UNIVERSITY OF CAPE TOWN

Investigating the relationship between mobile network performance metrics and customer satisfaction

by

Louwrens Labuschagne

A thesis submitted in partial fulfillment for the
degree of Masters in Data Science

in the
Statistics
Department of Science

June 2019

Declaration of Authorship

I, Louwrens Labuschagne, declare that this thesis titled, 'Investigating the relationship between mobile network performance metrics and customer satisfaction' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed by candidate

Signed: _____

Date: _____

UNIVERSITY OF CAPE TOWN

Abstract

Statistics
Department of Science

Masters in Data Science

by Louwrens Labuschagne

Fixed and mobile communication service providers (CSPs) are facing fierce competition among each other. In a globally saturated market, the primary differentiator between CSPs has become customer satisfaction, typically measured by the Net Promoter Score (NPS) for a subscriber. The NPS is the answer to the question: "How likely is it that you will recommend this product/company to a friend or colleague?" The responses range from 0 representing not at all likely to 10 representing extremely likely. In this thesis, we aim to identify which, if any, network performance metrics contribute to subscriber satisfaction. In particular, we investigate the relationship between the NPS survey results and 11 network performance metrics of the respondents of a major mobile operator in South Africa. We identify the most influential performance metrics by fitting both linear and non-linear statistical models to the February 2018 survey dataset and test the models on the June 2018 dataset. We find that metrics such as Call Drop Rate, Call Setup Failure Rate, Call Duration and Server Setup Latency are consistently selected as significant features in models of NPS prediction. Nevertheless we find that all the tested statistical and machine learning models, whether linear or non-linear, are poor predictors of NPS scores in a month, when only the network performance metrics in the same month are provided. This suggests that either NPS is driven primarily by other factors (such as customer service interactions at branches and contact centres) or are determined by historical network performance over multiple months.

Contents

List of Figures

List of Tables

Abbreviations

CSP	**C**ommunication **S**ervice **P**roviders
CEM	**C**ustomer **E**xperience **M**anagement
NPS	**N**et **P**romoter **S**core
OR	**O**perational **R**esearch
AUC	**A**era **U**nder the **C**urve
ROC	**R**eceiver **O**perator **C**urve
MSE	**M**ean **S**quared **E**rror
MAE	**M**ean **A**bsolute **E**rror
RMSE	**R**oot **M**ean **S**quared **E**rror
RSS	**R**esidual **S**um of **S**quares
MLE	**M**aximum **L**ikelihood **E**stimation
NLL	**N**egative **L**og **L**ikelihood
SLR	**S**imple **L**inear **R**egression
MLR	**M**ultiple **L**inear **R**egression
CART	**C**lassification **A**nd **R**egression **T**ree
Bagging	**B**ootstrap **Agg**regat**ing**
GSM	**G**lobal **S**ystem for **M**obile communication
GPRS	**G**eneral **P**acket **R**adio **S**ervices
EDGE	**E**nhanced **D**ata rates for **G**SM **E**volution
CS	**C**ircuit **S**witched
PS	**P**acket **S**witched
SMS	**S**hort **M**essage **S**ervice
1G	**F**irst **G**eneration wireless telecommunications technology
2G	**S**econd **G**eneration wireless telecommunications technology
3G	**T**hird **G**eneration wireless telecommunications technology

4G	**F**ourth **G**eneration wireless telecommunications technology
LTE	**L**ong **T**erm **E**volution
3GPP	**3**rd **G**eneration **P**artnership **P**roject
OSI	**O**pen **S**ystems **I**nterconnection
ISO	**I**nternational **O**rganisation for **S**tandardisation
KDE	**K**ernel **D**ensity **E**stimation
RTT	**R**ound-**T**rip **T**ime
PDP	**P**acket **D**ata **P**rotocol
CDR	**C**all **D**rop **R**ate
SFR	**C**all **S**etup **F**ailure **R**ate
MO	**M**obile **O**riginating
MT	**M**obile **T**erminating
SR	**S**uccess **R**ate
KPI	**K**ey **P**erformance **I**ndicators
VOI	**Voi**ce
BER	**Be**a**r**er

Chapter 1

Introduction

Fixed and mobile communication service providers (CSPs) are facing fierce competition amongst each other, trying to gain every bit of market share they can in a globally saturated market. Increased complexities of new technologies, higher subscriber requirements, paired with an increase in economic pressure is forcing CSPs to go to greater lengths to obtain new subscribers and even greater lengths to retain their current subscriber base.

To improve the experience of their subscribers, CSPs first need to find an adequate measure of the current experience of their subscribers. To do this CSPs need to come to grips with what matters most to their subscribers, followed by feeding this knowledge back into the organisation to make the required targeted investments and take action that aims to improve the subscriber experience (Spiess et al. (2014)). Customer Experience Management (CEM) is the umbrella term given to the domain involved in extracting and acting upon information gathered from subscribers to improve their experience.

A metric to measure and manage the experience of the subscriber base of an organisation was proposed by Reichheld (2003) in an article published in the Harvard Business Review titled: *The one number you need to grow.* In the article Reichheld suggested to do away with long, complicated customer satisfaction surveys and he proposed the *Net Promoter Score*, abbreviated as *NPS*. The use of NPS as a metric to measure customer satisfaction and loyalty within organisations has gained much traction since its inception in 2003, and many organisations have adopted this as one of their main measures of subscriber loyalty, happiness and satisfaction.

1.1 Net Promoter Score

The NPS of a company is calculated based on the survey responses of subscribers to the following question:

> Based on your interaction with company *X*, how likely are you to recommend us to a friend or colleague? On a scale from 0 to 10, were 0 represents not at all likely and 10 represents extremely likely.

The Net Promoter Score for the organisation is then calculated by taking the percentage of subscribers who are Promoters (responses between 9-10) and subtracting the percentage who are Detractors (responses between 0-6). Due to the simplicity of the survey - only one question, companies have turned to use the NPS survey results to try and explain churn, subscriber satisfaction and loyalty.

In summary, NPS works by quantifying if the service provided results in repeat business by measuring the likelihood that a subscriber would recommend the service to friends and family (Hamilton et al. (2014)).

As the NPS survey is quick and easy - there is usually many responses gathered in an NPS survey, compared to previously, longer more tedious surveys. These responses on their own can tell upper management that the subscriber base in a particular region might be unsatisfied, but it does not tell management why subscribers are unsatisfied. NPS then is usually a symptom of subscriber loyalty, not the cause.

1.2 Customer Experience Management

Today CSPs can acquire new subscribers and hold on to existing subscribers by focussing on improving subscriber experience, also known as Customer Experience Management, abbreviated as *CEM*. In the past, differentiating factors for CSPs included higher bandwidth, device innovations, unique services and mobile coverage. However with the advent of fast broadband, with technologies such as 3G, 4G and 5G in the near future, together with more devices supporting features like international video calling, instant messaging and video streaming out of the box using over the top (OTT) applications like YouTube and WhatsApp - customer experience has become a key differentiator.

It makes sense that customer experience is of utmost importance when you consider that the majority of dimensions of a CSP: network reliability, coverage, care, provisioning and billing all have an impact on the customer's perception of their service provider (Spiess et al. (2014)).

1.3 Interfaces, Services, Predictors and Models

We use subscriber network performance data gathered from the A, IuCS, Gb, IuPS and Gn interfaces (discussed in Section 3.3.3) for a large mobile operator. Across these five interfaces we have 11 performance metrics bundled into 3 services viz. VOI, BER and GN (discussed in Section 4.1.3). Four of the 11 features come from the voice (VOI) service: *Call Drop Rate*, *Call Setup Failure Rate*, *Mobile Originating and Terminating Call Duration*. Four features come from the bearer (BER) service: *Bearer Attach Duration*, *PDP Activation Duration*, *Bearer Attach Success Rate*, *PDP Activation Success Rate*, and three features come from the Gn (GN) service: *PDP Create Success Rate*, *Client Setup Time* and *Server Setup Time*.

We fit a linear regression, logistic regression, regression and classification decision trees and regression and classification random forests to describe the relationship between these 11 predictors and a subscriber's NPS response. For the regression approach we use the numeric NPS score (0 to 10), and for the classification approach, we cast subscribers to promoters if their NPS score is 9 or higher, and to detractors, if their score is less than 6, as per the definition of NPS. In both regression and classification approaches, we disregard subscribers with an NPS response of 7 or 8 as these subscribers are deemed neutral and we are interested in investigating what makes a subscriber a promoter or detractor.

We evaluate the regression models based on their Mean Squared Error (MSE) and adjusted R-squared (R^2_{adj}) scores and evaluate the classification models based on their F1-scores and Area Under the Curve (AUC) performance. In our modelling process, we are not solely interested in the best model based on these metrics, but rather what insights we can gain into the relationship between network performance metrics and subscriber satisfaction.

To gain insights the relationships between our dependent and independent variables we investigate how the models perform on the entire February dataset with all 11 features, followed by looking at how each feature performs individually, and finally using the features within their respective services to highly any within service interaction. As a final benchmark, we evaluate how the February trained models perform on the June survey to see which models generalise best.

1.4 Goals and Layout

The datasets in this thesis come from probing the live network of a large telecommunication organisation in South Africa. This thesis investigates whether using CEM data has any merit in describing and inferring subscriber NPS responses. Simply put, this thesis investigates whether network performance data (for example latency, number of dropped calls, data throughput) can give us an idea regarding what makes happy subscribers happy, but more importantly what makes unhappy subscribers unhappy. We take an operational research approach to model building, and we value insights about the data more than absolute accuracy.

In Chapter 2 we discuss the workings of linear and logistic regression, followed by how ensemble tree models partition a regression or classification space; also we discuss the different model evaluation metrics used in this thesis.

In Chapter 3 we give a broad overview of how NPS, CEM and how mobile technologies have evolved over the years. Further, we give a quick overview of how protocols and interfaces work in a telecommunications network and explain where we probe to source our features.

In Chapter 4 we visualise the distributions of our dependent variable (NPS survey responses) and independent variables (network performance metrics) and give some intuition into how these features relate to what subscribers might experience.

In Chapter 5 we explore 4 types of models viz. linear regression, logistic regression, tree-based regression and tree-based classification and discuss insights gained from each of these models.

In Chapter 6 we evaluate the performance of all the models on the February 2018 and June 2018 survey datasets and present some concluding arguments and recommendations for future work.

Chapter 2

Modelling Algorithms

Since the paper *ImageNet Classification with Deep Convolutional Neural Networks*, Krizhevsky et al. (2012), artificial neural networks in a variety of flavours have received much attention in machine learning literature. Moreover, they have earned the title for being one of the best universal function approximators available, as shown in Chen & Chen (1995). Neural networks can learn the mapping between any number of dependent and independent variables, or as Marcus (2018) states:

> In a world with infinite data and infinite computational resources, there might be little need for any other (than neural network) technique.

In this thesis, however, we are far less concerned about the accuracy with which we can predict whether a subscriber is a promoter or a detractor - we accept this is possible using neural networks. Instead, we would like to investigate whether poor network performance leads to subscribers being more prone to be a detractor - and more actionable to upper management - which subscriber interaction points lead *most* to subscribers being detractors on average. Management can then focus on these critical areas in an attempt to improve customer satisfaction.

We evaluate 4 different types of models in Chapter 4 and investigate the features that are the most significant within each of these models using different metrics. We compare the models on their relative accuracy and give more weight to the insights generated by more accurate models as these models fit the data better. By dissecting the features of the more accurate models, we hope to uncover insights as to which dimensions are most indicative of happy or unhappy subscribers.

Our modelling approach stems more from the operational research (OR) domain rather than the machine learning domain. In the OR domain, model building is more with

the intention to understand the underlying factors compared to the machine learning domain which uses an abundance of data to predict the correct outcome accurately. Pidd (2004) defines OR modelling principles that we employ in our investigation, some of which are:

- Model simple, think complicated.
- Be parsimonious; start small, then add.
- Divide and conquer - avoid mega models.

We are interested here in the causal relationship between a dependent variable (customer loyalty as indicated by a NPS survey response) and some collection of predictor variables (various network performance indicators, such as latency, call drop rate, etc.) We use statistical modelling as it encompasses the set of processes that are used to estimate and quantify the relationship between dependent and independent variables. We define a statistical model as a stripped-down representation of a particular real-world state or process. Or as Levins (1966) defines a statistical model:

> A model is neither a hypothesis nor a theory. Unlike scientific hypotheses, a model is not verifiable directly by an experiment. For all models of true or false, the validation of a model is not that it is "true" but that it generates good testable hypotheses relevant to important problems.

We use statistical modelling to investigate the relationship between one or more dependent variables (also called explained variables, response variables or predicted variables, usually denoted by y) and a collection of independent variables (also called the predictors, explanatory variables or control variables, usually denoted by $x_1, x_2, \ldots, x_n$). In this section, we give a broad overview of linear regression models, logistic regression models and ensemble tree models and discuss common metrics used to evaluate and compare these models.

2.1 Linear regression

Linear regression is an approach to model the relationship between the dependent and one or more independent variables. It assumes a linear relationship amongst the parameters in the model and is the most well understood and the longest studied statistical model to extract and investigate underlying causal relationships. According to Yan & Su (2009),

> Regression analysis ... is probably one of the oldest topics in the area of mathematical statistics dating back to about two hundred years ago. The earliest form of linear regression was the least squares method, which was published by Legendre in 1805, and by Gauss in 1809.

Yan & Su (2009) defines three types of regressions. The first being the simple linear regression (SLR) which is used for modelling the relationship between 2 variables, *one* dependent and *one* independent. The simple linear regression model usually comes in the form of Equation 2.1.

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{2.1}$$

where y is the dependent variable, β_0 is the intercept (also sometimes called the bias), β_1 is the slope or gradient of the regression line, x is the independent variable and ε is the random error. Usually it is assumed that ε is normally distributed, $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$, which is equivalent to saying $E\left[\varepsilon\right] = 0$ and $Var\left[\varepsilon\right] = \sigma^2$. In linear regression evaluating how the random error is distributed gives an indication of how well the data is being explained by the model.

The second type of regression Yan & Su (2009) defines, is the multiple linear regression, which assumes that the dependent variable is a linear combination of the model parameters and allows for multiple (p) independent variables. We define multiple linear regressions in the form of Equation 2.2.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon \tag{2.2}$$

In Equation 2.2 y is still referred to as the dependent variable, β_0 is still referred to as the intercept, however $\beta_1 \ldots \beta_p$ are now referred to as the regression coefficients and $x_1 \ldots x_p$ are referred to as the independent variables in the model. Here, as in the simple regression case, ε is assumed to be normally distributed around 0 with a variance of σ^2. SLR is used to explore the relationship between *one* dependent and *one* independent variable, where MLR focuses on the linear relationship between *one* dependent and *multiple* independent variables.

MLR involves more issues than SLR viz. collinearity among the independent variables, visualising regression results, variance inflation, as well as a higher difficulty detected outliers and influential observations. In general each slope $\beta_j, j = 1 \ldots p$ in Equation 2.2 is interpreted as the change in the average value of y for one unit change in x_j, holding all the other predictors fixed.

The third and last regression approaches Yan & Su (2009) describes is non-linear regression, which assumes that the relationship between the dependent and independent variables is not linear in the regression parameters. As we turn to ensemble methods to describe non-linear relationships in this paper, we refer to this type of linear regression here for completeness only. An example of such a model is the growth model defined in Equation 2.3.

$$y = \frac{\alpha}{1 + e^{\beta t}} + \varepsilon \tag{2.3}$$

where y is the growth of a particular organism as a function of time t, α and β are model parameters, and ϵ is the random error.

2.1.1 Linear Regression Estimations

If we let $\hat{\boldsymbol{\beta}}$ be the vector of all the beta coefficients, i.e. $\hat{\boldsymbol{\beta}} = \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_p \end{bmatrix}$, we can then estimate the model parameters ($\hat{\boldsymbol{\beta}}$) by choosing the values of $\hat{\boldsymbol{\beta}}$ that minimises a loss function. In regressions, the *residual sum of squares* (RSS) is widely used and is defined as in Equation 2.4.

$$RSS\left(\widehat{\beta}\right) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i - \ldots - \hat{\beta}_p x_i\right)^2 \tag{2.4}$$

A more commonly used loss metric is the *mean squared error* (MSE) and is closely related to RSS. MSE goes one step further and accounts for the number of observations, n, used to evaluate the metric. Equation 2.5 defines how the MSE is calculated.

$$MSE\left(\hat{\boldsymbol{\beta}}\right) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i - \ldots - \hat{\beta}_p x_i\right)^2 \tag{2.5}$$

Intuitively minimising the MSE amounts to choosing a regression line such that the predictions, $\hat{y}_i$, are as close as possible to the real observations y_i. To quantify the uncertainty in the estimations, we use the *standard error* of the estimates, which proxies for the standard deviation of the sampling distribution of the estimates (Everitt (2005)).

Here a low standard error implies that our estimate is reliable since it does not vary much between samples, whereas a high standard error means that our estimate is unreliable since we are likely to get something very different if we had to take another sample.

Suppose we have a value which has been sampled with an unbiased normally distributed error, Figure 2.1 shows the proportion of samples that would fall between 0, 1, 2, and 3 standard deviations above and below the actual value.

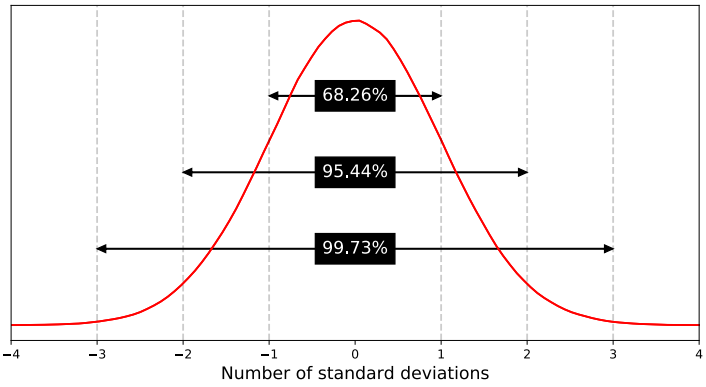


FIGURE 2.1: The percentage of samples (in black boxes) within 1, 2 and 3 standard deviations for a sample from an unbiased normally distributed error (Six-Sigma-Material (2018)).

Quantifying the uncertainty in our estimates enables us to test hypotheses about population parameters when we are trying to determine whether there is a relationship between X and Y in the population. For the SLR case if $\beta_1 = 0$, there is no relationship in the population. Suppose we had an estimated coefficient of $\hat{\beta} = 0.03$, would it be likely for us to observe such a point estimate if $\beta_1 = 0$? Whether or not we would have observed such a point estimate by chance depends how far β_1 is from zero and on how reliable our estimate is.

The Hypothesis testing framework provides a method to evaluate whether the estimated beta, $\hat{\beta}$, is an accurate representation of the population beta, β or whether we obtained the estimation by chance in the given sample we evaluated.

We can define a *Null* hypothesis, $H_0 : \beta_1 = 0$, which accounts for the case where there is no relationship, the alternative hypothesis, $H_1 : \beta_1 \neq 0$, then suggests that there is some relationship. To determine whether there is evidence against H_0, we first compute the t-statistic as in Equation 2.6, where SE denotes the standard error. In Equation 2.6 large values of $|t|$ would provide evidence against H_0 and small values evidence for H_0.

$$t = \frac{\hat{\beta}_1 - 0}{SE\left(\hat{\beta}_1\right)} \tag{2.6}$$

From the t-statistic, we can then compute the p-value, defined as *the probability of obtaining an estimate at least as extreme as ours if* H_0 *is true* and is shown visually in Figure 2.2.

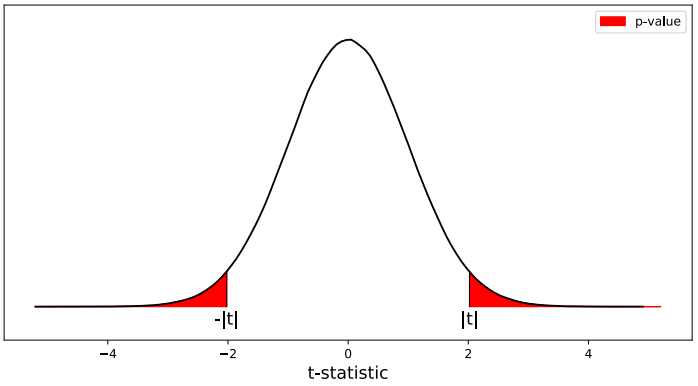


FIGURE 2.2: The relationship between the t-statistic and the p-value, where the p-value is the shaded red area, while the t-statistic is the value on the x-axis.

McShane et al. (2018) explains that in science publishing and as well as in other areas of research the status quo in history has been for any result to get into the academic literature, first it was required to have a p-value that surpasses the 0.05 threshold. If we had a small p-value (e.g. < 0.05) it would suggest that our observed estimate is highly unlikely if H_0 is true. We would then reject $H_0 : \beta_1 = 0$ and conclude that there is a relationship between X and Y in the population. If we had a significant p-value, it would suggest that the non-zero value of $\hat{\beta}_1$ we observed is likely to arise by chance even if $\beta_1 = 0$, thus we do not have evidence to reject the null hypothesis.

Although accepting and rejecting hypothesis significantly at a p-value of 0.05 has always been around, McShane et al. (2018) argues that the 0.05 threshold is somewhat arbitrary. McShane et al. (2018) suggests that the 0.05 threshold should not be the only cut-off requirement for publishing papers, but rather a more holistic stance of the evidence at hand should be taken, which includes the consideration of the neglected factors. In this thesis, we share a similar view, and in our model analysis, we too use the p-values generated by estimations as just another indicator, rather than being the determining factor for significance.

We can expand the idea of hypothesis testing from the SLR to the MLR model by determining if any of the predictors are related to the outcome in the population (Lomax & Hahs-Vaughn (2013)). This would lead us to define the following hypotheses test:

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$$
$$H_1 : \text{At least one } \beta_j \text{ is non-zero}$$

Examining the F-statistic for MLR allows one to compare the joint effect of all the variables together and whether all the variable together are related to the response variable. The F statistic and its the corresponding p-value indicates as to how well the estimated values perform in rejecting the Null hypothesis H_0. If we had a F-statistic which had a p-value less than say 0.05, we would conclude that at least one of the predictors is related to the dependent variable in the population.

2.1.2 Assessing Model Accuracy

We asses the goodness-of-fit of a regression model by examining the *coefficient of determination* or R^2 statistic. We interpret the R^2 statistic as the proportion of variation in Y that is explained by X (Draper & Smith (2014)). Therefore it is bounded between 0 and 1 where values closer to 1 indicate a better fit.

A caveat with the *coefficient of determination*, Draper & Smith (2014), is that the metric increases as we add more predictors to the model, even if the additional predictors explain minimal extra variation, we would rather keep the model simple and omit predictors that are not useful for predicting the outcome.

The adjusted R^2 statistic, R^2_{adj}, accounts for the added number of predictors in the model and is defined in Equation 2.7, where p is the total number of explanatory variables in the model (not including the constant term), and n is the sample size.

$$R^2{}_{adj} = 1 - \left(1 - R^2\right) \frac{n-1}{n-p-1} \tag{2.7}$$

2.1.3 Logistic Regression

Whilst linear regression deals with the case where we have a continuous response variable, logistic regression is used when we have a finite set of discrete categorical outcomes. For example, consider a binary outcome $Y \in \{0, 1\}$ and let the conditional probability that $Y = 1$ given a set of predictor values X be defined as in Equation 2.8.

$$f(X) = E[Y|X] = P(Y = 1|X) \tag{2.8}$$

For classification, our goal is to develop a statistical model to estimate the probability function in Equation 2.8. We can start by using the same linear form as we had with linear regression, shown in Equation 2.9.

$$f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p \tag{2.9}$$

The problem with this approach, however, is that the $f(X)$ in Equation 2.9 is not bounded and can take on any real value. To overcome this limitation logistic regression employs the logistic function shown in Equation 2.10. Using the logistic function, Hosmer et al. (2013) ensures that the outcome lies between 0 and 1, which means we can treat the outcome as a probability and we can define a cut-off probability above which $Y = 1$, otherwise $Y = 0$. We show the different ranges for linear and logistic regression in Figure 2.3.

$$f(x) = \frac{e^x}{1 + e^x} \tag{2.10}$$

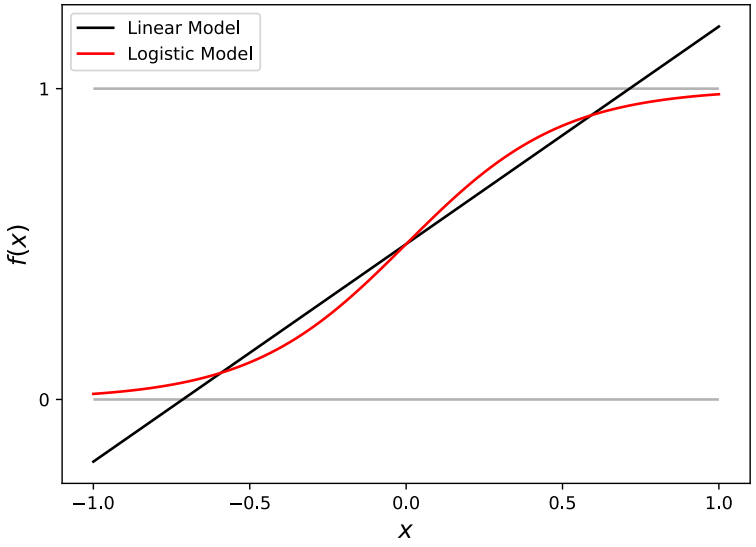


FIGURE 2.3: The unbounded range for linear (black) regression and the bounded range for logistic regression (red), Ldapwiki (2018).

The logistic regression model form for the multiple logistic regression is as shown in Equation 2.11.

$$f(X) = P(Y = 1|X) = \frac{e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p}} \tag{2.11}$$

As the relationship between $f(X)$ and X is not linear any more, the β parameters cannot be interpreted as gradients any more. If we rearrange Equation 2.11 we can find the odds that $Y = 1$ given that $X = x$ shown in Equation 2.12. We can then interpret increasing X_j by one unit as increasing the odds of $Y = 1$ by a multiple of e^{β_j} holding all other predictors constant.

$$\frac{P\left(Y=1|X=x\right)}{1-P\left(Y=1|X=x\right)} = e^{\beta_0+\beta_1 x_1+\ldots+\beta_p x_p} \tag{2.12}$$

For logistic regression, the z-statistic plays the same role as the t-statistic in regression, Hosmer et al. (2013), and provides evidence to support whether or not our estimated $\hat{\boldsymbol{\beta}}$'s are representative of the population $\boldsymbol{\beta}$'s. The equation to calculate the z-statistic is the same as in Equation 2.6 where large values of $|z|$ would provide evidence against H_0 and small values evidence for H_0.

One optimisation strategy for logistic regression is a maximum likelihood estimation (MLE) approach (Hosmer et al. (2013)). The labels that we are predicting are binary, and the output of our logistic regression function is the probability that the label is one. This means that we can interpret each label as a Bernoulli random variable: $Y \sim Ber\left(p\right)$ where $p = \sigma\left(\beta^T x\right)$.

To obtain the likelihood for a binary logistic regressor, we define the probability of one data point as in Equation 2.13 as the probability of one data point is in the form of the probability mass function of a Bernoulli distribution.

$$P\left(Y=y|X=x\right) = \sigma\left(\beta^T x\right)^y \times \left[1-\sigma\left(\beta^T x\right)^{1-y}\right] \tag{2.13}$$

Now that we know the probability mass function of one point, we can express the likelihood of all the data as in Equation 2.14.

$$L\left(\beta\right) = \prod_{i=1}^{n} P\left(Y=y^{(i)}|X=x^{(i)}\right) \tag{2.14}$$

Substituting the likelihood of a Bernoulli distribution we get Equation 2.15.

$$L\left(\beta\right) = \prod_{i=1}^{n} \sigma\left(\beta^T x^{(i)}\right)^{y(i)} \times \left[1-\sigma\left(\beta^T x^{(i)}\right)^{(1-y(i))}\right] \tag{2.15}$$

Taking the log of Equation 2.15 we get the log likelihood define as in Equation 2.16.

$$LL\left(\beta\right) = \sum_{i=1}^{n} y^{(i)} log\left[\sigma\left(\beta^T x^{(i)}\right)\right] + \left[1 - y^{(i)}\right] log\left[1 - \sigma\left(\beta^T x^{(i)}\right)\right] \tag{2.16}$$

2.1.4 Ridge Regression

As many of the features are correlated, we use ridge linear and logistic regression to identify features that are more descriptive than others. Ridge regression is just an extension of linear regression, in essence, it is a regularised linear regression model. The α parameter is a scalar that needs to be learned as well, using cross-validation (Ng (2004)).

An essential difference between ridge and lasso regression is that ridge regression enforces the β coefficients to be lower, but it does not force them to be zero - unlike lasso regression. In other words, we do not get rid of unnecessary features but rather minimise their impact on the trained model.

Suppose we use MSE as defined in Equation 2.17 as our loss function for a linear regression model. Ridge regression simply adds a penalisation term weighted by a positive scaler, α to transform Equation 2.17 into Equation 2.18.

$$MSE\left(\widehat{\boldsymbol{\beta}}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \left(\hat{\beta}_0 + \beta\boldsymbol{x}\right)\right)^2 \tag{2.17}$$

$$MSE_{ridge}\left(\widehat{\boldsymbol{\beta}}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \left(\hat{\beta}_0 + \beta\boldsymbol{x}\right)\right)^2 + \alpha\sum_{j=1}^{p}{\beta^2}_j \tag{2.18}$$

The impact of the regularisation term in Equation 2.18 is when $\alpha = 0$, there is no penalty and minimising this loss function returns the least squares estimates. As the regularisation term α gets larger, the regression coefficients are shrunk towards zero to minimise the penalised loss function, while we do not penalise the intercept term.

As we penalise the β coefficients for size, optimisation favours pushing one or more coefficients of correlated variables towards zero, only retaining one feature from correlated features - the feature that best minimises the chosen loss function.

The penalty term in the loss function can be extended to the binary logistic regression by adding the same $\alpha\sum_{j=1}^{p}{\beta^2}_j$ term to the loss function. In the case of binary logistic regression where loss function is the negative log likelihood defined as in Equation 2.19, ridge regression will have the loss function as define in Equation 2.20, with $\alpha > 0$.

$$NLL(\beta) = -\sum_{i=1}^{n} y^{(i)} log\left[\sigma\left(\beta^T x^{(i)}\right)\right] + \left[1 - y^{(i)}\right] log\left[1 - \sigma\left(\beta^T x^{(i)}\right)\right] \tag{2.19}$$

$$L(\beta) = NLL(\beta) + \alpha \sum_{i=1}^{p} {\beta_i}^2 \tag{2.20}$$

2.1.5 Summary

Linear regression aims to model the relationship between one dependent variable and one or more independent variable(s). Most commonly the conditional mean, $E[y|\mathbf{x}]$, is assumed to be an affine function[1] expressed as a linear function of $\mathbf{x}$. Linear regression, like other forms of regression analysis, is concerned with the conditional probability distribution of y given $\mathbf{x}$, rather than the joint probability distribution of y and $\mathbf{x}$, which is the domain of multivariate analysis.

Logistic regression is similar to linear regression as it expresses the probability of a category as a linear relationship between the predictor variables and the log odds of a category. It does, however, add the additional step of passing the sum of the predictor variables through the logistic function which can then provide probability values bounded between 0 and 1, rather than a continuous outcome value.

Regression models allow us to isolate the relationship between an outcome and an explanatory variable while other variables are held constant. If there exists an underlying linear relationship between the dependent and independent variables, this regression technique allows us to asses which x_j may or may not have a relationship with y, further it allows us to find which subset of x's contain redundant information about y.

2.2 Decision Trees

Decision trees were originally developed in Quinlan (1986) in an attempt to map subject matter expertise. A decision tree is a non-parametric supervised learning method used for both regression and classification. Intuitively, a decision tree encodes a set of if-else statements. The grown tree resulting from the if-else splits can then be used to gain insight into how the variables related to the output or the grown tree can then be used to predict a target value for a given set of features.

A decision tree is a graph like structure that partitions a feature space into nodes or sub-nodes. At each split the homogeneity of the resultant sub-node is increased, in other

[1] A function between affine spaces which preserves points, straight lines and planes

words, we say the purity of the sub-node increases with respect to the target variable. Decision trees scan through all possible features, then chooses a split that results in the most homogeneous sub-nodes. In Figure 2.4 we show an example of a decision tree deciding whether there are the right conditions to go for a swim.

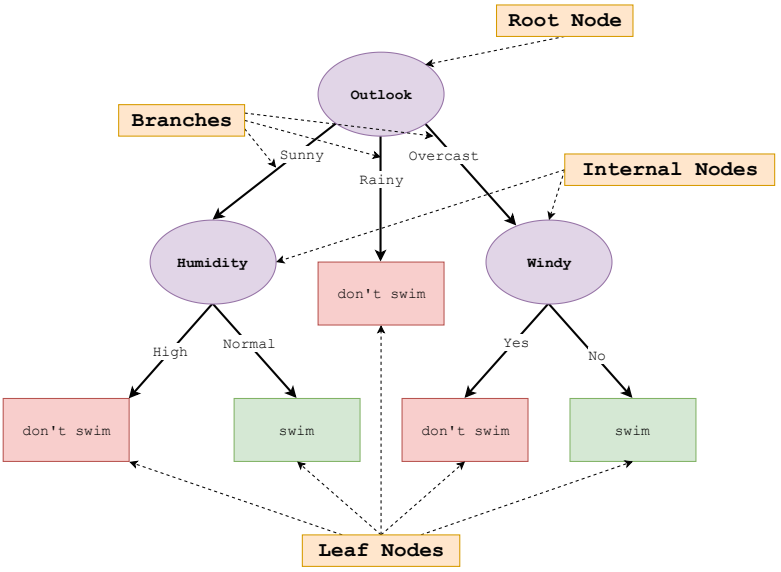


FIGURE 2.4: An example of a decision tree showing the *Root Node*, *Branches*, *Internal Nodes* and *Leaf Nodes* involved when deciding whether to swim or not.

In Figure 2.4 the various components are defined as:

- **Root Node**: Represents the whole sample or population and gets divided into two or more homogeneous sets.
- **Decision or Internal Node**: Where a sub-node splits into more sub-nodes.
- **Leaf or Terminal Nodes**: Nodes that are at the edge of a sub-node and do not split.
- **Pruning**: If we remove sub-nodes from the tree, it is called pruning.
- **Branch or Sub-Tree**: A subsection of the entire tree.
- **Parent or Child Node**: A node split into sub-nodes is called a parent node, whereas all sub-nodes are the children of a parent node.

According to Mazumdar (2018), there exist multiple algorithms to grow a decision tree, these different algorithms are shown in Table 2.1.

Algorithm	**Classification**	**Regression**
ID3	Information Gain	-
C4.5	Information Gain	-
C5.0	Information Gain	-
CART	Gini	Variance Reduction
CHAID	Chi-Squared Test For Independence	Chi-Squared Test For Independence

TABLE 2.1: The five different decision tree growing algorithms.

Though the algorithms vary, all of them employ a *greedy* algorithm that tries to search for the feature split which results in the maximum information gain or splits the data most homogeneously. In this thesis, we use the Classification And Regression Tree (CART) algorithm as this is a general purpose algorithm used for classification and regression.

Growing trees using the CART algorithm comprises of the recursive binary splitting of nodes to find the best split at each node considering all possible splits of all available predictive attributes. The best split is the split that maximises the splitting criterion. In the case of classification, the Gini index is used as the splitting criterion, whereas for regression least squares deviation is used.

2.2.1 Regression Trees

Using the CART algorithm, we can construct a regression tree in the following way. We let Y be a continuous outcome variable and let $X_1, \ldots, X_p$ be a set of predictor variables. Next, we would like to divide the feature space into J distinct, non-overlapping regions, $R_1, \ldots, R_J$. Further, we would like these J divisions to minimise the residual sum of squares for all the observations in the various regions using Equation 2.21. Here the predicted value, $\hat{y}_{R_j}$, for every observation that falls into R_J is merely the average of the outcome for all observations in R_j.

$$RSS = \sum_{j=1}^{J} \sum_{i:y_i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2 \tag{2.21}$$

As it is computationally infeasible to consider all possible regions, the algorithm uses a top-down greedy algorithm called *recursive binary splitting*. Recursive binary splitting works by first selecting a predictor X_j and cut-point c that splits the current branch

into 2 regions R_1 and R_2. The regions are split at $X_j = t$ which is the split that results in the lowest RSS. In mathematical terms R_1 and R_2 are defined as $R_1 = \{X|X_j < t\}$ and $R_2 = \{X|X_j \geq t\}$ respectively.

Next, the algorithm identifies the next feature X_j and cut-point c that splits both the newly defined regions, R_1 and R_2, with the lowest RSS. Until we meet any of the stopping criteria, the algorithm repeats this process. The minimum number of observations in a leaf node or the maximum depth of the tree are some examples of stopping criteria.

A graphical illustration of how the binary splitting process works is shown in Figure 2.5. Here we have two predictor variables, X_1 and X_2, and in split 1 we split the root node at $X_1 = t_1$ into regions R_1 and R_2. Next in split 2 we split the region R_1 into R_1 and R_2 at $X_2 = t_2$. The process is repeated and illustrates how binary splitting partitions a given feature space into sub regions that maximises a splitting criterion, here RSS.

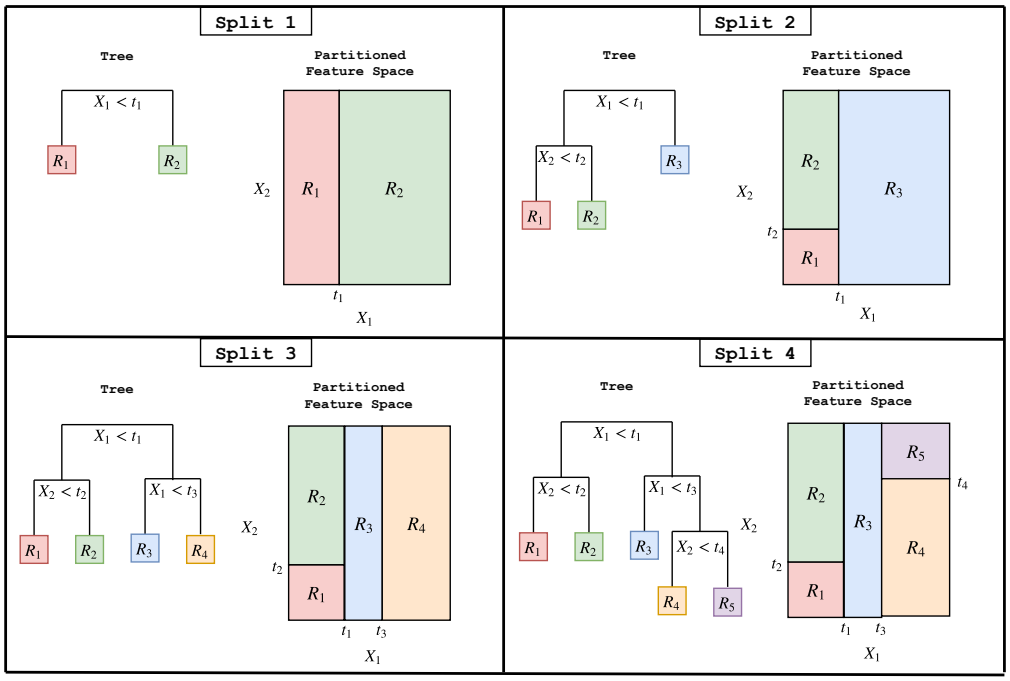


FIGURE 2.5: An example showing 4 consecutive recursive binary splits. Split 1 partitioning the feature space on X_1 at t_1, split 2 splitting on X_2 at t_2, then X_1 at t_3 and finally split 3 splitting on X_4 at t_4.

2.2.2 Classification Trees

The CART algorithm allows classification trees to work using the same greedy, recursive binary splitting principle as regression trees with the only difference being that for classification trees the splitting criterion changes from RSS to using the Gini index.

To illustrate how a classification tree partitions a feature space, consider Y to be a categorical variable $Y \in \{Promotor, Detractor\}$. We still partition the feature space into J distinct, non-overlapping regions, $R_1, \ldots, R_J$, but rather than splitting on the RSS, for a categorical Y, the class proportions in each terminal node give us an indication of the reliability of each classification rule.

To mimic a Bayes classifier, we would typically assign an observation with $X \in R_j$ to the most commonly occurring class in R. If false positives and false negatives have different costs, we could consider an alternative classification threshold. However, splitting on the reduction in classification error does not produce good trees, and we would instead use the Gini Index as a cost function.

To compute the Gini impurity for a set of items with J classes, let p_i be the fraction of items labelled with class i where $i \in \{1, 2, ..., J\}$.

In general the Gini impurity can be calculated as

$$I_{Gini}(p) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = \sum_{i=1}^{J} p_i (1 - p_i) \tag{2.22}$$

Simlifying Equation 2.22 a bit, we can obtain Equation 2.23.

$$I_{Gini}(p) = \sum_{i=1}^{J} (p_i - {p_i}^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} {p_i}^2 = 1 - \sum_{i=1}^{J} {p_i}^2 \tag{2.23}$$

As we would like the leaf nodes to be as homogeneous as possible; ideally, each leaf node would include observations of only *one* outcome category. At each step during tree growth, we choose the split that produces the most significant reduction in the Gini index.

For a binary response variable, where $p_2 = 1 - p_1$, Equation 2.23 reduces to Equation 2.24.

$$I_{Gini}(p) = 2p_1(1 - p_1) \tag{2.24}$$

Figure 2.6 shows Equation 2.24 visually. We see that the Gini index for 2 classes reaches a maximum when the two classes get split fifty-fifty. This property of the Gini index which results in the most homogeneous split at each node is why we prefer it as the splitting criterion in the CART algorithm versus other accuracy metrics such as classification accuracy.

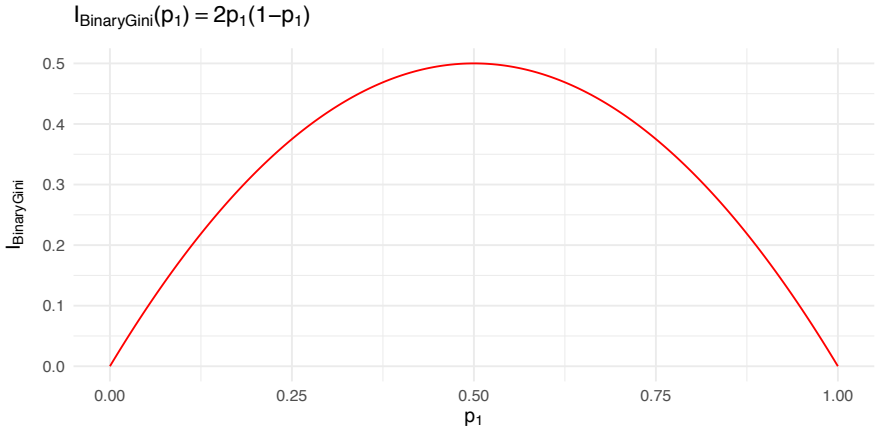


FIGURE 2.6: The Gini index for 2 classes.

2.2.3 Bagging

Although decision trees have a great visual interpretation, as in Figure 2.4, where we can quite intuitively gain an understanding when you would consider to swim. The biggest problem with a single decision tree is that they suffer from high sampling variability. What this means is if we had a different training sample we can and will most likely end up with an entirely different tree. Conversely, low variance procedures, such as linear regression, produce the same results for different samples from the same population.

Bootstrap Aggregating, commonly referred to as *bagging* (Breiman (1996)), is a all purpose procedure for reducing the variance of a statistical model. Bagging exploits the fact that averaging over a set of observations reduces the variance. Consider a set of n independent uncorrelated observations $Z_1, \ldots, Z_n$ each with variance σ^2. The variance of the average across all the observations, $\bar{Z}$ is reduced by $\frac{1}{n}$, in other words $Var\left[\bar{Z}\right] = \frac{\sigma^2}{n}$.

Suppose we had B training sets, we could grow a tree for each training set and use the average prediction of all B trees using an equation as in Equation 2.25.

$$\hat{y}_{ave} = \frac{1}{B}\sum_{b=1}^{B}\hat{y}_b \tag{2.25}$$

The average prediction, $\hat{y}_{ave}$, should have a lower sampling variance than the prediction of any single tree, $\hat{y}_b$. Although we do not have B different samples, we can obtain multiple bootstrap samples by repeatedly taking samples with the same size with replacement from the original sample.

The idea behind bagging then is to grow B trees on B bootstrap samples. For a continuous outcome variable, we take the average of the predictions produced by the B regression trees, whereas, for a categorical outcome, we take the majority vote, the overall prediction for observations then becomes the most commonly occurring category among the B predictions. A significant advantage of bagging is that a big value of B does not lead to overfitting.

We can determine the number of trees needed by assessing the out-of-bag sampling error. We can use the observations not used to grow a bagged tree to calculate an out-of-bag (OOB) error. The OOB error proxies an estimate of the test error, and we predict the outcomes for observation i using each of the bagged trees for which that observation was out-of-bag. We can then use the out-of-sample RSS for regression trees or the classification error for classification trees to choose the best number of trees, where after the OOB error does not improve.

The key advantage of a decision tree is ease of interpretation, as soon as we grow many trees, we lose this interpretability. To gain some insight into the variable importance of each predictor we can calculate for each predictor the amount by which the splitting criterion is improved for each tree and take the overall average over the B trees.

2.2.4 Random Forests

Bagging reduces the sampling variability, $Var[\hat{Z}]$, of predictions by averaging over many trees and for uncorrelated samples, the variance reduces with $\frac{1}{n}$ for every n trees added, as shown in Equation 2.26.

$$Var\left[\bar{Z}\right] = \frac{\sigma^2}{n} \tag{2.26}$$

If the trees, however, are highly correlated, which usually happens when one predictor is dominant in a dataset, it results in all trees containing this predictor. The variance gain is then not as significant as for uncorrelated samples case due to the variance of the average for a set of correlated random variables $Z_1, \ldots, Z_n$ with common variance σ^2 and correlations ρ_{ij} is as shown in Equation 2.27. In Equation 2.27 we see that if the trees are highly correlated the term containing the correlation coefficient, ρ_{ij}, begins to play a significant role.

$$Var\left[\bar{Z}\right] = \frac{\sigma^2}{n} + \frac{2\sigma^2}{n^2}\sum_{i\neq j}\rho_{ij} \tag{2.27}$$

Random forests attempt to improve bootstrap aggregating by decorrelating the trees (Strobl et al. (2009)). Similar to bagging, we construct a random forest by growing B bootstrapped decision trees, but at each split a random sample of $m < p$ variables are selected as split candidates. On average, $\frac{p-m}{p}$ of the splits does not consider a strong predictor variable, leading to decorrelation of the trees, resulting in the average predictions to be less variable between samples and thus more reliable.

When considering all the predictor variables, bagging is a particular case of random forests, i.e. $m = p$. As with bagging, we can gain insights into the importance of each predictor variable across all trees in the forest by calculating for each predictor the amount by which the splitting criterion is improved for each tree and then take the overall average over the B trees.

Using the random forest approach of subsetting the features at each split together with taking bootstrap samples, results in some trees fitting the overall structure of the data well and capturing the global trend, while other trees become "domain experts". The experts can catch subtle nuances that one tree trained on the entire dataset would have missed.

2.2.5 Summary

Decision trees encode a set of if-else statements based on a greedy algorithm that scans through all the features and splits the dataset based on some splitting criterion. For regression problems, this criteria is usually MSE, and the CART algorithm looks for the feature split that results in the highest reduction in MSE. For classification problems, the splitting criterion is usually the Gini Index which is a metric that ensures nodes split most homogeneously.

Bootstrap Aggregating, or bagging, is a general approach to reduce the sampling variance of models that suffer from high sampling variability. The approach creates different bootstrap samples and trains different trees on different samples; this results in models that generalise much better.

Random Forests take a subset of the features to consider at each split; this further reduces sampling variance by decorrelating the trees within the forest. By only showing a subset of features to individual trees, some trees become domain experts and capture subtle nuances within the data.

2.3 Evaluation Metrics

To compare the performance of our estimators we need some common metrics to evaluate their performance. For the regression estimators we consider the *Means Squared Error* (MSE), *Root Mean Squared Error* (RMSE) and *Mean Absolute Error* (MAE) to compare estimators with different parameters.

To evaluate our binary classification estimators we consider *Accuracy*, *Precision*, *Recall*, the *F1-score* and the *Receiver Operating Characteristic* ROC curve with the associated *Area Under the Curve* (AUC) metric.

2.3.1 Regression Metrics

In statistics, the mean squared error (MSE) is a metric to measures the average of the squares of the errors. In other words, MSE is the average squared difference between the actual values and the predicted values. MSE is calculated as shown in Equation 2.28 where n is the sample size, y_i is the observed value for observation i and $\hat{y}_i$ is the predicted value for observation i.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.28}$$

The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator and its bias. We measure MSE in the square of the units of our data, gaining a more intuitive understanding about how wrong an estimator the Root Mean Squared Error is sometimes considered as this brings the error into the same units as our data. The RMSE is merely the square root of the MSE, calculated as shown in Equation 2.29.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.29}$$

One criticism regarding the use of MSE is that the metric can be sensitive to outliers in the dataset. As the difference between the predicted and actual values are squared, outliers can exaggerate the MSE. Another metric to consider which is less sensitive to outliers is the Mean Absolute Error (MAE), which merely takes the absolute value between the difference of the actual and predicted values and is calculated as shown in Equation 2.30.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}| \tag{2.30}$$

2.3.2 Classfication Metrics

There are various metrics to evaluate classification estimators, but all of them stem from a confusion matrix. A confusion matrix is a table that describes the performance of a classification estimator. We show a confusion matrix for binary classification in Table 2.2, here we can see the different types of correct classifications - True Positive (TP) and True Negative (TN), and the different types of wrong classifications - False Negative (FN) and False Positive (FP).

	Actual	
Predicted	0	1
0	True Negative (TN)	False Negative (FN)
1	False Positive (FP)	True Positive (TP)

TABLE 2.2: A confusion matrix showing where *True Negatives*, *False Positives*, *False Negatives* and *True Positives* are situated.

- **True positives** are where the labels we were trying to predict are 1 and the classifier correctly predicted 1.
- **True negatives** are where the labels we were trying to predict are 0, and the classifier correctly predicted 0.
- **False positives**, also known as Type I errors, are where the labels we were trying to predict are 0 and the classifier wrongly predicted 1.
- **False negatives**, also known as Type II errors, are where the labels we were trying to predict are 1 and the classifier wrongly predicted 0.

When evaluating a classification estimator, we can use the counts of TP, TN, FP and FN to construct metrics that give us insights on how our classifier is performing. The metrics we will be interested in are Accuracy, Precision, Recall and the F1-score.

Accuracy can be thought of as the overall performance of the classifier, in other words, how often is the classifier correct? Accuracy is calculated as shown in Equation 2.31 where $N = TP + TN + FP + FN$ - the total number of observations being classified.

$$Accuracy = \frac{TP + TN}{N} \tag{2.31}$$

Precision is the fraction of relevant instances among the retrieved instances and is calculated as in Equation 2.32.

$$Precision = \frac{TP}{TP + FP} \tag{2.32}$$

Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances and is calculated as in Equation 2.33.

$$Recall = \frac{TP}{TP + FN} \tag{2.33}$$

F1-Score is a measure of a test's accuracy and considers both the precision and recall of the estimator. The F1-Score merely is the harmonic mean of a classifiers precision and recall and is calculated as in Equation 2.34.

$$F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.34}$$

As most classifier estimators can output a probability of an observation belonging to class 1 and class 0 we might be interested in what the performance of our classifier is if we vary the threshold between assigning a prediction to class 0 and class 1. The *Receiver Operating Characteristic* (ROC) curve is a visual metric that shows the predictive ability of a binary classifier as its discrimination threshold is varied (Fawcett (2006)).

The ROC curve is graphed by calculating the true positive rate ($TPR = \frac{TP}{TP+FN}$) against the false positive rate ($FPR = 1 - \frac{TN}{TN+FP}$) at various threshold settings. A comparison between 4 classifiers are shown in Figure 2.7.

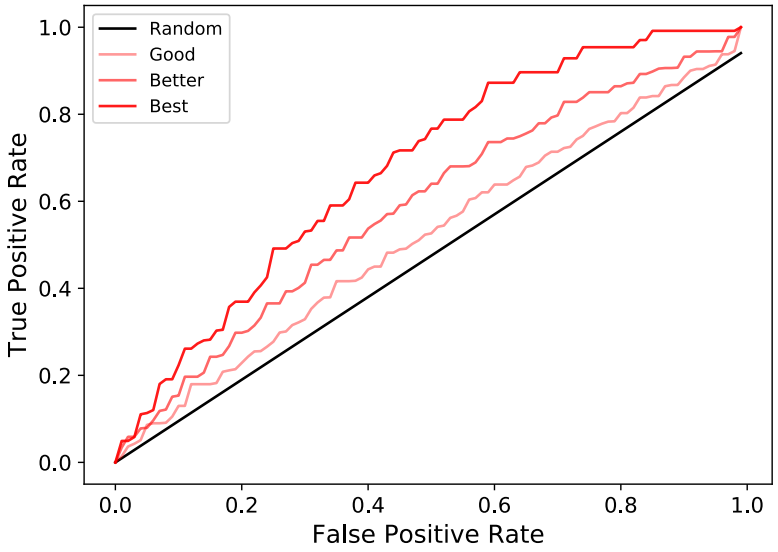


FIGURE 2.7: The ROC curve of 4 different classifiers. A random classifier in black, a good classifier in light red, a better classifier in a moderate shade of red, and the best classifier of the three in red.

As graphical metrics, like the ROC curve, can lend itself to subjectivity when comparing estimators a metric known as the Area Under the Curve (AUC) is a way to summarise the ROC curve into one number that can be used to compare and contrast different estimators. All the AUC essential is, is the area under the ROC curve for an estimator.

Looking at Figure 2.7, the AUC baseline of a random binary classifier is 0.5 as the area under the red curve is 0.5. Looking at the red estimator, we can see that it has the higher area under the curve as that of the random classifier, but also any of the others; therefore we select the red classifier to be the best classifier amongst the 4 based on the AUC metric.

Chapter 3

Introduction to Telecommunication

3.1 Net Promoter Score

Reichheld (2003) argued that most customer satisfaction surveys conducted were not providing any use to organisations, as most of these surveys were very long, they had low response rates and ambiguous root causes that made it difficult for line managers to act on and improve on the customer satisfaction reported. Reichheld (2003) further argued that senior executives, board members and investors rarely challenged or audited these surveys as they seldom correlated tightly with profits or growth.

In contrast, the Net Promoter Score of an organisation aims to serves as a proxy measure for customer loyalty and is derived from one question: "How likely is it that you will recommend this product/company to a friend or colleague?". The response options range from 0 representing not at all likely to 10 representing extremely likely; these responses are grouped into the following categories:

Promoters: Responses from 9-10
Passives: Responses from 7-8
Detractors: Responses from 0 to 6.

To get the net promoter score for an organisation you subtract the proportion of detractors from the proportion of promoters and convert this to a percentage, Figure 3.1 shows this graphically.

For example, if we survey 100 people, 40 promoters, 40 passives and 20 detractors, the NPS for the survey would be $\frac{40-20}{100} = 20\%$. A NPS of 20% then suggests that the organisation has +20% more promoters than detractors. The motivation behind NPS is that this positive word of mouth towards the organisation results in positive economic growth.

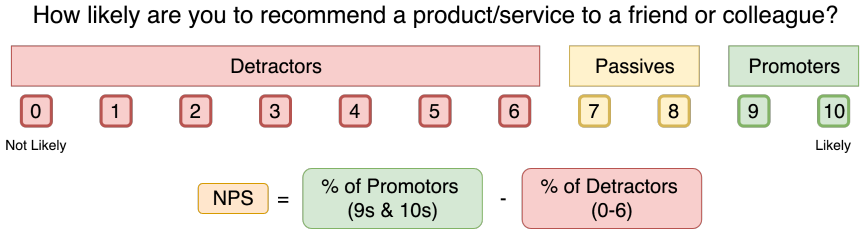


FIGURE 3.1: A graphical representation showing how NPS is calculated as the difference between the percentage of promoters and detractors.

One big driver for an organisation to use the Net Promoter Score is its simplicity, both to those surveyed, as well as to upper management interpreting the results afterwards. For the surveyed subscriber, they do not need to fill in a lengthy questionnaire disclosing their age, race and other personal information - they need only answer one question through various mediums like SMS, Phone Calls, emails or online. On the organisation's side, the simplicity remains - upper management need only look at one metric to get a feel for the happiness of the entire subscriber base, or they can break NPS down by other dimensions like region, device or price plan.

Another advantage of NPS is that it focusses the attention of the organisation away from lagging indicators like revenue per subscriber and focusses their attention to what the subscribers are experiencing at the present moment with minimal delay. It is much more difficult to predict the future growth of a company based on last quarters financial results, especially in the fast-changing technology industry.

Short term promotions and initiatives begin and frequently end in an organisation and a quick, subscriber centred metric probing subscriber satisfaction is highly valued. As an organisation, if you have a reasonable proxy for measuring future growth and revenue, then you might be able to improve next years revenue, while in the process, your customers can become happier as you reveal potential aches they might have.

Despite its simplicity, one significant criticism of the NPS statistic is that it reduces an 11 point Likert scale to a 3 point scale. This reduction has two significant consequences. Firstly it increases the required sample size to achieve the same level of precision when calculating the mean of the responses, and the margin of error is usually around twice

as wide compared to having 3 response options. Secondly, it is harder to detect differences between scores, either over time or compared to a competitor (Sauro (2012)). Rocks (2016) performed a thorough investigation of the statistical properties of the NPS statistic.

Despite the publicity and promotion for it being the *ultimate* question, there are many other, perhaps better questions for a specific industry. There exists many metrics of customer satisfaction which correlate with customer loyalty, however Reichheld (2006), points out that the likelihood to recommend question was the best or second best predictor of repeat purchases or referrals in 11 out of 14 industries (79%), one of which was the telecommunications industry.

Husgafvel (2011) investigated the relationship between customer satisfaction, customer retention and customer loyalty (Net Promoter Score) versus the value of shares for companies in the telecommunications industry. The hypothesis in the paper is that all the three non-financial metrics correlate to a company's performance - with performance proxied by the share value. The paper looked at the period from Q2/2007 to Q1/2011 and found supporting evidence for the first and second hypotheses, but had to reject the third stating that the Net Promoter Score does not seem to be significantly associated with company performance - if performance is proxied by the share price.

One pitfall that organisations should caution against is just collecting NPS. Albeit a good number to track, it is usually a symptom of subscriber loyalty, not the cause. If the NPS of your organisation is low one approach is to follow up with the detractors and try and find out why they are unhappy with the service they received, another approach is to try and identify related dimensions that detractors have in common and spend more time, money and energy on improving these dimensions. The last mentioned approach is where this paper tries to shed some light on - by identifying commonalities in the experience of promoters and detractors - this can aid upper management to allocate resources accordingly and hopefully improving NPS resulting in an increased market share.

3.2 Customer Experience Management

To be a successful business in today's fast-moving, global, borderless economy, as an organisation you need to ensure you are meeting the expectations and demands of your customers. Boohene & Agyapong (2010) explains that service quality today has become

not only the rhetoric of every organisation but also occupies a notable position in executive discourse. Further, customer satisfaction has become essential for the survival of current day corporate organisations.

Ojo (2010) and Boohene & Agyapong (2010) investigated the relationship between service quality and customer satisfaction in the telecommunication industry for Mobile Telecommunication Network (MTN) Nigeria and Vodafone Ghana respectively. Ojo (2010) initially tried to answer the following questions:

- Does customer service have a relationship with service quality perception?
- Does customer service have a relationship with customer satisfaction?
- Does customer satisfaction have a relationship with service quality perception?

To guide them in answering these questions, Ojo (2010) formulated three hypotheses:

- H_1: Customer service has effect on quality perception.
- H_2: Customer service has effect on customer satisfaction.
- H_3: There is a relationship between customer satisfaction and quality perception.

Boohene & Agyapong (2010) followed a similar method as Ojo (2010) but aimed to find out whether there exists a relationship between customer satisfaction and service quality in a public sector CSP in a developing country. Rather than having three hypotheses, this study reduced it to one hypothesis, namely:

> H_1: A high level of service quality exerts a strong influence on the overall level of customer satisfaction for Vodafone Ghana.

Both studies were guided by the service quality, SERVQUAL model to formulate questions to investigate the drivers of customer loyalty. In Parasuraman et al. (1985) service quality is defined as the customer's comparison between service expectation and service performance. The SERVQUAL model was introduced by Parasuraman et al. (1988) and is defined as a multi-item scale for measuring consumer perceptions of service quality. In many industries, including the telecommunication industry, the SERVQUAL model has been used to measure service quality by capturing respondents expectations and perceptions. The SERVQUAL model is built on the dis-confirmed expectancy paradigm and consists of questions spread over five dimensions: reliability, assurance, tangibles, empathy and responsiveness. The MTN Nigeria study had the following survey questions which were answered by 230 respondents:

- Reliability: MTN keeps customers informed about when network services will be performed
- Assurance: Behaviour of customer service personnel instils confidence
- Tangibles: How will you rate the customer service package of MTN
- Empathy: Courtesy of customer service personnel
- Responsiveness: MTN shows sincere interest in solving problems

The scoring of these 5 questions in the Ojo (2010) study formed the independent variables for the study. The study regressed these features onto both a quality perception score and a customer satisfaction score given by the respondents. Also assessing the correlation between quality perception and customer satisfaction, Ojo (2010) was able to reject the null hypothesis for all three of his hypothesis tests with an F-statistic at a significance level of 0.05.

The Boohene & Agyapong (2010) study, which had 460 completed questionnaires showed there is a relationship between customer satisfaction and service quality. The service quality questions spanned the 5 dimensions of the SERVQUAL model and the 5 SERVQUAL dimensions regressed onto customer satisfaction had an F-statistic with a significance level of 0.005. Both these studies then suggest that there is a strong relationship between service quality and customer satisfaction.

3.3 Telecommunication Terminology

In this section, we aim to provide a brief overview of where we obtain the network statistics for each subscriber. We use what is termed an *active monitoring* probing solution which mirrors the live network traffic and feeds this into a correlation engine which, through some logic, creates metrics reflecting what happened on the network by correlating different interfaces with one another.

3.3.1 Wireless Cellular Technology Evolution

The first generation wireless cellular technology, 1G, was introduced in the early 1980s and was an analogue based telecommunications standard. 1G was the world standard until the second generation *digital* wireless cellular technology, 2G, was introduced in 1991. Shortly after that, the Global System for Mobile communication (GSM) standard was defined by the European Telecommunications Standards Institute (ETSI) to

standardise the protocols and interfaces used in second generation (2G) digital cellular networks.

2G was initially developed for circuit switched[1] transport, optimised for full duplex voice telephony. GSM/2G was later enhanced to support data communications, first using circuit switched transport, then later expanding to support packet data via General Packet Radio Services (GPRS) and Enhanced Data rates for GSM Evolution (EDGE).

To develop 3G the 3rd Generation Partnership Project (3GPP) was established in 1998 with the scope to produce technical specifications and reports for a 3G network based on evolved GSM core elements and the radio access technologies. Their scope expanded with the demand for 4G technologies and as of 2007 the 3GPP's responsibilities span:

- GSM and related 2G and 2.5G standards, including GPRS and EDGE.
- Universal Mobile Telecommunications Service (UMTS) and related 3G standards, including High Speed Packet Access (HSPA).
- Long Term Evolution (LTE) and related 4G standards, including LTE Advanced and LTE Advanced Pro.
- Next generation and related 5G standards.
- An evolved IP Multimedia Subsystem (IMS) developed in an independent access manner.

The data gathered in this thesis for each subscriber covers voice and data metrics for 2G and 3G technologies.

3.3.2 Interfaces and Protocols

Any digital communication system is made up of a set of interfaces and protocols, where the interfaces describe the boundary across two or more separate components or nodes, and the (communication) protocols are the set of rules that allow for these nodes to communicate. As an analogy one can view the nodes as people, the protocols as the languages they speak and the interfaces as an intersection where people are speaking the same language can converse and exchange information.

Protocols very seldom exist in isolation and usually appear as a protocol stack where we abstract the lower layer protocols from the higher layer protocols. This abstraction

[1] Circuit switching is a method of having a telecommunications network where two nodes first establish a dedicated communication circuit through the network before the nodes can communicate.

of the lower layer protocols allows for higher layer protocols to operate independently of the lower layers. Being able to trust the lower levels allow for applications higher up in the stack to achieve more complex tasks. The 7 conceptual layers of the protocol stack called the Open Systems Interconnection (OSI) model was formalised in 1984 by the International Organisation for Standardisation (ISO) as standard ISO 7498.

The Open Systems Interconnection (OSI) model, shown in Figure 3.2 is sometimes used to conceptualise how the various protocols within a protocol stack communicate with each other. Rarely the original OSI model is implemented in practice as *seven* layers as very few protocols bundle all functionality into the defined layers. Instead, protocols either span over different layers or a protocol can make up a small portion of a layer.

An example of how we use the OSI model every day is a simple google request. Here whichever browser you are using serves you at the *Application* layer (layer 7). For your request to get to Google, the request needs to be passed all the way down to the *Physical* layer (layer 1) which transmits the 1's and 0's between your computer and Google's servers. Through well defined, standardised protocols and interfaces service providers can ensure that the data being passed from layer 7 to layer 1 does not get corrupted, intercepted or lost.

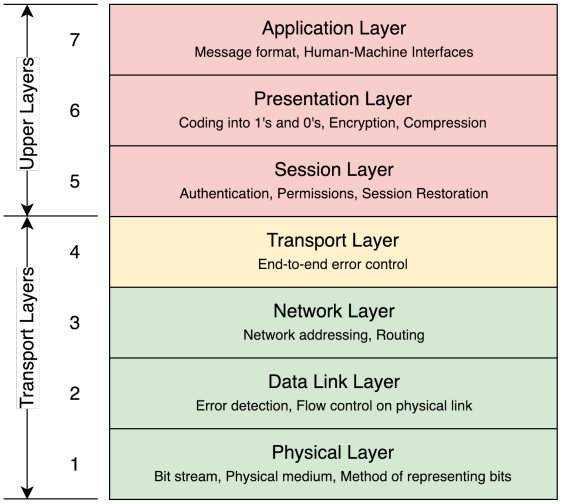


FIGURE 3.2: The OSI model showing the conceptual seven layers in a vanilla protocol stack.

3.3.3 Telecommunication Protocols and Interfaces

The telecommunication industry has a unique suite of protocols and interfaces defined by the 3rd Generation Partnership Project (3GPP). The 3GPP defines how everything should work in a telecommunication network to be interoperable with other telecommunication networks across the world.

The 3GPP standardise how the radio access network should work, through to the core network, to how billing should be processed as well as which speech encodings to use for international communication. Some older protocols and interfaces still exist in networks today due to a requirement of interoperability between new technology and legacy systems.

The interfaces we probed to collect the subscriber network performance data for our survey come from the A, Gb, IuCS, IuPS and Gn interfaces. Figure 3.3 shows a simplified diagram as to where these interfaces sit in the network relative to a subscriber.

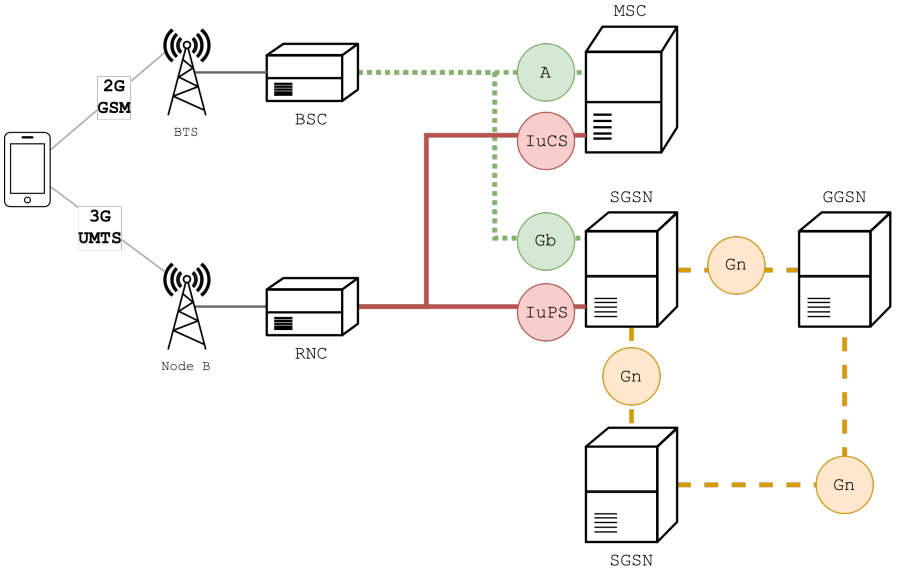


FIGURE 3.3: A simplified network diagram showing on which interfaces (A, IuCS, Gb, IuPS and Gn) we are gathering data for each subscriber. The A (voice) and Gb (data) interfaces (green) show through which nodes a subscriber's device connects to the network on the 2G GSM radio access technology. The IuCS (voice) and IuPS (data) interfaces (red) show through which nodes a subscriber's device connects to the network on the 3G UMTS radio access technology. The Gn interfaces (orange) show through which nodes a subscriber's device connects within the packet switched (data) network.

Table 3.1 describes which interfaces are related to which radio access technology, 2G or 3G, and what traffic we see on the A, IuCS, Gb, IuPS and Gn interfaces.

Interface	Technology	Description
A	2G	• Interface between the Base Station Controller (BSC) and the Mobile Switching Centre (MSC) • Used for signalling and media • The speech is transcoded whilst the signalling is handled by Signalling System No. 7 (SS7) • One user has the sole use of a dedicated physical resource
Gb	2G	• Interface between the BSC and Serving GPRS Support Node (SGSN) • Used for the exchange of signalling information and user packet switched data • In contrast to the A-Interface, users are multiplexed on the same physical resource and resources are allocated during activity periods
IuCS	3G	• Interface between the Radio Network Controller (RNC) and the Mobile Switching Centre (MSC) • Also called the interface between the 3G Radio Access Network (RAN) and the Circuit Switched Core Network (CS-CN) • Used to setup, manage and tear down bearers and also to transport circuit switched voice
IuPS	3G	• Interface between the RNC and the SGSN • Also called the interface between the 3G RAN and the Packet Switched Core Network (PS-CN) • This interface can be thought of as the gateway to the world wide web
Gn	2G & 3G	• Interface between SGSNs and other SGSNs and internal Gateway GPRS Support Nodes (GGSNs) • Uses GTP to manage subscriber sessions and mobility management of subscribers

TABLE 3.1: The radio access technology associated with the A, IuCS, Gb, IuPS and Gn interfaces, along with a short description of what traffic flows on these interfaces.

Ojo (2010) and Boohene & Agyapong (2010) investigated the relationship between service quality and customer satisfaction, where service quality in their studies was a wide range of customer touchpoints, to name a few:

- whether customers were informed about when network services would be performed
- courtesy of customer service personnel
- behaviour of customer service personnel instils confidence.

In this thesis, however, we investigate whether there is a relationship between technical, non-human, factors related to network performance and customer satisfaction. In other words, do technical, non-human, factors measured on the interfaces described above play an important role in predicting NPS scores.

Chapter 4

Dataset Description

4.1 Data Sources

The dataset used for our analysis came from an SMS based NPS survey from February 2018 for a major mobile operator in South Africa. The survey only targeted prepaid subscribers and received a total of 23986 responses. We also had the same dataset available for June 2018 which had 18859 responses, and we use the June dataset in Section 6 to evaluate how the trained models fair on an out of sample month.

To see if more history is useful in predicting NPS we further split our February and June dataset into a 2-week and 5-week dataset. For the 2-week dataset, we aggregate a subscribers' network metrics for 2 weeks before the date surveyed and 5 weeks prior for the 5-week dataset.

For the 2 weeks February dataset, some subscribers did not have any network activity on the interfaces we are considering; thus the 2-week dataset has 422 fewer survey responses with 23564 responses. For simplicity, we only look at the distributions for the 5-week dataset but report on the impact of the 2-week versus 5-week datasets in the modelling section.

Table 4.1 shows the overlap in subscribers between the four datasets considered and we see there are no overlapping subscribers between the February and June datasets. We note here that not all subscribers in the 5-week datasets appear in the 2-week datasets as they did not perform network activity during the 2 weeks, but they did perform network activity within the 3 weeks before the 2-week dataset.

	FEB2	FEB5	JUN2	JUN5
FEB2	23564	23450	0	0
FEB5	23450	23986	0	0
JUN2	0	0	18658	18571
JUN5	0	0	18571	18859

TABLE 4.1: Subscriber overlap between the four dataset considers. Between the FEB2 and FEB5 there are 23450 overlapping subscribers, between the JUN2 and JUN5 datasets there are 18571 overlapping subscribers and there are no overlapping subscribers between the February and June datasets.

4.1.1 NPS Distributions

As we are mainly interested in what makes a subscriber a promoter or a detractor, we remove subscribers that have an NPS response of 7 and 8 as these subscribers are neutral according to the definition of NPS. Removing the subscribers with scores of 7 and 8 is common practice in the industry as we ultimately are concerned with the extremes of what makes a subscriber happy or unhappy. Removing these subscribers decreases the number of responses by roughly 14.5% for both the 2-week and 5-week datasets.

Table 4.2 shows the count of subscribers which are promoters and detractors for the February and June 2018 surveys. From the percentage of subscribers that are promoters and detractors, we see that this operator has a positive NPS metric of $59.52\% - 40.48\% = 19.04\%$ indicating a positive word of mouth for February 2018. However, for the month of June, and we can see that this operator lost 3.42% NPS points from February to June ($57.81\% - 42.19\% = 15.62\%$).

Status	Dataset	Count	Percentage
Promoters	FEB5	12212	59.52 %
Detractors	FEB5	8306	40.48 %
Promoter	JUN5	9279	57.81 %
Detractor	JUN5	6771	42.19 %

TABLE 4.2: The split between promoters and detractor for the 5-week February and June datasets.

We plot the histograms of the NPS scores for the February and June 2018 survey in Figure 4.1. Figure 4.1 shows the number of subscribers that had a response of 10 decreased for June and the number of subscribers having a response of 0 to 6 stayed roughly the same, explaining the decrease in NPS in Table 4.2.

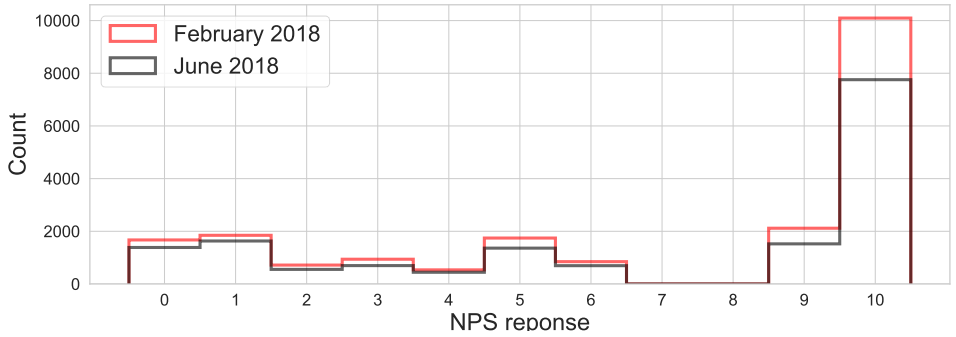


FIGURE 4.1: The kernel density estimation distribution for the February and June 2018 NPS scores.

4.1.2 Technology Concepts

Here we aim to provide some context for terms we use in the next section.

Call Drop Rate (CDR) and Call Setup Failure Rate (SFR) In a mobile telecommunications network, there are several root causes which result in calls to be successfully connected, terminated and reliably maintained - some of these causes are controllable while others are not. We split call failures into 2 different categories, viz. call drops and call setup failures. Call drops can occur only when a call has already been successful, and the call then drops, whereas a call setup failure occurs when a call could not get past the setup phase.

Latency For us, latency refers to network latency in a packet switched network and is the round-trip time (RTT) delay measurement for a packet from a subscribers device, to whichever server they are requesting information from and back to their device, usually measured in milliseconds. We make a distinction between client latency and server latency and define the client latency as the RTT from device to server, and the server latency as the RTT from the mobile core network to the server, ignoring latency caused by the air interface on the way down to the subscriber's device.

PDP Session PDP, short for Packet Data Protocol, is a network protocol used by external networks to set up sessions with GPRS networks. A PDP session contains session information about a subscriber, like their IP address for this session. In 2G and 3G mobile telecommunication networks, before access to the internet is possible a successfully connected PDP session is required.

4.1.3 Predictors

The dataset used has service groups from the A, Gb, IuCS, IuPS and Gn interfaces discussed in Section 3.3.3 and we group the statistics we have for each subscriber into the following services: VOI, BER and GN. We show the mapping of interfaces to service groups in Table 4.3.

Service	Interface
VOI	A
	IuCS
BER	IuPS
	Gb
GN	Gn

TABLE 4.3: The four different service groups our KPIs are divided into.

Within each service, we have various metrics for a subscriber. We refer to these metrics as Key Performance Indicators (KPIs), and in this thesis, we view these KPIs as our predictor variable for describing subscriber performance. For each subscriber, we have all or a subset of KPIs within each service depending on whether or not a subscriber performed any transactions on that interface during a given period.

Some interfaces gather more data due to the type of traffic on the interface, and consequently, data is not available for all subscribers across all of the interfaces and predictors. The VOI service for example which contains calls made and received for subscribers collects more traffic across the subscriber base compared to the GN or BER services which only consists out of data traffic that not all subscribers utilise.

In Table 4.4-4.6 we define how we calculate these KPIs and provide an alias and a short description of each KPI per service. The aliases are for ease of readability, and we use the aliases for the remainder of the thesis.

KPI	Alias	Description
CALLS_CONNECTED	N/A	Total number of mobile originating (MO) and mobile terminating (MT) 2G and 3G calls successfully connected between two or more parties. $$CALLS\_CONNECTED = (MO + MT)_{2G} + (MO + MT)_{3G}$$
CALLS_NW_DROPPED	N/A	Total number of MO and MT 2G and 3G call that where dropped due to network related issues. $$CALLS\_NW\_DROPPED = (MO + MT)_{2G} + (MO + MT)_{3G}$$
CALLS_ATTEMPTED	N/A	Total number of MO and MT 2G and 3G call that where attempted. $$CALLS\_ATTEMPTED = (MO + MT)_{2G} + (MO + MT)_{3G}$$
CALLS_NW_SETUP_FAILURES	N/A	Total number of MO and MT 2G and 3G call that failed to setup due to network related causes. $$CALLS\_NW\_SETUP\_FAILURES = (MO + MT)_{2G} + (MO + MT)_{3G}$$
VOI_CDR	CDR	Call drop rate (CDR) calculated as the ratio of calls dropped by network related causes and the total number of call connected for a subscriber. $$VOI\_CDR = \frac{CALLS\_NW\_DROPPED}{CALLS\_CONNECTED} \times 100$$
VOI_SFR	SFR	Setup failure rate calculated as the ratio of call that failed to set up due to network related causes and the number of call attempted for a subscriber. $$VOI\_SFR = \frac{CALLS\_NW\_SETUP\_FAILURES}{CALLS\_ATTEMPTED} \times 100$$
VOI_MOVOICEAVGDURATION	MO_DUR	Total number of milliseconds spent on 2G and 3G calls which were made by a subscriber - mobile originating.
VOI_MTVOICEAVGDURATION	MT_DUR	Total number of milliseconds spent on 2G and 3G calls which were received by a subscriber - mobile terminating.

Table 4.4: The feature descriptions for features divided into the VOI service group.

KPI	Alias	Description
BER_ATTACHAVGDURATION	ATT_DUR	The average duration in milliseconds to set up either a 2G or 3G bearer.
BER_PDPACTAVGDURATION	PDP_DUR	The average duration in milliseconds to set up a PDP Context with either the 2G or 3G PS core network.
BER_ATTACHSUCCESSRATE	ATT_SR	The ratio of successful bearer attaches and the total number of bearer attach attempts on both 2G and 3G. $$BER\_ATTACHSUCCESSRATE = \frac{(BER\_ATTACH\_SUCCESSS)_{2G\&3G}}{(BER\_ATTACH\_ATTEMPTS)_{2G\&3G}} \times 100$$
BER_PDPACTSUCCESSRATE	PDP_ACT_SR	The ratio of successful PDP attach procedures and the total number of PDP attach procedures attempted on both 2G and 3G. $$BER\_PDPACTSUCCESSRATE = \frac{(BER\_PDPACT\_SUCCESSS)_{2G\&3G}}{(BER\_PDPACT\_ATTEMPTS)_{2G\&3G}} \times 100$$

Table 4.5: The feature descriptions for features divided into the BER service group.

KPI	Alias	Description
GN_PDPCREATESUCCESSRATE	PDP_CRE_SR	The ratio of successful PDP create procedures and the PDP create procedures on the Gn interface for both 2G and 3G. $GN\_PDPCREATEUCCESSRATE = \frac{(GN\_PDPCREATE\_SUCCESSS)_{2G\&3G}}{(GN\_PDPCREATE\_ATTEMPTS)_{2G\&3G}} \times 100$
GN_HTTPAVGCLIENTSETUPTIME	C_LATENCY	The average roundtrip time in milliseconds of a HTTP request from the device of a subscriber to a server and back to the device.
GN_HTTPAVGSERVERSETUPTIME	S_LATENCY	The average roundtrip time in milliseconds of a HTTP request from the network to the server and back to the network.

TABLE 4.6: The feature descriptions for features divided into the GN service group.

4.2 Predictors Visualised

Table 4.7 shows the number of observations missing each feature. The missing observations are due to two reasons. Reason one is that the probes which collect the data were oversubscribed during the collection period and did not collect all the activity for a subscriber. Reason two is that each feature is a different counter aggregated up from user activity and a feature cannot be calculated for a subscriber if that subscriber did not perform any activity on a particular interface. An example of the later would be missing values for the call setup failure rate feature; if a subscriber did not make any calls, we can't calculate a setup failure rate for that subscriber. As prepaid subscribers are more inclined to use voice services rather than data service, we can see the VOI service has less missing values than the GN and BER services.

Feature	Service	NAs	NAs Percentage
SFR	VOI	527	2.20%
CDR	VOI	1044	4.35%
MT_DUR	VOI	1408	5.87%
MO_DUR	VOI	2253	9.39%
ATT_SR	BER	2956	12.32%
ATT_DUR	BER	3186	13.28%
PDP_ACT_SR	BER	3983	16.61%
PDP_DUR	BER	5193	21.65%
PDP_CRE_SR	GN	9674	40.33%
S_LATENCY	GN	9996	41.67%
C_LATENCY	GN	9997	41.68%

TABLE 4.7: The number and percentage of missing values (NAs) per feature.

To simplify our modelling approach, we only keep subscribers that have all the features in their respective services and drop all NA values amongst the features within a service.

In other words, within each service, we only keep subscribers that have values for all the features in that service. Table 4.8 shows the number of observations available per service.

Service	Week	Survey Size	Observations	Number of NAs	Percentage
VOI	5	20518	17901	2617	12.75
VOI	2	20143	15413	4730	23.48
BER	5	20518	15160	5358	26.11
BER	2	20143	12852	7291	36.20
GN	5	20518	10166	10352	50.45
GN	2	20143	8709	11434	56.76
FEB	5	20518	7822	12696	61.88
FEB	2	20143	5555	14588	72.42

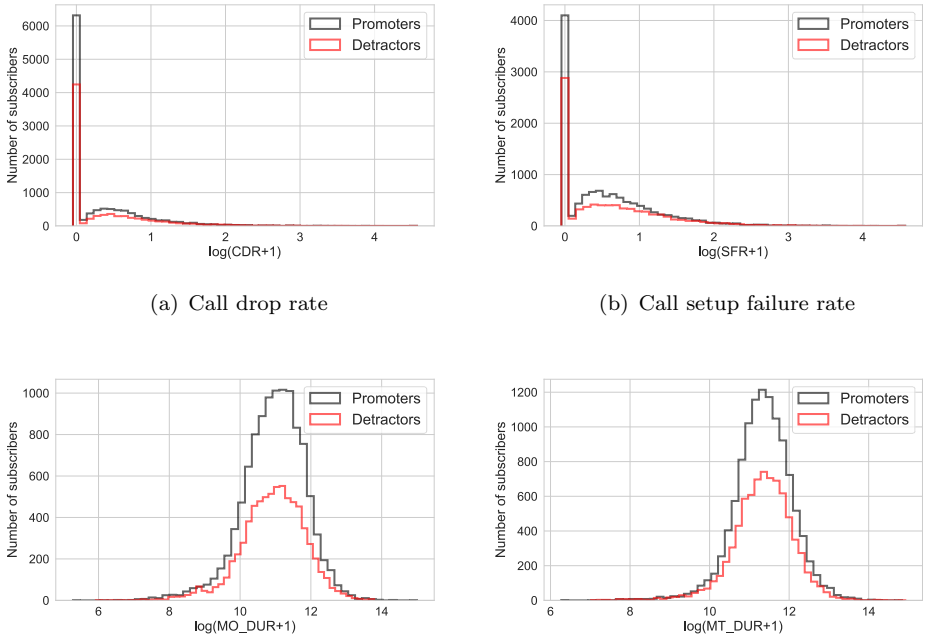
TABLE 4.8: The number and percentage of missing values (NAs) per service.

4.2.1 Distributions

In this section we inspect the distribution of each feature by looking at the natural logarithmic histograms for each feature, split based on whether a subscriber is a promoter or a detractor. We use the logarithmic histogram for each feature as most of the features are skewed and taking the logarithm of the values gives us better insights into the split between promoters versus detractors on an exaggerated scale. As we can not take the log of zero, we add 1 to each feature before taking the natural logarithm. For completeness, we show the non logarithm histogram for each feature in Appendix A.

We note that promoters and detractors are distributed very similarly for all the features which will make it difficult for any model to distiguise them based on these network performance metrics.

VOI service Figure 4.2 shows the logarithmic histograms for the call drop rate, call setup failure rate, mobile originating average voice duration and mobile terminiating average voice duration features.



(a) Call drop rate

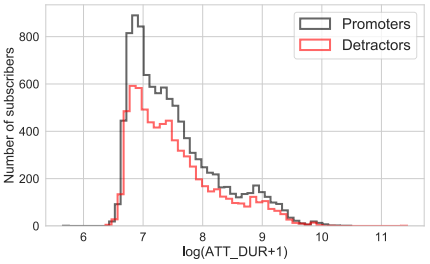
(b) Call setup failure rate

(c) Mobile originating average voice duration.

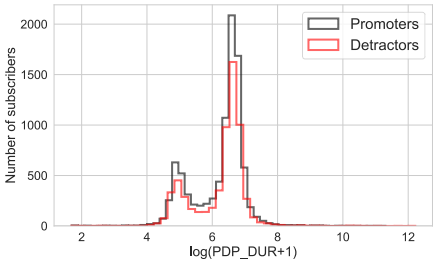
(d) Mobile terminiating average voice duration.

FIGURE 4.2: Logarithmic hisograms for the features in the VOI service.

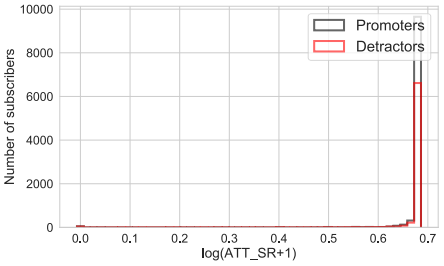
BER service Figure 4.3 shows the logarithmic historgrams for the average bearer attach duration, average PDP activation duration, bearer attach success rate and PDP activation success rate features.



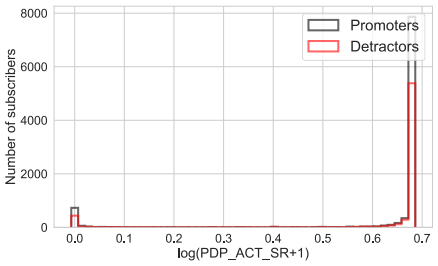
(a) Average bearer attach duration.



(b) Average PDP activation duration.



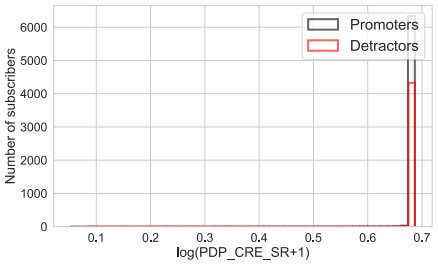
(c) Bearer attach success rate.



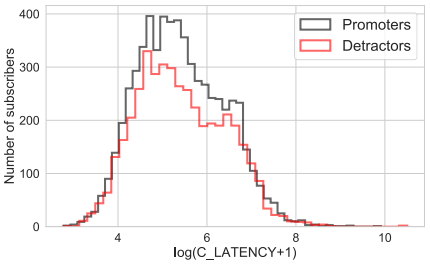
(d) PDP activation success rate.

FIGURE 4.3: Logarithmic hisograms for the features in the BER service.

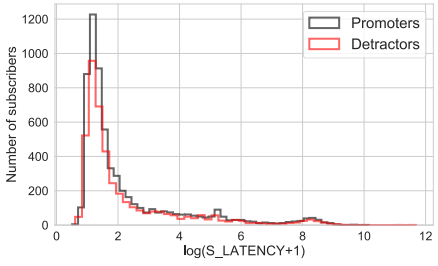
GN service Figure 4.4 shows the logarithmic histograms for the PDP create success rate, average client setup time and average server setup time features.



(a) PDP create success rate.



(b) Average client setup time.



(c) Average server setup time.

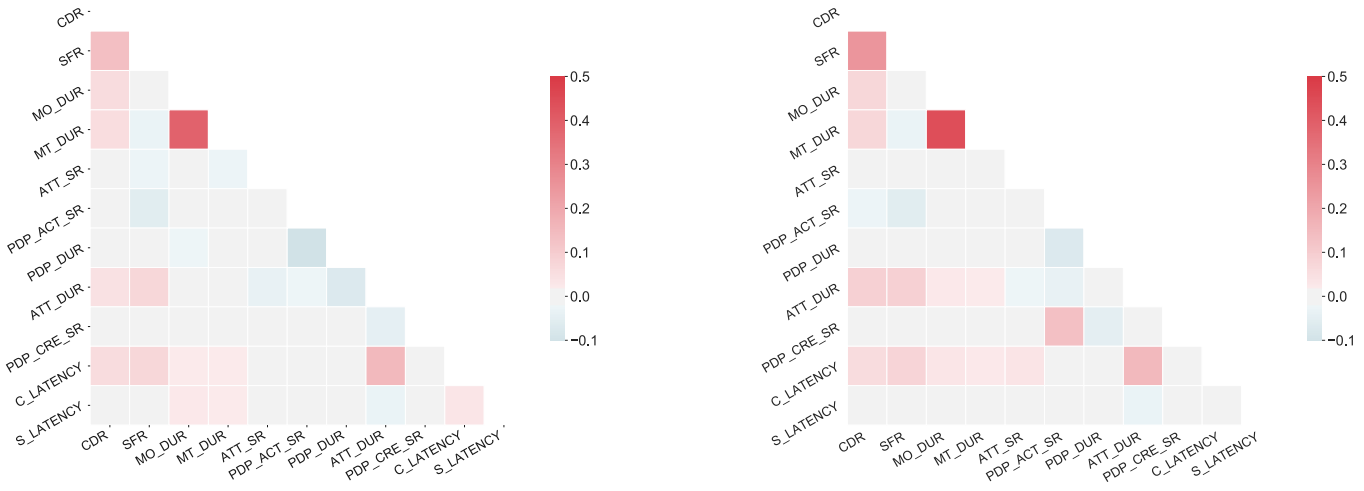
FIGURE 4.4: Logarithmic hisograms for the features in the GN service.

4.2.2 Correlations Matrices

To see how the features are correlated we visualise the correlations matrices for the February 2-week and 5-week dataset. We scale all the features between 0 and 1 to compare and contrast the correlations between the various features. We plot the correlation matrices for the services, *VOI*, *BER* and *GN* separately.

Although the correlations between the features are the same whether we are looking at the entire February dataset or within services, looking at the features within their respective service groups highlights some relationships that we miss due to the size of the entire February correlation matrix.

February In Figure 4.5(a) and Figure 4.5(b) we plot the correlation matrices for the 2-week and 5-week February datasets respectively. The correlation plots show that the MO_DUR and MT_DUR features are the most correlated, showing that subscribers who make long calls also receive long calls.



(a) 2-week dataset.

(b) 5-week dataset.

FIGURE 4.5: The correlation matrix between the features for the 2-week and 5-week February 2018 datasets.

Further, the *Call Drop Rate* feature and the *Call Setup Failure Rate* feature are also positively correlated, and this makes intuitive sense as an underlying factor between

these two features are the number of calls a subscriber makes. In other words, people who make more calls, fail to set up more calls and also drop more calls, and we see this from both the 2-week and 5-week datasets.

We note that the negative correlation between the PDP_ACT_SR and PDP_DUR features which intuitively shows that as the PDP activation success rate of a subscriber increase, his duration to set up a PDP context decreases.

Another intuitive correlation occurs between the PDP_CRE_SR and PDP_ACT_SR features as these both have successful PDP context as an underlying factor. In other words, both features measure the success rate of PDP context being set up, only on different interfaces, PDP_CRE_SR on the Gn interface and PDP_ACT_SR on the Gb and IuPS interfaces.

VOI service For the VOI service, we plot the correlation matrix in Figure 4.6 and we see the strong correlation between the CDR and SFR features. The correlation matrix shows another intuitive negative correlation between *Call Setup Failure Rate* and *Mobile Originating and Terminating Call Duration*.

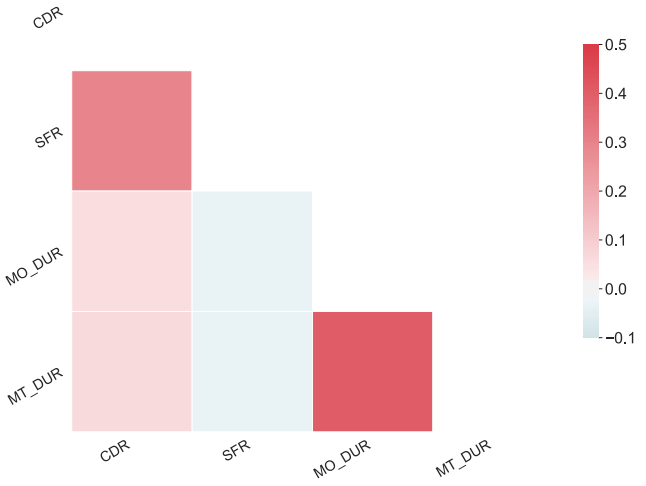


FIGURE 4.6: The correlation matrix for the features in the 5-week VOI service.

The negative correlation shows that subscribers who have longer calls fail to set up fewer calls - this is expected as these subscribers make fewer calls for longer, reducing

the chance for calls to fail to set up. Conversely, we see an intuitive, positive correlation between *Call Drop Rate* and *Mobile Originating and Terminating Call Duration*. This positive correlation shows subscribers who make longer calls, drop calls more often, which makes sense as the longer a call continues, the higher the chance is the call might drop.

BER service For the BER service, we plot the correlation matrix in Figure 4.7 and see an intuitive negative correlation between success rates and duration as we would expect a subscriber to have a longer setup up duration if they have a lower setup or activation success rate.

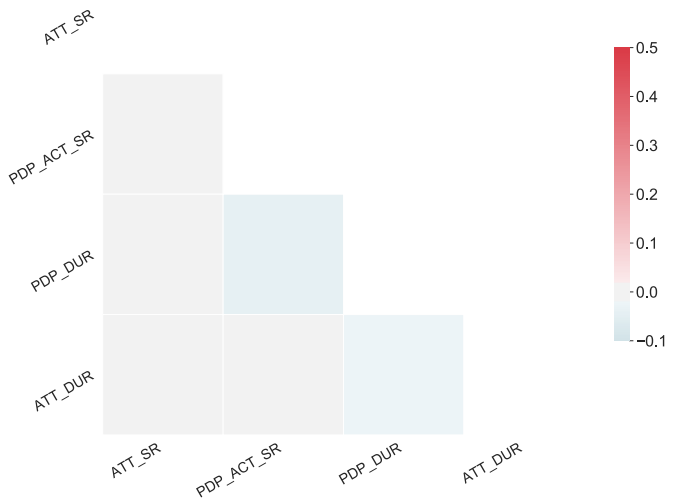


FIGURE 4.7: The correlation matrix for the features in the 5-week BER service.

An unintuitive correlation is the negative correlation between ATT_DUR and PDP_DUR as we would expect a subscriber to have a similar experience setting up PDP context than setting up bearers, but this negative correlation shows there are other factors than bearer activation duration influencing PDP context setup duration.

GN service For the GN service, we plot the correlation matrix in Figure 4.8 and see no highly correlated features within the GN service.

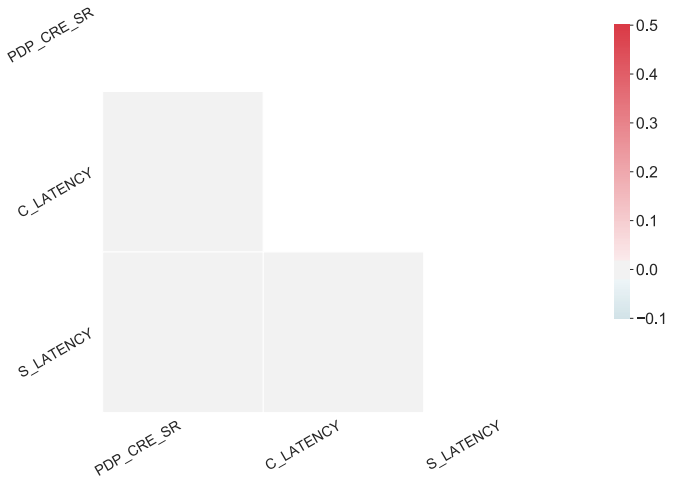


FIGURE 4.8: The correlation matrix for the features in the 5-week GN service.

Chapter 5

Models

In this chapter, we investigate how the various network performance features are related to a subscriber's NPS score. We fit both linear and non-linear models and consider both a regression and classification approach. For the regression approach we use the numeric NPS survey score between 0 and 10 as our dependent variable and for the classification approach we use the binary classes of 1 and 0, where 1 encodes a subscriber being a promoter (NPS score of 9 or 10) and 0 encodes a subscriber being a detractor (NPS score of 6 and below).

As we are interested in what influences a subscriber to be promoter or detractor, for all the approaches we ignore *neutral* subscribers having NPS scores of 7 or 8. Disposing of the subscribers with scores of 7 and 8 is common practice in the industry as we ultimately are concerned with the extremes of what makes a subscriber happy or unhappy. Table 5.1 shows the impact on the number of observations on each dataset if we discard all neutral subscribers. Across all the services and weeks we remove less than 20 % of the observations.

Dataset	Week	Train Observations Left	Train Neutrals Removed	Train Difference	Test Observations Left	Test Neutrals Removed	Test Difference
VOI	5	13431	2218	-16.51 %	4470	747	-16.71 %
VOI	2	11521	1962	-17.03 %	3892	603	-15.49 %
BER	5	11356	1993	-17.55 %	3804	646	-16.98 %
BER	2	9652	1660	-17.20 %	3200	571	-17.84 %
GN	5	7648	1388	-18.15 %	2518	494	-19.62 %
GN	2	6542	1205	-18.42 %	2167	416	-19.20 %
FEB	5	5876	1071	-18.23 %	1946	370	-19.01 %
FEB	2	4182	762	-18.22 %	1373	275	-20.03 %

TABLE 5.1: The impact of removing subscribers having a NPS score of 7 or 8.

To compare regression and classification models using classification metrics, we convert our regression results back into binary classes. The approach we consider is to cast

all subscribers having a predicted NPS score above the mean of all the predicted NPS scores as promoters as the model would view them as on average having a higher score, mathematically this can be expressed as in Equation 5.1.

$$\hat{f}_{clf}(X) = \begin{cases} 1 & \text{if} \hat{f}_{reg}(X) \geq \bar{\hat{f}}_{reg} \\ 0 & \text{if} \hat{f}_{reg}(X) < \bar{\hat{f}}_{reg} \end{cases} \tag{5.1}$$

We use linear regression and logistic regression to explain any linear correlation between any of our 11 features and a subscriber's NPS response. For the linear models, we scale our features between 0 and 1 as scaling all the features to the same interval allows us to compare and contrast the fitted coefficients for each feature. Our analysis approach for both linear and logistic regression is as follows:

- Fit a simple model to the 5 week February dataset to see if there are any highly significant features and get a baseline model.
- Perform ridge regression to see which of the features are less important due to the regularisation term.
- Fit a model for each feature individually to see which of the features has the most descriptive and predictive power.
- Fit a model for each service to see if any within service correlations aid or diminish the performance.
- Evaluate whether taking the logarithm of the predictors increases the performance of the models.

We use decision tree based ensemble methods for regression and classification to see if there are any non-linear relationships between any of our 11 features and a subscriber's NPS response. Unlike with the linear regression approach, here we do not have to scale or log any of the features as decision tree based methods are scale invariant. Our analysis approach for the tree-based regression and classification models is as follows:

- Fit a simple decision tree to the 5 week February dataset to see if there are any highly predictive features and get a baseline model.
- Use each feature to fit a single feature decision tree to gain some insights where these features partition the feature space.
- Perform a grid search to gain the best random forest parameters using the entire February dataset and each of the services.

Table 5.2 shows the classification metrics for a null model where we predict all subscribers to be promoters. We are most concerned with F1-Score and a F1-Score of 0.74 will be our base model performance.

Dataset	Observations	Null Accuracy	Null Recall	Null F1	Null Precision
FEB_2	5555	0.5926	1.0	0.7442	0.5926
FEB_5	7822	0.5896	1.0	0.7418	0.5896
VOI_5	17901	0.5977	1.0	0.7482	0.5977
BER_5	15160	0.5912	1.0	0.7431	0.5912
GN_5	10166	0.5917	1.0	0.7435	0.5917

TABLE 5.2: Null model classification metrics - predicting all subscribers to be promoters.

5.1 Linear Regression

We fit a simple linear regression model to all of the 11 features from the 3 different services for the 5 weeks leading up to the February 2018 NPS survey. Table 5.3 shows the fitted β-coefficient and p-values for each of the 11 features sorted by descending p-values.

Feature	β-coefficient	p-value
CONST	4.363	0.005673
S_LATENCY	-7.609	0.009349
CDR	-6.824	0.017266
SFR	-5.033	0.030437
ATT_SR	2.122	0.044532
PDP_ACT_SR	-0.537	0.073848
PDP_DUR	-4.117	0.099330
MO_DUR	0.762	0.311060
PDP_CRE_SR	0.999	0.402726
MT_DUR	0.473	0.641201
C_LATENCY	0.641	0.722525
ATT_DUR	0.112	0.945100

TABLE 5.3: The β-coefficients and significance for the simple linear regression on the 5-week February dataset sorted by descending p-value.

From Table 5.3 we see that the most significant features with negative β-coefficient causing a decrease in NPS scores are *Server Latency* with a coefficient of -7.61 and p-value of 0.0093, *Call Drop Rate* with a coefficient of -6.82 and p-value of 0.0173 and *Call Setup Failure Rate* having a coefficient of -5.03 and p-value of 0.0304. These coefficients are intuitive, and we expect the experience of subscribers to degrade if they experience a higher server latency, more call drops or they struggle to set up calls.

Conversely, we expect a subscriber to be more satisfied with the service received if his *Attach Success Rate* increases and we see that the model captures this with the coefficient for the *Attach Success Rate* feature being positive and having a value of 2.122 and p-value of 0.044532.

After the fourth most significant feature based on p-value, excluding the bias term, the coefficients become less intuitive. The *PDP Activation Success Rate* feature having a negative coefficient shows that as a subscriber experiences a higher success rate when activating PDP sessions, they are more likely to have a lower than average NPS score with all other features kept constant.

As the p-value for the *PDP Activation Success Rate* feature onwards is above 0.073848 which translates into there being a higher than 7.38% probability that we got these coefficients by chance even if the coefficient is 0 in the population. Consequently, we are less concerned about the unintuitive negative coefficient sign of the *PDP Activation Success Rate* feature (and coefficients with higher p-values) as the coefficient estimation has a high probability of occurring by chance even if there is no correlation between the feature and the NPS response of a subscriber in the population.

From the simple model, we take away the four most significant features as:

- Server Latency - negatively correlated
- Call Drop Rate - negatively correlated
- Call Setup Failure Rate - negatively correlated
- Attach Success Rate - positively correlated

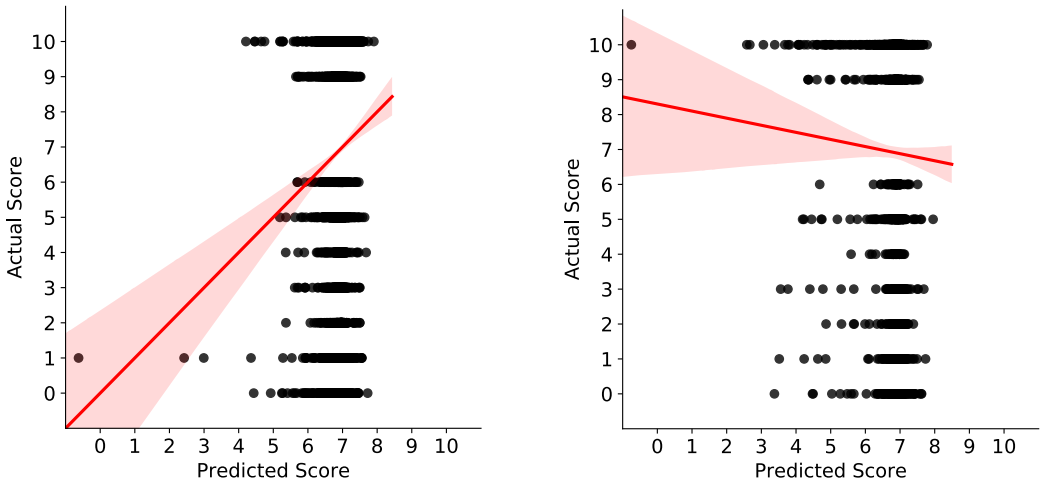
Metric	Test	Train	Difference (%)
Observations	1946	5876	201.95
MAE	3.49	3.44	-1.34
MSE	14.61	14.23	-2.59
RMSE	3.82	3.77	-1.30

TABLE 5.4: The performance metrics for the linear regression on the test and training set of the entire 5-week February dataset.

The linear regression model has an adjusted R^2 of 0.0035 showing that there is some correlation between the 11 features and the NPS responses of subscribers. We interpret the R^2 of 0.0035 as the variance in the predictors describing around 0.35% of the variance in the NPS scores. Also, the model has a p-value for the F-test for the overall significance

of 0.000892, which translates into the fit of our model using the 11 features versus an intercept-only model is significantly reduced.

In Figure 5.1(a) and Figure 5.1(b) we look at how well the linear regression model has performed by plotting the actual versus the predicted responses for the training and test set respectively. We interpolate a straight line between the actual and predicted values, and we see that the training set has fitted the data moderately well with the actual versus predicted interpolated line having a positive slope. In contrast, the actual versus predicted values for the testing set has a negative slope showing that the model does not perform well out of the sample.



(a) Training set.

(b) Testing set.

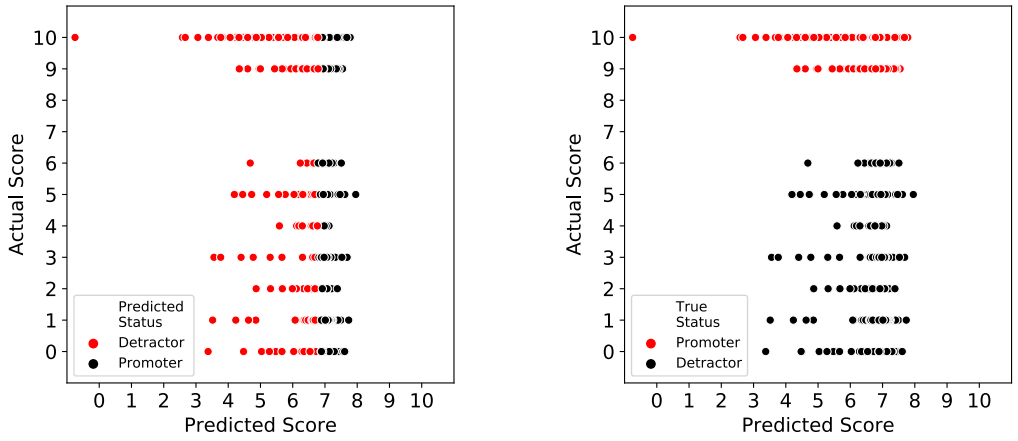
FIGURE 5.1: The true versus the predicted values for the February training and testing set.

From the actual versus predicted plots for both the training and testing set, we notice a significant shortfall of the linear regression model. If we disregard the few outliers, we see that the model only predicts NPS scores in the range of 3 to 8.

Next, we take a look at how well our simple regression model does when converting the regression predictions to classification predictions for the training and test set using the mean of the predicted values as our promoter detractor cut-off value.

Figure 5.2(a) and Figure 5.2(b) below contrasts how the predicted versus actual classification of a subscriber being a promoter or detractor fairs for the regression model casting subscribers to promoters if their predicted NPS response is above the mean of all the predicted NPS responses. As the predicted results only have a range between 3 and 8 the cut off around 7 miss classifies a lot of promoters and detractors, and we see

that our approach of converting our regression results to classification results using the mean of the predicted responses has its shortfalls.



(a) Predicted cast binary colouring.

(b) True binary colouring.

FIGURE 5.2: The true versus predicted values for the February training and testing set.

Table 5.5 shows the Accuracy, Recall, Precision and F1 score for both the training and testing set. It appears that the training set has a higher accuracy than the testing set showing overfitting, whereas the testing set has a higher F1-score due to a high recall.

Model	Set	F1-score	Recall	Precision	Accuracy
5-week February 2018	Test	0.647	0.719	0.588	0.533
5-week February 2018	Train	0.638	0.670	0.608	0.552

TABLE 5.5: The classification metrics for the linear regression predictions converted to binary classes - split based on the mean of all the predictions.

Converting our regression model to a classification model does have it's shortfalls and achieves a test accuracy of 0.533 and a test F1-score of 0.647 which is not better than the 5-week February null model with an accuracy of 0.590 and F1-score of 0.742.

5.1.1 Ridge Regression

We perform ridge regression to penalise the coefficients of some of the features to find which features carry the most predictive power. Figure 5.3 graphically shows the penalised β-coefficients for each feature after performing ridge regression with 15 fold cross-validation.

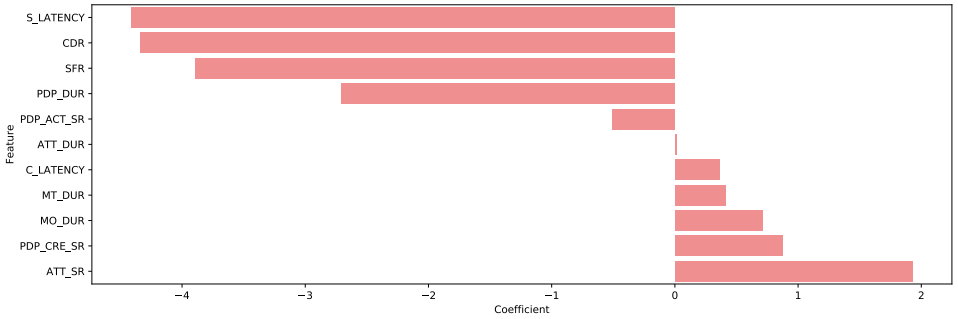


FIGURE 5.3: The β-coefficients after performing ridge regression on the entire 5-week February 2018 dataset.

Figure 5.3 shows that as with the simple linear regression the features *Call Setup Failure Rate*, *Call Drop Rate* and *Server Latency* appear to be the most significant contributors to subscribers having a lower than average NPS score, with *PDP Activation Success Rate* again having an unintuitive negative coefficient sign.

Further, we can see that the higher a subscriber's *Attach Success Rate* and *Mobile Originating Duration* the more likely they are to have a NPS score higher than average. The positive coefficient for *Mobile Originating Duration* shows that subscribers who make many calls tend to be satisfied with the service received regardless of other performance metrics.

feature	test	train	delta
MAE	3.47	3.45	-0.62
MSE	14.35	14.24	-0.74
RMSE	3.79	3.77	-0.37

TABLE 5.6: The regression performance metrics for the linear ridge regression model.

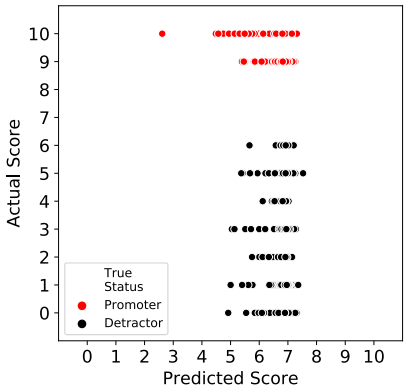


FIGURE 5.4: The true versus predicted values from the ridge regression model.

5.1.2 Individual Feature Regression

To evaluate how each of the features performs we fit a simple linear regression model with a bias coefficient to each of the 11 features in isolation. We also investigate the effect of time-delayed surveys as well as whether taking the logarithm of the feature improves the model at all. To evaluate the effect of time-delayed surveys we use 2 datasets, one containing aggregated statistic per subscriber for the 2 weeks leading up to their survey response and another dataset containing aggregated statistics for 5 weeks before their NPS response.

Table 5.7 shows the results of the simple linear regression models fitted to each of the features.

Feature	Week	Logged	coefficient	p-value	R^2	R^2_{adj}	F-Statistic p-value	Observations	Train MSE	Test MSE
CDR	5	True	-6.9294	0.0002	0.0011	0.0010	0.0002	13431	14.3878	13.9628
CDR	2	True	-6.2504	0.0002	0.0012	0.0011	0.0002	11521	14.0789	14.4115
CDR	2	False	-5.7165	0.0002	0.0012	0.0011	0.0002	11521	14.0789	14.4119
CDR	5	False	-6.3133	0.0002	0.0010	0.0009	0.0002	13431	14.3878	13.9633
SFR	5	True	-4.2779	0.0033	0.0006	0.0006	0.0033	13431	14.3878	13.9733
SFR	5	False	-3.6467	0.0068	0.0005	0.0005	0.0068	13431	14.3878	13.9727
S_LATENCY	5	False	-6.2988	0.0301	0.0006	0.0005	0.0301	7648	14.2965	14.2660
MT_DUR	5	False	1.3373	0.0325	0.0003	0.0003	0.0325	13431	14.3878	13.9866
MO_DUR	5	True	1.8168	0.0332	0.0003	0.0003	0.0332	13431	14.3878	13.9701
MT_DUR	5	True	1.5240	0.0335	0.0003	0.0003	0.0335	13431	14.3878	13.9861
S_LATENCY	5	True	-7.8148	0.0352	0.0006	0.0004	0.0352	7648	14.2965	14.2802
SFR	2	True	-2.9926	0.0370	0.0004	0.0003	0.0370	11521	14.0789	14.4096
MO_DUR	5	False	1.5491	0.0417	0.0003	0.0002	0.0417	13431	14.3878	13.9702
ATT_SR	2	False	1.6732	0.0506	0.0004	0.0003	0.0506	9652	14.1506	14.6686
PDP_ACT_SR	2	True	-1.0655	0.0515	0.0004	0.0003	0.0515	9652	14.1506	14.6578

TABLE 5.7: The simple linear regression models with a p-value less than 0.0525 sorted by descending p-value.

Using the p-values as our performance metric, we see that all 4 variants of the Call Drop Rate models (2 and 5 weeks and logarithmic and non-logarithmic) have the smallest p-value of 0.0002 showing that there is a tiny chance that we got the coefficient of around -6 by chance. Further, it shows that a subscriber is more likely to have a lower NPS score if they are dropping more calls irrelevant of whether the drops occurred 5 weeks or 2 weeks prior.

We take a look at the models with features who have p-values less than 0.0525 and identify the following features for our feature short-list to investigate as these features have a significant linear correlation with NPS survey scores of subscribers. These features are *Call Setup Failure Rate*, *Server Latency*, *Mobile Originating* and *Terminating Call Duration*, *Bearer Attach Success Rate* and *PDP Activation Success Rate*.

Looking at the SFR feature, it appears that a long history of data (5 weeks) results in a more significant correlation between SFR and the NPS survey scores. Both the

logarithmic and non-logarithmic 2 week SFR model still have a p-value less than 0.0525 with all of the SFR models having a negative coefficient for SFR around -4 - indicating that on average for every 1 per cent increase in call setup failure rate a subscriber's NPS score decreases by 0.04.

Looking at the S_LATENCY feature, it appears that only the 5-week aggregated dataset models have a p-value less than 0.00525 showing that there is more data needed to conclude that there is a negative correlation between the server latency and a subscribers NPS survey score. As we scaled the S_LATENCY feature to lie between 0 and 1, we unscale the fitted coefficient value using Equation 5.2.

$$\Delta_{NPS}|\Delta_{S\_LATENCY_s} = \frac{\beta_{S\_LATENCY}}{max\left(X_{S\_LATENCY\,s}\right)} = \frac{-6.3}{133,743} = -0,471 \tag{5.2}$$

We note that with a coefficient of -6.30 for the non-logarithmic scaled server latency feature we can on average expect a subscribers' NPS survey score to decrease by 0.471 for a 1-second increase in latency towards the server.

Looking at the mobile originating and terminating call durations it appears that taking the logarithm of the feature values does not improve the models that much. For the mobile terminating (MT) case we get a more significant model without taking the logarithm of the feature with a β-coefficient of 1.34 and a p-value of 0.0325, whereas if we take the logarithm we get a β coefficient of 1.52 with a p-value of 0.0335.

For the mobile originating case we get a more significant model if we take the logarithm of the feature with a β coefficient of 1.82 and an associated p-value of 0.0332. If we do not take the logarithm of the MO_DUR feature, we get a β coefficient of 1.55 and an associated p-value of 0.0417.

If we use the non-logarithmic coefficients of 1.34 and 1.55 for MT and MO respectively, we can analyse the impact of these features using the transformation equation described in Equation 5.3.

$$\Delta_{NPS}|\Delta_{X_j} = \frac{\beta_j}{max\left(X_j\right)} \tag{5.3}$$

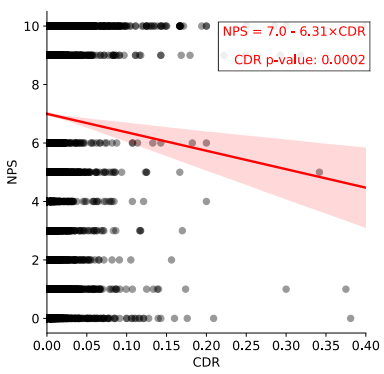
On average for a 1-second increase in mobile terminating calls received the NPS survey score of a subscriber increases by 0.0007449 or for a 1-hour increase a 2.68 increase in NPS survey score. Similarly on average for a 1-second increase in mobile originating calls being made the NPS survey score of a subscriber increases with 0.000773 or for a 1-hour increase a 2.78 increase in NPS survey score. The positive coefficient shows that

subscribers who make or receive longer duration calls are on average happier with the service they receive.

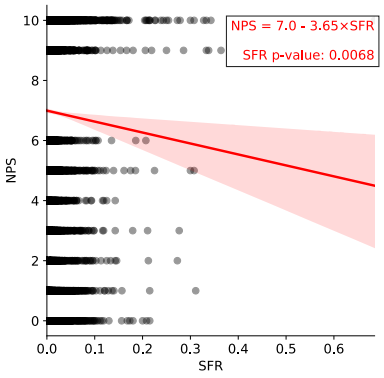
Next, we turn our attention to the correlation between the bearer attach success rate and the NPS survey scores. We notice that with a β coefficient of 1.67 for the non-logarithmic 2-week dataset model for the ATT_SR feature has a p-value of 0.0506. With the p-value hovering around the 0.05 significance level we take note of the positive correlation between this feature and the NPS survey scores, but we concede that NPS has more prominent features such as CDR, SFR, and call duration.

Last on our short-list of features to investigate is the PDP activation success rate model which has a PDP_ACT_SR β coefficient of -1.07 and a p-value 0.0515. The PDP success rate feature is the only one of our short-listed features where the sign of the coefficient does not make intuitive sense. We would expect a subscriber to be more satisfied if they have less trouble setting up a PDP context. The unintuitive sign of this coefficient might have some underlying feature that we are not capturing that most subscribers with high PDP success rates might have in common.

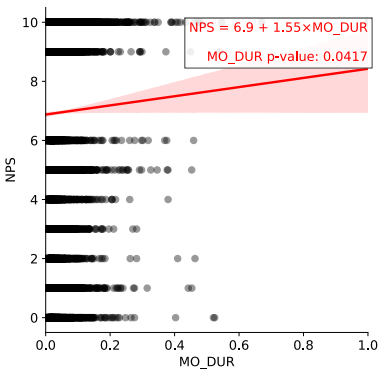
In Figure 5.5 we fit the model equation, $y = \beta_0 + \beta_1 x$ for the 5 week models with non-logarithmic features. Here β_0 is the fitted y-intercept and β_1 is the fitted slope for each of our 11 features, we also show the associated p-value of each feature.



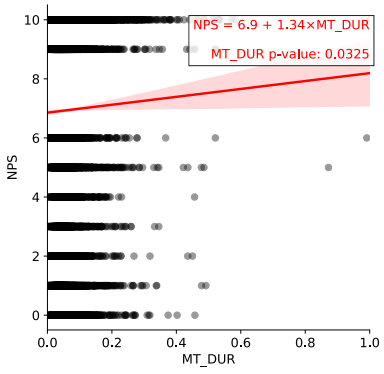
(a) CDR



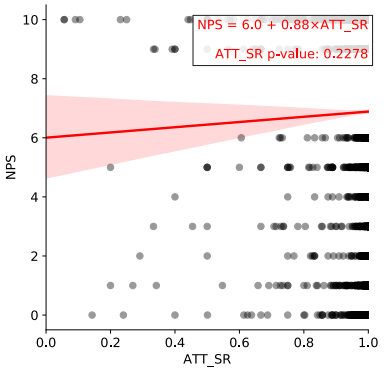
(b) SFR



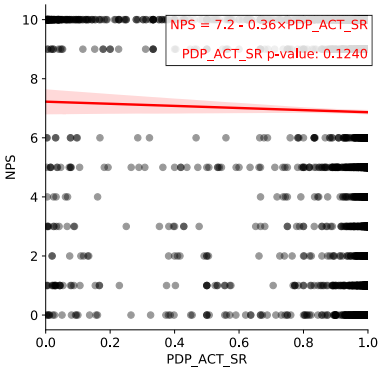
(c) MO_DUR



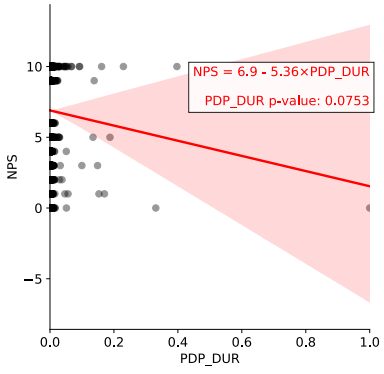
(d) MT_DUR



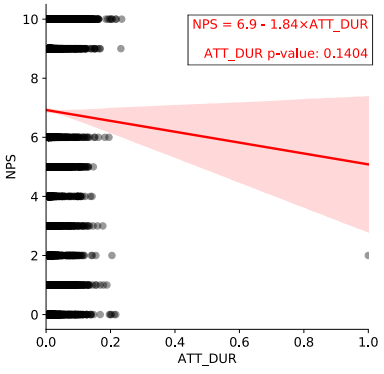
(e) ATT_SR



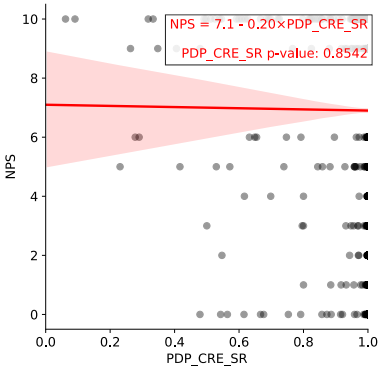
(f) PDP_ACT_SR



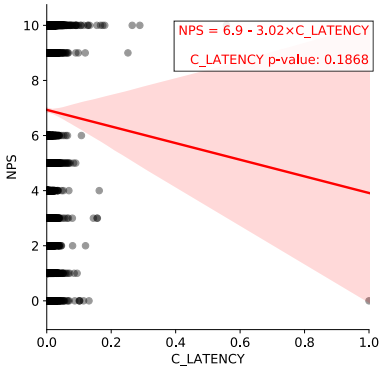
(g) PDP_DUR



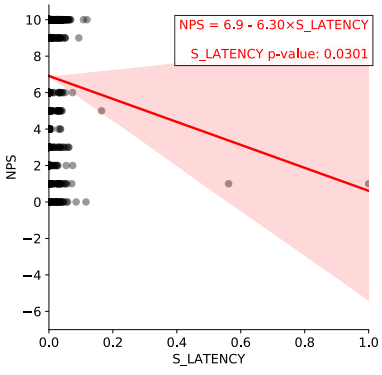
(h) ATT_DUR



(i) PDP_CRE_SR



(j) C_LATENCY



(k) S_LATENCY

FIGURE 5.5: The fitted linear regression line per feature superimposed on the training data with the associated p-value for each feature.

In Figure 5.5 we can see the narrow confidence intervals for the features: CDR, SFR, S_LATENCY, MT_DUR, MO_DUR, ATT_SR, PDP_ACT_SR compared to the wide confidence intervals for the PDP_DUR, ATT_DUR, PDP_CRE_SR, C_LATENCY and S_LATENCY features.

The aforementioned features with narrow confidence intervals are all features we identified as significant in our simple linear regression models in Table 5.7 with a p-values less than 0.0525. Figure 5.5 graphically shows why we have greater trust in the in the models with lower p-values as the linear model fits all the observations on average better.

In Table 5.8 we show the results for the various linear regression models with their predictions converted into binary classes based on the mean of the predicted values sorted by descending test F1-scores.

Feature	Week	Logarithm	Train F1	Train Recall	Train Precision	Train Accuracy	Test F1	Test Recall	Test Precision	Test Accuracy
PDP_CRE_SR	2	False	0.74	0.99	0.60	0.59	0.74	0.99	0.59	0.59
PDP_CRE_SR	2	True	0.74	0.99	0.60	0.59	0.74	0.99	0.59	0.59
S_LATENCY	2	False	0.72	0.92	0.59	0.58	0.73	0.94	0.60	0.58
S_LATENCY	2	True	0.72	0.92	0.59	0.58	0.73	0.94	0.60	0.58
S_LATENCY	5	False	0.72	0.93	0.59	0.58	0.72	0.91	0.59	0.57
ATT_SR	2	True	0.73	0.92	0.60	0.59	0.71	0.92	0.58	0.57
S_LATENCY	5	True	0.72	0.92	0.59	0.57	0.71	0.90	0.59	0.57
ATT_SR	5	False	0.70	0.87	0.59	0.57	0.71	0.87	0.60	0.58
ATT_SR	5	True	0.70	0.87	0.59	0.57	0.71	0.87	0.60	0.58
ATT_SR	2	False	0.73	0.92	0.60	0.59	0.71	0.92	0.58	0.57
C_LATENCY	2	False	0.64	0.69	0.60	0.54	0.66	0.72	0.61	0.56
C_LATENCY	2	True	0.64	0.69	0.60	0.54	0.66	0.72	0.62	0.57
CDR	2	False	0.67	0.75	0.61	0.56	0.66	0.76	0.59	0.55
CDR	2	True	0.67	0.75	0.61	0.56	0.66	0.76	0.59	0.55
CDR	5	False	0.66	0.73	0.61	0.56	0.66	0.74	0.60	0.55

TABLE 5.8: The classification metrics for the linear regression predictions converted to binary classes sorted by descending test F1-score.

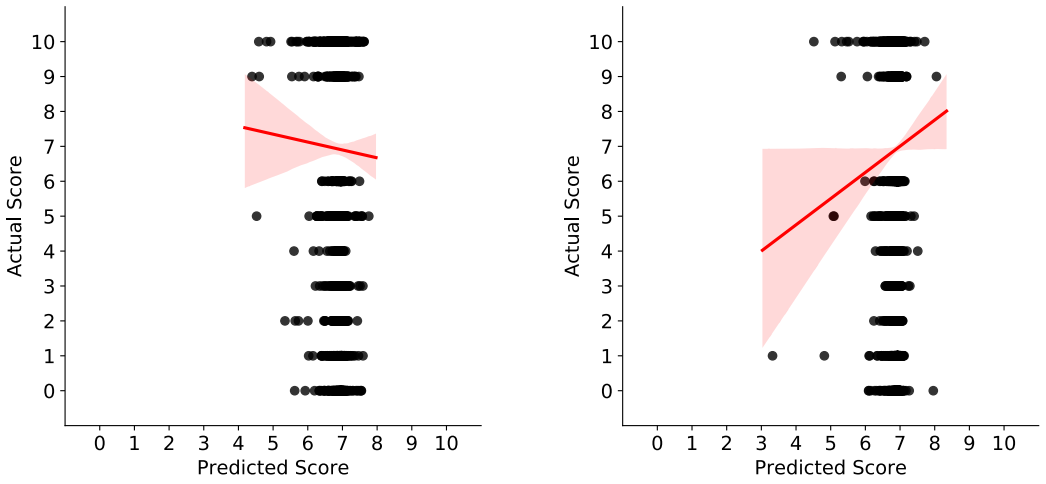
Although the best regression converted to a classification model has a test F1-score of 0.74. The high F1-score is due to a recall of almost 1. This recall close to 1 is due to the small range of predicted NPS scores and splitting the regression predictions at their mean predicts most as promoters resulting in a low number of true negatives and false negatives. The low number of false negatives drives up the recall to close to 1 and results in a high F1-score.

Comparing these metrics with the performance of the null models shown in Table 5.2 shows that the best single feature regression model converted to classification model has approached the null model, casting all subscribers as promoters.

5.1.3 Within Service Regression

In Figure 5.6(a) to Figure 5.7(b) we plot the actual versus predicted results of the test set for all the features contained in each of the services, viz. VOI, BER and GN - for comparison we have also included the FEB dataset containing all the features.

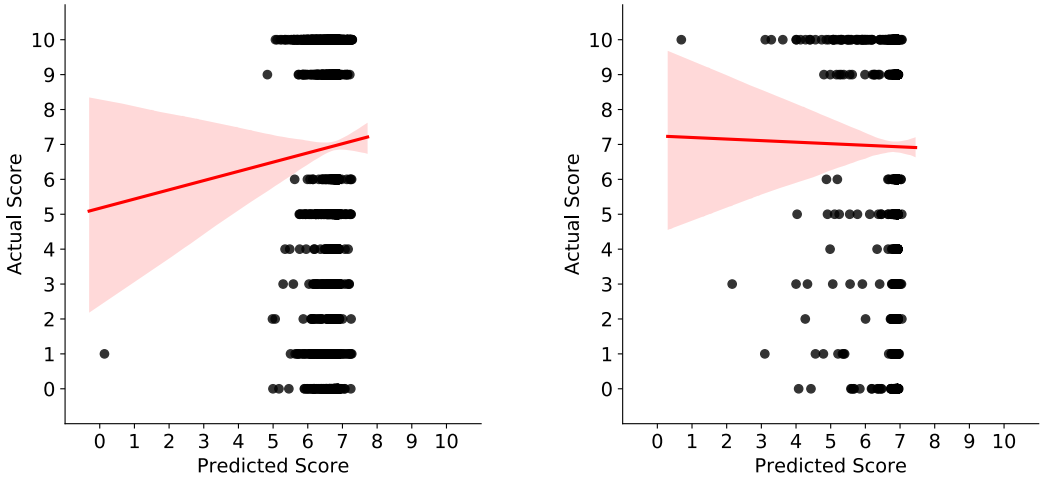
We see the VOI and BER services performing well on the test set, with the fitted line between the predicted and actual values having a positive slope whereas the FEB and GN services struggle to generalise well to unseen data as seen by the negative slope for the fitted line between predicted versus actual values.



(a) FEB service.

(b) VOI service.

FIGURE 5.6: The true versus predicted values for FEB and VOI service.



(a) BER service.

(b) GN service.

FIGURE 5.7: The true versus predicted values for BER and GN service.

Table 5.9 shows the top 10 service models based on their F-stat p-values. We see that the VOI service has the lowest F-stat p-value showing that we can be sure that at least one of the features within the VOI service is linearly correlated to the NPS survey responses. From the BER service onwards the p-value for the F-stat becomes greater than 0.048 showing that we can be less than 95% sure that one of the features in the service is linearly correlated to NPS survey responses even if there is no correlation between any of the features within the service and the NPS survey responses.

Service	Week	Logged	F_pvalue	R2	R2_adj	Nr_observations
VOI	5	True	0.000049	0.0019	0.001566	13431
VOI	5	False	0.000104	0.0017	0.001445	13431
VOI	2	True	0.001791	0.0015	0.001143	11521
VOI	2	False	0.002126	0.0015	0.001110	11521
BER	5	False	0.047840	0.0008	0.000493	11356
BER	5	True	0.054197	0.0008	0.000466	11356
BER	2	True	0.087848	0.0008	0.000425	9652
BER	2	False	0.089958	0.0008	0.000419	9652
GN	5	False	0.093932	0.0008	0.000444	7648
GN	5	True	0.114740	0.0008	0.000384	7648
GN	2	True	0.794108	0.0002	-0.000301	6542
GN	2	False	0.849101	0.0001	-0.000336	6542

TABLE 5.9: The top 10 linear regression models within service groups sorted by increasing F-stat p-value.

Table 5.10 show the classification metrics for the service level linear regression models sorted by descending F1-test scores. We see the opposite of what we see in Table 5.9 where based on F1-score, the GN service appears to perform the best. Upon further investigation, the high test recall again shows that these models predict most subscribers to be promoters with the model having very few false negatives.

Service	Week	Logarithm	Train F1	Train Recall	Train Precision	Train Accuracy	Test F1	Test Recall	Test Precision	Test Accuracy
GN	5	False	0.66	0.73	0.60	0.55	0.71	0.88	0.59	0.57
GN	5	False	0.66	0.73	0.60	0.55	0.71	0.88	0.59	0.57
GN	5	True	0.66	0.73	0.60	0.55	0.70	0.87	0.59	0.57
GN	5	True	0.66	0.73	0.60	0.55	0.70	0.87	0.59	0.57
GN	2	False	0.64	0.69	0.60	0.54	0.67	0.73	0.62	0.57
GN	2	False	0.64	0.68	0.60	0.54	0.67	0.73	0.62	0.57
GN	2	True	0.64	0.69	0.60	0.54	0.66	0.71	0.62	0.57
GN	2	True	0.64	0.68	0.60	0.54	0.66	0.71	0.62	0.57
BER	5	False	0.63	0.68	0.59	0.54	0.65	0.72	0.60	0.54
BER	5	False	0.64	0.69	0.59	0.54	0.65	0.72	0.60	0.54
BER	5	True	0.64	0.69	0.59	0.54	0.65	0.71	0.60	0.54
BER	5	True	0.63	0.68	0.59	0.54	0.65	0.71	0.60	0.54
BER	2	False	0.65	0.70	0.60	0.55	0.64	0.72	0.58	0.54
BER	2	True	0.65	0.70	0.60	0.55	0.64	0.70	0.58	0.54
BER	2	False	0.65	0.70	0.60	0.55	0.64	0.72	0.58	0.54

TABLE 5.10: The performance metrics for the linear regression predictions converted to a binary classes per service sorted by ascending F1-test.

None of these models performs better compared to any of the null models shown in Table 5.2 based on accuracy, F1-score or recall. However, some of the GN and BER service

models have got a slightly better precision compared to the null models which all have a precision around 0.59.

5.1.4 Summary

Regression Results Table 5.11 and Table 5.1.4 shows the results for all the linear regression models sorted by increasing F-stat p-value and decreasing R^2 respectively.

Model	Week	Logarithm	R2	F_pvalue	R2_adj	Nr_observations
VOI	5	True	0.0019	0.000049	0.001566	13431
VOI	5	False	0.0017	0.000104	0.001445	13431
CDR	5	True	0.0011	0.000154	0.000992	13431
CDR	2	True	0.0012	0.000208	0.001107	11521
CDR	2	False	0.0012	0.000223	0.001096	11521
CDR	5	False	0.0010	0.000228	0.000937	13431
FEB	5	True	0.0059	0.000273	0.004039	5876
FEB	5	False	0.0054	0.000892	0.003500	5876
VOI	2	True	0.0015	0.001791	0.001143	11521
VOI	2	False	0.0015	0.002126	0.001110	11521
SFR	5	True	0.0006	0.003301	0.000568	13431
SFR	5	False	0.0005	0.006752	0.000472	13431
S_LATENCY	5	False	0.0006	0.030073	0.000485	7648
MT_DUR	5	False	0.0003	0.032505	0.000266	13431
MO_DUR	5	True	0.0003	0.033150	0.000263	13431

TABLE 5.11: The top 15 feature and service results sorted by increasing f-stat p-value.

The VOI service in Table 5.11 has the lowest p-value for the F-stat showing that there is at least one variable within the VOI service that has a non zero β-coefficient. Looking at the third entry in Table 5.11 we see that the CDR is most likely the feature within the VOI service that has some linear correlation with NPS survey responses. With an F-stat p-value of 0.000154, there is a 99.9846% chance that the CDR feature has a non-zero β-coefficient in the population.

It is interesting to note that for the linear regression models the 5-week logged datasets seem to perform better based on F-stat p-values, showing that the longer the period of data we use, the better our models. Also, taking the logarithm of the features can amplify the effect of small nuances and give the feature more predictive power.

In Table we see that the FEB service has the highest R^2_{adj} followed by the VOI service and then the CDR feature. Although the R^2_{adj} metric should account for additional predictors, from the table, it is apparent that the FEB service dataset with all 11 features explains the most variance, around 0.4039%, of the variance in NPS survey responses, as expected with so many features.

Model	Week	Logarithm	R2	F_pvalue	R2_adj	Nr_observations
FEB	5	True	0.0059	0.000273	0.004039	5876
FEB	5	False	0.0054	0.000892	0.003500	5876
FEB	2	False	0.0047	0.051550	0.002052	4182
FEB	2	True	0.0043	0.078385	0.001710	4182
VOI	5	True	0.0019	0.000049	0.001566	13431
VOI	5	False	0.0017	0.000104	0.001445	13431
VOI	2	True	0.0015	0.001791	0.001143	11521
VOI	2	False	0.0015	0.002126	0.001110	11521
CDR	2	True	0.0012	0.000208	0.001107	11521
CDR	2	False	0.0012	0.000223	0.001096	11521
CDR	5	True	0.0011	0.000154	0.000992	13431
CDR	5	False	0.0010	0.000228	0.000937	13431
SFR	5	True	0.0006	0.003301	0.000568	13431
BER	5	False	0.0008	0.047840	0.000493	11356
S_LATENCY	5	False	0.0006	0.030073	0.000485	7648

TABLE 5.12: The top 15 feature and service results sorted by descreasing R^2_{adj}.

We see similar results in Table as in Table 5.11 with the VOI service explaining the second most variance followed by the CDR feature. What is noteworthy is that both the FEB and VOI services have more than one feature, whereas the CDR model only contains 1 feature showing, based on the R^2_{adj} metric, that the *Call Drop Rate* feature is the best to use when modelling NPS survey responses in a linear regression model.

Classification Results Table 5.13 shows the classification metrics for the regression results cast to binary classes sorted by descending F1-score. We see very different results compared to the best regression models based on regression metrics, and we have attributed the high F1-score to the high recall which is due to a high number of promoters predictions of subscribers due to our mean predicted NPS cut-off.

Model	Week	Logarithm	Train F1	Train Recall	Train Precision	Train Accuracy	Test F1	Test Recall	Test Precision	Test Accuracy
PDP_CRE_SR	2	False	0.74	0.99	0.60	0.59	0.74	0.99	0.59	0.59
PDP_CRE_SR	2	True	0.74	0.99	0.60	0.59	0.74	0.99	0.59	0.59
S_LATENCY	2	True	0.72	0.92	0.59	0.58	0.73	0.94	0.60	0.58
S_LATENCY	2	False	0.72	0.92	0.59	0.58	0.73	0.94	0.60	0.58
S_LATENCY	5	False	0.72	0.93	0.59	0.58	0.72	0.91	0.59	0.57
ATT_SR	5	False	0.70	0.87	0.59	0.57	0.71	0.87	0.60	0.58
ATT_SR	5	True	0.70	0.87	0.59	0.57	0.71	0.87	0.60	0.58
ATT_SR	2	False	0.73	0.92	0.60	0.59	0.71	0.92	0.58	0.57
ATT_SR	2	True	0.73	0.92	0.60	0.59	0.71	0.92	0.58	0.57
GN	5	False	0.66	0.73	0.60	0.55	0.71	0.88	0.59	0.57
GN	5	False	0.66	0.73	0.60	0.55	0.71	0.88	0.59	0.57
S_LATENCY	5	True	0.72	0.92	0.59	0.57	0.71	0.90	0.59	0.57
GN	5	True	0.66	0.73	0.60	0.55	0.70	0.87	0.59	0.57
GN	5	True	0.66	0.73	0.60	0.55	0.70	0.87	0.59	0.57
GN	2	False	0.64	0.68	0.60	0.54	0.67	0.73	0.62	0.57

TABLE 5.13: The top 15 feature and service regression predictions converted to binary classes sorted by descreasing test F1-score.

From the evidence at hand, the approach of converting our linear regression predictions to binary classes does not perform well and does not provide any additional insights as none of the models outperforms any of the null models.

5.2 Logistic Regression

For our simple logistic regression on the entire February dataset, we see similar results as with our linear regression model. The simple logistic regression model fit to the February dataset has a test accuracy of 0.579, a test recall of 0.943, a test precision of 0.591 and a test F1-score of 0.727.

In Table 5.14 we sort the features by their associated p-values and see that the ATT_SR, PDP_ACT_SR, CDR, SFR, S_LATENCY and MO_DUR (our short-listed features from our linear regression models) are in the top 7 most significant features - excluding the bias term.

As we would like to asses the impact of a feature on the NPS survey scores, we transform the β-coefficients using e^{B_j}. We can then interpret increasing X_j by one unit as increases the odds of $Y = 1$ by a multiple of e^{B_j} holding all other predictors constant.

Feature	Coefficient	p-value	e^{β}
ATT_SR	1.196579	0.035102	1.012038
PDP_ACT_SR	-0.322848	0.052696	0.996777
CDR	-2.773286	0.072098	0.972648
CONST	-1.415139	0.093508	0.985948
SFR	-1.976750	0.112358	0.980427
S_LATENCY	-4.207195	0.115815	0.958801
MO_DUR	0.629135	0.127923	1.006311
PDP_CRE_SR	0.924477	0.146109	1.009288
PDP_DUR	-1.805833	0.215179	0.982104
ATT_DUR	0.353164	0.691869	1.003538
MT_DUR	-0.167132	0.760162	0.998330
C_LATENCY	-0.139686	0.885679	0.998604

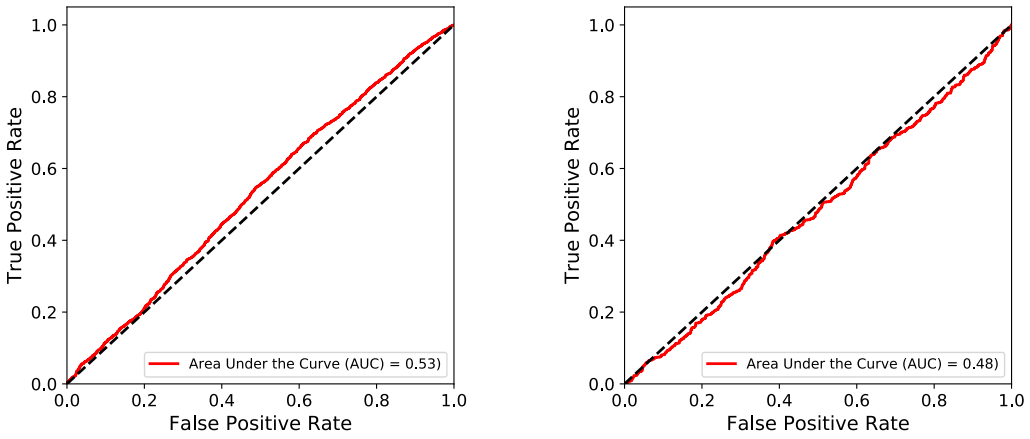
TABLE 5.14: The logistic regression features with their p-values and transformed β-coefficients.

From Table 5.14 we see that if we increase the *Attach Success Rate* of a subscriber by 1 per cent the odds of them being a promoter increases by 1.012 holding all other features constant.

As with our simple linear regression, we see the unintuitive relationship between the *PDP Attach Success Rate* feature and NPS responses. If we increase *PDP Attach Success Rate* with 1 per cent - this increases the odds of a subscriber being a promoter by 0.99678. As 0.99678 is less than 1, an increase in this feature causes a decrease in the odds of a subscriber being a promoter, at the same time increasing the odds of a subscriber being a detractor.

Here we also see the intuitive coefficient signs for *Call Drop Rate*, *Call Setup Failure Rate* and *Server Latency* showing that an increase in any of these features results in a decrease in the odds of a subscriber being a promoter. As the p-values for these features are higher than 0.072098, we take note of these results but with some scepticism and turn to the individual feature logistic regression analysis for more insights.

In Figure 5.8 we plot the ROC curve or the test and training set and see that the model did indeed learn some relationship between the 11 features the likelihood of a subscriber being a promoter or detractor in the training set with an area under the curve of 0.53.



(a) Training set.

(b) Testing set.

FIGURE 5.8: The ROC curves for the logistic regression model on the 5-week February 2018 dataset.

An AUC score above 0.5, however, is not the case for the test set with an AUC score of 0.48. The AUC score below 0.5 shows that random guessing to determine whether a subscriber is a promoter or a detractor outperforms this model.

Table 5.15 shows the confusion matrices for the train and test predictions made using the logistic regression model. The table shows that the logistic regression model favours predicting subscribers as promoters in both the train and test set.

(a) Training set.

		Actual	
		1	0
Predicted	1	TP 3408	FP 2365
	0	FN 47	TN 56

(b) Testing set.

		Actual	
		1	0
Predicted	1	TP 1091	FP 754
	0	FN 66	TN 35

TABLE 5.15: The confusion matrices for training and testing set for the logistic regression model fitted on the entire February 2018 dataset.

From the confusion matrices in Table 5.15 it is clear that the logistic regression model fitted on the entire February 2018 dataset performs similar to the null model as most observations are predicted to be promoters with the train set model only predicting 56 subscribers to be detractors and the test set model only predicting 35 subscribers to be detractors.

5.2.1 Ridge Regression

We perform 15 fold ridge logistic regression and present the coefficients in Figure 5.9.

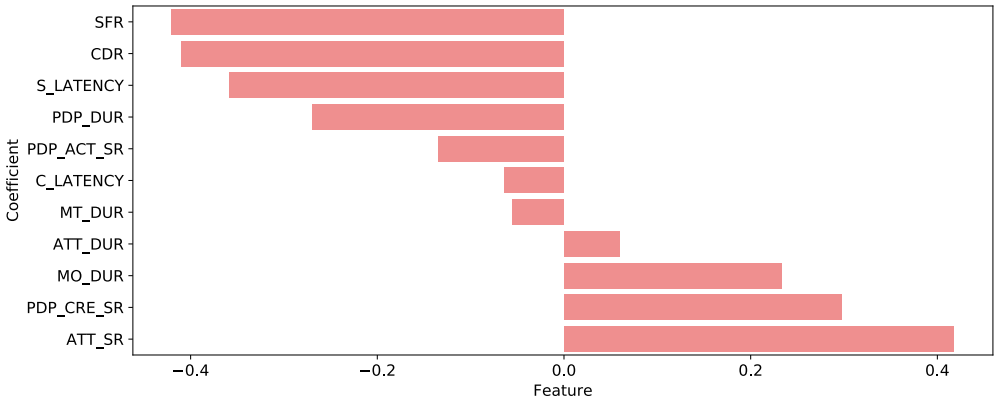


FIGURE 5.9: Coefficients after performing ridge logistic regression.

From the ridge logistic regression it appears that our most predictive features on whether a subscriber will be a promoter or a detractor are: Call Setup Failure Rate, *Call Drop Rate*, *Bearer Attach Success Rate*, *PDP Create Success Rate* and *Mobile Originating Call Duration*.

We identified all these features as significant in our linear regression analysis, all except the *PDP Create Success Rate* feature. Here the *PDP Create Success Rate* feature has

an intuitive, positive coefficient sign showing that as the *PDP Create Success Rate* of a subscriber increases so does his odds of them being a promoter.

5.2.2 Individual Feature Regression

Table 5.16 shows the β-coefficients and p-values for features having a p-value less than 0.172.

Feature	Week	Logged	Coefficient	p-value
CDR	5	True	-3.2520	0.0009
CDR	5	False	-2.9416	0.0013
CDR	2	True	-2.7536	0.0025
CDR	2	False	-2.4993	0.0031
SFR	5	True	-1.8692	0.0160
SFR	5	False	-1.5780	0.0280
SFR	2	True	-1.5298	0.0481
SFR	2	False	-1.3071	0.0673
PDP_ACT_SR	5	True	-0.3376	0.0807
PDP_ACT_SR	5	False	-0.2112	0.0959
MO_DUR	5	True	0.7722	0.0967
ATT_SR	2	False	0.7053	0.1229
MO_DUR	5	False	0.6376	0.1252
ATT_SR	2	True	1.1130	0.1590
S_LATENCY	5	True	-3.1844	0.1716

TABLE 5.16: The fitted logistic regression β-coefficients and p-value for each feature.

From our feature fitted logistic regression analysis we see that the most significant features based on p-values are *Call Drop Rate* and *Call Setup Failure Rate*. The models using the 5-week aggregated data seem to be more significant; also the models using the logged values of each feature for the two features appear to be more significant within the different week breakdowns.

We ignore the features following CDR and SFR as the p-values for each of the individually fitted features exceed 0.08 showing that there is a greater than 92% chance that we have obtained these fitted β-coefficient in the sample by chance even if they were 0 in the population.

To relate the CDR and SFR coefficients of our models to the impact on a subscriber, we transform the fitted coefficients using e^{β_j}. As we scaled the CDR and SFR features from between 0 and 100 to 0 and 1, we transform the fitted β_j using $e^{0.01\times\beta_j}$.

We see that a 1 per cent increase in call drop rate increases the odds of a subscriber being a promoter by 0.9710 - effectively decreasing the odds of a subscriber being a promoter. Similarly, for the SFR feature, we see that a 1 per cent increase in call setup failure rate on increases the odds of a subscriber being a promoter by 0.9843 - effectively decreasing the odds of a subscriber being a promoter.

Feature	Week	Logged	Train AUC	Test AUC	Train F1	Train Recall	Train Precision	Train Accuracy
MO_DUR	5	True	0.517	0.508	0.751	1.000	0.601	0.601
MO_DUR	5	False	0.517	0.508	0.751	1.000	0.601	0.601
ATT_DUR	2	False	0.514	0.500	0.746	1.000	0.595	0.595
ATT_DUR	2	True	0.514	0.500	0.746	1.000	0.595	0.595
CDR	5	True	0.513	0.511	0.750	0.997	0.601	0.600
CDR	5	False	0.513	0.511	0.750	0.997	0.601	0.600
SFR	5	True	0.512	0.498	0.750	0.999	0.601	0.600
SFR	5	False	0.512	0.498	0.750	0.999	0.601	0.600
MO_DUR	2	False	0.511	0.529	0.755	1.000	0.607	0.607
MO_DUR	2	True	0.511	0.529	0.755	1.000	0.607	0.607

TABLE 5.17: Classification metrics for logistic regression per feature sorted by descending AUC for the training set.

Table 5.18 shows the classification metrics for the test set sorted by descending test AUC scores. Although it appears that the MO_DUR feature outperforms the training set based on AUC, the recall of 1 shows that this is due to the class in balance in the testing set favouring the correct class. However, the MO_DUR feature model does do slightly better than all the null models which all have an accuracy around 0.59 and a F1-score around 0.74.

Feature	Week	Logged	Train AUC	Test AUC	Test F1	Test Recall	Test Precision	Test Accuracy
MO_DUR	2	False	0.511	0.529	0.738	1.000	0.585	0.585
MO_DUR	2	True	0.511	0.529	0.738	1.000	0.585	0.585
C_LATENCY	2	False	0.505	0.525	0.746	1.000	0.595	0.595
C_LATENCY	2	True	0.505	0.525	0.746	1.000	0.595	0.595
CDR	5	False	0.513	0.511	0.740	0.997	0.588	0.588
CDR	5	True	0.513	0.511	0.740	0.997	0.588	0.588
S_LATENCY	2	False	0.501	0.509	0.746	1.000	0.595	0.595
S_LATENCY	2	True	0.501	0.509	0.746	1.000	0.594	0.594
MO_DUR	5	True	0.517	0.508	0.741	1.000	0.589	0.589
MO_DUR	5	False	0.517	0.508	0.741	1.000	0.589	0.589

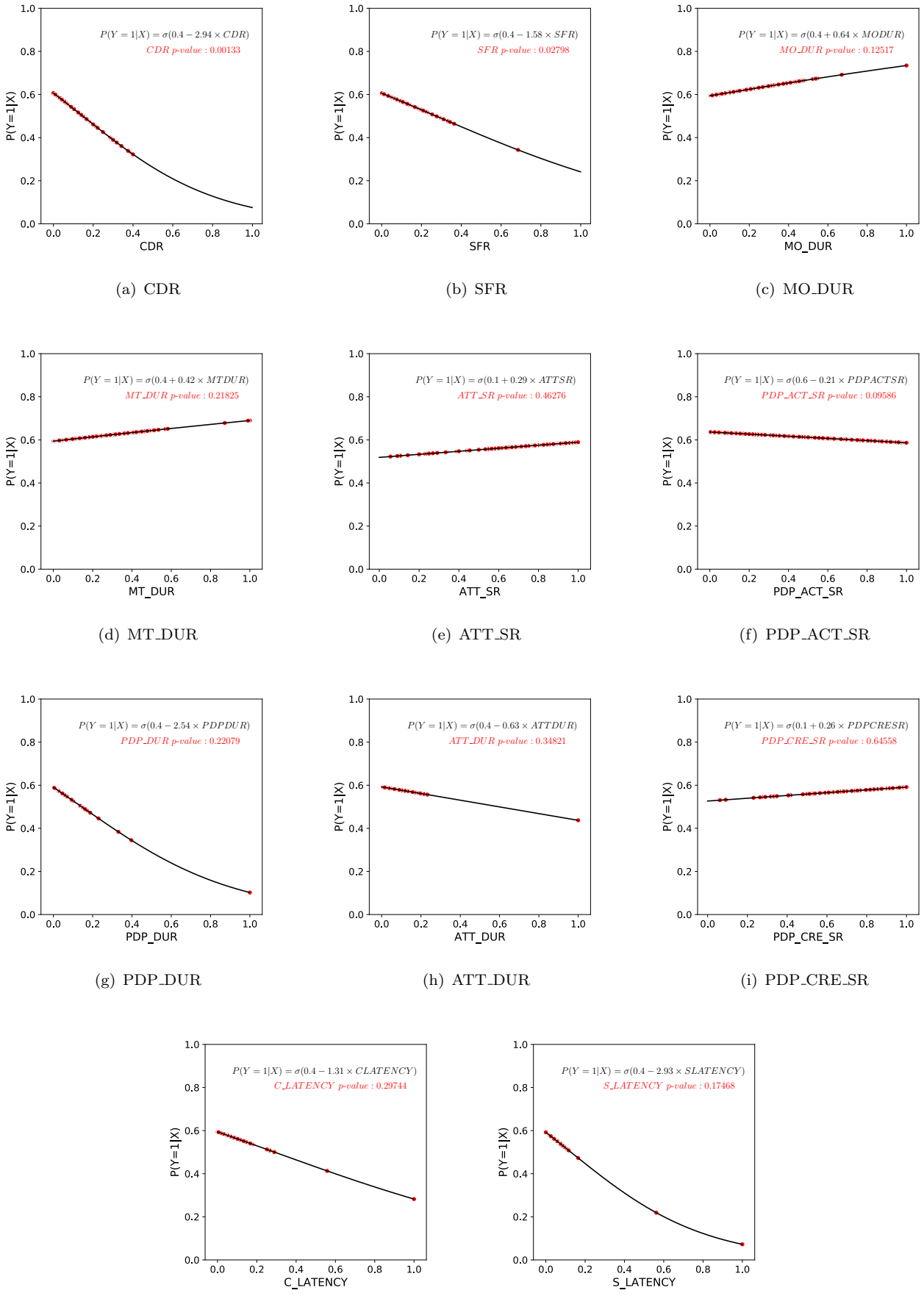
TABLE 5.18: Classification metrics for logistic regression per feature sorted by descending AUC for the testing set.

In Figure 5.10(a) to Figure 5.10(k) we plot the fitted logistic function, $P(Y = 1|X) = \sigma(\beta_0 + \beta_1 X)$ for each of the features and also show their respective p-values.

From the plots, we can see the probability of a subscriber being a promoter decreasing as the features: CDR, SFR, PDP_ACT_SR, PDP_DUR, ATT_DUR, C_LATENCY AND S_LATENCY increase. We expect a subscriber to be less likely to be a promoter if any of these features decrease, except for PDP_ACT_SR which we would expect a subscriber to be more satisfied if they have a higher success rate.

We also saw this unintuitive sign for the PDP_ACT_SR feature in our linear regression analysis showing there is some underlying factor that subscribers with a high PDP_ACT_SR have in common that might make them more likely to be a detractor. Although the fitted PDP_ACT_SR logistic regression model has a p-value of 0.09586

showing there is a 10% chance we got this coefficient estimation by chance even if it is not apparent in the population.



(a) CDR (b) SFR (c) MO_DUR

(d) MT_DUR (e) ATT_SR (f) PDP_ACT_SR

(g) PDP_DUR (h) ATT_DUR (i) PDP_CRE_SR

(j) C_LATENCY (k) S_LATENCY

FIGURE 5.10: The fitted logistic regression model per feature superimposed on the training data with the associated p-value for each feature.

5.2.3 Within Service Regression

Table 5.19 and Table 5.20 shows the classification metrics for a logistic regression model fitted to each service sorted by train AUC score and test AUC score respectively.

From Table 5.19 it appears that the 5-week FEB service with logged features was able to learn the most on the training set with a train AUC score of 0.545, followed by the 5-week VOI service with a train AUC score of 0.528. The train AUC scores that are just above 0.5 shows that the relationship between any of the feature and whether a subscriber is a promoter or detractor is challenging to model and is most likely not a linear relationship that a simple model such as logistic regression can capture fully.

Service	Week	Logged	Train size	Train AUC	Train F1	Train Recall	Train Precision	Train Accuracy
FEB	5	True	5876	0.545	0.59	0.55	0.62	0.54
FEB	2	True	4182	0.542	0.56	0.52	0.62	0.52
FEB	2	False	4182	0.533	0.57	0.52	0.62	0.52
FEB	5	False	5876	0.533	0.63	0.66	0.61	0.55
VOI	5	True	13431	0.528	0.63	0.65	0.62	0.55
VOI	5	False	13431	0.527	0.63	0.65	0.62	0.55
BER	2	True	9652	0.518	0.65	0.71	0.60	0.55
BER	2	False	9652	0.518	0.66	0.74	0.60	0.56
VOI	2	True	11521	0.518	0.65	0.69	0.62	0.55
VOI	2	False	11521	0.518	0.66	0.70	0.62	0.56

TABLE 5.19: The top 10 logistic regression models per service sorted by descending train AUC score.

From Table 5.20 it appears the learnings of the 5-week FEB service model does not get transferred to the test set with the GN service having the highest test AUC score of 0.553. For most of the services, it appears that the 2-week datasets generalise better to the test set when predicting whether a subscriber is a promoter or detractor. Sorting these models based on the AUC metric shows that none of them does better compared to the null models based on accuracy, recall or F1-score.

Service	Week	Logged	Test Size	Test AUC	Test F1	Test Recall	Test Precision	Test Accuracy
GN	2	True	2167	0.533	0.66	0.71	0.62	0.57
GN	2	False	2167	0.533	0.67	0.73	0.62	0.57
FEB	2	False	1373	0.523	0.54	0.48	0.62	0.52
VOI	2	True	3892	0.521	0.64	0.68	0.60	0.54
VOI	2	False	3892	0.519	0.64	0.69	0.60	0.55
FEB	2	True	1373	0.519	0.54	0.50	0.60	0.51
VOI	5	True	4470	0.515	0.64	0.70	0.60	0.54
VOI	5	False	4470	0.515	0.65	0.71	0.60	0.55
BER	2	True	3200	0.509	0.64	0.72	0.58	0.53
BER	2	False	3200	0.507	0.65	0.74	0.58	0.54

TABLE 5.20: The top 10 logistic regression models per service sorted by descending test AUC score.

5.2.4 Summary

Table 5.21 and Table 5.22 consolidates all the models fitted to each feature individually as well as fitted to each service. Table 5.21 sorts the models based on their train AUC score and Table 5.22 sorts them based on test AUC score.

From Table 5.21 we see the FEB service outperforming all other models with a train AUC score of 0.545 for the 5-week logged dataset. However, as we saw in the previous section, these learning in the training set could not generalise to the testing set with the 2-week GN service performing best with a test AUC score of 0.533.

Model	Week	Logged	Train Size	Train AUC	Train F1	Train Recall	Train Precision	Train Accuracy
FEB	5	True	5876	0.545	0.59	0.55	0.62	0.54
FEB	2	True	4182	0.542	0.56	0.52	0.62	0.52
FEB	2	False	4182	0.533	0.57	0.52	0.62	0.52
FEB	5	False	5876	0.533	0.63	0.66	0.61	0.55
VOI	5	True	13431	0.528	0.63	0.65	0.62	0.55
VOI	5	False	13431	0.527	0.63	0.65	0.62	0.55
BER	2	True	9652	0.518	0.65	0.71	0.60	0.55
BER	2	False	9652	0.518	0.66	0.74	0.60	0.56
VOI	2	True	11521	0.518	0.65	0.69	0.62	0.55
MO_DUR	5	True	13431	0.517	0.75	1.00	0.60	0.60
MO_DUR	5	False	13431	0.517	0.75	1.00	0.60	0.60
VOI	2	False	11521	0.517	0.66	0.70	0.62	0.56
BER	5	True	11356	0.514	0.60	0.59	0.60	0.53
ATT_DUR	2	False	9652	0.514	0.75	1.00	0.60	0.60
ATT_DUR	2	True	9652	0.514	0.75	1.00	0.60	0.60

TABLE 5.21: The top 15 logistic regression models for all features and services sorted by descending train AUC score.

From Table 5.22 it appears that based on test AUC score the best predictor to use when trying to model the relationship between whether a subscriber is a promoter or detractor is MO_DUR with a test AUC score of 0.523. Although this feature appears to generalise learnings in the training set to the testing set, the recall of 1 for both MO_DUR and C_LATENCY shows that the better than random test AUC score is due to the class in balance rather actual correct predictions. Again we note that the 2-week, shorter period datasets seem to generalise better to the testing set based on AUC test scores. Sorting these models based on the AUC metric shows that none of them does better compared to the null models based on accuracy, recall or F1-score.

Model	Week	Logged	Test Size	Test AUC	Test F1	Test Recall	Test Precision	Test Accuracy
GN	2	True	2167	0.533	0.66	0.71	0.62	0.57
GN	2	False	2167	0.533	0.67	0.73	0.62	0.57
MO_DUR	2	True	3892	0.523	0.74	1.00	0.59	0.59
MO_DUR	2	False	3892	0.523	0.74	1.00	0.59	0.59
C_LATENCY	2	True	2167	0.525	0.75	1.00	0.59	0.59
C_LATENCY	2	False	2167	0.525	0.75	1.00	0.59	0.59
FEB	2	False	1373	0.523	0.54	0.48	0.62	0.52
VOI	2	True	3892	0.521	0.64	0.68	0.60	0.54
VOI	2	False	3892	0.520	0.64	0.69	0.60	0.55
FEB	2	True	1373	0.519	0.54	0.50	0.60	0.51
VOI	5	True	4470	0.515	0.64	0.70	0.60	0.54
VOI	5	False	4470	0.515	0.65	0.71	0.60	0.55
CDR	5	True	4470	0.511	0.74	1.00	0.59	0.59
CDR	5	False	4470	0.511	0.74	1.00	0.59	0.59
S_LATENCY	2	False	2167	0.509	0.75	1.00	0.59	0.59

TABLE 5.22: The top 15 logistic regression models for all features and services sorted by descending test AUC score.

5.3 Tree Based Regression

So far, the linear models we have considered perform poorly. Can non-linear functions produce better results? In this section, we fit non-linear tree-based regression models to gain insights into how each feature partitions the subscriber base. First, we fit a decision tree regression model grown to a depth of 3 splits to all of the 11 features from the February 2018 NPS survey. The linear regression model fitted on the entire 5-week February dataset in Section 5.1 had a R^2_{adj} metric of 0.0035 and a test MSE of 14.61. Table 5.23 shows that the tree-based regression model performs better than the linear regression model based on test MSE.

The tree-based models testing set's MSE, RMSE and MAE is higher than the training set, showing that the model performs better in-sample, but does not generalise well out of sample. The model has a R^2_{adj} metric of 0.0135, which implies this model allows for the variance contained in the 11 features to explain around 1.35% of the variance in the NPS survey responses. Based on the R^2_{adj} metric the tree-based method captures more variance between the features and the NPS scores, showing that the relationship between the feature and NPS responses is more complicated than what the linear regression model can capture.

Metric	Train	Test	Difference (%)
Observations	5876	1946	-202
MSE	14.11	14.55	3.01
RMSE	3.76	3.81	1.52
MAE	3.42	3.47	1.42

TABLE 5.23: The train vs test set regression metrics for the simple regression decision tree.

To see where the decision tree partitions the feature space at each feature space partition we plot the grown tree in Figure 5.11. Figure 5.11 shows at which value of a particular feature the feature space is split and also what is the MSE, the percentage of samples and the estimated NPS survey response within each split.

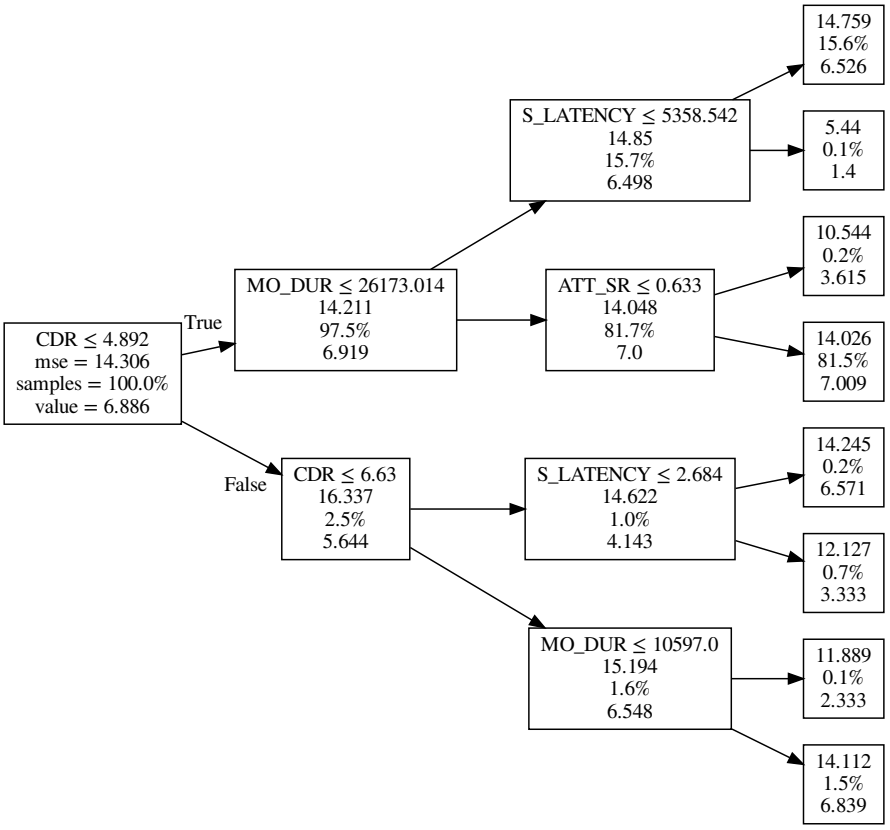


FIGURE 5.11: The 3-split deep decision tree fitted to the 5-week February 2018 dataset visualised.

From Figure 5.11 we see that the best first split leading to the most significant reduction in the MSE is partitioning the full feature space where $CDR \leq 4.89\%$. Splitting the 5876 training subscribers where $CDR \leq 4.89\%$ splits the 5876 subscribers into an internal node that has 5729 subscribers (97.5% of the training data) with an estimated NPS response of 6.919 and an internal node that has the remaining 147 subscribers (2.5% of the training data) having an estimated NPS score of 5.644. This split shows that subscribers with a call drop rate greater than 4.89% have lower estimated NPS responses on average.

Moving down the *True* branch after our initial split in Figure 5.11 we see the next split is made where $MO\_DUR \leq 26173ms$. This split agrees with what we see in our

linear regression modelling where subscribers with a high duration of mobile originating calls have higher NPS responses on average. We see that the bulk of the training observations (81.7%) fall into the group of subscribers who have $CDR \leq 4.89\%$ and $MO\_DUR \geq 26.173s$ with an estimated NPS response of 7.0.

If we continue down the *True* branch we see another intuitive split $S\_LATENCY \leq 5.358s$. The *Server Latency* feature partitions the subscriber between 917 subscribers with an estimated NPS response of 6.526 and 6 subscribers with an estimated NPS response of 1.4. Further, there is an intuitive slit at $ATT\_SR \leq 0.663$ where subscribers having a higher *Bearer Attach Success Rate* are placed into a leaf node containing 81.5% of all the training subscribers with an estimated NPS response of 7.009.

On the *False* branch after our initial split, we find an unintuitive split with the next best split on this branch occurring where $CDR \leq 6.63\%$ splitting subscribers having call drop rates lower than 6.63% into a node with an estimated NPS response of 4.143. Even though this one split is unintuitive if we keep in mind that subscribers on this branch already have a call drop rate higher than 4.892% we can expect lower than average estimated NPS responses. The remaining two splits on the *False* branch viz. $S\_LATENCY \leq 2.684s$ and $MO\_DUR \leq 10.597s$ split their respective partitioned feature spaces intuitively with subscribers having higher server latency having a lower estimated NPS response and subscribers with lower mobile originating call durations having lower NPS responses.

For our February decision tree we plot the variable importance for each feature in Figure 5.12. Here variable importance is calculated by evaluating which feature leads to a greater decrease in MSE at each split. As this decision tree is only grown to a depth of 3 splits, not all the features are used, and we can see the feature importance plot reflecting this with feature importance values assigned to only ATT_SR, $S\_LATENCY$, MO_DUR and CDR. We note that *Call Drop Rate* is the most important feature, followed by *Mobile Originating Call Duration*, *Server Latency* and *Bearer Attach Success Rate*.

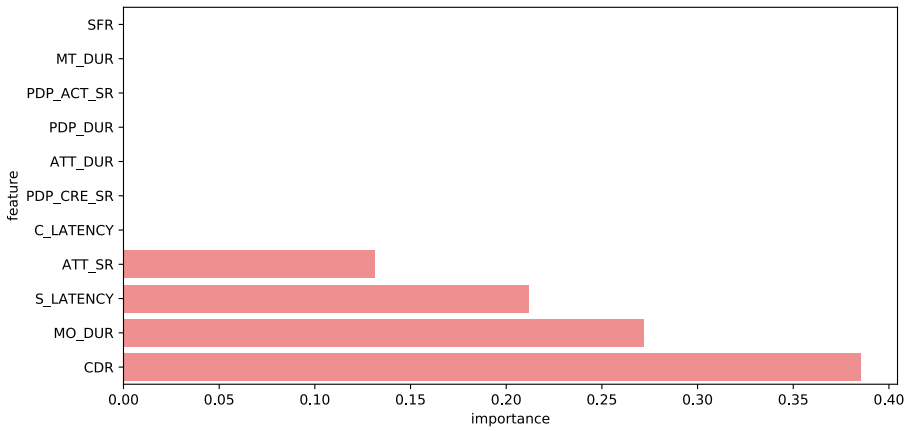
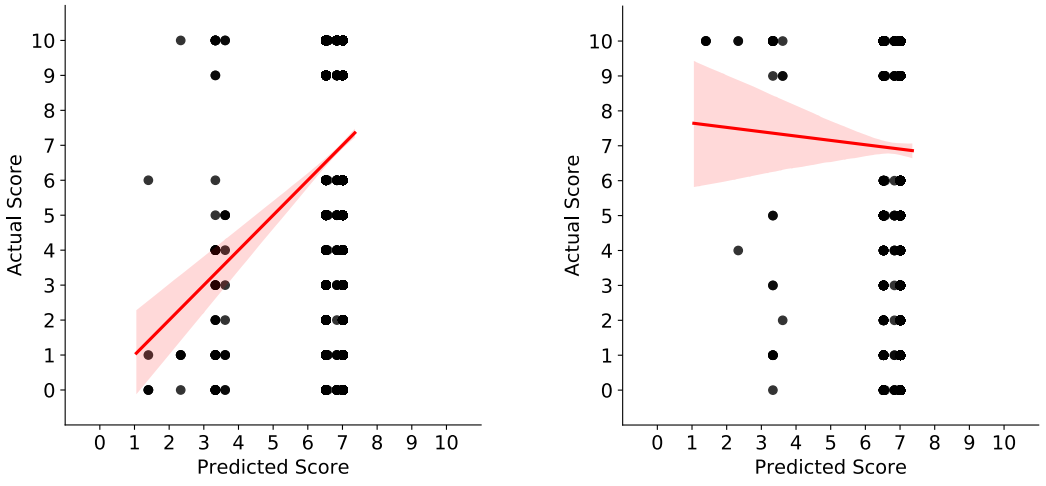


FIGURE 5.12: The feature importances of the simple decision tree fitted to the 5-week February 2018 dataset.

We plot the predicted versus actual NPS responses for the training set in Figure 5.13(a) and for the testing set in Figure 5.13(b). We see a positive correlation between the predicted and actual NPS responses for the training set, showing that the model did learn some relationship between the features and the NPS responses. However, it appears that what the model did learn from the training set could not be generalised to the testing set which has a slight negative slope.



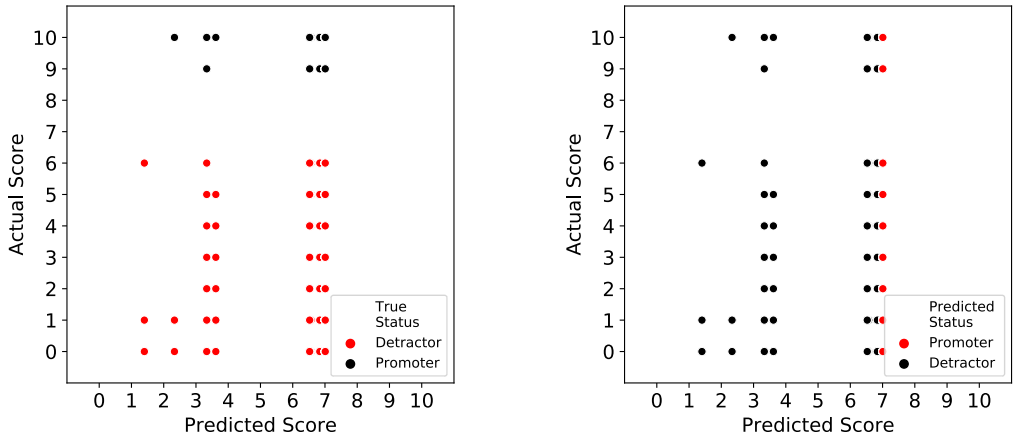
(a) Training set.

(b) Testing Set.

FIGURE 5.13: The true vs predicted values for training and testing set.

We cast our decision tree regression prediction to classification predictions by casting subscribers with a predicted NPS response greater than the average of all the predicted

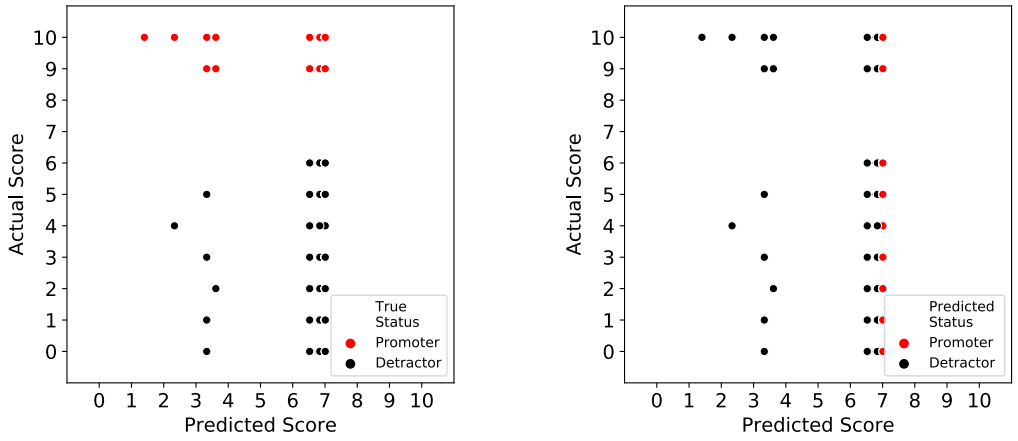
NPS responses as promoters and subscribers below the average as detractors. We plot the cast classes of each subscriber in Figure 5.14 and Figure 5.15 respectively.



(a) True status.

(b) Predicted status.

FIGURE 5.14: The regression predictions converted to classification predictions colored by the true and predicted status of each subscriber respectively for the training set.



(a) True status.

(b) Predicted status.

FIGURE 5.15: The regression predictions converted to classification predictions colored by the true and predicted status of each subscriber respectively for the testing set.

In Figure 5.14 and Figure 5.15 we see that it is impossible to draw a linear separation line on the predicted score axis that will separate subscribers who are actually promoters from those who are detractors. To evaluate the performance of our classifier we plot the confusion matrices for the training and test set in Table 5.24 and present the associated performance metrics in Table 5.25. In Table 5.25 we see that on all four classification

performance metrics the training set outperforms the testing set, showing that there is some signal that the model is picking up on, but it struggles to generalise this to unseen data.

(a) Training set.

		Actual	
		1	0
Predicted	1	TP 2890	FP 1900
	0	FN 565	TN 521

(b) Testing set.

		Actual	
		1	0
Predicted	1	TP 919	FP 629
	0	FN 238	TN 160

TABLE 5.24: The confusion matrices for train and test set for the simple decision tree fitted to the 5-week February 2018 dataset.

From Table 5.25 we see the simple decision tree model does not outperform any of the classification metrics for the null models shown in Table 5.2.

Metric	Test set	Train set	Difference (%)
Recall	0.79	0.84	-6.33
Precision	0.59	0.60	-1.69
Accuracy	0.55	0.58	-5.45
F1	0.68	0.70	-2.94

TABLE 5.25: The difference in classification metrics between the train and test set for the regression predictions converted to classification predictions.

5.3.1 Individual Feature Decision Trees

Below we fit a regression decision tree with a maximum of 3 splits to each of the features, followed by an analysis and discussion about how and where each of features partitions the subscriber base.

Call Drop Rate: In Figure 5.16 we show the decision tree grown using only the call drop rate feature. The feature partitioning in the figure is in line with our intuition and reinforces what we have seen with the linear regression model: for a higher call drop rate, a subscriber will on average have a lower NPS response.

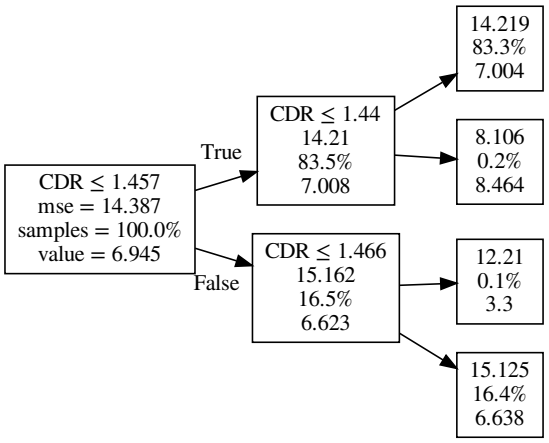


FIGURE 5.16: The tree visualised for a decision tree model fitted on the call drop rate feature.

Call Setup Failure Rate: In Figure 5.17 we show the decision tree grown using only the call setup failure rate feature. The grown tree mostly follows our intuition, with subscribers on the *True* branch after the initial split of $SFR \leq 1.794\%$ having a higher NPS response. However, on the *False* branch we see that the leaf node with 0.2% of the VOI service subscribers having $SFR \geq 21.556\%$ has an estimated NPS response of 8.3343. The 0.2% translates into 27 subscriber's NPS responses not influenced by a high call setup failure rate as we would intuitively expect.

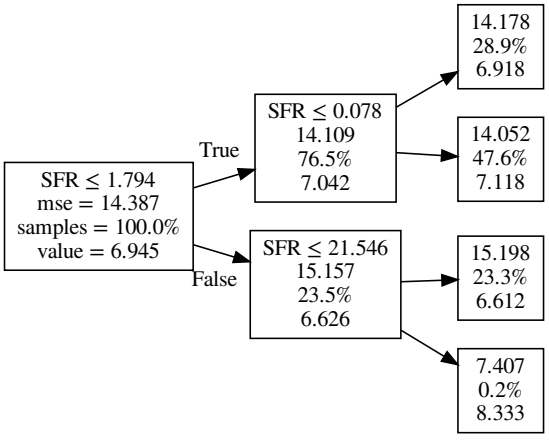


FIGURE 5.17: The tree visualised for a decision tree model fitted on the call setup failure rate feature.

Mobile Originating Call Duration: In Figure 5.18 we show the decision tree grown using only the mobile originating call duration feature. The feature space partitioning reinforces what we have seen in the linear regression model: subscribers who make longer calls have a higher NPS response on average. From Figure 5.18 we gain some insights into where some of these call duration cut-offs are.

Figure 5.18 shows that 0.4% of subscribers have an estimated NPS response of 8.429 even if they made calls for less than 3.13 seconds. Also, the figure shows that the interval $3.13s \leq MO\_DUR \leq 81.4s$ has the lowest estimated NPS responses of 6.847 with the most (59.7%) of the subscribers. From the figure, we see that if a subscriber has made calls for longer than 82 seconds over the 5 weeks, they have a NPS response on average greater than 7.

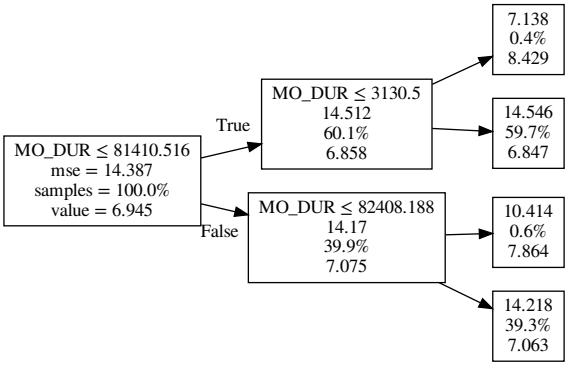


FIGURE 5.18: The tree visualised for a decision tree model fitted on the mobile originating call duration feature.

Mobile Terminating Call Duration: In Figure 5.19 we show the decision tree grown using only the mobile terminating call duration feature. The figure shows similar results to that of the linear regression model: the longer the duration of calls made to a subscriber the higher their NPS response.

Analysing the non-linear decision tree augments our insights as we can see that about 1% of subscribers having $MT\_DUR \geq 483.9s$ have a NPS response greater than 7. As the average NPS response for the dataset is 6.945 (seen in the root node as *value*), a reduction of 0.007 due to partitioning the feature space on MT_DUR shows that this feature does not influence a subscriber's NPS response as much as other features.

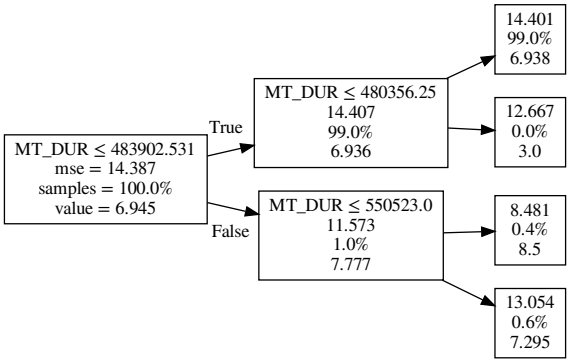


FIGURE 5.19: The tree visualised for a decision tree model fitted on the mobile terminating call duration feature.

Bearer Attach Success Rate: In Figure 5.20 we show the decision tree grown using only the bearer attach success rate feature. The figure shows similar results as the linear regression model: as the *Bearer Attach Success Rate* increases, so does the NPS response for a subscriber. From the non-linear decision tree, we see that the interval where most subscribers (97.5%) lie with regards to bearer attach success rate is the interval $91.6\% \leq ATT\_SR \leq 92.9\%$. In this majority partition, the estimated NPS response is the same as the average for the entire dataset, 6.88. This average shows that there are a few subscribers (2.5%) that have NPS responses influenced by their bearer attach success rate, but for the majority, this feature does not influence there promoter/detractor status.

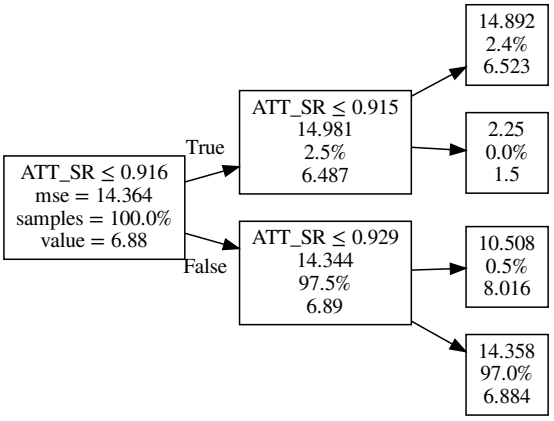


FIGURE 5.20: The tree visualised for a decision tree model fitted on the bearer attach success rate feature.

PDP Activation Success Rate: In Figure 5.21 we show the decision tree grown using only the PDP activation success rate feature. The grown tree shows some insights as to why our linear regression model fitted the wrong intuitive sign to this feature. We see subscribers with a PDP activation success rate less than 1.6% who have an estimated NPS response of 9.8; also subscribers with a PDP activation success rates in the interval $0.016 \leq PDP\_ACT\_SR \leq 0.019$ have an estimated NPS response of 7.896. The estimated NPS response for this partition shows that subscribers with meagre PDP activation success rates have higher than average NPS responses. However, the total number of subscribers on the *True* branch after the initial split at $0.016 \leq PDP\_ACT\_SR$ is only 0.07%.

If we look at the leaf node with 99.3% of the observations, we see that the estimated NPS response for the majority of the dataset is only slightly below the average of the dataset (6.873 versus 6.88). We argue that the unintuitive sign obtained in the linear regression model is due to the 0.07% of high NPS responses with meagre PDP activation success rates, coupled with the fact that most other subscribers have an average NPS response.

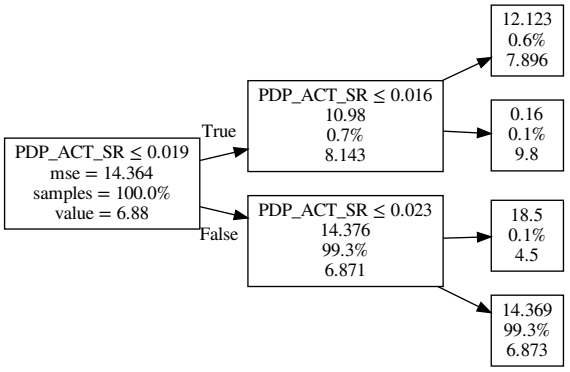


FIGURE 5.21: The tree visualised for a decision tree model fitted on the PDP activation success rate feature.

PDP Activation Duration: In Figure 5.22 we show the decision tree grown using only the PDP activation duration feature. We do not see a significant difference in the PDP activation duration after the first feature partitioning, 6.616 versus 6.91 with the dataset NPS response average being 6.88. The figure shows that the majority of the subscribers (68.2%) grouped into an interval of $543.02 \leq PDP\_ACT\_SR$. As the PDP activation duration feature has a mean of $804.15ms$, we conclude that a deeper tree is needed to say more about how this feature splits the subscriber base.

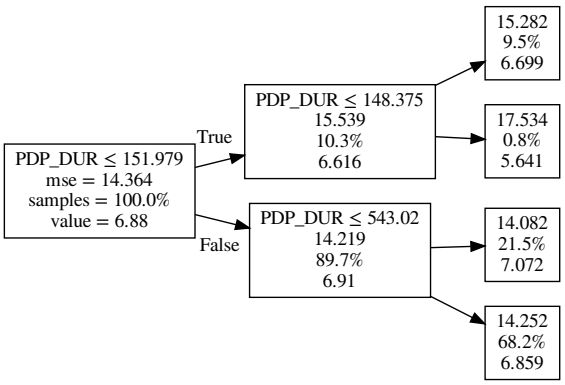


FIGURE 5.22: The tree visualised for a decision tree model fitted on the PDP activation duration feature.

Bearer Attach Duration: In Figure 5.23 we show the decision tree grown using only the bearer attach duration feature. The tree shows that 25.1% of subscribers have a bearer attach duration less than $1.1s$ and these subscribers have a NPS response of 7.02 which is higher than the average 6.88. Further, of the 25.1% subscribers, 86% (the 21.7% terminal node) have a bearer attach duration less than $1.057s$.

However, this lower bearer attach duration does not lead to significantly reduced NPS responses, with subscribers in the interval $1056.748 \leq ATT\_DUR \leq 1098.652$ having a higher estimated NPS response compared to subscribers with an activation duration less than $1.057s$. For the remaining 74.5% subscribers who have a bearer activation duration greater than $1.103s$, the tree shows that their NPS responses are on par with the average NPS response, showing that their NPS response is not influenced by their bearer activation duration.

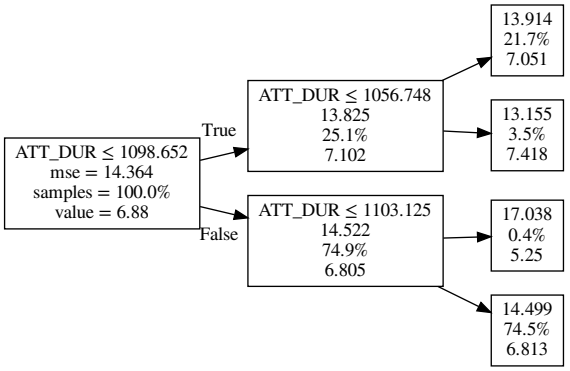


FIGURE 5.23: The tree visualised for a decision tree model fitted on the bearer attach duration feature.

PDP Create Success Rate: In Figure 5.24 we show the decision tree grown using only the PDP create success rate feature. As the mean for this feature is 0.996, and the first quartile is 1, the tree grown only highlights what is going on in the tail (< 0.441) of the feature space. Due to the tree only looking at a small part of the subscriber base, we conclude that a deeper tree is needed to say more about how this feature splits the subscriber base.

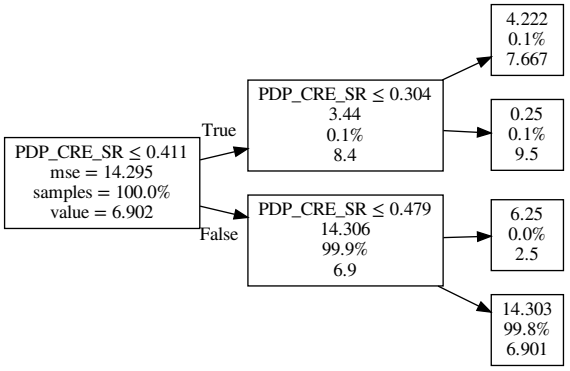


FIGURE 5.24: The tree visualised for a decision tree model fitted on the call drop rate feature.

Client Latency: In Figure 5.25 we show the decision tree grown using only the client latency feature. We see an intuitive first split at $C\_LATENCY \leq 45.462ms$ which separates the subscriber base into 2% and 98% respectively. The 2% of subscribers having a client latency less than $45.462ms$ has an estimated NPS response of 7.588, 0.686 higher than the average, whereas the 98% of subscribers have an estimated NPS response of 6.888, about the same as the average.

As with the *PDP Create Success Rate* feature, we only see the tail of the C_LATENCY feature as 92.8% of the subscribers have a client latency greater than 65.236ms. To make any further conclusions about how *PDP Create Success Rate* is related to NPS responses a deeper tree is needed.

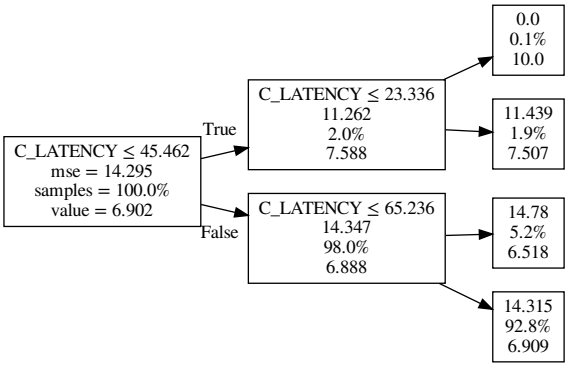


FIGURE 5.25: The tree visualised for a decision tree model fitted on the client latency feature.

Server Latency: In Figure 5.26 we show the decision tree grown using only the server latency feature. The figure shows an intuitive first split ($S\_LATENCY \leq 2.617ms$) which splits the subscriber base into 25.8% and 74.2% respectively. The 25.8% of subscribers with a server latency less than 2.617s have an estimated NPS response of 7.087, 0.185 higher than the average. Conversely, the 74.2% of subscribers with a server latency greater than 2.617s have an estimate NPS response of 6.838, 0.064 lower than the average.

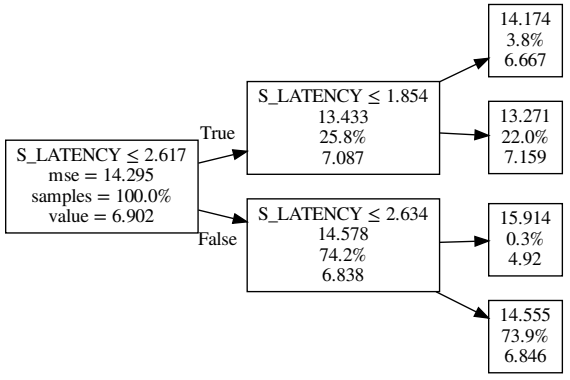


FIGURE 5.26: The tree visualised for a decision tree model fitted on the server latency feature.

Table 5.26 show the top 15 individual feature decision trees grown sorted by descending R^2_{adj}. The table shows that when fitting an individual decision tree the variance in the *Server Latency* feature describes the most variance in NPS responses with an R^2_{adj} of 0.28%.

The features in Table 5.26 contain all the features identified by the linear regression model as significantly correlated. *Call Drop Rate*, *Call Setup Failure Rate*, *Server Latency*, *Mobile Originating and Terminating Duration*, *Bearer Attach Success Rate* and *PDP Activation Success Rate* are all the features who had a p-values below 0.0525 when we fitted the individual feature linear regression models. Table 5.26 shows that the non-linear decision tree model also detected this correlation between the *Call Drop Rate*, *Call Setup Failure Rate*, *Server Latency* and *Mobile Originating and Terminating Duration* features and NPS responses.

Feature	Week	$R2$	R^2_{adj}	Number of Observations
S_LATENCY	2	0.00326	0.00280	6542
SFR	5	0.00307	0.00278	13431
SFR	2	0.00244	0.00209	11521
CDR	5	0.00230	0.00200	13431
CDR	2	0.00233	0.00198	11521
S_LATENCY	5	0.00222	0.00183	7648
MT_DUR	2	0.00216	0.00182	11521
ATT_SR	2	0.00216	0.00175	9652
ATT_DUR	5	0.00203	0.00168	11356
ATT_DUR	2	0.00201	0.00159	9652
C_LATENCY	2	0.00200	0.00155	6542
MO_DUR	5	0.00177	0.00147	13431
PDP_DUR	5	0.00166	0.00130	11356
PDP_DUR	2	0.00169	0.00128	9652
PDP_CRE_SR	2	0.00166	0.00121	6542

TABLE 5.26: The top 15 individual feature decision trees sorted by descending R^2_{adj}.

Table 5.27 shows the classification metrics for the regression predictions of each individual feature decision tree converted to binary outcomes based on the mean of the predicted values. Table 5.27 shows the models ranked by descending test F1 scores, and we notice a big pitfall with the approach of converting regression results to classification results when using shallow grown decision trees.

As trees with a depth of 2 only have 4 terminal nodes, the potential estimated NPS response values only have 4 possible values, irrespective of how many subscribers are in each terminal node. The impact of this is that we see many of the single feature decision trees with a recall of either 0.00 or 1.00 as there are either no true positives in the case where recall takes on a value of 0.00 or there are no false negatives when the recall is 1.00.

Feature	Week	Train F1	Train Recall	Train Precision	Train Accuracy	Test F1	Test Recall	Test Precision	Test Accuracy
PDP_CRE_SR	5	0.00	0.00	0.70	0.41	0.75	1.00	0.59	0.59
PDP_CRE_SR	2	0.75	1.00	0.60	0.60	0.75	1.00	0.60	0.60
C_LATENCY	2	0.74	0.98	0.60	0.60	0.74	0.98	0.59	0.59
C_LATENCY	5	0.73	0.95	0.59	0.59	0.73	0.95	0.59	0.58
ATT_SR	2	0.73	0.92	0.60	0.59	0.71	0.92	0.58	0.57
CDR	5	0.71	0.85	0.61	0.58	0.71	0.86	0.60	0.58
CDR	2	0.70	0.82	0.61	0.58	0.69	0.83	0.59	0.57
SFR	2	0.67	0.74	0.62	0.56	0.66	0.75	0.59	0.55
MO_DUR	2	0.62	0.63	0.62	0.54	0.62	0.64	0.61	0.55
SFR	5	0.55	0.50	0.63	0.52	0.54	0.49	0.60	0.51
MT_DUR	2	0.53	0.47	0.62	0.51	0.50	0.44	0.58	0.49
MO_DUR	5	0.50	0.42	0.62	0.50	0.49	0.42	0.60	0.49
ATT_DUR	2	0.50	0.41	0.62	0.50	0.48	0.41	0.59	0.49
ATT_DUR	5	0.37	0.26	0.62	0.47	0.35	0.25	0.59	0.45
PDP_DUR	5	0.33	0.22	0.62	0.46	0.32	0.22	0.62	0.45

TABLE 5.27: The classification metrics of the top 15 individual feature regression decision trees converted to classification predictions, sorted by descending test F1-scores.

5.3.2 Within Service Random Forests

In an attempt to find the best ensemble tree model we can fit to our dataset, we perform a cross-validation grid search and pick the model with the lowest MSE based on the following hyperparameter ranges:

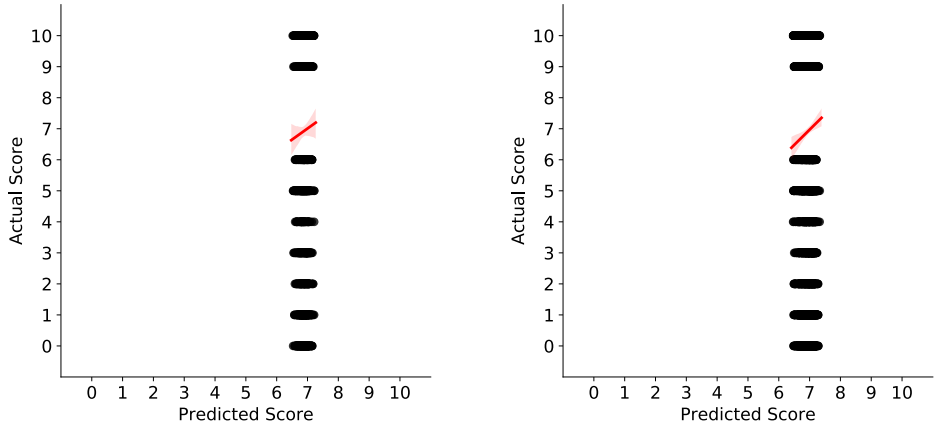
- max_features $\in$ {0.1,0.3,0.7}
- n_estimators $\in$ {10, 500, 1000}
- max_depth $\in$ {None, 5, 10, 50}
- min_samples_leaf $\in$ {1, 50, 500}
- min_samples_split $\in$ {2, 10, 50}

Table 5.28 shows the best hyperparameters for each service and week combination with the training and test set MSE and R^2 and R^2_{adj} metrics. The table is ranked by increasing MSE values, and it appears that the GN service has the lowest train MSE at 13.967, which means that this model mispredicts the NPS responses of subscribers on average by 3.74. What is interesting to note is that the 2-week datasets outperform the 5-week datasets across all the services and that the best random forest for the VOI service is only 5 splits deep with only 10 trees in the random forest.

Service	Week	Observations	Max Features	N Estimators	Max Depth	Min Samples Leaf	Min Samples Split	R^2	R^2_{adj}	Train MSE	Test MSE	MSE Diff (%)
GN	2	6542	0.1	1000	None	50	2	0.016	0.016	13.967	14.076	0.773
VOI	2	11521	0.1	10	5	500	2	0.004	0.004	14.020	14.368	2.418
FEB	2	4182	0.7	500	None	500	2	0.008	0.005	14.044	14.026	-0.130
BER	2	9652	0.1	1000	None	500	2	0.003	0.002	14.113	14.652	3.675
FEB	5	5876	0.3	1000	None	500	2	0.010	0.008	14.167	14.183	0.111
GN	5	7648	0.1	1000	None	500	2	0.002	0.001	14.273	14.094	-1.274
VOI	5	13431	0.1	1000	None	500	2	0.007	0.007	14.282	13.946	-2.410
BER	5	11356	0.1	500	5	500	2	0.003	0.003	14.314	14.191	-0.861

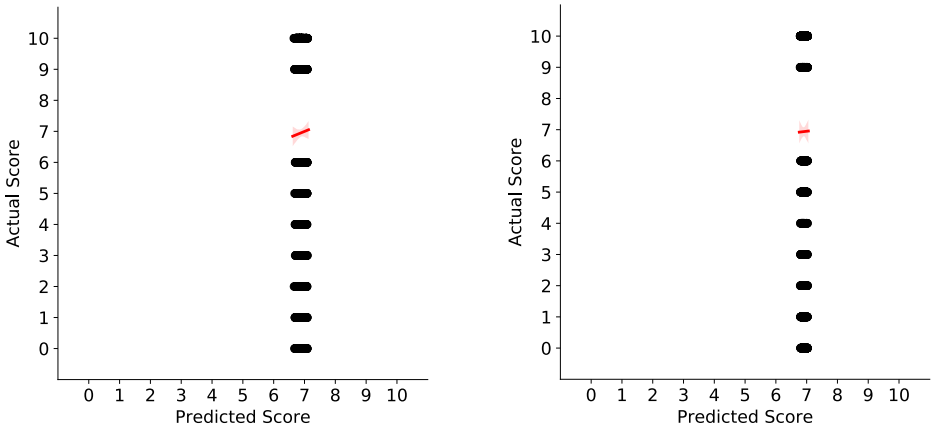
TABLE 5.28: The regression metrics for the within service random forest model along with their best grid search parameters.

For the 5 week dataset we plot the predicted versus actual NPS responses for the test set and for each service in Figure 5.27(a) to Figure 5.28(b). The figures show that all the services have a positive slope between the actual and predicted values showing that these models generalise better than any of the models we have had previously. One criticism on this model is that the range of the predicted values is tiny making it difficult to separate promoters from detractors.



(a) FEB (b) VOI

FIGURE 5.27: The true vs predicted values for the FEB and VOI service.



(a) BER (b) GN

FIGURE 5.28: The true vs predicted values for the BER and GN service.

Table 5.29 shows the F1, recall, precision and accuracy of the test and training set for the regression predictions converted to binary classes. The table shows that these classifiers perform more realistic compared to the individual feature regression classifiers as we do not have the recall jumping to 0.00 or 1.00 due to no observations cast into a particular category. This is partly due to the deeper trees as well as the trees having more features to use when partitioning the feature space. The table shows that the VOI service performs the best with a test F1 score of 0.59 and a test accuracy of 0.53 - which is still worse compared to the null model for the VOI service shown in Table 5.2.

service service	week week	train F1	train Recall	train Precision	train Accuracy	test F1	test Recall	test Precision	test Accuracy
VOI	2	0.60	0.56	0.63	0.54	0.59	0.57	0.61	0.53
VOI	5	0.61	0.59	0.63	0.55	0.59	0.58	0.61	0.53
FEB	5	0.59	0.55	0.65	0.56	0.57	0.53	0.61	0.52
FEB	2	0.59	0.54	0.64	0.55	0.56	0.52	0.61	0.52
BER	5	0.56	0.52	0.62	0.53	0.54	0.50	0.59	0.50
GN	2	0.62	0.56	0.70	0.59	0.53	0.47	0.61	0.50
BER	2	0.52	0.45	0.62	0.51	0.51	0.45	0.59	0.50
GN	5	0.50	0.43	0.61	0.50	0.51	0.43	0.61	0.50

TABLE 5.29: The classification metrics for regression predictions converted to binary classes sorted by descending test F1 score.

5.3.3 Summary

The tree-based regression analysis shows that for the 5-week February dataset the most important features are ATT_SR, $S\_LATENCY$, MO_DUR and CDR. On the 5-week February dataset, we fit a simple decision tree with 3 splits that captures 1.35% of the variance in NPS responses and mispredicts the NPS score of a subscriber in the test set by 3.81 on average.

The decision tree analysis on each of the individual features shows that the 4 features which had p-values below 0.0525 for the linear regression models and also describe the most variance in NPS responses in the non-linear setting are: *Call Drop Rate*, *Call Setup Failure Rate*, *Server Latency* and *Bearer Attach Success Rate*.

Performing a grid search for the best hyperparameters based on the lowest test MSE, we found a random forest model, fit using the GN service, that has a MSE of 14.076 ($\sqrt{14.076} = 3.75$) and captures 1.6% of the variance in NPS responses.

As with the linear regression models, we see that converting our regression models to classification models using the mean NPS as a cut-off does not work well. None of the models outperformed the null models based on accuracy, precision, recall or F1-score.

5.4 Tree Based Classification

In this section, we fit non-linear tree-based classification models to gain insights into how each feature partitions the subscriber base. First, we fit a classification decision tree with 3 splits to the entire 5-week February dataset to see which features are partitioning the promoter detractor feature space.

Table 5.30 shows the AUC, F1, recall, precision and accuracy for the simple decision tree. Here we can see that the model is performing slightly better than the null model on the test set based on AUC with an AUC of 0.51. However, based on accuracy, recall, precision and F1-score, the model performs worse compared to all the null models. Across all the metrics, the training set performs better than the testing set showing that our model is overfitting. Table 5.30(a) and Table 5.30(b) shows the confusion matrices for the training and test set, respectively.

Metric	Train	Test	Difference (%)
AUC	0.53	0.51	-3.46
F1	0.40	0.38	-5.11
Recall	0.29	0.28	-5.57
Precision	0.64	0.61	-4.10
Accuracy	0.49	0.47	-4.36

TABLE 5.30: The classification metrics for a classification decision tree fitted to the entire 5-week February 2018 dataset.

(a) Training set.

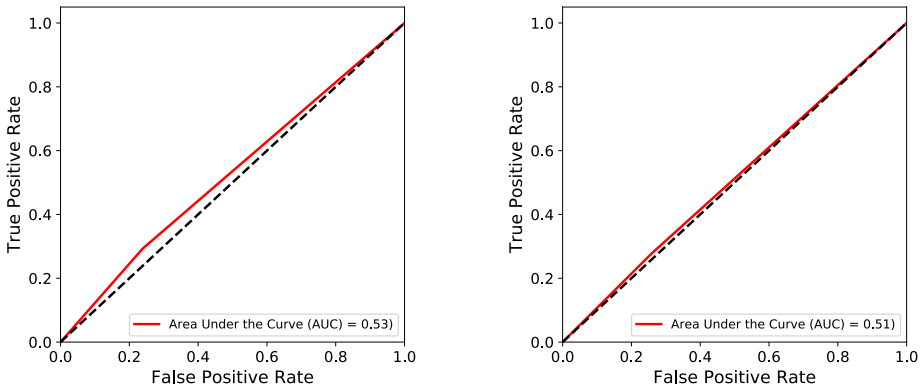
		Actual	
		1	0
Predicted	1	TP 1012	FP 578
	0	FN 2443	TN 1843

(b) Testing set.

		Actual	
		1	0
Predicted	1	TP 321	FP 204
	0	FN 836	TN 585

TABLE 5.31: The confusion matrices for the train and test set for a classifcation decision tree model fitted on the 5-week February 2018 dataset.

Figure 5.29 shows the the ROC curve for the training and testing set respectively. The figures show that the simple decision tree model does learn some mapping between the 11 features and the status of a subscriber in the training set with an AUC of 0.53; however, these learnings do not generalise to the testing set with an AUC of 0.51.



(a) Training set.

(b) Testing set.

FIGURE 5.29: The ROC curve for the training and testing set for a decision tree fitted on the 5-week February 2018 dataset.

To gain insights into where the decision tree partitions the feature space we plot the 3 split deep tree in Figure 5.30. The tree shows that the best first split leading to the most homogeneous split occurs at $MO\_DUR \leq 9.5s$. The split on mobile originating call duration divides the subscriber base into an internal node containing 3.1% of the subscriber base and another containing 96.9%.

The subscribers within the 3.1% node are made up of 62.9% detractors and 37.1% promoter, whereas the subscribers in the 96.9% node are spread more evenly with 49.6% detractors and 50.4% promoters. This tree shows that if a subscriber made less than 9.5 seconds of calls within the 5 weeks, there is a 62.9% probability that they are a detractor.

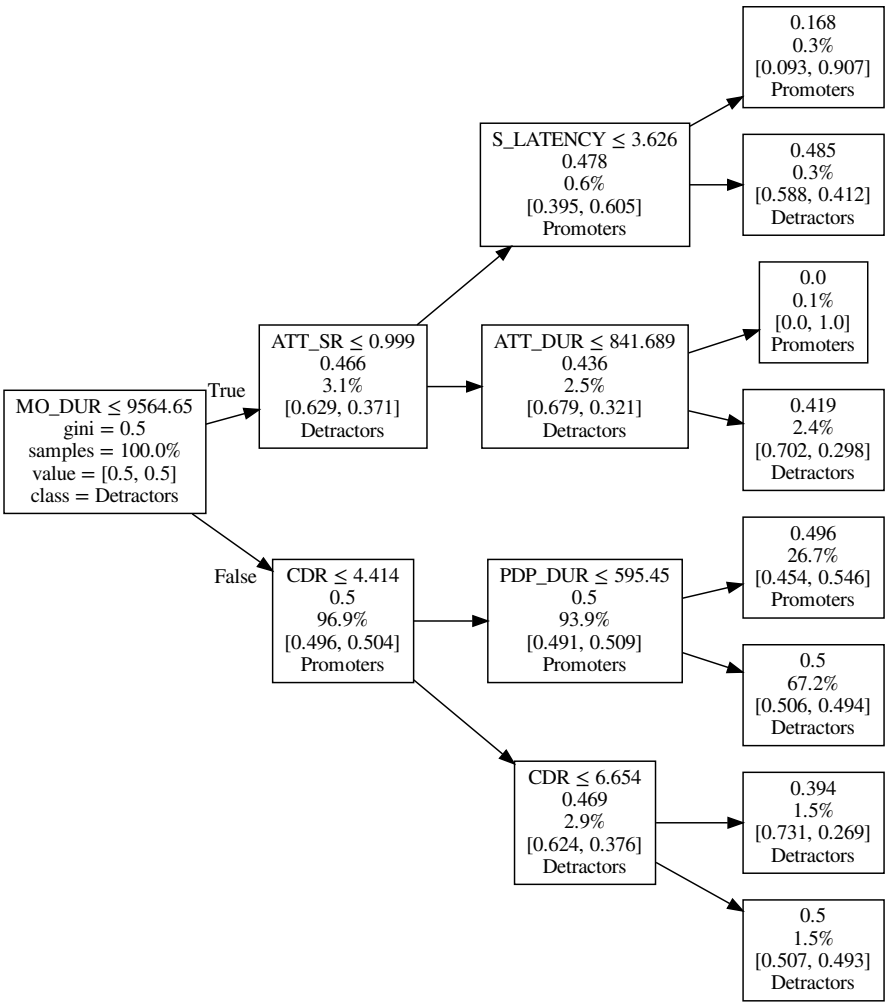


FIGURE 5.30: The Feature importances of simple decision tree fitted on the 5-week February 2018 dataset.

Continuing down the *True* branch we see the 3.1% of the subscriber base is further partitioned based on bearer attach success rate at the cut-off point $ATT\_SR \leq 0.999$. We see 0.6% of the subscriber base has a success rate less than 99.9% and even with the lower success rate, the internal node resulting from the split at $ATT\_SR \leq 0.999$ has 60.5% promoters and 39.5% detractors, showing a low bearer attach success rate is less correlated with subscribers who are detractors.

Similarly the remaining 2.5% of the original 3.1% on the *True* branch has subscribers with a bearer attach success rate greater than 99.9% and a higher probability for these subscribers to be a detractors, again showing that partitioning the feature space here based on bearer attach success rate results in the most homogeneous nodes, but does not follow our intuition of what we might expect.

We can interpret the leaf nodes of the classification decision tree as follows:

- if you are a subscriber that has made less than 9.5 seconds of originating calls and you have a bearer attach success rate less than 99.9%, and your server latency is less than $3.626ms$ then there is a 90.7% probability that you will be a promoter.
- if you are a subscriber that has made less than 9.5 seconds of originating calls and you have a bearer attach success rate less than 99.9%, and your server latency is greater than $3.626ms$ then there is a 58.8% probability that you will be a detractor.
- if you are a subscriber that has made less than 9.5 seconds of originating calls and you have a bearer attach success rate greater than 99.9%, and your bearer attach duration is less than $0.84s$ then there is a 100% probability that you will be a promoter.
- if you are a subscriber that has made less than 9.5 seconds of originating calls and you have a bearer attach success rate greater than 99.9%, and your bearer attach duration is greater than $0.84s$ then there is a 70.2% probability that you will be a detractor.

On the *False* branch after the initial split we show the next split leading the most homogeneous children nodes occurs where the call drop rate is less than 4.414%, $CDR \leq 4.414\%$. The split is intuitive with 93.9% of subscribers having a call drop rate below 4.414% being more likely to be a promoter, whereas the 2.9% of the subscriber base having a call drop rate higher than 4.414 is more likely to be a detractor.

Interpreting the leaf nodes we see that:

- if you are a subscriber that has made more than 9.5 seconds of originating calls and you have a call drop rate less than 4.414%, and your PDP setup duration is less than $0.595s$ then there is a 54.6% probability that you will be a promoter.
- if you are a subscriber that has made more than 9.5 seconds of originating calls and you have a call drop rate less than 4.414%, and your PDP setup duration is more than $0.595s$ then there is a 50.6% probability that you will be a detractor.
- if you are a subscriber that has made more than 9.5 seconds of originating calls and you have a call drop rate greater is between 4.414% and 6.654% then there is a 73.1% probability that you will be a detractor.
- if you are a subscriber that has made more than 9.5 seconds of originating calls and you have a call drop rate greater than 6.654% then there is a 50.7% probability that you will be a detractor.

To get an indication of which features results in the most significant reduction in the Gini index we plot the feature importance in Figure 5.31. As with our regression decision tree, because we grew a shallow tree, not all the feature were used, and the feature importance plot reflects this. We see that the most important feature in this model is the call drop rate of a subscriber followed by the duration of mobile originating calls.

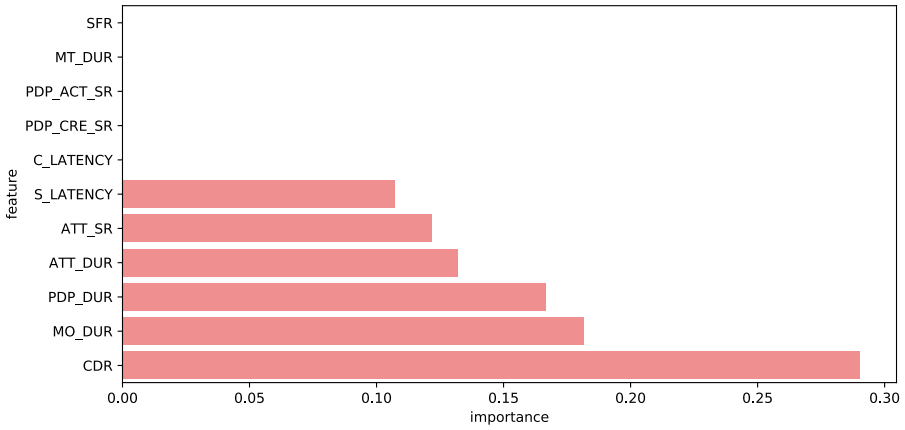


FIGURE 5.31: The feature importances of the simple decision tree fitted on 5-week February 2018 dataset.

5.4.1 Individual Feature Decision Trees

Below we fit a binary classification decision tree with a maximum of 3 splits to each of the features, followed by an analysis and discussion about how and where each of features partitions the subscriber base.

Call Drop Rate: In Figure 5.32 we show the decision tree grown using only the call drop rate feature. The first split divides the subscriber base into 89% of subscribers having a call drop rate less than 2.043% and 11% of subscribers having a call drop rate higher than 2.043%. The 89% node is almost entirely equally distributed with a Gini score of 0.5 with the node containing 49.3% detractors and 50.7% promoters. Conversely, the 11% of subscribers having a higher call drop rate is more likely to be a detractor with an estimated probability of 55.2% of being a detractor if your call drop rate is higher than 2.043%. The leaf nodes reinforce what we have seen with all our models: subscribers with a higher call drop rate are more likely to be a detractor.

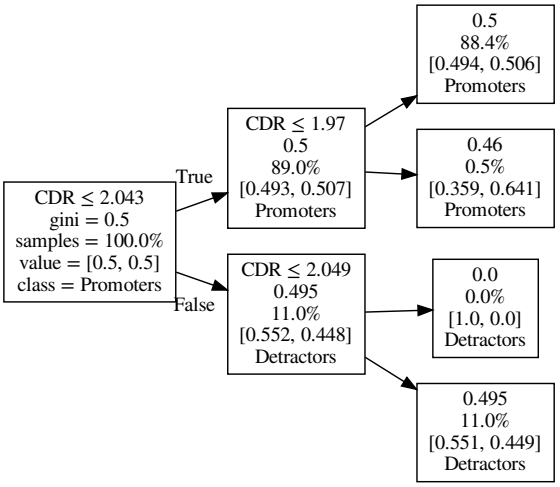


FIGURE 5.32: The tree visualised for a decision tree model fitted on the call drop rate feature.

Call Setup Failure Rate: In Figure 5.33 we show the decision tree grown using only the call setup failure rate feature. The first split divides the subscribers where $SFR \leq 1.794\%$ with 76.5% of the subscribers having a call setup failure rate less than 76.5% and these subscribers also being more likely to be a promoter. On the other side of the first split, we see 23.5% of subscribers having a call setup failure rate in the range $1.794\% \leq SFR \leq 31.942\%$ with a 54% probability of being a detractor if a subscriber's call setup failure rate falls into this range.

We see an intuitive leaf node where subscribers have a call setup failure rate lower than 0.082% being more likely to be a detractor. As the probability of being a promoter versus a detractor in this leaf node is 49% versus 51%. From this model, we see that the call setup failure rate does not influence the NPS responses for subscribers who have a call setup failure rate between 0% and 0.082 %. For the majority of the subscriber base (47.6%), we do however see that if a subscriber has a call setup failure rate in the range $0.082\% \leq SFR \leq 1.794\%$, there is a 52.6% probability that they are a promoter.

Here we can confirm the rate of call setup failures at which promoters become detractors that we saw in the the logged distribution of the *Call Setup Failure Rate* feature in Figure **??**. Figure 5.33 shows the most significant first split occurs at $SFR \leq 1.794\%$, in Figure 5.33 it appeared as if this split occurs at $log(SFR+1) = 1$, in other words $e^1 - 1 = 1,718$, which we can now put an exact value of 1.794 to.

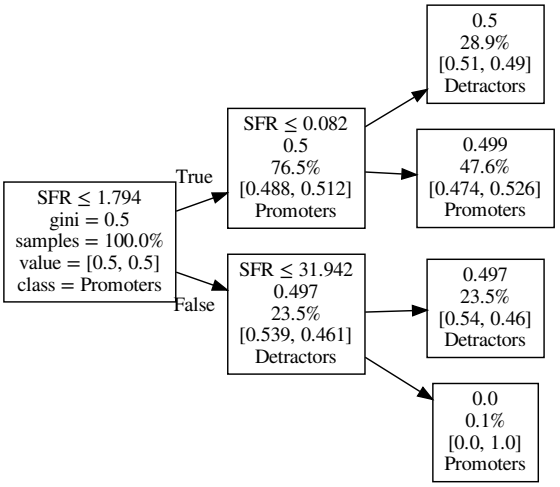


FIGURE 5.33: The tree visualised for a decision tree model fitted on the call setup failure rate feature.

Mobile Originating Call Duration: In Figure 5.34 we show the decision tree grown using only the mobile originating call duration feature. The first split dividing the subscriber base into 33.3% of subscribers having made calls lasting shorter than 45.9s during the 5 weeks agrees with what we have seen before with subscribers who make fewer calls being more likely to be a detractor, here with a probability of 52.3%. Conversely, we see subscribers who make longer calls have a higher probability to be a promoter, with 66% of the subscriber base making calls lasting longer than 46s and having a probability of 51.2% to be a promoter.

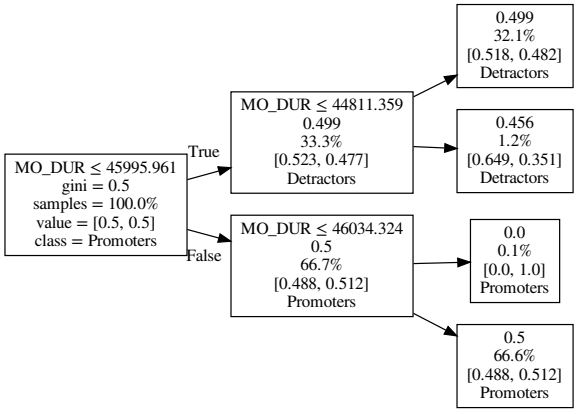


FIGURE 5.34: The tree visualised for a decision tree model fitted on the mobile originating call duration feature.

Mobile Terminating Call Duration: In Figure 5.35 we show the decision tree grown using only the mobile terminating call duration feature. We see that 99% of the of subscribers receive calls lasting less than 483s for the 5 weeks, with subscribers talking longer on received calls having a higher probability of 61.7% to be a promoter.

Of the 99.0% of subscribers having received calls lasting less than 483s for the 5 weeks, 25.5% talked for just over a minute (63.3s), and for those talking less, there is a probably of 51.8% to be a detractor. The 73.6% of subscribers that received calls lasting in the range $63.s \leq MT\_DUR \leq 483.9s$ are more likely to be a promoter, but only slightly with a probably of 50.5%.

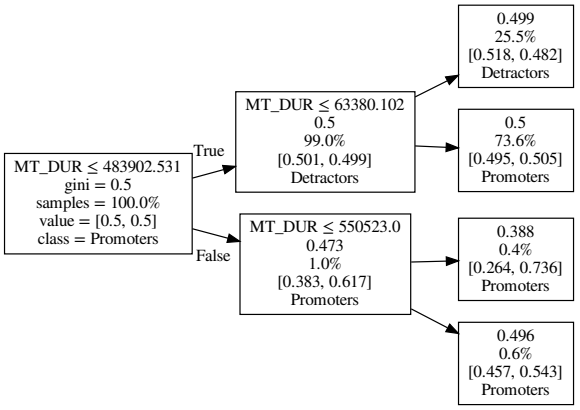


FIGURE 5.35: The tree visualised for a decision tree model fitted on the mobile terminating call duration feature.

Bearer Attach Success Rate: In Figure 5.36 we show the decision tree grown using only the bearer attach success rate feature. The first split dividing the subscriber base occurs at 89.2%. Subscribers with a success rate in the range $12.4\% \leq ATT\_SR \leq 89.2\%$ makeup 2% of the subscriber base and have a 56.9% probability of being a detractor.

Conversely, subscribers with a success rate in the range $89.2\% \leq ATT\_SR \leq 95.5\%$ makeup 2.6% of the subscriber base and have a 57.4% probability of being a promoter. The remaining 95.4% of subscribers have a success rate higher than 95.5%, and with this decision tree model, these subscribers do not tend towards either side with a node Gini score of 0.5.

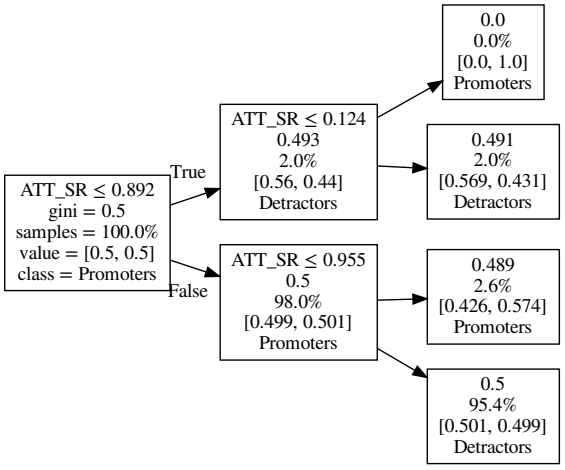


FIGURE 5.36: The tree visualised for a decision tree model fitted on the bearer attach success rate feature.

PDP Activation Success Rate: In Figure 5.37 we show the decision tree grown using only the PDP activation success rate feature. The tree shows that 0.6% of the subscriber base has a PDP activation success rate less than 1.6% and still this segment of subscribers appear to be made up by more promoters than detractors with subscribers in this node having a 63.9% probability of being a promoter.

Conversely, subscribers with a success rate greater than 1.9% do not appear to be dominated by either class with all nodes on the *False* branch after the initial split having a Gini score of 0.5. To gain any further insights into how the PDP activation success rate correlates to NPS responses a deeper tree would be needed.

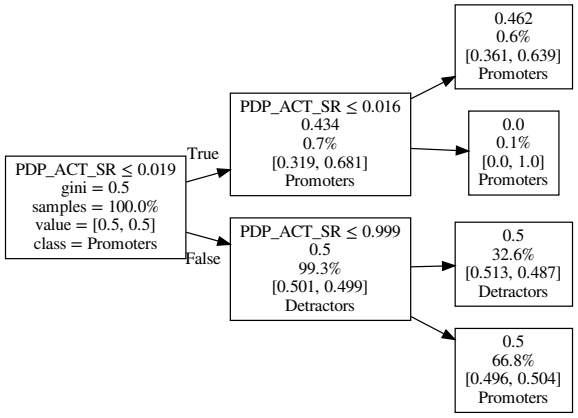


FIGURE 5.37: The tree visualised for a decision tree model fitted on the PDP activation success rate feature.

PDP Activation Duration: In Figure 5.38 we show the decision tree grown using only the PDP activation duration feature. The tree shows that the 32.6% of subscribers who have a PDP activation duration less than 0.559s are more likely to be a promoter with a 51.2% probability of being a promoter. Conversely, 64.8% of the subscriber base who has a PDP activation duration greater than 0.609s are only more likely to be a detractor with a probably of 50.4%. As most of the Gini values for nodes in this model are 0.5, the tree does not give us any insights into how the NPS responses are related to the PDP activation duration feature.

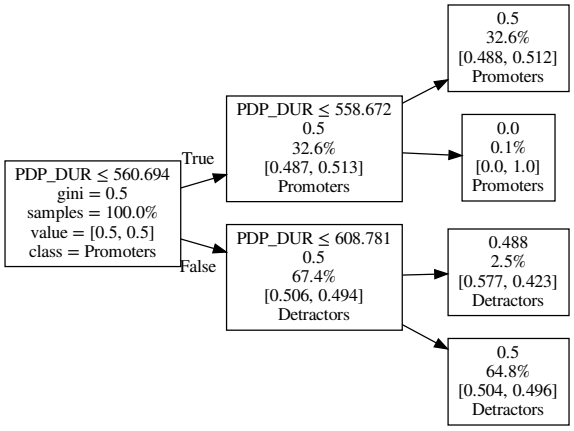


FIGURE 5.38: The tree visualised for a decision tree model fitted on the PDP activation duration feature.

Bearer Attach Duration: In Figure 5.39 we show the decision tree grown using only the bearer attach duration feature. The tree shows that the 27.4% of subscribers with a bearer attach duration of less than 1.128s have a 52.9% probability of being a promoter. Conversely, the 68.0% of subscribers with a attach duration greater than 1.204s are more likely to be detractors with a probability of 50.7%.

The tree shows that there are subscribers who notice a short bearer attach duration and are also more likely to be a promoter. On the other hand, subscribers with a high attach duration do not necessarily have low NPS responses.

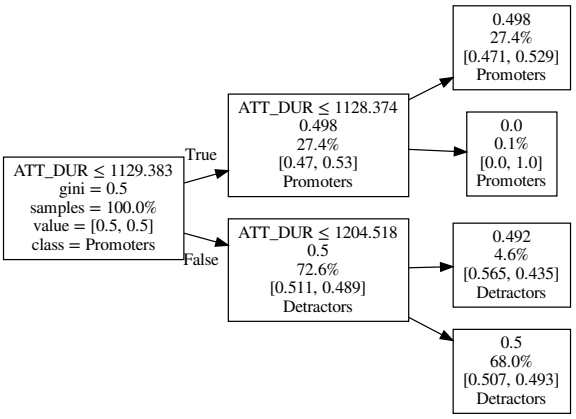


FIGURE 5.39: The tree visualised for a decision tree model fitted on the bearer attach duration feature.

PDP Create Success Rate: In Figure 5.40 we show the decision tree grown using only the PDP create success rate feature. The tree shows there is 0.7% of the subscriber base that has a PDP create success rate less than 80.9%, and it appears their NPS responses attest to the lousy success rate with subscribers having a 60% probability of being a detractor. The remaining 99.3% does not seem to influenced by a success rate greater than 80.9%.

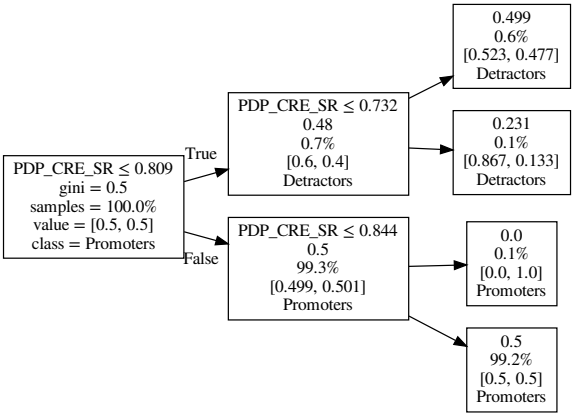


FIGURE 5.40: The tree visualised for a decision tree model fitted on the call drop rate feature.

Client Latency: In Figure 5.41 we show the decision tree grown using only the client latency feature. The tree shows that subscribers with a client latency less than $0.604s$ are more likely to be a promoter, but subscribers with a latency higher than $0.604s$ are

far more to be detractors. The 20.6% of the subscriber base with a latency higher than $0.604s$ have a probability of 52.8% to be detractors.

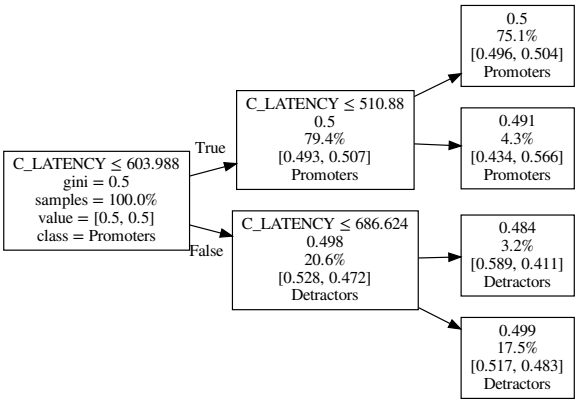


FIGURE 5.41: The tree visualised for a decision tree model fitted on the client latency feature.

Server Latency: In Figure 5.42 we show the decision tree grown using only the server latency feature. The initial split divides the subscriber base in 99.9% of the subscribers who have a server latency less than $9.35s$ and 0.01% who have a latency higher than $9.35s$. Looking further down the 99.9% split we see that the subscriber base is segmented (22.3%/77.6%) at a latency of $2.494ms$.

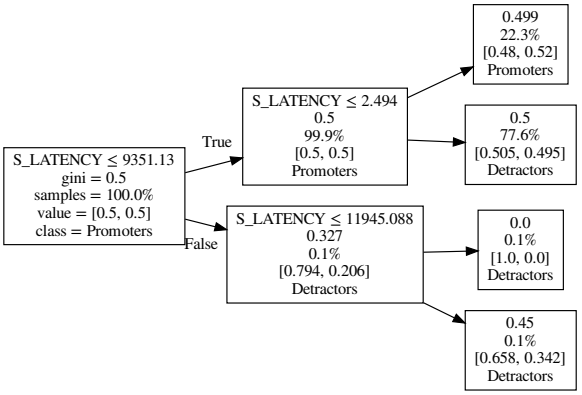


FIGURE 5.42: The tree visualised for a decision tree model fitted on the server latency feature.

In Table 5.32 we show the top 15 individual decision trees sorted by test AUC scores. From the table it appears that the 3 split decision trees could not generalise much of their training set learnings to the testing set with the S_LATENCY feature having the

best test AUC score of 0.506, only slightly better than flipping an unbiased coined. All of these models perform worse than the null models based on accuracy, precision, recall and F1-score.

Feature	Week	Train AUC	Test AUC	Train F1	Test F1	Train Recall	Test Recall	Train Precision	Test Precision	Train Accuracy	Test Accuracy
S_LATENCY	5	0.490	0.506	0.67	0.68	0.77	0.79	0.59	0.60	0.54	0.56
ATT_DUR	5	0.480	0.505	0.64	0.66	0.71	0.72	0.58	0.60	0.52	0.54
S_LATENCY	2	0.487	0.503	0.72	0.72	0.91	0.91	0.59	0.60	0.57	0.58
ATT_SR	5	0.493	0.502	0.73	0.74	0.97	0.97	0.59	0.60	0.58	0.59
PDP_ACT_SR	2	0.498	0.501	0.74	0.73	1.00	1.00	0.59	0.58	0.59	0.58
PDP_DUR	2	0.498	0.501	0.75	0.73	1.00	1.00	0.59	0.58	0.59	0.58
MT_DUR	2	0.481	0.500	0.47	0.49	0.39	0.42	0.59	0.59	0.46	0.49
MT_DUR	5	0.489	0.498	0.35	0.35	0.25	0.25	0.58	0.59	0.44	0.45
MO_DUR	5	0.483	0.497	0.41	0.43	0.32	0.34	0.58	0.58	0.45	0.47
PDP_CRE_SR	2	0.496	0.496	0.00	0.00	0.00	0.00	0.19	0.33	0.40	0.40
PDP_CRE_SR	5	0.498	0.495	0.01	0.01	0.01	0.00	0.49	0.28	0.41	0.40
PDP_DUR	5	0.489	0.495	0.62	0.63	0.67	0.67	0.58	0.59	0.52	0.53
CDR	5	0.487	0.494	0.17	0.17	0.10	0.10	0.55	0.55	0.41	0.42
ATT_SR	2	0.491	0.493	0.14	0.14	0.08	0.08	0.55	0.54	0.41	0.43
ATT_DUR	2	0.481	0.489	0.58	0.58	0.58	0.59	0.58	0.57	0.50	0.51
PDP_ACT_SR	5	0.490	0.488	0.41	0.41	0.32	0.31	0.58	0.58	0.46	0.45
CDR	2	0.487	0.487	0.28	0.26	0.18	0.17	0.58	0.55	0.42	0.43
SFR	5	0.471	0.486	0.54	0.54	0.50	0.51	0.58	0.58	0.48	0.49
C_LATENCY	5	0.483	0.485	0.29	0.28	0.20	0.19	0.56	0.56	0.44	0.43
SFR	2	0.477	0.479	0.39	0.38	0.29	0.28	0.58	0.57	0.44	0.45
MO_DUR	2	0.481	0.476	0.50	0.48	0.43	0.42	0.59	0.56	0.47	0.47
C_LATENCY	2	0.479	0.473	0.48	0.47	0.41	0.40	0.58	0.56	0.47	0.46

TABLE 5.32: The classification metrics for the top 15 individual decision trees sorted by descending test AUC score.

5.4.2 Within Service Random Forests

In an attempt to find the best ensemble tree model we can fit our dataset, we perform a cross-validation grid search and pick the model with the lowest AUC based on the following hyperparameters:

- max_features ∈ {0.1,0.3,0.7}
- n_estimators ∈ {10, 500, 1000}
- class_weight ∈ {"balanced","balanced_subsample",None}
- max_depth ∈ {None, 5, 10, 50}
- min_samples_leaf ∈ {1, 50, 500}
- min_samples_split ∈ {2, 10, 50}

Table 5.33 shows the best hyperparameters for each service and week combination with the training and test set metrics sorted by the descending test AUC score. We see that the 2-week VOI service has the highest test AUC score and the lowest difference in AUC

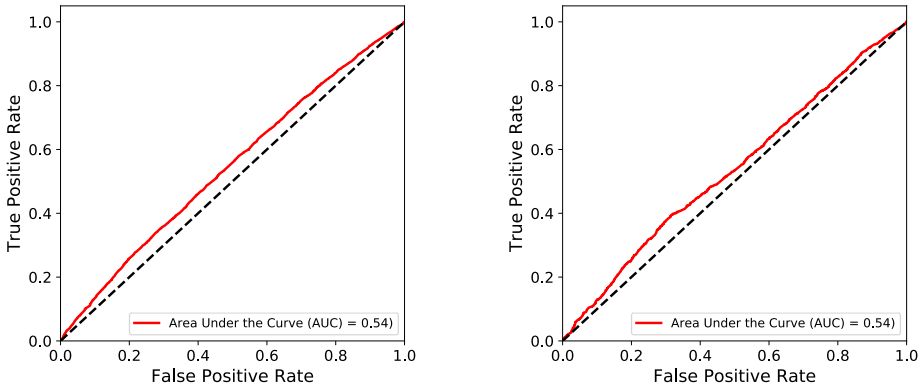
scores between the test and training set with only a 1.49% difference showing that the model learned something in the training set and was able to transfer those learnings to the testing set.

Table 5.33 shows that the best parameters for the 2-week VOI model only has 10 estimators, in other words, a random forest with only 10 trees selecting only 10% of the features at each split.

Service	Week	Observations	Max Features	N Estimators	Class Weight	Max Depth	Min Samples Leaf	Min Samples Split	AUC Train	AUC Test	Difference (%)
VOI	2	11521	0.1	10	balanced	None	500	2	0.54	0.54	-1.49
FEB	2	4182	0.7	1000	balanced_subsample	None	500	2	0.57	0.54	-6.11
VOI	5	13431	0.1	1000	balanced_subsample	None	500	2	0.56	0.53	-6.81
BER	2	9652	0.1	10	None	5	1	2	0.57	0.52	-9.62
FEB	5	5876	0.3	1000	None	None	1	10	1.00	0.51	-95.44
GN	2	6542	0.7	10	balanced_subsample	None	1	50	0.82	0.50	-63.67
BER	5	11356	0.7	10	balanced_subsample	5	500	2	0.55	0.50	-9.23
GN	5	7648	0.7	10	balanced	10	1	2	0.74	0.50	-49.22

TABLE 5.33: The grid search parameters for the best random forest, based on highest AUC, fitted to each service sorted by decreasing test AUC score.

Figure 5.43 shows the ROC curve for the training and test set for the 2-week VOI service. The figure shows that the classifier performs better than random for all possible promoter-detractor probability thresholds has the ROC curve is above the 0.5 random line for all false and true positive rates.



(a) Training set. (b) Testing set.

FIGURE 5.43: The training and testing set ROC curves for the 2-week VOI service.

In Table 5.34 we present the accuracy, precision, recall and F1-score for the test and training set for the best random forest parameters described in Table 5.33. The table shows that the 2-week VOI service has the highest test AUC score and has a test F1-score of 0.58 and a test accuracy of 0.52 showing some learned relationship between the

features in the VOI service and whether or not a subscriber is a promoter or a detractor. All of these models still perform worse than the null models based on accuracy, precision, recall and F1-score.

service	week	nr_observations	train F1	train Recall	train Precision	train Accuracy	test F1	test Recall	test Precision	test Accuracy
VOI	2	11521	0.59	0.56	0.63	0.54	0.58	0.55	0.60	0.52
FEB	2	4182	0.59	0.55	0.64	0.55	0.57	0.53	0.60	0.53
VOI	5	13431	0.60	0.56	0.64	0.55	0.58	0.56	0.61	0.53
BER	2	9652	0.75	1.00	0.60	0.60	0.73	0.99	0.58	0.58
FEB	5	5876	1.00	1.00	1.00	1.00	0.70	0.84	0.60	0.57
GN	2	6542	0.79	0.78	0.80	0.75	0.59	0.58	0.59	0.51
BER	5	11356	0.56	0.51	0.62	0.53	0.53	0.48	0.60	0.50
GN	5	7648	0.73	0.74	0.72	0.68	0.60	0.61	0.59	0.52

TABLE 5.34: The classification metrics for the decision tress fitted to each service sorted by descending test AUC score.

5.4.3 Summary

The tree-based classification analysis shows that for the 5-week February dataset the most important features in descending order are: CDR, MO_DUR, PDP_DUR, ATT_DUR, ATT_SR, S_LATENCY.

Performing a grid search for the best hyperparameters based on the lowest test AUC, we found a random forest model, fit using the VOI service, that has both a train and test AUC of 0.54 using only 10 fully grown trees, using only 1 feature (max_features=0.1) at each split.

5.5 Summary

In this section, we looked at four different models, two linear and two non-linear, tree-based models. We looked at predicting the actual NPS score as well as classifying subscribers into a binary class - promoter or detractor.

For the regression models, in general, there is a positive correlation between true and predicted values in the training data, but a negative correlation on the test set. This shows that the models do capture some relationship between performance metrics and NPS scores. However, as these models do not generalise well out of the sample, these learnings seem to be sample specific and not general relationships that we can expect in the broader population.

For the classification models, the best out of sample AUC score we were able to get was 0.54 for a hyper-parameter optimised non-linear random forest fitted to the 2-week VOI

service. Although this an AUC of 0.54 is only slightly better than random guessing, there does appear to be some predictive power in the dataset that generalised out of sample. Based on accuracy, precision, recall and F1-score all the models considered in this section perform worse than the null models. In the next section, we compare and evaluate the models more in-depth.

Chapter 6

Results, Insights and Conclusion

To tie all of the models and our analysis together, in this section, we evaluate how the various classification and regression models performed against each other. For the regression models we use the *test MSE* and R^2_{adj} to choose the best model, and for the classification models, we use the *test F1-score* and *test AUC* metric to choose the best model. We note that *best* here compares only the models considered, and a different modelling strategy could yield much better performing models. Further, we use the models to predict the NPS score, in the regression case, and the promoter status, in the classification case, for a NPS survey conducted in June 2018.

We discuss what insights we gained from the linear, and non-linear models fitted to the February 2018 dataset and discuss which models generalise the best out of sample using our hold out test set and the June 2018 survey as our out of month sample. We argue that the model that performs best out of sample, on the test set and the June dataset, best models the relationship between NPS scores and network performance metrics. Lastly, we give some concluding thoughts and some recommendations for future work.

6.1 Best Regression Models

In this section, we compare the R^2_{adj} and MSE metrics from the various regression models and motivate the selection of the best regression model. The regression models considered are made up of 14 linear regression models and 15 non-linear decision tree based models. The linear models consist of 11 simple linear regression models fitted to each feature and 3 multiple linear models fitted to each service. The tree-based non-linear models consist of 11 per feature, 3 split deep, decision tree models, and 4 hyper-parameter optimised random forests.

To differentiate between the different model types we introduce a *model type* category, $model\_type \in \{LR, DT, RF\}$, where LR denotes linear regression, DT denotes decision tree and RF denotes random forest.

In Section 5.1, we show the baseline linear regression model fit to the 5-week February dataset has an R^2_{adj} of 0.35%, a train MSE of 14.23 and a test MSE of 14.61. In Section 5.3, we show the baseline simple 3 split decision tree trained on the 5-week February dataset has a R^2_{adj} of 1.35%, a train MSE of 14.11 and a test MSE of 14.55.

Table 6.1 shows the top 5 models ranked by descending R^2_{adj}, Table 6.2 shows the top 5 models ranked by ascending train MSE, and Table 6.3 shows the top 5 models ranked by ascending test MSE.

Model	Model Type	Week	Logarithm	R^2	R^2_{adj}	Train MSE	Test MSE
GN	RF	2	False	0.0161	0.0157	13.9674	14.0762
FEB	RF	5	False	0.0097	0.0078	14.1671	14.1829
VOI	RF	5	False	0.0073	0.0070	14.2821	13.9461
FEB	RF	2	False	0.0076	0.0049	14.0438	14.0256
FEB	LR	5	True	0.0059	0.0040	14.3080	14.2978

TABLE 6.1: Top 5 regression models sorted by descending R^2_{adj}.

Model	Model Type	Week	Logarithm	R^2	R^2_{adj}	Train MSE	Test MSE
GN	RF	2	False	0.0161	0.0157	13.9674	14.0762
VOI	RF	2	False	0.0041	0.0037	14.0204	14.3678
SFR	DT	2	False	0.0024	0.0021	14.0434	14.4250
FEB	RF	2	False	0.0076	0.0049	14.0438	14.0256
CDR	DT	2	False	0.0023	0.0020	14.0448	14.4098
MT_DUR	DT	2	False	0.0022	0.0018	14.0472	14.4320

TABLE 6.2: Top 5 regression models sorted by ascending train MSE.

Table 6.1 and Table 6.2 shows that the best model based on R^2_{adj} and train MSE is a hyper parameter optimised random forest trained on the GN service which has a R^2_{adj} of 1.57%, a train MSE of 13.97 and a test MSE of 14.08.

However, Table 6.3 shows that the the GN service model does not generalise well to the February test set. We select the hyper-parameter optimised random forest trained on the 2-week VOI service as our best regression model with a test MSE of 13.95 and a R^2_{adj} of 0.7%.

Model	Model Type	Week	Logarithm	R^2	R^2_{adj}	Train MSE	Test MSE
VOI	RF	5	False	0.0073	0.0070	14.2821	13.9461
CDR	LR	5	True	0.0011	0.0010	14.3878	13.9628
CDR	LR	5	False	0.0010	0.0009	14.3878	13.9633
MT_DUR	DT	5	False	0.0012	0.0009	14.3696	13.9644
VOI	LR	5	True	0.0019	0.0016	14.3878	13.9644

TABLE 6.3: Top 5 regression models sorted by ascending test MSE.

From the test set evaluation metrics in Table 6.3, it is clear that the VOI service performs best on the test set followed by models made up of features within the VOI service (CDR, MT_DUR). The main take away from the regression models is that the features within the VOI service capture the relationship between a subscriber's NPS score and network performance best as these features make for linear and non-linear models that generalise best on unseen data.

6.2 Best Classification Models

In this section, we compare the various classification models based on the test F1 score, train AUC and test AUC. The classification models considered are made up of 14 logistic regression models and 15 non-linear decision tree based models. The linear models consist of 11 simple logistic regression models fitted to each feature and 3 multiple logistic regression models fitted to each service. The tree-based non-linear models consist of 11 per feature, 3 split deep, decision tree classifiers, and 4 hyper-parameter optimised random forest classifiers.

To differentiate between the different model types we introduce a *model type* category, $model\_type \in \{LR, DT, RF\}$, where LR denotes logistic regression, DT denotes decision tree and RF denotes random forest. To differentiate between the different classification model types we introduce a *class type* category, $class\_type \in \{clf, reg\_2\_clf\}$, where clf denotes a pure classification model, and reg_2_clf denotes regression predictions converted to classification predictions.

In Section 5.2, we show the simple logistic regression classifier fit to the 5-week February dataset has a baseline F1-score of 0.727, a train AUC of 0.53 and a test AUC of 0.48. In Section 5.4, we show the 3 split deep decision tree fit to the 5-week February dataset has a baseline test F1-score of 0.38, a train AUC of 0.53 and a test AUC of 0.51.

Table 6.4 shows the top 5 classification models ranked by test F1 score, Table 6.5 shows the top 5 classification models ranked by test AUC and Table 6.6 shows that the top 5 classification models ranked by test AUC.

Model	Model Type	Class Type	Week	Logarithm	Test F1	Test Recall	Test Precision	Test Accuracy
PDP_ACT_SR	LR	clf	5	False	0.7493	1.0000	0.5991	0.5991
PDP_ACT_SR	LR	clf	5	True	0.7493	1.0000	0.5991	0.5991
ATT_SR	LR	clf	5	False	0.7493	1.0000	0.5991	0.5991
ATT_SR	LR	clf	5	True	0.7493	1.0000	0.5991	0.5991
PDP_DUR	LR	clf	5	False	0.7491	0.9991	0.5992	0.5991

TABLE 6.4: Top 5 classification models sorted by descending test F1 Score.

Model	Model Type	Class Type	Week	Logarithm	Train AUC	Test AUC
FEB	RF	clf	5	False	1.0000	0.5117
GN	RF	clf	2	False	0.8236	0.5032
GN	RF	clf	5	False	0.7436	0.4983
BER	RF	clf	2	False	0.5685	0.5186
FEB	RF	clf	2	False	0.5680	0.5353

TABLE 6.5: Top 5 classification models sorted by descending train AUC.

Table 6.4 shows that the best model based on test F1 score is a logistic regression model fit only to the PDP_ACT_SR feature. However, upon further inspection, we see that the recall for this model is 1; in other words, the model predicts all subscribers to be promoters, which is not very useful. Further, all these models perform worse based on F1-score compared to the null models described in Table 5.2. Table 6.5 shows an example of an overfit model with the 5-week February dataset having a test AUC of 1.00, but a train AUC of 0.51, showing this model does not generalise well to unseen data.

Table 6.6 shows similar results to the regression results with the VOI service trained on the 2-week dataset generalising the best to the test set. With a train AUC of 0.5446 and test AUC of 0.5366, we select the hyper-parameter optimised random forest trained on the 2-week VOI service is the best classification model.

Model	Model Type	Class Type	Week	Logarithm	Train AUC	Test AUC
VOI	RF	clf	2	False	0.5446	0.5366
FEB	RF	clf	2	False	0.5680	0.5353
GN	LR	clf	2	True	0.5039	0.5334
GN	LR	clf	2	False	0.5037	0.5333
MO_DUR	LR	clf	2	True	0.5111	0.5289

TABLE 6.6: Top 5 classification models sorted by descending test AUC.

From the absence of reg_2_clf values in the *Class Type* column in Table 6.4 to Table 6.6 it is clear that none of the regression predictions converted to classification predictions faired well versus the pure classification models. This is expected as we have seen the implications of converting the limited numeric range predicted by the regression models in Section 5.1.4, Section 5.4.1 and Section 5.3.2.

Similarly to what we see for the best regression model in Section 6.1, using an optimised random forest and the features in the VOI service results in a model that generalises best to unseen data. Again, this shows that the features within the VOI service capture the relationship between a subscriber's NPS score and network performance best.

6.3 Out of sample - June 2018

In this section, we evaluate our models trained on the February 2018 survey on a NPS survey conducted in June 2018. We use test MSE as our regression metric to decide on a best model, and we use test AUC as our classification metric to decide on a best model.

6.3.1 Regression Performance

Table 6.7 shows the top 5 models ranked by ascending test MSE. Here test refers to the June dataset and not the holdout test set. Table 6.7 shows that a random forest model trained on the 2-week VOI service dataset performs best on the June dataset. The model has a test MSE of 14.74, which translates to the model on average predicting the NPS score of a subscriber wrong by 3.84.

Model	Logarithm	Week	Model Type	Test MSE
VOI	False	2	RF	14.74
MO_DUR	True	2	LR	14.75
MO_DUR	False	2	LR	14.75
VOI	True	2	LR	14.76
VOI	False	2	LR	14.76

Table 6.7: Top 10 regression models sorted by ascending test MSE.

Again, the VOI service appears to contain the most robust features when it comes to modelling the relationship between a subscriber's NPS score and network performance metric on unseen data. Interestingly, Table 6.7 shows that the MO_DUR feature is the only individual feature that performs well on the June dataset, suggesting that it is not network performance features that best model subscriber's NPS scores, but rather some feature identifying a subscriber's behaviour, proxied by how much they make calls in this case.

Figure 6.1 shows the predicted versus actual values for the 2-week VOI random forest model. Figure 6.1 shows how the model only predicts values in the range between 6.5 to 7.5 making it impossible to draw a linear decision boundary between promoters and detractors.

With a test MSE of 14.74 and a test RMSE of 3.84, although the VOI service random forest model is the best regression model compared to the models we looked at, it still misses the NPS score of a subscriber by 3.84 on average. Couple the lousy RMSE with the fact that the model does not utilise the full prediction range from 0 to 10, in practice, the model does not add much value for predicting the NPS score of a subscriber. The model, however, again confirms that it is the features within the VOI service that best model subscribers' NPS scores.

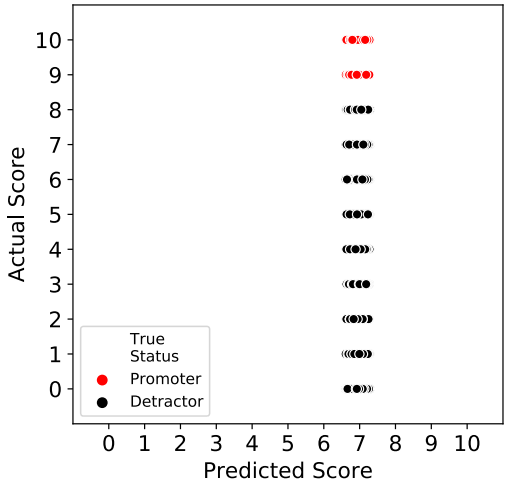


FIGURE 6.1: True versus predicted values for the random forest model on the 2-week June VOI service.

6.3.2 Classification Performance

Table 6.8 shows the top 5 classification models ranked by descending test F1-scores and Table 6.9 shows the top 5 classification models ranked by descending test AUC-score. Here test refers to the June dataset and not the hold out test set.

Model	Week	Logarithm	Model Type	Class Type	Test F1	Test Accuracy
MO_DUR	2	True	LR	clf	0.7399	0.5872
MO_DUR	2	False	LR	clf	0.7399	0.5872
MO_DUR	5	False	LR	clf	0.7367	0.5831
MO_DUR	5	True	LR	clf	0.7367	0.5831
FEB	2	False	LR	clf	0.7346	0.5828

TABLE 6.8: Top 5 classification models sorted by descending F1-score.

Model	Week	Logarithm	Model Type	Class Type	Test AUC	Test F1	Test Accuracy
MO_DUR	2	True	LR	clf	0.5218	0.7399	0.5872
MO_DUR	2	False	LR	clf	0.5218	0.7399	0.5872
MO_DUR	5	False	LR	clf	0.5205	0.7367	0.5831
MO_DUR	5	True	LR	clf	0.5205	0.7367	0.5831
VOI	5	False	RF	clf	0.5200	0.5374	0.5082

TABLE 6.9: Top 5 classification models sorted by descending test AUC-score.

Table 6.9 and Table 6.8 show that a linear regression model trained on only the MO_DUR feature performs best out of sample, based on test F1-score and test AUC. However, looking at the confusion matrix for this classifier in Table 6.10, it is clear that the model has just predicted all subscribers to be promoters, voiding the value of the top 4 MO_DUR classification models.

		Actual	
		1	0
Predicted	1	TP 7408	FP 5201
	0	FN 0	TN 0

Table 6.10: MO DUR LR 2 True confusion matrix.

As the results of the simple logistic regression model trained on the MO_DUR features are void, we select the random forest trained on the VOI service our best classification model for the June dataset. Table 6.11 shows the confusion matrix for the runner up classification model; a hyper-parameter optimised random forest trained on the 5-week VOI service. Again, all these models perform worse based on F1-score compared to the null models described in Table 5.2.

The confusion matrix shows that the VOI model does not bin all subscribers into one category, and Figure 6.2 shows that there is some predictive power in the VOI model with a test AUC of 0.52. However, in practice, the model is not great at predicting whether a subscriber is a promoter or detractor, but we have again confirmed that the VOI service best captures the relationship between a subscriber's NPS score and the available network performance metrics.

		Actual	
		1	0
Predicted	1	TP 4043	FP 3907
	0	FN 4217	TN 4414

Table 6.11: VOI FALSE 5 RF ConfusionnMatrix.

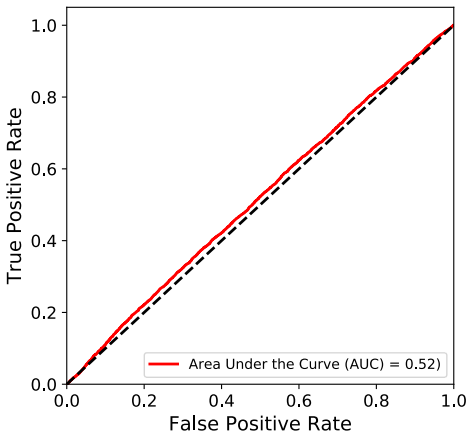


FIGURE 6.2: JUN VOI FALSE 5 RF AUC.

6.4 Recommendations

Although we were not able to construct very accurate models, in this section, we present some recommendations that we believe should improve model performance.

Evaluating 2G, 3G separately Although we aggregated 2G and 3G network performance metrics together in this thesis, evaluating the metrics within there respective technologies can provide more accurate models. The improvement is because of activity on each technology proxies for underlying subscriber behaviour, like device type, device capabilities and regional placement.

More features Most models perform better if they have more information about their observations. However, an investigation into the impact of correlated variables is required, as models like linear regression and logistic regression are highly sensitive to correlated features.

Voice metrics are the most predictive From our analysis, we see that the VOI service has the most promise to capture variance in NPS survey responses. However, the global trend in mobile device usage is focussed less on voice and more on data. There is thus a fundamental flaw in a model that relies on voice metrics, as such a model can not provide any insights into subscribers who only uses the data network and makes calls using the data network, Whatsapp calls for example.

Other evaluation metrics Although test MSE, R^2_{adj} and test AUC are standard metrics to evaluate the performance of supervised models, we suspect that when predicting NPS there are better evaluation approaches. One such approach would be to bin subscribers into deciles which would spread the contracted predicted range of 6.5 to 7.5 back to 1 to 10.

Alternative cut-off methods We looked at converting our regression predictions into binary classes based on the mean prediction, with below par performance. Another approach to consider would be to use the median of the predicted values rather than the mean, as we would get an even division between promoters and detractors using this method. Another approach would be to use the empirical proportions of promoters and detractors in the training sample, rank the predicted scores and allocate the sample proportions of these to promoters and detractors respectively.

Logging the features From our analysis applying a log transform to the features did not improve the models' performance much. Upon further investigation, we realised that most of our feature where already in the range 0 to 1 and logging values in this range does not exploit the variation as much as for feature on a much larger scale. An approach would be to log the features first, before scaling them between 0 and 1.

Including subscriber information In this thesis, we have only looked at the network performance metrics of a subscriber. Including features like device information, contract information and other subscriber dimensions should aid the accuracy.

6.5 Conclusion

From our analysis, we see that for both the regression and classification case, the random forest models perform better than the linear models, suggesting that the relationship between network performance metrics and subscriber NPS scores is non-linear.

From both the regression and classification models it was clear that features within the VOI service: *Call Drop Rate*, *Call Setup Failure Rate* and *Mobile Originating and Terminating Call Duration* generalised the best to unseen data. The robust out of sample performance suggests that the features within the VOI service best model the underlying relationship between the NPS score of a subscriber and the network performance metrics considered.

However, even the best hyper-parameter optimised VOI service random forest only had a R^2_{adj} of 0.0070, interpreted as the variance contained in all of the features within the VOI service capturing 0.7% of the total variance in NPS scores, which is not a significant portion of the variance. The low R^2_{adj} shows that there are many other factors, other than network performance metrics that influence the NPS score of a subscriber, also relying solely on VOI service features make it impossible to model subscribers who do not use any voice services.

We believe that including personal information about subscribers, for example, device type or the type of contract they have should improve the performance of the models considered. The need for more subscriber focused features eludes to the fact that some subscribers are aware of when they experience lousy network performance and are unhappy with the service received because of this lousy performance. While other subscribers do not notice bad network performance but are rather unhappy with the service, they received due to other factors like a high bill at the end of the month or lousy customer service.

We find that all the tested statistical and machine learning models, whether linear or non-linear, are poor predictors of NPS scores in a month when only the network performance metrics in the same month is available. This suggests that either NPS is driven primarily by other factors (such as customer service interactions at branches and contact centres) or are determined by historical network performance over multiple months.

A concern with our analysis is that the surveys considered had around 20 000 respondents, all of which were pre-paid subscribers. For an operator with upwards of 20 000 000 subscribers, it raises the question whether the insights gained are general for the entire subscriber base or only hold for pre-paid subscribers. Further analysis is needed to answer this question.

In summary, from all the models considered we gained much insight into how each of our 11 features is related to NPS scores. From both the linear and non-linear models, for the 11 features considered; Call Drop Rate, Call Setup Failure Rate, Server Latency and Bearer Attach Success Rate are the features most often selected as necessary in predicting the NPS scores of subscribers.

These four features were found to have the lowest p-values in the linear and logistic regression analysis and partitioned the feature space most significantly in the ensemble methods. Zaki et al. (2016) argues that the NPS metric used as a loyalty indicator does not explain the root cause or causes of low scores.

Further, the NPS measure is typically taken at the end of the customer journey, thus potentially masking the underlying issues of concern, which form the basis for identifying improvements. Our various model analyses resonate with the argument from Zaki et al. (2016), and we conclude that although our models identified a few network performance features as necessary, the models were all very poor predictors of NPS scores of subscribers.

In contrast, customer service has been shown in previous studies to be important in determining NPS scores. It may be that these are the primary factors, or that only network performance over periods significantly longer than the 5 weeks we were able to study here are significant.

Appendix A

Unlogged KPI Distributions

A.1 VOI Service

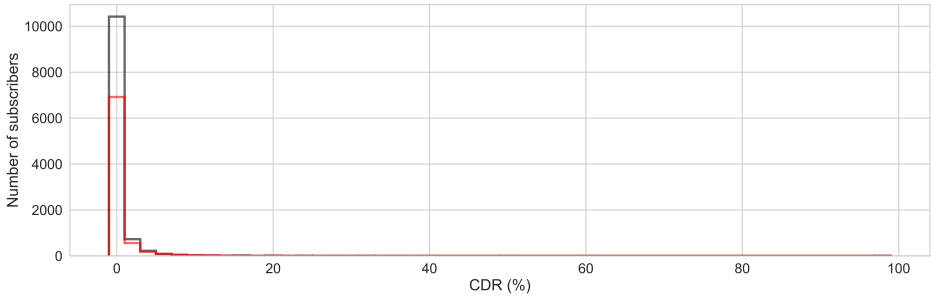


FIGURE A.1: The kernel density estimation distribution for the *CDR* feature.

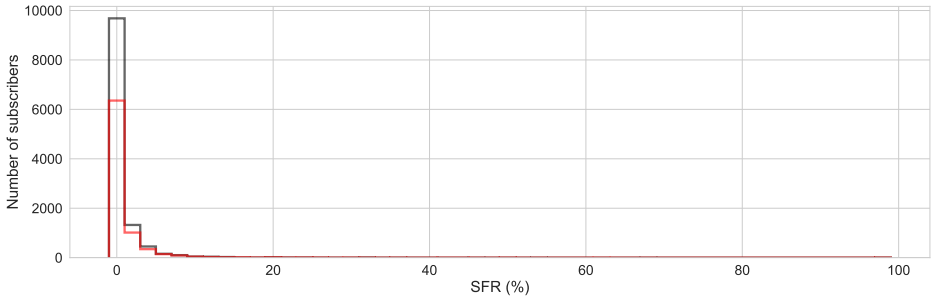


FIGURE A.2: The kernel density estimation distribution for the *SFR* feature.

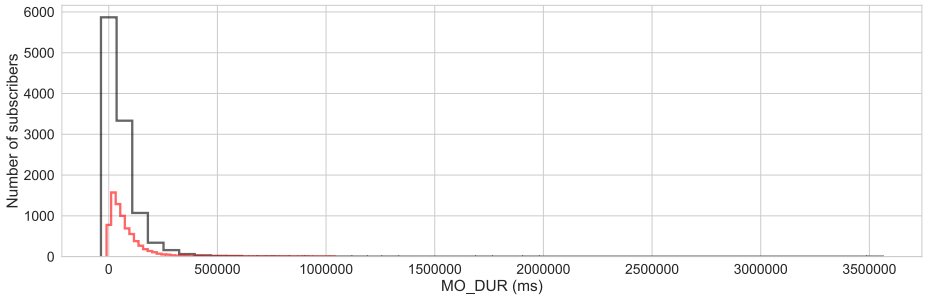


FIGURE A.3: The kernel density estimation distribution for the *MO DUR* feature.

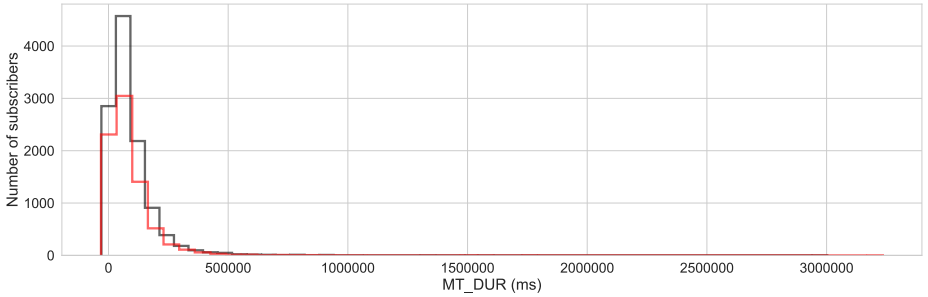


FIGURE A.4: The kernel density estimation distribution for the *MT DUR* feature.

A.2 BER Service

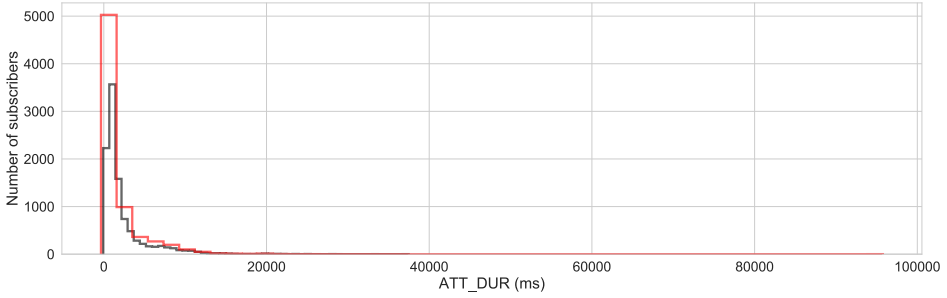


FIGURE A.5: The kernel density estimation distribution for the *ATT DUR* feature.

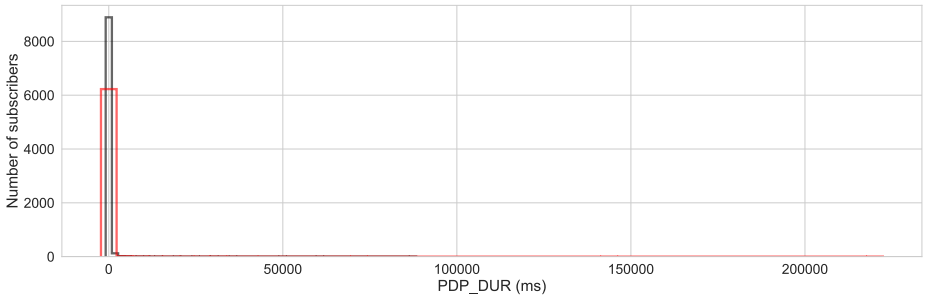


FIGURE A.6: The kernel density estimation distribution for the *ATT SR* feature.

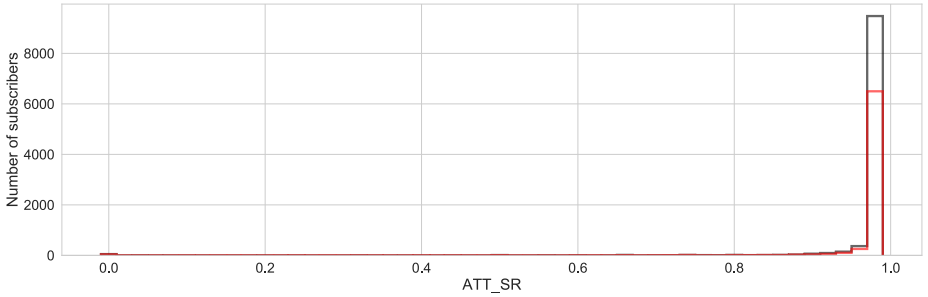


FIGURE A.7: The kernel density estimation distribution for the *PDP DUR* feature.

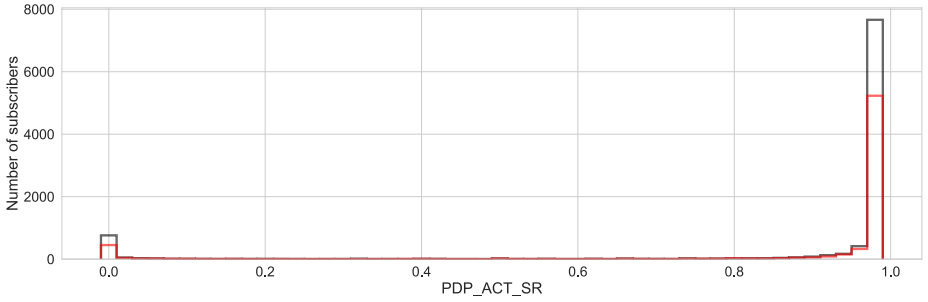


FIGURE A.8: The kernel density estimation distribution for the *PDP ACT SR* feature.

A.3 GN Service

FIGURE A.9: The kernel density estimation distribution for the *C LATENCY* feature.

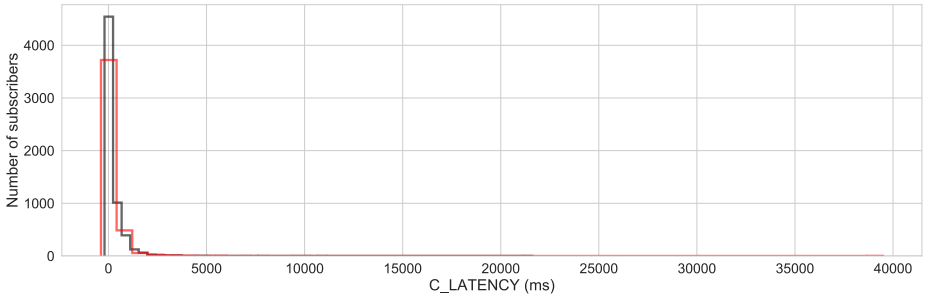


FIGURE A.10: The kernel density estimation distribution for the *S LATENCY* feature.

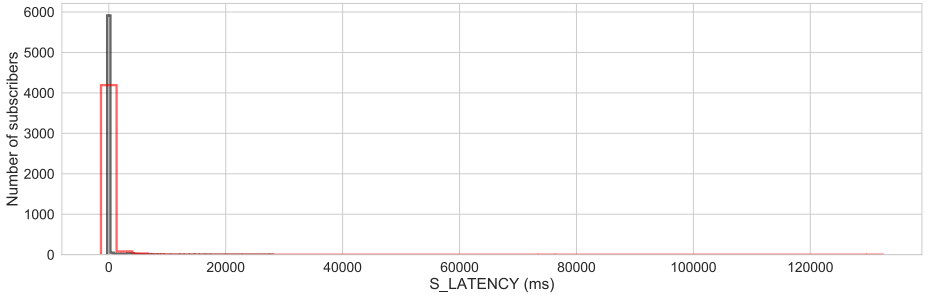


FIGURE A.11: The kernel density estimation distribution for the *PDP CRE SR* feature.

Bibliography

Boohene, R. & Agyapong, G. K. (2010), 'Analysis of the antecedents of customer loyalty of telecommunication industry in ghana: The case of vodafone (ghana)', *International Business Research* **4**(1), 229.

Breiman, L. (1996), 'Bagging predictors', *Machine learning* **24**(2), 123–140.

Chen, T. & Chen, H. (1995), 'Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems', *IEEE Transactions on Neural Networks* **6**(4), 911–917.

Draper, N. R. & Smith, H. (2014), *Applied regression analysis*, Vol. 326, John Wiley & Sons.

Everitt, B. S. (2005), *The Cambridge dictionary of statistics*, Cambridge University Press.

Fawcett, T. (2006), 'An introduction to roc analysis', *Pattern recognition letters* **27**(8), 861–874.

Hamilton, D., Lane, J. V., Gaston, P., Patton, J., Macdonald, D., Simpson, A. & Howie, C. (2014), 'Assessing treatment outcomes using a single question: the net promoter score', *Bone Joint J* .

Hosmer, D., Lemeshow, S. & Sturdivant, R. (2013), *Applied Logistic Regression*, Wiley Series in Probability and Statistics, Wiley.
URL: *https://books.google.co.za/books?id=64JYAwAAQBAJ*

Husgafvel, L. (2011), How non-financial customer based metrics are associated with company performance? an analysis of customer satisfaction, customer retention and net promoter score in telecommunications industry, G2 pro gradu, diplomity.
URL: *http://urn.fi/URN:NBN:fi:aalto-201201111221*

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'Advances in neural information processing systems', pp. 1097–1105.

Ldapwiki, L. (2018), 'Logistic function', https://ldapwiki.comth.

Levins, R. (1966), 'The strategy of model building in population biology', *American scientist* **54**(4), 421–431.

Lomax, R. G. & Hahs-Vaughn, D. L. (2013), *Statistical concepts: A second course*, Routledge.

Marcus, G. (2018), 'Deep learning: A critical appraisal', *CoRR* **abs/1801.00631**.
URL: *http://arxiv.org/abs/1801.00631*

Mazumdar, P. (2018), 'Decisiontrees'.
URL: *http://www.shogun-toolbox.org/static/notebook/current/DecisionTrees.html*

McShane, B., Gal, D., Gelman, A., Robert, C. & Tackett, J. (2018), Abandon statistical significance.
URL: *https://arxiv.org/pdf/1709.07588v2.pdf*

Ng, A. Y. (2004), Feature selection, l 1 vs. l 2 regularization, and rotational invariance, *in* 'Proceedings of the twenty-first international conference on Machine learning', ACM, p. 78.

Ojo, O. (2010), 'The relationship between service quality and customer satisfaction in the telecommunication industry: Evidence from nigeria', *BRAND. Broad Research in Accounting, Negotiation, and Distribution* **1**(1), 88–100.

Parasuraman, A., Zeithaml, V. A. & Berry, L. L. (1985), 'A conceptual model of service quality and its implications for future research', *the Journal of Marketing* pp. 41–50.

Parasuraman, A., Zeithaml, V. A. & Berry, L. L. (1988), 'Servqual: A multiple-item scale for measuring consumer perc', *Journal of retailing* **64**(1), 12.

Pidd, M. (2004), *Systems Modelling: Theory and Practice*, Wiley.
URL: *https://books.google.co.za/books?id=B4rsTSn_CEcC*

Quinlan, J. R. (1986), 'Induction of decision trees', *Machine learning* **1**(1), 81–106.

Reichheld, F. (2003), 'The one number you need to grow', *Harvard Business Review* (December Issue).
URL: *https://hbr.org/2003/12/the-one-number-you-need-to-grow*

Reichheld, F. (2006), *The Ultimate Question: Driving Good Profits and True Growth*, Harvard Business School Press. pg. 28.
URL: *https://books.google.co.za/books?id=52X3-A3U2VQC*

Rocks, B. (2016), 'Interval estimation for the net promoter score', *The American Statistician* **70**(4), 365–372.

Sauro, J. (2012), '10 things to know about net promoter scores and the user experience'.
URL: *https://measuringu.com/nps-ux/*

Six-Sigma-Material (2018), 'Normal distribution'.
URL: *http://www.six-sigma-material.com/images/NormalDistribution.JPG*

Spiess, J., T'Joens, Y., Dragnea, R., Spencer, P. & Philippart, L. (2014), 'Using big data to improve customer experience and business performance', *Bell Labs Technical Journal* **18**(4), 3–17.
URL: *http://dx.doi.org/10.1002/bltj.21642*

Strobl, C., Malley, J. & Tutz, G. (2009), 'An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests.', *Psychological methods* **14**(4), 323.

Yan, X. & Su, X. (2009), 'Linear regression analysis: theory and computing'.

Zaki, M., Kandeil, D., Neely, A. & McColl-Kennedy, J. R. (2016), 'The fallacy of the net promoter score: Customer loyalty predictive model'.