# GERMAN CREDIT DATA ANALYSIS

Goyal, Prakhar

## Table of Contents

# 1. Logistic Regression on German Credit Data

## 1.1 Problem and Approach

The data for this problem is taken from UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)). This dataset classifies people described by a set of attributes as good or bad credit risks. The report examines the dataset through exploratory data analysis. A lot of variables in the dataset have been encoded and we have used techniques such as chi-squared test to assess correlation between these categorical variables and the response variable. We then move on to find the best model for classification customers into good or bad credit.

In order to determine the best logistic regression model, the team used different variable selection methods including backward elimination and LASSO. We then compare the model Area Under the Curve (AUC) values to identify and select the best model for the given data. For the model selected, we report the out-of-sample AUC values and the asymmetric classification rate.

## 1.2 Major Results

We have used logistic link to make our model.
We have used multiple stepwise variable selection techniques, Lasso and Classification Tree methods to identify the best model.
The final model was selected using **backward selection with AIC criteria**.
We observed that using Cross-validation fetches better AUC and less Miss classification rates than randomly selecting training and testing data.

## 1.3 Exploratory Data Analysis

### 1.3.1 About German Credit Dataset

```
data.shape
```

```
(1000, 21)
```

The dataset consists of 1000 observations across 21 variables. It was observed that there aren't any missing values in the dataset.

```
data.describe()
```

| | duration | creditamount | installmentrate | residencesince | age | existingcredits | peopleliable | classification |
|---|---|---|---|---|---|---|---|---|
| **count** | 1000.000000 | 1000.000000 | 1000.000000 | 1000.000000 | 1000.000000 | 1000.000000 | 1000.000000 | 1000.000000 |
| **mean** | 20.903000 | 3271.258000 | 2.973000 | 2.845000 | 35.546000 | 1.407000 | 1.155000 | 0.700000 |
| **std** | 12.058814 | 2822.736876 | 1.118715 | 1.103718 | 11.375469 | 0.577654 | 0.362086 | 0.458487 |
| **min** | 4.000000 | 250.000000 | 1.000000 | 1.000000 | 19.000000 | 1.000000 | 1.000000 | 0.000000 |
| **25%** | 12.000000 | 1365.500000 | 2.000000 | 2.000000 | 27.000000 | 1.000000 | 1.000000 | 0.000000 |
| **50%** | 18.000000 | 2319.500000 | 3.000000 | 3.000000 | 33.000000 | 1.000000 | 1.000000 | 1.000000 |
| **75%** | 24.000000 | 3972.250000 | 4.000000 | 4.000000 | 42.000000 | 2.000000 | 1.000000 | 1.000000 |
| **max** | 72.000000 | 18424.000000 | 4.000000 | 4.000000 | 75.000000 | 4.000000 | 2.000000 | 1.000000 |

We however see that the numerical variables present in the dataset are on different scales, so we would need to standardize them before we move on to build a model. We also notice that the numerical variables consist of outliers which we will need to treat before modelling.

We binarize the response variable (classification) and see that there are 700 instances of good loans and 300 instances of bad loans.

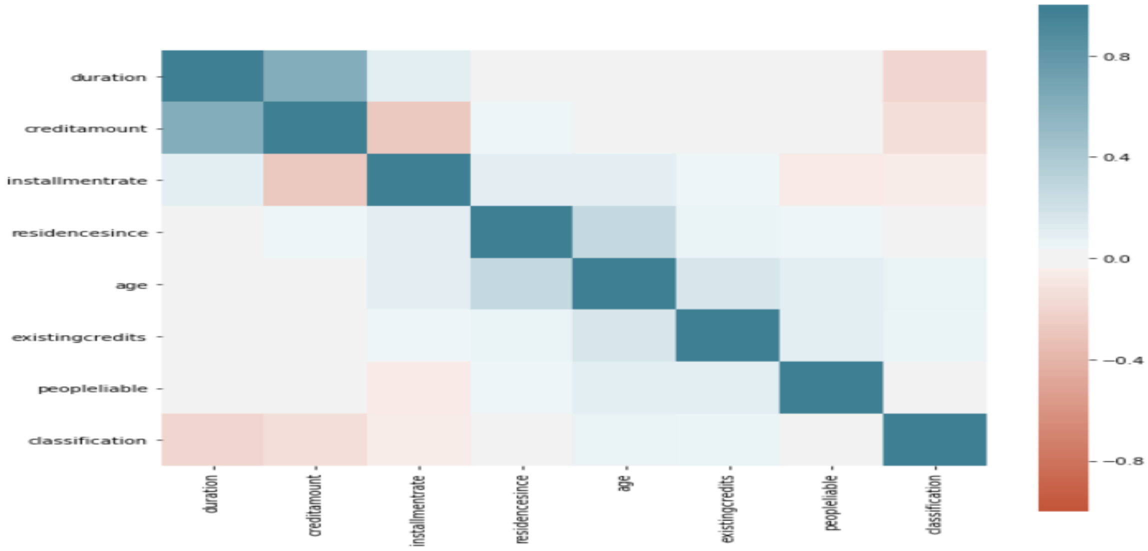
```
#Binarize the y output
data.classification.replace([1,2], [1,0], inplace=True)

data.classification.value_counts()
```

```
1    700
0    300
```

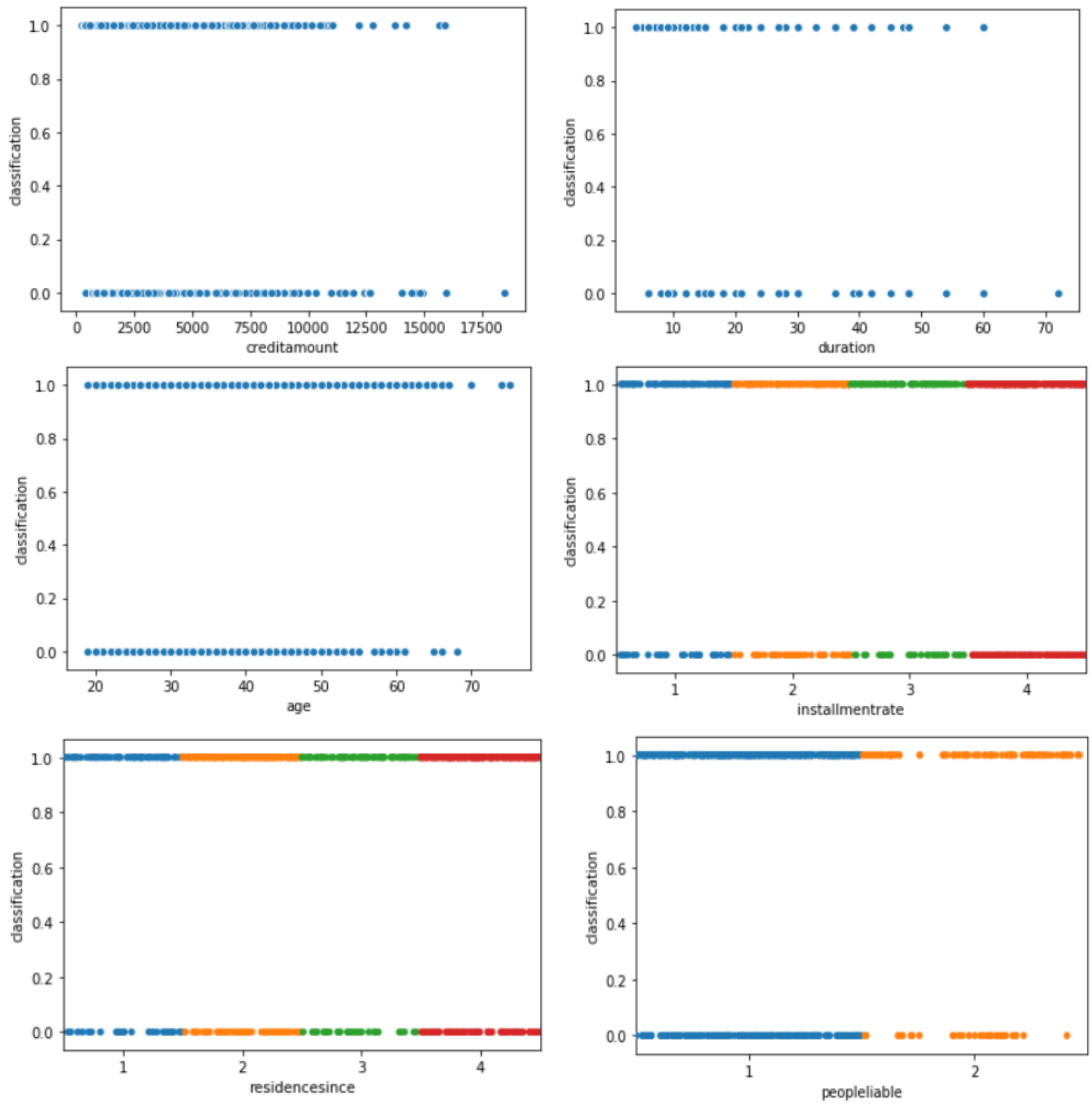
### 1.3.2 Exploratory Data Analysis

We have looked at pairwise correlations between the numerical variables and the response variable. For numerical variables we also look at jitter plots against the response variable. For categorical variables, we conduct chi-square test of categorical variables with the response variable.

#### *2.3.2.1 Pairwise Correlation of numerical variables with response variable*



### 1.3.2.2 Jitter plot of numerical variables with response variable




### 1.3.2.3 Chi-square tests

| Variable | Chi-squared test statistic | p-value | Related (Yes/No) |
|---|---|---|---|
| existingchecking | 97.13 | 2.2e-16 | Yes |
| credithistory | 52.80 | 9.34e-11 | Yes |
| purpose | 24.32 | 0.0038 | Yes |
| savings | 26.52 | 2.48e-05 | Yes |
| employmentsince | 22.45 | 0.00016 | Yes |
| statussex | 12.29 | 0.0064 | Yes |
| otherdebtors | 7.72 | 0.021 | Yes |
| property | 15.73 | 0.0012 | Yes |
| housing | 10.43 | 0.005 | No |
| job | 2.81 | 0.42 | No |
| foreignworker | 4.71 | 0.029 | Yes |

**Key Insights:**

- None of the numerical variables show a significant relationship with the response variable. However, it was observed that creditamount and duration are correlated to some extent
- Results of chi-squared test are summarized in a table above

## 1.4 Logistic Regression Model Building

### 1.4.1 Finding the appropriate link function

| | Name.of.the.Link | Deviance | AIC | BIC |
|---|---|---|---|---|
| 1 | Logit | 592.6301 | 690.6301 | 913.6330 |
| 2 | Probit | 592.6522 | 690.6522 | 913.6552 |
| 3 | Cloglog | 592.9673 | 690.9673 | 913.9702 |

*Figure 1:Comparison of various link families*

**Key Insights:**
We can observe that Logit has minimum Deviance, AIC and BIC values amongst the three Links. We will stick with logit because of easy of interpretability as well.

### 1.4.2 Variable Selection using AIC, BIC and LASSO

```
> Comparison_table
```

| | Model | AUC | MissClassification.Rate | Deviance | No_of_Variables |
|---|---|---|---|---|---|
| 1 | AIC | 0.8537073 | 0.3114286 | 606.4441 | 13 |
| 2 | BIC | 0.7896877 | 0.4700000 | 704.7924 | 4 |
| 3 | Lasso | 0.8227412 | 0.3600000 | NA | 11 |

*Figure 2:Comparison of values from different Models*

**Key Insights:**
We observe that AIC has less deviance, MR rate and high AUC. Even though AIC model is a bit complex than the Lasso Model in terms of number of predictor variables, the AUC and MR rate is significantly bad when compared to AIC.
Hence, we select AIC model to be our final model.

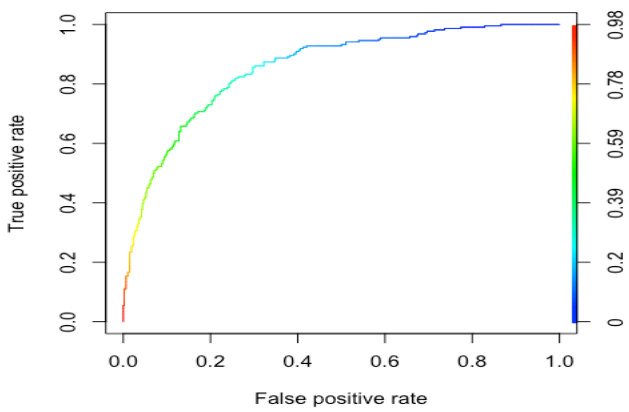



*Figure 3:ROC Curve AIC Model*

### 1.4.3 Final Model Testing

| | Model | AUC | MissClassification.Rate |
|---|---|---|---|
| 1 | Final_Model_Test | 0.7411065 | 0.4033333 |
| 2 | Final_Model_Train | 0.8537073 | 0.3114286 |

*Figure 4:AUC and MR Comparison of Testing vs Training Data*

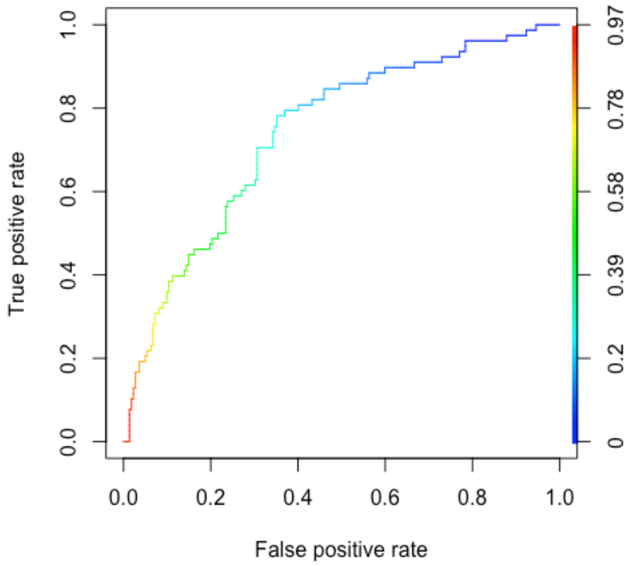



*Figure 5:ROC Curve on Testing Data*

**Key Insights:**

The AUC and Miss classification rate on testing dataset has decreased as expected.

### 1.4.4 Optimal Cut-off Probability

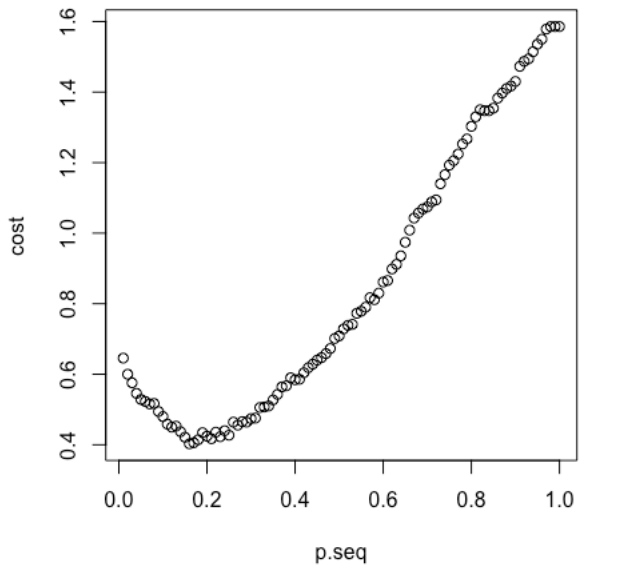



*Figure 6:Cost vs probability*

```
> optimal.pcut.glm.a
[1] 0.16
```

*Figure 7:Optimal Cut off Probability*

**Key Insights:**

The optimal Cut-off probability(also the minimum cost) is 0.16 for this model, which is quite close to the cut off probability weights(1:5) as expected.

### 1.4.5 3-fold Cross Validation

Adjusted cross-validation estimate→0.4826

```
                         Model       AUC MissClassification.Rate
1 Final_Model_prediction_Test 0.7411065                0.4033333
2                          CV 0.8202333                0.3400000
```

*Figure 8:AUC and MR Rate after Cross Validation*

**Key Insights:**

The above table states that AUC and MR are different for (iii) & (v). This proves that even though the model equation is same, by using the Cross-validation technique to divide the dataset into testing and training has yielded in a much better AUC and MR rates in comparison to the original model.

### 1.4.6 Using Classification Tree for Variable Selection

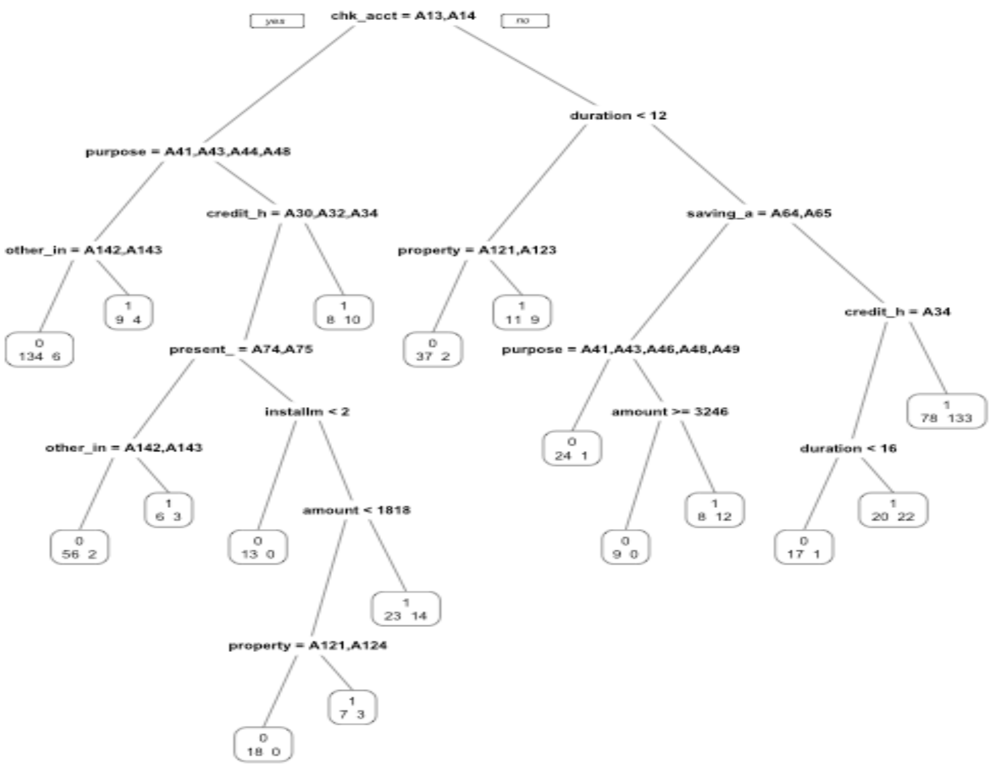



*Figure 9:Tree Map*

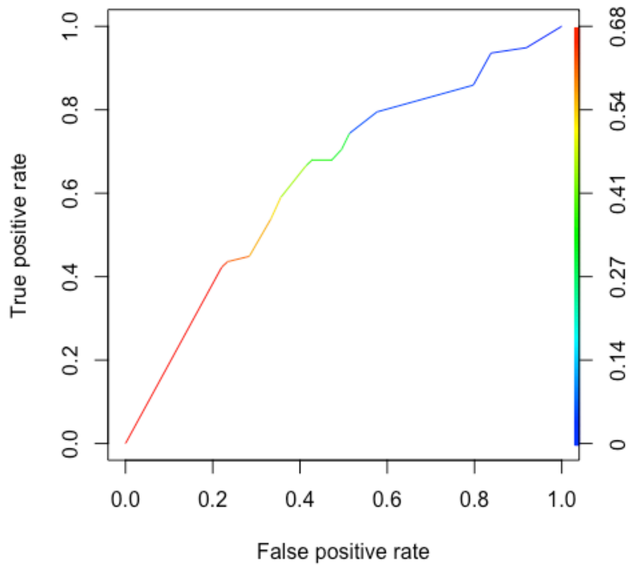



*Figure 10:ROC Curve for Tree Model*

```
> Comparison_table.final
                        Model       AUC MissClassification.Rate
1 Final_Model_prediction_Test 0.7411065               0.4033333
2                          CV 0.8202333               0.3400000
3   Classification Tree Model 0.6395241               0.4433333
```

*Figure 11:Comparison among the Models vs Tree*

**Key Insights:**

The above table states that AUC has decreased and MR has increased when we use the Classification Tree model.

### 1.4.6 Changing share of Training/Test dataset to 90/10

| | Training/Test :70/30 | | Training/Test :90/10 | |
|---|---|---|---|---|
| | AUC | Miss Classification Rate | AUC | Miss Classification Rate |
| Final_Model_AIC | 0.7411065 | 0.4033333 | 0.7380457 | 0.37 |
| Final_Model_CV | 0.8202333 | 0.3400000 | 0.8202333 | 0.34 |
| Classification Tree Model | 0.6395241 | 0.4433333 | 0.6658004 | 0.47 |

*Figure 12:Comparision*

**Key Insights:**

The above table shows that when we increased our training set to 90% of the total dataset, CV model is producing the same results as expected.

The AUC values should increase as the training dataset is increasing to 90% and would have more datapoints, and similarly Miss Classification rate should ideally increase as the testing data doesn’t have a large enough dataset.