Predictive analytics on German credit data.

Created By:
Geethanjali Shivanna
Juhi Sharma
Neelam Mishra
Neethu Sreerangam

Agenda:

- Introduction
- Project Objective
- Data Description
- Data Review
- Decision Tree(Juhi)
- Data cleaning
- Train and test split
- Model training
- Model Visualization
- Random forest(Juhi)
- Model training
- Model visualization

Introduction:

Credit risk modelling refers to the use of financial models to estimate losses a firm might suffer in the event of a borrower's default. Financial institutions deploy models that draw upon the credit history of borrowers, third-party data – such as rating agency data – and inputs from their own economic stress scenarios to measure credit risk.

Credit risk capital modelling refers to the use of these models to gauge minimum requirements to set aside as a buffer against such losses. Banks permitted to use this family of approaches must measure two components: a borrower's probability of default, and the bank's own loss given default. The results help banks allocate loss provisions and set regulatory capital, among other uses.

Project Objective

The objective of this project is to minimize loss from the bank's perspective, the bank needs a decision rule regarding who to give approval of the loan and who not to. An applicant's demographic and socio-economic profiles are considered by loan managers before a decision is taken regarding his/her loan application.

When a bank receives a loan application, based on the applicant's profile the bank has to make a decision regarding whether to go ahead with the loan approval or not. Two types of risks are associated with the bank's decision.

If the applicant is a good credit risk, i.e. is likely to repay the loan, then not approving the loan to the person results in a loss of business to the bank.

If the applicant is a bad credit risk, i.e. is not likely to repay the loan, then approving the loan to the person results in a financial loss to the bank.

Data Overview:

The German Credit Data contains data on 10 variables and the classification whether an applicant is considered a Good or a Bad credit risk for 1000 loan applicants. Here is the link German Credit data (https://www.kaggle.com/sanyalush/predicting-credit-risk/data). A predictive model developed on this data is expected to provide a bank manager guidance for deciding whether to approve a loan to a prospective applicant based on his/her profiles.

The original dataset contains 1000 entries with 20 categorial/symbolic attributes prepared by Prof. Hofmann. In this dataset, each entry represents a person who takes a credit by a bank. Each person is classified as good or bad credit risks according to the set of attributes. The dataset is taken as bank's records about the status of loan defaults and the profile of customers. Several columns are simply ignored, because in my opinion either they are not important, or their descriptions are obscure.

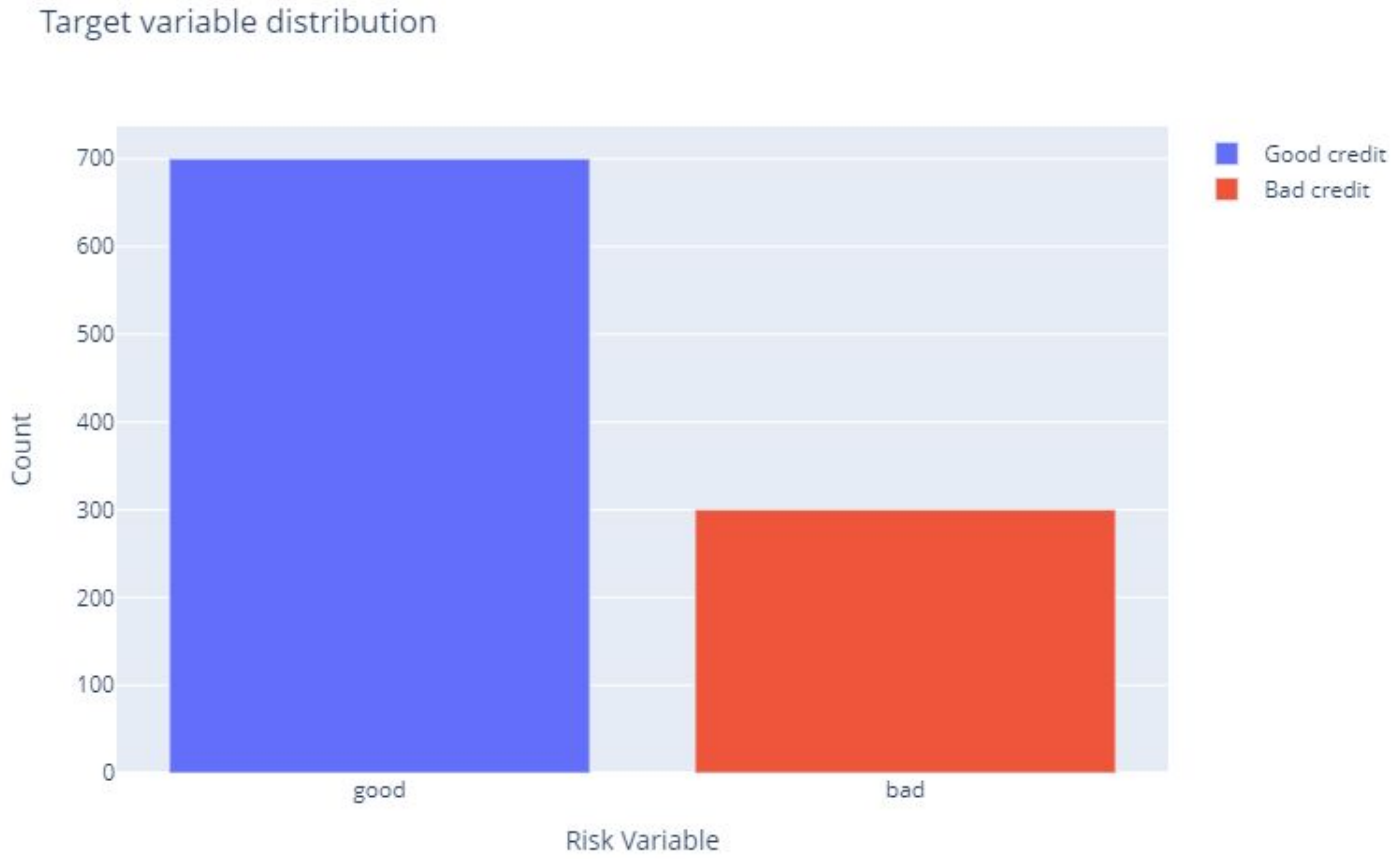
Attributes : Age , Sex , Job ,Housing ,Saving accounts ,Checking account ,Credit amount ,Duration , Purpose, & Risk.

Data Prep for Model building

- Dropping all Missing values from Checking Account & Savings Account Columns.
- Models have been applied on 522 observations and 11 Attributes.
- Train test Split has been done with 70% train and 30% Test dataset to check the effectiveness of model.

Predictors: Age , Sex , Job ,Housing ,Saving accounts ,Checking account ,Credit amount ,Duration , Purpose.

Target : Risk



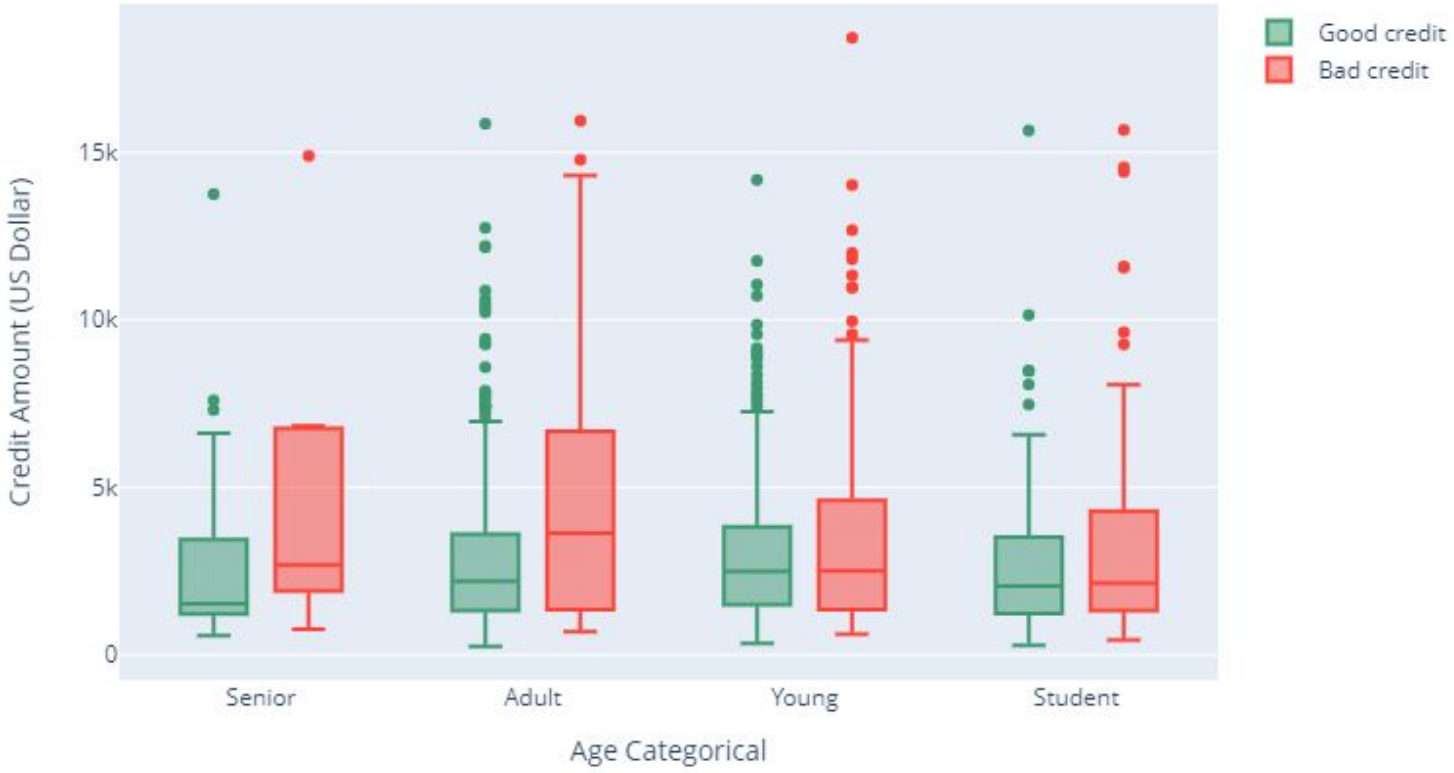
Distribution of Target variable:

The target variable has a good credit of 70% and 30% of bad credit. There already seems biasedness in the dataset. And that might affect our analysis outcome.

Distribution of Age Variable:

The distribution of age is skewed to left.

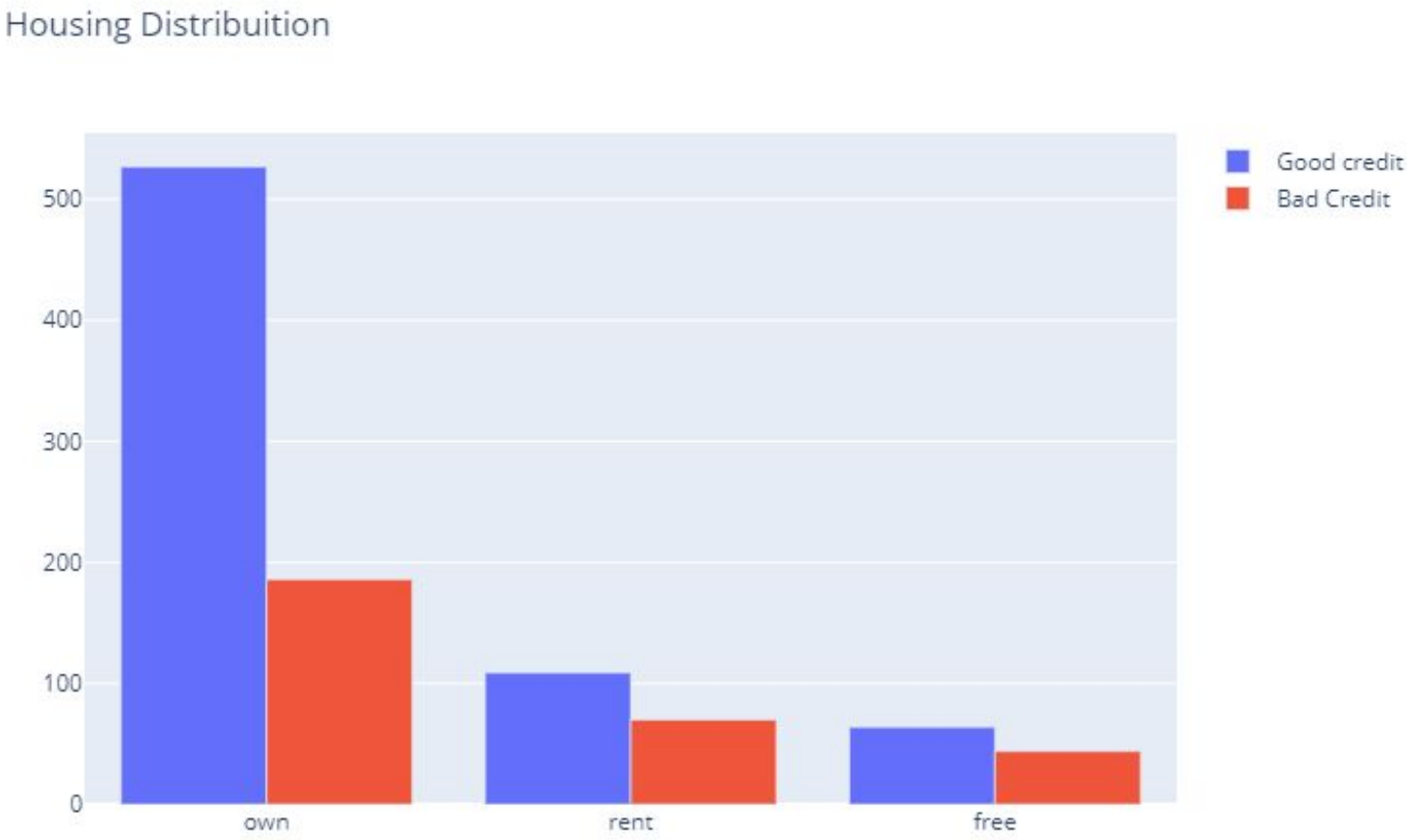
We can see that generally, throughout the age distribution, bad loans tend to be of a higher amount but when it comes to people over 55, this gap widens.



The age categories:

The bad credit could be seen more in the age group of Adult and Senior.

The good credit is seen second most in young then followed by students age group.

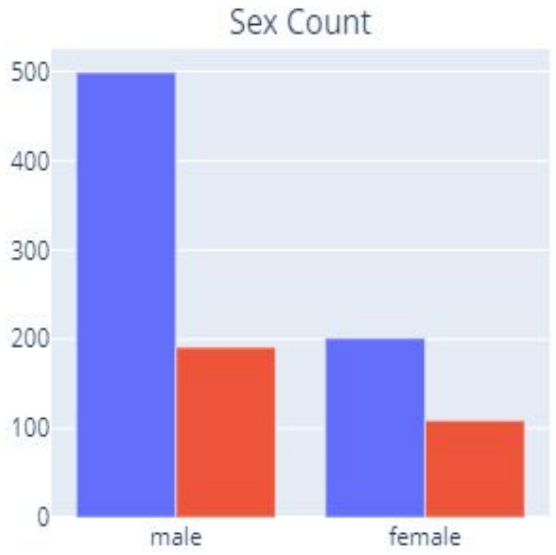


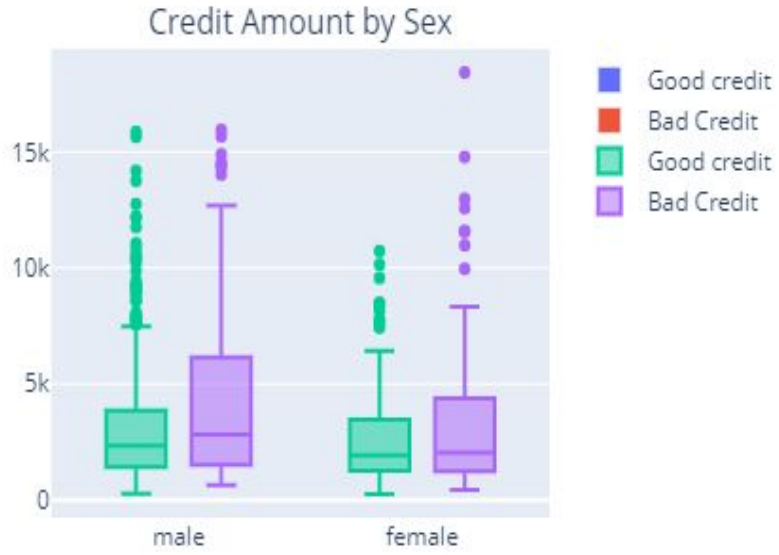
Distribution of Housing Variable:

The good credit is mostly seen in the people who have own houses.

The bad credit is also highest in the people who have own houses.

Sex Distribuition



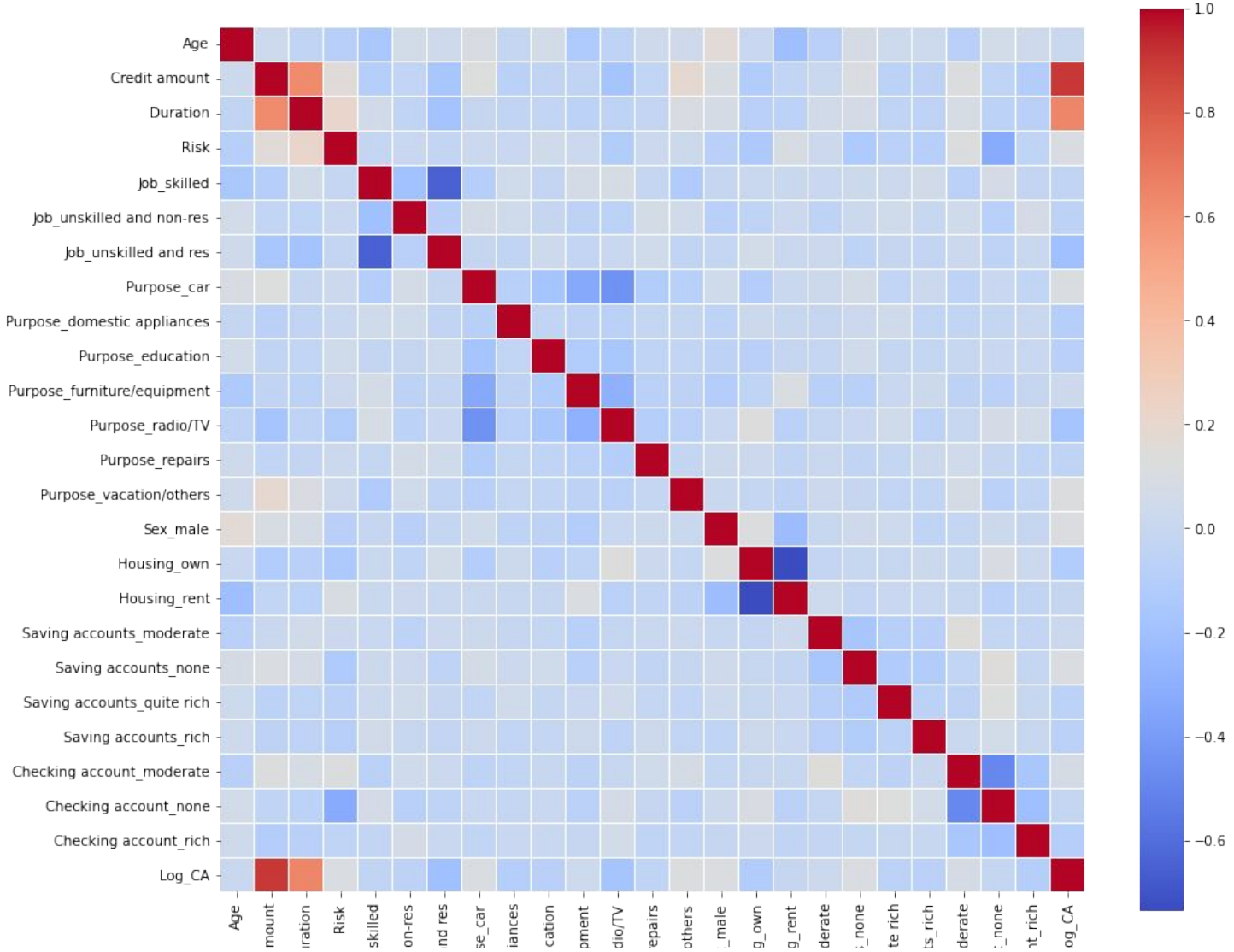


Distribution of Gender Variable:

Men tend to get a 'good' rating much more often than women. Eventually men tend to have bad credit amount when compared to women.

Distribution of saving accounts and credit amount:

In saving accounts, the richer you are, the more likely you are to be classified as good. In the savings accounts, there are visibly more good ratings than bad ones.



Feature Engineering, Correlation Plot

Correlation plot is begin used to find the multi collinearity between the variables and choose the final variables that suits for the model.

Support Vector Classification

```
svc = SVC(kernel='linear', gamma=10, C=0.8)
svc.fit(X_train, y_train)
y_pred_svc = svc.predict(X_test)
print(accuracy_score(y_pred_svc, y_test))
print(confusion_matrix(y_test, y_pred_svc))
print(classification_report(y_test, y_pred_svc))
```

```
0.76
[[129  12]
 [ 36  23]]
              precision    recall  f1-score   support

           0       0.78      0.91      0.84       141
           1       0.66      0.39      0.49        59

    accuracy                           0.76       200
   macro avg       0.72      0.65      0.67       200
weighted avg       0.75      0.76      0.74       200
```

Model Evaluation:

The accuracy of the SVC model = 0.76

TP +TN/TP+FP+TN+FN = 0.76

Precision = TP/TP +FN = 0.91

Recall = TN/TN+FP = 0.37

F1 Score=TP+1/TP+½ (FP+FN)= 84

4.1. Logistic Regression

In [31]:
```
log = LogisticRegression()
log.fit(X_train, y_train)
y_pred_log = log.predict(X_test)
print(accuracy_score(y_pred_log, y_test))
print(confusion_matrix(y_test, y_pred_log))
print(classification_report(y_test, y_pred_log))
```

```
0.74
[[126  15]
 [ 37  22]]
              precision    recall  f1-score   support

           0       0.77      0.89      0.83       141
           1       0.59      0.37      0.46        59

    accuracy                           0.74       200
   macro avg       0.68      0.63      0.64       200
weighted avg       0.72      0.74      0.72       200
```

Model Evaluation:

The accuracy of the Logistic Regression = TP +TN/TP+FP+TN+FN = 0.74.

Precision = TP/TP +FN = 0.59

Recall = TN/TN+FP = 0.37

F1 Score=TP+1/TP+½(FP+FN)= 0.46

AUC is coming as 0.615.

Decision Tree (data loading and cleaning):

- Data loading and cleaning
  - Dropped all null values from the data using

    drop_na(credit_data_df)

  - Since decision tree only works with numerical data and we had few non-numerical columns like Sex, Housing, Saving.accounts etc. We converted these columns into numerical values by applying

    transform(credit_data_df, Sex = as.numeric(as.factor(Sex)))

Decision Tree (train and test split)

- **Train and test split**

  - We divided the initial data into train and test data with 70-30 ratio.

```
split <- sample.split(credit_data_df_numeric, SplitRatio =  0.7)

train <- subset(credit_data_df_numeric, split == TRUE)

test <- subset(credit_data_df_numeric, split == FALSE)
```

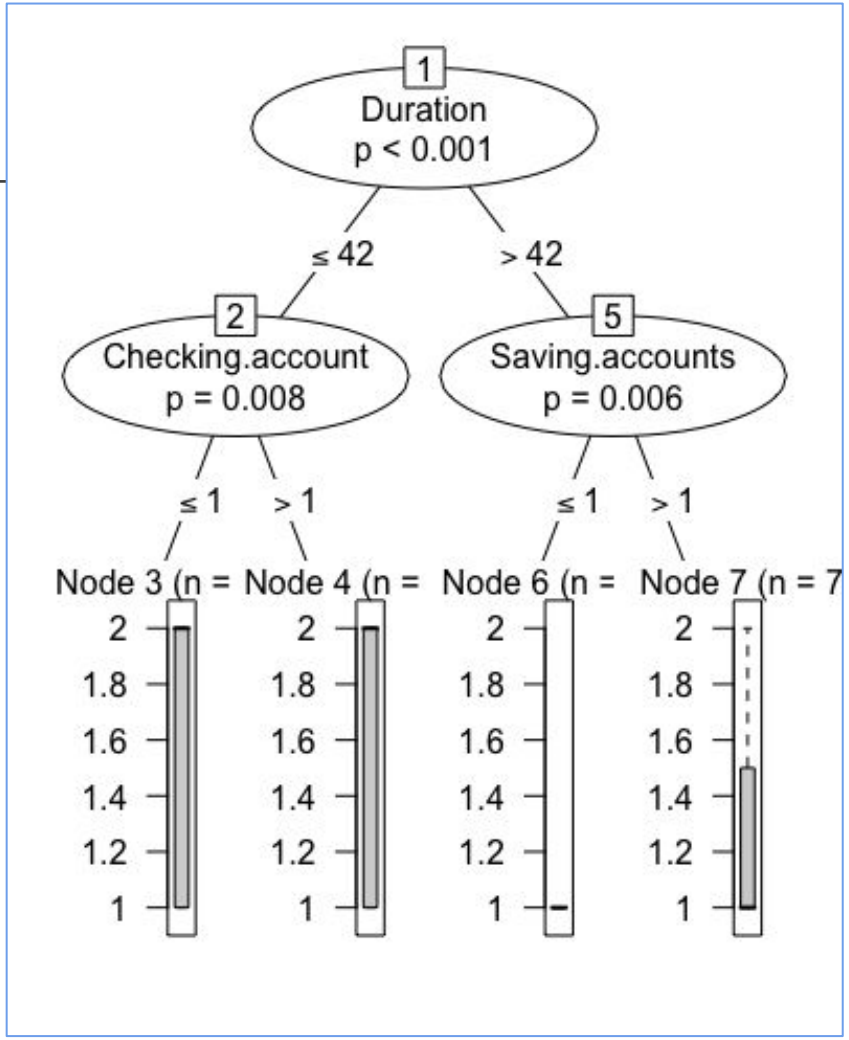
Decision tree (model training)

- **Model training**
  - Model was trained using decision tree algorithm using

```
ctree(Risk ~ Age + Sex + Job + Housing +
            Saving.accounts + Checking.account + Credit.amount +
            Duration + Purpose,
            data = train)
```

Decision tree (visualization)

- Response: Risk
- Inputs: Age, Sex, Job, Housing, Saving.accounts, Checking.account, Credit.amount, Duration, Purpose
- Number of observations: 332
- Duration <= 42; criterion = 1, statistic = 30.846
  - Checking.account <= 1; criterion = 0.992, statistic = 11.119 (Weights = 153)
  - Checking.account > 1 (Weights = 150)
- Duration > 42
  - Saving.accounts <= 1; criterion = 0.994, statistic = 11.566 (Weights = 22)
  - Saving.accounts > 1 (Weights = 7)



Random forest (model training)

- Used Risk against all other column

```
output.forest <- randomForest(Risk ~ Age + Sex + Job + Housing +
                                Saving.accounts + Checking.account + Credit.amount +
                                Duration + Purpose,
                              data = train)
```

Random forest (model visualization)

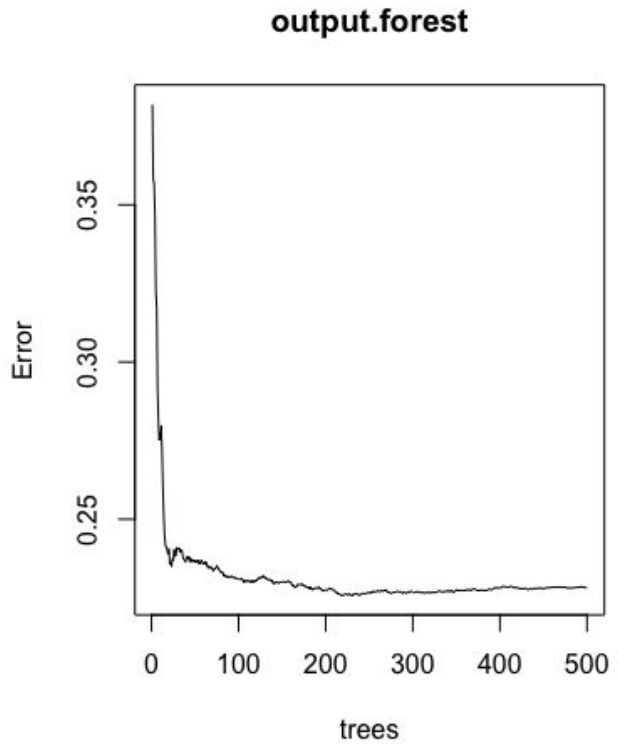
Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 3

Mean of squared residuals: 0.2282063

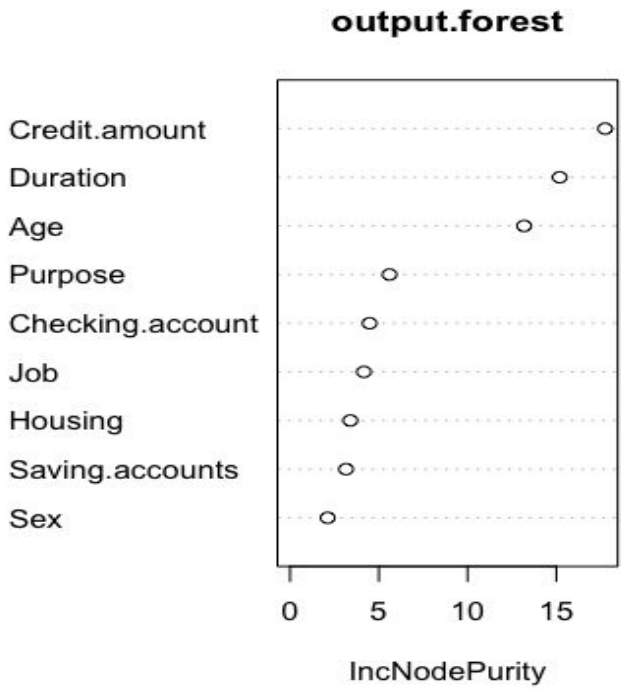
% Var explained: 7.97



Random forest (model visualization contd…)

This graph shows the relative importance of features in determining the results.

The Graph clearly indicates that the “credit.amount”, “Duration” and “Age” have higher importance than the other features in the data



XGI BOOST

```
xgb = XGBClassifier(eta=0.19, max_depth=8, n_estimators=150, subsample=0.8, colsample_bytree=1)
xgb.fit(X_train, y_train)
y_pred_xgb = xgb.predict(X_test)
print(accuracy_score(y_pred_xgb, y_test))
print(confusion_matrix(y_test, y_pred_xgb))
print(classification_report(y_test, y_pred_xgb))
```

```
0.75
[[123  18]
 [ 32  27]]
              precision    recall  f1-score   support

           0       0.79      0.87      0.83       141
           1       0.60      0.46      0.52        59

    accuracy                           0.75       200
   macro avg       0.70      0.66      0.68       200
weighted avg       0.74      0.75      0.74       200
```

Model Evaluation

The accuracy of the Logistic Regression =

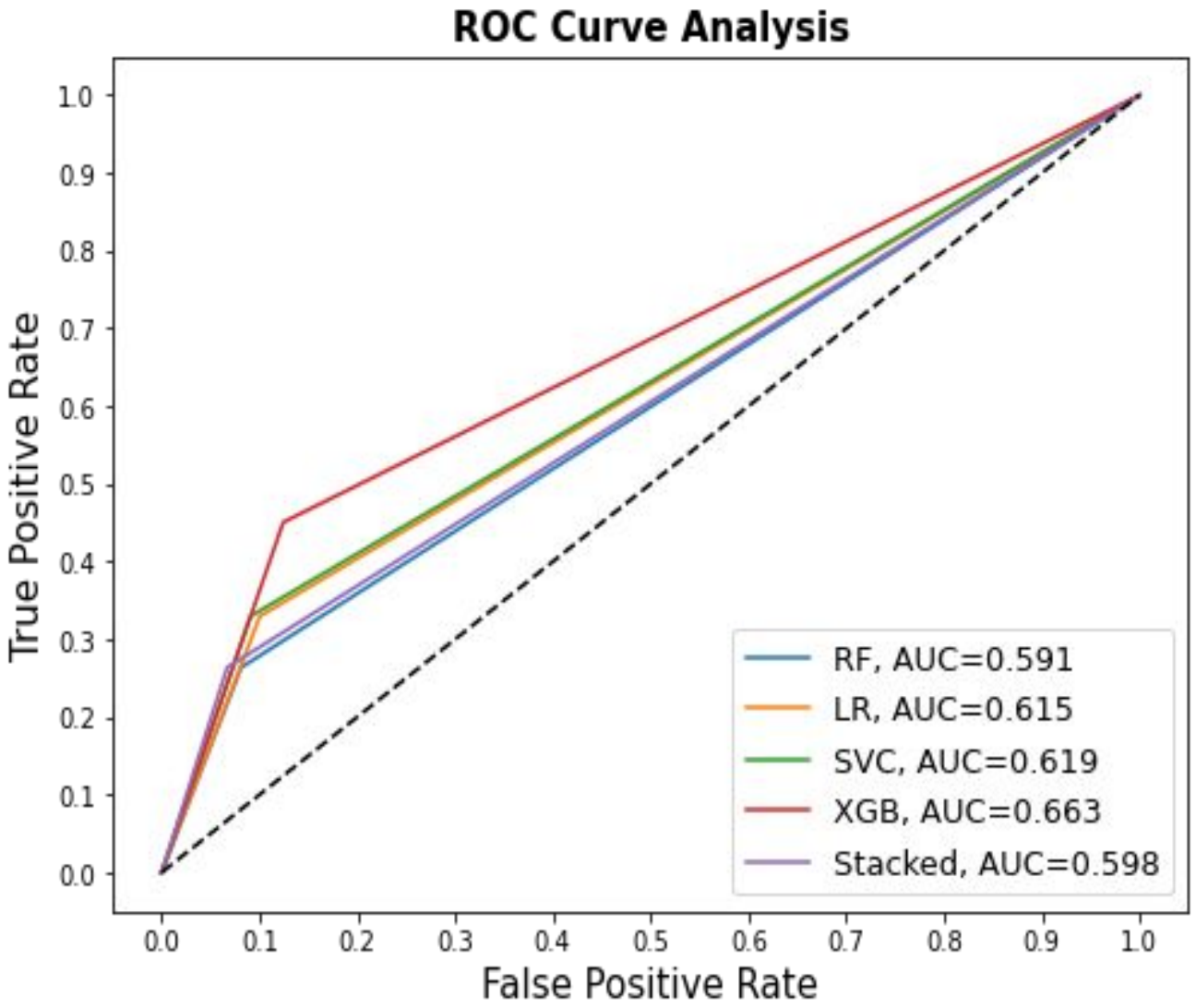
TP +TN/TP+FP+TN+FN = 0.75

Precision = TP/TP +FN

Recall = TN/TN+FP

F1-Score=TP+1/TP+½(FP+FN)=

0.46



Conclusion

XG Boost is the best model among the 4 models. The AUC value (0.663) is the highest as compared to other models tried on the dataset.


Thank
you