Early Prediction of Customer Abandonment in E-Business

A Thesis Submitted for the Degree of Master of Business Analytics

W.R.G.S.M. Weerasinghe

University of Colombo School of Computing

2021

I would like to dedicate this thesis to my supervisor, teachers and family.

ABSTRACT

Customer churn, customer retention and attrition are topics which have been discussed and studied in so many researches. However, most of them have evaluated the churn prediction only based on the history data. Here in this research conduct, it is intended to derive results based on the customer segmentation. Considering the whole customer base might deliver the desirable output yet, when applying the customer segmentation based on the primary customer types, the final outcome will be more precise. Considering the customer type such as, loyalty and seasonal customers who really gives a better return on investment for the business conduced more customer churn and retention evaluation.

By referring the E-Business dataset specifically with the access mode and channels of the customers, the tendency of leaving the system with the possible way is to be evaluated. Getting into the term ‘Churn’, this study has aware of the nature of the detachment considering the access channel or method of the customer as well.

The suggested methodology would be approaching in two different paths in order to evaluate the best fit and the performance by measuring the accuracy and the reusability of the model. Aligning with two methods namely logistic regression and deep neural network address artificial neural network, the predictive model would be implemented. Specifically with the use of RFM Analysis, the study will be directed to customer segmentation. The segmentation would be involved in clustering for the customer segmentation and as for the predictive model, classification model would be used with class variable for identification of the user churn.

Here in this study, main focus would be designing and building predictive models in different aspects under the similar criteria and mostly the evaluating the usefulness of segmentation of the customer over the total customer base that has been dedicated under specific clusters identified with the given set of criteria.

ACKNOWLEDGEMENTS

Introducing the new academic course “Master of Business Analytics” for the postgraduate degrees, University of Colombo School of Computing has open up another path for the students with novel experience. I would sincerely offer my gratitude to my institute UCSC, for granting this valuable opportunity by letting me follow this course. From the very beginning of this research module, I have been instructed, supervised and well mentored and monitored by my supervisor Dr Kasun Karunanayake and I must be much grateful for his proper guidance and the motivation.

I would also like to be thankful for my coordinators Dr T. Halloluwa and Dr M. Wickramasinghe for their guidance and helping hand all the time whenever necessary throughout the course. Last but not least, I should be sincerely thankful for my family and friends specially who have helped in various way to get this study completed successfully.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CLV	– Customer Lifetime Value
RFM	– Recency Frequency Monetary
EDA	– Exploratory Data Analysis
CART	– Classification and Regression Trees
SOM	– Self-Organizing Map
SMOTE	– Synthetic Minority-Over-sampling Technique
KNN	– K -Nearest Neighbour
OLS	– Ordinary Least Squares
MLE	– Maximum Likelihood Estimation
ANN	– Artificial Neural Network
AUC	– Area Under the Curve
ROC	– Receiver Operating Characteristics
API	– Application Programming Interface

CHAPTER 1
INTRODUCTION

Customer churn is the tendency of customers to stop doing business with a company in a given time period (Hossein Abbasimehra, Mustafa Setakb, M . J. Tarokhc, 2012). “Considering the level of competition prevailing in Business-to-Consumer (B2C) E-Commerce domain and the huge investments required to attract new customers, firms are now giving more focus to reduce their customer churn rate. Churn rate is the ratio of customers who part away with the firm in a specific time period” (Renjith, September 2015). This could be a permanent or temporary discontinuation of the transaction or interaction with the E-commerce system.

The terms such as ‘Customer Churn’, ‘Customer Retention’ and ‘Customer Attrition’ have drawn the attention most of the business and economic experts irrespective to the domain or the channel of the businesses. Along with the agile popularity of virtual business platforms, E-Business has become one main type of approachable channels even without having a physical implementation of the business, in real.

Continuous monitoring of this type of a business reveals valuable information that applicable for many use cases. Without limiting to the customer churn, identifying patterns and associations between favourable features and attributes direct to realize more informative consummation at the end.

In this research, the main intention is to evaluate the feasibility of generating and using a model for early identification and predicting the abandonment of the customers in E-Businesses platforms based on the customer segmentation.

1.1 Motivation

Most of the researchers have experimented on the area of ‘Customer Churn’ and interesting outcomes have been recorded so far in different domains by addressing different knowledge areas. However, there can be feasible solutions that could be derived from the same dilemma if think out of the box without following the same steps and procedures which have been already observed. Hence, suggesting an alternate solution which deviates from the current approaches will allow to study and generate new models. This innovative alternative has become the most influential motivational factor caused for conducting this research.

With a deviation from the usual way of customer churn prediction, finding out feasible and solution by assessing the generated models and patterns from available data will reveal new insights and study areas as well.

1.2 Statement of the problem

The existing studies have been conducted related to “Customer Churn” have mainly focused on the previous churn rate of the customers and based on those readings their conclusions have been made. In most of the studies, the customer type and detachment type have not taken into consideration when modelling or analysing the result. Also, if the nature and the type of the detachment could be identified beforehand, the flow of the business would be determined according to the need of the business.

When the concept “Customer Churn” is taken into consideration, finding answers to two main questions will enable better and deeper insights in this research, namely:

- **What is the type of the discontinuation or detachment of the customer with the system?**
  This can be occurred in many forms such as unsubscribing, cancellation of registration or membership, uninstalling the app or being inactive for the rest of the time.

- **Whether the act of inactiveness or detachment is temporary or permanent?** This is fact to be considered is much tricky and concerned based on the customer behaviour.

Considering mainly the above mentioned two factors, which have not discussed or covered in the bare context of “Customer Churn”, this research is intended to find the possible evaluations for predicting the early detection of the system abandonment by the customers in E-business platforms with customer segmentation rather predicting the customer churn.

The customer segmentation basically considers the customer types identified by the previous study. According to the blog (Anon., n.d.) on business and customer relationships management has specifically identified about these customer types by their behaviours. By combining this customer type with the above discussed detachment nature of the customer, the prediction model is to be trained and tested.

1.3 Research Aims and Objectives

1.3.1 Aim

Customer churn and the customer retention is the most challenging factors when continuing a business physically or virtually. Though, many of the researches related to 'Customer churn' have been conducted throughout the time period so far, in order to make it more precise and productive, it is wise to look for an option to predict and identify the possible loss and turnover beforehand within a specific time period.

Therefore, in this research, the primary objective is to evaluate the feasibility of using a machine learning approach for early identification of the customers who are going to abandon the business in an E-commerce system based on the historical data of customer behaviour, the type of the customer and the nature of the abandonment. Most of the previous studies have not critically addressed these two aspects in a combined manner instead the focus has been taken either of them. Here upon, the proposed solution critically considers these two aspects together for generating a highly relatable model.

1.3.2 Objectives

With the help of below mentioned secondary objectives, the model is going to be trained and used for predicting and identifying the customers with their related value-added feature attributes who are going to leave out the system in advance instead evaluating only the churning rate in general.

- To evaluate the behavioural factors of the customers (Comment/Review/Ratings)
- To discover and identify the nature of the abandonment
- To determine the suitable models for prediction of abandonment
- To identify the suitable transformation strategies for the dataset selected
- To determine the criteria of evaluation of the success of prediction
- To determine the limitations and constraints of the models identified

1.4 Scope

The target domain area is E-Business and the customer base with their behaviours. There are several types of customers that are intended to be classified according to their purchasing behaviours in the system as follows.

- **Loyal customers**: Customers that make up a minority of the customer base but generate a large portion of sales.
- **Impulse customers**: Customers that do not have a specific product in mind and purchase goods when it seems good at the time.
- **Discount customers**: Customers that shop frequently but base buying decisions primarily on markdowns.
- **Need-based customers**: Customers with the intention of buying a specific product.
- **Wandering customers**: Customers that are not sure of what they want to buy. (Anon., n.d.)

Above mentioned categorization should be considered whenever an analysis is performed since it will lead to misclassification otherwise. For instance, one-time customer is a profit for the business from the monetary values while it implies an inactive or never returning customer based on the visiting counts. Therefore, identifying the type of the customer is a major factor in this topic.

Also, the detachment or the discontinuation of the customer can be implied in several ways. Either it could be a permanent or temporary act of the customer which can be one of the proposed categorisations mentioned below.

- Unsubscribing from the system
- Cancellation of the registration or membership
- Inactive or idle act with the system
- Uninstallation of the app

Evaluation of the relevant features with regards to above mentioned factors is the underlying logic of this research. The major aspects which are going to be addressed based on the given above criteria are as follows:

- The early detection of customer abandonment of an E-commerce system
- Identify the behavioural factors of the customer that is affected to turn over
- Evaluate the probable time period that a correct prediction can be made
- Detecting the churning customers based on the user type in terms of E-commerce

Even though the data related to E-Businesses and E-Commerce platforms, are available in different sources, there are some problems that have been identified.

- No enough favourable attributes per dataset

- Not enough volume of data
- Higher cost of real customer data

In order to get a complete collection of insights, it is decided to generate the dataset randomly by studying the available real datasets. The main reasons for generating a random dataset have been discussed in detail in the methodology chapter.

1.5 Structure of the Thesis

The structure of this thesis is as mentioned below.

Chapter 2: Literature Review presents a previous related works on customer churn, attrition and retention analysis and applications and discuss the different types of models used so far.

Chapter 3: Methodology discuss about the proposed solution, the methods and models along with the selected tools, techniques and technology.

Chapter 4: Evaluation of the paper will critically evaluate the results of the study.

Chapter 5: Discussion will contain the important significant milestones throughout the process

Chapter 6: Conclusion will conclude the work, including the summary of the study.

CHAPTER 2

LITERATURE REVIEW

Customer Relations Management has two major aspects, one being the operational aspect and the other being the technical aspect. This technical aspect is also known as Customer Analytics. Customer Analytics can be broken into two major categories.

1. Descriptive Analytics: Customer Identification
2. Predictive Analytics: Customer Retention (Damith Senanayake, Lakmal Muthugama, Laksheen Mendis, Tiroshan Madushanka, 2015)

With the purpose of making prediction based on the customer data and their behaviour, the appropriate analytical approach for this research is predictive analytics. According to the previous researches, it has been emphasized that the customer churning is much ware of the customer retaining. In this study, it is contemplated to use nearly one million records of data which will be manipulated by automated analytical technique.

Moving to the online virtual platforms has become a trend in most of the production and service sectors nowadays. It is facilitated the almost needful with knowledge, systems, equipment and technology freely and effectively. When it comes to any business performing online, new customer gathering and retaining becomes easier when the business earns the loyalty of the customers. However, there is a high possibility of leaving out of any customer which could be a permanent lose for the business at any point which can be explained by the Customer Lifetime Value (CLV).

CLV is the value makes by a customer for the business in his total period of time with the business starting from his initial purchase. The reason why CLV is so important in terms of 'Customer Churn' is it costs the lesser value to retain the current customers than procuring new customers. Even though it is concerned more about the retention in CLV, Muzaffar Shah, Darshan Adiga, Shabir Bhat and Viveka Vyeth have mentioned in their research as 'while acquiring new customers is the backbone of the business growth but marketing experts suggest that equal importance should be given to retention policies' (Muzaffar Shah, Darshan Adiga, Shabir Bhat and Viveka Vyeth, n.d.), which is appeared to be contradictive to CLV.

Customer churn is the tendency of customers to stop doing business with a company in a given time period (Hossein Abbasimehra, Mustafa Setakb, M. J. Tarokhc, 2012). "Considering the level of competition prevailing in Business-to-Consumer (B2C) E-Commerce domain and the huge investments required to attract new customers, firms are now giving more focus to reduce their customer churn rate. Churn rate is the ratio of customers who part away with the firm in a specific time period" (Renjith, September 2015). This could be a permanent or temporary discontinuation of the transaction or interaction with the E commerce system.

2.1 A Literature Review

Customer churn analysis and prediction is a mostly discussed topic in various fields such as telecommunication and e-Business. Those researches and findings related to this topic is mainly considered the following domains.

- Telecommunication – Mobile users of different vendors
- B2C – Online and off line wholesale and retail businesses

Almost every study has based on the customer behaviour with their transactions. Every E-Commerce business house understands that only a part of their customers sticks with them, while the rest stop shopping after one or two transactions. According to a survey conducted by the Rockefeller Foundation, majority of customers mentioned that they move on from sellers as they are not being cared (Renjith, September 2015).

When referring time series data for predictive analysis of customer churn, it is challenging and somewhat biased if it is seasonal data. As Bryan Gregory mentioned in his paper, accurately predicting customer churn using large scale time-series data is a common problem facing many business domains. The creation of model features across various time windows for training and testing can be particularly challenging due to temporal issues common to time-series data (Gregory, 2018).

Even though Bryan Gregory has used Extreme Gradient Boosting method for testing, when there is any type of missing data, the model is having the issue of fitting well. In order to overcome this issue, the data set should be highly accurate. As per the data set of one-million records, evaluating for missing values will take unnecessary overhead over this type of algorithm and approach.

Moreover, the gathered data should be cleansed, processed, transform and integrated in order to proceed with modelling. In other words, we have to undergo with data mining phase accordingly. This will help with revealing important patterns and relationships among data attributes. As Vivek Bhambri has well explained in his paper about data mining with regards to customer churn in detail. The customer churn is becoming a major area of concern for the industry now a days. Identifying the churn beforehand and taking necessary steps to retain the customers would increase the overall profitability of the organization. Losing customers not only leads to opportunity lost because of reduced sales, but also to an increased need for attracting new customers, which is five to six times more expensive than customer retention (Bhambri, 2012).

When considering the customers' purchase pattern behaviour, there are four types of business settings have been identified in the previous studies.

- **Contractual and discrete business**: - In these types of businesses, purchase interval and sale amount both are fixed. They are also called the subscription-based business. Example: - Post-paid telecommunication, Netflix
- **Contractual and continuous business**: - Where purchase interval is fixed but the sale amount can vary. Example: - Credit card businesses
- **Non-contractual and discrete business**: - When the purchase interval can vary but the sale amount is fixed. Example: - Oil and Gas retail businesses
- **Non-contractual and continuous business:** - Where both sale amount and purchase interval can vary. Example: - Grocery stores, Retailer businesses

Further, each of these business categories can be classified as business-to-business or business to-consumer. However, the methods proposed in this work apply to both of these subcategories of businesses. Most of the churn prediction systems until now have been proposed for Contractual and discrete and Contractual and continuous types of businesses (Muzaffar Shah, Darshan Adiga, Shabir Bhat and Viveka Vyeth, n.d.). As per the above categorization, this research is referring to the Non-contractual and continuous business type.

Furthermore, in order to identify the influential relational patterns of the attributes, the data should undergo a thorough mining process after the data is cleansed and extracted accordingly. The importance of the data attributes in mining process in terms of customer churn, has been mentioned in the previous studies as follows irrespective the industry or field.

2.1.1 RFM Analysis

In RFM analysis, customer data are classified by Recency (R), Frequency (F) and Monetary (M) variables. It has been noted that RFM enables the practitioners to observe customer behaviour, as well as to segment customers in order to determine immediate customer value. (Cormac Dullaghan and Eleni Rozaki, 2017)

This consideration of market segmentation is applicable despite of the nature or the channel of the business. Using market segmentation along with the customer segmentation will be more insightful for deriving associations. In this study, it is more intended to

As per the above discussed connotation, it implies the selection and extraction of relevant data attributes will lead to a process of data mining which results accurate and informative data for utilizing for the next step in the learning and modelling process. This can be put into action by Exploratory Data Analysis (EDA) process which will result the insights to differentiate attributes purposively.

2.1.2 Data Mining Techniques

The possibility of customer churn and importance of customer retention can be easily understood based on following reasons.**Invalid source specified.**

1) Some customers are cherry pickers (they visit multiple places to shop the basket of goods)
2) Some customers are store switchers
3) Some uses few retailers for their shopping
4) Retaining existing customers is advantageous than acquiring new ones
5) Loyal customers are advantageous as they increase spending amount (upselling and cross selling), spread positive word-of-mouth, less costly to serve, less prone to noise by competitors.

Because of the mentioned reasons there exist many machine learning approaches majorly focusing on customer Retention Analysis or Customer churn analysis. One-to-one marketing, loyalty programs and complaint management are classified as key elements in customer retention. (E. W. Ngai, L. Xiu, and D. C. Chau, 2009)

Even in the domain telecommunication, for customer profiling and market segmentation data mining has been applied specially for prediction model. It has come to attention that those prediction model has been used and getting more popular since this derives values for marketing strategies for customer retention. These studies highlight the importance of the customer segmentation followed by data mining.

This contrasts the validity and value of customer segmentation based on the generic behavioural factors. By narrowing down the whole data set into one common base by leaving out the unnecessary type of customers would make the results more valid and accurate. This approach has been suggested for this study possibly by classifying the data set to identified customer segments beforehand.

2.1.3 Data Preparation and Extraction

Before starting to implement the prediction approach, we conducted several unstructured interview sessions with a product owner, an operations engineer, and a data analyst of the platform's development teams. They provided us valuable insights on the available data, the platform provider's interactions and relationships with their customers as well as important events in the lifecycle of each customer. (Iris Figalist, Christoph Elsner, Jan Bosch, and Helena Holmstr¨om Olsson, 2020)

For conducting a proper study, the data preparation and the feature extraction play a critical role. Irrelevant data will lead the study to end up with incorrect information and an inefficient model which tries to solve and interact with garbage data. Moreover, there is a high tendency of misreading of the produced model when it is trained well yet only due to these unrealistic data it has headed to the other way around. Data is very critical since it directly influence on the accuracy and the performance of the model.

For determining the relevance of data and relevant data, different data mining techniques can be utilized, when having a huge amount of data volume like telecommunication and E-Commerce platforms. The different data mining techniques includes decision tree, association rules, neural networks, Rough Sets and Classification and Regression Trees (CART), Self-Organizing Map (SOM), fuzzy clustering, Classification, Regression, Rule generation, Sequence analyses, genetic algorithms, Forecasting Process. (Amjad Khani, Zahid Ansari, 2014)

When moving to data extraction, even some of the data are there in the data set or in a persisted source, some data need to be derived from the available data in order to get meaningful values. Deriving data from the direct data is important yet time and resource consuming specially for huge volume of data with associations and dependencies. Still, this would be critical if the available data se is not enough for getting insights.

2.2 Concerns on Dataset

There are aspects where considerable attentions that need to be paid in order to examine the data set fairly for all the scenarios possibly. Following mentioned are some of those concerns that arise regardless the nature or domain of the data set.

2.2.1 Imbalanced Dataset

A classification problem may be a little skewed, such as if there is a slight imbalance. Alternately, the classification problem may have a severe imbalance where there might be hundreds or thousands of examples in one class and tens of examples in another class for a given training dataset.

- **Slight Imbalance**. An imbalanced classification problem where the distribution of examples is uneven by a small amount in the training dataset (e.g. 4:6).
- **Severe Imbalance**. An imbalanced classification problem where the distribution of examples is uneven by a large amount in the training dataset (e.g. 1:100 or more). (Brownlee, 2019)

There are different methods of handling imbalanced data in order to make the more sensible and accurate analysis in machine learning.

a) Under Sampling Majority Class

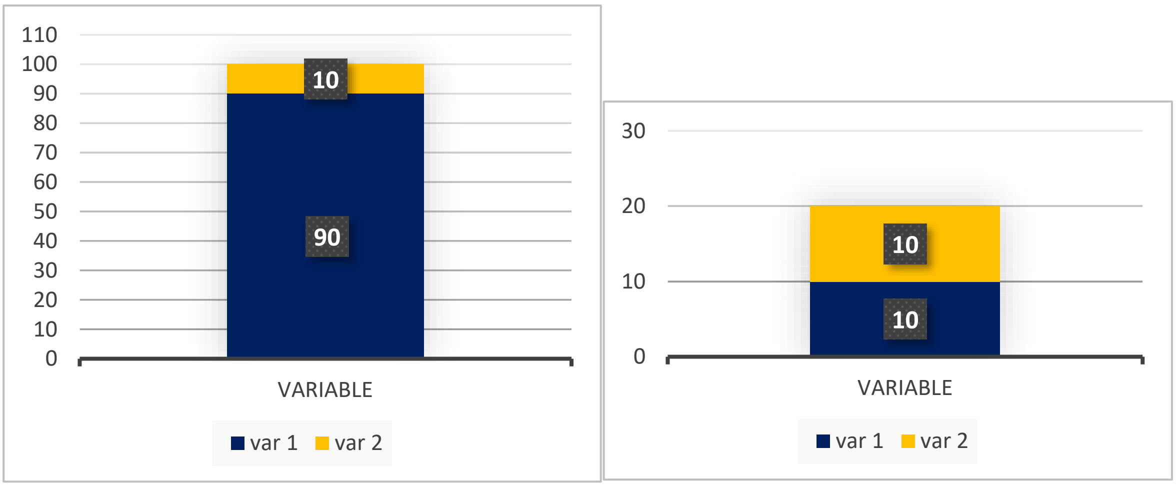


Figure 3:1 - Under Sampling Majority Class

Get a random sample from the majority class which is equal in amount to the minority class and discard all the other data from the majority sample. The randomly picked sample is combined with the minority class together and then this sample is going to be used for model training. Since, there are lot of data that has been neglected in this approach and as the above Figure 3:1 the total sample size is reduced, it does not consider as a better solution.

b) Over Sampling Minority Class by Duplication

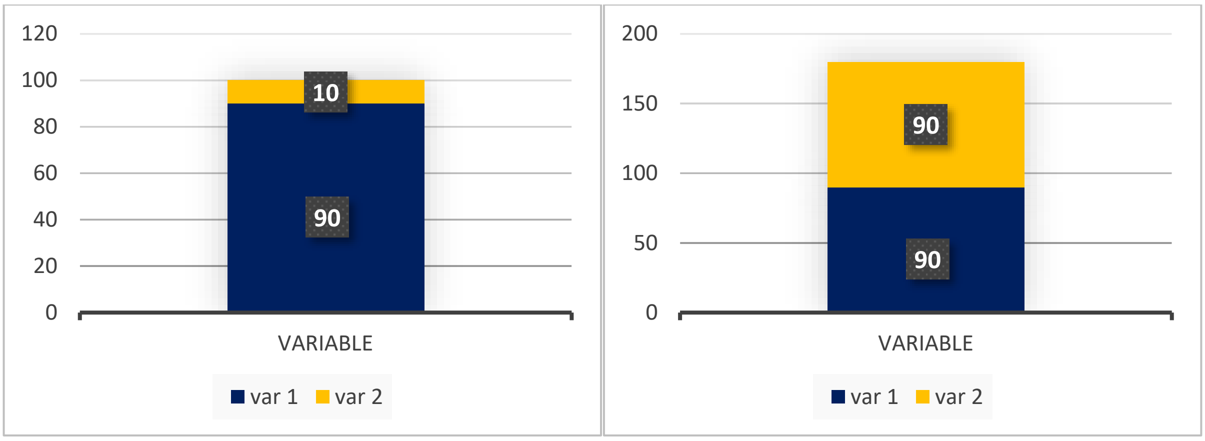


Figure 3:2 - Over Sampling Minority Class by Duplication

In this approach, as the above Figure 3:2, the minor sample is going to be duplicated by copying in order to be tallied with the majority sample class. This will generate new samples based on the current available sample and use the expanded sample for mode training. This is somewhat acceptable but this will replicate the data unnecessarily.

c) Over Sampling Minority Class by SMOTE

This method totally similar to the above mentioned yet the sampling approach is based on the K-Nearest Neighbor (KNN) algorithm. This produces synthetic samples using KNN algorithm by using minority sample class which elaborate SMOTE as Synthetic Minority-Over-sampling Technique.

d) Ensemble Method

In ensemble model, as the below mentioned Figure 3:3 the majority class is going to be divided into similar samples and they are combined with the minority sample batch wise. The resulting model is then going to be evaluated and get the majority vote which is similar to Random Forest sampling.

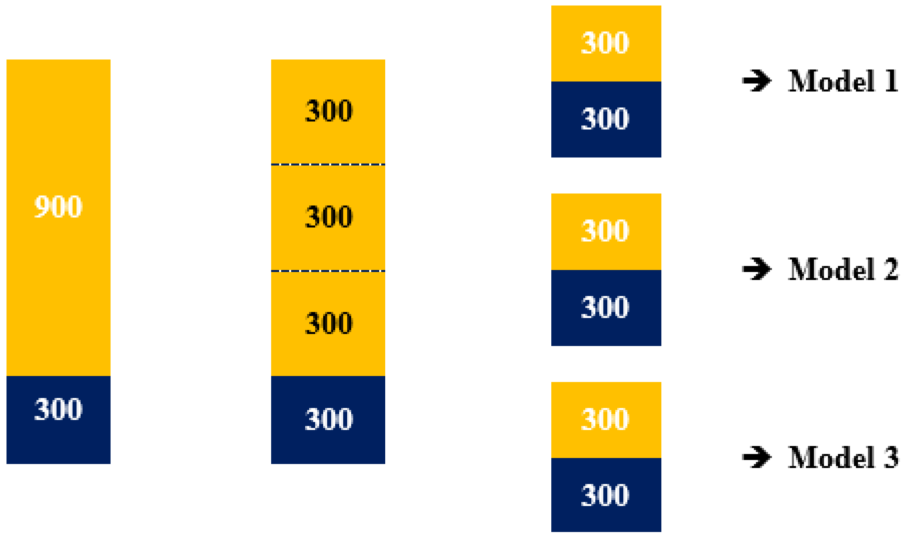


Figure 3:3 - Ensemble Method

e) Focal Loss

This method is spatial which is penalizing the majority class sample during loss calculation while giving more weight to the minority class sample

2.2.2 Feature Engineering

In constructing a set of behavioural metrics from the raw transactional data, we employ a data-centric approach that is intended to minimize the role of the analyst in determining which features are relevant to the task of modelling customer churn. (Muhammad Raza Khan, Joshua Manoj, Anikate Singh, Joshua Blumenstock, n.d.)

Even though there can be a huge set of attributes in a data set, it is possible to emerge a need of generating new features based on the existing features. Those features can be either derived from the existing or introduced with a logical explanation. Any of the studies may have this approach in order to handle and improve the usability of data.

2.3 Research Gap

Most of the previous work related to customer churn and prediction have used the real business data focusing on a specific area. Most of them have paid their attention for customer reviews, active engagement with the system and amount they spending on the product or services. The customer type's segmentation has not been considered thoroughly or at all for the aforementioned predictions.

The proposed prediction model is going to be trained and tested with the use of an E-Commerce dataset based which contains data collected over 5 years. The behaviour of the abandonment by a customer needs to be identified based on the customer segmentation, since there are different types of customers. For instance, if most of the data that have been collected are related to wandering customers, the business value which is going to be generated is less than the loyal customers. Even though it is open to use the data limitlessly, the particular readings should have to have a core business value that gives more accurate data feed as an input to the machine learning model.

Along with the customer segmentation, the intended models to be implemented as per the predictive models, will be evaluated against with under a same given criteria as with the performance of each model. Since the intended study will be covering the model designing, implementation, evaluation and conclusion thoroughly under different criteria, the scope that will be addressed is considerably broader.

CHAPTER 3

METHODOLOGY

The methodology chapter is dedicated for elaborating the suggested and carried out methodology including the in between major and minor processes, techniques and approaches. It has been attempted to describe all the steps starting from data preparation until the predictive model implementation.

3.1 Proposed Methodology

The suggested methodology has to be involved in phases of data mining, analytics and machine learning in order to generate the prediction model. This research is mainly aimed for the B2C domains which are sustaining their businesses virtually. Beginning from the customers' foot prints from signing up/signing in to sign out of the system data would be monitored and analysed for the research process.

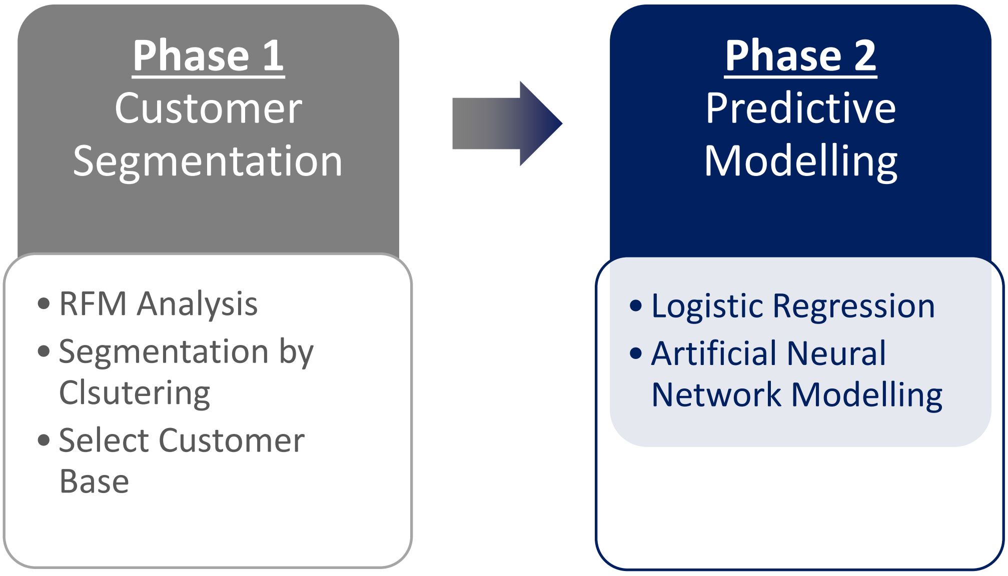


Figure 4:1 - Diagram of Proposed Methodology

The above Figure 4:1 proposed methodology is consisted with two phases as customer segmentation and predictive model building. The customer segmentation phase is directly involving with the data pre-processing, data transformation, feature engineering, feature selection and finally the RFM analysis.

This produces the valuable customer base as the input for the next phase of the study. Based on the selected customer segmentation data set, then the predictive model is supposed to be built and this study will utilize the logistic regression and the deep neural network approaches.

Finally, the evaluation model will be designed based on the selected two approaches and it will address the performance results of the study.

Selected Tools and Technology Stack

- Python 3
- MySQL : Data transformation, Type conversions, Data normalization
- Jupyter Notebook (Anaconda 3)
- Libraries: Pandas, Numpy, Matplotlib, ScikitLearn, Tensorflow

3.2 Dataset Selection and Preparation

Even though the E-Commerce and E-Business data are available persisted in most of the resources, the preferred needful of complete set of data for this study is not publicly available due to security concerns. Thus, by studying the available datasets, it has been selected the dataset of an international E-Commerce business, total volume of 50,000 unique records with the favourable attributes.

Data repository:
https://github.com/000407/customer_churn_ds/blob/main/data/customer_churn.csv

Though, it has not contained all of the needful feature attributes together as a single dataset, with the available data attributes the favourable attributes have been generated with logical assumptions. Hence, according to the customer type segmentation the relevant data is generated with the detailed attributes which can be found in a real E-Business database.

As the major aspects of data related to the prediction model has been categorized as follows.

Data Category	Composition
Customer information	Customer type and status of the availability (churned, not churned)
Sales/Order data	Order or purchases related data along with the timestamp, order count and order hike amount
Reviews/Comments	Feedback (satisfaction score) on products that have been purchased
Access logs data	Frequency, number of devices used, last access, recency of the customer’s visit

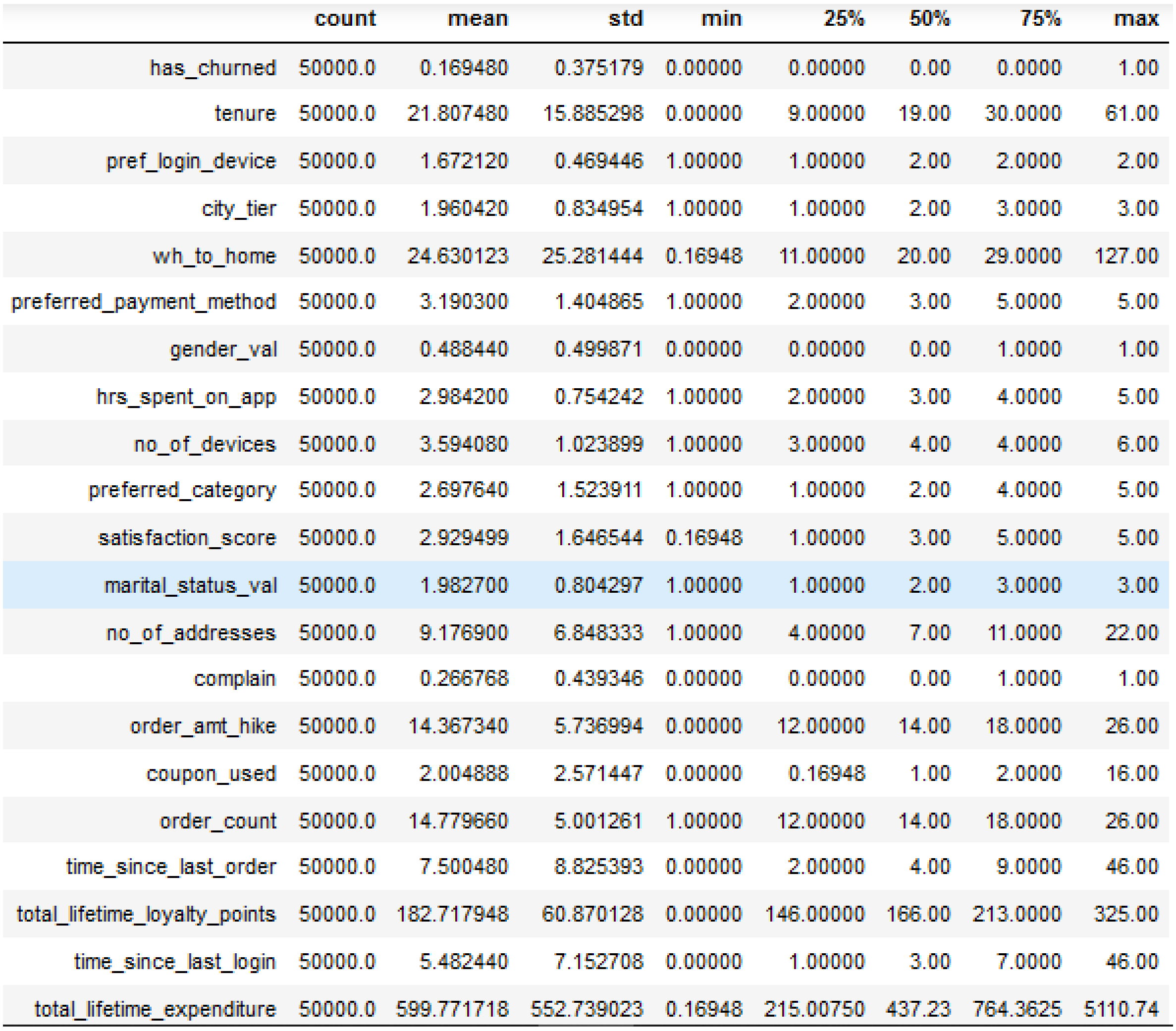
Table 4:1 - Customer data categorization

3.2.1 Data pre-processing

For pre-processing the data, it is advisable to have an overview and overall understand about the whole collection. Descriptive statistics and summary statistics are very basic yet powerful summarization of the collated data. For this proposed dataset, after deriving the summary statistics it provides quite understood dataset which then leads for Feature Engineering.

- **Descriptive Statistics**

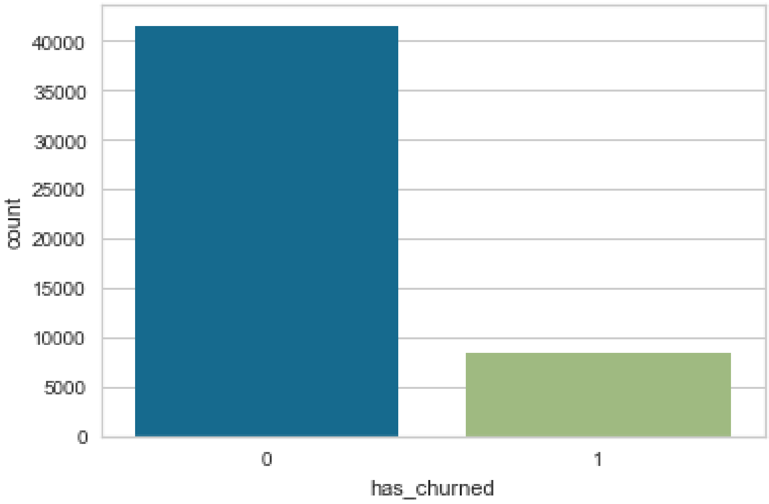
The selected dataset is having 50000 unique records that each record is mapped for one customer Id. The following Table 4:2 is tabulating the descriptive statistics of the data set along with all the attributes consisted in the dataset. In the data preparation stage using sql, the data type conversion has been done for all the attributes in to numeric initially. The null values that are presented in the dataset have been replaced by mean imputation.



	count	mean	std	min	25%	50%	75%	max
has_churned	50000.0	0.169480	0.375179	0.00000	0.00000	0.00	0.0000	1.00
tenure	50000.0	21.807480	15.885298	0.00000	9.00000	19.00	30.0000	61.00
pref_login_device	50000.0	1.672120	0.469446	1.00000	1.00000	2.00	2.0000	2.00
city_tier	50000.0	1.960420	0.834954	1.00000	1.00000	2.00	3.0000	3.00
wh_to_home	50000.0	24.630123	25.281444	0.16948	11.00000	20.00	29.0000	127.00
preferred_payment_method	50000.0	3.190300	1.404865	1.00000	2.00000	3.00	5.0000	5.00
gender_val	50000.0	0.488440	0.499871	0.00000	0.00000	0.00	1.0000	1.00
hrs_spent_on_app	50000.0	2.984200	0.754242	1.00000	2.00000	3.00	4.0000	5.00
no_of_devices	50000.0	3.594080	1.023899	1.00000	3.00000	4.00	4.0000	6.00
preferred_category	50000.0	2.697640	1.523911	1.00000	1.00000	2.00	4.0000	5.00
satisfaction_score	50000.0	2.929499	1.646544	0.16948	1.00000	3.00	5.0000	5.00
marital_status_val	50000.0	1.982700	0.804297	1.00000	1.00000	2.00	3.0000	3.00
no_of_addresses	50000.0	9.176900	6.848333	1.00000	4.00000	7.00	11.0000	22.00
complain	50000.0	0.266768	0.439346	0.00000	0.00000	0.00	1.0000	1.00
order_amt_hike	50000.0	14.367340	5.736994	0.00000	12.00000	14.00	18.0000	26.00
coupon_used	50000.0	2.004888	2.571447	0.00000	0.16948	1.00	2.0000	16.00
order_count	50000.0	14.779660	5.001261	1.00000	12.00000	14.00	18.0000	26.00
time_since_last_order	50000.0	7.500480	8.825393	0.00000	2.00000	4.00	9.0000	46.00
total_lifetime_loyalty_points	50000.0	182.717948	60.870128	0.00000	146.00000	166.00	213.0000	325.00
time_since_last_login	50000.0	5.482440	7.152708	0.00000	1.00000	3.00	7.0000	46.00
total_lifetime_expenditure	50000.0	599.771718	552.739023	0.16948	215.00750	437.23	764.3625	5110.74

Table 4:2 - Descriptive statistics of the data set

When considering the dataset descriptive statistics and specially the class variable ‘has_churned’, it gives the following graph in Figure 4:2 and it clearly depicted that there is an imbalance of the class variable with a big difference with 0.17 to 0.83 ratio.



```
Retained  customer percentage = 83.052 %
Churned customer percentage = 16.948 %
```

Figure 4:2 - Customer churn count spread

As the selected method for mitigating the biased analysis, SMOTE method approach has been chosen for data preparation. The underline reason that has been considered for this is, the approach is more logical and the new data generation would not just be random since it facilitates with K-Means algorithm to generate each and every data value based on the rest of the data values.

3.2.2 Data Mining Process

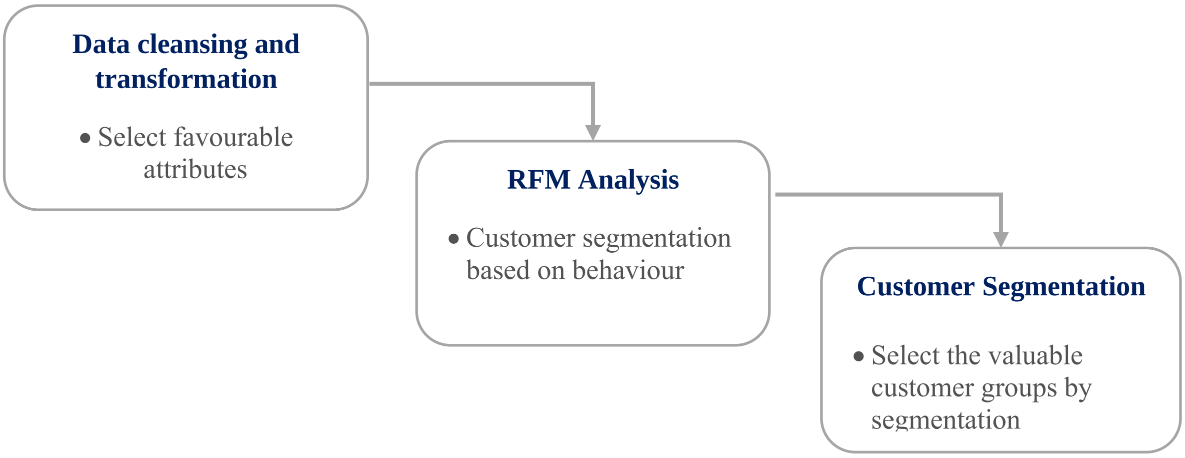

Data cleansing and transformation
• Select favourable attributes
RFM Analysis
• Customer segmentation based on behaviour
Customer Segmentation
• Select the valuable customer groups by segmentation

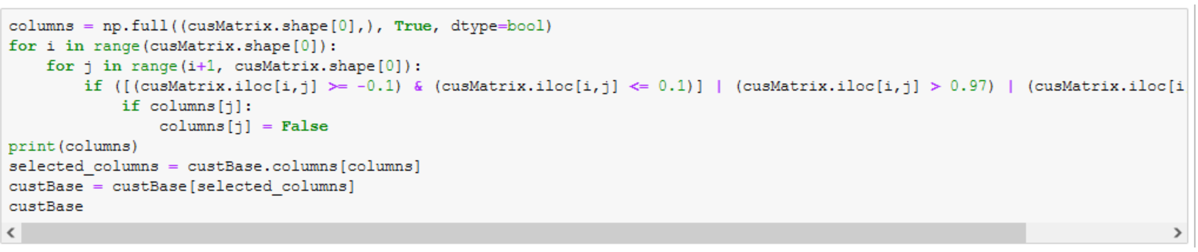
Figure 4:3 - Data mining process for feature selection

According to the Amjad and Zahid's (Amjad Khani, Zahid Ansari, 2014) study on customer churn for telecommunication domain, they have directly used the data of customers' call details which gives more insights efficiently. This implies that regardless the domain it is wise to choose the most relevant attributes for data mining and it could be better if this is consolidated with customer segmentation with profiling.

For extracting and filtering the relevant attributes from the data set, the data mining phase is conducted. The collected data should be filtered and cleansed first in order to proceed the data mining process. This step is more crucial since, only accurate data generates the accurate results. The importance of this phase is about the relevance and relationships among the selected attributes and the variables.

3.2.3 Feature Selection

As the next step, it has been identifying the related attributes which imply associations in between. Identifying the most relevant attributes is critical for the proposed prediction model's accuracy. Hence, as per the selection criteria described in the Figure 4:4 below, the highly negative and highly positive correlation have been eliminated with the value higher than +0.97 and lower than -0.97. Along with that, the neutral correlations within the range from +0.1 to -0.1 also have been eliminated.



```
columns = np.full((cusMatrix.shape[0],), True, dtype=bool)
for i in range(cusMatrix.shape[0]):
    for j in range(i+1, cusMatrix.shape[0]):
        if ([(cusMatrix.iloc[i,j] >= -0.1) & (cusMatrix.iloc[i,j] <= 0.1)] | (cusMatrix.iloc[i,j] > 0.97) | (cusMatrix.iloc[i
            if columns[j]:
                columns[j] = False
print(columns)
selected_columns = custBase.columns[columns]
custBase = custBase[selected_columns]
custBase
```

Figure 4:4 - Selecting the correlated attributes

- **Correlation Matrix**

The correlation matrix with the heat map illustration using the Pearson correlation coefficient has been used to identify the linear correlation between the attributes. The attributes which are having the positive and negative correlation has been selected using the python code segment.

The correlation matrix with the heat map illustration using the Pearson correlation coefficient has been used to identify the linear correlation between the attributes. The above mentioned Figure 4:5 is the final correlation matrix that has been resulted for the total customer data set.

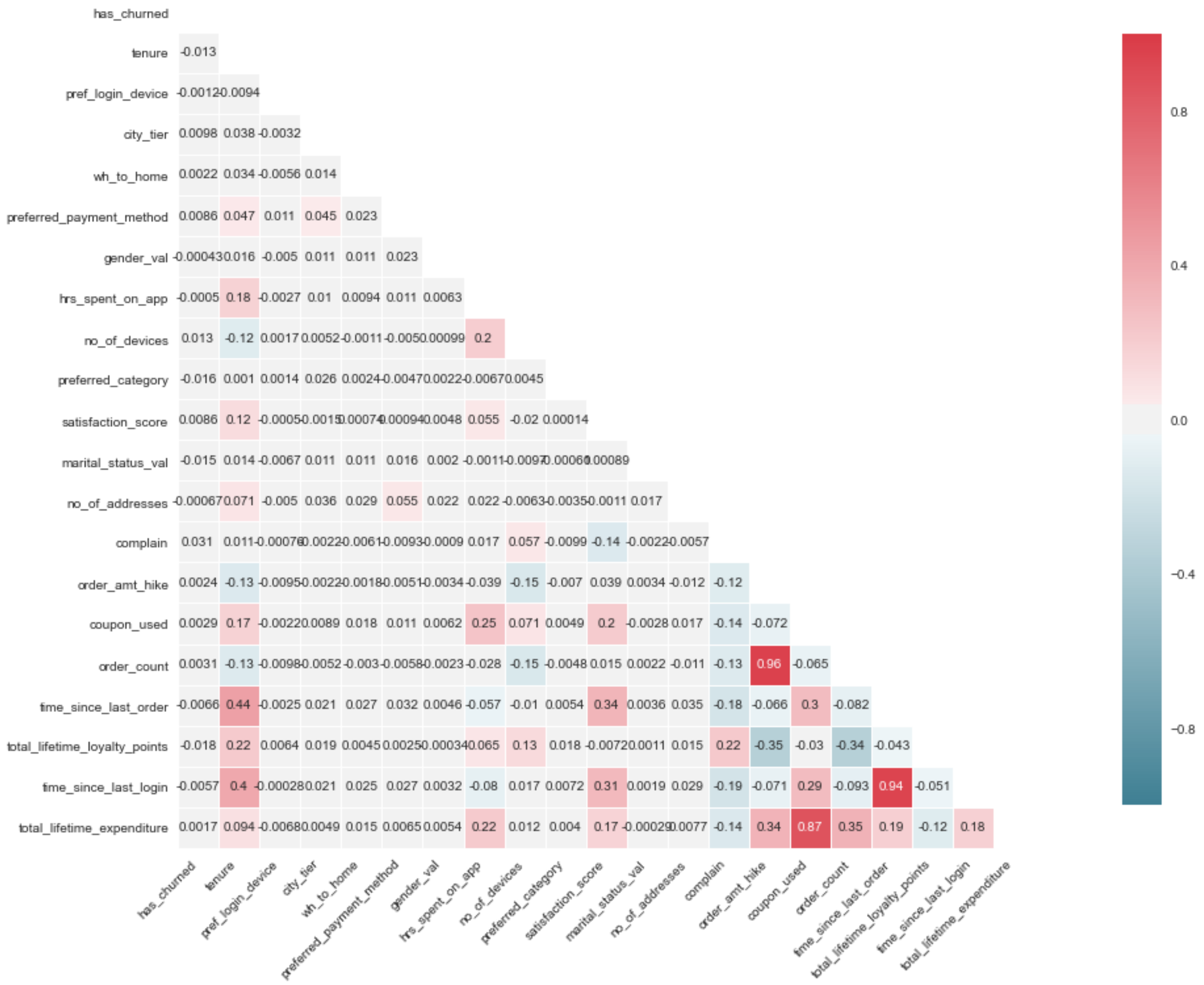


Figure 4:5 - Correlation Matrix

3.3 Phase 1: Customer Segmentation

Analytical techniques like predictive modelling and customer profiling can be leveraged as powerful tools to manage the problem of customer churn. (Renjith, September 2015). For the suggested predictive model, inputs are going to be the attributes that have been derived from the data mining step association rules.

3.3.1 Customer Segmentation by Clustering

The filtered extracted data then should be manipulated through a learning process in order to train the data set to utilize for the test data set. This facilitates the recognizing the repeating and related pattern midst the data attributes and variables. The machine learning algorithms and pattern recognition techniques are availed for this phase.

It is often used as a data analysis technique for discovering interesting patterns in data, such as groups of customers based on their behaviour. (Brownlee, 2020)

Here, based on the order and sales data along with the user logs data for each customer, the clustering of data need to be initiated as depicted in Figure 4:6. The clustering would be identified as per the customer types mentioned earlier.

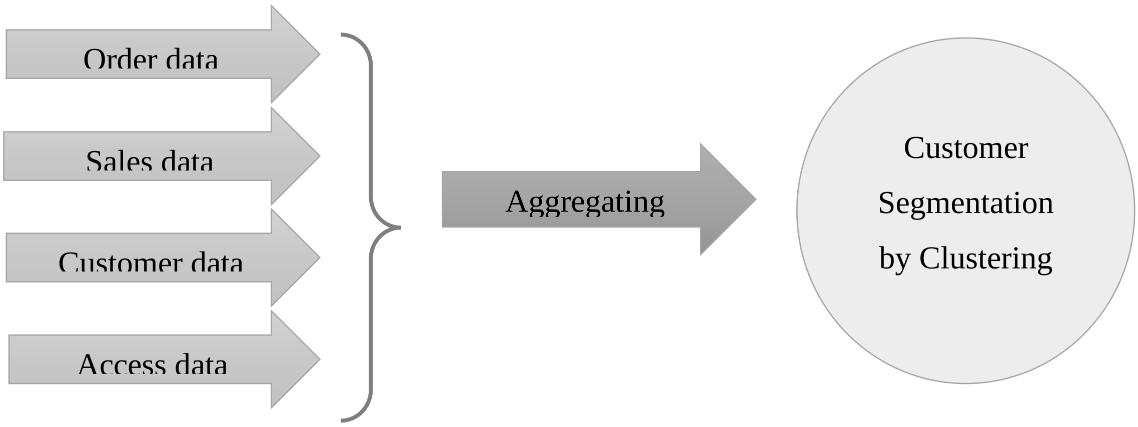


Figure 4:6 - Customer segmentation

For generating more accurate and refined clusters, it is necessary to follow a more relevant algorithm. Here, the selected algorithm is K-Means which continues an iterative mode. At each iteration, every single data point is going to be evaluated and assigned to the nearest cluster.

3.3.2 RFM Analysis on the Segmented Customers

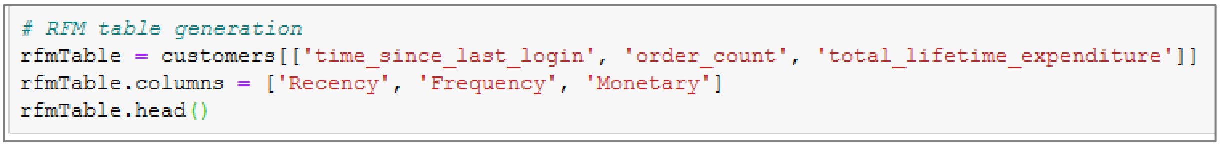
The main approach that has been used for the customer segmentation is RFM analysis which is recognized as a well performing marketing strategy in the business domain. By utilizing the RFM analysis, the whole dataset which is consisted of 50000 customers' records has been monitored and segmented into the preferred customer groups by K-Means algorithm.

As for the initial step in RFM analysis, the related attributes for Recency, Frequency and Monetary has been selected from the available dataset as follows.

Recency: time_since_last_login

Frequency: order_count

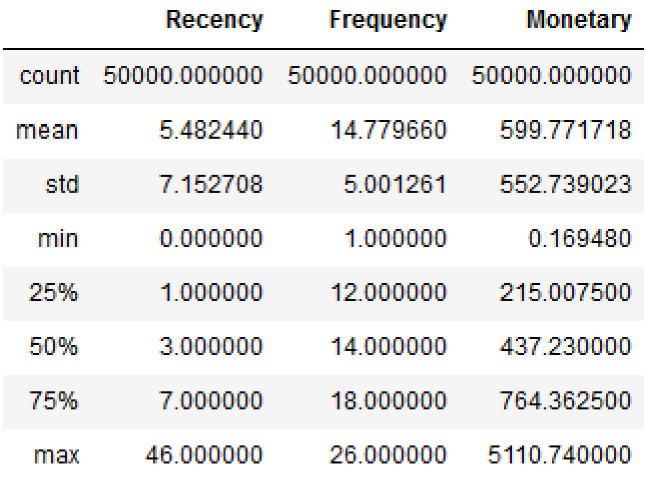
Monetary: total_lifetime_expenditure



```
# RFM table generation
rfmTable = customers[['time_since_last_login', 'order_count', 'total_lifetime_expenditure']]
rfmTable.columns = ['Recency', 'Frequency', 'Monetary']
rfmTable.head()
```

Figure 4:7 - Initial RFM table generation

As the above code segment in the Figure 4:7, the most suitable attributes have been selected from the whole customer dataset and generated the RFM table for further analysis is described in the following Table 4:3.



	Recency	Frequency	Monetary
count	50000.000000	50000.000000	50000.000000
mean	5.482440	14.779660	599.771718
std	7.152708	5.001261	552.739023
min	0.000000	1.000000	0.169480
25%	1.000000	12.000000	215.007500
50%	3.000000	14.000000	437.230000
75%	7.000000	18.000000	764.362500
max	46.000000	26.000000	5110.740000

Table 4:3 - Initial RFM table before scaling

As the above Table 4:3 tabulated, the data has been spread for different ranges and this should be avoided in order to generate accurate data with a specified range. The result has been depicted in the following Figure 4:8.

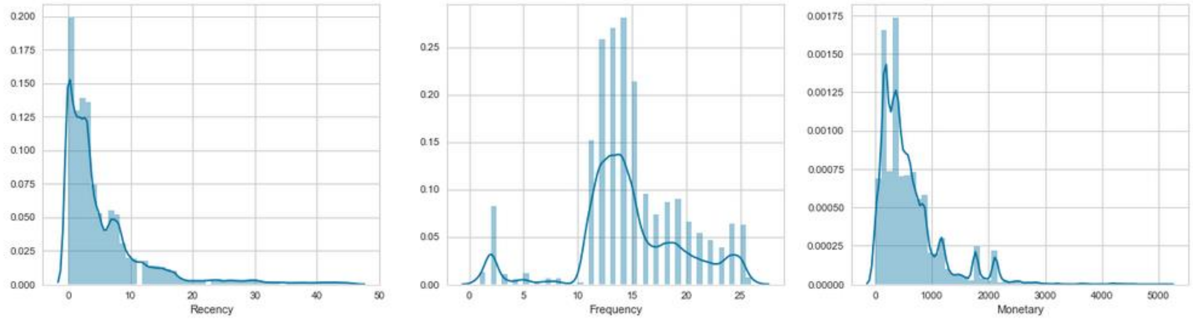
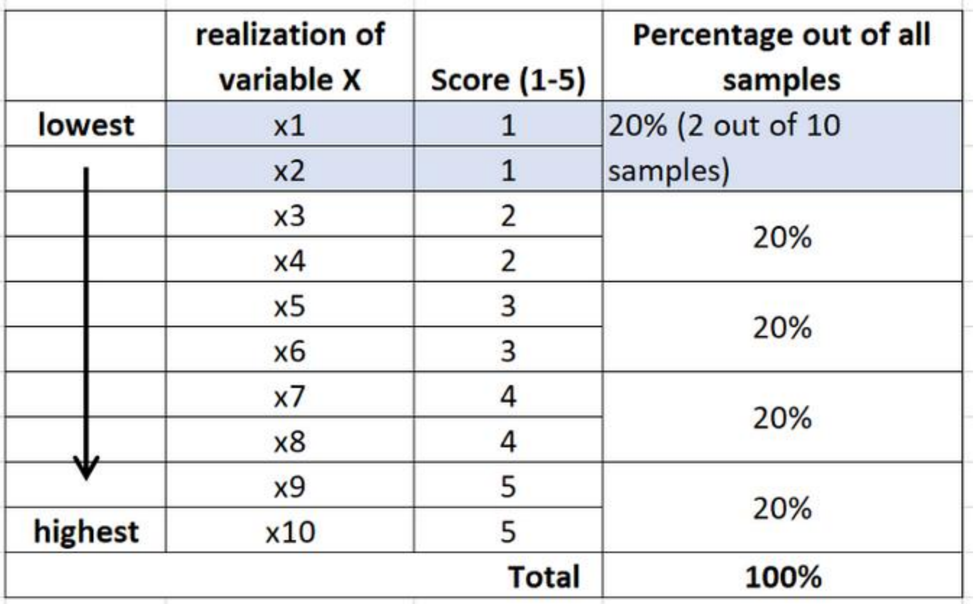


Figure 4:8- Graph of the RFM values before scaling

The values of each graph vary from different ranges and this makes the decision-making bias and interpretation inaccurate. These concerns need to be addressed beforehand for a proper data analysis.

3.3.3 RFM Analysis with RFM Scoring

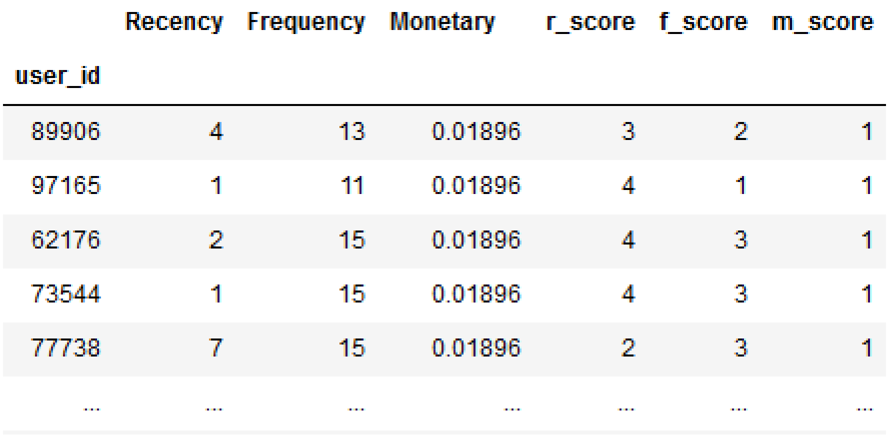
In most basic RFM analysis, for each variable with score 1 to *n*, we are dividing data into n equal-sized group. So, there are 1/n of samples have same score. (For example, selecting top 20% with lowest value for R variable and allocating score of 5. Subsequent 20% for score 4, and continues.) (Hshan.T, 2020)



	realization of variable X	Score (1-5)	Percentage out of all samples
lowest	x1	1	20% (2 out of 10 samples)
	x2	1	
	x3	2	20%
	x4	2	
	x5	3	20%
	x6	3	
	x7	4	20%
	x8	4	
	x9	5	20%
highest	x10	5	
Total			100%

Table 4:4 - Illustration of RFM scoring method

According to the RFM scoring, the criteria should be formed considering the number of bins that requires. For this study as the customers are willing to segment into 5 groups, and that is considered as the same number of bins for the score range. Accordingly, the score range has been selected within the range 1 to 5 which allocates the same size of data points to each bin or cluster as the above Table 4:4. The following Table 4:5 is a segment of the normalized customer dataset after assigning the RFM score for each individual user.



user_id	Recency	Frequency	Monetary	r_score	f_score	m_score
89906	4	13	0.01896	3	2	1
97165	1	11	0.01896	4	1	1
62176	2	15	0.01896	4	3	1
73544	1	15	0.01896	4	3	1
77738	7	15	0.01896	2	3	1
...	...	...	...	...	...	...

Table 4:5 - RFM Scoring Assigning

The generated result has been monitored for the correlation between each and every variable of the table as follows in order to identify the influence of them. And as the Figure 4:9 depicted below, it is clear that there is a considerable correlation between attributes m_score - f_score and m_score - r_score while there is no sufficient correlation between m_score and r_score.

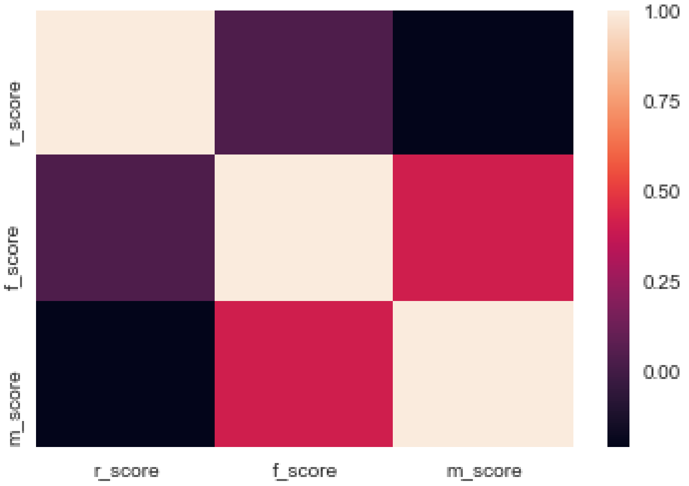


Figure 4:9 - Correlation of generated RFM table

Until this step of the process, the data set has been pre-processed, transformed, cleansed and finally the relevant attributes have been filtered for the next step of the process; the Clustering.

3.3.4 K-Means Algorithm

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. “A cluster refers to a collection of data points aggregated together because of certain similarities. (Garbade, 2018)

As an unsupervised learning mechanism, K-Means algorithm fulfills the need of grouping the similar data points into specific clusters based on the underlying patterns as depicted in Figure 4:10. It is advised to select the optimum number of clusters which denotes the "K" and the averaging values which denotes the "means" of the data set, in "K-Means" algorithm.

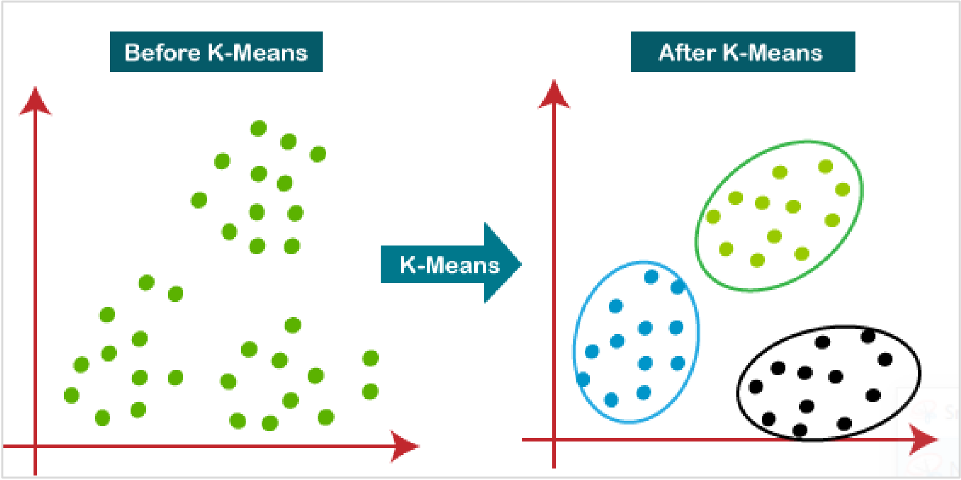
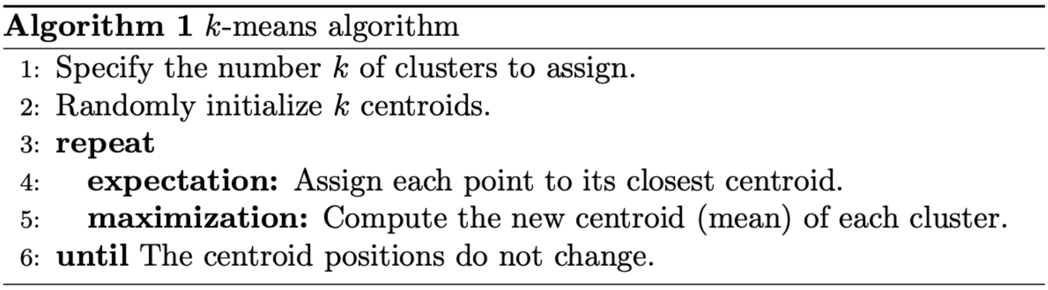


Figure 4:10 - K-Means algorithm application

K-Means Algorithm

1. Determine K- number of clusters
2. Select random K centroids which represents centre of the clusters
3. Assign the data points to the nearest centroid of the predetermined cluster
4. Repeat until the defined iterations are covered and clusters are stabilized and finalized



Algorithm 1 k-means algorithm

```
Specify the number $k$ of clusters to assign.
Randomly initialize $k$ centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.
```

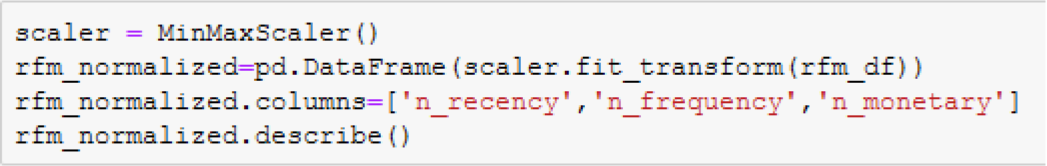
(1)

3.3.5 Standardization

When independent variables in training data are measured in different units, it is important to standardize variables before calculating distance. In order to make them comparable we need to standardize them which can be done by any of the following methods. (Bhalla, n.d.)

In so far as, it has been generated the relevant table the values are not standardized or scaled as it displays in the Table 4:3. The variable values have been in a large range where with a large variation. Since the K-Means evaluation acts upon the distance-based behavior of the data points, data should be adjusted in order to eliminate a biased model designing.

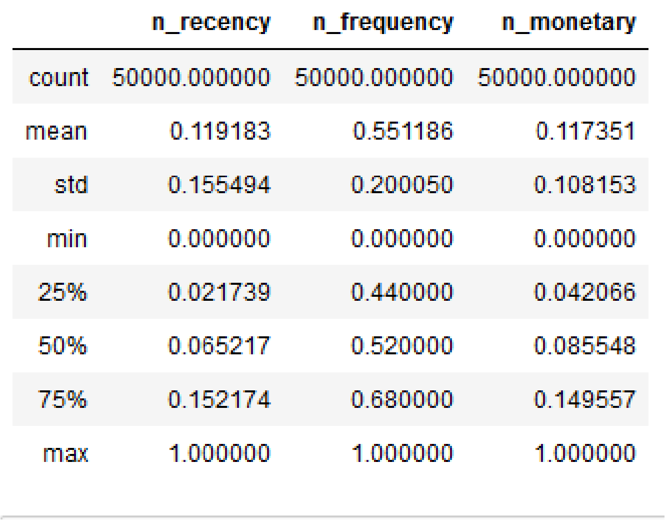
Thus, the selected dataset has been normalized by scaling using MinMaxScaler from scikitlearn library which reshapes the expansions of the attribute values between the minimum of 0 to maximum of 1 in order to get more readable and meaningful interpretation as the Figure 4:11 below.



```
scaler = MinMaxScaler()
rfm_normalized=pd.DataFrame(scaler.fit_transform(rfm_df))
rfm_normalized.columns=['n_recency','n_frequency','n_monetary']
rfm_normalized.describe()
```

Figure 4:11 - Scaling of RFM dataset

After standardizing the RFM dataset, the dataset has been scaled as the MinMaxScaler within the range of 0-1 as the below mentioned Table 4:6. The table is about the Five Number Summary along with the count, means and standard deviation. This standardized/scaled data set is the input this point on wards for the next step of the study.



	n_recency	n_frequency	n_monetary
count	50000.000000	50000.000000	50000.000000
mean	0.119183	0.551186	0.117351
std	0.155494	0.200050	0.108153
min	0.000000	0.000000	0.000000
25%	0.021739	0.440000	0.042066
50%	0.065217	0.520000	0.085548
75%	0.152174	0.680000	0.149557
max	1.000000	1.000000	1.000000

Table 4:6 - Standardized RFM Dataset

After standardizing the RFM dataset, it has resulted the following graph in Figure 4:12 which indicates the data range also within 0-1 for all three of Recency, Frequency and Monetary.

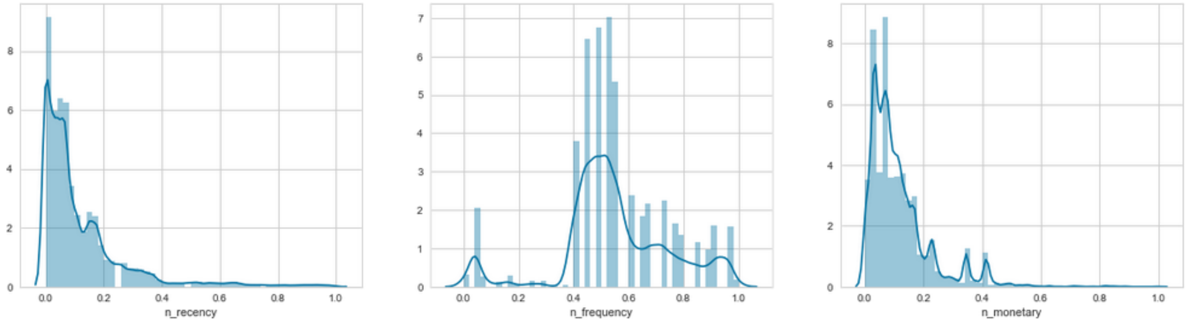
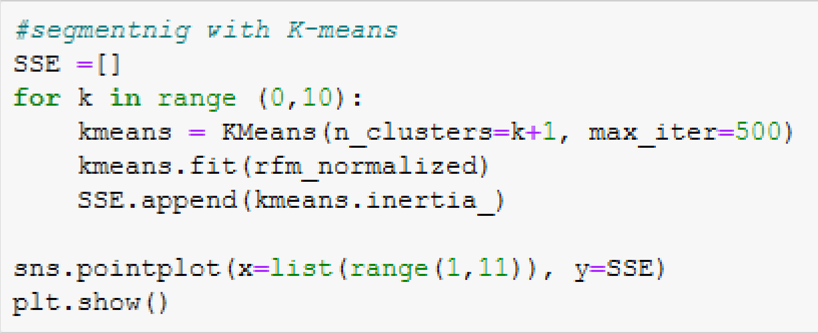


Figure 4:12 - Graph of the RFM values after scaling

3.3.6 Deciding the optimum number of K

Before applying the K-Means algorithm, it is necessary to identify the optimum number of ‘K’ i.e., the optimum number of clusters into which groups the dataset can be divided in the most meaningful way. For deciding this, the Elbow method has been used as indicating in the following Figure 4:13. According to the result, it indicated the subsidence of the inertia at the elbow point K = 5 and since the subsidence after wards is not that significant, it has been selected the optimum K as 5.



```
#segmentnig with K-means
SSE =[]
for k in range (0,10):
    kmeans = KMeans(n_clusters=k+1, max_iter=500)
    kmeans.fit(rfm_normalized)
    SSE.append(kmeans.inertia_)

sns.pointplot(x=list(range(1,11)), y=SSE)
plt.show()
```

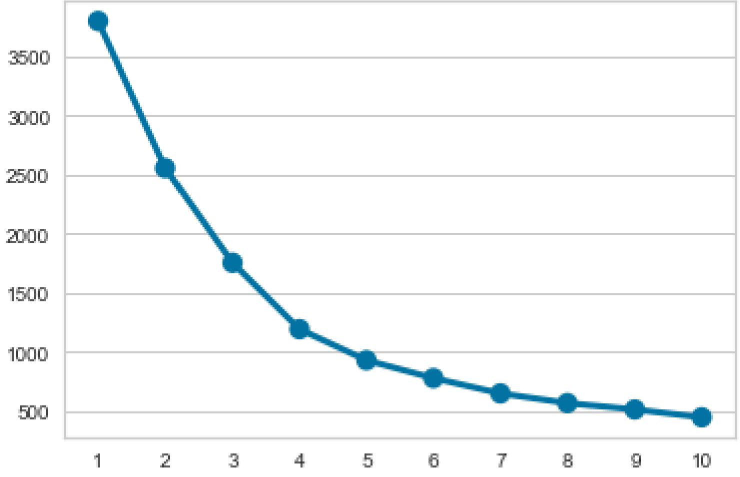


Figure 4:13 - Plot of inertia by Elbow method against optimal 'K'

Also, in this study it is willing to segment the dataset into 5 groups, the resulted optimal K is accepted. For indicate the result in well-structured graphical plot, the library ‘seaborn’ has been used.

3.3.7 Clustering with Silhouette Analysis

Silhouette analysis can be used to study the separation distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of [-1, 1] (Anon., n.d.).

According to the Silhouette visualizer official documentation, the analysis of clusters is measured by Silhouette coefficients within the range of -1 to +1 and when the result is on or near around the value of 0, it indicates the resulted sample is closer to the decision boundary. When the sample is closer to -1, it indicates the wrong assigning of data points to the wrong clusters meanwhile, the +1 indicating the sample is unnecessarily away from the belonging cluster.

After applying this Silhouette coefficient for clustering the normalized customer dataset, the result given as the following Figure 4:14 with 5 clusters. The clusters have been in the range of -0.1 to 0.7 and each and every cluster is having a proper spread of data points with a majority of them are on and closer to 0 which indicates better clustering of the data points. For all the identified clusters, average Silhouette score is 0.5 and the majority of the clusters are below to the average point and closer to 0.

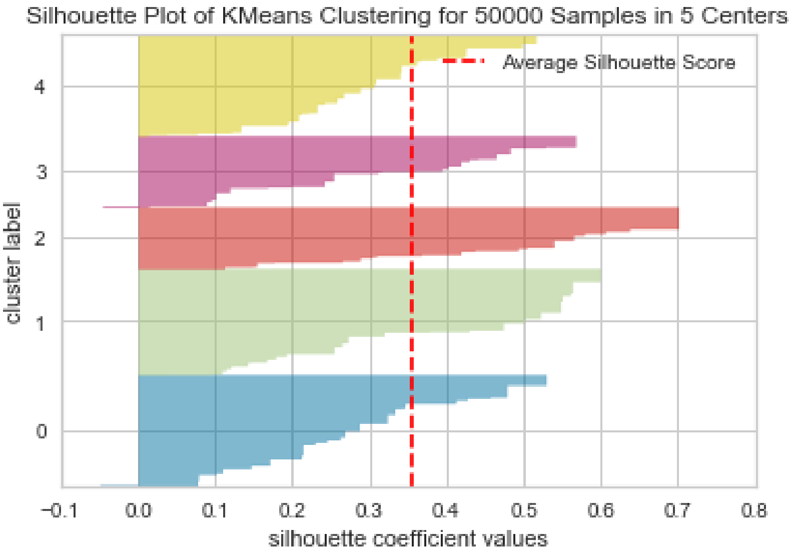


Figure 4:14 - Cluster analysis against Silhouette coefficient

The finalized clusters have been then graphed in a 3D scatter plot for more readability and it gives a real visualization of the way how these data points have been situated with the all three RFM score variables which indicates in the following Figure 4:15.

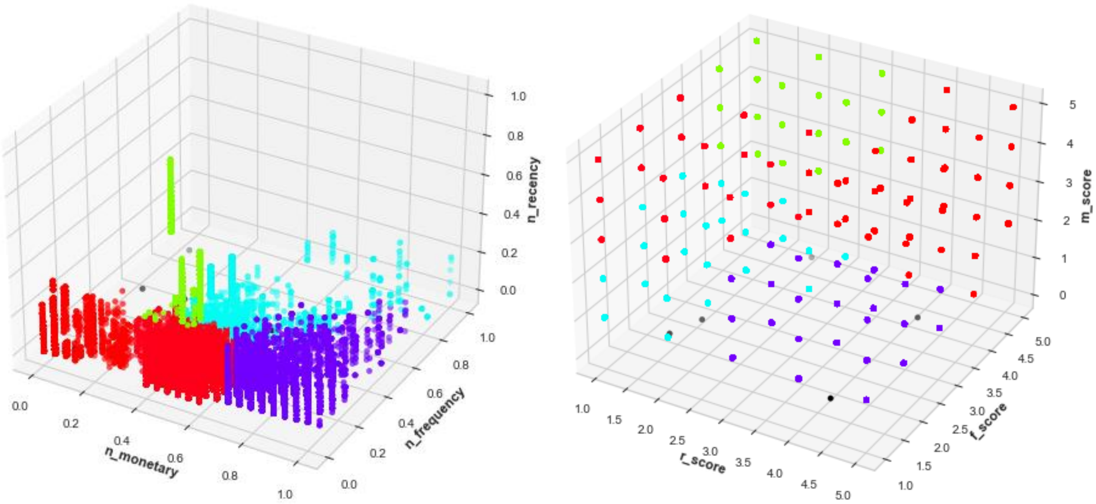
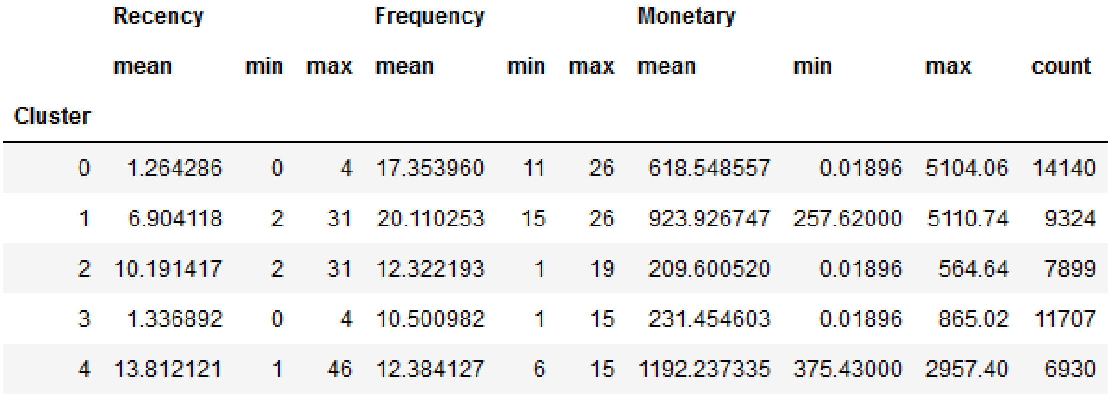


Figure 4:15 - 3D Scatter plot of the RFM with and without scoring

3.3.8 Selecting Customers based on the Clustering

As per the final step in the customer segmentation, the objective is to select the proper customer base according to the above clustering process which will be using as the valuable dataset for the further analysis in the study. In order to identify the each above depicted cluster's real values, the clusters have been then monitored and studies thoroughly for identifying the customer type or the customer profiling.



	Recency			Frequency			Monetary			
	mean	min	max	mean	min	max	mean	min	max	count
Cluster										
0	1.264286	0	4	17.353960	11	26	618.548557	0.01896	5104.06	14140
1	6.904118	2	31	20.110253	15	26	923.926747	257.62000	5110.74	9324
2	10.191417	2	31	12.322193	1	19	209.600520	0.01896	564.64	7899
3	1.336892	0	4	10.500982	1	15	231.454603	0.01896	865.02	11707
4	13.812121	1	46	12.384127	6	15	1192.237335	375.43000	2957.40	6930

Table 4:7 - Customer segmentation by cluster summary

As tabulated in the above Table 4:7, the dataset has been divided into 5 defined by the RFM score and the data points of similar behavior have been assigned to the closest cluster. By considering the above final segmentation, the clusters have been assigned to the particular customer type considering the mean values of the attributes Recency, Frequency and Monetary as following Table 4:8.

Cluster	Customer Type	Explanation
0	Loyal Customer	• Has minimum mean Recency value • Has the maximum sample size • Generates average level (biased to upper) mean monetary value
1	Discounted Customer	• Has the maximum Frequency value • Has 3rd maximum sample size(biased to upper) • Generates the second maximum mean monetary value
2	Impulse Customer	• Has the second maximum Recency value • Has the lowest monetary value • Has moderate mean Frequency
3	Need based Customer	• Has the second minimum Recency value • Has 2^{nd} minimum sample size(biased to lower) • Has the minimum mean Frequency
4	Wandering Customer	• Has the minimum sample size(6930) • Has the maximum Recency value(13 weeks) • Has the minimum Frequency value(12)

Table 4:8 - Final Customer segmentation

3.3.9 Customer Base Selection Criteria

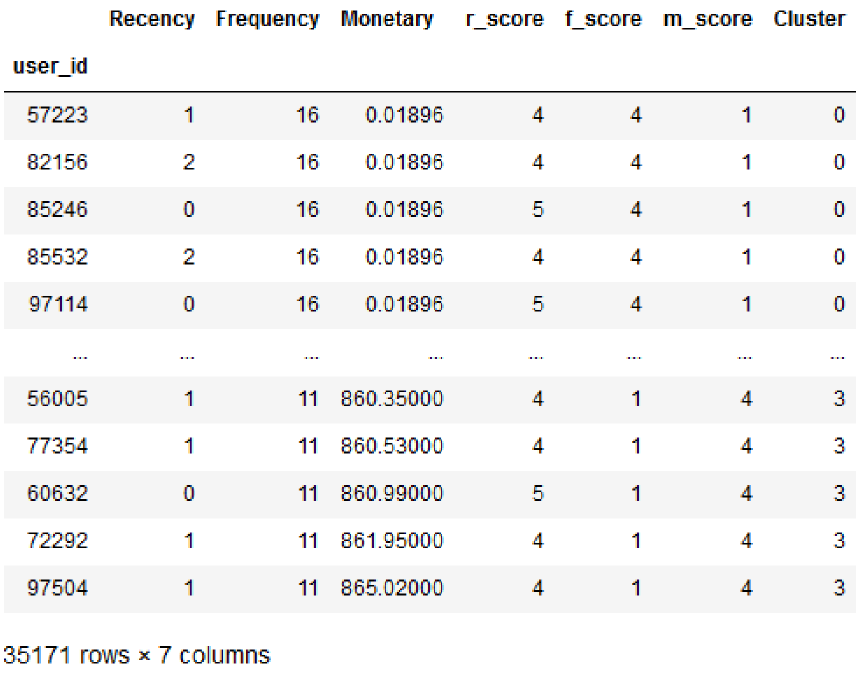
In order to select the optimal target customer base, the identified clusters have been cross checked and validated by the following selection criteria.

The target customer base should not contain the customers if;

Recency value: is higher than 6 weeks and,

Frequency value: minimum is greater than 6 weeks

Thus, Cluster 0-Loyal, 1-Discounted and 3-Need based customers have been selected as the customer base for modeling the predictive analysis onward. As the result, 35171 number of customers are considered in predictive analytics out of 50000 customers from the initial dataset as the below mentioned Table 4:9.



	Recency	Frequency	Monetary	r_score	f_score	m_score	Cluster
user_id							
57223	1	16	0.01896	4	4	1	0
82156	2	16	0.01896	4	4	1	0
85246	0	16	0.01896	5	4	1	0
85532	2	16	0.01896	4	4	1	0
97114	0	16	0.01896	5	4	1	0
...	...	...	...	...	...	...	...
56005	1	11	860.35000	4	1	4	3
77354	1	11	860.53000	4	1	4	3
60632	0	11	860.99000	5	1	4	3
72292	1	11	861.95000	4	1	4	3
97504	1	11	865.02000	4	1	4	3

35171 rows × 7 columns

Table 4:9 - Selected customer base

This customer base has been used this point on ward for predictive modelling and evaluation.

3.4 Phase 2: Predictive Modelling with Logistic Regression

The second phase of the study is about drawing the statistical inferences that can be finalized according to the identified patterns and utilize them for making the meaningful prediction for the upcoming situations in the scenarios. Statistical methods and analysis will be exhausted for this phase and reporting module and data visualization would be favourable.

As the selected two approach for the evaluation of the models in this phase, are Logistic Regression and Deep Neural Network. For the selected customer base, these two models of predictions have to be applied and the final outcome should be evaluated considering the accuracy and the reusability of the model.

3.4.1 Regression Analysis

In regression analysis, the variable you wish to predict is called the dependent variable. The variables used to make the prediction are called independent variables. In addition to predicting values of the dependent variable, regression analysis also allows you to identify the type of mathematical relationship that exists between a dependent variable and an independent variable, to quantify the effect that changes in the independent variable have on the dependent variable, and to identify unusual observations (Mark L. Berenson, David M. Levine, Timothy C. Krehbiel, 2012).

The linear regression model with n inputs can be represented by the formula:

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots\ldots + \beta nXn \tag{2}$$

Here, **y:** study/dependent variable

x1, x2 ... Xn: independent/explanatory variables

Corresponding formula in the logistic regression model is:

$$Q(X) = \frac{1}{1+ e^{-(\alpha + \beta 1X1 + \beta 2X2 + \ldots + \beta nXn)}} \text{ (Renjith, September 2015)} \tag{3}$$

a) Properties of Logistic Regression and Liner Regression

There are some of the differences and the similarities of Linear and Logistic regression, where the applicability and adaptability can be considered against each problem scenario as the following Table 4:10.

Linear Regression	Logistic Regression
Dependent variable follows the Normal distribution	Dependent variable follows the Bernoulli distribution
Output is continuous (E.g., Prices of goods, Rental of apartment)	Output is constant/discrete (E.g., churned or not, pass or fail)
Estimates using Ordinary Least Squares (OLS) – Minimizing distance approach	Estimates using Maximum Likelihood Estimation (MLE)

Table 4:10 - Linear Regression vs. Logistic Regression

b) Selecting Logistic Regression

When there is a problem concerning of classification and the classification is targeting two or binary class evaluation, the optimum regression analysis method is the Logistic Regression. Thus, the optimum approach for predictive modelling statistically for a binary classification is Logistic Regression. Hence, the study is targeting on identifying the customer for churning or non-churning basis, Logistic Regression method has been selected for the first model building.

As Renjith and Shini (Renjith, September 2015) described in his paper on regression models, he has compared and contrast about the linear and logistic regression models using the same dataset as following Figure 4:16.

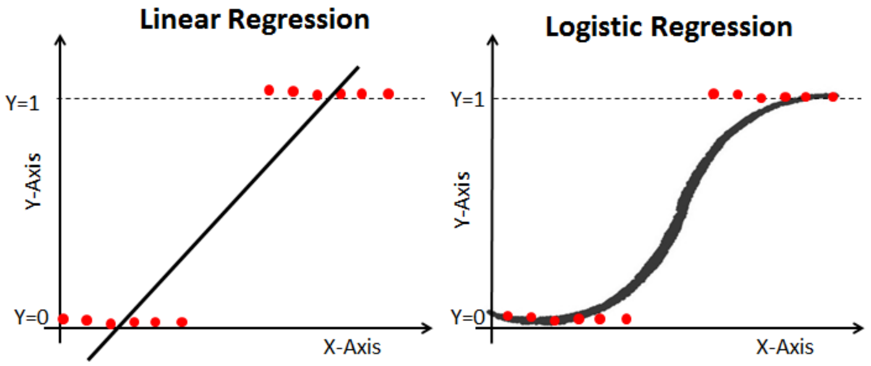


Figure 4:16 - Linear Regression vs. Logistic Regression

In linear regression, if the regression line is extended a few units upward or downward along the X axis, the predicted probabilities will fall outside legitimate range of 0.0 to 1.0. Logistic regression fits the correlation between X and Y with an S-shaped curve which is mathematically constrained to fit within the range of 0.0 to 1.0 on the Y axis.

In the B2C churn scenario, based on the predictive risk score the customers who are more likely to churn can be segmented out. The risk score along with other customer details available will form the input data for the next stage - cluster analysis to segment customers and apply strategies. (Renjith, September 2015)

The most suitable method for this predictive model is logistic regression model since it stands with classification problems as well. Since logistic regression model is useful with probabilities with binary classes and thus, it has become more applicable for the prediction model whether the specific customer tends to leave the business or not.

c) Advantages and Disadvantages of Logistic Regression

Even though it has been selected the approach as Logistic Regression, there can be both pros and cons come along with model itself irrespective to the problem scenario. The following Table 4:11 discusses the possible advantages and advantages over selecting the Logistic Regression.

Advantages	Disadvantages
Efficiency is higher	Not suitable for large number of categorical features
Easy to implement and interpret	Tend to be over fitting
No need of scaling the data	Cannot solve non-linear problem (needs additional transformation)
Provides probability score for evaluation	Provides inaccurate results if independent variable has not correlated with dependent variables

Table 4:11 - Pros and Cons of Logistic Regression

3.4.2 Logistic Regression Model Building

Based on the refined customer base derived from phase 1, the regression analysis has been applied. For understanding the nature of the target data variable, the initial data analysis has been drawn and since the data set is cleansed, transformed and pre-processed in the phase 1, it is not necessary for handling them.

As per the selected segmented customer base, there are 35171 customers who are going to be monitored for the predictive modeling. When considering the following Figure 4:17 of churned and non-churned customer count shows the spread of the customer count over the churning behavior.

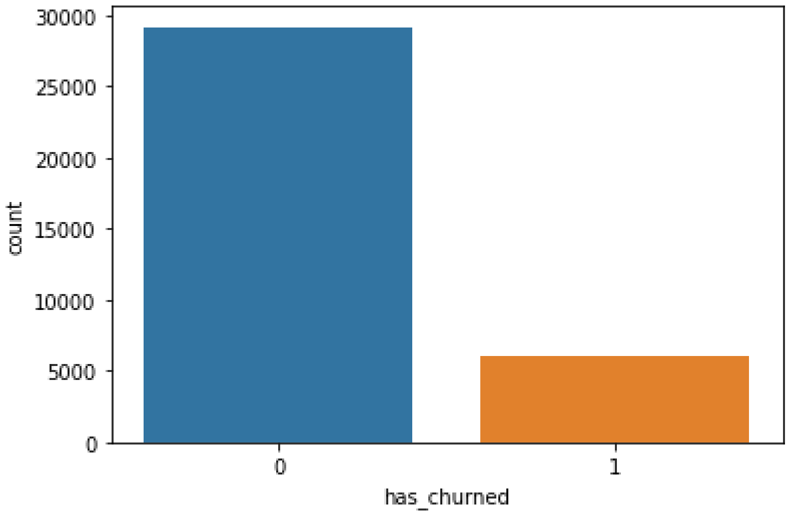
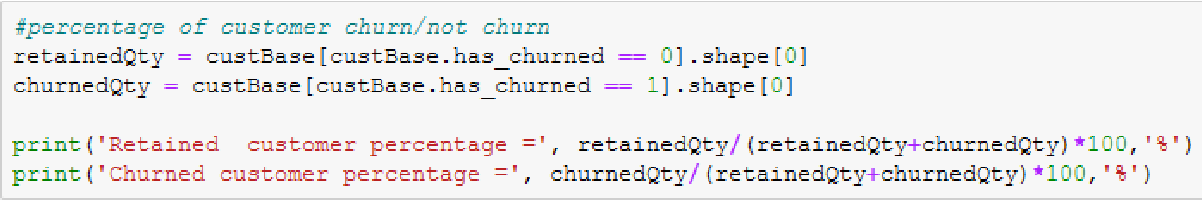


Figure 4:17 - Graph of churned and non-churned customer count

```
0    29155
1     6016
```

When calculating the churned and non-churned percentage of the whole customer base, it has given the following values which will be used for evaluating the accuracy of the regression model. The following Figure 4:18 indicates the spread of the data between the class level which is has_churned and retained customer percentages in values 82.89% and 17.1% respectively.



```
#percentage of customer churn/not churn
retainedQty = custBase[custBase.has_churned == 0].shape[0]
churnedQty = custBase[custBase.has_churned == 1].shape[0]

print('Retained  customer percentage =', retainedQty/(retainedQty+churnedQty)*100,'%')
print('Churned customer percentage =', churnedQty/(retainedQty+churnedQty)*100,'%')
```

Figure 4:18 - Dataset Spread in Class Level

```
Retained customer percentage = 82.8949987205368 %
```

```
Churned customer percentage = 17.105001279463195 %
```

When consider each attribute's behavior against the class variable 'has_churned', the result has been resulted through the graphs as depicted in the below Figure 4:19. The plots indicate that there is a biasness of 'has_churned' class variable towards the non-churned records since all the graphs are revealing a similar pattern with a high deviation.

In order to overcome this issue, in the later steps SMOTE has performed and mitigates the biasness in a considerable amount based on the minority class.

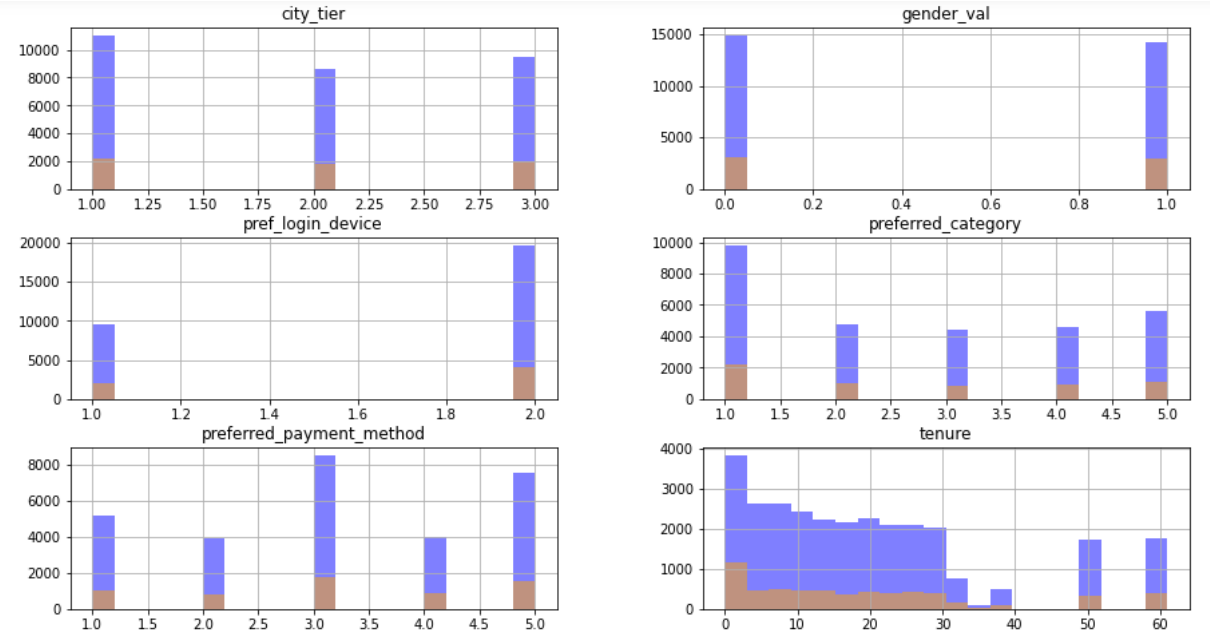
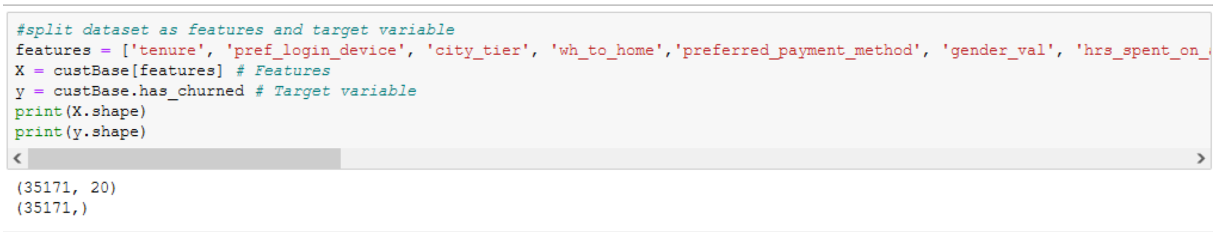


Figure 4:19 - **Data Visualization against Churn**

a) Splitting the dataset variables – Logistic Regression

For applying the regression model, first it is necessary to understand the study variable and the feature variables. There are few types of Logistic Regression methods and this study is about the Binary Logistic Regression since the target variable has only two possible values as churned and non-churned. As the dependent variable and the independent variable, the data frame's attributes have been considered as the below code segment in the following Figure 4:20.



```
#split dataset as features and target variable
features = ['tenure', 'pref_login_device', 'city_tier', 'wh_to_home','preferred_payment_method', 'gender_val', 'hrs_spent_on_
X = custBase[features] # Features
y = custBase.has_churned # Target variable
print(X.shape)
print(y.shape)
```

```
(35171, 20)
(35171,)
```

Figure 4:20 - Splitting the dataset variables

Accordingly, as the independent variables, tenure, pref_login_device, city_tier, wh_to_home, preferred_payment_method, gender_val, hrs_spent_on_app, no_of_devices, preferred_category, satisfaction_score, marital_status_val, no_of_addresses','complain, order_amt_hike, coupon_used, order_count, time_since_last_order, total_lifetime_loyalty_points, time_since_last_login, total_lifetime_expenditure have been selected meanwhile 'has_churned' has been selected as the independent or study variable.

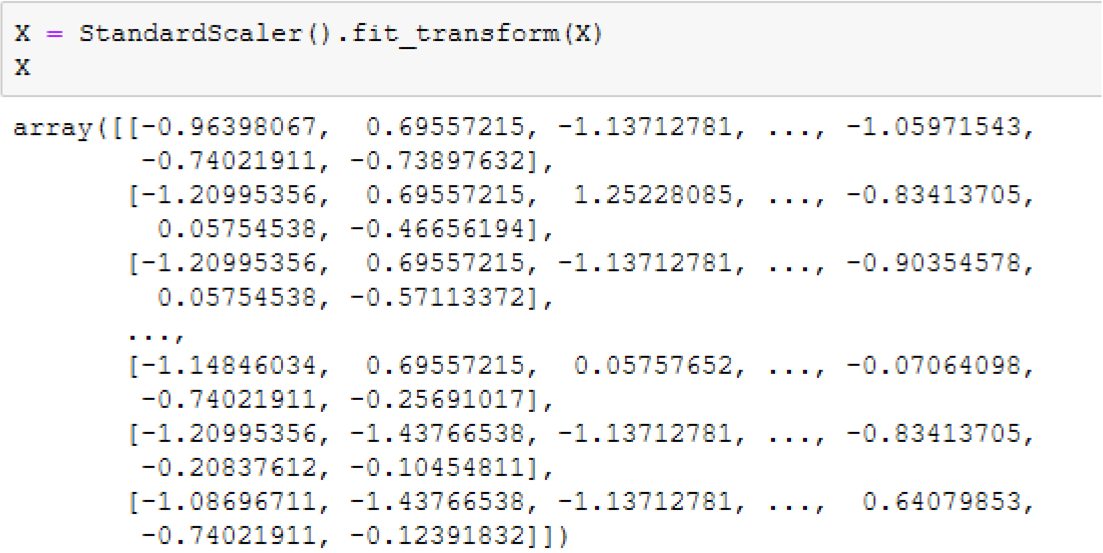
b) Standardizing/Scaling the Features – Logistic Regression

Machine learning algorithms like linear regression, logistic regression, neural network, etc. that use gradient descent as an optimization technique require data to be scaled. Take a look at the formula for gradient descent below:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{4}$$

The presence of feature value X in the formula will affect the step size of the gradient descent. The difference in ranges of features will cause different step sizes for each feature. To ensure that the gradient descent moves smoothly towards the minima and that the steps for gradient descent are updated at the same rate for all the features, we scale the data before feeding it to the model. "Having features on a similar scale can help the gradient descent converge more quickly towards the minima". (Bhandari, 2020)

Since the study has been done through the Logistic Regression for predictive modelling, the scaling of the feature variables is necessary to be done. As per the selected method, StandardScaler method has been used for scaling the features which scales the features by eliminating the mean and scaling them to the unit variance. And as the below Figure 4:21 the result has been scaled



```
X = StandardScaler().fit_transform(X)
X
```

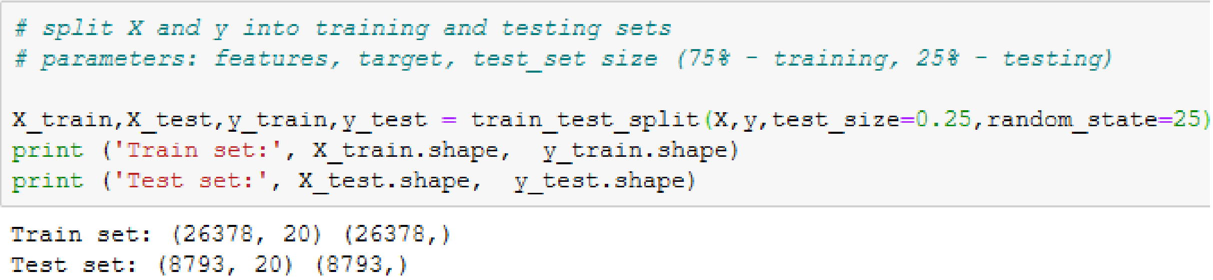
```
array([[-0.96398067,  0.69557215, -1.13712781, ..., -1.05971543,
        -0.74021911, -0.73897632],
       [-1.20995356,  0.69557215,  1.25228085, ..., -0.83413705,
         0.05754538, -0.46656194],
       [-1.20995356,  0.69557215, -1.13712781, ..., -0.90354578,
         0.05754538, -0.57113372],
       ...,
       [-1.14846034,  0.69557215,  0.05757652, ..., -0.07064098,
        -0.74021911, -0.25691017],
       [-1.20995356, -1.43766538, -1.13712781, ..., -0.83413705,
        -0.20837612, -0.10454811],
       [-1.08696711, -1.43766538, -1.13712781, ...,  0.64079853,
        -0.74021911, -0.12391832]])
```

Figure 4:21 - Scaling the feature variables by Standard Scalar

c) Selection of Training and Test Data Set – Logistic Regression

Splitting the dataset into training and test data set, a must do step in data pre-processing and it helps to avoid model "overfitting" or "underfitting". Overfitting of the model means that the even though the model performing well on the training dataset, it will be failing when applying the new or test data set. When the model is higher in complexity, the possibility of Overfitting is higher. Similarly, "Underfitting" means that the model performs poorly on the training data set due to the unsuitability of the model for the selected problem scenario. This could be due to model is less complex than to the expected level or problem.

As the following code segment in the Figure 4:22 explains, data has been split with the use of `scikit-learn` (`sklearn`) `train_test_split()` method, which 'Split arrays or matrices into random train and test subsets (Anon., n.d.). The training dataset's potion is 75% of the full data set meanwhile the test dataset contains 25% of it. The amount ratio to be used is not strictly emphasizing thus, there is a practice of dividing the dataset with 2/3 ratio for training and the rest for testing accordingly.



```
# split X and y into training and testing sets
# parameters: features, target, test_set size (75% - training, 25% - testing)

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.25,random_state=25)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

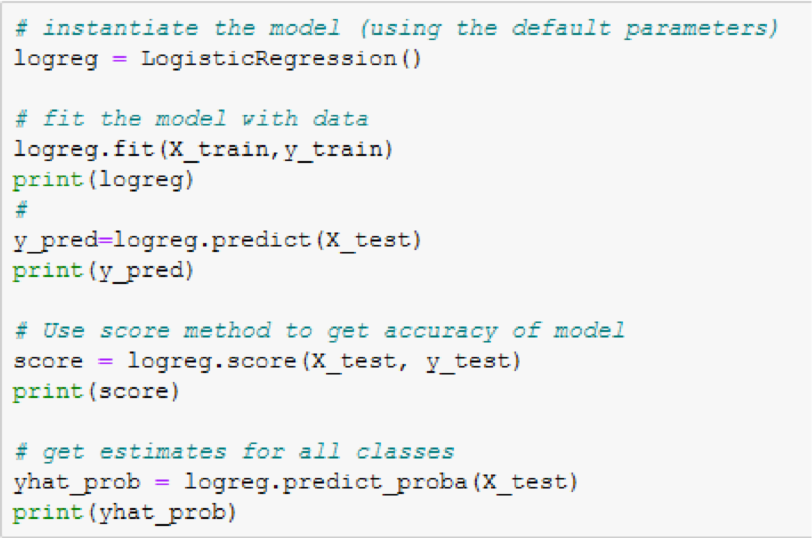
```
Train set: (26378, 20) (26378,)
Test set: (8793, 20) (8793,)
```

Figure 4:22 - Training and Test Data separation for Logistic Regression

d) Prediction Model Implementation – Logistic Regression

As per the final step of this phase, the prediction modelling is carried out. The followed approach is LogisticRegression and the training and test data set have been included for the study. The method has been taken from the `scikit-learn` (`sklearn`) library by using the default parameters which are **numerical optimizer** which help finding parameters, **C** parameter which indicates inverse of regularization strength with a positive float. Smaller the value, the regularization becomes stronger. There are some few numerical optimizers such as 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga' and the used one is the here default optimizer 'warn' solver. Logistic Regression of `scikit-learn` (`sklearn`) compatible with regularization which help solving model 'Overfitting' of machine learning models

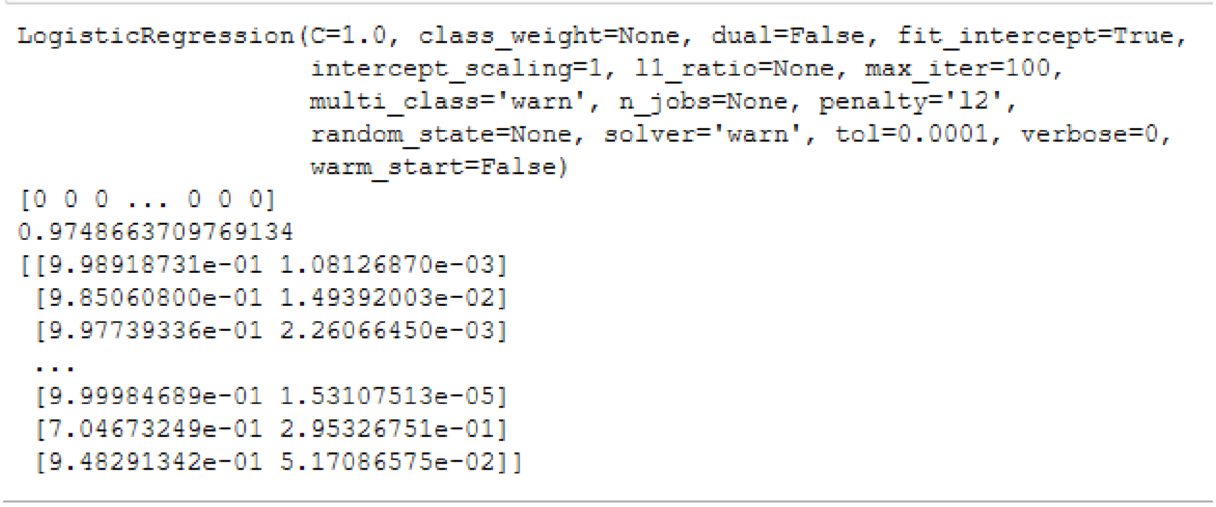
So, the model has been fit with the training set and testing with the test set using the code segment in the following Figure 4:23. As the code implies **predict_proba** gives estimations for all the study variable's classes, ordered by the label of classes. So, the first column gives the probability of class 1-has_churned, P(Y=1|X), while second column gives probability of class 0-not-churned, P(Y=0|X).



```
# instantiate the model (using the default parameters)
logreg = LogisticRegression()

# fit the model with data
logreg.fit(X_train,y_train)
print(logreg)
#
y_pred=logreg.predict(X_test)
print(y_pred)

# Use score method to get accuracy of model
score = logreg.score(X_test, y_test)
print(score)

# get estimates for all classes
yhat_prob = logreg.predict_proba(X_test)
print(yhat_prob)
```

Figure 4:23 - Applying Logistic Regression to the Model

According to the above process of Logistic Regression, the final outcome has been as following Figure 4:24. The result contains the accuracy score with a value of 97% and all the relevant details applied to the regression model. This is the final outcome of the model implementation of regression analysis model implementation.



```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
[0 0 0 ... 0 0 0]
0.9748663709769134
[[9.98918731e-01 1.08126870e-03]
 [9.85060800e-01 1.49392003e-02]
 [9.97739336e-01 2.26066450e-03]
 ...
 [9.99984689e-01 1.53107513e-05]
 [7.04673249e-01 2.95326751e-01]
 [9.48291342e-01 5.17086575e-02]]
```

Figure 4:24 - Logistic Regression Model Result

3.5 Phase 2: Predictive Modelling with Artificial Neural Network (ANN)

Most of the data pre-processing has been done in the segmentation phase, and some of the steps that have not been addressed before in the previous stages are only carried out in this phase when it is necessary.

3.5.1 Data Pre-processing for the Deep Neural Network Model Building

a) One Hot Encoding

The attribute values of the dataset should be in numerical form in order to feed the model properly. Hence, the available categorical variables' values should be encoded into binary vectors and One Hot Encoding is the process which has been used for type conversion in this phase. Most of the machine learning algorithms do not work well with categorical data and when there is a natural ordinal behavioural relationship in a categorical data it is possible to use data scaling directly. Even though, when there is no such ordinal structure but only nominal relationship, this needs a way of doing this programmatically. One hot encoding is such a technique that can be used for categorical variable vectorization by binary values.

Initially, all the categorical variable values have been mapped into numerical values in the study and since, each and every numeric value is represented as a binary vector which is all the zero values except the index of the integer, which is marked with a 1. The following attributes need to be binary vectorized by one hot encoding since they have only nominal integer values.

```
preferred_category: [1 5 2 3 4]
marital_status_val: [1 3 2]
preferred_payment_method: [2 1 3 5 4]
pref_login_device: [2 1]
city_tier: [1 3 2]
```

There are different ways of performing one hot encoding such as manual transforming, libraries like scikit-learn, Keras, pandas etc. and the selected method for this is, `pandas.get_dummies`(*data*, *prefix=None*, *prefix_sep='_'*, *dummy_na=False*, *columns=None*, *sparse=False*, *drop_first=False*, *dtype=None*) method implementation from pandas which converts categorical variable into dummy/indicator variables (Anon., n.d.). The result of the converted dummy variables are as follows. This is an instance of feature engineering where new features are introduced based on the existing features.

```
'city_tier_1', 'city_tier_2', 'city_tier_3',
'preferred_payment_method_1',
'preferred_payment_method_2',
'preferred_payment_method_3',
'preferred_payment_method_4',
'preferred_payment_method_5',
'preferred_category_1',
'preferred_category_2',
'preferred_category_3',
'preferred_category_4',
'preferred_category_5',
'marital_status_val_1',
'marital_status_val_2',
'marital_status_val_3'
```

b) Standardizing/Scaling the Features - ANN

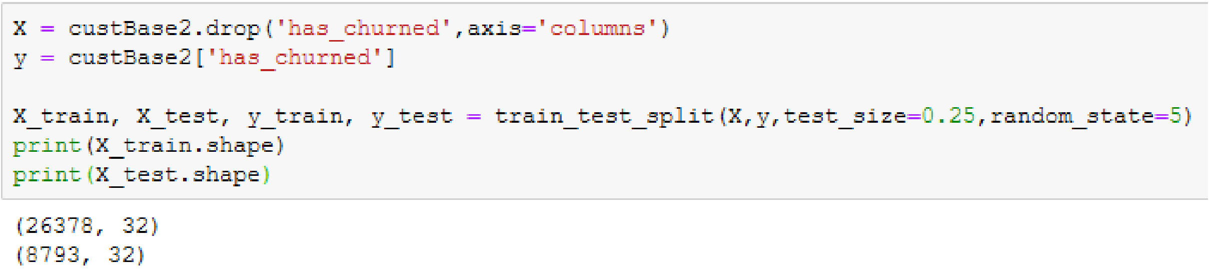
As in the previous steps, the scaling of data values gives more meaningful and organized interpretation of the data. The Scaling technique has been used here is MinMaxScaler of scikit-learn.processing and only the following selected columns need to be transformed.

```
['tenure','wh_to_home','hrs_spent_on_app','no_of_devices','no_of_addresses'
,'satisfaction_score','order_amt_hike','coupon_used','order_count','time_si
nce_last_order','total_lifetime_loyalty_points','time_since_last_login','to
tal_lifetime_expenditure']
```

Since the other attributes have been encoded by binary values within the 0-1 range, the MinMaxScaler is the most suitable technique to be followed. After scaling, the resulted outcome can be used for further analysis.

c) Selection of Training and Test Data Set - ANN

Similar to the above regression model building, the next step is the splitting and allocating the data set into training and test data sets. As the code segment of below Figure 4:25 describes, the training set has been considered with 75% percentage of total dataset and 25% has been allocated for testing purposes.



```
X = custBase2.drop('has_churned',axis='columns')
y = custBase2['has_churned']

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.25,random_state=5)
print(X_train.shape)
print(X_test.shape)
```

```
(26378, 32)
(8793, 32)
```

Figure 4:25 - Splitting the dataset variables

Accordingly, as the selected features for the model all the features including the encoded variables are considered with a count of 32 and the testing variable is ‘has_churned’. All the selected variables for the training set and the testing set are listed below along with the corresponding data types.

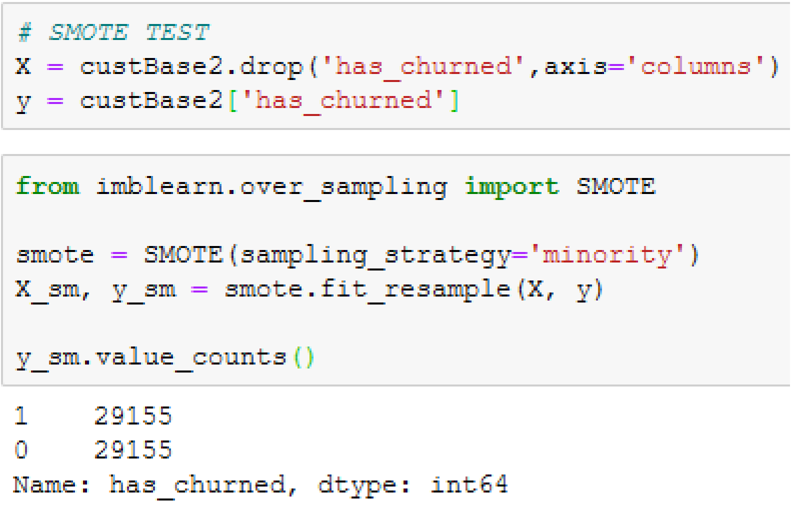
```
has_churned                          int64
tenure                               int64
pref_login_device                    int64
wh_to_home                         float64
gender_val                           int64
hrs_spent_on_app                     int64
no_of_devices                        int64
satisfaction_score                 float64
no_of_addresses                      int64
complain                           float64
order_amt_hike                       int64
coupon_used                        float64
order_count                          int64
time_since_last_order                int64
total_lifetime_loyalty_points      float64
time_since_last_login                int64
total_lifetime_expenditure         float64
city_tier_1                          uint8
city_tier_2                          uint8
city_tier_3                          uint8
preferred_payment_method_1           uint8
preferred_payment_method_2           uint8
preferred_payment_method_3           uint8
preferred_payment_method_4           uint8
preferred_payment_method_5           uint8
preferred_category_1                 uint8
preferred_category_2                 uint8
preferred_category_3                 uint8
preferred_category_4                 uint8
preferred_category_5                 uint8
marital_status_val_1                 uint8
marital_status_val_2                 uint8
marital_status_val_3                 uint8
```

The reason, for the increasing of number of attributes is encoding since, it introduces new attributes for each new binary attribute for the number of nominal categories it has.

3.5.2 Prediction Model Implementation – ANN

As per the final step of this phase, the prediction modelling of ANN is carried out. The followed approach is Deep Neural Network and the training and test data set have been included for the study. The method has been developed using tensorflow/keras module's sequential method. Sequential groups a linear stack of layers into a tf.keras.Model. Sequential provides training and inference features on this model (Anon., n.d.).

Since the filtered dataset is imbalanced as mentioned in Figure 4:18, the dataset needs to be enhanced in order to mitigate the biasness of the result. Imbalanced dataset has a direct impact on the machine learning model which will gives you an incorrect or biased reading of the trained model. For addressing this issue, SMOTE method discussed under Data pre-processing has been used as the following Figure 4:26 indicates.



```
# SMOTE TEST
X = custBase2.drop('has_churned',axis='columns')
y = custBase2['has_churned']
```

```
from imblearn.over_sampling import SMOTE

smote = SMOTE(sampling_strategy='minority')
X_sm, y_sm = smote.fit_resample(X, y)

y_sm.value_counts()
```

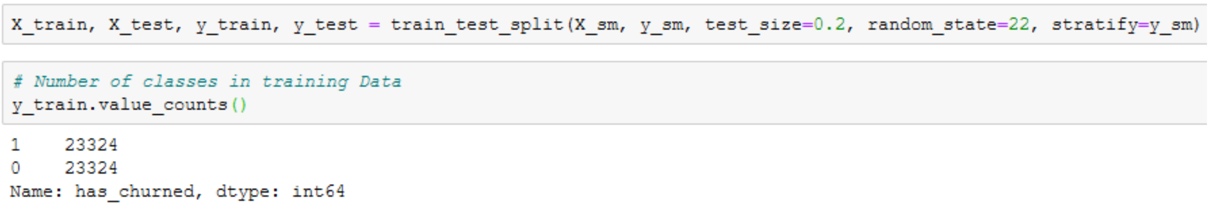
```
1    29155
0    29155
Name: has_churned, dtype: int64
```

Figure 4:26 - SMOTE on Dataset for Over-sampling

Since this method's strategy is to refine the minority class which is 0 with the count of 6016, is calculated to be matched with the amount of the majority class which is 1 with the count of 29155 based on the K-Means algorithm. Since the data is balanced completely now, then the model has been developed.

3.5.3 Selection of Training and Test Data Set – ANN

As the following code segment in the Figure 4:27 explains, data has been split with the use of scikit-learn (sklearn) train_test_split() method, which 'Split arrays or matrices into random train and test subsets (Anon., n.d.). The training dataset's potion is 75% of the full data set meanwhile the test dataset contains 25% of it. The amount ratio to be used is not strictly emphasizing thus, there is a practice of dividing the dataset with 2/3 ratio for training and the rest for testing accordingly



```
X_train, X_test, y_train, y_test = train_test_split(X_sm, y_sm, test_size=0.2, random_state=22, stratify=y_sm)
```

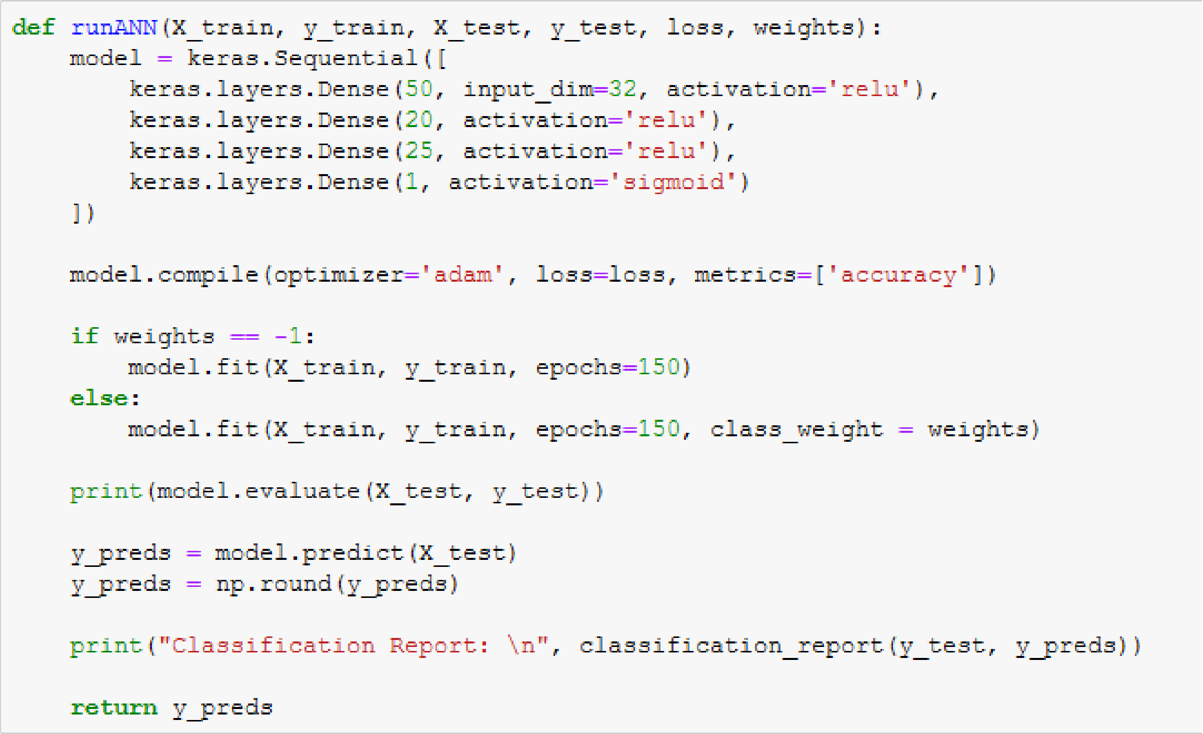
```
# Number of classes in training Data
y_train.value_counts()
```

```
1    23324
0    23324
Name: has_churned, dtype: int64
```

Figure 4:27 - Training and Test Data separation for ANN

3.5.4 Training the ANN Model

As the following code segment in the Figure 4:28, the followed neural network method is sequential and the input layer contains 50 neurones and 3 hidden layers with the activation function ‘relu’. For the hidden layers, it does not essential to specify the shape since it is derived from the input layer. As per the output layer, only one neurone is there and the result is based on 1 and 0, the activation function is ‘sigmoid’. When compiling the model, the optimizer that has been used is ‘adam’ with the loss function of ‘binary_crossentropy’ since the study model’s output is binary with the consideration of model accuracy.



```
def runANN(X_train, y_train, X_test, y_test, loss, weights):
    model = keras.Sequential([
        keras.layers.Dense(50, input_dim=32, activation='relu'),
        keras.layers.Dense(20, activation='relu'),
        keras.layers.Dense(25, activation='relu'),
        keras.layers.Dense(1, activation='sigmoid')
    ])

    model.compile(optimizer='adam', loss=loss, metrics=['accuracy'])

    if weights == -1:
        model.fit(X_train, y_train, epochs=150)
    else:
        model.fit(X_train, y_train, epochs=150, class_weight = weights)

    print(model.evaluate(X_test, y_test))

    y_preds = model.predict(X_test)
    y_preds = np.round(y_preds)

    print("Classification Report: \n", classification_report(y_test, y_preds))

    return y_preds
```

Figure 4:28 - ANN Model Implementation

The above ANN model has been tested with different number of epochs i.e., 50, 100, 150, 500 with the same number of neurons, hidden layers, and activation method. Based on the result, best model is chosen with the highest accuracy value.

CHAPTER 4
EVALUATION

This chapter includes the evaluation of the study that has been carried out in previous stages. The proposed evaluation technique mainly concerns about the accuracy and the recall of the model that has been built. For evaluating the results that have been derived from both the regression and neural network models, the common approach of performance measurement has been followed. The selected common method is Confusion Matrix Evaluation.

4.1 Confusion Matrix

When a machine learning process using for classification problem, with the output of binary or more classes, this approach can be followed. This is considering 4 different combinations of the resulting/actual values and expected/predicted values.

Actual Value

		Positive 1	**Negative 0**
Predicted Value	**Positive 1**	**TP**	**FP**
	Negative 0	**FN**	**TN**

True Positive (TP): The predicted value is to be positive and the result is true

True Negative (TN): The predicted value is to be negative and the result is true

False Positive (FP): The predicted value is to be negative and the result is false

False Positive (FN): The predicted value is to be positive and the result is false

Here, the actual values are identified as True/False while the predicted values are defined using Positive/Negative and the FP is referred as Type 1 error and FN is referred as Type 2 error. These terms are useful when dealing with the measurements with regards to confusion matrix such as Specificity, Accuracy, Recall, Precision, F-measure and AUC-ROC curves.

4.1.1 Confusion Matrix Evaluation Metrics

There are specific metrics that can be derived from a confusion matrix in different aspects. Regardless the problem scenario, nature or the problem domain, these metrics are elaborating some of the important values for particular problem study. The following listing is about the metrics that can be derived from a confusion matrix.

a) Precision

$$\boldsymbol{Precision = \frac{TP}{TP + FP}}$$

Precision is measuring the actual positive result values out of all the correctly predicted values. Precision is a useful metric in cases where False Positive is a higher concern than False Negatives (Bhandari, 2020). Higher precision value is expected to be a better model.

b) Recall

$$\boldsymbol{Recall = \frac{TP}{TP + FN}}$$

Recall describes, actual positive values that have been able to predict by using the built prediction model correctly. Recall is a useful metric in cases where False Negative trumps False Positive (Bhandari, 2020). Higher recall value is expected to be a better model.

c) Accuracy

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

Accuracy gives all the correct predictions that have been made regardless the positivity or negativity from all the classes. Higher accuracy value is expected to be a better model.

d) F1- Score/F1-Measure

$$\boldsymbol{F1\,score = \frac{2 * Recall * Precision}{Recall + Precision}}$$

Whenever the above values are in different ranges, i.e., higher recall with a lower precision or vice versa, it is difficult to come to conclusions and get a proper idea whether the model is performing well or not. In such scenarios, F1 score or F measure gives a harmonic means in arithmetic means term which is applicable for both precision and recall comparison at once.

4.2 Evaluation of Logistic Regression Model

As mentioned above as well, confusion matrix is a form of summarizing the final values of a classification problem in a tabular format. The final countdown is categorised based on the total amount of accurate and inaccurate prediction class-wise.

The following Figure 5:1 summarizes the final calculated confusion matrix in a heat map of the Logistic Regression.

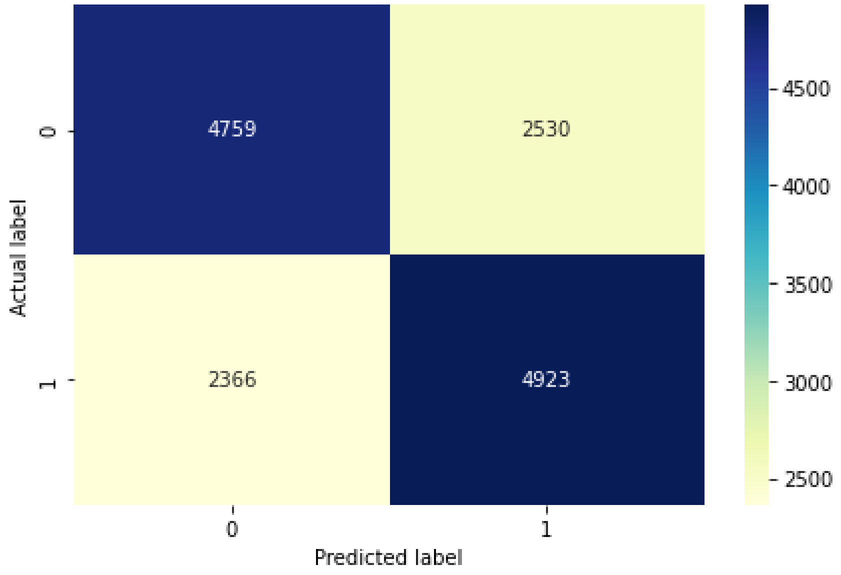
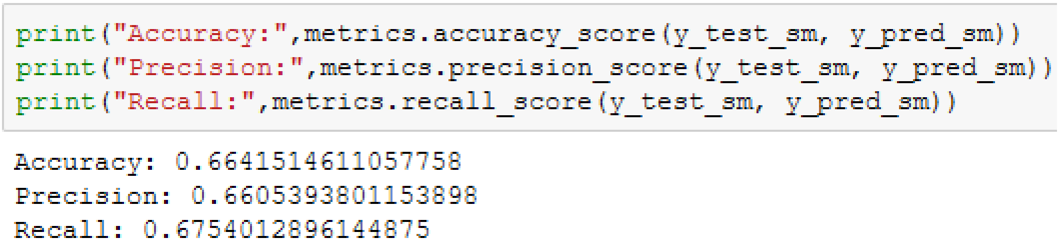


Figure 5:1 - Logistic Regression Confusion Matrix

The confusion matrix is depicting a promising amount of true predicted accurate values compared to the predicted incorrect values.

4.2.1 Evaluation Metrics of Logistic Regression

According to the above resulted confusion matrix, the evaluation measurements have been calculated as the below code segment in the Figure 5:2 which is followed by the final metrics values.

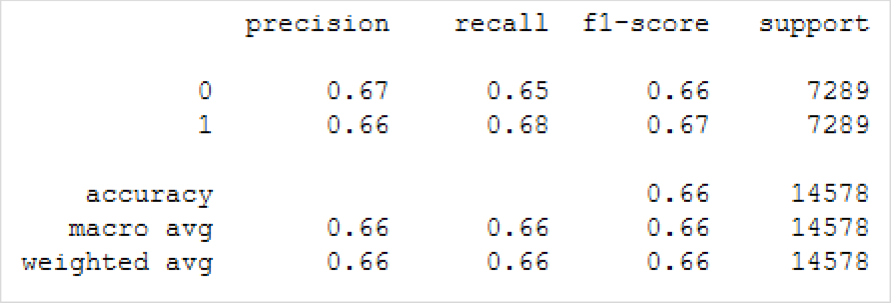


```
print("Accuracy:",metrics.accuracy_score(y_test_sm, y_pred_sm))
print("Precision:",metrics.precision_score(y_test_sm, y_pred_sm))
print("Recall:",metrics.recall_score(y_test_sm, y_pred_sm))
```

```
Accuracy: 0.6641514611057758
Precision: 0.6605393801153898
Recall: 0.6754012896144875
```

Figure 5:2 - Logistic Regression Evaluation Metrics

Since these values are referred generally for the test data values and predicted values only, the more systematic and complete approach for the evaluation is the classification report.



```
              precision    recall  f1-score   support

           0       0.67      0.65      0.66      7289
           1       0.66      0.68      0.67      7289

    accuracy                           0.66     14578
   macro avg       0.66      0.66      0.66     14578
weighted avg       0.66      0.66      0.66     14578
```

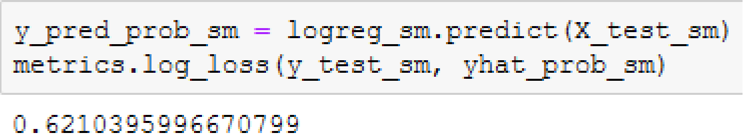
Table 5:1 - Classification Report for Logistic Regression

The above Table 5:1 illustrate that the trained logistic regression model gives the f1-score 0.66 i.e., 66% of accuracy while giving a recall value as 65% for class 0 i.e., the non-churned customers and 67% value for the class 1 i.e., the churned customers.

As the final measure, it is fair to assume that this model would appropriately work with a prediction rate of 66% for churn rate early identification.

4.2.2 Log Loss Evaluation

When it comes to logistic regression, in this scenario the probability results the churn rate of the customer which lies between 0 and 1 Log loss (Logarithmic loss) evaluates the performance of the particular classifier when the predicted output also lies between 0 and 1 which is a probability value. For the generated logistic regression model has a value of 0.62 log loss as in the Figure 5:3 which indicates an average strength as a classifier to be used.



```
y_pred_prob_sm = logreg_sm.predict(X_test_sm)
metrics.log_loss(y_test_sm, yhat_prob_sm)
```

```
0.6210395996670799
```

Figure 5:3 - Log loss evaluation of Logistic Regression

4.2.3 Area Under the Curve(AUC) Score Evaluation

AUC or Receiver Operating Characteristics (ROC) score also about the classifier’s strength or perfection. The value is plotted True Positive (TP) rate against the False Positive (FP) rate which indicates the deviation between the two measures.

After model generating with a balanced data set, as the final evaluation of AUC score for this scenario is 7156035530604539 as showing in the Figure 5:4.

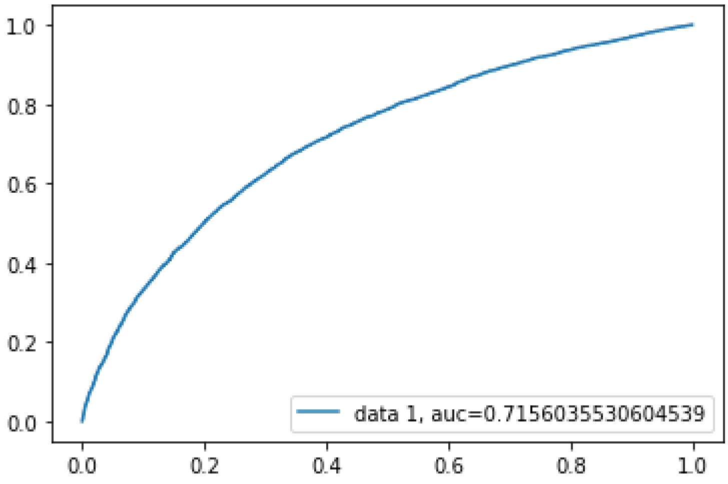


Figure 5:4 - AUC of the classifier

An AUC score 1 represents a perfect classifier for a given scenario and this model has generated 0.72 score which is a nearly better classifier to be used according to AUC.

4.3 Evaluation of Deep Neural Network Model

Similar to the model evaluation logistic regression in previous section, the trained deep learning model also need to be evaluated. For this also the derived confusion matrix is evaluated both with imbalanced and balanced data set with different epochs counts.

4.3.1 Evaluation Metrics of 50 epochs ANN

The following Figure 5:5 summarizes the final calculated confusion matrix in a heat map of the Logistic Regression. The corresponding numerical report calculated below tabulated in Table 5:2 is the classification report of the ANN model with 50 epochs and it has been selected as the initial number of epochs for training the model.

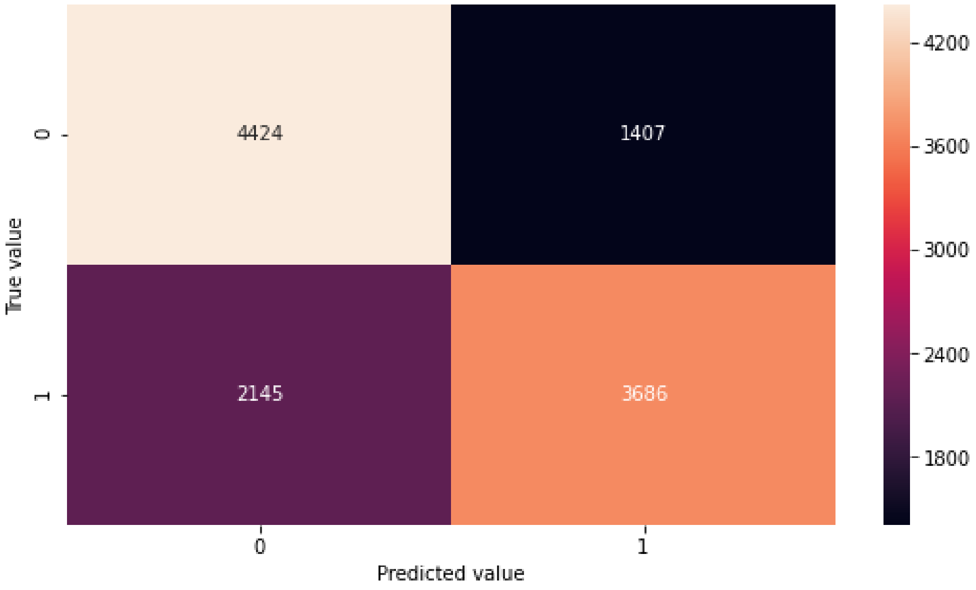
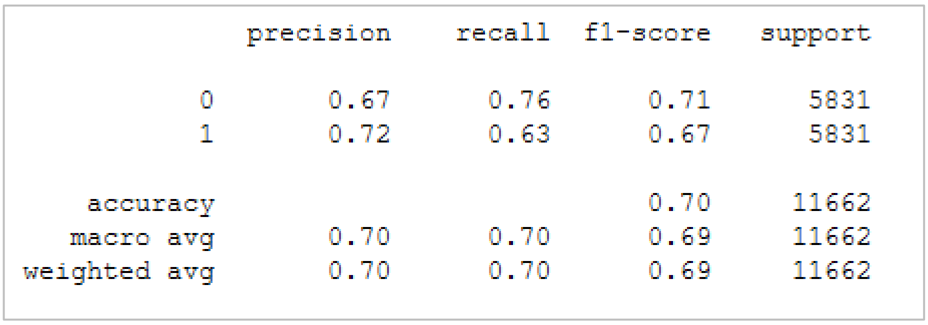


Figure 5:5 - ANN Model 50 Epochs Confusion Matrix

The model with 50 epochs has generated the accuracy rate of 70% as of f1-score which indicates a considerable good value to be considered.



```
              precision    recall  f1-score   support

           0       0.67      0.76      0.71      5831
           1       0.72      0.63      0.67      5831

    accuracy                           0.70     11662
   macro avg       0.70      0.70      0.69     11662
weighted avg       0.70      0.70      0.69     11662
```

Table 5:2 - Classification Report for ANN model 50 epochs

4.3.2 Evaluation Metrics of 100 epochs ANN

The following Figure 5:6 summarizes the final calculated confusion matrix in a heat map of the Logistic Regression. The corresponding numerical report calculated below tabulated in Table 5:3 is the classification report of the ANN model with 100 epochs and it has been selected as the initial number of epochs for training the model.

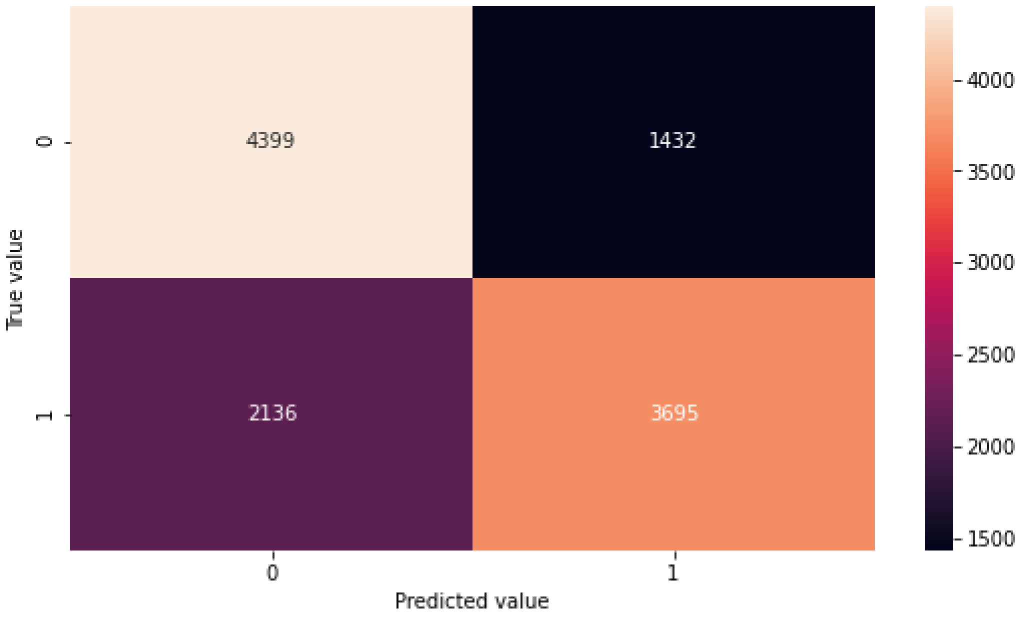
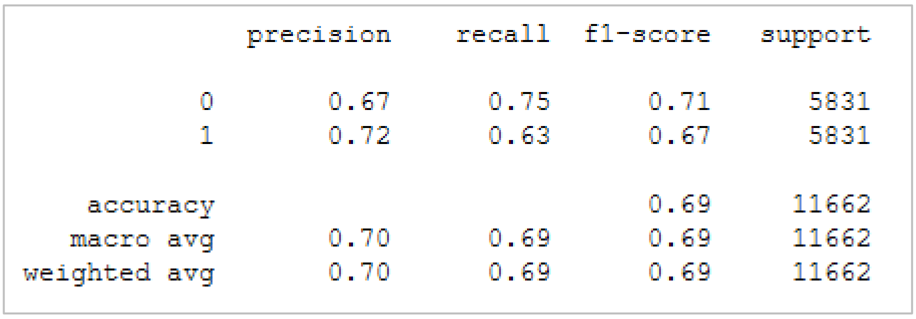


Figure 5:6 - ANN Model 100 Epochs Confusion Matrix

The model with 100 epochs has generated the accuracy rate of 69% as of f1-score which indicates a minor deduction of accuracy when it gets doubles the epochs count compared to previous step.



```
              precision    recall  f1-score   support

           0       0.67      0.75      0.71      5831
           1       0.72      0.63      0.67      5831

    accuracy                           0.69     11662
   macro avg       0.70      0.69      0.69     11662
weighted avg       0.70      0.69      0.69     11662
```

Table 5:3- Classification Report for ANN model 100 epochs

4.3.3 Evaluation Metrics of 150 epochs ANN

The following Figure 5:7 summarizes the final calculated confusion matrix in a heat map of the Logistic Regression. The corresponding numerical report calculated below tabulated in Table

5:4 is the classification report of the ANN model with 150 epochs and it has been selected as the initial number of epochs for training the model.

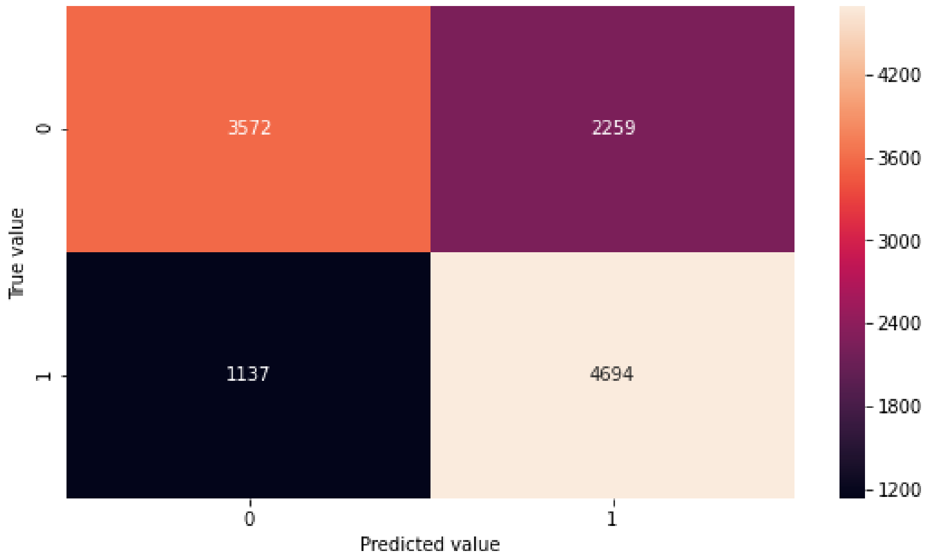
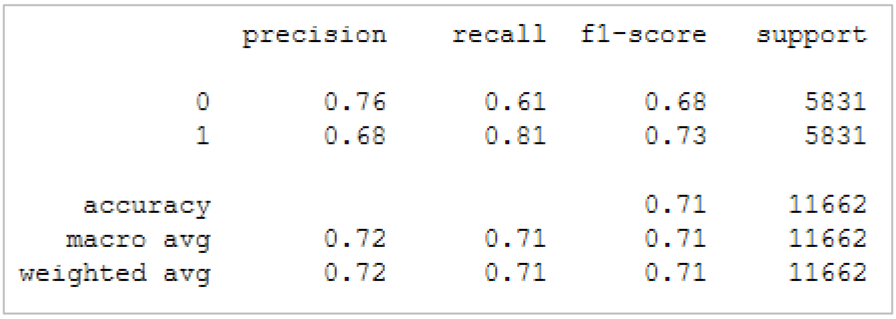


Figure 5:7 - ANN Model 150 Epochs Confusion Matrix

The model with 150 epochs has generated the accuracy rate of 71% as of f1-score which indicates a minor increase of accuracy when it gets doubles the epochs count compared to previous step.



```
              precision    recall  f1-score   support

           0       0.76      0.61      0.68      5831
           1       0.68      0.81      0.73      5831

    accuracy                           0.71     11662
   macro avg       0.72      0.71      0.71     11662
weighted avg       0.72      0.71      0.71     11662
```

Table 5:4 - Classification Report for ANN model 150 epochs

4.3.4 Evaluation Metrics with 500 epochs ANN

The following Figure 5:8 summarizes the final calculated confusion matrix in a heat map of the Logistic Regression. The corresponding numerical report calculated below tabulated in Table 5:5 is the classification report of the ANN model with 500 epochs and it has been selected as the initial number of epochs for training the model.

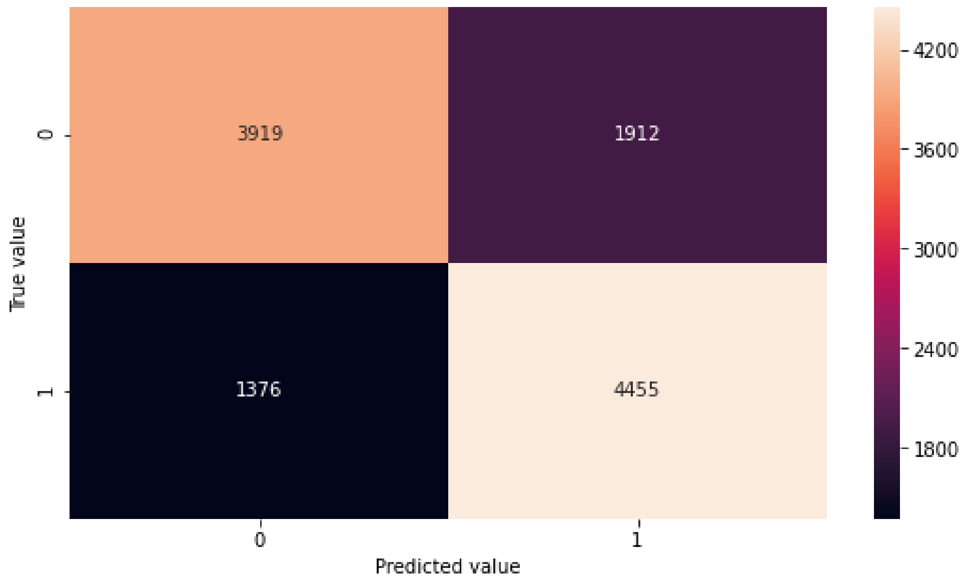
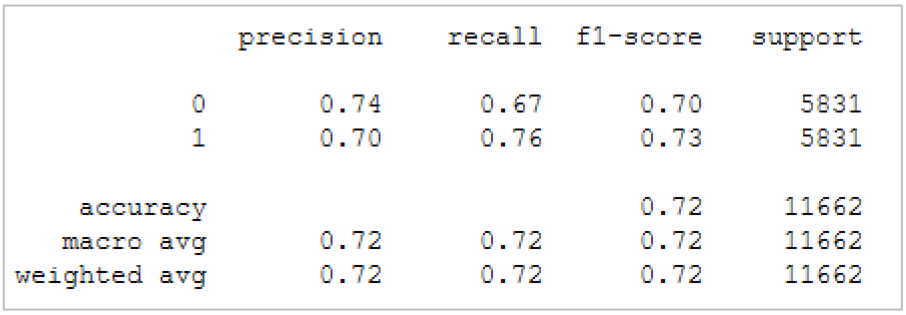


Figure 5:8 - ANN Model 500 Epochs Confusion Matrix

The model with 500 epochs has generated the accuracy rate of 72% as of f1-score which indicates a minor increase of accuracy when it gets doubles the epochs count compared to previous step.



```
              precision    recall  f1-score   support

           0       0.74      0.67      0.70      5831
           1       0.70      0.76      0.73      5831

    accuracy                           0.72     11662
   macro avg       0.72      0.72      0.72     11662
weighted avg       0.72      0.72      0.72     11662
```

Table 5:5 - Classification Report for ANN model 500 epochs

4.3.5 Evaluation Metrics with 1000 epochs ANN

The following Figure 5:9 summarizes the final calculated confusion matrix in a heat map of the Logistic Regression. The corresponding numerical report calculated below tabulated in Table 5:5 is the classification report of the ANN model with 1000 epochs and it has been selected as the initial number of epochs for training the model.

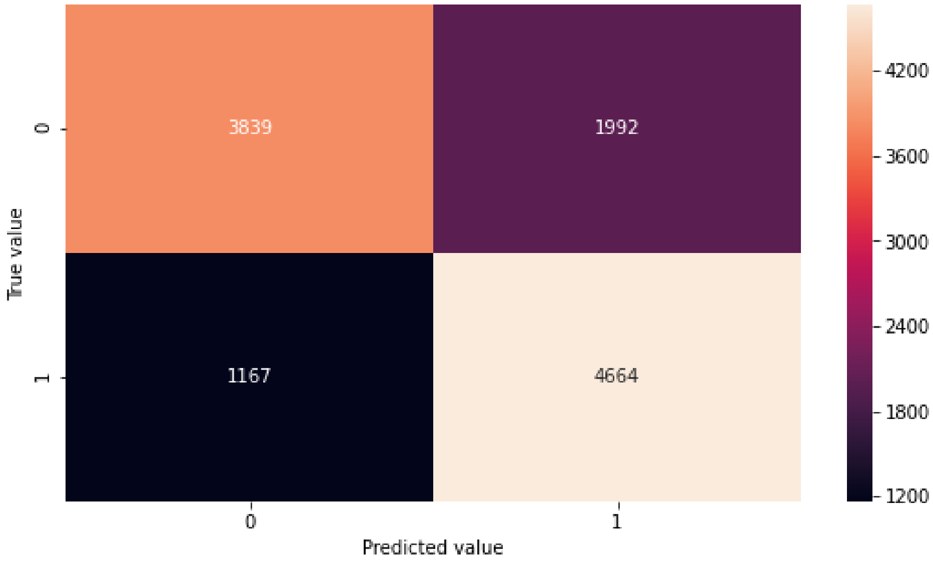
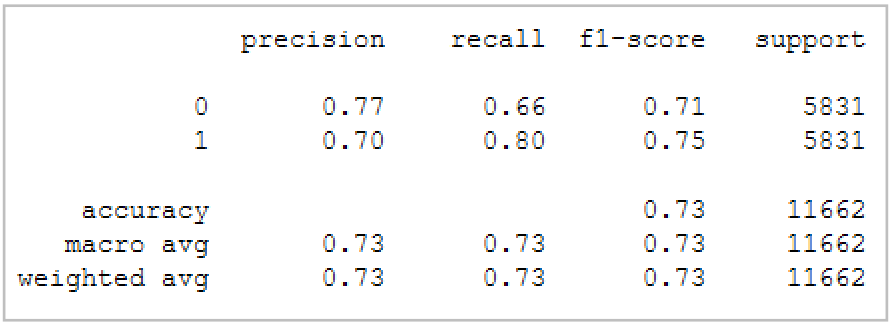


Figure 5:9 - ANN Model 1000 Epochs Confusion Matrix

The model with 1000 epochs has generated the accuracy rate of 73% as of f1-score which indicates a minor increase of accuracy when it gets doubles the epochs count compared to previous step.



```
              precision    recall  f1-score   support

           0       0.77      0.66      0.71      5831
           1       0.70      0.80      0.75      5831

    accuracy                           0.73     11662
   macro avg       0.73      0.73      0.73     11662
weighted avg       0.73      0.73      0.73     11662
```

Figure 5:10 - Classification Report for ANN model 500 epochs

4.3.6 Evaluation Comparison of the ANN Models

The below Figure 5:11 is a graph of the f1-score values against the different number of epochs and even though the number of epochs has been doubled in value, the improvement of f1-core is not that significant. So, as per the final result, it can be considered the f1 score for the ANN model that has been developed as 0.73.

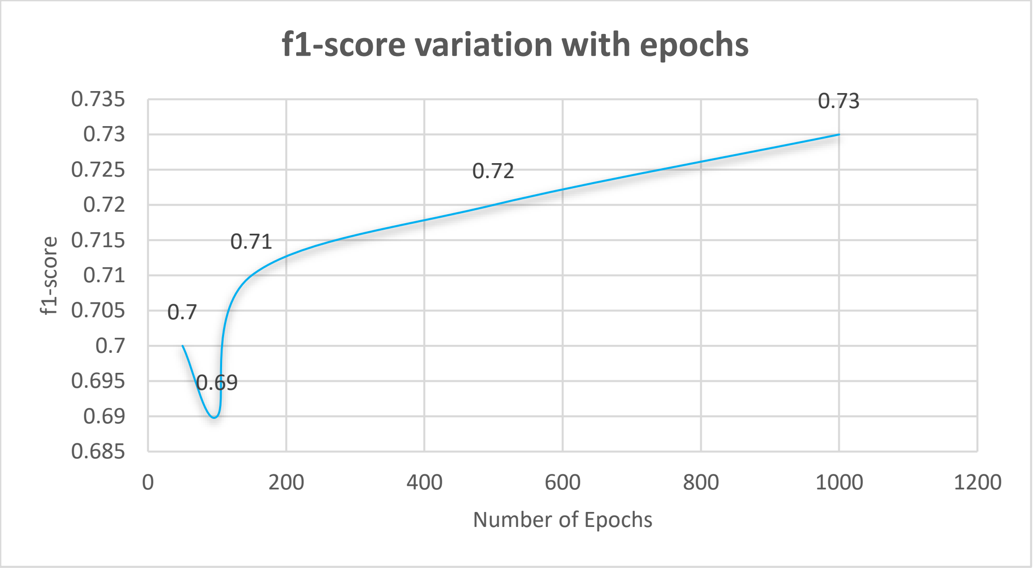


Figure 5:11 - Graph of f1-score against epochs

4.4 Evaluation of Models Built

As per the final step of the evaluation, the comparison of the built models has been made as follows and when considering the overall readings tabulated in the Table 5:6, the ANN model has represented more accuracy of 73% and recall of 80% and 60% which indicates the reusability of the built models. The reusability of the model for 'churn' prediction which has a higher percentage can be used for identifying the abandonment of the customers from a business beforehand by studying the underling behavioural factors.

	Regression Model	ANN Model
Iterations	1000	1000
f1-score	0.66	0.73
recall – Churn(1)	0.67	0.8
recall – Non Churn(0)	.65	0.6

Table 5:6 - Final evaluation of the models

CHAPTER 5

DISCUSSION

Although the generated prediction models have resulted with favorable result as intended, the in between processes have been faced with number of diversions due to unforeseen and unpredicted reasons. In this chapter, these processes and the actions that have been taken are discussed with the evidence provided. As well as the possible risks and concerns that can be identified and addressed beforehand in order to minimize the rework and enhance the performance of the models are also intended to be discussed in detail.

5.1 Imbalanced vs. Balanced Data

The importance and the necessity are unquestionable for data imbalance since it is theoretically proved and examined with enough of studies. In this study, the issue of the data imbalance-ness has been caused for the model and it has only revealed in the model analysis. The following discussed scenarios are related to the segmented and selected customer base.

5.1.1 Logistic Regression Model Evaluation with Imbalanced Data

Before identifying the cause that is influenced in the selected customer base, the logistic model has been built, trained and test with the same criteria. Along with the final evaluation of confusion matrix in the Figure 6:1 below it could identify the effect of data imbalance for generating the expected result.

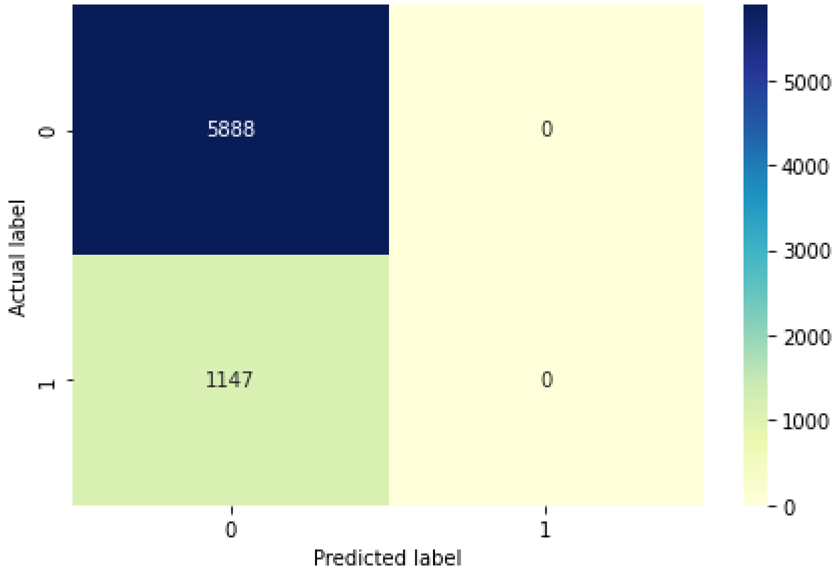
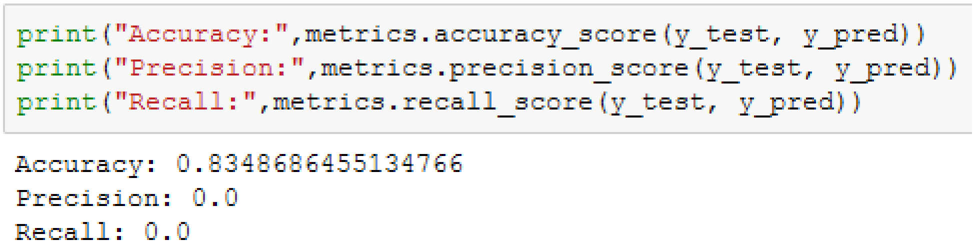


Figure 6:1 - Confusion matrix of logistic regression with imbalanced data

This result is biased towards the true prediction values, since the data set is biased in the same way towards the non-churned customer amount. When considering the evaluation metrics that have been generated based on the above confusion metrics, it has been given the below result as in the Figure 6:2.



```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8348686455134766
Precision: 0.0
Recall: 0.0
```

Figure 6:2 - Evaluation metrics of logistic regression with imbalanced data

Even though the accuracy is really high for the above model, the model reusability and the actual positive result are zero in value. If the reusability is none for a particular built model, it is not an approached to be followed and instead, suggesting another approach is better, if the reason is not an imbalanced data set.

As per the final evaluation unit, the AUC score also has been generated and the following Figure 6:3 describes the final result of the Sensitivity against the Specificity which is ended up with a value of 0.5 when 1.0 the perfect classifier indicates by the value 1.

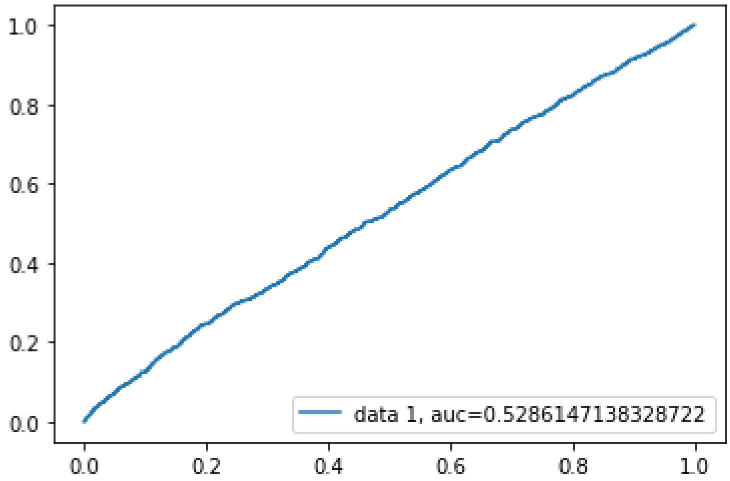


Figure 6:3 - AUC of the classifier for Imbalanced data set

Since the result is 50%, this classifier is not going to be useful for the expected model or reusable for another similar scenario as well.

5.1.2 Neural Network Model Evaluation with Imbalanced Data

Similar to the above discussed section 5.2.1, the deep neural network model also has been tested with the imbalanced data, before applying the data balancing techniques, here in which case has been used SMOTE for overcoming the effect causes for the study.

The Figure 6:4 blow depicted the spread of testing and training data set which is yet again thorough the biasness of positive data over the negative data values. This particular result has been generated with a model of 100 epochs with same criteria that has been applied for the model with balanced data set.

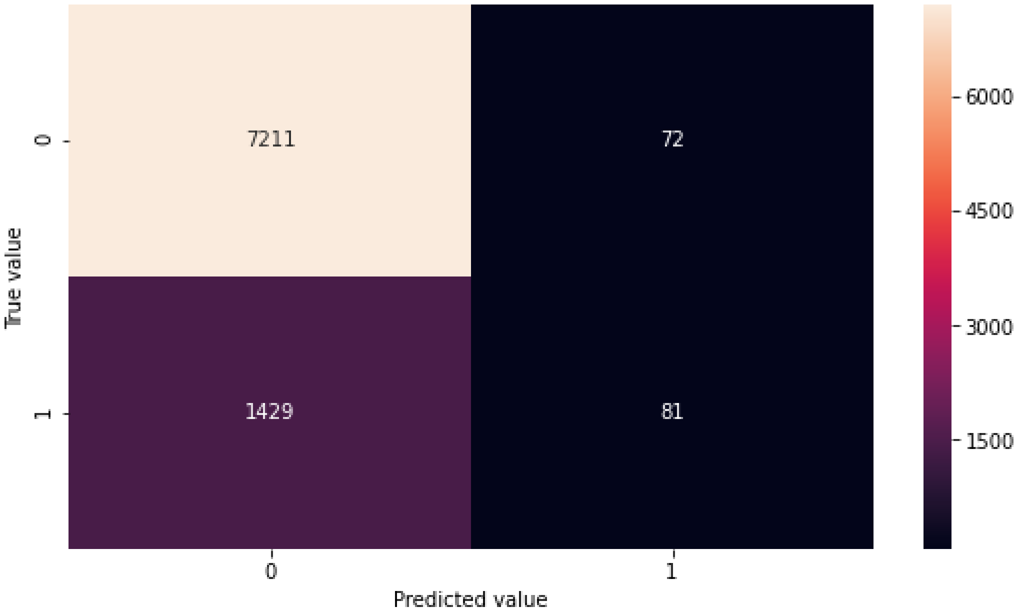
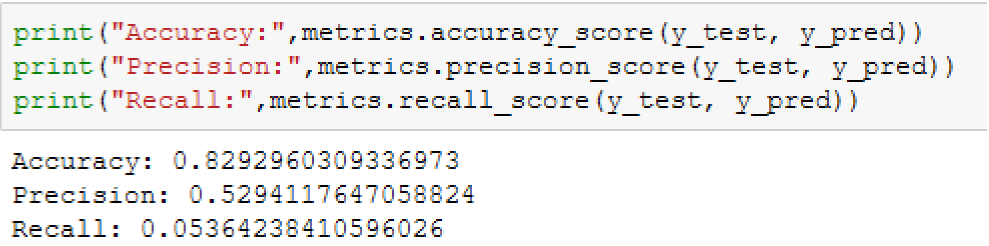


Figure 6:4 - Confusion matrix of Deep neural network with imbalanced data

The corresponding evaluation metrics for the above Figure 6:4 as following Figure 6:4 and it also indicates an accuracy of 83% which is higher and the precision and reusability is about 50% which does not indicate any significance to be used and derived by training a model. Still the data is based towards to True predictions which will tend the model to be a less accurate one.



```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8292960309336973
Precision: 0.5294117647058824
Recall: 0.05364238410596026
```

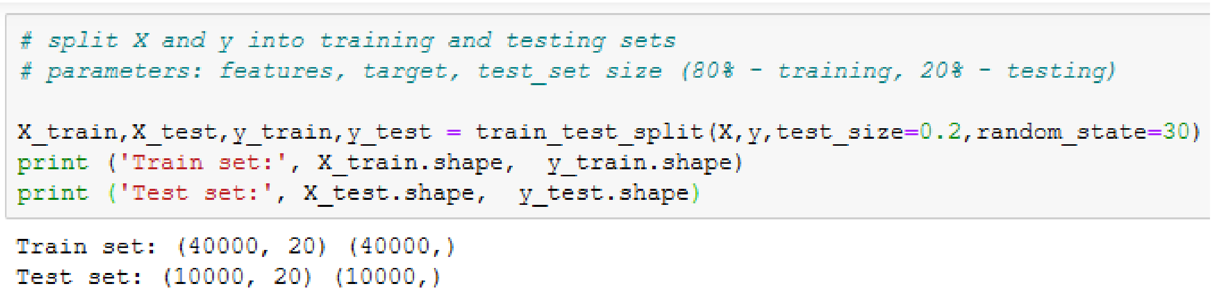
Figure 6:5 - Evaluation metrics of deep neural network with imbalanced data

5.2 Study of the Total Customer Base

This research study's predictive model designing and development is done solely based on the customer segmentation. It is fair to have the question of why the segmentation is done and what if there is no segmentation has been performed. For justifying those questionable facts, the study has focused and performed and evaluated some of the aspect that has been come across.

5.2.1 Logistic Regression Model Evaluation for Total Customer Base

The initial dataset of 50000 data records, have been included for the model building and the same set of code and implementation routine is conducted and the splitting of data is as following Figure 6:6.



```
# split X and y into training and testing sets
# parameters: features, target, test_set size (80% - training, 20% - testing)

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=30)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (40000, 20) (40000,)
Test set: (10000, 20) (10000,)
```

Figure 6:6 - Splitting the total dataset for training and testing

The total dataset has been split to the 4:1 ratio for training the model and the same model which has been used for the segmented customer data set is used and the final result of the confusion matrix was as the following Figure 6:7.

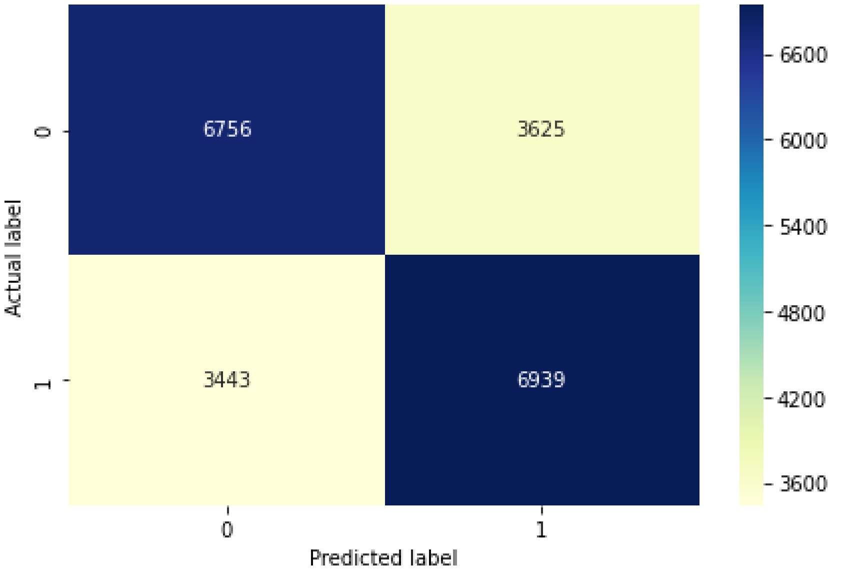
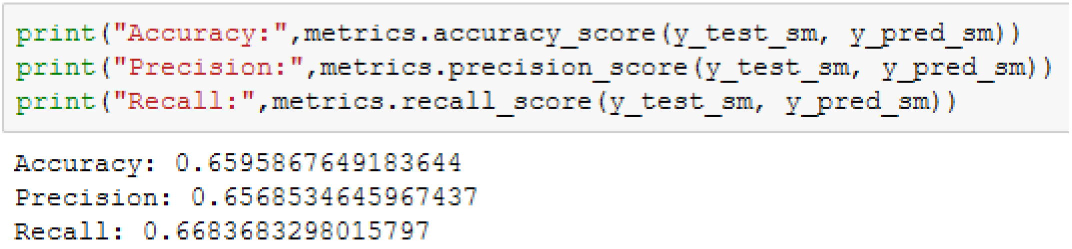


Figure 6:7 - Confusion matrix of the total customer base

The result of the logistic regression model built for the total customer base has derived the result mentioned as the following Figure 6:8 and there is a quite interesting readings worthy to be evaluated and considered.



```
print("Accuracy:",metrics.accuracy_score(y_test_sm, y_pred_sm))
print("Precision:",metrics.precision_score(y_test_sm, y_pred_sm))
print("Recall:",metrics.recall_score(y_test_sm, y_pred_sm))
```

```
Accuracy: 0.6595867649183644
Precision: 0.6568534645967437
Recall: 0.6683683298015797
```

Figure 6:8 - Evaluation metrics for total customer base

According to the evaluation metrics, this model's accuracy, reusability and predictability are similar to the segmented model in every aspect with a small deviation. Yet still, the biasness is not here towards the non-churned data since the data set is balance using the SMOTE model here as well. In this case, the significant difference has not been identified between these two approaches even though it could be verified with another similar set of data.

5.2.2 Neural Network Model Evaluation for Total Customer Base

Following the same approach that has been applied to the segmented customer base, the total customer base also was evaluated and following indicates the confusion matrix that has been resulted with 50 epochs.

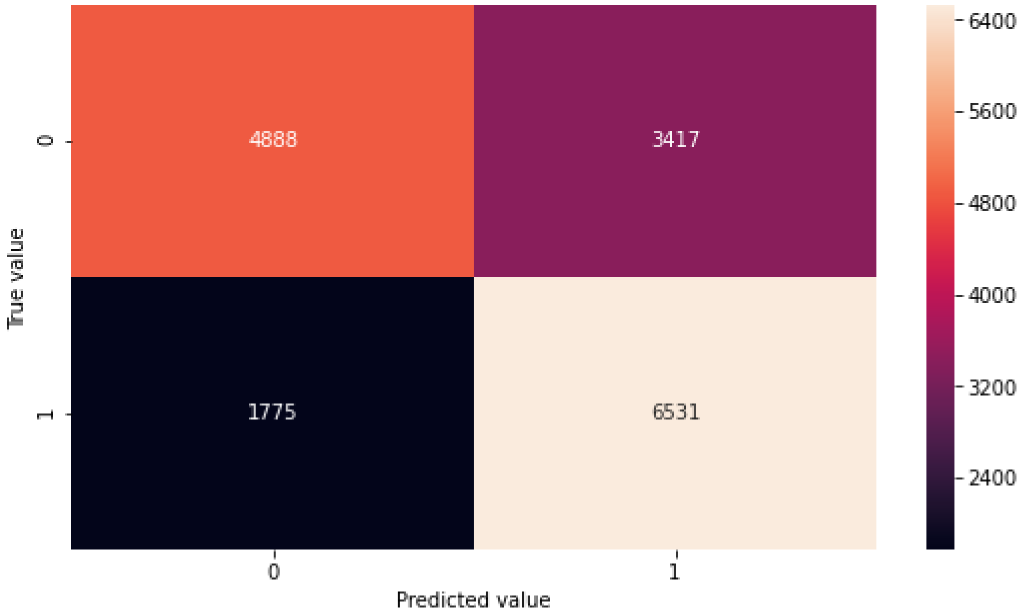
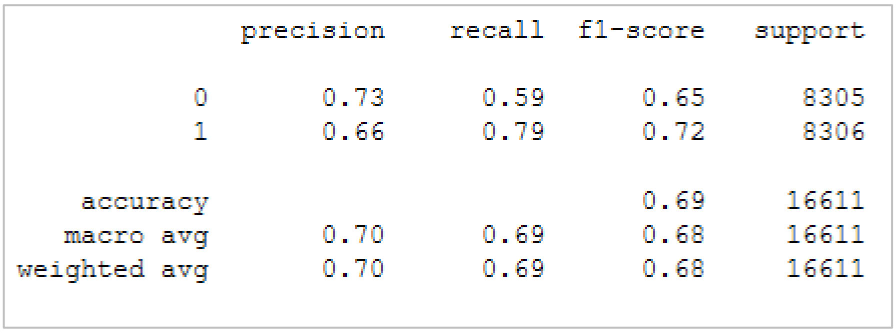


Figure 6:9 - Confusion matrix of deep neural network for total customer base

The above confusion matrix has generated an evaluation metrics tabulated as the following Table 6:1, and there are quite interesting readings that need to pay the attention.



```
              precision    recall  f1-score   support

           0       0.73      0.59      0.65      8305
           1       0.66      0.79      0.72      8306

    accuracy                           0.69     16611
   macro avg       0.70      0.69      0.68     16611
weighted avg       0.70      0.69      0.68     16611
```

Table 6:1 - Evaluation metrics of deep neural network for total customer base

The above table contains the reading for neural model trained for only 50 epochs and it is little lower than the segmented model that trained exactly with 50 epochs. Hence it is mindful to consider training the model with same number of epochs for better and complete evaluation.

When looking at the above observations, strategies and comparative experiments, it provides strong back ground with a proper backing up for the concepts that has been studied through this research. There are so many aspects that can be considered and evaluated with different measurements as well and some of those areas have been covered in this research study as discussed in this chapter. It would be facilitating handling those edge cases with the above-mentioned strategies and techniques followed.

CHAPTER 5
CONCLUSION

With the intention of the studying another aspect of customer abandonment or customer churn, this research has been initiated. The result has been critically evaluated and discussed thoroughly in the previous chapter. Finally, the research work that has been done so far has led to the point where the decision making.

When considering the topic "customer churn", most of the studies have addressed this in different aspect without any boundary. For instance, the fields such marketing, business, telecommunication banking and technology have paid their attention to this topic. Each and every industry and field is having their own definition and intention of this concern. This chapter will discuss the ultimate decisions that can be made based up on the gathered result.

6.1 Critical Review

As per the plan of this study, the whole procedure is modularized as phase based for better organizing so that the intended process is addressed properly. Since an incomplete or error some procedure leads the study for error some result, it should be verified that necessary aspects are included and evaluated.

From the initial step of data-cleansing to the final step of evaluation, it has been attempted to achieve the initially set objectives were evaluated consecutively. One of those objectives has been to evaluate the behavioural factors of the customers against the abandonment which will be providing a prepared background to initiate the study from. The chosen data set, which consisting the behaviours such as frequency of purchases, satisfaction score and loyalty points have been utilized for identifying this objective. Even though the intended attributes were not included in the data set, the above-mentioned attributes have been exposing the behavioural factors indirectly. Besides, the objective of discovering the nature of the abandonment was intended to be evaluated using the access logs level data, yet it has to be elicited using the attributes number of hours spent app, number of devices and preferred device. Consequently, the attributes have to be mapped to whatever the most suitable set of attributes that are available in the data set. Both of the mentioned objectives have been gained with feature engineering and RFM analysis in the first phase of the study; the Customer Segmentation.

When considering rest of the objectives which are directly associated with the second phase of the study; the Predictive Modelling. In here, the ideal data transformation strategies selection, evaluation criteria selection and the discovery of the limitations and constraints have been tested. Accordingly, the predictive modelling phase has provided a workspace and a platform to test these each and every objective with different approaches. Since there are two model implementations for the evaluation of predictive modelling, the researching area has been wider and it has led to more effective evaluation for the same scenario with different aspects.

Yet some of the area that was intended to identify in deeper understanding, have not been addressed by the evaluation and modelling process. The nature of the abandonment by the user based on the activity and technical action has been a main intention to be revealed. For this action all the data from the access level, user level, and authorization level is needed. Based on this type of secure and critical data, much more accurate modelling and evaluation could be done.

Hence, as an overall decision it can be concluded that the intended objectives have been addressed up to a considerable level with a good measure of accuracy as well.

6.2 Lesson Learnt

Even though not being novel by today, the Data Science, Business Statistics, Analytics and Machine Learning fields are getting well recognized attention in the industries irrespective to the industry nature. For applying the theoretical knowledge that has gained so far, practical approach and scenario is a must have. This research study has led us to most of the subject areas to be utilized and applied practically for real world scenarios. Just having only the knowledge is not helping after some point, since the real application needs to be evaluated with proper methods, tool selection and finally integration of them.

The same dataset will be behaving in different ways when applying the same strategy and evaluation method. These approaches are need to be thoroughly criticized first according to the theory before applying for a scenario. In this study, it was needed to perform the machine learning modelling process due to the imbalance-ness of the selected data set. So, these kind of situation needs to be identified beforehand in order to avoid any inconvenience and reworking.

Utilizing different technique approaches for each specific step in the process should be a mindful task. The compatibility of the result, integration of the result in between two different processes, conversion of result per necessity while protecting the existing data are the guidelines to end up with a proper and successful research study. The total outcome is measured as a whole of everything above mentioned. Hence, this research study has been a persuasion of a completion process from data, to knowledge discover in a logical and acceptable manner.

6.3 Future Work

The followed research study can be extended in the following aspects;

- Prediction modelling with a similar and different techniques, algorithm and approaches to evaluate the strength of the underlying concept
- Utilizing a dataset with a different industry or field
- Comparison of the same model implementations without handling the customer segmentations
- Applying the same model implementation for different set of customer segment selection
- Implementing an application programming Interface (API) for training similar type of models with selected attributes in a user-friendly manner. This option will enhance the usability of the model with non-technical users as well.

The above-mentioned future extensions will lead to significant improvements or deviations which will give more opportunity for new aspects as well.

APPENDIX

Appendix: A SQL Scripts Generated for Data Cleansing

Data cleansing and transforming of preferred category variable

```
use churn_data;
SELECT * FROM churn_data.user_churn;
select distinct(pref_category) FROM ii_research.user_churn;
SET SQL_SAFE_UPDATES=0;

-- Cleansing (Normalising the values with multiple representations)
UPDATE user_churn
SET pref_category='Mobile'
WHERE pref_category='Mobile Phone';

-- Cleansing (Normalising the values with multiple representations)
UPDATE user_churn
SET pref_category='Laptop&Accessory'
WHERE pref_category='Laptop & Accessory';

-- Create the table to track enumeration of values - device
CREATE TABLE pref_category (
id INT PRIMARY KEY,
value VARCHAR(25)
);

-- Add new column to hold the enumrated value
ALTER TABLE user_churn
ADD COLUMN preferred_category INT AFTER pref_category;

-- Set the value of newly added column for each distinct category
UPDATE user_churn
SET preferred_category = CASE
WHEN pref_category = 'Mobile' THEN 1
WHEN pref_category = 'Fashion' THEN 2
WHEN pref_category = 'Others' THEN 3
WHEN pref_category = 'Grocery' THEN 4
ELSE 5
END;

-- Extract enumerated key and value and insert them to the new table
INSERT INTO pref_category(id, value)
SELECT DISTINCT(preferred_category), pref_category FROM user_churn;

-- Drop the text based category column
ALTER TABLE user_churn
DROP COLUMN pref_category;
```

Data cleansing and transforming of Marital Status variable

```
use churn_data;
SELECT * FROM churn_data.user_churn;
select distinct(marital_status) FROM ii_research.user_churn;
SET SQL_SAFE_UPDATES=0;

-- Create the table to track enumeration of values - device
CREATE TABLE marital_status (
id INT PRIMARY KEY,
value VARCHAR(25)
);

-- Add new column to hold the enumrated value
ALTER TABLE user_churn
ADD COLUMN marital_status_val INT AFTER marital_status;

-- Set the value of newly added column for each distinct category
UPDATE user_churn
SET marital_status_val = CASE
WHEN marital_status = 'Single' THEN 1
WHEN marital_status = 'Married' THEN 2
ELSE 3
END;

-- Extract enumerated key and value and insert them to the new table
INSERT INTO marital_status(id, value)
SELECT DISTINCT(marital_status_val), marital_status FROM user_churn;

-- Drop the text based category column
ALTER TABLE user_churn
DROP COLUMN marital_status;
```

Data cleansing and transforming of Login Device variable

```
select distinct(preferred_login_device) FROM ii_research.user_churn;

UPDATE user_churn
SET preferred_login_device='Phone'
WHERE preferred_login_device='Mobile Phone';

-- Create the table to track enumeration of values - device
CREATE TABLE pref_login_device (
id INT PRIMARY KEY,
value VARCHAR(25)
);

-- Add new column to hold the enumrated value
ALTER TABLE user_churn
ADD COLUMN pref_login_device INT AFTER preferred_login_device;

-- Set the value of newly added column for each distinct category
UPDATE user_churn
SET pref_login_device = CASE
WHEN preferred_login_device = 'Computer' THEN 1
WHEN preferred_login_device = 'Phone' THEN 2
END;

-- Extract enumerated key and value and insert them to the new table
INSERT INTO pref_login_device(id, value)
SELECT DISTINCT(pref_login_device), preferred_login_device FROM user_churn;

-- Drop the text based category column
ALTER TABLE user_churn
DROP COLUMN preferred_login_device;
```

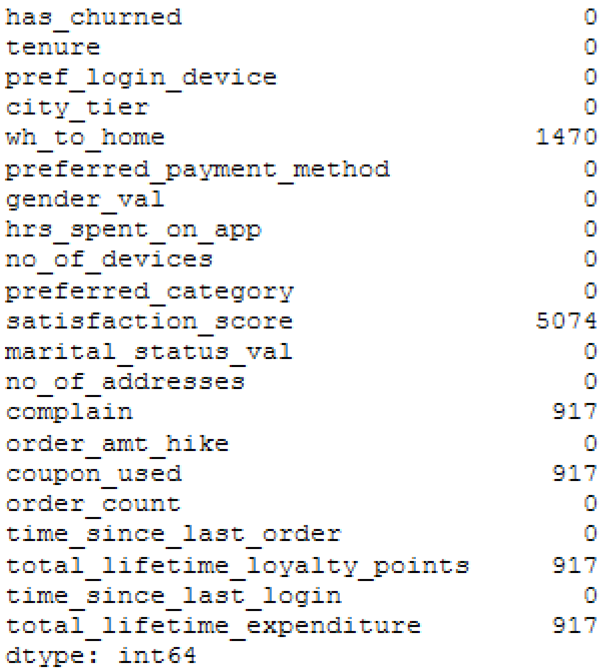
Data cleansing and transforming of Payment Method variable

```
use ii_research;
SELECT * FROM ii_research.user_churn;
```

```sql
select distinct(pref_payment_method) FROM ii_research.user_churn;
SET SQL_SAFE_UPDATES=0;

UPDATE user_churn
SET pref_payment_method='COD'
WHERE pref_payment_method='Cash on Delivery';

-- Cleansing (Normalising the values with multiple representations) --
payment method
UPDATE user_churn
SET pref_payment_method='CreditCard'
WHERE pref_payment_method='CC';

-- Cleansing (Normalising the values with multiple representations) --
payment method
UPDATE user_churn
SET pref_payment_method='CreditCard'
WHERE pref_payment_method='Credit Card';

-- Cleansing (Normalising the values with multiple representations) --
payment method
UPDATE user_churn
SET pref_payment_method='DebitCard'
WHERE pref_payment_method='Debit Card';

-- Cleansing (Normalising the values with multiple representations) --
payment method
UPDATE user_churn
SET pref_payment_method='Ewallet'
WHERE pref_payment_method='E wallet';

-- Add new column to hold the enumrated value  -- payment method
ALTER TABLE user_churn
ADD COLUMN preferred_payment_method INT AFTER pref_payment_method;

-- Create the table to track enumeration of values -- payment method
CREATE TABLE pref_payment_method (
id INT PRIMARY KEY,
value VARCHAR(25)
);

-- Set the value of newly added column for each distinct category-- payment
method
UPDATE user_churn
SET preferred_payment_method = CASE
WHEN pref_payment_method = 'DebitCard' THEN 1
WHEN pref_payment_method = 'UPI' THEN 2
WHEN pref_payment_method = 'CreditCard' THEN 3
WHEN pref_payment_method = 'Ewallet' THEN 4
WHEN pref_payment_method = 'COD' THEN 5
END;

INSERT INTO pref_payment_method(id, value)
SELECT DISTINCT(preferred_payment_method), pref_payment_method FROM
user_churn;

-- Drop the text based category column
ALTER TABLE user_churn
DROP COLUMN pref_payment_method;
```

Appendix: B   Missing Value Handling with Mean Imputation

```
customers.isnull().sum()
```

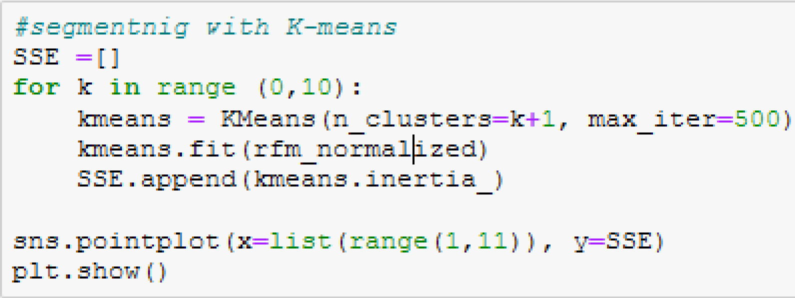


```
has_churned                          0
tenure                               0
pref_login_device                    0
city_tier                            0
wh_to_home                        1470
preferred_payment_method             0
gender_val                           0
hrs_spent_on_app                     0
no_of_devices                        0
preferred_category                   0
satisfaction_score                5074
marital_status_val                   0
no_of_addresses                      0
complain                           917
order_amt_hike                       0
coupon_used                        917
order_count                          0
time_since_last_order                0
total_lifetime_loyalty_points      917
time_since_last_login                0
total_lifetime_expenditure         917
dtype: int64
```

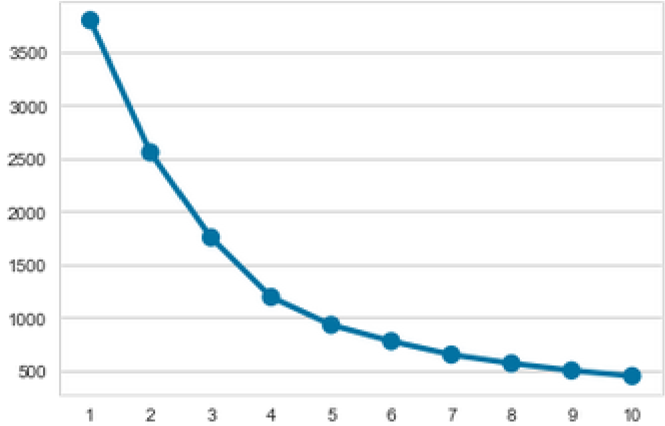
```
cols = ["wh_to_home", "satisfaction_score","complain",
"coupon_used","total_lifetime_loyalty_points","total_lifetime_expenditure"]
customers[cols]=customers[cols].fillna(customers.mean().iloc[0])
print(customers.shape)
customers.isnull().sum()
```

```
has_churned                          0
tenure                               0
pref_login_device                    0
city_tier                            0
wh_to_home                           0
preferred_payment_method             0
gender_val                           0
hrs_spent_on_app                     0
no_of_devices                        0
preferred_category                   0
satisfaction_score                   0
marital_status_val                   0
no_of_addresses                      0
complain                             0
order_amt_hike                       0
coupon_used                          0
order_count                          0
time_since_last_order                0
total_lifetime_loyalty_points        0
time_since_last_login                0
total_lifetime_expenditure           0
```

Appendix: C Segmentation by K-Means

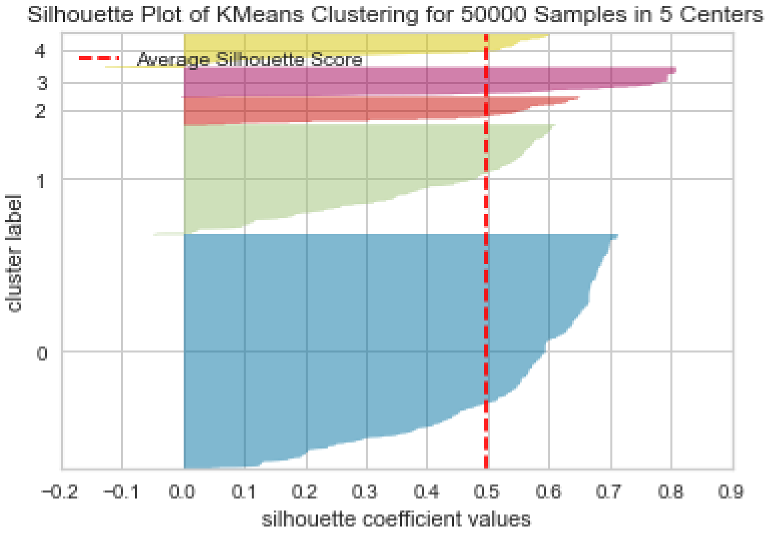


```
#segmentnig with K-means
SSE =[]
for k in range (0,10):
    kmeans = KMeans(n_clusters=k+1, max_iter=500)
    kmeans.fit(rfm_normalized)
    SSE.append(kmeans.inertia_)

sns.pointplot(x=list(range(1,11)), y=SSE)
plt.show()
```

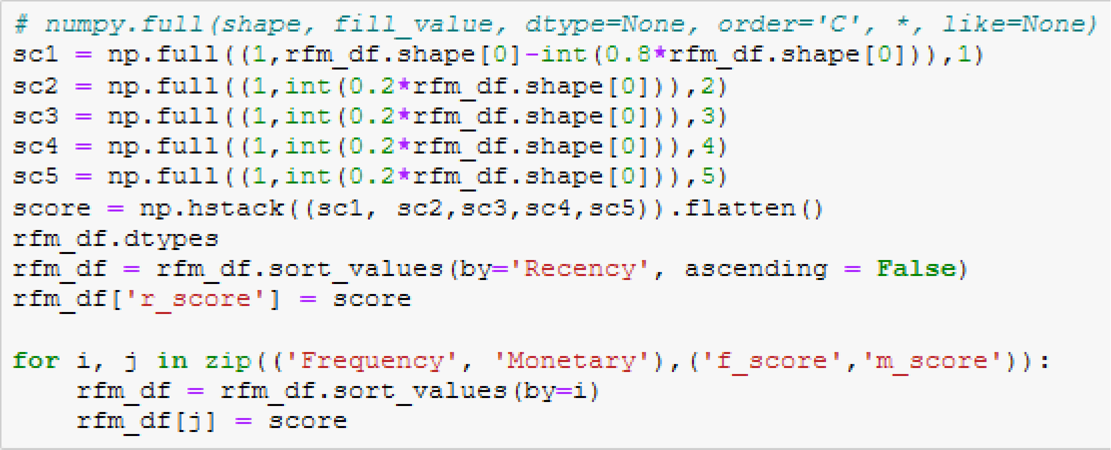


Appendix: D Clustering Total Customer Base by Silhouette Analysis

```
silhouette_visualizer(KMeans(5, random_state=42), rfm_normalized,
colors='yellowbrick')
```

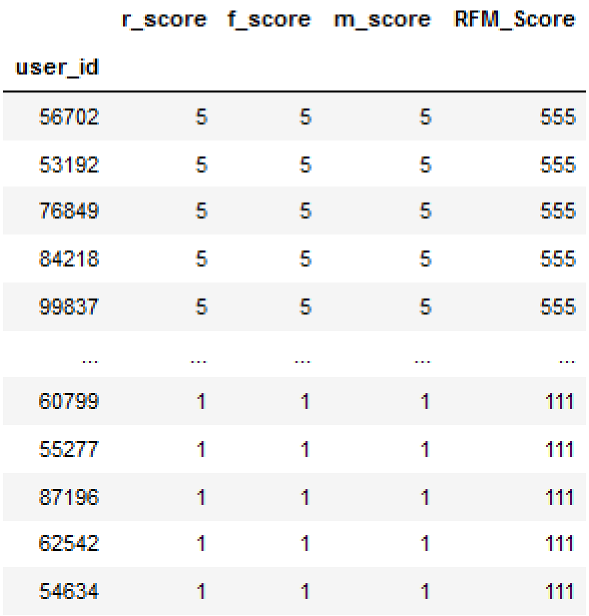


Appendix: E RFM Score Assigning for Clusters



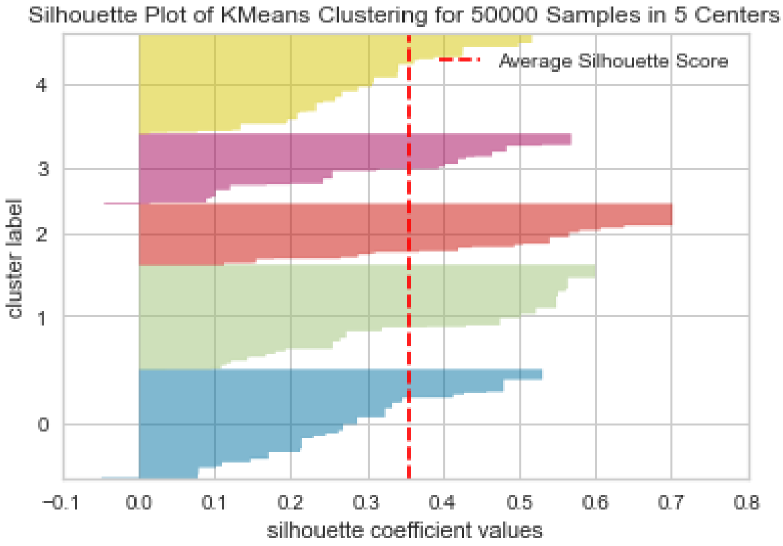
```
# numpy.full(shape, fill_value, dtype=None, order='C', *, like=None)
sc1 = np.full((1,rfm_df.shape[0]-int(0.8*rfm_df.shape[0])),1)
sc2 = np.full((1,int(0.2*rfm_df.shape[0])),2)
sc3 = np.full((1,int(0.2*rfm_df.shape[0])),3)
sc4 = np.full((1,int(0.2*rfm_df.shape[0])),4)
sc5 = np.full((1,int(0.2*rfm_df.shape[0])),5)
score = np.hstack((sc1, sc2,sc3,sc4,sc5)).flatten()
rfm_df.dtypes
rfm_df = rfm_df.sort_values(by='Recency', ascending = False)
rfm_df['r_score'] = score

for i, j in zip(('Frequency', 'Monetary'),('f_score','m_score')):
    rfm_df = rfm_df.sort_values(by=i)
    rfm_df[j] = score
```

Appendix: F RFM Score Table

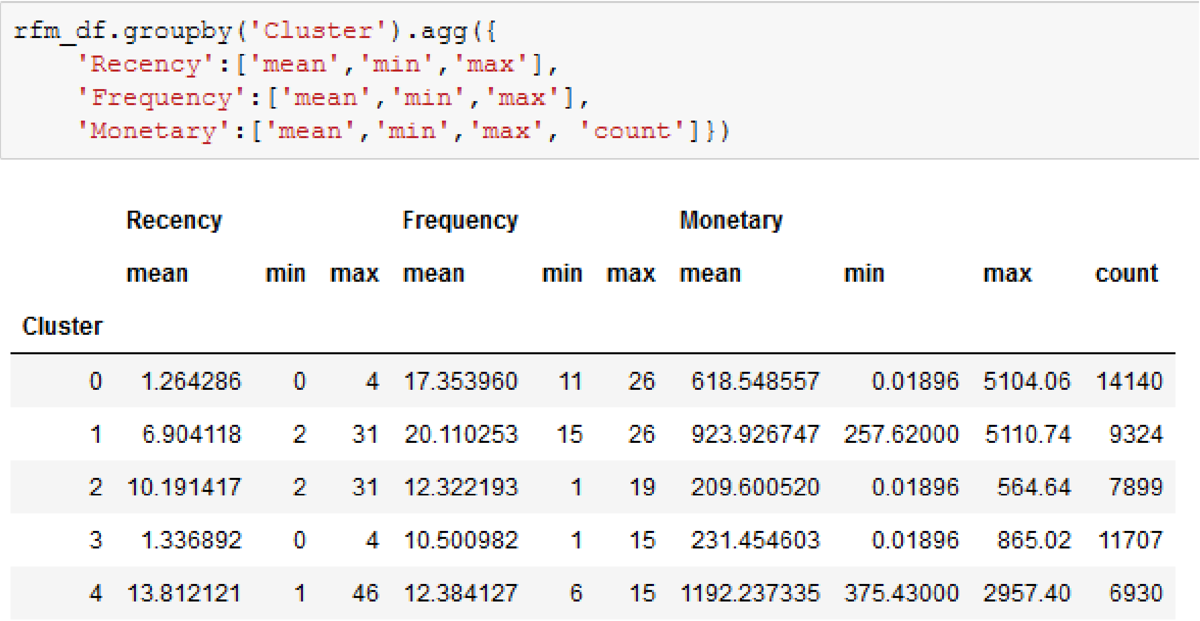


	r_score	f_score	m_score	RFM_Score
user_id				
56702	5	5	5	555
53192	5	5	5	555
76849	5	5	5	555
84218	5	5	5	555
99837	5	5	5	555
...	...	...	...	...
60799	1	1	1	111
55277	1	1	1	111
87196	1	1	1	111
62542	1	1	1	111
54634	1	1	1	111

Appendix: G Silhouette Analysis on RFM Table



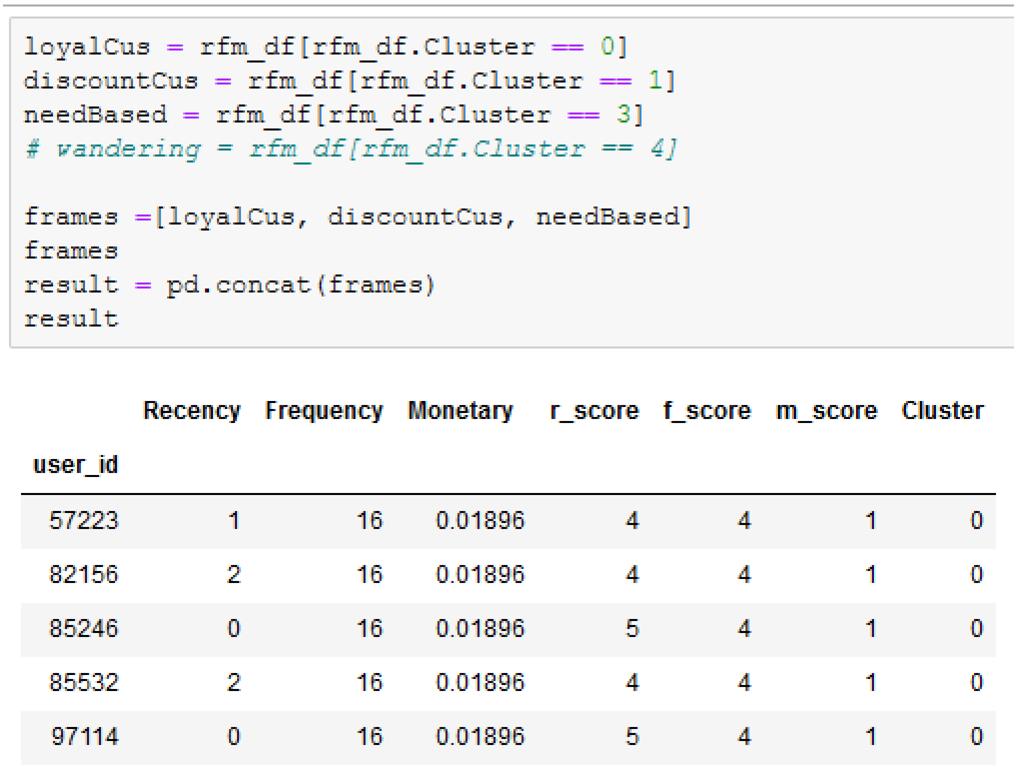
Appendix: H Finalize Cluster Table for Each Record



```
rfm_df.groupby('Cluster').agg({
    'Recency':['mean','min','max'],
    'Frequency':['mean','min','max'],
    'Monetary':['mean','min','max', 'count']})
```

	Recency			Frequency			Monetary			
	mean	min	max	mean	min	max	mean	min	max	count
Cluster										
0	1.264286	0	4	17.353960	11	26	618.548557	0.01896	5104.06	14140
1	6.904118	2	31	20.110253	15	26	923.926747	257.62000	5110.74	9324
2	10.191417	2	31	12.322193	1	19	209.600520	0.01896	564.64	7899
3	1.336892	0	4	10.500982	1	15	231.454603	0.01896	865.02	11707
4	13.812121	1	46	12.384127	6	15	1192.237335	375.43000	2957.40	6930

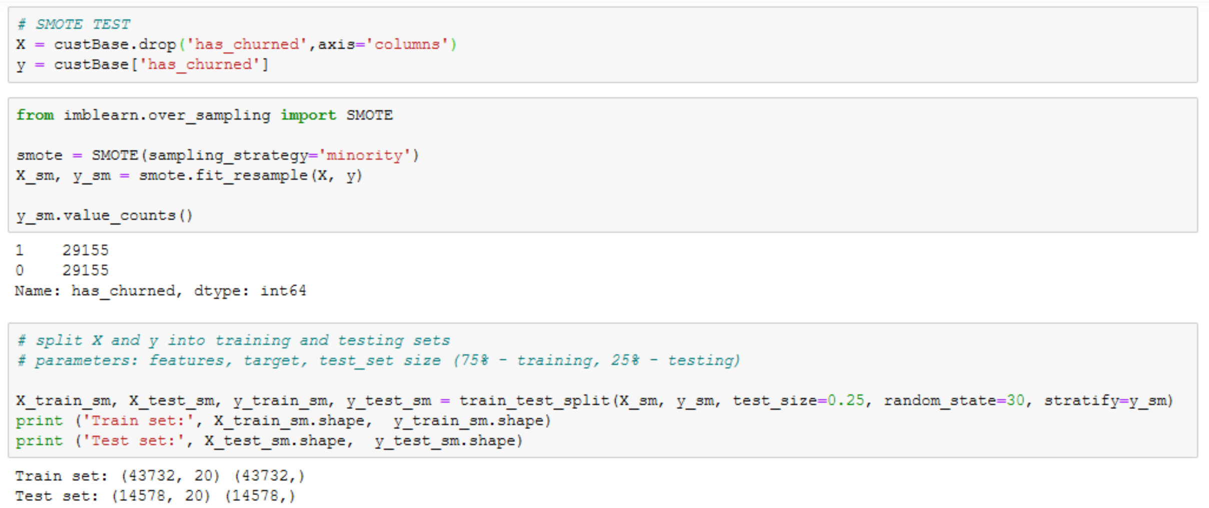
Appendix: I Selected Customer Base by Segmentation



```
loyalCus = rfm_df[rfm_df.Cluster == 0]
discountCus = rfm_df[rfm_df.Cluster == 1]
needBased = rfm_df[rfm_df.Cluster == 3]
# wandering = rfm_df[rfm_df.Cluster == 4]

frames =[loyalCus, discountCus, needBased]
frames
result = pd.concat(frames)
result
```

	Recency	Frequency	Monetary	r_score	f_score	m_score	Cluster
user_id							
57223	1	16	0.01896	4	4	1	0
82156	2	16	0.01896	4	4	1	0
85246	0	16	0.01896	5	4	1	0
85532	2	16	0.01896	4	4	1	0
97114	0	16	0.01896	5	4	1	0

Appendix: J SMOTE Technique on Training & Testing Data



```
# SMOTE TEST
X = custBase.drop('has_churned',axis='columns')
y = custBase['has_churned']
```

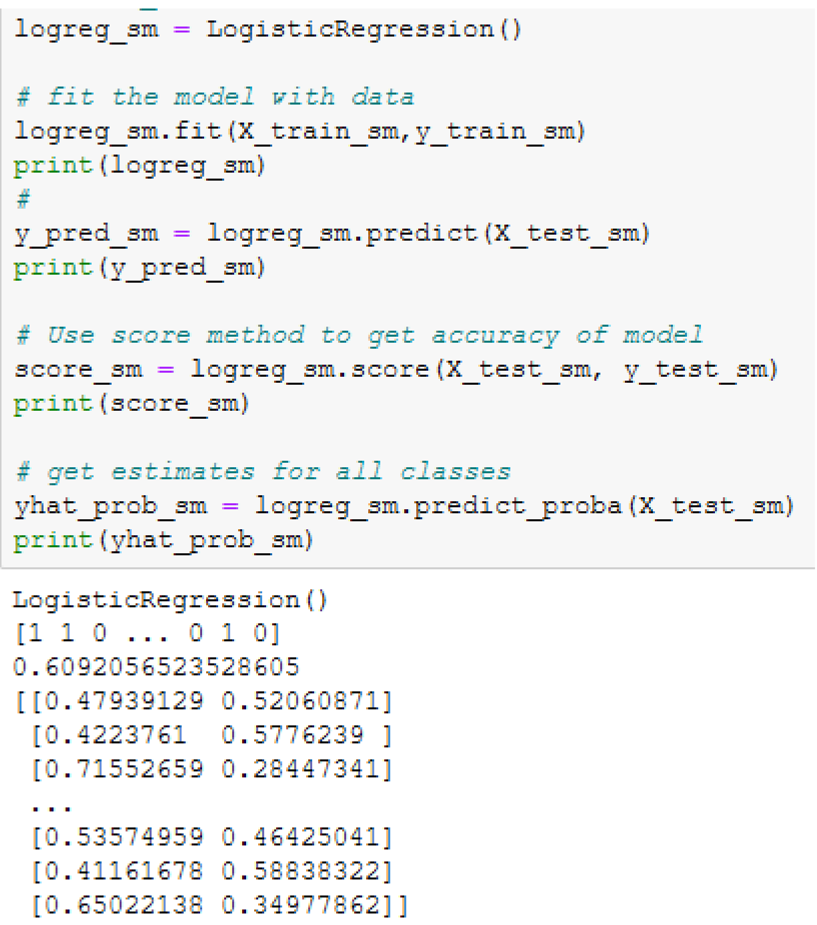
```
from imblearn.over_sampling import SMOTE

smote = SMOTE(sampling_strategy='minority')
X_sm, y_sm = smote.fit_resample(X, y)

y_sm.value_counts()
```

```
1    29155
0    29155
Name: has_churned, dtype: int64
```

```
# split X and y into training and testing sets
# parameters: features, target, test_set size (75% - training, 25% - testing)

X_train_sm, X_test_sm, y_train_sm, y_test_sm = train_test_split(X_sm, y_sm, test_size=0.25, random_state=30, stratify=y_sm)
print ('Train set:', X_train_sm.shape,  y_train_sm.shape)
print ('Test set:', X_test_sm.shape,  y_test_sm.shape)
```

```
Train set: (43732, 20) (43732,)
Test set: (14578, 20) (14578,)
```

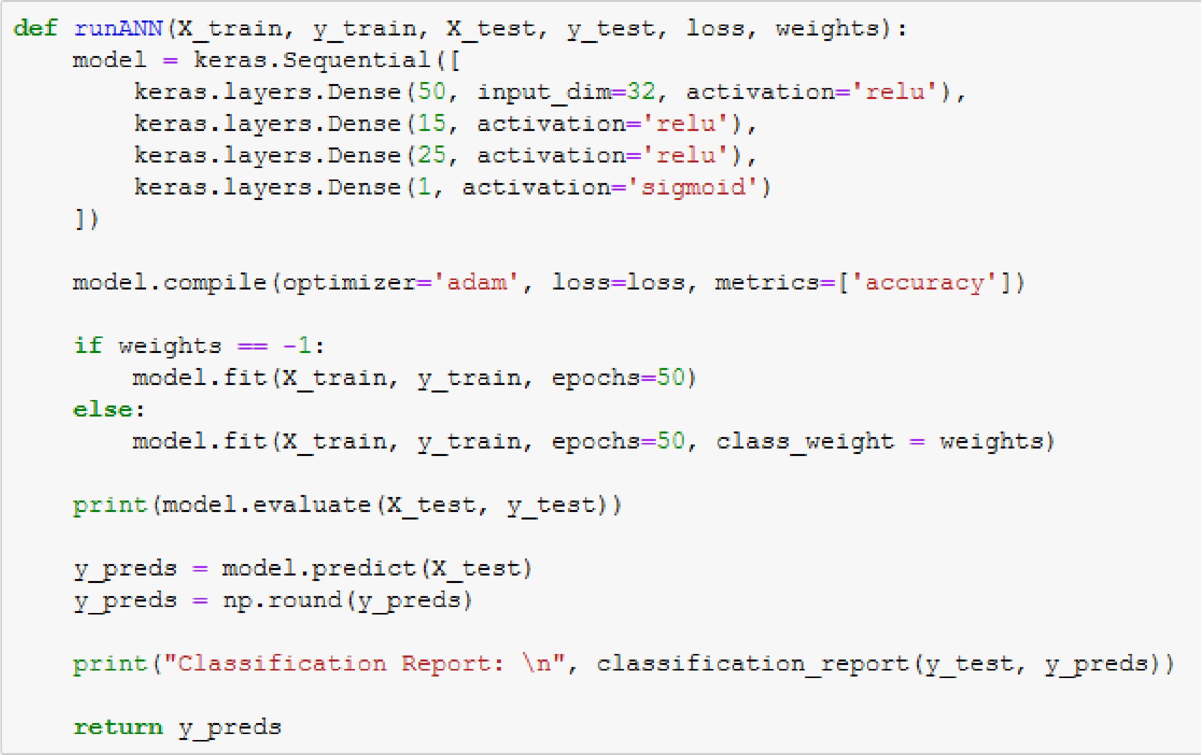
Appendix: K Logistic Regression Model



```
logreg_sm = LogisticRegression()

# fit the model with data
logreg_sm.fit(X_train_sm,y_train_sm)
print(logreg_sm)
#
y_pred_sm = logreg_sm.predict(X_test_sm)
print(y_pred_sm)

# Use score method to get accuracy of model
score_sm = logreg_sm.score(X_test_sm, y_test_sm)
print(score_sm)

# get estimates for all classes
yhat_prob_sm = logreg_sm.predict_proba(X_test_sm)
print(yhat_prob_sm)
```

```
LogisticRegression()
[1 1 0 ... 0 1 0]
0.6092056523528605
[[0.47939129 0.52060871]
 [0.4223761  0.5776239 ]
 [0.71552659 0.28447341]
 ...
 [0.53574959 0.46425041]
 [0.41161678 0.58838322]
 [0.65022138 0.34977862]]
```

Appendix: L Deep Neural Network Model



```
def runANN(X_train, y_train, X_test, y_test, loss, weights):
    model = keras.Sequential([
        keras.layers.Dense(50, input_dim=32, activation='relu'),
        keras.layers.Dense(15, activation='relu'),
        keras.layers.Dense(25, activation='relu'),
        keras.layers.Dense(1, activation='sigmoid')
    ])

    model.compile(optimizer='adam', loss=loss, metrics=['accuracy'])

    if weights == -1:
        model.fit(X_train, y_train, epochs=50)
    else:
        model.fit(X_train, y_train, epochs=50, class_weight = weights)

    print(model.evaluate(X_test, y_test))

    y_preds = model.predict(X_test)
    y_preds = np.round(y_preds)

    print("Classification Report: \n", classification_report(y_test, y_preds))

    return y_preds
```

REFERENCES

Amjad Khani, Zahid Ansari, 2014. Comparative Study of Data Mining Techniques in Telecommunications. *Internaiional Jounal of Emerging Technologies and Application,* 7(1), p. 8.

Anon., n.d. *pandas.get_dummies.* [Online]
Available at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html
[Accessed 14 06 2021].

Anon., n.d. *scikit-learn.* [Online]
Available at: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html
[Accessed 03 06 2021].

Anon., n.d. *sklearn.model_selection.train_test_split.* [Online]
Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
[Accessed 1 7 2021].

Anon., n.d. *The Sequential class.* [Online]
Available at: https://keras.io/api/models/sequential/
[Accessed 10 6 2021].

Anon., n.d. *What are the Different Types of Customers?.* [Online]
Available at: https://corporatefinanceinstitute.com/resources/knowledge/other/types-of-customers/
[Accessed 24 05 2021].

Bhalla, D., n.d. *Listen Data.* [Online]
Available at: https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html
[Accessed 25 05 2021].

Bhambri, V., 2012. Data Mining as a Tool to Predict Churn Behavior of Customers. *International Journal of Computer & Organization Trends,* 2(3), p. 5.

Bhandari, A., 2020. *Everything you Should Know about Confusion Matrix for Machine Learning.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/
[Accessed 1 07 2021].

Bhandari, A., 2020. *Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/
[Accessed 15 6 2021].

Brownlee, J., 2019. *A Gentle Introduction to Imbalanced Classification.* [Online]
Available at: https://machinelearningmastery.com/what-is-imbalanced-classification/
[Accessed 24 06 2021].

Brownlee, J., 2020. *10 Clustering Algorithms With Python.* [Online]
Available at: https://machinelearningmastery.com/clustering-algorithms-with-python/
[Accessed 25 05 2021].

Cormac Dullaghan and Eleni Rozaki, 2017. Integration of Machine Learning Tehniques to Evaluate Dynamic Customer Segmentation Analysis for Mobile Customers. *International Journal of Data Mining & Knowledge Management Process (IJDKP),* 7(1), p. 12.

Damith Senanayake, Lakmal Muthugama, Laksheen Mendis, Tiroshan Madushanka, 2015. Customer Churn Prediction: A Cognitive Approach. *nternational Journal of Computer, Electrical, Automation, Control and Information Engineering* , 9(3), p. 8.

developers, T. i.-l., 2014-2021. *Fitting model on imbalanced datasets and how to fight bias*¶. [Online]
Available at: https://imbalanced-learn.org/stable/auto_examples/applications/plot_impact_imbalanced_classes.html
[Accessed 1 7 2021].

E. W. Ngai, L. Xiu, and D. C. Chau, 2009. Application of data mining techniques in customer relationship management: A literature review and classification. *Expert systems with applications,* 36(2).

Garbade, D. M. J., 2018. *towards data science.* [Online]
Available at: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
[Accessed 01 07 2021].

Gregory, B., 2018. *Predicting Customer Churn: Extreme Gradient Boosting with Temporal Data.* Los Angeles, California USA, WSDM .

Harrison, O., 2018. *Machine Learning Basics with the K-Nearest Neighbors Algorithm.* [Online]
Available at: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
[Accessed 24 05 2021].

Hossein Abbasimehra, Mustafa Setakb, M . J. Tarokhc, 2012. *The application of Neuro-fuzzy classifiers on customer churn prediction.* s.l., Procedia Information Technology & Computer Science.

Hshan.T, 2020. *Exploring Customers Segmentation With RFM Analysis and K-Means Clustering With Python..* [Online]
Available at: https://medium.com/swlh/exploring-customers-segmentation-with-rfm-analysis-and-k-means-clustering-93aa4c79f7a7
[Accessed 23 05 2021].

Iris Figalist, Christoph Elsner, Jan Bosch, and Helena Holmstr¨om Olsson, 2020. *Customer Churn Prediction in B2B Contexts.* s.l., s.n.

Mark L. Berenson, David M. Levine, Timothy C. Krehbiel, 2012. 13 Simple Liner Regresion. In: S. Yagan, ed. *Basic Business Statistics: Concepts and Applications.* s.l.:Prentice , p. 890.

Muhammad Raza Khan, Joshua Manoj, Anikate Singh, Joshua Blumenstock, n.d. *Behavioral Modeling for Churn Prediction: Early Indicators and Accurate Predictors of Custom Defection and Loyalty,* s.l.: 1Information School, University of Washington, Seattle, WA, USA.

Muzaffar Shah, Darshan Adiga, Shabir Bhat and Viveka Vyeth, n.d. *Prediction and Causality Analysis of Churn Using Deep Learning,* Datoin Bangalore, India: Computer Science & Information Technology (CS & IT).

Renjith, S., September 2015. An Integrated Framework to Recommend Personalized Retention Actions to Control B2C E-Commerce Customer Churn. *International Journal of Engineering Trends and Technology (IJETT)*, 27 (3), p. 6.

Gmail **Maheshika Weerasinghe <maheshiweerasinghe486@gmail.com>**

Final Dissertation - Supervisor Version

Kasun Karunanayaka <ktk@ucsc.cmb.ac.lk> 14 September 2021 at 07:33
To: Maheshika Weerasinghe <maheshiweerasinghe486@gmail.com>

Dear Maheshika,
Turnitin report is also attached which shows the similarity. Thesis is good enough. Let's submit.
Best Wishes,

Kasun Karunanayaka
Senior Lecturer,
Department of Communication and Media Technologies
University of Colombo School of Computing
UCSC Building Complex, 35 Reid Avenue, Colombo 07
Sri Lanka.
Phone +94 -11- 2581245/ 7 | Fax +94 112 587239
Email: ktk@ucsc.cmb.ac.lk
Web: University Profile, Personal Profile

[Quoted text hidden]

Final Thesis-Turnitin.pdf
12980K