Analysis and Prediction of New Business Customers in the Auto Insurance Industry

Name: Martina Raftery (95549234)

School of Computer Science

National University of Ireland, Galway

Supervisor

Dr. Josephine Griffith

In partial fulfillment of the requirements for the degree of

MSc in Computer Science (Data Analytics)

August 31, 2020

DECLARATION I, Martina Raftery, do hereby declare that this thesis entitled "Analysis and Prediction of New Business Customers in the Auto Insurance Industry" is a bonafide record of research work done by me for the award of MSc in Computer Science (Data Analytics) from National University of Ireland, Galway. It has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: _____

Acknowledgement

I would like to thank my supervisor Dr. Josephine Griffith for her advice, guidance and motivation throughout this thesis.

I would also like to thank AutoUSA for allowing me to use their data for this research. Thanks to my employer and work colleagues, who have supported my decision to take a year out to pursue this masters.

Finally, I would like to thank my family and friends, with a special mention to my husband Daithí (Raff), son Fionn and daughter Cara for their constant support, encouragement and understanding throughout this research project and masters program.

Abstract

Many challenges within the Insurance Industry can be solved by data analysis techniques, e.g. generating new business, improving renewal churn rate, handling claims, reserving, fraud detection and improving the customer experience.

A large amount of data is being generated by the Insurance Industry every day making it imperative to have efficient and effective techniques perform data analysis. Machine learning techniques provide the potential to do pattern analysis on the insurance data to enable insights to be gained into the business.

The aim of this work is to use machine learning techniques to investigate and predict customer behaviour, which may ultimately lead to new business sales opportunities. Specifically, the research will carry out a comparative study on various supervised machine learning methods to classify customers with respect to how likely they are to select a particular insurance company when creating a new business car insurance quote.

The evaluation of the machine learning techniques will be done with the use of data from AutoUSA, a large American Auto-Insurance company. A significant portion of the AutoUSA's new business comes from 3rd party agency based comparative rating websites. Data is generated when these websites communicate with this company and it is this data that will be the use case for this research.

Keywords: Machine Learning, Data Mining, Feature Engineering, Imbalanced Data, Classification, Insurance Data, Churn Prediction.

Contents

List of Figures

List of Tables

List of Acronyms

ACORD Association from Cooperative Operations Research and Development. 19

AI Artifical Intelligence. 5

AIC Akaike Information Criteria. 8

ANOVA Analysis of Variance. 8

AUC Area Under the Curve. 17, 41, 43–45, 47–59, 62

CV Cross Validation. 44, 50, 51, 55–57

EDA Exploratory Data Analysis. 63

FPCA Functional Principal Component Analysis. 7

GAMs Generalised Additive Models. 8

GIGO Garbage In, Garbage Out. 24

GLM Generalized Linear Model. 8, 9

GLMs Generalized Linear Models. 6, 8–10

IOT Internet of Things. 1

KNN K-Nearest Neighbour. 44, 46, 54, 59

LR Logistic Regression. 44, 46, 59

ML Machine Learning. 11, 18, 25, 39

NN Neural Network. 9

NNs Neural Networks. 6, 9, 10

OECD Organisation for Economic Co-operation and Development. 45

PCA Principal Component Analysis. 5–7, 9, 39, 41

PII Personally Identifiable Information. 3, 4, 20, 24

RC1 Rate Call One. 2, 30, 35, 36, 63, 64

RF Random Forest. 44, 46, 59

ROC Receiver Operating Characteristic. viii, 17, 41, 47, 48, 58, 60

ROS Random Over Sampling. 44, 46

RTR Real Time Rater. 20, 21

RUS Random Under Sampling. 43

SMOTE Synthetic Minority Over-Sampling Technique. 43, 44

SQL Structured Query Language. 17, 19, 21

SVM Support Vector Machine. 45

VIN Vehicle Identification Number. 20, 22, 24

XGB Extreme Gradient Boosting. 44, 46, 59

XML Extensible Mark-up Language. 2, 18–21, 37, 63

XSLT Extensible Stylesheet Language Transformations. 18, 20

Chapter 1

Introduction

Many insurance companies generate vast amounts of data daily but are not utilising it to its full potential. Data only has value when useful information is extracted from it in order to provide beneficial knowledge to the business. This knowledge can help with decision making, customer prediction, process optimization and fraud detection.

The managing of this data can pose many challenges for these companies. Data may be stored in various repositories across different departments. Data can come from legacy or 3rd party systems so can have varying type and quality. There is a growing trend in the use of Internet of Things (IOT) applications and telematics in this industry which generates large volumes of data. This data has to be collected, stored, cleaned and organised in order for meaningful analysis to be carried out. Data security is also a challenge due to sensitive information contained in this data. Overcoming the challenges of big data and using this data effectively can create a competitive advantage for the organisation and allow the insurer to be more customer focused.

This research project involves analysing historical insurance data, using data mining and machine learning techniques in order to detect trends in the data so the company can gain insights into their customer profiles with the aim of growing their business and improving their competitiveness.

Note: At the request of the insurance company the pseudonym 'AutoUSA Insurance' has been used throughout this research as the data and the subsequent analysis is deemed business sensitive information.

1.1 Motivation

AutoUSA Insurance is an American auto-insurance company currently conducting business in multiple US states. A significant amount of their business originates from 3rd party comparative raters. As illustrated in Figure 1.1, an insurance agent, enters customer details into this web-based platform, selecting a rate button once complete. The data entered is sent to AutoUSA and other insurance carriers in XML format. This first stage is called Rate Call One (RC1). The data is then rated in real-time by the insurance companies if the companies business rules are adherred to. The premium and payment options are returned to the rater where applicable or a non-rateable message is displayed. If the agent then selects the returned AutoUSA premium, a Bridge button is selected on the rater and the entered details are uploaded directly onto AutoUSA's website where the quote can be purchased. This process is known as Uploading.

Only a small percentage of the raters RC1's get uploaded to AutoUSA's application and about half of quotes uploaded become policies. The challenge of this research is to analyse this RC1 data by using data mining and machine learning techniques in order to classify the data into a customer who will or not upload into AutoUSA. This will help AutoUSA target the type of customers they are not uploading in order to grow their customer base in the future.

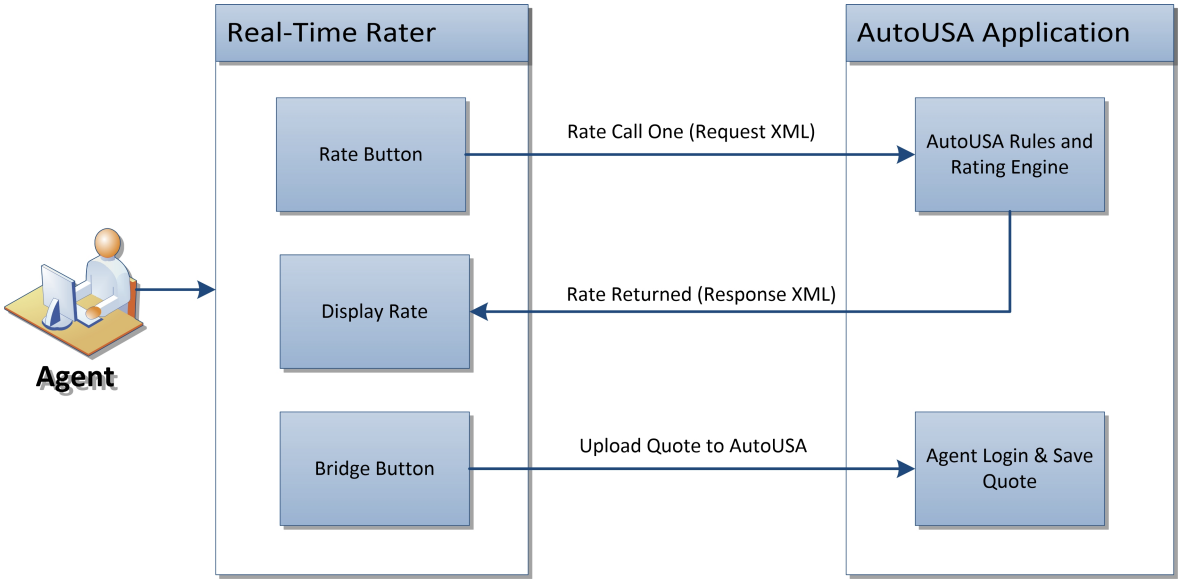

Figure 1.1: Rate Call One Process

1.2 Research Aim

The hypothesis of this research is to determine a suitable machine learning algorithm to forecast new business customers in the auto insurance company AutoUSA.

In order to answer this research question, the objectives of this work are defined as follows:

- Collect the required data from AutoUSA in order to carry out the research.
- Explore the data in order to identify any data issues, including missing data and outliers.
- Carry out exploratory data analysis so that trends and patterns can be determined and data can be better understood.
- Select appropriate machine learning algorithms to accurately carry out the classification prediction.
- Prepare and pre-process the data for the selected models using data mining techniques so that maximum performance results can be obtained from the models.
- Optimise the models to try and produce more accurate results.
- Analyse the results and compare the outcomes of the each of the models, selecting the best preforming model.

The research presents some challenges which will be tackled as part of this project, including protecting customers Personally Identifiable Information (PII), catering for imbalance in the dataset and dealing with poor quality data.

1.3 Thesis Layout

This thesis contains seven chapters. Chapter 1 provides an introduction to the research and an overview of the business domain of the project. Chapter 2 outlines a review of literature of the use of data mining and machine learning techniques in the insurance domain. Chapter 3 introduces some background information into the processes, techniques and tools used in this research. Chapter 4 details the data used in the research. This includes a description of the

data, how the data is collected, how the dataset was created and the steps carried out to pre-process the data. This includes the handling of missing data, outlier detection, data quality investigation, data transformation and handling of PII. Chapter 5 highlights the methodology of the research which includes feature engineering, sampling techniques, algorithm selection and optimisation. Chapter 6 evaluates the results and compares the various classification models. Chapter 7 provides an overview of the conclusions of the research and suggests possible future work.

Chapter 2

Related Work

2.1 Overview

Historically insurance pricing, underwriting and risk assessment was carried out using actuarial statistical analysis. Driven by increased data availability and proven success of machine learning techniques, this analysis can now be done faster and more effectively using machine learning methods.

A 2020 Accenture Technology Vision for Insurance Report by Rangwala et al. found that 79% of insurance executives believe collaboration between humans and machines will be critical to innovation in the future. Artifical Intelligence (AI) will transform the way insurers gain insights and interact with their customers. AI is a broad subject and encompasses many technologies. Current applications of AI in insurance include pricing, analysing customer behaviour, loss reserving, fraud detection, sentiment analysis, chatbots and telematics. This research will concentrate on data mining and machine learning methods only. An investigation will be carried out into the current use of these techniques in the insurance industry, discussing the different areas of this domain that benefits from machine learning and comparing these approaches.

Aleandri (2019), Roodpishi and Nashtaei (2015) and Parodi (2009) explores the use of unsupervised machine learning techniques such as Principal Component Analysis (PCA), Association Rules and Clustering to find patterns in insurance datasets, primarily to analyse customer behaviour. Wuethrich and Buser (2019) and Parodi (2009) compares classical actuarial processes

such as Generalized Linear Models (GLMs) with several supervised machine learning techniques such as Neural Networks (NNs), regression trees and bagging techniques for analysing different claim datasets both synthesised and real. Various use cases including rating factor selection, predictive modelling for reinsurance and claims reserving are investigated.

These papers and others highlight the different data mining and machine learning techniques used in the insurance sector and will be discussed in more detail in the next sections.

2.2 Data Mining

According to Witten and Frank (2005), data mining is defined as the process of discovering patterns in big data. The process must be automatic or (more usually) semiautomatic. The patterns discovered must be meaningful in that they lead to some advantage, usually an economic advantage. Data Mining can involve the following tasks: anomaly detection, feature selection and reduction, association rules and clustering. For this review, unsupervised learning data mining techniques were explored in the customer management aspect of the insurance domain.

Aleandri (2019) explains the use of PCA during the data preparation stage of the data mining process. PCA is used to perform dimension reduction of the data. The data used contained ten premium fields with high correlations between each other. Four of these had over 80% of a linear correlation with the others explaining less than 5%. After PCA the number of premium fields was reduced to four. Association Rules are used to select profiles of customers that are more likely to buy insurance products. This required numerical variables to be converted into categorical data. Each association rule is determined by an antecedent and a consequent. Limitations were added to avoid complicated or insignificant rules which led to 435 rules. These were represented on a scatter plot and analysed to detect marketing patterns. This analysis of a group of customers to find potential sales opportunities is known as market basket analysis. The results of these techniques are followed by logistic regression to predict the churn rate of the customer. This process boosted the performance of the regression performance.

Roodpishi and Nashtaei (2015) mentions the use of market basket analysis to better understand the customers behaviour which can lead to organisations being better able to deal with their

customer needs. Here clustering was carried out first and then association rules where applied to each cluster in order to determine hidden patterns in the insurance data. Segovia-Gonzalez et al. (2009) explains that the use of Functional Principal Component Analysis (FPCA) provides an improved approach to the conventional PCA. These two different versions of PCA were compared in the study of claims ratios. The FPCA method performed better, more information was gathered by a smaller number of components with less of an error rate. PCA on its own or together with association rules offer a good feature engineering option for the data preparation phase of insurance machine learning tasks.

Cluster analysis can be used to partition policyholders who display similar characteristics. Aleandri (2019) uses hierarchical clustering to analyse the reduction in variance in order to select a k value, i.e. k is the number of clusters used. A value of seven was selected which was then used to run the k-means algorithm. When the author applied logistic regression to predict churn rate to single clusters and the non-clustered dataset, improved performance was observed in the clustered dataset. This use of the two different clustering techniques tackles the disadvantages of k-means which includes the selection of the k value and the setting of initial conditions. This method saves computational time and cost. Parodi (2009) also discusses territorial clustering using k-means but proposes an alternative clustering technique called spectral clustering that he believes should be considered by the actuarial profession. Spectral clustering has its origins in graph theory. The author uses this type of clustering to collect claims history details and group them into clusters of similar behaviour before applying a regression model to the data. Spectral clustering resulted in only a slightly better accuracy over k-means, but he emphasised that it was a far more flexible method compared to k-means. Clustering techniques offer a useful tool for data analysis and can improve results of predictive modelling which is applied after clustering in a data mining pipeline.

2.3 Machine Learning

Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience, (Mitchell, 1997). A model learns from data and then uses this knowledge to make predictions on future or unseen data. Machine Learning has many

potential applications in the insurance industry.

Generalized Linear Model (GLM) is a classical modelling technique commonly used by actuaries in the tasks of insurance pricing, predicting claims frequency and reserving. GLM is an extension of the linear model with the main differences being that its response variable distribution is not limited to a normal distribution and its evaluation uses maximum likelihood instead of least squares error.

Wuethrich and Buser (2019) experiments with the Poisson GLM on a synthesised car insurance 3rd party liability claims dataset in order to model claim frequency. The authors carry out feature engineering steps including converting categorical variables to numerical representations, converting continuous features to categorical classes and applying data compression. They generated five different models using the GLM function in R programming language with different combinations of variables based on statistical analysis. The model with the best out-of-sample loss was one that included all variables. Generalised Additive Models (GAMs) were also tried on the same data which led to a slightly better result. Parodi (2009) also investigated the use of GLMs in non-life insurance. The author wanted to compare their use with other machine learning techniques such as Neural Networks. Like Wuethrich and Buser (2019), a controlled experiment was performed using artificial data in order to model the claims frequency as a Poisson GLM. Features used in the model were selected using a greedy forward selection approach i.e. selecting the simplest model first and then adding a feature at each step until the best feature subset was selected. Akaike Information Criteria (AIC) was used to decide best feature selection whereas Wuethrich and Buser (2019) used ANOVA. Parodi (2009) also mentioned using regularisation together with GLMs as a better alternative to the GLM.

GLMs are a well-developed and understood methodology in the actuarial world. They are flexible as they allow for different distributions, yet the distribution must be defined at model specification. The model selection process can be a complex and time-consuming task. The experiments in these papers only use between five and eight different attributes. If this number of attributes is greater or the data is high-dimensional, it would be very difficult to investigate all the different interactions between the variables and to manually add them to the model. When other techniques were used, better results were attained.

Neural networks are utilised by insurance companies to predict claims, for rate making and

reserving as an alternative to more traditional approaches i.e. GLMs. Parodi (2009) explains that Neural Networks can allow greater generality over GLMs and they also take less time for feature selection but mentions that there is a problem of "prediction without interpretation" so they can be difficult to justify in the insurance environment where transparency is required. The author does suggest that they can be useful for bench marking when comparing the performance of other models and for exploratory analysis. Styrud (2017) experiments with the use of Neural Networks for insurance rate making. A simulated car insurance dataset of 5,000 observations was trained using a NN with one hidden layer comprising of 3,4,5,6 and 7 nodes. A sigmoid activation function was selected and 5-fold cross validation was performed. A model with four hidden layers performed the best on the test set. When compared to the GLM, the NN returned slightly better precision results but was less stable. The author reiterates the difficulty of explaining the predictions obtained by the NN and highlights the fact that the process of fine-tuning the model can be time-consuming and complex. No practical implementation of a neural network was carried out by Parodi (2009) and a small non-real world dataset was used by Styrud (2017) with limited tuning. Therefore it seems that more research is required using a large accurate dataset with more sophisticated configuration settings in order to prove if there are any significant benefits of Neural Networks over GLMs.

2.4 Conclusions

The introduction of data mining and machine learning techniques into actuarial work has many benefits in the insurance domain. These main benefits include enhancing business profitability, improving customer satisfaction and better forecasting of sales and losses.

The use of data mining unsupervised learning techniques in order to discover patterns in the insurance data has been explored in various papers. PCA, association rules and clustering were experimented with on various datasets which led to more accurate prediction results when machine learning was carried out as a later step.

GLMs are a well-used regression model utilised by actuaries for insurance pricing. It is suitable for large data sets and is flexible but can be difficult to use with high dimensional data. Neural Networks offers an alternative to the GLM as they address some of their limitations. NNs work

better with high dimensional data and also do not require the analysis of variable interactions. They do posses drawbacks with the main disadvantage being its lack of explainability. More research would be beneficial to prove if NNs show a significant improvement over GLMs using real insurance datasets.

The majority of papers reviewed for this project, outlined the use of data mining and machine learning for pricing, profiling customers claim risk, claim reserving and using unsupervised learning techniques in the customer management area. The research for this project will concentrate on customer management but instead of using unsupervised learning methods will use different data analysis techniques to build up knowledge on why a new business customer selects an insurance company over others and then uses alternative machine learning methods to GLMs and NNs to predict this new business behaviour.

Chapter 3

Overview of Techniques and Technologies

3.1 Overview

The two main categories of machine learning techniques are supervised and unsupervised learning (Figure 3.1). Supervised learning uses labelled training data to make predictions. Unsupervised learning uses unlabelled data to try and find patterns or groupings in the data. The data used for supervised learning is made up of two types of values/variables:

- Target variable(Y), also referred to as the dependant or response variable.
- Attribute value (x), also referred to as the independent/input or predictor variable. There can be more than one of these and they are used to help predict the target variable.

Unsupervised learning only has input data and no corresponding target variable. Each row in the dataset is referred to as a data point or observation. An hypothesis function is used to predict the output variables on unseen data i.e. Y = f(x). The data used in this research is labelled, therefore various supervised ML algorithms will be trained on the data and compared in order to obtain the best prediction results. This research aims to predict whether a customer will upload a quote into the AutoUSA application or not. This is a binary classification task as the output variable is categorical and made up of two classes i.e. *QuoteUploaded* = 'Y' and

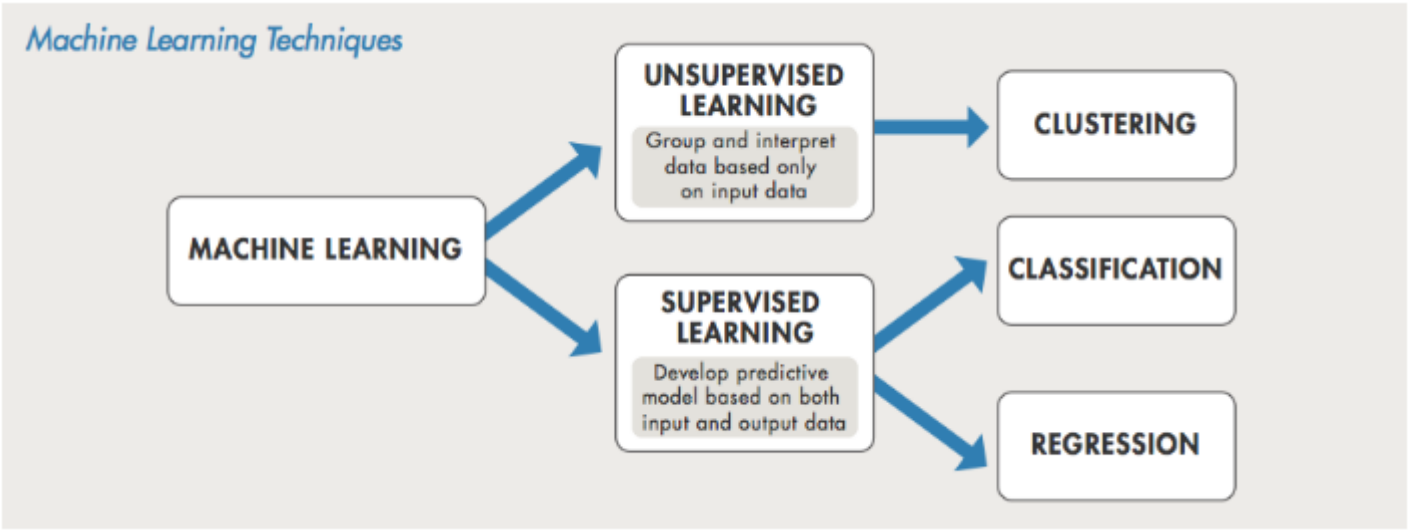

Figure 3.1: Supervised vs Unsupervised Machine Learning Techniques. (Source: Kumar)

$QuoteUploaded$ = 'N'. These classes will be encoded to numeric values before modelling i.e. $QuoteUploaded = 1$ and $QuoteUploaded = 0$. This chapter will explain the algorithms that will be used to perform this binary classification.

3.2 Logistic Regression

When explaining logistic regression, it is benefical to start with the linear regression model. A linear model is a simple, widely used statistical technique used in predictive modelling analysis. Linear regression is used to define the relationship between the input and output data point pairs using a straight line. The line should be as close as possible to each of the data points. For this model, the output variable is a continuous value. Instead logistic regression uses the natural logarithm function to find the relationship between the variables. Here the output variable is categorical in nature. Another name for the logarithm function is the sigmoid function:

$$y = \frac{1}{1 + e^{-x}}$$

It gives an 'S' shaped curve instead of a straight line in linear regression, see Figure 3.2. The algorithm outputs a probability between 0 and 1. If the output of the function is more than 0.5 we classify the outcome as 1 or 'Yes' and if it is less than 0.5 we can classify it as 0 or 'No'.

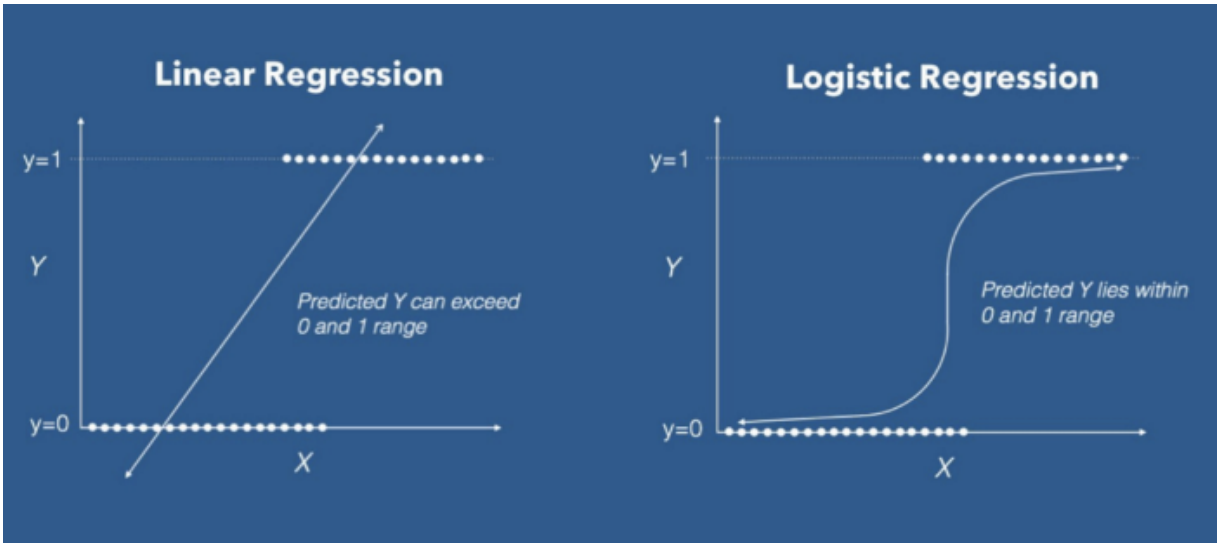

Figure 3.2: Linear vs Logistic Regression. (Source: Joshi)

The threshold value of 0.5 can be modified and is usually problem-dependant.

3.3 Random Forest

Random forest is an ensemble machine learning model which can be used for classification problems. It is made up of a combination of decision trees, as can be seen in Figure 3.3. Ensembles combine a group of weak learners to form a stronger learner. Each classifier is combined to return a final decision using a majority vote. A process called bagging (bootstrap aggregation) is used to select the data for each tree. Random sub-setting of the attributes and rows in the dataset for each tree is carried out in order to make sure each tree is partially independent. This process reduces the variance and introduces more randomness and diversity to the process. The results of each tree in the forest is aggregated through majority voting which leads to more accurate predictions.

3.4 K-Nearest Neighbour

The K-Nearest Neighbour algorithm presumes that similar data points exist close to each other. The concept around this algorithm is calculating the length of the straightest line between these data points. If there are two data points (x1, y1) and (x2, x2), use the euclidean distance to

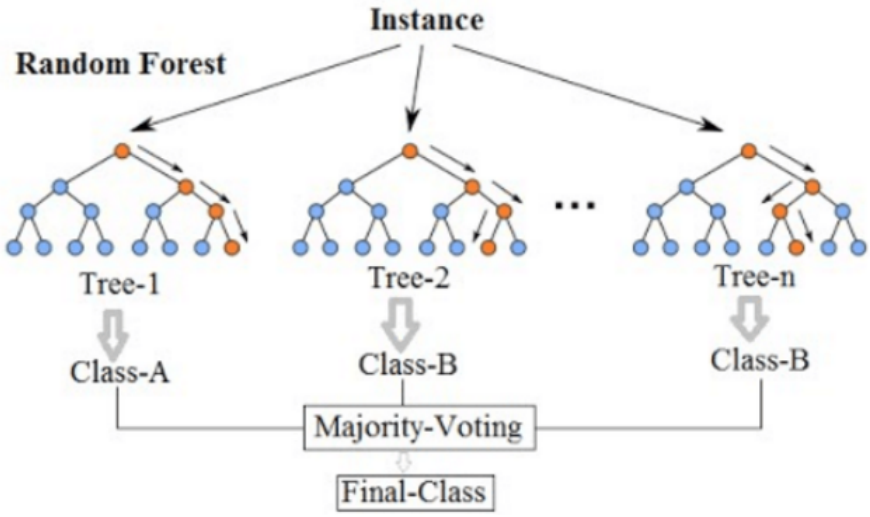

Figure 3.3: Random Forest Ensemble. (Source: Chakure)

work out the distance between these points using the following formula:

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

A value for k is selected with k being the number of data points grouped together. Figure 3.4 demonstrates a value of three being selected for k. If a value of three was selected for k, using the euclidean distance, the algorithm works out the three data points that are nearest to the example we want to predict. The majority label among these three data points would be returned and this will be the predicted class. This process can involve some trial and error to decide what value of k to use. Different values of k are tested and the value of k which returns the best results is selected.

3.5 XGBoost – Extreme Gradient Boosting

XGBoost is an ensemble machine learning algorithm like random forest. It's base model is also decision trees but it uses gradient boosting instead of bagging. Boosting is a more iterative approach to bagging. With bagging, the individual trees are trained independently from each other but with boosting they are trained in succession. The new tree model learns and corrects the errors of the previous tree model. Therefore the model is improving as new trees are added.

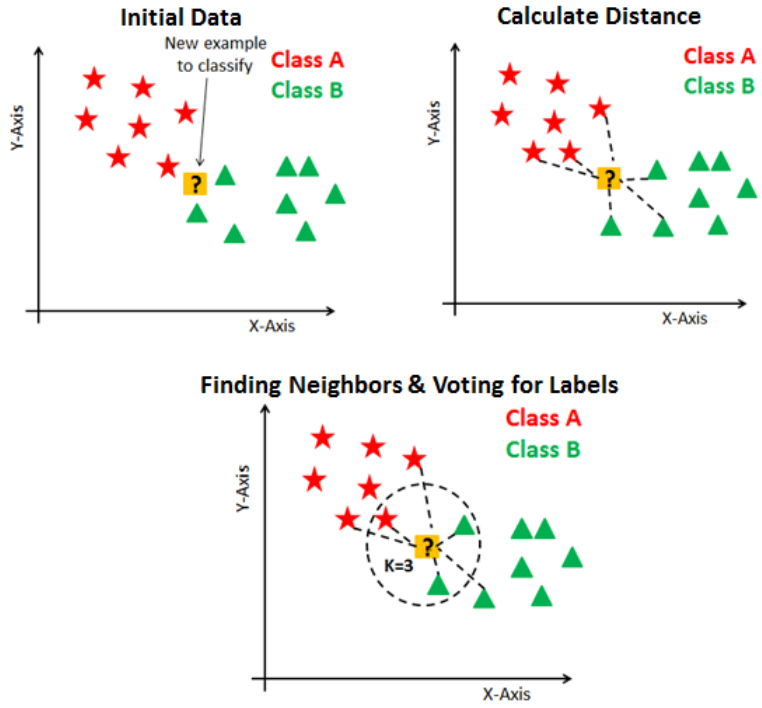

Figure 3.4: K-Nearest Neighbour. (Source: Navlani)

Weak learners are combined which results in a stronger end model.

3.6 Evaluation Techniques

3.6.1 Cross Validation

In order to evaluate how well a model has performed, various cross validation techniques can be used. The dataset can be split into training and testing data e.g. 80% training, 20% testing. The model is trained on the training set and then the results are validated on the hold out test set. An improved approach is k-fold cross validation. Here the data is split into k random subsets/folds, k-1 subsets are used for training the model and 1 subset for testing. Scores are recorded. The process is repeated until each fold of the k folds has been used as a testing set. The average of the scores for each fold becomes the performance metric of the model. Common values for k are 5 and 10. This method can prevent overfitting and is also useful when data is limited.

3.6.2 Classification Metrics

Various metrics exist to evaluate the performance of a binary classification problem. This section will provide a brief explanation and formulas of the metrics used in this research. Confusion matrices are a useful visual tool to allow for a detailed analysis of a models accuracy, see Table 3.1.

	Actually Positive	**Actually Negative**
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Table 3.1: Confusion Matrix Explained

It displays the following information:

- **True Positives (TP)**: Predict the class as Yes, when the actual class is Yes
- **True Negative (TN)**: Predict the class as No, when the actual class is No
- **False Positives (FP)**: Predict the class as Yes, when the actual class is No
- **False Negatives (FN)**: Predict the class as No, when the actual class is Yes

Other metrics include:

- **Accuracy**: Measures how many observations, both positive and negative, were classified correctly.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

- **True Positive Rate (TPR)**: Measures how many observations out of all positive observations, were classified as positive. This is also referred to as Sensitivity or Recall.

$$TPR = \frac{TP}{TP + FN}$$

- **True Negative Rate (TNR)**: Measures how many observations out of all negative observations, were classified as negative. This is also referred to as Specificity.

$$TNR = \frac{TN}{TN + FP}$$

- **False Positive Rate (FPR)**: Measures how many observations out of all positive observations, were classified as negative. This is also referred to a Type I Error.

$$FPR = \frac{FP}{FP + TN}$$

- **False Negative Rate (FNR)**: Measures how many observations out of all negative observations, were classified as positive. This is also referred to a Type II Error.

$$FNR = \frac{FN}{TP + FN}$$

- **ROC Curve**: The ROC Curve plots the true positive rate against the false positive rate using different threshold values between 0 and 1.

- **AUC Score**: This is the Area Under the ROC Curve or ROC AUC Score. It tells us how well the model is at distinguishing between the two different classes.

3.7 Technologies Used

Python is a free open-source software programming language. It has a large collection of libraries for loading and analysing data, statistical analysis, machine learning and data visualisation. A list of the Python packages used for this project are outlined in Table 3.2.

SQL is a common scripting language used for relational database manipulation and was required for extraction of some data from a database.

Package Name	Package Use	Description
os	Other	Used to interact with the operating system e.g. get and set the working directory
lxml	Data Manipulation	Used for handling XML data and carrying out XSLT transformations in Python.
date	Data Manipulation	Used to work with date objects in Python
pandas	Data Manipulation	Used to work with data structures in an easy intuitive way in Python
numpy	Data Manipulation	Used to manage arrays and matrices in Python
seaborn	Data Visualisations	This library provides a high-level interface for creating statistical graphics in Python. It is based on matplotlib.
matplotlib	Data Visualisations	Used to create 2-dimensensial graphs and plots
sklearn	Machine Learning	Allows the running of classification, clustering and regression algorithms. Also provides functions for feature selection and ML performance evaluation.
mlxtend	Machine Learning	This library provides machine learning extensions. Provides sampling techniques for machine learning algorithms
Xgboost	Machine Learning	Allows for the implementation of the gradient boosting decision tree algorithm XGBoost in Python
scipy	Statistics	Used for carrying out mathematical and scientific calculations

Table 3.2: Python Libraries and their Description

Chapter 4

Data Preparation and Exploration

The data for this research is made up of semi-structured data in the form of XML and structured data from a SQL relational database. XML is the prominent data type and does not conform to a defined data model like the SQL data but it does have easily identifiable groupings and hierarchy. The XML follows the ACORD XML Standard for Insurance. This facilitates the peer-to-peer communication between insurance agents and brokers, financial institutions, insurance organizations and software providers world-wide.

4.1 Raw Data Collection and Dataset Creation

A web service is used to pass the XML data between the rater and insurance company. This XML data is not currently saved by the insurance company due to its large volume. Quote data is not saved until it is uploaded. The XML passed between the rater and AutoUSA only gets generated when a configuration logging setting is switched on in the application and is normally only used for troubleshooting purposes. In order to carry out this analysis this configuration flag was switched on for a period of a week between the 26$^{\text{th}}$ of February and the 3$^{\text{rd}}$ of March, 2020. Approximately 150,000 files were created in this time period, half of which were request files i.e. data sent from rater to insurance company and the other half were matching response files i.e. data sent from insurance company back to rater. The name of the file contains a unique ID that links the request and response file.

The XML files contained Personally Identifiable Information (PII) including First Name, Last Name, Address, Phone number, Email Address, Vehicle VIN number and Driver's License Number. Instead of anonymising the data, XSLT files were created to extract the non-sensitive data only.

The request XML files contained three distinct sections which were extracted into three separate csv files using the *lxml* package in Python:

- Summary Data: includes quote level data such as the policy term, effective date, postal code of the customer, quote level coverage details and an RTR reference number. The summary csv file contains one row per XML file.
- Driver Data: includes details of each driver on the quote e.g. date of birth, driver gender, marital status, license type, occupation etc. The driver csv file contains one row per driver.
- Vehicle Data: includes details of each vehicle on the quote e.g. make of vehicle, year of vehicle, use of vehicle etc. The vehicle csv file contains one row per vehicle.

The XML file name links the request and response files and is unique to each Rate Call One. Created a column called *FileName* which contained the name of the request/response file and added it to the summary, vehicle and driver csv files described above. The RTR reference number i.e. *RTRRefNo* was also added to each row in the three csv files in order to link the data at a later stage.

The response XML files contains premium information, details on whether the response was a success or failure and details on how the vehicles are rated i.e. which vehicle is rated against which driver. This response data was extracted into one csv which contained one row per vehicle.

While carrying out this XML extraction, 257 files were noticed to have an "Index out of bounds" error. This error is caused by an error in the code that writes the XML to file. These were ignored by the extraction job.

The last piece of information that is required for this analysis was whether these quotes were uploaded to AutoUSA. This is not information that is available in the XML data as it is a step

DataFrame Name	Data Origin	No of Rows	No of Columns
summaryData	Request XML File	74,659	22
driverData	Request XML File	102,342	25
vehicleData	Request XML File	102,434	27
responseData	Response XML File	94,369	12
quoteData	SQL QuoteHeader Table	6,469	148

Table 4.1: Python Dataframes

after the point in time the response XML file is logged. This data is saved in the AutoUSA database. A SQL view was created to extract this information using the RTR reference number in the XML data, then the SQL export data utility was used to extract this data to csv format. All these csv files were then loaded into pandas data frames in Python for analysis and data preparation. Table 4.1 shows the breakdown of each of these data frames. As the data is spread across multiple files, we require the data to be joined into one dataset in order to do further analysis. Figure 4.1 outlines how the data frames were merged into one final dataset.

When carrying out this process, it was observed that the responseData data frame contained 12,924 rows with a status of 'Failure'. These are quotes that failed AutoUSA's business rules so therefore do not contain any rate information. These rows were dropped from the data frame before merging. The analysis of these failed quotes could be carried out in the future by the company but would be out of the scope of this research as they would result in gaps in the dataset due to no premium or rating information. The resulting merged data frame has 80,035 rows and 78 columns.

FileName and *RTRRefNo* are the two main ID fields in the dataset. *FileName* is generated by AutoUSA and used to link the data in the request/response XML files. There are duplicates of this ID in the dataset as a row in the dataset corresponds to one vehicle and the driver that is rated against it. If there are more than one vehicle on the quote, there will be more than one instance of this *FileName* ID in the dataset. *RTRRefNo* is an ID field generated by the rater and is unique to the rater quote. The agent creates a quote and this number is created and sent in the XML to AutoUSA. There may be a case where the agent might change some field(s) on the rater and re-rate the quote with this change. The XML will get sent again to AutoUSA with the same ID number. This can lead to duplicates of this ID in the dataset, which is legitimate as some aspect of the quote has changed, so the premium will more than

likely be different on the second rate of the quote.

4.2 Data Preparation

The comparative rater is a web based application which is made up of both required and optional fields that the agent has to fill in with the customer details. The majority of the fields are drop-down lists or strongly validated text input fields. These result in more structured, reliable data in our dataset. There are some free form text fields which will lead to unstructured and sometimes unreliable data due to user error. Optional data fields lead to empty fields in our dataset.

4.2.1 Data Cleaning

As part of a preliminary analysis of the dataset, we explored the missing value ratio of all variables in the dataset. 30 of the variables had over 40% empty values. This is a substantial percentage and would be difficult to try and fill in these values in a meaningful way in order to maintain the integrity of the data. After reviewing this list of variables, using domain knowledge of the variables, it was decided to drop these columns from the dataset. These included information that would not be significant in explaining the behaviour of the customer.

Of the remaining features, some contain few rows with missing values. These rows were dropped from the dataset as setting default values would not be appropriate i.e. *DrvDOB*, *State*, *VehCity*, *DrvSex*, *City*, *County*, *PostalCode*, *VehState* and *BICoverageRTR*. For other features, missing values were set to default values e.g. *DrvOccup* was set to 'UNKNOWN' and *DrvLicJur* to 'NONE'. Also dropped columns that have only one unique value as these have no variance i.e. *BillType*, *Status* and *DrvAccidents*.

The *VehSymbolRaterNum* field holds the symbol of the vehicle. This is a numerical field in the rater but has no validation. A symbol number is a way of matching premiums for each particular type of car to losses of that particular car. AutoUSA uses a 3rd party company to attain this symbol information based on a vehicles VIN number. The higher a vehicles symbol, the higher its rated premium, so it would be useful information to analyse. This company's

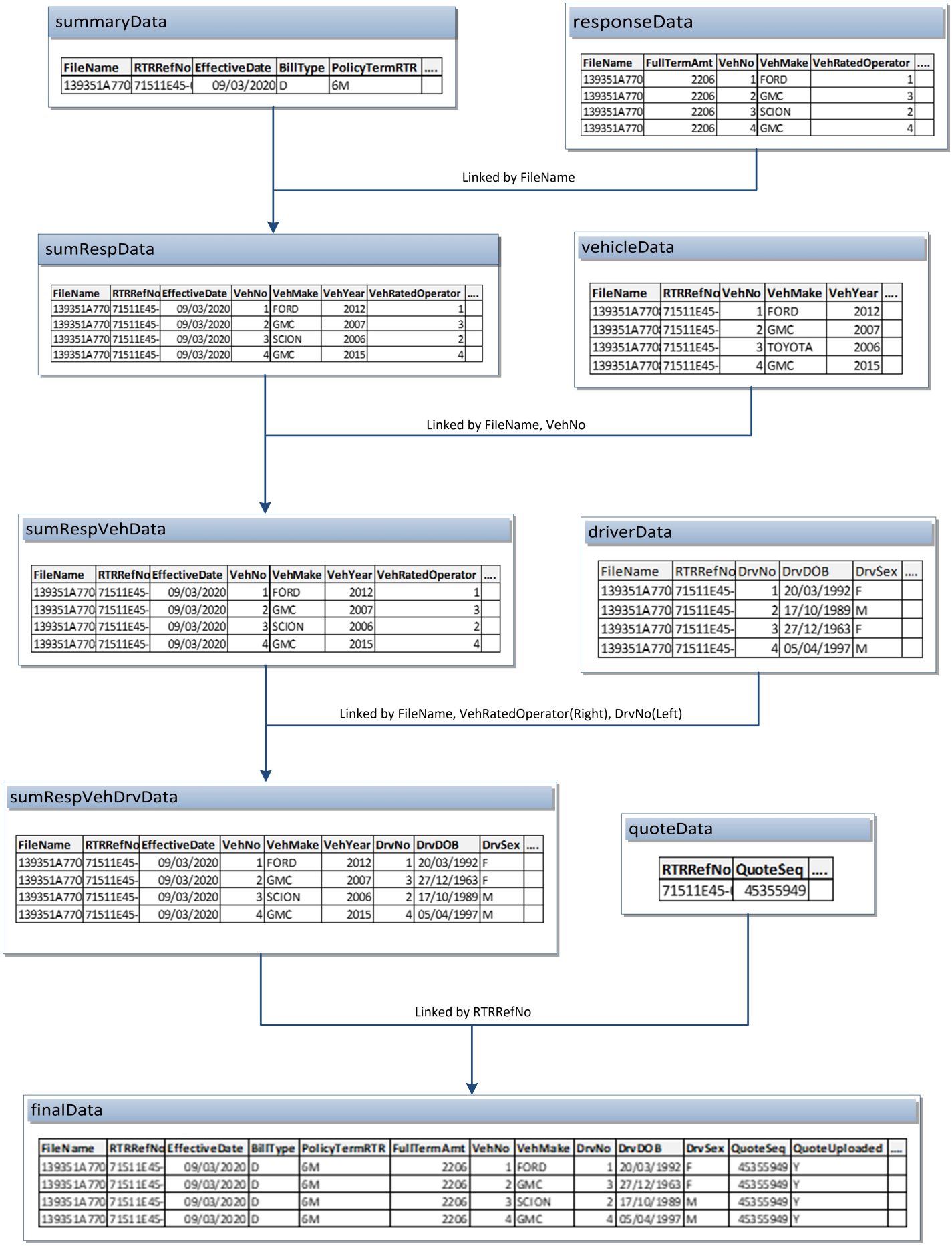

Figure 4.1: Process of Merging Data Frames to Create Final Dataset

symbol allocation has a range of 1 to 75. Reviewing the data in this column in our dataset, there are 36,018 rows with a symbol over 75, 38 with a value of 0 and 863 with missing value. Other insurance carriers use different companies to assign these symbol numbers which has led to the vast range of numbers contained in this field (i.e. 1,220 unique values). The symbol can also be obtained from the VIN number of the vehicles but this was not extracted as part of this project as it is Personally Identifiable Information (PII). Due to these reasons, this column was dropped from the dataset. Using domain knowledge of the remaining variables, we dropped certain date and location columns i.e. *EffectiveDate*, *PostalCode*, *City*, *VehCity*, *VehPostalCode*, *VehCityFirst*.

4.2.2 Data Quality Investigation

As well as missing values, the data could also contain inaccurate data. The quality of the data is important to the success of building a model and performance of a classification task. In Computer Science, Garbage In, Garbage Out (GIGO) is the concept that implies bad input will result in bad output, (Christensson, 2015). Due to the rater containing free form text fields, data entry errors can occur which can result in bad or duplicate data.

The data was analysed for quality and some discrepancies were found. Six coverage fields exist in the dataset in the format 'CoverCode' + *CoverageRTR* e.g. *BICoverageRTR*. All coverage categories should contain a letter followed by a number e.g. 'A0', 'B5'. 325 rows did not contain numbers in their coverage fields, so these rows were dropped from the dataset.

Two similar columns exist to record the employment information of the customer. *DrvEmployStatus* has 298 unique values and *DrvOccup* has 647. Most of the values in the field *DrvEmployStatus* has the values 'E' for Employed, 'H' for Homemaker, 'R' for Retired, 'S' for Student and 'U' for Unemployed but there are 15,904 rows with other inexplicable data and 3,787 empty rows. *DrvOccup* contains many unique values as this field was free form on rater. There are many spelling variations or different descriptions of the same occupation in this field. To improve the quality of this information, the empty values in the *DrvEmployStatus* column were set to 'X' for 'Unknown'. For any rows that had bad data in the *DrvEmployStatus* column, the *DrvOccup* was checked for existence of the text 'Homemaker', 'Retired', 'Student' , 'Unemployed' or 'Unknown' and if it did exist the *DrvEmployStatus* was set to the values 'H',

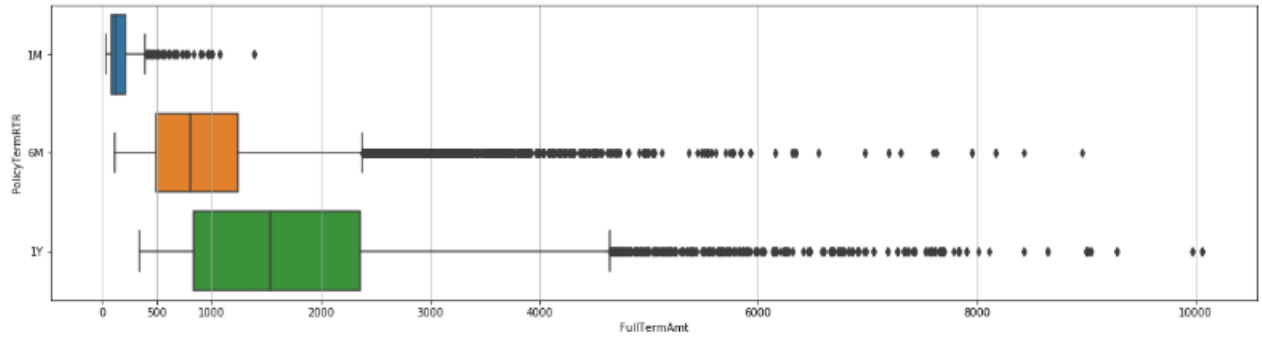

Figure 4.2: *FullTermAmt* Variable Boxplot

'R', 'S', 'U' or 'X' respectively. Then for all other rows where an occupation existed in the *DrvOccup* field, the *DrvEmployStatus* field was set to 'E' for Employed. *DrvOccup* column was then dropped from the dataset and *DrvEmployStatus* resulted in six distinct categories.

4.3 Data Pre-processing

Many feature engineering and machine learning techniques require variables to be qualitative. The transformation of continuous numerical data to discrete or categorical values is referred to as discretisation. This process involves splitting the values into separate bins across the range of the variable. This can make the variable easier to understand, more compatible with certain ML models and reduces noise by smoothing the effects of outliers in the data. *FullTermAmt* is the total calculated premium. The dataset it made up of 1 month, 6 month and annual quotes so therefore this value will need to be analysed on a per term basis, otherwise it will have a very skewed distribution.

The boxplot in Figure 4.2 shows the differences in the distribution of *FullTermAmt* for each term. This also shows that this premium value is right skewed with many outliers. If this value was to be split into three equal width bins across the range of the variable per term, we would see very little data points in the higher value range. Therefore, for this dataset it would be more suitable to split the data using equal frequency discretisation. Here each bin carries the same amount of rows. This is a better solution for skewed variables with outliers as it allows the observations to be spread over the different bins equally. We used the quantile-based discretisation function in Python (*qcut*). Table 4.2 displays the breakdown of the ranges of the

Term	Category	Premium Range
1M	Low	(35.999, 97.0]
	Medium	(97.0, 181.0]
	High	(181.0, 1389.0]
6M	Low	(118.999, 590.0]
	Medium	(590.0, 1068.88]
	High	(1068.88, 8961.0]
1Y	Low	(343.759, 1095.76]
	Medium	(1095.76, 2052.0]
	High	(2052.0, 10065.04]

Table 4.2: *FullTermAmt* Variable Discretisation

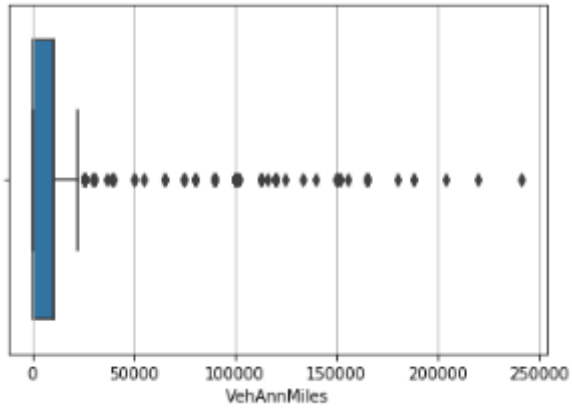

Figure 4.3: *VehAnnMiles* Variable Boxplot

three bins generated for the *FullTermAmt* variable for each of the term categories.

There were other options for handling this transformation. Looking at the boxplot in Figure 4.2, we could have dropped the outliers from the dataset or decided on a different max value and set values above this max value to the max value. This is the appropriate action, in this scenario as these premium values are a calculated value and not a data entry error. An error may have been entered in another field that caused this quote to have a very high premium but it is not an actual error in this field so we believe that the correct approach was used.

Figure 4.3, indicates that *VehAnnMiles* also has outliers. This is a free text field on the rater, therefore the outliers are more likely due to error e.g. the agent may have entered an extra zero at the end of the number in error. For this attribute, values over 100,000 will be dropped (68 rows). There are 22,804 rows with a value of zero. It wouldn't be legitimate to have annual mileage of 0 for this many instances. It may be legitimate to have a very small number of these

cases. These values will be set to a default value 12,000, which is the standard US average mileage. To discretise this attribute we used the following custom bands based on domain knowledge:

- 0 – 7,500 Low Mileage
- 7,500 – 15,000 Average Mileage
- 15,000+ High Mileage

90% of rows in dataset fall into the Average Mileage Category.

VehOdis is the one way mileage of the vehicle. This field has many outliers including values over 800 miles. There are 48,926 rows with the value 0, which is almost 64% of the dataset. We could set these zero values to the average value of the attribute but as there is a very significant proportion, the better solution would be to drop the column in order to keep the integrity of the data. The *DrvDOB* variable, is a date field which holds the date of birth of the driver. Machine learning algorithms work best with numerical data so therefore it would be more appropriate to use the age of driver instead of the date of birth. The age of driver was calculated using the date of birth and transaction date. This value was then discretised using a predefined age band that AutoUSA use for rating purposes, see Table 4.3. A similar transformation was carried out on the year of the vehicle, *VehYear* field. We calculated the age of the vehicle and then transformed this to vehicle age bands. A bin was created corresponding to the age of the vehicle, with a max value of 30. Any vehicle with an age over 30 years, was set to the 30 bin.

In order to transform *TransDateTime* field, the day of week was extracted creating a new column, *TransDayOfWeek*. This may be useful information to have when trying to find trends in the data. The two columns *DrvLicType* and *DrvLicTypeRater* contain very similar data. After analysing, we merged the contents where possible and dropped the *DrvLicType* field. We created a new column *OutOfStateLicense* and set this to 'Y' if *DrvLicJur* and State fields differed and set to 'N' otherwise. Similarly, we created a new column *GaragedOutOfState* and set to 'Y' if *VehState* and *State* were different and set to 'N' otherwise. Was then able to drop the *DrvLicJur* and *VehState* fields.

MinAge	MaxAge	Band Name
15	18	15
19	20	19
21	22	21
23	24	23
25	29	25
30	34	30
35	39	35
40	44	40
45	49	45
50	59	50
60	69	60
70	74	70
75	110	75

Table 4.3: *DrvAgeBand* Variable Discretisation

The final dataset contains 76,723 rows and 30 columns. A description of the dataset variables are contained in Table A.1.

4.4 Exploratory Data Analysis

During the data preparation phase it is useful to carry out a detailed analyses of the data in order to understand it and make informed decisions on how best to transform it and prepare it for the modelling stage. The target variable is *QuoteUploaded* which is a Boolean variable that denotes whether a customer will upload their quote to AutoUSA. In this section, we analysed the relationship between this target variable and a subset of other features in the dataset. This helped gain insight into trends that may exist in the data. Firstly we will look at the breakdown of *QuoteUploaded* in the dataset (Figure 4.4).

The dataset has the following breakdown:

- No of rows where *QuoteUploaded* is 'Y': 12,059 (15.72%)
- No of rows where *QuoteUploaded* is 'N': 64,664 (84.28%)

This is an imbalanced dataset i.e. classes are not represented equally. This may pose some issues in the modelling phase and will need a sampling technique to be applied to the data

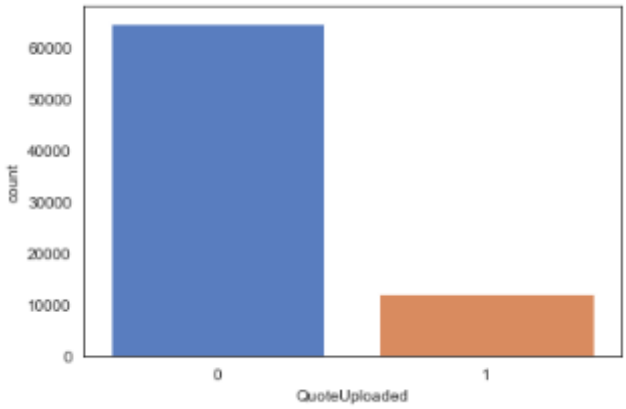

Figure 4.4: A Count Plot of the Target Variable *QuoteUploaded*

before training. We will go into more detail on this topic in Chapter 5 Section 5.3. Next, we will look at the premium field. Intuitively, the premium field would be a significant feature of an insurance quote so therefore would be an important field to analyse. Figure 4.5 shows a boxplot of the premium before discretisation, grouped by policy term.

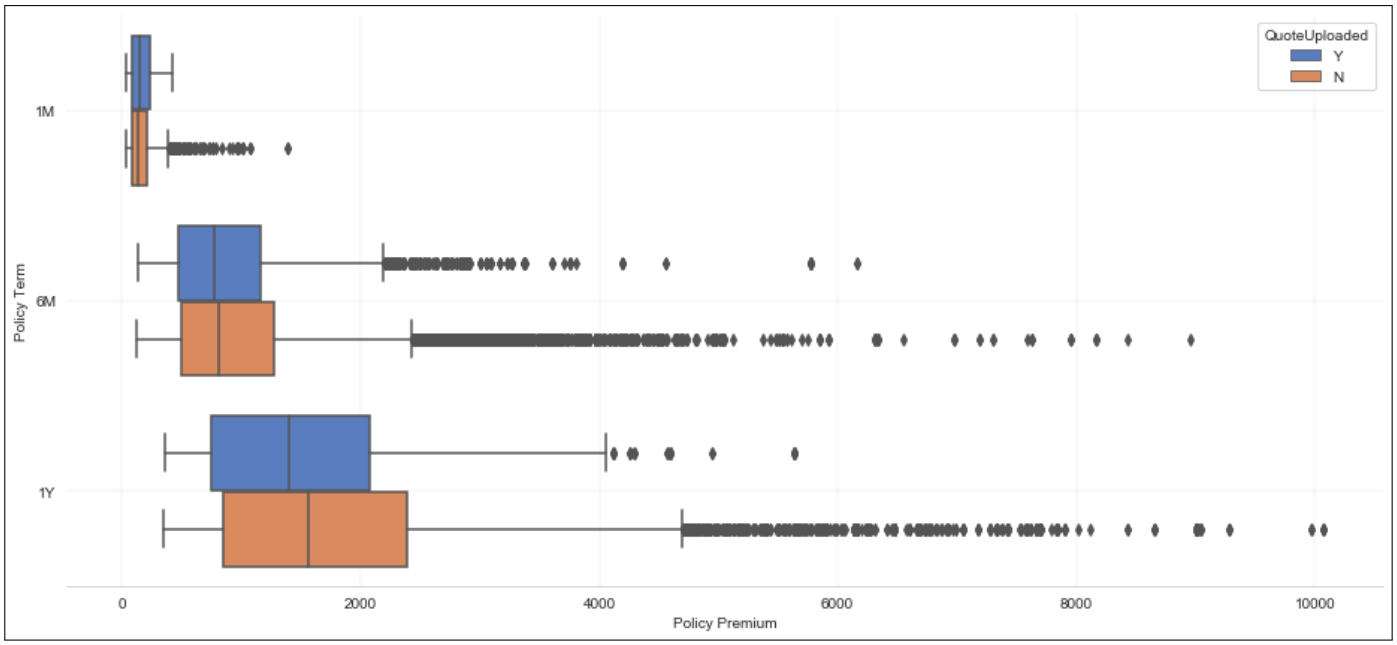

Figure 4.5: A Boxplot of the *FullTermAmt* Variable Per Policy Term

We can make the following observations:

- There are more outliers when *QuoteUploaded* = 'N' for all policy terms, which can be explained by the larger proportion of data in this category. There are more extreme

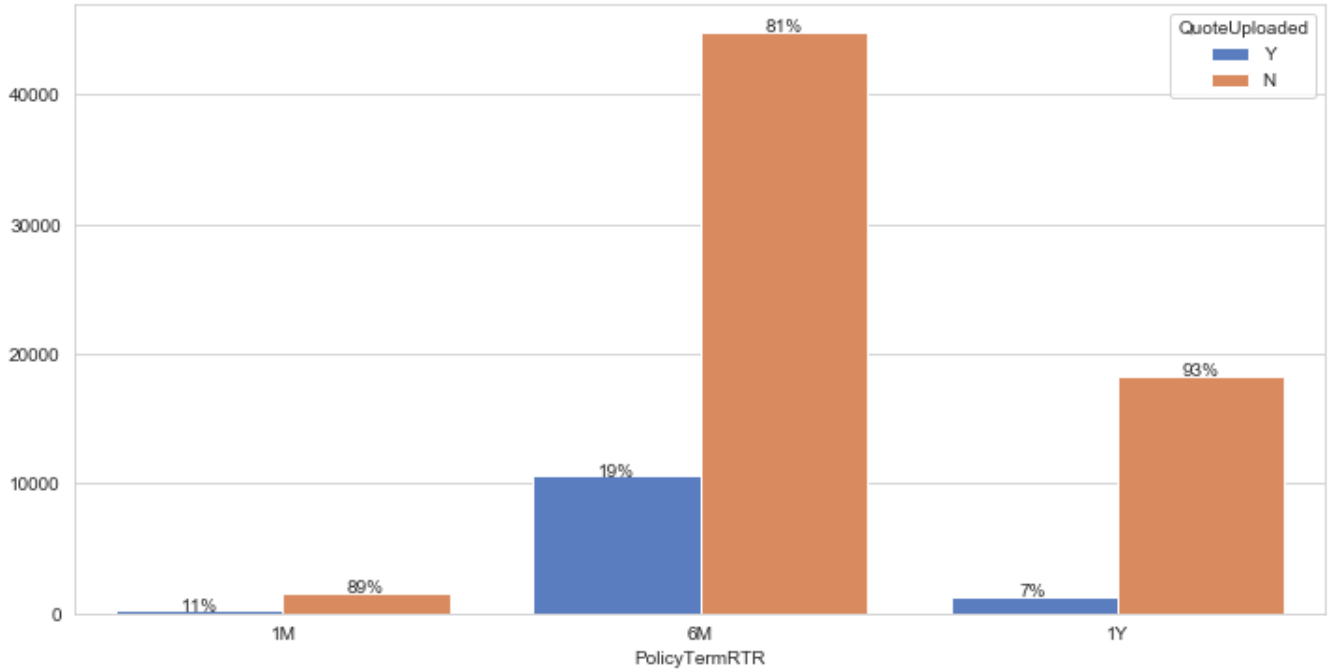

Figure 4.6: A Count Plot for the *PolicyTermRTR* Categorical Variable

outliers when *QuoteUploaded* = 'N', e.g. a 6 month premium of almost $9,000 and an annual premium over $10,000.

- All distributions are right skewed, with *QuoteUploaded* = 'N' more right skewed than *QuoteUploaded* = 'Y' in 6 month and 1 Year terms.
- For 1 Month term, the distribution of the target variable *QuoteUploaded* is similar, with similar min, max and median values also.
- For 6 Month and 1 Year term, RC1's have higher median and max values for *QuoteUploaded* = 'N'.

We selected various summary, driver and vehicle fields and created visualisations using the *Seaborn* library in Python to show the counts of observations in each category using bars. We grouped each category by *QuoteUploaded* value so we can show the breakdown of the target variable for each feature analysed.

As this is an imbalanced dataset with an approximately 85/15 split, we can see many of the splits in the categories have a similar breakdown. The following observations were noted:

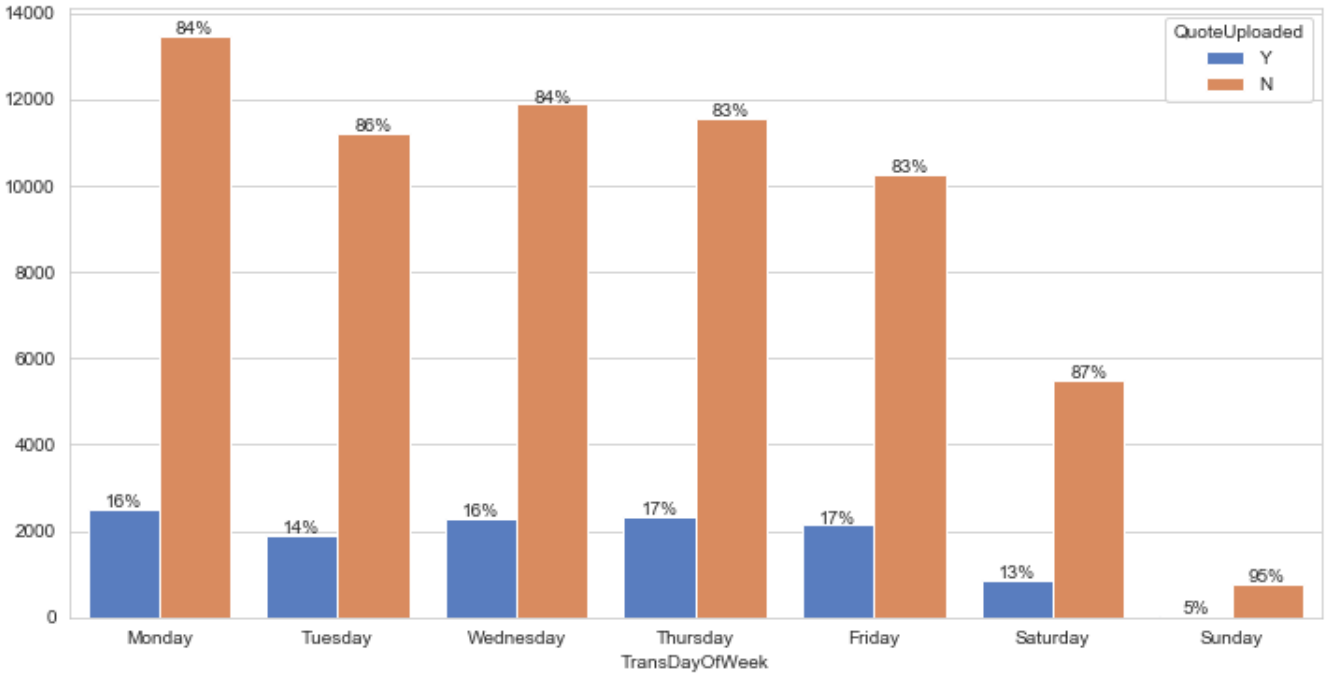

Figure 4.7: A Count Plot for the *TransDayOfWeek* Categorical Variable

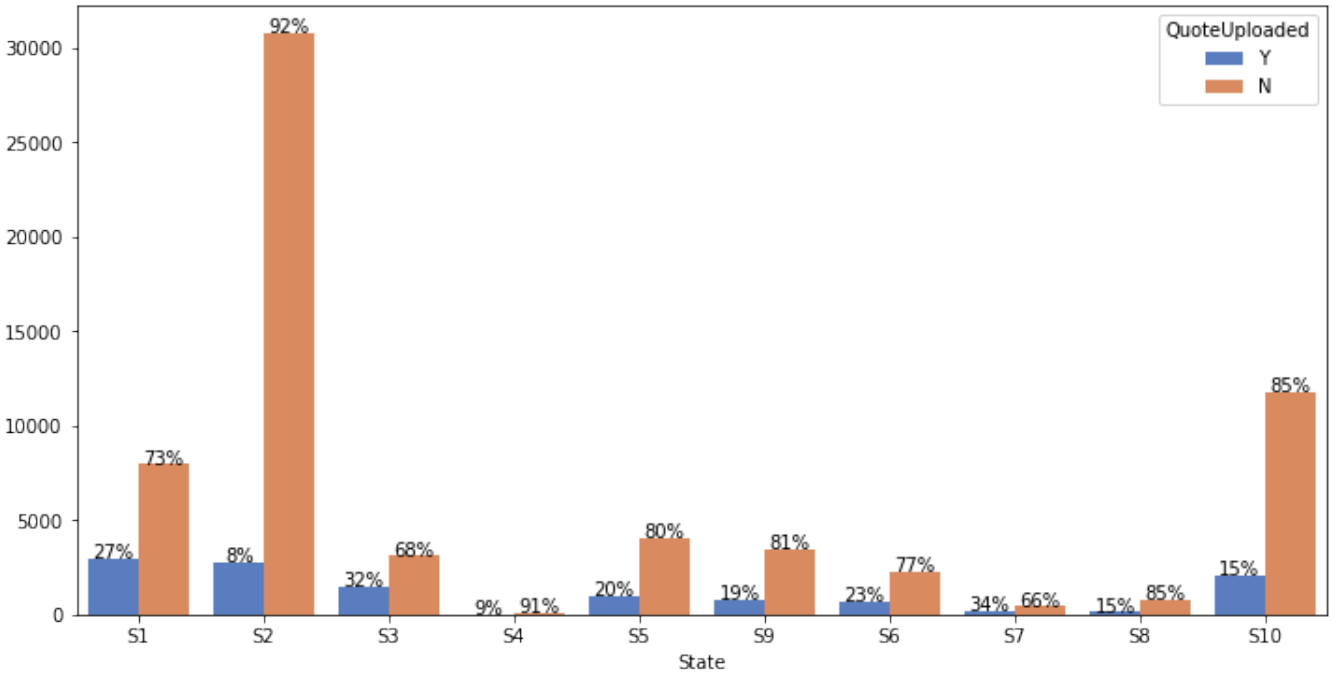

Figure 4.8: A Count Plot for the *State* Categorical Variable

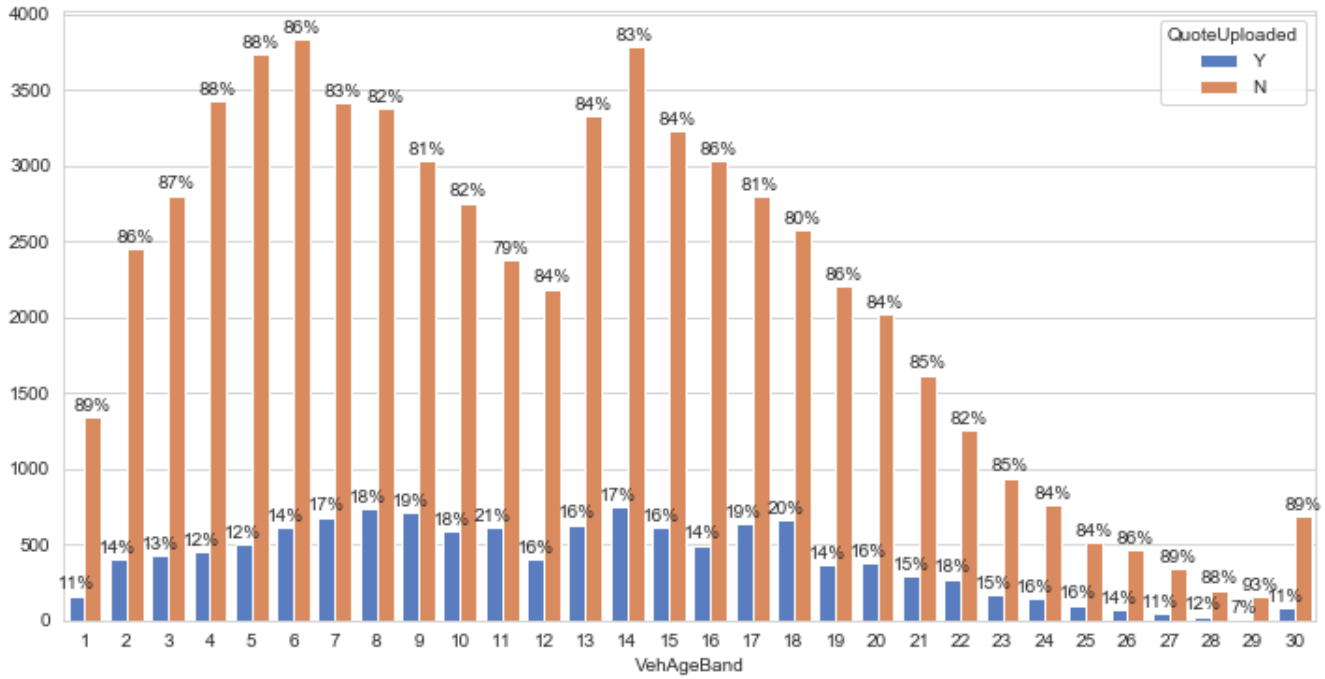

Figure 4.9: A Count Plot for the *VehAgeBand* Categorical Variable

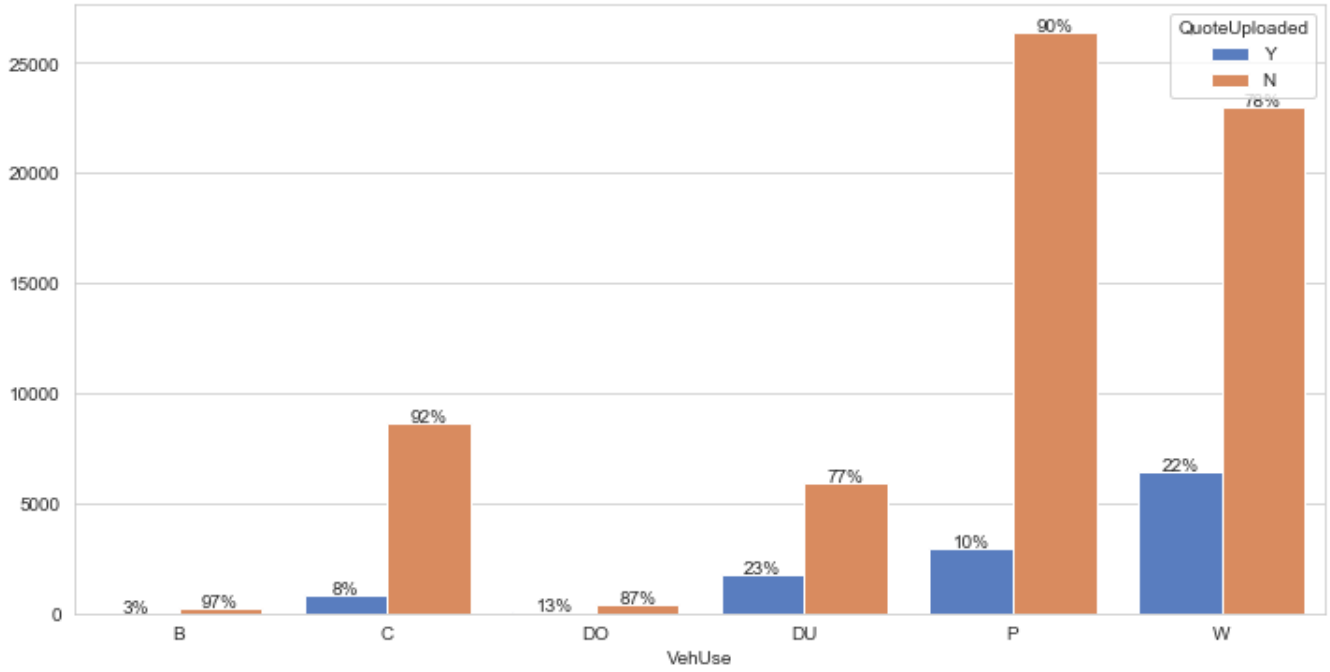

Figure 4.10: A Count Plot for the *VehUse* Categorical Variable

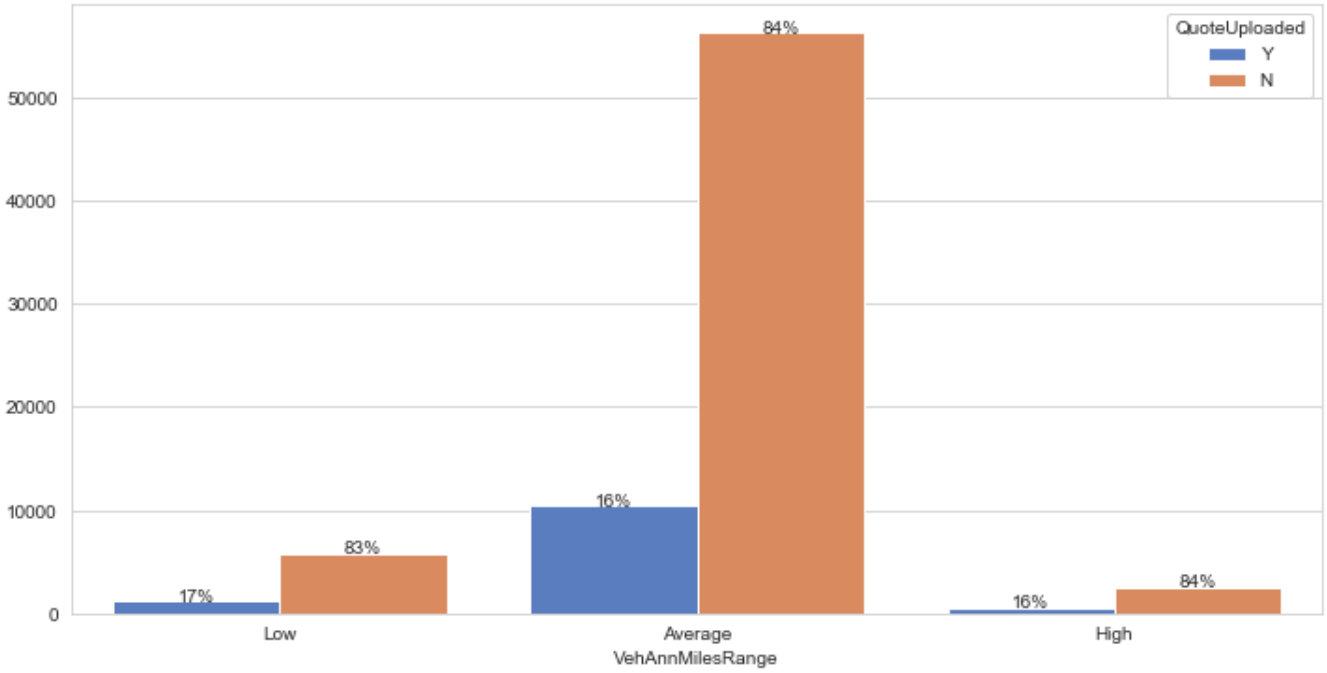

Figure 4.11: A Count Plot for the *VehAnnMilesRange* Categorical Variable

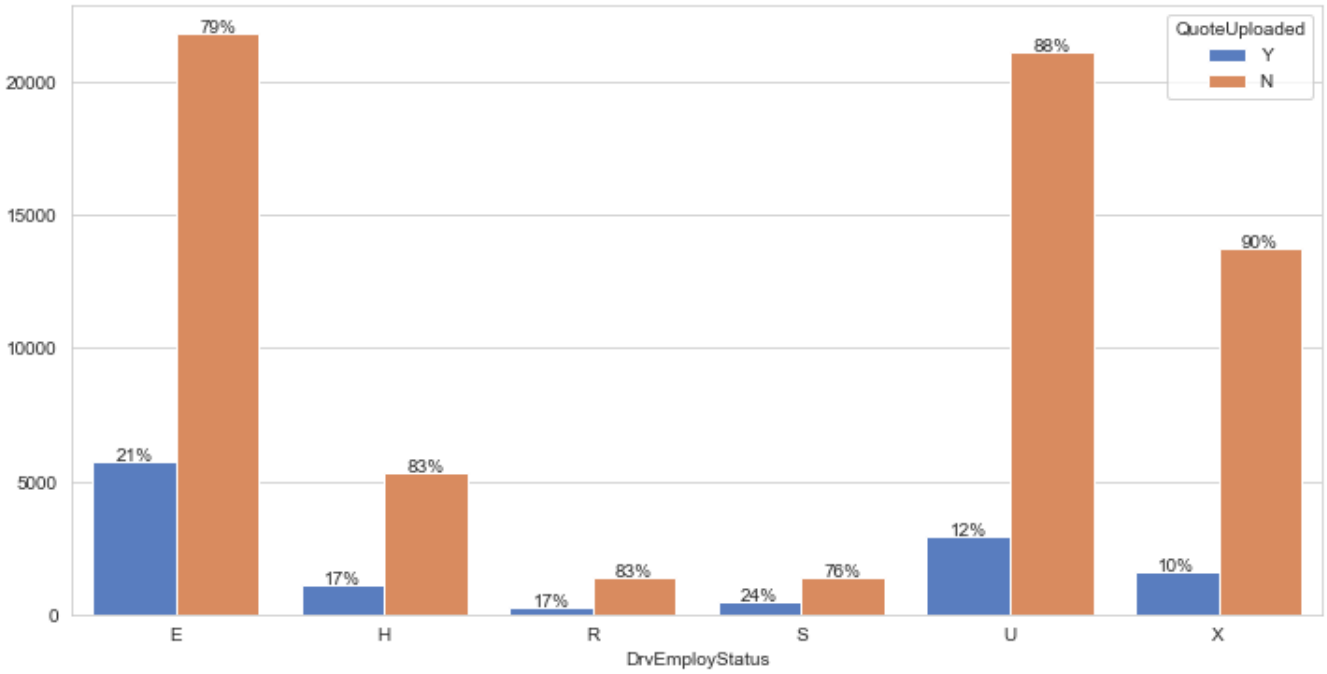

Figure 4.12: A Count Plot for the *DrvEmployStatus* Categorical Variable

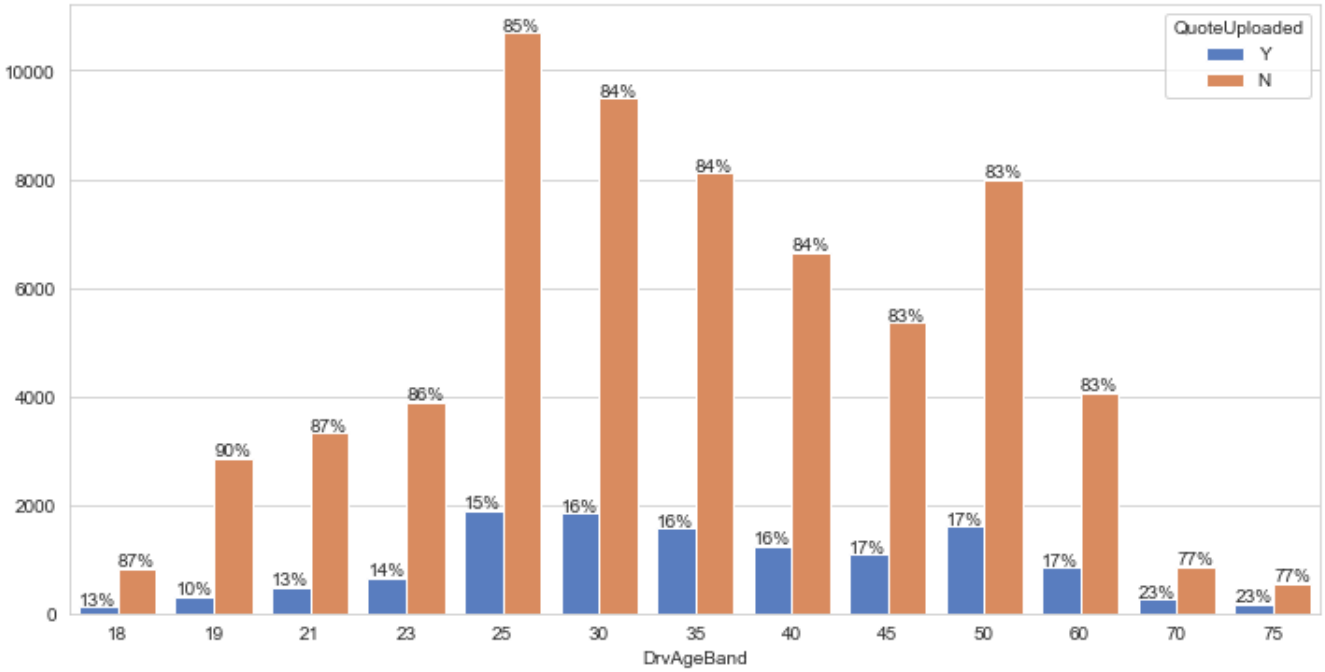

Figure 4.13: A Count Plot for the *DrvAgeBand* Categorical Variable

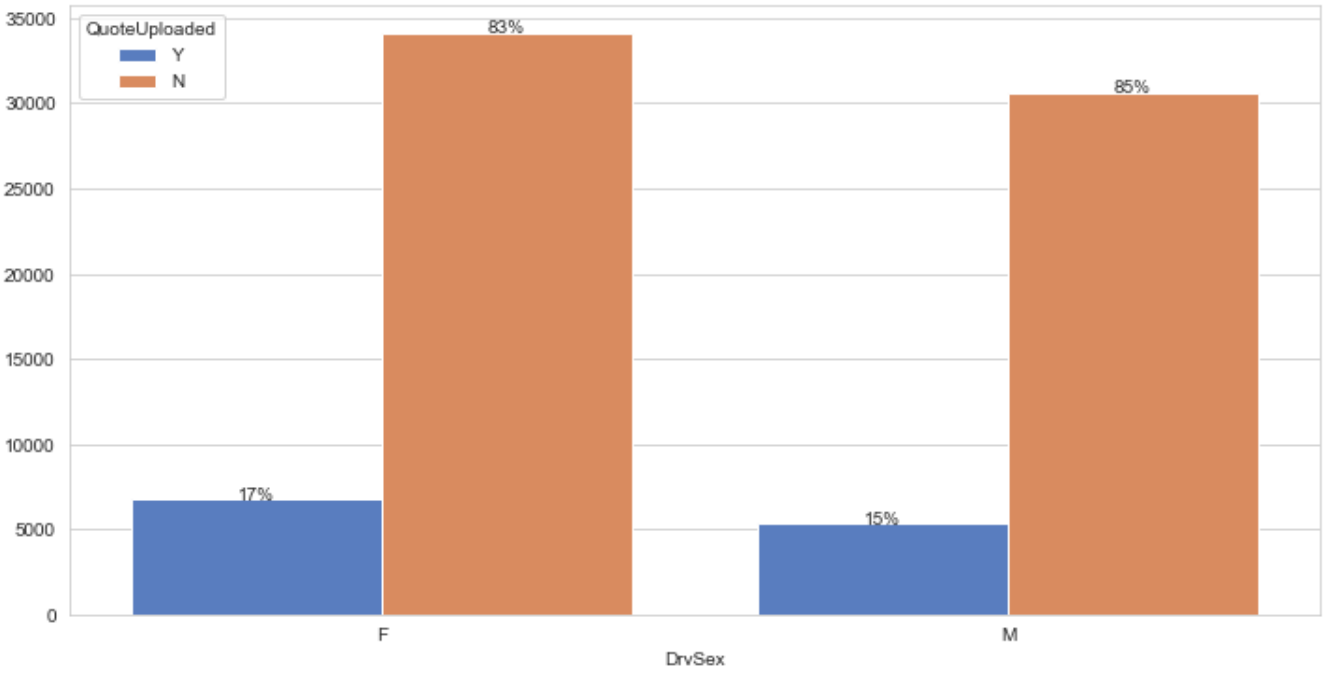

Figure 4.14: A Count Plot for the *DrvSex* Categorical Variable

- In Figure 4.6, we can see that RC1's with the *PolicyTermRTR* variable having a value of '1Y' i.e. annual term, have a proportion of 93% not uploaded to AutoUSA and only 7% uploaded.

- The *TransDayOfWeek* variable in Figure 4.7 shows a consistent split in between 83% and 87% for all days of the week except Sunday which has a 95%/5% split, but Sunday accounts for a very small proportion of the overall rows in the dataset.

- Looking at the *State* variable in the count plot in Figure 4.8, in the state 'S2', 92% are not uploaded with only 8% uploaded. The 92% accounts for approximately 30,000 observations, which is significant as it is almost 40% of the observations in the dataset. The state 'S4', has a similar high percentage of 91% not uploaded to AutoUSA, but it only accounts for a very small percentage of the overall observations, so therefore is not as significant. Note: the *State* field has been anonymised for business sensitivity reasons.

- Observing the *VehAgeBand* variable in Figure 4.9, we can see that no particular age band stands out as having a significantly different proportion to the 85/15 split of the overall dataset, except for the 29 vehicle age band which has 93% not uploaded to AutoUSA, which represents a very small number of observations.

- The *VehUse* variable in Figure 4.10 shows 90% of vehicles used for non-work purposes i.e. Pleasure, are not uploaded to AutoUSA, compared to 10% that are uploaded, the 90% accounting for 25,000 observations in the dataset. Also 92% of vehicles with *VehUse* having a value of 'C' for commercial are not uploaded to AutoUSA, representing approximately 8,000 rows in the dataset.

- The *VehAnnMilesRange* variable (Figure 4.11) follows the 85/15 split of the full dataset very closely.

- The category 'X' which stands for driver's where the employment status is unknown, has a proportion of 90% that is not uploaded to AutoUSA and can be seen in Figure 4.12.

- Observing the *DrvAgeBand* variable in Figure 4.13, 90% of drivers in the 19-20 age range are not uploaded with 10% uploaded. This accounts for a smaller proportion of the dataset i.e. approximately 3,000 observations.

- The *DrvSex* shows no significant difference between males and females (Figure 4.14).

AutoUSA should take this analysis into account when they are deciding what type of customers they are trying to attract. Any category with a significantly different breakdown to the 85/15 proportional split, should be investigated by the insurance company, i.e. 90% or over not uploaded to AutoUSA. They should specifically target the following quote characteristics: annual term, RC1 entered in the state 'S2', vehicle usage of commercial or pleasure and driver's between the age of 19 and 20.

Chapter 5

Methodology and Implementation

5.1 Research Process Flow

The diagram in Figure 5.1 shows the three main stages of this research project. The data preparation and exploration stage was described in Chapter 4. During this stage, using Python, raw XML data is transformed and merged into a complete dataset. Exploratory analysis, missing value treatment, outlier identification and data transformation work is carried out in order to better understand the data and to prepare for the modelling stage. This chapter describes in detail the second stage of the research, model development. In order to prepare the data for the modelling algorithms, feature engineering will be explored. This will involve feature reduction and the selection of the most important variables in the dataset. Sampling techniques will be explored in order to handle imbalance in the data. The various algorithms will be experimented with, selected and then optimised. The final stage of the research is model evaluation. This will be covered in detail in Chapter 6, where the results of the different models will be evaluated, compared and then interpreted to decide which performed best for this use case.

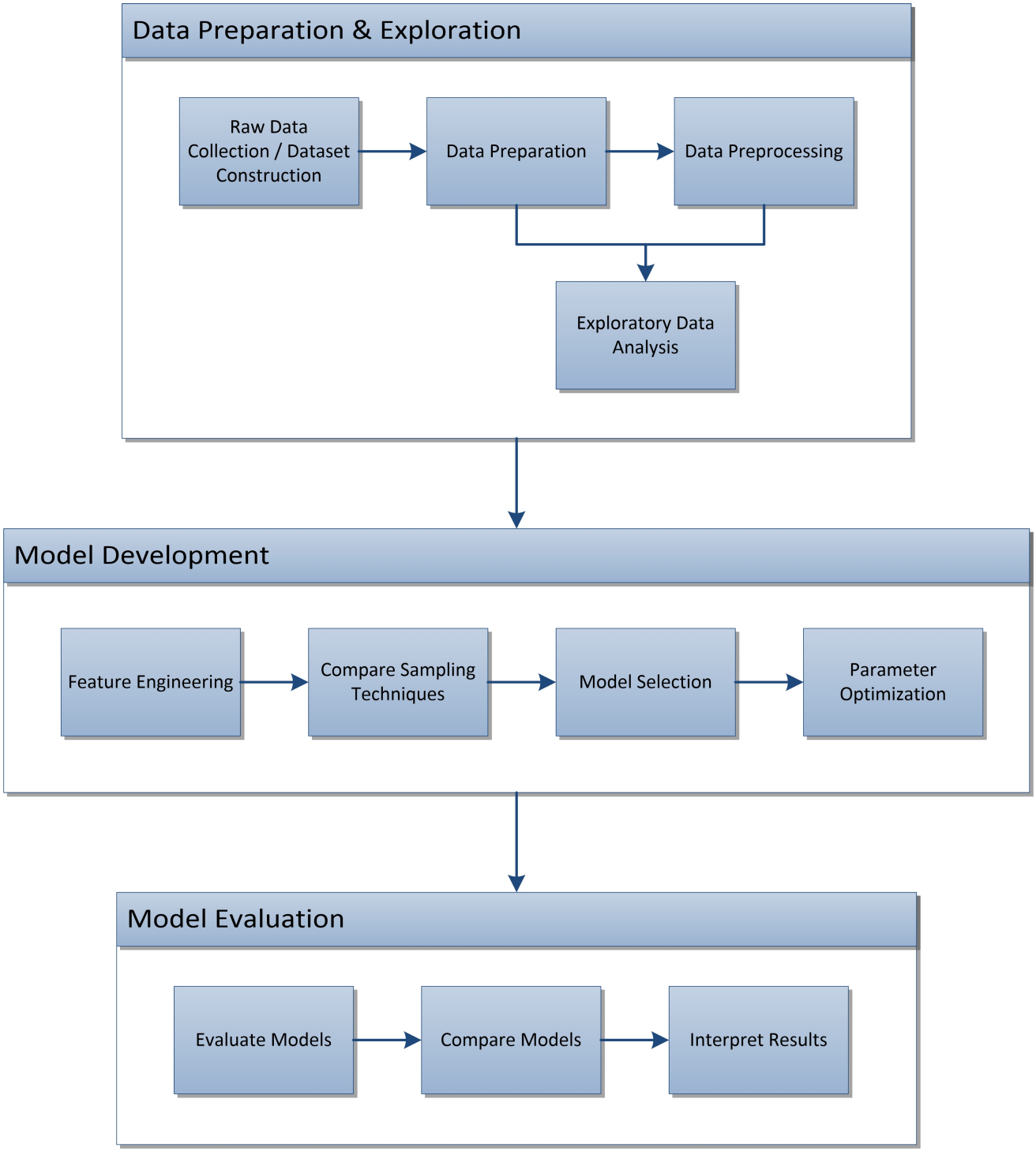

Figure 5.1: Research Process Flow

5.2 Feature Engineering and Selection

In order for machine learning algorithms to work effectively, features are required to have certain characteristics or particular relationships with each other. The preparation of the features for machine learning is referred to as feature engineering. Some of this preparation has been explained in Chapter 4 Section 4.2. This includes the handling of outliers, treatment of missing data, binning of variables using discretisation and the transformation of variables such as datetime fields to numerical fields. Other feature engineering tasks include categorical encoding techniques. Some machine learning will only accept numerical fields for example logistic regression. Therefore categorical data was converted into numbers, where each label in a category was assigned a unique integer based on alphabetical ordering. ML algorithms also require that variables are not highly correlated with each other, as correlated variables can provide the same or similar information. By removing one or more of the correlated variables, the model can be simplified and results may improve. The correlation plot in Figure 5.2 was created in order to investigate if any features are highly correlated with each other.

The correlation map shows that *CLCoverageRTR* is highly correlated (positive correlation) with *CPCoverageRTR*. On an insurance policy these coverage's are generally written together which explains the high correlation. *DrvNo* is highly correlated (positive correlation) with *VehRatedOperator*. *VehRatedOperator* is the driver number of the driver rated against the vehicle. *VehAgeBand* is highly correlated (negative correlation) with the *CPCoverageRTR* and *CLCoverageRTR* variables. Usually an insurer would require higher comprehensive and collision coverage the newer the car. Based on this analysis, *CLCoverageRTR*, *CPCoverageRTR* and *DrvNo* will be dropped from the dataset as they are correlated with other variables in the dataset.

Another approach to variable engineering, is analysing which features are the most important. This can lower the variance, prevent overfitting and simplify the model. Principal Component Analysis is a common feature reduction technique in the insurance domain as seen in the literary review for this research project. PCA selects the most important features in a dataset that captures maximum information about the dataset and converts them into principal components, which results in a reduced dataset. The original features are no longer available and the principal components are less interpretable and readable. PCA is more suitable for use with

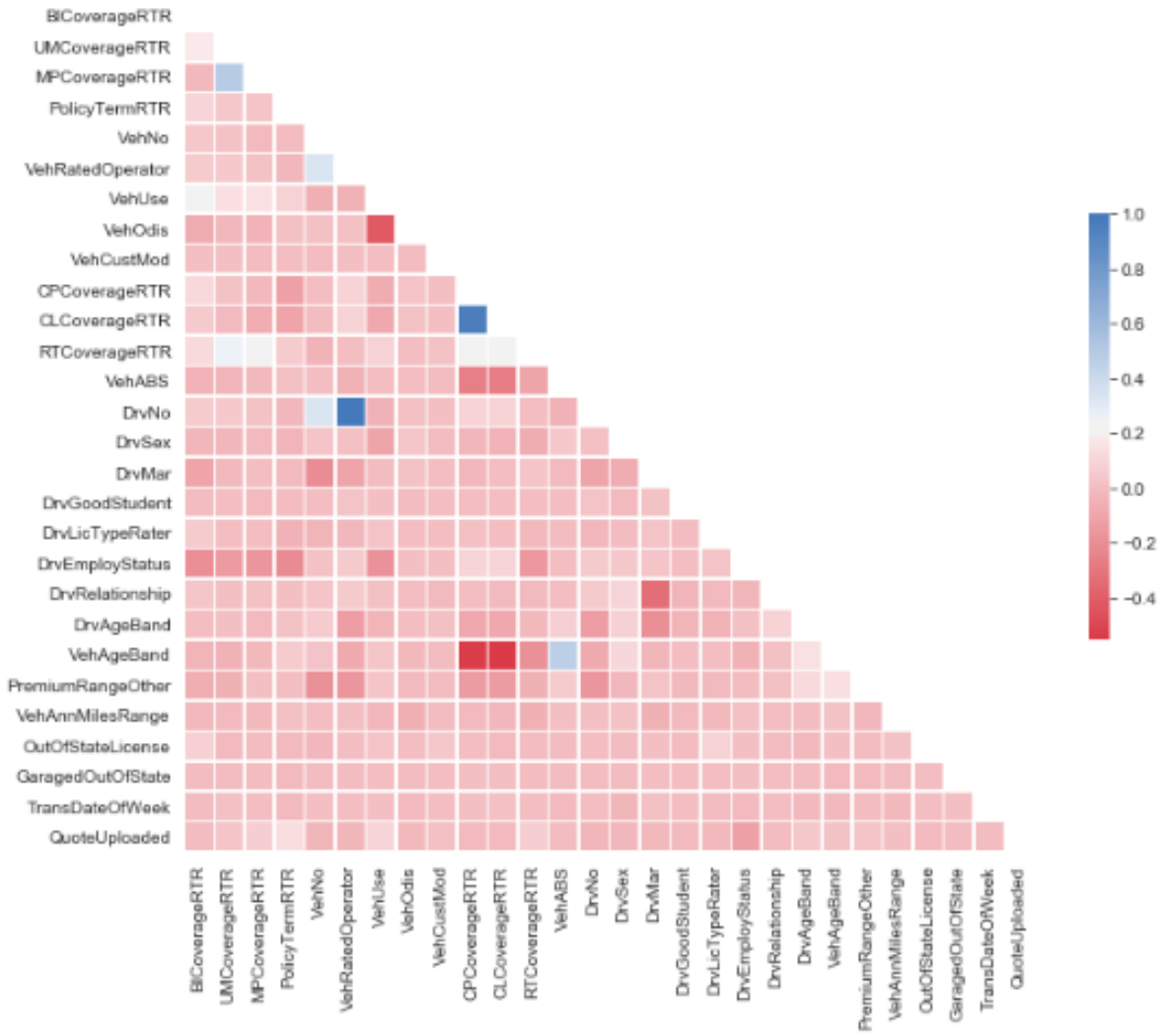

Figure 5.2: Correlation Map of Variables in Dataset

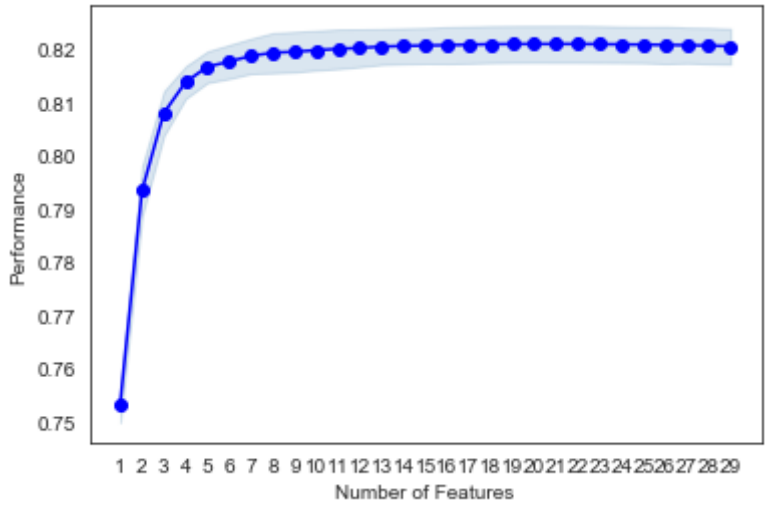

Figure 5.3: Logistic Regression Feature Selection

numerical variables and generally not recommended for categorical data which this dataset is predominately made up of. For this reason PCA was not used in this research. Instead a model forward selection process was used. Using the *SequentialFeatureSelector* greedy algorithm in the Python *mlxtend* library, we carried out feature selection for each of the classification models that will be used in this research. For each model in our analysis, features were added to the model incrementally and performance was recorded. Features were only included in the model if they added value. This also allows us to select the optimum number of features to gain the highest model performance. Default model parameters were used at this stage with 5-fold cross validation and data standardisation where appropriate. A ROC curve was created to plot the number of features against the AUC score. These ROC curves can be see in Figure 5.3, Figure 5.4, Figure 5.5 and Figure 5.6. We can see from these diagrams, the performance of the model increases dramatically after the first few features are added and then levels off for the rest of the features. Initially there are 30 features in the dataset. After this feature reduction process, the logistic regression model reduces the dataset to 21 features, 25 for random forest, 13 for K-nearest neighbour and 24 for the XGBoost Model.

5.3 Imbalanced Data and Evaluation Metric Selection

Failure to handle class imbalance in binary classification can cause poor model performance and low accuracy. Performance metric selection is crucial in working with imbalanced data.

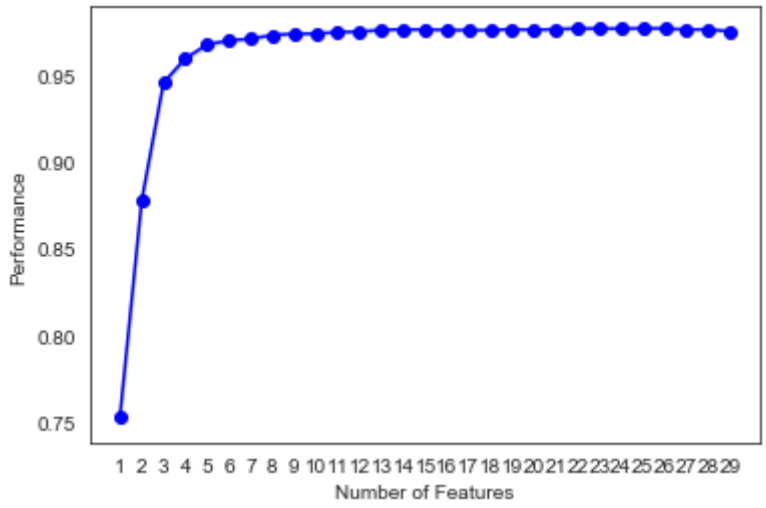

Figure 5.4: Random Forest Feature Selection

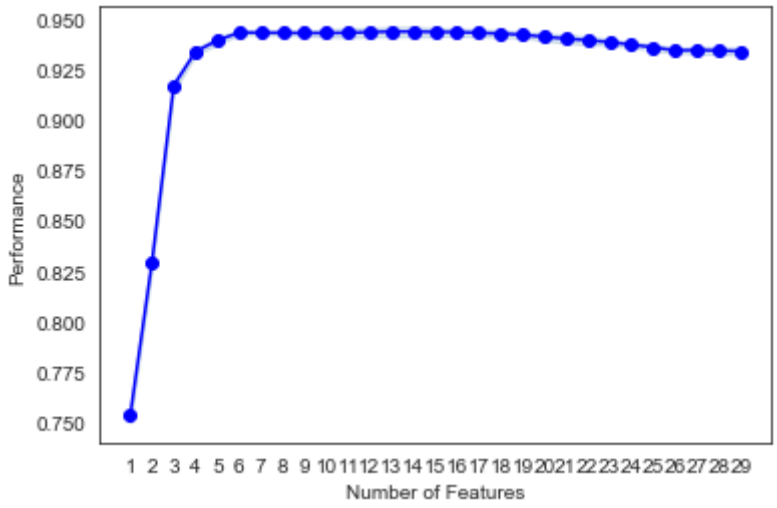

Figure 5.5: K-Nearest Neighbour Feature Selection

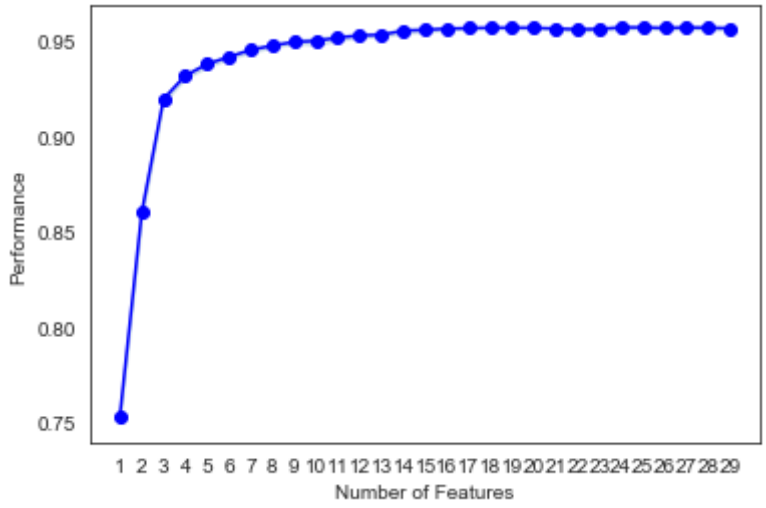

Figure 5.6: XGBoost Classifier Feature Selection

The most common performance metric in binary classification algorithm evaluation is accuracy. Accuracy is the sum of True Positives and True Negatives divided by the total population being evaluated. Therefore if all examples in the majority class were classified correctly, we could get an accuracy of approximately 85% even though no examples in the minority class were classified correctly, which shows that this metric can be misleading in imbalanced datasets. AUC scores, confusion matrices and precision and recall curves are more important when working on imbalanced datasets. Various tactics for handling imbalanced datasets will be tested to investigate which will result in the best improvement in the models performance. These include:

- Over-sampling. Randomly duplicate examples of the minority class.
- Generate synthetic samples using SMOTE.
- Adjustment of Class Weights (not applicable to all algorithms).

Under-sampling is another technique that can be used. Random Under Sampling (RUS) randomly deletes examples in the majority class until the classes are more balanced. We have not used this technique in these experiments, as important data from the majority class may be removed. For this research, we are more concerned with the majority class. The insurance company want to target the customers that are not uploading to AutoUSA currently. All these techniques were carried out on the training data on the selected algorithms using the full data set and the reduced features selected in the previous section. The AUC scoring was used to

Model	Features	# Features	Sampling	CV Score	Test Score
LR	All	30	None	0.82087471	0.751976285
LR	Reduced	21	None	0.821456318	0.751976285
LR	Reduced	21	ROS	0.821548999	0.74952427
LR	Reduced	21	Balanced Class Weight	0.821683669	0.753814625
LR	Reduced	21	SMOTE	0.821253706	0.752551849
RF	All	30	None	0.972938136	0.936364473
RF	Reduced	25	None	0.97435907	0.938515291
RF	Reduced	25	ROS	0.973071563	0.949258103
RF	Reduced	25	Balanced Class Weight	0.973071552	0.944556741
RF	Reduced	25	SMOTE	0.972442812	0.94438342
KNN	All	30	None	0.862118931	0.772483476
KNN	Reduced	13	None	0.971025534	0.772030218
KNN	Reduced	13	ROS	0.877213795	0.859618806
KNN	Reduced	13	SMOTE	0.875285458	0.850008915
XGB	All	30	None	0.954671013	0.85718656
XGB	Reduced	24	None	0.95473893	0.858596991
XGB	Reduced	24	ROS	0.95642953	0.901859997
XGB	Reduced	24	SMOTE	0.944423164	0.888764954

Table 5.1: Sampling Technique Cross Validation and Test Results per Model

carry out the comparison. The sampling techniques with the highest score on the test data will be used in the next step when optimising our model. We can see from the results in Table 5.1, Random Over Sampling (ROS) returns the best AUC score on the test data for all the algorithms except logistic regression.

Performance is significantly better in training data cross validation results for many of the models at this stage which could mean the models are overfitting. Overfitting occurs when models learn too well from the training data. They learn too much detail from the data including noise. When this happens, the model does not perform as well on unseen test data. We can see this problem occurring at this stage in the process, i.e. results on training data are better than on the test data. Overfitting can be prevented by sampling techniques. We can see from Table 5.1, the test score improves for different sampling methods and the difference between the CV score and test score is less. Another technique for preventing overfitting is using cross validation. For this process we used 5-fold cross validation. This will be improved at a later stage by using 10-fold cross validation.

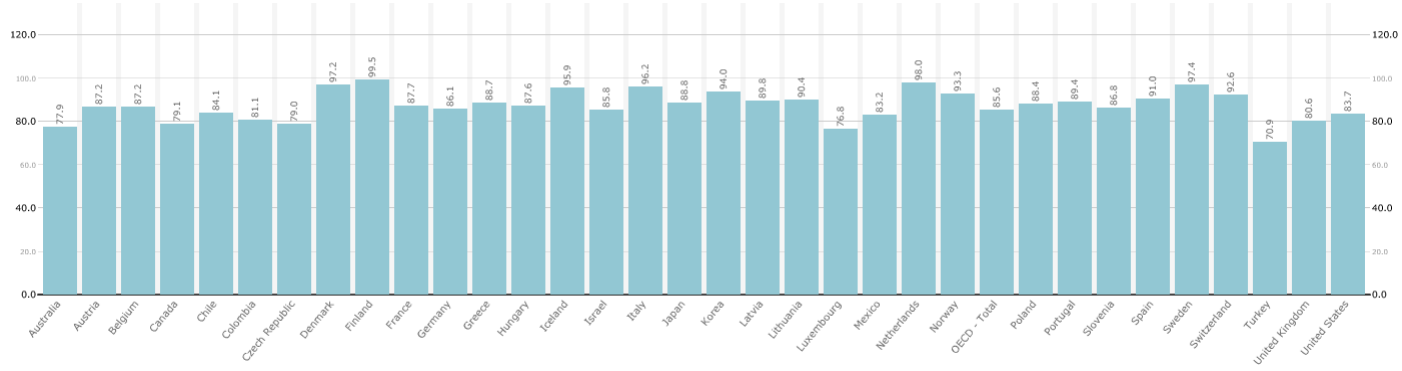

Figure 5.7: Non-life Insurance Retention Rate of OECD countries 2018

5.4 Model Selection

Churn prediction is a common use case in machine learning which exists in the insurance domain. Churn in the insurance domain refers to the loss of customers at renewal stage. It is very important for a business to try and target these customers before they churn. According to the OECD, the average retention ratio of non-life insurance amongst the OECD countries in 2018 is 86%, corresponding to a churn rate of 14% (OECD, 2018).

Therefore churn data is imbalanced. With churn prediction, the aim is to correctly classify the minority class i.e. customers that churn. Binary classification is used to try and predict the customers that churn. The use case for this research is very similar to churn prediction i.e. data is also imbalanced, except here our aim is to correctly classify the majority class, customers who are not uploading their quotes into AutoUSA. For this research we will use algorithms that have proven to perform well in churn prediction problems. Initially the following algorithms were researched and preliminary experiments were carried out on them: Logistic Regression, Random Forest, K-Nearest Neighbour and Support Vector Machine (SVM). We decided to use XGBoost instead of Support Vector Machine as SVM was too time and resource intensive. Also on an initial run of the SVM classifer very poor results were returned, i.e. AUC score of less than 40%. We selected XGBoost as it performed well in a churn prediction problem in the telecommunications domain outlined in the paper by Ahmad et al. (2019). Also XGBoost is a faster more efficient algorithm then SVM.

Model	Sampling	Best Parameters
LR	Class Weight Balanced	{'logisticregression__penalty': 'l2', 'logisticregression__C': 11.513953993264458}
RF	ROS	{'randomforestclassifier__n_estimators': 500, 'randomforestclassifier__min_samples_split': 14, 'randomforestclassifier__min_samples_leaf': 8, 'randomforestclassifier__max_features': 'auto', 'randomforestclassifier__max_depth': None, 'randomforestclassifier__criterion': 'entropy', 'randomforestclassifier__bootstrap': False}
KNN	ROS	{'kneighborsclassifier__weights': 'distance', 'kneighborsclassifier__n_neighbors': 14, 'kneighborsclassifier__leaf_size': 3, 'kneighborsclassifier__algorithm': 'kd_tree'}
XGB	ROS	{'xgbclassifier__colsample_bytree': 0.6948953189819347, 'xgbclassifier__learning_rate': 0.09113090404327123, 'xgbclassifier__max_depth': 7, 'xgbclassifier__min_child_weight': 2, 'xgbclassifier__n_estimators': 700, 'xgbclassifier__subsample': 0.5000892494264289}

Table 5.2: RandomizedSearchCV Best Parameters Results per Model

5.5 Model Optimisation

On the initial run of each of the models, default parameters were used with the full dataset. These become the base models for each algorithm to be used for comparison evaluation at a later stage in the process. Python offers two functions to carry out parameter optimisation experiments: *GridSearchCV* and *RandomizedSearchCV*. *GridSearchCV* carries out an exhaustive search of the model using a list of possible parameter options and tries every combination of these options. This can be very inefficient and time intensive when there are many parameter options and this proved to be the case on an initial run of this algorithm for the logistic regression algorithm. *RandomizedSearchCV* carries out a similar task but selects random combinations of parameters. *RandomizedSearchCV* is more efficient, can lead to good prediction results and runs in a fraction of the time *GridSearchCV* takes. This was also found to be the case in the work by Bergstra and Bengio (2012). We have used *RandomizedSearchCV* in this research, using the reduced dataset and the optimum sampling techniques as outlined in Section 5.3 above for each algorithm. Table 5.2 summarises the selected model, best sampling techniques per model and the best parameters as selected by the *RandomizedSearchCV* process.

Chapter 6

Results

In this Chapter, the predictive performance of each of the four models Logistic Regression, Random Forest, K-Nearest Neighbour and XGBoost will be evaluated by comparing the base model with their corresponding optimum model. To do this, the evaluation metrics accuracy, recall, precision and ROC AUC percentage scores are compared, for both the cross valuation mean score and the test score. The confusion matrix of the base model and optimum model are also analysed and compared. Finally, ROC curve and precision-recall curves are created for both base models and optimum models, where models can be easily visually compared and evaluated. The base models for each of the algorithms were run with the following characteristics:

- Default hyperparameters.
- Full dataset with 30 features.
- Data split into 75% training and 25% testing sets.
- 10-fold cross validation was used.
- No sampling techniques were used to handle the class imbalance.
- Scaling of data was carried out for Logistic Regression and K-Nearest Neighbour models.
- Accuracy, AUC, recall and precision metrics were recorded for both the cross validation and the test dataset. ROC curve and precision-recall curves were created.

Next optimised models were run for each of the algorithms and have the following characteristics:

- Best parameters returned from hyperparameter tuning using *RandomisedSearchCV*.
- Reduced dataset as returned from feature selection process for each algorithm.
- Sampling techniques were carried out in a pipeline.
- Data split into 75% training and 25% testing sets.
- 10-fold cross validation was used.
- The following metric scores were recorded: accuracy, AUC, recall and precision.
- ROC curve and precision-recall curves were created.

The breakdown of the results for each algorithm are outlined in the next sections.

6.1 Logistic Regression

6.1.1 Base Model

The base model was run using the logistic regression algorithm in the *sklearn* library in Python. The dataset was standardised before feeding it into the model using the *StandardScalar* function. This is a common requirement for certain machine learning algorithms. This function subtracts the mean of the variable and divides it by its standard deviation. A pipeline was created to run the scalar and model consecutively. Initially, the *liblinear* solver was selected for the model but it took too long to run. We changed to the *saga* solver as according to the sckit-learn documentation (Pedregosa et al., 2011), *saga* is a better more efficient choice for larger datasets. We set the maximum number of iterations to 1,000 as when using the default value of 100, the algorithm failed to converge.

Table 6.1 displays the results of the base logistic regression run, which includes the mean value of the 10-fold cross validation score and the test prediction scores. There is an insignificant difference between the cross validation score and the test scores except for AUC where there

is a 8% difference. This could suggest a slight overfitting of the model. This model has a high accuracy score of 92.15% and an AUC score of 75.2%. The confusion matrix in Figure 6.1 shows that the majority class was correctly predicted for all instances but only half of the minority class was classified correctly i.e. true positive rate is 100% but true negative rate is only 50%. It would be preferable if there was more of a balance between the true positive and negative rates i.e. balance between precision and recall.

Evaluation Measure	CV Mean Score	Test Score
Accuracy	92.26%	92.15%
Recall	50.70%	50.40%
Precision	100%	100%
AUC	82.11%	75.20%

Table 6.1: Base Model Results for Logistic Regression Model

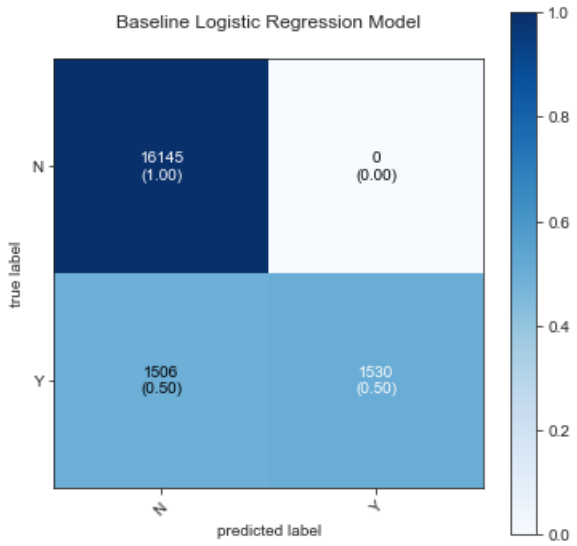

Figure 6.1: Confusion Matrix Logistic Regression Base Model

6.1.2 Optimised Model

For the optimised model, the reduced dataset of 21 features were used. The data was standardised and the balanced class weight parameter was set. The following hyperparamters were used:

- penalty = 'l2'

- C = 11.513953993264458
- solver = 'saga'
- class_weight = 'balanced'

Table 6.2 shows the results of the optimised logistic regression model. Here again there is very little difference between the CV and test scores except for AUC. The accuracy has reduced slightly to 89.1% with a similar AUC as the base model 75.38%. The recall has increased from 50.4% to 55.3% and precision has decreased from 100% to almost 70%. The confusion matrix (Figure 6.2) is showing an improvement in the classification of the minority class. True negatives have increased from 50% to 55%. Overall, there is a small improvement from the base to optimum Logistic Regression model as AUC has increased by 0.24% and there is more of a balance between the precision and recall scores.

Evaluation Measure	CV Mean Score	Test Score
Accuracy	89.18%	89.10%
Recall	55.26%	55.30%
Precision	69.58%	69.61%
AUC	82.18%	75.38%

Table 6.2: Optimised Model Results for Logistic Regression Model

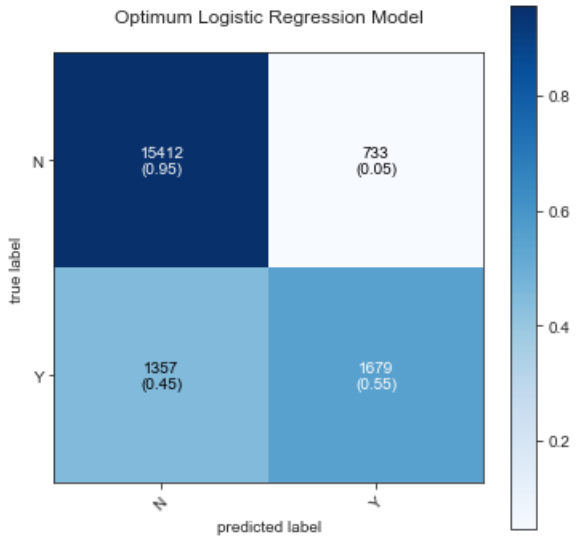

Figure 6.2: Confusion Matrix Logistic Regression Optimum Model

6.2 Random Forest Ensemble

6.2.1 Base Model

Using default parameters in the *RandomForestClassifier* function in *sklearn* library, no scaling was required, we trained the model with no sampling on the full original dataset. The results are noted in Table 6.3. Results are very good for a base model and much improved on the logistic regression results. There is insignificant differences between the CV and test scores which is a sign that the model is not overfitting. High accuracy and AUC of 97.35% and 93.65% respectively are recorded. Recall score of 88.21% and almost 95% precision score are returned which shows us that the model is able to distinguish between both classes and is classifying correctly most of the time. The confusion matrix in Figure 6.3 is showing improved precision and recall. There are only 1% false negatives i.e. quotes that were classified as uploaded but were not uploaded to AutoUSA. There were 12% false positives where customers were classified as uploaded into AutoUSA but were not uploaded.

Evaluation Measure	CV Mean Score	Test Score
Accuracy	97.05%	97.35%
Recall	85.71%	88.21%
Precision	95.03%	94.66%
AUC	97.64%	93.64%

Table 6.3: Base Model Results for Random Forest Model

6.2.2 Optimum Model

For the optimum model, a reduced dataset of 25 features, random oversampling and the following hyperparameters were used:

- n_estimators = 500
- min_samples_split = 14
- min_samples= 8
- max_features = 'auto'

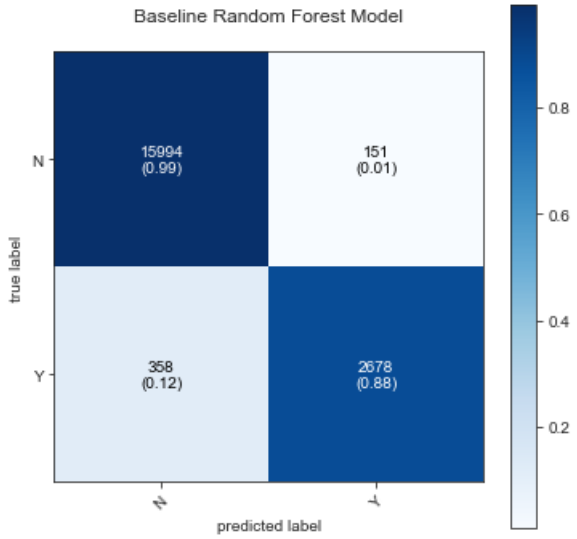

Figure 6.3: Confusion Matrix Random Forest Base Model

- max_depth = none
- criterion = 'entropy'
- bootstap = false

The results of the finely tuned random forest classifier is displayed in Table 6.4. The test score results are very similar to the test scores of the base model. There is a slight improvement on the recall from 88.21% to 89.03% and AUC score from 93.64% to 93.96% values but a slight drop in precision from 94.66% to 93.72% and accuracy 97.35% to 97.32%. The confusion matrix shows 1% false negatives and 11% false positives. For this research, we are more concerned with lowering the false negative rate. We want to predict the users that do not upload to AutoUSA. Therefore, for this model the base model would be slightly preferable to the optimised model. Random Forest is known to produce very good results even using default parameters and can produce very little variability of results after tuning according to Probst et al. (2018).

Evaluation Measure	CV Mean Score	Test Score
Accuracy	97.06%	97.32%
Recall	87.05%	89.03%
Precision	93.79%	93.72%
AUC	97.90%	93.96%

Table 6.4: Optimum Model Results for Random Forest Model

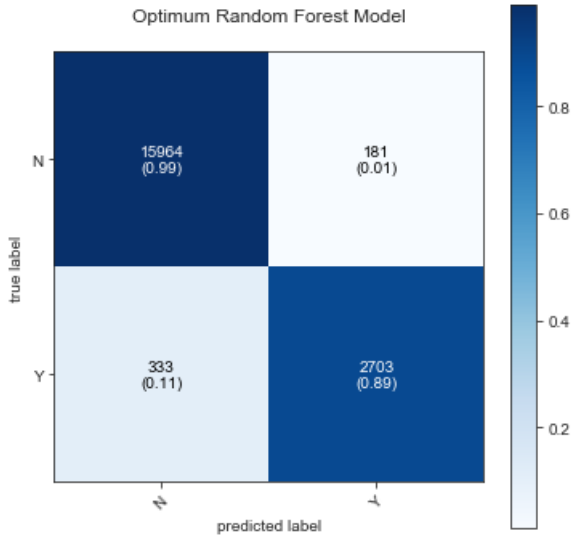

Figure 6.4: Confusion Matrix Random Forest Optimum Model

6.3 K-Nearest Neighbour

6.3.1 Base Model

The base K-Nearest Neighbour model was trained with a full dataset, no sampling and variable standardisation. Results in Table 6.5 show a good accuracy score of 94.15% but other evaluation measures of recall, precision and AUC could be improved having scores of 77.6%, 84.17% and 87.43% respectively. We can also see signs of overfitting on this model due to the AUC score having a 7% higher cross validation score over the test prediction score. The confusion matrix in Figure 6.5 shows 22% of quotes uploaded to AutoUSA and 3% of quotes not uploaded to AutoUSA being misclassified.

Evaluation Measure	CV Mean Score	Test Score
Accuracy	93.84%	94.15%
Recall	75.06%	77.60%
Precision	83.93%	84.17%
AUC	93.65%	87.43%

Table 6.5: Base Model Results for K-Nearest Neighbour

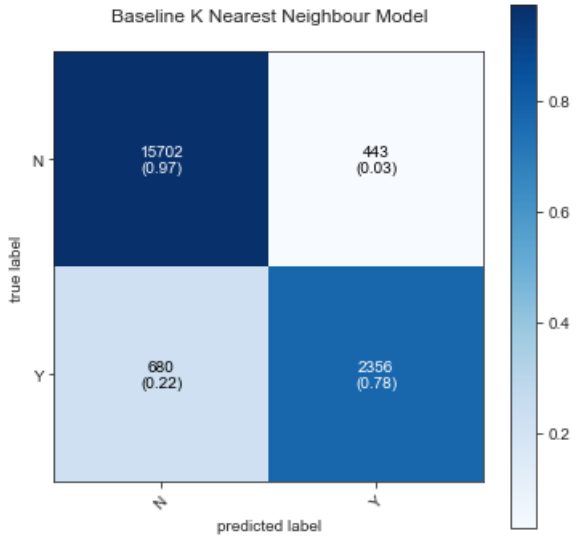

Figure 6.5: Confusion Matrix K-Nearest Neighbour Base Model

6.3.2 Optimum Model

For the K-Nearest Neighbour optimum model, a reduced dataset of 13 features, standard scaling, random oversampling and the following hyperparameters were used:

- weights = 'distance'
- n_neighbors = 14
- leaf_size = 3
- algorithm = 'kd_tree'

The results of the optimum K-Nearest Neighbour Algorithm is displayed in Table 6.6. There has been a significant increase in recall from 77.6% to 93.28%, an increase of almost 18%. The AUC score has increased from 87.43% to 93.43%. Precision has decreased from 84.17% to 73.22%, a drop of almost 14% and a slight decrease in accuracy from 94.15% to 93.54%. The confusion matrix (Figure 6.6) illustrates the significant improvement in the false positives from 22% to 7% between the base and optimum models. There is an improvement in the difference between the cross validation and test scores which shows that the issue of overfitting has been addressed in the optimum KNN model.

Evaluation Measure	CV Mean Score	Test Score
Accuracy	93.10%	93.54%
Recall	91.33%	93.28%
Precision	72.11%	73.22%
AUC	97.90%	93.43%

Table 6.6: Optimum Model Results for K-Nearest Neighbour

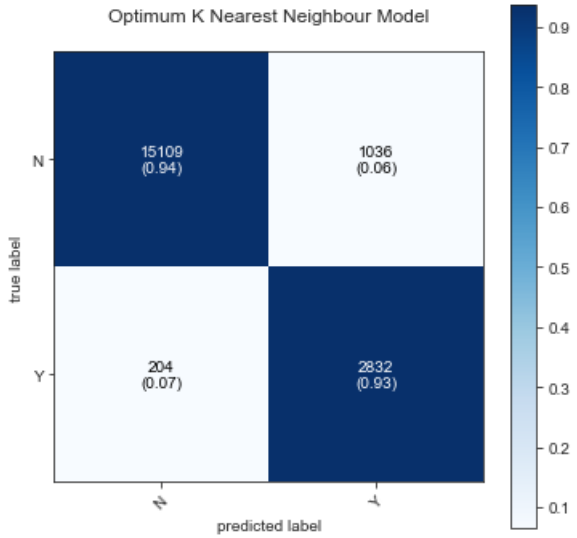

Figure 6.6: Confusion Matrix K-Nearest Neighbour Optimum Model

6.4 XGBoost Classifier

6.4.1 Base Model

The base XGBoost Classifier was run using the full dataset, no sampling technique and with no scaling required. The results in Table 6.7 show high accuracy of 95.18% and high precision score of 96.89% with lower scores for recall and AUC of 71.87% and 85.72% respectively. The cross validation and test predictions are very similar except for AUC which is reporting a 11% difference, with the cross validation having the higher score which may indicate overfitting. The confusion matrix (Figure 6.7) is showing a high false negative percentage, 28% of quote uploaded to AutoUSA are misclassified.

Evaluation Measure	CV Mean Score	Test Score
Accuracy	94.99%	95.18%
Recall	70.21%	71.87%
Precision	97.04%	96.89%
AUC	95.73%	85.72%

Table 6.7: Base Model Results for XGBoost Classifier

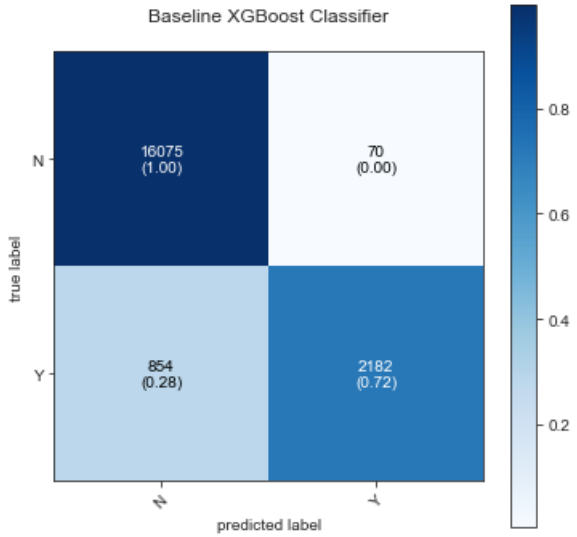

Figure 6.7: Confusion Matrix XGBoost Classifier Base Model

6.4.2 Optimum Model

The fine tuned XGBoost Model was trained using the random over sampling technique, and the following hyper parameters:

- colsample_bytree = 0.69
- learning_rate = 0.09
- max_depth = 7
- min_child_weight = 2
- n_estimators = 700
- subsample = 0.5

This optimisation resulted in an improvement in all metrics except precision. Accuracy increased from 95.18% to 96.56%, recall increased from 71.87% to 82.74% and AUC increased

from 85.72% to 90.95%. There is still a 6.5% difference between the CV score and the test score which may suggest that overfitting has not been fixed by the tuning of the model. The confusion matrix (Figure 6.8) shows decrease in the false negative with an improvement of the percentage from 28% to 17%.

Evaluation Measure	CV Mean Score	Test Score
Accuracy	96.51%	96.56%
Recall	82.00%	82.74%
Precision	95.07%	94.86%
AUC	97.09%	90.95%

Table 6.8: Optimum Model Results for XGBoost Classifier

Figure 6.8: Confusion Matrix XGBoost Classifier Optimum Model

6.5 Model Comparisons

This section analyses the results obtained for each of the machine learning algorithms selected for this research. Each model is compared using the evaluation metrics accuracy, recall, precision and AUC. A comparison will also be carried out between the base model and optimised version of each model. An experiment was carried out to find the best performing supervised learning model to predict the customers who will not upload their quote into the AutoUSA Insurance application. Accuracy has been recorded in these experiments as it is a common metric for evaluating performance. Precision, recall and AUC are more appropriate for this research

due to the imbalance in the data as they provide a more accurate picture of the performance of the different classes in our data. This is outlined in more detail in the paper by Ali et al. (2015).

It was observed that the optimised model performs better than the base model for all algorithms when comparing the AUC Score, with only very slight improvement for the logistic regression and random forest model. This can also be observed from the ROC curves in Figure 6.9 and Figure 6.10. The ROC curve plots the true positive rate against the false positive rate using different threshold values between 0 and 1. A model that displays a ROC curve closest to the top-left corner of the plot indicates the best performance. Therefore looking at the optimum ROC curve in Figure 6.10, random forest, K-nearest neighbour and XGBoost are showing lines very close together, which suggest very similar performance, with random forest having the best performance. The AUC score in the results tables is the Area Under the ROC curve, the closer this score is to 1 the better. Using this score, it can be seen that random forest is the best overall performing model with a AUC score of approximately 0.94, which is very close to 1.

The precision has decreased for all models in the optimised model but recall has improved. To evaluate how effective an algorithm is, both precision and recall need to be looked at together. When trying to maximise precision, recall will decrease, whereas trying to maximise recall, precision will decrease. For the use case in this research, precision is the more important metric. We want to minimise the misclassifications in the majority class i.e. customers that are not uploading to AutoUSA. Therefore, the base models are giving better results for this particular research problem. Looking at a precision-recall curve, the best performing algorithm will be where the line is the nearest to the top right hand corner of the graph. Random forest base model is showing as the best performing model in relation to precision as seen in Figure 6.12. Recall score of k-nearest neighbour from optimum model also performing very well with a score of 93%.

The performance results of the logistic regression model are significantly lower than the other three models. There are many potential reasons for this including:

- The representation of the features. Label encoding was used for all categorical data in the dataset. This converts categorical data to integers based on alphabetical ordering of

Evaluation Measure	LR	RF	KNN	XGB
Accuracy	92.15%	97.35%	94.15%	95.18%
Recall	50.40%	88.21%	77.60%	71.87%
Precision	100.00%	94.66%	84.17%	96.89%
AUC	75.20%	93.64%	87.43%	85.72%

Table 6.9: Comparison of Base Machine Learning Models

Evaluation Measure	LR	RF	KNN	XGB
Accuracy	89.10%	97.32%	93.54%	96.56%
Recall	55.30%	89.03%	93.28%	82.74%
Precision	69.61%	93.72%	73.22%	94.86%
AUC	75.38%	93.96%	93.43%	90.95%

Table 6.10: Comparison of Optimum Machine Learning Models

the category. This may not be appropriate for some of the features for example *County* and *State* fields. It may be more appropriate to use one-hot encoding for certain fields. With this type of encoding a new feature is created for each unique value in a category having a value of 0 or 1.

- Logistic regression assumes a linear relationship between the input variables with the output. This may not be the case for some of the input variables. More investigation would be required and input variables may need to be transformed to obtain this linear relationship.

- Logistic regression is sensitive to outliers which some of the features still contain.

The ensemble models random forest and XGBoost have performed the strongest on this dataset. Grouping together machine learning models creates more robust accurate models compared to the base models on their own. They help reduce overfitting by creating a more generalised model. K-nearest neighbour is a good alternative to the ensembles having very similar results but with reduced precision.

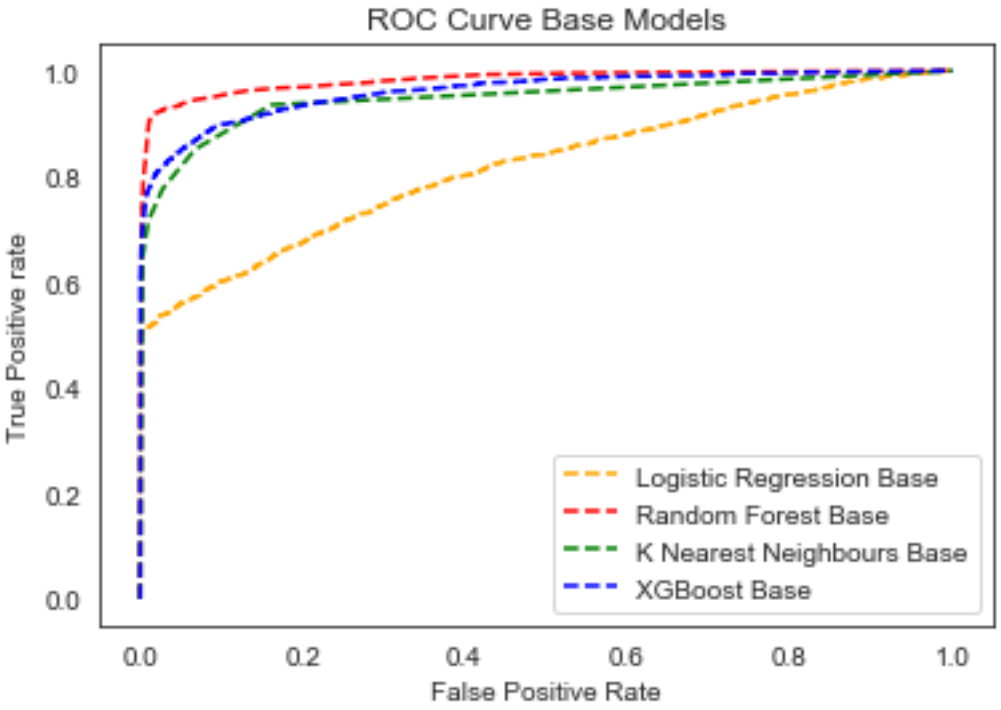

Figure 6.9: ROC Curve Base Models

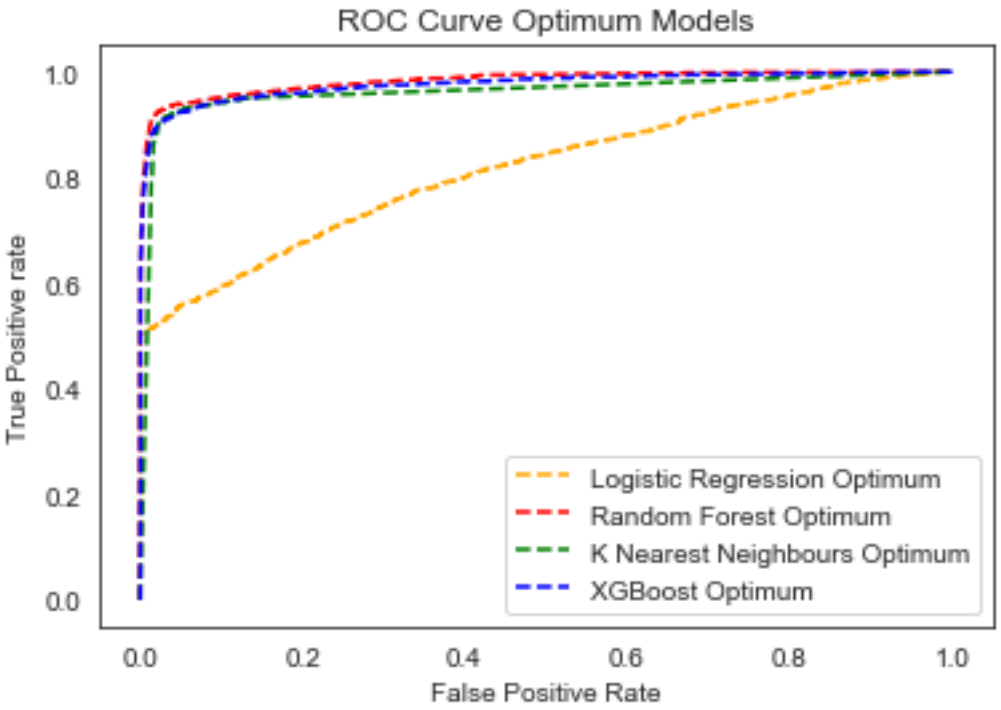

Figure 6.10: ROC Curve Optimum Models

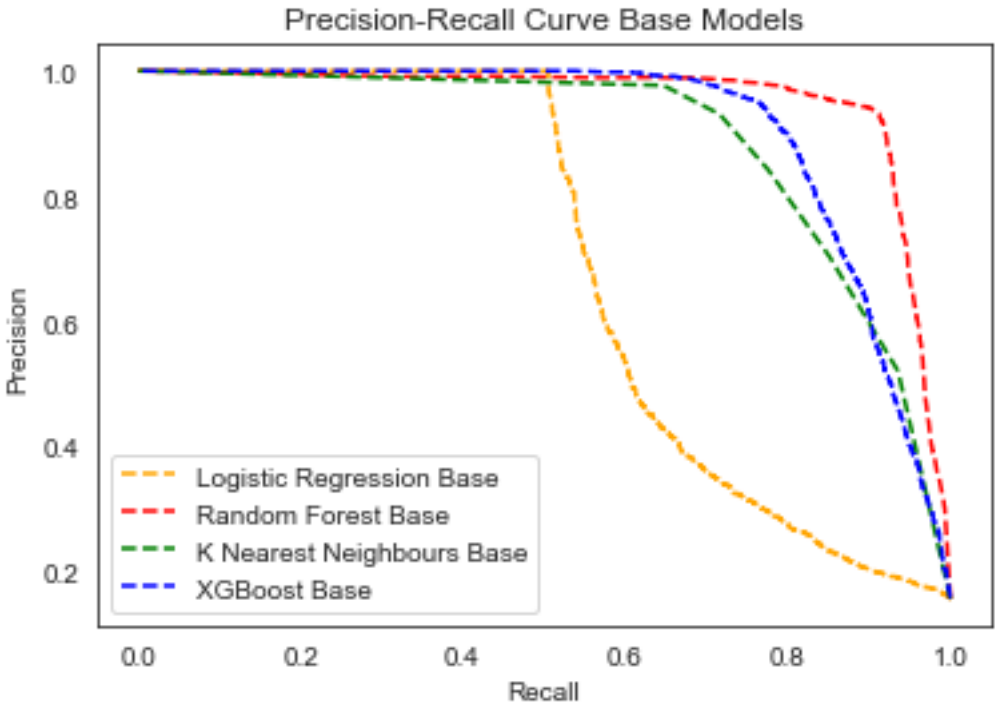

Figure 6.11: Precision-Recall Curve Base Models

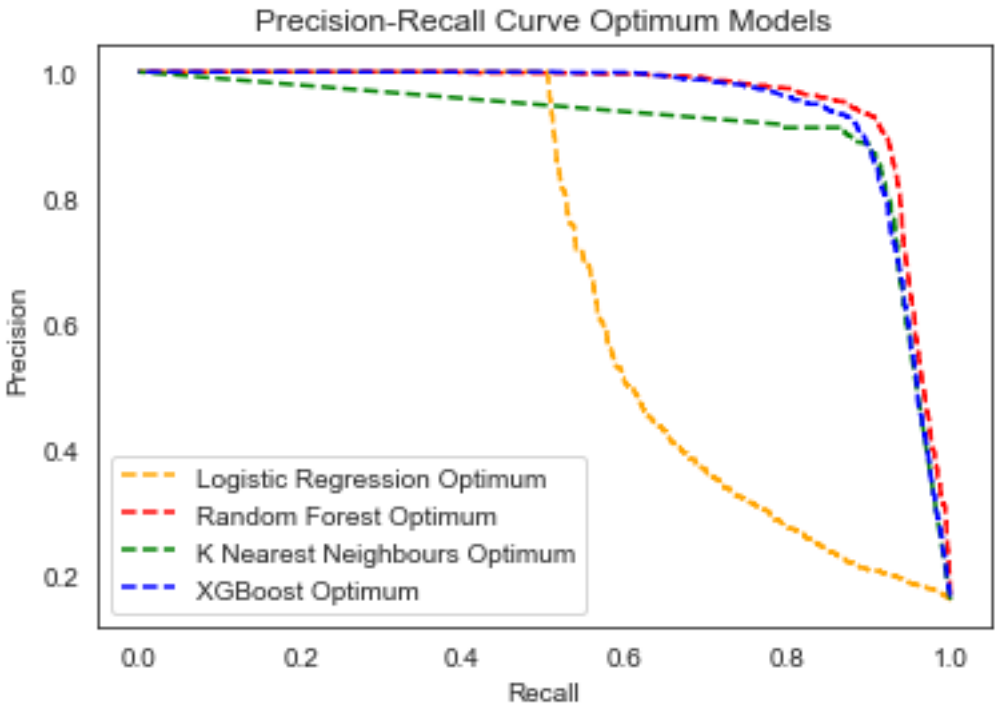

Figure 6.12: Precision-Recall Curve Optimum Models

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The primary aim of this research was to compare binary classification machine learning models on historical real-life data in order to predict whether insurance customers would upload a new business car insurance quote into a particular companies application that would then lead them to purchase the policy with the company. The process involved merging data from various data sources, cleaning the data, selecting the most important features, dealing with imbalance in the data and then fine tuning models in order to achieve the best results possible. In one of the main papers reviewed as part of this project, logistic regression was used to perform churn prediction on French car insurance data. (Aleandri, 2019). Using a stepwise selection process beforehand, logistic regression returned an AUC of approximately 66.4%. This value was slightly improved to 67.6% when an unsupervised k-means clustering technique was carried out beforehand. The use case for this research is similar to a churn prediction problem. When using logistic regression with feature selection and sampling techniques on AutoUSA's dataset, an AUC score of 75.4% was attained. This was the lowest performing of the four models in this research. The best performing model was random forest ensemble model, returning an AUC of almost 94% which was an improvement of almost 23% on logistic regression. Random forest has proven to be a robust, high preforming machine learning model for binary classification tasks, that is suitable for use on imbalanced insurance data.

7.2 Future Work

There are two main areas where future work would be of benefit to this research and use case.

7.2.1 Explore other machine learning techniques

Four machine learning techniques were used in this project on the RC1 insurance dataset. Other algorithms could be explored using this dataset. Neural networks are mentioned in the literature as a possible technique that could be used on insurance data. Parodi (2009) mentions using them for benchmarking against other models or EDA but discounts them for rate-making or pricing due to their lack of interpretability and transparency. Styrud (2017) experiments with neural networks on a small non-real world dataset with limited tuning for insurance rate making. As this use case is concerned with customer management, neural networks could be used and compared with the other models as interpretability is not as critical as working on rate making where transparency would be very important. Clustering and market basket analysis could be examined on this dataset to determine if these techniques can improve the performance of the modelling. This is something that was explored in other research and favourable results were attained.

7.2.2 Improve the collection of data

From the business perspective, improvements could be made on how the data is collected by AutoUSA. Instead of the data being logged to XML files, the data could be saved to a database. Personal data would not need to be recorded, only detailed information on the drivers and vehicles that are being quoted. Over 12,000 rows of data resulted in a failure and the quote was not rated with AutoUSA. These should be investigated to make sure they are legitimately business that AutoUSA does not want to attract. Real-time analysis with detailed results displayed on a dashboard would provide AutoUSA with valuable insights into the potential business they could be targeting. These insights could be communicated with the actuaries in the company who carry out the pricing strategies for the company. Only one weeks data was analysed for this research. It would be more beneficial to have a years worth of data, so that

time-series analysis could be carried out. Similar data mining and machine learning techniques could be carried out on the data that is uploaded into AutoUSA in order to gain insights into the characteristics of the quotes that do not become policies, so that AutoUSA could contact the customers before the quote is abandoned to prevent loss of business.

7.3 Final Thoughts

For this research we analysed RC1 data that was passed between a 3rd party rater and an insurance company AutoUSA. Only 2.5% of these RC1 quotes end up as a policy with AutoUSA. Therefore if the insurance company was able to target even a fraction of the 97.5% of RC1 quotes that did not end up as policies with AutoUSA, they could greatly increase their new business percentage. This research therefore could prove very valuable to AutoUSA's future.

Data Analytics can provide valuable insight into a companies' customers and can allow organisations to make better business decisions faster and more effectively. Car insurance while mandatory is extremely competitive. Insurance companies who take full advantage of the potential of big data and machine learning techniques, can have the competitive edge over their competitors.

Appendix A

Appendix

A.1 Code

The code that has been written for this project is located in the following github repository: https://github.com/martinaraftery/MscDataAnalyticsProject

A.2 Description of Dataset Features

Table A.1 provides a description of the final dataset that was created for this project.

Field Name	Description
BICoverageRTR	Bodily Injury Coverage (policy level coverage)
County	County in the US where the insured resides
DrvAgeBand	Age band of the driver
DrvEmployStatus	Driver employment status
DrvGoodStudent	Does the driver qualify for good student discount?
DrvLicTypeRater	Driver license type
DrvMar	Driver marital status
DrvRelationship	Driver's relationship with the named insured
DrvSex	Driver gender
DrvSWLicStatus	Driver license status
GaragedOutOfState	Is the vehicle garaged out of state?
MPCoverageRTR	Medical Payment Coverage (policy level coverage)
OutOfStateLicense	Does the driver have an out of state license?
PolicyTermRTR	Policy Term e.g. 1 month, 6 month, 1 year
PremiumRange	Premium range - low, average or high
QuoteBound	Was the quote bound i.e. did it become a policy?
QuoteUploaded	Was the quote uploaded into AutoUSA?
RTCoverageRTR	Rental Coverage (vehicle level coverage)
State	State in the US where the insured creates the RC1 quote.
TransDayOfWeek	Day of the week the transaction occurred
UMCoverageRTR	Uninsured Motorist Bodily Injury Coverage (policy level coverage)
VehABS	Vehicle anti-lock breaking system
VehAgeBand	Age band of the vehicle
VehAnnMilesRange	Vehicle annual mileage range - low, average or high
VehCustMod	Vehicle has customisations or modifications
VehMake	Make of the vehicle
VehModel	Model of the vehicle
VehNo	ID number of vehicle
VehRatedOperator	The number of the driver that vehicle is rated against
VehUse	Use of vehicle

Table A.1: Description of Variables in the Final Dataset

References

Abdelrahim Kasem Ahmad, Assef Jafar, and Kadan Aljoumaa. Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, 6, 2019. 45

Marco Aleandri. Data science in insurance: Some introductory case studies. *Institute and Faculty of Actuaries*, 2019. 5, 6, 7

Aida Ali, Siti Mariyam Shamsuddin, and Anca Ralescu. Classification with class imbalance problem: A review. 7:176–204, 01 2015. 58

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 2012. 46

Afroz Chakure. Random forest classification. URL `https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0`. vii, 14

P. Christensson. Gigo definition, 2015. URL `https://techterms.com/definition/gigo`. 24

Maithili Joshi. A comparison between linear and logistic regression. URL `https://medium.com/@maithilijoshi6/a-comparison-between-linear-and-logistic-regression-8aea40867e2d`. vii, 13

Ajitesh Kumar. Supervised vs unsupervised machine learning problems. URL `https://vitalflux.com/dummies-notes-supervised-vs-unsupervised-learning/`. vii, 12

Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. ISBN 0070428077. 7

Avinash Navlani. Knn classification using scikit-learn. URL `https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn`. vii, 15

OECD. Insurance indicators dataset : Retention ratio, 2018. URL `https://stats.oecd.org/`. 45

Pietro Parodi. Computational intelligence techniques for general insurance. *Accepted SA0 research dissertations, Institute and Faculty of Actuaries (IFoA)*, 2009. 5, 8, 9

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 48

Philipp Probst, Bernd Bischl, and Anne-Laure Boulesteix. Tunability: Importance of hyperparameters of machine learning algorithms. *arXiv:1802.09596v3 [stat.ML]*, 2018. 52

Abizer Rangwala, Andrew Starrs, and Emmanuel Viale. Accenture technology vision for insurance 2020. URL `https://www.accenture.com/th-en/insights/insurance/technology-vision-insurance-2020`. 5

Marzieh Vahidi Roodpishi and Reza Aghajan Nashtaei. Market basket analysis in insurance industry. *Management Science Letters*, 2015. 5, 6

M.M. Segovia-Gonzalez, F.M. Guerrero, and P.Herranz. Explaining functional principal component analysis to actuarial science with an example on vehicle insurance. *Insurance:Mathematics and Economics*, 45:278–285, 2009. 7

Lovisa Styrud. Risk premium prediction of car damage insurance using artificial neural networks and generalized linear models. *KTH Royal Institute of Technology*, 2017. 9

I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition.* The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2005. 6

Mario V. Wuethrich and Christoph Buser. Data analytics for non-life insurance pricing. *Swiss Finance Institute Research Paper No. 16-68*, 2019. 5, 8