Lisbon School of Economics & Management
Universidade de Lisboa

**Master**

Actuarial Science

# Master's Final Work

Internship Report

## Machine Learning Frameworks for Retention Models: An Application to a Motor Insurance Portfolio.

Diana Alexandra Montrond

Supervisors:

João Manuel Andrade e Silva

Francisco Alier Valentim Nogueira

October-2021

# Acknowledgements

The accomplishment of this project would not be possible without the contribution of the ones I mentioned here.
Firstly, I would like to express my gratitude to my supervisor João Manuel Andrade e Silva for presenting me the opportunity to do my internship at Liberty Seguros and for the exceptional and exhaustive help during the development of this report.
I would also like to thank all my co-workers from the Demand Modelling Team at Liberty for the patience and guidance. With a special thanks to my Supervisor, Francisco Nogueira, for giving me the idea for this project and the support to conclude it, as well as for being a kind and generous person during the whole period.
Finally, I want to thank my family and friends for the unconditional support during all my academic years and to challenge me to achieve more objectives. Special thanks to my parents and sisters for always being by my side and believing on my skills. Last but not least, a immense thank you to my great friend Mariana da Luz for being an amazing co-worker and companion during this new challenge, who listened to my struggles and helped overcome them.

# Abstract

In insurance market, predicting the retention is one key step to define strategies to obtain more clients as well as understand possible features that can prevent cancellation from current clients. Therefore, it exists a demand to have the best efficient modelling system that would lead to determinate better the needs of the market.
This project is one of those attempts, it selects three different models to be compared and chooses which of them behaves the best. The selected three models are: One belongs to the Generalised Linear Model (GLM), the Logistic Regression and two Machine Learning Algorithms, the Extreme Gradient Boosting (xGBoost) and the Light Gradient Boost (Light GBM). To evaluate those models, classifications such as Area Under the Curve (AUC), Receiver Operating Characteristic Curve (ROC Curve) and Accuracy were used as well as the McNemar's Test. This report also investigates ways to gain better knowledge about the Motor Insurance Portfolio, performing an Exploratory Data Analysis. It also covers methods of Feature Engineering to deal with data issues, such as missing data and with categorical variables. To obtain further improvements of the data, it was suggested two types of methods of training and testing split.
This analysis shows that the most efficient training and testing methods is randomly select $80\%$ of each month of the whole data set as a training set and the remaining as a testing set. It also concludes that the best retention model is the Light Gradient Boost, with very small difference from the Extreme Gradient Boosting. Those Machine Learning Algorithms performed significantly better than the traditional Logistic Regression. The model evaluations performed were mostly successful in discovering the differences between the models.

**Keywords:** Motor Insurance, Retention, Logistic Retention, Machine Learning, Extreme Gradient Boosting, Light Gradient Boost, Training and Testing, AUC, ROC Curves, McNemar's Test.

# Resumo

No mercado de seguradoras, prever a retenção é uma etapa importante para definir estratégias de forma a obter mais clientes, tal como, entender possíveis características que possam prever o cancelamento dos clientes atuais. Sendo assim, existe a necessidade de obter modelos mais eficientes e rigorosos para a necessidades do mercado.
Este projeto é uma dessas tentativa pois seleciona três modelos diferentes para serem comparados e escolhe o que têm melhor resultados. Os três modelos selecionados são: Um pertence ao Modelos Lineares e Generalizados (GLM), a Regressão Logística, e dois modelos de Machine Learning Algorithms, o Extreme Gradient Boosting (xGBoost) e o Light Gradient Boost (Light GBM). Para avaliar os modelos, classificações como Area Under the Curve (AUC), Receiver Operating Characteristic Curve (ROC Curve) e Exatidão são usados tal como o Teste de McNemar. Este relatório também investiga maneiras de aprofundar o conhecimento da carteira de seguro de automóvel, realizando uma Análise Exploratória dos Dados. Métodos de Feature Enginering foram usados para lidar com problemas de valores omissos e variáveis categóricas. Com objetivo de melhor os dados, foi sugerido dois tipos de métodos para a divisão de treino e teste.
Nesta análise foi descoberto que o método mais eficiente de divisão de treino e teste é selecionar aleatoriamente $80\%$ de cada mês para toda a base de dados como conjunto de treino e o restante como conjunto de teste. Também se encontrou o melhor modelo de demanda que é o Light Gradient Boost, com diferenças mínimas do Extreme Gradient Boosting. Os modelos de Machine Learning Algorithms obtiveram resultados mais significativos que a tradicional Regressão Logística. A comparação dos modelos foi bem-sucedida em desvendar as diferenças entre os modelos.

**Keywords:** Seguro Automóvel, Retenção, Regressão Logística, Machine Learning, Extreme Gradient Boosting, Light Gradient Boost, Treino e Teste, Correlação de Person, Cramér's V, AUC, ROC Curves, Teste de McNemar.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# 1 Introduction

The focus of this report is to address the project made during the internship at Liberty Seguros for the Portugal Demand Modelling team affiliated with Liberty Mutual Western European Market. The internship had a duration of 5 months, from March 2021 to July 2021.
The primary assignment of this experience was to develop this project as a Master's Final Work, the secondary assignments involved understanding the Motor Business and Motor Data in Portugal, develop skills in software such as Earnix and Python for modelling and deploy Demand models namely for Retention, Churn and Negotiation models. It also included validating Risk Models, both Frequency and Severity Models, and finalising with an impact analysis of the Risk Models and Demand Models on the companies objectives.
This project mainly consisted in exploring several tasks in order to perform a better estimation of the client's retention for the motor insurance portfolio. The tasks are: Researching better ways to distribute the data set in a diverse way to obtain as most information as possible; Find the best features to predict the retention; Explore different machine learning algorithms; Compare our results to the standard and usual regressions approaches in the modelling world.

## 1.1 Motivation

The main focus of this project consisted in demystifying the "black-box" characteristic that Machine Learning models tend to have for the insurance models as well as exhibit the performing results when compared to more usual models, like the Logistic Regression.

## 1.2 Objectives

The objectives of this project are listed bellow in a sequential order of how they are being addressed in this report:

- To carry out an Exploratory Data Analysis of a motor retention portfolio data set with the aim of understanding possible features used for retention models in insurance business. Additionally conduct feature engineering to comply with missing values and with categorical variables for Machine Learning Algorithms.
- To investigate different methods for splitting the data into training and testing mainly used to validate the performance of the models.
- To implement different frameworks while exploring algorithmic options to predict the retention when using a Motor Insurance portfolio. The main frameworks are the Extreme Gradient Boosting Algorithm (xGBoost) and Light Gradient Boost Algorithm (Light GBM) along with contrasting them with a Logistic Regression.
- To compare the models implemented by using classifications such as ROC Curves, AUC and Accuracy, and by using more traditional statistical tests such as the McNemar's Test with the purpose of evaluating the best possible model for predicting the retention.

## 1.3 Structure

This report is organised in 5 chapters, Chapter 1 is the Introduction, Chapter 2 refers to the software used through the analysis as well as detailed data description. Chapter 3 involves an explanation of the methods used during the Exploratory Data Analysis together with the exhibition and interpretation of the results. This Chapter also mentions the variable treatment of the covariates used in the modelling stage and presents the different methodologies of partition applied in the data set.
Next Chapter 4 includes the presentation of the three models, Logistic Regression, Extreme Gradient Boosting and Light Gradient Boost, as well as the methodologies of model evaluation, classification and tests. To finalise this chapter, the results and their interpretation were presented. Chapter 5 is a conclusion of the findings of this report along with the possible next steps.

# Chapter 2

# 2 Software and Data

The software SAS [1] was used to obtain the data set which had already been trough some cleaning and treatment by the IT team of the Liberty Seguros, to ensure the highest quality to the following analysis. The analysis and modelling for the retention was performed in the software Python [2].

The data set includes the Liberty's client portfolio of twelve months of renewals. In this database twenty variables were selected. Our data base contains $14$ quantitative and $6$ qualitative variables.

The quantitative variables are: Feature 2, Feature 3, Feature 7, Feature 8, Feature 9, Feature 11, Feature 12, Feature 13, Feature 14, Feature 15, Feature 16, Feature 17, Feature 18 and Feature 20. These values hold the information of Premium, Agent and Client related details, such as Loss Ratio and Renewals and Population characteristics.

In terms of qualitative variables, they are: Feature 1, Feature 4, Feature 5, Feature 6, Feature 10 and Feature 19. These variables contain the following information: Historical and Payment Behaviour, Client details and Bonus-Malus organisation. There is a single variable that is a date variable named $year\ month\ renew$, which relates the month of renew of the policyholder.

It is important to mention that due to confidentiality reasons these variables have a small description of their true meaning and have suffered some changes in the values, the continuous variables were multiplied by a factor and for the categorical variables the respective levels were transformed in anonymous levels, therefore it could provide some privacy and security for the company's when disclosure some of the results mentioned in the succeeding chapters.

[1] https://support.sas.com/en/software/enterprise-guide-support.html

[2] https://www.python.org/

# Chapter 3

# 3 Exploratory Data Analysis

Before modelling there is an important step that should be done, namely to investigate and to understand our data set so we can choose the best options when modelling. We also need to pay attention to possible drawbacks of some variables. Given this, the focus of this section will be analysing and interpreting the behaviour of the covariates used in a retention portfolio for a motor insurance business along with presenting the methods used to solve various problems that happen at this stage.

## 3.1 Individual Analysis

The exploratory data analysis will mainly consist in displaying visualisations for one of each type of covariate against the target variable.
Considering the categorical variables, the visualisations are not as complex as for the non-categorical, a possible visualisation is the bar chart. To extract more insights of each level distribution by the target variable, the variables were also plotted against the target giving for each level the percentage of retention, showing possible differences among levels.
For numerical variables it was used the box-plot and histogram.

### 3.1.1 Results

Due to confidential purposes we will only show the analysis for some variables although all variables went through the same process.
Figure 1 illustrates how the client's portfolio retention behaves during the twelve months. This graph shows the percentage of the the retention(in the red line and right scale) and the exposure(in the bars and left scale) of each month. This can be seen bellow.

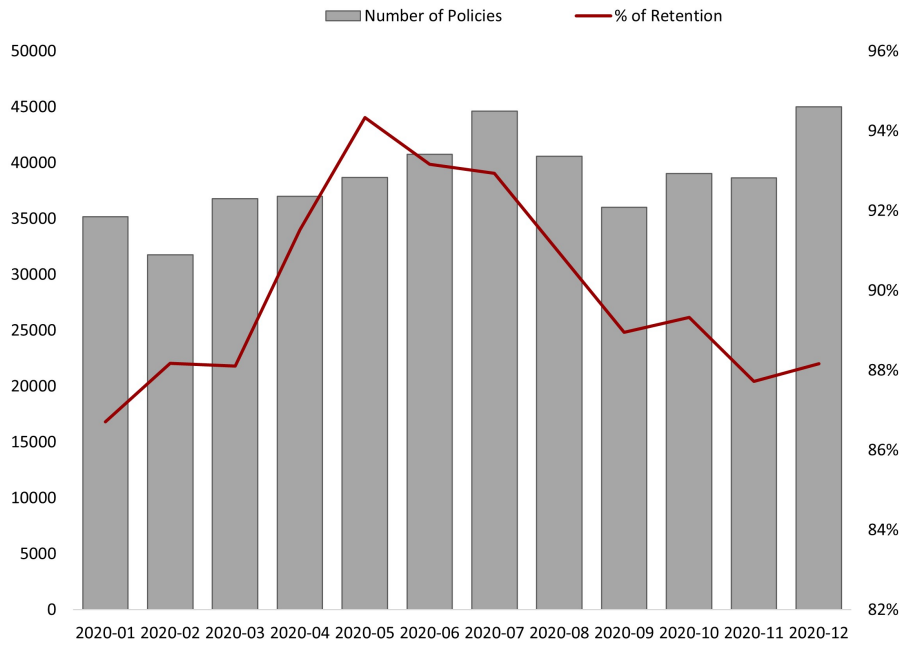


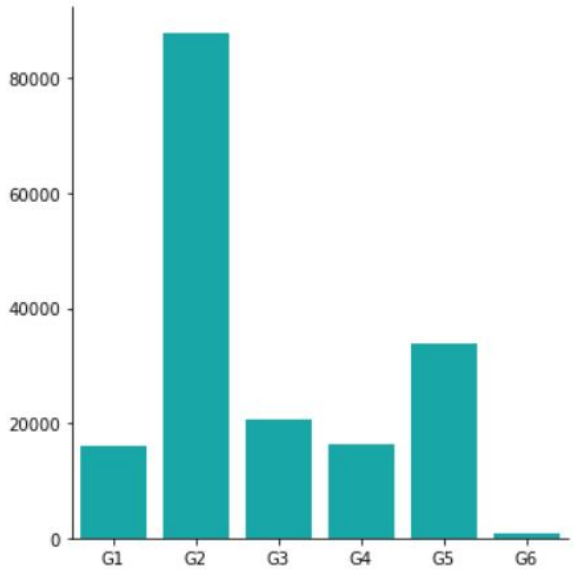

FIGURE 1: Combine bar and line chart for the number of policies and percentage of retention against the month of renew.

Figure 1 shows that during the twelve months the number of policies under renewals are more or less constant, reaching two peeks in July and December, although for the percentage of retention it can be seen that higher rates happen in May and June while the lower rates happen in January and November. Even tough the graph scale may indicate a very high difference between the months, once looked to the right scale, it shows that it is between $86.70\%$ and $94.33\%$. Given that, the proportion of retention are not very volatile. This is reinforced by the minimum and maximum having less than $8\%$ of difference. The mean percentage of retention is $90\%$.

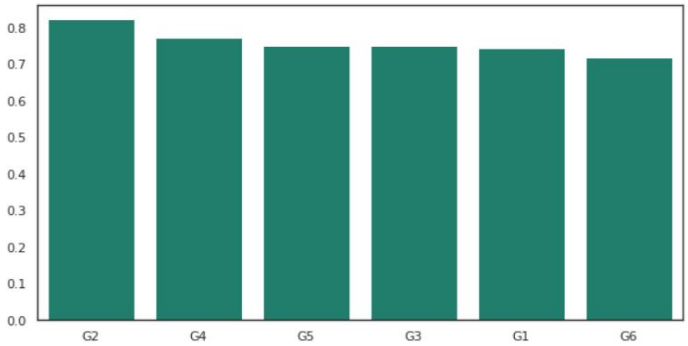
To illustrate a categorical variable, Feature 4 was chosen. It has six levels and even though there are categorical variables with less levels it is a good choice for a deep analysis. Figure 2 shows the bar chart and the bar chart by target for this feature.

In graph (a) of Figure 2 it can be seen that the level G2 is the most common group of this covariate with a really large difference when compared to the other's levels, although the remaining variables seem to be at similar values, besides G6 that has almost no observations. This indicates that most of the clients are inserted on the level G2. To investigate the effect of these levels on the target, the graph (b) of Figure 2 show them in a descendant ranking system, where the level that has the most retention appears first following after, the levels with smaller retention. Considering that, even with the differences shown in the previous graph, the retention does not have large differences between levels. To be more exact, the retention in level G2, the highest, is $82\%$ while the level G6 is nearly $70\%$.

Given that we predicting proportions, those small differences may be precise to obtain more accurate results. This will be checked in later stages of this report at the modelling stage chapter.




(a) Bar Chart for Feature 4




(b) Retention Rate by Feature 4

FIGURE 2: Categorical Graphs for Feature 4.

To illustrate the analysis of a numerical variable, we chose Feature 11 whose behaviour is more interesting to investigate given the nature of the variable being the number of renewals. The first graph for this covariate is an histogram.

Figure 3 shows a histogram where most of the observations are included in the intervals $[0; 5[$, combining with the decreasing shape, it indicates that this variable tends to not have very large observations. The rates of retention for each class were obtained separately and indicates that the rates tend to be higher for when the number of renewals are greater than $6$. Taking a look for the higher values, it seems to be well adjusted not having extreme values, but this will be checked in the next graph.

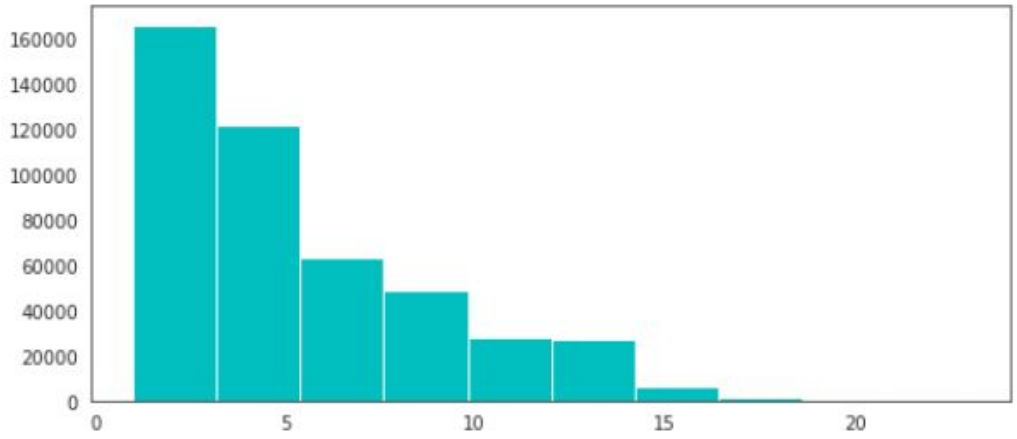



FIGURE 3: Histogram for Feature 11.

For finalising this section, we have the last figure shown below.

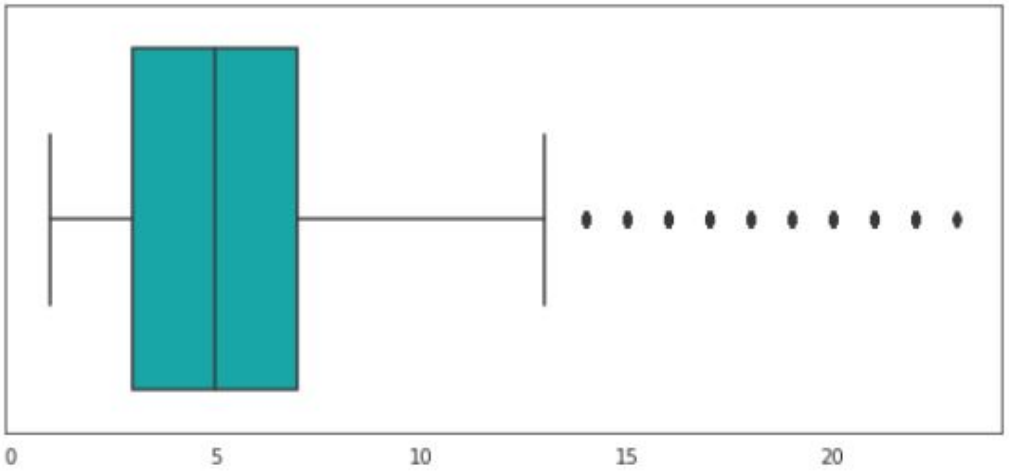



FIGURE 4: Box-Plot for Feature 11.

The box-plot in Figure 4 shows that the $Median$ of this covariate is $5$ and $3^{rd}Quantile$ is $7$ which indicates that $75\%$ of the sample distribution is smaller than 7, suggesting, once again, that this variable does not tend to have higher values. For the potential outliers that are shown, given the nature of the variable they cannot be considered as outliers. Those values give us important information of how clients are behaving when comes to retaining for one more year.

## 3.2 Correlation & Association Analysis

Managing the correlation between the variables is important to prevent bias or over-fitting in the modelling stages, therefore one of the crucial parts of the exploratory data analysis is to calculate the correlations among the variables and for an easier understanding of the results.

### 3.2.1 Person's Correlation

In terms of continuous variables, the most famous method which is applied to this analysis was the Person's Correlation, that is a measure of linear correlation between two variables $x$ and $y$, denoted as $r_{(x,y)}$. The correlation coefficient formula is defined in (3.1).

$$r_{(x,y)} = \frac{s_{x,y}}{s_x s_y} \tag{3.1}$$

Where $s_{x,y}$ stands for the covariance between variables $x$ and $y$ while $s_x$ and $s_y$ for the standard deviation of $x$ and $y$, respectively.

### 3.2.2 Cramér's V

To explore possible correlation between categorical variables, we use was the Cramér's V coefficient, $\phi_c$, [Anderson et al. (2007)] that works as measure of association between two categorical features.

Let's consider two categorical variables, $X$ and $Y$ which have the following subsets $\{x_1, x_2, \cdots, x_I\}$ and $\{y_1, y_2, \cdots, y_J\}$, respectively. For each variable there are n observations. Taking into account every possible combination of values $(x_i, y_j)$, let $n_{ij}$ denote the number of times that combination is shown in the sample. Defining the row and columns totals, we obtain the succeeding:

$$n_{i\bullet} = \sum_{j=1}^{J} n_{ij} \;\; and \;\; n_{\bullet j} = \sum_{i=1}^{I} n_{ij} \tag{3.2}$$

Given that, the Chi-Square Statistic is defined as,

$$\chi^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{(\frac{n_{ij} - n_{i\bullet}}{n})^2}{\frac{n_{i\bullet} n_{\bullet j}}{n}} \tag{3.3}$$

The Cramér's V statistic, denoted as $V$ or $\phi_c$, is a simple transformation of the Chi-Squared statistic mentioned in equation (3.3).

$$V = \sqrt{\frac{\chi^2/n}{min(I,J)-1}} \tag{3.4}$$

The result of equation (3.4) should be a numeric value that varies between $0$ and $1$, where $0$ represents no association between the two categorical variables and $1$ represents a perfect association between those variables.

### 3.2.3 Results

Figure 4 presents the Person's correlation coefficients for the quantitative variables as an heat graph. It gives different tones for each value of correlation in order to spot easily the most significant coefficients.


FIGURE 5: Correlation Plot for all Continuous Variables.

It can be seen in Figure 5 that Feature 7 and Feature 15 are highly correlated, which means that one of those two variables should be removed in the modelling stages. As explained before, highly correlated covariates might bring over-fitting to our models, consequently the decision was to remove Feature 15 since it was found in further analysis to be the less important feature to predict the retention.

The following table I shows the estimates of Cramér's V coefficients between all categorical variables, as it shows the variables are not associated in any level that should be excluded, therefore it was important to maintain those for further analysis in the project.

| Features | Feature 1 | Feature 4 | Feature 5 | Feature 6 | Feature 10 | Feature 19 |
|---|---|---|---|---|---|---|
| Feature 1 | 1 | 0.0897 | 0.049 | 0.0443 | 0.041 | 0.362 |
| Feature 4 | 0.0897 | 1 | 0.1513 | 0.0412 | 0.1204 | 0.0528 |
| Feature 5 | 0.049 | 0.1513 | 1 | 0.0497 | 0.1078 | 0.0193 |
| Feature 6 | 0.0443 | 0.0412 | 0.0497 | 1 | 0.191 | 0.0459 |
| Feature 10 | 0.041 | 0.1204 | 0.1078 | 0.191 | 1 | 0.0387 |
| Feature 19 | 0.362 | 0.0528 | 0.0193 | 0.0459 | 0.0387 | 1 |

TABLE I: Cramér's V Coefficients Matrix for the Categorical Variables.

## 3.3 Feature Engineering

One of the biggest problems when modelling is dealing with missing values on the given data set. The reasons for the existence of the missing values could vary and it is crucial to obtain the most valuable information possible contained in the covariates so we can achieve the best results in the modelling stages. The information obtained for the missing values in this data set were mainly originated by non-information in the policies. To solve this, our approach was to replace the missing values for the continuous variables by the median of the variable, in order to not be influenced by the extreme values that such variables can contain. For the categorical variables our approach was to add a new category for the missing's named *"Missing"*.

To model machine learning algorithms there is a need to pay special attention to the categorical variables. Given that the usual dummy variable method does not give the best results, it is important to search for more complex methods, for that matter, the approach chosen was the target label encoding. The target encoding [Micci-Barreca (2001)] is a numeration of categorical variables via target. In this method, the categorical variable is replaced with just one new numerical variable, where we replace each level of the categorical variable with its corresponding probability of the target. Instead of having a covariate with levels such as Y1, Y2, Y3 and Y4, we replace those values with probability associated with the corresponding occurrence probability of the target within each level. As we can see in the following table II as an example.

| Feature 1 | Target : Renew | New Feature 1 |
|---|---|---|
| Y1 | 1 | 0.6 |
| Y1 | 1 | 0.6 |
| Y3 | 1 | 0.75 |
| Y2 | 0 | 0.5 |
| Y4 | 1 | 0.75 |
| Y1 | 0 | 0.6 |
| Y2 | 1 | 0.5 |
| Y1 | 0 | 0.6 |
| Y4 | 1 | 0.75 |
| Y4 | 0 | 0.75 |
| Y1 | 1 | 0.6 |

TABLE II: Example Table of the Target Encoding Application.

The main drawbacks of this method is its dependency to the distribution of the target and in the perspective of deployment, would be updating the historical information. Once this is settled, the data is ready for the modelling stage.

## 3.4 Training & Testing

The major problems of modelling are dealing with over-fitting and the possibility that the model memorises the behaviour of the data. Such can happen when there is a full usage of our whole data set. This may lead to bad performances in unseen data. For that reason there is an urgent need to follow the approach of the train-test split methods before creating the models. The training and testing for insurance models will ensure that models are behaving correctly and will measure the adaptation of the model on unknown data.
One of the main reasons to do this project was with the intention to explore different techniques, two of them are presented here and tested. The first methodology was presented by the company and worked by selecting the first nine months, from January to September, for the train set and the remain three months, from October to December, for test set, shown in Figure 6.

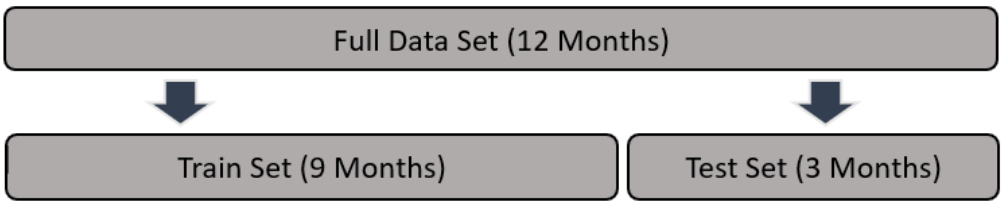



FIGURE 6: Partition Demonstration for the First Methodology.

This approach has an easy implementation and would capture all the information of those train months. Although it may be overlooking the seasonality that some of the months in the different set may contain (see Figure 1). For that reason, a new approach was implemented where a percentage of the data was selected in each month for the training data set and the remaining percentage was for the testing data.

The partition chosen for this split was $80\%$ to the training data set and $20\%$ to the testing data set. This can be seen in the next Figure 7.

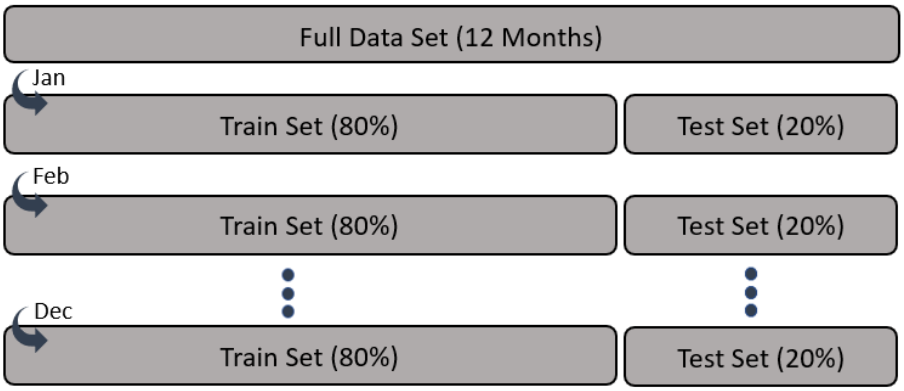



FIGURE 7: Partition Demonstration for the Second Methodology.

# Chapter 4

# 4 Modelling

Since the main objective is to compare different frameworks for modelling the retention for clients in the company's portfolio, this section explains the theory behind of them.
As mentioned in previous chapters, the models considered were Logistic Regression, Extreme Gradient Boost (xGBoost) and Light Gradient Boosting Machine (Light GBM).

## 4.1 Logistic Regression

The logistic regression is a common model for modelling these type of problems in insurance Guillen et al. (2003) and it will be used as a benchmark account for further comparisons with the Machine Learning models.
The Logistic Regression [Murphy et al. (2000)] is a Generalised Linear Model [Nelder & Wedderburn (1972)], this means it has the following form:

$$Y = h(\boldsymbol{X}\boldsymbol{\beta}) + \epsilon \tag{4.1}$$

where $h$ is a monotonic function and $Y$ is distributed to a selected exponential family of distributions. $\boldsymbol{X}$ is called the design matrix and it holds the information of the covariates.

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

where $x_{ij}$ denotes the $j^{th}$ feature and $i^{th}$ observation in the data set, $p$ is the number of features and $n$ the total number of observations in the data set.

Let $\eta$ represent the linear predictor that is given by:

$$\eta = \boldsymbol{X\beta} = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} \tag{4.2}$$

where $\{\beta_0, \beta_1, \cdots, \beta_p\}$ are the regression coefficients and the regression coefficient vector is $\beta = \begin{bmatrix}\beta_0 & \beta_1 & \cdots & \beta_p\end{bmatrix}^T$. The vector $x_i$ is defined as $\boldsymbol{x_i} = \begin{bmatrix}1 & x_{i1} & \cdots & x_{ip}\end{bmatrix}^T$. The link function is set as $g(\mu) = \eta$ where $\mu$ represents transformation of the linear predictor that is given by $h(\boldsymbol{X\beta})$.
Given the nature of our problem, $Y$ has a binary response that has the following output:

$$Y = \begin{cases} 0, & \text{if the Policyholder cancels the Policy} \\ 1, & \text{if the Policyholder renews the Policy} \end{cases}$$

where $Y \sim Bernoulli(\mu)$.

The goal of the model is to predict the occurrence of retention which can be settled as a conditional probability of a renewal $P(Y = 1|X) = \mu$.
To set a Logistic Regression the chosen link function is the **logit link function**, so we obtain the equation (4.3) as a function of $g(\mu)$.

$$g(\boldsymbol{\mu}) = \ln\big(\frac{\boldsymbol{\mu}}{1-\boldsymbol{\mu}}\big) \tag{4.3}$$

Consequently our regression model can be defined as:

$$\ln\big(\frac{\boldsymbol{\mu}}{1-\boldsymbol{\mu}}\big) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} = \boldsymbol{\beta^T x_i} \tag{4.4}$$

Through general computation, the regression model can be obtained in 4.5 and is shown bellow,

$$\boldsymbol{\mu(x, \beta)} = \frac{e^{\boldsymbol{\beta^T x}}}{1 + e^{\boldsymbol{\beta^T x}}} \tag{4.5}$$

This results on a odds of occurrence output, this conducts to a natural interpretation of the coefficients description on the effects of our predictor. If $\beta_j$ is positive then there's a increasing effect on the predictor for $x$ increases, if $\beta_j$ is negative the there's a decrease effect on the predictor in case if $x$ increases.

## 4.2 Extreme Gradient Boosting Algorithm

The first model to be used is the Extreme Gradient Boosting [Chen & Guestrin (2016)]. The nature of this algorithm is predictive modelling that mainly focus on predicting Y, our response variable, by using the predictors, also known as features, X. To build the model we need the response variable, $Y$ and a set of covariates $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_m\}$ as well as a set of data given by $\mathcal{D} = \{(Y_1, \mathcal{X}_1), (Y_2, \mathcal{X}_2), \cdots, (Y_n, \mathcal{X}_m)\}$ where $n$ is the total number of observations in the data set and the sample is assumed to be independent and identically distributed.
The prediction model is produced by a form of ensemble of weak predictors like decision trees. From this, a tree assembles a model with the given equation (4.6) that assumes $K$ additive functions to predict the output.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{4.6}$$

where $\mathcal{F} = \{f(x) = w_q(x)\}(q : R^m \to T, w \in R^T)$ is the space of regression trees, $q$ represents the structure of each tree that maps an example to the respective leaf index, $T$ is the number of leaves in the tree. Each $f_k$ represents an independent tree structure $q$ and leaf weights $w$. Each of this regression trees records a continuous score of each leaf, denoted as $w_i$. In order to obtain optimal prediction, we need to minimise the *regularised objective function* shown bellow:

$$\mathcal{L}(\phi) = \sum_i l\big(\hat{y}_i, y_i\big) + \sum_k \Omega(f_k) \tag{4.7}$$
$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

$l$ is a loss function which penalises the the difference between the prediction $\hat{y}_i$ and the target $y_i$. The second term $\Omega$ penalises the complexity of the model. This additional regularisation makes sure to smooth the final learnt weights to avoid over-fitting.

Following once again the [Chen & Guestrin (2016)], we can summarise the implementation of the Extreme Gradient Boosting which is based on the following two algorithms and enumerates over all the possible splits on all the features. Algorithm 1 in Figure 8 represents the exact greedy algorithm, it enumerates all the possible splits for continuous features.

**Input**: $I$, instance set of current node
**Input**: $d$, feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
**for** $k = 1$ ***to*** $m$ **do**
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  **for** $j$ *in sorted(I, by* $\mathbf{x}_{jk}$*)* **do**
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  **end**
**end**
**Output**: Split with max score

FIGURE 8: Exact Greedy Algorithm for Split Finding for Extreme Gradient Boosting. Source:[Chen & Guestrin (2016)]

Algorithm 2 in Figure 9 represents the Approximate Algorithm, it proposes candidate splitting points according to percentiles of feature distribution.

**for** $k = 1$ ***to*** $m$ **do**
  Propose $S_k = \{s_{k1}, s_{k2}, \cdots s_{kl}\}$ by percentiles on feature $k$.
  Proposal can be done per tree (global), or per split(local).
**end**
**for** $k = 1$ ***to*** $m$ **do**
  $G_{kv} \leftarrow= \sum_{j \in \{j | s_{k,v} \geq \mathbf{x}_{jk} > s_{k,v-1}\}} g_j$
  $H_{kv} \leftarrow= \sum_{j \in \{j | s_{k,v} \geq \mathbf{x}_{jk} > s_{k,v-1}\}} h_j$
**end**
Follow same step as in previous section to find max score only among proposed splits.

FIGURE 9: Approximate Algorithm for Split Finding for Extreme Gradient Boosting. Source:[Chen & Guestrin (2016)]

The algorithm, in Figure 9, maps the continuous features into buckets split by these candidate points and aggregates the statistics to find the best solution among proposals based on the aggregated statistics. Once found the best optimal tree, the output of this algorithm contains the AUC (Area Under the Curve) which will be explained in the next sections.

## 4.3 Light Gradient Boost Algorithm

As a good contender for the Extreme Gradient Boosting, we have the Light Gradient Boosting [Ke et al. (2017)] which follows the same process as the Extreme Gradient Boosting Algorithm as a decision tree-based algorithm. However the Light Gradient Boosting elaborates more precise algorithms to deal with continuous and categorical variables, which are the Gradient-based One-Side Sampling(GOSS) algorithm and Histogram Algorithm.

The GOSS is a predictive modelling algorithm to predict our response variable, Y, by using a data set given by n independent and identically distributed sample, $\{x_1, x_2, \cdots, x_n\}$, where each $x_i$ is a vector of dimension s in space of $\mathcal{X}^s$. For each interaction of gradient boosting, the negative gradients of the loss function with respect to the output of the model are denoted as $\{g_1, g_2, \cdots, g_n\}$.

The ranking method consists in ranking the train instances according to the absolute values of their gradients in descending order and keeping the top $a \times 100\%$ instances with the larger gradients and get an instance subset $A$. The remaining set $A^c$,$(1-a) \times 100\%$ instances, with smaller gradients, are randomly sampled with a subset $B$ with size $b \times |A^c|$. Ultimately, the instances are split according to the estimated variance gain $\widetilde{V}_j(d)$ over the subset $A \cup B$, i.e.,

$$\widetilde{V}_j(d) = \frac{1}{n}\Big(\frac{\big(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\big)^2}{n_l^j(d)} + \frac{\big(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\big)^2}{n_r^j(d)}\Big) \quad (4.8)$$

where $A_l = \{x_i \in A : x_{ij} \leq d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$, $B_l = \{x_i \in B : x_{ij} \leq d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$, and the coefficient $\frac{1-a}{b}$ is used to normalise the sum of the gradients over $B$ back to the size of $A^c$.

To execute this Gradient-based One-Side Sampling Algorithm, the implementation is shown in the following figure.

**Input**: $I$: training data, $d$: iterations
**Input**: $a$: sampling ratio of large gradient data
**Input**: $b$: sampling ratio of small gradient data
**Input**: $loss$: loss function, $L$: weak learner
models $\leftarrow$ { }, fact $\leftarrow \frac{1-a}{b}$
topN $\leftarrow$ a $\times$ len($I$) , randN $\leftarrow$ b $\times$ len($I$)
**for** $i = 1$ ***to*** $d$ **do**
    preds $\leftarrow$ models.predict($I$)
    g $\leftarrow$ $loss$($I$, preds), w $\leftarrow$ {1,1,...}
    sorted $\leftarrow$ GetSortedIndices(abs(g))
    topSet $\leftarrow$ sorted[1:topN]
    randSet $\leftarrow$ RandomPick(sorted[topN:len(I)], randN)
    usedSet $\leftarrow$ topSet + randSet
    w[randSet] $\times =$ fact $\triangleright$ Assign weight $fact$ to the small gradient data.
    newModel $\leftarrow$ L($I$[usedSet], $-$ g[usedSet], w[usedSet])
    models.append(newModel)

FIGURE 10: Gradient-based One-Side Sampling for Light Gradient Boost. Source: [Ke et al. (2017)]

Overall, the algorithm uses the gradient of each data instance as a proxy for sample weight and then drops the samples with low gradient, holding the subset $A$ with the highest gradients, with the purpose of saving computing time during the search for the best split. Algorithm 2 in Figure 10 represents the Histogram-based Algorithm, as the name indicates it produces an histogram where the bins record the feature values.

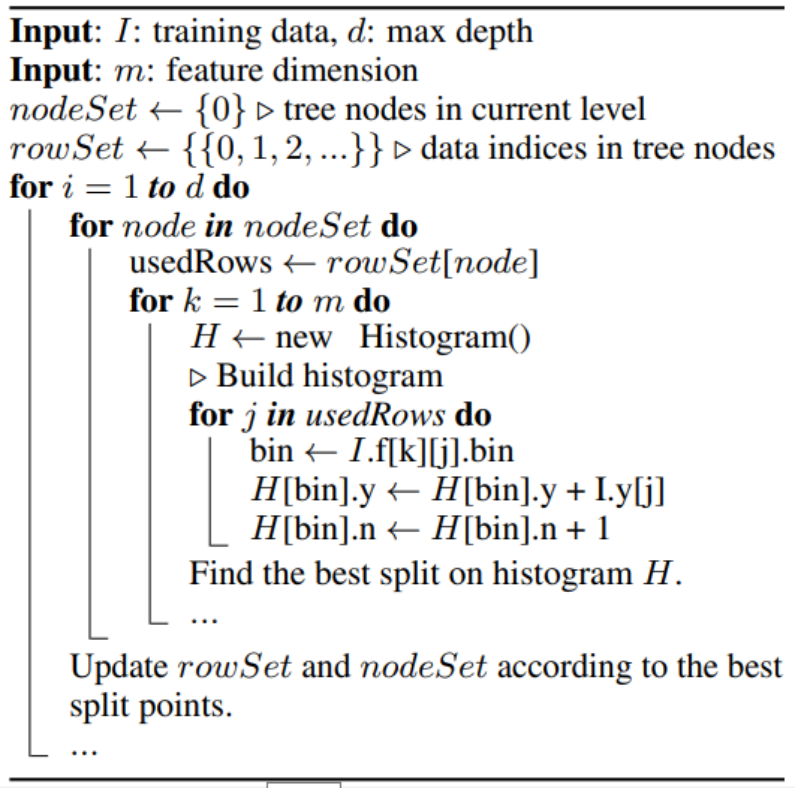



FIGURE 11: Histogram-based Algorithm for Light Gradient Boost.
Source: [Ke et al. (2017)]

The second Algorithm in Figure 11 allocates every feature value in the data set into a set of bins. Once those are created, for the continuous variables, the algorithm goes through all sample's accumulated statistics and finds the best split point giving the maximum gain when separate $k$ bins into two parts. For categorical variables, there's two methods, the first method explores possible splits for the levels of the categorical variable and tries to find out which split gains maximum output. The second method will accumulate statistics for each categories, subsequently it will sort these categories based on the value of the feature. Afterwards, from large to small and vice versa, it will be implemented a maximum bins that will be searching possible split points to obtain the one who gives the maximum gain.

This algorithm is usually designed as the simpler algorithm because it only explores the leaves that are significant, as a leaf-wise tree growing strategy, which makes the time process be faster. While the Extreme Gradient Boosting uses the scheme of a level-wise tree growing strategy.

## 4.4 Model Selection

Concerning our problem, it is important to explore methods of comparison between the models, so we can obtain valuable information of which model performs the best. For this reason four different methods were chosen:
The Area Under the Curve (AUC), Receiver Operating Characteristic Curve (ROC Curve) and Accuracy, known to be the simplest case and easier interpretation, and the McNemar's Test, a more robust and sophisticated test.

### 4.4.1 Area Under the Curve and Receiver Operating Characteristic curve

Receiver Operating Characteristic Curve, also known as ROC Curve [Fawcett (2006)], is a metric for binary classification problems that represents the probability curve that plots the True Positive Rates against the False Positive Rate at several threshold values, this means it analyses the cost against the benefits.
The Area Under the Curve (AUC) [Fawcett (2006)] measures the capacity of the classifier to distinguish between classes.
To obtain those metrics it is important to define the following. Let the following table III be defined as confusion matrix.

| | Actual Value Positive | Actual Value Negative |
|---|---|---|
| Predicted Value Positive | True Positive (TP) | False Positive (FP) |
| Predicted Value Negative | False Negative (FN) | True Negative (TN) |

TABLE III: Confusion Matrix.

Once the confusion matrix is found, it is important to obtain the following rates to get the ROC Curve and AUC.
The True Positive Rate, or $Sensitivity$, measures what proportion of positives that got correctly classified, and can be calculated by the following formula

$$TPR = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{4.9}$$

The False Negative Rate is the proportion of positives that got incorrectly classified is given by:

$$FNR = \frac{False\ Negative}{True\ Positive + False\ Negative} \tag{4.10}$$

The True Negative Rate, also known as $Specificity$, measures the proportion of negatives that got correctly identified by the classifier and the formula is shown below.

$$TNR = \frac{True\ Negative}{True\ Negative + False\ Positive} \tag{4.11}$$

The last measure is the False Positive Rate that tell us the proportion of negatives that got incorrectly classified, is calculated in the next equation:

$$FPR = \frac{False\ Positive}{True\ Negative + False\ Positive} = 1 - Specificity \tag{4.12}$$

Once these rates are found, both measures, ROC Curve and AUC can be calculated. The ROC Curve will calculate for each threshold the respective values of the True Positive Rates and the False Positive Rate so it can be plotted in a given chart.
The AUC can vary between $0$ and $1$, the higher the AUC gets to $1$ to better is our model predicting correctly. An AUC of $0.5$ is usually an indication of no skill in the model or randomness.
Although $Accuracy$ [Sokolova et al. (2006)] is not used to obtain the ROC or even the AUC, it is an important measure which holds the statistics of the corrected predictions in the total sample as a percentage, at threshold of $0.5$. Can be obtained using the values on the confusion matrix, and is as it follows:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \tag{4.13}$$

### 4.4.2 McNemar's Test

Firstly, we will start with a brief explanation of this test. The goal of the McNemar's test [Dietterich (1998)] is to analyse the statistical significance of differences in classifier performances.
The hypothesis test for this is given by:

$$H_0 : None\ of\ the\ two\ models\ performs\ better\ than\ the\ other.$$

$$vs.$$

$$H_1 : The\ performances\ of\ the\ two\ models\ are\ not\ equal.$$

This is based on a contingency table IV by $\mathbf{2} \times \mathbf{2}$. The cells includes the number of samples of the number of observations that are correctly and incorrectly identified by each model.

| | Model 2 Correct | Model 2 Incorrect |
|---|---|---|
| Model 1 Correct | Yes/Yes (a) | Yes/No (b) |
| Model 1 Incorrect | No/Yes (c) | No/No (d) |

TABLE IV: Contingency Matrix.

Once these values are obtained, we can calculate the statistical test which are stated subsequently:

$$\mathcal{X}_{stat} = \frac{(Yes/No - No/Yes)^2}{(Yes/No + No/Yes)} = \frac{(b-c)^2}{(b+c)} \sim \boldsymbol{\chi^2_{(1)}} \tag{4.14}$$

In order infer about the test we must obtain the exact p-value shown in (4.15).

$$p_{value} = 2\sum_{i=b}^{n} \binom{n}{i} 0.5^i (1-0.5)^{n-i} \tag{4.15}$$

where $n$ is the dimension of the sample.

If the $p - value$ is larger than a threshold $\alpha = 0.05$ for instance, then we fail to reject $H_0$, so there's no significant difference between the two classifiers in the test set. If the $p-value$ is smaller than the threshold, then we reject $H_0$, so there's significant differences between the two classifiers in the test set.
The drawback of this approach is that it does not give the best model but just confirms the clear differences, to conclude the finding of the best performance it should be combined with the AUC & ROC methods.

## 4.5 Results

Our first step is to analyse the output from our three models segmented by the two types of training and testing spilt methodologies. Figure 12 is related to the Logistic Regression and the applied training and testing methods.

| | coef | std err | z | P>\|z\| |
|---|---|---|---|---|
| feature9 | 0.9850 | 0.066 | 14.847 | 0.000 |
| feature13 | 1.6165 | 0.049 | 33.309 | 0.000 |
| feature2 | 5.416e-08 | 4.13e-09 | 13.108 | 0.000 |
| feature5 | 1.7877 | 0.168 | 10.623 | 0.000 |
| feature20 | 0.1218 | 0.093 | 1.302 | 0.193 |
| feature7 | 2.961e-07 | 1.23e-06 | 0.242 | 0.809 |
| feature12 | -4.975e-06 | 2.4e-06 | -2.077 | 0.038 |
| feature1 | 3.7315 | 0.192 | 19.398 | 0.000 |
| feature4 | 4.0714 | 0.167 | 24.310 | 0.000 |
| feature16 | -0.0067 | 0.001 | -5.260 | 0.000 |
| feature10 | -16.6613 | 0.468 | -35.633 | 0.000 |
| feature11 | 0.0278 | 0.002 | 17.343 | 0.000 |
| feature19 | 7.7973 | 0.122 | 63.890 | 0.000 |
| feature8 | -0.0041 | 0.000 | -21.784 | 0.000 |
| feature17 | -0.1415 | 0.034 | -4.166 | 0.000 |
| feature18 | -0.0006 | 3.13e-05 | -17.575 | 0.000 |
| feature3 | -0.0063 | 0.000 | -19.142 | 0.000 |
| feature6 | -0.0675 | 0.380 | -0.178 | 0.859 |
| feature14 | -0.0140 | 0.001 | -21.262 | 0.000 |

(a) Logistic Regression Output for the First Method Split

| | coef | std err | z | P>\|z\| |
|---|---|---|---|---|
| feature9 | 0.9343 | 0.063 | 14.860 | 0.000 |
| feature13 | 1.6936 | 0.047 | 36.142 | 0.000 |
| feature2 | 5.311e-08 | 3.78e-09 | 14.047 | 0.000 |
| feature5 | 1.4543 | 0.162 | 8.956 | 0.000 |
| feature20 | 0.1095 | 0.091 | 1.206 | 0.228 |
| feature7 | 4.663e-06 | 1.97e-06 | 2.362 | 0.018 |
| feature12 | -1.077e-05 | 2.27e-06 | -4.741 | 0.000 |
| feature1 | 4.2321 | 0.189 | 22.408 | 0.000 |
| feature4 | 4.1565 | 0.163 | 25.450 | 0.000 |
| feature16 | -0.0042 | 0.001 | -3.982 | 0.000 |
| feature10 | -17.5606 | 0.446 | -39.416 | 0.000 |
| feature11 | 0.0272 | 0.001 | 18.229 | 0.000 |
| feature19 | 7.7747 | 0.118 | 66.043 | 0.000 |
| feature8 | -0.0040 | 0.000 | -21.833 | 0.000 |
| feature17 | -0.1409 | 0.034 | -4.155 | 0.000 |
| feature18 | -0.0006 | 2.9e-05 | -20.910 | 0.000 |
| feature3 | -0.0054 | 0.000 | -17.441 | 0.000 |
| feature6 | 0.4698 | 0.360 | 1.306 | 0.192 |
| feature14 | -0.0130 | 0.001 | -20.890 | 0.000 |

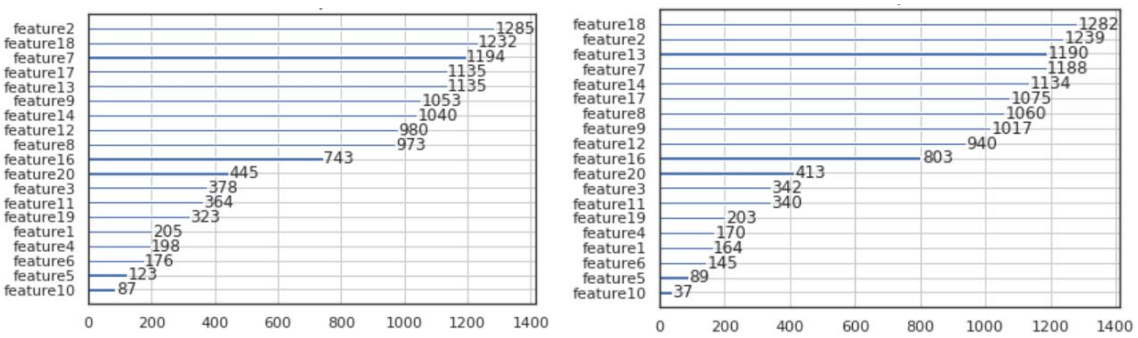
(b) Logistic Regression Output for the Second Method Split

FIGURE 12: Logistic Regression Python Output.

From the output (a) in Figure 12, we can see that most of the features are significant to explain our model, at a significance level of $5\%$. Only $Feature\ 6$, $Feature\ 7$ and $Feature\ 20$ showed no significant importance. From the coefficients perception on whether which of the variables have more influence, we can tell $Feature\ 10$ ($\beta_{Feature\ 10} = -16.6613$), $Feature\ 19$ ($\beta_{Feature\ 19} = 7.7973$) and $Feature\ 4$ ($\beta_{Feature\ 4} = 4.0714$) seemed to be affecting our response variable at higher rates. In the output (b) in Figure 12 there was a slightly improvement of the significance of the variables, $Feature\ 7$ is now an important variable meanwhile the remaining continued being not significant, this can indicate that the first split may lost information when using the partition of the first $9$ months against the remaining $3$ months. Looking once again to the most important variables they seem not to change from the previous model, although its now included the $Feature\ 1$ ($\beta_{Feature\ 1} = 4.2321$).

Considering the $Accuracy$ of each model, the first Logistic Regression scored $78.92\%$ while the second Logistic Regression scored $80.48\%$, this shows that the second split method on the Logistic Regression performed marginally better when predicting the right results.

The next figure represents the output from the Extreme Gradient Boost.




(a) xGBoost Feature Importance Graph for the First Split

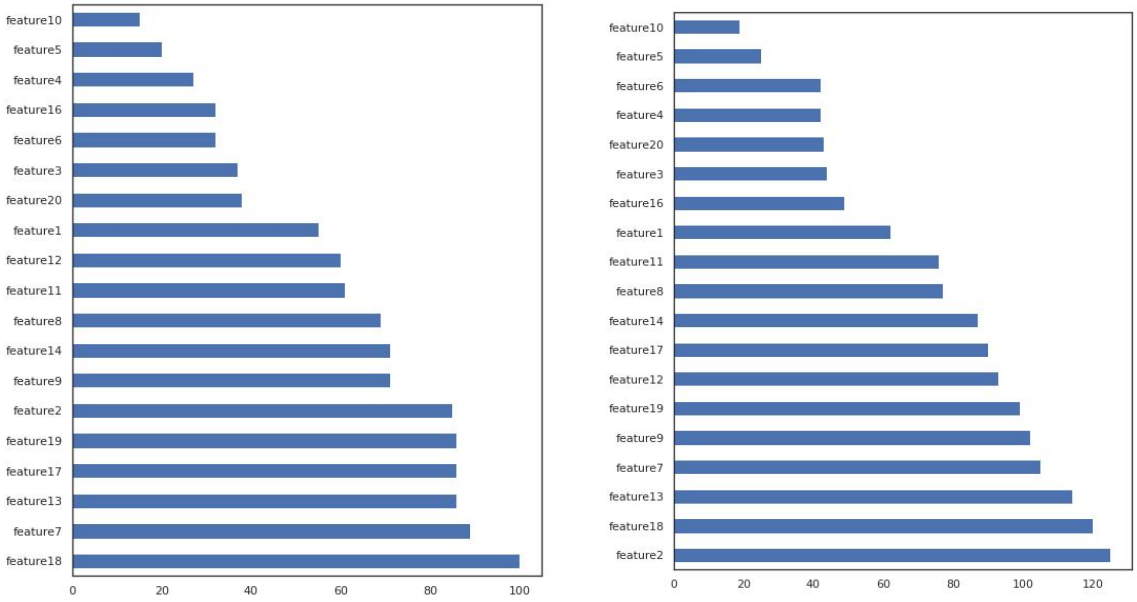
(b) xGBoost Feature Importance Graph for the Second Split

FIGURE 13: Extreme Gradient Boost Feature Importance Graphs.

Both outputs are called feature importance [Hastie et al. (2008)] plots, which represents the technique of scoring the features of given predictive model and indicates the importance of each variable in the classifier. This is usually calculated within the chosen algorithm. For the graph (a) in Figure 13 it shows that the most important variables are $Feature\ 2$, $Feature\ 18$ and $Feature\ 7$. In the graph (b) in Figure 13 the main variables are $Feature\ 18$, $Feature\ 2$ and $Feature\ 13$. When comparing those two models it seems not to have such big difference between the main variables. This may indicate that the algorithm is retaining effectively the same information in the different types of methodology split.

In terms of the $Accuracy$ applied to the test set of each classifier from the Extreme Gradient Boosting, the first method scored $79.07\%$, and the second method scored $80.68\%$ which leads to choosing the second model of the Extreme Gradient Boosting because it performed better in terms of accuracy.

To finalise, the following graphs are related to the Light Gradient Boost output, it is the same format as the previous figure.




(a) LightGBM Feature Importance Graph for the First Split

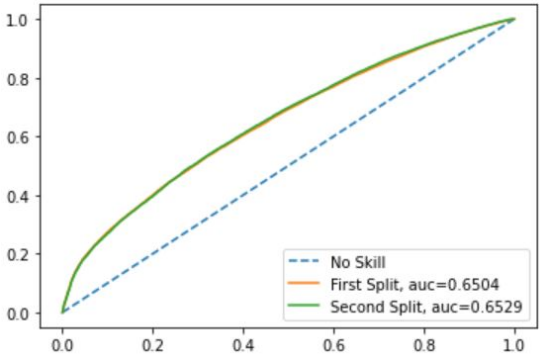
(b) LightGBM Feature Importance Graph for the Second Split

FIGURE 14: Light Gradient Boost Feature Importance Graphs.

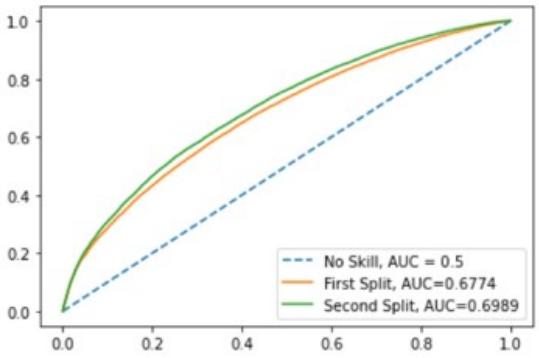
In these graphs it shows the important variables from the bottom to the top, instead of the top to the bottom as showed in the previous graphs, so for the plot in (a) of Figure 14 the most important variables are $Feature\ 18$, $Feature\ 7$ and $Feature\ 13$. The plot (b) in Figure 14, we have the following as major variables the $Feature\ 2$, $Feature\ 18$ and $Feature\ 13$. Concerning the Accuracy for the Light Gradient Boost models, the first model scored $79.07\%$, and the second model scored $80.68\%$, that indicates that once again the second model performed better than the first model according to predicting the equal results of the data set.

Given all this, we see that for the majority of models they have mostly the same important variables only showing the bigger differences in the logistic regression which is probably resulting by the difference in methodologies. Overall, in terms of the $Accuracy$, the second training and testing split method of the machine learning algorithms had the better performance, even that it was not large difference compared to the peers.

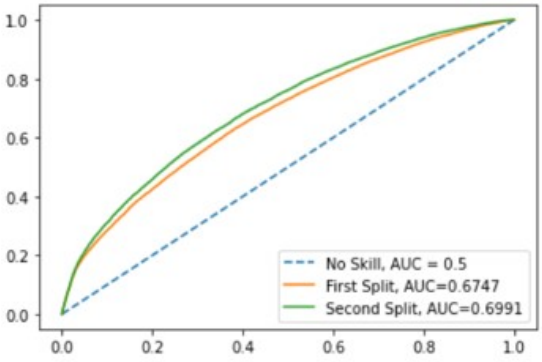
The next topic we are going to discuss is the ROC curves & AUC results for each of the models approaches. In Figure 15 we have three graphs related to the measures described above.




(a) Logistic Regression ROC Curves and AUC Scores.




(b) Extreme Gradient Boost ROC Curves and AUC Scores.




(c) Light Gradient Boost ROC Curves and AUC Scores.

FIGURE 15: ROC Curves and AUC Scores for each model.

As we can see in all graphs, they all behave similarly in terms of the ROC Curves, and they all show better curves for the second method of splitting when compared to the first one. Once we look to the AUC scores, we confirm what has been seen in the ROC curves: All AUC scores are greater in the second training and testing splitting methodology. This may confirm the hypothesis that the first methodology of splitting was restraining important information for predicting the proportion of retention. Comparing the results of each AUC individually we can see that the highest AUC score belongs to the Light Extreme Gradient Boosting with the second split method ($AUC_{Light\ GBM:2^{nd}\ Split} = 69.91\%$).
To finalise this section, the succeeding approach is the McNemar's Test for all the classifiers in order to obtain statistical inference about the real differences between them. The next set of tests refer to the data set where the **first method of split** was applied, given the complexity of this approach, we can only compare results were the same method of split were applied. It is not possible to perform the McNemar's test for models were in one it is applied the first method of training and testing and in the other the second method of training and testing.

Firstly, it is important to formulate the hypothesis tests for each statistical test applied, which are:

$H_0$ : Neither the two models, Logistic Regression and Extreme Gradient Boosting, perform better than each other. $vs.$

$H_1$ : The performances of the two models are not equal.

The obtained $p_{value}$ for this test is approximately $0$, we reject $H_0$ so there's enough evidence for the differences between the models.
The second test has the following hypothesis,

$H_0$ : Neither the two models, Logistic Regression and Light Gradient Boost, perform better than each other. $vs.$

$H_1$ : The performances of the two models are not equal.

Once again, the obtained $p_{value}$ for this test is approximately $0$, we reject $H_0$ therefore there's enough evidence for the differences between the models.
The final test and the more informative test for the purpose of this project, has the following hypothesis,

$H_0$ : Neither the two models, Extreme Gradient Boosting and Light Gradient Boost, perform better than each other. $vs.$

$H_1$ : The performances of the two models are not equal.

The obtained $p_{value}$ for this test is approximately $0.869$, we do not reject $H_0$ therefore there's not enough evidence for the differences between the models.
The succeeding set corresponds to the **second method of split** which, once more, was applied the McNemar's test.
The first test we have the following hypothesis,

$H_0$ : Neither the two models, Logistic Regression and Extreme Gradient Boosting, perform better than each other. $vs.$

$H_1$ : The performances of the two models are not equal.

The $p_{value}$ is approximately $0$, we reject $H_0$ so there's enough evidence for the differences between the models.

The second test can be denoted by the following hypothesis,

$H_0$ : Neither the two models, Logistic Regression and Light Gradient Boost, perform better than each other. $vs.$

$H_1$ : The performances of the two models are not equal.

Once again, the obtained $p_{value}$ for this test is approximately $0$, we reject $H_0$ therefore there's enough evidence for the differences between the models.
The final test has the following hypothesis,

$H_0$ : Neither the two models, Extreme Gradient Boosting and Light Gradient Boost, perform better than each other. $vs.$

$H_1$ : The performances of the two models are not equal.

The obtained $p_{value}$ for this test is approximately $0.637$, we do not reject $H_0$ therefore there's not enough evidence for the differences between the models.
Given all this information, we can tell that the Machine Learning classifiers had statistically significant differences when compared to the Logistic Regression although when comparing the Extreme Gradient Boosting and Light Gradient Boost there's no statistically significant differences given their similarities as gradient boosting machines. Despite the fact that McNemar's showed no differences, the AUC scores displayed slightly differences, having the Light Gradient Boost has better performance score.

# Chapter 5

# 5 Conclusions

The analysis that took place in this report were quite successful in answering the main problems raised in this project, those include in effectively supervise the data set in order to prevent bias and over-fitting as well as managing the data set where it was encountered a collection of data omissions. It was also successful in finding the most important features to model Liberty's client retention.
In terms of the training and testing, it was possible to determinate that a more structured split for the data set, in our case the second method, results in a better accuracy and AUC scores of every model produced.
Taking in account on whether the Machine Learning Algorithms (MLA) performed better than the Logistic Regression, it was proven that Machine Learning Algorithms statistically had better performance results. It was possible to even determinate that the Light Gradient Boost classifier had the best overall AUC scores, even tough it showed a very small difference between them.

## 5.1 Next Steps

As any procedure that includes producing models in insurance companies it is necessary to make sure our model represents as closely as possible to the actual portfolio retention before deploying it into production. Therefore to ensure that, the next steps should be followed:

- Perform an Hyper-Parameter Optimisation in the Machine Learning Algorithms, with the aim of choosing a group of optimal hyper-parameters so it can obtain the minimum of the loss function or the maximum of accuracy leading then to a better performing model.

- Proceed with a more exact and detailed statistical test so its possible to obtain a more solid conclusion on the best model possible.
- To finalise an Impact Analysis should be performed in order to test the potential impacts our model can have in the market, this should include Key Performance Indicators (KPIs) and a detailed analysis of the Efficient Frontier.

# References


Anderson, D., Feldblum, S., Modlin, C., Schirmacher, D., Schirmacher, E. & Thandi, N. (2007), *A Practitioner's Guide to Generalized Linear Models*, Towers Watson.
**URL:** *https://www.casact.org/sites/default/files/database/dpp_dpp04_04dpp1.pdf*

Chen, T. & Guestrin, C. (2016), 'XGBoost: A scalable tree boosting system', *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* p. 785–794.
**URL:** *https://doi.org/10.1145/2939672.2939785*

Dietterich, T. G. (1998), 'Approximate statistical tests for comparing supervised classification learning algorithms', *Neural Computation,* **10**(7), 1895–1923.
**URL:** *https://doi.org/10.1162/089976698300017197*

Fawcett, T. (2006), 'An introduction to ROC analysis', *Pattern Recognition Letters,* **27**(8), 861–874.
**URL:** *https://doi.org/10.1016/j.patrec.2005.10.010*

Guillen, M., Parner, J., Densgsoe, C. & Perez-Marin, A. M. (2003), 'Using logistic regression models to predict and understand why customers leave an insurance company', *Intelligent and Other Computational Techniques in Insurance: Theory and Applications,* pp. 465–490.
**URL:** *https://doi.org/10.1142/9789812794246_0013*

Hastie, T., Tibshirani, R. & Friedman, J. (2008), *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y. (2017), 'LightGBM: A highly efficient gradient boosting decision tree', *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems,* p. 3149–3157.
**URL:** *https://dl.acm.org/doi/10.5555/3294996.3295074*

Micci-Barreca, D. (2001), 'A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems', *ACM SIGKDD Explorations Newsletter,* **3**(1), 27–32.
**URL:** *https://doi.org/10.1145/507533.507538*

Murphy, K. P., Brockman, M. J. & Lee, P. K. W. (2000), 'Using generalized linear models to build dynamic pricing systems', *Casualty Actuarial Society Forum,* pp. 107–140.

Nelder, J. A. & Wedderburn, R. W. M. (1972), 'Generalized linear models', *Journal of the Royal Statistical Society, Series A,* **135**(3), 370–384.
**URL:** *https://doi.org/10.2307/2344614*

Sokolova, M., Japkowicz, N. & Szpakowicz, S. (2006), 'Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation', *Australasian Joint Conference,* **4304**, 370–384.
**URL:** *https://doi.org/10.1007/11941439_114*