Illinois Institute of Technology

MATH 564 - Applied Statistics

CREDIT CARD CUSTOMER CHURN

Instructed by : Prof. Lulu Kang

Submitted by :

Akshay Singh - A20498211

Amogh Kori - A20491465

Rahul M Bharadwaj - A20502085

I. Introduction

The main objective is to select the model that predicts customer churning best by comparing them with the performance of different Machine Learning models using R. Customer churn is a common measure of lost customers. By minimizing customer churn, a company can maximize its profits. Companies recognize that existing customers are one of the most valuable assets a company could have and customer retention is critical for a good marketing strategy. The prevention of customer churn through customer retention is the core problem of Customer Relationship Management. Here, an analysis is done on purchasing behavior of bank customers from a certain dataset. A detailed analysis is worked out to convert raw customer data into meaningful and useful data that suits the buying behavior and in turn, converts this meaningful data into knowledge for which predictive data mining techniques are adopted.

II. Data Description

This dataset includes the information of banks along with its existing and attrited customers. We will use the attrition tag that is within the dataset as our target variable, we will train and test with this variable. To define the success of the solution that we will deliver let's define the metrics as: Accuracy and Recall score. These metrics were chosen since normally churn problems are imbalanced, but all depends on the definition of churn and the cost driven by each scenario. The Dataset consists of 10127 Rows and 21 Columns.

Link to Dataset - https://www.kaggle.com/varunbarath/credit-card-customers-bank-churners

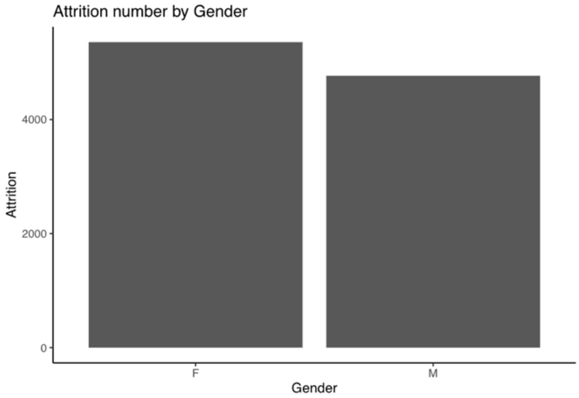
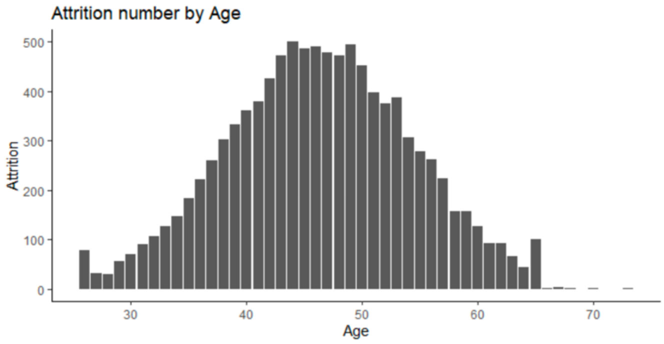
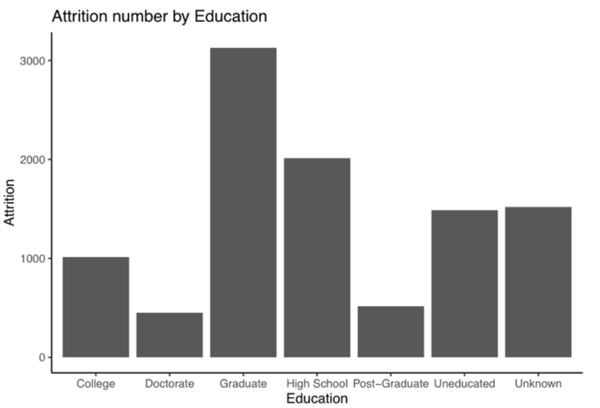
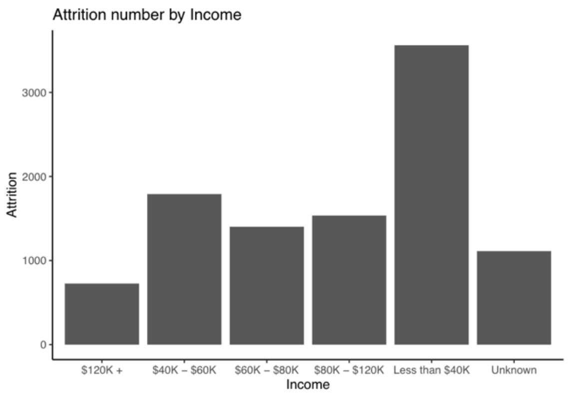
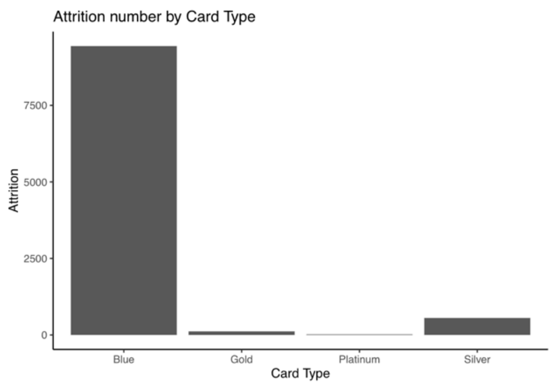
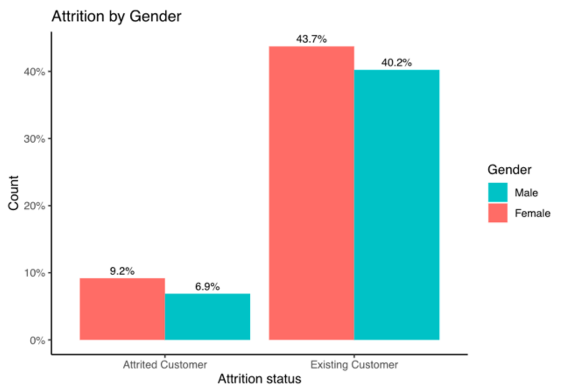
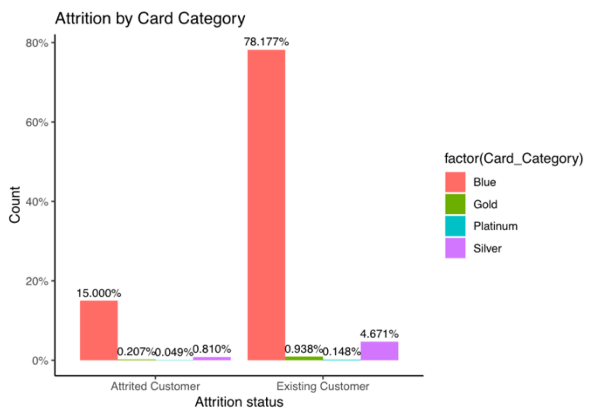
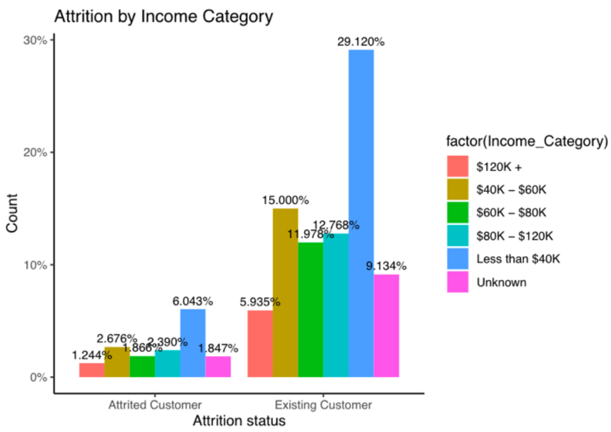
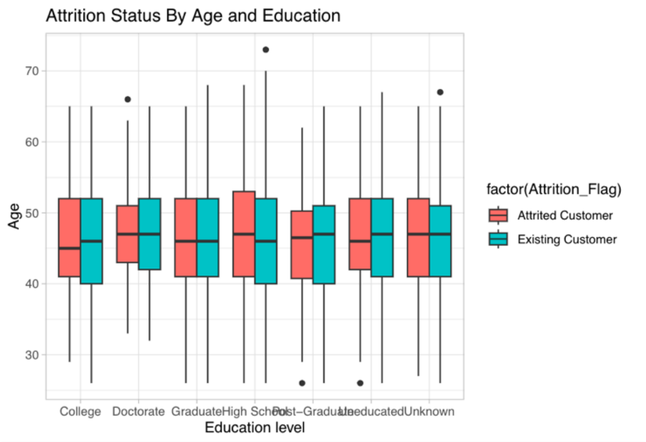
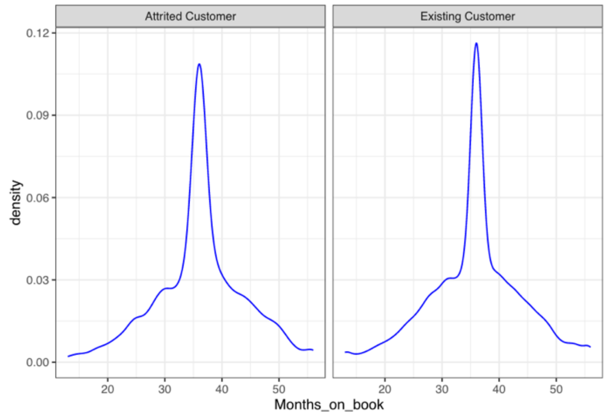
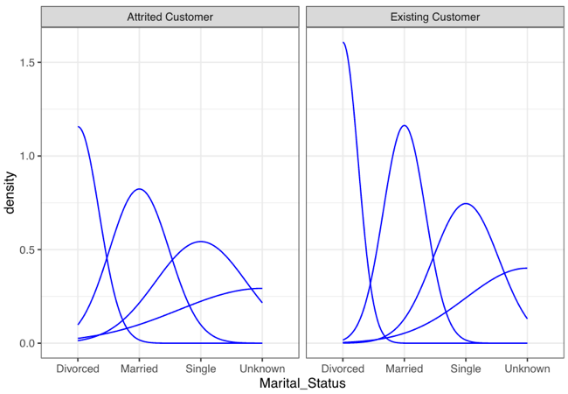
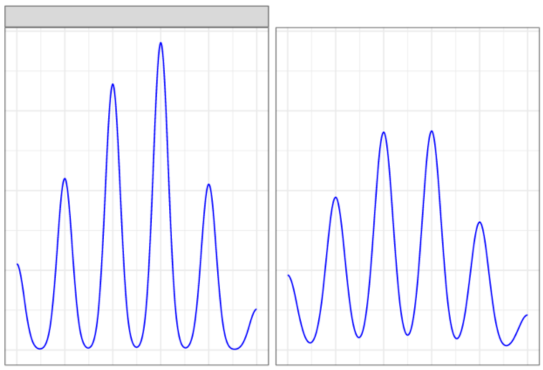
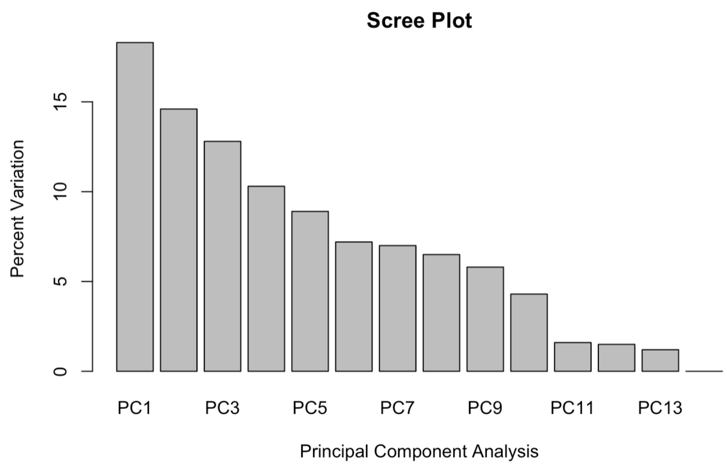
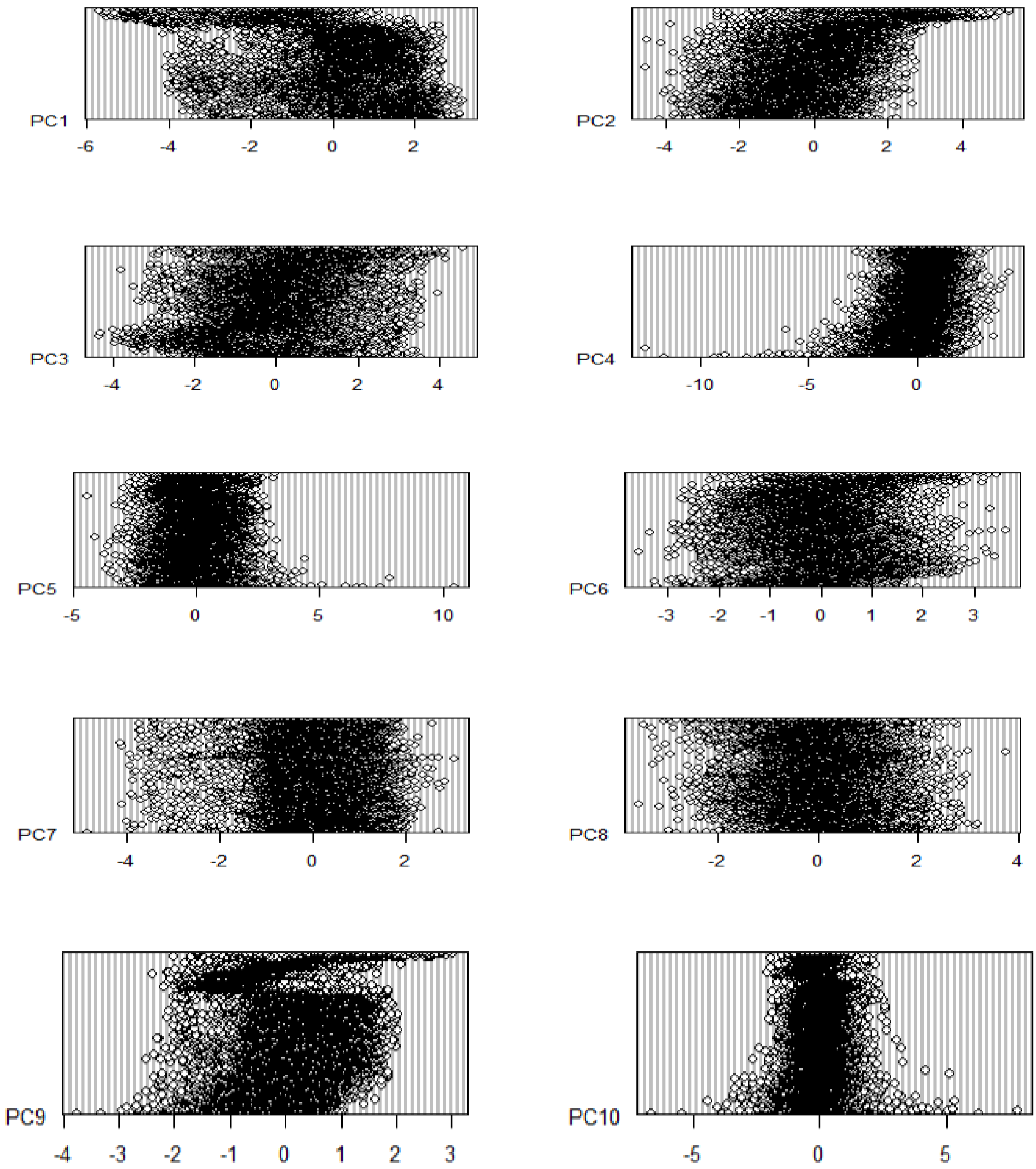
Predicted attribute: Attrition Flag

```
## 'data.frame':    10127 obs. of  23 variables:
##  $ CLIENTNUM
##  $ Attrition_Flag
##  $ Customer_Age
##  $ Gender
##  $ Dependent_count
##  $ Education_Level
##  $ Marital_Status
##  $ Income_Category
##  $ Card_Category
##  $ Months_on_book
##  $ Total_Relationship_Count
##  $ Months_Inactive_12_mon
##  $ Contacts_Count_12_mon
##  $ Credit_Limit
##  $ Total_Revolving_Bal
##  $ Avg_Open_To_Buy
##  $ Total_Amt_Chng_Q4_Q1
##  $ Total_Trans_Amt
##  $ Total_Trans_Ct
```

```
##  $ Total_Ct_Chng_Q4_Q1
##  $ Avg_Utilization_Ratio
##  $ Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educati
##  $ Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educati
sapply(churn, function(x) sum(is.na(x)))
```

III. Data Cleaning

In this step, we will check for missing and duplicate values. We will also drop the last 2 columns (Naive_Bayes_Classifier….) because it contains garbage values and will not be useful for our model.

```
```{r}
sapply(churn, function(x) sum(is.na(x)))

churn$Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12
_mon_1  <- NULL
churn$Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12
_mon_2 <- NULL
churn$CLIENTNUM <- NULL
```
```

Missing Attribute Values: None

We also converted columns to numeric data for ease of training and testing our model.

```
```{r}
#Converting all features to categorical data
churn[sapply(churn, is.character)]<- lapply(churn[sapply(churn, is.character)], as.factor)
```
```

IV. Exploratory Data Analysis





Attrition number by Gender

Attrition number by Age

You can see through this graph that Females are more prone to attrition compared to males.
You can see through this graph that Age 44 are more prone to attrition compared to people of other ages.





Attrition number by Education

Attrition number by Income

You can see through this graph that graduates are more prone to attrition compared to people of other education streams.

You can see through this graph that people earning less than 40k are more prone to attrition compared to people of other incomes.





Attrition number by Card type

Attrition Status by Gender against existing customer

You can see through this graph that people with blue card type are more prone to attrition compared to people of other card types.

Comparison of the Attrition Status between Attrited Customer Existing customers is shown above.





Attrition Status by Card Category against existing customer

Attrition Status by Income Category against existing customer

Comparison of the Attrition Status between Attrited Customer Existing customers is shown above.

Comparison of the Attrition Status between Attrited Customer Existing customers is shown above.





Attrition Status by Age and Education against existing customer Attrition Status by Months on Book against existing customer

Boxplot showing attrition status of the card holders based on age and education of the individual

Comparison of the Attrition Status between Attrited Customer Existing customers is shown above.





Attrition Status by Marital Status against existing customer Attrition Status by Dependent Count against existing customer

Comparison of the Attrition Status between Attrited Customer Existing customers is shown above.

Comparison of the Attrition Status between Attrited Customer Existing customers is shown above.

V. Principal Component Analysis

Principal component analysis (PCA) is a technique for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance.
Although our dataset does not have a huge amount of features in the traditional sense, we still wanted to use PCA to learn how it works and reduce the dimensions and to see if using PCA is a good strategy for when the number of features are about 20.

```
```{r}
#PCA
churn.pca <- prcomp(scale(churn[,c(2,4,9:20)]), center = TRUE)
summary(churn.pca)
```
```

We have only added the 14 numerical data that were there in our data to generate 14 principal components. We have also made sure to normalize the data.

Before PCA, we standardize/ normalize data. Usually, normalization is done so that all features are at the same scale. Normalization is important in PCA since it is a variance maximizing exercise. It projects your original data onto directions which maximize the variance.

In theory it is possible to apply PCA on discrete variables as well. This can be done by one hot encoding categorical variables and then applying PCA on it. However, it is not advised to do so. General rule of thumb is, if your variables don't belong on a coordinate plane, then do not apply PCA on them.

```
```{r}
#How much variation in the original data does PCA account for
churn.pca.var <- churn.pca$sdev^2
churn.pca.var.per <- round(churn.pca.var/sum(churn.pca.var)*100,1)
churn.pca.var.per
```
```

This snippet shows what percentage of variation each principal component is responsible for.

```
##  [1] 18.3 14.6 12.8 10.3  8.9  7.2  7.0  6.5  5.8  4.3  1.6  1.5  1.2  0.0
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5    PC6     PC7
## Standard deviation     1.6025 1.4301 1.3408 1.2024 1.11491 1.0019 0.99250
## Proportion of Variance 0.1834 0.1461 0.1284 0.1033 0.08879 0.0717 0.07036
## Cumulative Proportion  0.1834 0.3295 0.4579 0.5612 0.64998 0.7217 0.79203
##                            PC8     PC9    PC10    PC11    PC12    PC13
## Standard deviation     0.95112 0.89829 0.77448 0.47086 0.45909 0.40948
## Proportion of Variance 0.06462 0.05764 0.04284 0.01584 0.01505 0.01198
## Cumulative Proportion  0.85665 0.91429 0.95713 0.97297 0.98802 1.00000
##                             PC14
## Standard deviation     1.067e-15
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

Next thing we will do is create a scree plot to find out how much variation in the original data does PCA account for

```
```{r}
#Plotting PCA percentages
barplot(churn.pca.var.per, main="Screen Plot", xlab="Principal Component Analysis",
        names = c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9", "PC10",
                  "PC11", "PC12", "PC13","PC14"), ylab="Percent Variation", )

#PCA ends here
```
```



Dot chart

A dot plot or dot chart is similar to a scatter plot. The main difference is that the dot plot in R displays the index (each category) in the vertical axis and the corresponding value in the horizontal axis, so you can see the value of each observation following a horizontal line from the label.



From the above plot PC2 is left skewed. PC1, PC4 are right skewed. And PC3, PC4, PC5,PC6,PC7,PC8,PC10 are uniformly distributed.

VI. Dimension Reduction

Upon examining the scree plot and cumulative proportion, we can observe that the first 10 principal components are responsible for 95.713% of variance. Therefore, we have decided to drop the last 4 principal components while creating our model. This will reduce the model dimensions from 19 to 15.

```
```{r}
pc_data <- churn.pca$x[,1:10]
cat_data <- churn[,c(1,3,5:8)]
churn_pca <-data.frame(cat_data, pc_data)
```
```

Here, pc_data represents the data in principal components 1 through 10 and cat_data contains all the columns with categorical data in them. We have combined these two to create a single data frame by the name of 'churn_pca' on which we will create our models and compare the results with the data frame containing just features and no principal components.

VII. Data Preprocessing

```
```{r}
churn_pca[sapply(churn_pca, is.character)]<- lapply(churn_pca[sapply(churn_pca, is.character)], as.factor)
summary(churn_pca)
```
```

```
```{r}
#Converting all features to categorical data
churn[sapply(churn, is.character)]<- lapply(churn[sapply(churn, is.character)], as.factor)
```
```

Here, we are transforming all the categorical features to factors. This is a necessary step before we could create models. Upon completing this step, we begin to split our datasets. From here on we will refer to **Dataset I for the churn_pca dataset** containing principal components and **Dataset II for the churn dataset** for regular data with all the features for clarity.

Dataset I

```{r}
#Splitting the pca dataset
intrain_pca<- createDataPartition(churn_pca$Attrition_Flag, p=0.80, list = FALSE)
training_pca<- churn_pca[intrain_pca,]
testing_pca<- churn_pca[-intrain_pca,]
dim(training_pca); dim(testing_pca)
#summary(training_pca)
#summary(testing_pca)
```

```
[1] 8102   16
[1] 2025   16
```

Dataset II

```{r}
#Splitting the regular dataset
intrain_reg<- createDataPartition(churn$Attrition_Flag, p=0.80, list = FALSE)
training_reg<- churn[intrain_reg,]
testing_reg<- churn[-intrain_reg,]
dim(training_reg); dim(testing_reg)
#summary(training_reg)
#summary(testing_reg)
```

```
[1] 8102   20
[1] 2025   20
```

We have split our datasets in 80:20 ratio for training and testing data.

VIII. Models

Random Forest

Random forest is a Supervised Machine Learning Algorithm that works by combining results of multiple decision trees to reach one single result. It works well for both regression and classification problems. In a random forest, instead of working on just one decision tree, it takes into consideration each and every tree and aggregates them together to predict the most popular result. A singular decision tree is more prone to problems of bias and overfitting, as compared to multiple trees ensembled together in the random forest algorithm. Thus, they predict results with more accuracy, especially in cases where individual decision trees are correlated with each other.

In our case, dataset I predicted attrition with **91.26** accuracy and dataset II produced predictions with an accuracy score of **96.35**

Logistic Regression

Logistic Regression is a Supervised Learning technique. It’s mainly used for predicting the categorical variable by mapping the independent input-output pairs.
The outcome is a categorical or discrete value like, true or false, right or wrong, 0 or 1 etc. Though, it doesn't give exact values. It gives a probabilistic value. One major distinction between Logistic Regression and Linear Regression is that Linear Regression predicts a regression line whereas Logistic Regression is used to solve classification problems.

In dataset I, we got the accuracy score of Logistic Regression to be **90.07** and **76.05** in dataset II.

SVM

Support Vector Machine is another Supervised Learning algorithm, which is widely used for regression as well as classification problems. Although it has more real-life use cases for classification than regression in Machine Learning. The way it works is, it generates a best line or decision boundary that is the maximum distance between data points of both classes. This helps us in assigning future data to their respective class or category with ease and confidence. This model selects outliers as vectors while creating the hyperplane. These outlier vector cases are called support vectors, and hence the name Support Vector Machine.

How does SVM help in predicting attrition?
SVM models can create hyperplanes among different categorical features from our dataset and train on it. It can use the most extreme case, and create a decision boundary. With the help of the groundwork provided by the decision boundary, predict which category would be more suitable when we input our test data.
SVM’s accuracy in database I came out to be **87.21** and **86.86** for database II.

Naive Bayes

It is another supervised learning model and is used for classification problems. It is based on Bayes theorem of conditional probability. This model is easy to set up and performs really well for large datasets. It's not just easy to use, but also powerful enough to compete with and even outperform some sophisticated classification methods. A few use cases of Naive Bayes Algorithm include: Sentimental analysis, spam filtration and classification of articles.

In dataset I, the accuracy of this model is **89.78** and in dataset II, the accuracy is **89.33**.

Decision Tree

Decision Tree is a one more type of Supervised learning technique which is primarily used to solve classification problems, but in cases, can be used to solve Regression problems as well. The main idea of Decision Trees is to continuously make decisions like yes or no based on certain rules and split the dataset till the point each data point belonging to a different class is isolated. This phenomenon creates a Tree like structure, hence the name. In this tree structure, internal nodes represent features, branches represent decision rules and each leaf node represents the outcome.

In our case, the decision tree model produced an accuracy of **88.74** on dataset I and **94.07** on dataset II.

IX. Model Evaluation and Interpretation

Confusion Matrix and Statistics

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

Random Forest

Dataset I

```
Confusion matrix:
                  Attrited Customer Existing Customer class.error
Attrited Customer               675               627  0.48156682
Existing Customer               109              6691  0.01602941
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer               166                18
  Existing Customer               159              1682

               Accuracy : 0.9126
                 95% CI : (0.8994, 0.9245)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6066

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.51077
            Specificity : 0.98941
         Pos Pred Value : 0.90217
         Neg Pred Value : 0.91363
             Prevalence : 0.16049
         Detection Rate : 0.08198
   Detection Prevalence : 0.09086
      Balanced Accuracy : 0.75009

       'Positive' Class : Attrited Customer
```

Dataset II

```
Confusion matrix:
                  Attrited Customer Existing Customer class.error
Attrited Customer              1086               216  0.16589862
Existing Customer                82              6718  0.01205882
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer               265                20
  Existing Customer                60              1680

               Accuracy : 0.9605
                 95% CI : (0.9511, 0.9686)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8457

 Mcnemar's Test P-Value : 1.299e-05

            Sensitivity : 0.8154
            Specificity : 0.9882
         Pos Pred Value : 0.9298
         Neg Pred Value : 0.9655
             Prevalence : 0.1605
         Detection Rate : 0.1309
   Detection Prevalence : 0.1407
      Balanced Accuracy : 0.9018

       'Positive' Class : Attrited Customer
```

Logistic Regression

Dataset I

```
Confusion Matrix and Statistics

      target
y_pred    1    2
     1  165   41
     2  160 1659

               Accuracy : 0.9007
                 95% CI : (0.8869, 0.9134)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : 1.038e-15

                  Kappa : 0.5676

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.50769
            Specificity : 0.97588
         Pos Pred Value : 0.80097
         Neg Pred Value : 0.91204
             Prevalence : 0.16049
         Detection Rate : 0.08148
   Detection Prevalence : 0.10173
      Balanced Accuracy : 0.74179

       'Positive' Class : 1
```

Dataset II

```
      target
y_pred    1    2
     1   44  204
     2  281 1496

               Accuracy : 0.7605
                 95% CI : (0.7413, 0.7789)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : 1.0000000

                  Kappa : 0.017

 Mcnemar's Test P-Value : 0.0005586

            Sensitivity : 0.13538
            Specificity : 0.88000
         Pos Pred Value : 0.17742
         Neg Pred Value : 0.84187
             Prevalence : 0.16049
         Detection Rate : 0.02173
   Detection Prevalence : 0.12247
      Balanced Accuracy : 0.50769

       'Positive' Class : 1
```

SVM

Dataset I

```
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer                94                12
  Existing Customer               231              1688

               Accuracy : 0.88
                 95% CI : (0.865, 0.8938)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : 1.563e-07

                  Kappa : 0.3879

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.28923
            Specificity : 0.99294
         Pos Pred Value : 0.88679
         Neg Pred Value : 0.87962
             Prevalence : 0.16049
         Detection Rate : 0.04642
   Detection Prevalence : 0.05235
      Balanced Accuracy : 0.64109

       'Positive' Class : Attrited Customer
```

Dataset II

```
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer                62                 5
  Existing Customer               263              1695

               Accuracy : 0.8677
                 95% CI : (0.8521, 0.8821)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : 0.0002322

                  Kappa : 0.2766

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.19077
            Specificity : 0.99706
         Pos Pred Value : 0.92537
         Neg Pred Value : 0.86568
             Prevalence : 0.16049
         Detection Rate : 0.03062
   Detection Prevalence : 0.03309
      Balanced Accuracy : 0.59391

       'Positive' Class : Attrited Customer
```

Naïve Bayes

Dataset I

```
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer               146                28
  Existing Customer               179              1672

               Accuracy : 0.8978
                 95% CI : (0.8838, 0.9106)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : 2.646e-14

                  Kappa : 0.5329

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.44923
            Specificity : 0.98353
         Pos Pred Value : 0.83908
         Neg Pred Value : 0.90330
             Prevalence : 0.16049
         Detection Rate : 0.07210
   Detection Prevalence : 0.08593
      Balanced Accuracy : 0.71638

       'Positive' Class : Attrited Customer
```

Dataset II

```
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer               203               127
  Existing Customer               122              1573

               Accuracy : 0.877
                 95% CI : (0.8619, 0.891)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : 1.156e-06

                  Kappa : 0.5465

 Mcnemar's Test P-Value : 0.7999

            Sensitivity : 0.6246
            Specificity : 0.9253
         Pos Pred Value : 0.6152
         Neg Pred Value : 0.9280
             Prevalence : 0.1605
         Detection Rate : 0.1002
   Detection Prevalence : 0.1630
      Balanced Accuracy : 0.7750

       'Positive' Class : Attrited Customer
```

Decision Tree

Dataset I

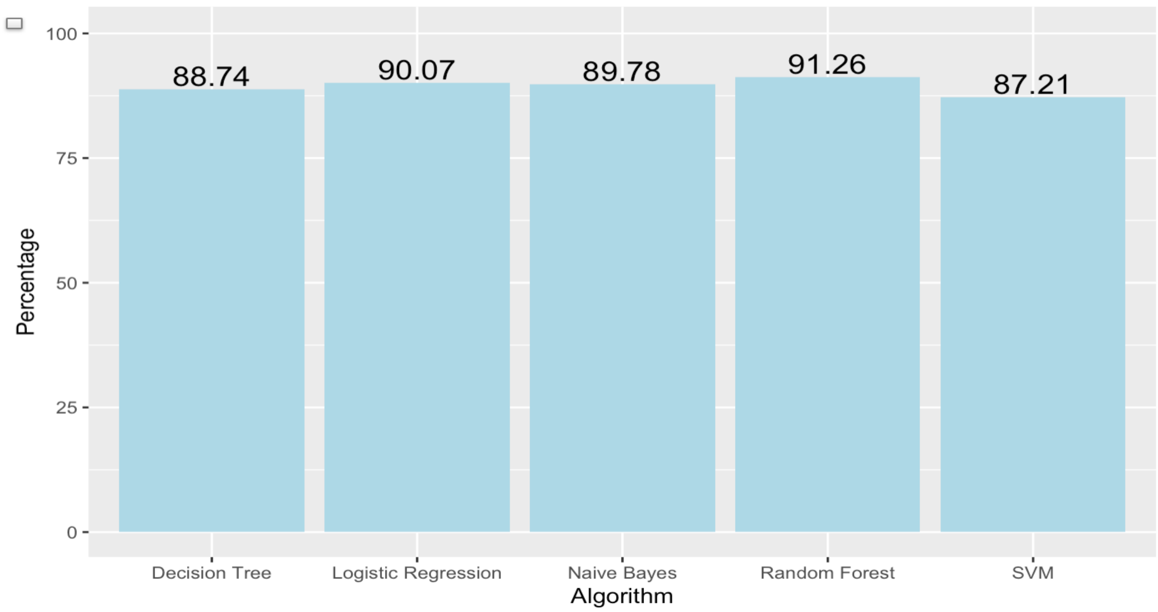
```
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer               171                74
  Existing Customer               154              1626

               Accuracy : 0.8874
                 95% CI : (0.8728, 0.9009)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : 5.090e-10

                  Kappa : 0.536

 Mcnemar's Test P-Value : 1.678e-07

            Sensitivity : 0.52615
            Specificity : 0.95647
         Pos Pred Value : 0.69796
         Neg Pred Value : 0.91348
             Prevalence : 0.16049
         Detection Rate : 0.08444
   Detection Prevalence : 0.12099
      Balanced Accuracy : 0.74131

       'Positive' Class : Attrited Customer
```

Dataset II

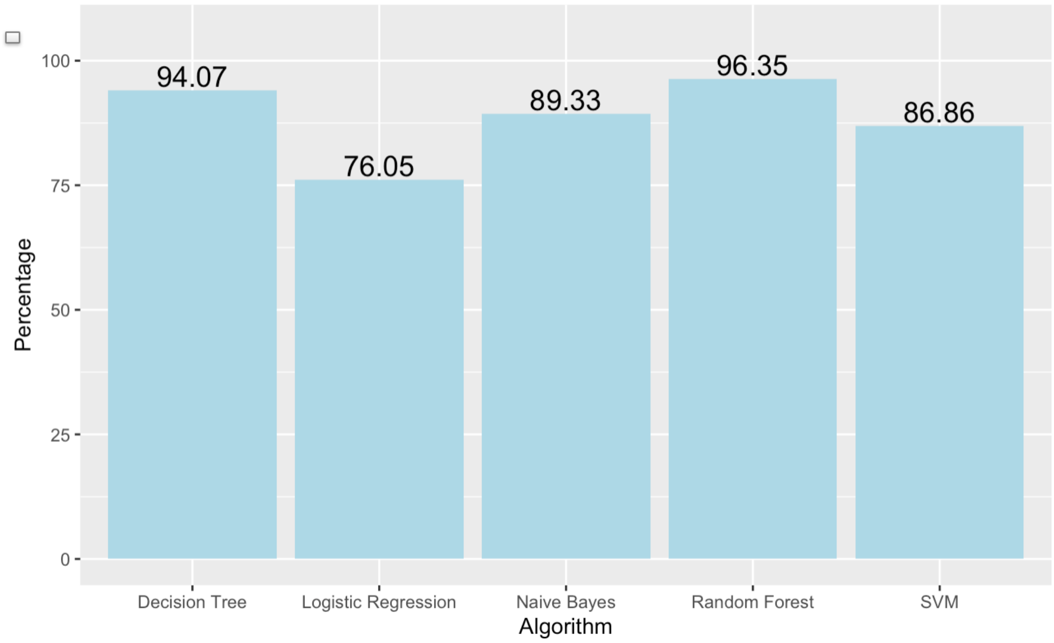
```
Confusion Matrix and Statistics

                   Reference
Prediction          Attrited Customer Existing Customer
  Attrited Customer               246                59
  Existing Customer                79              1641

               Accuracy : 0.9319
                 95% CI : (0.92, 0.9424)
    No Information Rate : 0.8395
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7406

 Mcnemar's Test P-Value : 0.1058

            Sensitivity : 0.7569
            Specificity : 0.9653
         Pos Pred Value : 0.8066
         Neg Pred Value : 0.9541
             Prevalence : 0.1605
         Detection Rate : 0.1215
   Detection Prevalence : 0.1506
      Balanced Accuracy : 0.8611

       'Positive' Class : Attrited Customer
```

Accuracy Percentage of All Models

Dataset I



Dataset II



X. Conclusion

We can conclude with certainty that 'Random Forest' will be the best model for predicting credit card customer attrition for both datasets since it predicted customer attrition with the highest accuracy in both datasets – with reduced dimensions and principal components, and the regular one.
We can also observe that Naïve Bayes and SVM perform a tiny bit better with PCA data whereas the accuracy of Decision Tree and Random Forest dipped a bit when we used the PCA data. From this, we can infer that a dataset with 20 dimensions isn't big enough where we would need to perform PCA to reduce dimensions.

SVM model takes more time to execute than expected. Currently, we are working with a database with about 10,000 rows, and although the time taken is manageable. But, in case we have to work on an even bigger database in future, we will have to be careful about using this model.

Time management and coordination – Since all 3 of us have completely different other 2 subjects, and had projects and assignments in those subjects as well, it was a bit challenging working together as a team and getting everyone to show up at the library at a time available to all three.

XI. References

1. *Dataset link - https://www.kaggle.com/code/varunbarath/credit-card-customers-bank-churners/data*
2. *Logistic Regression - https://www.javatpoint.com/logistic-regression-in-machine-learning*
3. *SVM - https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm*
4. *Decision tree - https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm*

XIII. Contributions

Akshay Singh - Exploratory Data Analysis, Comparison between PCA and without PCA, Documentation
Amogh A Kori - Model Building, Documentation
Rahul M Bharadwaj - PCA, Debugging, Model Evaluation and Confusion Matrix, Documentation