Credit Card Fraud Prediction

MOHAMED FARHAN

TARGET PROBLEM

- UCI credit card dataset contains financial information using which we intend to predict if a person is likely to default on their payment.
- Some of the factors used to make such predictions include information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients.
- The output variable is categorical which can have values either 0 (NOT default) and 1 (defaults).
- Since we are expected to predict the label for observations using predictor variables, such a model will be identified as a supervised learning model.

DATASET DESCRIPTION

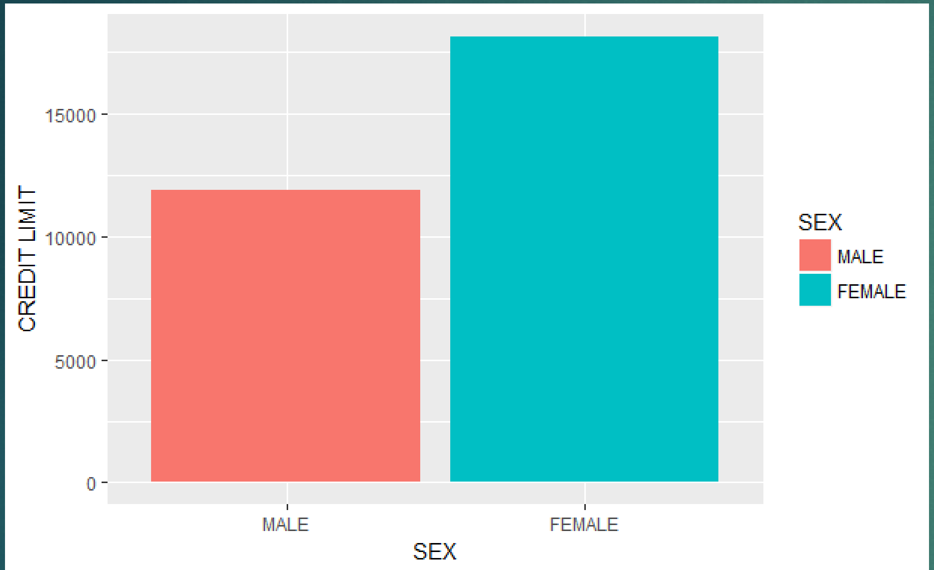
- **Name:** UCI_Credit_Card.csv
- **Number of samples:** 30000
- **Number of attributes:** 25
- **Source:** https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
- **Target Variable:**
- Default.payment.next.month will be the target variable for my project. It will determine if a
- person will default on their credit card bill payment (1=yes, 0=no).

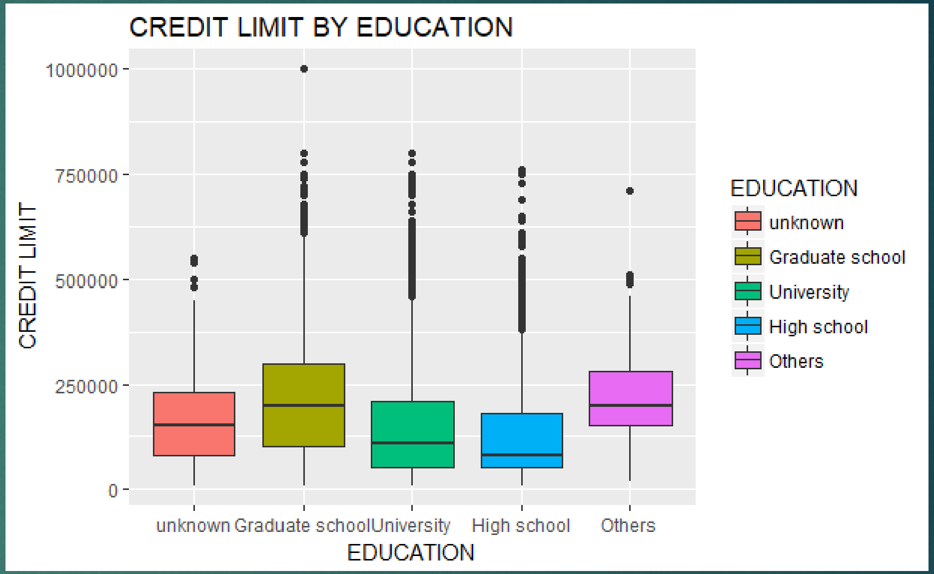
DATA PREPROCESSING

- We look for n/a, NaN and infinity values in the dataset which may produce skewed results. This is an integral step in preprocessing as it may influence the model's accuracy significantly.
- We find that no such values are present in the dataset.

```
> sapply(credit,function(count) sum(is.na(count)))
                   ID           LIMIT_BAL                 SEX           EDUCATION
                    0                   0                   0                   0
             MARRIAGE                 AGE               PAY_0               PAY_2
                    0                   0                   0                   0
                PAY_3               PAY_4               PAY_5               PAY_6
                    0                   0                   0                   0
            BILL_AMT1           BILL_AMT2           BILL_AMT3           BILL_AMT4
                    0                   0                   0                   0
            BILL_AMT5           BILL_AMT6            PAY_AMT1            PAY_AMT2
                    0                   0                   0                   0
             PAY_AMT3            PAY_AMT4            PAY_AMT5            PAY_AMT6
                    0                   0                   0                   0
default.payment.next.month
                    0
```

INTERESTING PLOTS





CORRELATION

- Correlation matrix can be used to analyze which variables in the dataset are strongly connected to each other.
- Correlation can be both positive or negative.
- If correlation between two variables is high, we can drop any one of the two variables to reduce collinearity among independent variables.

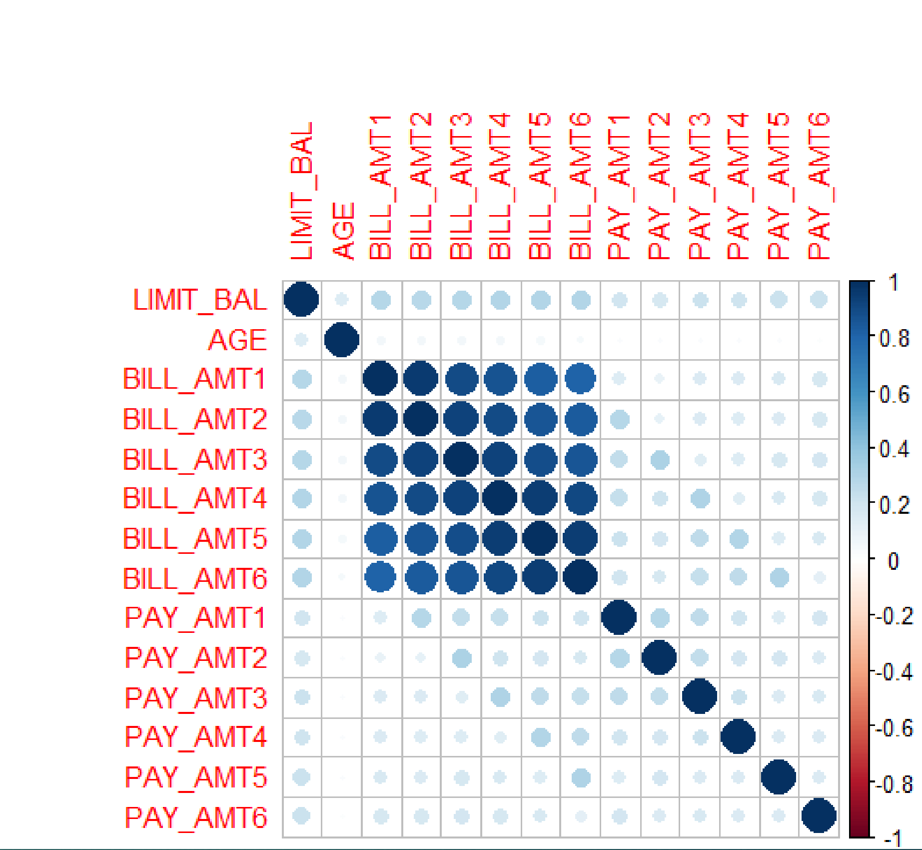
```
> cor_res<-cor(credit_newfact,use="pairwise.complete.obs",method="pearson")
> round(cor_res,3)
```

	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
LIMIT_BAL	1.000	0.145	0.285	0.278	0.283	0.294	0.296	0.290	0.195	0.178
AGE	0.145	1.000	0.056	0.054	0.054	0.051	0.049	0.048	0.026	0.022
BILL_AMT1	0.285	0.056	1.000	0.951	0.892	0.860	0.830	0.803	0.140	0.099
BILL_AMT2	0.278	0.054	0.951	1.000	0.928	0.892	0.860	0.832	0.280	0.101
BILL_AMT3	0.283	0.054	0.892	0.928	1.000	0.924	0.884	0.853	0.244	0.317
BILL_AMT4	0.294	0.051	0.860	0.892	0.924	1.000	0.940	0.901	0.233	0.208
BILL_AMT5	0.296	0.049	0.830	0.860	0.884	0.940	1.000	0.946	0.217	0.181
BILL_AMT6	0.290	0.048	0.803	0.832	0.853	0.901	0.946	1.000	0.200	0.173
PAY_AMT1	0.195	0.026	0.140	0.280	0.244	0.233	0.217	0.200	1.000	0.286
PAY_AMT2	0.178	0.022	0.099	0.101	0.317	0.208	0.181	0.173	0.286	1.000
PAY_AMT3	0.210	0.029	0.157	0.151	0.130	0.300	0.252	0.234	0.252	0.245
PAY_AMT4	0.203	0.021	0.158	0.147	0.143	0.130	0.293	0.250	0.200	0.180
PAY_AMT5	0.217	0.023	0.167	0.158	0.180	0.160	0.142	0.308	0.148	0.181
PAY_AMT6	0.220	0.019	0.179	0.174	0.182	0.178	0.164	0.115	0.186	0.158

	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
LIMIT_BAL	0.210	0.203	0.217	0.220
AGE	0.029	0.021	0.023	0.019
BILL_AMT1	0.157	0.158	0.167	0.179
BILL_AMT2	0.151	0.147	0.158	0.174
BILL_AMT3	0.130	0.143	0.180	0.182
BILL_AMT4	0.300	0.130	0.160	0.178
BILL_AMT5	0.252	0.293	0.142	0.164
BILL_AMT6	0.234	0.250	0.308	0.115
PAY_AMT1	0.252	0.200	0.148	0.186
PAY_AMT2	0.245	0.180	0.181	0.158
PAY_AMT3	1.000	0.216	0.159	0.163
PAY_AMT4	0.216	1.000	0.152	0.158
PAY_AMT5	0.159	0.152	1.000	0.155
PAY_AMT6	0.163	0.158	0.155	1.000

CORRELATION (CONTINUED)

- We observe that variables BILL_AMT1 & BILL_AMT2, BILL_AMT5 & BILL_AMT6, BILL_AMT3 & BILL_AMT4 are highly correlated.
- Therefore we can drop either of the variables from each pair.



PRINCIPAL COMPONENT ANALYSIS (PCA)

- **PCA tells us how many principal components put together represent the most important or relevant information.**

```
> summary(all_pca)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
Standard deviation     1.7988 1.2148 0.99753 0.94896 0.93359 0.92150 0.87269 0.84208 0.38779 0.22595
Proportion of Variance 0.3236 0.1476 0.09951 0.09005 0.08716 0.08492 0.07616 0.07091 0.01504 0.00511
Cumulative Proportion  0.3236 0.4712 0.57066 0.66071 0.74787 0.83279 0.90895 0.97986 0.99489 1.00000
```

- **From the above summary, we can say that the first 7 PCs cover the maximum relevant/important information to perform analysis or to fit a model.**

TRAINING AND TEST SETS

- The training set contains 60% of the entire dataset.
- The test set contains remaining 40% of the dataset.

NEED FOR OVERSAMPLING

Before oversampling

- **We find that both the training and test sets have a very small percentage of the positive (1) class.**
- **Therefore, we need to balance both the classes before fitting the model.**

```
> table(train$default.payment.next.month)

    0     1
14028  3972
> table(test$default.payment.next.month)

   0    1
9336 2664
```

After oversampling

- **To overcome this, ROSE() from library ROSE is used to ensure that the classes are balanced.**

```
> train.rose<-ROSE(default.payment.next.month~.,data=train,seed=1)$data
> table(train.rose$default.payment.next.month)

   0    1
9063 8937
>
> test.rose<-ROSE(default.payment.next.month~.,data=test,seed=1)$data
> table(test.rose$default.payment.next.month)

   0    1
6040 5960
```

LOGISTIC REGRESSION MODEL

- Logistic regression model uses logistic function and also takes into consideration the p-value. Variables with large p-values can be omitted.
- We use the first 8 principal components from our PCA along with other categorical predicators to fit the model.
- Here we use both forward and backward method for variable selection.
- We observe that the model chooses to use only 14 predictors out 18 predictors.

```
> summary(model_log.fit3)

Call:
glm(formula = default.payment.next.month ~ SEX + EDUCATION +
    MARRIAGE + PAY_0 + PAY_3 + PAY_4 + PAY_5 + PAY_6 + PC1 +
    PC2 + PC3 + PC5 + PC6 + PC8, family = "binomial", data = train.rose)
```

RESULTS OBTAINED USING LOGISTIC REGRESSION MODEL

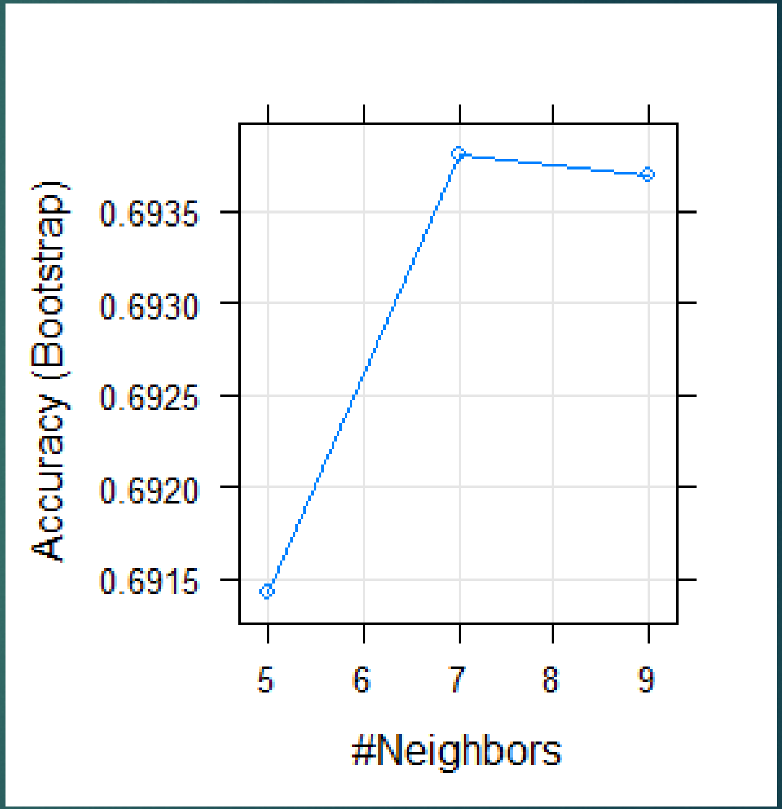
- After running the model on the test set, the following statistics are obtained:
- Accuracy: 55.6%
- Error rate: 44.4%

```
> confusionMatrix(ifelse(p_3>0.5,1,0),test.rose$default.payment.next.month)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1430  887
         1 3109 3573

               Accuracy : 0.556
```

k-NEAREST NEIGHBORS (kNN) MODEL

- We use the first 8 principal components from our PCA along with other categorical predicators to fit the model.
- The data is centered and scaled before fitting.

```
> plot(knnFit_pca)
> knnFit_pca
k-Nearest Neighbors

21001 samples
   17 predictor
    2 classes: '0', '1'

Pre-processing: centered (76), scaled (76)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 21001, 21001, 21001, 21001, 21001, 21001, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.6914236  0.3830607
  7  0.6938054  0.3877955
  9  0.6936877  0.3875275

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was k = 7.
```



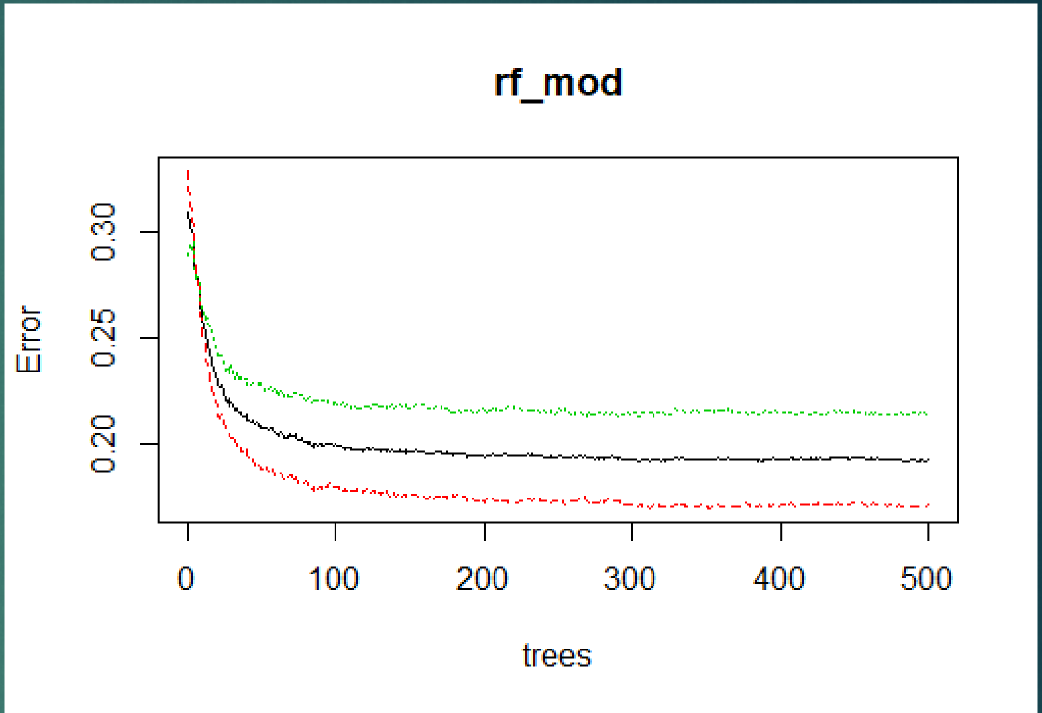
RESULTS OBTAINED USING kNN MODEL

- After running the model on the test set, the following statistics are obtained:
- Accuracy: 65.86%
- Error rate: 34.14%
- Optimal k: 7

```
> knnPredict_pca <- predict(knnFit_pca,newdata = test.rose)
> confusionMatrix(knnPredict_pca, test.rose$default.payment.next.month )
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2927 1460
         1 1612 3000

               Accuracy : 0.6586
                 95% CI : (0.6487, 0.6684)
    No Information Rate : 0.5044
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.3174
 Mcnemar's Test P-Value : 0.006442

            Sensitivity : 0.6449
            Specificity : 0.6726
         Pos Pred Value : 0.6672
         Neg Pred Value : 0.6505
             Prevalence : 0.5044
         Detection Rate : 0.3253
   Detection Prevalence : 0.4875
      Balanced Accuracy : 0.6588

       'Positive' Class : 0
```

RANDOM FOREST MODEL

- Random Forest is a powerful model when we have several categorical predictors in our data.
- We use the first 8 principal components from our PCA along with other categorical predicators to fit the model.
- We obtain Out-of-Bag (OOB) error as 25.69%




```
> rf_pca

Call:
 randomForest(formula = default.payment.next.month ~ ., data = train.rose)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 25.69%
Confusion matrix:
     0    1 class.error
0 8493 2064   0.1955101
1 3331 7113   0.3189391
```

RESULTS OBTAINED USING RANDOM FOREST MODEL

- After running the model on the test set, the following statistics are obtained:
- Accuracy: 72.99%
- Error rate: 27.01%
- Larger the Mean Decrease Gini, more important the variable is for the model.

```
> confusionMatrix(rf_pred_pca, test.rose$default.payment.next.month )
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3698 1590
         1  841 2870

               Accuracy : 0.7299
```

```
> importance(rf_pca)
          MeanDecreaseGini
SEX               117.2366
EDUCATION         242.0976
MARRIAGE          140.4475
PAY_0            1165.8346
PAY_2             556.5555
PAY_3             340.1938
PAY_4             314.1530
PAY_5             247.6370
PAY_6             261.3334
PC1               842.2002
PC2               989.4754
PC3               839.1043
PC4               861.8754
PC5               927.0761
PC6               863.7516
PC7               800.7965
PC8               943.3868
```

CONCLUSION

- Three different models were used to measure performance
- Logistic regression model
- kNN model
- Random forest model.

We observed that Random forest model performs better compared to other models discussed providing us with an accuracy of 72.99%.