

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE MATEMÁTICA

ISCTE
INSTITUTO UNIVERSITÁRIO DE LISBOA
DEPARTAMENTO DE FINANÇAS



**Ciências
ULisboa**

iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Redes Neurais LSTM em Python: Previsão do S&P500

André Filipe Ferreira Lourenço da Silva

Mestrado em Matemática Financeira

Dissertação orientada por:

Doutora Diana A. Mendes

2023

Agradecimentos

Fico contente por concluir esta fase da minha vida académica, que não seria possível sem o apoio da minha família. Em particular agradeço aos meus pais e irmãos, pelo apoio e motivação que me deram ao longo dos anos.

Agradeço à minha namorada Catarina pelo apoio e carinho neste percurso.

Também agradeço à professora Doutora Diana Mendes pela atenção, esforço e dedicação.

Resumo

Nesta dissertação, iremos explorar um pouco do universo que são os algoritmos *Deep Learning*, para séries temporais. Em particular, através do algoritmo LSTM (*Long Short-Term Memory*), foi desenvolvido um modelo capaz de prever o S&P500, com resultados significativos, no período de análise desde 1 de janeiro 2020 até 31 de dezembro de 2022.

A linguagem de programação Python foi escolhida para implementar o código, dado que disponibiliza uma grande quantidade de algoritmos pré-definidos em bibliotecas que facilitam tanto a escrita, como a leitura e interpretação do código.

Além disto, foram escolhidos vários indicadores fundamentais, macroeconômicos e técnicos, com expectativa de que o modelo LSTM consiga captar relações entre eles.

Após serem abordados os conceitos chave para compreensão da análise, exploramos de forma detalhada a metodologia, raciocínio e técnicas usadas no tratamento de dados, desenvolvimento, treino e *tuning* dos hiperparâmetros do modelo LSTM.

De seguida, passamos à análise de sensibilidade, onde é avaliada a relevância de cada indicador e parâmetro para o resultado do modelo, realizando ajustes caso necessário e por fim comparamos o modelo final com o modelo *Naive*.

Concluimos esta dissertação respondendo a algumas questões e interpretando os resultados do algoritmo do ponto de vista econômico.

Palavras-chave: Redes Neurais, LSTM, *tuning*, *hidden layers*, RMSE

Abstract

We will explore in this dissertation, a small part of the universe of Deep Learning algorithms for time series data. Using the LSTM (Long Short-Term Memory) algorithm, we will create a model capable of predicting the S&P500 with significant results. The data period analysed spans from January 1, 2020, to December 31, 2022.

The Python programming language was chosen to implement the code because it provides a vast number of predefined algorithms in libraries make it easier, both code writing, reading and interpretation.

We have selected several fundamental, macroeconomic, and technical indicators, with the expectation that the LSTM model could capture relationships between them.

Frist we will address the key concepts necessary for understanding the analysis and them explore in detail the methodology, reasoning, and techniques used in data preprocessing, model training, and tuning of LSTM model hyperparameters.

We then proceed to the sensitivity analysis, where we assess the relevance of each indicator and parameter to the model's outcome, adjusting if necessary and finally, we compare our final model to the Naive model.

Then we conclude this dissertation by answering some questions and interpreting the model results from an economic perspective.

Keywords: Neural Networks, LSTM, tuning, hidden layers, RMSE

Índice

Agradecimentos	iii
Resumo	v
Abstract	vii
Capítulo 1. Introdução	1
Capítulo 2. Revisão da Literatura	3
Capítulo 3. Revisão Teórica	5
3.1. Redes Neurais Recorrentes (<i>Recurrent Neural Networks</i>)	5
3.2. LSTM (<i>Long Short-Term Memory</i>)	6
3.3. Indicadores	7
3.3.1. Fundamentais	7
3.3.2. Macroeconómicos	8
3.3.3. Técnicos	8
Capítulo 4. Metodologia	11
4.1. Indicadores	11
4.2. Tratamento de dados	12
4.3. Treino do modelo	13
4.4. Especificações do modelo LSTM	14
4.5. <i>Tuning</i> dos Hiperparâmetros	15
Capítulo 5. Análise de Sensibilidade	17
5.1. Relevância de Indicadores	17
5.2. Divisão da amostra	18
5.3. Janela Deslizante (<i>Rolling Window</i>)	19
5.4. Ajuste de Hiperparâmetros	19
5.5. Número de <i>Hidden Layers</i>	19
5.6. Modelo Ingênuo (<i>Naive Model</i>)	19
Capítulo 6. Conclusão	21
Referências Bibliográficas	23
Anexos	25

Índice de Quadros

Quadro 1: Indicadores, Fonte e Frequência dos registros	11
Quadro 2: Parâmetros <i>default</i> usados em cada Indicador Técnico	11
Quadro 3: Avaliação da Relevância dos Indicadores	18

Índice de Figuras

Figura 1: Representação do Neurónio da Rede Neuronal	5
Figura 2: Representação da Rede Neuronal	6
Figura 3: Representação do algoritmo LSTM	7
Figura 4: Matriz de correlação de Pearson dos Indicadores	12
Figura 5: Representação das Funções Ativação Sigmoide, ReLU, TanH, e respetivas derivadas	25
Figura 6: Representação gráfica dos indicadores normalizados EFFR, UNRATE e UMCSSENT	26
Figura 7: Representação gráfica do indicador SPY, da previsão LSTM e da previsão <i>Naive</i>	26

Lista de Abreviações

API	<i>Application Programming Interface</i>
ATR	<i>Average True Range</i>
BLS	<i>Bureau of Labor Statistics</i>
CBOE	<i>Chicago Board Options Exchange</i>
DXY	<i>US Dollar Index</i>
EFFR	<i>Effective Federal Funds Rate</i>
EMA	<i>Exponential Moving Average</i>
EUA	Estados Unidos da América
FED	<i>Federal Reserve Bank</i>
FRED	<i>Federal Reserve Economic Data</i>
FVX	<i>5 Year Treasury Note Yield Index</i>
IRX	<i>13 Week Treasury Bill Yield Index</i>
LSTM	<i>Long Short-Term Memory</i>
MACD	<i>Moving Average Convergence Divergence</i>
MAE	<i>Mean Absolute Error</i>
OBV	<i>On-Balance Volume</i>
ReLU	<i>Rectified Linear Unit</i>
RMSE	<i>Root Mean Square Error</i>
RSI	<i>Relative Strength Index</i>
S&P500	<i>Standard & Poor's 500</i>
SPY	<i>SPDR S&P 500 ETF Trust</i>
TanH	<i>Tangente Hiperbólica</i>
TNX	<i>10 Year Treasury Note Yield Index</i>
TSCV	<i>Time Series Cross-Validation</i>
UMCSSENT	<i>University of Michigan Consumer Sentiment Index</i>
UNRATE	<i>Unemployment Rate</i>
VIX	<i>Volatility Index</i>

CAPÍTULO 1

Introdução

Prever o S&P500 ou qualquer ativo relacionado com os mercados financeiros de forma consistente, na prática é perto de impossível, quer seja porque é impossível representar as infinitas variáveis que influenciam o mercado, ou pela falta de poder computacional necessário para processar esse mesmo número infinito de dados. De um ponto de vista filosófico, o modelo teria de se considerar a si mesmo na previsão, pois a partir do momento em que o modelo “prevê” como o mercado irá agir, essa previsão também é uma variável que teria de ser levada em consideração.

Simplificando, estamos a tentar prever o futuro o que é claramente impossível, levantando assim a questão, porquê que existem tantas pessoas e instituições, a investir vastos recursos humanos e financeiros na resolução deste problema? Na verdade, não estão realmente a tentar prever o futuro, nem os mercados. Estão a investir numa vantagem ou dica que lhes permita no longo prazo ter lucros com menor risco. Por exemplo, se investíssemos 1€ para ganhar 1€ no jogo de lançar uma moeda equilibrada (ou justa) ao ar, sabemos que 50% das vezes o resultado será cara e 50% será coroa. Deste modo, no longo prazo não teríamos lucro, no entanto se soubermos que a moeda retorna 51% das vezes o resultado coroa, então ao investirmos em coroa após 100 jogadas em média teríamos 51 vitórias e 49 derrotas, ou seja, um lucro médio de 2€ a cada 100 jogadas, garantido apenas com 1% vantagem.

A capacidade de captar estas pequenas, mas relevantes vantagens no mercado é o que faz a diferença entre um bom e um mau investidor. Para isto há quem use uma grande variedade de indicadores e nesta dissertação iremos analisar a relevância para o modelo de indicadores fundamentais, macroeconómicos, de análise técnica e combinações entre os mesmos.

Para executar o algoritmo LSTM foi escolhido a linguagem de programação Python pela sua vasta quantidade de algoritmos, já implementados nas suas bibliotecas, pela facilidade de leitura e escrita do código. Por estes motivos Python é bastante usado no ramo de Ciência de Dados. É de notar que em termos de volume e velocidade de processamento de dados, há outras linguagens mais adequadas dependente do modelo e tipo de dados.

O mercado acionista dos EUA é considerado o melhor, tanto pela sua dimensão, como pela elevada liquidez e reduzida volatilidade. Desta forma escolheu-se o S&P500, como objeto de análise, com expectativa de reduzir o ruído e volatilidade dos dados, representando de forma global o mercado dos EUA (cerca de 80%¹ a agosto 2023).

É de notar que, o valor de mercado do S&P500 é sobre representado pelo setor tecnológico (superior a 25%² a agosto 2023), pelo facto que as empresas com maior valorização no mercado são tecnológicas. Ainda assim, não é motivo para invalidar a análise.

O período de análise escolhido abrange a série temporal, de 1 de janeiro de 2020 a 31 de dezembro de 2022, com frequência diária, onde usamos o valor de fecho do mercado. Este período foi escolhido por ser o mais atual e por apenas a considerar os dados registados após o início da pandemia covid-19, tendo o objetivo de reduzir a variância dos dados, resultado das medidas impostas pelos países e bancos centrais que alteraram as suas políticas monetárias, afetando desde pessoas a mercados financeiros.

¹ <https://www.investopedia.com/ask/answers/040215/what-does-sp-500-index-measure-and-how-it-calculated.asp>

² <https://www.spglobal.com/spdji/pt/indices/equity/sp-500/#data>

Com isto, deixo algumas questões a ser respondidas no fim da análise:

- A. Quais os indicadores e parâmetros mais relevantes para a previsão do S&P500, usando o modelo LSTM?
- B. O modelo que usa os indicadores mais relevantes supera a capacidade de previsão do modelo que usa a totalidade de indicadores?
- C. Aumentar o número de *hidden layers* melhora a capacidade de previsão do modelo?
- D. O modelo LSTM tem maior capacidade de previsão que o modelo *Naive*?

Esta dissertação está estruturada em 6 capítulos. No capítulo 1 introduzimos os objetivos da análise. No capítulo 2 fazemos uma revisão da literatura, onde relacionamos literaturas com o tema da dissertação. No capítulo 3 abordamos uma revisão teórica dos conceitos chave que permitem uma melhor compreensão da dissertação. No capítulo 4, exploramos a metodologia, raciocínio e técnicas usadas na análise. No capítulo 5 analisamos e exploramos a relevância para o resultado do modelo dos indicadores, corte da amostra entre treino e teste, período da janela deslizante, hiperparâmetros e número de *hidden layers*. Por fim, no capítulo 6 reunimos a informação essencial da dissertação, respondendo às questões indicadas e apresentamos as conclusões finais.

CAPÍTULO 2

Revisão da Literatura

As redes neuronais têm ganho popularidade nos últimos anos, à medida que os recursos computacionais vão-se tornando cada mais rápidos, acessíveis e eficientes a processar algoritmos complexos.

O trajeto que levou ao desenvolvimento dos modelos atuais começou com a formulação do modelo matemático de um simples neurônio. Foi desenvolvido em 1943 pelo neurocientista Warren McCulloch e pelo cientista Walter Pitts. O modelo ficou conhecido como “neurônio de McCulloch-Pitts”, originalmente publicado no *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133 (McCulloch & Pitts, 1943).

As redes neurais recorrentes começaram a aparecer a partir de 1980, mas devido ao problema da dissipação do gradiente (*vanishing gradient*), não eram capazes de lidar com sequências longas de dados. Esta limitação foi formalmente exposta por Rumelhart David, Hinton Geoffrey & Williams Ronald (Rumelhart et al., 1986).

Esta situação levou a que não tenha havido grandes desenvolvimentos neste ramo até à introdução do conceito de redes neuronais com a capacidade adicional de captar dependências de longo prazo (Hochreiter & Schmidhuber, 1997), dando origem ao modelo LSTM e abrindo a porta a uma grande variedade de aplicações.

Mais recentemente, têm existido várias tentativas de melhorar a capacidade preditiva dos modelos, em todos os aspetos, desde a seleção de indicadores (Hoseinzade & Haratizadeh, 2019) até à combinação de modelos LSTM com outros modelos preditivos (Li et al., 2023).

Também foram desenvolvidos modelos mais exóticos, como a criação de modelos com objetivo de captar possíveis relações entre diferentes mercados como o ouro, os combustíveis ou o S&P500 em relação ao preço do Bitcoin (Parvini et al., 2022).

Já em 2023, houve melhorias no ramo da Inteligência Artificial, que tem criado grandes expectativas e aumentado as possibilidades na construção de modelos. Será este mais um passo em frente no desenvolvimento de novos os modelos de previsão? Esperemos que sim.

Revisão Teórica

O foco principal deste capítulo é apresentar conceitos essenciais para entender o contexto e metodologia da análise desenvolvida nesta dissertação.

Em detalhe, será apresentado uma visão geral do algoritmo LSTM e uma descrição dos indicadores.

Em teoria, cada indicador exógeno tem uma relação direta ou indireta com o indicador *target* SPY e é a significância desta relação que será analisada.

3.1 Redes Neurais Recorrentes (*Recurrent Neural Networks*)

Redes Neurais Recorrentes são modelos matemáticos muito usados para resolver problemas de regressão e classificação não-lineares. Os elementos-chave, o neurónio artificial e a rede neuronal, são inspirados pelo cérebro humano, em que deste dezenas a milhares de nodos, quando combinados desenvolvem uma relação de interconectividade entre si. Devido à complexidade destas relações as Redes Neurais são consideradas “*black boxes*”, pela frequente dificuldade de explicar completamente a conclusão dos resultados.

Cada nodo denominado como neurónio (*neuron*) recebe *input* de dados, executa cálculos em que o modelo atribui um peso (*weight*) a cada variável, que representa a relevância dessa variável para o resultado *output* do *neuron* e atribui um *bias*, sendo este o parâmetro responsável por compensar positiva ou negativamente o resultado.

O *output* em cada neurónio, é o resultado do cálculo da função de ativação (pré-definida) aplicando o somatório dos ($weights \times inputs$) + *bias*.

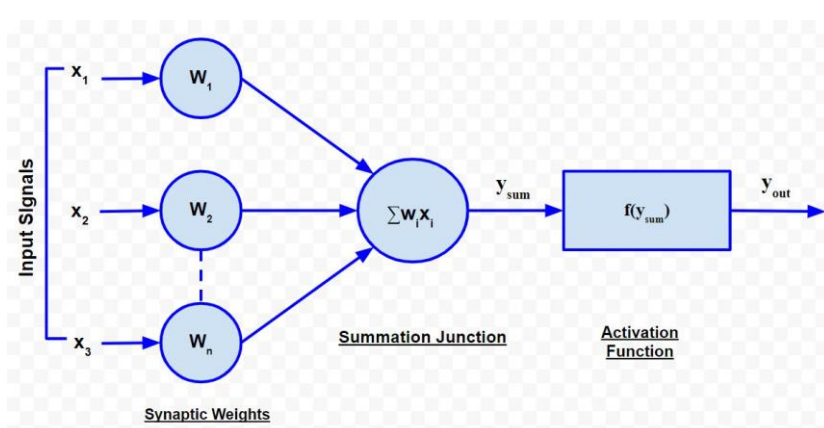


Figura 1: Representação do Neurónio da Rede Neuronal³

Durante o *tuning*, os valores dos *weights* e *bias* são ajustados automaticamente pelo algoritmo.

Uma rede neuronal é essencialmente dividida em 3 camadas, *Input Layer*, *Hidden Layers* e *Output Layer*. A *Input Layer* é a primeira camada de neurónios, onde se recebe os dados originais como *input*, atribui-se um peso a cada *input* de cada neurónio, efetua-se os cálculos e por fim cada neurónio da primeira camada devolve um *output*, que será usado nos cálculos de cada neurónio da camada seguinte, na *Hidden Layer* (camada oculta) e por aí adiante.

³ Fonte: https://www.geeksforgeeks.org/types-of-activation-function-in-ann/?ref=ml_lbp

As *Hidden Layers* podem ser compostas por uma ou mais camadas de neurónios conforme a complexidade adequada ao modelo.

Por fim a *Output Layer* é a camada final, responsável por retornar os resultados do modelo⁴.

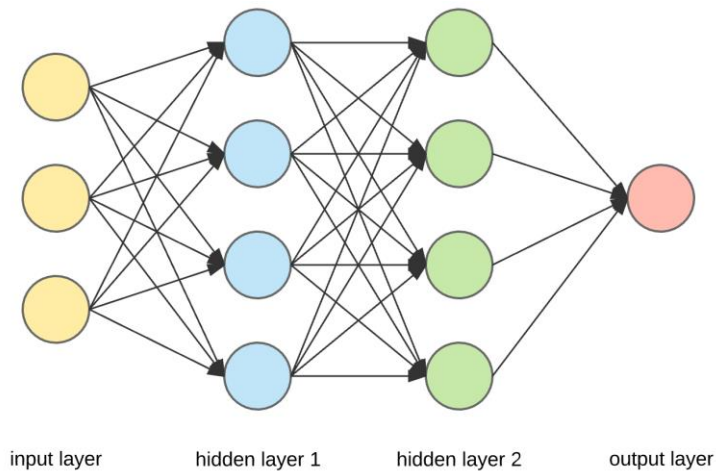


Figura 2: Representação Rede Neuronal⁵

3.2 LSTM (*Long Short-Term Memory*)

O modelo LSTM é baseado em Redes Neurais Recorrentes e distingue-se dos outros principalmente pela capacidade de capturar relações complexas entre os indicadores em sequências temporais de dados.

Tal como *Long Short-Term Memory* sugere, a célula, elemento base da arquitetura da rede, foi desenhada de forma a captar e relembrar sequências de curto e longo prazo. Consegue fazer isto principalmente tendo por base 3 “portões”.

O portão de esquecimento (*Forget Gate*), o portão de entrada (*Input Gate*) e o portão de saída (*Output Gate*), que têm diferentes funções, nomeadamente:

- O portão de esquecimento determina a quantidade de informação da célula anterior h_{t-1} que deve ser mantida ou esquecida na célula atual. Este valor, juntamente com o valor *input* atual X_t são alimentados à função ativação sigmoide (σ), que retorna um valor entre 0 e 1. Valores próximos de 0 indicam que mais informação será “esquecida”, enquanto valores próximos de 1 indica que mais informação será “mantida”.
O resultado da função sigmoide é multiplicado pelo valor do estado interno da célula anterior c_{t-1} , que representa a informação de longo prazo. O *output* é atribuído ao estado interno atual da célula c_t .
- O portão de entrada é responsável por determinar a quantidade de informação que deve ser adicionada ao estado interno atual da célula. Da mesma forma que o portão de esquecimento, também recebe informação da célula anterior h_{t-1} e o *input* atual X_t .
Estes valores em cálculos separados são então alimentados às funções de ativação sigmoide e tangente hiperbólica (TanH, valores entre -1 e 1).

⁴ Esta é uma visão simplificada do modelo. Conforme a análise de diferentes problemas e respetivas complexidades poderá ser necessário fazer alterações aos parâmetros do modelo. Para mais informações, a documentação oficial pode ser consultada em: https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM

⁵ Fonte: <https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6>

A função sigmoide controla quais informações são mais relevantes, enquanto a função tangente hiperbólica representa a quantidade de informação que será preservado a longo prazo. Os *outputs* destas funções são multiplicados entre si e em seguida, é adicionado ao estado interno da célula c_t .

- O portão de saída controla a saída de informação h_t para a próxima célula LSTM. Mais uma vez, recebe informação da célula anterior h_{t-1} , o input atual X_t e são alimentados à função sigmoide. O valor do estado interno da célula c_t é alimentado à função tangente hiperbólica, cujo resultado é multiplicado pelo resultado da função sigmoide. Este *output* é o novo valor h_t , que representa informações de curto prazo que serão passadas à próxima célula LSTM, juntamente com o estado interno da célula c_t .

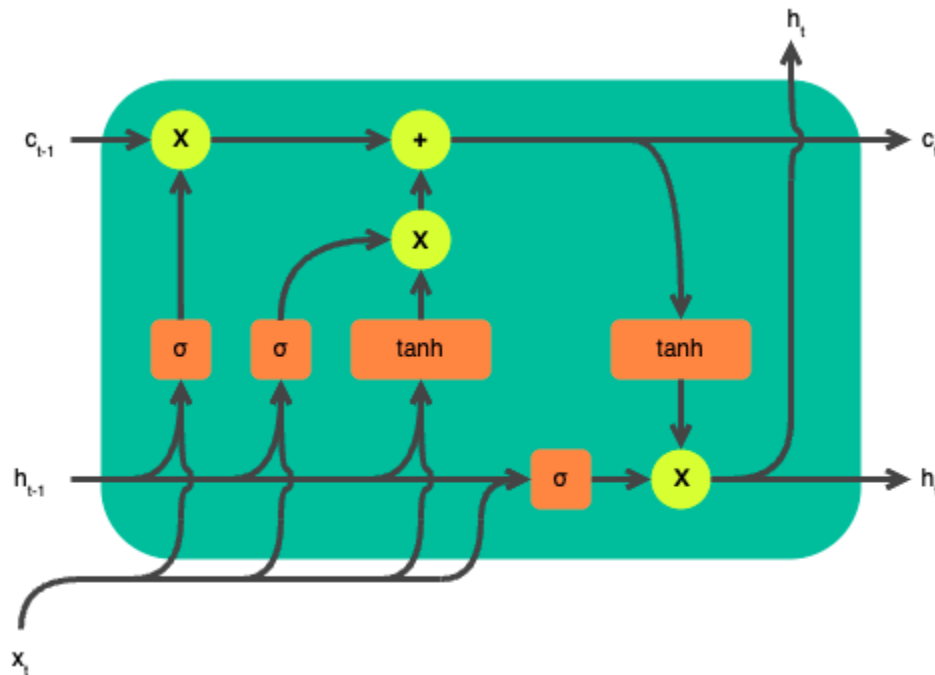


Figura 3: Representação do algoritmo LSTM⁶

3.3 Indicadores

3.3.1 Fundamentais

Indicadores fundamentais fornecem informações sobre o desempenho/valor de uma empresa ou ativo financeiro.

- SPY: Mais conhecido por S&P500, é o índice que representa as 500 maiores empresas do mercado acionista americano.
- VIX: *CBOE Volatility Index*, também conhecido por “Índice do medo”, é calculado com base na volatilidade das opções de compra e venda do S&P500. Na prática representa o nervosismo dos investidores quanto ao mercado. Quanto mais elevado for este valor, piores são as expectativas dos investidores para o mercado.
- DXY: *US Dollar Index* (Índice do Dólar dos EUA), representa a força da moeda dólar em contra um conjunto ou “cesto” de moedas estrangeiras.

⁶ Fonte: <https://www.machinelearningnuggets.com/tensorflow-lstm/>

3.3.2 Macroeconómicos

Os indicadores macroeconómicos fornecem informações sobre a saúde e desempenho da economia de uma forma geral.

- **IRX:** *CBOE 13 Week Treasury Bill Yield Index*, representa as taxas de juro dos Títulos do Tesouro dos EUA com vencimento a 13 semanas (ou trimestral). De forma geral dá aos investidores a perceção de risco no mercado a curto prazo e é usado como referência para investimentos de baixo risco no curto prazo.
- **FVX:** *CBOE 5 Year Treasury Note Yield Index*, representa as taxas de juro dos Títulos do Tesouro dos EUA com vencimento de 5 anos. De forma geral dá aos investidores a perceção de risco no mercado a médio prazo e é usado como referência para investimentos de baixo risco no médio prazo.
- **TNX:** *CBOE 10 Year Treasury Note Yield Index*, representa as taxas de juro dos Títulos do Tesouro dos EUA com vencimento de 10 anos. De forma geral dá aos investidores a perceção de risco no mercado a longo prazo e é usado como referência para investimentos de baixo risco no longo prazo.
- **EFFR:** *Effective Federal Funds Rate*, representa a taxa de juros à qual os bancos emprestam dinheiro entre si durante o dia nos EUA (taxa de juro interbancária). É usada pelo FED para controlar a política monetária e influenciar a atividade econômica. De forma resumida, um aumento/redução das taxas de juro representa um aumento/redução no custo de novos créditos e créditos de taxa variável, tornando o crédito mais caro/barato e restringindo/estimulando a economia.
- **UNRATE:** *Unemployment Rate*, representa a taxa de desemprego dos EUA. É usado como indicativo da saúde econômica de um país porque quando a taxa de desemprego é alta, pode indicar fraqueza na economia e de forma análoga, quando uma taxa baixa pode indicar robustez na economia.
- **UMCSENT:** *University of Michigan Consumer Sentiment Index*, mede o sentimento dos consumidores em relação à economia. Aumento deste valor sugere um maior otimismo dos consumidores, que pode refletir-se num aumento de consumos, que desta forma aumenta atividade econômica. Decréscimo do deste valor sugere naturalmente o oposto.

3.3.3 Técnicos

Indicadores técnicos fornecem informações mais detalhadas sobre o desempenho/valor de uma empresa ou ativo financeiro, mais concretamente pela análise de tendências, gestão de risco, sinais de compra ou venda. Essencialmente ajudam a identificar padrões.

- **MACD:** *Moving Average Convergence Divergence*, é um indicador de tendência. Ajuda a identificar a força e a direção da tendência do mercado. Valores positivos sugerem uma subida do mercado e de forma análoga valores negativos indicam uma tendência de queda do mercado.
- **MACD Signal:** É a média móvel exponencial do MACD, que suaviza as flutuações. Em conjunto com a linha MACD quando esta cruza acima da linha MACD Signal, é visto como um sinal potencial de compra, enquanto um cruzamento abaixo da linha MACD Signal é considerado um sinal potencial de venda.
- **MACD Histograma:** É calculado pela diferença entre o valor de MACD e o MACD Signal. Quando o valor é maior que zero, indica que o MACD está acima do MACD Signal, que representa um sinal potencial de compra.

- *ATR: Average True Range*, é um indicador de volatilidade. Mede a faixa média de variação dos preços num determinado período.
- *RSI: Relative Strength Index*, é um indicador de momento. Mede a tendência das compras e vendas dos ativos, num determinado período. Os valores variam de 0 a 100, em geral os investidores consideram que acima de 70 o ativo está sobre comprado e abaixo de 30 é considerado que está sobre vendido.
- *Volume*: Indica o número de vezes que um ativo financeiro em específico foi negociado num determinado período de tempo.
- *OBV: On-Balance Volume* é um indicador cumulativo do Volume no mesmo período de tempo. Essencialmente usado para ajudar a identificar tendências no mercado financeiro.

CAPÍTULO 4

Metodologia

4.1. Indicadores

Com base na investigação feita para esta dissertação e nos conhecimentos que adquiridos sobre o mercado de ações, identificou-se um conjunto de indicadores, amplamente usados por investidores na expectativa de prever a tendência dos mercados. Estes indicadores podem-se categorizar em indicadores fundamentais, macroeconómicos e técnicos.

A maioria dos indicadores foram obtidos diretamente pelo *Yahoo Finance*, ou através de transformações destes mesmos dados. A exceção foram os indicadores macroeconómicos EFFF, UNRATE e UMCSENT que necessitaram ser extraídos do site da *Federal Reserve Economic Data* (FRED). Além do registo no site é necessário pedir autorização da FRED para que através de uma API possa-se exportar os dados diretamente em Python (facilitado pela biblioteca “fredapi”).

Indicadores	Símbolo	Fonte	Frequência
Fundamentais	DXY	Yahoo Finance	Diário
	SPY		
	VIX		
Macroeconómicas	FVX	Yahoo Finance	Diário
	IRX		
	TNX		
	EFFF	FRED	Diário
	UMCSENT		Mensal
	UNRATE		
Técnicos	ATR	Yahoo Finance	Diário
	MACD		
	MACD Histograma		
	MACD Signal		
	OBV		
	RSI		
	Volume		

Quadro 1: Indicadores, Fonte e Frequência dos registos

Os indicadores técnicos foram obtidos pelos valores do SPY (‘Close’, ‘High’, ‘Low’, ‘Volume’) e os parâmetros usados em cada indicador foram os *default*, ou por outras palavras os mais popularmente usados para cada indicador.

Técnicos	Parâmetros Default (período em dias)
MACD	período curto EMA = 12 período longo EMA = 26 período media movel = 9
MACD Signal	período = 9
ATR	período = 14
RSI	período = 14

Quadro 2: Parâmetros *default* usados em cada Indicador Técnico

Usando também o indicador SPY acrescentou-se aos indicadores duas variáveis de desfasamento, que representam o indicador SPY, mas observado um dia e uma semana no passado. Iremos avaliar se é vantajoso permitir que o modelo faça previsões, levando em consideração observações passadas em simultâneo com as atuais.

4.2. Tratamento de dados

É importante notar que os indicadores UNRATE e UMCSSENT, são reportados mensalmente, então para podermos incorporar estes indicadores na análise foi necessário transformar a frequência dos registos para diária. Isto foi alcançado assumindo que o valor reportado não muda ao longo do mês, até que haja um novo reporte. A representação destes indicadores torna-se assim uma função não continua.

Posteriormente, a totalidade dos indicadores foram agrupados num único *dataframe*, garantindo a consistência dos índices.

Em seguida, foi aplicada a matriz de correlação de Pearson aos indicadores para medir a relação linear entre cada par. Observamos que os pares EFFR/ IRX e FVX/ TNX têm uma correlação de 0.99 entre si, ou seja, uma relação positiva quase perfeita. Isso sugere que irão contribuir a mesma informação, aumentando desnecessariamente a complexidade e recursos usados no modelo.

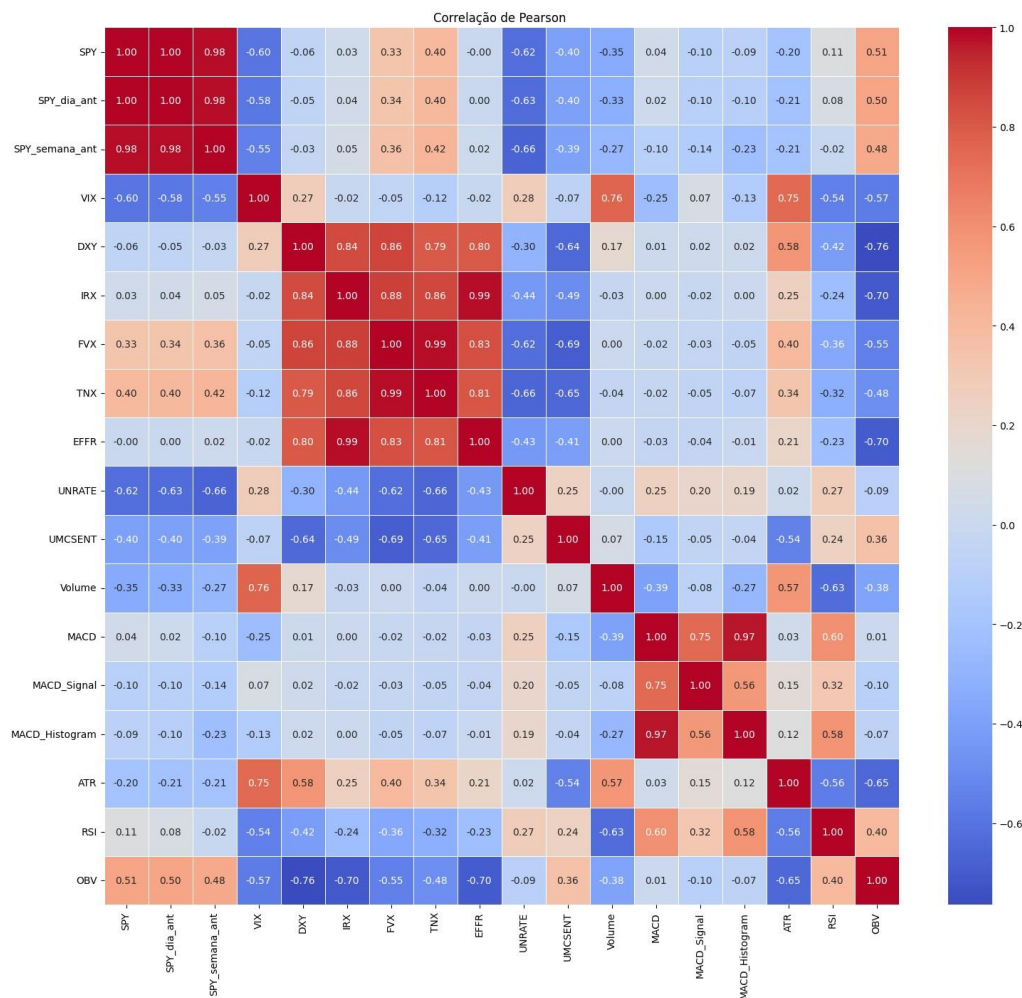


Figura 4 – Matriz de correlação de Pearson dos Indicadores

Analisando do ponto de vista económico, faz sentido EFR e IRX apresentem uma elevada correlação, porque ambas representam alterações à política monetária de taxas de juro no curto prazo. Em detalhe, EFR representa a taxa de juro interbancária definida pela FED nos EUA. Quando se decide baixar/aumentar esta taxa, resulta numa diminuição/aumento das taxas de juro que os bancos praticam, diminuindo/aumentando a procura títulos de Tesouro de curto prazo, que por sua vez resulta na diminuição/aumento do valor do IRX.

De forma semelhante, faz sentido os indicadores FVX e TNX terem uma elevada correlação, já que representam as taxas de juro dos Títulos do Tesouro dos EUA com vencimento no médio e longo prazo respetivamente. Assim sendo, ambos são sensíveis às mesmas decisões de política monetária, expectativas de investidores e fatores económicos.

Para decidir quais indicadores manter foi efetuado uma análise de variância entre cada par, excluindo o indicador com maior variância. Manteve-se no modelo os indicadores EFR e TNX.

Também observamos que o par MACD/MACD Histograma têm uma correlação de 0.97 entre si. Pela análise de variância manteve-se no modelo o indicador MACD Histograma. Esta elevada correlação é justificada porque o indicador MACD Histograma foi calculado a partir do MACD.

A seguir foram excluídos os registos com pelo menos um valor *null*/omisso. Estes valores resultam da existência de pequenas diferenças nas datas dos indicadores e porque alguns dos indicadores técnicos necessitam de registos de histórico para poderem ser calculados. Por exemplo, o RSI só pode ser calculado a partir do décimo quarto registo. No final ficaram elegíveis para análise um total de 680 registos.

As diferentes escalas dos valores dos indicadores, implica a necessidade de normalização dos dados, porque caso contrário os indicadores com escalas maiores iriam ser sobre representadas na sua significância para o modelo. Usar o normalizador Min-Max entre valores de 0 e 1 resolve esta questão e preserva as relações entre os dados e a continuidade temporal.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

No entanto, é preciso notar algumas especificações que podem prejudicar o modelo. O normalizador é sensível a *outliers* e também está a limitar indicadores que em teoria não têm limite superior. Tendo isto em conta é razoável o uso do normalizador Min-Max, porque a sua aplicação aos dados continua a ter balanço positivo para o modelo.

4.3. Treino do modelo

Como estão a ser analisadas series temporais, é crucial para o modelo LSTM usar a abordagem da janela deslizante (*rolling window*), porque facilita o processo de aprendizagem, já que permite o modelo LSTM use sequencias temporais de dimensão ‘n’ pré-definido, preservando a ordem e continuidade temporal. Nesta abordagem, o período temporal usado foi o equivalente a 10 registos que na prática representa duas semanas (equivalente a 10 dias úteis), permitindo que o modelo se adapte a tendências neste período.

O próximo passo é a divisão dados entre treino e teste, tendo em consideração que é essencial não perder a continuidade temporal dos registos, já que o algoritmo irá tentar captar padrões temporais.

Usando outras análises como referência é comum o uso da regra dos 80% para treino e 20% para teste. Outra estratégia comum é adicionar mais uma divisão, acrescentado um conjunto para validação, dividindo de dados por exemplo em 80% treino, 10% validação e 10% teste. Dito isto, uma divisão à priori dos dados entre treino e teste não é a melhor abordagem, porque a decisão de onde fazer a divisão em si arbitrária e pode afetar drasticamente o desempenho do modelo.

Conseguimos ultrapassar esta limitação, usando o método de validação cruzada TSCV (*Time Series Cross-Validation*), dividindo os dados em treino e teste 'n' vezes ('n' número pré-definido de divisões). Para esta análise escolheu-se 4 divisões, ou seja, os dados serão divididos em quatro conjuntos distintos de treino e teste, permitindo avaliar o desempenho do modelo em quatro combinações diferentes.

4.4. Especificações do modelo LSTM

O algoritmo LSTM encontra-se pré-definido em Python e pode ser facilmente importado pela biblioteca 'tensorflow.keras'. No entanto, o principal desafio na construção do modelo é a escolha dos hiperparâmetros, onde existe uma grande flexibilidade para adaptar especificações consoante o objetivo do modelo.

Desta forma, definimos uma função que tem como parâmetros sequências de registos (*rolling window*) 'X' (*input features*) e 'y' (*target*) para treino e teste, e os hiperparâmetros *neurons*, *epochs*, *batch_size*, *dropout_rate*, *activation_function*, *optimizer*, *loss_function* e *L2 regularization*.

Um problema comum neste tipo de modelos é o *overfitting*. Acontece quando o modelo se adapta em excesso aos dados de treino e depois tem dificuldades na fase de previsão. Desta forma, adotou-se medidas preventivas como a utilização dos parâmetros *dropout* e regularizador L2.

O parâmetro *dropout* é usado para desativar aleatoriamente uma percentagem pré-definida dos neurónios durante o treino do modelo, evitando assim que o modelo memorize em excesso os dados de treino, que por sua vez reduz o *overfitting*.

O parâmetro regularizador L2 é usado para assegurar que os pesos (*weights*) das variáveis na camada de *Output* não são demasiado grandes, ou seja, reduz a capacidade do modelo se adapte em excesso aos dados (*overfitting*).

Outro problema comum é o elevado tempo de processamento no *tuning* dos hiperparâmetros. Esta situação é difícil de resolver na totalidade, mas podemos otimizar o modelo. Implementou-se assim o método *Early Stopping* que permite que o treino do modelo seja interrompido caso não existam melhorias significativas no valor de erro (*validation loss*) durante 10 épocas (*epochs*). Estes parâmetros podem ser ajustados consoante as necessidades.

O modelo LSTM desenvolvido apenas possui uma *hidden layer* para diminuir a complexidade do algoritmo, diminuindo também assim o tempo de *tuning* dos hiperparâmetros. Numa análise posterior iremos avaliar se existe melhoria significativa do resultado aumentando o número de *hidden layers*.

Para o cálculo do grau de confiança do modelo entre os valores previstos e os valores reais, definimos o erro quadrático médio RMSE (*Root Mean Square Error*). Escolhemos esta métrica por ser mais amplamente usada neste tipo de análises, já que penaliza mais os *outliers*, do que por exemplo o erro médio absoluto MAE (*Mean Absolute Error*).

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

Em que, a cada registo, de 1 a N observações:

- y_i : i -ésimo valor original da serie temporal
- \hat{y}_i : i -ésimo valor previsto pelo modelo
- N : Número de observações.

4.5. Tuning dos Hiperparâmetros

A escolha dos hiperparâmetros é a essência de um bom modelo LSTM. Idealmente percorreríamos todas as combinações de hiperparâmetros possíveis, aplicando-as ao modelo e saberíamos em concreto qual o melhor conjunto. No entanto, isto não é uma abordagem realista.

Se assumirmos por exemplo, que queremos avaliar 8 hiperparâmetros e que cada um possui apenas 3 valores possíveis. Adicionando a validação cruzada com 4 divisões através do método TSCV, resultaria em $3^8 \times 4$, ou seja, 26.244 possíveis combinações. Se também assumirmos que uma máquina demora cerca de 10 segundos a processar o modelo por conjunto, então a *tuning* levaria aproximadamente 3 dias para concluir o processo computacional.

No entanto, após esse período, temos 3 dias adicionais de registos, tornando os resultados obsoletos. Neste sentido temos duas hipóteses: melhorar o processo de seleção de hiperparâmetros ou aumentamos o poder computacional.

A primeira abordagem é claramente a melhor pois qualquer aumento no número de hiperparâmetros ou aumento na complexidade do modelo, resultaria no constante aumento do tempo de processamento, o que se torna irrealista.

A solução escolhida foi usar o método de *Random Search* (pesquisa aleatória), que gera ‘n’ conjuntos aleatórios de hiperparâmetros para alimentar o modelo, podendo assim selecionar com melhor resultado entre eles (‘n’ número de conjuntos definido à priori).

A lista de hiperparâmetros analisada foi a seguinte:

- Nº de neurónios: [16, 32, 64, 128, 192, 256]
- Nº de épocas: [8, 16, 32, 64, 96]
- *Batch Size*: [4, 8, 16, 32]
- Taxa de *Dropout*: [0.2, 0.3, 0.4]
- Função de ativação: ['relu', 'tanh']
- Otimizadores: ['Adam', 'Adadelta', 'Nadam', 'RMSprop', 'Adagrad']
- *Loss Function*: ['mean_squared_error', 'mean_absolute_error']
- Parâmetro de regularização L2: [0.01, 0.001, 0.0001]

Algo que pode saltar à vista é a ideia de que se encontrarmos um bom resultado com o método *Random Search*, então poderíamos simplesmente alterar os hiperparâmetros à volta desse ponto e encontrar o ponto ótimo. Isto é o que procuramos fazer na análise de sensibilidade, mas não é correto construir o modelo dessa forma pois se apenas procurarmos minimizar a *loss function* facilmente ficaríamos “presos” num ponto ótimo local (mínimo local).

Durante a *Random Search* foi analisado cerca de 10% do conjunto de combinações possíveis de hiperparâmetros (total 15.120), para cada uma das 4 divisões distintas da validação cruzada TSCV.

Minimizando o valor do RMSE entre os valores previstos pelo modelo e os valores reais, fazendo o corte na amostra a 17 de junho de 2022, sendo 544 registros para treino e 136 para teste (que equivale a 80% e 20% da amostra para treino e teste respectivamente), o conjunto ótimo de hiperparâmetros tem os seguintes valores:

- N° de neurónios: 192
- N° de épocas: 96
- *Batch Size*: 32
- Taxa de *Dropout*: 0.2
- Função de ativação: tanh
- Otimizador: Adam
- *Loss Function*: mean_squared_error
- Parâmetro regularização L2: 0.0001

Com valor RMSE igual a 0.1582.

Análise de Sensibilidade

Após obter o melhor conjunto de hiperparâmetros vamos analisar a robustez do modelo e caso necessário, ajustar os parâmetros.

Novamente, o objetivo será minimizar o RMSE entre os valores previstos pelo modelo e os valores reais dos dados. Para garantir que os resultados obtidos pelo cálculo do RMSE não são situações isoladas, será executada cada alteração 3 vezes e no final calculado a média dos resultados para reduzir a variância.

Consideremos como melhoria significativa a redução em pelo menos 0.01 no valor do RMSE.

Devido à grande discrepância de valores entre as escalas de cada indicador, a análise e cálculo do RMSE serão feitos com base na escala normalizada, com valores entre 0 e 1.

5.1 Relevância de Indicadores

A escolha dos indicadores é também uma parte crucial do modelo, porque caso existam indicadores que não estão a contribuir para o resultado, no mínimo estão a prejudicar a performance do modelo, já que quanto maior o número de indicadores, maior será a complexidade e tempo de processamento, que além de tornar mais difícil interpretar os resultados, também pode levar a *overfitting*.

Além disto, também podem existir indicadores que estão diretamente a prejudicar o resultado.

Desta forma, vamos avaliar a relevância de cada indicador, procedendo à sua exclusão progressiva do modelo.

Após cada exclusão, calculamos o RMSE e caso este valor tenha variação:

- 1) Menor que 0.01 do RMSE obtido anteriormente, será considerado que o indicador não tem relevância para o resultado do modelo e será removido.
- 2) Superior a 0.01, será considerado que o indicador está a ter um impacto negativo no resultado e será removido.
- 3) Inferior a -0.01, será considerado que o indicador está a ter um impacto positivo no resultado e é mantido no modelo.

O progresso do valor do RMSE, à medida que cada indicador é excluído do modelo é apresentado no Quadro 3 (Coluna Remover N°1).

Pode-se observar o uma tendência decrescente progressiva no valor RMSE, que indica que a remoção da maioria dos indicadores tem pouca relevância no valor do RMSE, com exceção dos indicadores VIX e UNRATE.

O modelo foi simplificado aos indicadores SPY, VIX e UNRATE, com RMSE igual a 0.0328.

O resultado obtido é considerado aceitável, ainda assim, iremos voltar a aplicar o mesmo método para avaliar a relevância de cada indicador no resultado.

Pelo Quadro 3 (Coluna Remover N°2) podemos observar que o modelo melhora o valor RMSE para 0.0234, quando usa apenas com os indicadores SPY e UNRATE.

Indicadores	Remover Nº1		Remover Nº2		Adicionar	
	RMSE	Decisão	RMSE	Decisão	RMSE	Decisão
SPY_dia_ant	0.1574	Sim	-	-	0.0228	Não
SPY_semana_ant	0.1565	Sim	-	-	0.0255	Não
VIX	0.1729	Não	0.0362	Sim	0.0328	Não
DXY	0.1555	Sim	-	-	0.0321	Não
TNX	0.1539	Sim	-	-	0.0468	Não
EFFR	0.0771	Sim	-	-	0.0764	Não
UNRATE	0.0894	Não	0.0234	Não	-	-
UMCSENT	0.0772	Sim	-	-	0.0279	Não
Volume	0.0714	Sim	-	-	0.0620	Não
MACD Signal	0.0781	Sim	-	-	0.0328	Não
MACD Histograma	0.0646	Sim	-	-	0.0346	Não
ATR	0.0594	Sim	-	-	0.0286	Não
RSI	0.0434	Sim	-	-	0.0493	Não
OBV	0.0328	Sim	-	-	0.0434	Não

Quadro 3: Avaliação da Relevância dos Indicadores

Tendo em conta que a ordem de exclusão dos indicadores é de certa forma arbitrária, podem existir outros conjuntos de indicadores que tenham um melhor resultado, quando alteramos os hiperparâmetros, número de *hidden layers* ou o corte da amostra. Infelizmente não iremos abordar estas possibilidades devido aos elevados recursos computacionais que seriam necessários.

Ainda assim, iremos continuar a procurar o melhor conjunto de indicadores, de forma semelhante ao que foi feito anteriormente, mas adicionando progressivamente cada indicador ao modelo, analisando se existe alguma redução significativa (menor que 0.01) no valor do RMSE e caso se verifique adicionar o indicador ao conjunto ótimo.

Como pode ser observado no Quadro 3 (Coluna Adicionar), não houve melhoria.

Terminamos este ponto com o conjunto de indicadores SPY e UNRATE, valor RMSE igual a 0.0234.

Este já pode ser considerado um bom resultado, mas iremos prosseguir com a análise de sensibilidade.

5.2 Divisão da amostra

Neste momento faz sentido ajustar o valor que usamos como referência 0.01, para considerar que houve uma redução significativa no valor RMSE, já que a ordem de magnitude deste valor é próxima do RMSE atual 0.0234.

Desta forma passaremos a considerar que houve melhoria significativa se o valor RMSE diminuir cerca de 10%, ou seja, se a variação do valor for inferior a -0.002 do RMSE atual.

O método *Time Series Cross-Validation*, com quatro divisões, segmenta a amostra nos índices 136, 272, 408 e 544.

O ponto de divisão atual 544 divide a amostra em 80% treino e 20% teste. Mantendo constantes todos os outros parâmetros, realizou-se a alteração no ponto de divisão para 408 e 272, e observamos que não houve melhoria no valor do RMSE que registaram 0.0390 e 0.0567 respetivamente.

5.3 Janela Deslizante (*Rolling Window*)

Conforme referido anteriormente, no método janela deslizante usamos series temporais com dimensão igual a 10 registos (equivalente a duas semanas ou 10 dias uteis). Ao ajustar este valor para 5 e 20, também não melhorou significativamente o valor do RMSE (0.0261 e 0.0239 respetivamente).

5.4 Ajuste de Hiperparâmetros

Fazendo uma análise exaustiva das combinações de hiperparâmetros, à volta do conjunto atual, foi possível melhorar de forma significativa o valor RMSE para 0.0192.

Em particular, durante a análise notou-se que os melhores conjuntos usavam com frequência o parâmetro *dropout* com valor 0.2. Sendo assim decidiu-se baixar este valor para 0.1, apesar de não estar incluído na lista inicial de hiperparâmetros e de facto melhorou o resultado.

O novo conjunto ótimo tem os hiperparâmetros:

- N° de neurónios: 256
- N° de Épocas: 128
- *Batch Size*: 16
- Taxa de *Dropout*: 0.1
- Função de ativação: tanh
- Otimizador: Adam
- *Loss Function*: mean_absolute_error
- Parâmetro regularização L2: 0.0001

5.5 Número de *Hidden Layers*

Como já foi referido o modelo LSTM distingue-se pela sua capacidade de capturar dependências de longo prazo e aumentar o número de *hidden layers*, também aumenta a capacidade de memorização, que pode potenciar a capacidade do modelo a captar padrões mais complexos nas series temporais.

Naturalmente, aumentar a complexidade do modelo vai também aumentar a necessidade de mais recursos computacionais e maior tempo de processamento. Além disto o aumento de complexidade também pode aumentar o risco de *overfitting* e prejudicar o resultado do modelo.

Foi precisamente isto que aconteceu quando aumentamos o número de *hidden layers*, o valor do RMSE piorou para 0.0265 com duas e para 0.0269 com três *hidden layers*.

5.6 Modelo Ingênuo (*Naive Model*)

O modelo *Naive* representa um modelo definido de forma que a previsão atual é sempre igual à última observação registada. É usado para avaliar se os modelos de previsão em comparação, estão realmente a fornecer uma melhoria significativa nas previsões.

Avaliando novamente pelo RMSE, verificasse que o modelo *Naive* tem valor 0.0181, ou seja, um valor um pouco melhor que do nosso modelo segundo esta métrica.

CAPÍTULO 6

Conclusão

Esta dissertação teve como objetivo explorar o modelo LSTM sobre diferentes indicadores, com a expectativa de conseguir uma previsão satisfatória e que supere o modelo *Naive*.

Começemos por responder às questões propostas na introdução:

- A. Quais os indicadores e parâmetros mais relevantes para a previsão do S&P500, usando o modelo LSTM?

Os indicadores do resultado ótimo foram o SPY e o UNRATE, dividindo a amostra em 80% treino e 20% teste, apenas usando uma *hidden layers* no modelo LSTM, janela deslizante com series temporais de 10 registos e o conjunto de hiperparâmetros listado no ponto 5.4.

- B. O modelo que usa os indicadores mais relevantes supera a capacidade de previsão do modelo que usa a totalidade de indicadores?

Sim, após a análise de sensibilidade dos indicadores o valor RMSE do modelo com a totalidade dos indicadores reduziu drasticamente de 0.1582 para 0.0192.

O modelo LSTM pode estar a ajustar-se em excesso e captar o ruído dos dados (*overfitting*) ou o excesso de indicadores pode estar a prejudicar a capacidade de o modelo captar padrões nos dados.

- C. Aumentar o número de *hidden layers* melhora a capacidade de previsão do modelo?

Não, um maior número de *hidden layers* não está correlacionado com melhoria do modelo.

Este fator irá depender de cada modelo, consoante os desafios e necessidades. Mesmo quando existe uma melhoria gradual no valor RMSE, consoante o aumento do número de *hidden layers*, será sempre necessário haver uma análise posterior para avaliar se o aumento da complexidade e recursos necessários para o modelo, causado pelo aumento do número de *hidden layers*, justifica a melhoria do valor RMSE.

- D. O modelo LSTM tem maior capacidade de previsão que o modelo *Naive*?

O resultado do modelo LSTM pareceu promissor, mas por pouco não conseguiu superar o modelo *Naive*. Para ultrapassar este objetivo seria necessária uma análise mais aprofundada e com maiores recursos computacionais para ser possível explorar mais combinações e de maior complexidade, que não houve oportunidade de analisar nesta dissertação.

Também podemos analisar este resultado como sendo expectável, segundo a Hipótese dos Mercados Eficientes (Fama, 1970). Esta teoria define um mercado com eficiência “forte”, como um mercado que já tem refletido em si todas as informações existentes, quer sejam públicas ou privadas. Neste sentido é impossível desenvolver modelos com capacidade de obter lucros de forma consistente, porque qualquer nova fonte de informação seria refletida nos preços do mercado, deixando de ser relevante.

Podemos então concluir que cumprimos os objetivos principais da dissertação, apesar do resultado não ser completamente satisfatório.

É interessante observar que os indicadores SPY e UNRATE, foram os que melhor se ajustaram. Ou seja, o modelo sugere que o mercado acionista das 500 maiores empresas dos EUA é bastante influenciado pela taxa de desemprego, publicada pela agência do governo americano BLS (*Bureau of Labor Statistics*), que além destes dados, é responsável por publicar outras diversas estatísticas econômicas.

De forma mais aprofundada, um aumento do valor UNRATE representa um aumento da taxa de desemprego. Quando este aumento é significativo sugere que as populações em geral passam a ter menos fontes de rendimento, o que por sua vez, as obriga como consumidor a reduzir os seus gastos.

Esta perspetiva afeta diretamente a cotação das empresas no mercado de ações, influenciando por consequência o S&P500 (SPY). As empresas são avaliadas em grande parte com base nas expectativas de rendimentos futuros e capacidade de crescimento, logo a simples expectativa de que os consumidores irão reduzir os seus gastos, sugere que os lucros e crescimento das empresas, em geral, ficarão abaixo das projeções iniciais.

Perante esta realização, os mercados ajustam as valorizações das empresas de acordo, o que causa a desvalorização do SPY. De forma análoga, a diminuição do valor UNRATE pode refletir uma valorização do SPY.

A mudança de expectativas, que também é denominada como a incerteza sobre o futuro dos mercados. A relação desta incerteza com o desemprego foi explorada em detalhe por Choi & Loungani (2015).

Nesta análise apenas foi possível tocar na ponta do universo de conhecimento que existe à volta dos modelos de previsão. Portanto, sem sair muito do tema deixo algumas propostas para trabalhos futuros:

- Analisar outros indicadores como as *Fibonacci Retracemet Lines*.
- Treinar o algoritmo no muito curto prazo, com o período temporal máximo de 1 dia e frequência de 1 ou 5 minutos. A este tipo de estratégia dá-se o nome *Scalping*.
- Usar o algoritmo para criar sinais reais de “Compra”, “Venda” ou “Manter” e testar versus a estratégia de *Buy&Hold*.
- Desenvolver um modelo híbrido, complementando a capacidade preditiva do modelo LSTM com outro modelo de previsão.

Referências Bibliográficas

- Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications*, 113, 457–480. <https://doi.org/10.1016/j.eswa.2018.07.019>
- Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. C. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, 9, 100320. <https://doi.org/10.1016/j.mlwa.2022.100320>
- Choi, S., & Loungani, P. (2015). Uncertainty and unemployment: The effects of aggregate and sectoral channels. *Journal of Macroeconomics*, 46, 344–358. <https://doi.org/10.1016/j.jmacro.2015.10.007>
- Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. In *Neurocomputing* (Vol. 503, pp. 92–108). Elsevier B.V. <https://doi.org/10.1016/j.neucom.2022.06.111>
- Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), 383–417. <https://doi.org/10.2307/2325486>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129, 273–285. <https://doi.org/10.1016/j.eswa.2019.03.029>
- Li, K., Huang, W., Hu, G., & Li, J. (2023). Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network. *Energy and Buildings*, 279. <https://doi.org/10.1016/j.enbuild.2022.112666>
- Liang, L., & Cai, X. (2020). Forecasting peer-to-peer platform default rate with LSTM neural network. *Electronic Commerce Research and Applications*, 43, 100997. <https://doi.org/https://doi.org/10.1016/j.elerap.2020.100997>
- Liang, L., & Cai, X. (2022). Time-sequencing European options and pricing with deep learning – Analyzing based on interpretable ALE method. *Expert Systems with Applications*, 187, 115951. <https://doi.org/https://doi.org/10.1016/j.eswa.2021.115951>
- Lin, Y., Lin, Z., Liao, Y., Li, Y., Xu, J., & Yan, Y. (2022). Forecasting the realized volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM. *Expert Systems with Applications*, 206. <https://doi.org/10.1016/j.eswa.2022.117736>
- Lv, S.-X., Peng, L., Hu, H., & Wang, L. (2022). Effective machine learning model combination based on selective ensemble strategy for time series forecasting. *Information Sciences*, 612, 994–1023. <https://doi.org/https://doi.org/10.1016/j.ins.2022.09.002>
- Mcculloch, W. S., & Pitts, W. (1990). A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY* n. In *Bulletin of Mothemnticnl Biology* (Vol. 52, Issue 2).
- Michańków, J., Sakowski, P., & Ślepaczuk, R. (2022). LSTM in Algorithmic Investment Strategies on BTC and S&P500 Index. *Sensors*, 22(3). <https://doi.org/10.3390/s22030917>

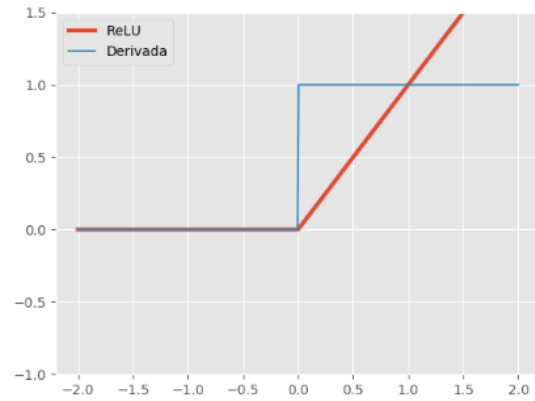
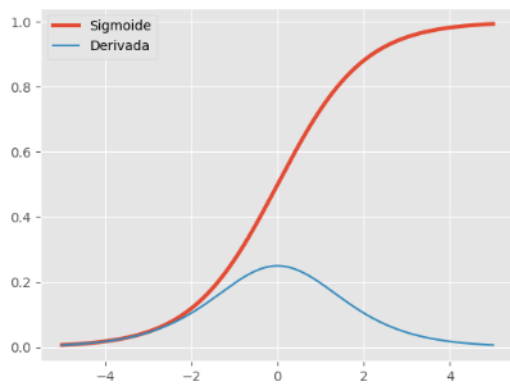
- Park, H. J., Kim, Y., & Kim, H. Y. (2022). Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing*, 114. <https://doi.org/10.1016/j.asoc.2021.108106>
- Parvini, N., Abdollahi, M., Seifollahi, S., & Ahmadian, D. (2022). Forecasting Bitcoin returns with long short-term memory networks and wavelet decomposition: A comparison of several market determinants. *Applied Soft Computing*, 121. <https://doi.org/10.1016/j.asoc.2022.108707>
- Pascanu, R., Mikolov, T., & Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. <http://arxiv.org/abs/1211.5063>
- Prasad, A. (2019). Prediction and Analysis of Time Series Data using Tensorflow. <https://towardsdatascience.com/prediction-and-analysis-of-time-series-data-using-tensorflow-2136ef633018>
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). 1986 Learning representations by back-propagating errors - David E. Rumelhart, Geoffrey E. Hinton e Ronald J. Williams. Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA, 323.
- Sang, C., & Di Pierro, M. (2019). Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network. *Journal of Finance and Data Science*, 5(1), 1–11. <https://doi.org/10.1016/j.jfds.2018.10.003>
- Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, 04(12), 310–316. <https://doi.org/10.33564/IJEAST.2020.v04i12.054>
- Wang, K., Yu, M., Niu, D., Liang, Y., Peng, S., & Xu, X. (2023). Short-term electricity price forecasting based on similarity day screening, two-layer decomposition technique and Bi-LSTM neural network. *Applied Soft Computing*, 136. <https://doi.org/10.1016/j.asoc.2023.110018>

Anexos

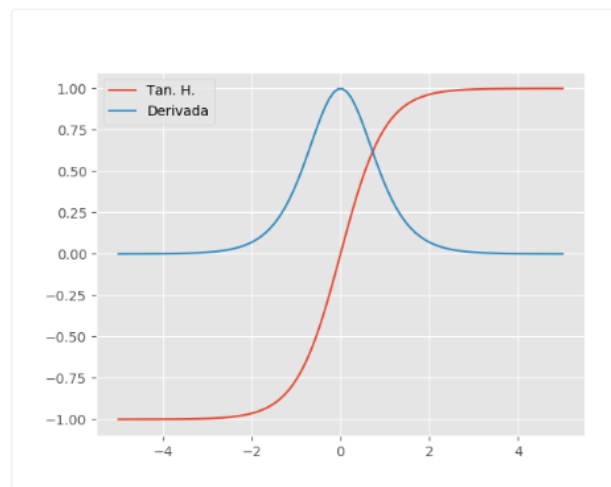
Figura 5: Representação das Funções Ativação Sigmoide, ReLU, TanH, e respectivas derivadas ⁷

$$\sigma(x) = \frac{1}{1 + e^x} \quad \sigma'(x) = \sigma(x)(1 - \sigma(x))$$

$$\text{ReLU}(x) = \max\{0, x\} \quad \text{ReLU}'(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{c.c.} \end{cases}$$



$$\tanh(x) = 2\sigma(2x) - 1 \quad \tanh'(x) = 1 - \tanh^2(x)$$



⁷ Fonte: <https://matheusfacure.github.io/2017/07/12/activ-func/>

Figura 6: Representação gráfica dos indicadores normalizados EFR, UNRATE e UMCSNT

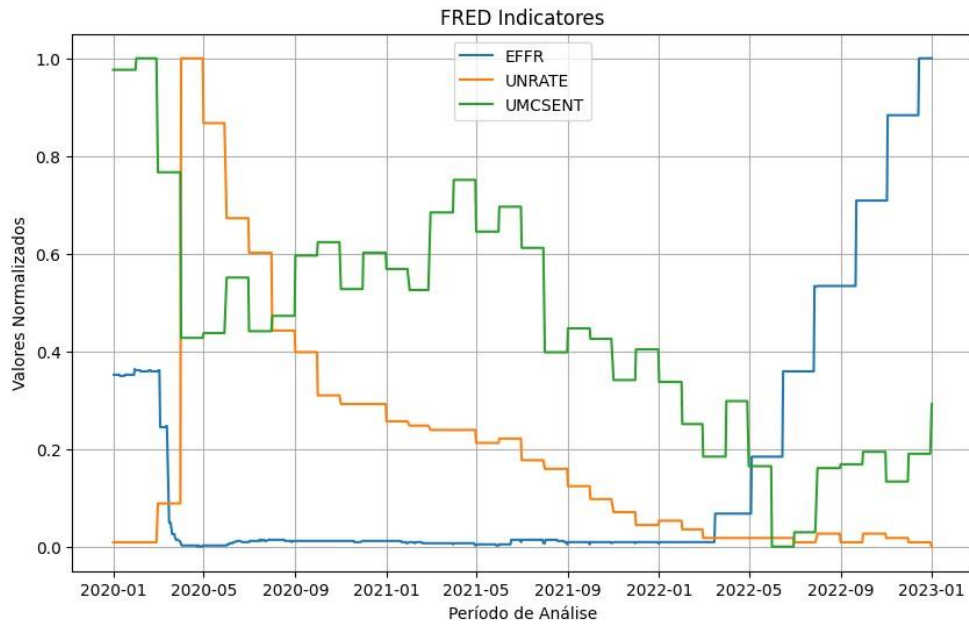


Figura 7: Representação gráfica do indicador SPY, da previsão LSTM e da previsão Naive

