# CUSTOMER SEGMENTATION AND CREDIT RISK ANALYSIS USING COBRA

A project report submitted
for the course

**MA 691 Advanced Statistical Algorithms**
Asst. Prof. Arabin Kumar Dey

by

**AB Satyaprakash** (180123062)
**Kartikay Goel** (180101033)
**Samiksha Sachdeva** (180123040)
**Jatin Dhingra** (180123060)
**Himanshu Yadav** (180123016)


*to the*

**DEPARTMENT OF MATHEMATICS**
**INDIAN INSTITUE OF TECHNOLOGY GUWAHATI**
**GUWAHATI - 781039, INDIA**

*November 2021*

# ABSTRACT

Credit risk analysis is a form of analysis performed by a credit analyst on potential borrowers to determine their ability to meet debt obligations. The main goal of credit analysis is to determine the creditworthiness of potential borrowers and their ability to honor their debt obligations. This is a very important area of study because it's one of the most common situations that financial institutions like banks encounter on a daily basis. In this work, predict the credit risk of borrowers on two very famous datasets namely - the **German Credit** dataset and the **Australian Credit** dataset. We use **COmbined Regression Alternative** (**COBRA**) and get an improved accuracy of **70.1%** and **85.3%** for these datasets respectively. Our work can also be extended to other datasets and put to common use to accurately segment customers based on credit risk by financial institutions.

We have presented our work in the form of 2 notebooks - German Credit Risk Analysis and Australian Credit Risk Analysis. A web application has been deployed on Heroku based on the German dataset here. An API has also been deployed on Heroku for both Australian and German datasets for anyone to explore and make new applications on.

# 1. INTRODUCTION: Credit Risk Analysis

## 1.1. Credit Risk

Credit Risk is the probable risk of loss resulting from a borrower's failure to repay a loan or meet contractual obligations. If a company offers credit to its client, then there is a risk that its clients may not pay their invoices.

## 1.2. Types of Credit Risk

- **Good Risk:** An investment that one believes is likely to be profitable. The term most often refers to a loan made to a creditworthy person or company. Good risks are considered exceptionally likely to be repaid.

- **Bad Risk:** A loan that is unlikely to be repaid because of bad credit history, insufficient income, or some other reason. A bad risk increases the risk to the lender and the likelihood of default on the part of the borrower.

## 1.3. Objective

Based on the attributes, classify a person as a good or bad credit risk.

## 1.4 Practical Applications

Credit Risk analysis is something financial institutions encounter on a daily basis. Loan approval, credit-card applications, customer classification, etc. are essential for their functioning and thus our work is of practical importance.

## 2. **COBRA:** A Combined Regression Alternative

COBRA which stands for Combined Regression is a method used to combine multiple weak learners. Given a set of preliminary estimators r1, .., rM, the idea behind this combining method is a unanimity concept. It creates a prediction mapping for each weak learner on the training data. These are then used while predicting the test data to find existing data points that are close to the considered point. The prediction corresponding to these data points is used to generate the final prediction by taking the help of some summary metric (mean in our implementations).

## 3. **Dataset Description:** An overview of the 2 datasets

In this section, we will take a look at the 2 famous datasets namely the German and the Australian Credit Risk analysis datasets. Both the dataset have been taken from the UCI Machine Learning Reposiotry.

### 3.1. German Credit Risk Dataset

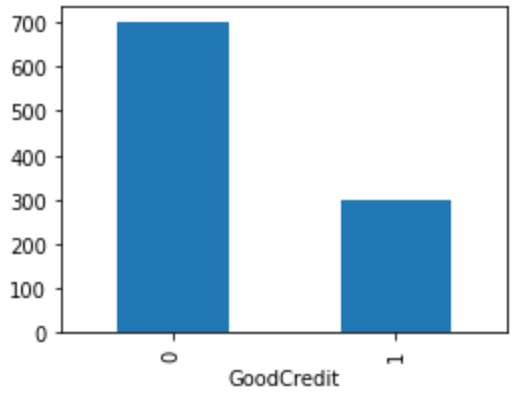
The German Credit Risk dataset classifies people described by a set of attributes as good or bad credit risks. The dataset was prepared by Professor Dr. Hans Hofmann, from the Institut f"ur Statistik und "Okonometrie Universit".

The dataset contains **1000 entries** with **20 independent variables** (**7 numerical, 13 categorical**) and **1 target variable**. Each person is classified as a good or bad credit risk according to the set of attributes.

**The attributes in this dataset are** - Status of the existing checking account, Duration in months, Credit history, Purpose, Credit amount,

Savings account/bonds, Present employment, Personal status and sex, Other debtors/guarantors, Present residence since, Property, Age in years, Other installment plans, Housing, Number of existing credits at this bank, Job, Number of people being liable to provide maintenance for, Telephone, and Foreign worker.

Fig. German dataset count for good vs bad credit score.




Since all attributes are clear for this dataset, it makes it an ideal choice for a web application that we have made using this.

## 3.2. Australian Credit Approval Dataset

The Australian Credit Approval dataset concerns credit card applications. Each person's credit card application has been classified as accepted or rejected. The dataset was prepared by anonymous sources.

The dataset contains 690 entries with 14 independent variables (6 numerical, 8 categorical) and 1 target variable. The labels for attributes and values are modified to protect data confidentiality. This dataset is interesting because there is a good mix of attributes -- continuous, nominal with small numbers of values, and nominal with larger numbers of values.

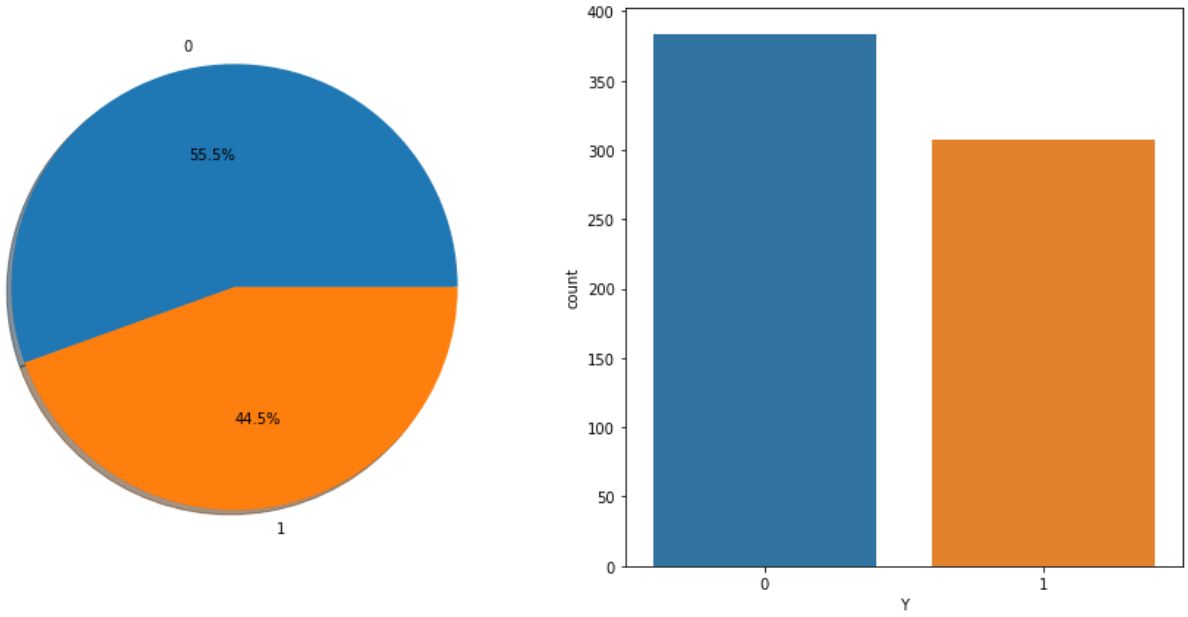
There are also a few missing values.

**37 cases (5%) had one or more missing values**. The missing values from particular attributes were:

**A1**: 12 ; **A2**: 12 ; **A4**: 6 ; **A5**: 6 ; **A6**: 9 ; **A7**: 9 ; **A14**: 13

For **categorical** attributes, these were replaced by the **mode** of the attributes and for **continuous** attributes, these were replaced by the **mean** of the attributes.

Fig. Australian dataset count for accepted vs rejected credit card applications.



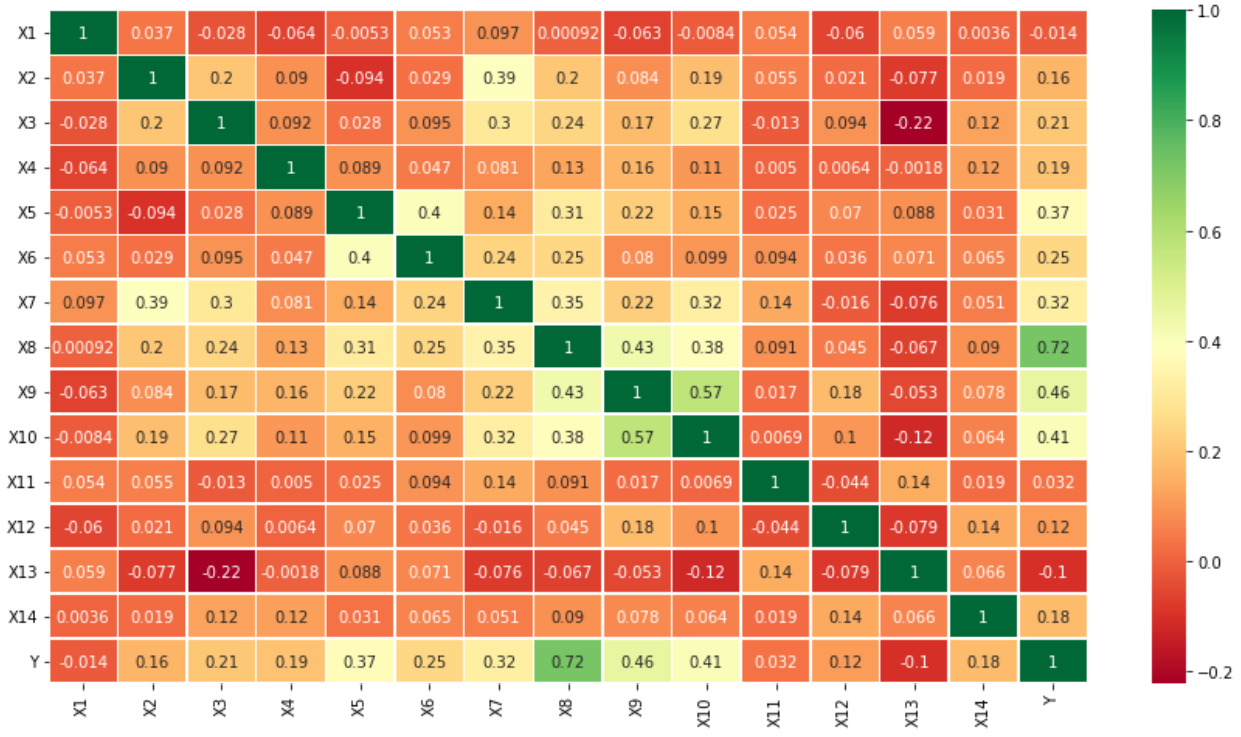
# 4. Methodology:

In this section, we discuss the methodology of our work in detail. We will cover this over smaller subsections below.

## 4.1. Dataset Description

Our first step was understanding the dataset and attributes. We have already given a complete overview of the 2 datasets in section 3. Apart from exploration, we made sanity tests and checked for null or missing values in the dataset, and made appropriate changes to them.

Fig. Heatmap for depicting correlation in Australian dataset.

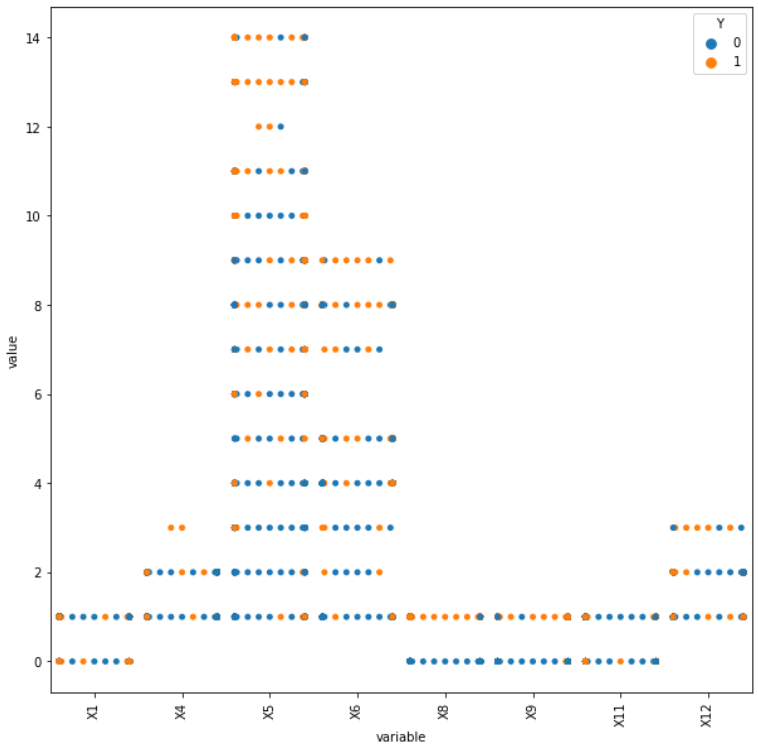



## 4.2. Exploratory Data Analysis (EDA)

Exploratory data analysis was important to discover patterns, spot

anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations. For example in the German dataset, we observe that more people who are foreign workers perform transactions in the bank as compared to native people. For this purpose, we made several plots - bar plots, heatmaps, swarm plots, and so on.

Fig. Swarmplot for categorical attributes in Australian dataset.




## 4.3. Feature Selection

Since we have a large number of attributes, and some of them might be correlated or even irrelevant to the classification, feature selection is an important step. For this, we perform the following 2 tests coupled with several other plots to come to a conclusion.

- **ANOVA Test** - Analysis of variance (ANOVA) is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples.
- **Chi-square Test** - Chi-squared test is used to determine whether there is a statistically significant difference between the expected frequencies and the observed frequencies in one or more categories of a contingency table.

### 4.4. Data Preprocessing for Machine Learning

In order to make data more suitable for models, some preprocessing is necessary. For instance, labels are replaced by numbers, and then the categorical attributes are split into dummies using the *get_dummies()* function in pandas. The continuous attributes are scaled and normalized using the *normalize()* and *MinMaxScaler()* functions in preprocessing module of scikit-learn.

### 4.5. Train-test split

For performing our model analysis we split the dataset into training and testing data using a **split ratio** of **0.3** and doing a random split.

- For the **German** dataset, this gives us **700** data points in the **train set** and **300** in the **test set**.
- For the **Australian** dataset, this gives us **483** data points in the **train set** and **207** in the **test set**.

This split has been taken because the initial number of data points is itself small, and to get some statistically significant results we need enough test data points.

## 4.6. COBRA

Multiple initial estimators of the regression function can be combined, instead of building a linear or convex optimized function over a selection of basic estimators. This can be used as a collective indicator of the proximity between training data and test results. This local-distance approach is fast and efficient, which performs asymptotically in the L2 sense as the best combination of the basic estimators in the collective. The increased accuracy of the COBRA strategy can help achieve better classifier performance on the Australian and German credit datasets.

### 4.6.1. Model overview

We divide the training dataset into two groups of equal size and then use distinct models to predict the indicator functions in our implementation. These models are trained on one of these halves and predict the indicator function on the other. Reference-training data are the prediction results that are saved for each of these models and used to predict outcomes.
For predicting the indicator outcome for an input from the test data, we do the following:

- We discover each machine's prediction on the test point under consideration, then cycle through the prediction table to locate near matches. For those machines, we make a note of these forecasts.

- Then we select the reference-training data points in which all machines make accurate predictions.

- We acquire the outcome corresponding to our necessary test data point by taking the mean of prediction probabilities corresponding to these data points, which we already have because this is training data.

### 4.6.2. Constituent Classification Models

We have used the following classification models (machines in COBRA):

1. **Decision Tree Classifier**: builds a Decision tree to classify samples into different classes.
2. **KNN Classifier**: finds k closest neighbors of a given test point and classifies it into the majority class using a distance measure.
3. **Naive Bayes Classifier**: classifier based on Bayes' Theorem and assumes that predictors are independent of one another.
4. **Support Vector Classifier**: creates a model that allocates new samples to one of two categories, making it a binary linear classifier that is non-probabilistic.
5. **Random Forest Classifier**: a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
6. **Gradient Boosting Classifier**: builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions.
7. **Logistic Regression**: a classification algorithm generates a decision boundary between two classes based on the training data
8. **MLP Classifier**: Multi-layer Perceptron classifier optimizes the log-loss function using LBFGS or stochastic gradient descent.

### 4.6.3. Hyper-paramter Tuning

Parameters that define the model architecture are referred to as hyperparameters and thus this process of searching for the ideal model architecture is referred to as hyperparameter tuning. In order to get the best accuracy and performance from a model, its hyperparameters are tuned using *GridSearchCV()* function.

#### 4.6.4. Metrics for evaluation

We have used the following metrics for evaluation -

1. **Accuracy**: The percentage of match between the predicted and the true values in the evaluating dataset. This has been done in several different methods namely - Randomized Repeated Holdout, Stratified Repeated Holdout, Randomized K-fold, Stratified K-fold, and Leave One Out.
2. **F1 score:** The F-score or F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive.
   The F1 score is the harmonic mean of precision and recall.

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

3. **ROC-AUC score:** It is the computed Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.

## 5. Results and Implementation:

The experiments were conducted in python with the help of libraries like Scikit-Learn, Scikit-Survival, Numpy, Pandas, and Matplotlib. Scikit-Learn was used to train classification models discussed in section 4.6.2.
Scikit-Learn preprocessing, stats, and feature selection libraries were also used. Furthermore, Sklearn metrics like accuracy, f1_score, confusion_matrix, classification_report, and roc_auc score were used for evaluating the model.

The experiment is executed on Google Colaboratory having the following configuration:
(i) Intel(R) Xeon(R) CPU @ 2.20GHz
(iii) 12.69 GB Memory / 108 GB Disk

### 5.1. German Dataset

The evaluation was done for each of the individual models and then for COBRA. The results for each model are tabulated below:

| Classification Model | Accuracy from sample data | Cross-Validation Accuracy | ROC AUC Score |
|---|---|---|---|
| Logistic Regression | 0.76 | 0.74 | 0.69 |
| Decision Tree | 0.67 | 0.7 | 0.59 |
| Random Tree | 0.72 | 0.74 | 0.63 |
| Adaboost Classifier | 0.73 | 0.74 | 0.66 |
| XGBoost Classifier | 0.72 | 0.75 | 0.64 |
| K-Nearest Neighbors | 0.74 | 0.7 | 0.67 |
| Support Vector | 0.73 | 0.75 | 0.65 |
| Naive Bayes | 0.72 | 0.7 | 0.71 |

We then train COBRA using the complete dataset. The following results are obtained for COBRA after cross_validation.

```
Accuracy values for 10-fold Cross-Validation:
 [0.8153605  0.55799534 0.68853333 0.76870219 0.71563636 0.68637933
 0.68221388 0.56773919 0.69152783 0.66655197]

Final Average Accuracy of the model: 0.6841
```

Plots were also made corresponding to each machine for roc_auc score and feature importance. Two of the plots (for others please see notebook) are shown here for **Decision Tree Classifier.**

Fig. roc_curve for Decision Tree Classifier.
Note the auc=0.59 matches to the one in the table above

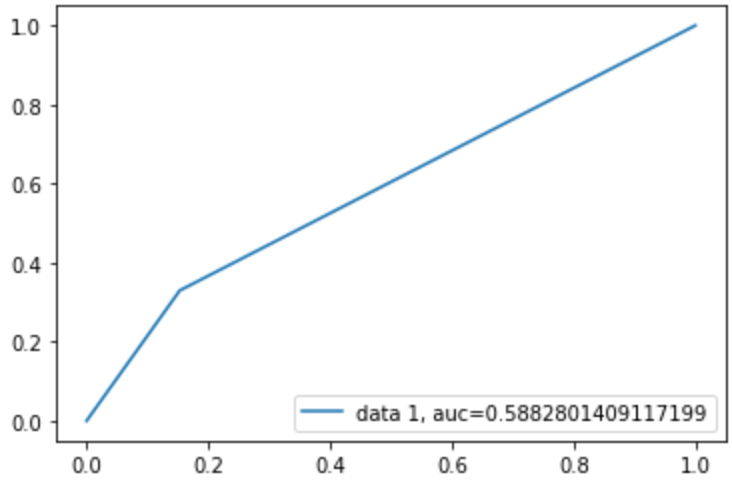
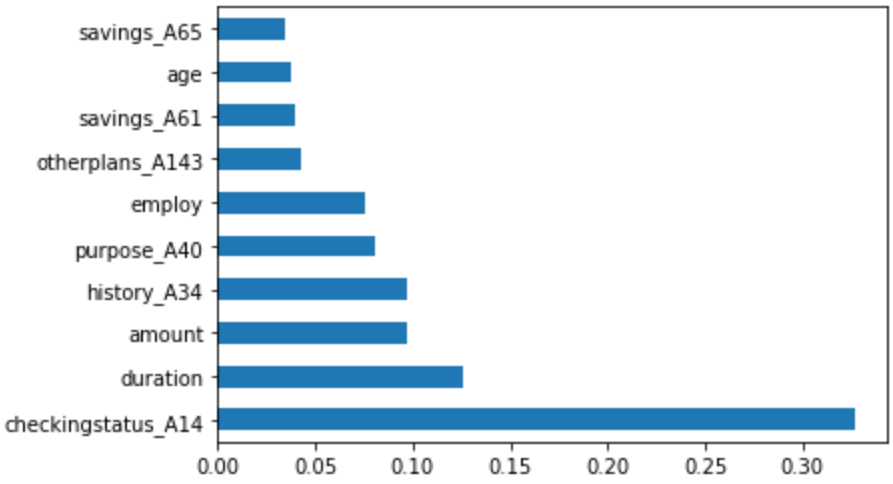



Fig. Feature importance plot for Decision Tree Classifier.



## 5.2. Australian Dataset

The evaluation was done for each of the individual models and then for COBRA. Accuracy was calculated using several different techniques of splitting the datasets and the results are as below: (images copied from notebooks)

1. Randomized Repeated Holdout

| | Accuracy |
|---|---|
| Gradient Boosting | 0.870614 |
| Logistic Regression | 0.870175 |
| CART | 0.865789 |
| KNN | 0.865789 |
| SVM | 0.860965 |
| MLP | 0.856579 |
| Random Forest | 0.854825 |
| Naive Bayes | 0.812281 |

2. Stratified Repeated Holdout

| | Accuracy |
|---|---|
| Logistic Regression | 0.856579 |
| SVM | 0.855263 |
| MLP | 0.850439 |
| KNN | 0.849123 |
| Random Forest | 0.848246 |
| Gradient Boosting | 0.843860 |
| CART | 0.803947 |
| Naive Bayes | 0.775439 |

3. Randomized k-Fold

| | Accuracy |
|---|---|
| Gradient Boosting | 0.870614 |
| Logistic Regression | 0.870175 |
| CART | 0.865789 |
| KNN | 0.865789 |
| SVM | 0.860965 |
| Random Forest | 0.859211 |
| MLP | 0.855702 |
| Naive Bayes | 0.812281 |

4. Stratified k-Fold

| | Accuracy |
|---|---|
| Gradient Boosting | 0.870614 |
| Logistic Regression | 0.870175 |
| CART | 0.865789 |
| KNN | 0.865789 |
| Random Forest | 0.862719 |
| SVM | 0.860965 |
| MLP | 0.853070 |
| Naive Bayes | 0.812281 |

5. Leave One Out

| | Accuracy |
|---|---|
| **CART** | 0.879710 |
| **Gradient Boosting** | 0.878261 |
| **KNN** | 0.866667 |
| **Logistic Regression** | 0.863768 |
| **Random Forest** | 0.856522 |
| **SVM** | 0.855072 |
| **MLP** | 0.852174 |
| **Naive Bayes** | 0.836232 |

We then train COBRA using the complete dataset. The following results are obtained for COBRA after cross_validation.

```
Accuracy values for 10-fold Cross-Validation:
 [0.91308001 0.91270903 0.8400908  0.86939959 0.80870518 0.91270903
 0.71872629 0.86945526 0.88361204 0.82476472]

Final Average Accuracy of the model: 0.8553
```

In a similar manner as the German dataset fashion plots were made for the Australian dataset too.

## 6. Deployment:

We have deployed a web application to showcase our work and also provide a platform for users to fill in their details and get predictions of the credit risk of a customer. The website features 2 sections -

a. Web portal for German credit risk dataset
b. Analysis for Australian credit risk dataset

For the German dataset web application, we ask the user for the following parameters:

1. Employment status
2. Age (in years)
3. Amount (in Deschutes Mark)
4. Duration (in months)
5. Status of existing checking account
6. Credit history
7. Purpose of loan
8. Savings
9. Sex and Marital Status

The prediction made is shown as a message below the web app and tells the user whether the customer has a good or bad credit risk.

For the Australian dataset page, we just show the analysis since making a web app for such a dataset is unrealistic.

We have also deployed an API for making predictions so that developers can make websites and applications using our API and make further use of the trained model.

## 7. References:

[1] COBRA: A combined regression strategy Gérard Biaua, Aurélie Fischer, Benjamin Guedj, James D. Malleye.
[2] UCI Machine Learning Repository: Statlog (German Credit Data) Data Set
[3] UCI Machine Learning Repository: Statlog (Australian Credit Approval) Data Set