Customer Churn Prediction: A Machine Learning Approach

Paul Kennedy

MSc in Data Science
The University of Bath
2021/2022

Customer Churn Prediction: A Machine Learning Approach

Submitted by: Paul Kennedy

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Master of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

With the field of data science continuing to grow, the opportunity to convert raw data into relevant actionable knowledge is becoming ever more present. Businesses in all fields are working hard and investing time and resources into analysing their data. Subscription based E-Commerce businesses are no different. Two of the most topical and difficult questions that companies in this sector are attempting to answer are 'what factors make a customer cancel?', and 'how can we predict when a customer is likely to cancel?'. Using data from a Men's Health E-commerce subscription based company, I will analyse some of the best feature selection methods for identifying the factors that indicate a customer's propensity to cancel. I will then outline the process of how I built different machine learning models that can accurately predict customers who are likely to churn in this context, and I will compare their performance.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to extend huge thanks to the following people: My supervisor Dr Alan Hayes for his help and willingness to answer my queries throughout the project. My family for their constant support, ensuring that during this project I didn't have to worry about anything else. My girlfriend for her encouragement and help proof reading the paper. The team from the E-commerce company who supplied the data. Finally, Dr Alessio Guglielmi and the other University of Bath staff for providing helpful resources to guide the project writing.

Chapter 1

Introduction

With the field of data science continuing to grow, the opportunity to convert raw data/information into relevant actionable knowledge is becoming ever more present. Businesses in all fields are working hard and investing time and resources into analysing their data, and subscription based E-Commerce businesses are no different Akter and Wamba (2016). Two of the most topical and difficult questions that companies in this sector are attempting to answer are 'what factors make a customer cancel?', and 'how can we predict when a customer is likely to cancel?'. Using data from a Men's Health E-commerce subscription based company, I will analyse some of the best feature selection methods for identifying the factors that indicate an E-commerce customer's propensity to cancel. I will then outline the process of how I built different machine learning models that can accurately classify customers who are likely to churn in this context, and I will compare their performance.

Customers, for obvious reasons, are a central part of any business. For some businesses simply attaining customers is enough. However, for subscription based businesses just attaining customers does not suffice. Customer retention is key for these businesses if they are going to make profits and succeed in the long term. It has been well documented in research that, for the average company, an increase in the customer retention rate by as little as 5% can result in an increase in revenue of at least 25% Singh and Khan (2012). Customer retention rates can be effectively increased through identifying factors that influence customer churn and changing these factors, such as pricing Hamilton, Rust and Dev (2017). Another method of increasing customer retention is through offering incentives, or simply reaching out to customers who are likely to cancel in the near future Ben Rhouma and Zaccour (2018). In this paper effective machine learning methods to help businesses with both identifying these factors, and classifying customers who are likely to cancel will be studied. These methods should ease the process of increasing customer retention rates for businesses that can effectively implement them.

I will be putting an emphasis on feature selection, with the objective of giving the reader a good idea of how to select a refined feature set to be used to train a model that will generalise well to unseen data. From the research that I have conducted, there is an indication that feature selection has often been neglected in research papers in the area.

The performance of a number of the most popular machine learning methods in the area of customer churn prediction will be compared in this paper, and the best model in this context will be identified. A comprehensive comparison of this nature has not yet been done in a research paper for the E-commerce industry, so the work will be novel work, adding well to the

existing literature in the area.

Over the course of the early stages of this study, I worked closely with a growing E-commerce business to extract and process their raw, anonymised, data in to a usable format for my analysis. Once this preprocessing was completed, I was able to begin my exploratory analysis on the data to get a better understanding of the features, before embarking on the main parts of the work.

The core idea is that outlining the process of building the machine learning models used for churn prediction in this study will give the reader the required framework to be able to build accurate churn prediction models of their own, and give businesses the opportunity to retain customers through the use of these methods. It will also give the reader an interesting look at the different factors that can cause a customer to cancel their subscription, and how to use machine learning to convert raw customer information into predictions about customer behaviour.

Given time constraints, this is as far as I will cover in this study. However, in the months following the end of the study, I will be planning to put the best performing machine learning model to use in an automated system for identifying customers who are likely to churn in the E-commerce company. As mentioned already, this could provide huge value for the company, as once these customers are identified there are proven methods to retain them through different incentives, emails and so on, with the aim of proving the company's loyalty to them Almohaimmeed (2019).

Ultimately, the system that will be the outcome of this study will run internally for the company, taking active customer data points as input and outputting the customers who are likely to churn.

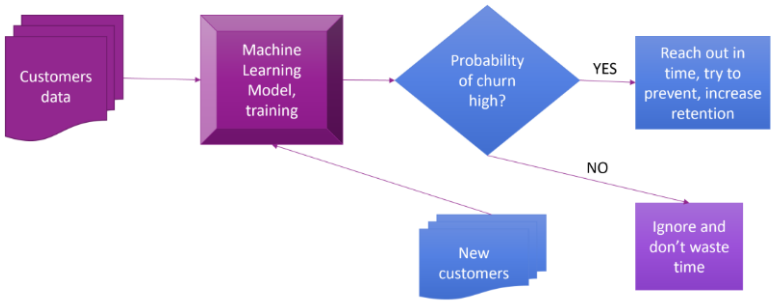

Figure 1.1: Block Diagram of System

The flow of the final system, which will use the best performing models from this study can be seen in Figure 1.1.

The paper will be broken into seven chapters following this introductory chapter. The next chapter, chapter 2, will be the literature review, covering similar studies done on churn prediction spanning different industries. chapter 3 will cover ethics in the context of this study. chapter 4 will give descriptions of all methods used in the study. chapter 5 will give an idea of the steps taken to utilise these methods. chapter 6 will cover results, with some comparison and analysis. chapter 7 will include a discussion of the study, with an attempt at critiquing the work. Finally, chapter 8 will give the conclusions of the paper.

Chapter 2

Literature and Technology Survey

In this chapter the reader will be given an insight into the research that has been completed in the area of customer churn prediction.

Initially, a definition of what customer churn prediction is in the context of a business will be given. This will be followed by brief sections on the research work done in the telecommunications and banking industries, with the concluding section focusing on customer churn prediction research specifically for E-commerce.

2.1 Customer Churn Prediction

Customer churn refers to the act of the customer ending their relationship with the company, often to bring their business to a competitor Dahiya and Bhatia (2015).

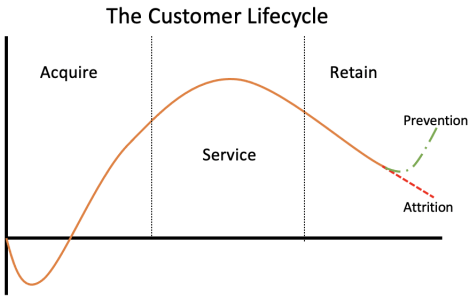

Figure 2.1: Customer Life Cycle

There are a number of stages in the lifetime of a customer as seen above, and if the stage of customer churn, (attrition in the above diagram), arrives too early, it can be detrimental for a company.

Customer churn prediction research has been conducted in many industries. I have found that the most prominent industries involved are Telecommunications and Banking, with research in the area of E-Commerce Customer Churn beginning to grow. The machine learning methods used for prediction in all industries remain reasonably consistent.

2.1.1 Telecommunications

As stated, the telecommunications industry has had consistent research in this area since the early 2000's.

In 2013, a decade review of techniques used for predicting customer churn in telecommunications was conducted Hashmi, Butt and Iqbal (2013). They found that the methods most frequently used were Decision Trees, Neural Networks, Logistic Regression and Clustering. This aligns with other findings during this period. Disappointingly, this paper doesn't include a summary of the performances of the models used.

A paper from 2012 introduced a number of new variables that could be used to predict customer churn in the telecom industry Huang, Kechadi and Buckley (2012). More interestingly, the methods used were Logistic Regression, Naive Bayes, Decision Trees, Neural Networks, Support Vector Machines and the Evolutionary Data Mining Algorithm. The authors found decision trees and SVMs to be useful for classifying churn and logistic regression worked well for estimating churn probability. Some critiques that I would have would be that the authors didn't refer to dealing with imbalanced data and they didn't indicate any use of feature selection methods.

In a 2015 paper comparing machine learning techniques for predicting customer churn in the telecommunications industry, Artificial Neural Networks, Decision Trees/Random Forests, Logistic Regression, Support Vector Machines and Naive Bayes were identified as some of the most popular and best performing techniques for the problem Vafeiadis et al. (2015). Given that the dataset will be different for an E-Commerce business, it will be interesting to compare whether similar techniques work as well in this study. In their research, they also went on to use a boosting algorithm to improve performance. Another 2022 churn prediction paper Lalwani et al. (2022) stated that boosting considerably improved performance of classifiers, so this is something that was explored in my work.

There have been a number of recent reviews of machine learning methods for customer churn prediction in the telecommunications industry. In a 2019 paper Śniegula, Poniszewska-Marańda and Popović (2019), the authors set out to investigate which machine learning models were best suited to predicting telecom customer churn. Their comparison was limited to three methods; K-means, Decision trees, and Neural Networks. Decision trees outperformed the other models, however, a downfall in the research conducted by the authors was that, while they identified the problem of imbalanced data in their dataset, they didn't deal with this issue. This could effect the analysis conducted, as the performance of neural networks can be significantly deteriorated due to imbalanced data Buda, Maki and Mazurowski (2018). Imbalanced data is an issue that is commonly faced in similar fields such as fraud detection. Some useful techniques for dealing with this issue, such as undersampling and oversampling, are covered in the comprehensive 2012 paper, 'An overview of classification algorithms for imbalanced datasets' Ganganwar (2012).

Another recent paper, had a similar aim to the aforementioned paper Śniegula, Poniszewska-Marańda and Popović (2019). The authors set out to conduct a full review of the methods, factors and techniques present in the area of telecom customer churn prediction from 2005-2020 Jain, Khunteta and Srivastava (2021). Feature extraction was found to be extremely important, and evaluating performance using confusion matrices is recommended by the authors. Following from the identification of feature extraction as an important factor, deep learning methods, and particularly Convolutional Neural Networks, are recommended. The reason for this is

that no feature extraction is needed. Neural networks effectively extract their own features. They can, in fact, be used for feature extraction Wiatowski and Bölcskei (2017). Principal Component Analysis and using co-relation matrices are recommended for feature extraction for other methods, dealing with the curse of dimensionality.

It could also be possible to experiment with different hybrid machine learning techniques for predicting customer churn. Some novel methods have been implemented in recent years. In a paper released in 2021, again with the goal of predicting customer churn likelihood in the telecommunications industry, hybrid ensemble learning approaches were used Tavassoli and Koosha (2021). The approaches that were used were based on bagging and boosting, with three different hybrid approaches being tested. The approaches, were combining bagging and boosting (performing boosting on each bagged sample), bagging and bagging (performing bagging again on each bagged sample) and bagging of neural network (applying a simple neural network to each sample). The hybrid approaches generally achieved better performance when compared with many of the most popular techniques applied to the same problem, so it is worth investigating the use a hybrid approaches in similar areas. The authors of the paper made a final note that it may be beneficial for one to try a different algorithm instead of a neural network to change their third hybrid ensemble technique.

2.1.2 Banking

Banking has been another popular industry for research into methods for predicting customer churn.

In a 2016 paper, Decision Trees were used to classify whether a banking customer was a churner or non-churner Keramati, Ghaneei and Mirmohammadi (2016). The authors proposed the use of forward selection and backward elimination methods for feature selection, with the backward elimination method performing better. A limitation of the paper was that only one model was tested. Decision trees were evaluated on various metrics and performed well across the board.

Six different machine learning methods were compared in a 2020 paper Dias, Godinho and Torres (2020). The six methods used were, random forests, support vector machine, stochastic boosting, logistic regression, classification and regression trees and multivariate adaptive regression splines, with stochastic boosting performing the best among them. The authors took an interesting approach for attempting to predict churning customers over each month in a six month period. They built six separate models to each predict a month in the period, which appeared to be an effective method.

In a 2020 paper with the goal of predicting customer churn likelihood in the context of banking, random forests and logistic regression techniques were implemented COŞER et al. (2020). Perhaps more interestingly, the optimisation technique used in this paper was the Grid Search optimisation technique. Grid search is used to obtain the optimal predictive model through trialing different combinations of hyperparameters. This is, again, an effective method that I found in my research and chose to implement and utilise in my study.

2.2 Customer Churn Prediction in E-Commerce

In my initial research, it became apparent that this problem hasn't been tackled very many times in scientific papers. A reason for this is that E-Commerce companies have neglected the possibility of predicting customer churn, due to the absence of face-to-face interaction with their customer base. However, E-commerce companies can use big data to understand the behaviour of their customers evermore in today's world Alrumiah and Hadwan (2021).

In a 2011 paper, the problem of customer churn prediction in E-Commerce was tackled Yu et al. (2011). The authors proposed an Extended Support Vector Machine method to predict whether a customer was likely to cancel or not. The authors tested their novel method against Support Vector Machines, Neural Networks, and Decision Trees, with their model outperforming all others. This extended SVM model introduced non-linearities to deal with the complexity of the dataset, and introduced two new variables to deal with the imbalance of churners/non-churners in the dataset. These changes had a positive impact on the accuracy of the standard SVM model. It is important to note that the data was highly imbalanced and complex in their case. The authors dealt with these specific issues in their novel method, so their extended model would not be necessary for most datasets.

More recently, the authors of a 2017 paper again applied a variation of Support Vector Machines to predict customer churn in the E-commerce industry Gordini and Veglio (2017). They applied an AUC parameter-selection technique to select the hyperparameters and found that paired with SVM, it outperformed baseline methods when applied to the noisy, imbalanced and nonlinear marketing data. The authors of the paper pointed out that more research should be done into selecting the correct kernel for the SVM.

Area under curve parameter selection has been effectively used in the above mentioned paper Gordini and Veglio (2017). This method of parameter selection uses cross validation and aims to maximize the classification power of the selected model measured by the area under the curve metric Jiang, Huang and Zhang (2013). The grid search optimisation that I used in my study makes use of this.

In a recent 2022 paper, Logistic Regression and Support Vector Machines were compared Xiahou and Harada (2022). This paper is the most comprehensive comparison that I have found applied to an E-commerce dataset. The authors used K-means clustering to first segment the customers, which improved performance. They found that SVM performed better than LR. A critique of this paper would be that more clustering methods should have been compared, and that feature selection wasn't investigated.

In terms of E-Commerce data there is an interesting dataset on Kaggle Ecommerce Customer Churn Analysis and Prediction (2010). Subsequently, a few notebooks implementing churn prediction models on this dataset have also been posted, one achieving high accuracy with various machine learning models E-commerce Customer Churn (2010). The models used were; Logistic Regression, Linear Discriminant Analysis, Decision Tree, Random Forest, K-Nearest Neighbours and XGBoost. Again, these models are consistent with what has appeared throughout my research.

2.3 Chapter Summary

The reader should now feel aware of the algorithms and machine learning methods that have been used in these areas and have an idea of the more effective ones. The chosen methods will be formally introduced in a later chapter, chapter 4.

Chapter 3

Ethical Discussion

In this chapter I will briefly cover the ethical issues in the context of this study, giving the reader an insight into this central component of a data driven piece of research.

Ethical considerations are a key part of all scientific studies, determining what's right or wrong in the process. They can be thought of as a set of principles that guide you throughout your research. Scientists must always adhere to these principles if they want to maintain scientific and moral integrity. Ethics is a burning issue in the field of data science, the protection of personally identifiable information (PII) being at the heart of this issue. In recent years there have been many general data protection regulation (GDPR) laws passed around the world, including here in the UK. These laws were enacted to uphold seven principles that act together to protect the individual and their data from companies looking to take advantage of them. These laws have been a step in the right direction for eradicating some of the ethical issues in the field of data science, individuals now have to explicitly opt-in to their data being used in countries that have enacted GDPR laws. Unfortunately, there are still many countries that have not passed these laws as of yet, such as the United States.

3.1 Ethics in This Study

In keeping with what I have outlined in the previous section, it was imperative for me to investigate the ethics of this study prior to beginning my work. The two main questions that needed to be answered to validate that there were no ethical issues were as follows:

1. Are the company fully GDPR compliant?
2. Has the data been fully anonymised? i.e. No PII in the data.

Firstly, like any online business in the UK, the company are required to be fully compliant with GDPR to avoid legal action and face detrimental fines. It is not only the fines that would effect the company, they would risk losing credibility and, along with that, a large portion of their customer base. It has been proven to me by the data team in the company that they are fully GDPR compliant. Explicit consent is given by the customer to store and use the data for analytics, and the company have already been making use of this.

Secondly, the data was fully anonymised by the company prior to me accessing it. This included the removal of names, email addresses (replaced with numerical IDs), full addresses, and any other PII.

Chapter 4

Methods Used in this Study

In this chapter the methods used for feature selection, the algorithms used for classification, the performance metrics used for evaluation, and the tools used for implementing these methods will be outlined and descriptions will be given where appropriate. The reader should expect to get an idea of the workings of the machine learning algorithms used in the study and how they can be implemented over the course of the chapter.

Note: all code for this project can be found in a github repository, which a link is provided for in the appendix.

4.1 Feature Selection Methods

Feature selection methods are used to refine the feature set before training a machine learning model. The reasons for implementing this step include reducing the chance of overfitting, speeding up model training, increasing interpretability/decreasing complexity (Occam's Razor), and improving performance. There is often a real emphasis put on this aspect of the process in building an accurate prediction model when dimensions are high Anukrishna and Paul (2017). Given that the dataset's feature space is large and to avoid the curse of dimensionality, a number of different feature selection methods were utilised. The methods fall into three different categories: filter methods, wrapper methods, and embedded methods.

4.1.1 Filter Methods

Filter methods are feature selection methods that filter out features from the initial set based on how they score in different statistical tests for their correlation with the dependent variable Porkodi (2014). The two filter based methods that I chose for this study were Pearson's Correlation, and Chi-Square.

Pearson's Correlation

Pearson's correlation is a measure of the linear dependency of two variables. It takes values between -1 (highly negatively linearly dependent) and +1 (highly positively linearly dependent), the higher the absolute value the better. The equation for this value is given as follows:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{4.1}$$

where x bar and y bar refers to the mean values.

Chi-Square

We can apply the chi-square test to groups of features to investigate the likelihood of correlation or association between them, again giving us an indication of which variables might be good predictors. The equation for the chi-square statistic is as follows:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{4.2}$$

where O is the observed value and E is the expected value.

4.1.2 Wrapper Methods

Wrapper based methods are iterative feature selection methods that require one machine learning model to be implemented. Features are added/removed based on the performance of the model Gnana, Balamurugan and Leavline (2016). They are very effective methods, but can be computationally expensive. The three most common methods are forward selection, backward elimination, and recursive feature selection.

Forward Selection

Forward selection refers to a feature selection method which begins with an empty model and adds in variables one by one, adding the variable that improves performance most at each step Macedo et al. (2019).

Backward Elimination

On the other hand, backward elimination refers to a feature selection method which begins with a model with all variables included and removes variables one by one, removing the variable that contributes the least to performance at each step Narin, Isler and Ozer (2014).

Recursive Feature Elimination

Ultimately, the method that I chose to implement in this study was recursive feature elimination. This is a greedy process, the goal of which is to find the best set of features Yan and Zhang (2015). In this method models are repeatedly created, with the most and least important features put aside at each run until the full feature set is exhausted. The features are ranked based on how early they were eliminated.

4.1.3 Embedded Methods

Embedded feature selection methods use machine learning algorithms which have their own built in way of ranking feature importance to refine the feature set, in a way they combine the

benefits of both filter and wrapper methods Maldonado and López (2018). Three of the most popular embedded methods that exist utilise the random forest, ridge regression, and lasso regression algorithms. I made use of each of them in the study.

Random Forest

The random forest algorithm measures feature importance based on node impurities using either the Gini index or information gain metrics Louppe (2014). In our case the Gini index was used, the equation for which can be seen below:

$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i) \tag{4.3}$$

where $p(c_i)$ is the percentage of class c_i in a node.

Ridge Regression

Ridge regression performs $L2$ regularization to avoid overfitting van Wieringen (2015). This means that a penalty is added equivalent to the square of the magnitude of coefficients. If the feature is irrelevant, the regularization will penalise it more (i.e. bring the coefficient close to 0) and vice versa.

Lasso Regression

Lasso regression performs $L1$ regularization to avoid overfitting Ranstam and Cook (2018). This means that a penalty is added equivalent to the absolute value of the magnitude of coefficients. Again if the feature is irrelevant, the regularization will penalise it more (i.e. bring the coefficient close to 0) and vice versa.

4.2 Classification Methods

One of the main aims of the study was naturally to find the model that is best suited to predicting customer churn on the E-commerce dataset. From the many prediction methods that have been evaluated throughout the years on the problem of customer churn prediction, I have selected the most popular and best performing models to apply to the problem and compare. The following are the models that were compared in this study: Logistic Regression (baseline), Random Forest, Adaptive Boosting, and Support Vector Machine.

4.2.1 Logistic Regression

Logistic regression (LR) is a very popular method for classification and it appeared frequently throughout the research in this area. LR is often outperformed by other models. However, LR has it's advantages, including, extremely fast learning speed and easy interpretability. LR can sometimes be the best performing model, for example in a study done predicting fetal abnormalities it was found that no other machine learning model outperformed it Kuhle et al. (2018). LR has proved a very useful and reliable baseline model for this study.

Logistic regression makes use of the sigmoid/logistic function, which outputs a value of between 0 and 1. This output value can be interpreted as a probability for classification problems Sperandei (2014). The sigmoid/logistic function can be seen below:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.4}$$

Relating the sigmoid equation back to linear regression where we previously had $y(x) = w^T x + b$, we now have $y(x) = \frac{1}{1+e^{-(w^T x+b)}}$ in logistic regression. The usual threshold value for logistic regression is 0.5, meaning that any value greater than 0.5 is predicted to be of class 1 and any below is predicted as class 0.

4.2.2 Random Forests

Decision trees and, in particular, random forests have remained highly effective for classification in recent times. This has been seen throughout my research and on other complex datasets Fratello and Tagliaferri (2018). Simple implementations of decision trees have also frequently been used as a baseline model for classification studies, along with logistic regression as defined above. Decision trees make use of recursive splitting to reach final outputs at the leaf nodes.

Random forests combine decision trees and ensemble learning. The algorithm makes use of many decision trees as weak learners, aggregating the predictions at the end Ali et al. (2012). The ensemble learning technique used is called bagging, short for bootstrap aggregating. The process that the random forest algorithm follows is:

1. A random subset of features is selected and a bootstrapped sample is obtained.
2. A decision tree of depth equal to the maximum depth specified is constructed using the bootstrapped sample of data points.
3. Steps 1 and 2 are repeated until the desired amount of decision trees have been constructed.
4. The final classifier output is given by majority vote using all trees in the forest.

If we take a training set S, with f different features and n data points, we can define a bootstrap training sample S_k sampled from S with replacement. S_k now contains n data points, but with m random features ($m \leq f$).

A random forest is defined as follows: It is a classifier consisting of a collection of tree based classifiers $C_k(x, S_k)$, $k = 1, ..., N$, where S_k are bootstrapped training samples, and each tree casts a vote for the most probable class of input x Oshiro, Perez and Baranauskas (2012).

One of the advantages of using bagging is that the out-of-bag error can be calculated on an ongoing basis, which gives a good estimation of the generalisation error of the random forest. The out-of-bag error is calculated by finding the mean of the prediction error on each training sample x_i, using only the votes from the trees that didn't have x_i in their bootstrapped training sample, S_k.

4.2.3 Adaptive Boosting

Boosting hasn't been used frequently in the area, but has proven to be very effective for problems of this nature. For example, in the similar area of fraud detection gradient boosting algorithms are popular Dhieb et al. (2019). One of the most popular gradient boosting algorithm is adaptive boosting or AdaBoost for short. The AdaBoost algorithm has been applied to the problem of E-commerce customer churn in a 2016 paper, achieving good accuracies of above the 80% mark Wu and Meng (2016).

AdaBoost, like random forests, is an ensemble technique that uses decision trees as the weak learner. However, in AdaBoost the depth of the decision trees is limited to 1 split, what we call decision stumps. The algorithm builds models sequentially, attempting to predict the error from the previous model each time Brownlee (2016). In AdaBoost equal weights are initially given to all the data points. Then, as the model is trained and tested, it assigns higher weights to points that are wrongly classified. With this method, the points which have higher weights are given more importance in the next model, giving the model a better chance to learn from it's errors. A mathematical description of this process is as follows:

Assign equal weights to every one of the n data points in the training set, S:

$$w_m(i) = \frac{1}{n} \tag{4.5}$$

Iteratively, for $m = 1, ..., k$, a weak classifer $C_m(x)$ is trained on the training set S, (at each iteration the weights, w_m, are updated), with the goal of minimizing the following error:

$$E_m = \sum_{i=1}^{n} w_m(i) I(C_m(x_i) \neq y_i) \tag{4.6}$$

I here is the indicator function, which gives the value 1 if the statement inside the function is true and 0 if it's argument is false.

Note: the weights are updated as follows:

$$w_{m+1}(i) = w_m(i) exp(a_m I(C_m(x_i) \neq y_i)) \tag{4.7}$$

where, $a_m = \frac{1}{2} ln(\frac{1-e_m}{e_m})$ and $e_m = \frac{E_m}{\sum_{i=1}^{n} w_m(i)}$

The final classifier is given by:

$$C(x) = \arg\max_{l \in Y} \sum_{m=1}^{k} a_m I(C_m(x_i) = l) \tag{4.8}$$

4.2.4 Support Vector Machines

Support Vector Machines (SVMs) have been prominent throughout my research so far, generally performing well for predicting customer churn. The goal of an SVM is to separate data points in each class by constructing a separating hyperplane Suthaharan (2016). SVMs make use of kernel functions to map the data to new, often higher-dimensional, spaces. The idea behind

this is that a non-linear decision boundary can become a linear one in higher dimensions. The support vectors are data points close to the hyperplane, which help define the plane itself.

In the context of this study's case of a binary classification task: We have training set with input vectors $x_i \in R^N$ and labels $y_i \in \{-1, 1\}$. We can then define a separating hyperplane to classify points as follows:

$$w \cdot x + b = 0 \tag{4.9}$$

We can define two equations for calculating the margin of the classifier:

$$w \cdot x + b = -1, \;\; w \cdot x + b = 1 \tag{4.10}$$

In order for the classifier to generalise well, we want to maximise the margin distance between the equations in 4.10. The distance between is given by $\frac{2}{||w||}$. So, if we want to maximise this, it's the same as minimising $\frac{||w||}{2}$.

So, we now have the following optimisation problem:

$$\min \phi(x) = \frac{||w||^2}{2} = \frac{1}{2}(w \cdot w) \tag{4.11}$$

such that $y_i((w \cdot x_i) + b) - 1 \geq 0$ for $i = 1, 2, ..., n$.

The SVM algorithm then makes use of the Lagrange function to transform this problem Yin and Hou (2016). The following is the new formulation:

$$L(w, b, \alpha) = \frac{1}{2}(w \cdot w) - \sum_{i=1}^{n} \alpha_i [y_i((w \cdot x_i) + b) - 1] \tag{4.12}$$

The α here is the Lagrange multiplier.

We want $L(w, b, \alpha)$ to be minimised with respect to both w and b, giving us the following two equations:

$$\frac{\delta L}{\delta w} = 0, \;\; \frac{\delta L}{\delta b} = 0 \tag{4.13}$$

From here, we can find that the following optimal w and b are obtained:

$$w^* = \sum_{i=1}^{n} y_i \alpha_i x_i, \;\; \sum_{i=1}^{n} y_i \alpha_i = 0, \;\; b^* = y_i - \sum_{i,j=1}^{n} y_i \alpha_i (x_i \cdot x_j) \tag{4.14}$$

So, we now have the SVM classifier function:

$$y = sign((w \cdot x) + b) = sign(\sum_{i=1}^{n} y_i \alpha_i (x \cdot x_i) + b^*) \tag{4.15}$$

Note: the dot product can be replaced by kernel functions to extend the SVM classifier function to nonlinear problems. This is one of the most powerful aspects of the SVM classifier.

The kernel functions used in cross validation during this study were the linear, polynomial, radial basis function and sigmoid kernels, all of which can be very effective depending on the nature of the problem.

4.3 Performance Metrics

A key part of the process is, of course, evaluating the performance of our models. An extremely popular way of evaluating the performance of binary classification models is by constructing the confusion matrix Hossin and Sulaiman (2015). A number of metrics can be calculated from the values in the confusion matrix. Another effective method is plotting the receiver operating characteristic (ROC) curve, and finding the area under the curve (AUC) Hajian-Tilaki (2013). The ROC curve plots the false positive (FP) rate versus the true positive (TP) rate.

For the purpose of evaluating the performance of the churn prediction models we will focus on four metrics, accuracy, precision, recall, and AUC.

The accuracy metric is an evaluation of the overall performance of the model. It is the number of correctly predicted samples divided by the full sample size.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.16}$$

The recall metric is an evaluation of the coverage of the model. It is the number of correctly predicted positive samples divided by the full amount of predicted positive samples.

$$Recall = \frac{TP}{TP + FN} \tag{4.17}$$

The precision metric is an evaluation of the performance of the model on positive samples. It is the number of correctly predicted positive samples divided by the full amount of positive samples.

$$Precision = \frac{TP}{TP + FP} \tag{4.18}$$

The AUC metric is used to measure how well the models can distinguish between the two different classes. Like the other metrics, a higher value means better performance here.

4.4 Tools Utilised

There are an abundance of open source tools available to aid the implementation of machine learning projects, in this section I will list the tools that were utilised in this study.

Firstly, the coding language used was Python, the integrated development environment (IDE) chosen was Visual Studio Code, and the code was written in Jupyter Notebooks within Visual Studio Code. Python is the most popular language in the field of data science currently, so it

was an easy choice for this study, while Visual Studio Code provided a user friendly environment to create the Jupyter Notebooks.

One of the benefits of coding in Python is the vast amount of libraries/packages that you can import as tools for your work. The libraries that were imported to complete the code for this study were as follows:

- Pandas Wes McKinney (2010) - Package to aid reading the files into the format of dataframes, and aid with preprocessing.
- Numpy Harris et al. (2020) - Very useful package for completing any numerical tasks.
- Matplotlib Hunter (2007) - Package to plot and visualise data.
- Seaborn Waskom (2021) - Another package to plot and visualise data.
- Scikit-Learn Pedregosa et al. (2011) - Package to aid preprocessing, feature selection, model building, optimisation, and evaluation of performance.
- DateTime - Package to convert data points into timestamps.

4.5 Chapter Summary

This chapter has briefly covered the main bulk of the concepts and tools that were used to build the predictive models. The reader should have gained an overview of the methods used and how they could be implemented. However, further reading on these topics would be recommended.

If the reader would like to delve deeper into the theory, statistics and mathematics underlying the methods in this chapter, the following two books would be recommended: Géron (2017) and Murphy (2012).

Chapter 5

Empirical Analysis

In this chapter the flow of work in the study will be provided, including the stages of initial analysis of the data, preprocessing of the data, feature selection and model optimisation. The more complex of the methods used in this chapter were introduced in chapter 4. The reader should expect to be somewhat immersed into the practical work that went in to producing the results in this study, giving a rough framework for a similar project.

5.1 Data

Three datasets were extracted as Microsoft Excel files from the company's database containing information regarding customer details and behaviour. The three datasets were chosen due to the relevancy of the variables included in them. The first of which included data on the customer's subscription, i.e. which product they had chosen etc., the second of which included data on customer information i.e. date of birth etc., and the third of which included data again on customer information (similar data points, stored in a different file). The company had changed the method of collecting the customer information, so this is the reason for the two separate files.

Many unnecessary columns were removed from the data, and the files were read into python and merged using the Pandas package. After merging, the final dataset was formed containing behaviour data on 25,046 subscribers. This dataset now contained the dependent variable; churn (binary), and 16 predictor variables; subscription date, order count (number of orders), failed payment (binary), failed payment count, price of plan, total spent, orders count (number of orders for current product), discount amount, lifetime refund amount, used finasteride (binary), used minoxidil (binary), hair loss level, date of birth, lifetime value, product on subscription and shipping city.

5.2 Data Preprocessing

Data preprocessing is often an essential step before analysing the data, especially when dealing with raw real world data García, Luengo and Herrera (2015). As expected, the data from the company was initially unclean, and appeared quite problematic. There were some missing values in a number of the variables which were dealt with accordingly, for example by filling in with median values. The datetime package was utilised to correctly format both the subscription

date and date of birth columns. Next step was to implement some feature engineering to create some more, perhaps more relevant, predictor variables. Firstly, there were two problematic categorical variables that required engineering, these were product on subscription and shipping city. When the value counts for product on subscription were investigated, there were 400 different combinations of products, which is too many to encode as an effective input variable. Therefore, I chose to retain the 8 main products which accounted for almost 90% of the customers and grouped the rest of the product values under the label, other. Similarly, when the value counts for shipping city were investigated, there were 4,646 different cities listed. For this variable, I decided that an effective solution would be to divide the cities into 4 tiers based on the number of customers in each. Another variable, that I decided was worth adding was monthly/order spend. This variable was added simply by dividing the total spent column by the order count column in the dataframe. The next step was to encode the categorical variables, so they were ready for use as input variables. I used the label encoder from Scikit-learn for this task.

Once the data preprocessing was fully completed, the data was split into two strictly separate datasets; the test dataset, making up around 30% of the whole dataset and set aside for the sole purpose of testing our models performance on unseen data, and the training dataset, making up around 70% of the whole dataset and to be used for feature selection and training of our models. This part of the process was of particular importance, as it could skew performance if done incorrectly. Code for splitting the data can be seen below (note paused refers to churn):

```
X = df.drop('paused',axis=1)
y = df['paused'].astype('category')
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size =
    0.3, random_state = 0)
```

5.3 Feature Selection

As we know, not all features would necessarily be important and effective for the task of predicting customer churn. It has been well researched that having irrelevant, redundant features in a feature set for a classification problem can slow down convergence and negatively impact performance Tang, Alelyani and Liu (2014).

So, the next key step in the process was implementing feature selection. Many effective feature selection methods exist, which are all well suited to different problems. It is often difficult to decide which one to use. Therefore, I decided to use an ensemble approach Seijo-Pardo et al. (2017). Six different feature selection methods were implemented. Subsequently, each variable was given a tally based on how many times they had been selected overall. This ensemble method should give a robust indication of which features are better suited as predictors. The methods chosen were a mix of filter, wrapper and embedded methods. They were as follows; Pearson's Correlation (Pearson), Recursive Feature Elimination (RFE), Ridge Regression (Ridge), Lasso Regression (Lasso), Random Forest (RF), and Chi-Squared (Chi^2). The tallies for the features can be seen in the below table, (True meaning the feature was selected using that method, and False if not).

As we can see from 5.1 each feature was selected by at least one of the methods that were implemented. This sparked the idea that the performance of models trained on different sets of features could be tested. To ensure that there would be no crossover between the models,

Feature	Pearson	RFE	Ridge	Lasso	RF	Chi^2	Total
value	True	True	True	False	True	True	5
monthly_spend	True	True	True	False	True	True	5
used_min_or_reg	True	True	False	True	False	True	4
used_fin_or_prop	True	True	False	True	False	True	4
type	True	True	False	True	False	True	4
total_spent	True	False	True	False	True	True	4
price	True	False	True	False	True	True	4
failed	True	True	False	True	False	True	4
order_count	False	True	True	False	False	True	3
sub_month	False	True	True	False	False	False	2
orders_count	False	True	True	False	False	False	2
hair_loss_level	False	True	False	False	False	True	2
discount	False	True	False	False	True	False	2
age	False	True	True	False	False	False	2
refund_count	False	True	False	False	False	False	1
prod	False	True	False	False	False	False	1
failed_count	False	True	False	False	False	False	1
city_tier	False	True	False	False	False	False	1

Table 5.1: Feature Selection Tallies.

three separate notebooks of code were utilised. One using the full feature set during training and testing, one using only the features with a count of at least two in 5.1 (consisting of 14 features), and one with only the features with a count of at least three in 5.1 (consisting of 9 features). Another reason for implementing this method was that all of the features that were only selected once were selected by RFE. RFE has been highly effective when applied to similar problems in the area of churn prediction Farquad, Ravi and Raju (2014).

The features that appeared to be the most prominent predictor variables were value, monthly spend, used min or reg (referring to use of minoxidil), used fin or prop (referring to use of finasteride), subscription type (subscription frequency), total spent, price, failed (whether the customer has failed payments), and order count.

5.4 Hyperparameter Tuning

Once feature selection was completed, it was time to train the models on the training data. The aim of training the models is to learn the model parameters and optimise the model hyperparameters.

Hyperparameters are parameters in a machine learning model who's values are chosen before the model is trained, and alter the behaviour of the model. Hyperparameter tuning/optimisation refers to identifying the optimal set of hyperparameters that maximise the performance of the predictive model. For this task, I opted to use GridSearchCV from the Sci-Kit Learn package. This combines the grid search optimisation technique with K-fold cross validation Kartini et al. (2021).

5.4.1 Grid Search Optimisation

This is a very simple, intuitive hyperparameter tuning technique. We define a grid of possible values for each hyperparameter, and train the model with each combination of these values on the training data. Performance metrics are calculated for each combination using the validation data (which is a portion of the training data that's withheld during training and not used to directly train the model). The values trialled for each model during grid search optimisation can be seen in the below code:

```
paramsLR = [{'clf':[LogisticRegression(max_iter=10000)],
             'clf__C':[0.0001,0.001,0.01,.1,1],
             'clf__solver':['lbfgs','liblinear']
             }]

paramsRF = [{'clf':[RandomForestClassifier()],
            'clf__n_estimators': [100,200,300,400],
             'clf__max_features': ['sqrt', 'log2'],
             'clf__max_depth' : [8,9,10,11],
            'clf__criterion':['gini','entropy'],
            }]

paramsAda = [{'clf': [AdaBoostClassifier()],
              'clf__n_estimators': [300,400,500,600,700,800],
              'clf__learning_rate': [0.0001,0.001,0.01,.1,1]
             }]

paramsSVM = [{'clf': [svm.SVC()],
              'clf__C': [0.5,1,1.5,2],
              'clf__kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
              'clf__degree': [2,3,4],
              'clf__gamma': ['scale', 'auto']
            }]
```

5.4.2 Cross Validation

Cross validation uses different sets of the data to train and test the models, giving a better indication of performance than just using one variation Wong (2015). A commonly used and highly effective method is K-fold cross validation, and it is what I opted to use. The dataset is split into K folds, we then iterate through the folds until each of the k folds has been used as the validation set. The performance metrics are then averaged over each run of the model giving us a good indication of performance before we predict the test data Wong and Yeh (2019). In the case of GridSearchCV each set of possible input hyperparameters are trained on the k folds.

5.5 Chapter Summary

The reader should now have a mapping of the workflow in the main bulk of the practical side of this study. Each of the four sections in this chapter are essential steps in the process of building accurate machine learning models.

Please note that separate optimisation was done for each of the three different feature sets: Full, refined, and further refined.

It should now be clear to the reader that upon completion of the practical work covered in this chapter, the models have been built and optimised. This gave the optimal model parameters and hyperparameters for the four models (the combinations that performed best for each of the four models when GridSearchCV was implemented on the training data). These models can now be used to make predictions on the test data, the results of which will be given in the next chapter. The code for the classifier class can be seen below:

```
class classif(BaseEstimator):

    def __init__(self, estimator=None):
        self.estimator = estimator

    def fit(self, X, y=None):
        self.estimator.fit(X,y)
        return self

    def predict(self, X, y=None):
        return self.estimator.predict(X,y)

    def predict_p(self, X):
        return self.estimator.predict_p(X)

    def score(self, X, y):
        return self.estimator.score(X, y)
```

Chapter 6

Results and Analysis

In this section, the predictive performance of the selected models when applied to the unseen test data will be evaluated. The performance of the models trained on the full feature set, the refined 14 feature set, and the further refined 9 feature set will all be given. These performances will then be analysed and comparisons will be made between the different models and the different feature sets.

The reader will be given the values for all the confusion matrices in this section, along with the values for the performance metrics which were introduced in chapter 4. The reader should also expect some analysis and comparison of the performances.

6.1 Prediction Performance

This section will be broken down into three subsections to give the performance metrics for each of the final models on each of the three feature sets: full, refined, and further refined.

6.1.1 Full Feature Set

Below I will summarise the prediction performance of each of the four models when the full feature set was utilised for training:

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1495	877
Actual Churn (1)	645	4497

Table 6.1: Confusion Matrix - Logistic Regression on Full Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1793	579
Actual Churn (1)	423	4719

Table 6.2: Confusion Matrix - Random Forest on Full Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1678	694
Actual Churn (1)	516	4626

Table 6.3: Confusion Matrix - AdaBoost on Full Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1656	716
Actual Churn (1)	460	4682

Table 6.4: Confusion Matrix - SVM on Full Feature Set.

Model	Accuracy	Recall	Precision	AUC
Logistic Regression	0.7974	0.8368	0.8746	0.7524
Random Forest	0.8666	0.8907	0.9177	0.8348
AdaBoost	0.8389	0.8695	0.8996	0.8029
SVM	0.8435	0.8674	0.9105	0.8043

Table 6.5: Performance Metrics on Full Feature Set.

We can see from Table 6.5 that, as expected, LR performs the worst out of the four models on each of the metrics here. However, the LR algorithm converges extremely quickly and it provides a good baseline performance for the purpose of comparison.

The best performing algorithm was random forest, not only in the accuracy metric but also on recall, precision and AUC, this was followed by SVM, with AdaBoost performing third best.

Random forest had an accuracy of 86.7%, recall of 89.1%, precision of 91.8%, and AUC of 83.5%.

Note: precision is particularly important for us in the context of this study, we want to identify as many of the churners as possible (precision is a measure of how accurate the model is on positive/churn samples).

6.1.2 Refined Feature Set

Below we will summarise the prediction performance of each of the four models when the refined feature set with 14 features was utilised for training:

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1448	924
Actual Churn (1)	626	4516

Table 6.6: Confusion Matrix - Logistic Regression on Refined 14 Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1729	643
Actual Churn (1)	423	4719

Table 6.7: Confusion Matrix - Random Forest on Refined 14 Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1642	730
Actual Churn (1)	526	4616

Table 6.8: Confusion Matrix - AdaBoost on Refined 14 Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1619	753
Actual Churn (1)	421	4721

Table 6.9: Confusion Matrix - SVM on Refined 14 Feature Set.

Model	Accuracy	Recall	Precision	AUC
Logistic Regression	0.7937	0.8301	0.8783	0.7446
Random Forest	0.8581	0.8801	0.9177	0.8267
AdaBoost	0.8328	0.8634	0.8977	0.7923
SVM	0.8438	0.8624	0.9181	0.8000

Table 6.10: Performance Metrics on Refined 14 Feature Set.

We can see from Table 6.10 that, as expected, LR again performs the worst out of the four models on each of the metrics here. Again, the LR algorithm converges extremely quickly and provides a good baseline performance.

The best performing algorithm on accuracy, recall and AUC was random forest, while the SVM algorithm performed best on precision. However, SVM only marginally outperformed random forest on precision, so random forest was again chosen as the best performing model here.

Random forest trained on the refined 14 feature set had an accuracy of 85.8%, recall of 88.0%, precision of 91.8%, and AUC of 82.7%.

6.1.3 Further Refined Feature Set

Below I will summarise the prediction performance of each of the four models when the further refined feature set with 9 features was utilised for training:

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1442	930
Actual Churn (1)	633	4509

Table 6.11: Confusion Matrix - Logistic Regression on Refined 9 Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1675	697
Actual Churn (1)	504	4638

Table 6.12: Confusion Matrix - Random Forest on Refined 9 Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1601	771
Actual Churn (1)	573	4569

Table 6.13: Confusion Matrix - AdaBoost on Refined 9 Feature Set.

	Predicted Non-Churn (0)	Predicted Churn (1)
Actual Non-Churn (0)	1498	874
Actual Churn (1)	485	4657

Table 6.14: Confusion Matrix - SVM on Refined 9 Feature Set.

Model	Accuracy	Recall	Precision	AUC
Logistic Regression	0.7919	0.8290	0.8769	0.7424
Random Forest	0.8402	0.8694	0.9020	0.8032
AdaBoost	0.8211	0.8556	0.8886	0.7818
SVM	0.8191	0.8420	0.9057	0.7686

Table 6.15: Performance Metrics on Refined 9 Feature Set.

We can see from Table 6.15 that, as expected, LR again performs the worst out of the four models on each of the metrics here. Extremely fast convergence and good baseline performance is provided once again by the LR algorithm.

Similar to the refined 14 feature set, the best performing algorithm on accuracy, recall and AUC was random forest, while the SVM algorithm performed best on precision. SVM again only marginally outperformed random forest on precision, so random forest was chosen as the best performing model here.

Random forest trained on the refined 9 feature set had an accuracy of 84.0%, recall of 86.9%, precision of 90.2%, and AUC of 80.3%.

6.2 Analysis and Comparison

As I have noted in the first section, the clear best performing algorithm was random forest. With the results that I obtained, this is the model I recommend for use in similar churn prediction settings. Random forest is a long standing, reliable machine learning algorithm that is easy to interpret, relatively simple to implement and often suited to problems of a similar nature, so this is a positive result. As we have noted in chapter 4, Occam's Razor tells us complexity is not always necessary.

6.2.1 Feature Selection Analysis

The full feature set, with 18 features, had the best performance throughout all models, so the random forest algorithm trained on the full feature set is the best performing model from the study. There was a clear drop in performance across the metrics as the feature set was refined. This is surprising as an emphasis was put on the feature selection methods in this study. One

of the benefits of refining the feature set is the increased speed of training, this benefit was observed in the study. However, the drop in training time was not enough of a saving to justify the drop in performances of the algorithms. We can note here that the accuracy of random forest dropped by over 2.5% between the full feature set and the further refined 9 feature set.

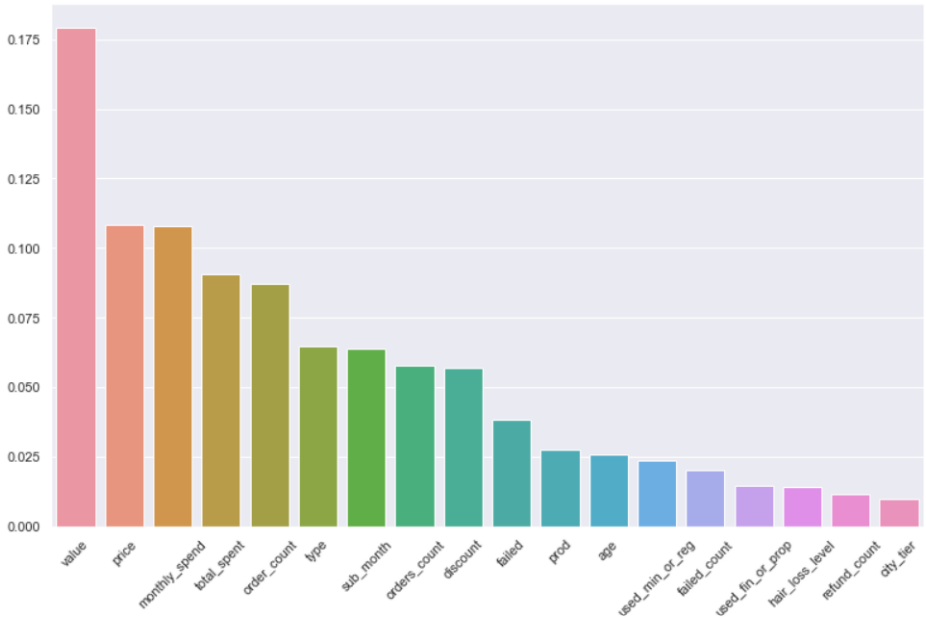

Figure 6.1: Feature Importance Random Forest

Studying the feature importance of the features in the random forest model trained on the full feature set in Figure 6.1, we can see that value was of the most importance. Contrary to the indication from our feature selection tallies, used min or reg and used fin or prop were of little importance as predictor variables. The features that this model tells us are of little importance are as follows: failed, prod, age, used min, used fin, failed count, hair loss level, refund count, and city tier. Subsequently, the RF algorithm was fitted to the data again without these features and performed well. However, the performance didn't improve upon the results from RF on the full feature set.

6.3 Chapter Summary

The reader should now be aware of the predictive results of this study, taking note of the confusion matrices and performance metric tables displayed in this chapter. Analysis has been provided, but the reader is encouraged to make insights and critiques of their own.

Note that, in the next chapters, the implications of these results for the business and in the context of the research will be discussed and analysed.

Chapter 7

Discussion

In this chapter, I will critique the process of the study, keeping in mind the results from chapter 6. An emphasis will be put on outlining some of the failings of the study. I will discuss system evaluation, and throughout the chapter I will analyse the process as a whole.

As we know, the applications of the models built in this study are ultimately business minded, and I have not yet discussed the implications of the results with respect to the company that provided the data. So, in this chapter I will also delve deeper into what this work means for the company, and whether the models are accurate enough to help influence business decisions to achieve data driven growth for the company.

The reader can expect to be given suggestions of future work that could be done throughout the chapter. It should also become clear what the future holds for the system.

7.1 Critique of the Process

Upon analysing the work, the areas that I will critique can be split into three subsections: Data, Methods, and System Evaluation. These three headings cover most aspects of the study. So, by identifying the failings in each I hope to convey what could be done to improve the models in the future.

7.1.1 Data

Firstly, the data was one thing in the process that was lacking. Data is key to building a successful classification model in all scenarios, so this is the most important critique to note for improving the study.

I was expecting that the company would provide access to more customer behaviour data such as: customer logins, customer amended orders, time spent on website, preferred login device, and other such data points. More work could also be done on engineering further features to add to our data. For example, variables relating to the times between orders could be taken into account as indicator variables, which has been successfully implemented in a 2020 churn prediction system Deligiannis and Argyriou (2020). In the case of this study's dataset, the times would have to be calculated with respect to the subscription frequency of the customer, i.e. how much the frequency of orders was deviating from the set frequency. These added

features could improve the performance of the system if they become available in the future, which is something that the data team at the company are working on making possible.

Another thing to note with respect to data is that datasets from subscription based E-commerce businesses can vary hugely. Data points are generally unique from business to business. Not surprisingly, training and testing on datasets from multiple businesses in the broad sector would be more effective. This would aid us in determining which models have a very good generalisation ability. However, it would be very time and resource consuming but if it becomes possible it would be highly recommended for future work.

7.1.2 Methods

As is the case with any study, the methods could always be altered, extended or changed to attempt to improve performance of the models. There are some key things that I will outline in this section to be further investigated on this dataset. The time constraints impacted on the scope of what was completed in this study, especially in terms of trialling different machine learning methods.

SVMs could be further experimented with. The SVM algorithm showed promising performance, even achieving the best precision score of any model, see Table 6.10. The use of kernel functions can be key to improving the performance of the SVM algorithm Savas and Dovis (2019). The use of custom kernel functions can be highly effective in SVM models for certain problems Ayush and Sinha (2019). Due to time constraints and SVMs being the slowest out of the models to train, no custom kernels were experimented with. Given the complexity of this task, this work could be reserved for it's own future project.

The feature selection methods failed in a sense, given that the full feature set provided the best performing model Table 6.2. Therefore, this part of the process should be further investigated. More methods could always be used, analysis of which would work best was lacking in this study. Principal component analysis provides an effective feature engineering/selection method which also reduces the issue of dimensionality Bro and Smilde (2014). This is a method that could improve the models speed and performance if implemented.

There are, of course, further machine learning models that could be implemented on this problem. Neural networks were not experimented with in the study and have proved to perform well on other binary classification problems Peng et al. (2018). Another boosting algorithm could also be trialled, for example extreme gradient boosting which has been shown to be an effective binary classifier Torlay et al. (2017).

Customer segmentation is also something worth studying in this area in future work. Customer segmentation could be implemented in many different ways to further develop this system. One core idea of segmenting the customers in this context would be to divide the customers into high, medium, and low characteristically loyal customer groups, i.e. the high group would have the lowest churn rate. This addition to the system would allow the company to formulate better targeted marketing strategies using the subdivided groups of customers. It could also improve the overall accuracy of the models if the segmentation was done before the prediction, or if a hybrid approach was used. In a similar paper, forecast results were found to be more accurate after segmenting the customers ZHUANG (2018). Along with that, a hybrid approach of both prediction and segmentation was found to have worked effectively in a 2017 paper Peker, Kocyigit and Eren (2017). K-means clustering appears to be the chosen segmentation method in churn prediction literature.

7.1.3 System Evaluation

While the results that were displayed in chapter 6 give a demonstration of the prediction performance of the models, the system has not yet been evaluated at a business level. In fact, the automated customer churn prediction system is something that I plan to finalise and put in to practice in the coming months.

After this automated system is running and identifying customers who are likely to churn, the evaluation of the performance at a business level will take a number of months. I am currently working with the data team from the company to get the system automated and running internally. The idea is that the models will be run daily on the active customer base, identifying customers who are likely to churn. The company will then have the opportunity to retain more customers through tailored incentives or other methods of convincing them to stay active. The ultimate evaluation of the system will come after a few months of tracking customer retention rates, hopefully seeing a decrease in churn percentage.

7.2 Research Implications

Although the reader should be aware of the implications of this study from a research aspect throughout, I will briefly take the time to explicitly note the main additions to the literature in this section. Given that there hasn't been many research papers published in this area, this comparison of the performance of four different classifiers of customer churn will be a very valuable addition to the current limited literature in the area. The second addition to the literature here is that the ensemble feature selection method used in this paper is novel in the area of churn prediction, this could prove to be a valuable method for future use.

7.3 Business Implications

In terms of the practical use of the models, the results of the study are encouraging. The random forest model trained on the full feature set achieved relatively high scores on all performance metrics Table 6.2, showing promising results before application in a business context. Building a functional churn prediction system with highly accurate models would prove very useful for the company. Identifying the customers who aren't satisfied with the current conditions would also allow the business to identify operational issues, product or pricing plan weak points, and customer preferences. Through analysis of the identified customers, the business could reduce the list of reasons for churn proactively. If the models can be utilised effectively, the company could reduce both customer churn and needless marketing spend. Overall, this could help the company's financial health hugely.

7.4 Chapter Summary

The reader should now have some ideas for future work that could be done following on from this study. As before, the reader is encouraged to analyse the study as a whole at this stage, note any other future work that should be conducted, and also critique the process of the study. System evaluation has also been introduced in this chapter. The reader should now understand the final use of the models that have been built, and the value that they could provide for the company once the final system has been put in motion.

Chapter 8

Conclusions

In the previous chapter, the sentiment may have come across quite negative in relation to the outcomes of the study. However, the study has been a very successful one given the time constraints. Within this concluding chapter, I will attempt to convey what this study has achieved in relation to the motivations behind the research and the performance of the models. It should first be noted that the RF model trained on the full feature set has been deemed to have a high enough performance to be included in the final system. This, alone, is a very strong achievement.

This paper has shown that, even with the failings I have outlined in chapter 7, machine learning models can be built that highly accurately predict customer churn on the chosen E-commerce dataset. Aligning with similar literature, the performance of various machine learning models were compared. The four chosen models were: LR, RF, AdaBoost, and SVM. As we noted in chapter 7, an addition to the literature in this paper was that 6 different feature selection methods were used to select the features and subsequently three different feature sets were trialled in training (using three different notebooks of code to ensure no overlap): full feature set, refined 14 feature set, and further refined 9 feature set. As we have seen, the RF model trained on the full feature set performed the best, and the training time saved for the refined feature sets wasn't significant enough to take into account. Although the feature selection methods failed to improve the models, this idea is still a valuable addition to the literature and should be used in future work.

The motivations for this research as outlined in chapter 1 were, firstly, to identify some of the feature selection methods that are best for E-commerce customer churn prediction, also with a view to outputting important predictor variables. Secondly, it was to find machine learning models that could be used in an accurate churn prediction system. As I have touched on already, the first of these motivations was investigated and further work is recommended on feature selection due to the full feature set providing the best model. Secondly, the results in this paper demonstrated that accurate machine learning models could be built for classifying churn on data from a Men's Health E-commerce subscription based company. As desired, the LR model performed as a baseline model with the top three performing models being RF, SVM, and AdaBoost. In terms of the application of this study, the motivation was also there to provide the reader with a framework for building machine learning models to apply to similar problems. Reflecting on the paper, I believe the tools and steps are now there to apply this approach for churn prediction or other similar classification problems. So, one of the main things for the reader to take forward should be the clear framework provided to embark on a

similar project.

Reflecting upon the outcomes of the work, this study has made it clear to me that machine learning should be utilised for customer churn prediction by online subscription businesses. Customer data points are there waiting to be analysed, and I have shown that models can be built to accurately classify based on these data points. Time and resources should be put into better understanding customers, and predicting what their next move may be. This paper should provide a good starting point for businesses.

Bibliography

Akter, S. and Wamba, S.F., 2016. Big data analytics in e-commerce: a systematic review and agenda for future research. *Electronic markets*, 26(2), pp.173–194.

Ali, J., Khan, R., Ahmad, N. and Maqsood, I., 2012. Random forests and decision trees. *International journal of computer science issues (ijcsi)*, 9(5), p.272.

Almohaimmeed, B., 2019. Pillars of customer retention: An empirical study on the influence of customer satisfaction, customer loyalty, customer profitability on customer retention. *Serbian journal of management*, 14(2), pp.421–435.

Alrumiah, S.S. and Hadwan, M., 2021. Implementing big data analytics in e-commerce: Vendor and customer view. *Ieee access*, 9, pp.37281–37286.

Anukrishna, P. and Paul, V., 2017. A review on feature selection for high dimensional data. *2017 international conference on inventive systems and control (icisc)*. IEEE, pp.1–4.

Ayush, K. and Sinha, A., 2019. Improving classification performance of support vector machines via guided custom kernel search. *Proceedings of the genetic and evolutionary computation conference companion*. pp.159–160.

Ben Rhouma, T. and Zaccour, G., 2018. Optimal marketing strategies for the acquisition and retention of service subscribers. *Management science*, 64(6), pp.2609–2627.

Bro, R. and Smilde, A.K., 2014. Principal component analysis. *Analytical methods*, 6(9), pp.2812–2831.

Brownlee, J., 2016. A gentle introduction to the gradient boosting algorithm for machine learning. `https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/`.

Buda, M., Maki, A. and Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106, pp.249–259.

COŞER, A., Aldea, A., Maer-Matei, M.M. and BEŞIR, L., 2020. Propensity to churn in banking: what makes customers close the relationship with a bank? *Economic computation & economic cybernetics studies & research*, 54(2).

Dahiya, K. and Bhatia, S., 2015. Customer churn analysis in telecom industry. *2015 4th international conference on reliability, infocom technologies and optimization (icrito)(trends and future directions)*. IEEE, pp.1–6.

Deligiannis, A. and Argyriou, C., 2020. Designing a real-time data-driven customer churn risk indicator for subscription commerce. *International journal of information engineering & electronic business*, 12(4).

Dhieb, N., Ghazzai, H., Besbes, H. and Massoud, Y., 2019. Extreme gradient boosting machine learning algorithm for safe auto insurance operations. *2019 ieee international conference on vehicular electronics and safety (icves)*. IEEE, pp.1–5.

Dias, J., Godinho, P. and Torres, P., 2020. Machine learning for customer churn prediction in retail banking. *International conference on computational science and its applications*. Springer, pp.576–589.

E-commerce customer churn, 2010. `https://www.kaggle.com/code/ankitverma2010/e-commercecustomerchurn`.

Ecommerce customer churn analysis and prediction, 2010. `https://www.kaggle.com/datasets/ankitverma2010/ecommerce-customer-churn-analysis-and-prediction`.

Farquad, M.A.H., Ravi, V. and Raju, S.B., 2014. Churn prediction using comprehensible support vector machine: An analytical crm application. *Applied soft computing*, 19, pp.31–40.

Fratello, M. and Tagliaferri, R., 2018. Decision trees and random forests. *Encyclopedia of bioinformatics and computational biology: Abc of bioinformatics*, 1, p.3.

Ganganwar, V., 2012. An overview of classification algorithms for imbalanced datasets. *International journal of emerging technology and advanced engineering*, 2(4), pp.42–47.

García, S., Luengo, J. and Herrera, F., 2015. *Data preprocessing in data mining*, vol. 72. Springer.

Géron, A., 2017. Hands-on machine learning with scikit-learn and tensorflow: Concepts. *Tools, and techniques to build intelligent systems*.

Gnana, D.A.A., Balamurugan, S.A.A. and Leavline, E.J., 2016. Literature review on feature selection methods for high-dimensional data. *International journal of computer applications*, 136(1), pp.9–17.

Gordini, N. and Veglio, V., 2017. Customers churn prediction and marketing retention strategies. an application of support vector machines based on the auc parameter-selection technique in b2b e-commerce industry. *Industrial marketing management*, 62, pp.100–107.

Hajian-Tilaki, K., 2013. Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation. *Caspian journal of internal medicine*, 4(2), p.627.

Hamilton, R.W., Rust, R.T. and Dev, C.S., 2017. Which features increase customer retention. *Mit sloan management review*, 58(2), pp.79–84.

Harris, C.R., Millman, K.J., Walt, S.J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M.H. van, Brett, M., Haldane, A., Río, J.F. del, Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. and Oliphant, T.E., 2020. Array programming with NumPy. *Nature* [Online], 585(7825), pp.357–362. Available from: `https://doi.org/10.1038/s41586-020-2649-2`.

Hashmi, N., Butt, N.A. and Iqbal, M., 2013. Customer churn prediction in telecommunication a decade review and classification. *International journal of computer science issues (ijcsi)*, 10(5), p.271.

Hossin, M. and Sulaiman, M.N., 2015. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2), p.1.

Huang, B., Kechadi, M.T. and Buckley, B., 2012. Customer churn prediction in telecommunications. *Expert systems with applications*, 39(1), pp.1414–1425.

Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. *Computing in science & engineering* [Online], 9(3), pp.90–95. Available from: `https://doi.org/10.1109/MCSE.2007.55`.

Jain, H., Khunteta, A. and Srivastava, S., 2021. Telecom churn prediction and used techniques, datasets and performance measures: a review. *Telecommunication systems*, 76(4), pp.613–630.

Jiang, D., Huang, J. and Zhang, Y., 2013. The cross-validated auc for mcp-logistic regression with high-dimensional data. *Statistical methods in medical research*, 22(5), pp.505–518.

Kartini, D., Nugrahadi, D.T., Farmadi, A. et al., 2021. Hyperparameter tuning using gridsearchcv on the comparison of the activation function of the elm method to the classification of pneumonia in toddlers. *2021 4th international conference of computer and informatics engineering (ic2ie)*. IEEE, pp.390–395.

Keramati, A., Ghaneei, H. and Mirmohammadi, S.M., 2016. Developing a prediction model for customer churn from electronic banking services using data mining. *Financial innovation*, 2(1), pp.1–13.

Kuhle, S., Maguire, B., Zhang, H., Hamilton, D., Allen, A.C., Joseph, K. and Allen, V.M., 2018. Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: a retrospective cohort study. *Bmc pregnancy and childbirth*, 18(1), pp.1–9.

Lalwani, P., Mishra, M.K., Chadha, J.S. and Sethi, P., 2022. Customer churn prediction system: a machine learning approach. *Computing*, 104(2), pp.271–294.

Louppe, G., 2014. Understanding random forests: From theory to practice. *arxiv preprint arxiv:1407.7502*.

Macedo, F., Oliveira, M.R., Pacheco, A. and Valadas, R., 2019. Theoretical foundations of forward feature selection methods based on mutual information. *Neurocomputing*, 325, pp.67–89.

Maldonado, S. and López, J., 2018. Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for svm classification. *Applied soft computing*, 67, pp.94–105.

McKinney Wes, 2010. Data Structures for Statistical Computing in Python [Online]. In: Stéfan van der Walt and Jarrod Millman, eds. *Proceedings of the 9th Python in Science Conference*. pp.56 – 61. Available from: `https://doi.org/10.25080/Majora-92bf1922-00a`.

Murphy, K.P., 2012. *Machine learning: a probabilistic perspective*. MIT press.

Narin, A., Isler, Y. and Ozer, M., 2014. Investigating the performance improvement of hrv indices in chf using feature selection methods based on backward elimination and statistical significance. *Computers in biology and medicine*, 45, pp.72–79.

Oshiro, T.M., Perez, P.S. and Baranauskas, J.A., 2012. How many trees in a random forest? *International workshop on machine learning and data mining in pattern recognition*. Springer, pp.154–168.

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

Peker, S., Kocyigit, A. and Eren, P.E., 2017. A hybrid approach for predicting customers' individual purchase behavior. *Kybernetes*.

Peng, A.Y., Sing Koh, Y., Riddle, P. and Pfahringer, B., 2018. Using supervised pretraining to improve generalization of neural networks on binary classification problems. *Joint european conference on machine learning and knowledge discovery in databases*. Springer, pp.410–425.

Porkodi, R., 2014. Comparison of filter based feature selection algorithms: an overview. *Int. j. innov. res. technol. sci*, 2(2), pp.108–113.

Ranstam, J. and Cook, J., 2018. Lasso regression. *Journal of british surgery*, 105(10), pp.1348–1348.

Savas, C. and Dovis, F., 2019. The impact of different kernel functions on the performance of scintillation detection based on support vector machines. *Sensors*, 19(23), p.5219.

Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V. and Alonso-Betanzos, A., 2017. Ensemble feature selection: homogeneous and heterogeneous approaches. *Knowledge-based systems*, 118, pp.124–139.

Singh, R. and Khan, I.A., 2012. An approach to increase customer retention and loyalty in b2c world. *International journal of scientific and research publications*, 2(6), pp.1–5.

Śniegula, A., Poniszewska-Marańda, A. and Popović, M., 2019. Study of machine learning methods for customer churn prediction in telecommunication company. *Proceedings of the 21st international conference on information integration and web-based applications & services*. pp.640–644.

Sperandei, S., 2014. Understanding logistic regression analysis. *Biochemia medica*, 24(1), pp.12–18.

Suthaharan, S., 2016. Support vector machine. *Machine learning models and algorithms for big data classification*. Springer, pp.207–235.

Tang, J., Alelyani, S. and Liu, H., 2014. Feature selection for classification: A review. *Data classification: Algorithms and applications*, p.37.

Tavassoli, S. and Koosha, H., 2021. Hybrid ensemble learning approaches to customer churn prediction. *Kybernetes*.

Torlay, L., Perrone-Bertolotti, M., Thomas, E. and Baciu, M., 2017. Machine learning–xgboost analysis of language networks to classify patients with epilepsy. *Brain informatics*, 4(3), pp.159–169.

Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G. and Chatzisavvas, K.C., 2015. A comparison

of machine learning techniques for customer churn prediction. *Simulation modelling practice and theory*, 55, pp.1–9.

Waskom, M.L., 2021. seaborn: statistical data visualization. *Journal of open source software* [Online], 6(60), p.3021. Available from: `https://doi.org/10.21105/joss.03021`.

Wiatowski, T. and Bölcskei, H., 2017. A mathematical theory of deep convolutional neural networks for feature extraction. *Ieee transactions on information theory*, 64(3), pp.1845–1866.

Wieringen, W.N. van, 2015. Lecture notes on ridge regression. *arxiv preprint arxiv:1509.09169*.

Wong, T.T., 2015. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern recognition*, 48(9), pp.2839–2846.

Wong, T.T. and Yeh, P.Y., 2019. Reliable accuracy estimates from k-fold cross validation. *Ieee transactions on knowledge and data engineering*, 32(8), pp.1586–1594.

Wu, X. and Meng, S., 2016. E-commerce customer churn prediction based on improved smote and adaboost. *2016 13th international conference on service systems and service management (icsssm)*. IEEE, pp.1–5.

Xiahou, X. and Harada, Y., 2022. B2c e-commerce customer churn prediction based on k-means and svm. *Journal of theoretical and applied electronic commerce research*, 17(2), pp.458–475.

Yan, K. and Zhang, D., 2015. Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Sensors and actuators b: Chemical*, 212, pp.353–363.

Yin, Z. and Hou, J., 2016. Recent advances on svm based fault diagnosis and process monitoring in complicated industrial processes. *Neurocomputing*, 174, pp.643–650.

Yu, X., Guo, S., Guo, J. and Huang, X., 2011. An extended support vector machine forecasting framework for customer churn in e-commerce. *Expert systems with applications*, 38(3), pp.1425–1430.

ZHUANG, Y., 2018. Research on e-commerce customer churn prediction based on improved value model and xg-boost algorithm. *Management science and engineering*, 12(3), pp.51–56.

Appendix

A link to the github repository containing all code from this project is provided below:

code