UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Candidate disease gene prediction: a deep learning approach

Matilde Baptista Neves de Carvalho

Mestrado em Informática

Dissertação orientada por:
Professora Doutora Márcia Cristina Afonso Barros

2024

Acknowledgments

I would like to express my gratitude to my advisor, Professor Doctor Marcia Barros, for her guidance, which was a huge instrumental factor in the success of my research and my academic development.

These acknowledgments would be incomplete without expressing my deep and immense gratitude, overflowing like the rising sun over the horizon, to my family – my parents, my sisters, Sisi, and Mikas, the true architects of my path, my light. I thank you for all the investment, encouragement, and support throughout this journey. Even in the most challenging moments, you have always been by my side, and words are not enough to convey the immense love and appreciation I feel for each of you. Thank you for everything, for staying by my side and believing in me. I sincerely appreciate all the sacrifices you made to contribute to the realization of my dreams, for putting my needs ahead of yours, for being the beacon in my foggy days, and for safeguarding my happiness. This journey is not just mine, it is ours. The support and dedication of each of you have made it possible to achieve this great milestone in my life. I am deeply grateful to have such an incredible family by my side and for helping me become the best version of myself. You are my strength, my safe harbor, and my love for you surpasses any meaning that the word 'love' can encompass. I love you! *Obrigada por fazerem tudo por mim!*

I would like to dedicate this thesis to my beloved grandparents, whose memories and influences continue to guide and inspire me every day, even after their departure. To my grandfather Vítor, whose wisdom illuminated my path and whose teachings were fundamental to my education and the passion I inherited for numbers. I know you would be immensely proud of my work, and your presence is felt in every achievement of my life and every page of this thesis. To my grandfather Carlos, who always spoke of me with pride that shone in his eyes. His passing, one day after I received the papers to submit this thesis, left an immense void, but also a legacy of love, support and unconditional investment in my education. The strength of your encouragement and the trust you placed in me are an integral part of this work. Your absence is deeply felt, but your legacy lives on intensely in every written line. Thank you for believing in me and for always speaking of me with such pride. To both of you, my eternal gratitude and longing. This work is a tribute to the values and lessons you taught me, and my way of honoring the memory and love you always gave me. *Obrigada por todo o amor!*

Thanks a million to my fabulous and beloved godmother, Betty! Your charm, guidance, and support have been the secret sauce of my life. Cheers to believing in me and being the amazing soul that you are. For boosting me up, academically and personally, and for investing in this masterpiece-in-progress. You're not just a godmother; you're my fairy god-awesome!

I want to express heartfelt gratitude to my friend Raquel for being my constant support on this journey through English and university life. Thank you for always believing in me, even when I doubted my abilities, and for helping me overcome one of the greatest challenges of my life. Your confidence in me, especially in the realm of research, and all the support and opportunities you've provided are truly invaluable. You've inspired me with your enthusiasm, passion, and commitment, showing me that making a difference in the academic world and people's lives is possible. Thank you from the bottom of my heart, and thanks for being a great friend!

A massive thank you to my friends Inês and Sandra for the incredible emotional journey and the wisdom they've shared with me. I feel truly blessed to have such special friends by my side. Inês, biology has become a fun adventure with you. Thank you for your patience, explanations, and the amazing articles you've shared. I appreciate you being my friend through thick and thin, bringing so much positivity and joy into my life! Sandra, thank you for helping keep calm and guiding the way forward. I am deeply grateful to have such dedicated and fun people in my life.

A huge shoutout to Pedro and Eduardo, the humor engineers and masters of friendship! For all the notes, the laughter in tough times, and the top-notch friendship over the years! They say true friends are made in college, but ours began way before, and, behold, it became one of the best things in my life. With memes, laughter, and lots of camaraderie, you guys are simply incredible. Thanks for being such great friends, for making days funnier, and for being the true pillars of my sanity (or lack thereof). Here's to us, the masters of joy and epic friendship!

To Paulo, I want to dedicate a special note, as our friendship is one of the greatest blessings in my life. Thank you for all the unwavering support, even in moments when I contemplated giving up. I appreciate everything you do for me, the investments in my education, expanding my horizons, to helping me develop skills and knowledge that will be invaluable for my professional and personal growth, the priceless advices, and for being an unique and special presence in my life. My debt to you is truly immeasurable. Thank you from the bottom of my heart.

I also want to give a big shout-out to João and Marta for their incredible patience, emotional support, and friendship during this crucial time. I can't thank you enough for all the help and invaluable friendship you've given me. Without you, this journey would have been far more challenging. To João, for guiding me on the right path, and to Marta, for always bringing laughter into my life. I owe you both my eternal gratitude.

Sob o manto da meia-luz dos meus pensamentos, onde sussurros se entrelaçam em melodias no silêncio da minha essência, agradeço aos que dançam na quietude, guardiões de mistérios que desenham caminhos invisíveis nesta jornada. Entre os véus que separam realidade e transcendência, a vossa presença é uma teia invisível que entrelaça o meu destino.

A vós, mensageiros, que me recordam que o Universo reserva grandiosas batalhas aos destemidos, onde cada desafio é uma trilha de crescimento nas sombras do oculto. Aceito com honra os desafios que apresentam, agradeço pela bússola que me oferecem, revelando o Norte a seguir. E assim prossigo, sob uma imensidão de luz que desvenda os seus segredos apenas para mim, como o vento que dança de forma invisível aos olhos alheios, acariciando toda a minha alma, enquanto, em cada eco melódico das mensagens proferidas, a vossa presença é anunciada como o cântico dos sinos aos meus ouvidos.

A gratidão que carrego junto a mim transcende as fronteiras do entendimento, pois é nos enigmas sussurrados que descubro a essência verdadeira do rumo que a vida me propõe, como páginas reveladoras de um antigo pergaminho, cujos segredos ecoam através das eras, como versos enigmáticos de uma sagrada escritura tecelada nas páginas do tempo.

Aos ferreiros das grandiosas batalhas, moldadores da guerreira que sou, expresso o meu profundo agradecimento. Cada desafio foi uma forja que esculpiu resiliência, coragem e a afiada lâmina que empunho. Nas tempestades desafiadas, na dança eterna entre luz e sombra, avanço, guiada pelas vossas mãos hábeis, como segredos entrelaçados nos mapas sagrados de destemidos exploradores do desconhecido.

Obrigada pela luz que emana do poder da chama que está dentro de mim.

Muito obrigada!

Aos que partiram, aos que cá estão e aos que estarão por vir.

Abstract

The recommendation of candidate genes plays a crucial role in genetic research and medicine, given the vast and complex nature of the human genome, which harbors a multitude of genes potentially associated with various diseases. In this context, recommendation systems assume particular relevance by streamlining the identification process, acting as tools that suggest specific candidate genes related to a particular disease. Consequently, this approach expedites the phases of scientific investigation and discovery.

Additionally, in the current landscape marked by the information age and big data, the implementation of recommendation systems enhances efficiency in genetic studies. Given the continuous accumulation of genetic information at an unprecedented rate, these systems assist researchers in navigating and prioritizing potential candidates, optimizing the use of time and resources. Healthcare professionals can also, with the aid of these systems, personalize treatments based on individual genetic composition, leading to more tailored and effective medical interventions and therapies.

Given the intrinsic relevance of the subject, this study proposes a comprehensive approach to address contemporary challenges in genomic research. The creation of the recommendation system named `RecSysModel` in PyTorch stands out, with its architecture inspired by Neuronal Collaborative Filtering, aiming to recommend candidate genes for diseases listed in the Disease Ontology based on the level of scientific evidence in the literature. Data is imported directly from the SQLite database, designated `DiseaseGene`, into the system. It was designed to map diseases in the Disease Ontology, aligning with those recorded in the DisGeNET database, to contain detailed information about genes, diseases present in the ontology, and interactions between them, providing a robust approach for the analysis and recommendation of genes associated with these conditions.

It is crucial to highlight that the approach adopted for the proposed model stands out for its innovation, as it was not identified during the state-of-the-art review. This absence underscores the unique and original contribution of this methodology, filling a gap in the scientific literature, emphasizing the singularity of the approach in this study, and the innovation presented to the scientific community in this research field.

Keywords: Recommender system, Recommending candidate genes, Neural Collaborative Filtering, Disease Ontology, Pytorch

Resumo

O século atual marcou uma era de avanços tecnológicos significativos, impulsionados pela disseminação generalizada da Internet. Este fenómeno resultou numa explosão sem precedentes de dados e informações, muitas vezes referida como o ”petróleo do século”. Contudo, essa abundância de dados trouxe consigo um desafio crucial: como analisar eficientemente volumes massivos de informações para extrair dados relevantes e identificar padrões significativos.

Para superar esta problemática, surgiram os sistemas de recomendação, que utilizam algoritmos complexos para realizar análises sofisticadas de dados, respondendo à complexidade imposta pela abundância de informações disponíveis. A nível global, estes sistemas representam uma transformação paradigmática na abordagem e processamento de dados, permitindo uma exploração eficaz da riqueza de informações disponíveis para atender às necessidades individuais e fundamentar decisões informadas.

Em termos simples, os sistemas de recomendação convertem dados em sugestões personalizadas. No âmago da sua funcionalidade, empregam algoritmos avançados para analisar padrões comportamentais e preferências dos utilizadores, baseando-se nas suas interações passadas. Ao compreender os interesses individuais, esses sistemas são capazes de antecipar e sugerir conteúdos relevantes, produtos ou serviços. Dessa forma, a personalização inerente proporcionada pelos sistemas de recomendação não apenas melhora a experiência do utilizador, mas também otimiza os resultados para as empresas, aumentando a probabilidade de conversões e a fidelização do cliente. Essa capacidade de adaptação a preferências individuais torna esses sistemas ferramentas valiosas na era da informação, moldando a forma como interagimos com dados e conteúdos online.

Deste modo, a abrangência do conceito dos sistemas de recomendação transcende setores específicos, encontrando aplicação na resolução de desafios multifacetados, desde o comércio eletrónico até plataformas de música, vídeos e filmes, redes sociais, publicidade online, sites de notícias, saúde e investigação. No contexto da investigação biomédica, os sistemas de recomendação destacam-se como instrumentos essenciais, capazes de recomendar genes candidatos, desempenhando um papel indispensável na identificação eficiente de genes associados a condições específicas. Esta prática enfrenta os desafios intrínsecos à complexidade dos dados genómicos, essenciais para o progresso e a medicina personalizada.

No âmbito da medicina personalizada, as recomendações de genes candidatos desempenham um papel crucial, na medida em que uma compreensão abrangente da variabilidade genética entre indivíduos é imperativa para o desenvolvimento mais preciso de tratamentos. Os sistemas de

recomendação têm por isso, a capacidade de identificar marcadores genéticos relevantes para a resposta de um indivíduo aos medicamentos, permitindo o desenvolvimento de abordagens terapêuticas personalizadas que melhoram a sua eficácia e minimizam os efeitos adversos.

A identificação eficiente de genes candidatos revela-se crucial para a compreensão das causas genéticas de doenças complexas, sobretudo aquelas influenciadas por múltiplos fatores. Assim, os sistemas de recomendação agilizam a filtragem de vastos conjuntos de dados genômicos, identificando genes que requerem investigação mais aprofundada. A descoberta destes genes não só amplia o conhecimento sobre os mecanismos subjacentes, como também possibilita o desenvolvimento de intervenções terapêuticas direcionadas e novos fármacos.

Além disso, a sugestão de potenciais genes candidatos aprimora a eficácia das investigações biomédicas. Ao aproveitar o extenso volume de dados genômicos disponíveis, esses sistemas automatizados facilitam triagens eficientes e direcionadas, resultando em economia de tempo e recursos. Num cenário em que a pesquisa biomédica é dinâmica e exige respostas imediatas para avançar na compreensão de doenças, isso é de extrema importância. Assim, a recomendação de genes candidatos é um componente crítico da investigação biomédica contemporânea.

Dada a intrínseca relevância do tema, o presente estudo propõe uma abordagem abrangente para enfrentar os desafios contemporâneos na pesquisa genômica. Destaca-se a criação do sistema de recomendação denominado `RecSysModel` em PyTorch, cuja arquitetura é inspirada na abordagem de Filtragem Colaborativa Neuronal, e visa recomendar genes candidatos a doenças constantes na Ontologia de Doenças, com base no nível de evidência científica existente na literatura.

Devido à carência de conjuntos de dados estruturados que estabeleçam relações entre as doenças da Ontologia de Doenças e os genes, tornou-se imperativo criar uma base de dados em SQLite denominada `DiseaseGene`. Esta base de dados foi projetada para mapear as doenças na Ontologia de Doenças, alinhando-se com aquelas registradas na base de dados DisGeNET. Dessa maneira, a base de dados `DiseaseGene` contém informações detalhadas sobre genes, patologias e as interações entre ambos, proporcionando uma abordagem robusta para a análise e recomendação de genes associados a essas condições.

Diante da falta de correspondência direta entre os identificadores da Ontologia de Doenças (DOID) e as doenças presentes no DisGeNET, foi realizado um mapeamento para garantir uma integração harmônica das abrangentes relações entre genes e doenças. Este procedimento visa fortalecer a coesão e a utilidade do sistema desenvolvido, assegurando uma integração eficaz das informações provenientes de ambas as fontes de dados.

Adicionalmente, foi desenvolvido um esquema da base de dados para facilitar a compreensão e conduzida uma breve análise. O conjunto de dados abordado por este estudo inclui um total de 7101 identificadores DOID únicos, cobrindo 9929 doenças, 20024 genes e 541102 associações entre essas doenças e genes. Esta abordagem robusta proporciona uma base sólida para a análise e recomendação de genes associados a condições específicas, fortalecendo a integridade e utilidade do sistema desenvolvido.

Os dados são incorporados de forma direta do banco de dados SQLite para o modelo, visando uma integração eficiente da informação. Essa prática espelha um ambiente operacional real, simulando meticulosamente todo o fluxo de processamento antes de ser empregado como entrada no modelo para a geração de recomendações personalizadas. A importação direta de dados da base para o modelo representa uma prática eficaz nas operações, proporcionando benefícios notáveis. Isso simplifica e agiliza o processo de recomendação, eliminando etapas intermediárias e otimizando a eficiência. Além disso, essa abordagem reduz potenciais fontes de erro ou discrepâncias nos dados, garantindo uma representação mais precisa e atualizada das informações do banco de dados.

É fundamental destacar que a abordagem adotada para o modelo proposto, sobressai pela sua inovação, uma vez que não foi identificada durante a revisão do estado da arte. Esta ausência enfatiza a contribuição única e original desta metodologia, preenchendo uma lacuna na literatura científica, o que realça a singularidade da abordagem adotada neste estudo e a inovação apresentada à comunidade científica neste campo de pesquisa.

Concluindo este esforço, a fusão da extração de informações sobre doenças, construção de bancos de dados e desenvolvimento do sistema de recomendação estabeleceu uma base sólida para o avanço da compreensão das associações gene-doença. O modelo proposto não apenas produz previsões precisas, mas também serve como uma ferramenta versátil com aplicações potenciais em diversos cenários médicos e de pesquisa.

No dinâmico cenário da informática biomédica, este trabalho contribui não apenas para o aprimoramento dos sistemas de recomendação, mas também para o escopo mais amplo da descoberta e aplicação do conhecimento no domínio da genética e das doenças. A jornada empreendida destaca a importância do tratamento meticuloso dos dados, do desenvolvimento de modelos inovadores e da otimização contínua, refletindo a natureza dinâmica da investigação científica. À medida que avançamos, os resultados obtidos com este trabalho pavimentam o caminho para uma exploração mais aprofundada, colaboração contínua e avanços na integração do conhecimento no campo.

As principais contribuições desta dissertação são:

- Base de dados:
  - Criação da base de dados `DiseaseGene` em SQLite;
  - Contém informações sobre 7101 doenças da Ontologia de Doenças, abrangendo 9929 doenças identificadas por Sistema Unificado de Linguagem Médica (UMLS), 20024 genes e 541102 associações entre doenças e genes.
- Modelo:
  - Desenvolvimento do modelo denominado `RecSysModel`.
  - Objetivo: Recomendar genes candidatos com base no nível de evidência científica.
  - Utilização do PyTorch.
  - Arquitetura semelhante ao método Filtragem Colaborativa Neuronal.

  - Implementação inovadora, não identificada durante a revisão do estado da arte, representando uma contribuição significativa à comunidade científica.

- Técnica de Partição de Dados:
  - Desenvolvimento de uma técnica de partição de conjunto de dados original para superar desafios de algumas bibliotecas.
  - Garante a presença de todas as doenças no conjunto de treino, evitando a eliminação de pares únicos de doença e gene.

- Integração Eficiente:
  - Incorporação direta de dados do banco de dados SQLite para o modelo, garantindo uma integração eficiente da informação.

- Documentação e Exemplos Práticos:
  - Fornecimento de exemplos de código e uma explicação concisa da abordagem matemática utilizada.
  - Análise de recomendações para grupos de doenças, doenças raras e doenças classificadas como fenótipos.

- Código-fonte no GitHub:
  - O código-fonte do projeto está disponível neste repositório do GitHub e os dados estão disponíves no Figshare. Esta iniciativa promove a transparência, permitindo que outros investigadores e profissionais examinem, contribuam e construam sobre o trabalho desenvolvido.

Palavras-chave: Sistema de recomendação, Recomendação de genes candidatos, Filtragem Colaborativa Neuronal, Ontologia de Doenças, PyTorch.

Contents

List of Figures

List of Tables

Listings

Acronyms

CPU Central Processing Unit

CSV Comma-separated values

DO Disease Ontology

GPU Graphics Processing Unit

MSE Mean Squared Error

RAM Random Access Memory

ReLU Rectified Linear Unit

RMSE Root Mean Square Error

UMLS Unified Medical Language System

1
Introduction

The onset of the current century witnessed a remarkable surge in technological advancements and the widespread adoption of the Internet. Consequently, this has led to an unprecedented proliferation of data and information accessible to users and individuals alike. Therefore, it can be argued that data serves as a valuable resource in the current era, driving advancements and facilitating unprecedented levels of innovation. In tandem with the advancements and developments in technology, there are significant challenges that emerge as a result of the proliferation of data. One such challenge pertains to the efficient and expeditious discovery of targeted information within the expansive digital realm.

The proliferation of information has presented as a notable challenge: the difficulty of effectively analyzing vast quantities of data to identify pertinent content that aligns with the user's profile while also considering their individual preferences, interests, and tastes. As the quantity of data, continues to grow. The task of discerning significant and pertinent information, has become akin to searching for a needle in a haystack. Therefore, to address this issue, the field of information systems and artificial intelligence has witnessed the urge and priority of recommendation systems as a viable solution.

In light of this challenge, there is a pressing need to develop methodologies and algorithms that possess the capability to unravel this intricate data enigma. The aforementioned requirement has consequently led to the development of recommendation systems as an innovative tool created to relieve the strain caused by the overwhelming amount of information generated every second. Recommendation systems use complex algorithms to analyze data and identify intricate patterns. These enable them to offer personalized content that aligns with the unique profiles, preferences, and inclinations of users.

In this sequency, the emergence of recommendation systems represents a significant paradigm

shift in our approach to navigating the extensive realm of information. Through the analysis of user behavior, preferences, and historical data, these systems not only enable the discovery of content, but also, enhance user engagement and satisfaction. The capacity to anticipate and to propose content that users may perceive as pertinent, not only expedites their digital interactions, but also enhances business operations by fostering customer loyalty and facilitating conversions.

Hence, these systems hold significant importance across several domains and exhibit extensive applicability, including:

- E-commerce platforms such as Amazon[1], eBay[2], and Alibaba[3] employ recommendation systems to propose product suggestions to customers, leveraging data on consumer purchasing history, product views, interactions, and preferences [81].

- The phenomenon of media streaming encompasses the provision of music, movies, and series through digital platforms, exemplified by popular services like Spotify[4], Netflix[5], and YouTube[6]. These platforms employ recommendation systems to offer personalized suggestions of songs, films, and TV shows that align with the individual preferences of global users [83].

- Social networks [104], including Facebook[7], Instagram[8], and TikTok[9], employ recommendation systems to present users with pertinent content within their news feeds, proposing connections with other individuals, and suggest groups or pages that are aligned with their corresponding interests.

- Online advertising involves the utilization of digital advertising platforms that employ recommendation systems to determine and display advertisements that are pertinent to users [83]. These ones are based on users' online activities, browsing history, and personal preferences.

- News websites and blogs, employ recommendation systems, to provide users with personalized suggestions for articles and contents that align with their own individual interests and also their reading habits [76].

- Recommendation systems are used on libraries and online bookstores, to provide suggestions for books, magazines, and other reading materials that align with the literary preferences of users.

[1] https://www.amazon.com/
[2] https://www.ebay.co.uk/
[3] https://www.alibaba.com/
[4] https://open.spotify.com/
[5] https://www.netflix.com/
[6] https://www.youtube.com/
[7] https://www.facebook.com/
[8] https://www.instagram.com/
[9] https://www.tiktok.com

- In the domains of health and scientific research [83], recommendation systems have proven to be a valuable tool for suggesting relevant genes, microbes, and scientific articles.

As previously mentioned, recommendation systems play as a crucial role on various domains, necessitating the development of improved algorithms that can surpass existing ones in terms of efficiency and performance criteria.

Among this, recommendation systems have become a prominent tool in navigating the complex landscape of data proliferation. Search engines, not only streamline the process of information retrieval, but also serve as a prime example of the mutually beneficial relationship between human engineering and technological progress. This partnership propels humans into an era where the abundance of data can be effectively utilized to cater to our personal requirements and facilitate informed decision-making.

1.1 Motivation

The implementation of recommendation systems in biomedical research is crucial for associating biological components with diseases, as it offers an effective method of managing the vast and growing volume of available data. Modern medicine, which concentrates, not only on the treatment of diseases, but also on the in-depth study of Protamines, Microbiomics[86], and Genetics, requires this background.

In the pursuit of accurate information regarding the DNA of pathogenic microbiological species, recommendation algorithms improve the efficiency of the research process. When confronted with a substantial volume of data in this field, these systems efficiently sort and present the most critical information in a systematic way, thereby optimizing the time of researchers.

Comprehension of the intricate connections that exist between biological entities and maladies may prove challenging. Sophisticated algorithms employed in recommendation systems empower them to identify connections that are imperceptible at initial examination, thereby unveiling patterns and insights that may serve as a guide for further research.

By customizing recommendation systems to suit the particular preferences and needs of researchers, a more focused approach can be achieved. This ensures that the recommendations are customized to the specific interests of each user, thereby increasing the pertinence of the content provided.

Recommendation systems accelerate scientific discovery through the automation of data processing and filtration. This liberates researchers from the burden of interpretive and practical considerations, facilitating more rapid advancements in the comprehension of microorganisms, proteins, and genes associated with diseases.

Recommendation systems can handle the flow of biological data, which is frequently disseminated across numerous sources and formats. The ability to cohesively incorporate data provides a more comprehensive understanding, which is essential for gaining a profound understanding of

the connections between diseases and biological entities.

In clinical settings, rapid identification of biological entities associated with diseases is critical. Healthcare professionals benefit from recommendation systems as they facilitate more precise diagnoses and encourage well-informed clinical judgments.

The dynamic nature of biological research necessitates continuous updating. Recommendation systems are an exceptional method for informing health professionals and researchers of the most recent findings, which contributes to the implementation of more effective and current practices.

The utilization of recommendation systems to establish connections between diseases and biological entities is, at its core, indispensable for enhancing research efficiency, advancing scientific knowledge, and facilitating a comprehensive understanding of contemporary medical fields including proteomics, microbiomics, and genetics. In this critical sector, these technologies are indispensable for managing complexity and addressing the ever-growing volume of information.

1.2 Goals

The candidate gene recommendation system is an essential component in biomedical research, serving as an indispensable instrument to efficiently identify genes that are linked to particular conditions. This practice is critical for a multitude of reasons that address the inherent difficulties of genomic data complexity and the imperative to progress in the fields of functional genomics and personalized medicine.

To begin with, it is critical to identify candidate genes in order to further comprehend the genetic causes of complex diseases. Numerous medical conditions, particularly those with multiple contributing factors, necessitate a comprehensive strategy that incorporates genomic data in order to ascertain the implicated genes. In this regard, recommendation systems are indispensable, as they facilitate the efficient filtering of vast quantities of genomic data in order to identify genes that merit additional investigation.

Additionally, the identification of potential therapeutic targets is facilitated by the recommendation of candidate genes. The discovery of disease-associated genes, not only expands comprehension of the underlying mechanisms, but also facilitates the creation of targeted therapeutic interventions. These suggestions have the potential to draw attention to particular genes that are crucial in biological pathways associated with diseases, thereby offering significant knowledge for the advancement of novel pharmaceuticals and more efficient treatment strategies.

Personalized medicine is an additional domain in which candidate gene recommendations are crucial. A holistic comprehension of genetic variability among individuals is imperative in order to more precisely develop treatments. Genetic markers that are pertinent to an individual's response to medications can be identified by recommendation systems. This capability empowers the development of personalized therapeutic approaches, which not only enhance efficacy but also mitigate adverse effects.

Furthermore, the suggestion of potential genes enhances the efficacy of biomedical investigations. By leveraging the extensive volume of genomic data at hand, these automated systems

facilitate efficient and focused screening, resulting in time and resource savings. In a scenario where biomedical research is dynamic and necessitates immediate responses to advance disease comprehension, this is of the utmost importance.

In conclusion, candidate gene recommendation is a critical component of contemporary biomedical research. Through the optimization of gene identification associated with particular conditions, these systems not only enhance comprehension of the genetic foundations of diseases but also facilitate the development of personalized therapeutic approaches, distinguishing themselves as an essential instrument in the scientific repertoire that contributes to the pursuit of more precise and efficacious medicine.

Given the relevance of the topic and its importance today, ***the goal of this project is to develop a recommender system for recommending candidate genes to diseases.***

Main objectives:

1. Creation of a major standard dataset with the interactions between the diseases in the Disease Ontology (DO) and the known genes. The objective is to create a $\langle disease,\ gene,\ interaction,\ (other features)\rangle$ dataset.
2. Develop a recommendation system algorithm capable of recommending candidate genes using the generated dataset.

1.3 Structure of the document

This document is organised as follows:

- **Chapter 2**, provides an introduction to recommendation systems, including an examination of how to address the initial problem, feedback types, a comprehensive delineation of feasible approaches throughout the recommendation system's development process, methods for evaluating the system and encountered challenges.
- **Chapter 3**, presents a comprehensive examination of the scientific literature, detailing the advancements made thus far and the endeavors undertaken in recommending potential candidate genes.
- **Chapter 4**, complete overview of the procedure that culminates on the formulation of the definitive set of data, including fundamental sources, data extraction methods, and their processing. In addition, a summed up examination of the data is provided.
- **Chapter 5**, delves into the implementation and environment factors influencing the model. It thoroughly explores the input data preparation, emphasizing the division into training, validation, and test sets while elucidating procedures ensuring data quality and integrity. Investigating a Neural Collaborative Filtering model in PyTorch with the Torch library, it describes the structure, features, and crucial aspects, supported by code snippets and simplified mathematical explanations.

- **Chapter 6** scrutinizes performance metrics to assess the model's effectiveness, accuracy, and resilience, culminating in a meticulous evaluation of outcomes. Additionally, it compares the proposed model with an existing one.
- **Chapter 7**, concludes the written work with a summary of contributions, areas for improvement, and potential future changes.

2
Recommender Systems

The fundamental concept of recommender systems is to offer users personalized recommendations or suggestions for items, utilizing previously collected data to discern their interests. A recommendation system encompasses the dynamic interplay between users and items, aiming to exert an influence on subsequent recommendations. This tool should be considered as an integral component for various industries [10, 11, 29, 66], particularly those associated with consumer behavior, as it influences various aspects such as modes of thought, the development of new preferences, and the generation of consumer demands through media collaborations. This particular system, operates based on a shared income, while encountering challenges related to varying economic and social strata.

In this context, there are two important roles to retain:

User: Referring to the entity to which recommendations are directed (it doesn't have to be a person).

Item: Referring to the product or item that has been viewed or is being recommended.

There are two potential approaches to formulating a recommendation system [8]:

1. **The predictive variant of the problem:** The primary aim of this approach is to forecast the rating or preference value of a user for a particular combination of user and item. Specifically, it seeks to determine the likelihood that *User 1 will like item Y* given their positive response to *item X*. In order to implement this methodology, it is necessary for the training set to contain vectors comprising the $\langle user, item, rating \rangle$. The primary difficulty lies in generating precise forecasts for the unobserved data in the test set, specifically for the items that the user has not yet engaged with or evaluated. Therefore, the test set includes vectors $\langle user,\ item,\ rating\ = ? \rangle$.

2. **The ranking version of the problem:** This is a variant that involves assigning a relative order or position to a set of items based on certain criteria or attributes. The approach to formulating this system, is commonly referred to as the *top-k problem*. The primary aim is to provide a recommendation of the *top-k* items (or users) that are most pertinent to a specific user (or item). This focus is on identifying relevance rather than accurately predicting the rating that a user assigns to a particular item. As a result, the precise numerical values of the anticipated ratings are not of significant importance. Consequently, the focus lies in suggesting the most pertinent k items, taking into consideration the user's probability of preference. Instead of assigning a discrete numerical classification to each item, the system arranges the items in a ranked order based on their relevance to the user. The *top-k* items in this ordered list are then presented as recommendations. The process bears resemblance to the *top-k* users.

2.1 Feedback

Recommender systems endeavor to predict whether a given user or a group of users will evince interest in a particular product or item or even when is generated a list with the most *top-k* relevante items. The identification of discernible patterns in this regard necessitates the acquisition of feedback, a process that bifurcates into explicit and implicit feedback mechanisms.

So that, explicit feedback refers to information, that is provided by users in a clear and straightforward manner. This phenomenon can be observed on the form of quantifiable ratings, on a numerical scale, or by seeking user feedback on particular products. Besides that, implicit feedback refers to those that are derived from the actions and behaviors of users. When a user interacts with a news article by clicking on it, it can be inferred that he or she may have a potential interest in accessing articles that are similar in nature to the one clicked on. It can also encompass various factors, including, but not limited to view counts, the amount of time spent on viewing a video, and other relevant metrics [8]. Thus, each of these feedback categories has unique merits and demerits.

2.1.1 Explicit Feedback

Explicit feedback refers to the process of soliciting user input through the use of a predetermined scale, where in users are requested to quantitatively express their level of satisfaction with a particular item [80]. For instance, *"What rating would you assign to this film on a scale ranging from* 1 *to* 5*?"*. In this scenario, the numerical value of 1 denotes the least favorable rating, indicating that the user did not have a positive opinion of the film. Conversely, a rating of 5 signifies that the user expressed a favorable view of the film.

There are several advantages and challenges to consider.

Advantages: Explicit feedback offers clear and unambiguous insights into the prefer-

ences of users, frequently by means of quantifiable metrics such as numerical ratings. This determines whether a user evaluates a specific item or product positively or negatively.

Challenges: One of the primary difficulties related to explicit feedback is its intermittent lack of accessibility. It is not always the case that users consistently offer explicit feedback, resulting in gaps in the available data and desired insights.

2.1.2 Implicit Feedback

In the context of implicit feedback, it is not possible to ascertain the user's preference towards a specific product with direct certainty. The extent of user interaction with the product is ascertainable [70]. For example, a user who has engaged with news articles.

There are several advantages and challenges to consider.

Advantages: Implicit feedback is obtained indirectly through user behavior, eliminating the need for direct forms of interaction such as questionnaires or ratings. This approach circumvents the necessity for increased user involvement.

Challenges: The interpretation of implicit feedback poses a significant challenge. Ambiguities may arise when individuals engage with content, such as reading an article, as this engagement does not necessarily imply a clear endorsement or appreciation of the content in question.

The selection between explicit and implicit feedback, in the domain of recommender systems, is contingent upon the accessibility of data and the particular application context. While explicit feedback is known for its clarity, implicit feedback provides a more subtle yet occasionally complex perspective through which user preferences can be inferred. Therefore, the selection of the most suitable feedback modality is contingent upon striking a balance between these benefits and obstacles in the endeavor to construct resilient and efficient recommendation algorithms.

2.2 Recommender systems approaches

Once the problem has been formulated, specifically determining whether it pertains to the problem of perdition or the *top-k* problem, it becomes crucial to justify the selection of appropriate approaches for implementing the recommendation system. In this context, there are various methodologies that facilitate the development of a recommendation system, namely collaborative filtering, content-based filtering, and hybrid approaches, illustrated in Figure 2.1.

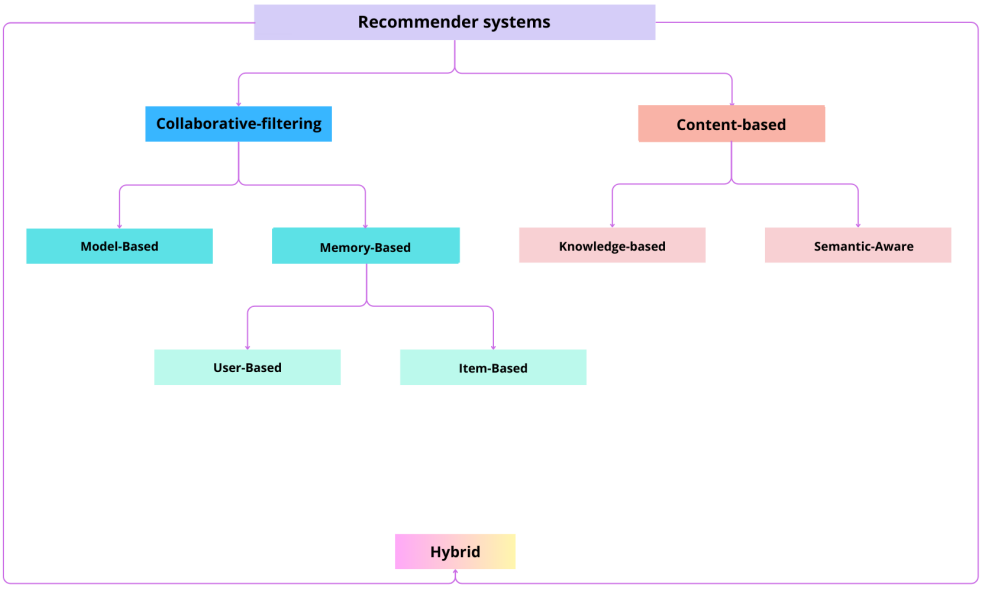


Figure 2.1: Recommender Systems main approaches.

Figure 2.2 illustrates the primary difference between these two methodologies, collaborative filtering and content-based. In the context of collaborative filtering recommender systems, it is seen that both *User 1*, and *User 2*, have engaged with common articles or books. Therefore, these consumers exhibit similarities in their preferences. *User 1*, proceeded to peruse a subsequent academic publication or book, which is poised to be suggested to *User 2*. In the context of content-based techniques, *User 1* has engaged with a certain article or book. Consequently, a second article or book, that bears similarity to the one *User 1*, has already read its identified and then suggested to *User 1*. Content-based techniques need the availability of a comprehensive set of functionalities. In some domains, like the realm of film or literature, these characteristics may be readily delineated. In the context of movies, the attributes that might be considered, include the director, genre, and actors. In other disciplines, the process of determining the appropriate traits is less straightforward.

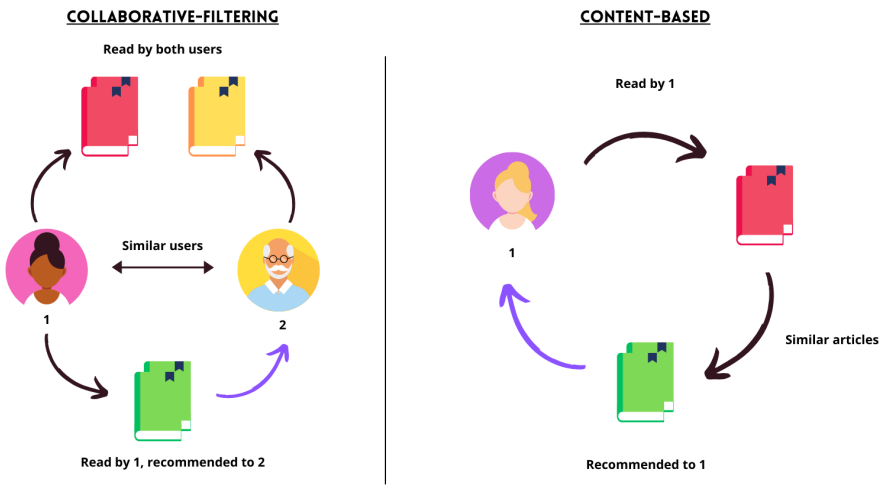


Figure 2.2: Collaborative-filtering vs content-based.

2.2.1 Collaborative Filtering

The initial instance of a collaborative filtering system can be traced back to the year 1992, as documented in Goldberg et al.'s [44] seminal work. This pioneering system, known as Tapestry, marked the advent of collaborative filtering technology. The system under discussion was developed by researchers affiliated with Xerox PARC (Palo Alto Research Center Incorporated) and its primary objective was to leverage the concept of user collaboration in order to provide content recommendations tailored to individual preferences and interaction history. Specifically, the system utilized feedback from users regarding previously read emails to suggest relevant content to other users who might share similar interests. The emergence of the concept of collaborative filtering can be attributed to an approach in this recommendation systems. This approach involves the collection and analysis of data pertaining to user preferences. Otherwise, leveraging the similarity between users' past interests, it aims to predict their current interests and recommend relevant items (Figure 2.2). Therefore, there are several strategies for implementing the memory-based and model-based collaborative filtering techniques (Figure 2.1) [49].

2.2.1.1 Memory-Based

Memory-based collaborative filtering is a foundational technique used in recommender systems, which involves the examination of users' historical interactions with items or the inherent connections among items. This technique is commonly employed to provide recommendations to users by leveraging the concept of similarity, either between users or between the items being considered. One of the primary components of memory-based collaborative filtering entails the utilization of neighborhood-based, wherein the direct associations between users and items are

taken into account. Hence, the underlying assumption proposes that individuals who share comparable patterns of interaction are likely to exhibit similar preferences towards items that have not been observed. There are two primary methods for achieving this objective: User-based collaborative filtering and Item-based collaborative filtering (Figure 2.1) [36].

1. **User-Based Collaborative Filtering (UBCF):** User-Based Collaborative Filtering is a methodology that aims to identify users who exhibit similar behavioral patterns to the target user, often referred to as *'User A'*. This is achieved by computing the similarity between *User A* and other users in a rating matrix, which typically contains user-item ratings. Similarity calculations can be performed using metrics such as the Pearson correlation coefficient (Equation 2.2) or the Cosine similarity coefficient (Equation 2.1), measuring linear correlation or directional similarity between users' rating vectors, respectively. Once similarity is established, the system proceeds to suggest items that have received positive ratings from users who exhibit similar preferences, but have not yet been evaluated by *User A*. User-Based Collaborative Filtering aims to identify a cluster of users who share similarity with *User A* and makes recommendations based on the preferences and choices of this user cluster, providing personalized recommendations.

2. **Item-Based Collaborative Filtering (IBCF):** approach prioritizes the identification of items that exhibit the highest similarity to the target *item B*, using the ratings provided by users to lead them on basis for comparison. The initial stage involves the creation of a set of items, denoted as *S*, which exhibit the highest degree of similarity to *item B*. Subsequently, the evaluations furnished by users pertaining to the elements encompassed within set *S* are employed to forecast *User A*'s affinity towards *item B*. The primary objective of Item-based collaborative filtering is to identify items that exhibit similarity to the ones prefered by the current *User*, relying on past reviews as a basis [30].

As previously stated, the Cosine similarity (Equation 2.1) and Pearson correlation coefficient (Equation 2.2) are the prevailing metrics utilized to compute the similarity between the rows or columns of a matrix, depending on the context (whether it pertains to users or items). The concept of Cosine similarity pertains to the quantification of similarity between two vectors within a vector space, wherein it assesses the cosine value of the angle formed between said vectors [98]. The following formula can be used to express the aforementioned concept:

$$Cosine\ Similarity(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \tag{2.1}$$

Where:

x and y are two non-null vectors.

The Cosine similarity metric spans from -1, representing complete dissimilarity, to 1, indicating perfect similarity, while a value of 0, signifies no similarity. In specific applications, the variable is constrained to the interval $[0, 1]$, contingent upon the context in which all values are positive.

The Pearson correlation coefficient [84] (Equation 2.2) is a statistical measure used to assess the strength and direction of the linear relationship between two sets of data. The aforementioned method is employed with the purpose of calculating the level of correlation between the ratings provided by two users (or items). The correlation coefficient is a mathematical measure which quantifies the relationship between two variables. It is obtained by dividing the covariance of the variables by the product of their standard deviations. Thus, it provides a standardized measure of covariance, with values ranging from -1 (indicating a perfect negative correlation) to 1 (indicating a perfect positive correlation), and a value of 0 indicating no linear correlation between the variables. Similar to covariance, the measure solely captures a linear correlation between variables, disregarding various other forms of relationships or correlations. The following formula can be used to express the aforementioned concept:

$$Pearson\ correlation(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.2}$$

Where:

x_i and y_i are the individual sample points indexed with i.

$\bar{x}$ and $\bar{y}$ are the sample mean.

n is the sample size.

It is imperative to acknowledge that in spite of the memory-based collaborative filtering exhibits intuitive and effective characteristics in numerous scenarios. It also possesses certain limitations, which include its sensitivity to alterations in user behavior and challenges that are associated with sparsity in extensive datasets. Therefore, in order to enhance the quality of recommendations in recommender systems, model-based techniques such as Singular Value Decomposition (SVD) or machine learning algorithms, are frequently employed.

2.2.1.2 Model-based

Model-based approaches in recommender systems offer an alternative methodology to memory-based collaborative filtering. This is a method that relies on the direct utilization of users' past interactions with items. In contrast, model-based collaborative filtering involves the construction of statistical or mathematical models that describe the underlying relationships between users and items [101]. There exist various techniques and approaches for the implementation of model-based methods in recommender systems, which are outlined as follows [6]:

1. **Decision Trees:** Decision trees are computational models that depict decision-making processes in a hierarchical tree structure [61]. In the realm of collaborative filtering, models can be employed to facilitate the process of assigning ratings to items by leveraging user and item attributes. One potential approach that can be employed, is the utilization of the Random Forest algorithm. They are a type of ensemble learning method that consist of multiple decision trees, which can be effectively employed in collaborative filtering tasks. Every individual tree within the forest possesses the capability to forecast user-item ratings or preferences by taking into account the attributes of both the user and the item.

2. **Rule-Based Models:** Rule-based models are constructed using logical or heuristic rules as their foundation. Sets of rules can be employed to establish recommendations by analyzing patterns of user behavior. The utilization of Association Rule Mining can be employed with this approach [79], such as Apriori or FP-growth, which are commonly utilized to uncover patterns in user behavior. For instance, the identification of patterns such as *"If a consumer purchases product A, there is a high probability of them also purchasing product B"* and leveraging these patterns to generate recommendations.

3. **Bayesian Methods:** The Bayesian methodology integrates probabilities and Bayesian theory to construct models that represent the user's preferences and the likelihood of favoring or disfavoring a particular item. This enables the development of probabilistic models for generating recommendations. One potential strategy in this particular scenario entails the utilization of Bayesian Personalized Ranking (BPR) [77], which is a probabilistic approach employed in matrix factorization models to facilitate recommendation systems that pairwise ranking task is the problem. The objective is to optimize the ranking of items based on user-item interactions.

4. **Latent Factor Models:** Latent factor models represent a robust methodology employed in recommendation systems. The latent factor space is utilized to represent both users and items, wherein the similarity between users and items are computed by considering their respective latent representations. One example of a technique that falls into this category, is Matrix Factorization (MF) [18] (Equation 2.3), such as Singular Value Decomposition (SVD) [90] and Alternating Least Squares (ALS) [51], are utilized to decompose the user-item interaction matrix into latent vectors representing users and items. The latent vectors in question, serve the purpose of capturing underlying patterns and are subsequently employed in the context of recommendation systems. Additionally, Logistic Matrix Factorization (LMF) is a notable technique that leverages logistic functions to model user-item interactions and it extends the concepts of matrix factorization to handle binary user-item interactions, making it suitable for scenarios where explicit ratings may not be available [56]. Finally,

the Neural Collaborative Filtering (NCF) [47] method, employs neural networks to directly glean insights into user-item interactions from available data. This novel methodology integrates embeddings of users and items into the network, which enables the discovery of complex patterns and interactions that exceed the capabilities of conventional matrix factorization techniques. Significantly, Neural Collaborative Filtering demonstrates the capacity to incorporate numerous layers and activation functions, which enables it to model intricate connections between users and objects with efficacy. Through the utilization of these neural network functionalities, it introduces a robust framework for recommendation systems that is able to comprehend intricate and non-linear connections within user behavior.

$$R = U \cdot V^T \tag{2.3}$$

Where:

$R \in \mathbb{R}^{users \times items}$ is the user-item rating matrix.

$U \in \mathbb{R}^{users \times latent\ factors}$ contains the user's latent factors.

$V \in \mathbb{R}^{items \times latent\ factors}$ contains the item's latent factors.

Resuming, model-based methods utilize mathematical models that have been trained using historical item ratings in order to make predictions about user preferences for items that have not been rated. These models are able to capture intricate patterns of user behavior and consequently, several hybrid models integrate multiple techniques, and the selection of the model or neural network, is contingent upon the specific requirements, data availability, and scalability of the recommendation system. Model-based recommender systems presents an alternative approach to memory-based collaborative filtering, employing statistical and mathematical techniques to deliver recommendations that are tailored and efficient. Some examples of models commonly used in data analysis and machine learning include decision trees, rule-based models, bayesian methods, and latent factor models. One of the primary challenges associated with the collaborative filtering approach pertains to the cold start problem, which encompasses the difficulties encountered when dealing with new items, new users, and data sparsity. The phenomenon known as "cold start" occurs when a user has not provided ratings for any items in the dataset, or when a new item has not yet received ratings from any users. Section 2.4 discusses and examines the aforementioned challenges.

2.2.2 Content-based

Content-based recommendation systems are a fundamental methodology for the development of recommendation systems. These methods distinguish themselves from the collaborative filtering technique by not relying exclusively on user ratings to generate high-quality recommendations. However, these systems rely on the descriptive attributes of the items and the development of user

profiles. Hence, the objective of the content-based approach is to identify items that exhibit the highest degree of similarity to those that users have previously encountered (Figure 2.2). This session will explore the core ideas and unique characteristics of content-based recommendation systems [3, 74].

1. **Item Descriptive Characteristics:** The fundamental principle underlying content-based systems is centered around the utilization of the descriptive attributes of items. The variability of these characteristics is contingent upon the specific field of application. For instance, when providing movie recommendations, the relevant attributes may encompass the film's title, synopsis, genre, principal cast members, director, and other pertinent details. The aforementioned information is derived directly from the items themselves, without reliance on user ratings or reviews. This is particular important when dealing with scientific items, such as chemical compounds or genes, due the specificity of the fields.

2. **User Profile Creation:** In the context of content-based recommendation systems, individual users are assigned profiles that capture their interests and preferences. These profiles are constructed by analyzing their past interactions or by incorporating information explicitly provided by the user. The user profile is constructed based on the attributes of items that user has expressed preference for or engaged with in previous instances.

3. **User Profile Comparison with Items:** After obtaining the user profile and the characteristics of the items, the system proceeds comparing these two sets of information in order to generate recommendations. The process is accomplished through the utilization of algorithms that compute the degree of similarity between user's profile and attributes of the items.

4. **Relevance Score Calculation:** The degree of similarity observed between the user profile and the inherent characteristics of the items under consideration leads to the computation of a relevance score for each individual item. The relevance score increases as the item exhibits greater similarity to the user's profile and within receive the highest scores are suggested as recommendations to the user. Its calculation, between items and the user profile, can be achieved by utilizing either the Cosine similarity measure (Equation 2.1) or, in the case of structured data, Euclidean distance metric [74]. The following formula can be used to express the aforementioned concept of Euclidean distance between two vectors x and y:

$$Euclidean\ distance(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.4}$$

Where:

x_i and y_i are the individual sample points indexed with i.

n is the sample size.

In addition to these metrics, machine learning techniques, such as clustering, can be employed [9].

5. **Personalized Recommendations:** One of the primary benefits of content-based systems is their capacity to offer customized recommendations by considering the unique interests of individual users and the attributes of the items being recommended.

6. **Managing Novel Users:** One potential constraint of content-based systems, lies in the challenge of providing recommendations to new users who lack an established user profile. This limitation is primarily due to the absence of an interaction history necessary for the system, to generate reliable recommendations. In this context, while content-based systems do not face the problem of a ”cold start” for new items, a significant issue persists when dealing with new users [63].

7. **Use of Textual Data:** The utilization of textual data frequently assumes a pivotal part in the depiction of objects and the formation of user profiles. The utilization, of Natural Language Processing (NLP) methodologies is frequently employed for the purpose of extracting significant information from textual data, like movie synopses or product descriptions. For this reason, ontologies are considered one of the essential tools for content-based recommendation systems [92]. In the realm of knowledge representation, an ontology is a meticulously organized collection of terms and concepts that serves as a representation of our understanding of the universe[1]. Specifically, ontologies offer well curated vocabularies consisting of terms and their corresponding definitions, enabling the representation of entities within a specific domain of study [16, 32, 94].

In the realm of content-based recommendation systems, two prominent strategies for generating suggestions have emerged: the knowledge-based approach and the semantic-aware approach (Figure 2.1). Both methodologies share the common objective of enhancing the caliber of recommendations, while diverging in their respective strategies and information sources.

2.2.2.1 Knowledge-based

Knowledge-based recommendation systems integrate explicit knowledge regarding items and users into their recommendation algorithms [22, 93]. This may encompass comprehensive data regarding the characteristics of items, such as attributes, metadata, tags, expert ratings, and other sources of enriching information. It relies on identifying similarities between users' preferences

[1] `https://dicionario.priberam.org/ontologia`

and also the descriptions of items. The proposed methodology leverages the collected knowledge from products in order to create a more accurate mapping of users' preferences and needs, consequently enhancing the relevance of suggestions. As a result, this methodology demonstrates its efficacy in situations involving sporadically purchased products such as the real estate market, or items with intricate qualities such as cars and cell phones. Nevertheless, the execution of this type of system may present difficulties owing to the requirement of constructing and sustaining a comprehensive and current information repository. It is worth that Knowledge-Based approach does not encounter a "cold start" problem, but rather is constrained by the explicit knowledge provided by users, thus missing in terms of novelty [5].

2.2.2.2 Semantic-Aware

In contrast, Semantic-Aware systems utilize Natural Language Processing (NLP) techniques and semantic representations to augment the comprehension of items and users' preferences. This is so as elucidated in point 7 of Section 2.2.2. By using this kind of methodology, it aims to comprehend the inherent significance of words and phrases, enabling the system to gain a more comprehensive understanding of the context and semantic connections between items and users. It can be also achieved by leveraging domain-specific knowledge, such as ontology and Linked Data [31]. Within, it has the potential to yield recommendations that are more contextually relevant and precise, particularly in situations when there is a lack of explicit knowledge about the products.

In conclusion, content-based recommender systems represent a robust methodology for delivering customized suggestions to users, especially in situations when comprehensive item data is accessible. These systems demonstrate the capability to operate well even in scenarios with limited user collaboration. They offer a particular value in domains where descriptive attributes significantly influence user decision-making, such as movies, music, products, and other relevant areas.

2.2.3 Hybrid

Recommendation systems play a vital role in many modern applications, helping users discovering items of interest in an immeasurable sea of information. Two primary methodologies have been extensively employed into the development of these systems, namely collaborative filtering and its content-based. Nevertheless, it is important to acknowledge that both of these methodologies possess inherent limitations that may have an impact on the quality of recommendations and the overall user experience.

The generation of recommendations in collaborative filtering, is heavily dependent on user ratings and interactions. This implies that in situations where there is limited user data or for newly introduced items, collaborative filtering may encounter challenges in delivering pertinent recommendations [2]. Furthermore, the system is susceptible to scalability issues, data sparsity, and the "cold start" predicament, which arises when a novel user or item is introduced to the system.

However, content-based systems encounter difficulties when it comes to managing the wide range of user preferences. The recommender system often suggests items that closely resemble those with whose the user has previously engaged, thereby constraining the exploration of novel interests. Furthermore, the efficacy of these systems may be diminished in cases where items are devoid of comprehensive descriptive information or when user preferences are not clearly articulated [2].

In order to overcome these constraints, researchers have developed hybrid recommendation systems. These systems aim to integrate the strengths of both collaborative filtering and content-based approaches [19]. There exist multiple methodologies for the development of hybrid systems.

Fusion of Results: It's a technique that involves the generation of recommendations through the independent application of collaborative filtering and content-based filtering methods. Subsequently, the outcomes are amalgamated in a manner that yields conclusive recommendations. This can be achieved by assigning varying degrees of importance to the recommendations provided by each method, or alternatively, by integrating them into a unified approach [25, 73]. Equations 2.5 and 2.6 present several metrics, namely S_{CFI1} and S_{CBI1}. S_{CFI1} represents the score achieved for *Item 1* through the utilization of a collaborative-filtering algorithm, while S_{CBI1} denotes the score obtained for *Item 1* using a content-based algorithm.

Multi-layer Models: This kind of utilization models involves the implementation of distinct recommendation across various layers. Initially, a template is employed to generate an initial set of recommendations. Subsequently, an additional model is employed to enhance and optimize these recommendations by considering supplementary factors and contextual information.

Dynamic alternation: This refers to the capability of hybrid systems to flexibly switch between different methods based on the specific context or user requirements which enables the system to select the most suitable methodology for a given situation.

$$Metric\,1 = S_{CFI1} \times S_{CBI1} \tag{2.5}$$

$$Metric\,2 = \frac{S_{CFI1} + S_{CBI1}}{2} \tag{2.6}$$

Where:

S_{CFI1} represents the score achieved for *Item 1* using a collaborative-filtering algorithm.

S_{CBI1} denotes the score obtained for *Item 1* using a content-based algorithm.

Hybrid systems aim to address the inherent limitations of collaborative and content-based filtering approaches by integrating both methodologies. An illustrative instance involves the implementation of a hybrid system that combines collaborative filtering and content-based approaches. This hybridization effectively addresses the challenges associated with the cold start problem pertaining to new items, as well as the diversity problem. A hybrid approach combining, collaborative filtering and knowledge-based methods, has the potential to address the challenges associated with the cold start problem for both new items and also new users [4].

One potential avenue to improvement lies in enhancing the precision of recommendations, addressing the challenge of cold start scenarios, and offering a wider array of suggestions. Consequently, these systems have demonstrated significant efficacy across diverse scenarios, enhancing user experience and the utility of recommendations within recommendation systems. Hence, their contribution is vital in the ongoing advancement of this field of study and in providing recommendations of exceptional quality for practical implementations.

2.3 Evaluation methods

The assessment of a system holds significant importance, as it is crucial to guarantee the quality of the recommendations produced by the system. Adding the fact that, the primary purpose of recommendation systems is to provide suggestions, it is crucial that these recommendations are in accordance with the User's profile and cater to their individual tastes and preferences. As a result, it is imperative to exercise meticulous deliberation when selecting evaluation metrics, considering both the available resources and the specific objectives of the recommendation system under scrutiny [45, 65].

To provide an example, it's helpful to consider an extensively utilized video-sharing platform that utilizes a recommendation system, similar to YouTube[2] or TikTok[3]. In this context, the evaluation process can be carried out by means of online tests, wherein the efficacy of recommendations is assessed by analyzing genuine user interactions. One commonly employed method for evaluating online platforms, is the A/B Test methodology [14], which is typically implemented as follows: Users are randomly allocated into two groups in an arbitrary manner; *Group A* is assigned a particular recommendation approach, while *Group B* is assigned to another approach. One group receives suggestions based on the current algorithm, while the other group is exposed to a variation, such as an improved recommendation algorithm. The efficacy of these methodologies is evaluated through performance measures, such as the conversion rate, which signifies the ratio of users who engage in desired actions, such as subscribing to a content creator or interacting with a suggested video. Additionally, the click-through rate is employed to indicate the percentage of users who click on the presented recommendations. This approach provides valuable insights into the impact of different recommendation strategies, allowing for data-driven decisions to enhance the overall user experience.

[2] `https://www.youtube.com/`
[3] `https://www.tiktok.com/`

A/B utilization tests, enable this video-sharing platform to ascertain the efficacy of its recommendation strategies in attaining desired objectives on the platform, including user engagement, prolonged platform usage, new design features, and potential content dissemination. They perform an essential role in enhancing the recommendation algorithm and consistently improving the user experience, guaranteeing that recommendations are more pertinent and appealing to users.

Nevertheless, it is crucial to acknowledge that developers of recommender systems frequently encounter challenges when attempting to access online data for the purpose of evaluation. Thus, it is common practice to utilize offline data, predominantly comprising datasets that encompass details pertaining to users' historical preferences.

It is imperative to acknowledge that a significant drawback of the offline evaluation method is the absence of immediate access to a user's real-time content preferences. For instance, in the event that a user is seeking content pertaining to cooking in the morning, the recommendation system should possess the capability to curate a personalized 'For You Page' featuring a selection of cooking videos. Despite the fact that the user primarily engages with videos pertaining to book recommendations in the afternoon, the system should possess the same level of proficiency in suggesting content that revolves around books. The fluctuating nature of preferences over time presents a significant obstacle for offline assessment.

However, its utilization on evaluating recommendation systems, offers a notable benefit as it enables the testing and advancement of novel recommendation models without the immediate requirement of implementing an online platform, so that, gather real-time interactions could be disposable for users. In brief, offline evaluation offers a significant opportunity to assess and enhance the efficacy of recommender system models. If these novel models exhibit superior performance in comparison to established models, they can be deployed on online platforms with a high level of assurance.

When conducting an assessment of offline datasets, a crucial step is to partition the datasets into distinct subsets for training and testing purposes. These datasets are derived from past historical data. The model is trained using the training data so that it could be enable the system to discern patterns within the data. Subsequently, the trained model is employed to evaluate the quality of recommendations made by the system using the test data. Consequently, the process of dividing the data into distinct training and testing sets, enables the model to undergo training using a subset of the data, followed by an evaluation of its capacity to generate precise and pertinent recommendations for unseen data [82, 85]. Moreover, it is imperative to acknowledge that, under certain circumstances, the utilization of an additional dataset known as a validation set can prove advantageous. The utilization of the validation set is a common practice in order to fine-tune the hyperparameters of the model, that ultimately enhance the effectiveness of the recommendation system. This practice aids in mitigating the issue of overfitting the model to the training data, which can lead to suboptimal recommendations for novel users or items.

For dividing the data into training and testing sets, there are two widely used techniques. Firstly, the hold-out method involves partitioning the dataset into a specific percentage for training

$\alpha\%$, typically around 80%, while allocating the remaining portion $(1-\alpha)\%$, usually 20%, for testing. Secondly, the cross-validation method is employed by dividing the dataset into q equal sets. In each iteration of this process, $q-1$ sets are utilized for training, and 1 set is reserved for testing. This meticulous allocation of dataset subsets during each evaluation ensures a thorough examination of the entire dataset, effectively guarding against overfitting. Notably, this method eliminates the necessity for a dedicated validation set, as mentioned in reference [7].

Nevertheless, in specific scenarios where K-fold cross-validation is concurrently employed for the fine-tuning of hyperparameters and the estimation of errors, a validation set is introduced as outlined in reference [28]. This validation set serves a crucial role solely during the optimization of hyperparameters. Now, it is deemed crucial to shift focus toward the evaluation method that is particularly well-suited for sequence-aware recommender systems: the leave-one-out technique [27, 65]. In this approach, the final item in the sequence is temporarily concealed during the testing phase. The retention of this final item holds paramount significance since sequence-aware recommender systems are meticulously designed to forecast the subsequent optimal item within the sequence. Consequently, it seamlessly aligns with their overarching objective. Within the realm of validation, it is customary to conceal all items except one.

As stated earlier, there exist various approaches to constructing a recommendation system, and consequently, the assessment of said system is contingent upon the specific objective of the recommendation system.

When considering user evaluations algorithms, that aim to predict its specific items, is advisable to employ evaluation metrics such as Mean Squared Error (MSE) or Root Mean Square Error (RMSE) to effectively assess algorithm performance.

The Mean Squared Error (MSE) is a quantitative measure used to assess the accuracy of a recommendation algorithm by calculating the average of the squared differences between the predicted ratings given by the recommendation algorithm and the actual ratings for a given set of n items under analysis [95]. The formula for Mean Squared Error (MSE) is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.7}$$

Where:

y_i represents the actual rating given by a user for item i.

$\hat{y}_i$ represents the estimate or prediction made by the recommendation system for the classification of item i.

n represents the total number of reviews or ratings made by users.

The Mean Squared Error (MSE) assigns more significance to larger errors due to the squaring of the differences. This implies that MSE value is more significantly influenced by substantial errors, by analyzing when its value is low which corresponds to an improvement performance of

the recommendation system that suggests that the predicted ratings are more closely aligned with the actual evaluations.

The Root Mean Square Error (RMSE) is a modified variation of Mean Square Error (MSE) that incorporates the square root of MSE. This adjustment enhances the interpretability of the metric, as it is measured on the same scale as the original assessments. The formula for Root Mean Square Error (RMSE) is as follow:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.8}$$

Where:

y_i represents the actual rating given by a user for item i.

$\hat{y}_i$ represents the estimate or prediction made by the recommendation system for the classification of item i.

n represents the total number of reviews or ratings made by users.

Similar to Mean Squared Error (MSE), a recommendation system's performance is considered better when the Root Mean Squared Error (RMSE) value is lower. A lower Root Mean Square Error (RMSE) value signifies the system's proficiency in generating precise predictions, whereas a higher RMSE value suggests the presence of significant errors in this kind of system's.

In the context of recommender systems, Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) can be employed as evaluation metrics to assess the degree of concordance between the system's ratings or predictions and the users' actual ratings or preferences. This evaluation process facilitates the refinement and optimization of algorithms, thereby enhancing the accuracy of recommendations provided to users. Nevertheless, it is crucial to acknowledge that these metrics may not comprehensively encompass the caliber of a recommendation system, as there are other pertinent metrics, such as ranking metrics, that are also significant in assessing its efficacy.

When dealing with algorithms that aim to provide a ranked list of items, specifically the *top-k* items, it is common to utilize ranking metrics such as $Precision@k$, $Recall@k$, $F-measure@k$, Mean Reciprocal Rank, Discounted Cumulative Gain [82], and Normalized Discounted Cumulative Gain [53]. These metrics are particularly useful in evaluating the performance of systems in terms of accurately classifying and selecting items that are relevant to users.

1. **Precision in K (Precision@k)**:

   - $Precision@k$ measures the proportion of relevant items among the *top-k* recommended items.
   - It is calculated as follows:

$$Precision@k = \frac{relevant\ items@k}{k} \tag{2.9}$$

- This metric provides a measure of the quality of the recommended items at the beginning of the list. Their values range from 0 to 1, where a value closer to 1 indicates greater precision.

2. **Recall in K (Recall@k)**:

   - $Recall@k$ measures the proportion of relevant items found in the *top-k* items in relation to the total available relevant items.
   - The formula is as follows:

$$Recall@k = \frac{relevant\ items@k}{total\ relevant\ items} \tag{2.10}$$

   - This metric provides a measure of the recommender system's ability to retrieve all relevant items, and thus higher recall values indicate that a greater number of relevant items are being retrieved in the recommendations.

3. **F-measure in k (F-measure@k)**:

   - $F - measure@k$ combines $Precision@k$ and $Recall@k$ to provide a single metric that balances precision and recall. It can be calculated using the harmonic mean:

$$F\text{-}measure@k = 2 \times \frac{Precision@k \times Recall@k}{Precision@k + Recall@k} \tag{2.11}$$

   - This metric is useful for achieving a balanced evaluation of the recommendation system.

4. **Mean Reciprocal Rank (MRR)**:

   - Mean Reciprocal Rank evaluates the quality of the classification of the most relevant item. It is the average of the reciprocal ratings of the most relevant item for each query or user.
   - The formula to calculate Mean Reciprocal Rank is:

$$MRR = \frac{1}{n\_users} \sum_{i=1}^{n\_users} \frac{1}{rank_i} \tag{2.12}$$

   - Where $rank_i$ refers to the rank of the first relevant item for $user_i$. MRR is utilized to assess how effectively the most relevant item is positioned at the top of the recommendation list. A higher MRR value indicates that the most relevant item tends to be ranked higher on average, reflecting the recommendation system's improved performance in promptly returning relevant items to users.

5. **Discounted Cumulative Gain (DCG)**:

   - DCG takes into account the position and relevance of the items in the recommended list, so assign decreasing weights to items as they move down the list of recommendations, prioritizing the most relevant items and those that appear higher in the list.

- The formula for DCG is:

$$DCG@k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} \tag{2.13}$$

where rel_i is the relevance of the item at position i and k represents the first k items on the recommended list.

- DCG aids in evaluating the distribution of relevance across the list, where a higher value signifies a better distribution of relevance, prioritizing more relevant items towards the top of the list.

6. **Normalized Discounted Cumulative Gain (nDCG)**:

- nDCG normalizes the DCG by dividing it by the ideal DCG, which is obtained when the relevant items are perfectly ranked at the top.
- The formula for nDCG is:

$$nDCG@k = \frac{DCG@k}{IDCG@k} \tag{2.14}$$

where $IDCG@k$ is the ideal DCG.

- Normalized nDCG provides a metric between 0 and 1, making it easier to compare different recommended lists.

In conclusion, the metrics discussed in this chapter, provide essential tools for evaluating recommendation systems that could enable us to measure their performance and fine-tune them for optimal user's experiences.

2.4 Challenges

Similar to any other domain, recommendation systems face also obstacles and constraints. There exist several restrictions pertaining to both implementation and ethical considerations that give rise to moral problems [62, 64, 78, 83].

- **The Phenomenon of Filter Bubble:** Recommender systems have the potential to generate filter bubbles, wherein users are only presented with information and perspectives that align with their pre-established views [15]. This phenomenon has the potential to result in polarization and a dearth of diverse viewpoints.

- **Privacy and Security leak:** The act of acquiring personal information for the purpose of tailoring suggestions carries the potential risk of privacy infringements, particularly when there is mishandling or unauthorized disclosure of the collected data [97]. Additionally, it is possible that there are security vulnerabilities that could potentially compromise the confidentiality of sensitive information.

- **Fairness and Bias:** Recommendation systems carry the potential for bias, exhibiting a preference towards some demographic [35], cultural, or socioeconomic cohorts while disregarding others. This phenomenon has the potential to result in discriminatory practices and an absence of fairness in the formulation of recommendations.

- **Exploitation vs. Exploration:** Striking a balance between the exploration of novel items and the exploitation of familiar ones poses a significant difficulty [40]. Certain systems may become trapped in what is known as ”local maximums” when they consistently provide recommendations for the same categories of content.

- **Data Scarcity:** This refers to the limited availability or insufficient quantity of data for a certain research or analysis. Recommendation systems often rely on user data to generate suggestions, namely the assessments provided by users. The scarcity of data poses a significant challenge, particularly for inexperienced users or goods with limited interactions. This directly impacts the density of the ratings matrix, resulting in a high level of sparsity. The issue discussed has a substantial impact on recommendation systems that rely on collaborative filtering. In systems with a large number of items, it is often observed that several items have limited or no reviews available. Consequently, this is a challenge for the system to generate accurate and reliable suggestions.

- **Cold Start Problem:** This issue refers to the challenge of addressing situations where there, is no prior contact history available for new users or new things [60]. In these cases, it is imperative for systems to identify and implement efficient strategies for generating recommendations that arise when a novel item or user, is introduced into the system as there is an absence of information or feedback pertaining to new items from users. Hence, this matter is prevalent in collaborative filtering methodologies, so that could untested items that would never receive recommendations. In contrast, the content-based method mitigates this issue by focusing on the inherent characteristics of items. In this case, enabling evaluations to be conducted based on their features. The issuing of incorporating new users into the system, poses a similar challenge for both collaborative filtering and content-based approaches. This is because these users have not yet provided any evaluations for items, rendering it impossibles to determine their preferences. Consequently, the creation of a user profile and the identification of users with similar interests become an arduous task.

- **Evaluating Suggestion Quality:** Establishing suitable evaluation measures is crucial in assessing the effectiveness of user recommendations, making this task multifaceted.

- **Dynamic User Interests:** User interests have the potential to undergo changes throughout the course of time. It is imperative for systems to possess the capability to identify and perceive these alterations afterwards adapting their recommendations in accordance with such modifications.

- **Regulatory Compliance:** The adherence to regulations, such as the General Data Protection Regulation (GDPR) [43] implemented in the European Union[4], entails limitations on the acquisition and utilization of personal data. Recommender systems are required to adhere to diverse legislation that differ across countries, hence posing challenges in achieving global compliance with all applicable regulations.

- **Manipulation and malicious attacks:** Recommendation systems are susceptible to manipulation by individuals with evil intent, who aim to deceitfully manipulate suggestions.

- **Long-term Impact:** Recommender systems possess the potential to exert a lasting influence on user behavior, thereby shaping their preferences and interests over an extended period of time. The aforementioned phenomenon may give rise to many social and ethical ramifications.

- **The Relationship between Sustainability and Information Ecology:** The overconsumption of recommended content has the potential to result in issues related to information saturation and cognitive overload. It is imperative for systems to take into account the well-being of people.

- **System scalability:** The issue of scalability, poses a significant obstacle in recommender systems, since it directly impacts the capacity of these systems to effectively handle a substantial volume of users, items, and interactions. As the database expands in size and user and item counts increase, recommender systems encounter substantial obstacles that can detrimentally affect their performance and efficiency. One of the primary scalability issues encountered in recommender systems pertains to the persistent expansion of the item collection. So that, as the collection expands with the addition of new products, recommender systems encounter it challenges of managing an increasingly extensive search field. The augmentation of recommendation algorithms not only amplifies their intricacy, but also presents, a formidable challenge in identifying pertinent goods for users, particularly inside extensive inventories. Moreover, with the significant increase on user base, the task of efficiently customizing recommendations for every unique user becomes a formidable problem. The act of customizing requires a greater allocation of computer resources and processing time, potentially resulting in system overload. One further challenge encountered is to the effective handling of substantial quantities of interaction data. The proliferation of recorded interactions, including but not limited to reviews, purchases, clicks, and various forms of feedback, has led to a substantial increase in the volume of data that needs to be processed. The augmentation in the quantity of data might have a negative effect on the duration needed to train recommendation models, hence impacting the overall efficiency of the system. In order to successfully address the issue of scalability, it is imperative to accurately determine the appropriate dimensions of the hardware and software infrastructure. The process

[4] `https://european-union.europa.eu/index_en`

of accommodating a substantial volume of concurrent user requests, can provide intricate challenges and significant expenses, hence requiring substantial investments in resilient resources to mitigate potential performance limitations. Furthermore, it is imperative to prioritize the inclusion of diverse recommendations, particularly when the quantity of things and users expands. Ensuring proposals possess diversity and relevance across a broad spectrum of interests is of paramount importance, albeit presenting notable difficulties when implemented on a significant magnitude.

- **Unstructured data:** The management of unstructured data is a significant obstacle, particularly in recommendation systems that employ a content-based methodology. In such systems, the conversion of unstructured data, such as textual content extracted from reviews, into a structured dataset of distinct features becomes imperative.

In conclusion, these issues necessitate a multidisciplinary approach that includes technical, ethical, legal, and social factors. Recommender system developers must consider the impact of their technologies on society and work to mitigate these challenges while providing useful and relevant recommendations to users. These multifaceted problems require creative engineering and technical solutions. maintaining recommendation performance and quality requires efficient algorithms and resource allocation.

3
State-of-art

In the state-of-the-art chapter, exploring the association between genes and diseases is paramount to understanding the genetic foundations of various pathological conditions. The process of identifying candidate genes and predicting their relevance involves innovative approaches, including machine learning and deep learning techniques. This chapter not only addresses the identification of candidate genes but also highlights the growing role of advanced technologies in predicting these genes through robust predictive methods. Additionally, the development of recommender systems is examined, playing a crucial role in suggesting candidate genes associated with diseases and providing a comprehensive insight into contemporary strategies for advancing the understanding of the complex relationship between genetics and pathologies.

3.1 Association between genes and diseases identification

A better comprehension of the significance of Natural Language Processing (NLP) and text mining in the field of medicine, particularly in the context of disease-related textual data, is a fundamental aspect of contemporary medical practice. In addition to the treatment of diseases, the field of medical science endeavors to comprehend and investigate domains such as protamines, microbiomics, and genetics. The investigation of these specific domains is of utmost importance in the study of the genetic composition of microbial species that are recognized as causative agents of diseases. Nevertheless, the identification of prospective novel entities, such as genes, proteins, or microbes, presents a formidable task owing to the extensive and continuously expanding body of knowledge within this field.

The role of scientific literature in the field of medicine is considered like greatest importance as it serves an essential channel for the dissemination of novel discoveries and advancements.

For instance, the PubMed[1] platform, encompasses a vast collection of over 30 million articles pertaining to biomedical literature. But in fact, a significant obstacle in this domain that pertains to the absence of organized and uniform datasets that incorporate the interactions between all diseases documented in the Disease Ontology (DO) and all identified genes, encompassing their respective variations. This dearth of comprehensive datasets, poses a hindrance to expediting scientific advancements.

A significant resolution to this issue, was presented in the research paper [24], which presents a novel system that has been developed to extract associations between diseases and their corresponding genes from MedLine [2] documents, utilizing Medical Subject Headings[3] (MeSH) terms. As an integral component of this study, a team of researchers constructed an all-encompassing lexicon by harnessing data derived from multiple publicly accessible databases pertaining to disease articulating and genetic information. This investigators, initially compiled a comprehensive dictionary which was encompassing gene-related information by utilizing various databases such as HUGO[4] and RefSeq[5]. In parallel, for the disease dictionary, they employed the Unified Medical Language System (UMLS). Hence, they employed the approach of identifying potential relationships by utilizing dictionary matching. Within the adding of a machine learning technique that was devised by the researchers, which leveraged Named Entity Recognition (NER) to effectively eliminate erroneous outcomes arising from dictionary matching. This study represented a significant advancement in comprehending the intricate associations between diseases and genes within the realm of biomedical literature.

The study [33] presented a novel methodology for streamlining the retrieval of disease-related information and establishing a robust framework to forthcoming applications that establish connections between diseases, treatments, causes, and pertinent information. The inclusion of entity normalization in the design of the NCBI Disease Corpus was of utmost importance, as it facilitated the mapping of disease mentions to standardized identifiers sourced from databases and ontologies such as Medical Subject Headings (MeSH) and Online Mendelian Inheritance in Man (OMIM)[6]. The conducted experiments, have provided evidence that this corpus possesses the potential to be utilized as a foundational resource for the purpose of training machine learning models. This finding emphasizes the significance of the corpus in the field of biomedical text mining research.

The article [34] presents a comprehensive dataset of uncommon monogenic disorders that have a well-established genetic basis. The dataset encompasses comprehensive information pertaining to diseases, including the genes responsible for their manifestation, as well as the scientific publications that initially elucidated the diseases and identified the associated genes. This information has been sourced from reputable databases such as Online Mendelian Inheritance in Man (OMIM),

[1] https://pubmed.ncbi.nlm.nih.gov/
[2] https://www.nlm.nih.gov/medline/medline_home.html
[3] https://www.nlm.nih.gov/mesh/meshhome.html
[4] https://www.genenames.org/
[5] https://www.ncbi.nlm.nih.gov/refseq/
[6] https://www.omim.org/

PubMed, Wikipedia[7], and other relevant sources. The examination of the dataset elucidates the temporal sequence of uncommon medical conditions and the identification of genes responsible for their occurrence, along with their correlation to advancements in research methodologies.

To tackle the challenge of identifying and elucidating the connections between diseases and genes, researchers have devised multiple tools, placing significant emphasis on the DisGeNET Cytoscape App [75] and BiONT [88]. The DisGeNET Cytoscape application effectively combines the features of Cytoscape with the comprehensive collection of disease-related genes and variants provided by DisGeNET[8]. The platform facilitates users in executing queries, conducting analyses, and generating visualizations of various network representations related to the connections between genes and diseases, as well as variants and diseases. In contrast, BiONT employs an unique approach by combining deep learning methods with biomedical ontologies, resulting in promising results in the advancement of biomedical relationship extraction. The deep learning system, refered to as BiOnt, utilizes four distinct categories of biomedical ontologies to extract and analyze relationships related to genetic products, phenotypes, diseases, and chemical compounds. The experiments conducted on three datasets demonstrated statistically significant improvements in F-score compared to existing methods that are considered state-of-art.

3.2 Candidate gene prediction

Candidate gene prediction is an essential domain in genetic research that further advances comprehension of diseases and facilitates the development of novel therapeutic approaches, notwithstanding the absence of data. The proliferation of genetic data enables the elucidation of intricate genetic landscapes and the resolution of the mysteries surrounding complex diseases. This research is important in the field of personalized medicine, which aims to optimize treatment efficacy while minimizing adverse effects by customizing therapies to an individual's genetic composition. As the intricacies of the genetic code of life continue to be uncovered, the scientific community has been motivated to develop machine learning models to assist in the prediction of candidate genes.

The research [71] investigates the prediction of candidate disease genes for genetically heterogeneous diseases via protein-protein interactions. The researchers compared the interaction partners of identified disease loci that lack causative genes with protein interaction data from a variety of species. Using this interactions significantly increased the probability of discovering candidate disease genes, with an average of ten percent or higher of these genes predicted to be authentic disease genes. In addition, randomization and benchmark tests were conducted to validate the study's findings. It was noted that the use of protein-protein interactions to predict disease genes has certain limitations, such as the presence of noise in high-throughput interaction data and the difficulty in mapping interactions from other species to human proteins. In summary, the research demonstrated that protein-protein interactions have the potential to serve as a valuable

[7] https://en.wikipedia.org/wiki/Main_Page
[8] https://www.disgenet.org/

indicator for candidate disease genes.

The purpose of the research [41] is to use protein-protein interaction data to prioritize candidate genes for type 1 diabetes. The researchers compiled a list of known type 1 diabetes disease genes as well as positional candidate genes derived from known linkage loci. They discovered that the protein-protein interaction network of known type 1 diabetes genes had distinct topological features and a greater number of interactions. It was also defined by them new candidate disease genes as positional candidates which were first-degree protein-protein interaction neighbors of known disease genes. Cross-validation revealed that this method outperformed random selection and using linkage information alone. The new candidates were also found to be significantly cited in type 1 diabetes related publications and to be over-represented in type 1 diabetes-related gene ontology terms. Sequence analysis also revealed that they had more type 1 diabetes related protein domains. This researching lend indirect support to the newly predicted candidates. The study suggests that protein-protein interaction information can be used to prioritize type 1 diabetes candidate genes, and it emphasizes its possibility by using topological features, functional and sequence information, and literature citation. The study also discusses the approach's limitations as well as the need for an integrative approach to understanding complex diseases.

The study [57] evaluates a cutting-edge method designed to extract gene-phenotype associations from biomedical literature. Emphasizing the pivotal role of such associations in understanding disease mechanisms and advancing treatment strategies, the paper highlights that while some associations are available in standard resources, a significant portion remains undiscovered in the literature. To overcome this challenge, the research team proposes a synergistic approach, combining statistical methods with an ontology-based strategy that focuses on text mining gene-phenotype associations. This innovative approach not only successfully retrieves known associations from existing resources, resulting in remarkable Area Under the Curve (AUC) values, but also showcases its efficacy in predicting potential disease-causing genes by analyzing semantic resemblances among phenotypes. The evaluation of their disease candidate prediction model demonstrates consistently high Area Under the Curve (AUC) values for gene-disease association datasets. Manual analysis further validates the accuracy of the method in extracting both established and novel gene-phenotype associations. In summary, the findings suggest that this approach adeptly uncovers associations between genes and phenotypes within published research.

3.2.1 Prediction of candidate genes using machine learning and deep learning techniques

The PERCH framework [38] described in the paper, is a versatile tool for gene prioritization in its discovery research and variant classification in clinical genetic testing. Both, it integrates multiple factors such as deleteriousness, allele frequency, call quality, segregation, association, and biological relevance that prioritize disease genes and classify variants of unknown significance. One of the methods employed involves the Naïve Bayesian model, which generates a global score predicting the probability of a genetic variant being pathogenic. The framework is shown to be

more accurate and powerful than existing methods until the date of publication, and it allows for customization based on study designs. However, there are some limitations to consider, such as the potential overfitting to known variants and the need for specifying a genetic model for the disease. Overall, PERCH provides a comprehensive and efficient approach for interpreting genetic variants discovered through next-generation sequencing.

The authors present OPA2Vec [87], a machine learning approach aimed like a producing vector representations of biological entities within ontologies. This is accomplished by combining formal ontology axioms and annotation axioms derived from ontology metadata. Researchers use a pre-existing Word2Vec model to generate feature vectors from a corpus, abstracts, or full-text articles, and then validate their approach (a neural network algorithm) by predicting protein-protein interactions and gene-disease associations. The results show that OPA2Vec outperforms other methodologies in predicting gene-disease associations up to the date of the article's publication. It also shows promise for identifying candidate genes associated with rare and orphan diseases. In general, OPA2Vec shows great promise as a methodology for generating vector representations of biomedical entities using ontologies.

Integrating critical data regarding phenotypes, gene functions, and anatomical sites of gene expression is the primary aim of paper [21], which seeks to improve disease-gene prioritization. In pursuit of this objective, the research utilizes a variety of ontologies to annotate genes and their corresponding phenotypes, including the Gene Ontology[9] (GO), Mouse Phenotype[10] (MP), Uberon Anatomy Ontology[11], and PhenomeNET Ontology. Furthermore, the research makes use of the Tissue Expression Profiles (GTEx) dataset in order to ascertain patterns of gene expression in a wide variety of tissues. The authors have successfully devised an advanced algorithm for graph-based embeddings, which they refer to as DL2Vec. In order to discover vector representations for biological entities, this algorithm utilizes structural information, axioms, and ontology-driven annotations. Vector representations of words are obtained using the Word2Vec model, which is predicated on their co-occurrence in a given contextual window. To forecast the relationships between genes and diseases, the authors implement a pointwise learning-to-rank model that calculates a similarity score between two vectors provided as input. The training of this model is executed efficiently by employing binary cross-entropy as the loss function. For the assessment of the model's performance, the study relies on the Receiver Operating Characteristic (ROC) curve and the Area Under the Receiver Operating Characteristics Curve (ROCAUC) metric, and the authors also provide data on the recall at rank n (Hits@n). Every inconsistency in the ROCAUC is thoroughly assessed using the Mann-Whitney U test. For the purpose of advancing the prioritization of disease genes, this research paper presents an innovative strategy that combines ontologies, embedding methods, and machine learning in a seamless way. The model's performance surpasses significantly that of contemporary approaches, thereby substantially broadening the scope of genes that can be prioritized based on phenotype, function, or site of expression compared to existing

[9] https://geneontology.org/
[10] https://www.mousephenotype.org/
[11] https://obophenotype.github.io/uberon/

methods up to the date of its publication.

DeepSVP, a computational method introduced in [13], stands at the forefront of addressing the challenges posed by structural genomic variants in understanding human variability and diseases. By consolidating diverse datasets encompassing gene functions, phenotypic annotations, gene expression profiles from databases such as the Mouse Genome Informatics, Human Phenotype Ontology[12] (HPO), Tabula Muris Consortium, and GTEx tissue expression database, it revolutionizes the prioritization of structural variants linked to genetic diseases. Leveraging machine learning techniques, specifically an Artificial Neural Network (ANN) that incorporates genomic features and prediction scores generated by the DL2Vec-based phenotype prediction model, DeepSVP achieves effective categorization. This is demonstrated by its targeted applications, accurately linking genes and diseases based on their genetic attributes. The evaluation, conducted using performance metrics as is the Area Under the Precision–Recall Curve (PRAUC), Diagnostic Odds Ratio (DOR), Area Under the Receiver Operating Characteristics Curve (ROCAUC), and F1-score, showcases DeepSVP's proficiency in identifying causative variants associated with patient phenotypes within real genomes. Its success in uncovering novel pathogenic structural variations, especially in consanguineous families, underscores its potential for transformative impact in the field of genetic disease research.

In the exploration of phenotype ontologies and semantic machine learning, the study presented in [12] meticulously examines the identification of genes linked to diseases, providing in-depth insights into results, models, and data. The researchers leveraged integrated phenotype ontologies to gather diverse phenotype data from model organisms such as the mouse, zebrafish, fruit fly, and fission yeast. Using various techniques, they quantified phenotypic similarity between these model organisms and human diseases. To predict gene-disease associations based on these similarities, the research employed both a multilayer perceptron (MLP) and a naïve classifier, taking special care to address and evaluate annotation bias. Notably, the study underscored the significance of mouse genotype-phenotype data as the most valuable dataset for discerning genes associated with human diseases. Methodologically, the evaluation included the calculation of false-positive and true-positive rates at each rank, accompanied by the generation of Receiver Operating Characteristic (ROC) curves and the computation of Area Under the Receiver Operating Characteristics Curve (ROCAUC) using this data. Overall, this research bears profound importance in advancing integrated phenotype ontologies and elevating the utilization of phenotypes from model organisms in the interpretation of human genetic variants.

The research paper [42] introduces GenePredict-KG, a disease-gene prediction system based on a knowledge graph that is generated by establishing connections between entities extracted from diverse genotypic and phenotypic databases. Utilizing knowledge graph embeddings, the system forecasts novel disease-gene interactions. By considering performance metrics (Area under receiver operating characteristic - AUROC, Area under precision-recall - AUPR, and Mean reciprocal rank - MRR), GenePredict-KG surpasses other contemporary approaches. Nevertheless,

[12] `https://hpo.jax.org/app/`

additional research is required to address certain constraints, including the weighting of neighboring entities, data class imbalance, establishing causal relationships, and identifying truly novel disease-gene associations. GenePredict-KG exhibits potential in the identification of genes associated with diseases.

As a result of significant advancements in genomics and candidate gene prediction, the relationship between genes and diseases is now better understood. The availability of data and the evolution of computational models have been instrumental in the advancement of this field. With fresh insights into candidate gene identification and the intricate connections between genes and diseases, this thesis endeavors to augment comprehension in these areas. Profound implications of this research extend to personalized medicine and genomics, potentially fostering progress in the identification and management of genetic disorders. In general, these investigations serve to augment the body of knowledge and offer significant perspectives for medical and genetic research, thereby advancing the trajectory toward scientific breakthroughs in the field of medicine.

3.2.2 Recommender systems to recommend candidate genes associated with diseases

The article [72] centers on the inference of gene regulatory networks (GRNs), which consist of proteins, metabolites, and genes, as well as the interactions among them. By attributing the challenge of gene regulatory networks inference to a model based on collaborative filtering, the research employs the notion of recommendation systems. Afterwards, this model is implemented to forecast genetic relationships by leveraging a variety of data sources. The computation of gene similarity is performed by aggregating attributes from various data sets, with Cosine similarity (Equation 2.1) being the preferred method. In order to select neighbors that represent genes with similar characteristics, Pareto dominance and similarity values are taken into account. The selection of regulated genes is achieved via collaborative filtering, which considers the established connections among adjacent genes. In order to assess the efficacy of the system, the metrics $Precision@k$, $Recall@k$, and $F1 - score$ are selected. The findings reinforce the efficacy of recommender systems methodology in forecasting genetic relationships, particularly when data from numerous sources is incorporated.

Although the study [99] question is not overtly classified as a recommendation system, upon closer inspection, it becomes apparent that it is structured as a collaborative filtering recommendation system. The issue recognized by academics pertains to the tendency of numerous candidate gene recommendation methods to concentrate solely on the specific modeling of each disease or phenotype. However, this methodology fails to account for common patterns that may exist among various conditions or disease categories.

In order to address this constraint, scientists implement the factorization technique of a gene-phenotype matrix with sparsely distributed infill. The primary aim is to predict the unidentified entries within this matrix. To enhance the precision of the gene-phenotype matrix completion, the team expands upon the traditional Bayesian factorization method by integrating numerous

secondary sources of information. The innovation is distinguished not solely by its integration of gene-related data sources, but also by its incorporation of data sources that contain Human Phenotype Ontology terms.

The obtained results offer encouraging prospects for the prediction of genes associated with a range of diseases, particularly those associated with congenital malformations and the nervous system.

The article [50] introduces a significant trend in the field, highlighting the application of recommendation systems to genes, specifically targeting the exploration of gene interests (Gi) and the more accurate recommendation of pathogenic human genes to patients. To address this, the researchers proposed a novel Gene TOP-N-based Collaborative Filtering algorithm (GeneCF), leveraging patients' gene interests (Gi) to enhance the precision of gene recommendations. By employing this algorithm, which utilizes a collaborative filtering strategy, a gene similarity matrix, gene expression matrix, and the gene jaccard similarity algorithm, the study aimed to provide healthcare providers with top recommendations of genes (top N) relevant to individual patient profiles.

The data used for this research was extracted from Stanford University's[13] genome database, focusing specifically on liver cancer. With gene expression data available for 3964 genes and a sample pool comprising 82 individuals diagnosed with hepatocellular carcinoma (HCC) along with 74 samples from non-tumor liver tissue, the study ensured a comprehensive analysis.

Evaluating the efficacy of the GeneCF algorithm involved the use of the Gene Precision-Coverage (GPC) method, which balanced precision and coverage in gene recommendations by considering metrics such as Gene Precision (GP) and Gene Coverage (GC). The results of this investigation revealed the algorithm's notable efficacy in computing gene interest for individual patients, thereby facilitating precise gene recommendations. These ones were validated through testing with carcinoma staining and RNA expression data.

As a consequence of employing the GeneCF algorithm, doctors were empowered to provide more intelligent and personalized care for cancer patients. The algorithm's application led to the discovery of six genes that could potentially be linked to liver cancer, signifying a significant leap in identifying crucial genetic markers associated with the disease.

The research paper [100] presents Deep Collaborative Filtering (DCF), a model that effectively ranks potential genes associated with diseases. The architecture of the system integrates deep learning techniques to hierarchically extract genetic information, thereby enhancing the accuracy of gene-disease association predictions. It is crucial to note that in the absence of negative examples, this architecture also employs positive-unlabeled learning (PU) to complete low-dimensional classification matrices. When assessing the system, the application of the precision-recall metrics curve emerges as a notable feature, underscoring its exceptional capability to identify genuine associations and categorize emerging disease phenotypes, thereby suggesting promise in the investigation of obscure connections. Beyond attaining scalability in the incorporation of varied

[13] `https://www.stanford.edu/`

genetic and disease attributes, the research proposes investigating alternative loss functions for the binary matrix. Subsequently, it is suggested that deep canonical correlation analysis should be used to identify common attributes among various auxiliary data sources. Additionally, the Deep Collaborative framework could be expanded to include Convolutional Neural Networks (CNN) and other deep learning models. Using filtration in an effort to enhance efficacy further.

Exome sequencing (ES) has emerged as a crucial diagnostic tool for Mendelian disorders in recent years. The analysis of hundreds of variants presents a substantial obstacle, impeding the identification of causative genes despite its potential. In response to the acknowledged necessity for thorough clinical analysis in the interpretation of Exome sequencing data, PhenoApt was created [23]. Utilizing clinical knowledge, this phenotype-driven gene prioritization tool permits phenotype-specific weighting via a machine learning algorithm. Functioning as a recommendation system, it places emphasis on the associations between discernible phenotypes and genes that are linked to hereditary disorders. It is worth noting that PhenoApt outperformed prior instruments in establishing top-10 lists of causative genes during baseline evaluations.

So as to prioritize tasks, PhenoApt employs a range of data sources, such as the Human Phenotype Ontology (HPO), Online Mendelian Inheritance in Man (OMIM), and Orphanet[14]. These databases provide information on phenotypes, diseases, and the genes associated with them. The tool creates a directed graph that illustrates the connections between phenotypes, diseases, and genes and utilizes graph embedding techniques to convert each node in the graph representing phenotypes, genes, and diseases into a vector representation. In this way, these vector representations are employed to quantify the associations between phenotypes and genes. PhenoApt computes the resemblance between vectors that represent the phenotypes of a patient and the genes that are recognized to be responsible for those phenotypes. The similarity is utilized to produce a prioritized catalog of potential genes, indicating the ones that are most probable to be linked with the observed phenotypes in patients. Furthermore, it enables medical professionals and investigators to allocate personalized importance to various phenotypes according to their clinical expertise.

PhenoApt, although proficient in ranking potential genes according to phenotypes, encounters constraints. The accuracy of diagnoses, particularly in cases involving new or multiple conditions, can be compromised by the assignment of weights to unrelated phenotypes and the lack of certain phenotypes in databases. Furthermore, since PhenoApt is primarily focused on phenotypes, it may not prioritize all pertinent genes as the highest-ranking ones, thus requiring a more thorough gene curation method.

In conclusion, PhenoApt employs phenotype, disease, and gene data to rank candidate genes, assisting in the identification of genes that are probably linked to observed phenotypes in patients with genetic disorders.

The article [58] introduces a recommendation system aimed at identifying causative variants in rare diseases while minimizing misdiagnosis risks caused by artefactual variants. This retrospective study harnesses clinical genetic interpretation and genomic data analysis from a specific

[14] `https://www.orpha.net/consor/cgi-bin/index.php`

period to enhance the accuracy of variant predictions in rare genetic diseases. Leveraging the random forest machine learning framework, this system effectively excludes extraneous variants and adeptly integrates clinical data, family history, and variant databases. As a result, it demonstrates exceptional precision in detecting pathogenic variants.

The system's training and testing against other prioritization models involve a fusion of various factors, including disease semantic similarity, pathogenicity probabilities, and quality control statistics. Evaluation via recall measures the system's accuracy in suggesting variants for patients within a predetermined rank. To gauge the impact of quality control-related statistics on system performance, the assessment includes 5-fold cross-validation and an ablation test. Additionally, post hoc analysis methods such as Shapley additive explanations (SHAP) and permutation feature importance provide valuable insights into feature significance and model interpretation.

3.3 Pytorch and recommendation systems

PyTorch[15] is a highly regarded and extensively used open source framework in the fields of machine learning and scientific computing, particularly emphasizing the design and implementation of deep neural networks. The exponential growth in popularity of this tool may be attributed to its efficiency, versatility, and active community support, as it provides a wide range of important features that are crucial for the development and training of deep learning models.

PyTorch's tensor structure[16] is a notable feature that enables efficient manipulation of multi-dimensional data, which is crucial for performing complicated numerical operations in machine learning. Additionally, it incorporates an autograd system[17], which enables the automated computation of gradients for the purpose of optimizing the model throughout the training process.

A diverse range of modules and functions are available for the construction of neural networks[18], including convolutional layers, linear layers, activation functions, and optimizers[19]. The versatility of PyTorch makes it a perfect option for conducting experiments and research, enabling developers to easily design and evaluate neural network structures.

In addition, it has inherent integration with GPUs, allowing for substantial acceleration in model training by using the computing capabilities of this specialized hardware[20]. The aforementioned skill plays a pivotal role in effectively managing ever bigger data sets and more intricate models.

PyTorch has gained significant popularity in the field of recommendation systems due to its unique characteristics [39, 52, 54, 67, 91]. It has been widely acknowledged for its efficacy in developing and training embedding models. These models produce dense vector representations of items and users, capturing crucial attributes that enable the recommendation system to acquire

[15] https://pytorch.org/
[16] https://pytorch.org/docs/stable/tensors.html
[17] https://pytorch.org/docs/stable/autograd.html
[18] https://pytorch.org/docs/stable/nn.html#module-torch.nn
[19] https://pytorch.org/docs/stable/optim.html
[20] https://pytorch.org/docs/stable/cuda.html

similar relationships between them.

Furthermore, it provides a diverse range of specialized modules and features that may be used to construct neural networks tailored for recommendation systems. These networks are particularly valuable in representing the interaction between users and items, especially when used in the context of collaborative filtering [54, 55]. In this application, the system is capable of predicting users' preferences by using previous interactions and item attributes.

PyTorch showcases its adaptability in more intricate situations, such as recommendation systems that include a series of user interactions over time (e.g., suggesting things in an online shop), by facilitating the development of attention models and sequence models [69]. By including temporal and sequence patterns in user preferences, these models enhance the accuracy of recommendations.

Moreover, the inherent capability of PyTorch to facilitate training on Graphics Processing Units (GPUs) confers a significant benefit, especially in the context of recommendation systems that often handle large datasets [1, 102]. GPU processing enables enhanced acceleration, resulting in expedited and more effective model training, hence bolstering the scalability of these systems.

PyTorch features the TorchRec library[21] [52], which is specifically designed to provide innovative solutions in the area of recommendation systems, in order to strengthen its position in this sector. The library provides further functionalities and specialized tools to streamline the construction and training of recommendation models, making PyTorch a favored option for scholars and professionals engaged in this particular domain of inquiry.

In conclusion, PyTorch, together with its TorchRec library, presents itself as a robust and adaptable solution for tackling intricate problems in recommender systems. It provides sophisticated and effective functionalities for constructing, training, and implementing high-performing, tailored models.

[21] `https://pytorch.org/torchrec/`

4
Data

This chapter furnishes a thorough and detailed description of the diverse data sources harnessed in this study. It encompasses an in-depth elucidation of the methodology employed for the meticulous collection of data, delineates the intricate process involved in crafting the definitive SQLite database essential for seamless integration into the model, and provides a concise yet illuminating exploratory analysis of the resultant data. Hence, it meticulously delineates every step taken to accomplish the *primary objective*: Creation of a major standard dataset with the interactions between the diseases in the Disease Ontology (DO) and the known genes. The objective is to create a $\langle disease,\ gene,\ interaction,\ (other features)\rangle$ dataset.

4.1 Data sources

4.1.1 Disease Ontology (DO)

Disease Ontology[1], denoted as DO, is an ontology intended to methodically classify and arrange data pertaining to a wide range of diseases. The primary purpose of this endeavor is to achieve a standardized vocabulary, which is crucial for enhancing interoperability and maintaining consistency in the fields of biomedicine and clinical practice. Understanding the intricate interrelationships and classifications that are intrinsic to the field of diseases is enhanced through the systematic organization made possible by the DO's structured framework.

An examination of the historical development of Disease Ontology reveals that its inception was inspired by the urgent requirement for a standardized method to categorize and structure disease-related data. In response to the growing intricacy of biomedical and clinical data, the

[1] `https://disease-ontology.org/`

initiative was initiated to establish a systematic framework that would foster collaboration and integration of information, in addition to enhancing comprehension.

A pivotal moment in the progression of Disease Ontology is characterized by its dedication to the integration of semantics. The integration of DO terms with various well-established medical vocabularies, such as Medical Subject Headings[2] (MeSH), International Classification of Diseases[3] (ICD), NCI thesaurus[4], Systematized Nomenclature of Medicine[5] (SNOMED), and Online Mendelian Inheritance in Man[6] (OMIM), is accomplished via comprehensive cross-mapping. Through the strategic alignment of its terminology with these universally acknowledged vocabularies, DO guarantees that its organized data remains interconnected with more extensive medical and scientific understanding. This integration not only enhances the practicality of the information but also integrates it into the broader clinic and biomedical domain.

Among the attributes and advantages of the Disease Ontology are the following:

- **Standardized Terminology:** DO serves as a controlled and standardized vocabulary for the purpose of describing diseases. Its implementation guarantees consistency in the way disease-related information is represented across various data sources and research investigations.
- **Hierarchy of Terms:** The diseases classified under the DO are arranged in a hierarchical fashion, establishing a systematic taxonomy that mirrors the interconnections among various ailments. A classification and navigation of disease-related information is facilitated by this hierarchical structure.
- **Terms and Definitions:** Each term in the Disease Ontology is accompanied by a precise definition, providing clarity and context for researchers, clinicians, and data analysts. These definitions contribute to the unambiguous interpretation of disease-related terms.
- **Relationships Among Diseases:** The Disease Ontology encompasses relationships among diseases, including parent-child relationships that signify, respectively, more general and specific disease categories. This facilitates the investigation of correlations and parallels among various diseases.
- **Integration with Other Ontologies:** The Disease Ontology is designed to be interoperable with other ontologies, databases and resources in the biomedical domain. This interoperability enhances the integration of disease-related information from diverse sources.
- **Cross-References:** DO incorporates cross-references to additional pertinent databases and ontologies, facilitating smooth transitions between diverse resources and enhancing the user's comprehension of diseases in a contextual manner.

[2] https://www.nlm.nih.gov/mesh/meshhome.html
[3] https://icd.who.int/en
[4] https://ncithesaurus.nci.nih.gov/ncitbrowser/
[5] https://www.snomed.org/
[6] https://www.omim.org/

- **Application in Biomedical Research:** Disease Ontology is employed by researchers to annotate and scrutinize clinical records, experimental data, and literature, thereby augmenting the collective comprehension of disease pathology and biology.
- **Consistent Updates:** The Disease Ontology undergoes periodic expansions and updates to integrate emerging insights and knowledge in the domains of medicine and biology, thereby guaranteeing its ongoing pertinence and precision.

Disease Ontology Identifiers (DOIDs) adhere to the format DOID: followed by a number. For instance, DOID:9744 is assigned to the disease type 1 diabetes mellitus, and the data pertaining to it was obtained from the disease ontology website (Figure 4.1) at 10:58 am on November 10, 2023.

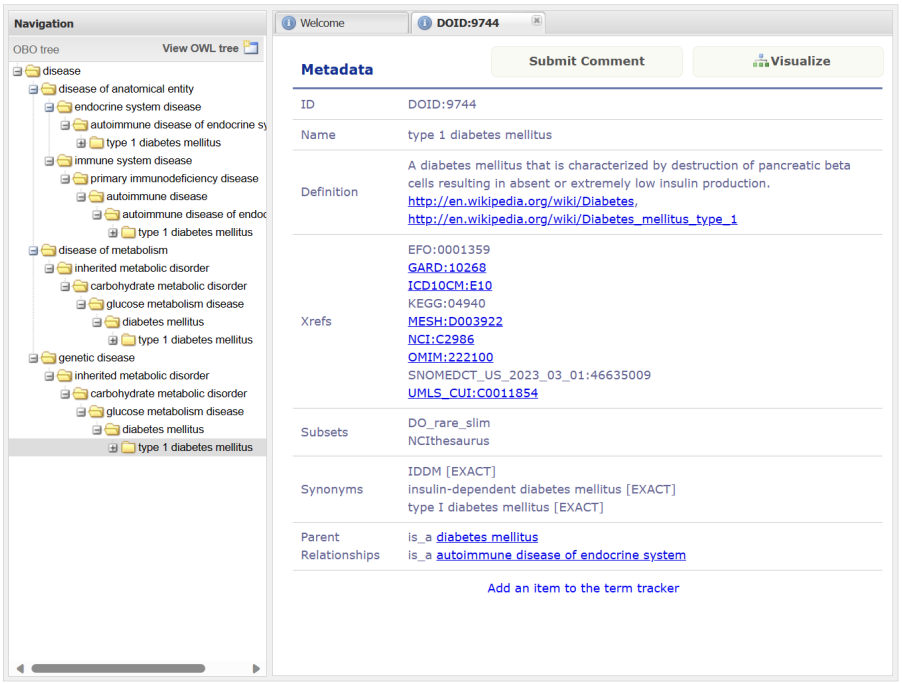


Figure 4.1: Metadata for DOID:9744

Fundamentally, the Disease Ontology serves as a dynamic and evolving instrument that not only functions as a repository of information but also enables the scientific and medical community to decipher the intricate nature of diseases. The Disease Ontology, by virtue of its systematic structure, incorporation of semantics, and ability to accommodate changes over time, is an exceptionally valuable resource that fosters collaboration and knowledge development in the complex realm of biomedical and clinical research.

4.1.2 DisGeNET

DisGeNET[7] is a database that compiles data from diverse sources, including scientific literature, genomic databases, and disease associations, in order to ascertain information regarding the

[7] https://www.disgenet.org/

associations between human genes and diseases.

Its principal objective is to provide an all-encompassing and cohesive perspective on the interrelationships among genes and diseases. This encompasses information regarding genetic variants, evidence of associations between particular genes and various pathological conditions, and comprehensive details regarding the characteristics and resilience of these associations. DisGeNET offers extensive information pertaining to diseases and genes in order to facilitate the interpretation and analysis of associations between diseases and genes and their variations. Diseases are represented by the UMLS and numerous databases, including but not limited to Online Mendelian Inheritance in Man (OMIM), Medical Subject Headings (MeSH), Disease Ontology, Human Phenotype Ontology[8], and Orphanet Rare Disease Ontology[9] (ORDO). In accordance with the HUGO Gene Nomenclature Committee[10] (HGNC), all gene-related information, including official symbols and complete names, is extracted from the Gene[11] database maintained by the National Center for Biotechnology Information[12] (NCBI). The platform classifies these associations according to the abundance and quality of available evidence, in addition to cataloging them. The evidence score associated with each association in this classification aids researchers in evaluating the dependability and pertinence of the data that is being presented.

With the ability to conduct in-depth research and detailed analyses on the genetic basis underlying a vast array of human diseases, the platform is an indispensable resource for the scientific community and health professionals. It not only aids in the advancement of research, but also contributes to the formulation of therapies, diagnoses, and treatment strategies for a wide range of medical conditions through the provision of current and organized data.

4.2 Creation of the DiseaseGene database

The fundamental file pertaining to diseases was extracted from the original account of the Disease Ontology, located on GitHub (`https://github.com/DiseaseOntology`), in order to acquire the requisite information. Critical to the comprehension and analysis of the disease-related data contained in the ontology, the aforementioned file, entitled `'HumanDo.json'`, was downloaded at 15:26 on October 30, 2023 from the repository HumanDiseaseOntology. In total were discovered 13863 entities associated with diseases, identified through the use of IDs in the Disease Ontology.

The pertinent data for this study were obtained via the `DisGeNET SQLite 2020-v7.0` SQLite database on November 22, 2023, at 3:57 pm. Specifically, the files `disease_mappings.tsv` and `disease_mappings_to_attributes.tsv` were also downloaded DisGeNET plataform for the purpose of mapping the diseases in the SQLite database to the diseases present in the Disease Ontology.

[8] `https://hpo.jax.org/app/`
[9] `https://www.orpha.net/consor/cgi-bin/index.php`
[10] `https://www.genenames.org/`
[11] `https://www.ncbi.nlm.nih.gov/gene/`
[12] `https://www.ncbi.nlm.nih.gov/`

In the course of this investigation, the DiseaseAttributes table from the SQLite database was used. This table comprises the following attributes: 'diseaseNID', 'diseaseId' (UMLS), 'diseaseName' (disease name), and 'type' (disease, phenotype, group). A primary goal was to augment this table with a novel column designated 'DOID', which would facilitate the mapping of disease names to their corresponding entries in the Disease Ontology.

The procedure commenced with the mapping of the UMLS values from the JSON file, entitled `HumanDo.json` to the corresponding UMLS values in the database's 'diseaseId' column. Following this, associations were established between the disease names and synonyms in the JSON file and the disease names in the `disease_mappings_to_attributes.tsv` file. Diseases in the JSON file were subsequently mapped using the `disease_mappings.tsv` file, which contained the DOID, OMIM, and ORDO values in addition to their names and synonyms. The objective of this procedure was to enhance and broaden the mapping of disease-related data across various data sources. In order to streamline the process of mapping DOID values, a new column named 'mappingDOID' was introduced. This column contains integer values, each of them corresponds to a certain DOID value.

Considering that the reference database was already structured in a format compatible with SQLite, its use facilitated the integration and manipulation of data in an efficient and consistent way. Thus, a SQLite database named `DiseaseGene` was successfully generated for the specific purpose of this research. The database `DiseaseGene`, comprises a table called Disease, which incorporates the aforementioned modifications. The geneAttributes table from the DisGeNET database was duplicated as Genes in the `DiseaseGene` database, with the exclusion of the 'pLI', 'DSI', and 'DPI' columns, since they are irrelevant to the research objective. Finally, the geneDiseaseNetwork table from the DisGeNET database was replicated into the `DiseaseGene` database under the name DiseaseGeneNetwork, with the addition of the columns 'DOID' and 'mappingDOID' to facilitate data input in the model, and the columns 'association' and 'EL' being removed.

To summarize, a `DiseaseGene` SQLite database was created, consisting of three tables: Disease, Genes, and DiseaseGeneNetwork. The structure of the database is seen in Figure 4.2, and the characteristics of each table are outlined in Tables 4.1, 4.2, and 4.3, respectively.

Table 4.1: Disease Table

Column	**Description**
diseaseNID	Unique identifier for each disease
diseaseId	Disease identifier
diseaseName	Disease name
type	Type of disease
DOID	Disease Ontology ID
mappingDOID	Mapping DOID

Table 4.2: Genes Table

Column	Description
geneNID	Unique identifier for each gene
geneId	NCBI gene identifier
geneName	NCBI Official Full Name
geneDescription	Gene description

Table 4.3: DiseaseGeneNetwork Table

Column	Description
NID	Unique identifier for each record
diseaseNID	Reference to diseaseNID in Disease table
DOID	Disease Ontology ID
geneNID	Reference to geneNID in Genes table
source	Source of information
associationType	Type of association
sentence	Associated sentence
pmid	PubMed ID
score	DisGeNET Score
EI	Evidence Index - Evidence Leve
year	Year of publication
mappingDOID	Mapping DOID

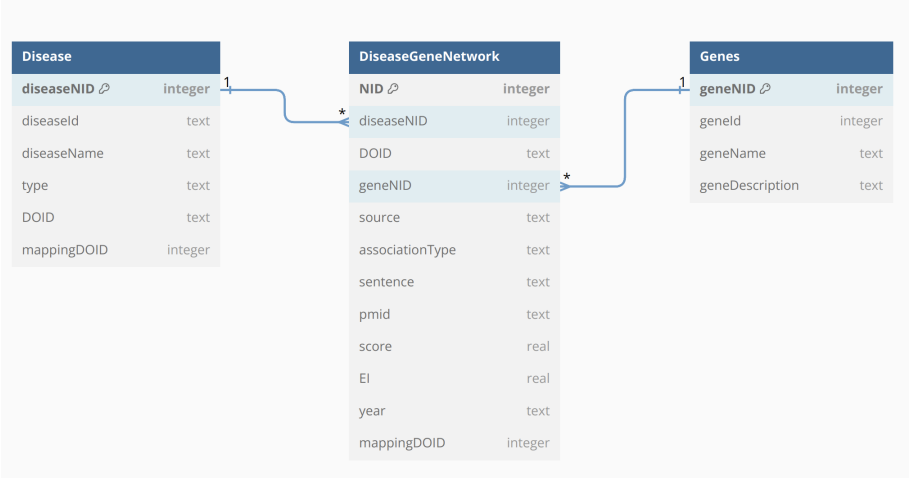


Figure 4.2: DiseaseGene schema

4.2.1 Data Analysis

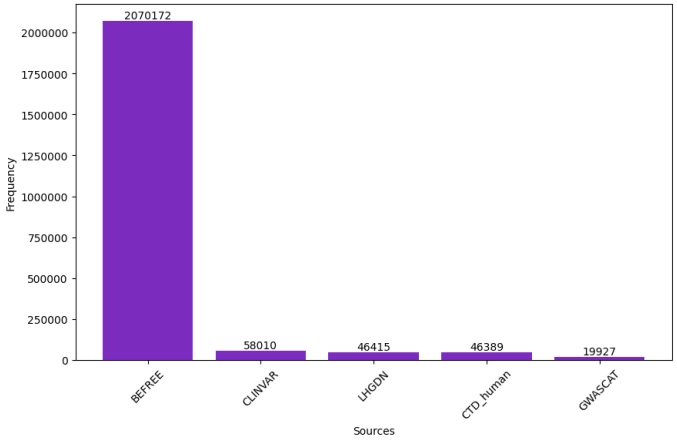
The Diseases table comprises 9929 unique UMLS values representing diseases and 7101 distinct DOID values. As a result, it's possible for a single DOID value to be associated with multiple UMLS values. With 7101 unique DOID identifiers, this dataset covers a spectrum of 9929 diseases. Among these illnesses, they can be classified into three distinct categories: Disease, Group, and Phenotype. Upon examining Table 4.4, it was observed that the dominant category is 'Disease'.

Table 4.4: Frequency of different types of diseases

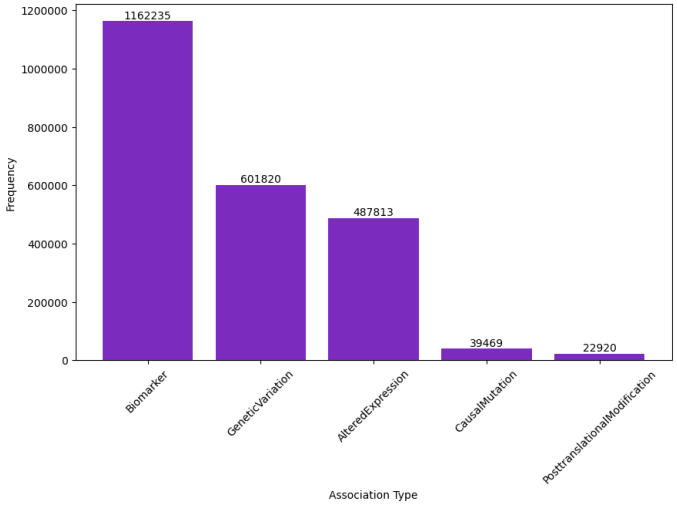
Disease	**Group**	**Phenotype**
9159	449	321

Regarding the Genes table, it has a total of 20024 genes.

The DiseaseGeneNetwork table comprises 706258 distinct scientific articles (PubMed IDs), and Figure 4.3(a) displays the five principal information sources. Furthermore, Figure 4.3(b) illustrates the five most prevalent varieties of associations. In regards to the quantity of articles per disease, Figure 4.4 predominantly indicates that there exists a significantly larger number of diseases with only one or two associated articles compared to others. This phenomenon may be attributed to the rarity of certain diseases or to recent research interest in these conditions.



(a) Top-5 information sources



(b) Top-5 types of gene-disease associations

Figure 4.3: Brief analysis of literature sources, and gene-disease associations present in the DiseaseGeneNetwork table.

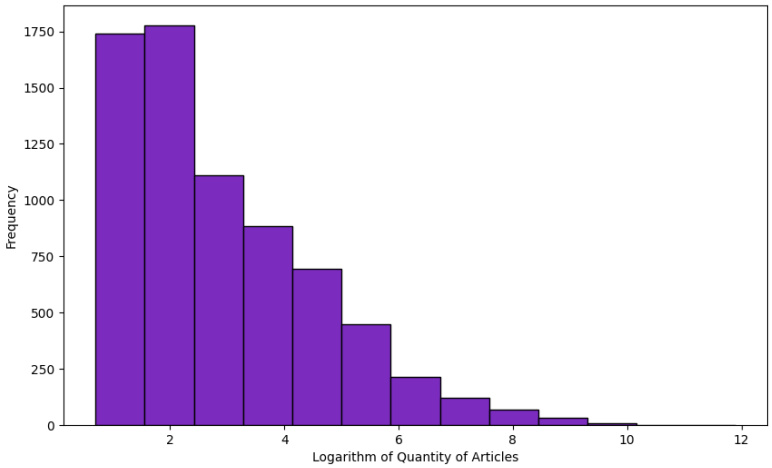


Figure 4.4: Distribution of the Number of Articles by Disease

By grouping the DiseaseGeneNetwork table in unique pairs (mappingDOID, geneNID), 541102 unique pairs can be confirmed. In establishing this grouping, the mean quality of scientific evidence (the 'EI' column) was utilized. The fundamental metrics (Table 4.5) comprise the mean ($\bar{x}$), with a value of 0.973373, and the standard deviation (σ), with a value of 0.138928, which signify the dispersion of data around the mean value.

Additionally, the Table 4.5 presents the minimum and maximum values that were recorded in the sample, signifying the data's lower and upper boundaries (0 and 1, respectively). The values that represent quartile measurements— 25%, 50%, and 75% — are denoted as quartiles Q_1, median (Q_2), and Q_3, accordingly. By dividing the distribution into four equal sections, these values offer a more comprehensive examination of the dispersion of the data. As a result, the quartile values are roughly 1, indicating a strong correlation between the gene's presence or variation and the risk or occurrence of the disease in question. The robust correlation observed may suggest that the gene is, in fact, implicated in the development or susceptibility to a particular disease.

Table 4.5: Statistics for the level of evidence ('EI') for the DiseaseGeneNetwork grouped table

Mean ($\bar{x}$)	**Standard deviation (σ)**	**Minimum**	**25%**	**50%**	**75%**	**Maximum**
0.973373	0.138928	0.00	1.00	1.00	1.00	1.00

5 Model

This chapter provides an analysis of the implementation and environment factors that influenced the development of the model. Following this, a comprehensive analysis of the input data for the proposed model is conducted. This provides an in-depth analysis of the data preparation procedure, focusing on the approach of partitioning the data into distinct sets for training, validation, and testing purposes. Furthermore, the methods implemented to guarantee the quality, representativeness, and integrity of the utilized data sets are detailed.

The system under consideration is a PyTorch[1] implementation that utilizes the Torch library and employs a Neuronal Collaborative Filtering methodology. A comprehensive description is provided, encompassing its attributes, structure, and fundamental parameters. Code examples are included to enhance the illustrative value, and a concise mathematical explanation is accompanied by examples of the model's internal operations. In addition, technical intricacies are meticulously emphasized in order to furnish a thorough comprehension of the functioning of the proposed model.

Subsequently, an examination of the performance parameters employed to assess the model's efficacy, precision, and robustness is undertaken, furnishing a meticulous evaluation of the acquired outcomes. In this way, research objective number 2 is achieved.

5.1 Implementation and Environment

The project was developed using Python, using a modular and structured approach to enhance scalability and maintainability of the source code. The project repository may be accessed on GitHub using the following hyperlink.

[1] https://pytorch.org/

Throughout the implementation, many important libraries were used for different phases of the procedure, encompassing data processing, model implementation, training, and assessment. Notable libraries, such as Pandas[2], Scikit-learn[3], and Sqlite3[4], were used for direct data manipulation inside the database and pre-processing.

The distinguishing factor of this research, was the deliberate utilization of the PyTorch framework for implementing, training, and evaluating the model. PyTorch distinguishes itself by providing a very versatile interface that enables the creation and adjustment of intricate neural network structures. This is possible by its dynamic framework, which simplifies the process of defining and modifying these structures within the realm of machine learning.

PyTorch showcases efficient integration with tensor operations, optimizing crucial mathematical computations during neural network training, while also providing flexibility. This improvement enhanced operational efficiency and agility, leading to expedited and more efficient processing.

An important aspect to consider is the capability to execute computations on GPUs, which result in a substantial improvement in speeding up the processes of training, validation, and evaluation. This is particularly beneficial in situations where large datasets are involved, as it results in faster training times and, consequently, more precise and efficient outcomes.

Therefore, the decision to use PyTorch as the primary framework for this project facilitated the implementation of the recommendation model and created a strong and very effective training environment. This ultimately culminated in exceptional performance and more precise outcomes, conforming to the goals set for the project.

The code was executed in a computer with a 13th Gen Intel(R) Core(TM) i9-13900H Central Processing Unit (CPU) with a Max Turbo Frequency of 5.40 GHz . The CPU had a total of 14 cores and 20 threads. The system was equipped with 32 GB of Random Access Memory (RAM). Additionally, it features an NVIDIA® GeForce RTX™ 4070 Laptop Graphics Processing Unit (GPU) with 4608 CUDA Cores, a Boost Clock ranging from 1230 to 2175 MHz, and 8 GB of GDDR6 memory. The operating system used was Windows 11 Home, the CUDA version was V11.8.89, and the PyTorch version was 2.0.0+cu118.

5.2 Input Data

The initial phase of this endeavor entailed the careful manipulation and preparation of input data to generate datasets that were appropriate for the validation, training, and testing of the recommendation model. Combining libraries and methods, including Pandas, Sqlite3, and PyTorch, which assisted with data import, transformation, and formatting, enabled the achievement of this goal.

The information was obtained directly from the `DiseaseGene.db` SQLite database, which

[2] `https://pandas.pydata.org/`
[3] `https://scikit-learn.org/stable/`
[4] `https://docs.python.org/3/library/sqlite3.html`

was developed in Chapter 4 utilizing the Sqlite3 library. Once the database connection is established, a SQL query is executed to retrieve the pertinent attributes, including 'mappingDOID', 'geneNID', and 'EI', from the DiseaseGeneNetwork table. Then, the Pandas library was utilized to perform preprocessing on the unprocessed data. During this phase, the pairs 'mappingDOID' and 'geneNID' were grouped for the purpose of data aggregation. Additionally, statistical summaries were computed, including the mean and sum of 'EI' interactions (referred to as 'sum_EI' and 'mean_EI', respectively). In order to produce a set of distinct pairings (mappingDOID, geneNID), the scientific evidence levels for each pair were added together and subsequently averaged.

It was crucial to divide the data into training, validation, and test sets after processing. This division was established in consideration of the uniqueness of 'mappingDOID' identifiers. To ensure a fair and inclusive distribution among the diverse data sets, the samples were partitioned into distinct sets based on the quantity of samples per identifier. The model will undergo evaluation using two separate training datasets.

The division of the first dataset was predicated on the subsequent line of reasoning: When a mappingDOID value is associated with a single sample, that particular sample is explicitly allocated to the training set. One sample is allocated to the training set and the other to the testing set when a mappingDOID contains two samples. If the mappingDOID contains three samples, two of them will be allocated for training purposes, while the remaining one will be used for testing. For mappingDOIDs with more than three samples, a custom split strategy is implemented, in which 80% of the samples for a given mappingDOID are allocated to the training set, while the remaining 20% is divided equitably between the validation and test sets (10% each). This methodology was implemented to guarantee the inclusion of all mappingDOID, or diseases, in the training set and due to the fact that certain diseases are associated with a mere one or two genes. Therefore, this training set will be referred to as the *Complete Training Set*.

The second dataset follows the same data partitioning methodology, except that diseases with just a single associated gene are excluded from the training set. Thus, this training set will be named *Training Set Excluding Unique Pairs*, which excludes diseases that have only one associated gene.

To ensure the integrity of the datasets used in this study, no intentional anti-test examples were included in the training, testing, and validation sets.

The datasets were converted to numeric values utilizing the sklearn.preprocessing[5] package and LabelEncoder [6]. This procedure facilitated the representation of data in a format that is more conducive to model training and ensured that the identifiers commenced with the value 0. This is a prerequisite for certain libraries and models, thereby promoting competent programming methodology.

Utilizing PyTorch, the data were transformed into tensors. An extension of PyTorch's Dataset class, known as the `DiseaseGeneDataset` class, was developed (Listing 5.1) with the purpose

[5] `https://scikit-learn.org/stable/modules/preprocessing.html`

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html`

of enabling the conversion of data into tensors containing the 'diseases', 'genes', and 'ei' attributes needed for training the model. In the `__init__` method of the class, the 'diseases', 'genes', and 'ei' variables are initialized. The code includes two essential methods: `__len__` and `__getitem__`. The `__len__` method returns the quantity of samples in the data set, while `__getitem__` returns a dictionary containing PyTorch tensors for each of the attributes ('diseases,' 'genes,' and 'ei') at the specified index.

Listing 5.1: DiseaseGeneDataset class in Python.

```
import torch
from torch.utils.data import Dataset, DataLoader

class DiseaseGeneDataset(Dataset):
    def __init__(self, diseases, genes, ei):
        self.diseases = diseases
        self.genes = genes
        self.ei = ei

    def __len__(self):
        return len(self.diseases)

    def __getitem__(self, idx):
        return {
            "diseases": torch.tensor(self.diseases[idx],
                                     dtype=torch.long),
            "genes": torch.tensor(self.genes[idx],
                                  dtype=torch.long),
            "ei": torch.tensor(self.ei[idx],
                               dtype=torch.float),
        }
```

A later development utilized the DataLoader method in PyTorch to load data in batches ($batch\_size$), a feature that proves advantageous when training machine learning models on extensive datasets (Listing 5.2). The $batch\_size$ variable (used in the training set but not in the validation/test set) was set to 6 and the parameter refers to the number instances in the batch, in this case it means that there are 6 occurrences in each batch. This number differs from the number of batches. The $num\_workers$ parameter specifies the number of subprocesses to be used for data loading (which can speed up loading for large data sets) was set to 0. In addition, the code checks for GPU availability and, if one is present, (how is the case), transfers the data imported with the DataLoader to the GPU in order to enhance the computational efficacy of the model training process. At this moment, the data can be imported into the model.

Listing 5.2: Using DataLoader in PyTorch to load data into GPU if available.

```
import torch
from torch.utils.data import DataLoader
```

```
# Assuming that the variables train_dataset, val_dataset and
# test_dataset have already been previously defined

# Check for GPU availability
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
# print('Using device:', device)

# Loading data using DataLoader

batch_size = 6
train_loader = DataLoader(dataset=train_dataset,
                          batch_size=batch_size,
                          shuffle=True,
                          num_workers=0)
val_loader = DataLoader(dataset=val_dataset,
                          batch_size=batch_size,
                          shuffle=False,
                          num_workers=0)
test_loader = DataLoader(dataset=test_dataset,
                          batch_size=batch_size,
                          shuffle=False,
                          num_workers=0)
```

It is noteworthy to mention that the datasets produced throughout the stages of training, validation, and testing were inherently archived in Comma-separated values (CSV) files to facilitate the monitoring and analysis of individual sets. By means of a label decoding procedure, the values associated with the initial database data are maintained in these files. This decoding reversed LabelEncoder's previous label encoding, ensuring that the values in the CSV files accurately reflect the original values in the database.

Why was the level of scientific evidence ('EI') selected as opposed to an alternative metric?

In the context of this investigation, the "Evidence Index" (EI) was chosen as a suitable metric for determining the association between a gene and a disease. This decision was made in consideration of the index's calculation method and its pertinence to the study's objectives. The computation of this index is based on data sourced from reputable organizations, including BeFree and PsyGeNET [46], the latter of which has undergone expert validation to certify its reliability. It is possible to express it as follows:

$$EI = \frac{N_{pubs_{positive}}}{N_{pubs_{total}}} \tag{5.1}$$

Where:

$N_{pubs_{positive}}$ focuses only on publications that directly support the gene-disease association.

$N_{pubs_{total}}$ considers all available publications that offer some type of support for that specific association, regardless of whether they are positive or not.

An EI value of 1 signifies that all publications provide support for the association between the gene variant and the disease, whereas an EI value less than 1 indicates that some publications assert the lack of association between the gene/variant and the disease.

5.2.1 Data Analysis

5.2.1.1 With the Complete Training Set

The meticulous analysis of the training, validation, and test datasets unveils pivotal insights into the interaction between diseases and genes. A wide spectrum of information is observed, delineated by the count of unique diseases, distinct genes, and unique disease-gene pairs in each set (Table 5.1).

The training set encompasses a total of 7101 distinct diseases and 19394 unique genes, forming a staggering count of 431289 unique disease-gene pairs. This extensive diversity implies a wealth of potential relationships between genetic factors and associated diseases, laying a robust foundation for model construction and training.

Conversely, the validation set comprises 3451 diseases, 11700 genes, and 53437 unique disease-gene pairs. This reduction means a more focused and specific sample compared to the training set while still retaining a substantial volume of information essential for validation and model refinement.

The test set displays numbers akin to the validation data, featuring 4889 diseases, 11912 genes, and 56376 unique disease-gene pairs. This coherence fortifies the reliability of the gathered information and its replicability across diverse datasets.

By employing the method of dividing the data into training, validation, and test sets, previously presented in the Section 5.2, it becomes apparent that approximately 79.72% of the data corresponds to training data, around 9.87% of the total data was allocated for the validation set, and roughly 10.41% was designated for the test set.

Table 5.1: Summary of Unique Diseases, Genes, and Pairs in Datasets

Dataset	Nº of unique Diseases	Nº of unique Genes	Nº of unique pairs (Disease, Gene)
Train	7101	19394	431289
Validation	3451	11700	53437
Test	4889	11912	56376

The Table 5.2 encapsulates descriptive statistics pertaining to the level of evidence ('EI') across the training, validation, and test datasets. Remarkably, a consistent pattern emerges across all datasets, showcasing a mean level of evidence hovering around 0.97, accompanied by relatively low standard deviations, denoting a concentrated distribution around this mean.

Moreover, the uniformity is strikingly apparent in the minimum, Q_1, median (Q_2), and Q_3, and maximum values, all consistently recorded as either 0 or 1. This suggests that the majority of

observations are situated at the value of 1, while a minor proportion may be attributed to the value 0.

In the context of this analysis, these statistics serve as pivotal indicators revealing the distributional characteristics and variability inherent in the evidence level across distinct datasets. Elucidating the significance of this metric within the specific domain of study, this data enriches the understanding and aids in the nuanced interpretation essential to this research endeavor.

Upon comparing the statistics of the complete data (Table 4.5) with those of the individual training, validation, and test sets (Table 5.2), a striking resemblance in the descriptive values of the evidence level ('EI') becomes evident. The mean (0.973373) of the complete dataset closely mirrors that of the training (0.973450), validation (0.972727), and test (0.973402) sets. Furthermore, the standard deviation values are highly proximate across all samples, implying a consistent spread of the data.

Notably, related to the individual sets, the complete dataset also exhibits minimum, Q_1, median (Q_2), and Q_3, and maximum values consistently positioned at 0 and 1. This bolsters a predominant concentration of observations at the value 1, while a smaller fraction aligns with the value 0.

This statistical parallelism between the complete dataset and the individual sets confirms the consistency of the evidence level, indicating uniformity in the distribution and characteristics of 'EI' across all analyzed sets. Such uniformity is pivotal in ensuring the reliability and representativeness of the data utilized in this study. Thereby, strengthening the groundwork for in-depth analyses and conclusions regarding the predictive capacity of the model.

Table 5.2: Statistics for the level of evidence ('EI') for Training, Validation and Test datasets

Dataset	**Mean ($\bar{x}$)**	**Standard deviation (σ)**	**Minimum**	**25%**	**50%**	**75%**	**Maximum**
Train	0.973450	0.138738	0.00	1.00	1.00	1.00	1.00
Validation	0.972727	0.140171	0.00	1.00	1.00	1.00	1.00
Test	0.973402	0.139193	0.00	1.00	1.00	1.00	1.00

5.2.1.2 With the Training Set Excluding Unique Pairs

The meticulous analysis of the training, validation, and test datasets unveils pivotal insights into the interaction between diseases and genes. A wide spectrum of information is observed, delineated by the count of unique diseases, distinct genes, and unique disease-gene pairs in each set (Table 5.3).

Before being partitioned into training, validation and test sets, the data (referred to as *All data*) presents a total of 4889 distinct diseases and 20015 unique genes, forming a staggering count of 538890 unique disease-gene pairs.

The training set encompasses a total of 4889 distinct diseases and 19377 unique genes, forming a staggering count of 429077 unique disease-gene pairs. This extensive diversity implies a wealth of potential relationships between genetic factors and associated diseases, laying a robust foundation for model construction and training.

Conversely, the validation set comprises 3451 diseases, 11700 genes, and 53437 unique disease-gene pairs. This reduction means a more focused and specific sample compared to the training set while still retaining a substantial volume of information essential for validation and model refinement.

The test set displays numbers akin to the validation data, featuring 4889 diseases, 11912 genes, and 56376 unique disease-gene pairs. This coherence fortifies the reliability of the gathered information and its replicability across diverse datasets.

By employing the method of dividing the data into training, validation, and test sets, previously presented in the Section 5.2, it becomes apparent that approximately 79.62% of the data corresponds to training data, around 9.91% of the total data was allocated for the validation set, and roughly 10.47% was designated for the test set.

Table 5.3: Summary of Unique Diseases, Genes, and Pairs in Datasets

Dataset	**Nº of unique Diseases**	**Nº of unique Genes**	**Nº of unique pairs (Disease, Gene)**
Train	4889	19377	429077
Validation	3451	11700	53437
Test	4889	11912	56376
All data	4889	20015	538890

Table 5.4 encapsulates descriptive statistics pertaining to the level of evidence ('EI') across the training, validation, and test datasets and from data before partition, as well as for the entire dataset before partitioning (*All data*). Remarkably, a consistent pattern emerges across all datasets, showcasing a mean level of evidence hovering around 0.97, accompanied by relatively low standard deviations, denoting a concentrated distribution around this mean.

Moreover, the uniformity is strikingly apparent in the minimum, Q_1, median (Q_2), and Q_3, and maximum values, all consistently recorded as either 0 or 1. This suggests that the majority of observations are situated at the value of 1, while a minor proportion may be attributed to the value 0.

In the context of this analysis, these statistics serve as pivotal indicators revealing the distributional characteristics and variability inherent in the evidence level across distinct datasets. Elucidating the significance of this metric within the specific domain of study, this data enriches the understanding and aids in the nuanced interpretation essential to this research endeavor.

Upon comparing the statistics of the complete data (*All data*) with those of the individual training, validation, and test sets (Table 5.4), a striking resemblance in the descriptive values of the evidence level ('EI') becomes evident. The mean (0.973284) of the complete dataset closely mirrors that of the training (0.973338), validation (0.972727), and test (0.973402) sets. Furthermore, the standard deviation values are highly proximate across all samples, implying a consistent spread of the data.

Notably, related to the individual sets, the complete dataset also exhibits minimum, Q_1, median (Q_2), and Q_3, and maximum values consistently positioned at 0 and 1. This bolsters a predominant concentration of observations at the value 1, while a smaller fraction aligns with the value 0.

This statistical parallelism between the complete dataset and the individual sets confirms the consistency of the evidence level, indicating uniformity in the distribution and characteristics of 'EI' across all analyzed sets. Such uniformity is pivotal in ensuring the reliability and representativeness of the data utilized in this study. Thereby, strengthening the groundwork for in-depth analyses and conclusions regarding the predictive capacity of the model.

Table 5.4: Statistics for the level of evidence ('EI') for All data, Training, Validation and Test datasets

Dataset	Mean ($\bar{x}$)	Standard deviation (σ)	Minimum	25%	50%	75%	Maximum
Train	0.973338	0.139007	0.00	1.00	1.00	1.00	1.00
Validation	0.972727	0.140171	0.00	1.00	1.00	1.00	1.00
Test	0.973402	0.139193	0.00	1.00	1.00	1.00	1.00
All data	0.973284	0.139142	0.00	1.00	1.00	1.00	1.00

5.3 Collaborative filtering model

The objective of this study is to develop a recommendation system that identifies genes associated with diseases. In this regard, the primary focus is to create a model following an architecture similar to the approach of Neural Collaborative Filtering. This model aims to predict the scientific evidence level associated with a specific pair comprising a disease and a gene.

To achieve this, PyTorch was utilized to design a model named `RecSysModel`. This model is engineered to accept disease and gene indices, convert them into dense representations (embeddings), and forecast the association between them by generating a continuous value (Listing 5.3).

Listing 5.3: RecSysModel class in Python.

```
class RecSysModel(nn.Module):
    def __init__(self, n_diseases, n_genes, n_factors=16):
        super().__init__()
        self.diseases_embed = nn.Embedding(n_diseases, n_factors)
        self.genes_embed = nn.Embedding(n_genes, n_factors)
        self.out = nn.Linear(n_factors*2, 1)

    def forward(self, diseases, genes):
        diseases_embeds = self.diseases_embed(diseases)
        genes_embeds = self.genes_embed(genes)
        output = torch.cat([diseases_embeds, genes_embeds], dim=1)
        output = self.out(output)
        output = torch.sigmoid(output)
        return output.squeeze()
```

Initialization `__init__`:

- The `RecSysModel` class is defined as a PyTorch model, inheriting from the `nn.Module` class.

- In the `__init__` method, the main components of the model are defined. `n_diseases` and `n_genes` are the numbers of different types of diseases and genes, respectively, used to create embeddings.
- `n_factors` is the size of the vectors of latent factors (density representations) for diseases and genes (standard: 16).
- The model uses two layers of embedding: `diseases_embed` and `genes_embed`, which transform disease and gene indices into dense vectors of size `n_factors`.
- `self.out` is a linear layer that receives the concatenation of the embeddings of diseases and genes (with dimension `n_factors * 2`) and produces a dimension 1 output to predict a continuous value.

Forward method `__forward__`:

- The forward method defines how data flows through the model during inference.
- It receives tensors of disease indices (`diseases`) and genes (`genes`).
- These indices are converted into embeddings through the `diseases_embed` and `genes_embed` layers.
- The embeddings are concatenated into a single tensor (`output`) along dimension 1, because the objective is to predict only one value, the evidence index (`ei`).
- This tensor is passed through the linear layer (`self.out`) to obtain a continuous output.
- The Sigmoid (Equation 5.2) activation function is applied to normalize the output between 0 and 1.
- Finally, the method returns the output, reducing the dimension to a one-dimensional tensor.

Why use the Sigmoid activation function instead of the Rectified Linear Unit (ReLU)?

Activation functions play a fundamental role in neural network by determining the output of a single node based on the signals it receives from connected neurons. These functions introduce nonlinearity, allowing the network to learn complex patterns in data. Additionally, activation functions guide how information is passed to subsequent layers or to the model's final output.

For effective training, activation functions must be differentiable. Differentiability allows for the assessment of how changes in weights impact the output of the neural network, a key aspect of the backpropagation process. This characteristic is essential for gradient-based optimization algorithms, such as Adam [59] or Stochastic Gradient Descent, to adjust the model's parameters in a way that reduces the loss function and enhances performance during training.

The Sigmoid function is a mathematical function that transforms any real number into a value within the range of 0 to 1. It is widely used in neural networks and machine learning, where the goal is to predict probabilities.

When it comes to negative input values far from zero, the Sigmoid function maps them to values close to 0. For instance, -10 or -100, when passed through the Sigmoid function, will result in values very close to 0, but never exactly 0.

On the other hand, positive values far from zero are mapped to values close to 1. Thus, very large positive numbers, such as 10 or 100, when passed through the Sigmoid function, will result in values very close to 1, but again, never exactly 1.

Between these extremes, the Sigmoid function gradually increases around the value 0. In other words, it starts to rise slowly with negative values and continues increasing toward close to 1 with positive values, always maintaining a smooth and continuous behavior. The Sigmoid function (Figure 5.1) is commonly represented as:

$$\sigma : \mathbb{R} \to (0, 1), \quad Sigmoid(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{5.2}$$

Essentially, the Sigmoid function is useful for measuring the positivity or 'activation' of a neuron in a neural network. It transforms the weighted sum of inputs and weights (a common operation in neural networks) into values that represent the probability of belonging to a certain class (if it's a binary classification problem) or the activation level of the neuron.

On the other hand, the ReLU function is an activation function that returns 0 if the input value x is less than or equal to 0, and returns the input value x itself if it's greater than 0. The Rectified Linear Unit (Figure 5.1) function is commonly represented as:

$$\text{ReLU} : \mathbb{R} \to [0, \infty[, \quad \text{ReLU}(x) = \max(0, x) \tag{5.3}$$

The main advantage of adopting ReLU is its computational efficiency, outperforming complex functions, such as the Sigmoid, due to its reliance on a straightforward maximum and comparison process. ReLU not only improves efficiency but also mitigates the issue of 'vanishing gradient'. In functions such as the Sigmoid, gradients can become very small in certain regions, leading to learning stagnation in deep neural networks. ReLU overcomes this problem by maintaining a gradient of 1 for positive values, which guarantees effective gradient propagation and facilitates the training of deeper networks.

Nevertheless, ReLU does have its limitations since it might cause neurons to become inactive or non-responsive throughout the training process. Consequently, neurons with negative values remain inactive and don't change their weights, which may lead to a decline in learning.

The choice of using the Sigmoid activation function instead of ReLU to predict the level of scientific evidence of a gene concerning a specific disease, is grounded in several considerations.

The primary objective is to predict a relevance index ('EI' - Evidence Index) that varies within the range of $[0, 1]$, where a value of 1 signifies that all publications support the association between the genetic variant and the disease. Conversely, a value below 1 indicates that some publications refute the association between the gene/variant and the disease.

The Sigmoid function stands out in this context due to its ability to map values to a specific range (in this case, the interval between 0 and 1). This characteristic is crucial since the relevance

index needs to lie within this interval to reflect the spectrum of scientific support related to the gene's association with the disease.

Furthermore, the Sigmoid function is known for its capability to model probabilities and provide an output on a probability scale. This property is advantageous in expressing the certainty or uncertainty associated with the relationship between the gene and the disease based on the quantity and quality of scientific publications.

The nature of the ReLU function, which returns 0 for negative values, doesn't align directly with the need to express an evidence index that ranges from complete support to the refutation of the association between the gene and the disease. Moreover, it lacks the direct capability to furnish a probability scale within the precise interval necessary for this particular context.

Therefore, considering the necessity to express the strength of the association between the gene and the disease within a range varying from full support to potential refutation, the Sigmoid function stands out as the most appropriate choice for this specific problem.

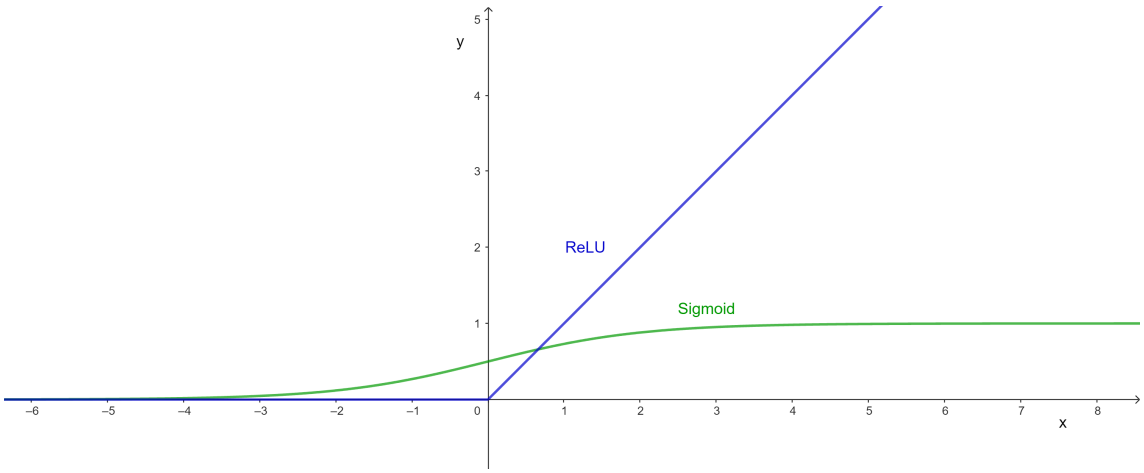


Figure 5.1: Graphs of Sigmoid and ReLU functions

5.3.1 Mathematically, what happens?

Initialization `__init__`:

- `nn.Embedding(n_diseases, n_factors)` and `nn.Embedding(n_genes, n_factors)` create matrices of embeddings for diseases and genes, respectively. Each row of these matrices represents a dense vector of size `n_factors`.
- `nn.Linear(n_factors*2, 1)` creates a linear layer that receives the concatenation of disease and gene embeddings, resulting in a single continuous value. The input dimension is `n_factors*2` because disease and gene embeddings are concatenated before being passed to this layer.
- The `Linear` method applies a linear transformation to the input data, described by the equation:

$$y = xA^T + b \tag{5.4}$$

Where:

y is the output.

x is the input.

A represents the transformation weights.

b is the bias variable.

In PyTorch, the bias is typically initialized as true by default and is commonly initialized from a uniform distribution $\mathcal{U}(-\sqrt{k}, \sqrt{k})$, where $k = \frac{1}{\text{in\_features}}$.

Forward method `__forward__`:

- `diseases_embeds = self.diseases_embed(diseases)` and `genes_embeds = self.genes_embed(genes)` are operations that search the embedding matrices for dense vectors corresponding to disease indices (`diseases`) and genes (`genes`), respectively.
- `output = torch.cat([diseases_embeds, genes_embeds], dim=1)` concatenates the disease and gene embedding vectors into a single tensor along dimension 1.
- `output = self.out(output)` applies the linear layer, multiplying the concatenation of embeddings by the linear layer weight matrix and summing the biases.
- `output = torch.sigmoid(output)` applies the Sigmoid (Equation 5.2) activation function to transform the result into a value between 0 and 1.

A brief example:

Define two-dimensional embeddings for three diseases and four genes ($n\_factors = 2$). For simplicity, let's depict the embedding matrices for diseases and genes, along with the linear layer weight matrix.

Disease Embeddings (`diseases_embed`):

$$\begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \\ 0.5 & 0.6 \end{bmatrix}$$

Genes Embeddings (`genes_embed`):

$$\begin{bmatrix} 0.7 & 0.8 \\ 0.9 & 1.0 \\ 1.1 & 1.2 \\ 1.3 & 1.4 \end{bmatrix}$$

Let D_2 and G_3 be the embeddings for the second disease and the third gene, respectively.

$$D_2 = \begin{bmatrix} 0.3 & 0.4 \end{bmatrix} \quad \text{(Second Disease)}$$

$$G_3 = \begin{bmatrix} 1.1 & 1.2 \end{bmatrix} \quad \text{(Third Gene)}$$

The concatenation of these embeddings (X) is:

$$X = \begin{bmatrix} D_2 & G_3 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.4 & 1.1 & 1.2 \end{bmatrix}$$

Let W represent the weights of the linear layer. These weights are chosen for example purposes and do not represent the actual model weights.

$$W = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}$$

Let b be the initial bias value. For the linear layer that receives the concatenation of disease and gene embeddings, the input dimension ($in\_features$) is $n\_factors \times 2$. In this case, $n\_factors$ is 2, as each embedding is of size two, so $k = \frac{1}{4}$.

Hence, the bias value can be calculated using the formula:

$$b = \mathcal{U}(-\sqrt{k}, \sqrt{k})$$

Substituting k:

$$b = \mathcal{U}(-\sqrt{\frac{1}{4}}, \sqrt{\frac{1}{4}})$$

Numerically, this translates to:

$$b = \mathcal{U}(-0.5, 0.5)$$

The matrix multiplication with the weight matrix W and addition of the bias b is given by:

$$WX + b = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} 0.3 & 0.4 & 1.1 & 1.2 \end{bmatrix} + b$$

Calculating this product:

$$WX + b = (0.1 \times 0.3) + (0.2 \times 0.4) + (0.3 \times 1.1) + (0.4 \times 1.2) + b$$

Now, apply the Sigmoid (Equation 5.2) function to obtain the model output:

$$\sigma(WX + b) = \sigma((0.1 \times 0.3) + (0.2 \times 0.4) + (0.3 \times 1.1) + (0.4 \times 1.2) + b)$$

Numerically:

$$\sigma(0.03 + 0.08 + 0.33 + 0.48 + b) = \sigma(0.92 + b)$$

Considering $b = 0$, the final expression is:

$$\text{output} = \frac{1}{1 + e^{-0.92}} \approx 0.715$$

Weights of linear layers are learnt model parameters throughout the training process. When the linear layer is constructed utilizing `nn.Linear(n_factors*2, 1)`, a random initialization of the weights (weight array) and biases occurs. These weights are adjusted iteratively by the optimization algorithm (such as gradient descent) during training in order to minimize the loss function, or ensure that the model's predictions closely approximate the true values of the training data.

Additionally, the model's performance is evaluated on a separate set of data, called the validation set. This step assesses how well the model generalizes to unseen data. By optimizing the weights in accordance with the gradient of the loss function with respect to these weights, the model is capable of progressively refining its predictions and discerning patterns within the data. These weights are anticipated to accurately represent pertinent patterns in the data and generate predictions upon completion of the training process.

5.3.2 Model Training and Validation

In order to train the proposed model, the `RecSysModel` (Listing 5.3) following an architecture similar to the approach of Neural Collaborative Filtering, was utilized. The architecture underwent initialization using precise parameters, including the quantity of diseases ($n\_diseases$) and genes (n_genes) that have been identified within the dataset. Both genes and diseases have been represented by $16 - dimensional$ embeddings. This number was set to 16 due to the lack of substantial enhancements observed in the model results with alternative values.

Prior to initiating the training and optimization process, the subsequent critical variables were established:

- Loss function: It plays a crucial role by measuring the difference between predicted and actual outputs, guiding the model's parameter adjustments during training. Specifically, `nn.MSELoss()` computes the average squared difference between predictions and ground truth, offering a clear insight into prediction quality. By iteratively minimizing errors, the model fine-tunes its parameters, enhancing its predictive capacity. Choosing an appropriate loss function significantly impacts the model's learning, with `nn.MSELoss()` enabling iterative refinement of predictions, improving alignment with actual data and overall accuracy.

- Optimizer: The optimizer holds a pivotal role in adjusting the model's weights based on the gradient's direction and magnitude, aiming to minimize the loss function specified by the criterion. Its primary goal is to refine the model parameters to best align with the data, fostering more precise and effective learning. The ADAM algorithm, referenced from [59], stood out as the optimal choice, boasting a learning rate (lr) of 0.01. Its chief objective revolves around fine-tuning the model's weights throughout the training phase to minimize the loss function. Widely prevalent in deep learning, Adam's adaptive nature allows for tailored adjustments to the learning rate for each parameter. This adaptability significantly

contributes to expediting the model's convergence, ensuring a more efficient learning process.

- Scheduler: It plays a pivotal role in model training by dynamically adjusting the learning rate throughout the procedure. Utilizing StepLR, the learning rate undergoes reduction by a factor ($gamma$), in this case equal to 0.7, at regular $step\_size$ intervals, in this context, occurring every 1 epoch. This mechanism serves to prevent unwanted fluctuations, ensuring both the stability of the training process and the gradual convergence of the model. The scheduler's function lies in facilitating a controlled decrease in the learning rate, fostering a more consistent and stable convergence of the model across epochs. By regulating the learning rate, it encourages a smoother and more reliable convergence, ultimately enhancing the model's performance.

The model underwent training for a maximum of seven epochs (iterations), during which losses were calculated for both the training and validation sets. Continuously monitoring this process was pivotal to comprehending the model's progressive development. Notably, to optimize the neural model's training and performance, datasets were pre-processed and partitioned into batches to optimize the neural model's training. This batching strategy facilitated efficient data processing by allowing the model to focus on limited sets of samples concurrently. Following the processing of each batch, individual losses were computed and accumulated. At the end of each epoch, representing a complete cycle through the dataset, the average losses from all batches were computed. This aggregated metric provided an overview of the model's performance across the entire dataset, yielding the average loss observed throughout the training and validation phases.

Throughout the training process, performance metrics including loss and Root Mean Squared Error (RMSE) were diligently computed to assess the model's performance across both the training and validation datasets. This continuous assessment was instrumental in understanding how well the model performed across diverse datasets (training and validation), providing insights into its generalization capabilities.

To ensure computational efficiency, GPU memory was systematically released after each iteration. This approach was pivotal in maximizing the utilization of available resources while preemptively preventing potential memory constraints or bottlenecks.

An early stopping mechanism was implemented to halt training if there's no improvement in validation loss after a designated number of epochs, set at 1 in this case. The implementation of this approach is critical in mitigating the model's tendency to excessively adapt to the training data, thus ensuring its capacity to generalize efficiently. This practice not only contributes to the robustness of the model, but also saves computational resources by stopping training as soon as there is no additional benefit.

After every epoch, the scheduler was updated to dynamically adjust the learning rate. This approach facilitated a seamless adaptation of the learning rate throughout the training process, significantly aiding in optimizing the model's convergence.

5.3.2.1 With the Complete Training Set

The results obtained were the following:

Listing 5.4: Results from the process of training and validating the model

```
Epoch [1/7], Train Loss: 0.0199, Validation Loss: 0.0199,
Train RMSE: 0.0706, Validation RMSE: 0.0708

Epoch [2/7], Train Loss: 0.0194, Validation Loss: 0.0201,
Train RMSE: 0.0672, Validation RMSE: 0.0668

Early stopping at epoch 2 as there is no improvement in
validation loss.
```

Examining the outcomes (Listing 5.4) acquired throughout the training process of the model provides vital insights into its efficacy and ability to generalize. The training and validation loss measurements, denoted as Train Loss and Validation Loss, respectively, exhibited an initially promising trend of convergence in the first epochs, considering that the Training Loss value reduces as the number of epochs increases. However, from the second epoch onwards, an increase of 0.0002 in Validation Loss is observed, indicating a possible difficulty for the model in generalizing to unobserved data. Consequently, the training was halted in the second phase using the Early Stopping strategy. This early termination of training was used to prevent potential overfitting of the model to the training data, while maintaining its ability to generalize. Hence, the pivotal choice to implement Early Stopping at this juncture was imperative in order to prevent the model from being too specialized to the training data, enhancing its capacity to generalize to novel data sets.

It is noteworthy that, prior to arriving at this particular architecture and parameter choices, various approaches were explored. These included experimenting with the L2 regularization technique [68], introducing hidden layers, using different values in the optimizer, the scheduler, the number of factors, and early stopping parameters. However, none of these alternatives demonstrated substantial improvements when compared with the results obtained with the presented model. Consequently, these methodologies were excluded from the final model due to their limited contribution to enhancing its performance or generalization capacity. Therefore, the simplicity of the presented model proved to be the most effective and efficient choice for achieving the desired outcome.

Upon analyzing the RMSE values, is it possible to observe a positive trend in both measures for both training and validation data. The training Root Mean Square Error (RMSE) reduced by 0.0034 across epochs, achieving a value of 0.0672. Similarly, the validation RMSE fell to 0.0668, reflecting a positive change of 0.004. The reduction in both training and validation RMSE is a positive sign of the model's capacity to fit the data and make predictions on unseen data sets. The similarity of the RMSE values across the training and validation sets indicates a consistent ability to make accurate predictions across various data sets.

Furthermore, at completion of this procedure, the model was stored for future utilization. By employing this method, the model's state is conserved post-training, hence eliminating the ne-

cessity of restarting training from the beginning. This optimizes time and computing resources, enabling effortless access and utilization of the model for subsequent predictions or ongoing training with fresh data, while preserving the progress achieved in earlier training.

At the end, the model's recommendations for the test set can be printed and saved in a CSV file. Employing a label decoding procedure preserves the values associated with the initial database data in these files. This decoding reverses the label encoding applied by the LabelEncoder, ensuring that the values in the CSV file accurately mirror the original values in the database.

5.3.2.2 With the Training Set Excluding Unique Pairs

The results obtained were the following:

Listing 5.5: Results from the process of training and validating the model

```
Epoch [1/7], Train Loss: 0.0201, Validation Loss: 0.0203,
Train RMSE: 0.0715, Validation RMSE: 0.0711

Epoch [2/7], Train Loss: 0.0194, Validation Loss: 0.0206,
Train RMSE: 0.0665, Validation RMSE: 0.0664

Early stopping at epoch 2 as there is no improvement in
validation loss.
```

Examining the outcomes (Listing 5.5) acquired throughout the training process of the model provides vital insights into its efficacy and ability to generalize. The training and validation loss measurements, denoted as Train Loss and Validation Loss, respectively, exhibited an initially promising trend of convergence in the first epochs, considering that the Training Loss value reduces as the number of epochs increases. However, from the second epoch onwards, an increase of 0.0003 in Validation Loss is observed, indicating a possible difficulty for the model in generalizing to unobserved data. Consequently, the training was halted in the second phase using the Early Stopping strategy. This early termination of training was used to prevent potential overfitting of the model to the training data, while maintaining its ability to generalize. Hence, the pivotal choice to implement Early Stopping at this juncture was imperative in order to prevent the model from being too specialized to the training data, enhancing its capacity to generalize to novel data sets.

It is noteworthy that, prior to arriving at this particular architecture and parameter choices, various approaches were explored. These included experimenting with the L2 regularization technique [68], introducing hidden layers, using different values in the optimizer, the scheduler, the number of factors, and early stopping parameters. However, none of these alternatives demonstrated substantial improvements when compared with the results obtained with the presented model. Consequently, these methodologies were excluded from the final model due to their limited contribution to enhancing its performance or generalization capacity. Therefore, the simplicity of the presented model proved to be the most effective and efficient choice for achieving the desired outcome.

Upon analyzing the RMSE values, is it possible to observe a positive trend in both measures

for both training and validation data. The training Root Mean Square Error (RMSE) reduced by 0.005 across epochs, achieving a value of 0.0665. Similarly, the validation RMSE fell to 0.0664, reflecting a positive change of 0.0047. The reduction in both training and validation RMSE is a positive sign of the model's capacity to fit the data and make predictions on unseen data sets. The similarity of the RMSE values across the training and validation sets indicates a consistent ability to make accurate predictions across various data sets.

Furthermore, at completion of this procedure, the model was stored for future utilization. By employing this method, the model's state is conserved post-training, hence eliminating the necessity of restarting training from the beginning. This optimizes time and computing resources, enabling effortless access and utilization of the model for subsequent predictions or ongoing training with fresh data, while preserving the progress achieved in earlier training.

At the end, the model's recommendations for the test set can be printed and saved in a CSV file. Employing a label decoding procedure preserves the values associated with the initial database data in these files. This decoding reverses the label encoding applied by the LabelEncoder, ensuring that the values in the CSV file accurately mirror the original values in the database.

6
Results and Discussion

The present chapter explores the assessment of models that have been developed from two separate training sets, providing valuable insights into their performance when applied to the test set. The present research conducts a thorough analysis to evaluate the efficacy of these models in comparison to a well-established k-Nearest Neighbors (kNN) model across a range of metrics. Furthermore, this chapter elucidates the complexities behind the suggestions offered forward by the system, providing insight into certain genes that have been recognized as significant for distinct disorders. By exploring these features in more depth, a thorough comprehension of the advantages and constraints of each model is presented, providing important perspectives on their practical use and potential for future improvement.

6.1 Evaluation of the model trained with the Complete Training Set

Following the completion of model training, an assessment of performance was carried out using the test set. This evaluation was based on fundamental metrics that gauge the precision and efficacy of the model's predictions. The metrics used were Root Mean Squared Error (RMSE), Precision, and Recall, employing the approaches of $Precision@k$ and $Recall@k$.

The Root Mean Squared Error (RMSE) (Equation 2.8) is a crucial metric for evaluating models. It measures the quadratic mean of the differences between the predicted values of the model and actual values. In this case, the RMSE value on the test set was 0.06129063179340693. A reduced Root Mean Square Error (RMSE) is very desirable as it signifies a diminished average discrepancy between the predictions made by the model and the actual values. This indicates that the model has a tendency to provide more precise predictions, closely resembling the actual values in the test dataset. Moreover, a low RMSE value serves as a favorable indication of the model's

ability to generalize, meaning its proficiency in utilizing acquired knowledge from training to accurately forecast outcomes on unseen data. The model's flexibility and capacity to produce correct predictions in unfamiliar settings or data not seen during training are essential aspects.

The decreased disparity between forecasts and observed values enhances the model's interpretability and comprehensibility, as its predictions exhibit more consistency with actual data. To summarize, a smaller Root Mean Square Error (RMSE) indicates that the model has a greater capacity to create precise and consistent predictions with actual values. This is highly regarded when evaluating the model's quality and dependability. Therefore, the resulting value for this measure clearly demonstrates that the model is appropriate for predicting the level of scientific evidence linking a disease and a specific gene.

In addition to RMSE, it is crucial to consider metrics such as $Precision@k$ (Equation 2.9) and $Recall@k$ (Equation 2.10) when evaluating the effectiveness of a recommendation system. While RMSE quantifies the quadratic mean of the differences between the predicted values of the model and actual values, $Precision@k$ assesses the quality of the recommendations at the top of the list as it helps measure how well the system performs in terms of providing relevant items early in the list of recommendations produced by the model, and $Recall@k$ measures the proportion of relevant items found in the *top-k* recommendations. These metrics are essential for evaluating the quality of recommendations, especially in systems where relevance and precision are paramount, as is the case in recommending genes for diseases. Thus, the model was assessed using the metrics $Precision@k$ and $Recall@k$, where k was defined as the set $k = \{1, 2, 3, 4, 5\}$.

It is crucial to acknowledge that the objective of these metrics is to determine the relevance[1] of a gene to a particular disease. A threshold [48] value of 0.973373 was determined for this particular context, which corresponds to the mean scientific evidence index of the aggregated entire data set (Table 4.5). Therefore, in the process of assessing the predictive efficacy of the model for a specific disease-gene pair, the gene will be declared irrelevant if the predicted level of scientific evidence falls below the predetermined threshold. Conversely, in the event that the value predicted by the model is within or above the predetermined threshold, the gene will be declared relevant to the specific disease under consideration. The utilization of this relevance criterion offers a lucid methodology to interpret and classify the importance of genes in relation to the disease that is being studied.

Analyzing Table 6.1 and Figure 6.1, along with the metrics for various k values, leads to several conclusions.

Table 6.1: Metrics for different values of k

Metric	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$Precision@k$	0.8321	0.9485	0.9588	0.9567	0.9540
$Recall@k$	0.4231	0.4301	0.4331	0.4450	0.4517

[1] `https://surprise.readthedocs.io/en/latest/FAQ.html`

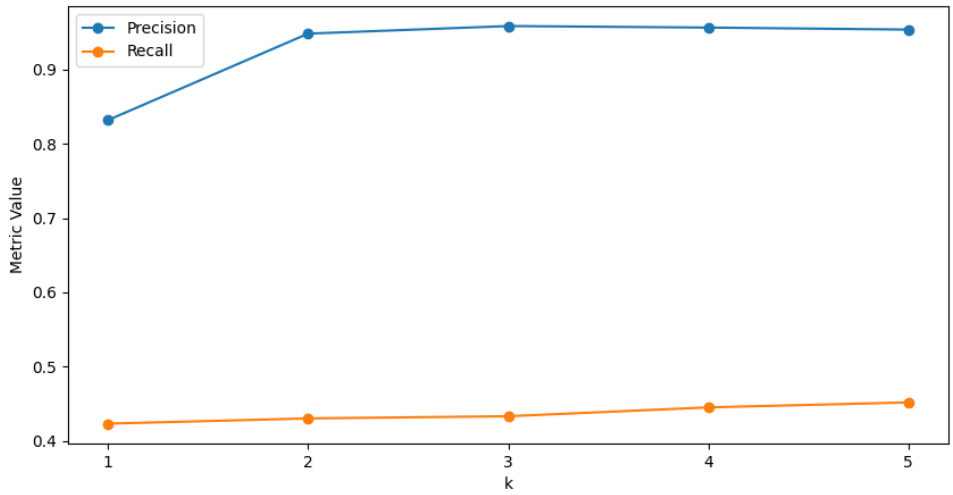


Figure 6.1: Metrics for different values of k

Regarding $Precision@k$, the values in the table for different k (from 1 to 5) emphasize the precision dynamics as the number of recommended items varies, maintaining a value around 0.83 and 0.96. Remarkably, it is interesting to observe that precision consistently remains high, indicating that even with the expansion of the recommendation set, a significant proportion remains correct. However, starting from $k = 3$, a slight downward trend is noticeable. This decline suggests that, with the inclusion of more items in the recommendation list, there is a modest reduction in the proportion of relevant items. In fact, although this decrease is relatively minor, considering the proximity of the values to each other and the fact that the values remain high (above 95%), it indicates that the majority of recommendations are accurate.

In the context of $Recall@k$, a continuous upward trend is observed as k increases, maintaining within a range of 0.43 to 0.46. At the initial point, $k = 1$, recall is recorded at 0.4231, indicating that only 42.31% of relevant items were retrieved in the first recommendation. This gradual growth trend reaches its peak at $k = 5$. The consistent evolution of recall suggests that, as the number of items included in the recommendations increases, there is a progressive ability to retrieve all relevant items. This overall increase in recall reflects a continuous advancement as the recommendation system expands its suggestions, indicating ongoing development in ensuring the inclusion of all truly relevant items.

In contextualizing the analysis of the $Precision@k$ and $Recall@k$ metrics within the scope of predicting the level of scientific evidence for a gene related to a specific disease, $Precision@k$ stands out as a fundamental metric as it reflects the proportion of recommended genes in the *top-k* set that are relevant for a specific disease, taking into account the level of scientific evidence between the recommended gene and the disease. A high $Precision@k$, evidenced by values close to 1, indicates that the vast majority of genes recommended in the *top-k* are highly relevant, demonstrating a significant level of scientific evidence associated with the respective disease.

This observation suggests that the developed recommendation system is capable of identifying

and suggesting genes with a high level of scientific evidence associated with a specific disease. In other words, the genes recommended by the system are highly worthy of consideration concerning their relationship with the investigated disease. This enhances confidence in the system's ability to direct attention to genes that have a substantial probability of being associated with the pathology in question, highlighting the practical utility and effectiveness of the recommendation system in the context of biomedical research and the identification of disease candidate genes.

On the other hand, $Recall@k$ assesses the recommendation system's ability to capture all relevant genes with a high level of scientific evidence among the *top-k* recommendations. An overall increase in recall is observed, signifying that as the recommendation system expands its suggestions, it ensures the inclusion of all genes that are genuinely relevant. A balanced $Recall@k$, as is the case, indicates the system's capability to capture relevant genes with a high level of evidence, suggesting improvement in the identification of crucial genes as the recommendation list expands.

While a high $Recall@k$ is desirable to ensure that no relevant gene is missed in the recommendations, it is important to consider the balance between recall and precision. A higher recall may lead to more recommended relevant genes but could include some with lower scientific evidence, affecting precision. From the observed data, it is clear that there is a delicate equilibrium between providing precise recommendations with high precision and thoroughly identifying all pertinent genes with recall.

The meticulous analysis of all employed metrics confirms the robustness of the developed recommendation system in selecting candidate genes. This strength enhances its practical utility, broadening the capability to identify promising genes with potential applications in biomedical research and clinical practices. Consequently, the system makes a significant contribution to the advancement and understanding of genomics associated with specific diseases, standing out as a valuable tool in identifying associations between genes and diseases, driving relevant progress in this field.

It is noteworthy that the computation of precision and recall does not include values when the number of samples in the test set is less than the value of k (when $n\_samples \leq k$). In other words, if a disease has only two genes in the test set, precision and recall will not be calculated for the values of $k = \{3, 4, 5\}$, so as not to influence these metrics.

6.1.1 Proposed model vs k-Nearest Neighbors algorithm

In order to evaluate the model's efficacy, the outcomes were examined through the implementation of the k-Nearest Neighbors (k-NN) algorithm. Thus, for an equal-footing comparison, the training, validation, and test datasets utilized for both models were identical. Utilizing the Surprise[2] library, the model was constructed. The following model parameters were obtained through parameter optimization using the validation set: k is set to 20, $similarity\_measure$ is set to mean squared difference (msd), and $user\_based$ is set to $True$.

The k-Nearest Neighbors (k-NN) algorithm is a collaborative filtering technique that generates

[2] `https://surpriselib.com/`

predictions by calculating the similarity between items or users, taking into account their proximity. Nevertheless, k-Nearest Neighbors may face difficulties when confronted with extremely sparse datasets in which the majority of inputs are unknown, as is frequently the situation. Moreover, k-Nearest Neighbors necessitates ongoing computation of the similarity matrix, a process that can be computationally intensive and impede scalability, especially when new data is added.

On the contrary, the methodology employed by the suggested model, which is founded upon neural networks, is unique and is employed to discover complex patterns in recommendation data. In contrast to sparse datasets, the robustness of this model is improved as a result of the capacity of neural networks to acquire latent representations even when comprehensive information is lacking. Furthermore, in comparison to sparser and larger datasets, its capacity for generalization exceeds k-Nearest Neighbors, enabling the identification of more intricate and subtle patterns as well as the scalability of models — exactly the goals that recommendation systems strive to accomplish.

The findings suggest that the `RecSysModel` (Listing 5.3) demonstrated a higher level of confidence in forecasting correlations between candidate genes and diseases in comparison to the k-Nearest Neighbors model. The aforementioned assertion is corroborated by the assessment metric called Root Mean Squared Error (RMSE).

The Root Mean Square Error (RMSE) quantifies the precision of a model's forecasts in relation to the true values. As RMSE decreases, the model's fit to the observed data improves. Within the given context, the `RecSysModel` attained a Root Mean Square Error (RMSE) of 0.0613, whereas the k-Nearest Neighbors model exhibited a marginally higher RMSE of 0.1411.

The disparity in Root Mean Square Error (RMSE) values indicates that the `RecSysModel` exhibits superior predictive performance in proximity to the true values when compared to the k-Nearest Neighbors model. In other words, the `RecSysModel` demonstrated a higher degree of consistency between its predictions and the observed outcomes, which signifies an enhanced capability to forecast correlations between candidate genes and diseases.

This superiority of `RecSysModel` can be attributed to its ability to use neural networks to learn complex patterns in recommendation data, allowing it to better handle sparse datasets and learn latent representations even in the absence of information.

Therefore, based on the RMSE results, `RecSysModel` emerges as a more promising choice for the specific problem of predicting associations between candidate genes and diseases compared to k-Nearest Neighbors.

Both algorithms exhibit competitive performance in terms of $Precision@k$ and $Recall@k$ (Table 6.2 and Figure 6.2). Nevertheless, it is worth mentioning that `RecSysModel` demonstrates a marginal superiority in terms of precision, particularly in situations when the value of k is lower. These findings indicate that `RecSysModel` tends to provide a slightly greater percentage of correct suggestions compared to actual gene-disease connections when dealing with a smaller range of recommendations. The capacity to precisely detect noteworthy correlations between genes and illnesses is especially advantageous in scenarios when the precision of recommendations is crucial, such as when just one alternative is being suggested.

In terms of recall, the `RecSysModel` model exhibits a progressive rise as the value of k grows, but a different trend is observed in the k-Nearest Neighbors when k is equal to 2. Nevertheless, it is worth mentioning that `RecSysModel` routinely demonstrates superior performance compared to the k-Nearest Neighbors method in terms of recall. This suggests that as the number of suggested genes increases, `RecSysModel` has a comparatively higher capacity to accurately identify a growing proportion of genuine gene-disease connections in comparison to k-Nearest Neighbors.

Overall, when evaluating measures such as RMSE, precision, and recall, `RecSysModel` proves to be a more robust and efficient choice for the particular task of forecasting connections between genes and illnesses, relying on scientific evidence, and suggesting potential genes.

Table 6.2: Metrics for Different Values of k and models

Metric	Algorithm	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$Precision@k$	RecSysModel	**0.8321**	**0.9485**	0.9588	0.9567	0.9540
	k-NN	0.8178	0.9368	**0.9606**	**0.9636**	**0.9596**
$Recall@k$	RecSysModel	**0.4231**	**0.4301**	**0.4331**	**0.4450**	**0.4517**
	k-NN	0.4134	0.3807	0.3853	0.3954	0.4054

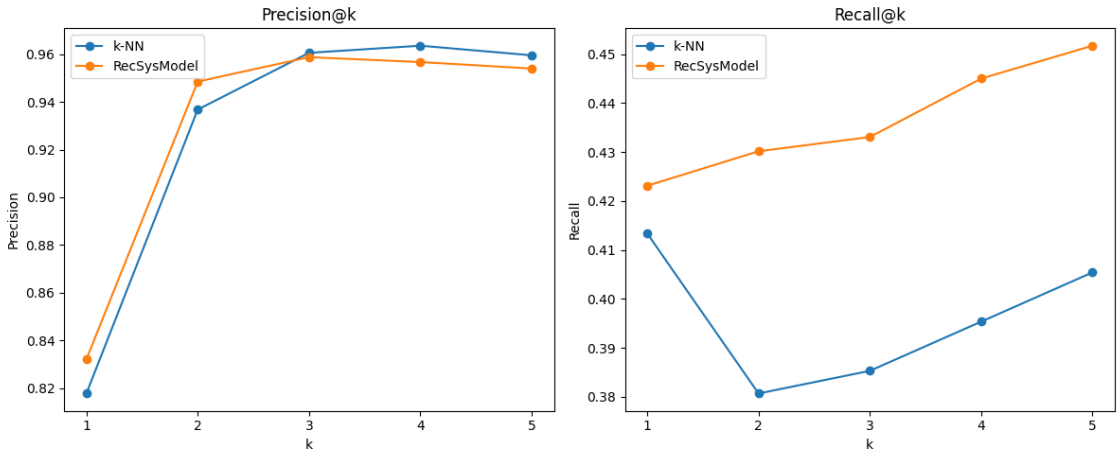


Figure 6.2: Metrics for Different Values of k and models

6.1.2 Examples of candidate gene recommendations

During this pivotal phase in the advancement of genetic recommendation systems, the evaluation of `RecSysModel` (Listing 5.3) recommendations emerges as a critical element. Through the examination of distinct diseases and the genes suggested by the model, a comprehensive understanding of potential genetic correlations in specific health conditions can be obtained. In addition to offering insight into potential genetic factors that may underlie a disease, these recommendations serve as a guide for future therapeutic developments and research.

The significance of this evaluation resides in the recommendation system's capability to filter and emphasize genes that might otherwise remain undetected amidst extensive collections of

genomic data, with the objective of suggesting candidate genes in accordance with the level of scientific evidence for disease. These suggestions enable the detection of pertinent genes, which in turn promotes the exploration of therapeutic targets, comprehension of pathological mechanisms, and customization of medical strategies.

Consequently, the examination of recommendations serves to enhance comprehension while also furnishing a valuable instrument to direct subsequent investigations, clinical progressions, and the formulation of treatment strategies that are more precise and efficacious — in essence, these are the fundamental aims of `RecSysModel` (Listing 5.3). With this in mind, the scientific evidence substantiated by the model will be utilized to evaluate the candidate gene recommendations for four distinct diseases. The findings were organized in Tables 6.3, 6.4, 6.5 and 6.6, which provide a comprehensive synopsis of the genetic associations that have been emphasized. Every entry into tables comprises the following information: the gene identification number (Gene ID), the official symbol denoting the gene (Official symbol), the name of the gene (Gene Name), the model's prediction for the level of evidence, and the actual value (Real value) that corresponds to the level of scientific evidence.

Gestational Diabetes

Gestational diabetes is a type of diabetes distinguished by hyperglycemia that manifests during pregnancy [89]. This condition develops when the body is unable to adequately produce insulin to fulfill the heightened demands, thereby posing potential risks for the mother and the baby. The categorization of gestational diabetes as a phenotype holds paramount significance in scientific investigations into the fundamental genetic mechanisms underlying this condition (DOID:11714). The Table 6.3 lists the five principal genes that the `RecSysModel` (Listing 5.3) has identified, emphasizing their significance according to real-world predictions and values.

The PROK1 gene, which exhibits the highest prediction score of 0.9991, demonstrates a robust correlation with Gestational Diabetes, implying a potential involvement in the regulation of biological processes associated with this condition [37]. In a similar vein, EBAG9 is recognized as an additional highly significant candidate, with a score of 0.9991, suggesting a potential association with the hormonal mechanisms involved in Gestational Diabetes.

The AMH gene, highlighted with a score of 0.9990, suggests a potential role in hormonal regulation and a possible influence on the condition. Additionally, S100A9 (prediction score of 0.9989) is pointed out as a possible relevant marker, indicating association with inflammatory processes. Based on its score of 0.9987, the MSC gene is identified as a potential candidate that may have an effect on Gestational Diabetes. This suggests that the gene may have an influence on processes related to cellular differentiation and development.

The remarkable correspondence between predicted values from the `RecSysModel` (Listing 5.3) and the observed values enhances the trustworthiness of these genes in relation to Gestational Diabetes, thereby establishing a robust foundation for subsequent investigations. A thorough comprehension of these genes is essential for the development of more efficacious therapeu-

tic strategies, thereby enhancing the overall understanding of the genetic components underlying the condition and providing valuable insights for investigations pertaining to maternal and infant health.

Table 6.3: Top-5 genes recommended by the model for the disease Gestational Diabetes (DOID:11714).

k	Gene ID	Official symbol	Gene Name	Prediction	Real value
1	84432	PROK1	Prokineticin 1	0.9991	1.0
2	9166	EBAG9	estrogen receptor binding site associated antigen 9	0.9991	1.0
3	268	AMH	anti-Mullerian hormone	0.9990	1.0
4	6280	S100A9	S100 calcium binding protein A9	0.9989	1.0
5	9242	MSC	musculin	0.9987	1.0

Apert syndrome

The genetic analysis of Apert Syndrome (DOID:12960) through the `RecSysModel` (Listing 5.3) reveals valuable information about this rare genetic condition, known for its craniofacial and skeletal anomalies [96].

Among the main genes recommended by the `RecSysModel` (Listing 5.3), the spotlight is on RAB18, a member of the RAS oncogene family, and CCND1, associated with cell cycle regulation. Gene RAB18 receives a significant prediction score of 0.9956, indicating a strong association with Apert Syndrome. Similarly, CCND1, with a score of 0.9737, is identified as another relevant candidate.

Despite a smaller number of identified genes compared to Gestational Diabetes, attributed to the limited samples in the test set, the confidence in the predictions remains substantial, as demonstrated in Table 6.4. This suggests a potential connection of these genes with the underlying mechanisms of Apert Syndrome. Understanding the basic mechanisms behind the condition may be significantly helped by the expression of these genes, which also offer interesting targets for further research and treatment approaches. The high degree of prediction confidence highlights the reliability of these findings and demonstrates the benefit of the model in identifying key genes for uncommon genetic disorders. This strong performance offers a useful method for studying fewer prevalent genetic diseases, for which early identification of important genes is essential to comprehension and treatment.

Table 6.4: Top-5 genes recommended by the model for the disease Apert syndrom (DOID:12960).

k	Gene ID	Official symbol	Gene Name	Prediction	Real value
1	22931	RAB18	RAB18, member RAS oncogene family	0.9956	1.0
2	595	CCND1	cyclin D1	0.9737	1.0

Achondroplasia

Genetic analysis performed by the gene prediction model revealed remarkable results for Achondroplasia (DOID:4480), a rare genetic condition characterized by abnormalities in bone growth [17]. The Table 6.5 presents the five genes that have been identified as the primary candidates by the model. These genes have demonstrated exceptional accuracy in their predictions. The FGF1 gene obtained the highest prediction score of 0.9983, indicating a robust correlation with Achondroplasia. This implies that FGF1 might play a crucial role in the biological processes that are responsible for this disease. The STAT5B gene has been identified as a highly relevant candidate with a prediction score of 0.9957. Their correlation implies potential implications in cellular signaling processes associated with Achondroplasia. The NPPC gene obtained a score of 0.9956, indicating its significant involvement in underlying processes related to the disease. The EVC gene has been noted as a significant candidate, with a prediction score of 0.9944, suggesting potential links with ciliary complexes and their impact on Achondroplasia. The IFT20 gene has been identified as a potential significant contributor to biological processes associated with the disorder, with a score of 0.9922. The strong correlation between the model's predictions and the actual results enhances trust in the reliability of these projections, particularly when considering uncommon genetic illnesses. The genes FGF1, STAT5B, NPPC, EVC, and IFT20, which have been detected, possess promise both as biomarkers and as crucial contributors to the genetic factors that underlie Achondroplasia. The accuracy of these predictions not only confirms the efficacy of the model in uncommon instances, but also establishes a strong foundation for future study and the creation of focused therapy methods. The commendable achievement is promising in terms of tackling rare genetic disorders.

Table 6.5: Top-5 genes recommended by the model for the disease Achondroplasia (DOID:4480).

k	Gene ID	Official symbol	Gene Name	Prediction	Real value
1	2246	FGF1	fibroblast growth factor 1	0.9983	1.0
2	6777	STAT5B	signal transducer and activator of transcription 5B	0.9957	1.0
3	4880	NPPC	natriuretic peptide C	0.9956	1.0
4	2121	EVC	EvC ciliary complex subunit 1	0.9944	1.0
5	90410	IFT20	intraflagellar transport 20	0.9922	1.0

Metabolic Diseases

The assessment of the `RecSysModel` (Listing 5.3) has revealed a remarkable ability to identify candidate genes for Metabolic Diseases (DOID:0014667), a comprehensive grouping of diverse metabolic conditions. Table 6.6 highlights the top five genes recommended by the model, indicating significant performance in identifying genes relevant to this category of diseases.

The gene S100A12 received the highest prediction score, reaching 0.9999, standing out as strongly associated with Metabolic Diseases. With a prediction score of 0.9999, the gene ADRA2B is identified as another highly relevant candidate, its association with adrenergic receptors suggests

possible implications in regulatory processes related to metabolic conditions. The gene TREM2 also scored 0.9999, indicating its strong association with Metabolic Diseases. Its role in myeloid cells suggests potential connections with the immune system in these conditions.

With a prediction score of 0.9998, the gene F7 is identified as a possible key participant in blood clotting processes associated with metabolic diseases. The gene PIN1 also scored 0.9998, highlighting its potential relevance in regulating metabolic processes.

Table 6.6: Top-5 genes recommended by the model for Metabolic Diseases (DOID:0014667).

k	Gene ID	Official symbol	Gene Name	Prediction	Real value
1	6283	S100A12	S100 calcium binding protein A12	0.9999	1.0
2	151	ADRA2B	adrenoceptor alpha 2B	0.9999	1.0
3	54209	TREM2	triggering receptor expressed on myeloid cells 2	0.9999	1.0
4	2155	F7	coagulation factor VII	0.9998	1.0
5	5300	PIN1	peptidylprolyl cis/trans isomerase, NIMA-interacting 1	0.9998	1.0

The incorporation of genes with diverse functions, such as immune system regulation (S100A12, TREM2), and blood coagulation (F7), underscores the holistic approach of the model. This ability to identify genes related to different biological processes is valuable when studying groups of metabolic diseases, which often share underlying pathophysiological mechanisms.

The notable consistency between the model's predictions and actual values reinforces confidence in the utility of these genes as potential targets for future studies. Thus, the developed recommendation system not only provides valuable insights for recommending candidate genes for individualized approaches and rare diseases but also represents a promising recommendation tool for broader investigations into disease groups.

6.2 Evaluation of the model trained with the Training Set Excluding Unique Pairs

Following the completion of model training, an assessment of performance was carried out using the test set. This evaluation was based on fundamental metrics that gauge the precision and efficacy of the model's predictions. The metrics used were Root Mean Squared Error (RMSE), Precision, and Recall, employing the approaches of $Precision@k$ and $Recall@k$.

The Root Mean Squared Error (RMSE) (Equation 2.8) is a crucial metric for evaluating models. It measures the average discrepancy between the predicted values of the model and the actual values. In this case, the RMSE value on the test set was 0.06281747327541579. A reduced Root Mean Square Error (RMSE) is very desirable as it signifies a diminished average discrepancy between the predictions made by the model and the actual values. This indicates that the model has a tendency to provide more precise predictions, closely resembling the actual values in the test dataset. Moreover, a low RMSE value serves as a favorable indication of the model's ability to

generalize, meaning its proficiency in utilizing acquired knowledge from training to accurately forecast outcomes on unseen data. The model's flexibility and capacity to produce correct predictions in unfamiliar settings or data not seen during training are essential aspects.

The decreased disparity between forecasts and observed values enhances the model's interpretability and comprehensibility, as its predictions exhibit more consistency with actual data. To summarize, a smaller Root Mean Square Error (RMSE) indicates that the model has a greater capacity to create precise and consistent predictions with actual values. This is highly regarded when evaluating the model's quality and dependability. Therefore, the resulting value for this measure clearly demonstrates that the model is appropriate for predicting the level of scientific evidence linking a disease and a specific gene.

In addition to RMSE, it is crucial to consider metrics such as $Precision@k$ (Equation 2.9) and $Recall@k$ (Equation 2.10) when evaluating the effectiveness of a recommendation system. While RMSE quantifies the average difference between model predictions and actual values, $Precision@k$ assesses the quality of the recommendations at the top of the list as it helps measure how well the system performs in terms of providing relevant items early in the list of recommendations produced by the model, and $Recall@k$ measures the proportion of relevant items found in the *top-k* recommendations. These metrics are essential for evaluating the quality of recommendations, especially in systems where relevance and precision are paramount, as is the case in recommending genes for diseases. Thus, the model was assessed using the metrics $Precision@k$ and $Recall@k$, where k was defined as the set $k = \{1, 2, 3, 4, 5\}$.

It is crucial to acknowledge that the objective of these metrics is to determine the relevance[3] of a gene to a particular disease. A threshold [48] value of 0.973284 was determined for this particular context, which corresponds to the mean scientific evidence index of the aggregated entire data set (Table 5.4). Therefore, in the process of assessing the predictive efficacy of the model for a specific disease-gene pair, the gene will be declared irrelevant if the predicted level of scientific evidence falls below the predetermined threshold. Conversely, in the event that the value predicted by the model is within or above the predetermined threshold, the gene will be declared relevant to the specific disease under consideration. The utilization of this relevance criterion offers a lucid methodology to interpret and classify the importance of genes in relation to the disease that is being studied.

Analyzing Table 6.7 and Figure 6.3, along with the metrics for various k values, leads to several conclusions.

Table 6.7: Metrics for different values of k

Metric	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$Precision@k$	0.8613	0.9520	0.9574	0.9563	0.9536
$Recall@k$	0.4540	0.4545	0.4466	0.4562	0.4637

[3] https://surprise.readthedocs.io/en/latest/FAQ.html

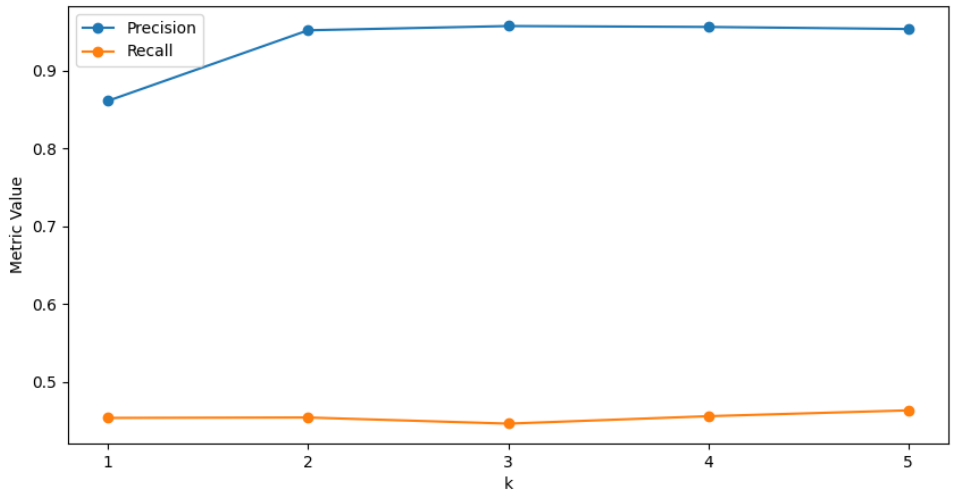


Figure 6.3: Metrics for different values of k

Regarding $Precision@k$, the values in the table for different k (from 1 to 5) emphasize the precision dynamics as the number of recommended items varies, maintaining a value around 0.86 and 0.96. Remarkably, it is interesting to observe that precision consistently remains high, indicating that even with the expansion of the recommendation set, a significant proportion remains correct. However, starting from $k = 3$, a slight downward trend is noticeable. This decline suggests that, with the inclusion of more items in the recommendation list, there is a modest reduction in the proportion of relevant items. In fact, although this decrease is relatively minor, considering the proximity of the values to each other and the fact that the values remain high (above 95%), it indicates that the majority of recommendations are accurate.

In the context of $Recall@k$, a continuous upward trend is observed as k increases, maintaining within a range of 0.45 to 0.46, except when $k = 3$, where there is a decline. At the initial point, $k = 1$, recall is recorded at 0.4540, indicating that only 45.40% of relevant items were retrieved in the first recommendation. This gradual growth trend reaches its peak at $k = 5$. The consistent evolution of recall suggests that, as the number of items included in the recommendations increases, there is a progressive ability to retrieve all relevant items. This overall increase in recall reflects a continuous advancement as the recommendation system expands its suggestions, indicating ongoing development in ensuring the inclusion of all truly relevant items.

In contextualizing the analysis of the $Precision@k$ and $Recall@k$ metrics within the scope of predicting the level of scientific evidence for a gene related to a specific disease, $Precision@k$ stands out as a fundamental metric as it reflects the proportion of recommended genes in the *top-k* set that are relevant for a specific disease, taking into account the level of scientific evidence between the recommended gene and the disease. A high $Precision@k$, evidenced by values close to 1, indicates that the vast majority of genes recommended in the *top-k* are highly relevant, demonstrating a significant level of scientific evidence associated with the respective disease.

This observation suggests that the developed recommendation system is capable of identifying

and suggesting genes with a high level of scientific evidence associated with a specific disease. In other words, the genes recommended by the system are highly worthy of consideration concerning their relationship with the investigated disease. This enhances confidence in the system's ability to direct attention to genes that have a substantial probability of being associated with the pathology in question, highlighting the practical utility and effectiveness of the recommendation system in the context of biomedical research and the identification of disease candidate genes.

On the other hand, $Recall@k$ assesses the recommendation system's ability to capture all relevant genes with a high level of scientific evidence among the *top-k* recommendations. An overall increase in recall is observed (except when $k = 3$, where there is a decline), signifying that as the recommendation system expands its suggestions, it ensures the inclusion of all genes that are genuinely relevant. A balanced $Recall@k$, as is the case, indicates the system's capability to capture relevant genes with a high level of evidence, suggesting improvement in the identification of crucial genes as the recommendation list expands.

While a high $Recall@k$ is desirable to ensure that no relevant gene is missed in the recommendations, it is important to consider the balance between recall and precision. A higher recall may lead to more recommended relevant genes but could include some with lower scientific evidence, affecting precision. From the observed data, it is clear that there is a delicate equilibrium between providing precise recommendations with high precision and thoroughly identifying all pertinent genes with recall.

The meticulous analysis of all employed metrics confirms the robustness of the developed recommendation system in selecting candidate genes. This strength enhances its practical utility, broadening the capability to identify promising genes with potential applications in biomedical research and clinical practices. Consequently, the system makes a significant contribution to the advancement and understanding of genomics associated with specific diseases, standing out as a valuable tool in identifying associations between genes and diseases, driving relevant progress in this field.

It is noteworthy that the computation of precision and recall does not include values when the number of samples in the test set is less than the value of k (when $n\_samples \leq k$). In other words, if a disease has only two genes in the test set, precision and recall will not be calculated for the values of $k = \{3, 4, 5\}$, so as not to influence these metrics.

6.2.1 Proposed model vs k-Nearest Neighbors algorithm

In order to evaluate the model's efficacy, the outcomes were examined through the implementation of the k-Nearest Neighbors (k-NN) algorithm. Thus, for an equal-footing comparison, the training, validation, and test datasets utilized for both models were identical. Utilizing the Surprise[4] library, the model was constructed. The following model parameters were obtained through parameter optimization using the validation set: k is set to 20, $similarity\_measure$ is set to mean squared difference (msd), and $user\_based$ is set to $True$.

[4] `https://surpriselib.com/`

The k-Nearest Neighbors (k-NN) algorithm is a collaborative filtering technique that generates predictions by calculating the similarity between items or users, taking into account their proximity. Nevertheless, k-Nearest Neighbors may face difficulties when confronted with extremely sparse datasets in which the majority of inputs are unknown, as is frequently the situation. Moreover, k-Nearest Neighbors necessitates ongoing computation of the similarity matrix, a process that can be computationally intensive and impede scalability, especially when new data is added.

On the contrary, the methodology employed by the suggested model, which is founded upon neural networks, is unique and is employed to discover complex patterns in recommendation data. In contrast to sparse datasets, the robustness of this model is improved as a result of the capacity of neural networks to acquire latent representations even when comprehensive information is lacking. Furthermore, in comparison to sparser and larger datasets, its capacity for generalization exceeds k-Nearest Neighbors, enabling the identification of more intricate and subtle patterns as well as the scalability of models — exactly the goals that recommendation systems strive to accomplish.

The findings suggest that the `RecSysModel` (Listing 5.3) demonstrated a higher level of confidence in forecasting correlations between candidate genes and diseases in comparison to the k-Nearest Neighbors model. The aforementioned assertion is corroborated by the assessment metric called Root Mean Squared Error (RMSE).

The Root Mean Square Error (RMSE) quantifies the precision of a model's forecasts in relation to the true values. As RMSE decreases, the model's fit to the observed data improves. Within the given context, the `RecSysModel` attained a Root Mean Square Error (RMSE) of 0.0628, whereas the k-Nearest Neighbors model exhibited a marginally higher RMSE of 0.1411.

The disparity in Root Mean Square Error (RMSE) values indicates that the `RecSysModel` exhibits superior predictive performance in proximity to the true values when compared to the k-Nearest Neighbors model. In other words, the `RecSysModel` demonstrated a higher degree of consistency between its predictions and the observed outcomes, which signifies an enhanced capability to forecast correlations between candidate genes and diseases.

This superiority of `RecSysModel` can be attributed to its ability to use neural networks to learn complex patterns in recommendation data, allowing it to better handle sparse datasets and learn latent representations even in the absence of information.

Therefore, based on the RMSE results, `RecSysModel` emerges as a more promising choice for the specific problem of predicting associations between candidate genes and diseases compared to k-Nearest Neighbors.

Both algorithms exhibit competitive performance in terms of $Precision@k$ and $Recall@k$ (Table 6.8 and Figure 6.4). Nevertheless, it is worth mentioning that `RecSysModel` demonstrates a marginal superiority in terms of precision, particularly in situations when the value of k is lower. These findings indicate that `RecSysModel` tends to provide a slightly greater percentage of correct suggestions compared to actual gene-disease connections when dealing with a smaller range of recommendations. The capacity to precisely detect noteworthy correlations between genes and illnesses is especially advantageous in scenarios when the precision of recommendations

is crucial, such as when just one alternative is being suggested.

In terms of recall, the `RecSysModel` model exhibits a progressive rise as the value of k grows (except when $k = 3$, where there is a decline), and a similar trend is seen in the k-Nearest Neighbors model but when k is equal to 2. Nevertheless, it is worth mentioning that `RecSysModel` routinely demonstrates superior performance compared to the k-Nearest Neighbors method in terms of recall. This suggests that as the number of suggested genes increases, `RecSysModel` has a comparatively higher capacity to accurately identify a growing proportion of genuine gene-disease connections in comparison to k-Nearest Neighbors.

Overall, when evaluating measures such as RMSE, precision, and recall, `RecSysModel` proves to be a more robust and efficient choice for the particular task of forecasting connections between genes and illnesses, relying on scientific evidence, and suggesting potential genes.

Table 6.8: Metrics for Different Values of k and models

Metric	Algorithm	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$Precision@k$	RecSysModel	**0.8613**	**0.9520**	0.9574	0.9563	0.9536
	k-NN	0.8407	0.9379	**0.9615**	**0.9637**	**0.9596**
$Recall@k$	RecSysModel	**0.4540**	**0.4545**	**0.4466**	**0.4562**	**0.4637**
	k-NN	0.4359	0.3830	0.3880	0.3976	0.4072

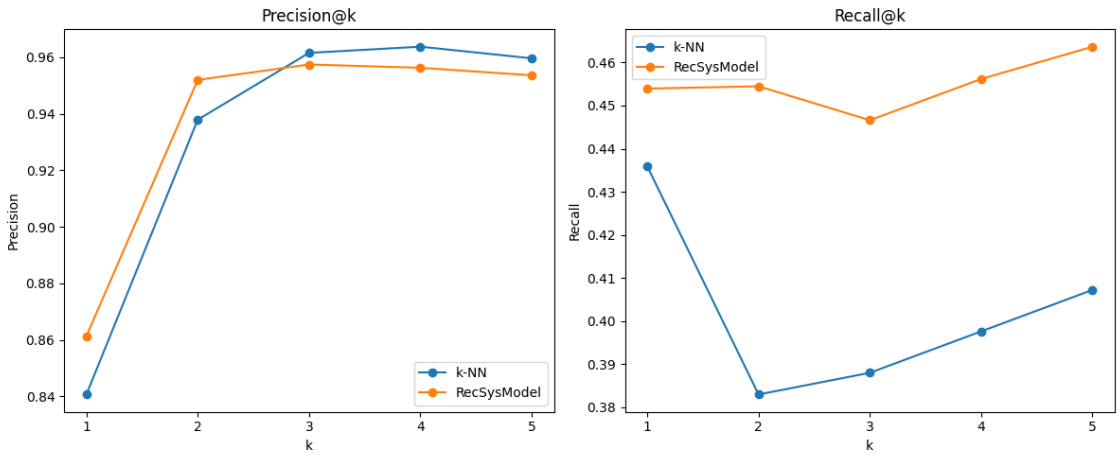


Figure 6.4: Metrics for Different Values of k and models

7 Conclusions

7.1 Summary of Contributions

This thesis establishes a recommendation system capable of suggesting candidate genes for diseases based on their level of scientific evidence. To achieve this objective, a recommendation system was created titled, `RecSysModel`, inspired by the Neural Collaborative Filtering method, utilizing PyTorch. The absence of this model in the state-of-art review underscores the relevance and innovation this study brings to the scientific community.

Due to the lack of structured data relating to the diseases present in the Disease Ontology and possible genes associated with them, the DisGeNET database was explored. However, it does not contain references to all Disease Ontology's DOID values thus, it necessitated the creation of a SQLite database named `DiseaseGene`. Through meticulous mapping, this database seamlessly integrates complex relationships between genes and diseases. Consequently, the created database contains information about diseases present in the Disease Ontology, genes, and their interactions. It encompasses details about 7101 diseases from the Disease Ontology, covering 9929 diseases identified by UMLS, 20024 genes, and 541102 associations between diseases and genes.

The developed data partitioning technique overcomes challenges, ensuring all diseases are present in the training set. The efficient integration of data directly from SQLite into the model optimizes the effectiveness of the recommendation process.

Detailed analysis of datasets, efficient transformation into tensors using PyTorch, GPU, and implementation of techniques such as batching data and early stopping strategies highlight the commitment to the model's robustness and efficiency. The results proved highly positive when assessed by metrics such as Root Mean Square Error (RMSE), $Precision@k$, and $Recall@k$, validating the model's effectiveness in recommending candidate genes.

The final evaluation emphasizes the model's effectiveness, precision, and resilience, marking its success in achieving the established research objectives. The amalgamation of information about diseases present in the Disease Ontology, database construction, and recommendation system development establishes a solid foundation to advance our understanding of gene-disease associations.

This research not only refines recommendation systems but also contributes to the discovery and application of knowledge in genetics and diseases. In an ever-evolving biomedical informatics landscape, the insights gained pave the way for further exploration and advancements in the integration of data science and genomics.

In summary, this thesis provides not only a systematic and detailed procedure for creating a robust dataset and developing an efficient recommendation system but also significantly contributes to bioinformatics and precision medicine. It offers a valuable tool for gene-disease associations, with the potential to enhance understanding of disease mechanisms and facilitate targeted therapeutic interventions in personalized medicine. The notable consistency between the model's predictions and actual values reinforces confidence in the utility of these genes as potential targets for future studies, representing a promising recommendation tool not only for individualized approaches and rare diseases but also for broader investigations into disease groups.

7.2 Future Work

Nevertheless, there are a few aspects that warrant further improvement. The massive influx of information in scientific literature pertaining to genes linked to diseases presents substantial difficulties in upholding up-to-date databases and identifying pertinent research. To propel advancements, upcoming efforts should delve into incorporating cutting-edge technologies such as Gemini[1] or ChatGPT[2]. This task includes not only, the act of updating the database created, but also the process of finding the most pertinent research to genes and diseases. Given the intricate nature resulting from the vast amount of literature on genes related to diseases, these models can have a crucial impact on improving the database by helping to extract relevant information from scientific literature. Moreover, these technologies might potentially improve the computation of the scientific evidence index in the database.

In future research, it would be intriguing to compute the semantic similarity across illnesses, using methods like DiShIn[3]: Disjunctive Shared Information for Semantic Similarity Measures [26], given that diseases are included inside the Disease Ontology.

These sophisticated algorithms may greatly enhance the precision of extracting information from publications, making it easier to identify crucial sentences that either confirm or contradict the connections between genes and illnesses and the computation of the similarity across illnesses. This integrated approach offers a more effective and comprehensive method of updating

[1] https://deepmind.google/technologies/gemini/
[2] https://openai.com/blog/chatgpt
[3] http://labs.rd.ciencias.ulisboa.pt/dishin/

the database established in light of the continuously expanding landscape of scientific knowledge.

Exploring the TorchRec[4] library, which is specifically designed for Recommendation Systems in the PyTorch domain, can provide substantial advantages when it comes to coding. The library facilitates the utilization of hybrid model-parallelism and data-parallelism, allowing for the creation of robust and effective models that can be executed on diverse devices and nodes. Moreover, the inclusion of GPU inference support and specialized techniques for managing extensive datasets renders TorchRec an especially pertinent selection.

These characteristics are essential for improving the computational efficiency of the system, allowing it to be highly scalable and capable of handling the challenges of large-scale recommendation systems, as is the case in this project. Utilizing TorchRec offers a strong basis for enhancing system efficiency, utilizing the unique features of PyTorch, and effectively tackling issues related to large amounts of data.

Exploring a hybrid recommendation system that integrates the proposed approach with collaborative filtering, incorporating patients' genetic data, is an intriguing research area. The careful integration of genetic information holds the potential to significantly enhance recommendations, creating a personalized and accurate system. By focusing on patients, this hybrid approach promises a more individualized experience, contributing to practical advancements in the system. Additionally, considering the incorporation of temporal information into the dataset is crucial, offering the opportunity to model the evolution of interactions between diseases and genes over time. This holistic approach, uniting individual genetic information and considering temporal evolution, aims to enhance system accuracy, adaptability, and sensitivity to nuances, resulting in a more robust and patient-centric solution.

Recently, creating an application or website that utilizes the developed system would be a valuable and easily accessible tool for researchers, the scientific community, and medical professionals. In addition to streamlining the search for genes linked to diseases, this tool would have a pivotal role in advancing treatments. The direct implementation of this platform would have a huge impact on the advancement of healthcare by providing an intuitive platform to explore and apply genetic research findings in clinical practice. This contribution seeks to promote a well-informed and effective medical environment, facilitating the utilization of genetic research to enhance global health.

[4] `https://pytorch.org/blog/introducing-torchrec/`

Glossary

Alternating Least Squares Alternating Least Squares (ALS) is an optimization algorithm used in linear algebra and collaborative filtering for matrix factorization. ALS iteratively alternates between fixing one matrix and solving a least-squares problem to update the other matrix. This technique is commonly applied in recommendation systems to approximate user-item interaction matrices and discover latent factors representing underlying patterns in collaborative filtering scenarios. 14

Apriori Association Rule Mining, specifically the Apriori algorithm, is a data mining technique used to discover interesting relationships and patterns within large datasets. Apriori identifies associations among items by establishing rules based on their co-occurrence frequencies. Widely applied in market basket analysis and recommendation systems, Apriori helps reveal valuable insights into the associations between items in transactional databases. 14

Bayesian theory Bayesian theory, rooted in probability theory, is an approach to statistical inference that combines prior knowledge with observed data to update and refine the probability of hypotheses. It provides a framework for reasoning under uncertainty, allowing for the incorporation of prior beliefs and iterative learning. Bayesian methods are widely used in various fields, including machine learning, data analysis, and decision-making, providing a flexible and coherent approach to statistical modeling. 14

clustering Clustering is a machine learning and data analysis technique that involves grouping similar data points together based on certain criteria. The goal is to identify inherent structures or patterns in the data, where items within the same cluster share more similarities with each other than with those in other clusters. Clustering is widely used in various domains, including customer segmentation, image segmentation, and anomaly detection. 17

disease A pathological condition of a part, organ, or system of an organism resulting from various

causes, such as infection, genetic defect, or environmental stress. Diseases are often characterized by specific symptoms and signs that negatively impact health. xix, 3–5, 29–38, 40–47, 50, 52–54, 56–62, 68–80, 82–84

DNA Deoxyribonucleic Acid, a molecule that carries the genetic instructions used in the development and functioning of all known living organisms. DNA consists of two long chains of nucleotides twisted into a double helix. Each nucleotide contains a phosphate group, a sugar molecule (deoxyribose), and one of four nitrogenous bases: adenine (A), thymine (T), guanine (G), and cytosine (C). The sequence of these bases encodes genetic information. 3

embeddings Embeddings are numerical representations of objects, such as words, phrases, or entities, in a continuous vector space. These representations, often learned through techniques like Word Embeddings or Graph Embeddings, capture semantic relationships and similarities between objects. Embeddings play a crucial role in natural language processing, information retrieval, and various machine learning tasks by encoding meaningful information about the objects in a compact and expressive form. 15, 33, 56, 57, 59–62

FP-growth Association Rule Mining, particularly the FP-growth algorithm, is a data mining technique used to discover frequent patterns and relationships within large datasets. FP-growth constructs a compact data structure called a frequent pattern (FP) tree to efficiently extract associations between items, making it a powerful approach for analyzing transactional data and extracting valuable insights from diverse domains. 14

framework A framework is a structural or conceptual foundation that provides organizational and developmental guidelines for software. It consists of a set of tools, libraries, and conventions that facilitate the creation and maintenance of applications, promoting good design practices and architecture. Frameworks are designed to expedite the development process by offering standardized solutions for common tasks, reducing complexity, and ensuring consistency in code. They are widely used in various areas, including web development, mobile apps, desktop applications, and machine learning. 32, 38, 40, 41, 49

gene A gene is a segment of DNA that encodes the instructions for the synthesis of functional molecules, typically proteins. Genes serve as the basic units of heredity, carrying genetic information from one generation to the next, and play a crucial role in the development and functioning of living organisms. 3, 29–35, 56, 59, 61, 62, 68, 69, 73, 77, 78

genetics Genetics is the scientific study of genes, heredity, and genetic variation in living organisms. It explores the principles governing the transmission of traits from one generation to the next and the molecular mechanisms that underlie genetic processes. Genetics encompasses the study of DNA, genes, chromosomes, and their roles in determining an organism's characteristics. This field plays a fundamental role in understanding inheritance, genetic disorders, and the genetic basis of biological diversity. 29

GitHub GitHub is a web-based platform for version control and collaborative software development. It provides tools for code hosting, project management, and team collaboration, making it a popular choice among developers for managing and sharing code repositories. 43, 48

machine learning Machine learning is a branch of artificial intelligence (AI) that focuses on the development of algorithms and statistical models enabling computers to perform tasks without explicit programming. It involves the use of data to train systems to recognize patterns, make predictions, and improve their performance over time, making it a key technology in areas such as data analysis, pattern recognition, and automated decision-making. 13, 15, 17, 29–31, 33, 34, 37, 38, 49, 51, 57

matrix factorization Matrix Factorization is a mathematical technique used in linear algebra and machine learning to decompose a matrix into the product of two or more matrices. In the context of recommendation systems, matrix factorization is employed to approximate a user-item interaction matrix, enabling the discovery of latent factors that represent underlying patterns and preferences. This approach is particularly valuable for collaborative filtering in recommendation algorithms. 14, 15

metadata Metadata refers to descriptive information that provides context and details about other data. It includes attributes such as the origin, format, structure, and content of data, serving to facilitate the organization, retrieval, and understanding of information. Metadata is crucial in various domains, including data management, library sciences, and information systems, as it enhances the accessibility, usability, and overall management of data assets. 17, 33

microbe Microbe, short for microorganism, refers to a microscopic living organism that includes bacteria, viruses, fungi, and single-celled organisms. Microbes play diverse roles in ecosystems, ranging from decomposition and nutrient cycling to causing infectious diseases, and they are essential for various biological processes on Earth. 3, 29

microbiomics Microbiomics is the study of the collective genetic material, microbial communities, and their interactions within a specific environment. It encompasses the analysis of microorganisms, such as bacteria, viruses, and fungi, to understand their diversity, functions, and impact on the host organism or ecosystem. Microbiomics plays a crucial role in fields like human health, ecology, and agriculture, providing insights into the complex relationships between microorganisms and their surroundings. 29

Natural Language Processing Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and human languages. NLP techniques enable machines to understand, interpret, and generate human-like language, facilitating tasks such as sentiment analysis, machine translation, text summarization, and

language modeling. NLP plays a pivotal role in bridging the gap between computers and human communication, allowing for the development of intelligent systems capable of processing and understanding natural language. 17, 18, 29

Naïve Bayesian A probabilistic classification technique based on Bayes' theorem, assuming independence among features. It's often used in machine learning for classification tasks. 32

neural network A Neural Network is a computational model inspired by the structure and function of the human brain, composed of interconnected nodes (neurons) organized into layers. It is used for machine learning tasks, including pattern recognition, classification, and regression. Neural networks learn from data by adjusting the weights of connections between neurons during a training process, allowing them to generalize and make predictions on new, unseen data. 15, 33, 49, 57, 58

phenotype The observable physical or biochemical characteristics of an organism, resulting from the interaction of its genotype (genetic makeup) with the environment. Phenotypes can include traits such as appearance, behavior, and other measurable characteristics. 34

protamines Protamines are small, positively charged proteins found in the sperm of many species. Known for their role in packaging and stabilizing the DNA within the sperm nucleus, protamines replace histones during sperm maturation, enabling efficient compaction of the genetic material. This unique characteristic of protamines contributes to the structural integrity of sperm and is essential for successful fertilization and the transmission of genetic information during reproduction. 29

protein A protein is a large, complex molecule composed of amino acids, essential for the structure, function, and regulation of cells and tissues in living organisms. Proteins play diverse roles, serving as enzymes, structural components, antibodies, and signaling molecules. The sequence of amino acids in a protein determines its unique structure and function, and variations in protein structure contribute to the vast biological diversity observed in living organisms. 29, 31, 32

Python Python is a high-level, versatile programming language known for its simplicity and readability. It is widely used in various fields, including web development, data analysis, machine learning, and scientific computing. 48

RNA Ribonucleic Acid, a type of nucleic acid that plays a crucial role in various biological processes. RNA is involved in protein synthesis, gene regulation, and other cellular functions. It consists of a single strand of nucleotides, including adenine (A), uracil (U), guanine (G), and cytosine (C). There are different types of RNA, such as messenger RNA (mRNA), transfer RNA (tRNA), and ribosomal RNA (rRNA). 36

Singular Value Decomposition Singular Value Decomposition (SVD) is a fundamental concept in linear algebra, representing the factorization of a matrix into three matrices, capturing the singular values and corresponding singular vectors. Widely used in data analysis, signal processing, and machine learning, SVD plays a crucial role in dimensionality reduction, noise reduction, and uncovering latent structures within matrices. 13, 14

SQL Structured Query Language (SQL) is a specialized programming language designed for managing and manipulating relational databases. SQL provides a standardized way to interact with databases, allowing users to define, query, update, and control the data stored in a relational database management system (RDBMS). It includes commands for tasks such as creating and modifying database schemas, inserting, updating, and retrieving data, as well as managing user permissions and transactions. 50

tensor A tensor is a mathematical object that generalizes the concept of vectors and matrices. In the context of machine learning and physics, tensors are multi-dimensional arrays capable of representing complex data structures. Tensors play a fundamental role in various scientific and engineering applications, including deep learning and computational physics. 49–51, 57, 60, 82

text mining Text mining, also known as text analytics, is the process of extracting valuable insights and patterns from unstructured textual data. It involves techniques from natural language processing, information retrieval, and machine learning to analyze and discover knowledge from large volumes of text. Text mining is applied in various domains, including sentiment analysis, document categorization, and information extraction, to transform unstructured text into structured and actionable information. 29, 30, 32

vanishing gradient Vanishing Gradient is a phenomenon in neural network training where the gradients of the loss function with respect to the weights become extremely small during backpropagation. This can impede the effective training of deep neural networks, particularly in architectures with many layers. The vanishing gradient problem often occurs when using certain activation functions, such as the sigmoid or hyperbolic tangent (tanh), as these functions squash input values into a limited range, causing the gradients to approach zero for extreme inputs. Strategies to mitigate the vanishing gradient problem include using alternative activation functions (e.g., ReLU), batch normalization, and employing specialized weight initialization techniques. 58

Word2Vec Word2Vec is a word embedding technique that represents words as vectors in a continuous vector space, enabling the capture of semantic and syntactic relationships between words in natural language processing (NLP) tasks. 33

Bibliography

[1] Muhammad Adnan, Yassaman Ebrahimzadeh Maboud, Divya Mahajan, and Prashant J Nair. Heterogeneous acceleration pipeline for recommendation system training. *arXiv preprint arXiv:2204.05436*, 2022.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[3] Charu C. Aggarwal. Content-based recommender systems. In *Recommender Systems: The Textbook*, pages 139–168. Springer International Publishing, 2016.

[4] Charu C. Aggarwal. Ensemble-based and hybrid recommender systems. In *Recommender Systems: The Textbook*, pages 199–224. Springer International Publishing, 2016.

[5] Charu C. Aggarwal. Knowledge-based recommender systems. In *Recommender Systems: The Textbook*, pages 167–197. Springer International Publishing, 2016.

[6] Charu C. Aggarwal. Model-based collaborative filtering. In *Recommender Systems: The Textbook*, pages 71–138. Springer International Publishing, 2016.

[7] Charu C Aggarwal and Charu C Aggarwal. Evaluating recommender systems. *Recommender Systems: The Textbook*, pages 225–254, 2016.

[8] Charu C Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016.

[9] Rishabh Ahuja, Arun Solanki, and Anand Nayyar. Movie recommender system using k-means clustering and k-nearest neighbor. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 263–268, Jan 2019.

[10] Marie Al-Ghossein. *Context-aware recommender systems for real-world applications*. Theses, Université Paris Saclay (COmUE), February 2019.

[11] Marie Al-Ghossein. *Context-aware recommender systems for real-world applications*. PhD thesis, Université Paris Saclay (COmUE), 2019.

[12] Sarah M Alghamdi, Paul N Schofield, and Robert Hoehndorf. Contribution of model organism phenotypes to the computational identification of human disease genes. *Dis Model Mech*, 15(7):dmm049441, 2022.

[13] Abdulrahman Althagafi, Lamia Alsubaie, Nagarajan Kathiresan, Katsuhiko Mineta, Tareq Aloraini, Fuad Al Mutairi, Majid Alfadhel, Takashi Gojobori, Ahmed Alfares, and Robert Hoehndorf. Deepsvp: integration of genotype and phenotype for structural variant prioritization using deep learning. *Bioinformatics*, 38(6):1677–1684, 2022.

[14] Xavier Amatriain. Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations Newsletter*, 14(2):37–48, 2013.

[15] Qazi Mohammad Areeb, Mohammad Nadeem, Shahab Saquib Sohail, Raza Imam, Faiyaz Doctor, Yassine Himeur, Amir Hussain, and Abbes Amira. Filter bubbles in recommender systems: Fact or fallacy—a systematic review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(6):e1512, 2023.

[16] M. Barros and F. M. Couto. Knowledge representation and management: a linked data perspective. *Yearb Med Inform*, 2016(1):178–183, 2016.

[17] Geneviève Baujat, Laurence Legeai-Mallet, Georges Finidori, Valérie Cormier-Daire, and Martine Le Merrer. Achondroplasia. *Best Practice & Research Clinical Rheumatology*, 22(1):3–18, 2008. Orphan Skeletal Diseases.

[18] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science*, 49:136–146, 2015. Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15).

[19] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370, 2002.

[20] Francesco Camilli and Marc Mé zard. Matrix factorization with neural networks. *Physical Review E*, 107(6), jun 2023.

[21] Jun Chen, Azza Althagafi, and Robert Hoehndorf. Predicting candidate genes from phenotypes, functions and anatomical site of expression. *Bioinformatics*, 37(6):853–860, 10 2020.

[22] Li Chen and Pearl Pu. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22:125–150, 2012.

[23] Zefu Chen, Yu Zheng, Yongxin Yang, Yingzhao Huang, Sen Zhao, Hengqiang Zhao, Chenxi Yu, Xiying Dong, Yuanqiang Zhang, Lianlei Wang, Zhengye Zhao, Shengru Wang, Yang Yang, Yue Ming, Jianzhong Su, Guixing Qiu, Zhihong Wu, Terry Jianguo Zhang, and Nan Wu. Phenoapt leverages clinical expertise to prioritize candidate genes via machine learning. *The American Journal of Human Genetics*, 109(2):270–281, 2022.

[24] Hyonwoo Chun, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Noriko Nagata, Teruhisa Hishiki, and Jun'ichi Tsujii. Extraction of gene-disease relations from medline using domain dictionaries and machine learning. In *Pac Symp Biocomput*, pages 4–15, 2006.

[25] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*, volume 60, pages 1853–1870. Citeseer, 1999.

[26] Francisco M Couto and Andre Lamurias. Semantic similarity definition., 2019.

[27] Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4225–4232, 2023.

[28] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

[29] Yashar Deldjoo, Fatemeh Nazary, Arnau Ramisa, Julian Mcauley, Giovanni Pellegrini, Alejandro Bellogin, and Tommaso Di Noia. A review of modern fashion recommender systems, 2023.

[30] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[31] Tommaso Di Noia and Vito Claudio Ostuni. *Recommender Systems and Linked Open Data*, pages 88–113. Springer International Publishing, Cham, 2015.

[32] Dejing Dou, Hao Wang, and Haishan Liu. Semantic data mining: A survey of ontology-based approaches. In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pages 244–251, 2015.

[33] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10, 2014.

[34] Friederike Ehrhart, Egon L Willighagen, Martina Kutmon, Marit van Hoften, Leopold M G Curfs, and Chris T Evelo. A resource to explore the discovery of rare diseases and their causative genes. *Sci Data*, 8(1):124, 2021.

[35] Michael D Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In *Conference on fairness, accountability and transparency*, pages 172–186. PMLR, 2018.

[36] Mehdi Elahi, Danial Khosh Kholgh, Mohammad Sina Kiarostami, Sorush Saghari, Shiva Parsa Rad, and Marko Tkalčič. Investigating the impact of recommender systems on user-based and item-based popularity bias. *Information Processing & Management*, 58(5):102655, 2021.

[37] Jemma Evans, Rob D Catalano, Kevin Morgan, Hilary OD Critchley, Robert P Millar, and Henry N Jabbour. Prokineticin 1 signaling and gene regulation in early human pregnancy. *Endocrinology*, 149(6):2877–2887, 2008.

[38] Bing-Jian Feng. Perch: A unified framework for disease gene prioritization. *Hum Mutat*, 38(3):243–251, 2017.

[39] Amin Firoozshahian, Joel Coburn, Roman Levenstein, Rakesh Nattoji, Ashwin Kamath, Olivia Wu, Gurdeepak Grewal, Harish Aepala, Bhasker Jakka, Bob Dreyer, et al. Mtia: First generation silicon targeting meta's recommendation systems. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–13, 2023.

[40] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI open*, 2:100–126, 2021.

[41] Shang Gao and Xiaosheng Wang. Predicting type 1 diabetes candidate genes using human protein-protein interaction networks. *J Comput Sci Syst Biol*, 2:133, 2009.

[42] Zhe Gao, Yiran Pan, Pengyong Ding, and Rui Xu. A knowledge graph-based disease-gene prediction system using multi-relational graph convolution networks. *AMIA Annu Symp Proc*, 2022:468–476, 2023.

[43] General Data Protection Regulation (GDPR). Regulation (eu) 2016/679 of the european parliament and of the council, 2016.

[44] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, dec 1992.

[45] Asela Gunawardana, Guy Shani, and Sivan Yogev. Evaluating recommender systems. In *Recommender systems handbook*, pages 547–601. Springer, 2012.

[46] Alba Gutiérrez-Sacristán, Solène Grosdidier, Olga Valverde, Marta Torrens, Àlex Bravo, Janet Piñero, Ferran Sanz, and Laura I. Furlong. PsyGeNET: a knowledge platform on psychiatric disorders and their genes. *Bioinformatics*, 31(18):3075–3077, 05 2015.

[47] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 173–182, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[48] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[49] Jun Hong, Xiaoyuan Su, and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:421–425, 10 2009.

[50] Jinyu Hu, Sugam Sharma, Zhiwei Gao, and Victor Chang. Gene-based collaborative filtering using recommender system. *Computers & Electrical Engineering*, 65:332–341, 2018.

[51] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Dec 2008.

[52] Dmytro Ivchenko, Dennis Van Der Staay, Colin Taylor, Xing Liu, Will Feng, Rahul Kindi, Anirudh Sudarshan, and Shahin Sefati. Torchrec: a pytorch domain library for recommendation systems. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 482–483, 2022.

[53] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[54] Shashidhar Reddy Javaji and Krutika Sarode. Hybrid recommendation system using graph neural network and bert embeddings. *arXiv preprint arXiv:2310.04878*, 2023.

[55] Kalyan Kumar Jena, Sourav Kumar Bhoi, Chittaranjan Mallick, Soumya Ranjan Jena, Raghvendra Kumar, Hoang Viet Long, and Nguyen Thi Kim Son. Neural model based collaborative filtering for movie recommendation system. *International Journal of Information Technology*, 14(4):2067–2077, 2022.

[56] Christopher C Johnson et al. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems*, 27(78):1–9, 2014.

[57] Şenay Kafkas and Robert Hoehndorf. Ontology based text mining of gene-phenotype associations: application to candidate gene prediction. *Database (Oxford)*, 2019:baz019, 2019.

[58] Ho Heon Kim, Junwoo Woo, Dong-Wook Kim, Jungsul Lee, Go Hun Seo, Hane Lee, and Kyoungyeul Lee. Disease-causing variant recommendation system for clinical genome interpretation with adjusted scores for artefactual variants. *bioRxiv*, 2022.

[59] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[60] Daniel Kluver and Joseph A Konstan. Evaluating recommender behavior for new users. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 121–128, 2014.

[61] S. B. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, April 2013.

[62] Pushpendra Kumar and Ramjeevan Singh Thakur. Recommendation system techniques and related issues: a survey. *International Journal of Information Technology*, 10:495–501, 2018.

[63] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. *Content-based Recommender Systems: State of the Art and Trends*, pages 73–105. Springer US, Boston, MA, 2011.

[64] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics reports*, 519(1):1–49, 2012.

[65] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Exploring data splitting strategies for the evaluation of recommendation models. In *Proceedings of the 14th ACM conference on recommender systems*, pages 681–686, 2020.

[66] Frank Meyer. Recommender systems in industrial contexts, 2012.

[67] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*, 2019.

[68] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[69] A Noorian, A Harounabadi, and M Hazratifard. A sequential neural recommendation system exploiting bert and lstm on social media posts. *Complex & Intelligent Systems*, pages 1–24, 2023.

[70] Douglas W Oard and Jinmook Kim. Implicit feedback for recommender systems. 1998.

[71] Martin Oti, Berend Snel, Martijn A Huynen, and Han G Brunner. Predicting disease genes using protein-protein interactions. *J Med Genet*, 43(8):691–8, 2006.

[72] Makbule Gulcin Ozsoy, Faruk Polat, and Reda Alhajj. Inference of gene regulatory networks via multiple data sources and a recommendation method. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 661–664, 2015.

[73] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13:393–408, 1999.

[74] Michael J. Pazzani and Daniel Billsus. *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[75] Janet Piñero, Josep Saüch, Ferran Sanz, and Laura I. Furlong. The disgenet cytoscape app: Exploring and visualizing disease genomics data. *Computational and Structural Biotechnology Journal*, 19:2960–2967, 2021.

[76] Shaina Raza and Chen Ding. News recommender system: a review of recent progress, challenges, and opportunities. *Artificial Intelligence Review*, pages 1–52, 2022.

[77] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[78] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender systems: introduction and challengess*, pages 1–34. Springer Publishing Company, Incorporated, 2nd edition, 2015.

[79] J. J. Sandvig, Bamshad Mobasher, and Robin Burke. Robustness of collaborative recommendation based on association rule mining. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, page 105–112, New York, NY, USA, 2007. Association for Computing Machinery.

[80] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pages 291–324. Springer, 2007.

[81] J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, 1999.

[82] Gunnar Schröder, Maik Thiele, and Wolfgang Lehner. Setting goals and choosing metrics for recommender system evaluations. In *UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA*, volume 23, page 53, 2011.

[83] James Scott and Nick Polson. *Inteligência Artificial: Como funciona e como podemos usá-la para criar um mundo melhor*. Vogais, março 2020. Classificação temática: Livros em Português, Informática, Inteligência Artificial.

[84] Philip Sedgwick. Pearson's correlation coefficient. *Bmj*, 345, 2012.

[85] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. *Recommender systems handbook*, pages 257–297, 2011.

[86] Sanjay K Shukla, Narayana S Murali, and Murray H Brilliant. Personalized medicine going precise: from genomics to microbiomics. *Trends in molecular medicine*, 21(8):461–462, 2015.

[87] Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. OPA2Vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics*, 35(12):2133–2140, 11 2018.

[88] Diana Sousa and Francisco M. Couto. Biont: Deep learning using multiple biomedical ontologies for relation extraction. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, *Advances in Information Retrieval*, pages 367–374, Cham, 2020. Springer International Publishing.

[89] Caroline Spaight, Justine Gross, Antje Horsch, and Jardena Jacqueline Puder. Gestational diabetes mellitus. *Novelties in diabetes*, 31:163–178, 2016.

[90] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

[91] K Suriyakrishnaan, L Charan Kumar, and R Vignesh. Recommendation system for agriculture using machine learning and deep learning. In *Inventive Systems and Control: Proceedings of ICISC 2022*, pages 625–635. Springer, 2022.

[92] John K. Tarus, Zhendong Niu, and Ghulam Mustafa. Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50(1):21–48, 2018.

[93] Shari Trewin. Knowledge-based recommender systems. *Encyclopedia of library and information science*, 69(Supplement 32):180, 2000.

[94] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[95] D. Wallach and B. Goffinet. Mean squared error of prediction as a criterion for evaluating and comparing system models. *Ecological Modelling*, 44(3):299–306, 1989.

[96] Tara L Wenger, Anne V Hing, and Kelly N Evans. Apert syndrome. 2019.

[97] Yu Xin and Tommi Jaakkola. Controlling privacy in recommender systems. *Advances in neural information processing systems*, 27, 2014.

[98] Jun Ye. Cosine similarity measures for intuitionistic fuzzy sets and their applications. *Mathematical and Computer Modelling*, 53(1):91–97, 2011.

[99] Pooya Zakeri, Jaak Simm, Adam Arany, Sarah ElShal, and Yves Moreau. Gene prioritization using Bayesian matrix factorization with genomic and phenotypic side information. *Bioinformatics*, 34(13):i447–i456, 06 2018.

[100] Xiangxiang Zeng, Yinglai Lin, Yuying He, Linyuan Lü, Xiaoping Min, and Alfonso Rodríguez-Patón. Deep collaborative filtering for prediction of disease genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(5):1639–1647, 2020.

[101] Xian-Da Zhang. Machine learning. In *A Matrix Algebra Approach to Artificial Intelligence*, pages 223–440, Singapore, 2020. Springer Singapore.

[102] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

[103] Fernando Zhapa-Camacho, Maxat Kulmanov, and Robert Hoehndorf. mOWL: Python library for machine learning with biomedical ontologies. *Bioinformatics*, 39(1):btac811, 12 2022.

[104] Xujuan Zhou, Yue Xu, Yuefeng Li, Audun Josang, and Clive Cox. The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review*, 37:119–132, 2012.