Customer retention and addressing customer churn through Predictive Analytics in telecom industry

P R A Nonis

2021

Customer retention and addressing customer churn through Predictive Analytics in telecom industry

A dissertation submitted for the Degree of Master of Computer Science

P R A Nonis
University of Colombo School of Computing
2021

DECLARATION

I hereby declare that the thesis is my original work, and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: P R A Nonis

Registration Number: 2017/MCS/057

Index Number: 17440577

2021/11/27

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr. P R A Nonis under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by,

Supervisor Name: Dr. D. A. S. Atukorale

29/11/2021

Signature of the Supervisor & Date

I would like to dedicate this thesis to my loving wife Dhanya and my most precious daughter Rayeli for being with me at all times with their immense love and support throughout this project.

Thank you for all the sacrifices you made, and it was you who made me walk through this amazing endeavor and milestone. I truly consider you both as a blessing to my life.

ACKNOWLEDGEMENTS

Firstly, I praise and thank God Almighty for all the choicest blessings showered upon during my research work and for the successful completion of the research.

I would like to express my deep and sincere gratitude to my research supervisor, **Dr. D. A. S. Atukorale**, who gave me the opportunity to do research and provided invaluable guidance throughout this research. I was inspired by his knowledge, vision, and motivation. He taught me the methodology for conducting research and presenting research work as clearly as possible. It was a great honor and a privilege to work and study under his supervision. I am very grateful for what he has provided me. I would also like to thank him for his friendliness, sympathy, and good sense of humor.

I am very grateful to my parents for the education they gave me. Even they are not alive today, I make this a moment to thank them for all the sacrifices they have made for me. I am very grateful to my wife Dhanya, and my daughter Rayeli for their continuous love, understanding, prayers, and support to complete this research work.

I express my special thanks to the Chief Information Officer of Xcom for giving me the approval to conduct this research based on their production data. It was he who made this research more realistic and valuable. Also, the insights he provided me regarding the core areas of data gathering were very helpful during the research.

Finally, I would like to thank all the people who supported me to complete the research work, directly or indirectly.

P R A Nonis

ABSTRACT

Churn has become a major issue to almost all the telecom companies. Predicting churn draws considerably a higher importance which would help retain the existing customers of the telecom organization. Cost of acquiring a new customer is always higher than retaining an already existing customer who is about to leave the company. In order to predict the potential customers who would churn, past data must be analyzed to build the relationships between the derived variables. This is quite a challenging task as this entire exercise is based on the production dataset provided by the telecom organization. It should contain some knowledge within the multi-dimensional set of data, and this will be possible only if a proper exploratory data analysis is done. The purpose of this research is to identify the potential churners in the current customer base who would leave the company in the time to come. That entire knowledge is hidden in the production dataset and by using machine learning models, we will identify the set of customers who has a higher potential to leave the company. Out of the many machine learning models in existence, Regression Logistic model, Decision Tree model and Multi-Layer Perceptron model will be used in this study and based on the descriptive metrics of evaluation such as accuracy, recall, precision and F1 score, the best model will be identified. Once the model is identified, it will be able to intake any production dataset arranged as per the specification and to predict the potential churners in a targeted proactive manner who would leave the company. Once identified, the telecom organization will have the liberty to retain those potential churners by offering various types of offers, discounts and benefits only to those targeted customers. By making this prediction as accurate as possible, it will not only retain the existing customers, but also it will save a lot of money from untargeted and mass advertising on offers and other service-related discounts.

Keywords: Churn, Attrition, Predictive Analytics, Customer Retention, Machine Learning, Prediction, Logistic Regression, Decision Tree, Neural Networks, Multi-Layer Perceptron, Churn Prediction Model, Telecom industry

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The rapid growth of the market in almost every sector is leading to subscriber base for the service providers and as a result, the competition for customer acquisition is skyrocketing by making more and more competitors and innovative business models. Therefore, in such an environment, the importance of the retention of existing customers is at utmost high importance. This makes the providers put their best efforts to prediction and prevention of churn. According to telecom market, the process of subscribers (either prepaid or post-paid) switching from one service provider is called “customer churn”.

Telecommunications industry is highly competitive. Retaining the existing customers of a telecom organization has become super important. Cost of acquiring new consumers is much greater and challenging than retaining the currently present customers [1]. As a result, the worldwide telecommunications industry is facing a serious loss of revenue and the biggest revenue leakages in telecom industry is caused due to customer churn behavior [8]. If the existing customers could be retained by identifying the possible churners effectively and precisely, this problem can be minimized to a greater level. However, that identification has become extremely challenging with the current context along with the uprising competition.

Almost all the service providers bring so many promotions frequently in order to retain the existing customers, but its effectiveness or efficacy is highly doubtful, and it will be great if those promotions can address customer churn/attrition at the right time to solidify the business by being able to retain the possible churners. In summary, this research will explore and investigate the solutions to the prediction of customer churn by using predictive analytics. The main and the most important objective of this study is to identify the potential churners and non-churners so that the churners could be addressed separately with value retention plans.

This research will cover the Broadband customers and hence, the scope of this study will be identifying the potential churners and non-churners within the boundaries of broadband and a sample of 2000 customers and their usage data will be considered from January 2020 to August 2020. In addition, this study could be expanded in future to consider the customer support related data within the above-mentioned period to understand if there were any customer support tickets and queries made within the specified period and how it has affected the attrition.

Customer relationship management (CRM) is a vital area in terms of retaining the existing customers. CRM tools hold a large set of information related to each and every customer and apart from this, customer support data is of equal importance. Analyzing these data can bring a huge amount of unforeseen knowledge and find multiple ways and means to minimize customer attrition more precisely.

Data mining techniques and methods are applied in telecommunication databases for various purposes. The usage of the telco data varies based on the purpose. The data generated by telecom industries are broadly grouped into 3 categories.

- Customer data (Demography)
- Network data
- Bill data.

In summary, this study will vastly contribute to identify the possible churners and to take necessary steps to retain them for the betterment of the telecom organization while ensuring better quality of service. Machine learning will play a major role in the computer science domain to derive the respective models which will help identify the possible churners.

1.1 Motivation

The main interest for conducting this research is to identify the customers who leave the company/service provider and thus, to minimize the cost of attracting new customers by retaining the existing ones.

1.2 Statement of the problem

Telecommunications industry is highly competitive and retaining the existing customers of a telecom organization has become super important. Cost of acquiring new consumers is much greater and challenging than retaining the currently present customers [1]. As a result, the worldwide telecommunications industry is facing a serious loss of revenue. If the existing customers could be retained by identifying effectively, this problem can be minimized to a greater level. However, that identification has become extremely challenging with the current context along with the uprising competition.

Almost all the service providers bring so many promotions frequently in order to retain the existing customers, but its effectiveness is highly doubtful, and it will be great if those promotions can address customer churn/attrition to solidify the business. In summary, this research will explore and investigate the solutions to the prediction of customer churn.

1.3 Research Aims and Objectives

This will be a comprehensive study which will dig deep into the real production data of an Australian based telecommunication company called Xcom which is situated in Sydney, Australia.

Xcom is an Australian internet service provider and a reseller which provides a range of internet related services to its customers. It has 3 types of customers which are residential, Small & medium business, and Corporate. Out of these, the highest customer base is on residential type which spans more than 100,000 customers. Broadband, hosting, and VoIP are the mainstream services provided by Xcom. All the data centers and switching centers are spread across mainland Australia. As a top reseller in the country, Xcom buys services from telco giants like NBN (National Broadband Network), Telstra, Optus and AAPT and resells to its customers at a better rate. Xcom does not deploy and maintain its own infrastructure. Instead, it operates as a wholesale customer of telecommunication providers in the category of Tier 1. For the most part, Tier 1 discount telco suppliers are the ones who give IP Transit, Inter-capital Transmission and on top of that, the different private and business grade access network items are given, coordinated and overseen by them for its clients.

The company started in 1990s as a firm which provided technical and management consultation and went on until it decided to become an internet service provider. Starting from 2004 until now, Xcom has been offering its services to the customers while improving along with the latest technology stacks. In the meantime, it started to activate its own VoIP switches and take the industrial leadership among the counterparts of the similar capacity. The Point of Presence of Xcom are circulated in many territories of Australia like New South Wales (NSW), Victoria (VIC), Queensland(QLD), South Australia(SA), Western Australia(WA), Australian Capital Territory(ACT) and Auckland.

Xcom was recorded as a rapidly developing autonomous affiliate of internet providers in numerous business magazines and postings throughout the long term and this was among the quickest developing 50 Australian innovation organizations with an additional high total development rate in a given long term period.

Below are the services provided by Xcom

- NBN
- Wireless Ethernet
- Mobile Voice Services
- Mid-Band Ethernet/Ethernet over Copper
- ADSL/ADSL2+
- Mobile Broadband Services
- Residential and business web facilitating
- POTS communication
- Home Wireless Broadband Services
- Ethernet over Fiber
- VoIP

Xcom has an extremely high reach data transmission board of more than 300 Gbps and they utilize various 10Gbit associations with many looking organizations to guarantee greatest substance accessibility and organization contiguousness. Since its initiation, XCom has executed an assortment of procedures to convey the most significant levels of on-net substance, including Google, Akamai, and Netflix stores, to guarantee the most ideal end-client experience.

At the point when XCom went live, they have made a strange stride of furnishing clients with significant "free" information to deal with their data transmission all the more viably. This effectively urges clients to perform enormous quantities of downloads during what is currently alluded to as the "uncounted" or "off-top" period. The time and remittances during this period have likewise changed since the strategy was first carried out in February 2004. As of July 21, 2009, the off-top period has been reached out from 12 PM to early afternoon at AEST, with a month-to-month recompense of 60GB. This period and its recompenses are accessible to all private ADSL and ADSL2 + clients, aside from packaged ADSL or Zero Quarter ADSL2 + plan clients.

There is a characterized limit for uncounted/off-top periods, however Xcom doesn't effectively keep clients from downloading past that breaking point. Beforehand, clients who surpassed the cutoff at whatever month were set in discrete transfer speed pools for the remainder of the month to forestall such activities, however on 1st of February 2008. By then, we began applying overage charges for downloads that surpassed the off-top cutoff.

Broadband, NBN, and versatile now have new terms that incorporate different information limits, including free/limitless information alternatives.

Xcom has two kinds of SMB clients which are known as VISP (Virtual Internet Service Provider) and the other sort is Reseller. Xcom offers its administrations to VISP clients where they can exchange them to their own clients. Affiliates also do the equivalent however the meaning of VISP is that they are ready to exchange benefits as well as, they can oversee them. As of now both VISP and Resellers have separate online entries which assists them with overseeing and incorporate the administrations that will give the best to their clients.

The other main customer type is Corporate. The Corporate customers bring along a higher income in Xcom telco business and at the moment they have nearly 5000 active corporates. Master accounts are provided to easily manage the multiple accounts with Xcom and at the moment there are around 3000 master accounts.

Despite how good the company is performing, currently, Xcom is facing a customer churn/attrition problem and that could more descriptively be shown through below graph which shows the count of deactivations is larger than the count of service activations for a given month.

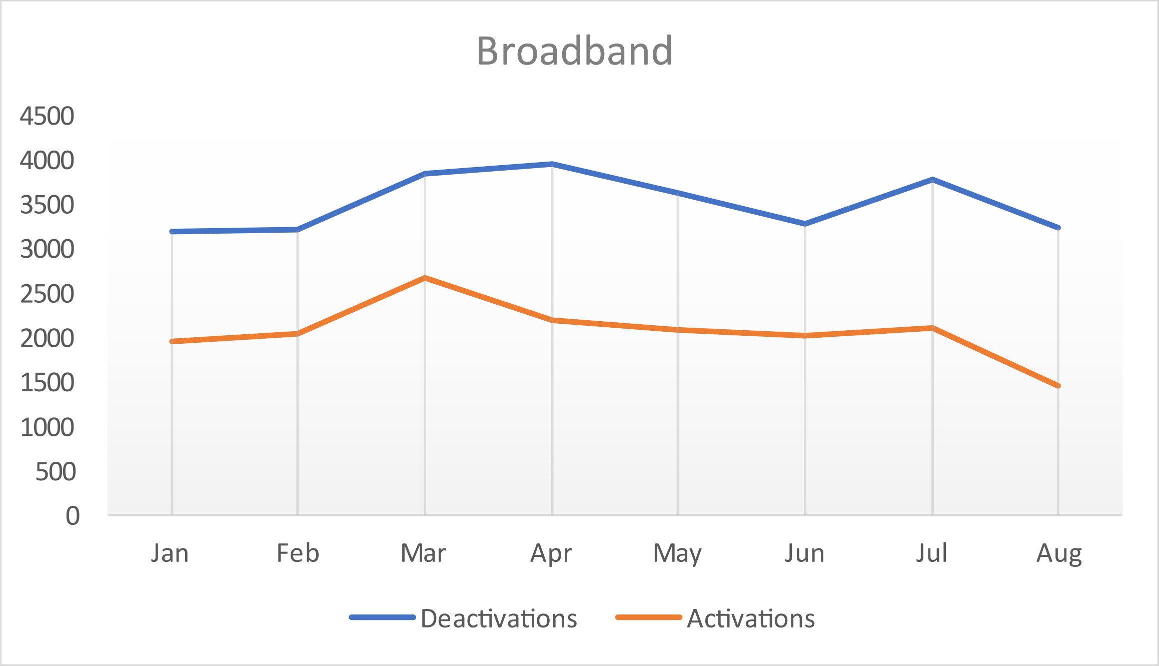


Figure 1. Service activations and deactivations

Reason behind this is not certain unless a proper investigation and a data analysis is done. However, the huge competition between the counterparts is playing a major role here. Even then, if the company is performing well and keeping its customers happy, there cannot be any reason for the customers to churn or leave the company by discontinuing the contract. If a proper analysis can be done as to identify these customers beforehand, it would greatly benefit the company to retain at a bigger scale.

1.3.1 Aim

Xcom is mainly focused on having higher profits by selling its products and services and the main target or the aim of this research is merely to solidify the business by having a strong customer base which has a higher growth rate. Existing customers must be retained well on top of new customers joining into the company. Hence, the aim is to safeguard the tenure of the existing customers with Xcom who will maintain a long service record with the company.

1.3.2 Objectives

The main and the most important objective of this study is to identify the potential churners and non-churners so that the churners could be addressed separately with value retention plans. Once the potential churners are proactively identified, they will be addressed by granting various types of offers that would drastically decrease the churn rate. As a result, the existing customers could be retained at a greater level which would keep the tenure to be continued with the company.

In order to make this objective successful, this study must produce a solid model which will predict the potential churners with a higher accuracy and precision. By using Machine Learning models and Data Analytics, the most appropriate model will be identified.

1.4 Scope

There are multiple product categories in Xcom which will eventually have many potential churners. Hence the scope of this study will be within the boundaries of Residential Broadband by using Machine Learning and Predictive Analytics.

This being said, if this research goes well along with the production data, the next steps would be to expand the model derived in this research to the other product types. However, the scope of this research will be only limited to Broadband customers of Xcom who are in the residential category.

1.5 Structure of the Thesis

In terms of the structure, this thesis is comprised of an in-depth literature review which talks about what churn is and the possible reasons for the customers to churn. There will be many previous studies which were referred when doing this research within the same telco domain as well as some other industrial and commercial domains. Since churn is not specific to a certain geography or customer type, the studies done for many other countries have been centralized when reviewing literature for this study.

There will be multiple models used in this study and out of them, the model with the highest accuracy will be selected as the churn prediction model. Below are the selected Machine Learning models for this study:

- Logistic Regression
- Decision Tree
- Neural Networks – MLP (Multi Layer Perceptron)

Logistic Regression model will be ideal as the target variable, which is churn, will be a binary output. The respective model will have to identify if the customer is a churner or not. If the customer was found a potential churner, the customer retention plans and bonuses could be applied on that particular customer and as a result, that same customer would create a high probability of not moving to a competitor for a better value.

Linear regression will provide a continuous output while Logistic Regression will provide a discrete output and could be used to predict if the customer is a potential churner or not. Also, the logistic regression is estimated via Maximum Likelihood Estimation.

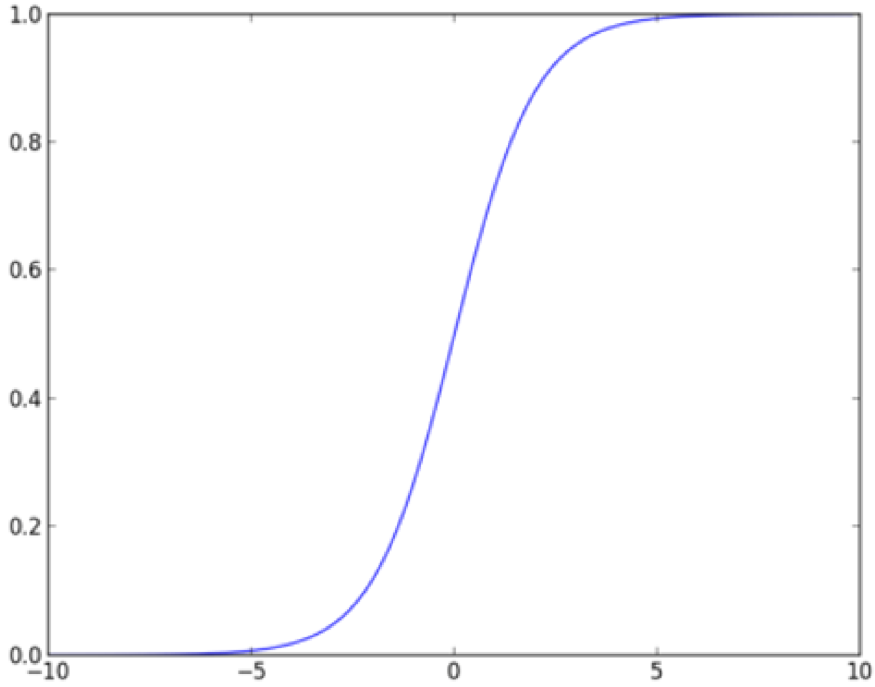


Figure 2. Logistic Regression

Decision trees are comparatively convenient to visualize as they have a tree like model of decisions. Also, when it comes to optimizing the decision tree model, it will gain a higher accuracy when predicting the potential churners.

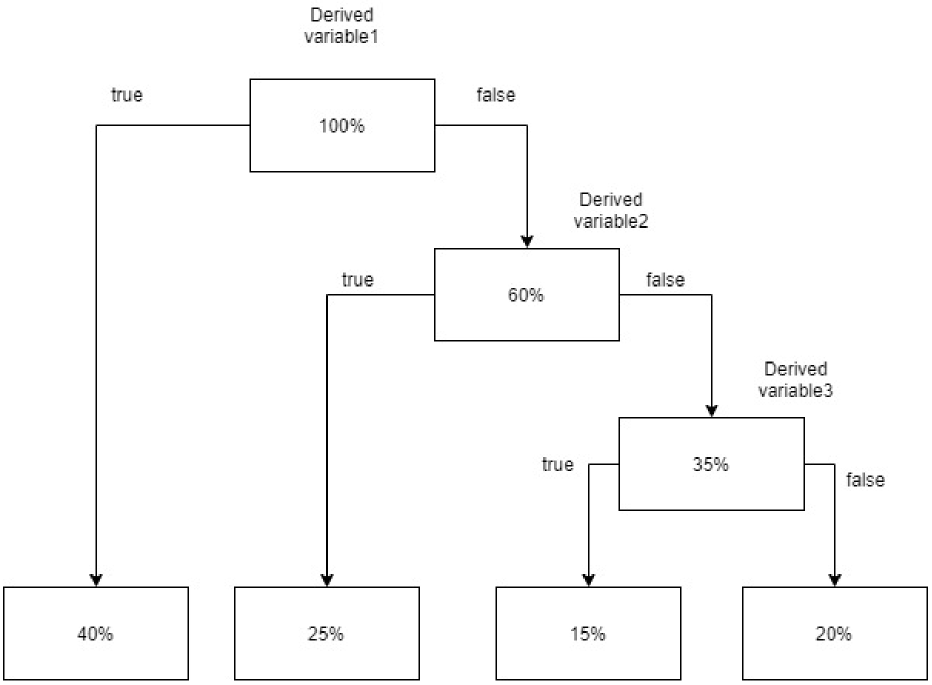


Figure 3. Decision Tree

Finally, this study will investigate the Neural Networks by using the Multi-Layer Perceptron model where MPL is the classical type of this kind. As this is multi-layer, there will be multiple layers of neurons where the input layer gets the data feed and there will be one or more levels of hidden layers and the predicting is done at the output layer. Technically, this output layer can be called as the visible layer. MLPs can be well used for regression prediction problems where in this case, it will predict the churning of a customer where churn is the target variable with a set of data inputs.

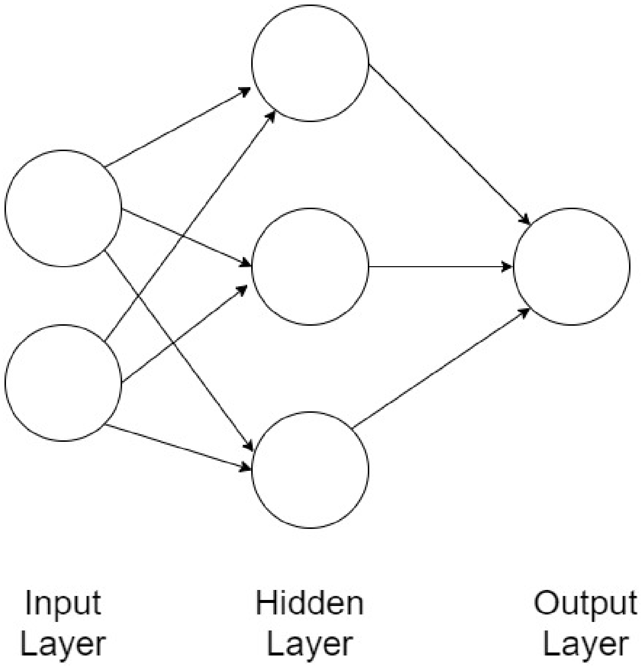


Figure 4. Multi-Layer Perceptron

Once the above-mentioned models are derived, evaluation will be done by using the confusion matrix and the usual descriptive metrics.

Table 1.Confusion Matrix

	Actual-Churner	**Non-Churner**
Predicted-Churner	TP	FP
Predicted-Non-churner	FN	TN

Also, the Confusion Matrix will do the visual representation of the below metrics:

- True Positive (TP) – Predicted churner is an actual churner
- True Negative (TN) – Predicted Non-churner is an actual non-churner
- False Positive (FP) – Predicted churner is an actual non-churner
- False Negative (FN) – Predicted non-churner is an actual churner

Below are the most commonly used descriptive metrics used for model evaluation by using the confusion matrix:

- **Accuracy** = (TP+TN) / total

  Out of all the classes (positive and negative), how many of them we have predicted correctly.

- **Precision** = TP / (TP+FP)

  Out of all the classes we have predicted as positive, how many are actually positive.

- **Recall** = TP / (TP+FN)

  Out of all the positive classes, how many were predicted correctly.

- **F1-Score** = (2* Recall* Precision)/(Recall+ Precision)

  F1 Score helps to measure Recall and Precision at the same time.

On top of this, as an alternative to the confusion matrix, we will be using the Receiver Operator Characteristic Curve (ROC) and Area Under Curve (AUC). ROC is a visual chart/graph which will delineate/illustrate the genuine positive rate against the bogus/false positive pace of our classifier.

The AUC will give a particular numeric measurement to analyze rather than a visual portrayal/representation. An AUC = 1 would address an ideal classifier, and an AUC = 0.5 addresses a classifier which just has half accuracy. This measurement evaluates the general exactness of the classifier model.

Lastly, this study talks about the conclusion and future work of how this can be improved by using the **XG-boost** which is an interpretation-focused method.

CHAPTER 2

LITERATURE REVIEW

2.1 A Literature Review

Customer churn could be of several types [7]:

- Involuntary churn – Termination of service due to payment failure or dishonest/fraudulent usage.
- Unavoidable churn – When the customer dies/move forever from the market.
- Voluntary churn – Leaving from one operator to the other as a result of better value.

Practically it is very difficult to differentiate between unavoidable and voluntary churn hence this study will consider them both as one [7]. Also, it is estimated that there is a 2.2% average churn rate per month for mobile telecommunications and about 27% of a given carrier's subscribers are dropped every year [1].

Churn prediction is a global issue where almost all the telecom organizations are facing worldwide. Hence there are so many studies based on this research area. However, the tools, techniques and methodologies used will vary and using Predictive Analytics, KDD (Knowledge Discovery in Databases) [3] will add more value when it comes to deriving the models for potential churner identification.

In business terms, churn means the termination or discontinuation of an ongoing contract and there are three types of churn [2]:

- Active/Deliberate – Customer decides to leave his contract and switch to another provider.
- Rotational/Incidental – Customer leaves the contract without the aim of switching to a competitor.
- Passive/Non-voluntary – Company discontinues/suspends the contract itself.

When addressing customer churn, there are two basic approaches which are Targeted and Untargeted. Both these approaches are based on contacting the customer at different points in their service period. An untargeted approach relies on the entire customer base by mass advertising to increase brand loyalty with the expectation of retaining customers. But a targeted approach relies on identifying the customers who has a higher probability to churn. These customers could be given promotions and incentives through a customized service plan with the expectation of retention. Even in targeted approach, it further divides into two sub-divisions which are Reactive and Proactive. In a targeted reactive approach, the telecom company waits until the customer contacts them for a cancellation of contract and at this point, the customer is offered with promotions and incentives or even discounts on the due amounts/bills. But in a targeted proactive approach, the telecom company tries to identify/predict the customers who are likely to churn in a future date and offers them a set of promotions and incentives to retain them in the business. Hence this study is mainly focused on the Targeted Proactive approach in addressing churn. However, with this approach the churn prediction process needs to be high in accuracy otherwise it will be an utter waste to give promotions and offers to the customers who would anyway retain with the company [1].

Grouping is one of the most widely recognized exploratory information investigation or data analysis strategies/techniques used to get an instinct with regards to the design of the information and there are some renowned bunching calculations which will be profoundly valuable while deciding the ideal attrition/churn prediction model. K-means is an iterative clustering algorithm that tries to partition the dataset into K number of pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. DBSCAN (Density-Based Spatial Clustering of Application with Noise) is another clustering algorithm for very large data sets, and it will be used in this study in order to cluster the data [4]. In addition, this study could be expanded in future which would consider using the FCM (Fuzzy c-means) clustering method [5] in comparison to DBSCAN and, some other clustering methods could be considered to identify the most ideal one for clustering data in this research. When it comes to building the churn prediction model, below models will be considered in this study in order to compare and identify the model with the highest accuracy [3]:

- Logistic regression model
- Decision Tree model
- Artificial Neural Networks – Multi-Layer Perceptron (MLP) model

Data mining will dig data and find the trends and patterns to get the undiscovered knowledge where we could use to apply to identify the potential churners [6]. Sometimes we might have to use derived variables based on the original variables to identify some underlying customer behavior. Derived variables are new variables based on the original variables [17]. When it comes to most effective derived variable, those are the ones which represent something in real world, and which can be used to depict the actual customer behavior in a better way. However, the original variables are derived variables already but those derived variables from the original variables are better than the original variable. This is due to the ability to explain customer behavior. Below are some examples of derived variables:

- The average amount of data consumed in last 8 months.
- The ratio of download and upload
- Average payment amount of last 8 months
- Average number of plan changes in last 8 months
- Customer tenure in months

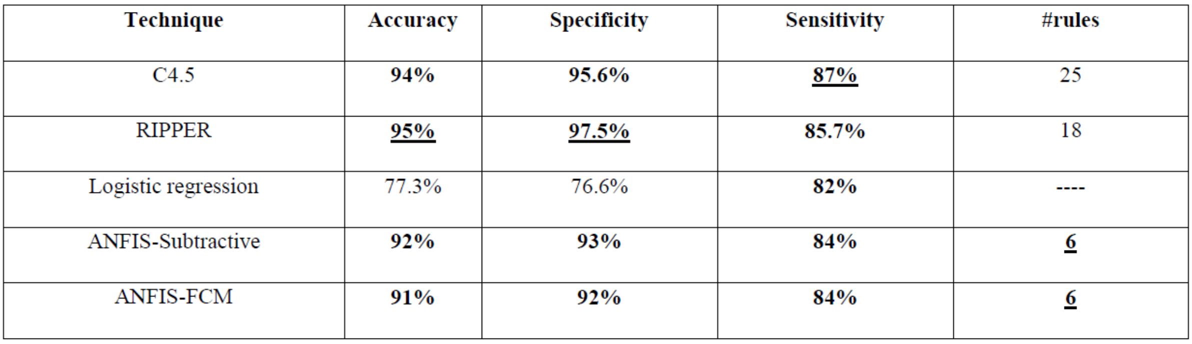
Churn is not specific only to telecommunications industry; it could be in any domain which has customer bases. Banking and insurance is one example where churn/attrition takes place competitively. Over the past two decades this has become adversely high along with the introduction of new technological enhancements to the banking and insurance sector. Eventually, with the introduction of fintech, most banking and insurance organizations were involved in a much higher competition, and this has resulted in customers moving across counterparts by creating a competition over the organizations. As a result, the banking and insurance companies are stimulated to have customer relationship management programs implemented and it was found that a bank can increase its revenues by 85% just by enhancing the retention rate in 5% of the total consumer base [5].

2.1.1 Comprehensibility

When we are dealing with a churn prediction model, the most common descriptive metrics are accuracy, precision, recall and F1 score. Something that needs to be kept in mind is that they are not the only important aspects when it comes to evaluation. Most of all, the model being developed for churn prediction has to be comprehensible and accurate. The term comprehensibility of a model causes it to express the knowledge on churn drivers of broadband consumers. Such knowledge can be extracted in the form of conditional expressions when developing an effective and precise churn prediction model. Hence, as a matter of fact, to consider the comprehensibility is of great importance [5]. In simple terms, the term comprehensibility is all about how well we can pull out the understanding of the churn drivers.

Hoissen and Mostafa [5] has tried building churn prediction models using C4.5, RIPPER, Logistic Regression, ANFIS-Subtractive, ANFIS-FCM models and they mention that only RIPPER, C4.5, ANFIS-Subtractive, ANFIS-FCM models generate comprehensible rules out of a dataset. Furthermore, the Logistic Regression model doesn't seem to be supporting the rule based or conditional representation of knowledge. Based on their findings, below table has been derived to show the comparison between descriptive metrics and the comprehensibility rules.

Table 2. Algorithm Performance [5]



Technique	Accuracy	Specificity	Sensitivity	#rules
C4.5	**94%**	**95.6%**	**87%**	25
RIPPER	**95%**	**97.5%**	**85.7%**	18
Logistic regression	77.3%	76.6%	**82%**	----
ANFIS-Subtractive	**92%**	**93%**	**84%**	**6**
ANFIS-FCM	**91%**	**92%**	**84%**	**6**

2.1.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an utmost important phase in performing a data analytics study. As Prasad Patil [17] explains in his data science web article, "it refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations."

Every study which focuses on building a churn prediction model, has an extended EDA. Despite of the target variable which is churn, the other derived numerical and categorical features are explored in the EDA, and it helps to demystify the hidden knowledge to a greater extent by using the above-mentioned summary statistics and graphical representations. In the study of Telecom churn prediction done by Umayaparvathi and Iyakutti [3], they have selected derived variables grouped together under 4 categories as mentioned below:

- Client Demography
- Bill and Payment
- Call Detail Record
- Client Care Service

Under the customer demography they have identified that the age of customers between 45 and 48 have more potential to churn. Customers between the age of 25 and 30 are more prone to churn. Corporate account holders are high in churn rate and that is identified as the customer class. Final variable under the customer demography is the days to contract expiry. That is mainly due to most of the customers who join the company in order to obtain a new device would eventually leave the network when soon after the contract is expired.

Bill and Payment is the second category out of their 4 derived categories. Under this category, average bill amount is the first derived variable which comes to a point that if the client's average bill total for the previous 2 quarters is equal to a certain amount, the churn potential is comparatively high. Secondly, they have derived the average pay amount variable which is if the customer’s average monthly payment for the past 2 quarters is less than $100 or if it is between $520 and $550, there is a higher chance of churn. The last derived variable under this category is the overdue payment count which talks about if the count is between 0 and 4, there is a much higher chance to churn.

The third main category is the Call Detail Record. Under this category, the first derived variable is the average minimum outbound call minutes. If the average number of minutes is less than 168, there is a higher chance of churning. Second derived variable under this category is the total past delink. If the total count is greater than 3, these customers will eventually churn. Thirdly, total distinct international calls count makes the derived variable. If the total count is greater than 6, the churn potential is higher as usual.

Under the fourth category, which is Customer Care Service, the subscription plan change flag makes the first derived variable. If the customer has a higher frequency of plan change, there is a higher tendency for that customer to churn. The second derived variable is the ID change flag. If the customer changes the account information, they are more towards churning. Next derived variable is the blacklist count. If the count of being blacklisted is great than 2, those type of customers have a higher potential in churning. Telephone number change flag is the next derived variable and if this count is greater than 2, it shows that the customer is not much happy with the company and has a higher probability to churn. Last derived variable under this category is the payment method change. If the customer has changed the payment for more than 3 times, it shows that the customer is about the leave the company.

2.1.3 Predictive Modeling Process and Experiment Architecture Subsection

In every organization under the telecommunication industry, managing data is at critical importance. Without handling the proper data storage and querying techniques, it would be extremely difficult to manage and maintain the transactions on a daily basis. Hence, almost all the telecom companies use the cutting-edge, high-performance databases in order cope up with the on-demand transactions and service reliability.

When it comes to exploring and experimenting data, there will be certain number of steps which must be followed in order to derive the meaningful and realistic variables for the predictive analytics. Once the numerical and categorical variables are derived on top of the target variable, the machine learning models will get trained and generate the intended results. Yang and Chiu [18] in their study of knowledge discovery of churn predictions, describes the steps required to perform the experiment architecture as a section in predictive modelling process.

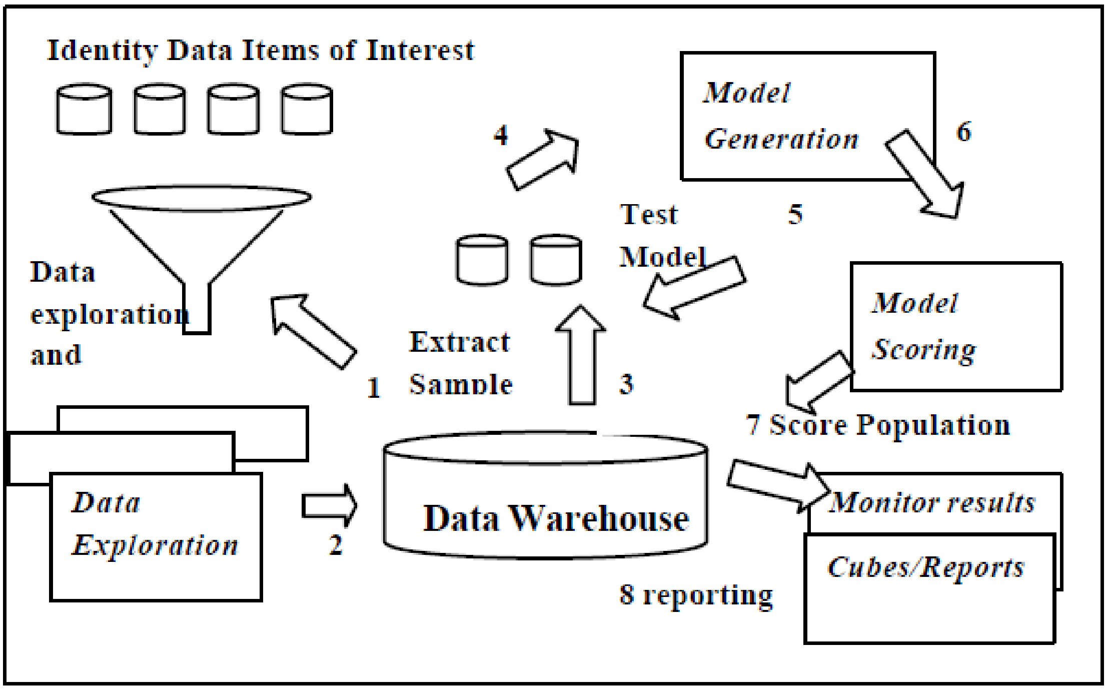


Figure 5. Experiment Architecture [18]

Below are the steps shown in the image above:

- Data extracted from the warehouse for exploration and to identify the data items of interest.
- Data extraction and exploration programming scripts.
- Further analysis of data.
- Recursive interaction with the data warehouse.
- Automate and move or transfer the discover rules of modelling results using SQL scripts.
- Generate the customer list of higher churn probability.

On top of the above, when building the model, they have used below success criteria in order to assess if the model meets the expected levels of acceptance:

- Model Performance
- Model Interpretability
- Model efficiency
- Model Maintainability

2.2 Presentation of Scientific Material

Below are the types of churn:

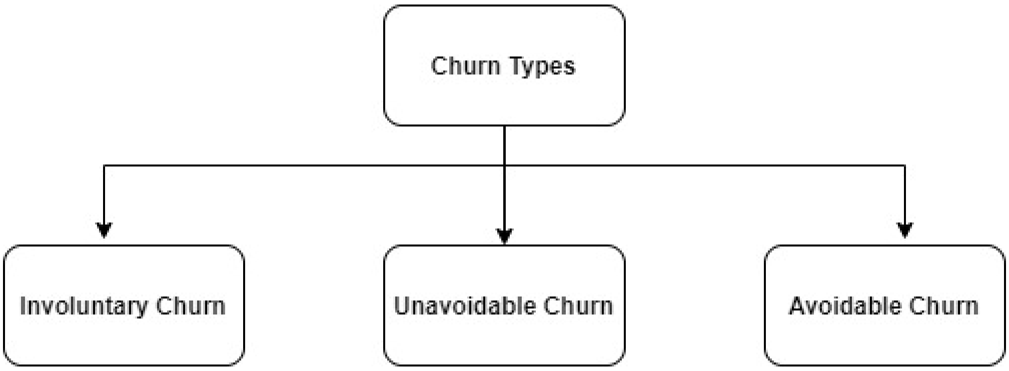


Figure 6: Churn Types

Below are the ways of addressing churn:

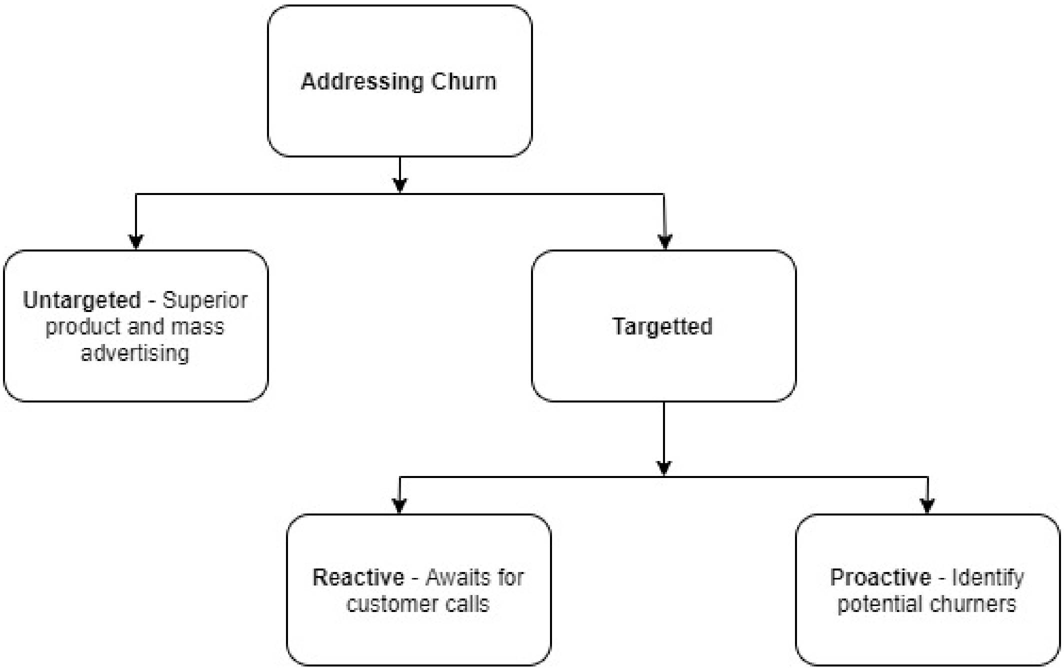

Addressing Churn
Untargeted - Superior product and mass advertising
Targetted
Reactive - Awaits for customer calls
Proactive - Identify potential churners

Figure 7: Ways of addressing churn.

CHAPTER 3

METHODOLOGY

The entire study is based on the historic data mining which will help to build up the models required to identify who the potential churners are going to be within the current active customers. Therefore KDD (Knowledge Discovery in Databases) along with Predictive Analytics will help identify the models required and those models will be applied on the existing customer base to identify who the potential churners are going to be [3].

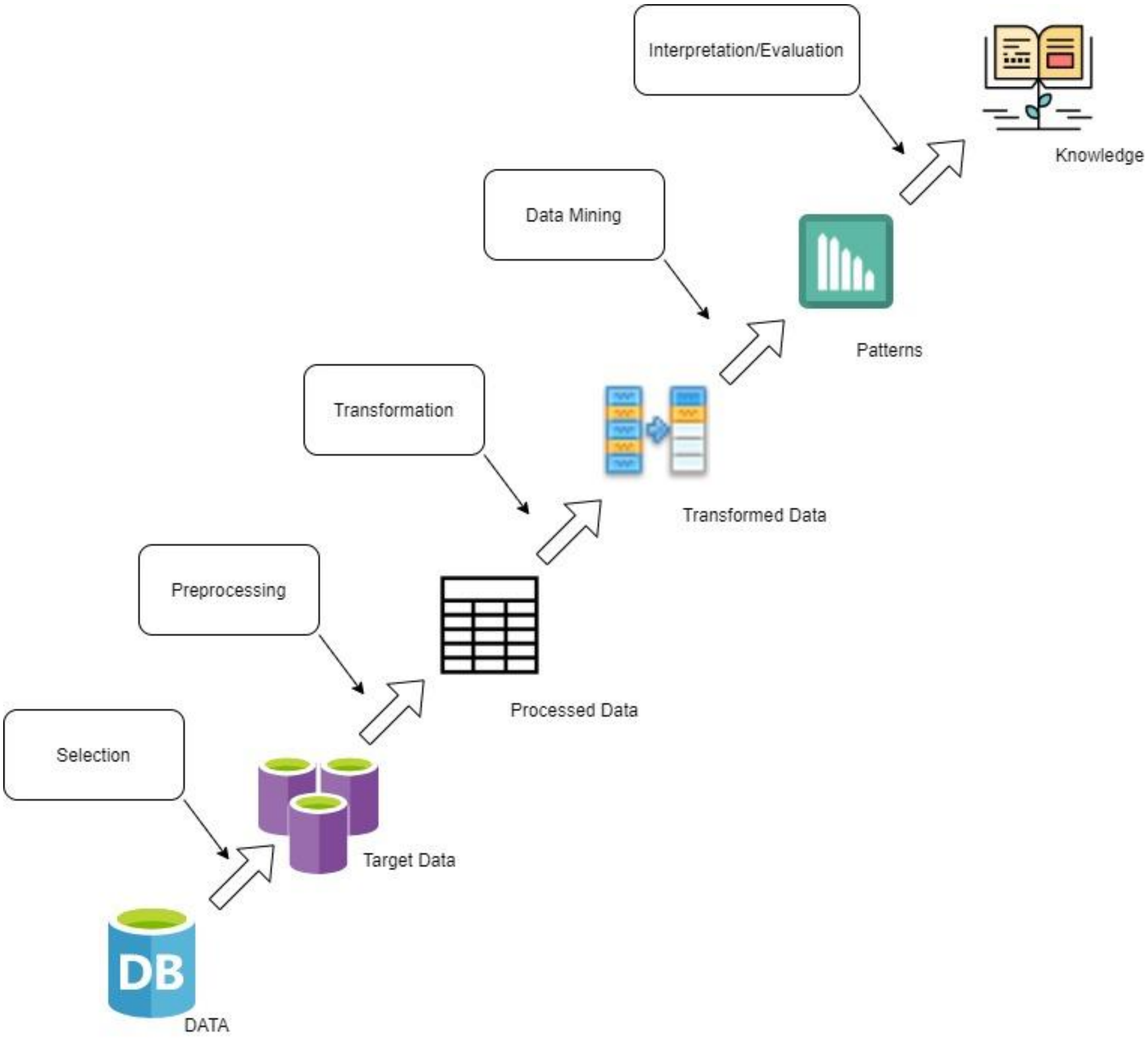


Figure 8: Knowledge Discovery Process

Basically, the study will comprise of below steps:

- Data acquisition
- Data preparation
- Derived variables
- Variable extraction
- Model Construction

As mentioned in the previous chapters, this study will generate below models:

- Logistic regression model
- Decision tree model
- Neural networks – Multi-Layer Perceptron

In this process, data acquisition and data preparation take an imperative significance as it will be the basis for all the models and decisions to be made throughout the study. Data will be aggregated for the past 8 months, and the customer behavior of that data will be used to predict the churners during the 9th month and so on.

As linear regression models are ideal to predict the continuous variables, the logistic regression models are much appropriate for binary or discrete attributes. By using logistic regression, the approximate probability of churn is estimated by the logit function.

3.1 Data acquisition

Obtaining data for this type of a research is undoubtedly challenging as the real production data can only be found from real service providers. Data related to this study was received from the Xcom organization and as mentioned above, 8 months of production data was given after masking the customer details. Usage data for nearly 2500 customer were received along with their active status, joined date and service cancellation dates (if present). All the data received are from the production databases and are 100% accurate. Churn prediction models depend on the past usage or behavior data for a specific period of time.

Below is one of the SQL queries used to retrieve the production datasets:

```
SELECT  dsl_service.id DSL_service_id, IF(active_status='I',1,0) churn,
active_status, date_format(dsl_service.creation_date, "%Y-%m-%d") creation_date,
dsl_service.deactivation_date, plan_type_id, plan_type.telstra_line_speed,
plan_type.sale_price, service_contract_period_id, service_contract_period.in_month
tenure, reseller_id, is_relocation, provider_status_type.product_type,
dsl_service.peak_period_type_id, reseller_id_xcom,
dsl_service_order_type.description,
payment_type.description, customer_type.description
 from dsl_service
LEFT JOIN customer ON dsl_service.customer_id=customer.id
LEFT JOIN service_contract_period ON
dsl_service.service_contract_period_id=service_contract_period.id
LEFT JOIN provider_status_type on
provider_status_type.provider_status=dsl_service.provider_status
LEFT JOIN plan_type ON plan_type.id=dsl_service.plan_type_id
LEFT JOIN dsl_service_order_type ON
dsl_service_order_type.id=dsl_service.dsl_service_order_type_id
LEFT JOIN payment_type ON payment_type.id=customer.payment_type_id
LEFT JOIN customer_type ON customer_type.id=customer.customer_type_id
where dsl_service.id in (selected service ID list from 2020-01-01 to 2020-08-31)
```

By using the above query, the main dataset was extracted from the production database. It contains all the raw data elements which needs to be prepared in the following steps of this study.

3.2 Data preparation

This is where Exploratory Data Analysis (EDA) comes into action. It helps to analyze the production dataset and to derive the variables. The raw dataset is retrieved from Xcom in order to understand the churn likelihood to a greater depth. However, before building any classification model, it is vital that we perform an Exploratory Data Analysis to get a better interpretation of our data. Target is to identify if the customer would churn or not. In simple terms, Yes or No. Hence, we will transform the churn into a binary outcome. If the customer is a churner, value would be 1 else 0.

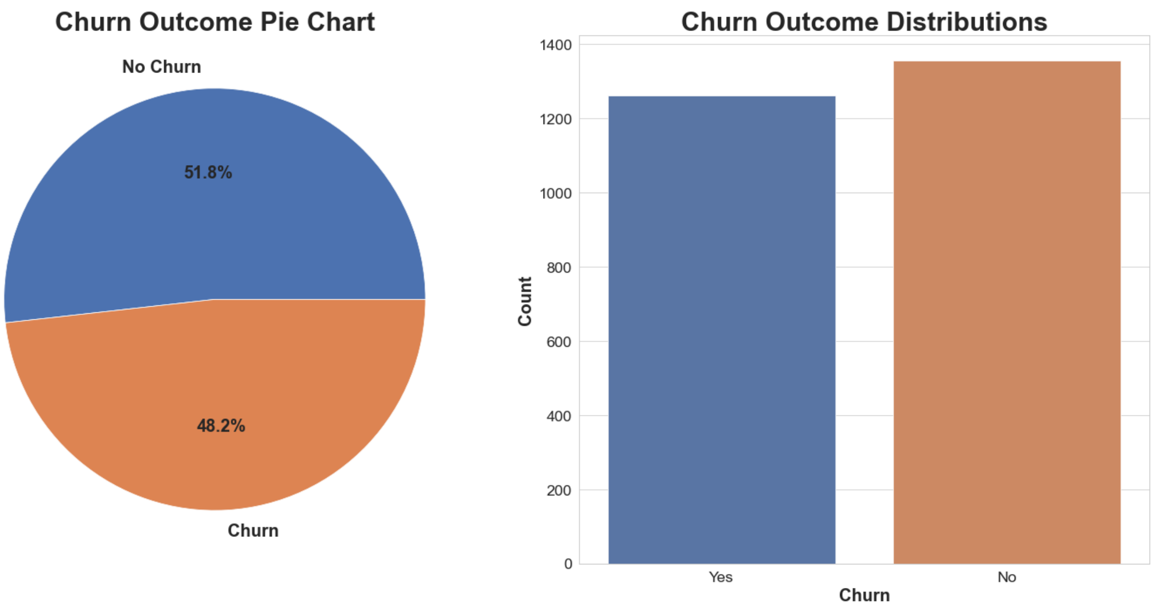
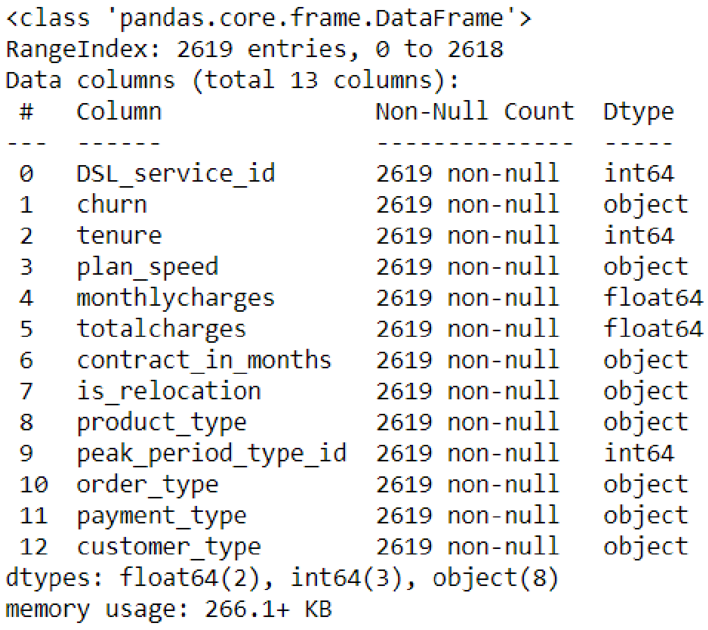


Figure 9. Churn vs non-Churn

3.3 Derived variables

This was done using the Jupyter Notebook powered by Python. There are 2619 data entries in total and below is the set of derived variables:

Table 3. Derived Variables



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2619 entries, 0 to 2618
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   DSL_service_id       2619 non-null   int64  
 1   churn                2619 non-null   object 
 2   tenure               2619 non-null   int64  
 3   plan_speed           2619 non-null   object 
 4   monthlycharges       2619 non-null   float64
 5   totalcharges         2619 non-null   float64
 6   contract_in_months   2619 non-null   object 
 7   is_relocation        2619 non-null   object 
 8   product_type         2619 non-null   object 
 9   peak_period_type_id  2619 non-null   int64  
 10  order_type           2619 non-null   object 
 11  payment_type         2619 non-null   object 
 12  customer_type        2619 non-null   object 
dtypes: float64(2), int64(3), object(8)
memory usage: 266.1+ KB
```

3.4 Variable extraction

The target variable is churn and we will see the correlation between churn and the numeric and categorical features. Numeric data will help us understand the distribution of data. Below are the identified numeric features which we can compare against the target value, churn.

- Sale Price (Monthly)
- Tenure

Below are the categorical features which we can compare against the target value, churn:

- Product Type
- Plan Speed
- Service Contract Period
- Relocation
- Peak Period Type
- Order Type
- Payment Type
- Customer Type

We will use the Kernel Density Estimation (KDE) plot which is a non-parametric way to assess the probability density function of a random variable which may have an impact on the target variable. Below is the KDE Plots for Tenure and Monthly charges (Sale price):

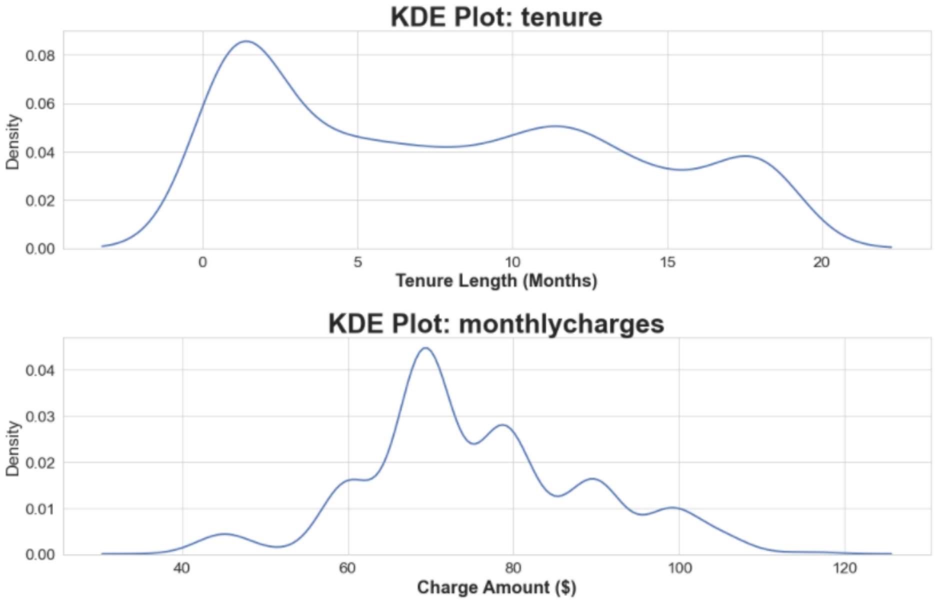


Figure 10. KDE Plots

Tenure is a derived numeric variable which means the amount of time which the customer has been with the company. Below is the comparison between the target variable churn and the numeric variable tenure:

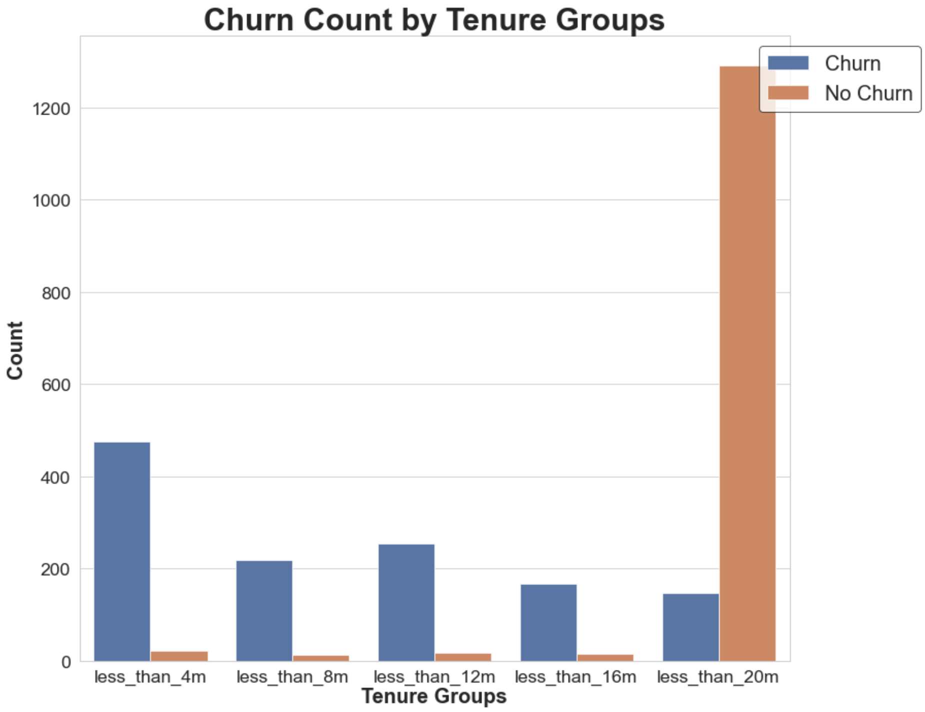


Figure 11. Tenure Groups

Tenure is grouped into 4-month groups and the maximum churn count can be found in 1 month to 4 months tenure where customers were with the company for less than 4 months.

Below is the comparison between the target variable churn and the derived numeric variable Sale Price:

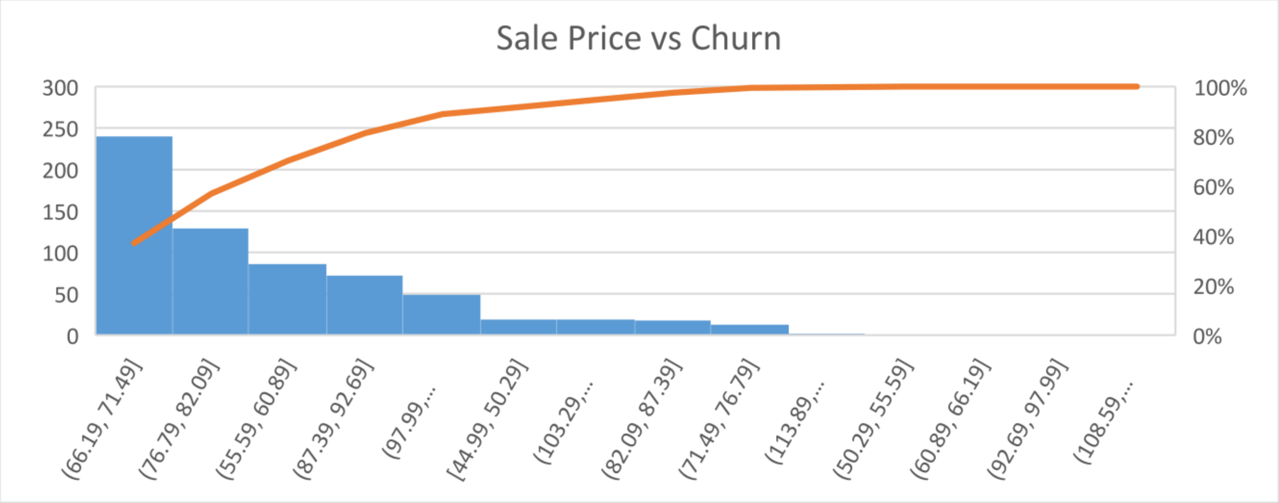


Figure 12. Sale Price vs Churn

As per the above graph, the maximum churn rate is from the monthly sale price of $ 66 to $71. Below are the data visualization comparisons between the target variable churn vs the categorical features.

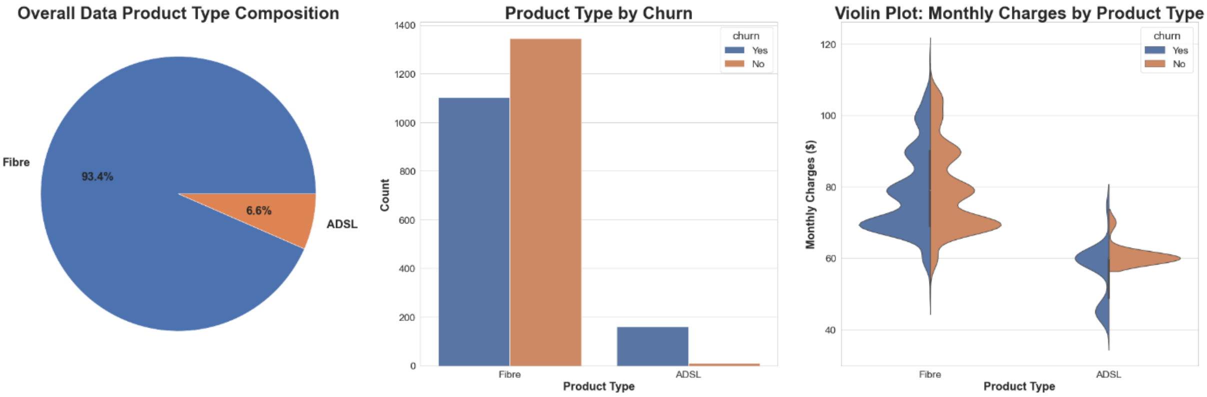


Figure 13. Churn vs Product Type

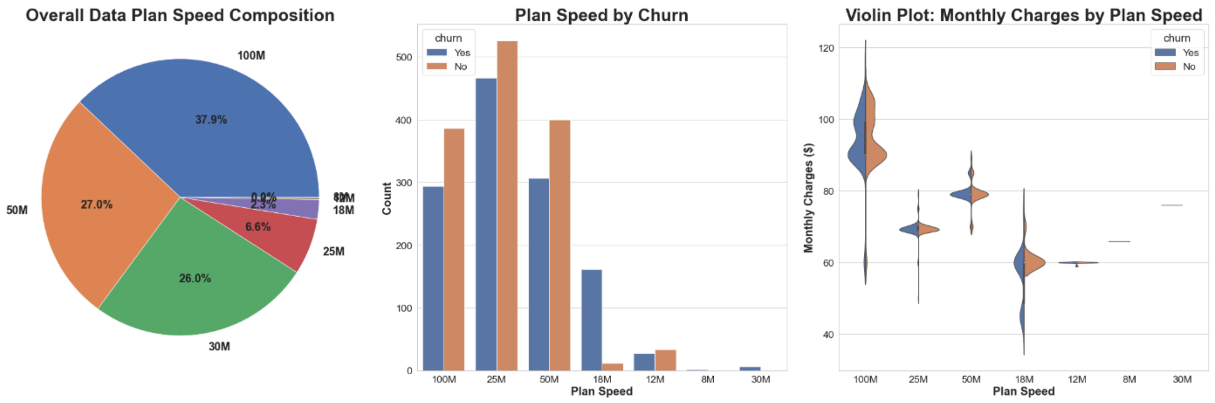


Figure 14. Churn vs Plan Speed

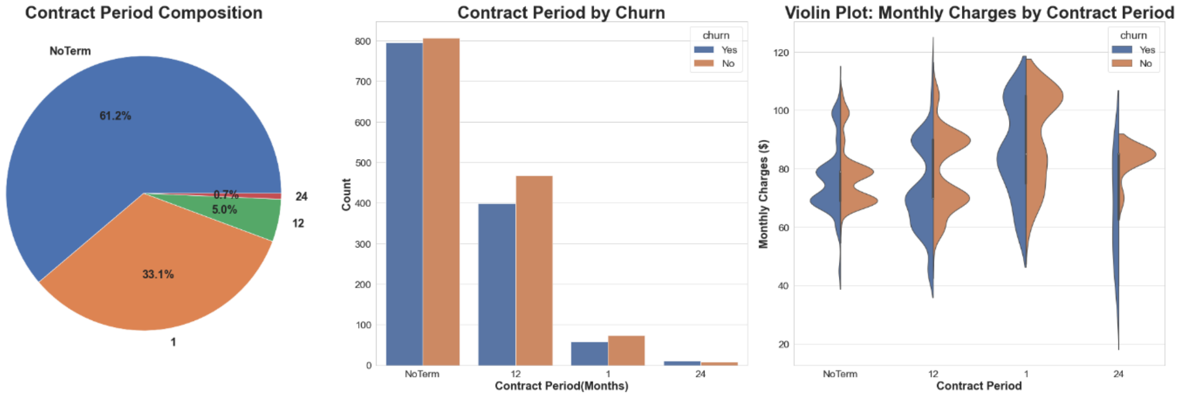


Figure 15. Churn vs Contract Period

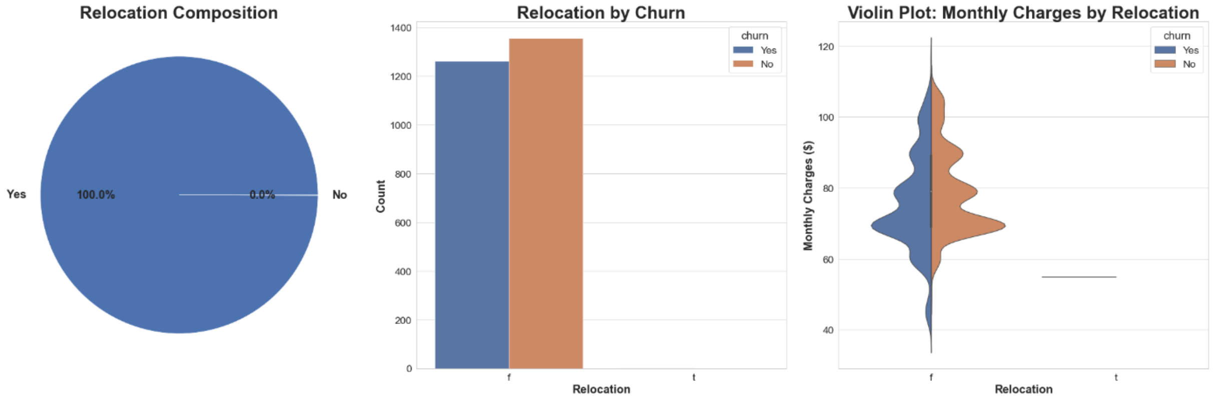


Figure 16. Churn vs Relocation

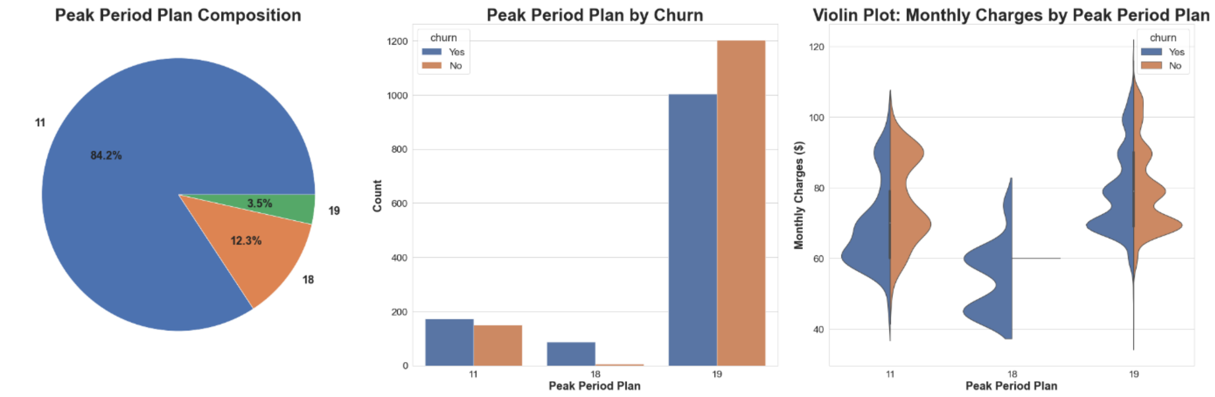


Figure 17. Churn vs Peak Period Plan

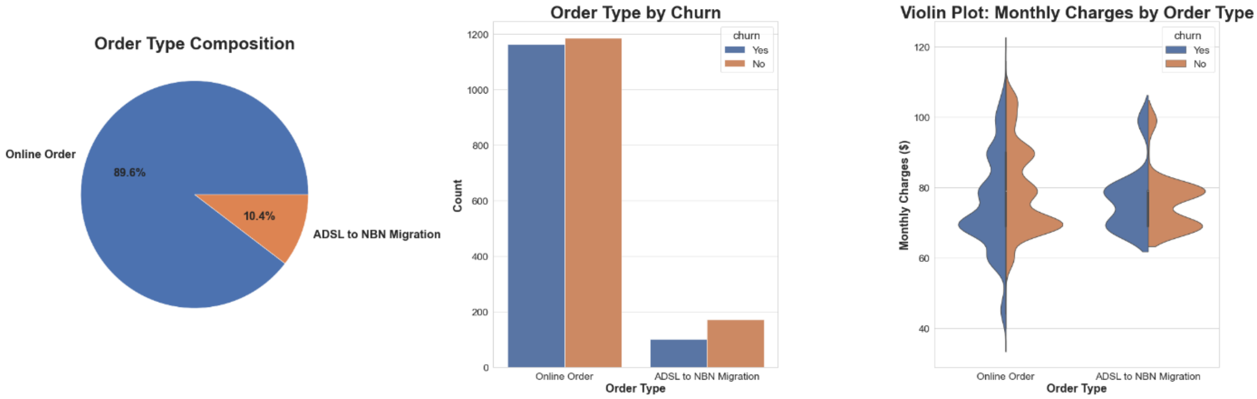


Figure 18. Churn vs Order Type

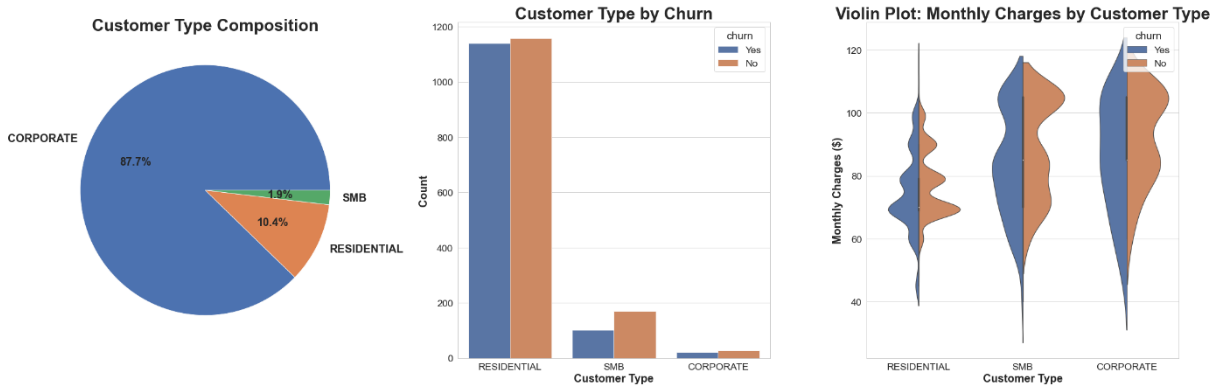


Figure 19. Churn vs Customer Type

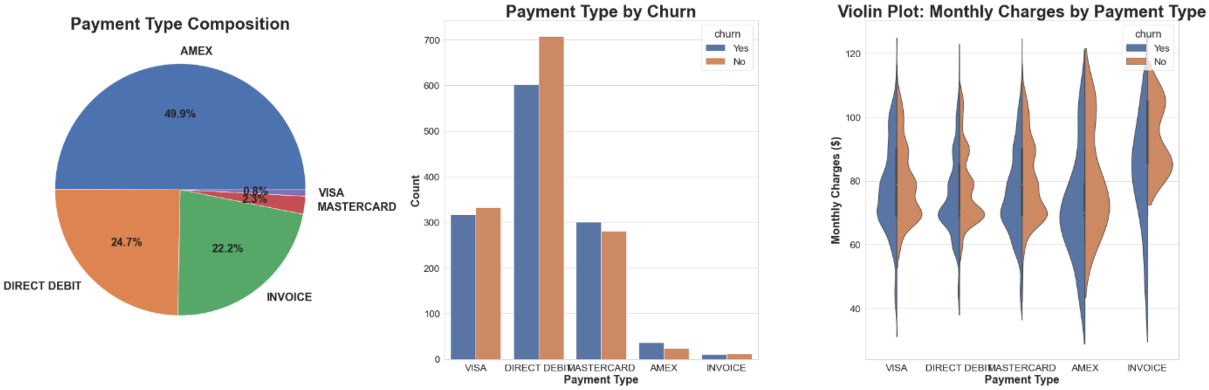


Figure 20. Churn vs Payment Type

3.5 Model construction

3.5.1 Logistic Regression Model

The logistic regression model will help us understand the correlation between churn and the various other derived variables and it will make its way to guess the probability of belonging to one selected group or another group. In Logistic Regression, we will have the predicted outcome value between 0 and 1. Here, our target feature is churn and the model will identify the relationships between churn and the other remaining features. By applying probabilistic calculations, we will be able to identify if the customer is a possible churner or not.

As mentioned earlier, the logistic regression model is a non-linear transformation of a linear regression model. The standard representation of the logistic regression is known as the logit function [2] and the estimated probability of churn is given by the below formula:

$$Pr[churn] = \frac{1}{1 + e^{-T}}$$

Above formula is based on the Sigmoid activation function which exists between 0 and 1 and it is an ideal choice when predicting the probability as an output.

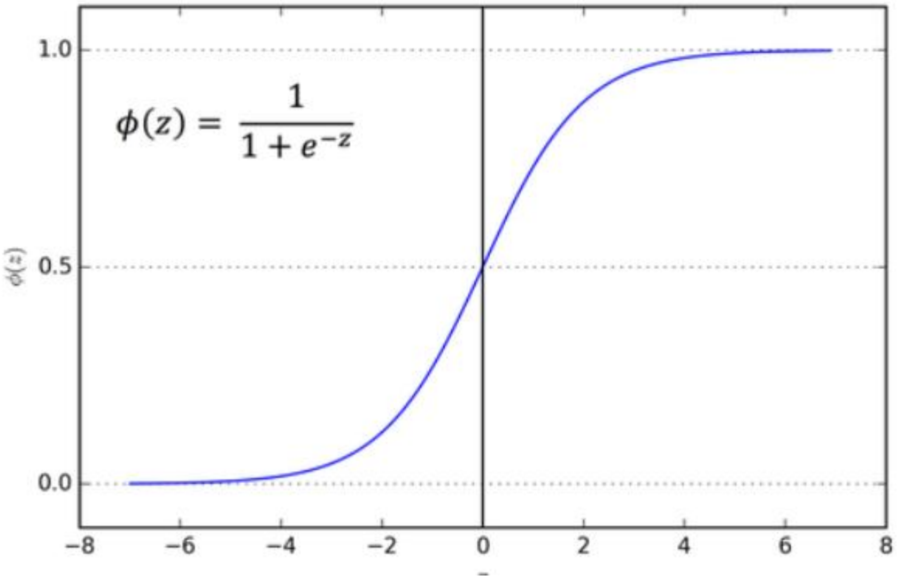


Figure 21. Sigmoid Function [10]

In the above formula, T= a + BX and **‘a’** is a constant term. **‘X’** represents the predictor attributes vector and **‘B’** is the coefficient vector for the predictor attributes. If ‘**T**’ equals 0 the probability is 0.5 and this means the probability of a customer being a churner and a non-churner is equal to each other which is 0.5 and as the T increases, the probability becomes close to 1 so that the customer becomes more probable towards becoming a churner. When T is becoming small, the probability of churn is moving towards 0. [2]

Also, Logistic regression is not only one of the most widely used classification algorithms, but also it is a discriminative probabilistic model as it models the posterior probability distribution P(Y|X), where Y is the target variable and X is the set of features. The logistic regression model will return the probability distribution over Y when X is given. In the binary classification problem, the output of the sigmoid function is interpreted as a probability of a particular sample belonging to a positive class [9].

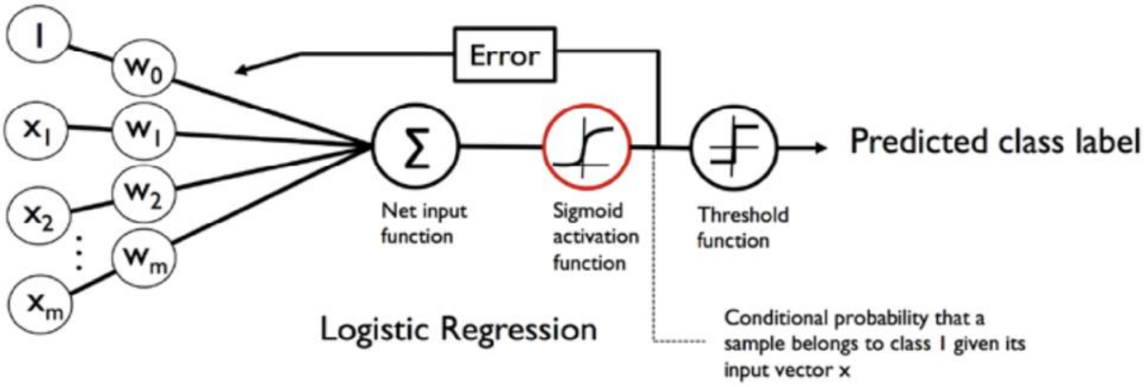


Figure 22. Logistic Regression Model Architecture [9]

By using Python and its inbuilt libraries, we can perform the logistic regression model by using the production data. When using the production data, the separation between the training and testing are 75% and 25% respectively.

In Python, **liblinear** solver method is used and a Logistic Regression model will be instantiated without an intercept. C is set to a large number.

LogisticRegression package is used in sklearn Python library, and the model is set to fit our x and y training sets as shown in below code snippet:

```
logreg = LogisticRegression(fit_intercept = False, C = 1e12, solver = 'liblinear’)

logreg.fit(X_train, y_train)
```

3.5.2 Decision Tree Model

Decision trees are considered as one of the most widely used Machine Learning models when predicting classifying the future events. Developing a decision tree has two main stages where the first is building and the second is pruning. In the building stage, the dataset is partitioned recursively until most of the records in each partition contain an identical value and in the second phase the branches with noisy data will be removed [2].

When building a decision tree, each node in it is a testing condition and the branching happens based on the value of the attribute being tested. Every decision tree represents a collection of multiple rule sets and when evaluating the production data, the classification is done by traversing through the tree until the leaf node is reached where the label of this leaf node is assigned to the data record being tested [2].

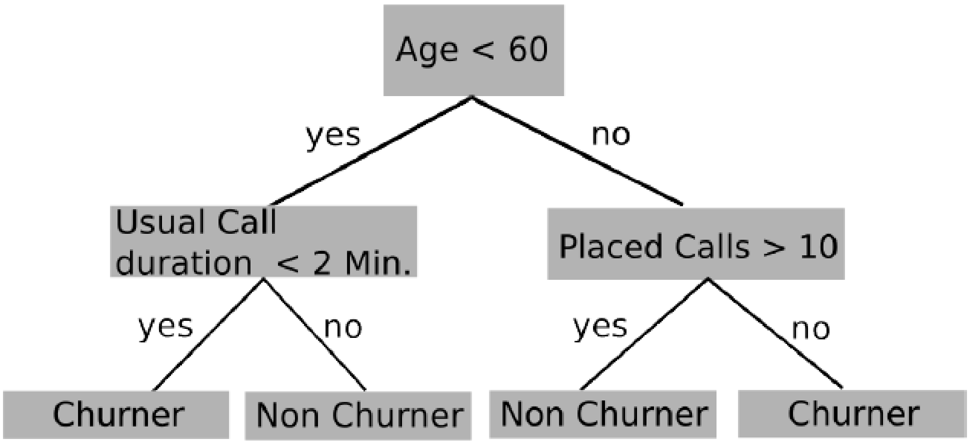


Figure 23. Simplified churn prediction decision tree [2]

Most of the time, tree-based predictive model has high accuracy, stability, and ease of interpretation and as Renato [9] describes in his research, decision trees are highly attractive models if the concern is about interpretability where the information split is performed by the Information Gain (IG). Further, he describes that the goal of IG is to split the nodes at the most informative features to the samples at each node where all belong to the same class. Below is the formula for computing IG [9]:

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

f = feature to perform the split

I = Impurity measure

D_p = data set of the parent node

D_{left} = left child node

D_{right} = right child node

N_p = total number of samples at the parent node

N_{left} = number of samples at the left node

N_{right} = number of samples at the right node

Model building is done with Jupyter notebook by using the already available Python libraries. DecisionTreeClassifier package is used in sklearn Python library, and the model is set to fit our x and y training sets as shown in below code snippet:

```
clf1 = DecisionTreeClassifier(random_state = 33)
clf1.fit(X_train, y_train)
test_predictions1 = clf1.predict(X_test)
```

3.5.3 Neural Networks – MLP using Python

Artificial Neural Networks (ANN) are based on the biological neural network of the human brain. ANNs are adaptive, fault tolerant and can learn by example. As in human brain, the unit of composition in an ANN is the neuron. Hence, an ANN is composed of as a set of connected neurons/nodes which are organized into layers [10].

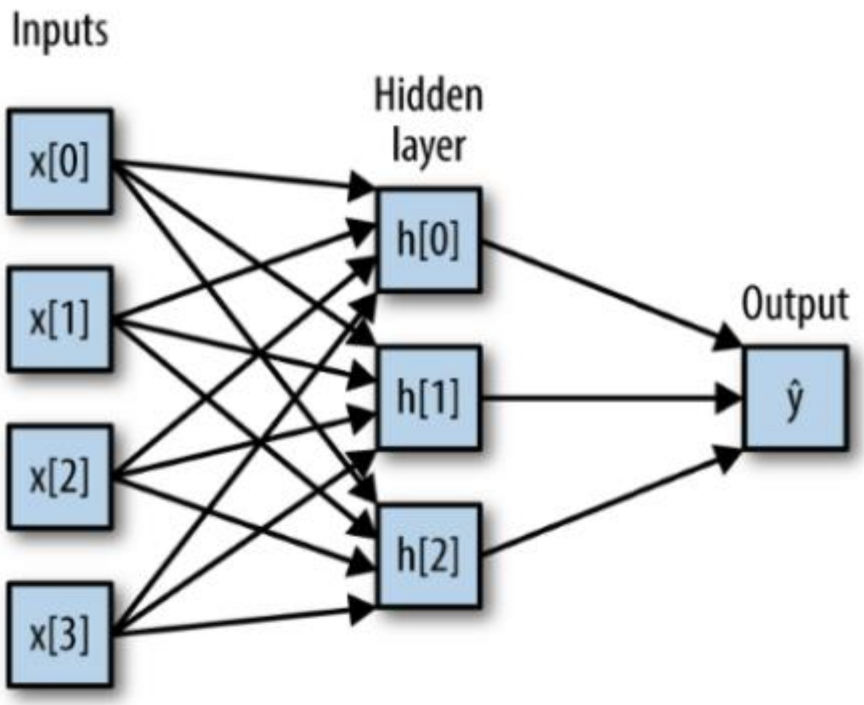


Figure 24. Artificial Neural Network [12]

As the above image shows, the input layer communicates with one or more hidden layers where the final hidden layer communicates in turn with the output layer. The connection between the layers is by weighted links and this is analogous to the synapsis in human neurons. These links are responsible in carrying the signals between the neurons and in ANNs the signals are sent in the form of real numbers. Each neuron will have an output and it will be a function of the weighted sum. In the learning phase, the weights on the connection are adjusted to represent the connections between the links. Artificial Neural Networks are ideal for churn prediction problems and in this study, the Multi-Layer Perceptron will be used where it has at least three layers [10].

The MLP built for this study comprises of 16 inputs which are feature columns in the production dataset and there are 8 output units in the hidden layer, and there is one output unit in the output layer which is the probability of customer churn.

In order to build the model, the in-built Python libraries will be used. Sequential and Dense packages are used in keras Python library, and the model is set to fit our x and y training sets as shown in this code snippet:

```
#import Sequential and Dense packages from Keras library
from keras.models import Sequential
from keras.layers import Dense
# define target variable and features
target = 'churn'
features = [x for x in list(df_trans.columns) if x != target]
model = Sequential()
model.add(Dense(16, input_dim=len(features), activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid’))
# compile the above model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy’])
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df_trans[features],
                                                    df_trans[target],
                                                    test_size=0.2,
                                       random_state=23)
```

The prototypes for the above models will be demonstrated by using Jupyter Notebook where a separate program has been developed to demonstrate all three models used in this research.

CHAPTER 4

EVALUATION AND RESULTS

This research is about customer retention and addressing customer churn through Predictive Analytics in telecom industry. Therefore, the target variable will be the churn outcome. To be more specific, the churn outcome will be yes or no. In predictive analytics we forecast what will happen in future. Hence, this research will predict the customers who may churn and move to the other competitors in future by making a potential impact to Xcom.

Since the real production data is used throughout the research and out of that a training data set (75% of full dataset) will be selected to train the models and the remaining 25% will be for test data. By using below descriptive metrics, the model with the highest accuracy will be selected as the churn prediction model:

- **Accuracy** = (TP+TN) / total
  Out of all the classes (positive and negative), how many of them we have predicted correctly.

- **Precision** = TP / (TP+FP)
  Out of all the classes we have predicted as positive, how many are actually positive.

- **Recall** = TP / (TP+FN)
  Out of all the positive classes, how many were predicted correctly.

- **F1 Score** = (2 * Recall * Precision) / (Recall + Precision)
  F1 Score helps to measure Recall and Precision at the same time.

[TP = True Positive, TN = True Negative, FP=False Positive, FN = False Negative]

Accuracy of the test data set will verify the validity of the model, and this could be used as an evaluation to validate the derived model. Once this is done, any new user data could be applied to the churn prediction model and identify/predict the potential churners.

Above parameters will be put into a confusion matrix and evaluated.

- True Positive (TP) – Predicted churner is an actual churner
- True Negative (TN) – Predicted Non-churner is an actual non-churner
- False Positive (FP) – Predicted churner is an actual non-churner
- False Negative (FN) – Predicted non-churner is an actual churner

Table 4: Confusion Matrix

	Actual-Churner	**Non-Churner**
Predicted-Churner	TP	FP
Predicted-Non-churner	FN	TN

Below are some of the hypotheses which will be covered in this research:

- Customers with a lower tenure are most likely to churn.
- Customers with monthly contract plans are most likely to churn.
- Customers who pay a higher monthly charge are highly likely to churn.

With the research outcome, there could be many other hypotheses which we could test in.

Initially an in-depth data preparation is done along with an exploratory data analysis (EDA). When it comes to churn prediction, identifying the data and the derived variables is quite important. Hence, based on the datasets, we must identify the relationship between the target and the other features. In our case, the target is churn which is the dependent variable as well as a Boolean outcome. Next step is to identify the numeric and categorical features.

When analyzing the numeric variables such as tenure, monthly charge and total charge, the distribution of data will be quite important. Hence, a Kernel Density Estimation (KDE) will be used which will visualize the probability distribution. KDE can be used as a non-parametric way to approximate/estimate the probability density function of the random variables of our model.

4.1 Logistic Regression Model Evaluation

Table 5. Results analysis using a Confusion Matrix.

	Actual Churner	Non Churner
Predicted Churner	316	13
Predicted Non churner	56	270

- Accuracy = 0.89
- Precision = 0.95
- Recall = 0.83
- F1 Score = 0.89

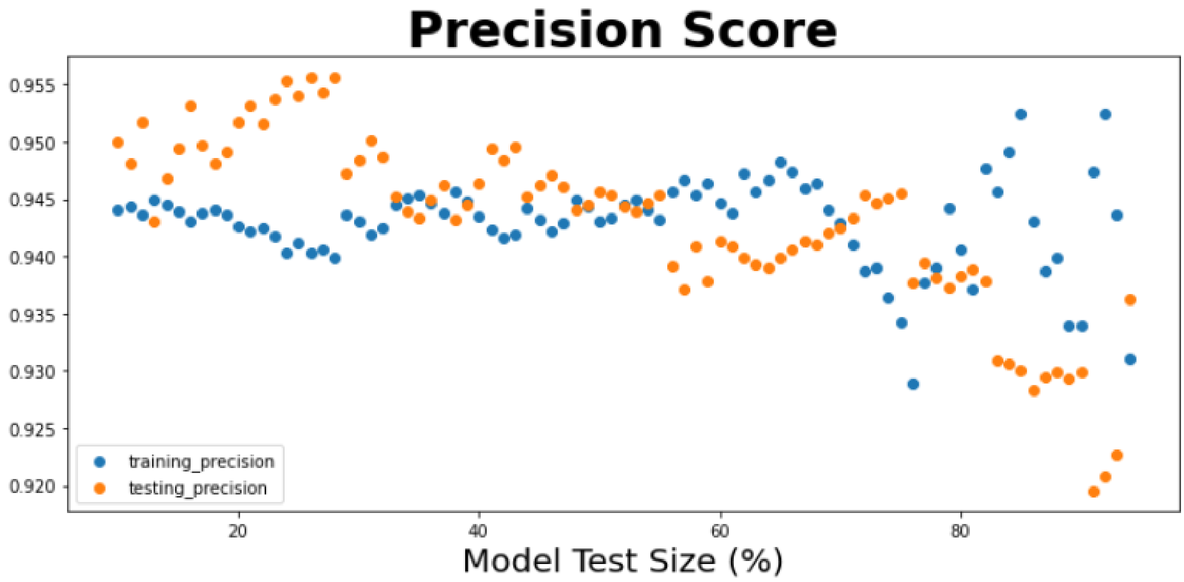


Figure 25. Logistic Regression – Precision Score

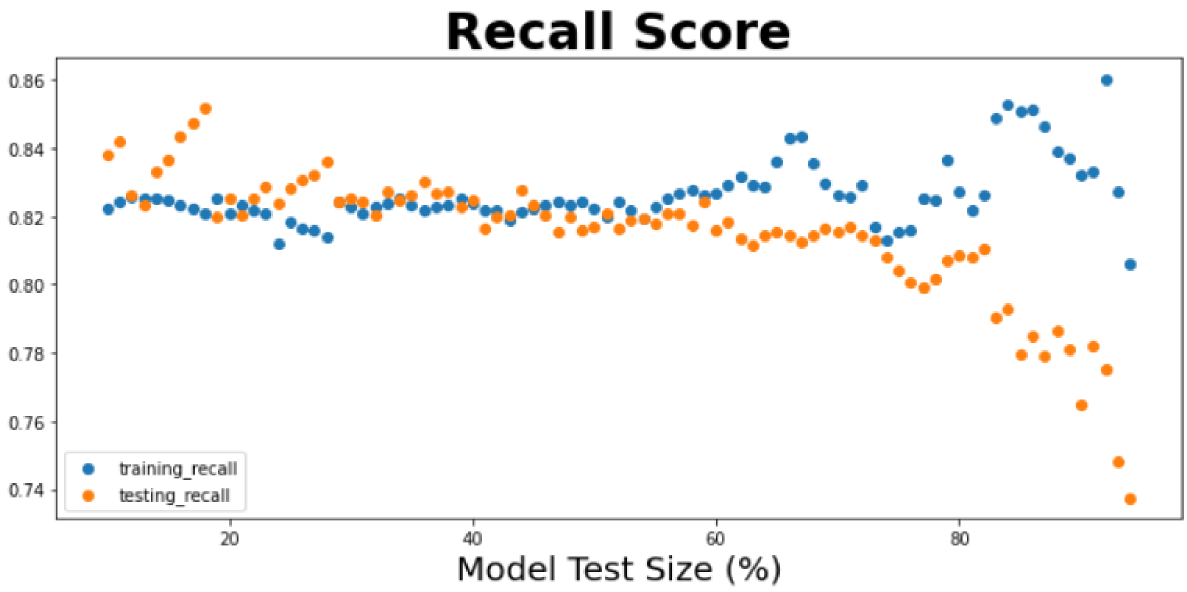


Figure 26. Logistic Regression – Recall Score

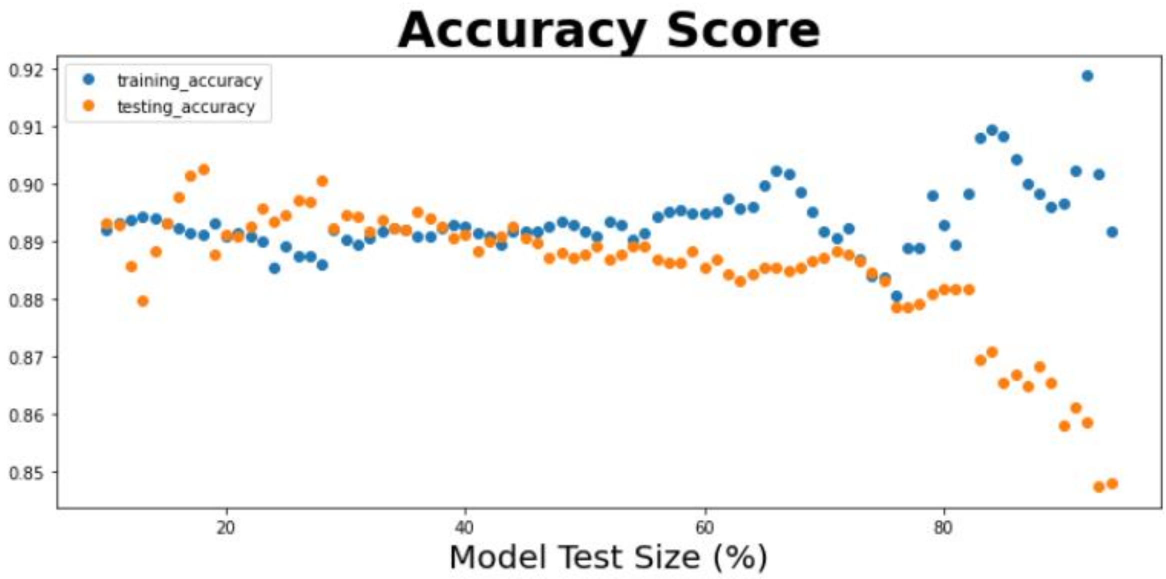


Figure 27. Logistic Regression – Accuracy Score

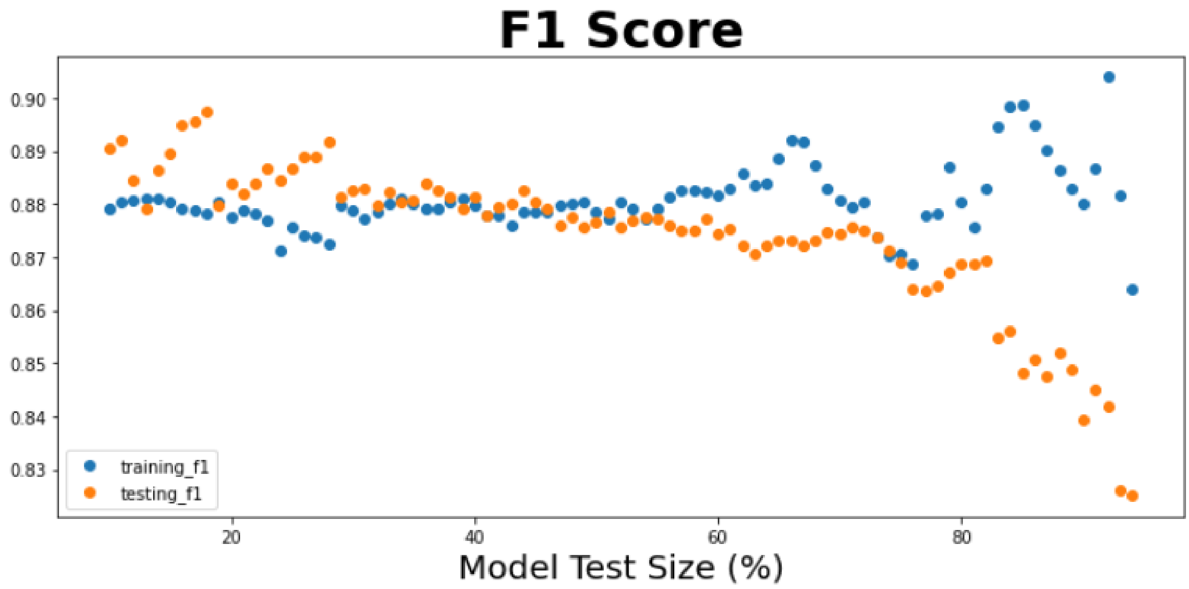


Figure 28. Logistic Regression – F1 Score

By using the above model, we will now generate the same metrics for the benchmark dataset in order to see the validity of the logistic regression model.

Table 6. Confusion Matrix – Logistic Regression

	Actual-Churner	**Non-Churner**
Predicted-Churner	1153	128
Predicted-Non-churner	224	256

- Accuracy = 0.80
- Precision = 0.66
- Recall = 0.53
- F1 Score = 0.59

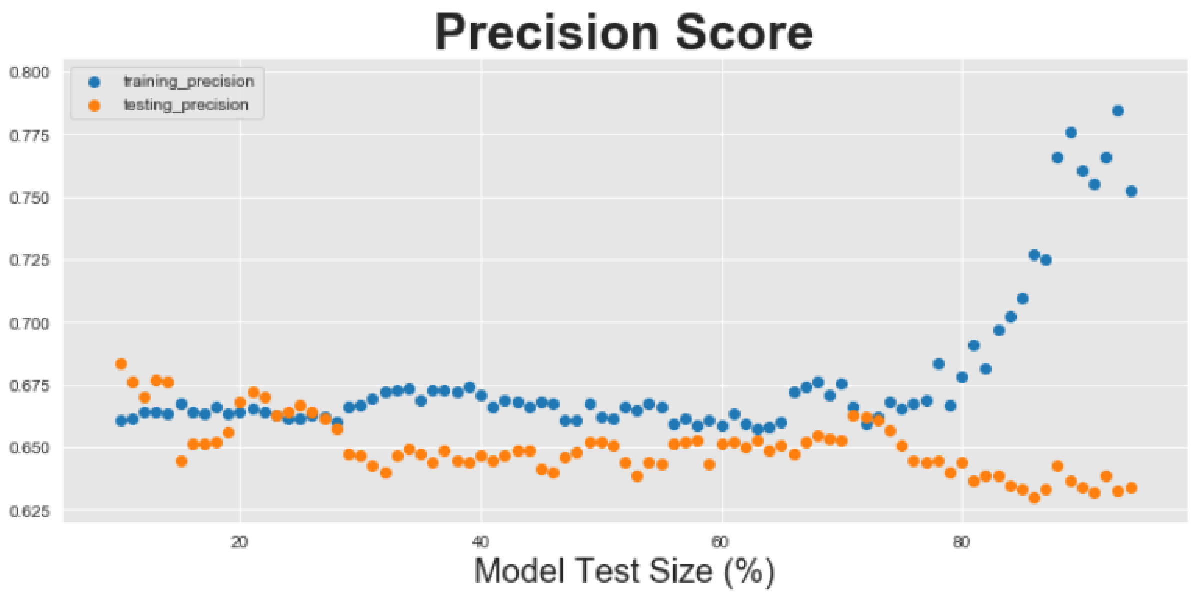


Figure 29. Logistic Regression (benchmark data) – Precision Score

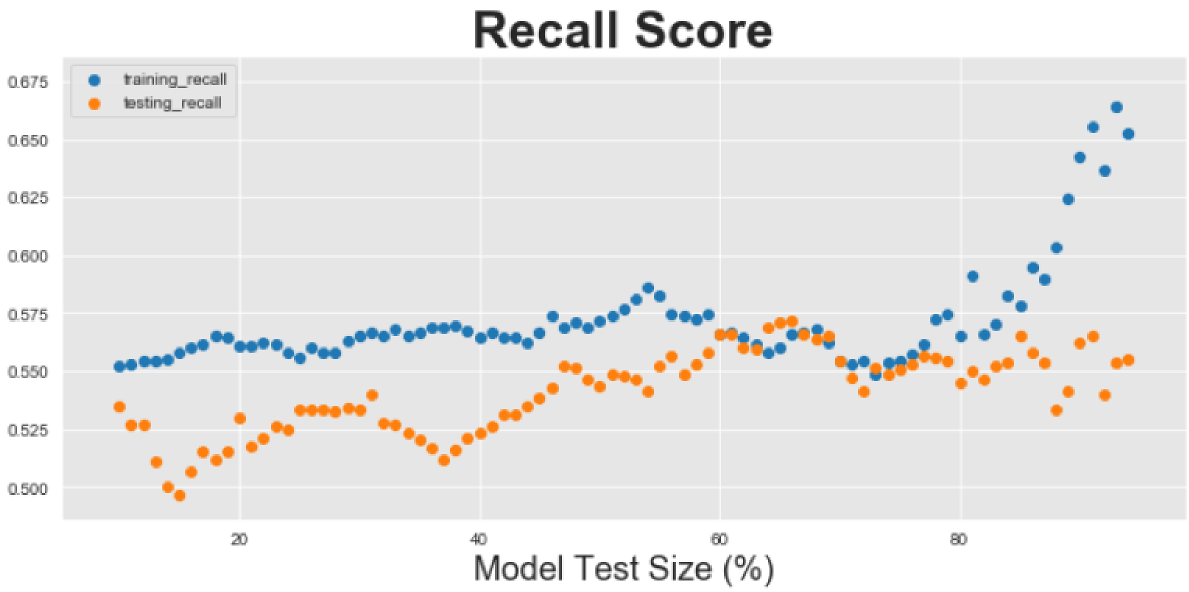


Figure 30. Logistic Regression (benchmark data) – Recall Score

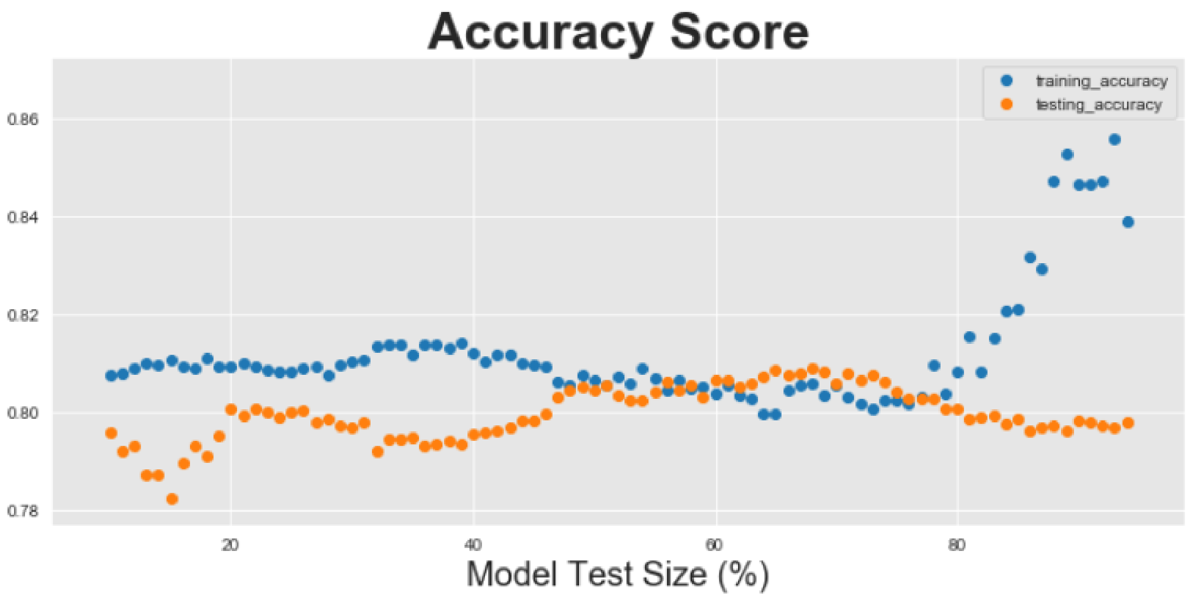


Figure 31. Logistic Regression (benchmark data) – Accuracy Score

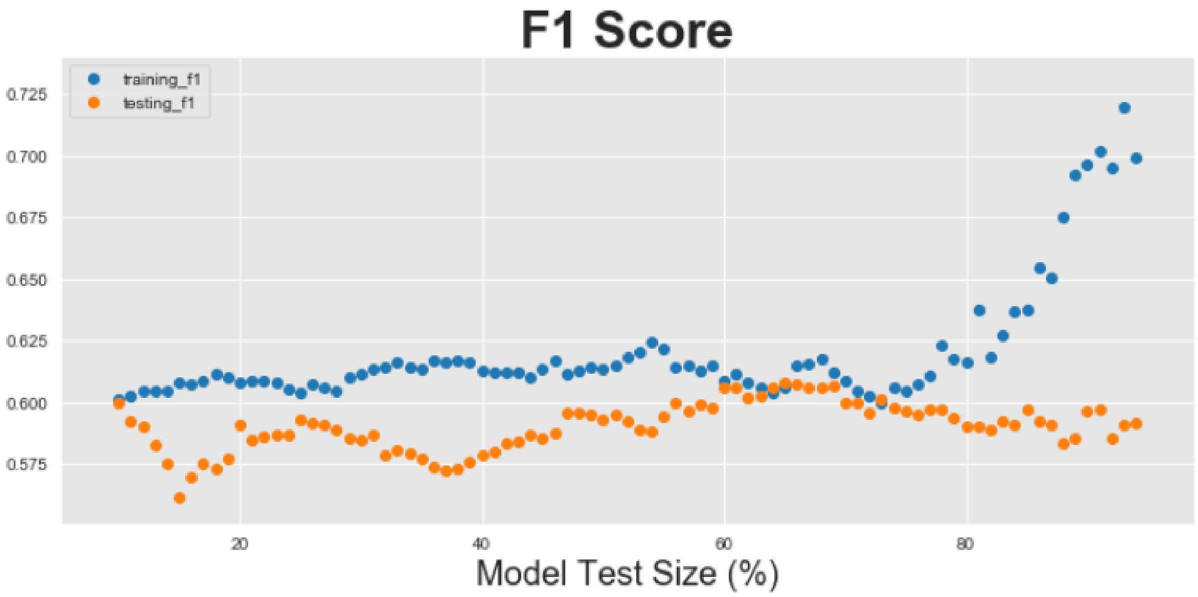


Figure 32. Logistic Regression (benchmark data) – F1 Score

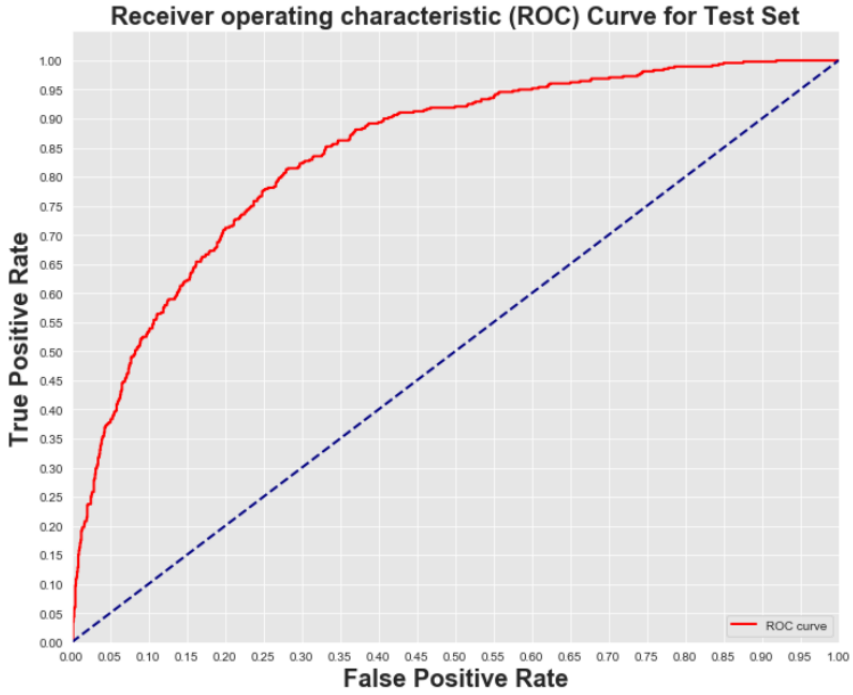


Figure 33. Logistic Regression (benchmark data) – AUC = 0.83

4.2 Decision Tree Model Evaluation

Table 7. Confusion Matrix – Decision Tree

	Actual Churner	**Non Churner**
Predicted Churner	**312**	**17**
Predicted Non churner	**66**	**260**

- Accuracy = 0.87
- Precision = 0.93
- Recall = 0.79
- F1 Score = 0.86

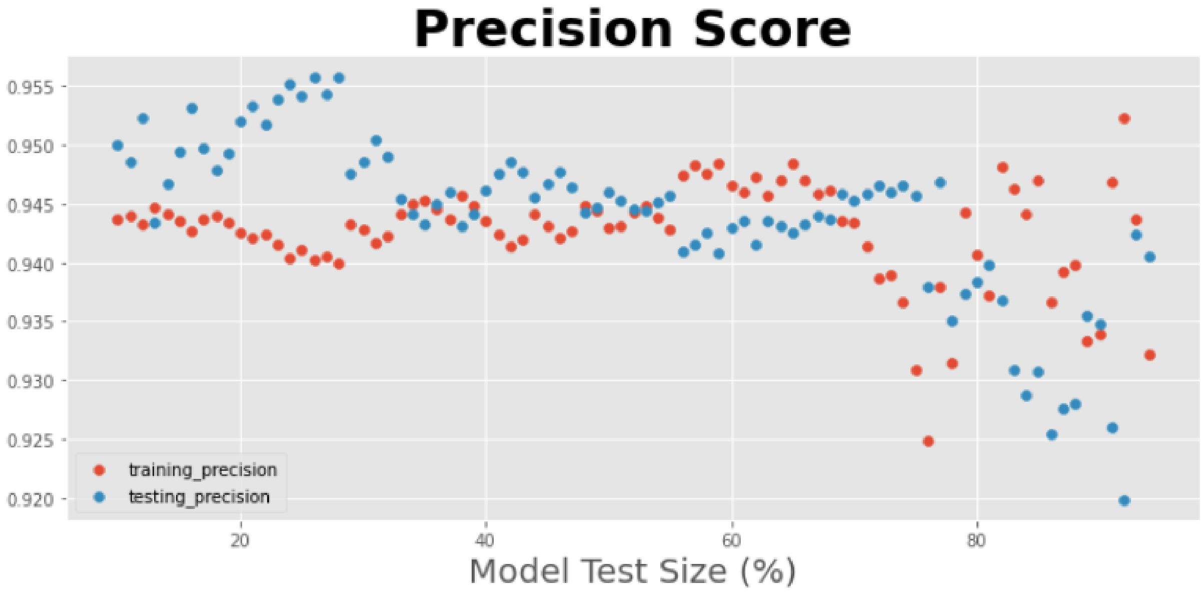


Figure 34. Decision Tree – Precision Score

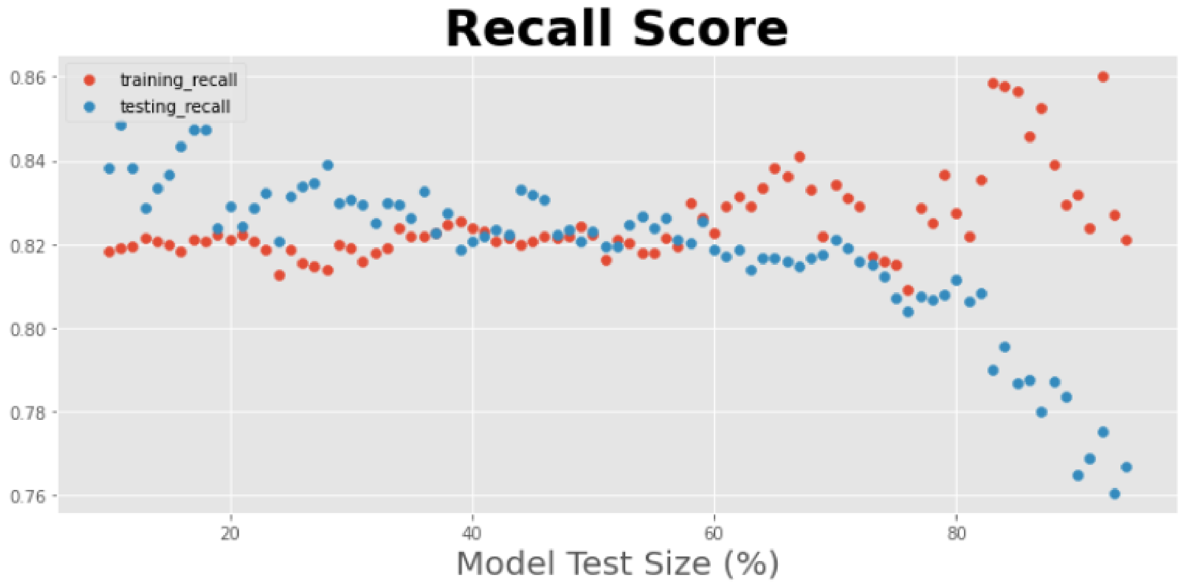


Figure 35. Decision Tree – Recall Score

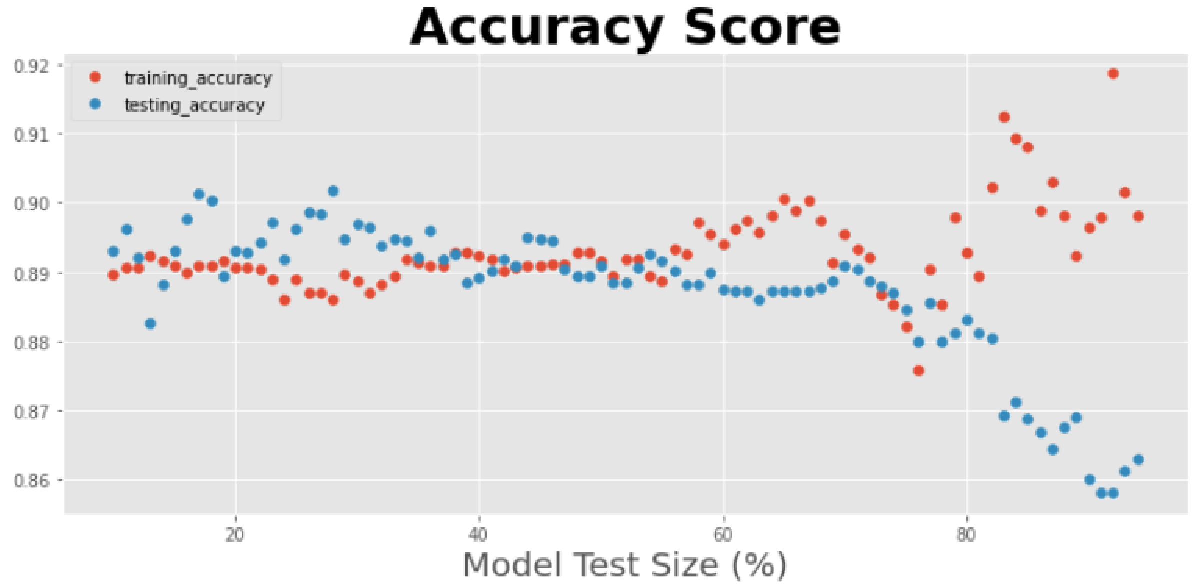


Figure 36. Decision Tree – Accuracy Score

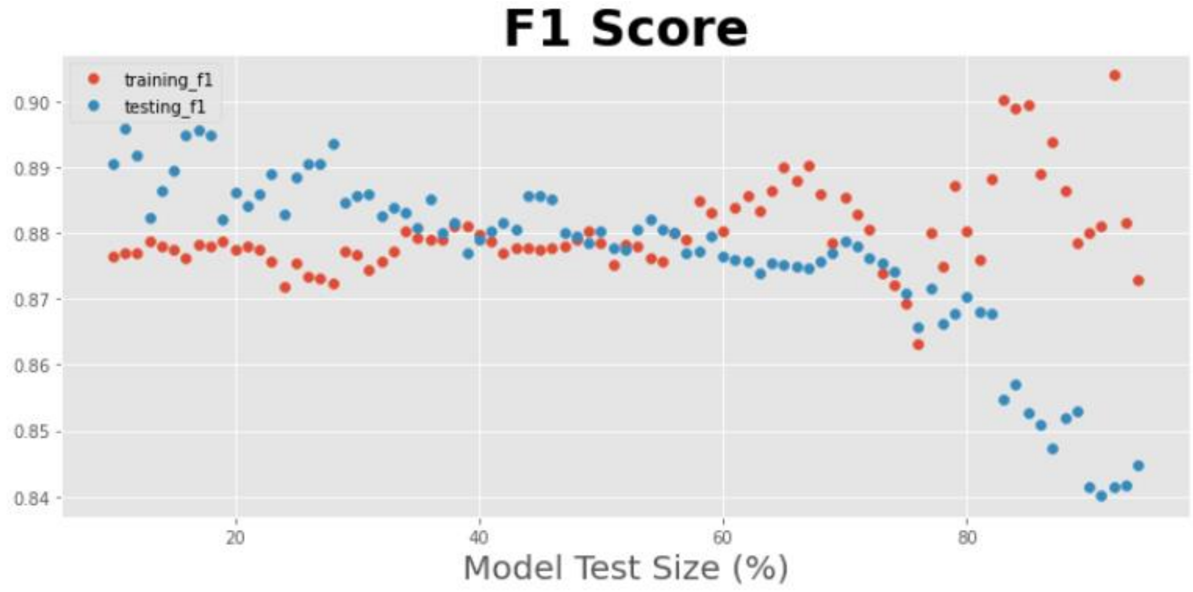


Figure 37. Decision Tree – F1 Score

By using the above model, we will now generate the same metrics for the benchmark dataset in order to see the validity of the Decision Tree model.

Table 8. Benchmark Data Confusion Matrix – Decision Tree

	Actual Churner	Non Churner
Predicted Churner	1053	228
Predicted Non churner	259	221

- Accuracy = 0.72
- Precision = 0.49
- Recall = 0.46
- F1 Score = 0.47

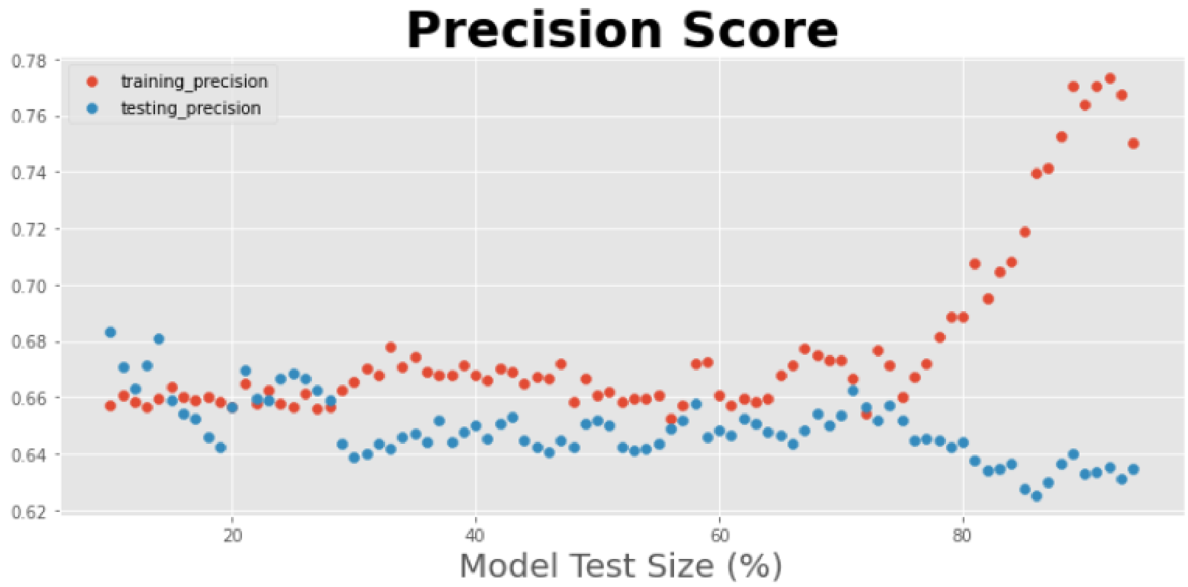


Figure 38. Decision Tree (benchmark data) – Precision Score

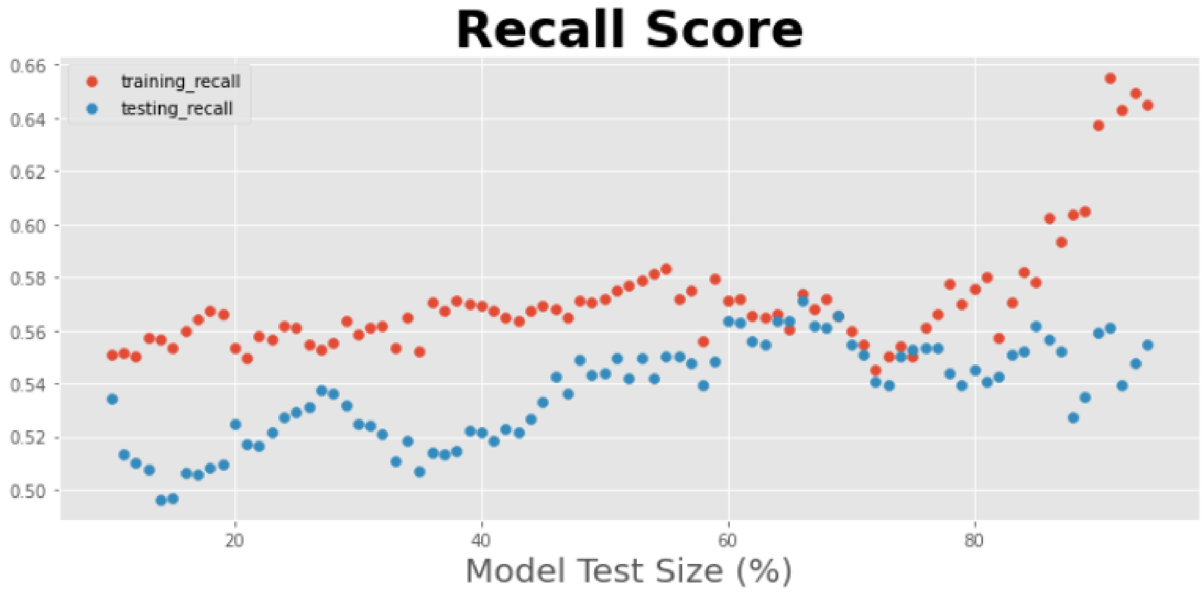


Figure 39. Decision Tree (benchmark data) – Recall Score

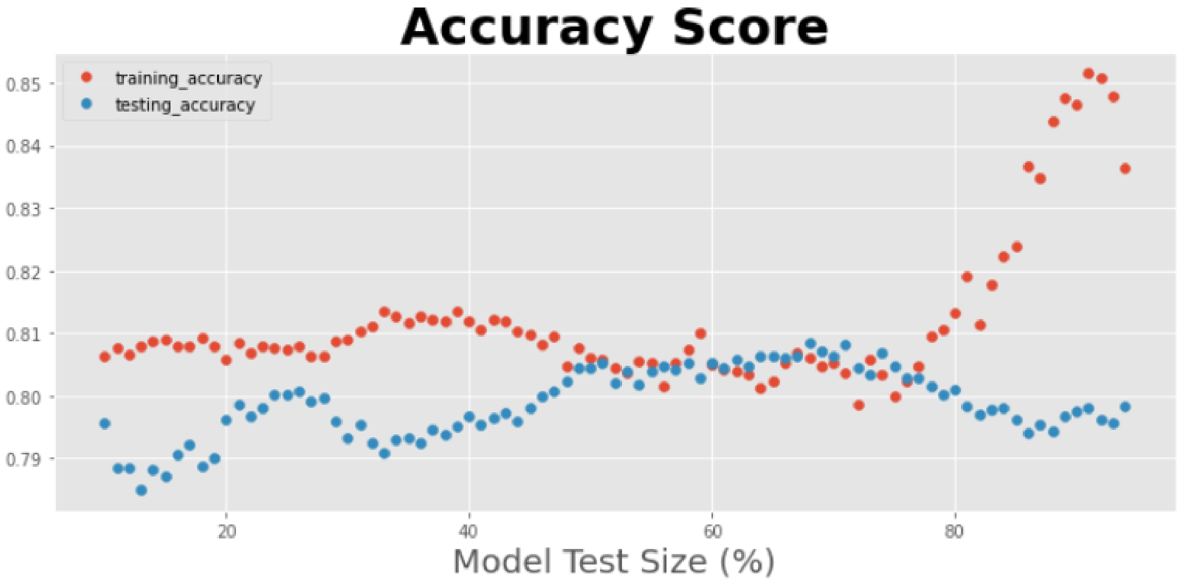


Figure 40. Decision Tree (benchmark data) – Accuracy Score

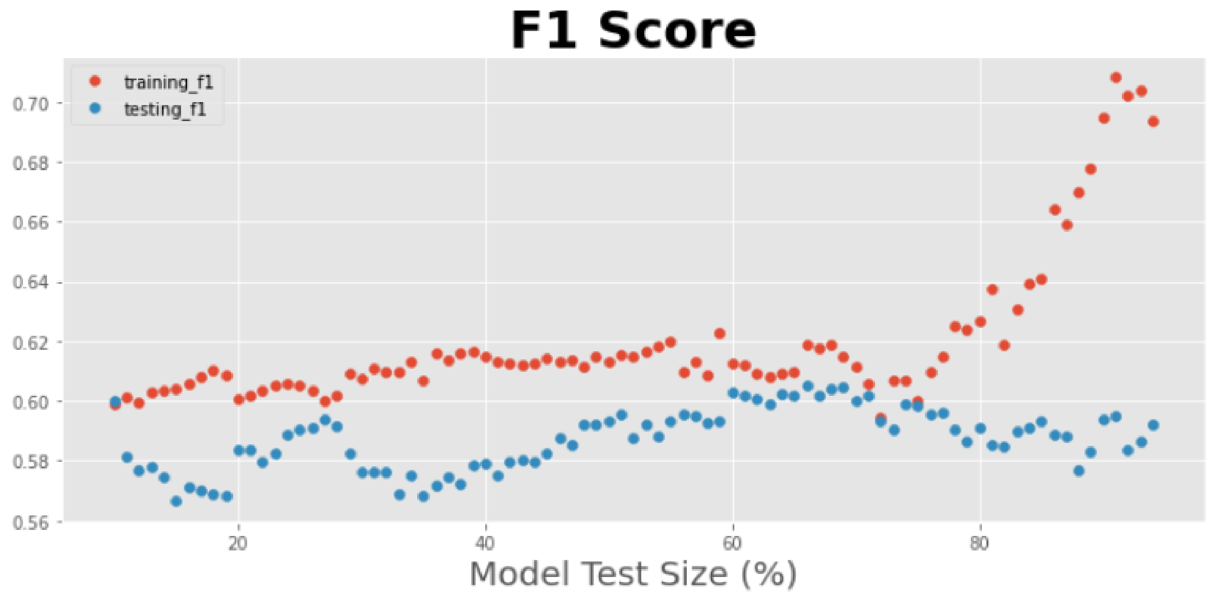


Figure 41. Decision Tree (benchmark data) – F1 Score

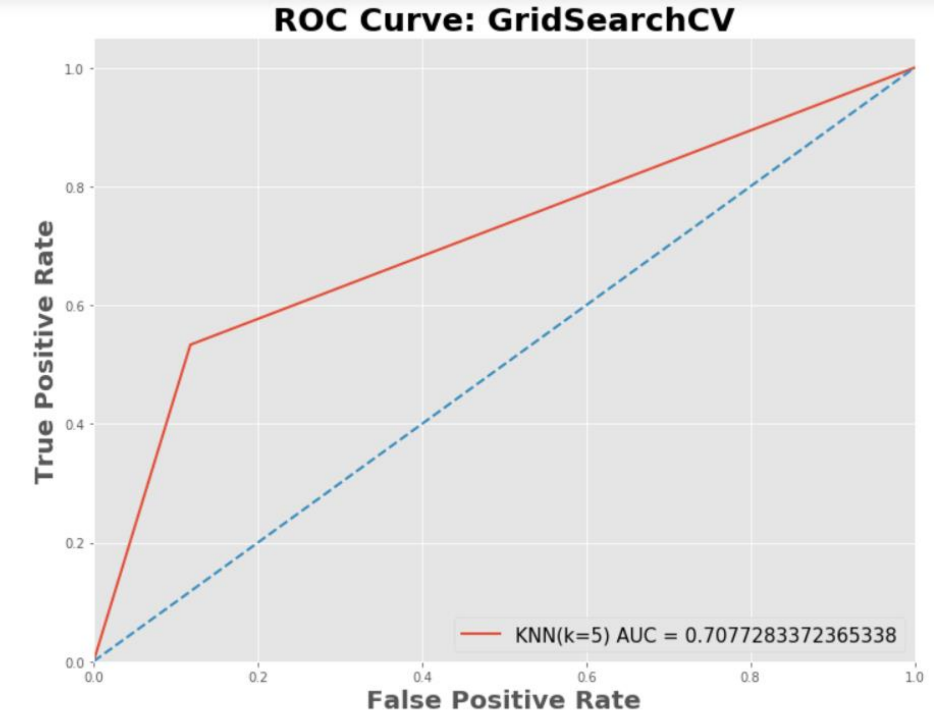


Figure 42. Decision Tree (benchmark data) – AUC = 0.70

4.3 Neural Networks MLP Model Evaluation

Table 9. Results analysis using a Confusion Matrix

	Actual Churner	Non Churner
Predicted Churner	**264**	**18**
Predicted Non churner	**29**	**213**

- Accuracy = 0.91
- Precision = 0.92
- Recall = 0.88
- F1 Score = 0.90

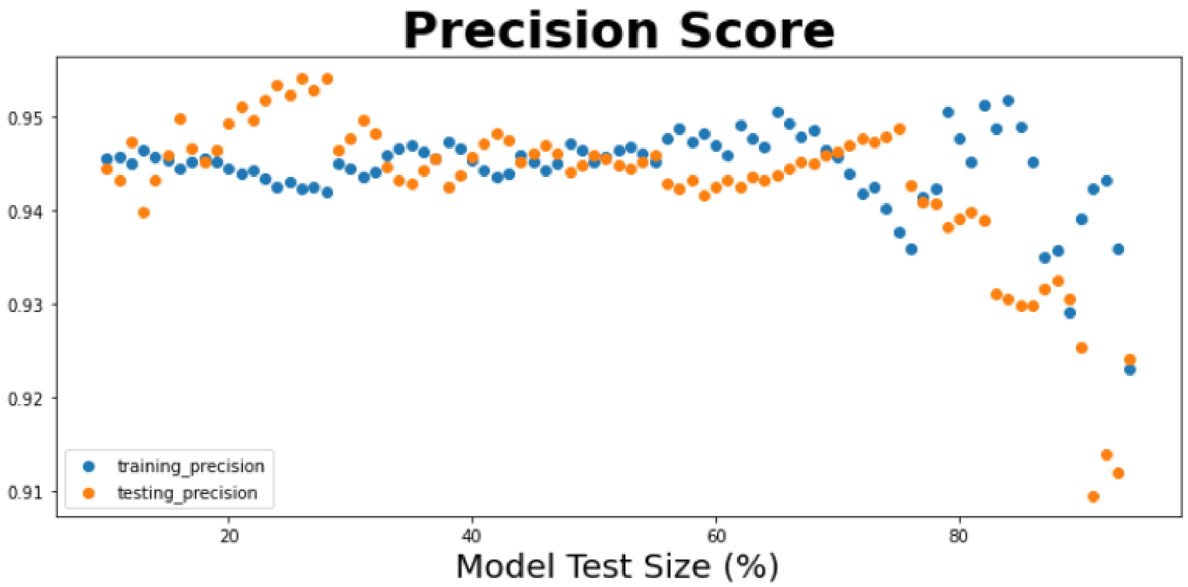


Figure 43. MLP – Precision Score

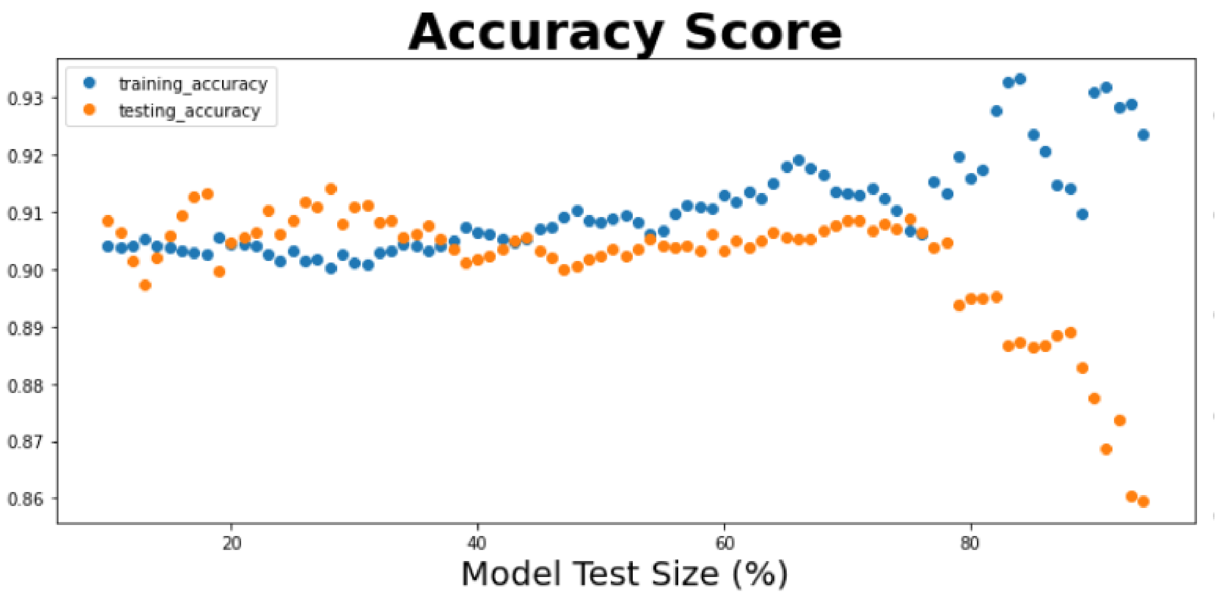


Figure 44. MLP – Accuracy Score

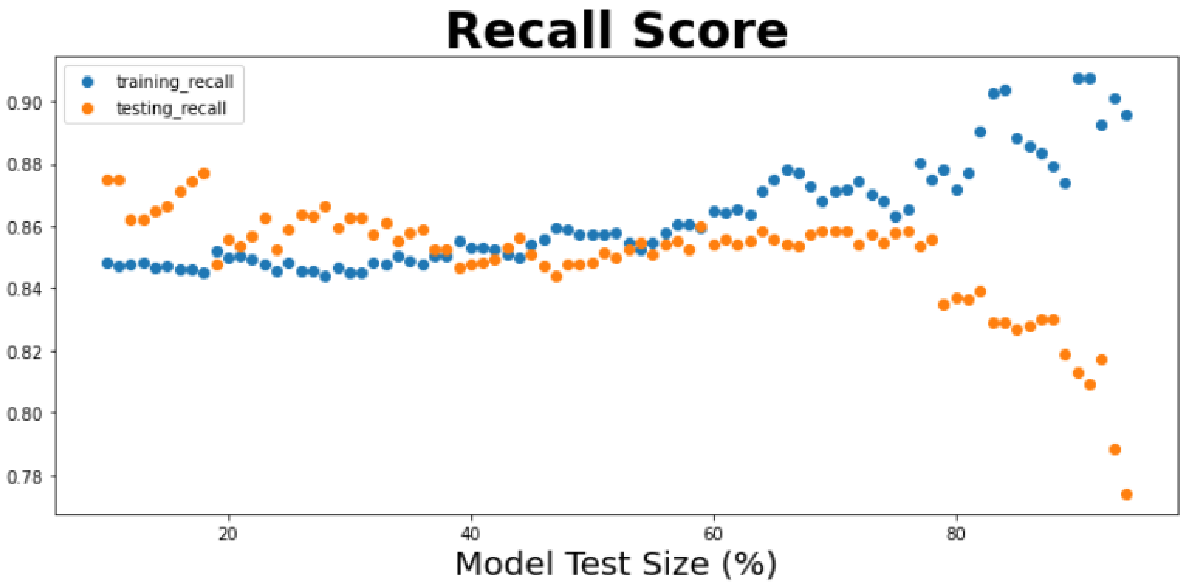


Figure 45. MLP – Recall Score

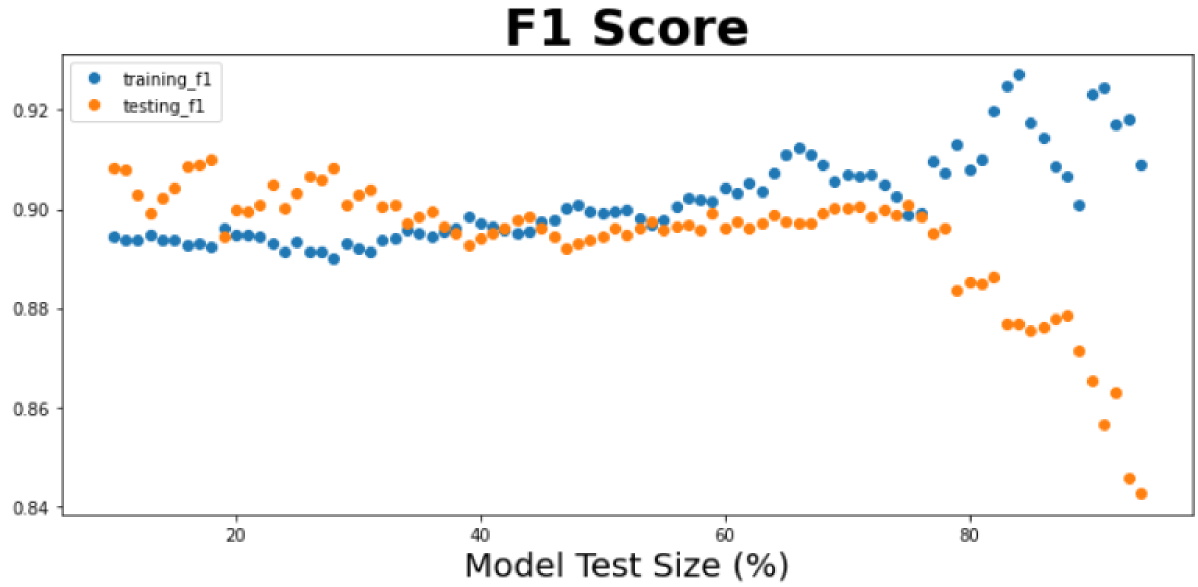


Figure 46. MLP – F1 Score

By using the above model, we will now generate the same metrics for the benchmark dataset in order to see the validity of the MLP model.

Table 10. Benchmark Data Confusion Matrix – Multi-Layer Perceptron Model

	Actual Churner	**Non Churner**
Predicted Churner	939	75
Predicted Non churner	203	190

- Accuracy = 0.81
- Precision = 0.70
- Recall = 0.49
- F1 Score = 0.57

Table 11. Descriptive metrics comparison – Production Data

	Logistic Regression	Decision Tree	MLP
Accuracy	**0.89**	**0.87**	**0.91**
Precision	**0.95**	**0.93**	**0.92**
Recall	**0.83**	**0.79**	**0.88**
F1 Score	**0.89**	**0.86**	**0.90**

4.4 Model Comparison

As per the above metrics, MLP has the highest values except for the precision. However, the highest level of accuracy can be seen in Multi-Layer Perceptron in Neural Networks. Hence, we can consider this as the model which can be used to predict the potential churners based on the features of the production data.

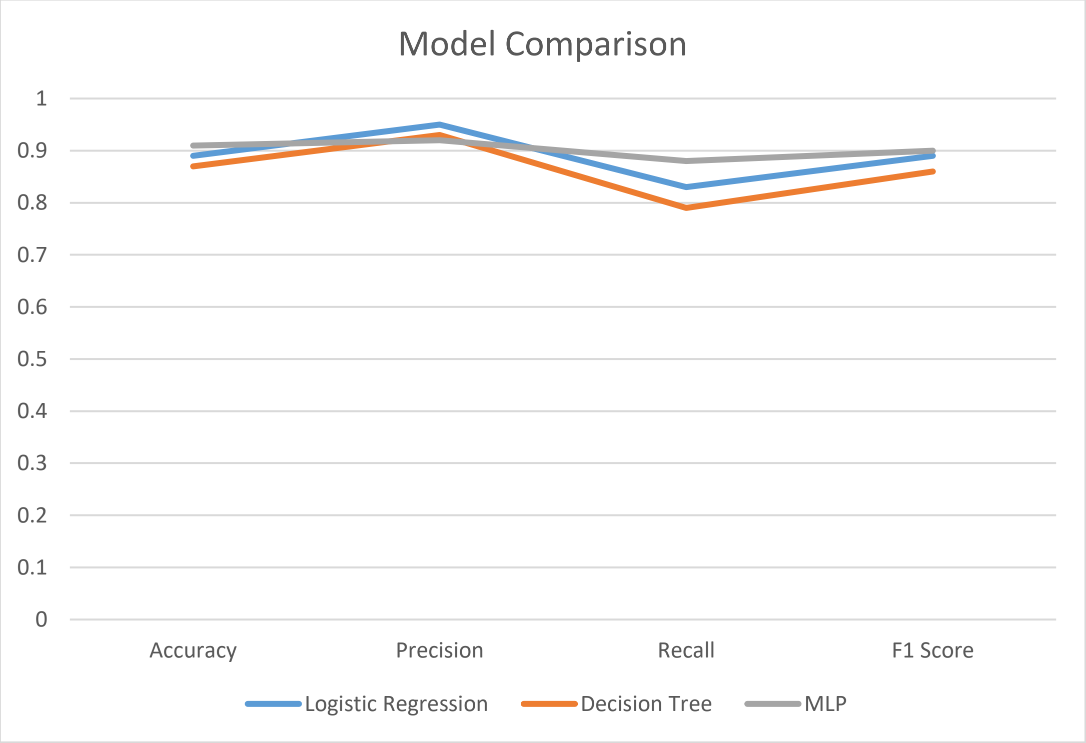


Figure 47. Model Comparison

On top of the above results and, as an alternative to the confusion matrix, we will be using the Receiver Operator Characteristic Curve (ROC) and Area Under Curve (AUC).

ROC is a visual graph which will illustrate the true positive rate against the false positive rate of our classifier.

The AUC will give a singular numeric metric to compare instead of a visual representation. An AUC = 1 would represent a perfect classifier, and an AUC = 0.5 represents a classifier which only has 50% precision. This metric quantifies the overall accuracy of the classifier model.

4.4.1 ROC & AUC for Logistic Regression Model

- AUC = 0.9345

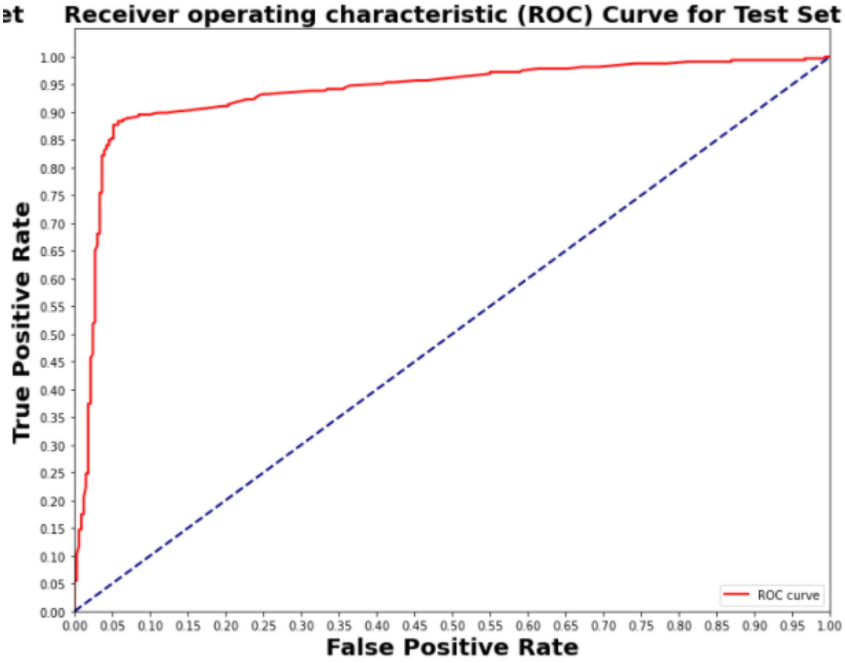


Figure 48. Logistic Regression model ROC Graph

4.4.2 ROC & AUC for Decision Tree Model

- AUC = 0.8729

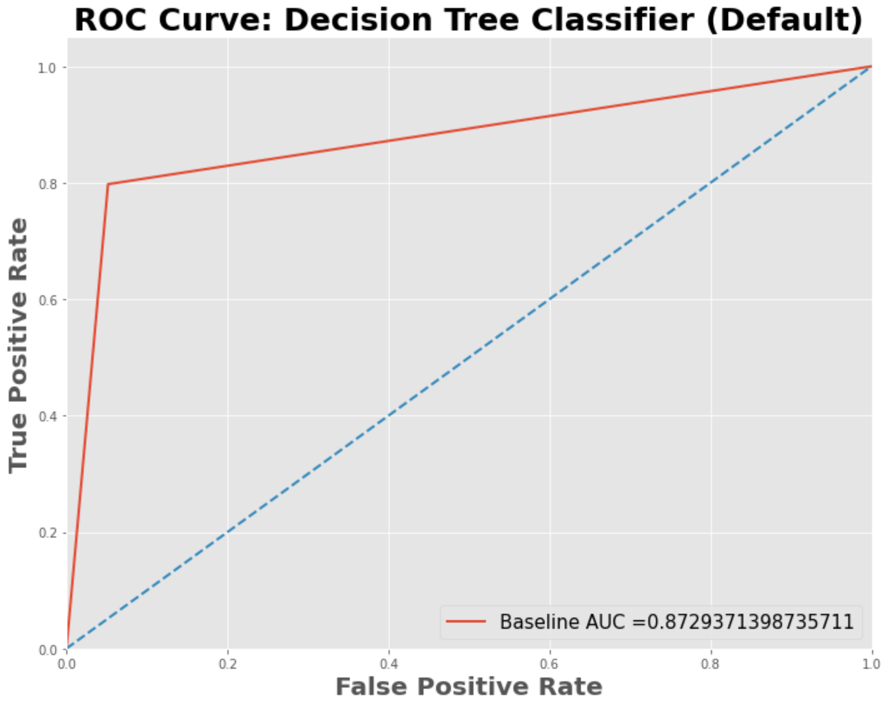


Figure 49. Decision Tree model ROC Graph

4.4.3 ROC & AUC for MLP Model

- AUC = 0.9163

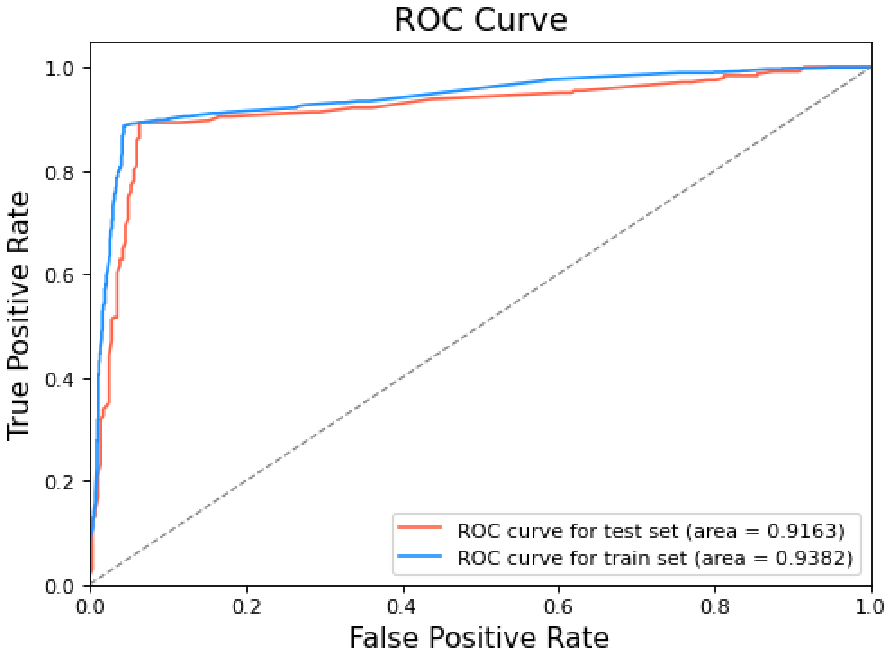


Figure 50. MLP ROC Graph

4.5 Tools in Use

Main tool used throughout this research is the Jupyter Notebook powered by Python. It is an open-source web application which can be used to create and share documents that contain live code, equations, visualizations, and narrative text.

Figure 51. Python Logo

Figure 52. Jupyter Notebook Logo

Starting from Exploratory Data Analysis (EDA), we have used it for data cleaning and transformation, numerical simulation, classification models and data visualization. It has been a comprehensive tool which was used to generate images, graphs and most of all, have all the findings in a centralized place which makes it so convenient to access at any given time. On top of that, the Jupyter Notebook was launched through an all-in-one navigator called Anaconda, which is a combined distribution of both R and Python programming languages mainly focused on scientific computing. Also, Anaconda aims at convenient and simplified deployment and package management [13].

Figure 53. Anaconda Logo

CHAPTER 5

CONCLUSION AND FUTURE WORK

This study is mainly focused on to find a solution to customer churn or attrition by predicting the potential churners through a targeted proactive manner. Churn prediction was performed to a telecommunication company called Xcom which is based in Sydney, Australia. Identifying potential churners is a highly challenging task and if the analysis is not done properly, there could be many losses to the company Xcom in both time and money. Hence, this study analyses a produced dataset given by Xcom and through a wide analytical phase for exploratory data analysis, the features or the derived variables were discovered. On top of that, by using machine learning models, the potential churners were identified. The models were evaluated by using the descriptive metrics accuracy, precision, recall and F1-score. ROC and AUC values made it even more helpful in the evaluation process.

Based on the evaluation results, the Multi-Layer Perceptron model of Artificial Neural Networks was identified as the most accurate model. It gave comparatively good readings for the above-mentioned descriptive metrics. As mentioned before, this study was carried out by using a production dataset released by the company Xcom. In order to make the churn detection models much accurate and realistic, we may also consider using the broadband usage data on top of the generic service-related data.

However, this study can be extended by including much more advanced machine learning models such as Extreme Gradient Boost or XGBoost. Among the tree-based machine learning algorithms, XGBoost is a decision tree-based ensemble machine learning algorithm. It uses gradient boosting [14] framework which is a technique for regression and classification.

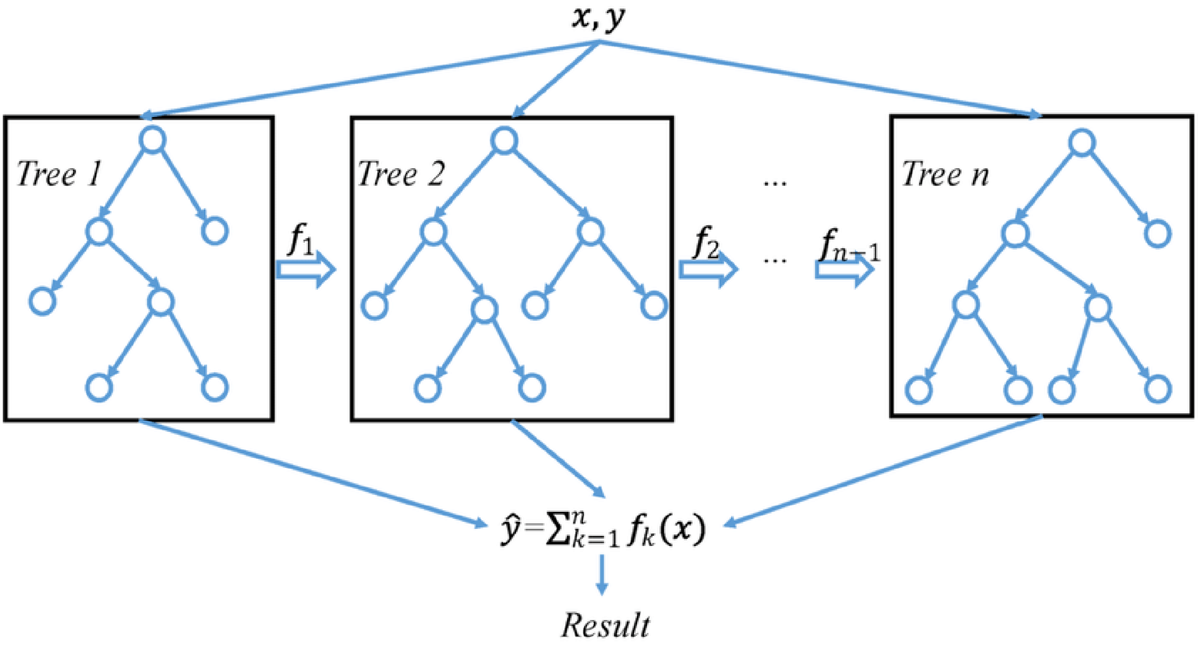
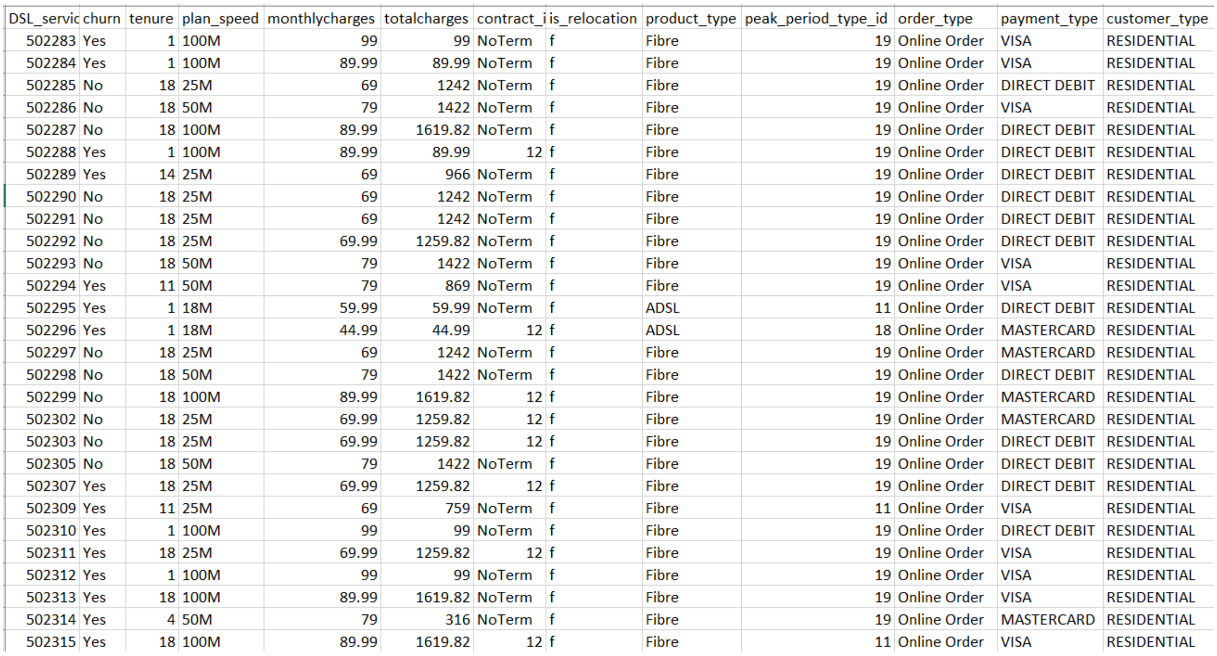


Figure 54. General architecture for XGBoost [16]

LIST OF APPENDICES

APPENDIX 1 – RAW PRODUCTION DATA

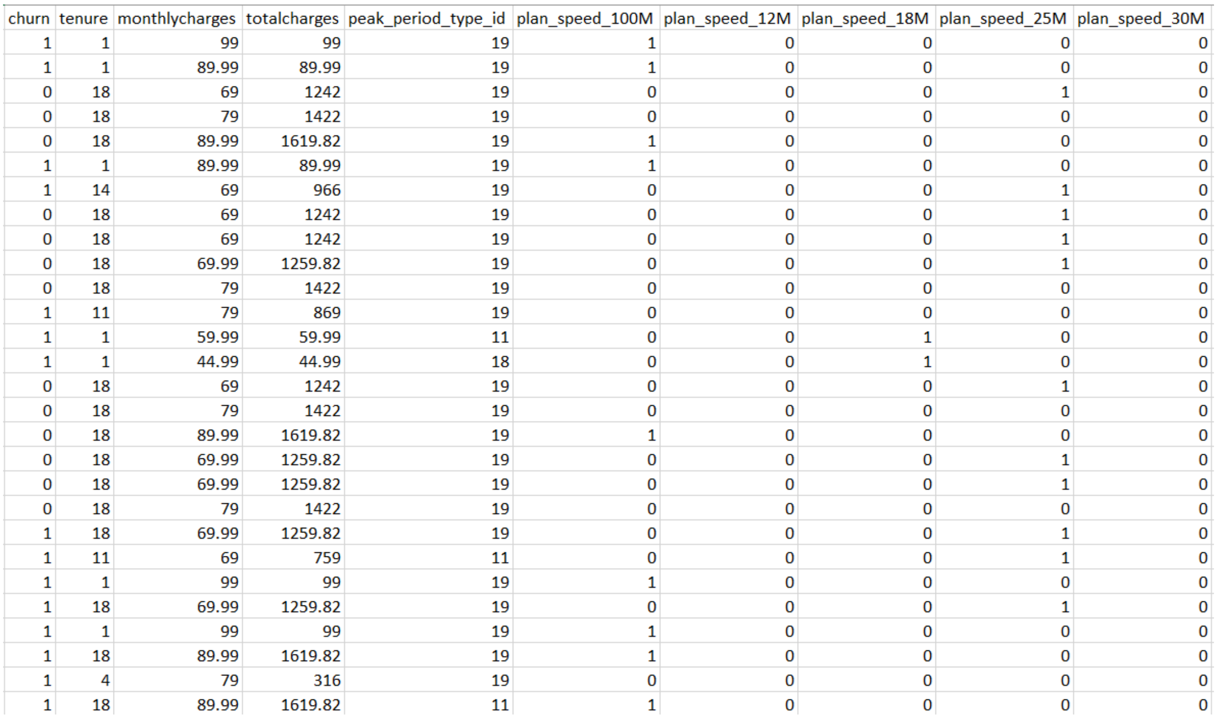
Table 12. Extract of Production Data



DSL_servic	churn	tenure	plan_speed	monthlycharges	totalcharges	contract_i	is_relocation	product_type	peak_period_type_id	order_type	payment_type	customer_type
502283	Yes	1	100M	99	99	NoTerm	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502284	Yes	1	100M	89.99	89.99	NoTerm	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502285	No	18	25M	69	1242	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502286	No	18	50M	79	1422	NoTerm	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502287	No	18	100M	89.99	1619.82	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502288	Yes	1	100M	89.99	89.99	12	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502289	Yes	14	25M	69	966	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502290	No	18	25M	69	1242	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502291	No	18	25M	69	1242	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502292	No	18	25M	69.99	1259.82	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502293	No	18	50M	79	1422	NoTerm	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502294	Yes	11	50M	79	869	NoTerm	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502295	Yes	1	18M	59.99	59.99	NoTerm	f	ADSL	11	Online Order	DIRECT DEBIT	RESIDENTIAL
502296	Yes	1	18M	44.99	44.99	12	f	ADSL	18	Online Order	MASTERCARD	RESIDENTIAL
502297	No	18	25M	69	1242	NoTerm	f	Fibre	19	Online Order	MASTERCARD	RESIDENTIAL
502298	No	18	50M	79	1422	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502299	No	18	100M	89.99	1619.82	12	f	Fibre	19	Online Order	MASTERCARD	RESIDENTIAL
502302	No	18	25M	69.99	1259.82	12	f	Fibre	19	Online Order	MASTERCARD	RESIDENTIAL
502303	No	18	25M	69.99	1259.82	12	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502305	No	18	50M	79	1422	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502307	Yes	18	25M	69.99	1259.82	12	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502309	Yes	11	25M	69	759	NoTerm	f	Fibre	11	Online Order	VISA	RESIDENTIAL
502310	Yes	1	100M	99	99	NoTerm	f	Fibre	19	Online Order	DIRECT DEBIT	RESIDENTIAL
502311	Yes	18	25M	69.99	1259.82	12	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502312	Yes	1	100M	99	99	NoTerm	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502313	Yes	18	100M	89.99	1619.82	NoTerm	f	Fibre	19	Online Order	VISA	RESIDENTIAL
502314	Yes	4	50M	79	316	NoTerm	f	Fibre	19	Online Order	MASTERCARD	RESIDENTIAL
502315	Yes	18	100M	89.99	1619.82	12	f	Fibre	11	Online Order	VISA	RESIDENTIAL

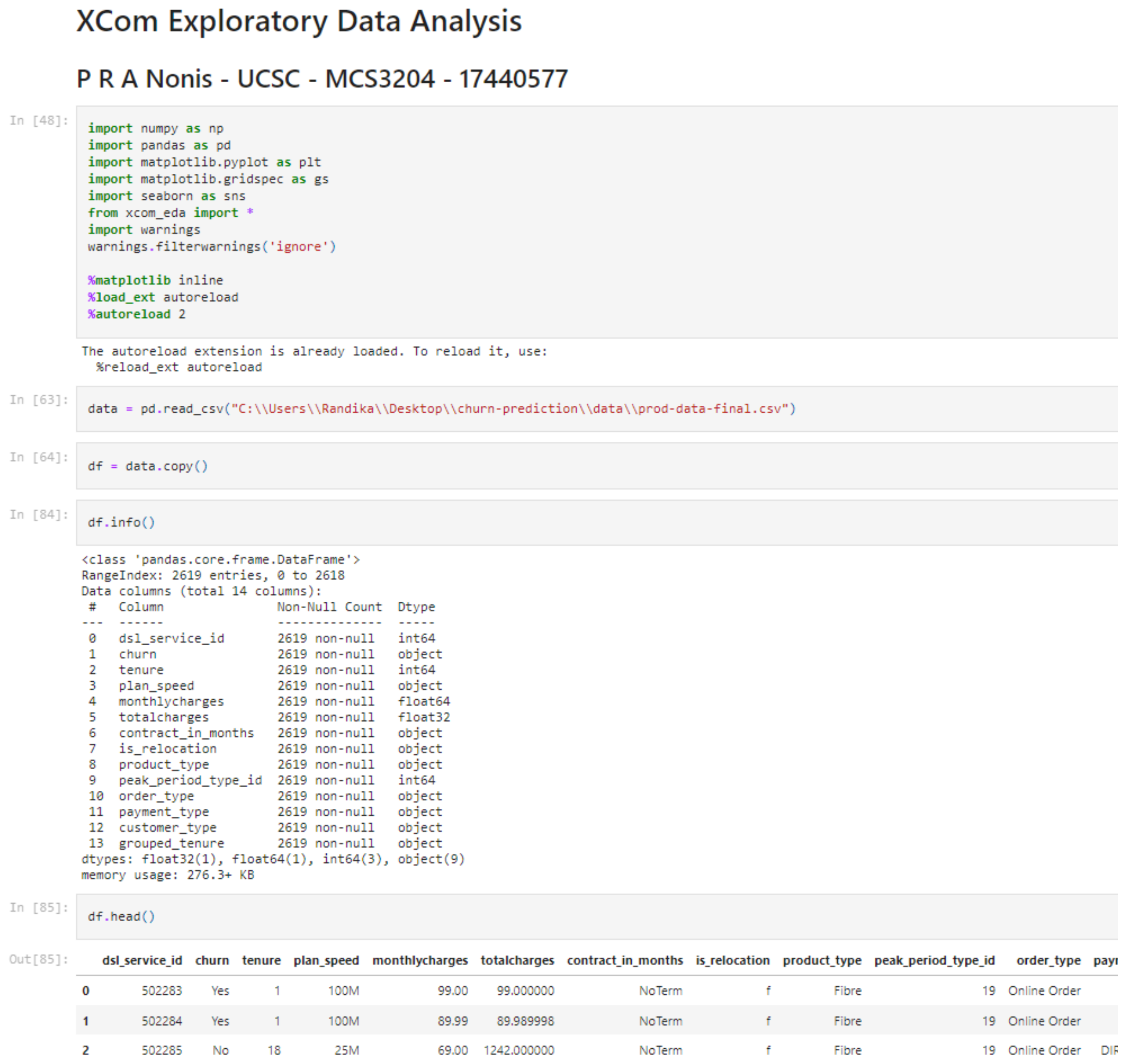
APPENDIX 2 – RAW PRODUCTION DATA BINARY

Table 13. Extract of Production Data in Binary Form



churn	tenure	monthlycharges	totalcharges	peak_period_type_id	plan_speed_100M	plan_speed_12M	plan_speed_18M	plan_speed_25M	plan_speed_30M
1	1	99	99	19	1	0	0	0	0
1	1	89.99	89.99	19	1	0	0	0	0
0	18	69	1242	19	0	0	0	1	0
0	18	79	1422	19	0	0	0	0	0
0	18	89.99	1619.82	19	1	0	0	0	0
1	1	89.99	89.99	19	1	0	0	0	0
1	14	69	966	19	0	0	0	1	0
0	18	69	1242	19	0	0	0	1	0
0	18	69	1242	19	0	0	0	1	0
0	18	69.99	1259.82	19	0	0	0	1	0
0	18	79	1422	19	0	0	0	0	0
1	11	79	869	19	0	0	0	0	0
1	1	59.99	59.99	11	0	0	1	0	0
1	1	44.99	44.99	18	0	0	1	0	0
0	18	69	1242	19	0	0	0	1	0
0	18	79	1422	19	0	0	0	0	0
0	18	89.99	1619.82	19	1	0	0	0	0
0	18	69.99	1259.82	19	0	0	0	1	0
0	18	69.99	1259.82	19	0	0	0	1	0
0	18	79	1422	19	0	0	0	0	0
1	18	69.99	1259.82	19	0	0	0	1	0
1	11	69	759	11	0	0	0	1	0
1	1	99	99	19	1	0	0	0	0
1	18	69.99	1259.82	19	0	0	0	1	0
1	1	99	99	19	1	0	0	0	0
1	18	89.99	1619.82	19	1	0	0	0	0
1	4	79	316	19	0	0	0	0	0
1	18	89.99	1619.82	11	1	0	0	0	0

APPENDIX 3 – Jupyter Notebook Captures



XCom Exploratory Data Analysis

P R A Nonis - UCSC - MCS3204 - 17440577

In [48]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as gs
import seaborn as sns
from xcom_eda import *
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [63]:

```
data = pd.read_csv("C:\\Users\\Randika\\Desktop\\churn-prediction\\data\\prod-data-final.csv")
```

In [64]:

```
df = data.copy()
```

In [84]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2619 entries, 0 to 2618
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   dsl_service_id      2619 non-null   int64  
 1   churn               2619 non-null   object 
 2   tenure              2619 non-null   int64  
 3   plan_speed          2619 non-null   object 
 4   monthlycharges      2619 non-null   float64
 5   totalcharges        2619 non-null   float32
 6   contract_in_months  2619 non-null   object 
 7   is_relocation       2619 non-null   object 
 8   product_type        2619 non-null   object 
 9   peak_period_type_id  2619 non-null   int64  
 10  order_type          2619 non-null   object 
 11  payment_type        2619 non-null   object 
 12  customer_type       2619 non-null   object 
 13  grouped_tenure      2619 non-null   object 
dtypes: float32(1), float64(1), int64(3), object(9)
memory usage: 276.3+ KB
```

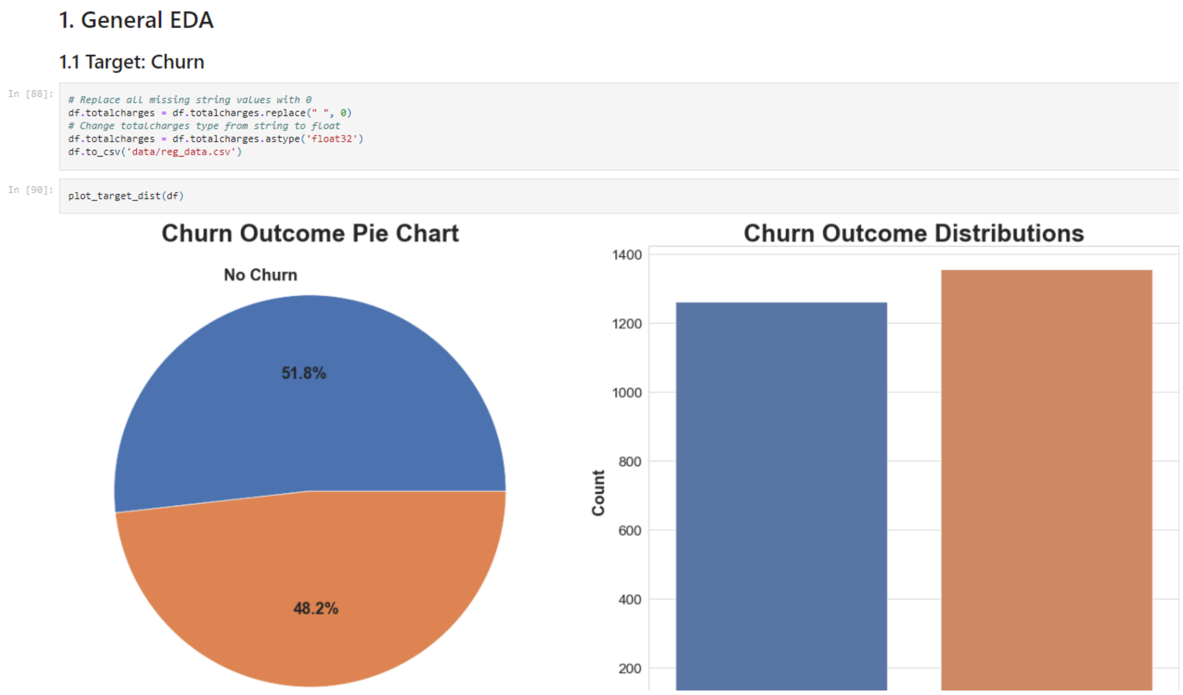
In [85]:

```
df.head()
```

Out[85]:

	dsl_service_id	churn	tenure	plan_speed	monthlycharges	totalcharges	contract_in_months	is_relocation	product_type	peak_period_type_id	order_type	payı
0	502283	Yes	1	100M	99.00	99.000000	NoTerm	f	Fibre	19	Online Order	
1	502284	Yes	1	100M	89.99	89.989998	NoTerm	f	Fibre	19	Online Order	
2	502285	No	18	25M	69.00	1242.000000	NoTerm	f	Fibre	19	Online Order	DIF

Figure 55. Exploratory Data Analysis



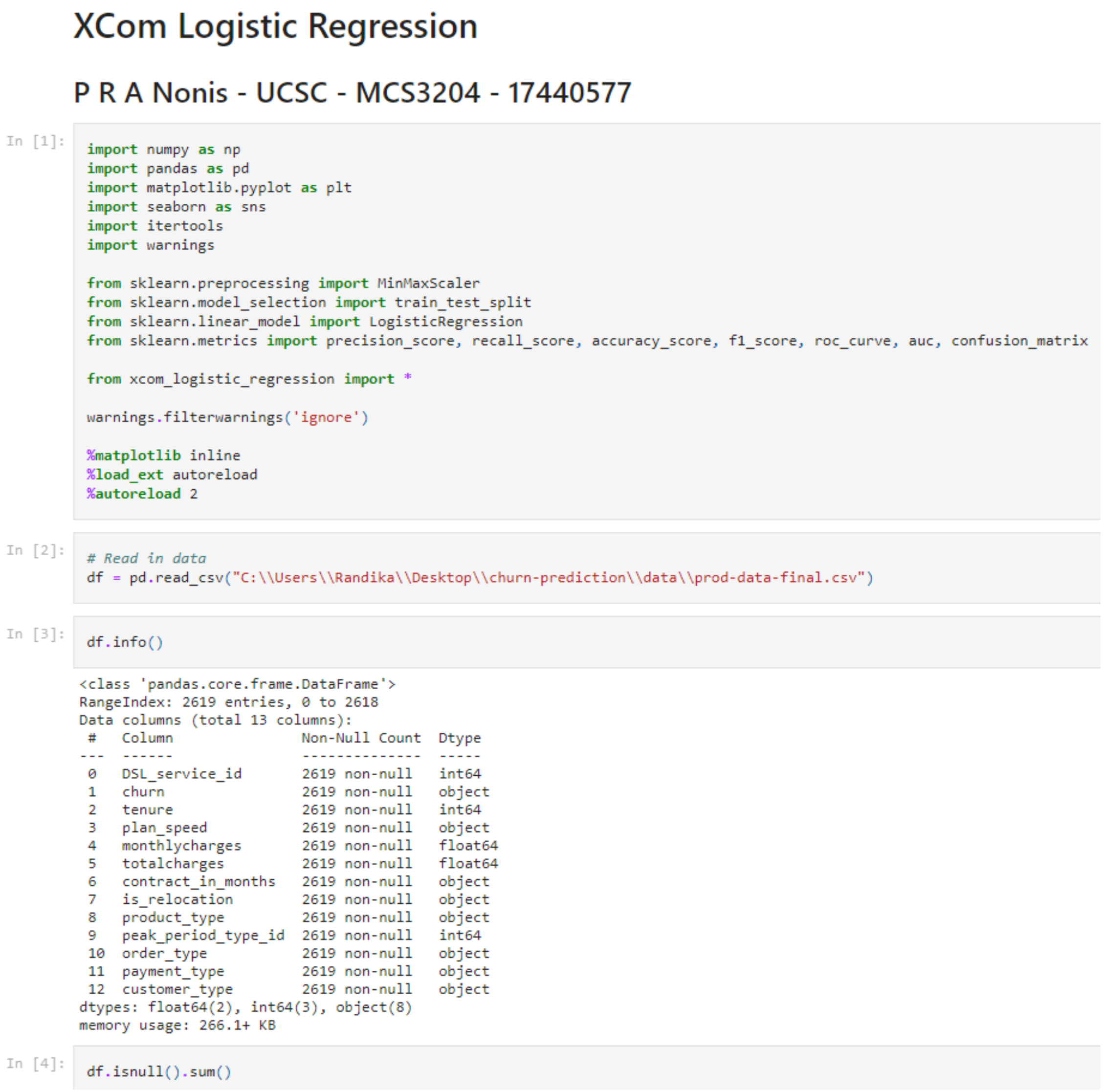
1. General EDA

1.1 Target: Churn

```
In [88]:
# Replace all missing string values with 0
df.totalcharges = df.totalcharges.replace(" ", 0)
# Change totalcharges type from string to float
df.totalcharges = df.totalcharges.astype('float32')
df.to_csv('data/reg_data.csv')
```

```
In [90]:
plot_target_dist(df)
```

Figure 56. Exploratory Data Analysis – Generated charts and graphs



XCom Logistic Regression

P R A Nonis - UCSC - MCS3204 - 17440577

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
import warnings

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import precision_score, recall_score, accuracy_score, f1_score, roc_curve, auc, confusion_matrix

from xcom_logistic_regression import *

warnings.filterwarnings('ignore')

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

In [2]:

```
# Read in data
df = pd.read_csv("C:\\Users\\Randika\\Desktop\\churn-prediction\\data\\prod-data-final.csv")
```

In [3]:

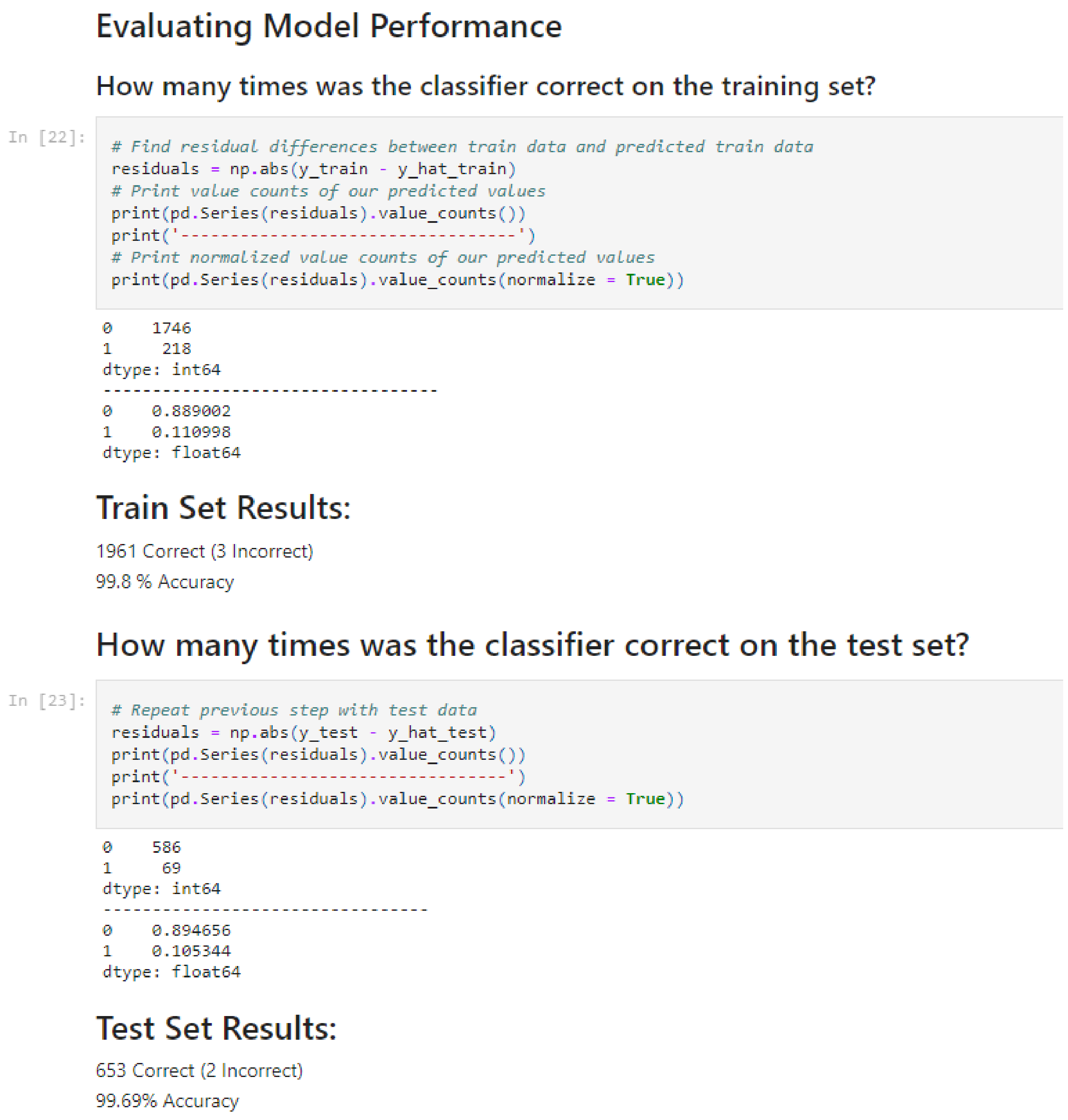
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2619 entries, 0 to 2618
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   DSL_service_id       2619 non-null   int64  
 1   churn                2619 non-null   object 
 2   tenure               2619 non-null   int64  
 3   plan_speed           2619 non-null   object 
 4   monthlycharges       2619 non-null   float64
 5   totalcharges         2619 non-null   float64
 6   contract_in_months   2619 non-null   object 
 7   is_relocation        2619 non-null   object 
 8   product_type         2619 non-null   object 
 9   peak_period_type_id  2619 non-null   int64  
 10  order_type           2619 non-null   object 
 11  payment_type         2619 non-null   object 
 12  customer_type        2619 non-null   object 
dtypes: float64(2), int64(3), object(8)
memory usage: 266.1+ KB
```

In [4]:

```
df.isnull().sum()
```

Figure 57. Logistic Regression



Evaluating Model Performance

How many times was the classifier correct on the training set?

In [22]:

```
# Find residual differences between train data and predicted train data
residuals = np.abs(y_train - y_hat_train)
# Print value counts of our predicted values
print(pd.Series(residuals).value_counts())
print('----------------------------------')
# Print normalized value counts of our predicted values
print(pd.Series(residuals).value_counts(normalize = True))
```

```
0    1746
1     218
dtype: int64
----------------------------------
0    0.889002
1    0.110998
dtype: float64
```

Train Set Results:

1961 Correct (3 Incorrect)

99.8 % Accuracy

How many times was the classifier correct on the test set?

In [23]:

```
# Repeat previous step with test data
residuals = np.abs(y_test - y_hat_test)
print(pd.Series(residuals).value_counts())
print('---------------------------------')
print(pd.Series(residuals).value_counts(normalize = True))
```

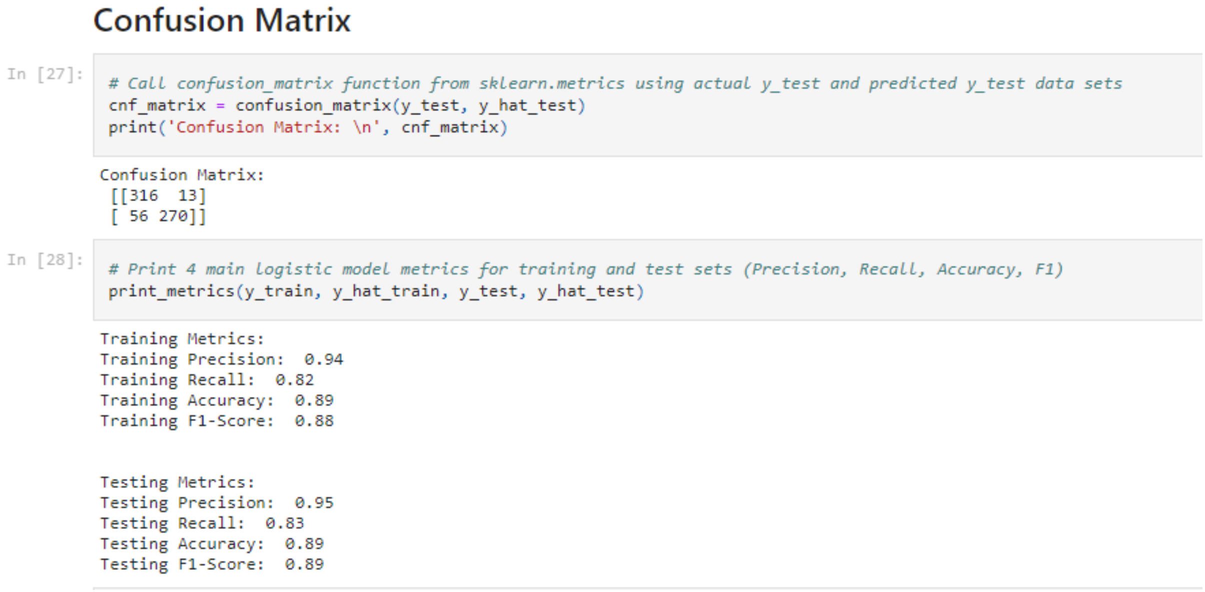
```
0    586
1     69
dtype: int64
---------------------------------
0    0.894656
1    0.105344
dtype: float64
```

Test Set Results:

653 Correct (2 Incorrect)

99.69% Accuracy

Figure 58. Logistic Regression – Model Evaluation



Confusion Matrix

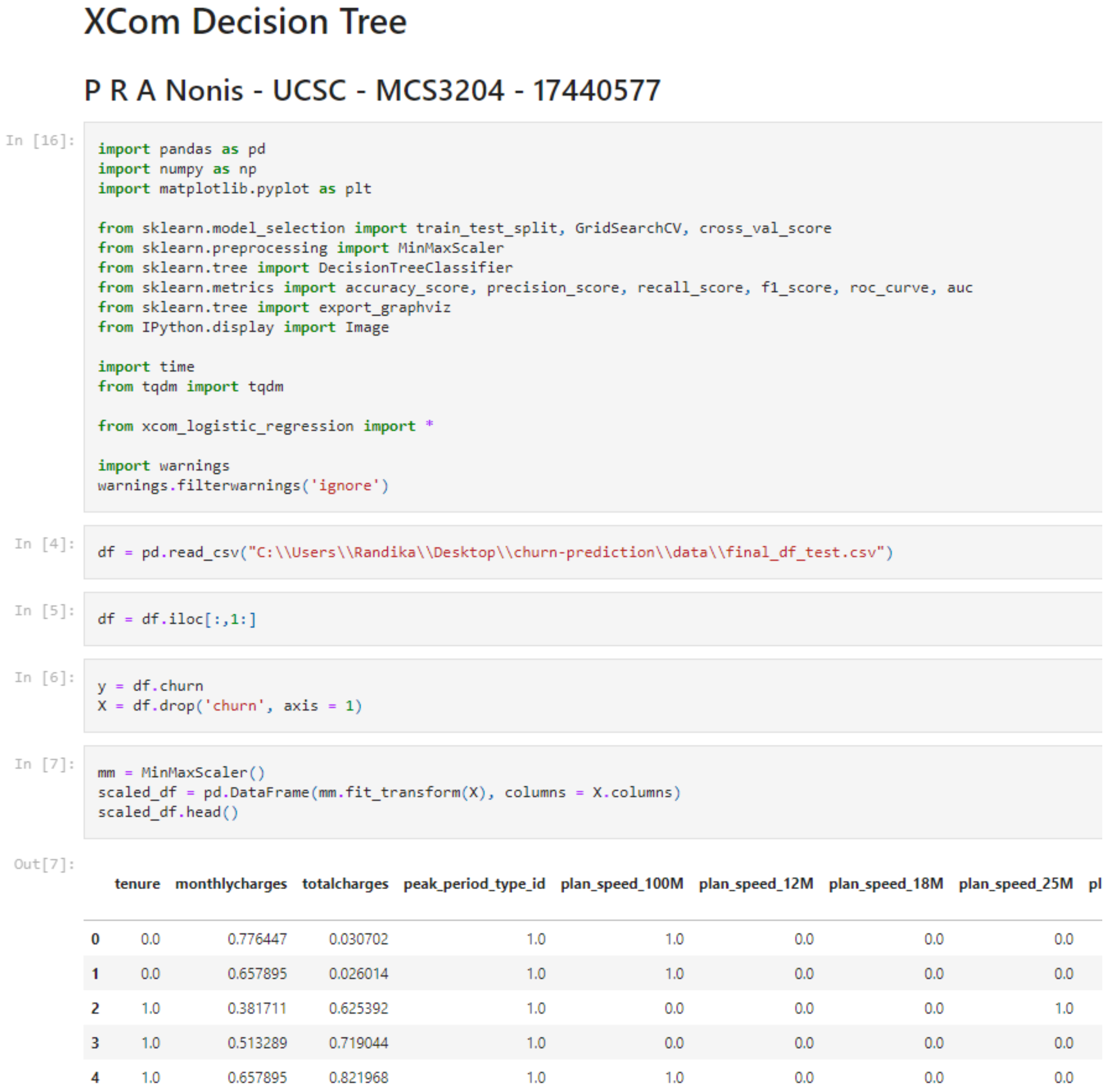
```
In [27]:
# Call confusion_matrix function from sklearn.metrics using actual y_test and predicted y_test data sets
cnf_matrix = confusion_matrix(y_test, y_hat_test)
print('Confusion Matrix: \n', cnf_matrix)
```

```
Confusion Matrix:
 [[316  13]
 [ 56 270]]
```

```
In [28]:
# Print 4 main logistic model metrics for training and test sets (Precision, Recall, Accuracy, F1)
print_metrics(y_train, y_hat_train, y_test, y_hat_test)
```

```
Training Metrics:
Training Precision:  0.94
Training Recall:  0.82
Training Accuracy:  0.89
Training F1-Score:  0.88


Testing Metrics:
Testing Precision:  0.95
Testing Recall:  0.83
Testing Accuracy:  0.89
Testing F1-Score:  0.89
```

Figure 59. Logistic Regression – Confusion Matrix



XCom Decision Tree

P R A Nonis - UCSC - MCS3204 - 17440577

In [16]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_curve, auc
from sklearn.tree import export_graphviz
from IPython.display import Image

import time
from tqdm import tqdm

from xcom_logistic_regression import *

import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
df = pd.read_csv("C:\\Users\\Randika\\Desktop\\churn-prediction\\data\\final_df_test.csv")
```

In [5]:

```
df = df.iloc[:,1:]
```

In [6]:

```
y = df.churn
X = df.drop('churn', axis = 1)
```

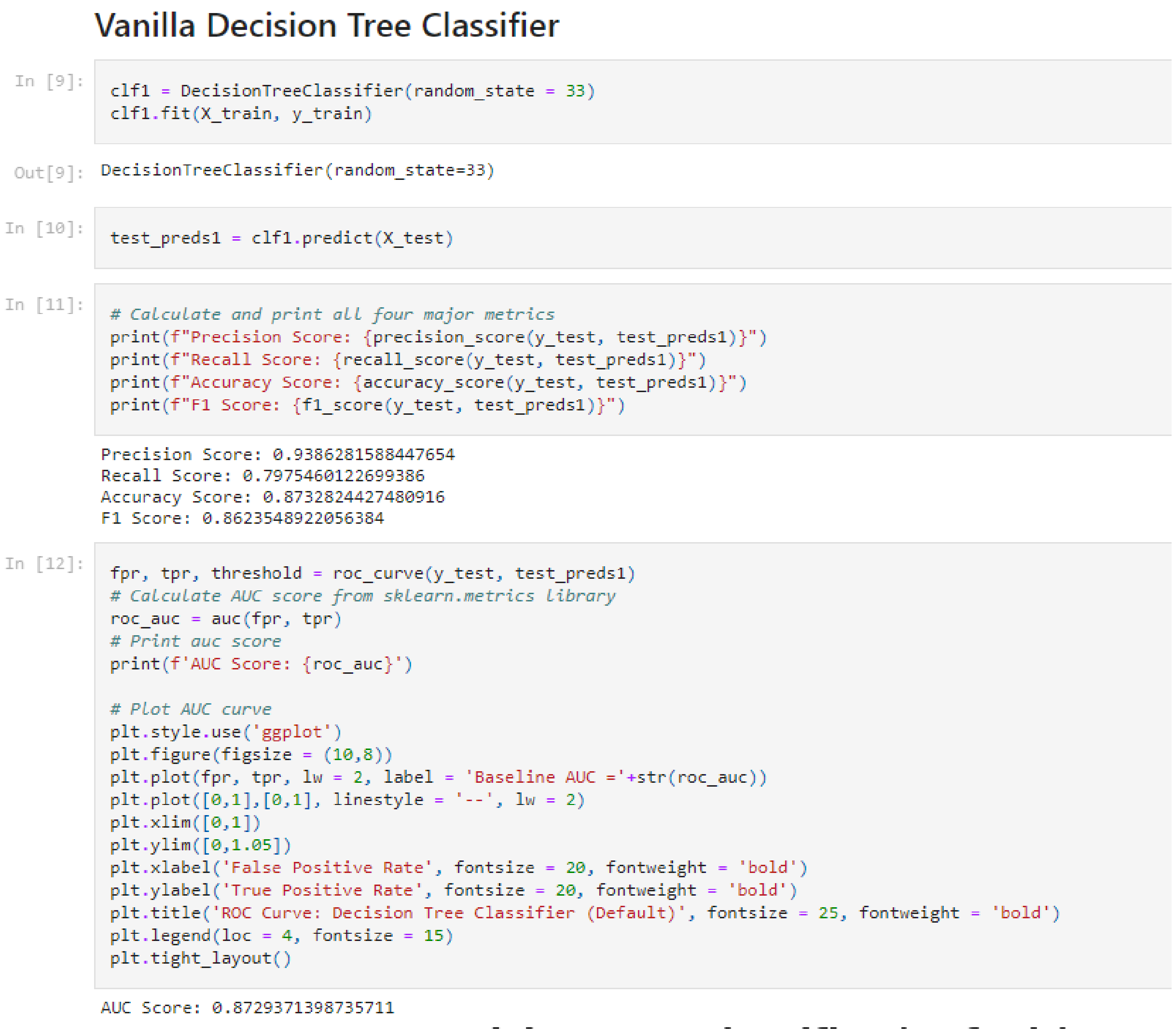
In [7]:

```
mm = MinMaxScaler()
scaled_df = pd.DataFrame(mm.fit_transform(X), columns = X.columns)
scaled_df.head()
```

Out[7]:

	tenure	monthlycharges	totalcharges	peak_period_type_id	plan_speed_100M	plan_speed_12M	plan_speed_18M	plan_speed_25M	pl
0	0.0	0.776447	0.030702	1.0	1.0	0.0	0.0	0.0	
1	0.0	0.657895	0.026014	1.0	1.0	0.0	0.0	0.0	
2	1.0	0.381711	0.625392	1.0	0.0	0.0	0.0	1.0	
3	1.0	0.513289	0.719044	1.0	0.0	0.0	0.0	0.0	
4	1.0	0.657895	0.821968	1.0	1.0	0.0	0.0	0.0	

Figure 60. Decision Tree



Vanilla Decision Tree Classifier

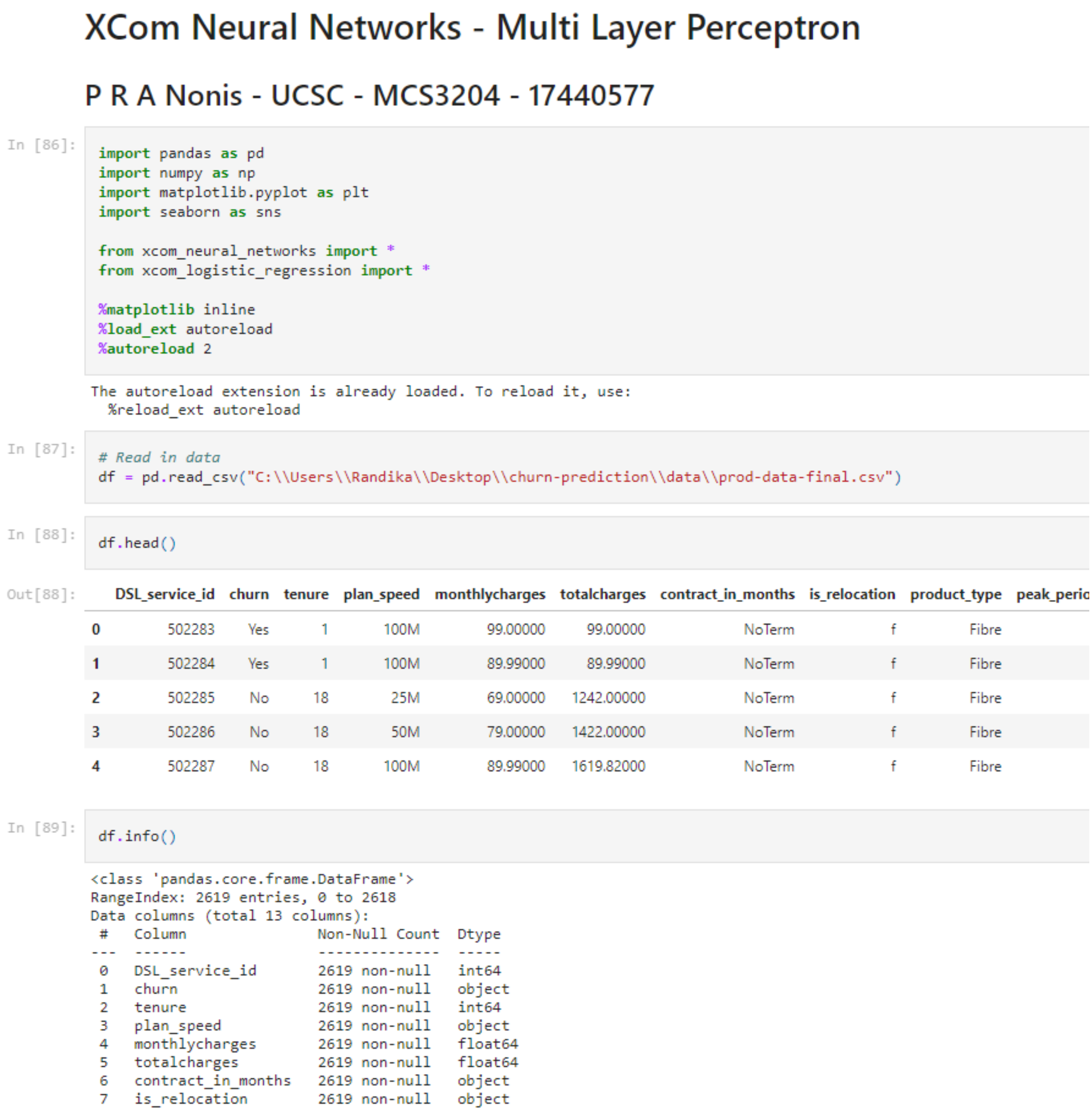
```
In [9]:  clf1 = DecisionTreeClassifier(random_state = 33)
          clf1.fit(X_train, y_train)

Out[9]:  DecisionTreeClassifier(random_state=33)
```

```
In [10]:  test_preds1 = clf1.predict(X_test)
```

```
In [11]:  # Calculate and print all four major metrics
           print(f"Precision Score: {precision_score(y_test, test_preds1)}")
           print(f"Recall Score: {recall_score(y_test, test_preds1)}")
           print(f"Accuracy Score: {accuracy_score(y_test, test_preds1)}")
           print(f"F1 Score: {f1_score(y_test, test_preds1)}")

           Precision Score: 0.9386281588447654
           Recall Score: 0.7975460122699386
           Accuracy Score: 0.8732824427480916
           F1 Score: 0.8623548922056384
```

```
In [12]:  fpr, tpr, threshold = roc_curve(y_test, test_preds1)
           # Calculate AUC score from sklearn.metrics library
           roc_auc = auc(fpr, tpr)
           # Print auc score
           print(f'AUC Score: {roc_auc}')

           # Plot AUC curve
           plt.style.use('ggplot')
           plt.figure(figsize = (10,8))
           plt.plot(fpr, tpr, lw = 2, label = 'Baseline AUC ='+str(roc_auc))
           plt.plot([0,1],[0,1], linestyle = '--', lw = 2)
           plt.xlim([0,1])
           plt.ylim([0,1.05])
           plt.xlabel('False Positive Rate', fontsize = 20, fontweight = 'bold')
           plt.ylabel('True Positive Rate', fontsize = 20, fontweight = 'bold')
           plt.title('ROC Curve: Decision Tree Classifier (Default)', fontsize = 25, fontweight = 'bold')
           plt.legend(loc = 4, fontsize = 15)
           plt.tight_layout()

           AUC Score: 0.8729371398735711
```

Figure 61. Decision Tree – Model Evaluation



XCom Neural Networks - Multi Layer Perceptron

P R A Nonis - UCSC - MCS3204 - 17440577

In [86]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from xcom_neural_networks import *
from xcom_logistic_regression import *

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [87]:

```
# Read in data
df = pd.read_csv("C:\\Users\\Randika\\Desktop\\churn-prediction\\data\\prod-data-final.csv")
```

In [88]:

```
df.head()
```

Out[88]:

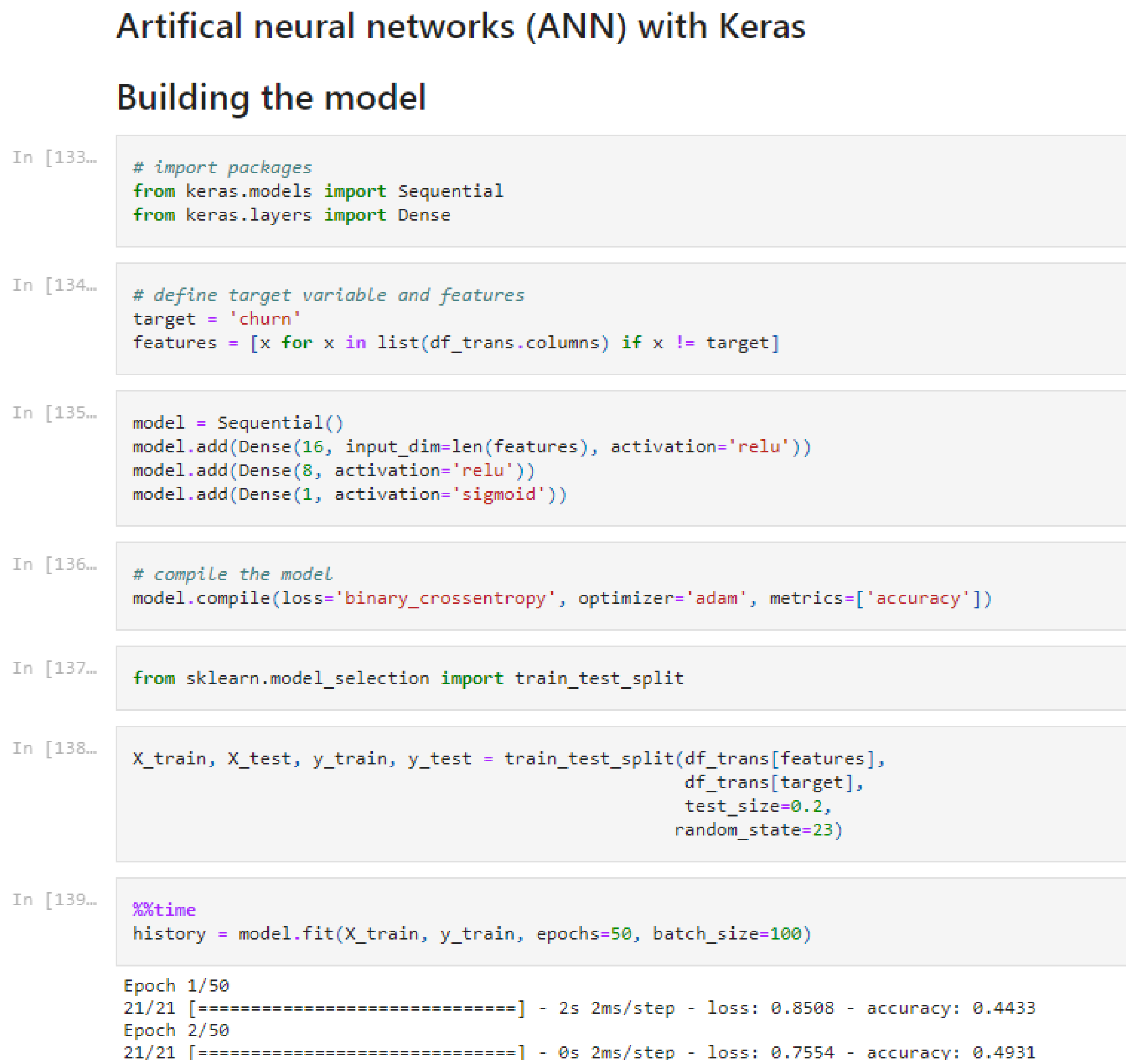
	DSL_service_id	churn	tenure	plan_speed	monthlycharges	totalcharges	contract_in_months	is_relocation	product_type	peak_perio
0	502283	Yes	1	100M	99.00000	99.00000	NoTerm	f	Fibre	
1	502284	Yes	1	100M	89.99000	89.99000	NoTerm	f	Fibre	
2	502285	No	18	25M	69.00000	1242.00000	NoTerm	f	Fibre	
3	502286	No	18	50M	79.00000	1422.00000	NoTerm	f	Fibre	
4	502287	No	18	100M	89.99000	1619.82000	NoTerm	f	Fibre	

In [89]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2619 entries, 0 to 2618
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   DSL_service_id      2619 non-null   int64  
 1   churn               2619 non-null   object 
 2   tenure              2619 non-null   int64  
 3   plan_speed          2619 non-null   object 
 4   monthlycharges      2619 non-null   float64
 5   totalcharges        2619 non-null   float64
 6   contract_in_months  2619 non-null   object 
 7   is_relocation       2619 non-null   object 
```

Figure 62. Multi-Layer Perceptron



Artifical neural networks (ANN) with Keras

Building the model

In [133…

```
# import packages
from keras.models import Sequential
from keras.layers import Dense
```

In [134…

```
# define target variable and features
target = 'churn'
features = [x for x in list(df_trans.columns) if x != target]
```

In [135…

```
model = Sequential()
model.add(Dense(16, input_dim=len(features), activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

In [136…

```
# compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [137…

```
from sklearn.model_selection import train_test_split
```

In [138…

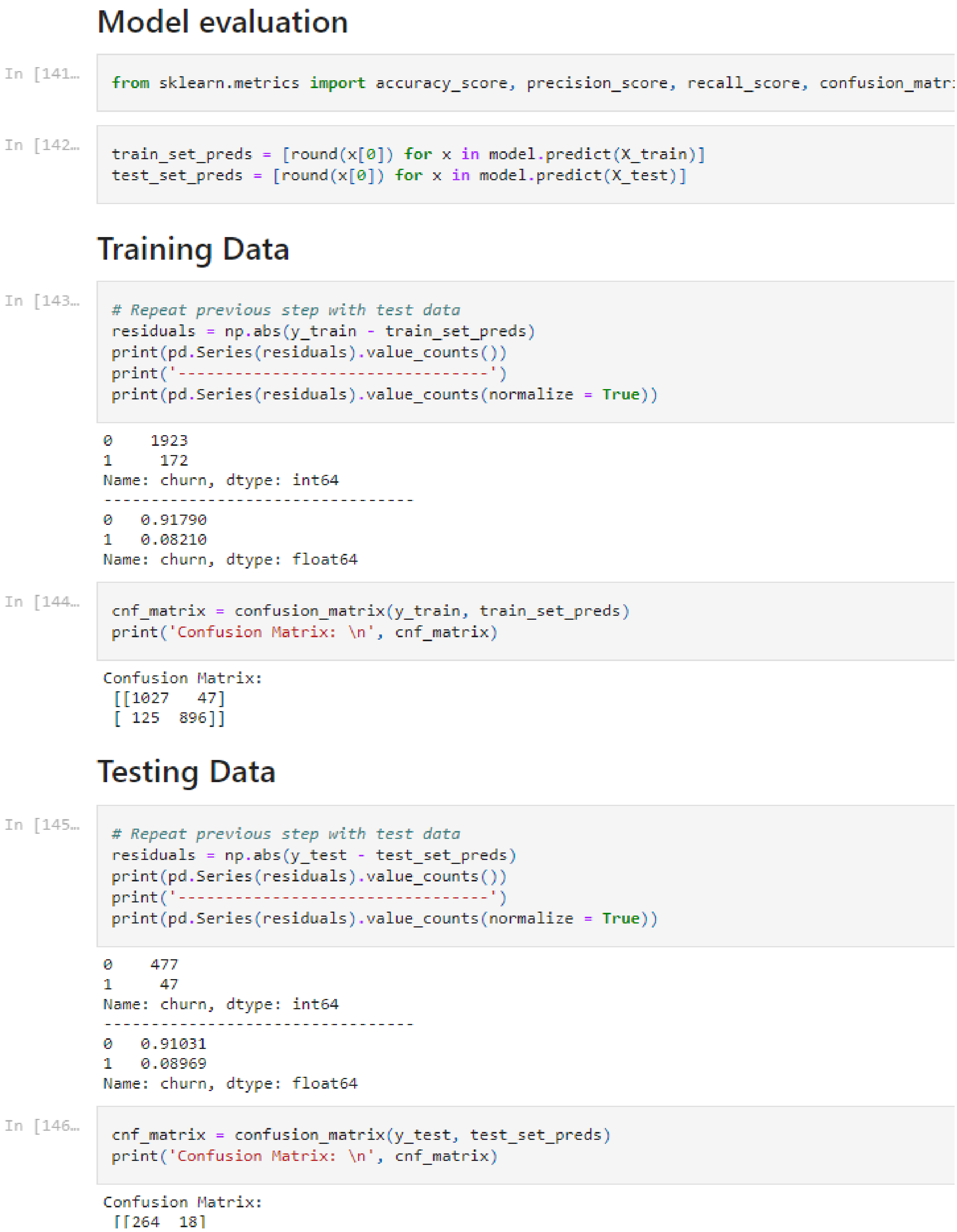
```
X_train, X_test, y_train, y_test = train_test_split(df_trans[features],
                                                    df_trans[target],
                                                    test_size=0.2,
                                                   random_state=23)
```

In [139…

```
%%time
history = model.fit(X_train, y_train, epochs=50, batch_size=100)
```

```
Epoch 1/50
21/21 [==============================] - 2s 2ms/step - loss: 0.8508 - accuracy: 0.4433
Epoch 2/50
21/21 [==============================] - 0s 2ms/step - loss: 0.7554 - accuracy: 0.4931
```

Figure 63. Multi-Layer Perceptron – Model Building



Model evaluation

```
In [141…  from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matr:
```

```
In [142…  train_set_preds = [round(x[0]) for x in model.predict(X_train)]
          test_set_preds = [round(x[0]) for x in model.predict(X_test)]
```

Training Data

```
In [143…  # Repeat previous step with test data
          residuals = np.abs(y_train - train_set_preds)
          print(pd.Series(residuals).value_counts())
          print('----------------------------------')
          print(pd.Series(residuals).value_counts(normalize = True))
```

```
0    1923
1     172
Name: churn, dtype: int64
----------------------------------
0   0.91790
1   0.08210
Name: churn, dtype: float64
```

```
In [144…  cnf_matrix = confusion_matrix(y_train, train_set_preds)
          print('Confusion Matrix: \n', cnf_matrix)
```

```
Confusion Matrix:
 [[1027   47]
 [ 125  896]]
```

Testing Data

```
In [145…  # Repeat previous step with test data
          residuals = np.abs(y_test - test_set_preds)
          print(pd.Series(residuals).value_counts())
          print('----------------------------------')
          print(pd.Series(residuals).value_counts(normalize = True))
```

```
0    477
1     47
Name: churn, dtype: int64
----------------------------------
0   0.91031
1   0.08969
Name: churn, dtype: float64
```

```
In [146…  cnf_matrix = confusion_matrix(y_test, test_set_preds)
          print('Confusion Matrix: \n', cnf_matrix)
```

```
Confusion Matrix:
 [[264   18]
```

Figure 64. Multi-Layer Perceptron – Model Evaluation

REFERENCES

[1] Khan, A., Jamwal, S. and Sepehri, M., 2010. Applying Data Mining to Customer Churn Prediction in an Internet Service Provider. International Journal of Computer Applications, 9(7).

[2] Lazarov, V. and Capota, M., n.d. Churn Prediction. Technische Universität München.

[3] Umayaparvathi, V., Iyakutti, K., 2012. Applications of Data Mining Techniques in Telecom Churn Prediction. International Journal of Computer Applications 42, 5–9. https://doi.org/10.5120/5814-8122

[4] Karahoca, A., KARA, A., 2006. Comparing Clustering Techniques for Telecom Churn Management.

[5] Abbasimehr, H., Mostafa, S., Tarokh, M.J., 2011. A Neuro-Fuzzy Classifier for Customer Churn Prediction. International Journal of Computer Applications 19, 0975–8887.

[6] Brandusoiu, I., Toderean, G., 2013. Churn Prediction in the Telecommunications Sector Using Support Vector Machines. Presented at the ANNALS OF THE ORADEA UNIVERSITY. Fascicle of Management and Technological Engineering. https://doi.org/10.15660/AUOFMTE.2013-1.2772

[7] YANG, L.-S., Chiu, C., 2006. Knowledge Discovery on Customer Churn Prediction.

[8] Jadhav, R., Pawar, U., 2011. Churn Prediction in Telecommunication Using Data Mining Technology. International Journal of Advanced Computer Sciences and Applications 2. https://doi.org/10.14569/IJACSA.2011.020204

[9] Torres, Renato & Ohashi, Orlando & Pessin, Gustavo. (2019). A Machine-Learning Approach to Distinguish Passengers and Drivers Reading While Driving. Sensors. 19. 3174. 10.3390/s19143174.

[10] F., S., 2018. Machine-Learning Techniques for Customer Retention: A Comparative Study. International Journal of Advanced Computer Science and Applications, 9(2).

[11] https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[12] Medium. 2021. *Predict Customer Churn with Neural Network*. [online] Available at: <https://towardsdatascience.com/predict-customer-churn-with-neural-network-1ef8f1a1c6ab> [Accessed 12 March 2021].

[13] En.wikipedia.org. 2021. *Anaconda (Python distribution) - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)> [Accessed 7 May 2021].

[14] Medium. 2021. *XGBoost Algorithm: Long May She Reign!*. [online] Available at: <https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d> [Accessed 20 March 2021].

[15] En.wikipedia.org. 2021. *Gradient boosting - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Gradient_boosting> [Accessed 23 April 2021].

[16] Wang, Yuanchao & Pan, Z. & Zheng, J. & Qian, L. & Mingtao, Li. (2019). A hybrid ensemble method for pulsar candidate classification. Astrophysics and Space Science. 364. 10.1007/s10509-019-3602-4.

[17] Medium. 2021. *What is Exploratory Data Analysis?*. [online] Available at: <https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15> [Accessed 24 February 2021].

[18] YANG, L. and CHIU, C., 2006. Knowledge Discovery on Customer Churn Prediction. *Proceedings of the 10th WSEAS Interbational Conference on APPLIED MATHEMATICS, Dallas, Texas, USA, November 1-3,* pp.523-528.

[19] Bell, D. and Mgbemena, C., 2017. Data-driven agent-based exploration of customer behavior. *SIMULATION*, 94(3), pp.195-212.