Capstone project2

Ahmed Al-Jifri

2023-12-06

Overview

For the purpose of this project, 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 easy to import into R and start preparing. The dataset has a label (Churn), 18 predictors and 5630 unique observations. To get an idea about the dataset, below is basic summary and structure of the dataset:

```
summary(commerce_data)
```

```
##    CustomerID       Churn            Tenure      PreferredLoginDevice
##  Min.   :   1   Min.   :0.0000   Min.   : 0.00   Length:5630
##  1st Qu.:1408   1st Qu.:0.0000   1st Qu.: 2.00   Class :character
##  Median :2816   Median :0.0000   Median : 9.00   Mode  :character
##  Mean   :2816   Mean   :0.1684   Mean   :10.19
##  3rd Qu.:4223   3rd Qu.:0.0000   3rd Qu.:16.00
##  Max.   :5630   Max.   :1.0000   Max.   :61.00
##                                  NA's   :264
##     CityTier     WarehouseToHome  PreferredPaymentMode    Gender
##  Min.   :1.000   Min.   :  5.00   Length:5630          Length:5630
##  1st Qu.:1.000   1st Qu.:  9.00   Class :character     Class :character
##  Median :1.000   Median : 14.00   Mode  :character     Mode  :character
##  Mean   :1.655   Mean   : 15.64
##  3rd Qu.:3.000   3rd Qu.: 20.00
##  Max.   :3.000   Max.   :127.00
##                  NA's   :251
##  HourSpendOnApp  NumberOfDeviceRegistered PreferedOrderCat   SatisfactionScore
##  Min.   :0.000   Min.   :1.000            Length:5630        Min.   :1.000
##  1st Qu.:2.000   1st Qu.:3.000            Class :character   1st Qu.:2.000
##  Median :3.000   Median :4.000            Mode  :character   Median :3.000
##  Mean   :2.932   Mean   :3.689                               Mean   :3.067
##  3rd Qu.:3.000   3rd Qu.:4.000                               3rd Qu.:4.000
##  Max.   :5.000   Max.   :6.000                               Max.   :5.000
##  NA's   :255
##  MaritalStatus      NumberOfAddress     Complain
##  Length:5630        Min.   : 1.000   Min.   :0.0000
##  Class :character   1st Qu.: 2.000   1st Qu.:0.0000
##  Mode  :character   Median : 3.000   Median :0.0000
##                     Mean   : 4.214   Mean   :0.2849
##                     3rd Qu.: 6.000   3rd Qu.:1.0000
##                     Max.   :22.000   Max.   :1.0000
```

```
##
##  OrderAmountHikeFromlastYear   CouponUsed       OrderCount
##  Min.   :11.00               Min.   : 0.000   Min.   : 1.000
##  1st Qu.:13.00               1st Qu.: 1.000   1st Qu.: 1.000
##  Median :15.00               Median : 1.000   Median : 2.000
##  Mean   :15.71               Mean   : 1.751   Mean   : 3.008
##  3rd Qu.:18.00               3rd Qu.: 2.000   3rd Qu.: 3.000
##  Max.   :26.00               Max.   :16.000   Max.   :16.000
##  NA's   :265                 NA's   :256      NA's   :258
##  DaySinceLastOrder CashbackAmount
##  Min.   : 0.000    Min.   :  0.0
##  1st Qu.: 2.000    1st Qu.:145.8
##  Median : 3.000    Median :163.3
##  Mean   : 4.543    Mean   :177.2
##  3rd Qu.: 7.000    3rd Qu.:196.4
##  Max.   :46.000    Max.   :325.0
##  NA's   :307
```

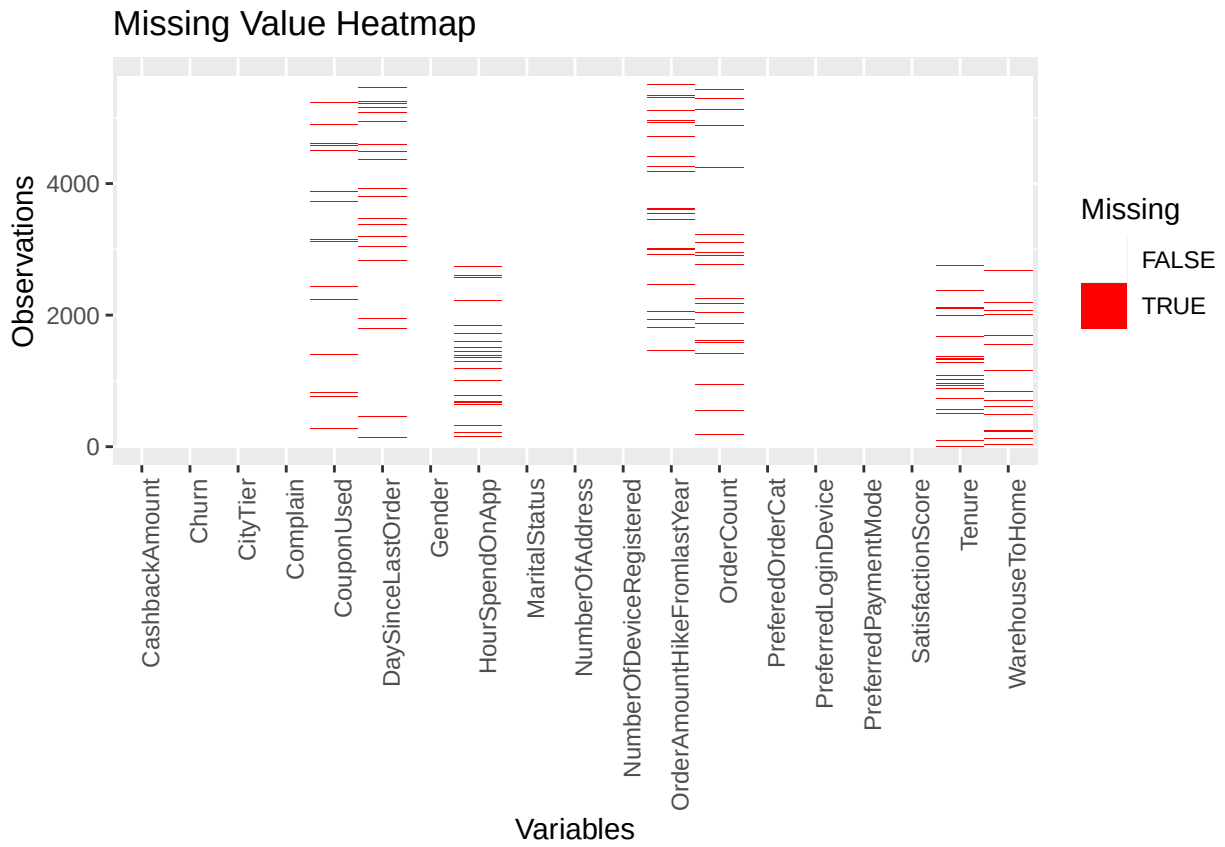
```
str(commerce_data)
```

```
## tibble [5,630 x 20] (S3: tbl_df/tbl/data.frame)
##  $ CustomerID                : int [1:5630] 1 2 3 4 5 6 7 8 9 10 ...
##  $ 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 & Accesso
##  $ 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 ...
```

Nevertheless, the purpose of this project is to build machine learning models that can accurately predict customers' churn in an e-commerce platform. the key steps performed were data preprocessing, data visualization, implementation and analysis of the machine learning algorithms used.

Methodology

First step lets preprocess the data and clean it for proper machine learnignimplementation. lets start by looking at missing values, it is evident from the summaries above that we do have some missing values. A visual representation can give us a good idea as shown below:

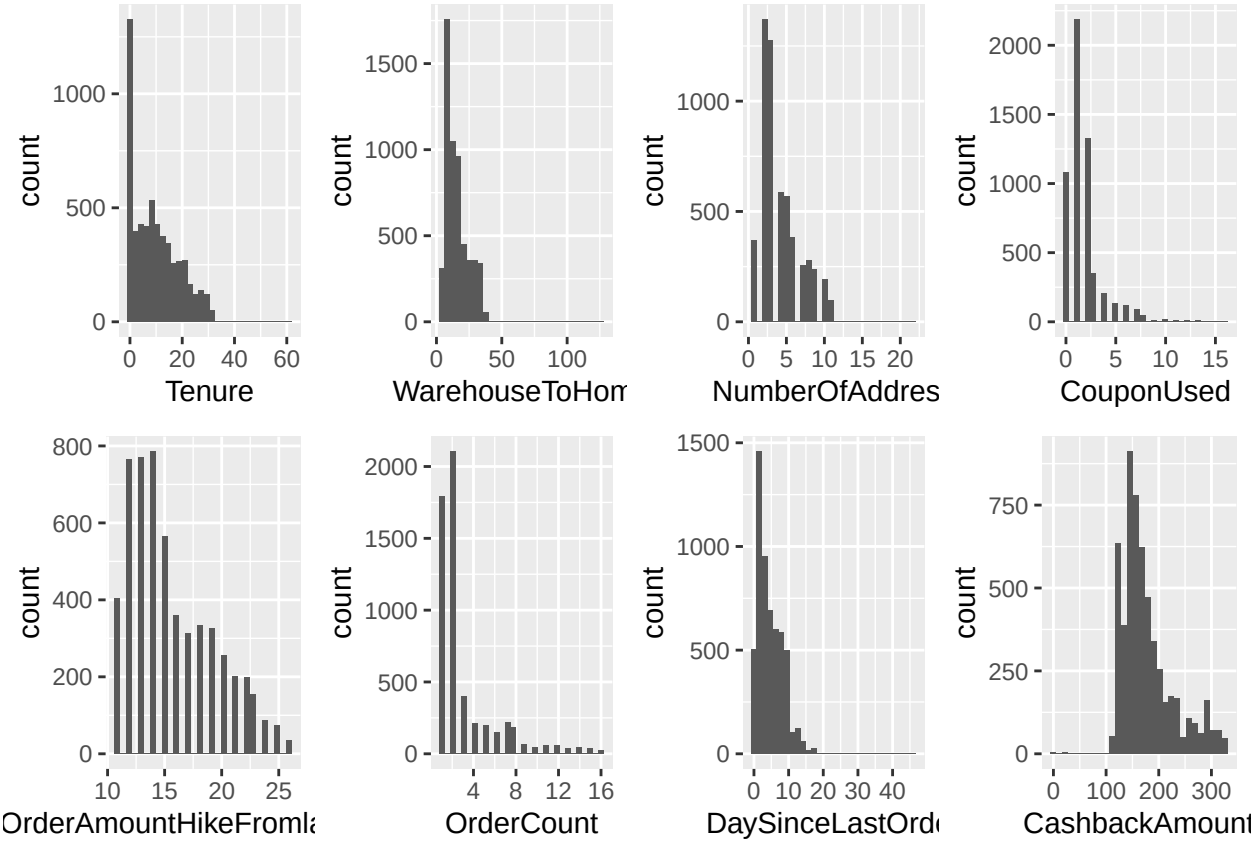


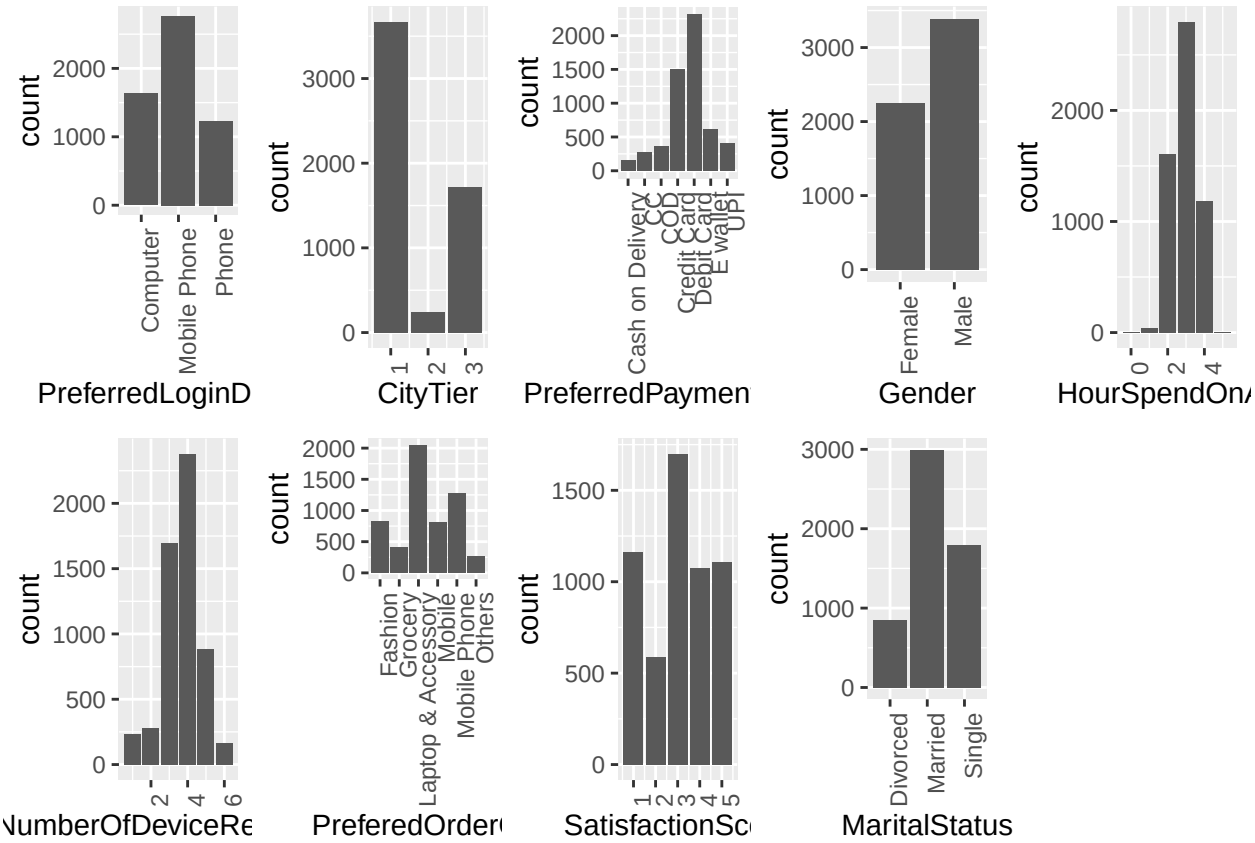
We can see there are 7 variables with missing data in them.There are many ways to handle missing data, in the report we opt for using a function called mice , which provides a solution for handling Nas by generating multiple imputations for multivariate missing values. This approach utilizes the Fully 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. After using the function mice() we can check for na using the following code:

```
sum(is.na(commerce_data_complt))
```

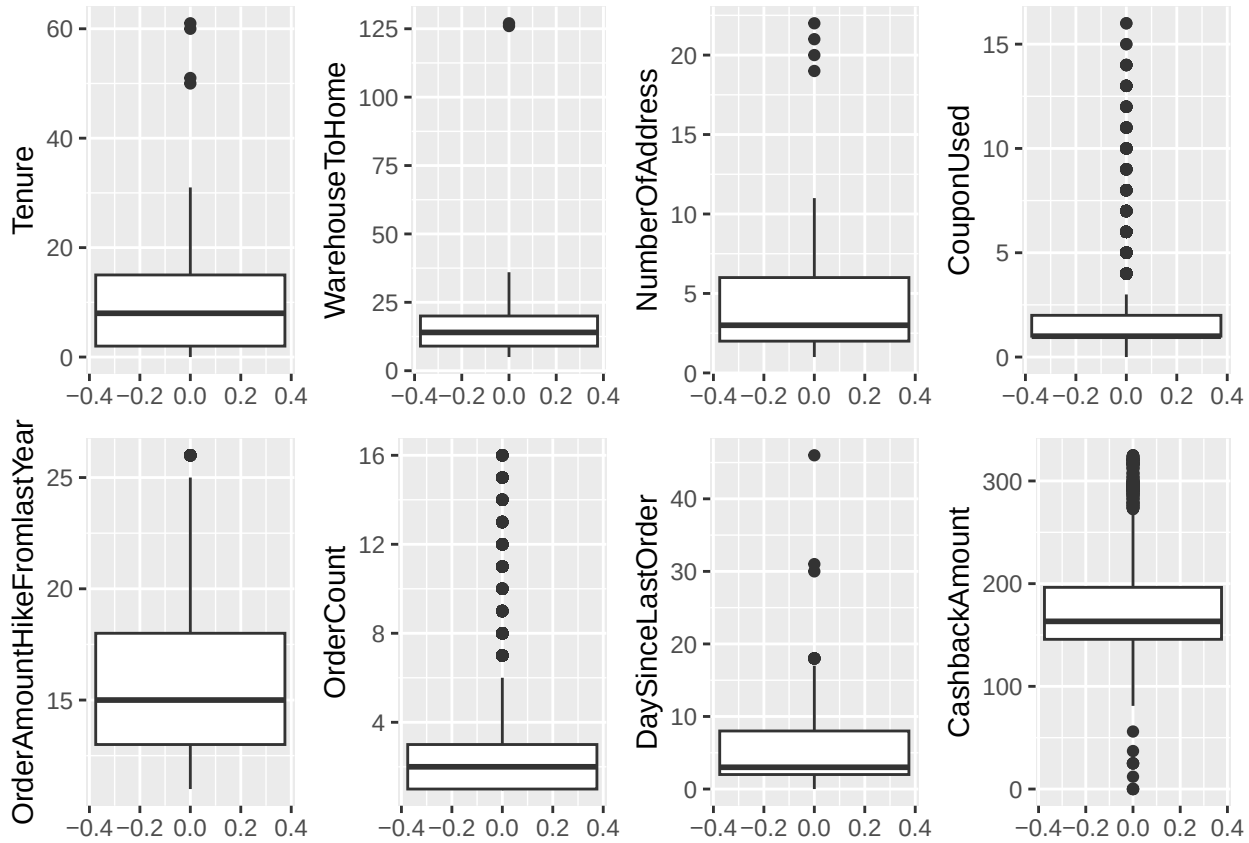
```
## [1] 0
```

Now that we've handled the missing variables, we can look into the dataset. Predictors have been manually categorized as a preliminary step into 3 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 shown below:


count
Tenure
WarehouseToHom
NumberOfAddres
CouponUsed
OrderAmountHikeFromla
OrderCount
DaySinceLastOrd
CashbackAmount



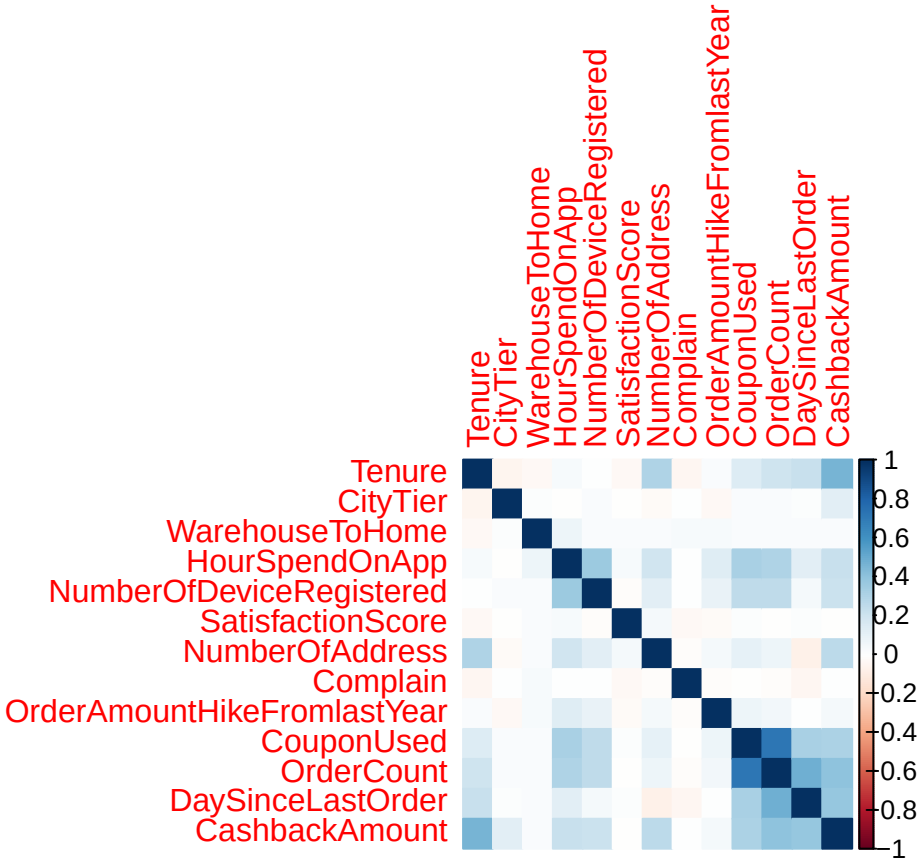
Looking briefly at Figure.1 and Figure.2 it is evident that there are outliers, however we're gonna leave them be for now and test their impact later on in ml implementation. On the other hand, categorical variables have some inconsistencies in their levels. We can see in their corresponding barplots that some variables have redundant levels which will be rectified. Moreover, boxplots are generated for numerical variables to get an idea about outliers and their corresponding statistics as shown in below:



Next, we need to split the dataset into training and testing parts with training set accounting for 80% of the original dataset and the rest 20% for the testing. to do this we use the function createDataPartition() as shown below:

```
set.seed(1991)
test <- createDataPartition(commerce_data_complt$Churn, times = 1, p = .2, list = FALSE)
train_commerce <- commerce_data_complt[-test,]
test_commerce <- commerce_data_complt[test,]
```

Furthermore, we can do correlation analysis between the numerical variables to check for highly correlated



variables.
We can see relatively high correlation between some of the variables. This shall be used later in tuning in the model.

Now we are ready for ml implementation. The choice of ml methods to use is set to be logistic regression as a base model and K-Nearest Neighbors.

Results

First of all, lets try logistic regression,after some experimentation, the best results were achieved when removing the variables OrderCount, DaysSinceLastOrder and CashbackAmount given high correlation between them and other variables and transform numerical variables with logarithmic scale adding a 0.5 buffer to avoid 0 entries. After that, we need to transform all categorical and binary variables to factors and handle redundancies of some of their levels.

the results of this model can be shown below using a confusion matrix:

```
##fit model
fit_log <- predict(model_log,newdata = test_commerce_logreg, type = "response")

y_h <- ifelse(fit_log > 0.5, 1, 0)
confusionMatrix(as.factor(y_h), as.factor(test_commerce_logreg$Churn2))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
```

```
##          0 900  73
##          1  34 119
##
##                Accuracy : 0.905
##                  95% CI : (0.8863, 0.9215)
##     No Information Rate : 0.8295
##     P-Value [Acc > NIR] : 3.437e-13
##
##                   Kappa : 0.6346
##
##  Mcnemar's Test P-Value : 0.0002392
##
##             Sensitivity : 0.9636
##             Specificity : 0.6198
##          Pos Pred Value : 0.9250
##          Neg Pred Value : 0.7778
##              Prevalence : 0.8295
##          Detection Rate : 0.7993
##    Detection Prevalence : 0.8641
##       Balanced Accuracy : 0.7917
##
##        'Positive' Class : 0
##
```

The model resulted in 0.9636 sensitivity and 0.6198 specificity, however, since we are interested in churns which is measured by the specificity it's quite low.

secondly, lets try KNN, for knn the best results were when scaling numerical variables, leaving categorical ordinal variables as is, however on-hot-encode categorical variables.

Below we can check the accuracies of our knn model through a confusion matrix:

```
fit_knn <- predict(model_knn, newdata = test_commerce_sc)
confusionMatrix(fit_knn, as.factor(test_commerce_sc$Churn))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 921   7
##          1  13 185
##
##                Accuracy : 0.9822
##                  95% CI : (0.9727, 0.9891)
##     No Information Rate : 0.8295
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.938
##
##  Mcnemar's Test P-Value : 0.2636
##
##             Sensitivity : 0.9861
##             Specificity : 0.9635
##          Pos Pred Value : 0.9925
```

```
##          Neg Pred Value : 0.9343
##              Prevalence : 0.8295
##          Detection Rate : 0.8179
##    Detection Prevalence : 0.8242
##       Balanced Accuracy : 0.9748
##
##        'Positive' Class : 0
##
```

We can see that knn outperformed logreg by far, having sensitivity of 0.9861 and specificity of 0.9635 which is very good!

Conclusion

In this project we have worked with a dataset provided publicly from Kaggle website, the dataset holds data with mixed datatypes about customers in an online e commerce platform labelled with churn. The purpose of this project was to create machine learning algorithms to classify our customers based on churning or not. Two models have been chosen for this purpose, logistic regression and knn. Logistic regression poorly classified churns due to the fact that the assumption of linearity between variables and the log odds. On the other hand, knn performed very well, giving highly accurate results. An interesting look would be trying out neural network in this context, however, it is out of the scope of this course.