Project Report

Problem Statement:

File ds1.csv contain data of the customers who were offered a card along with their response in "card_offer." If they bought it, the value would be TRUE else it will be FALSE. Use this data to build a model to predict the customer response.

I. Dataset Information:

There are Two datasets test1.csv
(Training dataset) and test2.csv
(Testing dataset)

Number of Instances: 10000
Number of Attributes: 12
Missing Values: No

Knowing the Dataset:

1. We started our dataset with finding the number of columns and number of rows.

```
ncol(customer)
12
nrow(customer)
10000
```

2. Now we structured the dataset and find the type of the variables.

```
$ customer_id      : int  167317 393970 435082 952844 22454 68256 878177 361131 65721
511114 ...
$ demographic_slice: Factor w/ 4 levels "AX03efs","BWEsk45",..: 3 3 2 1 3 4 1 1 1 3 ...
$ country_reg      : Factor w/ 2 levels "E","W": 1 1 2 1 1 1 2 2 2 2 ...
$ ad_exp           : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 1 1 2 2 ...
$ est_income       : num  76868 89873 45309 26767 96224 ...
$ hold_bal         : num  19.63 21.6 8.02 20.71 18.76 ...
$ pref_cust_prob   : num  0.0958 0.7518 0.411 0.3292 0.5083 ...
$ imp_cscore       : int  556 602 604 539 630 760 686 698 531 673 ...
$ RiskScore        : num  739 634 525 843 567 ...
$ imp_crediteval   : num  24 25.3 25.6 23.5 25.5 ...
$ axio_score       : num  0.0897 0.54 0.5229 0.0386 0.5642 ...
$ card_offer       : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
```

3. We also concluded the X-Variables and Y-Variable from the dataset.

II. Pre-processing of Data:

1. Check for NA, and Blanks values:

Null, NA, Blanks or ? are not present in dataset.

a) Dropping Columns:

i) Customer_id

It is unique identifier of an customer_id, it will not be required in any of the analysis.

A. Graphical Representation:
a) Variable Distributions

1. Axio_Score Distribution

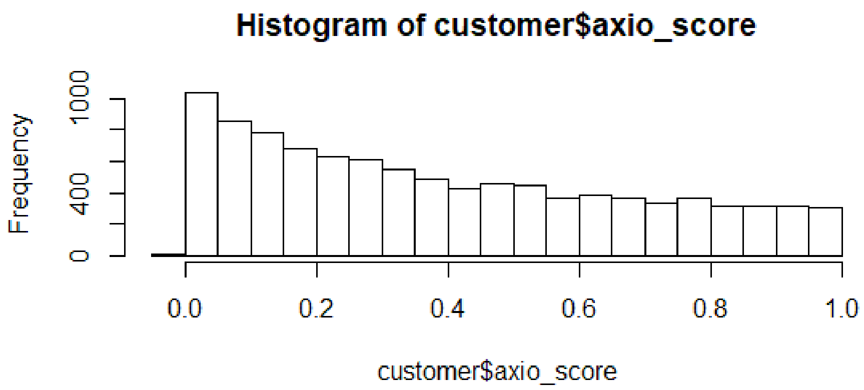
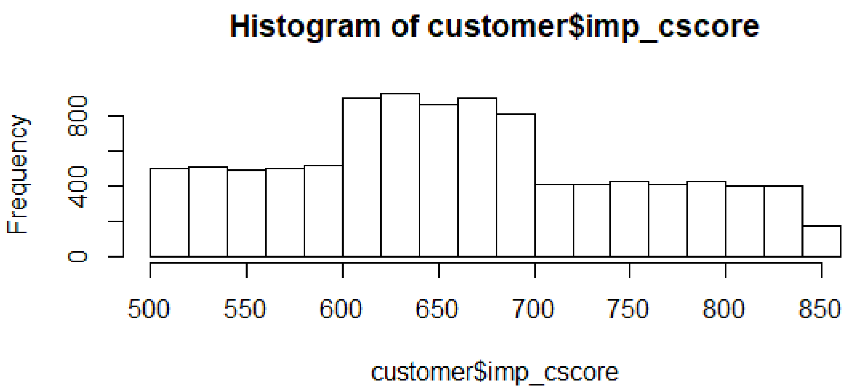


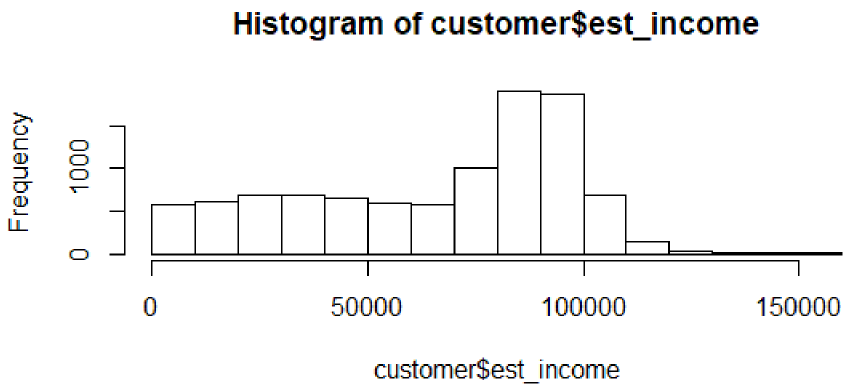
Fig 1: Axio_Score Distribution

We can see the normal distribution is left skewed.

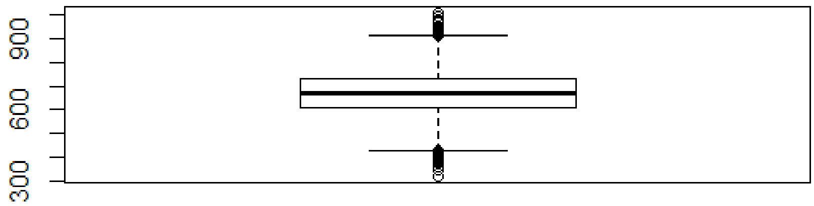
2. Imp_Score Distribution



3. Est_income



4. Boxplot for Risk_score



We can see outliers are present in Risk_score column.

III. Building Training and Testing Model

We randomized (to avoid any selection bias) and divided the clean data obtained into two parts: Training and Test Data, in a 70:30 ratio, which allowed us to train our models on 70% of the data and use the other 30% to assess the performance of our models.

IV. Selecting models

1. Decision Trees: By iteratively and hierarchically observing the level of certainty of predicting whether someone would be readmitted or not, we find the relative importance of different factors using a more human-like decision making strategy in establishing this determination.

2. Random Forests: By considering more than one decision tree and then doing a majority voting, random forests helped in being more robust predictive representations than trees as in the previous case. For both Decision Trees and Random Forests, we removed the interaction terms from the feature set since these are already accounted for in tree-based models.

3. Logistic Regression : Logistic Regression is used for Binary classification. Logistic Regression is extension of linear regression. Models the probability of an event occurring (Y) based on the Independent variables (x1, x2,… xn) **that are numeric or categorical** in nature.

4. Support Vector Machines: Support Vector Machines can help model linearly inseparable data, thus allowing us to explain complex non-linear relationships. However, because of high-dimensional structure and complexity, they are limited by their interpretability to gain insights on how different features are weighted/assigned importance.

5. K-nearest Neighbors: While K-nearest neighbors provide decent predictions, they cannot help in deciding the features that contribute to this decision the most, since features are weighted equally (assuming we normalize them) based on simply their contribution to the proximity/distance function.

Model Selection:

We can use different algorithms to predict model from that dataset(test1.csv). I used Logistic Regression and Random Forest for building model. After comparing these models, I have done final prediction(on test2.csv) using Random Forest because Random Forest gives better accuracy/prediction in training and testing dataset of test1.csv dataset.

1. Using Logistic Regression Algorithm After Building Model

1st Model :

```
Call:
glm(formula = card_offer ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.7504  -0.0515  -0.0047  -0.0001   3.3239

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -2.792e+01  2.044e+00 -13.665  < 2e-16 ***
demographic_sliceBWEsk45  1.152e+00  3.338e-01   3.452 0.000557 ***
demographic_sliceCARDIF2 -9.056e-02  2.637e-01  -0.343 0.731281
demographic_sliceDERS3w5  1.070e+00  3.330e-01   3.212 0.001317 **
country_regW             -7.159e+00  3.962e-01 -18.069  < 2e-16 ***
ad_expY                  -6.488e-02  1.468e-01  -0.442 0.658516
est_income                1.518e-04  7.525e-06  20.171  < 2e-16 ***
hold_bal                 -1.267e-01  9.690e-03 -13.070  < 2e-16 ***
pref_cust_prob            2.524e+01  1.028e+00  24.545  < 2e-16 ***
imp_cscore                9.940e-03  2.288e-03   4.345 1.39e-05 ***
RiskScore                -7.985e-04  8.078e-04  -0.988 0.322950
imp_crediteval            4.293e-02  1.037e-01   0.414 0.678864
axio_score                4.696e-02  2.527e-01   0.186 0.852570
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6112.1  on 6999  degrees of freedom
Residual deviance: 1232.1  on 6987  degrees of freedom
AIC: 1258.1

Number of Fisher Scoring iterations: 9
```

Confusion matrix for Training Dataset:

```
        actual
predicted    0    1
        0 5761  132
        1  132  975
```

Confusion matrix for Testing Dataset:

```
         actual
predicted    0    1
        0 2526   51
        1   50  373
```

"Best accuracy = 0.845"

2nd Model :

```
Call:
glm(formula = card_offer ~ ., family = "binomial", data = train1)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-4.3759  -0.0448  -0.0041  -0.0001   3.1373  

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)    
(Intercept)              -2.846e+01  1.415e+00 -20.107  < 2e-16 ***
demographic_sliceBWEsk45  7.720e-01  3.423e-01   2.256   0.0241 *  
demographic_sliceCARDIF2 -4.185e-02  2.648e-01  -0.158   0.8744    
demographic_sliceDERS3w5  8.286e-01  3.328e-01   2.490   0.0128 *  
country_regW             -6.666e+00  3.863e-01 -17.254  < 2e-16 ***
est_income                1.545e-04  7.651e-06  20.194  < 2e-16 ***
hold_bal                 -1.064e-01  9.430e-03 -11.287  < 2e-16 ***
pref_cust_prob            2.623e+01  1.085e+00  24.166  < 2e-16 ***
imp_cscore                9.866e-03  1.245e-03   7.924  2.3e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 6092.0  on 6999  degrees of freedom
Residual deviance: 1174.6  on 6991  degrees of freedom
AIC: 1192.6

Number of Fisher Scoring iterations: 9
```

Confusion matrix for Training Dataset:

```
         actual
predicted    0    1
        0 5775  122
        1  124  979
```

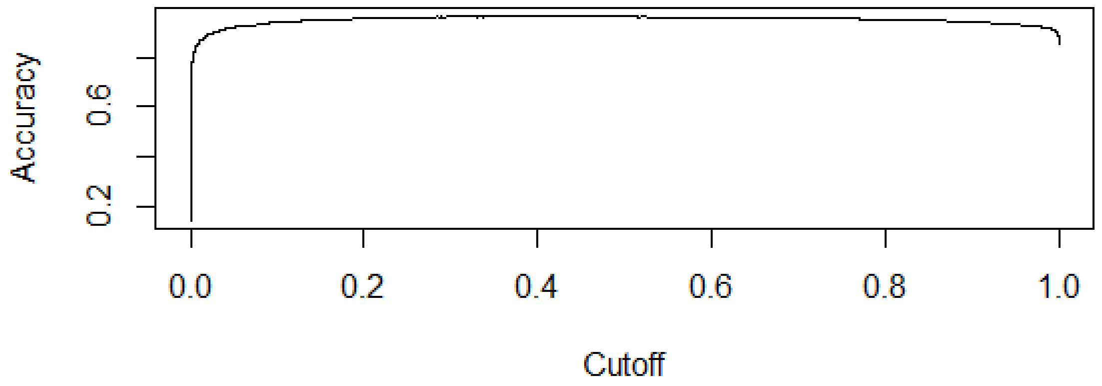
Confusion matrix for Testing Dataset:

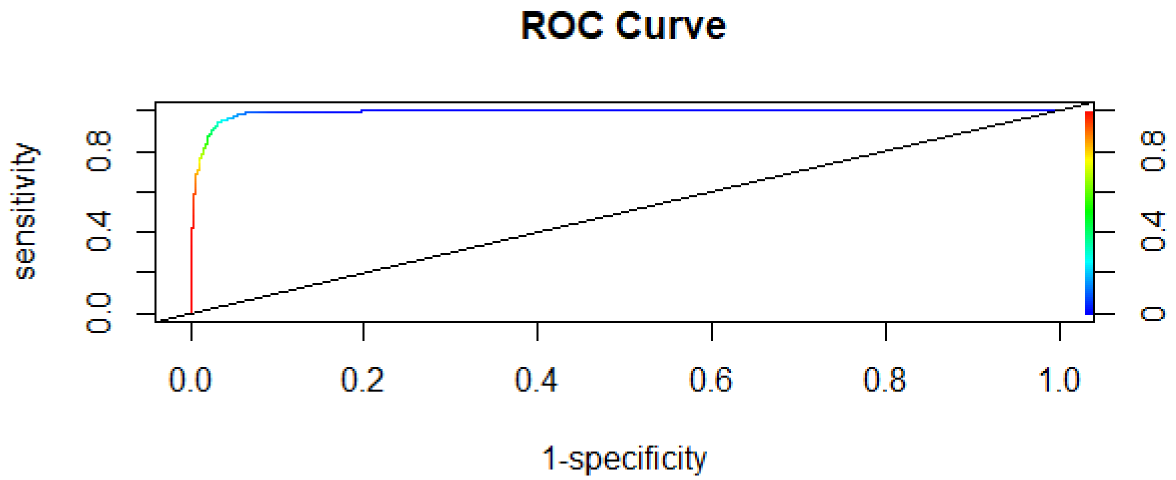
```
         actual
predicted    0    1
        0 2512   46
        1   58  384
```

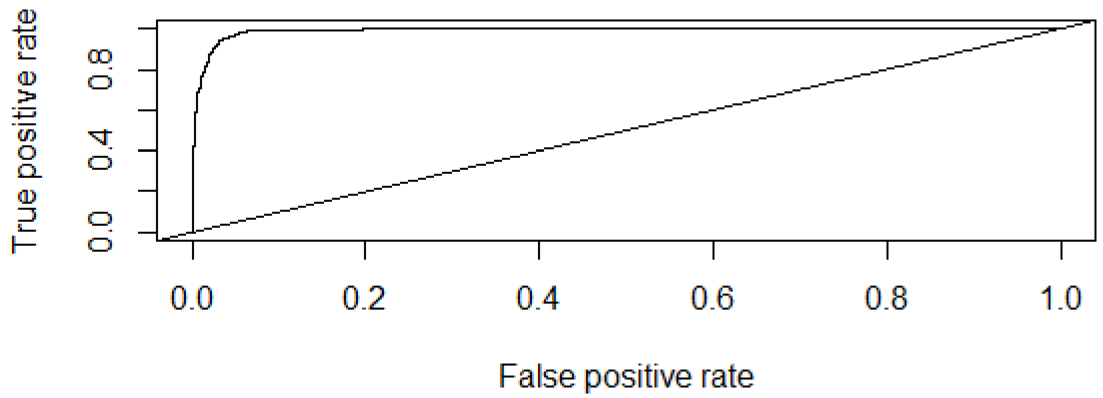
"Best accuracy = 0.966 Best cutoff = 0.5399"

Some Important Graphs After Model Building (test1.csv dataset)

ROC Curve:


Accuracy
0.2
0.6
0.0
0.2
0.4
0.6
0.8
1.0
Cutoff


ROC Curve
sensitivity
0.0
0.4
0.8
0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0
1-specificity


True positive rate
0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

2.Using Random Forest Algorithm After Building Model

1st Model:

```
Call:
 randomForest(x = train_x, y = factor(train_y))
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 2.51%
```

Confusion matrix for Training Dataset:

```
     0   1 class.error
0 5887  59 0.009922637
1  117 937 0.111005693
```

Confusion matrix for Testing Dataset :

```
         actual
predicted    0    1
        0 2556   39
        1   20  385
```

2nd model :

```
Call:
 randomForest(x = train1_x, y = factor(train1_y))
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 2.16%
```

Confusion matrix for Training Dataset :

```
     0   1 class.error
0 5860  47 0.007956662
1  104 989 0.095150961
```

Confusion matrix for Testing Dataset :

```
         actual
predicted    0    1
        0 2538   40
        1   24  398
```

We can see here, second model of random forest algorithm gives better accuracy so I used that model for predicting "card_offer" of customers in actual testing dataset (i.e. test2.csv).