CIND119 Final Project Submission- German Credit Data Analysis

Report By: Daniel Platnick
Student Number: 500645521

For: Dr. Lak

1. Abstract

My client is a bank who supplies loans to customers. They must continuously classify customers on the basis of creditworthiness and decide if they have good creditability. Ideally, the company would only like to provide loans to customers who will pay back the loan. Currently the method of classification is inefficient and limited. There is lots of room for improvement in the efficiency of the loan-decision making process. Through the use of machine-learning strategies like decision trees and the naive Bayes classifier, I intend to develop a model which increases efficiency and accurately classifies customers as having good creditability or bad creditability. In addition, I also wish to provide the bank with some insights and helpful analytics which can help them understand the basis of the classification of a customer.

2. Data Preparation

The German Credit dataset consists of 1000 observations with 21 attributes including the class attribute. Of the 1000 observed instances, 700 were classified as good credit, and 300 as bad credit. There were zero null values present in the dataset, so there were no incomplete instances. The dataset consists of a mix of numerical, and categorical variables. Of the 21 attributes, 6 are numerical and the remaining 15 are categorical variables. Eleven of the categorical variables are nominal, and the remaining 4 are ordinal variables. Figure 1 displays some summary statistics of the numerical variables computed. Python was the only software tool used in this experiment.

Figure 1: Summary Statistics of Numerical Variables

	Duration of Credit (month)	Credit Amount	Instalment per cent	Age (years)	No of Credits at this Bank	No of dependents
count	1000.000000	1000.00000	1000.000000	1000.00000	1000.000000	1000.000000
mean	20.903000	3271.24800	2.973000	35.54200	1.407000	1.155000
std	12.058814	2822.75176	1.118715	11.35267	0.577654	0.362086
min	4.000000	250.00000	1.000000	19.00000	1.000000	1.000000
25%	12.000000	1365.50000	2.000000	27.00000	1.000000	1.000000
50%	18.000000	2319.50000	3.000000	33.00000	1.000000	1.000000
75%	24.000000	3972.25000	4.000000	42.00000	2.000000	1.000000
max	72.000000	18424.00000	4.000000	75.00000	4.000000	2.000000

Figure 1 describes how the average customer is about 35, has roughly $3,300.00 in outstanding credit, and has held credit for about 21 months. It can also be seen that the ages of customers range from 19-75, and the maximum amount of credit held by a customer is $18,424.00. The display of quantiles also gives a quick overview about the distribution. Additionally, figure 1 communicates how the longest outstanding credit duration is 72 months. In order to understand the spread of each numerical variable better some boxplots were computed to display any outliers as well as show the inter-quartile range for each variable.

Figure 2: Boxplots of Numerical Variables

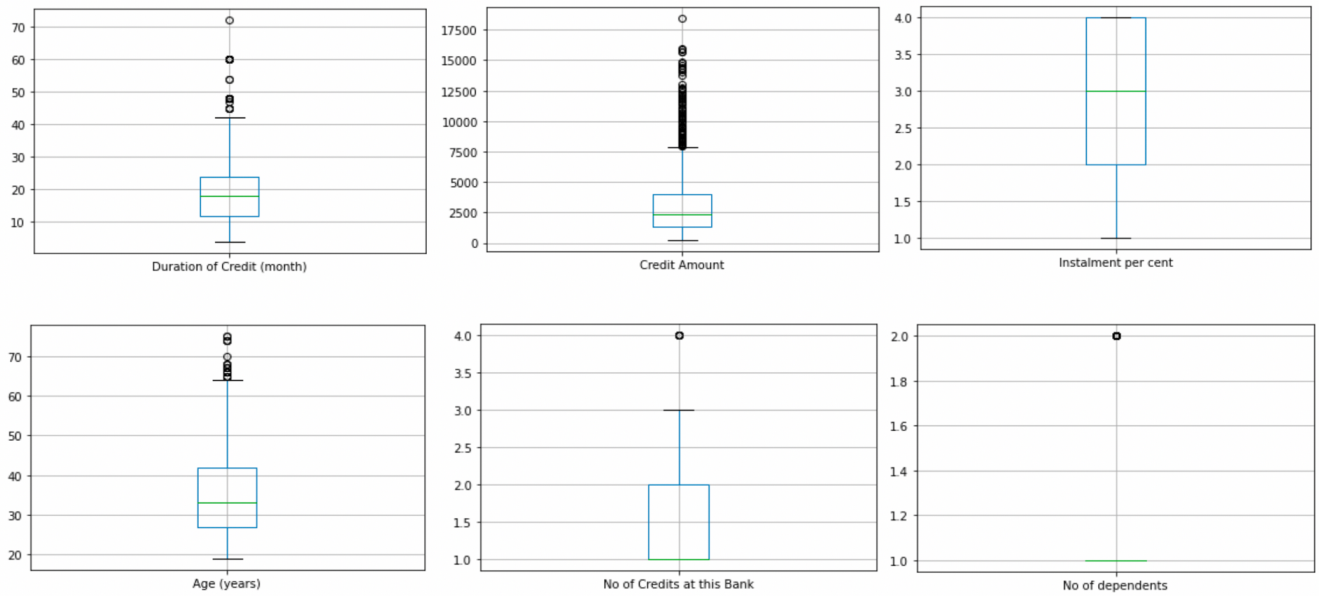


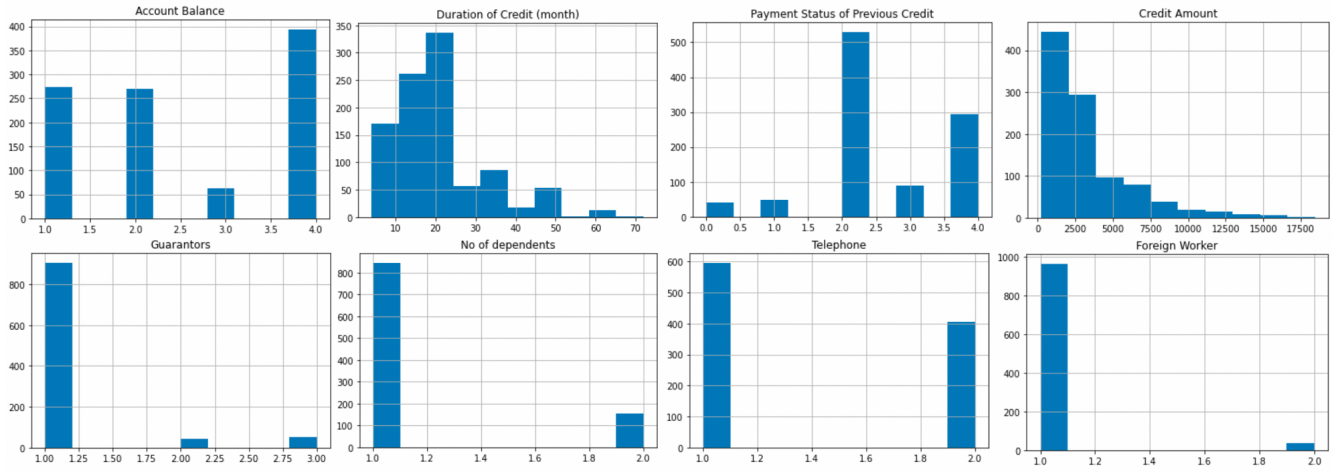
Figure 2 describes the distribution of each variable. Figure 2 expresses how most customers are between ages 29-41, and there is a group of outliers who are aged 64-75. The boxplot also gives important information about the credit amount. It can be seen that most people have about $1,500-$4,500 in outstanding credit, and there is another cluster of people whose credit ranges from $7,500-$15,000. Figure 2 also describes how most people have 1-2 loans taken out by the bank, and how most people have had outstanding credit for 11-23 months. There are some outliers in the dataset such as observations having a credit duration period of over 70 months, however no extreme outliers. There are many outlier observations forming a cluster of instances with credit amounts totaling $8,000-$15,000 which indicates a subset of customers with high credit amount. No outliers were removed during the data filtration process, in order to hold the integrity of the sample. Figure 3 displays a correlation heatmap of every attribute in the original dataset.

Figure 3: Correlation Heatmap of Attributes

In deploying multivariate statistical models, it is important to have an understanding of the correlation between different covariates, and most importantly the class variable. Figure 3 displays a correlation heatmap of every attribute from the original German Credit dataset. With the correlation heatmap it is possible to compare and view the various correlations between each variable. From figure 3 it can be seen that account balance is the highest correlated attribute with the class variable creditability, having a Pearson coefficient of 0.35. This makes sense because customers with higher account balances will likely have enough money to pay off their loan. A highly negatively correlated variable (Pearson of -0.21) is duration of credit (months), which means that customers with a longer duration of outstanding credit tend to be less creditable.

Some of the other highest negative and positively correlated variables are payment status of previous credit, credit amount and value of savings/stocks. Additionally, some of the least correlated variables are duration in current address, number of dependents, telephone, and purpose of credit. A variable like duration in current address has very low correlation with the class attribute (Pearson of -0.003). Figure 4 displays some histograms that were generated on the 4 most and least correlated attributes to the class attribute using python.

Figure 4: Histograms of Highest and Lowest Correlated Attributes

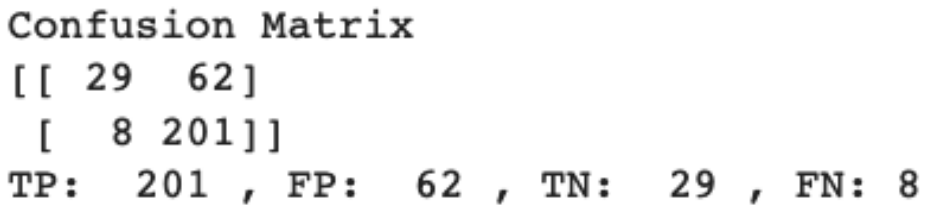


Viewing the distributions of your variables can provide additional insights. From figure 4 it can be seen that most individuals have credit amounts totaling less than $5,000. We can see that the distribution of credit amount is right skewed and that as the credit amount goes up, the amount of people who own that much credit goes down. Duration of credit is similarly right skewed. Additionally, in viewing the distribution of certain variables such as guarantors and foreign worker, it is such that few observations hold different values, so there is not much variation to capture in these variables. Variables with very low correlation such as telephone, number of dependents, guarantors, foreign worker, and also duration at current address were removed from the original dataset in the data filtration process. Removing variables with low correlation will help the model focus on attributes which explain more about individual customer creditability.

3. Predictive Modeling/Classification

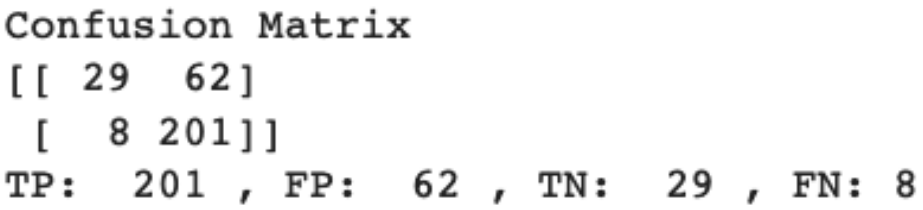
I used a classic 70/30 train-test split for all classification models discussed. This method was compared to a 75/25 as well as a 65/35 train-test split, however the 70/30 split yielded the most robust findings. The decision tree was set to a max depth of 5 which is a metric of how large and complex the tree should grow. The two of the most common decision tree decision metrics are the Gini index (impurity), and entropy/information gain. In practice, the Gini index is known to be less computationally expensive when compared to the entropy decision criteria. For the purposes of a small dataset of 1000 observations, the decision was made to use Gini impurity as the measure of partitioning the data in the decision tree.

Figure 5: Model 1's Confusion Matrix



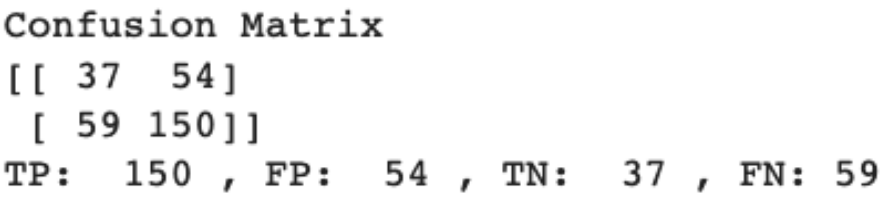
Model 1 is a Gini impurity decision tree classifier used with the original unfiltered German Credit dataset. The Gini impurity measures the probability of a variable being wrongly classified when it is randomly chosen. Model 1's decision tree algorithm chose the variable account balance as the root node in the tree with a Gini index of 0.419. The next two nodes highest up in the tree are duration of credit (month) and concurrent credits. Duration of credit (month) has a Gini impurity of 0.496. Concurrent credits have a Gini impurity of 0.211. Therefore, the decision tree has computed that account balance is the most important attribute when deciding a person's creditworthiness, and duration of credit (month) as well as concurrent credits are the second and third most important attributes. Model 1's confusion matrix has 201 true positive's, 62 false positives, 8 false negatives, and 29 true negatives. The accuracy measure for model 1 is 77%. Model 1's precision is 78% and in the context of credit analysis being precise is important because you will lose lots of money if a customer defaults on a loan. Therefore, in this context precision is a good measure of model effectiveness.

Figure 6: Model 2's Confusion Matrix



Model 2 is similar to model 1 as it uses the Gini impurity decision tree method as well as a 70/30 train-test split method. When comparing model 1 and model 2, the confusion matrices are identical. Model 2 was run on a filtered dataset, which consisted of 16/21 variables from the original dataset. The dropped variables were foreign worker, number of dependents, duration in current address, guarantors and telephone. They were dropped because of having low correlation with the target variable or an imbalanced frequency distribution. Model 2's decision tree is almost identical to model 1's because the tree pruned out most of the dropped variables anyways. However, model 1 uses the variable duration at current address (Gini impurity 0.142) and it is absent in the decision tree for model 2. Models 1 and 2 are equal in accuracy, precision and overall, there is no practical difference in choosing between them- however model 2's algorithm may be marginally more efficient due to less need for pruning.

Figure 7: Model 3's Confusion Matrix



Model 3 is a naive Bayes classifier implemented on the original unfiltered dataset. The naive Bayes classifier is a classification technique which is based on Bayes' Theorem. Naive Bayes works under the assumption that each of the attribute variables are dependent of one another.

The naive Bayes classifier was computed in python and generated a confusion matrix with 150 true positive's, 54 false positives, 37 true negatives and 59 false negatives. The precision is 39% for this model and the recall is 41%, with an accuracy measure of 63%. Model 3 performs worse than model 1 and model 2 in the areas of precision as well as recall.

Figure 8: Model 4's Confusion Matrix

```
Confusion Matrix
[[ 29  62]
 [  8 201]]
TP:  201 , FP:  62 , TN:  29 , FN: 8
```

Model 4 is the same naive Bayes classifier used on the filtered dataset. The filtered dataset is a subset of the original dataset with the attribute's foreign worker, number of dependents, duration in current address, guarantors and telephone removed. These variables were removed from the dataset in an effort to raise the accuracy of the model, because it helps the model ignore less important attributes. After running the naive Bayes classifier on the filtered dataset, it was apparent that removing the discussed variables had a positive effect on the model. Model 4 generated a confusion matrix with 201 true positive's, 62 false positives, 29 true negatives and 8 false negatives. Model 4's results seem more robust when compared to model 3. The precision of model 4 is 78% and the recall is 32%. Model 4's accuracy is 77%. It seems that removing the discussed features raised the precision from 39% to 78% between models 3 and 4 which is a great difference.

4. Conclusions and Recommendations

In terms of accuracy and precision, the most successful models were both of the decision trees, and the naive Bayes classifier which was used on the filtered dataset. Both decision trees yielded results similar to that of the naive Bayes classifier used on filtered data. The confusion matrix for decision tree 1, decision tree 2 and the filtered data naive Bayes classifier are identical in terms of recall, precision and accuracy. Although these models are identical in terms of the confusion matrix, they all outperform the naive Bayes classifier on the unfiltered dataset. Additionally, these models differ in that the decision tree produces an easily followed graph, outlining exactly how the decision parameters come to a classification value. In contrast the naive Bayes model is much less easy to interpret. I believe that in the setting of business consulting, a clear communication of statistical methods to your stakeholders is an invaluable tool. I believe that the decision tree produces equally effective results, however it is more easily understood compared to the naive Bayes and therefore would be a better recommendation for the bank to implement. When comparing model 1 (decision tree unfiltered) and model 2 (decision tree filtered), the tree on pre-filtered data is marginally more computationally efficient when compared to the decision tree on the unfiltered data (less pruning necessary). Therefore, my conclusive recommendation would be to employ the decision tree on filtered data, as it yields the highest precision and at the same time can be broken down and visualized by stakeholders.