ASSESSMENT 4 – GROUP REPORT

Cyrus Kwan: 45200165

Naveed Huda: 45185697

Yan Tung Lo: 44500513

Weixin Zou: 44166524

CONTENTS

BACKGROUND:

Determining the credit worthiness of an individual is crucial to maximise profit and minimise risk for a bank. Rejecting a customer with strong creditworthiness will result in the bank losing business whereas giving a loan to someone with poor creditworthiness will cost the bank money.

To provide a solution to assist the bank regarding who is given a loan and in turn minimise the risk and maximise profit, the bank has hired us to produce a report outlining the most efficient method of determining an individual's creditworthiness. This report will summarise the results of applying predictive and clustering analysis models to The German Credit data set with the aim of selecting a model to predict creditworthiness.

DATA PREPARATION: COMPLETED USING R SOFTWARE PACKAGE

SUMMARY OF GERMAN CREDIT DATA

```
> anyNA(credit_train)
[1] FALSE

> anyNA(credit_test)
[1] FALSE
```

Output 1: anyNA() returns no null values meaning that the data is balanced and no imputation is required.

```
> str(credit_train)
'data.frame':   800 obs. of  22 variables:
$ ID                                 : int  151 152 153 154 155 156
157 158 159 160 ...
$ Account.Balance                    : int  1 1 1 1 3 2 3 4 1 4 ...
$ Duration.of.Credit..month.         : int  48 21 18 15 4 21 10 6 9
24 ...
$ Payment.Status.of.Previous.Credit: int  2 4 2 2 2 4 2 3 4 2 ...
$ Purpose                            : int  4 0 3 9 0 9 0 3 5 3 ...
$ Credit.Amount                      : int  3051 571 1345 806 1494
3652 1240 935 1288 3235 ...
$ Value.Savings.Stocks               : int  1 1 1 1 5 1 2 1 2 3 ...
$ Length.of.current.employment       : int  3 5 3 3 2 4 5 3 5 5 ...
$ Instalment.per.cent                : int  3 4 4 4 1 2 1 3 3 3 ...
$ Sex...Marital.Status               : int  3 3 4 2 3 3 2 2 3 1 ...
$ Guarantors                         : int  1 1 1 1 1 1 1 1 3 1 ...
$ Duration.in.Current.address        : int  4 4 3 4 2 3 4 2 4 2 ...
$ Most.valuable.available.asset      : int  3 1 1 2 1 2 4 1 1 3 ...
$ Age..years.                        : int  54 65 26 22 29 27 48 24
48 36 ...
$ Concurrent.Credits                 : int  3 3 3 3 3 3 3 3 3 3 ...
$ Type.of.apartment                  : int  2 2 2 2 2 2 3 2 2 2 ...
$ No.of.Credits.at.this.Bank         : int  1 2 1 1 1 2 1 1 2 1 ...
$ Occupation                         : int  3 3 3 2 2 3 2 3 3 4 ...
$ No.of.dependents                   : int  1 1 1 1 2 1 2 1 2 1 ...
$ Telephone                          : int  1 1 1 1 1 1 1 1 1 2 ...
$ Foreign.Worker                     : int  1 1 1 1 2 1 1 1 2 1 ...
$ Creditability                      : int  0 1 0 1 1 1 0 1 1 1 ...
```

Output 2: str() shows that raw data is coded as numerical values according to the data dictionary which is difficult to interpret without constantly referring to the data dictionary.

```
> str(credit_test)
'data.frame':   150 obs. of  21 variables:
$ ID                                 : int  1 2 3 4 5 6 7 8 9 10 ...
$ Account.Balance                    : int  1 4 4 2 4 2 4 1 4 4 ...
$ Duration.of.Credit..month.         : int  24 15 28 24 24 7 22 24 36
12 ...
$ Payment.Status.of.Previous.Credit: int  2 2 4 2 4 2 2 4 2 2 ...
$ Purpose                            : int  2 1 3 1 3 3 3 3 2 0 ...
$ Credit.Amount                      : int  2359 4657 2743 12579 1516
2415 2675 1231 3349 1101 ...
$ Value.Savings.Stocks               : int  2 1 1 1 4 1 3 4 1 1 ...
$ Length.of.current.employment       : int  1 3 5 5 3 3 5 5 3 3 ...
$ Instalment.per.cent                : int  1 3 4 4 4 3 3 4 4 3 ...
$ Sex...Marital.Status               : int  1 3 3 2 2 3 3 2 2 4 ...
$ Guarantors                         : int  1 1 1 1 1 3 1 1 1 1 ...
$ Duration.in.Current.address        : int  1 2 2 2 1 2 4 4 2 2 ...
$ Most.valuable.available.asset      : int  2 3 3 4 1 1 3 2 3 1 ...
$ Age..years.                        : int  33 30 29 44 43 34 40 57 28
27 ...
$ Concurrent.Credits                 : int  3 3 3 3 3 3 3 3 3 3 ...
$ Type.of.apartment                  : int  2 2 2 3 2 2 2 1 2 2 ...
$ No.of.Credits.at.this.Bank         : int  1 1 2 1 2 1 1 2 1 2 ...
$ Occupation                         : int  3 3 3 4 2 3 3 4 4 3 ...
$ No.of.dependents                   : int  1 1 1 1 1 1 1 1 1 1 ...
$ Telephone                          : int  1 2 1 2 1 1 1 2 2 2 ...
$ Foreign.Worker                     : int  1 1 1 1 1 1 1 1 1 1 ...
```

Output 3: Shows all categorical values coded as numerical for both training and test sets

DATA CLEANING –

RECODING ATTRIBUTES AS CHARACTERS

As shown in output 2, many categorical attributes were difficult to initially interpret without the data dictionary. Hence, all categorical values were recoded as character values corresponding to the data dictionary. In contrast, although coding attributes as dummy variables may have produced better predictive and clustering models the number of attributes would have been unreasonable to manage (70 total).

ADDING FEATURES

Looking at the data dictionary, the attribute “Sex & Marital Status” has inconsistent categories between sexes. Thusly, the attribute was split into “Sex” and “Marital Status”.

DROPPING ATTRIBUTES

As mentioned above, producing “Sex” and “Marital Status” features caused the initial “Sex & Marital Status” attribute to become redundant and was subsequently dropped from the dataset.

RESULTING DATASETS

```
> str(code_train)
'data.frame':   800 obs. of  23 variables:
$ ID                                : int  151 152 153 154 155 156 157 158 159 160 ...
$ Account.Balance                   : chr  "< 0 EU" "< 0 EU" "< 0 EU" "< 0 EU" ...
$ Duration.of.Credit..month.        : int  48 21 18 15 4 21 10 6 9 24 ...
$ Payment.Status.of.Previous.Credit: chr  "Existing Paid" "Critical Account" "Existing
Paid" "Existing Paid" ...
$ Purpose                           : chr  "Appliances" "Car(New)" "Radio/Television"
"Business" ...
$ Credit.Amount                     : int  3051 571 1345 806 1494 3652 1240 935 1288
3235 ...
$ Value.Savings.Stocks              : chr  "< 100 EU" "< 100 EU" "< 100 EU" "< 100 EU" ...
$ Length.of.current.employment      : chr  "1 <= ... < 4 Years" ">= 7 Years" "1 <= ... < 4
Years" "1 <= ... < 4 Years" ...
$ Instalment.per.cent               : chr  "25% - 35%" "> 35%" "> 35%" "> 35%" ...
$ Guarantors                        : chr  "None" "None" "None" "None" ...
$ Duration.in.Current.address       : chr  "> 7 Years" "> 7 Years" "4 - 7 Years" "> 7
Years" ...
$ Most.valuable.available.asset     : chr  "Car or Other" "Real Estate" "Real Estate"
"Savings Agreement/Life Insurance" ...
$ Age..years.                       : int  54 65 26 22 29 27 48 24 48 36 ...
$ Concurrent.Credits                : chr  "None" "None" "None" "None" ...
$ Type.of.apartment                 : chr  "Own" "Own" "Own" "Own" ...
$ No.of.Credits.at.this.Bank        : chr  "1" "2 or 3" "1" "1" ...
$ Occupation                        : chr  "Skilled Official" "Skilled Official" "Skilled
Official" "Unskilled Resident" ...
$ No.of.dependents                  : chr  "< 3" "< 3" "< 3" "< 3" ...
$ Telephone                         : chr  "None" "None" "None" "None" ...
$ Foreign.Worker                    : chr  "Yes" "Yes" "Yes" "Yes" ...
$ Creditability                     : chr  "Bad" "Good" "Bad" "Good" ...
$ Sex                               : chr  "Male" "Male" "Male" "Female" ...
$ Marital.Status                    : chr  "Single" "Single" "Divorced/Separated/Married"
"Divorced/Separated/Married" ...
```

Output 4: Cleaned training set

```
> str(code_test)
'data.frame':   150 obs. of  22 variables:
$ ID                                : int  1 2 3 4 5 6 7 8 9 10 ...
$ Account.Balance                   : chr  "< 0 EU" "No Account" "No Account" "0 <= ... <
200 EU" ...
$ Duration.of.Credit..month.        : int  24 15 28 24 24 7 22 24 36 12 ...
$ Payment.Status.of.Previous.Credit: chr  "Existing Paid" "Existing Paid" "Critical
Account" "Existing Paid" ...
$ Purpose                           : chr  "Furniture/Equipment" "Car(Used)"
"Radio/Television" "Car(Used)" ...
$ Credit.Amount                     : int  2359 4657 2743 12579 1516 2415 2675 1231 3349
1101 ...
$ Value.Savings.Stocks              : chr  "100 <= ... < 500 EU" "< 100 EU" "< 100 EU" "<
100 EU" ...
$ Length.of.current.employment      : chr  "Unemployed" "1 <= ... < 4 Years" ">= 7 Years"
">= 7 Years" ...
$ Instalment.per.cent               : chr  "< 20%" "25% - 35%" "> 35%" "> 35%" ...
$ Guarantors                        : chr  "None" "None" "None" "None" ...
$ Duration.in.Current.address       : chr  "< 1 Year" "1 - 4 Years" "1 - 4 Years" "1 - 4
Years" ...
$ Most.valuable.available.asset     : chr  "Savings Agreement/Life Insurance" "Car or
Other" "Car or Other" "Unknown" ...
$ Age..years.                       : int  33 30 29 44 43 34 40 57 28 27 ...
$ Concurrent.Credits                : chr  "None" "None" "None" "None" ...
$ Type.of.apartment                 : chr  "Own" "Own" "Own" "For Free" ...
$ No.of.Credits.at.this.Bank        : chr  "1" "1" "2 or 3" "1" ...
$ Occupation                        : chr  "Skilled Official" "Skilled Official" "Skilled
Official" "Highly Qualified" ...
$ No.of.dependents                  : chr  "< 3" "< 3" "< 3" "< 3" ...
$ Telephone                         : chr  "None" "Registered" "None" "Registered" ...
$ Foreign.Worker                    : chr  "Yes" "Yes" "Yes" "Yes" ...
$ Sex                               : chr  "Male" "Male" "Male" "Female" ...
$ Marital.Status                    : chr  "Divorced/Separated/Married" "Single" "Single"
"Divorced/Separated/Married" ...
```

Output 5: Cleaned testing set

DATA ANALYSIS: COMPLETED USING ORANGE

UNIVARIATE ANALYSIS

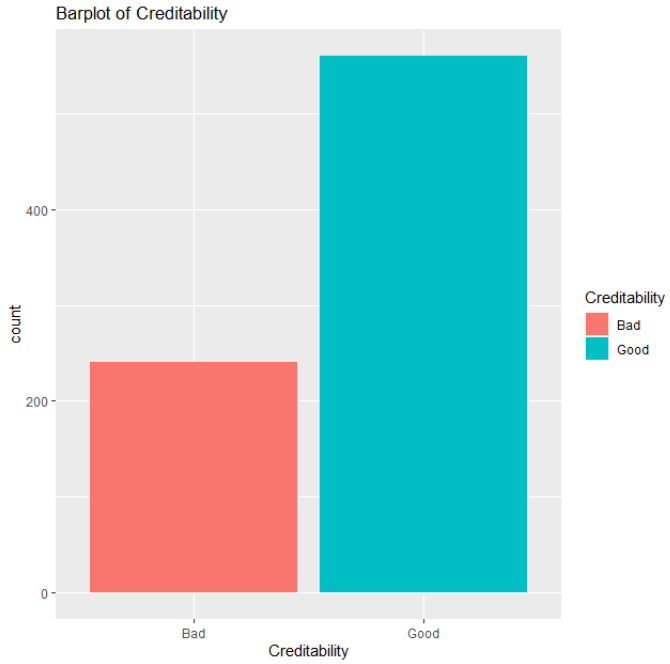

Figure 1: The majority of creditors are good.

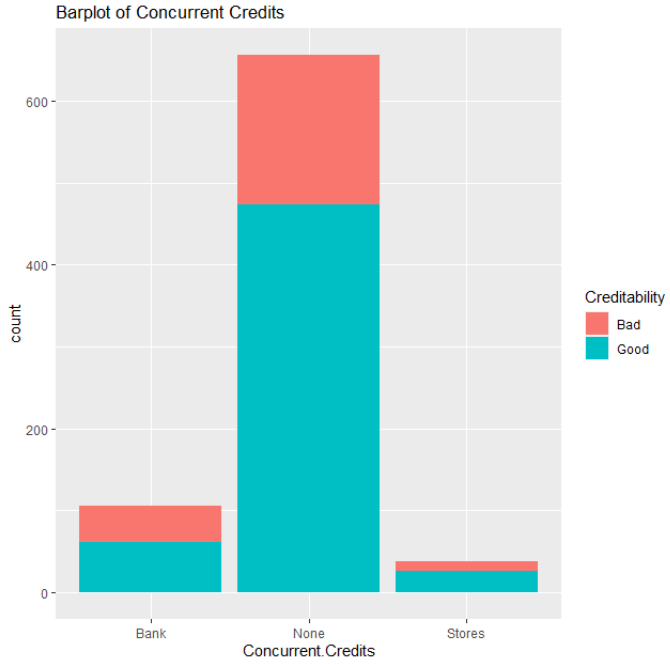

Figure 2: Creditors with only a single credit are more likely to be good creditors compared to those who have concurrent creditors in banks and/or stores. Additionally, they are the largest user base of creditors at this bank.

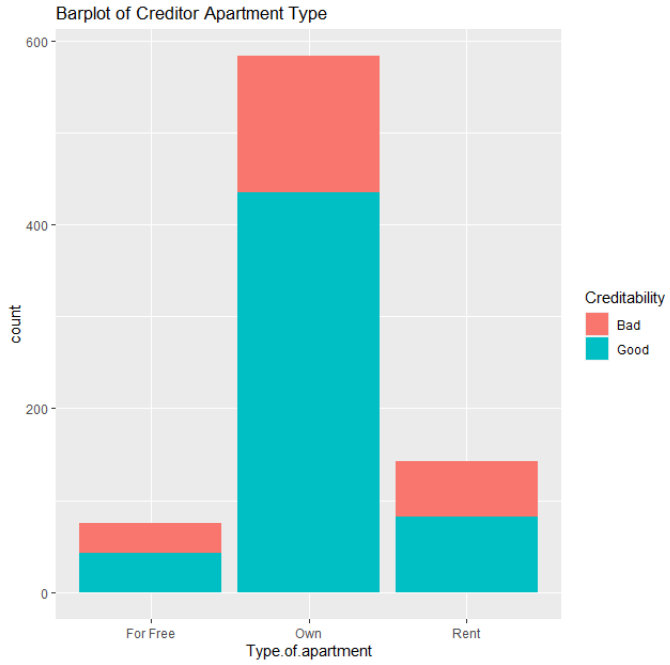

Figure 3: Proportionally, creditors who own their apartment are more likely to be good creditors compared to people who are renting or living for free. Creditors who own their apartment make up most users.

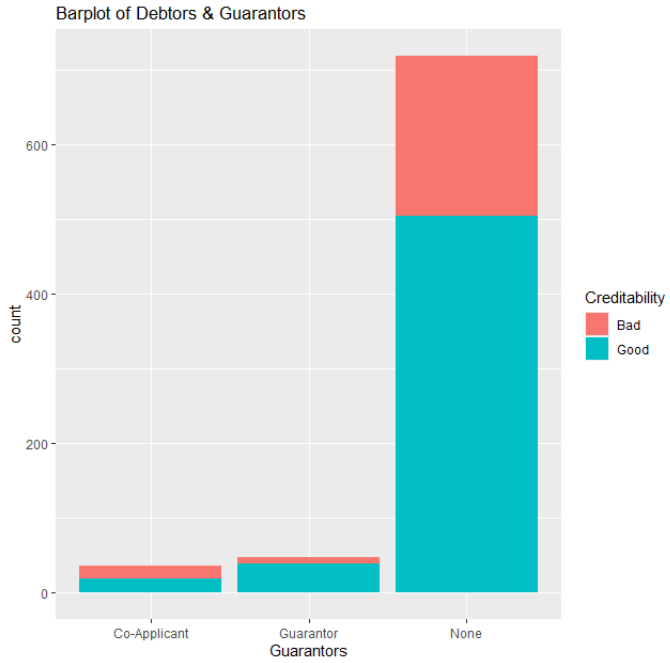

Figure 4: The largest user base is comprised of creditors without debtors or guarantors. Despite this, users with guarantors are more likely to be good creditors.

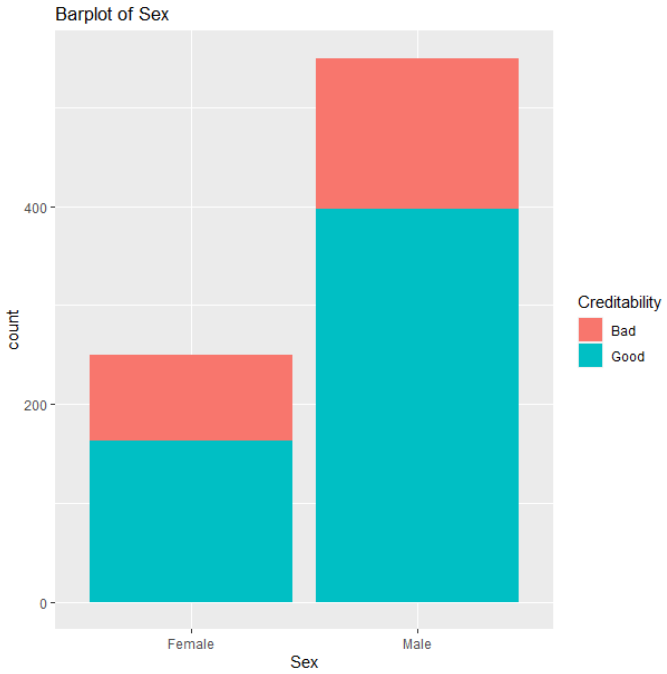

Figure 5: There are over 2 times the number of male creditors than there are female creditors. Additionally, there does not appear to be any significant difference between the proportion of good and bad creditors with regards to sex.

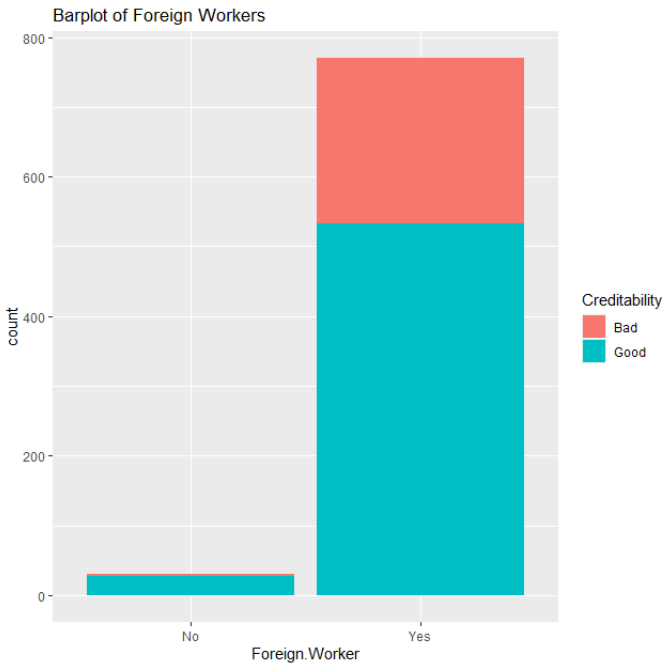

Figure 6: Most creditors are foreign workers. Advertising efforts for the bank could be directed overseas.

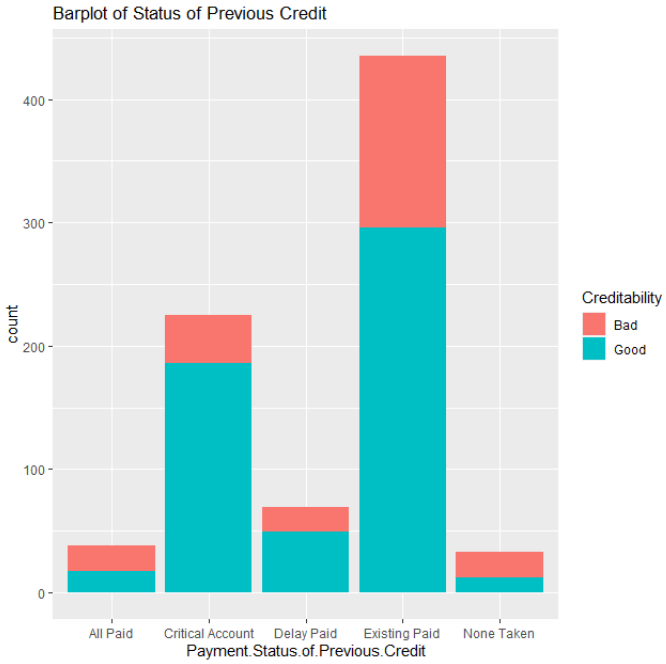

Figure 7: Creditors who previously had critical accounts are more likely to presently be a good creditor. Most creditors consistently take out more than 2 credit plans indicating that the bank has a loyal customer base.

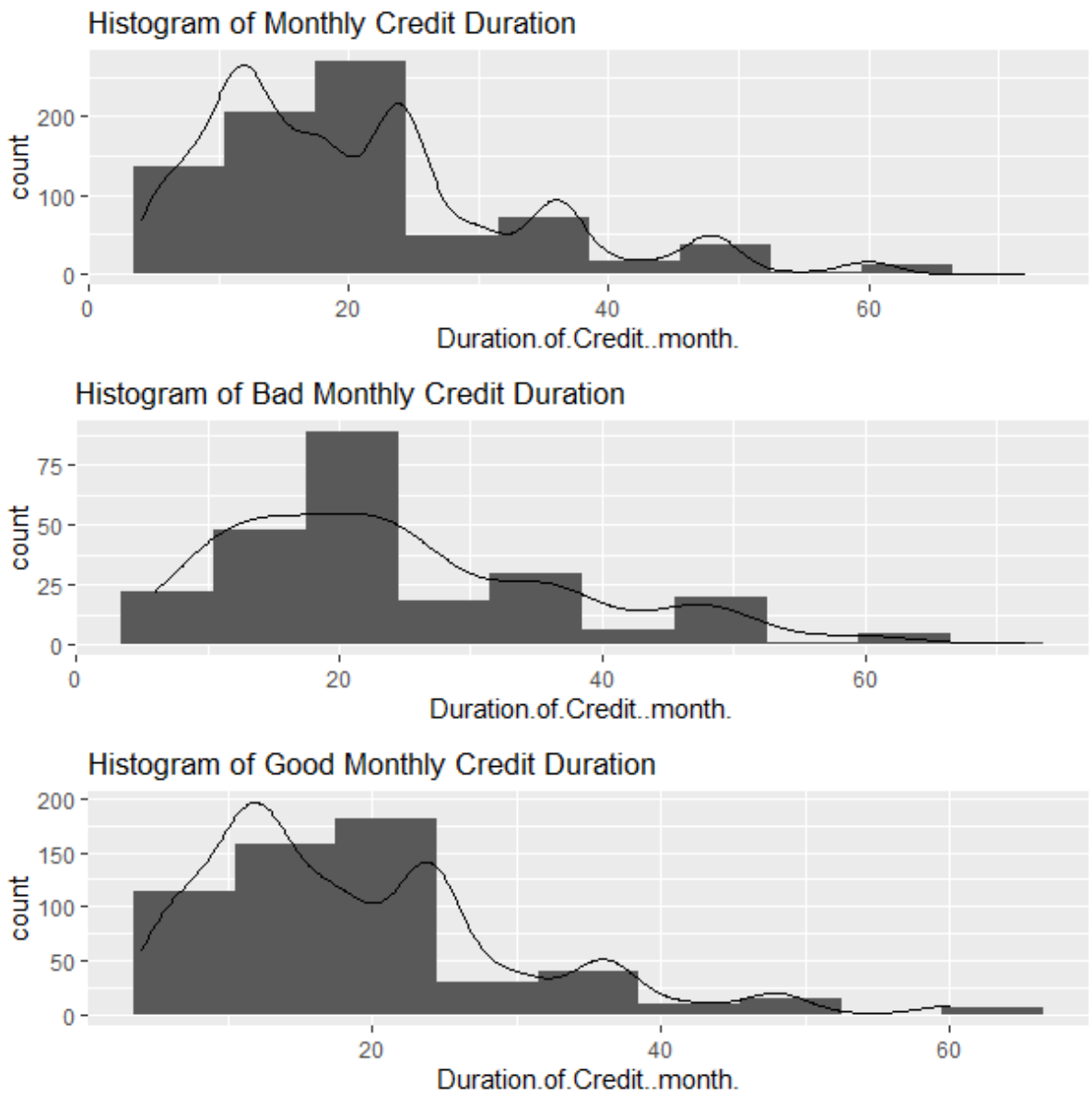

Figure 8: The distribution of bad creditors compared to good creditors shows that there are a larger proportion of good creditors where duration is below 20 months. Hence, a strategy to increase the amount of good creditors could be to promote shorter credit durations.

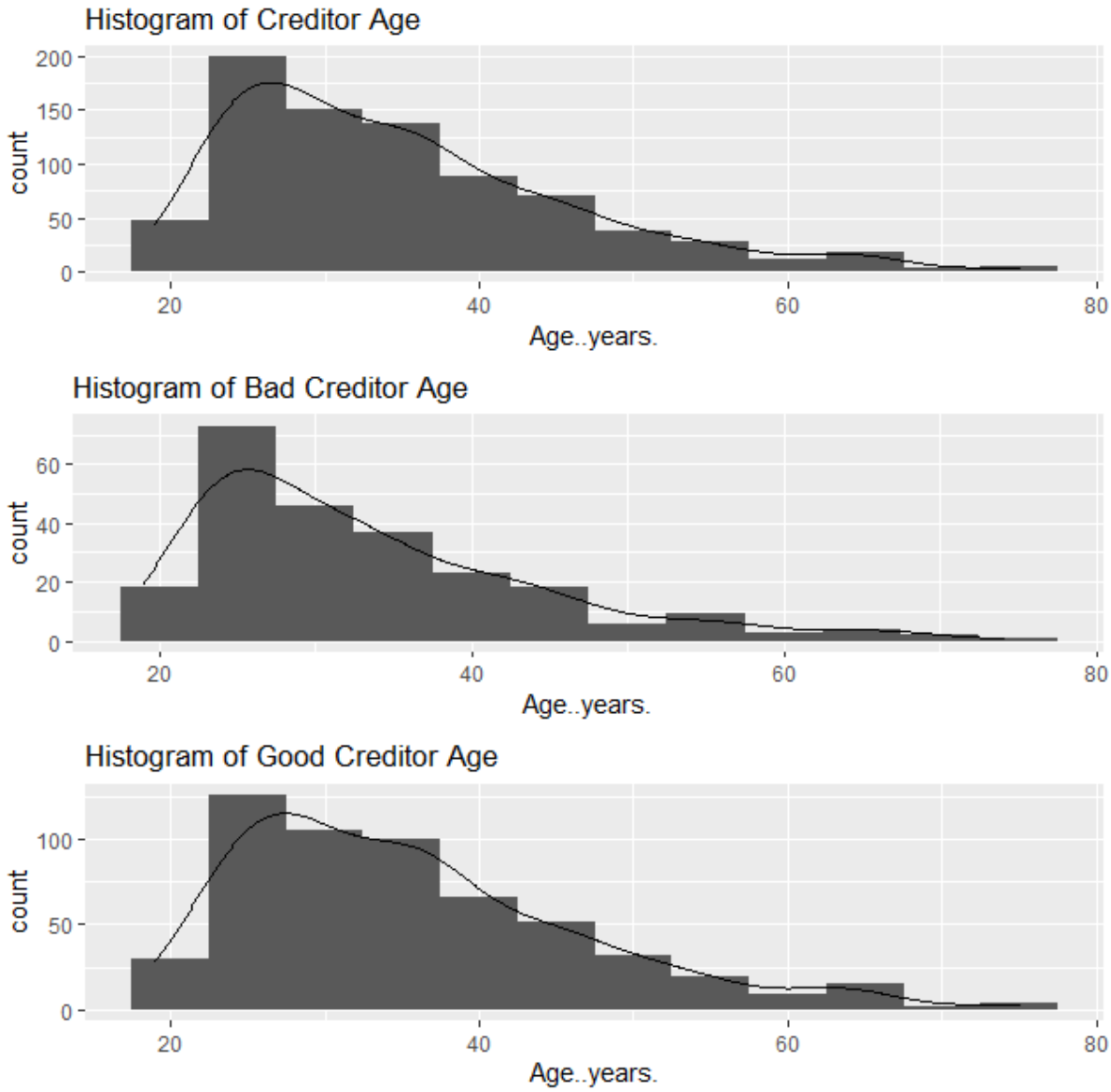

Figure 9: Distribution of good creditors to bad show that good creditors are more likely to be aged over 25 years.

BIVARIATE ANALYSIS

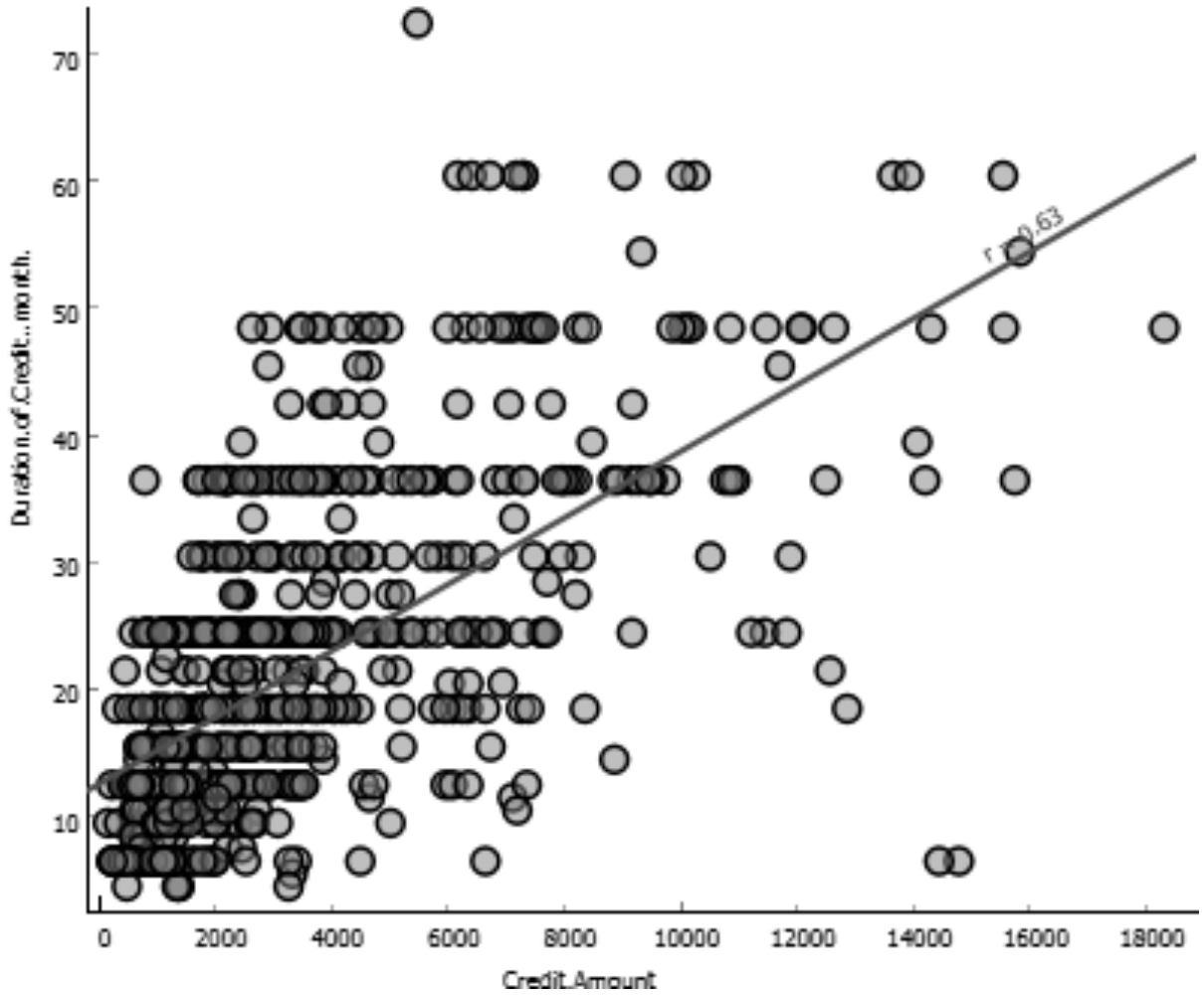

Figure 10: Logically, as the credit amount increases so does duration of credit month. As seen in figure 8, the bank could promote smaller credit amounts.

DATA REDUCTION

Principal components analysis was not performed on the dataset as it was shown to be not particularly helpful.

```
> train_eigen
       eigenvalue variance.percent cumulative.variance.percent
Dim.1   3.3311156         6.168733                    6.168733
Dim.2   2.5154171         4.658180                   10.826912
Dim.3   2.1132120         3.913355                   14.740268
Dim.4   1.8638208         3.451520                   18.191788
Dim.5   1.7181624         3.181782                   21.373570
Dim.6   1.6410561         3.038993                   24.412563
Dim.7   1.5281063         2.829826                   27.242389
Dim.8   1.4986775         2.775329                   30.017718
Dim.9   1.4034494         2.598980                   32.616698
Dim.10  1.3176311         2.440058                   35.056756
Dim.11  1.3047963         2.416289                   37.473045
Dim.12  1.2846987         2.379072                   39.852117
Dim.13  1.2403774         2.296995                   42.149112
Dim.14  1.2237824         2.266264                   44.415376
Dim.15  1.1868735         2.197914                   46.613290
Dim.16  1.1599504         2.148056                   48.761346
Dim.17  1.1353497         2.102499                   50.863846
Dim.18  1.1088219         2.053374                   52.917220
Dim.19  1.0949547         2.027694                   54.944914
Dim.20  1.0527991         1.949628                   56.894542
Dim.21  1.0419818         1.929596                   58.824137
Dim.22  1.0138881         1.877571                   60.701708
Dim.23  0.9895592         1.832517                   62.534225
```

Output 6: Cumulative eigenvalues explain about 62% of the variance in the dataset (measured up to 23 principal components). The whole data set is not explained even using the same number of principal components as there are attributes.

METHOD:

PREDICTIVE ANALYSIS: COMPLETED USING ORANGE

This report largely used classification models to train and predict 'German Credit' data as the dataset is composed of 20 categorical variables and only 3 numerical.

LOGISTIC REGRESSION

Logistic regression was chosen because it typically does well when trying to predict a target output of only 2 groups ('Creditability').

ROC PLOTS

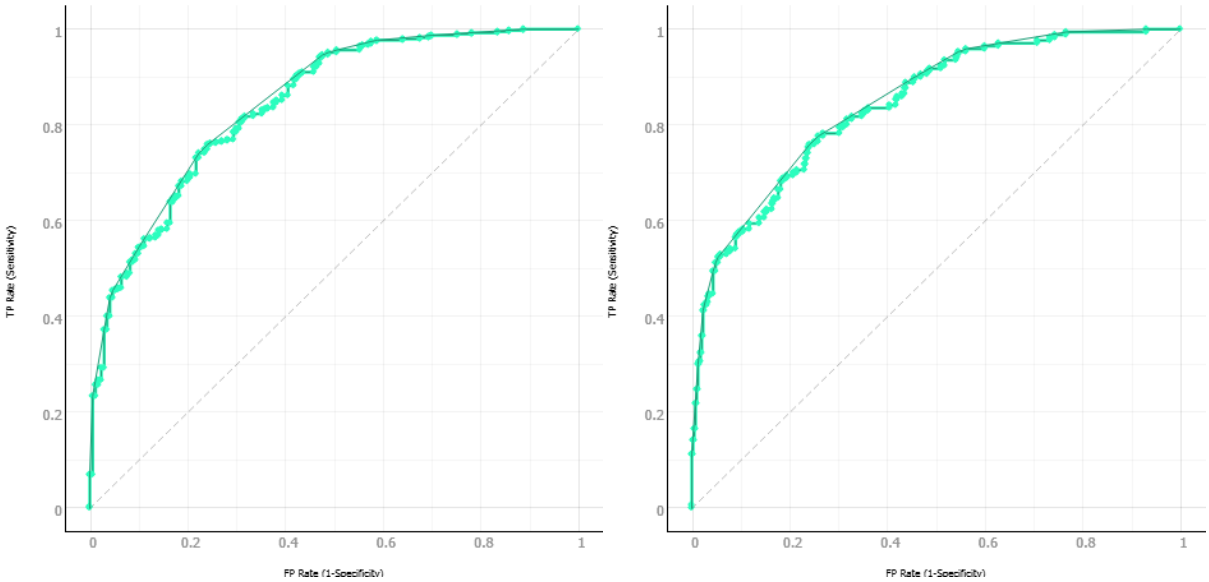

Figure 11: Left = good creditor, Right = bad creditor
There is no clear elbow for either ROC curve. The bank should likely choose some cut-off values for true positives.

F1 CALIBRATION CURVE

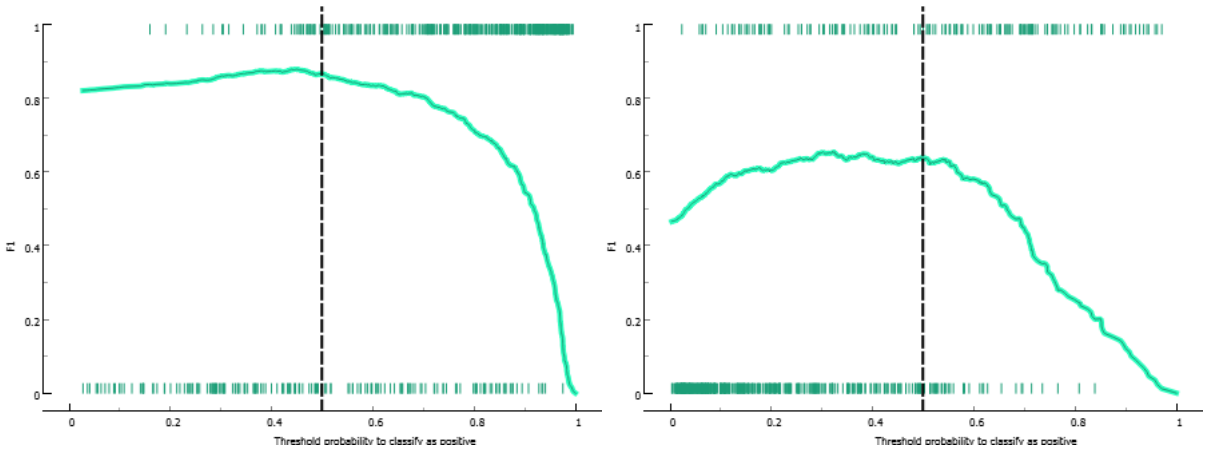

Figure 12: Left = good creditor, Right = bad creditor
Calibration plots show that in terms of F1 score, logistic regression performs much better when predicting good creditors compared to when it is predicting bad creditors.

RANDOM FOREST

Random forest models are very well equipped to deal with categorical data due to its tree algorithms.

ROC PLOTS

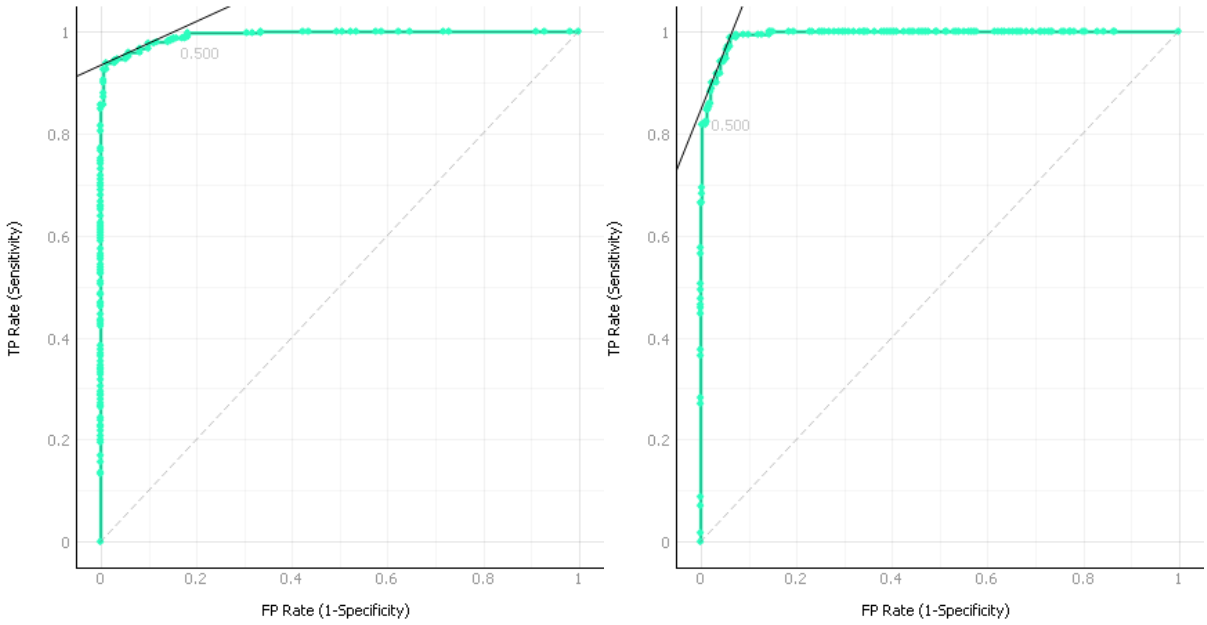

Figure 13: Left = good creditor, Right = bad creditor

For both ROC plots, there is a clear elbow showing the optimal false positive to true positive rate for both good and bad creditors. Additionally, the elbow almost perfectly represents a 100% accurate model.

F1 CALIBRATION CURVE

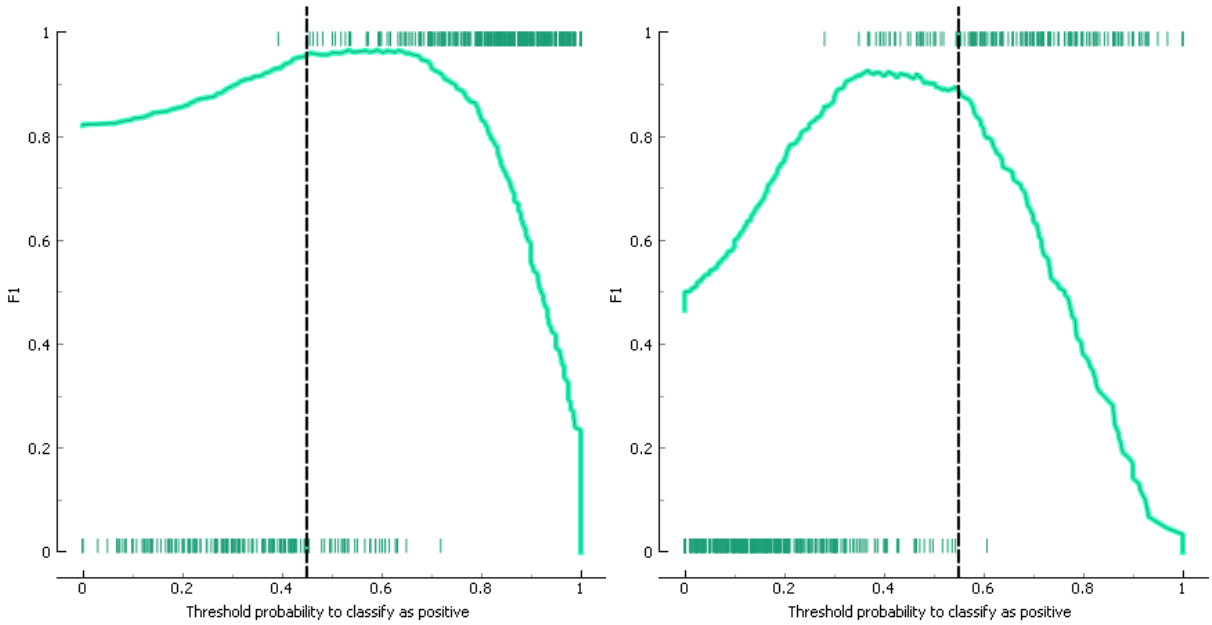

Figure 14: Left = good creditor, Right = bad creditor

As mentioned in figure 14, the random forest model very accurately can predict good and bad creditors based on training data. By moving the threshold probability, the random forest would be able to more accurately predict whether a creditor was good or bad e.g. bad creditor F1 score at 0.95 when probability threshold is at around 0.4.

NAÏVE BAYES

Naïve bayes is a fast classification model that is designed for data with few correlations between them. As seen in the data reduction section, no meaningful principal components were produced hence it could be interpreted as a good model to use for 'German Credit' data.

ROC PLOTS

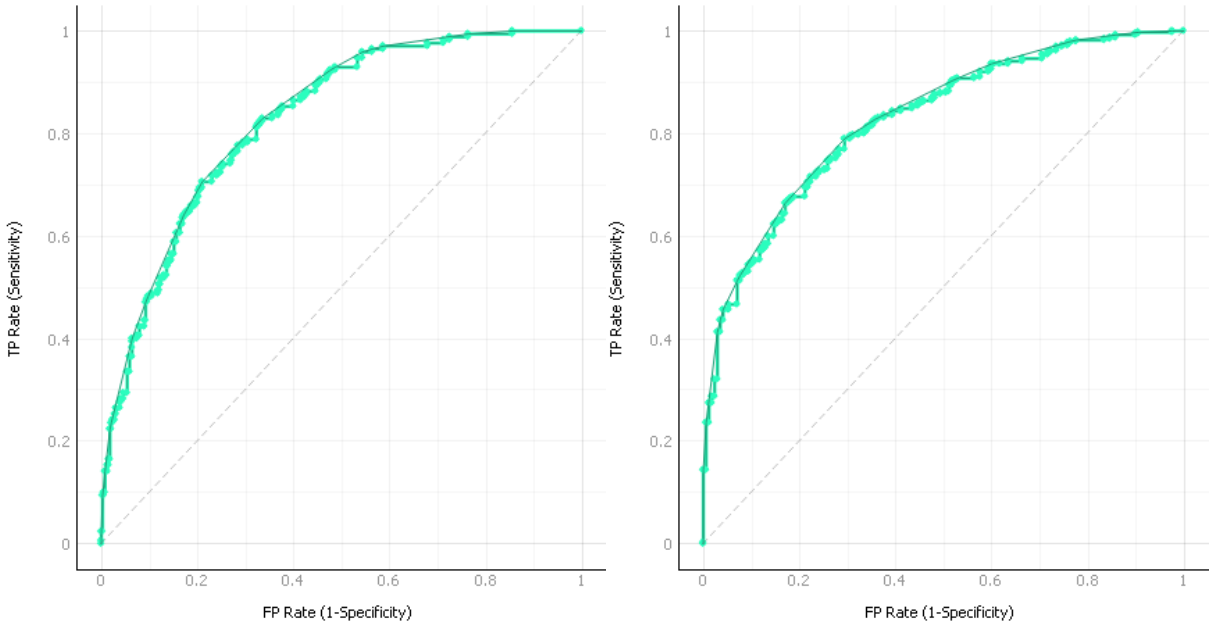

Figure 15: Left = good creditor, Right = bad creditor
Naïve bayes also shows no clear elbow. Its predictions of bad creditors is steep up until about 60% true positive rate where the model begins to lose accuracy. Predictions of good creditors is a smooth curve.

F1 CALIBRATION CURVE

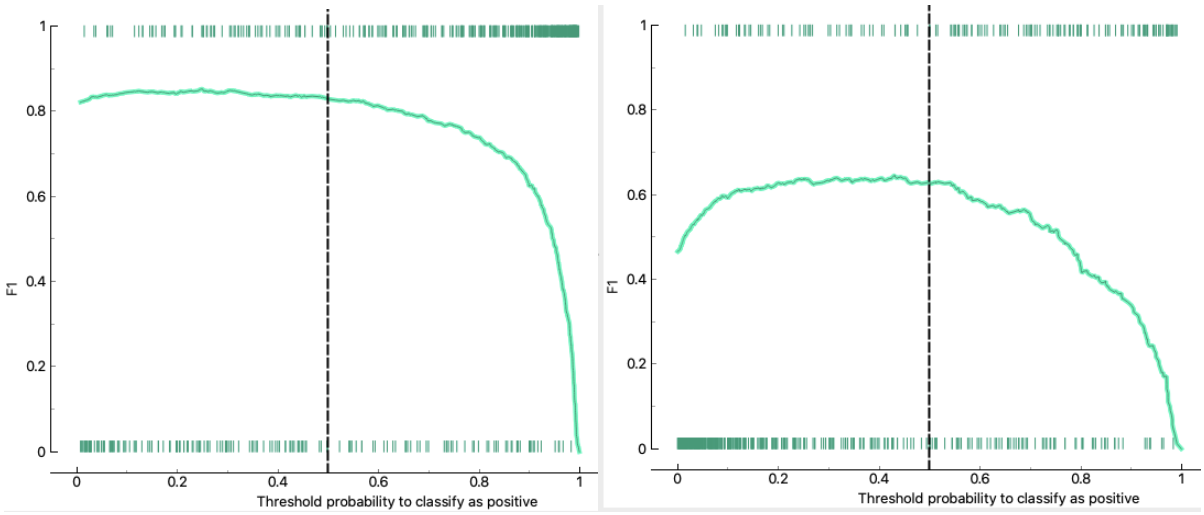

Figure 16: Left = good creditor, Right = bad creditor
F1 calibrations shares mostly the same story as with figure 15. Like logistic regression, naïve bayes is worse at predicting bad creditors compared to good ones.

SVM

Support vector machine (SVM) models are used for mixed data. With the presence of the 3 numerical attributes, it is likely a good model to train in this context. Additionally, SVM most effective when there is a significant clear margin between classes, and when there are high dimensional spaces (23 attributes).

ROC PLOT

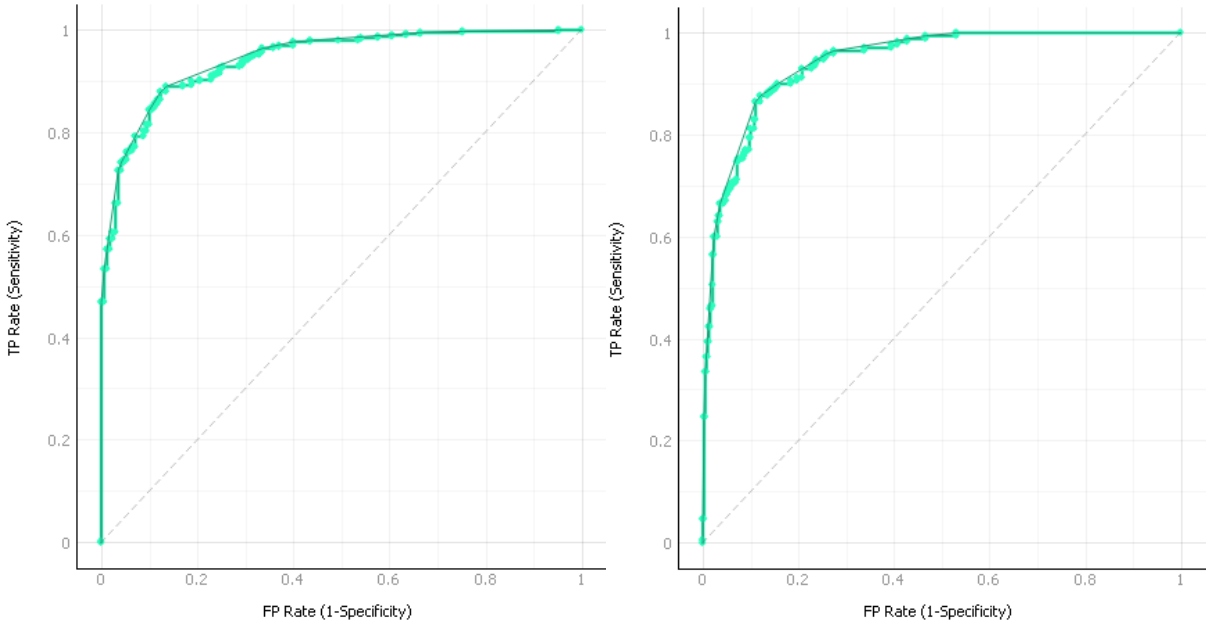

Figure 17: Left = good creditor, Right = bad creditor
SVM appears to be a slightly less accurate model compared to random forest while still out-performing logistic regression and naïve bayes models.

F1 CALIBRATION CURVE

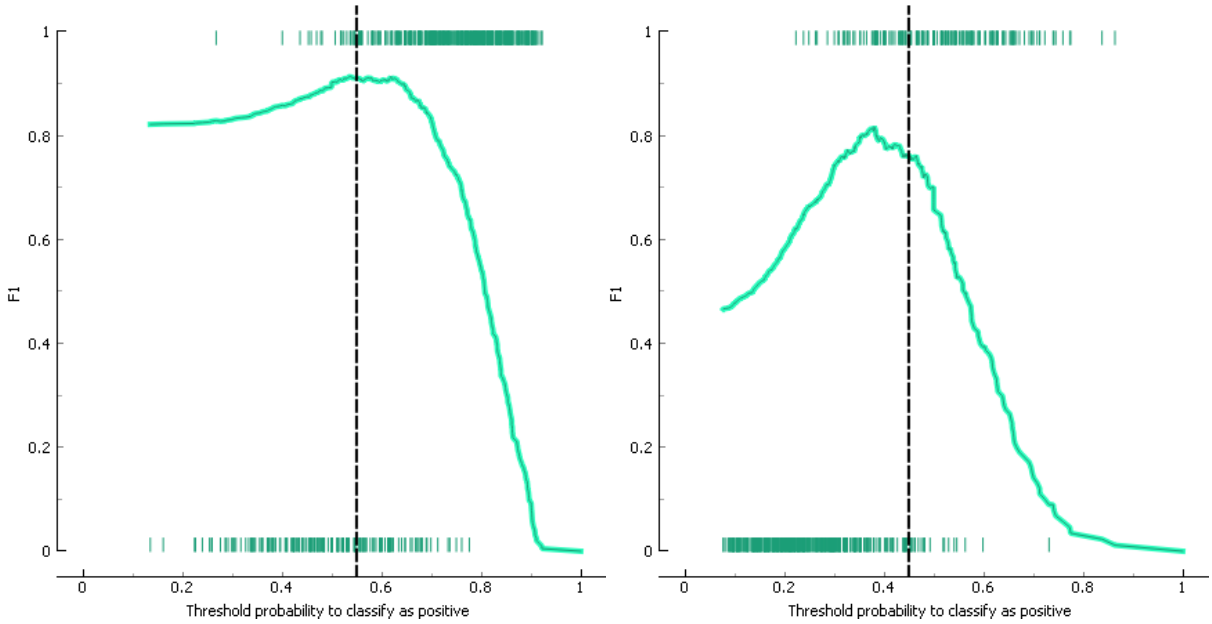

Figure 18: Left = good creditor, Right = bad creditor
Like random forest, SVM shows that it can more accurately predict bad creditors when moving the threshold probability to around 0.4. In this instance, moving the probability threshold to 0.4 is recommended as the SVM model still maintains a greater than 0.8 F1 score when predicting good creditors.

STACKING

Stacking was used as an attempt to reduce the faults of each model and find an equilibrium prediction between all models.

ROC PLOT

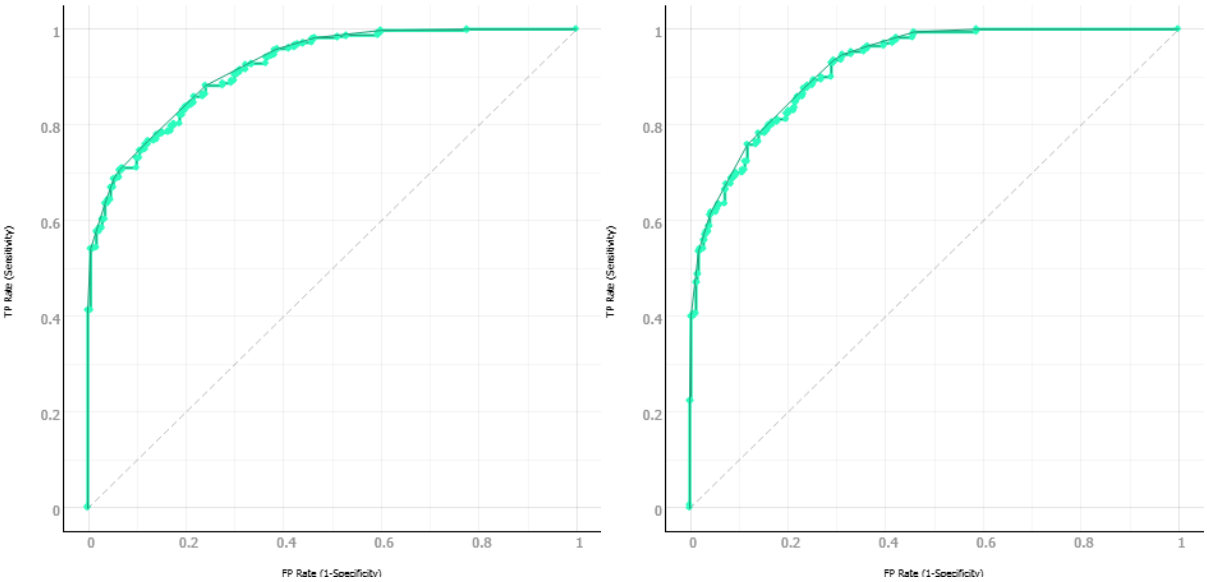

Figure 19: Left = good creditor, Right = bad creditor
Logically the combination of all individual models produces a merged ROC. The model appears to produce area under the curve results of equal value.

F1 CALIBRATION CURVE

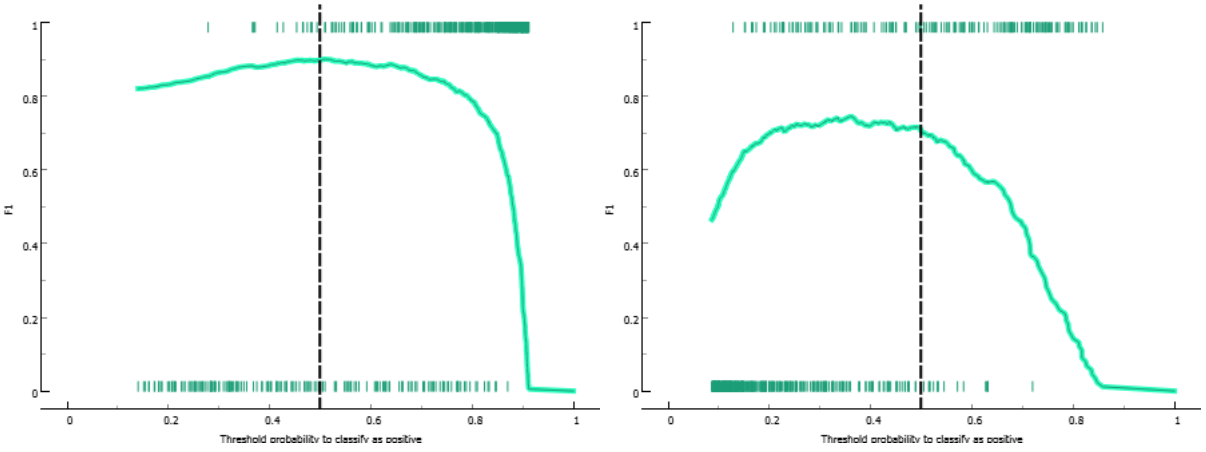

Figure 20: Left = good creditor, Right = bad creditor
In terms of F1, stacking multiple predictive models appears to still have problems when predicting bad creditors at just under 0.8.

TRAINING RESULTS

Model	**AUC**	**CA**	**F1**	**Precision**	**Recall**
Logistic Regression	0.841	0.804	0.796	0.797	0.804

Random Forest	0.993	0.945	0.944	0.946	0.945
Naïve Bayes	0.824	0.766	0.769	0.772	0.766
SVM	0.941	0.866	0.869	0.880	0.866
Stacking	0.940	0.870	0.863	0.872	0.870

CLUSTERING ANALYSIS: COMPLETED USING ORANGE

Most clustering algorithms measure the numerical distance between entries. Hence, many clustering algorithms should not show particularly useful information since 'German Credit Data' consists mostly of categorical values. Regardless, for this report Louvain, k-means, and DBSCAN algorithms were used

LOUVAIN

Louvain clustering is typically used for community detection. Although there is no clear network, the algorithm was able to produce some insights with regards to the limited numerical attributes. Results produced using categorical data appeared mostly insignificant.

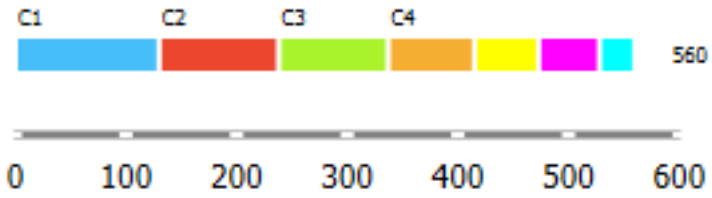

Figure 21: Louvain clustering identified 7 different clusters

Age	***Credit Amount***	***Duration of Credit Month***

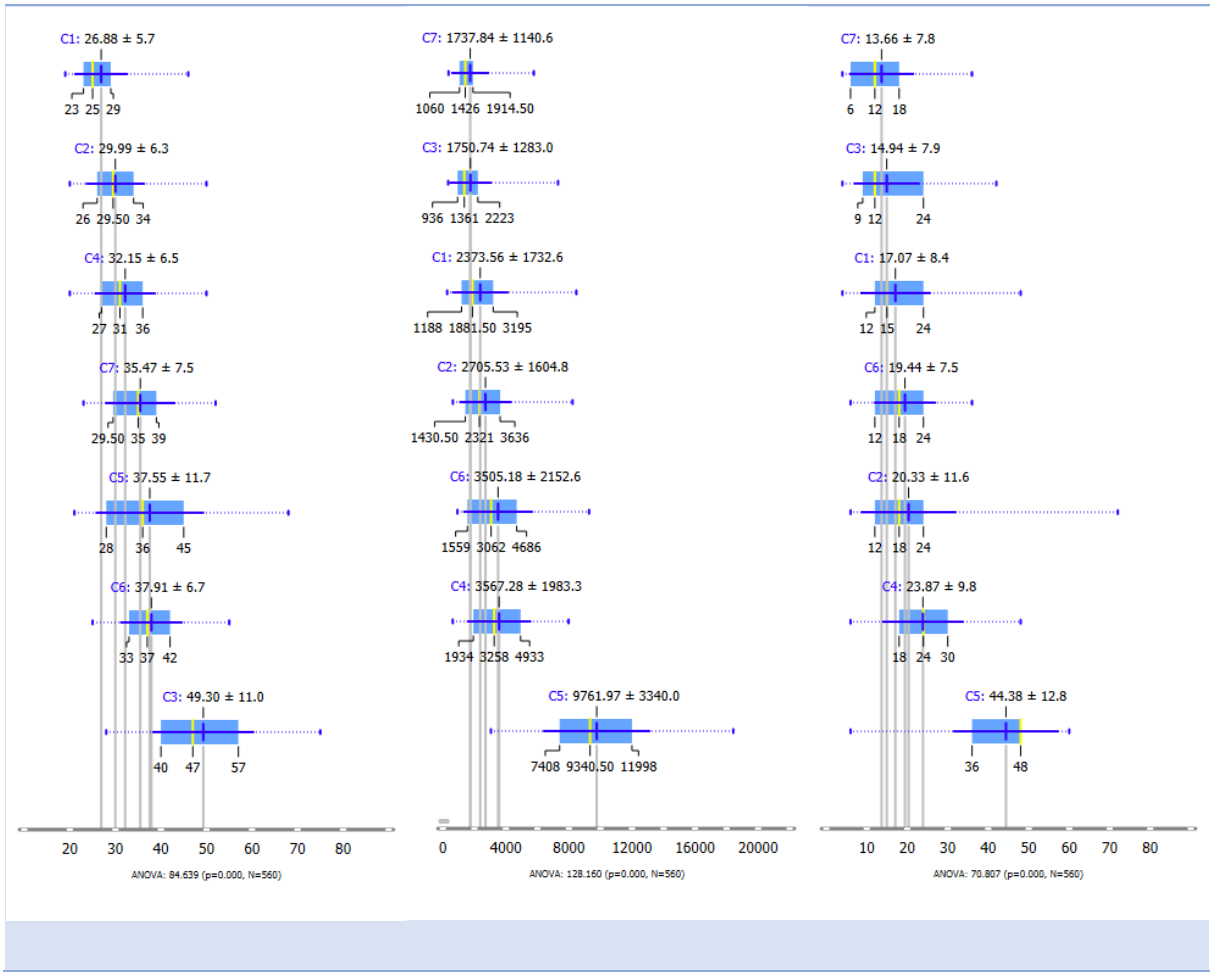

Figure 22: Consistently shows that aged creditors are more likely to have higher credit amounts and duration.

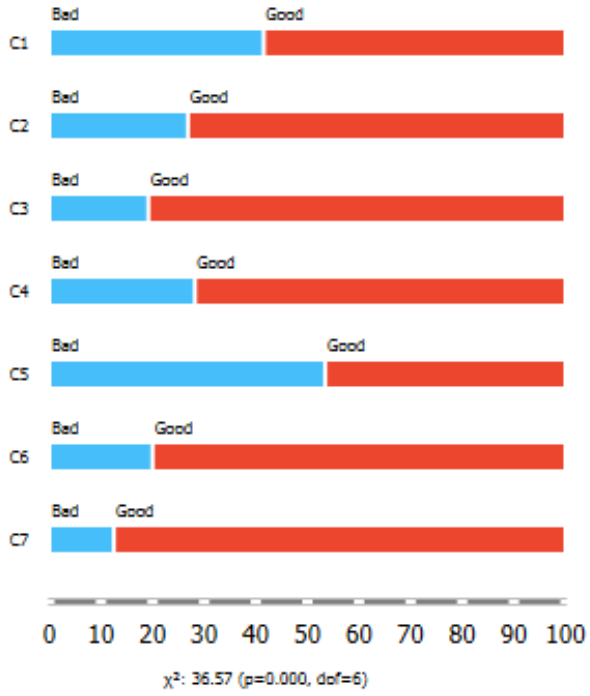

Figure 23: In contrast to figures 8 and 9, figure 22 shows that aged people (cluster 7) are more likely to have good credit when they have longer credit durations. However, the distribution of aged creditors appears to be quite low in figure 9. Hence it can be argued that the sample size is not statistically large enough for the produced insights to be significant.

K-MEANS

By applying K-Means analysis to our data and creating 2 clusters we were able to analyse the relationship of the clusters with different variables. By analysing all figures 24, 25 and 26, we can see that the younger population group is more likely to take out a loan. It is also shown that the younger population is also more likely to take out a loan for purchases such as furniture, television, or a new car whereas the older population is more likely to take out a loan for a used car. These insights are good for management to have to create target markets for their products however are insignificant in determining creditworthiness.

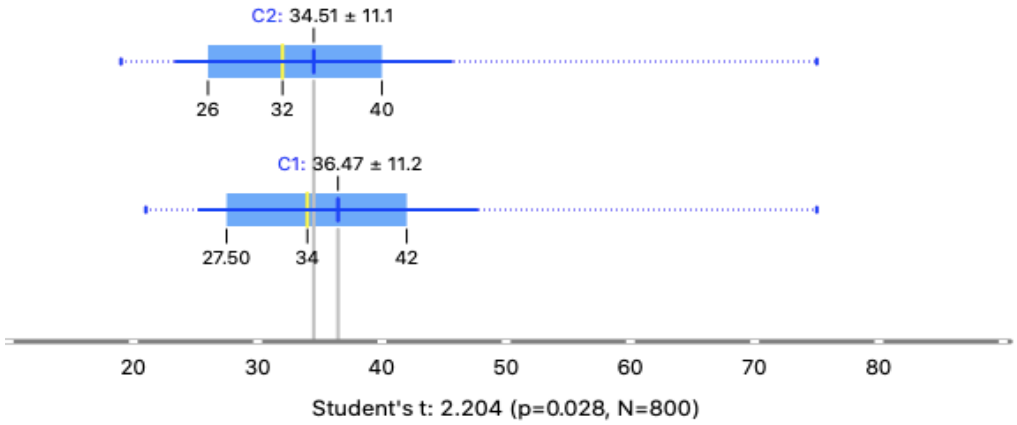

Figure 24: Here we can see which clusters have a higher average age – Cluster 1.

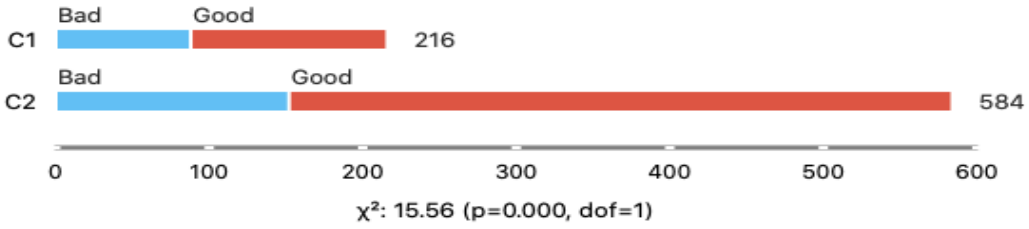

Figure 25: This figure shows the size of each cluster as well as their creditability.

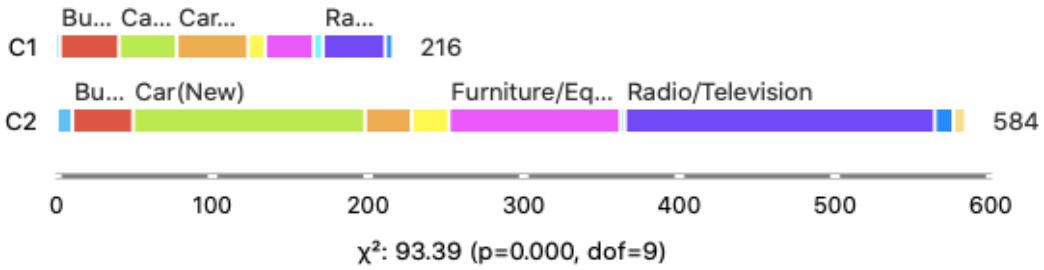

Figure 26: Here we can see the purpose of each person's loan.

DBSCAN

As DBSCAN is not an ideal clustering method for categorical data, no significant clusters are formed when using our clean data. Continuizing our data into numerical variables causes the model to lose accuracy as the new variables do not accurately represent the dataset.

RESULTS:

PREDICTIVE ANALYSIS

Although training results in the previous section presented AUC, accuracy, precision, and recall, this report will focus on producing the highest F1 score as per the results of the German credit Kaggle competition.

TESTING

LOGISTIC REGRESSION

F1 Score = 0.87608

RANDOM FOREST

F1 Score = 0.90628

NAÏVE BAYES

F1 Score = 0.85312

SVM

F1 Score = 0.87182

STACKING

F1 Score = 0.88275

AVERAGE

F1 Score = 0.89736

WEIGHTED AVERAGE

Produced a weighted average using the mean square error of each model.

$$Inverse\ MSE\ (IMSE) = \frac{1}{MSE}$$

$$Model\ Weight = \frac{IMSE_{model}}{IMSE_{LR} + IMSE_{RF} + IMSE_{NB} + IMSE_{SVM}}$$

F1 Score = 0.90254

THRESHOLD

Done by moving the threshold probability of combined models. Most successful attempt was where the threshold probability was set to 0.4.

F1 Score = 0.92706

CONCLUSION:

In conclusion, clustering analysis solutions are not highly recommended to determine creditworthiness as the results shown do not improve or influence prediction accuracy of any models. Clustering analysis solutions are recommended if management would like to determine whether there are any distinct market segments amongst their customers.

The best results are achieved using a combination of the models and use a threshold probability of 0.4 to as it has the highest accuracy at 92.70%. Its results can be compared to the accuracy of the less complex random forest model with an accuracy of 90.63% as it is relatively close and less time consuming.

As the bank's main priority is to minimise loss by correctly identifying individuals with good and bad creditworthiness, it is recommended that the combined models with threshold probability is the chosen model for the bank. The accuracy discrepancy of 2% will have a significant impact on whether the bank chooses the right candidates to give loans to or not, therefore, ensuring the bank maximises profit and minimises loss.