Degree Thesis

Master's level (Second cycle)

Predicting Customer Churn in a Subscription-Based E-Commerce Platform Using Machine Learning Techniques

Author: Ahmed Aljifri
School: Dalarna University
Supervisor: Somayeh Aghanavesi
Examiner: Mia Xiaoyun Zhao

Subject/main field of study: Micro Dada Analysis
Course code: MI4001
Credits: 30 hp
Date of examination: 16/01/2024

Abstract:

This study investigates the performance of Logistic Regression, k-Nearest Neighbors (KNN), and Random Forest algorithms in predicting customer churn within an e-commerce platform. The choice of the mentioned algorithms was due to the unique characteristics of the dataset and the unique perception and value provided by each algorithm. Iterative models 'examinations, encompassing preprocessing techniques, feature engineering, and rigorous evaluations, were conducted. Logistic Regression showcased moderate predictive capabilities but lagged in accurately identifying potential churners due to its assumptions of linearity between log odds and predictors. KNN emerged as the most accurate classifier, achieving superior sensitivity and specificity (98.22% and 96.35%, respectively), outperforming other models. Random Forest, with sensitivity and specificity (91.75% and 95.83% respectively) excelled in specificity but slightly lagged in sensitivity. Feature importance analysis highlighted "Tenure" as the most impactful variable for churn prediction. Preprocessing techniques differed in performance across models, emphasizing the importance of tailored preprocessing. The study's findings underscore the significance of continuous model refinement and optimization in addressing complex business challenges like customer churn. The insights serve as a foundation for businesses to implement targeted retention strategies, mitigating customer attrition, and promote growth in e-commerce platforms.

Keywords: Customer churn prediction, E-commerce, Machine learning algorithms, Logistic Regression, k-Nearest Neighbors (KNN), Random Forest, Feature engineering, Preprocessing techniques, Model evaluation, performance measures, supervised machine learning, classification, confusion matrix.

Table of Contents

Introduction

Background

Customer retention is an important concept for an e-commerce business to sustain and progress its market share value. As a new startup, normally one would focus on customer acquisition to start establishing itself. However, to succor longevity in the market, an e-commerce business must focus on customer retention as customers are the building block of its success (“What is Customer”, 2023).

In recent years there has been a sizable growth in the e-commerce industry which led to an increase in customer churn rates. This surge is mainly due to the growing competition from startups, which has put established platforms in a position when it comes to retaining their customer base. Based on a Bain & Company report, “A 5% increase in customer retention produces more than a 25% increase in profit” (Reichheld, 2001). That shows how greatly customer retention impacts revenues. Therefore, proactivity in predicting customers’ churn will in turn sustain revenues and promote growth.

A very common metric that quantifies customer retention is customer churn. It is a metric that within a time frame provides a percentage of customers who have unsubscribed or left the business. To have a benchmark, in a subscription-based businesses a 5% churn is considered average, therefore businesses in the industry should keep that as a reference going forward (“Ecommerce Churn”, 2022).

Purpose and Research Objectives

This research strives to add value in the realm of e-commerce analytics, aiming to unravel the complexities of customer churn prediction through an in-depth exploration of machine learning models. This exploration not only benchmarks the employed methodologies but also aspires to improve predictive accuracy. The research holds practical significance for e-commerce businesses, offering insights to refine customer retention strategies. This endeavour contributes not only to industry practices but also to the broader scientific understanding of customer churn dynamics.

Research Gap and Scientific Value:

The research fills a crucial void in the literature by focusing on the unique characteristics of the data for customer churn prediction within e-commerce, which has received limited attention. The scientific value lies in the application of machine learning techniques to derive actionable insights for businesses in the e-commerce domain. Previous research has often introduced novel approaches for finding out intricacies within the data by for instance, examining customer segmentation, behavioural vs non-behavioural data, transactional vs non-transactional data, temporal aspects. However, the goal is seldom to find the best model in terms of predictive accuracy of churn, but rather compare known machine learning models’ performances with no focus on specific characteristics of data.

Motivation and Research Objectives:

The primary motivation behind this research is to empower e-commerce businesses with accurate and efficient tools for predicting customer churn. By understanding the factors influencing customer retention or churn, businesses can implement proactive strategies to enhance customer retention and, consequently, overall business performance.

Therefore, the purpose of this thesis is to identify, build and compare machine learning models based on dataset characteristics in order to obtain the most accurate model for predicting customers' churn in an e-commerce platform. Furthermore, a secondary purpose is to identify the most impactful variables on churn in order for decision makers to be proactive in their strategies and action plans.

Literature Review

The realm of e-commerce is continually evolving, and the ability to predict customer churn holds immense significance for businesses aiming to maintain a competitive edge. This consolidated literature review synthesizes insights from three distinct papers, each contributing valuable perspectives to the field of customer churn prediction in diverse e-commerce domains. Three distinct, recent and innovative papers have been reviewed to support this research, the details of which is outlined next.

"B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM" by Xiancheng Xiahou and Yoshio Harada:

This paper delves into the critical importance of anticipating customer churn in Business-to-Consumer (B2C) e-commerce. Acknowledging customer churn as a substantial threat, the paper emphasizes the proactive implementation of retention strategies to ensure sustained profitability and customer loyalty. The focus here is on leveraging advanced machine learning models, particularly K-Means clustering and Support Vector Machines (SVM), to predict churn.

Furthermore, the paper highlights the significance of customer segmentation for formulating effective marketing strategies. Techniques like k-means clustering are discussed as integral to categorizing customers into distinct groups, enabling businesses to tailor retention strategies based on the unique characteristics of each segment. The paper contributes to ongoing discussions by comparing the predictive performance of Support Vector Machines and Logistic Regression models, with SVM demonstrating better accuracy in customer churn prediction.

One of the key themes explored in this paper is the role of behavioural data, including variables such as page views, purchases, cart interactions, and favourites. Time variables, segmented into different hours, emerge as crucial features, reflecting the temporal aspect of customer shopping behaviours. The literature review underscores the significance of variable selection, with recent studies highlighting the importance of time variables, particularly during Night and PM hours. Techniques such as Random Forest variable selection are discussed as effective tools for identifying critical features based on metrics like the Gini index.

The challenge of imbalanced datasets in predictive modelling is addressed, with a focus on data balancing techniques like the Synthetic Minority Over-sampling Technique (SMOTE) to ensure fair representation of both churn and non-churn instances. The evaluation of churn prediction models is explored through metrics such as accuracy, recall rate, precision, and the Area Under the Receiver Operating Characteristic (AUC-ROC) curve, collectively providing insights into overall predictive performance.

"Customer Churn Prediction in Influencer Commerce: An Application of Decision Trees" by Sulim Kima and Heeseok Lee:

This paper addresses the evolving landscape of influencer commerce. This distinct form of e-commerce involves influencers leveraging their social media presence to directly sell products to their followers. The study, conducted by Sulim Kim and Heeseok Lee from KAIST in Korea, focuses on predicting customer churn within this unique context, employing the Decision Trees (DT) algorithm.

The paper begins by establishing the increasing significance of customer churn prediction, particularly in industries where establishing long-term relationships with customers is crucial. Predicting churn is emphasized as a cost-effective strategy for refining retention strategies.

The challenges specific to predicting customer churn in e-commerce are explored, where data is often unbalanced, and defining churn points can be challenging. The study describes the dataset used, collected by an influencer marketing agency in Korea, spanning customer purchase details from August 2018 to October 2020. The Decision Trees (DT) algorithm, implemented through the Rapidminer software program, is chosen for predicting customer churn.

The paper presents the results and evaluation metrics, including recall, precision, accuracy, and F-measure, highlighting the Decision Trees model's impressive prediction accuracy of 90% based on F-measure. The conclusion emphasizes the importance of predicting customer churn in influencer commerce and suggests further research to explore the application of various algorithms to enhance the model's applicability.

“Predicting Customer Churn in E-Commerce Using Statistical Modelling and Feature Importance Analysis: A Comparison of Random Forest and Logistic Regression Approaches" by Amanda Rudälv:

Lastly, the third paper addresses the dynamic landscape of e-commerce. The study, conducted by an undisclosed author, contributes to the body of knowledge by conducting a comprehensive analysis that compares the performance of two widely used models, Random Forest and Logistic Regression, in predicting customer churn.

The paper recognizes the critical role of customer churn prediction in customer relationship management within the e-commerce domain. Traditional statistical modelling and modern machine learning techniques are explored to enhance the accuracy and interpretability of churn prediction models. The study distinguishes itself by conducting a detailed analysis comparing Random Forest, known for handling complex relationships, and Logistic Regression, a conventional yet interpretable model.

Non-transactional behaviour is given due consideration in understanding customer churn, including features like session patterns and intervals. The study investigates the impact of deviations in customer behaviour on prediction accuracy, contributing nuanced insights into potential pain points in the customer journey. Feature importance analysis is a notable aspect of the research, shedding light on the factors significantly contributing to predicting customer churn, such as session intervals and page views.

The comparison of Random Forest and Logistic Regression models extends beyond traditional metrics, emphasizing the importance of probability-based metrics in evaluating model performance. The paper advocates for continued exploration of Random Forest approaches and suggests the integration of SHAP (SHapley Additive exPlanations) importance for a more comprehensive understanding of feature contributions.

In conclusion, these three papers collectively provide a nuanced and comprehensive understanding of customer churn prediction in the dynamic landscape of e-commerce. By leveraging advanced machine learning techniques, addressing challenges, and comparing different modelling approaches, these studies contribute valuable insights that can inform businesses in crafting effective strategies for customer retention in the digital marketplace. However, they lack the specific goal of identifying the most accurate machine learning model for predicting customer’s churn in the e-commerce sector, which this thesis further investigates in detail.

Data Exploration and Preprocessing

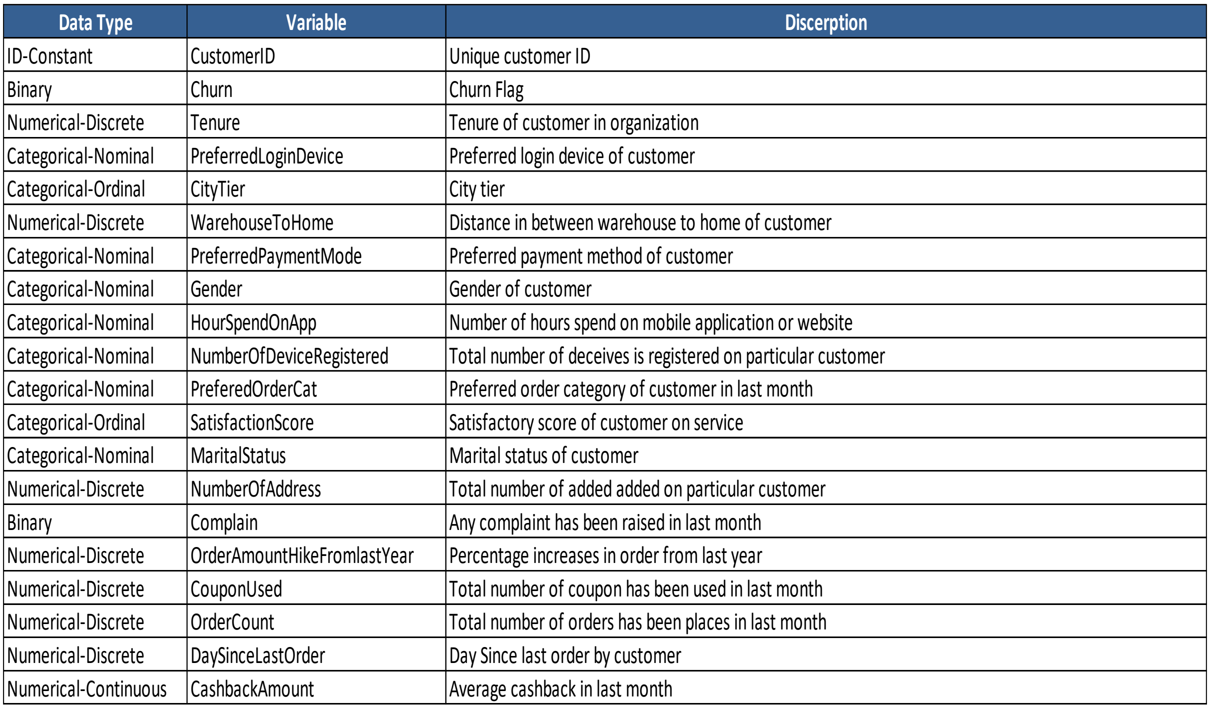
This chapter provides a comprehensive overview of the dataset on hand, main data science topics that arise during data exploration, data cleaning and preprocessing employed for analysis. Furthermore, the variables included in the dataset are highlighted and elaborated to establish a better understanding of the dataset.

Data Exploration:

First step in any data analysis/modelling is always data exploring and cleaning. Usually, datasets acquired for analysis are not in the desired format, whether that be data types are stored in improper format, dataset is not in tidy format, meaning each row doesn’t represent a unique observation, or in case of for example web scarping and text mining, data can be complexly messy is various ways. Therefore, spending time on these steps are crucial for ease and accuracy of analysis in later stages. In addition, data cleaning is mostly an iterative process going back and forth between the cleaning step and implementing ML algorithms, since during implementation further adjustments and tunings are needed on the data to achieve desired results (Turner, 2020).

For the purpose of this thesis, a public dataset has been retrieved from Kegel website that outlines customers’ churn and their corresponding information for a highly reputable E-commerce online based company (name left out for privacy concerns). The dataset is already in tidy format retrieved as an excel sheet which makes it efficient to import into R and start preparing. The dataset has a label (Churn), 18 predictors and 5630 unique observations, description of variables shown below:

Table.1 Outline of Variables in The Dataset and Their Data Types



Data Type	Variable	Discerption
ID-Constant	CustomerID	Unique customer ID
Binary	Churn	Churn Flag
Numerical-Discrete	Tenure	Tenure of customer in organization
Categorical-Nominal	PreferredLoginDevice	Preferred login device of customer
Categorical-Ordinal	CityTier	City tier
Numerical-Discrete	WarehouseToHome	Distance in between warehouse to home of customer
Categorical-Nominal	PreferredPaymentMode	Preferred payment method of customer
Categorical-Nominal	Gender	Gender of customer
Categorical-Nominal	HourSpendOnApp	Number of hours spend on mobile application or website
Categorical-Nominal	NumberOfDeviceRegistered	Total number of deceives is registered on particular customer
Categorical-Nominal	PreferedOrderCat	Preferred order category of customer in last month
Categorical-Ordinal	SatisfactionScore	Satisfactory score of customer on service
Categorical-Nominal	MaritalStatus	Marital status of customer
Numerical-Discrete	NumberOfAddress	Total number of added added on particular customer
Binary	Complain	Any complaint has been raised in last month
Numerical-Discrete	OrderAmountHikeFromlastYear	Percentage increases in order from last year
Numerical-Discrete	CouponUsed	Total number of coupon has been used in last month
Numerical-Discrete	OrderCount	Total number of orders has been places in last month
Numerical-Discrete	DaySinceLastOrder	Day Since last order by customer
Numerical-Continuous	CashbackAmount	Average cashback in last month

Predictors have been manually categorized as a preliminary step into three different data types: 8 Numerical, 9 Categorical and 1 Binary variables, and shall be modified accordingly during cleaning and ML implementation. Histograms/bar plots of Numerical and Categorical variables are displayed below in order to get an idea of data types/distributions and a sense of outliers:

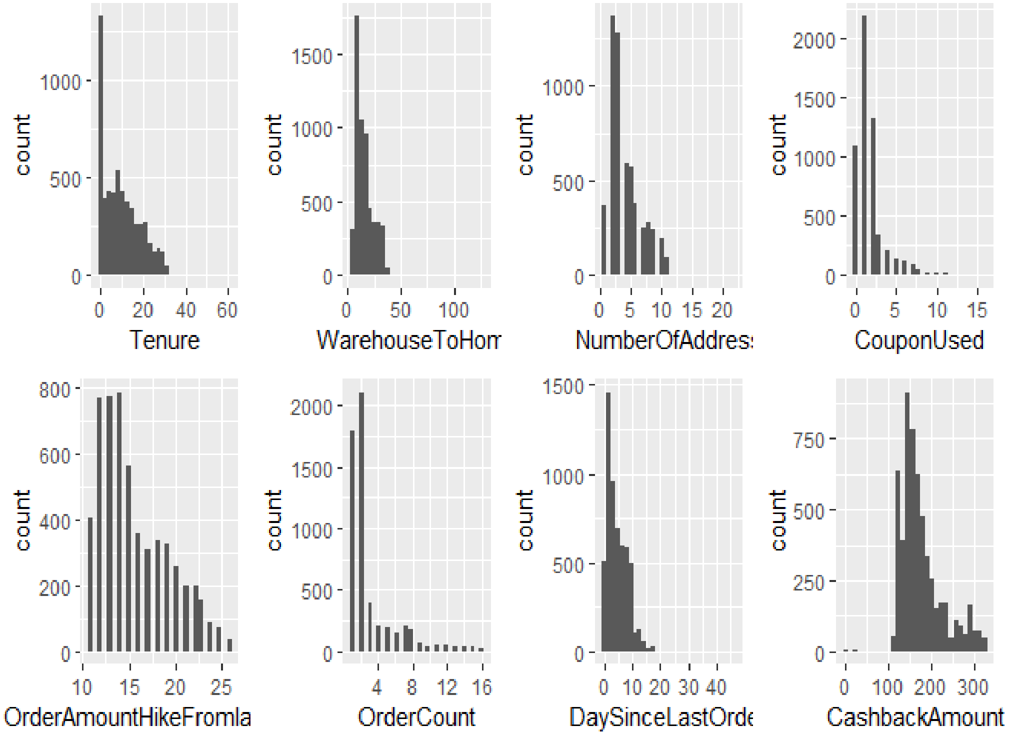


Figure.1 Histograms of Numerical Variables in The Dataset

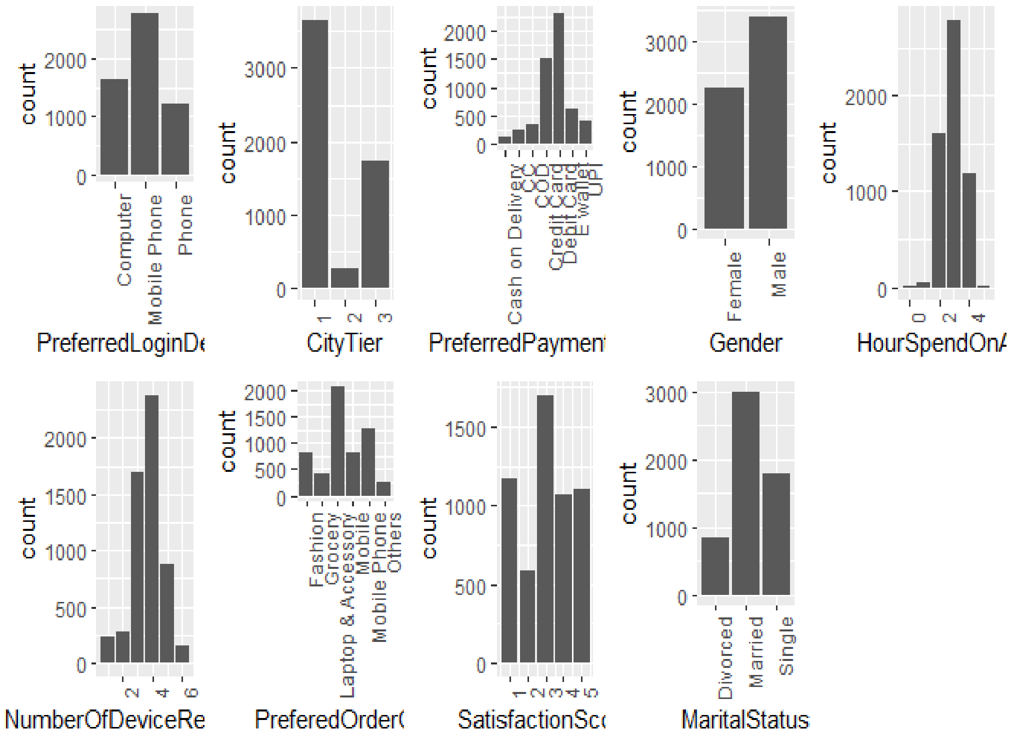


Figure.2 Bar Plots of Categorical Variables in The Dataset

Outliers and redundancies:

Looking briefly at Figure.1 and Figure.2 it is evident that there are outliers in the variables. Outliers are values that are extreme with respect to their average value, whether way greater or way smaller than the corresponding average. Outliers generally exist because of measurements' variability, human/experimental errors or occurring randomly due to the nature of object being studied. Moreover, they can introduce anomalies in analysis if not thoroughly investigated. Therefore, when talking about outliers there is a critical question that comes to mind for data analysts, should we keep or remove them? Well, it is not as straightforward as it might seem, there are two main reasons why outliers should be investigated carefully:

1- They can impact analysis negatively and induce faulty results.
2- They could be a crucial factor for the project or analysis being done.

Therefore, having the project's purpose in mind coupled with domain knowledge allows one to think of outliers as point one and remove them if they do impact analysis negatively, or as point two and monitor their impact closely (Sequitin, 2021).

Moreover, there are two main types of outliers:

1- Univariate outliers: an extreme value that is present out of one variable independently.
2- Multivariate outliers: an extreme value that is present out of at least two variables dependently while the values are within normal range by themselves with respect to their corresponding variable.

Nevertheless, in order to detect and map outliers there are several ideas available to explore. For Numerical variables, the main two ideas are boxplots and scatter plots for their ease of use and readability. For Categorical variables, bar plots along with investigating levels of factorised variables can give an insight as well (Sequitin, 2021).

As described above boxplots are generated for Numerical variables to get an idea about outliers and their corresponding statistics as shown in Figure.3 below.

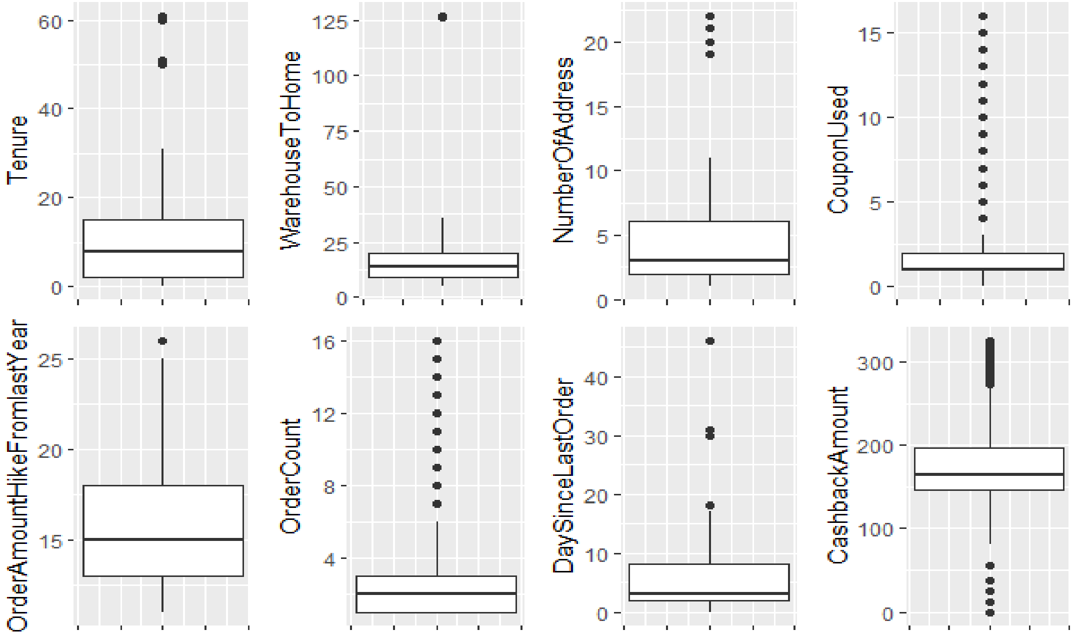
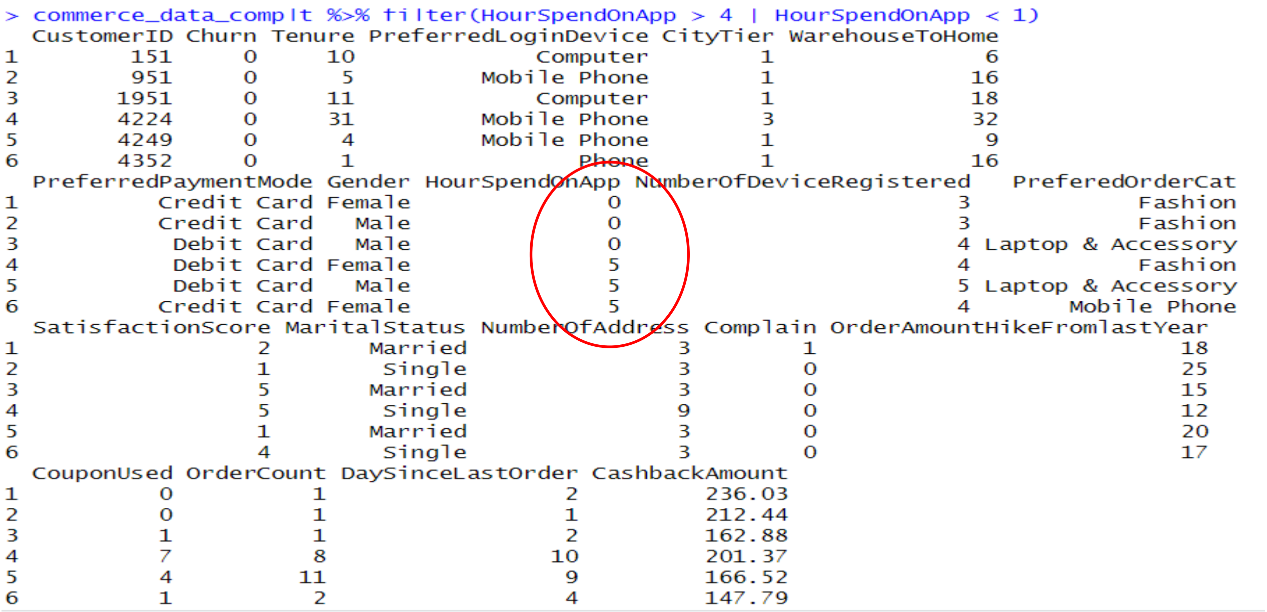


Figure.3 (Boxplots Num Vars)

It is clear from looking at Figure.3 that there are quite some outliers, however, as described above the choice of keeping or removing them could directly impact results of machine learning algorithms to be employed. As the research objectives do not imply any emphasis on outliers' analysis, these outliers can be investigated further during ML implementation whether they do impact our results or not. Making an educated choice could prevent models' performance issues such as overfitting and underfitting.

On the other hand, looking at the bar plots in Figure.2 it is evident that the variable "HourSpendOnApp" has six levels (0,1,2,3,4,5). By further investigation, it was clear that levels (0, 5) have very few observations. Filtering those two levels it is observed that each has only three entries which can be considered outliers as displayed in Figure.4.



```
> commerce_data_complt %>% filter(HourSpendOnApp > 4 | HourSpendOnApp < 1)
  CustomerID Churn Tenure PreferredLoginDevice CityTier WarehouseToHome
1        151     0     10             Computer        1               6
2        951     0      5         Mobile Phone        1              16
3       1951     0     11             Computer        1              18
4       4224     0     31         Mobile Phone        3              32
5       4249     0      4         Mobile Phone        1               9
6       4352     0      1                Phone        1              16
  PreferredPaymentMode Gender HourSpendOnApp NumberOfDeviceRegistered    PreferedOrderCat
1          Credit Card Female              0                        3             Fashion
2          Credit Card   Male              0                        3             Fashion
3           Debit Card   Male              0                        4 Laptop & Accessory
4           Debit Card Female              5                        4             Fashion
5           Debit Card   Male              5                        5 Laptop & Accessory
6          Credit Card Female              5                        4        Mobile Phone
  SatisfactionScore MaritalStatus NumberOfAddress Complain OrderAmountHikeFromlastYear
1                 2       Married               3        1                          18
2                 1        Single               3        0                          25
3                 5       Married               3        0                          15
4                 5        Single               9        0                          12
5                 1       Married               3        0                          20
6                 4        Single               3        0                          17
  CouponUsed OrderCount DaySinceLastOrder CashbackAmount
1          0          1                 2         236.03
2          0          1                 1         212.44
3          1          1                 2         162.88
4          7          8                10         201.37
5          4         11                 9         166.52
6          1          2                 4         147.79
```

Figure.4 (Filtered HourSpendOnApp)

Lastly, variables “PreferredLoginDevice”, “PreferredPaymentMethod” and “PreferredOrderCategory” have some redundancies in their levels as seen in the bar plots of Figure.2. “PreferredLoginDevice” has (Mobile Phone, Phone), “PreferredPaymentMethod” has (COD, Cash On Delivery) and (CC, Credit Card) and “PreferredOrderCategory” has (Mobile, Mobile Phone) which shall be addressed in Data Cleaning and Preparation section.

NA values:

An NA value is defined as a placeholder which represents an entry that is not stored or missing in one or more variables in a given dataset (Tamboli, 2021). NA is an abbreviation of “Not Available” or “Not Applicable”, therefore it is used to represent situations where the data point is unknown, unrecorded, or simply not applicable (Dealing with missing values, 2016).

Nevertheless, there are three main types of NA values that are encountered in datasets described as (Tamboli, 2021):

1- **Missing Completely at Random (MCAR):** The probability of a missing entry is the same for all variables; the missing entries are completely independent of other data. A main takeaway from MCAR is that the statistical analysis remains unbiased.

2- Missing At Random (MAR): The probability of a missing entry for a given variable is conditioned by another variable; the missing entries of a given variable is dependent on another. A main takeaway from MAR is that the statistical analysis may be biased if missing values were not handled correctly.

3- Missing Not at Random (MNAR): Missing values depend on unseen data. That is, unlike MAR there is some pattern observed on missing observations. If the data were to be collected on an entire population, however, observations of a stratum of such population is missing due to many reasons such as: reluctance of a certain group to provide information. Same as MAR, the statistical analysis may be biased.

The simplest way for detecting NA values is to use the function summary() in R. the function outlines simple statistics, box plot statistics, in addition to highlighting how many NA values are present if any for each variable in the dataset. Table.2 outlines NA entries extracted from the outcome of summary().

Table.2 NA Values of Variables in The Dataset

Variable Name	NAs
Tenure	264
WarehouseToHome	251
HourSpendOnApp	255
OrderAmountHikeFromlastYear	265
CouponUsed	256
OrderCount	258
DaySinceLastOrder	307

In addition to the NA summaries shown in Table.2, a visualization of NAs can be obtained through a heatmap as shown in Figure.5.

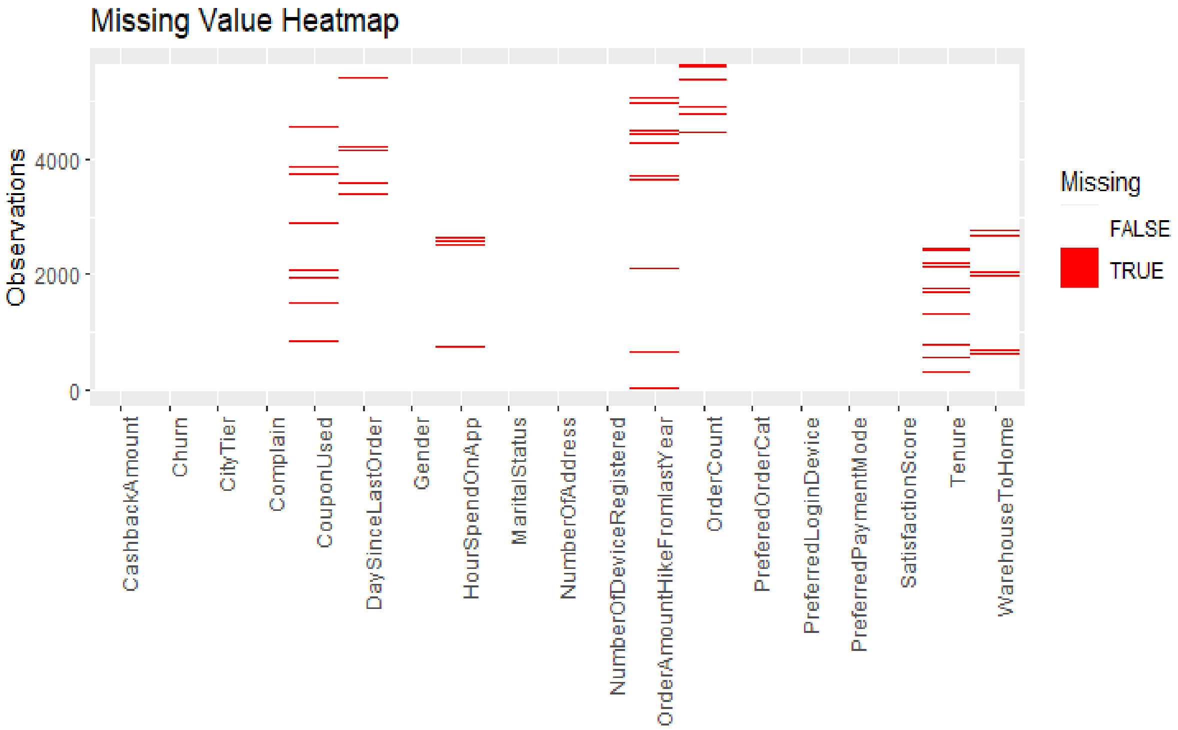


Figure.5 Missing Values Heatmap

Having discussed and explained missing values implications and types, it is possible to explore some effective ways of handling such missing values. Based on Tamboli N. (2021), there are two primary ways of dealing with missing values:

1- Imputing the missing values
2- Deleting the missing values

Imputing the missing values:
There are many methods to impute missing values, most common ways are: replacing with an arbitrary value, with the mean value, with the mode value, with the median value, with the previous value (forward fill), with the next value (backward fill), with interpolation or using some statistical methods available in R such as PMM (Predictive Mean Matching) (Tamboli, 2021).

Deleting the missing values:
There is always the choice of removing missing values, which can be done by removing whole observations or removing whole variables. If an observation has many missing values one might want to remove said row. On the other hand, if a variable contains many missing values one might want to remove said variable (Tamboli, 2021).

The choice of which method to use is highly dependent on the nature of the data on hand. It needs some domain experience to be able to effectively choose the right method in order to maintain the original variability of the data, and the integrity plus unbiases of the results (Tamboli, 2021).

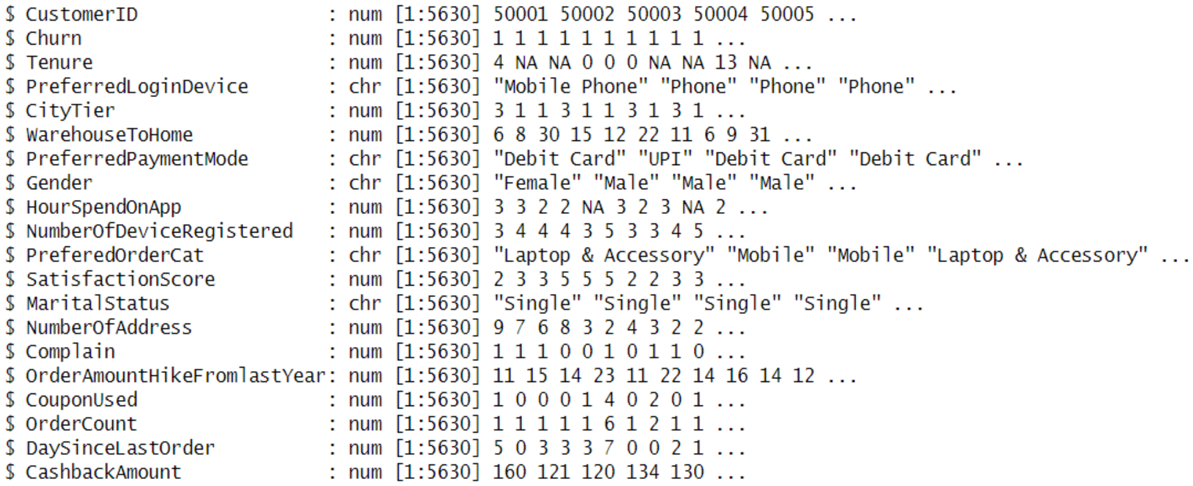
Data Cleaning and Preparation

The data cleaning process is an essential step in any data analysis or research study. It involves the thorough examination, identification, and rectification of inconsistencies, errors, and missing values within a dataset. The quality and reliability of analysis heavily depend on the accuracy and integrity of the data being used. In this section, let us expand on the data exploration section where we explored and visualized the dataset on hand and deal with the inconsistencies, errors and missing values to ensure that the dataset is properly prepared for ML implementation and analysis. Moreover, the data cleaning process ensures that the subsequent analysis and insights drawn are accurate, reliable, and trustworthy
(Turner, 2020).

Outliers and redundancies

Outliers' analysis shall be discussed further during ML implementation, this section will focus on rectifying inconsistencies and redundancies in the variables in question.

To get an idea about the data structure, str() function in R can be used. Figure.6 outlines the data structure of the given dataset.



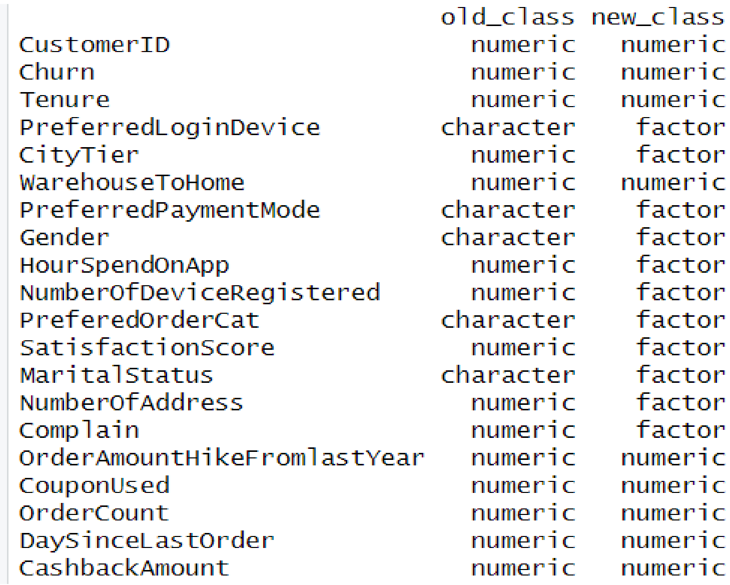
```
$ CustomerID                 : num [1:5630] 50001 50002 50003 50004 50005 ...
$ Churn                      : num [1:5630] 1 1 1 1 1 1 1 1 1 1 ...
$ Tenure                     : num [1:5630] 4 NA NA 0 0 0 NA NA 13 NA ...
$ PreferredLoginDevice       : chr [1:5630] "Mobile Phone" "Phone" "Phone" "Phone" ...
$ CityTier                   : num [1:5630] 3 1 1 3 1 1 3 1 3 1 ...
$ WarehouseToHome            : num [1:5630] 6 8 30 15 12 22 11 6 9 31 ...
$ PreferredPaymentMode       : chr [1:5630] "Debit Card" "UPI" "Debit Card" "Debit Card" ...
$ Gender                     : chr [1:5630] "Female" "Male" "Male" "Male" ...
$ HourSpendOnApp             : num [1:5630] 3 3 2 2 NA 3 2 3 NA 2 ...
$ NumberOfDeviceRegistered   : num [1:5630] 3 4 4 4 3 5 3 3 4 5 ...
$ PreferedOrderCat           : chr [1:5630] "Laptop & Accessory" "Mobile" "Mobile" "Laptop & Accessory" ...
$ SatisfactionScore          : num [1:5630] 2 3 3 5 5 5 2 2 3 3 ...
$ MaritalStatus              : chr [1:5630] "Single" "Single" "Single" "Single" ...
$ NumberOfAddress            : num [1:5630] 9 7 6 8 3 2 4 3 2 2 ...
$ Complain                   : num [1:5630] 1 1 1 0 0 1 0 1 1 0 ...
$ OrderAmountHikeFromlastYear: num [1:5630] 11 15 14 23 11 22 14 16 14 12 ...
$ CouponUsed                 : num [1:5630] 1 0 0 0 1 4 0 2 0 1 ...
$ OrderCount                 : num [1:5630] 1 1 1 1 1 6 1 2 1 1 ...
$ DaySinceLastOrder          : num [1:5630] 5 0 3 3 3 7 0 0 2 1 ...
$ CashbackAmount             : num [1:5630] 160 121 120 134 130 ...
```

Figure.6 Data Structure of Variables “Before adjustments”

Firstly, the structure of our variables in the dataset will be investigated. We shall go through the data types of said variables and modify what is needed to reflect the data types elaborated in Table.1. In addition, the inconsistencies and redundancies highlighted in the previous section will be rectified.

First modification is the variable “CustomerID”, its range goes from 50001:55630. To be consistent, its range will be rearranged into 1:5630. After that, all the categorical variables plus the variable “Complain” are factorised to reflect their categorical nature. Moreover, “PreferredLoginDevice” variable has three levels two of which are redundant as seen in Figure.6. Therefore, it can be squeezed into two levels: “Computer” and “Phone” and then coded as binary where 0 denotes “Computer” and 1 denotes “Phone” and factorised afterwards.

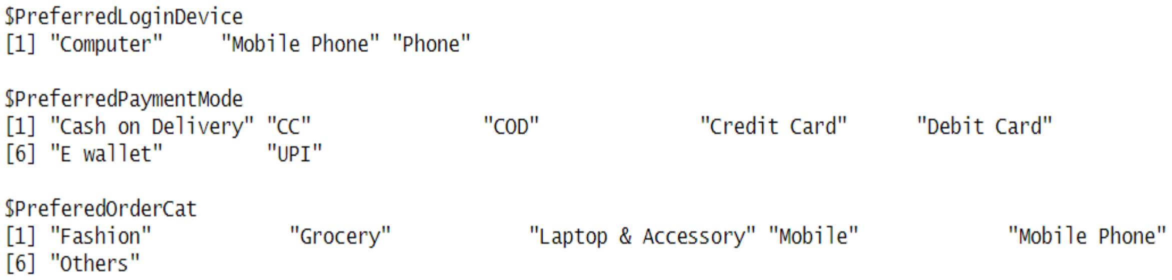
Nevertheless, “Gender” variable is coded into a binary variable as well: 0 for “Female” and 1 for “Male”, and then factorised with the rest of the categorical and binary variables.



```
                             old_class new_class
CustomerID                     numeric   numeric
Churn                          numeric   numeric
Tenure                         numeric   numeric
PreferredLoginDevice         character    factor
CityTier                       numeric    factor
WarehouseToHome                numeric   numeric
PreferredPaymentMode         character    factor
Gender                       character    factor
HourSpendOnApp                 numeric    factor
NumberOfDeviceRegistered       numeric    factor
PreferedOrderCat             character    factor
SatisfactionScore              numeric    factor
MaritalStatus                character    factor
NumberOfAddress                numeric    factor
Complain                       numeric    factor
OrderAmountHikeFromlastYear    numeric   numeric
CouponUsed                     numeric   numeric
OrderCount                     numeric   numeric
DaySinceLastOrder              numeric   numeric
CashbackAmount                 numeric   numeric
```

Figure.7 Old Variables’ Classes and Their Corresponding Modified Class

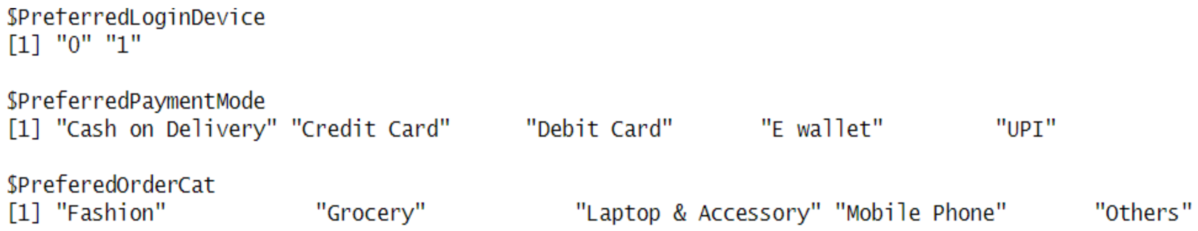
Figure.7 summarises the class adjustments of the variables in conjunction with Table.1. In addition to the redundancy of the variable “PreferredLoginMode”, the variables “PreferredPaymentMode” and “PreferedOrderCat” are adjusted according to the redundant levels. That is, “CC, COD” with “Credit Card, Cash On Delivery” respectively and “Mobile, Mobile Phone” shown in Figure.8.



```
$PreferredLoginDevice
[1] "Computer"     "Mobile Phone" "Phone"

$PreferredPaymentMode
[1] "Cash on Delivery" "CC"               "COD"              "Credit Card"      "Debit Card"
[6] "E wallet"         "UPI"

$PreferedOrderCat
[1] "Fashion"            "Grocery"            "Laptop & Accessory" "Mobile"             "Mobile Phone"
[6] "Others"
```

Figure.8 Redundant Levels of Specified Categorical Variables

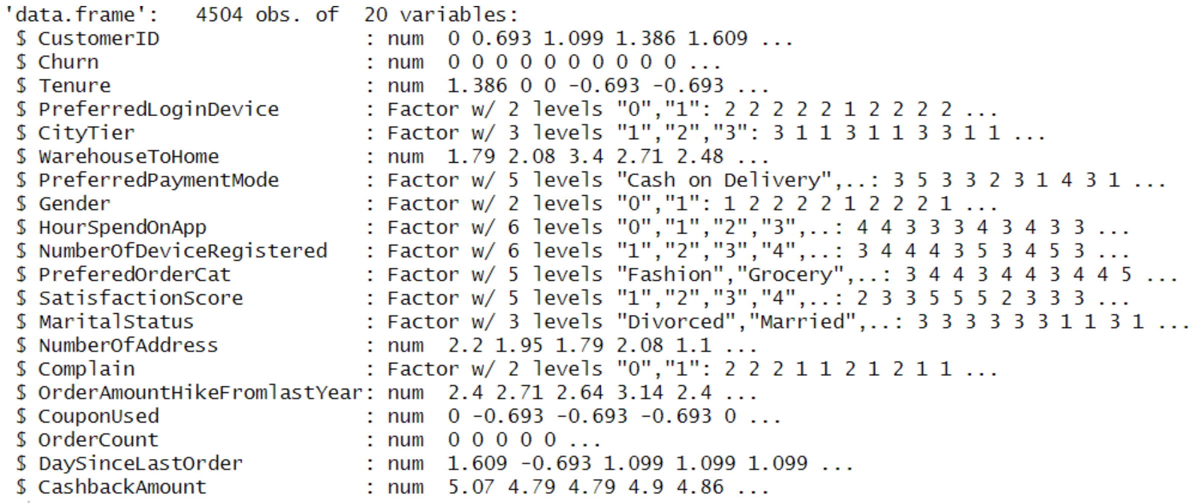
Figure.9 displays the modifications made on the redundant levels of the specified variables.



```
$PreferredLoginDevice
[1] "0" "1"

$PreferredPaymentMode
[1] "Cash on Delivery" "Credit Card"      "Debit Card"       "E wallet"         "UPI"

$PreferedOrderCat
[1] "Fashion"            "Grocery"            "Laptop & Accessory" "Mobile Phone"       "Others"
```

Figure.9 Adjusted Levels Redundancy of Specified Categorical Variables

Lastly, Adjustments regarding inconsistencies of data types/classes plus redundancies can be shown in Figure.10 and compared in conjunction with Figure.6 for clarification.



```
'data.frame':   4504 obs. of  20 variables:
 $ CustomerID                 : num  0 0.693 1.099 1.386 1.609 ...
 $ Churn                      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Tenure                     : num  1.386 0 0 -0.693 -0.693 ...
 $ PreferredLoginDevice       : Factor w/ 2 levels "0","1": 2 2 2 2 2 1 2 2 2 2 ...
 $ CityTier                   : Factor w/ 3 levels "1","2","3": 3 1 1 3 1 1 3 3 1 1 ...
 $ WarehouseToHome            : num  1.79 2.08 3.4 2.71 2.48 ...
 $ PreferredPaymentMode       : Factor w/ 5 levels "Cash on Delivery",..: 3 5 3 3 2 3 1 4 3 1 ...
 $ Gender                     : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 2 2 2 1 ...
 $ HourSpendOnApp             : Factor w/ 6 levels "0","1","2","3",..: 4 4 3 3 3 4 3 4 3 3 ...
 $ NumberOfDeviceRegistered   : Factor w/ 6 levels "1","2","3","4",..: 3 4 4 4 3 5 3 4 5 3 ...
 $ PreferedOrderCat           : Factor w/ 5 levels "Fashion","Grocery",..: 3 4 4 3 4 4 3 4 4 5 ...
 $ SatisfactionScore          : Factor w/ 5 levels "1","2","3","4",..: 2 3 3 5 5 5 2 3 3 3 ...
 $ MaritalStatus              : Factor w/ 3 levels "Divorced","Married",..: 3 3 3 3 3 3 1 1 3 1 ...
 $ NumberOfAddress            : num  2.2 1.95 1.79 2.08 1.1 ...
 $ Complain                   : Factor w/ 2 levels "0","1": 2 2 2 1 1 2 1 2 1 1 ...
 $ OrderAmountHikeFromlastYear: num  2.4 2.71 2.64 3.14 2.4 ...
 $ CouponUsed                 : num  0 -0.693 -0.693 -0.693 0 ...
 $ OrderCount                 : num  0 0 0 0 0 ...
 $ DaySinceLastOrder          : num  1.609 -0.693 1.099 1.099 1.099 ...
 $ CashbackAmount             : num  5.07 4.79 4.79 4.9 4.86 ...
```

Figure.10 Data Structure of Variables "After adjustments"

NA values:

In the previous section of Data Exploration, missing values were meticulously elaborated along with their common types that are encountered generally to data scientists or analysts. Furthermore, some of the most common and effective techniques of handling Nas were introduced and explained.

In this section we delve into the handling of NA values in the given dataset along with the techniques used to tackle them. First of all, an idea can be grasped about the type of the NA values encountered in the given dataset using some visual representation tools and statistical analysis.

The "mice" package in R is designed to help visualize, analyse and handle missing data in a dataset. It provides various functions and tools to understand the patterns and types of missing data in a given dataset. One of the key functions in this package is md.pattern(), which creates a matrix-style plot showing the presence and absence of missing values for each variable shown in Figure11.

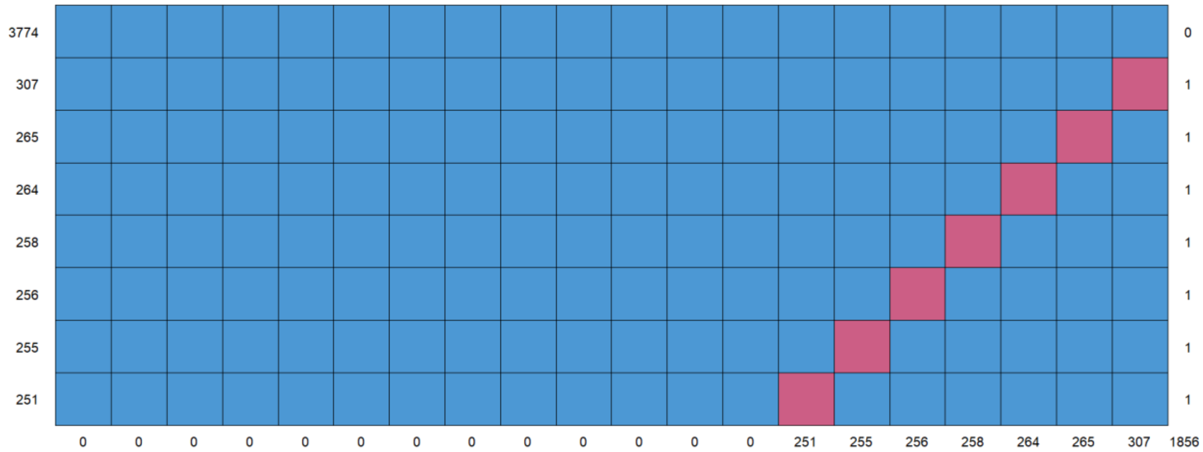


Figure11 NAs Pattern Visualisation

Observing Figure11, each column represents a variable, and each row represents a pattern of missingness. Furthermore, each pink box represents missing values with respect to the row’s corresponding pattern, and blue boxes mean no missing values. The column with numbers on the left side of the figure expresses the pattern of missing values observed with the upper first value expressing the total number of non-missing rows (observations). The column with numbers right to the figure expresses how many variables are missing with respect to each pattern with the bottom value expressing total number of missing values which Is the sum of the bottom row that represents number of missing values for each variable.

The last seven variables are the ones with missing values as displayed which can be identified in Figure.5. Moreover, each variable has all its missing values present in only one pattern, this indicates that no missingness pattern is shared among variables. This can relay that the missingness encountered is not related and there is no systematic missingness in the data.

Next comes the issue of how to handle our missing data. As mentioned in the Data Exploration chapter, there are mainly two options, either delete the rows/variables having missingness or impute them with one of the methods mentioned.

Hereafter, another key function of “mice” package is mice() function, which provides a solution for handling Nas by generating multiple imputations for multivariate missing values. This approach utilizes the Fully Conditional Specification technique, imputing incomplete variables with separate models. The function is capable of imputing various data types like continuous, binary, unordered categorical, and ordered categorical data. The default imputation methods in the function encompass various techniques. The package employs PMM (Predictive Mean Matching) for numeric data, LOGREG (logistic regression) for binary data and factors with two levels, POLYREG (polytomous regression) for unordered categorical data with more than two levels, and POLR (proportional odds) for ordered categorical data with more than two levels.

Since all of our missing values are numeric, before adjusting data types/classes as shown in Figure.6, PMM shall be used as a method of imputation by mice() (Buuren & Groothuis-Oudshoorn, 2011).

Furthermore, to get an idea about how PMM works, consider a scenario where a variable x has some instances with missing data, while a set of variables y, devoid of any missing data, are utilized to impute x. The following five steps are involved (Allison, 2022):

1. For instances with complete data, calculate a linear regression of x on y, generating a set of coefficients denoted as 'b'.
2. Derive a random draw from the "posterior predictive distribution" of 'b', resulting in a new set of coefficients termed 'b*'. Typically, this involves a random draw from a multivariate normal distribution, with 'b' as the mean and the estimated covariance matrix of 'b' (with an additional random draw for residual variance).
   This step introduces the necessary variability in the imputed values and is a standard practice for all robust methods of multiple imputation.

3. Using 'b*', predict values for x across all instances, encompassing both those with missing x and those with available x data.
4. For instances with missing x, identify a subset of instances with observed x values whose predicted values closely match the predicted value for the instance with missing data.
5. From among the closely matched instances, randomly select one and substitute the missing value.

In addition, in contrast to numerous imputation techniques, the primary objective of the linear regression in this context is not to directly produce imputed values. Instead, its role is to establish a measure for aligning instances with missing data to comparable instances with complete data (Allison, 2022).

After that, the outcome of the function mice() is of class “mids” (Multiple Imputed Datasets) which is commonly used in “mice” package. To obtain a data frame of the completed data set the function complete() in “mice” package does that with its argument as the output obtained from the function mice().

Methodologies

The methodologies employed in this study encompass a diverse range of statistical and machine learning techniques. For the purpose of this thesis, supervised machine learning algorithms are to be employed given the nature of the dataset on hand being labelled with focus on classification algorithms since we are dealing with a classification problem. Three distinct approaches, namely **Logistic Regression, k-Nearest Neighbours (kNN)**, and **Random Forest**, have been strategically chosen to make informed predictions of the dataset under examination.

The choice of those mentioned classification supervised learning algorithms is made based on two main points:

1- The unique characteristics of the dataset on hand.
2- The unique perception and value provided by each algorithm given the unique characteristics of dataset.

The two points shall be elaborated in detail to motivate the choice of the algorithms mentioned.

The unique characteristics of this dataset can be inferred based on the nature of the variables and their potential impact on customer churn prediction. Here are key characteristics:

- **Customer Churn Prediction:** The dataset is primarily focused on predicting customer churn, which renders the modelling to be a classification problem for a binary label.
- **Mixed Data Types:** The dataset contains a mix of data types, including binary, categorical, ordinal, and numerical variables.
- **No Assumption About Data Distribution:** The predictors in the dataset are not clearly following any statistical distribution and no assumption has been made to assert so.
- **Class Imbalance:** Since the label "Churn" is binary (1 for churn, 0 for no churn) and most of which are 0s, the dataset exhibits class imbalance, which is important to consider during modelling.

Moreover, a brief definition of the chosen algorithms along with the unique perception and value provided by each can be summarised as follows:

Logistic Regression is a fundamental statistical method that assumes the form of a logistic function to model the relationship between a binary dependent variable and one or more independent variables. In this study, Logistic Regression is utilized to uncover associations and the impact of predictor variables on a binary outcome. Its interpretability and ability to estimate probabilities make it invaluable for examining relationships within the dataset.

Unique perception and value: Logistic Regression is well-suited to the dataset's characteristics because (Saini, 2021):

- It can handle mixed data types, including numerical, binary, and categorical features.
- It is interpretable, which is beneficial when one wants to understand the impact of each feature on churn.
- It does not assume a specific data distribution. It works well when the data distribution is not clearly Gaussian or when one does not have prior knowledge of the data distribution.

Nearest Neighbours (kNN), on the other hand, is an instance-based machine learning algorithm known for its simplicity and efficacy. kNN operates by identifying the "k" nearest neighbours to an observation, thus assigning a class or value based on their collective characteristics. In this study, kNN serves as a versatile tool for classification tasks, utilizing the similarity between data points to make predictions.

Unique perception and value: kNN excels at capturing local information of predictors. This can be valuable if churn patterns are influenced by local customer behaviour or preferences which are conveyed by multiple predictors in the dataset. Link to dataset’s characteristics can be summarised as follows (Harrison, 2019):

- It can handle both numerical and categorical data(needs preprocessing), making it suitable for the dataset on hand.
- it does not make strong assumptions about the underlying data distribution.
- it is generally robust to outliers since it relies on the similarity of data points.
- It does not get affected by class imbalance in some cases (which is the case here, more on this in Results chapter) if one chooses k = 1, therefore classify based on nearest point.

Random Forest, a robust ensemble learning method, harnesses the collective wisdom of numerous decision trees to enhance predictive accuracy and mitigate overfitting. It excels in both classification and regression tasks, making it well-suited for uncovering complex relationships within the data.

Unique perception and value: According to Sruthi E.R.(2023), random Forest is a strong choice for the given dataset due to the following links to the dataset’s characteristics:

- It can effectively handle mixed data types, including numerical, binary, and categorical features (needs preprocessing).
- It can capture non-linear relationships (e.g., interactions between PreferredLoginDevice and HourSpendOnApp) and complex patterns.
- it provides feature importance scores, allowing one to identify which features have the most influence on the target variable ('Churn').
- It can handle large datasets efficiently, making it suitable for datasets with a significant number of observations.
- It tends to be robust to the inclusion of irrelevant features, reducing the risk of overfitting due to noisy predictors.

Each of these methodologies is chosen with a specific objective in mind along with their adherence to unique characteristics of the dataset. While Logistic Regression provides interpretability, kNN capitalizes on local patterns in data, and Random Forest excels at capturing complex interactions.

Finally, having discussed the algorithms to be employed in this thesis, more elaboration shall be provided on how to evaluate the performance of classification models. The performance of a classification problem is analysed through a confusion matrix. Table.3 displays the components of a confusion matrix as follows.

Table.3 Confusion Matrix General Form

Confusion Matrix	**Actual Positive**	**Actual Negative**
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Furthermore, the following metrics are commonly derived and studied through a confusion matrix (Kundu, 2020):

- **Sensitivity** (True Positive Rate or Recall): Sensitivity measures the proportion of actual positive cases that are correctly identified by the model. It is calculated as the ratio of true positives to the sum of true positives and false negatives. A high sensitivity indicates that the model is good at identifying actual positives, minimizing false negatives.

  $$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{1}$$

- **Specificity** (True Negative Rate): Specificity measures the proportion of actual negative cases that are correctly identified by the model. It is calculated as the ratio of true negatives to the sum of true negatives and false positives. A high specificity indicates that the model is good at identifying actual negatives, minimizing false positives.

  $$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \tag{2}$$

- **Precision**: Precision assesses the accuracy of positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives. A high precision score signifies that among the instances predicted as positive, a large proportion are genuinely positive, reducing the occurrence of false positives in the model's predictions.

  $$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{3}$$

- **F1 Score**: The F1 score is the harmonic mean of precision and recall (or sensitivity). It balances precision and recall. F1 score provides a single metric that considers both false positives and false negatives, making it useful when there is an imbalance between classes.

$$F1\ Score = 2 * \frac{Precison * Sensitivity}{Precison + Sensitivity} \quad (4)$$

Moreover, the F1 score can be adjusted to account for weights to precision and recall, since there are different costs associated with different errors.

- **Weighted F1 Score:** It provides a single value that considers both false positives (FP) and false negatives (FN) by giving more or less importance to specific classes. β is a constant used to specify weights, values higher than 1 provides more weight to sensitivity, while values between 0 < β <1 provide more weight to precision the less the value is.

  $$Weighted\ F1\ Score = \frac{(1+\beta^2) Precison * Sensitivity}{(\beta^2 * Precison) + Sensitivity} \quad (5)$$

These metrics offer insights into how well a classification model performs in distinguishing between classes, and they help to gauge the trade-offs between correct identifications and misclassifications in different scenarios (Kundu, 2022).

In the context of this thesis, sensitivity is associated with the model's capabilities in recognizing customers who shall remain. On the other hand, specificity is associated with the model's capabilities in predicting customers who shall churn. Therefore, given the nature of this study is the focus on customers' churn, hence, more focus shall be directed towards specificity.

Nevertheless, in the context of business, when analysing the two measures sensitivity and specificity, each one is associated with a cost to the decision makers. For instance, say a model's performance through sensitivity is 30%, however its specificity is 95%. In conditional probabilities context, if Y = 1 is the reference churn and $\hat{Y}$ = 1 is the predicted churn events, then:

Sensitivity = P($\hat{Y}$ = 0|Y = 0) = 0.3, that is the probability of a customer who remains given they actually remained. On the other hand, its compliment (type 2 error):

P($\hat{Y}$ = 1|Y = 0) = 0.7 is the probability that a customer churns given they actually remained which is quite high. Therefore, this error of falsely predicting a churn of customers who actually will remain is a cost that the company would incur; by for instance, giving out promotions or vouchers to customers who were never at risk of churning.

Specificity = P($\hat{Y}$ = 1|Y = 1) = 0.95, that is the probability of a customer who churns given that they actually churned. On the other hand, its compliment (type 1 error):

P($\hat{Y}$ = 0|Y = 1) = 0.05 is the probability that a customer remains given they actually churned which is quite low and at a desirable level. However, if this error was to be high, then, the lost opportunity cost here would be higher than type 2 error's cost since we will be losing a customer and their future purchases (Reichheld, 2001).

To sum up, emphasis is needed on the specificity measure, or reducing type 1 error as much as possible, since it is the core metric for this paper. However, type 1 error should as well be at an acceptable level for the model to make sense in practice. Therefore, for performance metrics of algorithms employed in this paper shall be the analysis of both sensitivity and specificity along with accuracy, since the former two measures together capture the essence of the performance of the classification models to be obtained, and the latter provides a view of how one metric alone could be misleading in case of data imbalance.

Results and Analysis

This chapter dives into the results obtained from data cleaning and preprocessing and the results using the three mentioned approaches along with expounding on implementation and analysis. Additionally, it will elaborate on how each model has been through many iterations in order to achieve its most effective form possible in order to address the research questions. The aim is to uncover the most effective model in terms of performance along with analysing predictive insights and identifying influential factors for customer retention in an e-commerce online platform.

Having done some preprocessing to the dataset, it is roughly ready for machine learning implementation; however, more specific tuning is needed for each algorithm to attain better results. Moreover, many iterations of data preprocessing have been applied to try and improve results, and they shall be elaborated on.

Six iterations have been conducted to reach the final model for each of the employed algorithms. Each iteration was involved in different trials of data preprocessing in order to obtain the best possible final model. It happened that all algorithms needed six iterations in order to obtain their corresponding final best possible model. For the sake of simplicity, confusion matrices of each iteration have been tabulated and displayed at the beginning of the results part of each algorithm.

Logistic regression

Using glm() function with the argument family = “binomial” will create our logistic model. Next, using the function predict() with the argument type =”response” will result in a vector of probabilities for each observation being admitted to the specified class. After that, the choice of a threshold for classifying points into 1 (Churn) or 0 (No Churn), based on the predicted probabilities of Logistic Regression, can experimented with in order to find the optimal threshold that produces the most accurate results. After some experimentation of different thresholds, a threshold of 0.5 has been chosen as it yielded the best results, such that if predicted_probability > 0.5 classify 1 and 0 otherwise.

Results

Table.4 Logistic Regression Iteration 1-a Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	899	96
	1	35	96
Accuracy	0.9147		
Sensitivity	0.9625		
Specificity	0.5000		

Table.5 Logistic Regression Iteration 1-b Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	898	91
	1	36	101
Accuracy	0.9182		
Sensitivity	0.9615		
Specificity	0.5260		

Table.6 Logistic Regression Iteration 2-a Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	899	94
	1	33	97
Accuracy	0.9166		
Sensitivity	0.9646		
Specificity	0.5079		

Table.7 Logistic Regression Iteration 2-b Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	891	90
	1	41	101
Accuracy	0.9174		
Sensitivity	0.9560		
Specificity	0.5288		

Table.8 Logistic Regression Iteration 4 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	898	77
	1	36	115
Accuracy	0.9288		
Sensitivity	0.9615		
Specificity	0.5990		

Table.9 Logistic Regression Iteration 5 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	904	70
	1	30	122
Accuracy	0.9358		
Sensitivity	0.9679		
Specificity	0.6354		

Table.10 Logistic Regression Iteration 6 Results-Final Iteration

Confusion Matrix

		Reference	
		0	1
Prediction	0	772	29
	1	162	163
Accuracy	0.8304		
Sensitivity	0.8266		
Specificity	0.8490		

First iteration was to try a logistic model with the configuration of data types seen in Figure10. However, the variables “NumberOfAdressess”, “OrderAmountHikeFromlastYear” and “CouponUsed” seem to be behaving somewhere between numerical-discrete and categorical-ordinal variables. Therefore, the model shall be tested with them being examined as both interchangeably, after that, the model shall be assessed and adjusted accordingly. The results are shown in Table.4. Investigating the results in Table.4 shows that the model is performing well in capturing customers who shall remain, displayed by the sensitivity measure, however, it’s quite low basically a coin flip in capturing customers who shall churn displayed by specificity which is our focus for this study.

Trying logistic regression again, however, treating “NumberOfAdressess”, “OrderAmountHikeFromlastYear” and “CouponUsed” as factors improved the results a bit in terms of specificity measure as shown in Table.5. In the second iteration, scaling numerical predictors and removing outliers were examined. Since numerical predictors are not on the same scale, it might be a promising idea to try and scale them around 0 and test the results. The results are displayed in Table.6. It is evident that scaling and removing outliers have resulted in a less performing model than the previous iteration, displayed by the specificity measure. However, having examined only scaling numerical predictors or only removing outliers, the problem seemed to be associated with scaling. Having only removed outliers in this iteration resulted in the results displayed in Table.7. Nevertheless, compared to the final performance of the first iteration displayed in Table.5, the results are almost the same. Therefore, for the sake of generalization and not overfitting our model, opting for not removing outliers seems to be the wise choice in this case.

For the third iteration, one hot encoding categorical variables has been examined. Namely, “PreferredPaymentMode”, “PreferedOrderCat” and “MaritalStatus”. Moreover, what this does is each category within a categorical feature is transformed into a binary vector where each category is represented by a column, a '1' indicates the presence of that category while '0' indicates absence. However, the model’s performance was almost the same with or without one hot encoding the mentioned categorical variables, mainly due to the fact that logistic regression internally deals with that issue.

For the fourth iteration, transforming numerical predictors was examined. There are many ways to transform numerical predictors such as log and square root transformations. Hereafter, natural log, log2, log10 and square root transformations were examined, and the best performance was obtained using the natural log transformation. Moreover, a buffer of 0.5 was added to values which are 0 in order to avoid undefined outcomes. Table.8 display the results of this iteration. The natural log transformation of numerical predictors shows a more promising model’s performance than scaling. While sensitivity measure is almost the same, there is a considerable improvement in the observed specificity which is almost 60%.

Going to the fifth iteration, unfactorising the three predictors mentioned in iteration one was tested, since more preprocessing has been performed, along with studying correlation between the numerical predictors will be examined. Having examined unfactorising the three mentioned variables interchangeably, unfactorising “CouponUsed” enhanced the model by fractions, specifically, it improved sensitivity by fractions which in turn made the accuracy hit 90% mark. Therefore, it shall be kept as a numerical-discrete variable. After that, correlation analysis has been performed shown in Figure12.

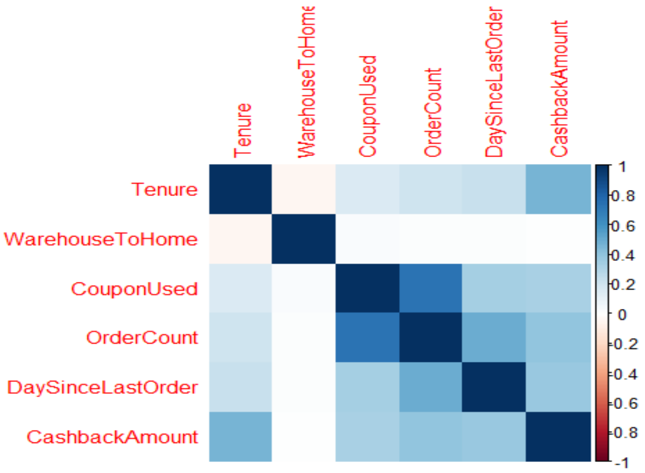


Figure12 Correlation Heat Map of Numerical Variables

Figure12 shows some correlations between the variable “CashbackAmount” and some others. In addition, there is a somewhat strong correlation between the variables “OrderCount” and “CouponUsed”. Lastly, some correlation can be seen between the variable “DaySinceLastOrder” and some others as well. Therefore, removing those variables interchangeably have been examined and it seems that the choice of removing the variables “DaySinceLastOrder”, “OrderCount” and “CashbackAmount” was the optimal combination and have improved the results as displayed in Table.9 .As displayed in Table.9, it is evident that the results have improved through our main measure specificity, enhancing the model even further from a specificity of almost 60% to 63% while improving a bit on the sensitivity as well.

On the last sixth iteration of the logistic model, the issue of unbalanced label shall be examined. The function downSample() has been used to randomly reduce the sample size of the dataset from the majority label 0 to match the minority label in size, so we end up having a balanced dataset of which 50% is labelled 0 and the rest 1. Moreover, again another examination of unfactorising the two variables “NumberOfAddress” and “OrderAmountHikeFromlastYear” interchangeably to examine their effect as numerical variables resulted in an improved specificity by 2% when only unfactorising “OrderAmountHikeFromlastYear”. The results of this iteration are displayed in Table.10. Despite the sensitivity reduction and therefore the accuracy’s as well, the specificity has improved significantly reaching 84% while its sensitivity counterpart is at 82% which is a more desirable performance than the previous iteration given the aim of this study.

Analysis

The approach in examining a logistic regression model is done by mainly looking into two aspects:

1- The coefficients associated with each predictor.
2- Their corresponding P-value calculated using hypothesis testing under Null hypothesis that the coefficient is 0 which is obtained by default through the model.

Appendix A has the two aspects laid out for all predictors which was obtained using a summary of the final model in iteration 6.

In logistic regression, the log odds of the probability of a certain outcome (e.g., event occurrence) is calculated based on the predictor variables. When fitting the logistic regression model, the coefficients are estimated for each predictor variable. These coefficients indicate the change in log odds of the predicted outcome associated with a one-unit change in the predictor, assuming all other predictors remain constant (Kanade, 2022).

$$\textit{Odds Ratio} = \frac{p}{1-p} \qquad (6)$$

Where: p is the probability of success or in this case the probability of predicting class 1

After obtaining the coefficients from the logistic regression model, the log odds for a particular observation is calculated using the formula:

$$\textit{Log odds} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \qquad (7)$$

Where:

- β_0 is the intercept term.
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for the predictor variables $x_1, x_2, \ldots, x_n$.

These log odds can then be transformed into probabilities using the logistic function (sigmoid function) to obtain the conditional probability of the event occurring:

- $P(Y = 1|X) = \frac{1}{1+ e^{-\,log\,odd}}$ *(8)*

Where: Y is the binary outcome and X is the set of predictors.

This probability is what logistic regression predicts based on the given values of predictor variables for a particular observation (Kanade, 2022).

Furthermore, since there are two main data types, numerical and categorical, the coefficients obtained for numerical variables can be interpreted by just exponentiating them to get the odds ratio. For instance, the variable “Tenure” has a coefficient of -1.18696, therefore, exp(-1.18696) = 0.305 and since the value is less than one this means that a 1-unit increment in “Tenure” reduces the odds of a churn by a factor of 0.305 assuming all other predictors remain constant. On the other hand, the coefficients obtained for categorical variables are compared to a reference level of the category. For instance, the categorical variable “CityTier” has three levels (1,2,3), in the model’s summary the statistics are only shown for levels (2,3) because they are being compared to level (1). Hence, the coefficient of “CityTier2” level is 0.33422, exp(0.33422) = 1.397 and since it’s more than one this means that the odds of a churn given a customer is from “CityTier2” is increased by a factor of 1.397 when compared to a customer from “CityTier1”.

Nevertheless, a more detailed elaboration of the model’s summary is tabulated as below. Note that the brief analysis assumes that all other variables are held constant when making those statements. The coefficients mentioned below are the estimates obtained from the logistic regression as stated in equation 7. A full summary of the algorithm’s result including P-values and the coefficients is available in Appendix A.

Variable	Brief Analysis
Tenure	The P-value is almost 0, hence, the coefficient’s implication is statistically significant. Coefficient is negative which means the more the Tenure of customers the less likely a churn occurs.
PreferredLoginDevice Reference level is 0 (Computer), 1 is (Phone)	The P-value is almost 0, hence, the coefficient’s implication is statistically significant. Coefficient is negative which means that customers logging in from phones indicate less likelihood of churn compared to logging in from computers.
CityTier Reference level is 1	CityTier2 P-value is > 0.05 which indicates that the coefficient is not statistically significant, however CityTier3 P-value indicates that its coefficient is statistically significant. Coefficient of CityTier3 is positive which means customers from CityTier3 have more likelihood of churn compared to customers from CityTier1.
WarehouseToHome	The P-value is almost 0, hence, the coefficient’s implication is statistically significant. Coefficient is positive which means the more

	the WarehouseToHome distance for customers the more likely a churn occurs.
PreferredPaymentMode Reference level is CashOnDelivery	The levels Credit Card, Debit Card, Wallet and UPI All have P-values <= 0.05 which indicates that their corresponding coefficients are statistically significant. Their corresponding coefficients are all negative, which indicates that customers who pay using these levels are less likely to churn compared to customers who pay using CashOnDelivery. Among the mentioned levels, Credit Card has the most negative coefficient, meaning when comparing customers who pay using the mentioned levels against customers who pay using cash, the ones using credit cards are the least likely to churn.
Gender Reference level is 0 (Female), 1 (Male)	The P-value associated with the level Male > 0.05, hence, the corresponding coefficient is statistically insignificant. Meaning, being Male or Female does not provide information for customer churn.
HourSpendOnApp Reference level is 0	The levels 1, 2, 3, 4, 5 all have P-values > 0.05 which indicates that their corresponding coefficients are statistically insignificant. Meaning, the time spent on app by customers ,being any of the mentioned levels, when compared to spending 0 hours doesn't provide information on customer churn.
NumberOfDevice Registered Reference level is 1	The P-value associated with level 2 > 0.05 which indicates that its corresponding coefficient is statistically insignificant, hence, having 2 devices compared to 1 does not provide information on customer churn. However, for levels 3, 4, 5, 6 the P-values are <= 0.05 which indicates that their corresponding coefficients are statistically significant. Their corresponding coefficients are all positive, which means that having 3, 4, 5 or 6 devices registered indicates more likelihood of churn compared to having 1 registered device. Moreover, their coefficients are increasing as the level increases, meaning that the more the devices the more likelihood of churn compared to having 1 device registered.
PreferedOrderCat Reference level is Fashion	The levels Grocery, Laptop & Accessory and Mobile Phone all have P-values <= 0.05 which indicates that their corresponding coefficients are statistically significant. On the other hand, the level Others has P-value > 0.05 which indicates that its coefficient is statistically insignificant. The levels Grocery, Laptop & Accessory and Mobile Phone all have negative coefficients implying that customers with preferred order category being one of the mentioned three is less likely to churn compared to customers with Fashion as their preferred category. Furthermore, the level Laptop & Accessory has the most negative coefficient, indicating that among the three mentioned categories, when compared to customers who prefer Fashion, customers who prefer Laptop & Accessory are the least likely to churn. Conversely, customers who prefer Others are more likely to churn compared to customers who prefer Fashion.
SatisfactionScore Reference level is 1	The P-value of level 2 is > 0.05 which indicates that its coefficient is statistically insignificant, hence its coefficient does not provide information on customer churn. For levels 3, 4, 5 their P-values are <= 0.05 which indicates that their corresponding coefficients are statistically significant. Their corresponding coefficients are positive and increasing, as the level increases, which means that the more the satisfaction score the more likely the customer will churn compared

	to customers leaving a 1 score. (counterintuitive)*
MaritalStatus Reference level is Divorced	The levels Married and Single both have P-values <= 0.05 which indicates that their corresponding coefficients are statistically significant. The coefficient of Married is negative, which implies that Married customers have less likelihood of churn. On the other hand, the coefficient of Single is positive, which implies that Single customers have more likelihood of churn both when compared to Divorced customers.
NumberOfAddress Reference level is 1	The levels 2 to 11 have P-values <= 0.05 which indicates that their corresponding coefficients are statistically significant. On the other hand, the remaining levels 19 and 20 are outliers with P-values almost 1 and no added information on customer churn. The coefficients of levels 2 to 11 are positive and increase as the level increases, which means that as the addresses of customers increase so does their likelihood of churn compared to customers with only 1 address.
Complain Reference level is 0 (no complain), 1 (complain)	The P-value of the level Complain is almost 0, hence, its corresponding coefficient is statistically significant. Coefficient is positive which means that customers who raise complaints are more likely to churn compared to customers no complaints.
OrderAmount HikeFromlastYear	The P-value is almost 0, hence, the coefficient's implication is statistically significant. Coefficient is negative which means that the more the OrderHike is the less likely a churn occurs.
CouponUsed	The P-value is > 0.05 which indicates that tis coefficient is statistically insignificant.

* The results for the impact of SatisfactionScore on Churn seems to be counterintuitive. There may be many assumptions/explanations which can be drafted in this case which needs further analysis. One for instance could be that since the results are compared to the reference level 1, customers who are loyal will be dissatisfied when the service is not at their desirable level, therefore, they vote 1 to show their dissatisfaction. On the other hand, customers who vote 3,4 or 5 compared to the ones who vote 1 might be temporary customers who just wanted to buy a product and never return. However, it is only an assumption with no supporting evidence, further investigations are beyond the scope of this paper.

kNN

Using the function train() from caret package in R, with argument method = "kNN" would train our model and bootstrap resample from the dataset to find the best tune for the parameter k, which by default k is examined as (5,7,9). After that, the function predict() will classify our unseen test dataset to obtain the desired results.

Results

Table.11 kNN Iteration 1 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	909	126
	1	25	66
Accuracy	0.8659		
Sensitivity	0.9732		
Specificity	0.3438		

Table.12 kNN Iteration 2 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	909	84
	1	25	108
Accuracy	0.9032		
Sensitivity	0.9732		
Specificity	0.5625		

Table.13 kNN Iteration 3 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	919	67
	1	15	125
Accuracy	0.9272		
Sensitivity	0.9839		
Specificity	0.6510		

Table.14 kNN Iteration 5 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	924	21
	1	10	171
Accuracy	0.9725		
Sensitivity	0.9893		
Specificity	0.8906		

Table.15 kNN Iteration 6 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	921	7
	1	13	185
Accuracy	0.9822		
Sensitivity	0.9861		
Specificity	0.9635		

For the first iteration, using the same preprocessed dataset from the fifth iteration of logistic model was used as a base iteration, reason for not using the final sixth iteration is that we do not need to balance dataset. The results are displayed in Table.11.
The model accurately predicts customers who shall remain given the high sensitivity measure, however it does horribly in predicting customers who shall churn given the extremely low specificity measure. This is mainly due the specific data processing to the logistic model, having transformed numerical variables with a logarithmic scale.

In the second iteration, scaling all variables which have numerical entries, which are all variables besides binary and categorical-nominal variables, therefore, all categorical-ordinal and numerical-discrete/continuous have been examined while leaving binary and categorical-nominal variables as is. The goal here is to try and assess the model having all numeric entries scaled around 0 since kNN is very sensitive to that, the details of the updated data structures are present in Appendix B-Table.1. The results are displayed in

Table.12.The results did improve from the previous iteration, displayed by the enhanced performance in the specificity measure. However, it is still not performing as well as desired and more tuning is needed.

For the third iteration, leaving binary along with ordinal variables as is and scaling the rest of numerical variables along with one hot encoding categorical-nominal, namely "PreferredPaymentMode", "PreferedOrderCat" and "MaritalStatus" variables were examined, the details of the updated data structures are present in Appendix B-Table.2. The results are shown in Table.13. There is a considerable improvement on performance for this iteration. Both sensitivity and specificity have gone up, while the latter have improved considerably.

Moreover, going into the fourth iteration, the same preprocessing done to iteration three was applied along with removing outliers. However, there was no improvement and the model retained same performance. This is due to the fact that all the variables with outliers have been scaled, therefore, the effect of outliers could not be prominent.

In the fifth iteration, the tuning parameter K was modified from the values of (5,7,9) to (1:5). Since the prevalence of our class 1 is very low, it is expected that trying lower values may improve results, especially our specificity measure. Table.14 elaborates on the results for this iteration. Indeed, having K = 1, hence choosing the nearest point only to classify new datasets has improved the model's predictive capabilities immensely. The specificity now is almost at 90% which is a noticeable improvement of the model's performance.

Furthermore, going to the last and sixth iteration, examining those numerical variables that have been scaled further to try and see if a certain combination is to be scaled and not all would enhance the performance of the model even further. Having interchangeably examined many combinations of scaling the numerical variables, scaling only the variables "WarehouseToHome", "NumberOfAddress", "CouponUsed" and "CashbackAmount" have improved the performance further. Table.15 displays the results of this iteration. As seen in Table.15, the results have improved even further from the previous iteration. The model's predictive capabilities are now almost perfect, with both measures, sensitivity and specificity being at a very desirable level.

Analysis

In kNN algorithm, Euclidean distance is a common metric used to measure the distance between two points in a multidimensional space. Given two points in an n-dimensional space, p and q, the Euclidean distance between them is calculated using the following formula:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \qquad (9)$$

A breakdown of the components is elaborated as follows:

- p and q are two points in the n-dimensional space, where p can be a point (observation in our dataset), and q can be a new point to be classified.
- pi and qi represent the ith component of points p and, respectively.

- n is the number of dimensions (features) in the dataset.

The calculation involves subtracting corresponding coordinates of each dimension, squaring the differences, summing these squared differences across all dimensions, and finally taking the square root of the sum to obtain the Euclidean distance (Abba, 2023).

In kNN, during the prediction phase, when a new data point is presented, the algorithm calculates this Euclidean distance between the new point and all the points in the training dataset. It then identifies the 'k' nearest neighbours (data points) to the new point based on these distances, in our final iteration k was set to 1. And finally, the class of the nearest point is used to predict the label of the new data points.

Furthermore, compared to logistic regression, kNN does not provide any more insights on the complexities inside the dataset and between predictors and the target variable. However, for such classification problems, it indeed possesses powerful prediction capabilities.

Random Forest

Using train() function from caret package in R with the argument method = “rf” would train our model with random forest algorithm. Moreover, this method has a tuning parameter called mtry, which determines the number of variables randomly sampled at each split. The default examining of mtry is (2,14,26) predictors. After that, using the function predict() on the test dataset will classify the new datapoints according to our model.

Results

Table.16 Random Forest Iteration 1 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	924	25
	1	10	167
Accuracy	0.9689		
Sensitivity	0.9893		
Specificity	0.8698		

Table.17 Random Forest Iteration 2 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	924	24
	1	10	168
Accuracy	0.9698		
Sensitivity	0.9893		
Specificity	0.8750		

Table.18 Random Forest Iteration 3 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	924	22
	1	10	170
Accuracy	0.9716		
Sensitivity	0.9893		
Specificity	0.8854		

Table.19 Random Forest Iteration 4 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	923	21
	1	11	171
Accuracy	0.9716		
Sensitivity	0.9882		
Specificity	0.8906		

Table.20 Random Forest Iteration 5 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	925	20
	1	9	172
Accuracy	0.9742		
Sensitivity	0.9904		
Specificity	0.8958		

Table.21 Random Forest Iteration 6 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	857	8
	1	184	184
Accuracy	0.9245		
Sensitivity	0.9176		
Specificity	0.9583		

First iteration was to examine the preprocessed dataset from the last iteration of kNN. Table.16 displays the performance of this iteration. The model seems to be performing well observing Table.16, it is performing way better than the final iteration of logistic regression. Furthermore, the model is performing very well in predicting customers who shall remain with almost 99% accuracy displayed by the sensitivity. However, compared to kNN it is still inferior in predicting customers who shall churn displayed by the specificity. More preprocessing might improve the performance of the model, which shall be examined further in the coming iterations.

In the second iteration, scaling all numerical or seemingly numerical variables was examined. Specifically, variables (3,6,14,16,17,18,19,20) respectively as listed in Figure10, along with one hot encoding categorical variables and retaining ordinal/binary variables as is. Table.17 displays the results of this iteration. Moreover, the performance of the model is the same as iteration 1 aside from one churn observation corrected.

In the third iteration, instead of scaling the mentioned variables, examining the natural log transformation was the next move. Since using the natural log transformation improved the performance in the logistic regression part, perhaps random forest would react positively to this as well. Table.18 outlines the results of the current iteration. Indeed, natural log transformation instead of scaling numerical predictors strengthened the model's performance further by capturing three more churns compared to the first iteration,

Going to the fourth iteration, instead of log transforming all mentioned variables, the variables that were optimally chosen to be scaled in the final kNN model were the only ones to be log transformed in this iteration. Table.19 displays the results of this iteration. The choice of log transforming the selected numerical variables indeed is optimal and enhanced the performance even further. Both measures are performing well, however, there is still room for further improvement.

In the fifth iteration, examining correlation between the numerical predictors is performed. During the fifth iteration of logistic regression, the variables "DaySinceLastOrder", "OrderCount" and "CashbackAmount" were the optimal combination to remove that improved the model. Examining omitting these variables interchangeably, resulted in the optimal choice of removing "OrderCount" and "CashbackAmount" which improved the results. Table.20 displays the performance of this iteration. Correlation analysis have improved the model even further. Both sensitivity and specificity have improved by fractions. Moreover, this iteration records the highest sensitivity, or in other terms predicting customers who shall remain, however, still short in what is more crucial in this paper which is the specificity measure compared to the final kNN model.

The last and sixth iteration examined balancing the dataset used in iteration five. The dataset has been balanced in the same matter used in logistic regression by down sampling. Table.21 displays the results of this iteration. As shown in Table.21, the sensitivity has been reduced mainly to the loss of information incurred by down sampling. On the other hand, the specificity measure improved tremendously to almost 96% which is close to the specificity obtained in the last kNN model.

Analysis

Random Forest operates by utilizing the use of many decision trees. It uses a metric called Gini Impurity, this is the metric used in our model some other models use different metrics, criterion in decision-making at each node within its assembly of decision trees. This process involves evaluating various features at each node to identify the most effective feature for splitting the data. The primary objective is to achieve nodes with the purest class distributions possible.

To determine the best split, the “rf” algorithm used here calculates the Gini impurity for each feature. It subsequently iterates through all available features and their potential thresholds, seeking the split that maximally reduces the Gini impurity in the child nodes. This recursive process constructs a tree structure by segmenting the data based on the selected features.

$$\textit{Gini Impurity} = 1 - \sum_{i=1}^{n}(p_i)^2 \qquad (10)$$

Where:

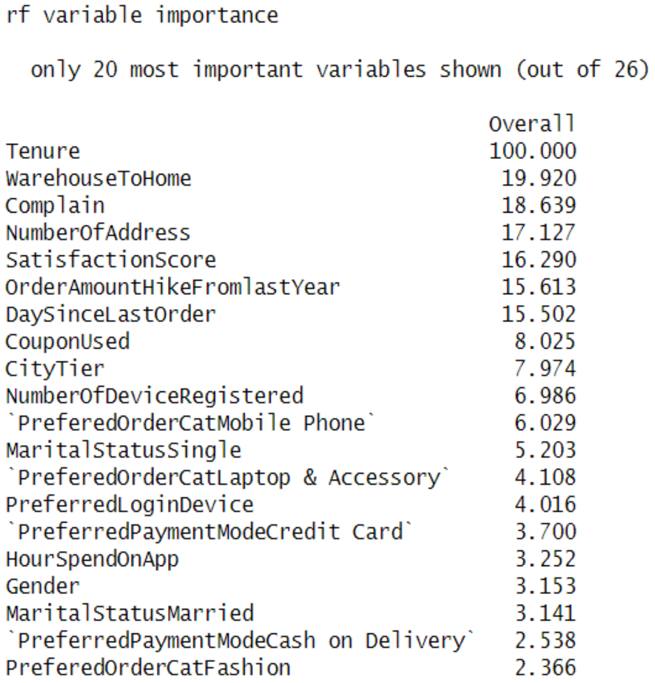
- n is the number of classes
- p_i the proportion of class i at each split

RF does not rely on a single decision tree but instead constructs multiple trees by bootstrapping the data and utilizing distinct subsets of features at each node. The model then combines the predictions from these trees, employing majority voting for classification tasks.

In classification scenarios, RF algorithm consolidates the individual tree predictions to assign the final class label through a majority voting scheme. This collective approach of utilizing multiple decision trees, each trained on varied data subsets and features, results in a robust and accurate model for classification purposes.

Moreover, the more a variable is utilized to make decisions that reduce impurity across all trees in the forest, the more important it is. Variable importance is computed by averaging the impurity decrease over all trees where a variable is utilized for splitting. Variables that lead to higher average impurity reduction across trees are considered more important. The table for variable importance obtained through the final model is displayed below in Table.22.

Table.22 Random Forest Variables Importance Based on Gini Index



```
rf variable importance

  only 20 most important variables shown (out of 26)

                                       Overall
Tenure                                 100.000
WarehouseToHome                         19.920
Complain                                18.639
NumberOfAddress                         17.127
SatisfactionScore                       16.290
OrderAmountHikeFromlastYear             15.613
DaySinceLastOrder                       15.502
CouponUsed                               8.025
CityTier                                 7.974
NumberOfDeviceRegistered                 6.986
`PreferedOrderCatMobile Phone`           6.029
MaritalStatusSingle                      5.203
`PreferedOrderCatLaptop & Accessory`     4.108
PreferredLoginDevice                     4.016
`PreferredPaymentModeCredit Card`        3.700
HourSpendOnApp                           3.252
Gender                                   3.153
MaritalStatusMarried                     3.141
`PreferredPaymentModeCash on Delivery`   2.538
PreferedOrderCatFashion                  2.366
```

Investigating Table.22, it is evident that by far the most important variable is “Tenure”, meaning that compared to the others it is the most utilized variable for reducing uncertainty through the criterion Gini Impurity when making decisions and it provides the most useful information for splitting the data across the trees in the forest. The rest of the variables’ importance are compared to “Tenure” as a proportion of its mean decrease Gini value across the trees, Therefore, observing the second most important variable “WarehouseToHome” its Gini value is merely 19% or a fifth of the “Tenure”’s importance and so it goes for the rest of the variables.

Discussion

Model Performance Comparison

In the final iteration of each model—logistic regression, kNN, and Random Forest—the evaluation based on the confusion matrices reveals distinct performance differences.

Table.10 Logistic Regression Iteration 6 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	772	29
	1	162	163
Accuracy	0.8304		
Sensitivity	0.8266		
Specificity	0.8490		

Logistic Regression: This model demonstrated moderate predictive capabilities, achieving an accuracy of 83.04%. However, its sensitivity and specificity stood at 82.66% and 84.90%, respectively. Despite its interpretability, it fell short in predicting churn compared to the other models. That is mainly due to the fact that the algorithm has stricter assumptions than the other two models. One of such, is an assumption of linearity between the log odds values of the predicted outcome and the predictors. The model works best if there was a clear linear relationship between the two, however this has not been the case with the dataset on hand, Appendix C shows this relationship for the numerical variables of the final model.

Table.21 Random Forest Iteration 6 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	857	8
	1	184	184
Accuracy	0.9245		
Sensitivity	0.9176		
Specificity	0.9583		

Random Forest: In its final iteration, the Random Forest model showcased an accuracy of 92.45%. While its sensitivity and specificity were higher than logistic regression (91.76% and 95.83%, respectively), it still lagged behind the kNN model in predictive accuracy.

Table.15 kNN Iteration 6 Results

Confusion Matrix

		Reference	
		0	1
Prediction	0	921	7
	1	13	185
Accuracy	0.9822		
Sensitivity	0.9861		
Specificity	0.9635		

kNN (K-Nearest Neighbours): Remarkably, the kNN algorithm emerged as the most accurate classifier. With an accuracy of 98.22%, its sensitivity and specificity were superior, reaching 98.61% and 96.35%, respectively. These metrics clearly outperformed both logistic regression and Random Forest, marking it as the best model for customer churn prediction in this study.

There are many measures to visualize the trade-offs between the examined classification algorithms. The main two measures used in these classification problems are Precision-Recall Curves and ROC (Receiver Operating Characteristic) curves (Brownlee, 2023). Since the focus of performance measurements for the designated algorithms were sensitivity (recall) and specificity, it makes sense to display a comparison of ROC curves for the final models obtained. ROC curves have been plotted to visualise and compare the performance of final models of each algorithm in Figure13 below.

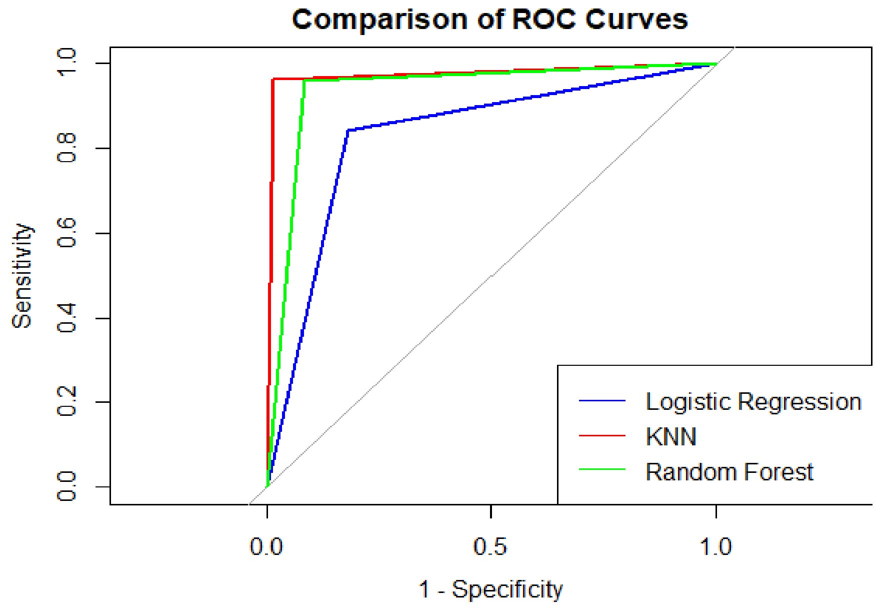


Figure13 ROC Curves of Logistic Regression, kNN and Random Forest

On the other hand, Logistic Regression and Random Forest have served the purpose of quantifying and recognising predictors' importance in the dataset. Random Forest has deemed the variable "Tenure" the most important variable by far with Logistic regression supporting that claim having a near zero p-value and an impactful coefficient.

Impact of Preprocessing Techniques

Throughout the iterations, various preprocessing techniques were explored, aiming to enhance models’ performances. These techniques significantly contributed to improving the results which solidifies the importance of data preprocessing in data science.

Logistic regression did not require one hot encoding categorical variables since the model deal with this issue internally. On the other hand, logarithmic transformation on selected numerical variables has proven more effective than scaling transformation for this dataset.

Nevertheless, one hot encoding categorical variables was a key step in both kNN and Random Forest. Both models are not equipped to deal with characters inputs, therefore, transforming those into binary switches through one hot encoding allow the models to realize the presence or absence of each lever during implementation.

Moreover, kNN benefited from scaling transformation of selected numerical variables since the model calculates Euclidean distances to classify new points, therefore, having similar scales help capture the similarities of new points to existing ones. Finally, Random Forest benefited most from logarithmic transformation of selected numerical variables opposed to scaling, this is mainly due to the impact of log transformation on variance stabilization in the data. Variables with high variance might dominate the splitting criterion in decision trees, impacting the model's performance. Transforming variables with high variance using logarithms can help balance their effect in tree-based models.

Advantages and Disadvantages of the Chosen Approach

kNN stands out for its simplicity and adaptability to complex, nonlinear datasets, making it an appealing choice for classification problems. Its straightforward logic, relying on proximity-based classification, makes it easy to implement and interpret. However, its practicality diminishes with larger datasets due to its computational demands. This computational complexity can lead to slower processing times and increased memory requirements. Additionally, the model's performance can be sensitive to the choice of the "k" parameter and is susceptible to noise and outliers, affecting its predictive accuracy. Addressing these challenges often involves careful parameter tuning and preprocessing steps to strike a balance between computational efficiency and predictive power.

Future Directions

Future research endeavours could focus on expanding the analysis to encompass temporal patterns, user behaviour analysis, and deeper customer segmentation strategies. Additionally, exploring the adaptability of the kNN algorithm in diverse e-commerce settings would provide a more comprehensive understanding of its applicability.

Delimitations

The scope of this study is delimited by the utilization of a specific dataset. The analysis and models constructed for churn prediction are contingent upon the characteristics and behaviors captured within the dataset. While the employed dataset offers a comprehensive representation of customer interactions, the study acknowledges the potential existence of broader contextual elements or evolving patterns in customer behavior beyond the confines of this specific dataset. Thus, the generalizability of the findings to different contexts warrants cautious consideration and might require validation or extension with additional data sources or over different time periods to enhance the robustness and applicability of the predictive models.

Conclusion

In this study, a comprehensive analysis was undertaken to investigate the performance of three distinct machine learning models - Logistic Regression, k-Nearest Neighbours (kNN), and Random Forest - in predicting customer churn within the e-commerce industry. The choice of these algorithms was based on the dataset's unique characteristics and the unique perception and value added of each algorithm.

The main objective was to identify which model possessed the best capabilities in terms of performance, along with identifying the most impactful variables on customer churn, thereby aiding businesses in implementing proactive retention strategies. Moreover, through comprehensive analysis, it was determined that kNN emerged as the most effective predictive model for churn, based on the distinctive data characteristics. Furthermore, among the variables examined, Tenure exhibited the highest impact on churn, followed by warehouseToHome distance and customer Complaint frequency, highlighting critical areas for strategic intervention.

The iterative process of ML implementation involved in-depth preprocessing steps, feature engineering, rigorous model evaluations and hyperparameter sensitivity analysis to optimize predictive performance. Throughout the iterations, it became evident that while all three models showcased competence, each exhibited distinctive strengths and weaknesses.

The Logistic Regression model demonstrated moderate predictive capabilities, achieving an accuracy of 83.04%, with reasonable sensitivity and specificity rates. However, it fell short in accurately identifying potential churners compared to the kNN and Random Forest models in their final iterations. Yet, it provided insight on variable importance, strengthening the findings of Random Forest in identifying the most impactful variables in predicting churn.

Conversely, the kNN model proved to be exceptionally adept at classifying both classes, achieving an accuracy of 98.22%. It exhibited superior sensitivity and specificity, surpassing the other models in predicting both customer retention and churn. Therefore, it was deemed the most accurate algorithm in predicting customer's churn in an e-commerce platform, satisfying the main objective of this study.

The Random Forest model, while exhibiting good predictive performance with an accuracy of 92.45%, showcased a slightly lower sensitivity rate than the kNN model. However, it excelled in achieving a high specificity rate, signifying its efficacy in correctly identifying customers who would churn making it a close competitor to kNN.

Moreover, the analysis highlighted the significance of feature engineering and model hyperparameter tuning in augmenting predictive performance. The findings underscore the importance of continuous refinement and optimization in machine learning approaches to effectively tackle complex business challenges like customer churn prediction.

Nevertheless, this study contributes significantly to the data science field by addressing a notable research gap in customer churn prediction within the e-commerce domain. Through an extensive examination of various machine learning models and feature importance analyses, valuable insights have been uncovered. Firstly, the research aims to identify the machine learning model with the best predictive capabilities in terms of performance for customer churn prediction within the e-commerce domain. By benchmarking and comparing various machine learning models based on dataset characteristics, the study endeavours to uncover the most accurate model for predicting customer churn. This objective directly addresses the need for accurate and efficient tools to empower e-commerce businesses in predicting and mitigating customer churn.

Secondly, the research aims to identify the most impactful variables on customer churn. By exploring the characteristics of the dataset and conducting in-depth analyses, the study seeks to uncover the key factors influencing customer retention or churn. This aspect of the research holds practical significance for decision-makers within e-commerce businesses, as it provides actionable insights that can inform proactive retention strategies and action plans.

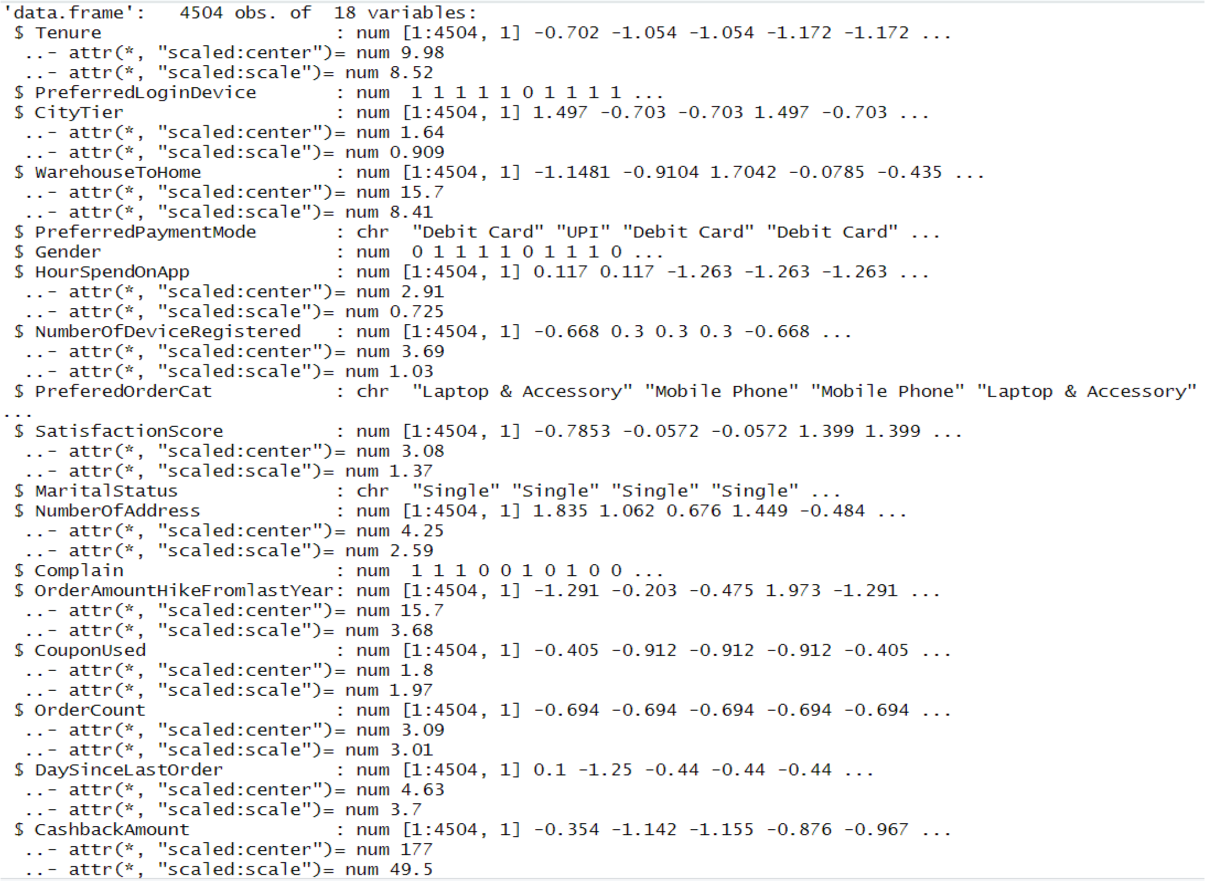
In summary, the research objectives outlined align closely with the overarching purpose of the thesis, which is to contribute to the field by identifying the machine learning model with the best predictive capabilities along with understanding of customer churn dynamics in e-commerce and provide valuable insights and tools for businesses to enhance customer retention and overall business performance.

Appendix A

```
                     Logistic regression summary for final iteration
Coefficients:                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                                 -12.09954  622.97489  -0.019 0.984504
Tenure                                       -1.18696    0.07376 -16.093  < 2e-16 ***
PreferredLoginDevice1                        -0.52556    0.16893  -3.111 0.001864 **
CityTier2                                     0.33422    0.39248   0.852 0.394452
CityTier3                                     0.79064    0.20414   3.873 0.000107 ***
WarehouseToHome                               0.69585    0.15759   4.416 1.01e-05 ***
PreferredPaymentModeCredit Card              -1.31039    0.30094  -4.354 1.33e-05 ***
PreferredPaymentModeDebit Card               -0.97508    0.29321  -3.325 0.000883 ***
PreferredPaymentModeE wallet                 -0.69724    0.38171  -1.827 0.067758 .
PreferredPaymentModeUPI                      -0.94758    0.41224  -2.299 0.021526 *
Gender1                                       0.04843    0.16201   0.299 0.765017
HourSpendOnApp1                              11.88362  622.97490   0.019 0.984781
HourSpendOnApp2                              12.45509  622.97360   0.020 0.984049
HourSpendOnApp3                              12.55921  622.97360   0.020 0.983916
HourSpendOnApp4                              12.28551  622.97363   0.020 0.984266
HourSpendOnApp5                              -2.53706 1080.43140  -0.002 0.998126
NumberOfDeviceRegistered2                    -0.08286    0.57410  -0.144 0.885244
NumberOfDeviceRegistered3                     0.84973    0.44549   1.907 0.056465 .
NumberOfDeviceRegistered4                     0.96555    0.44829   2.154 0.031251 *
NumberOfDeviceRegistered5                     1.39026    0.47405   2.933 0.003360 **
NumberOfDeviceRegistered6                     1.96381    0.60674   3.237 0.001209 **
PreferedOrderCatGrocery                      -1.15308    0.41665  -2.768 0.005648 **
PreferedOrderCatLaptop & Accessory           -1.81693    0.26196  -6.936 4.03e-12 ***
PreferedOrderCatMobile Phone                 -0.64204    0.25371  -2.531 0.011387 *
PreferedOrderCatOthers                        0.44897    0.43315   1.037 0.299951
SatisfactionScore2                           -0.13434    0.32765  -0.410 0.681800
SatisfactionScore3                            0.52744    0.23590   2.236 0.025360 *
SatisfactionScore4                            0.58347    0.26577   2.195 0.028133 *
SatisfactionScore5                            0.93030    0.25750   3.613 0.000303 ***
MaritalStatusMarried                         -0.50548    0.24373  -2.074 0.038084 *
MaritalStatusSingle                           0.61649    0.24774   2.488 0.012832 *
NumberOfAddress2                              0.84280    0.36991   2.278 0.022704 *
NumberOfAddress3                              1.05748    0.38914   2.717 0.006578 **
NumberOfAddress4                              1.27605    0.44146   2.891 0.003846 **
NumberOfAddress5                              1.85342    0.44118   4.201 2.66e-05 ***
NumberOfAddress6                              2.24395    0.46817   4.793 1.64e-06 ***
NumberOfAddress7                              2.14719    0.49061   4.377 1.21e-05 ***
NumberOfAddress8                              2.41479    0.46618   5.180 2.22e-07 ***
NumberOfAddress9                              2.98349    0.51878   5.751 8.87e-09 ***
NumberOfAddress10                             2.34726    0.50585   4.640 3.48e-06 ***
NumberOfAddress11                             2.06940    0.60457   3.423 0.000619 ***
NumberOfAddress19                            14.68308  882.74354   0.017 0.986729
NumberOfAddress20                            12.65618  882.74355   0.014 0.988561
Complain1                                     1.71397    0.16661  10.287  < 2e-16 ***
OrderAmountHikeFromlastYear                  -0.91902    0.34551  -2.660 0.007817 **
CouponUsed                                    0.19372    0.11425   1.696 0.089973 .
```

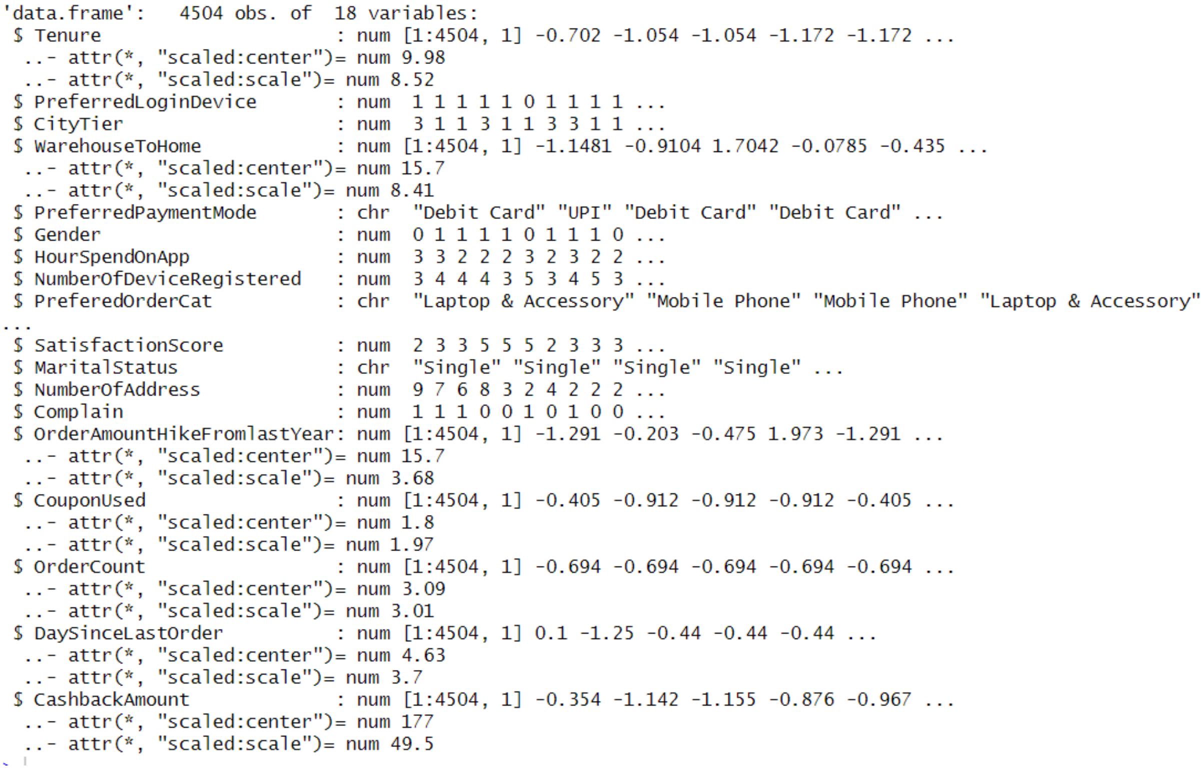
```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2096.1  on 1511  degrees of freedom
Residual deviance: 1143.9  on 1466  degrees of freedom
AIC: 1235.9

Number of Fisher Scoring iterations: 13
```

Appendix B



```
'data.frame':   4504 obs. of  18 variables:
 $ Tenure                     : num [1:4504, 1] -0.702 -1.054 -1.054 -1.172 -1.172 ...
  ..- attr(*, "scaled:center")= num 9.98
  ..- attr(*, "scaled:scale")= num 8.52
 $ PreferredLoginDevice       : num  1 1 1 1 1 0 1 1 1 1 ...
 $ CityTier                   : num [1:4504, 1] 1.497 -0.703 -0.703 1.497 -0.703 ...
  ..- attr(*, "scaled:center")= num 1.64
  ..- attr(*, "scaled:scale")= num 0.909
 $ WarehouseToHome            : num [1:4504, 1] -1.1481 -0.9104 1.7042 -0.0785 -0.435 ...
  ..- attr(*, "scaled:center")= num 15.7
  ..- attr(*, "scaled:scale")= num 8.41
 $ PreferredPaymentMode       : chr  "Debit Card" "UPI" "Debit Card" "Debit Card" ...
 $ Gender                     : num  0 1 1 1 1 0 1 1 1 0 ...
 $ HourSpendOnApp             : num [1:4504, 1] 0.117 0.117 -1.263 -1.263 -1.263 ...
  ..- attr(*, "scaled:center")= num 2.91
  ..- attr(*, "scaled:scale")= num 0.725
 $ NumberOfDeviceRegistered   : num [1:4504, 1] -0.668 0.3 0.3 0.3 -0.668 ...
  ..- attr(*, "scaled:center")= num 3.69
  ..- attr(*, "scaled:scale")= num 1.03
 $ PreferedOrderCat           : chr  "Laptop & Accessory" "Mobile Phone" "Mobile Phone" "Laptop & Accessory"
...
 $ SatisfactionScore          : num [1:4504, 1] -0.7853 -0.0572 -0.0572 1.399 1.399 ...
  ..- attr(*, "scaled:center")= num 3.08
  ..- attr(*, "scaled:scale")= num 1.37
 $ MaritalStatus              : chr  "Single" "Single" "Single" "Single" ...
 $ NumberOfAddress            : num [1:4504, 1] 1.835 1.062 0.676 1.449 -0.484 ...
  ..- attr(*, "scaled:center")= num 4.25
  ..- attr(*, "scaled:scale")= num 2.59
 $ Complain                   : num  1 1 1 0 0 1 0 1 0 0 ...
 $ OrderAmountHikeFromlastYear: num [1:4504, 1] -1.291 -0.203 -0.475 1.973 -1.291 ...
  ..- attr(*, "scaled:center")= num 15.7
  ..- attr(*, "scaled:scale")= num 3.68
 $ CouponUsed                 : num [1:4504, 1] -0.405 -0.912 -0.912 -0.912 -0.405 ...
  ..- attr(*, "scaled:center")= num 1.8
  ..- attr(*, "scaled:scale")= num 1.97
 $ OrderCount                 : num [1:4504, 1] -0.694 -0.694 -0.694 -0.694 -0.694 ...
  ..- attr(*, "scaled:center")= num 3.09
  ..- attr(*, "scaled:scale")= num 3.01
 $ DaySinceLastOrder          : num [1:4504, 1] 0.1 -1.25 -0.44 -0.44 -0.44 ...
  ..- attr(*, "scaled:center")= num 4.63
  ..- attr(*, "scaled:scale")= num 3.7
 $ CashbackAmount             : num [1:4504, 1] -0.354 -1.142 -1.155 -0.876 -0.967 ...
  ..- attr(*, "scaled:center")= num 177
  ..- attr(*, "scaled:scale")= num 49.5
```

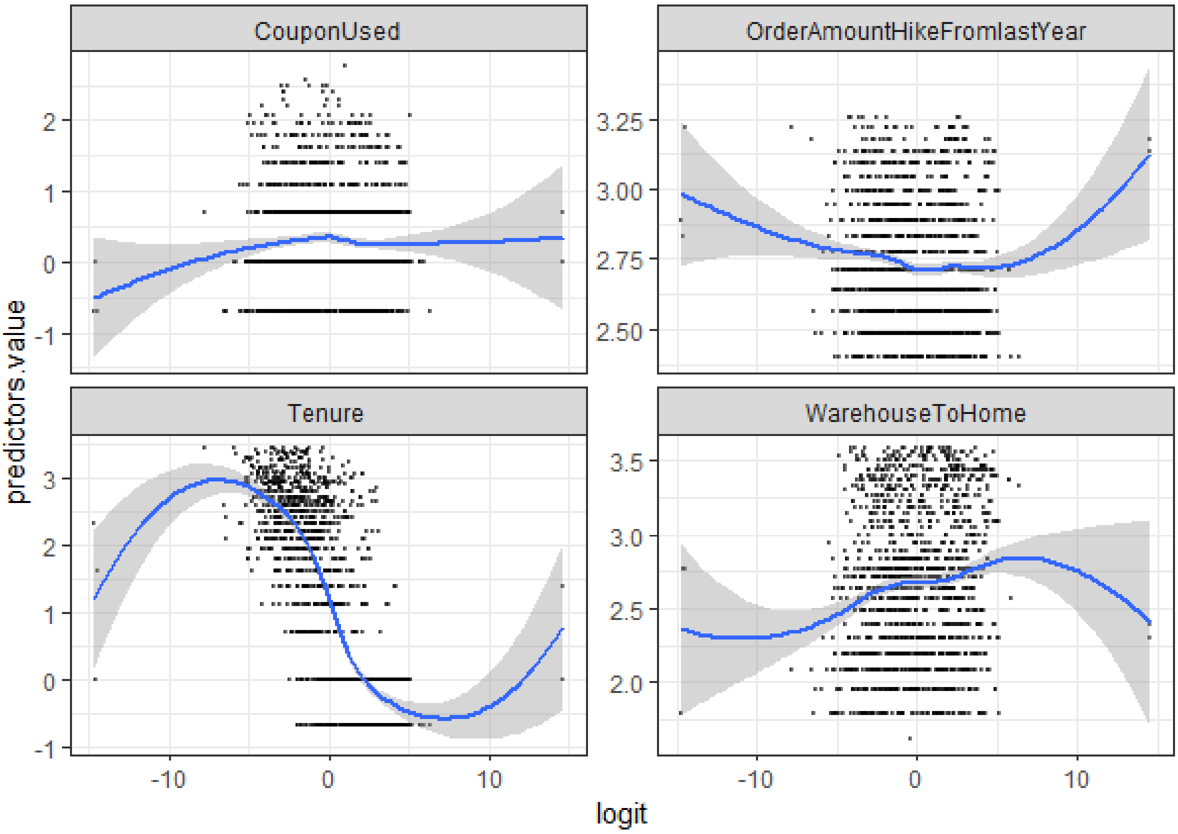
Table.1 (Data structure for iteration 2 KNN)



```
'data.frame':   4504 obs. of  18 variables:
 $ Tenure                     : num [1:4504, 1] -0.702 -1.054 -1.054 -1.172 -1.172 ...
  ..- attr(*, "scaled:center")= num 9.98
  ..- attr(*, "scaled:scale")= num 8.52
 $ PreferredLoginDevice       : num  1 1 1 1 1 0 1 1 1 1 ...
 $ CityTier                   : num  3 1 1 3 1 1 3 3 1 1 ...
 $ WarehouseToHome            : num [1:4504, 1] -1.1481 -0.9104 1.7042 -0.0785 -0.435 ...
  ..- attr(*, "scaled:center")= num 15.7
  ..- attr(*, "scaled:scale")= num 8.41
 $ PreferredPaymentMode       : chr  "Debit Card" "UPI" "Debit Card" "Debit Card" ...
 $ Gender                     : num  0 1 1 1 1 0 1 1 1 0 ...
 $ HourSpendOnApp             : num  3 3 2 2 2 3 2 3 2 2 ...
 $ NumberOfDeviceRegistered   : num  3 4 4 4 3 5 3 4 5 3 ...
 $ PreferedOrderCat           : chr  "Laptop & Accessory" "Mobile Phone" "Mobile Phone" "Laptop & Accessory"
...
 $ SatisfactionScore          : num  2 3 3 5 5 5 2 3 3 3 ...
 $ MaritalStatus              : chr  "Single" "Single" "Single" "Single" ...
 $ NumberOfAddress            : num  9 7 6 8 3 2 4 2 2 2 ...
 $ Complain                   : num  1 1 1 0 0 1 0 1 0 0 ...
 $ OrderAmountHikeFromlastYear: num [1:4504, 1] -1.291 -0.203 -0.475 1.973 -1.291 ...
  ..- attr(*, "scaled:center")= num 15.7
  ..- attr(*, "scaled:scale")= num 3.68
 $ CouponUsed                 : num [1:4504, 1] -0.405 -0.912 -0.912 -0.912 -0.405 ...
  ..- attr(*, "scaled:center")= num 1.8
  ..- attr(*, "scaled:scale")= num 1.97
 $ OrderCount                 : num [1:4504, 1] -0.694 -0.694 -0.694 -0.694 -0.694 ...
  ..- attr(*, "scaled:center")= num 3.09
  ..- attr(*, "scaled:scale")= num 3.01
 $ DaySinceLastOrder          : num [1:4504, 1] 0.1 -1.25 -0.44 -0.44 -0.44 ...
  ..- attr(*, "scaled:center")= num 4.63
  ..- attr(*, "scaled:scale")= num 3.7
 $ CashbackAmount             : num [1:4504, 1] -0.354 -1.142 -1.155 -0.876 -0.967 ...
  ..- attr(*, "scaled:center")= num 177
  ..- attr(*, "scaled:scale")= num 49.5
```

Table.2 (Data structure for iteration 3 KNN)

Appendix C



CouponUsed
OrderAmountHikeFromlastYear
Tenure
WarehouseToHome
predictors.value
logit
2
1
0
-1
3.25
3.00
2.75
2.50
3
2
1
0
-1
3.5
3.0
2.5
2.0
-10
0
10
-10
0
10

References

Abba, I. V. (2023, January 25). *KNN algorithm – K-nearest Neighbours classifiers and model example*. freeCodeCamp.org. https://www.freecodecamp.org/news/k-nearest-Neighbours-algorithm-classifiers-and-model-example/

Allison, P. (2022, July 18). *Imputation by predictive mean matching: Promise & peril*. Statistical Horizons. https://statisticalhorizons.com/predictive-mean-matching/

Brownlee, J. (2023, October 10). *How to use ROC curves and precision-recall curves for classification in Python*. MachineLearningMastery.com. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/#:~:text=Two%20diagnostic%20tools%20that%20help,Curves%20and%20Precision%2DRecall%20curves.

Dealing with missing values. Dealing with Missing Values · UC Business Analytics R Programming Guide. (2016). https://uc-r.github.io/missing_values

Ecommerce churn rate: How to calculate and Reduce Churn. Shopify. (2022, October 21). Retrieved April 26, 2023, from https://www.shopify.com/blog/churn-rate-in-ecommerce#:~:text=However%2C%20to%20give%20a%20reference,cohort%20would%20be%20considered%20average.

R, S. E: (2023, July 5). *Understand random forest algorithms with examples (updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Harrison, O. (2019, July 14). *Machine learning basics with the K-nearest Neighbours algorithm*. Medium. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-Neighbours-algorithm-6a6e71d01761

Kanade, V. (2022, April 18). *Logistic regression: Equation, assumptions, types, and best practices*. Spiceworks. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

Kundu, R. (2022, December 16). *F1 score in Machine Learning: Intro & Calculation*. V7. https://www.v7labs.com/blog/f1-score-guide

Reichheld, F. (2001, September). *Prescription for cutting costs - bain & company*. Bain. Retrieved April 26, 2023, from https://media.bain.com/Images/BB_Prescription_cutting_costs.pdf

Saini, A. (2021, August 26). *Conceptual understanding of logistic regression for data science beginners*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

Sequitin, K. (2021, October 5). *Data Analytics explained: What is an outlier?*. CareerFoundry. https://careerfoundry.com/en/blog/data-analytics/what-is-an-outlier/

Stef van Buuren, Karin Groothuis-Oudshoorn (2011). mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67. DOI 10.18637/jss.v045.i03.

Tamboli, N. (2023, July 14). *Effective strategies for handling missing values in data analysis (updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/

Turner, D. S. (2020, February 11). *Explore and clean: First steps of any data project.* Medium. https://towardsdatascience.com/explore-and-clean-first-steps-of-any-data-project-976a1d80d7aa

What is customer churn in Ecommerce for your shopfiy store? ReturnLogic. (2023, April 10). Retrieved April 26, 2023, from https://www.returnlogic.com/blog/what-is-customer-churn-in-ecommerce/