CREDIT RISK SCORING

PART A

Introduction

Credit risk scoring is a statistical analysis performed with the use of multiple tools by Banks or other financial organizations to determine if a customer or a business is credit worthy, and the Bank/organization will stand to make profit by lending money to the customer. Some of the simplest forms of credit risk scoring is Logistic and Linear Regressions. The goal of these 2 types of regression is the creation of a predictive model that will be able to classify future customers into "good" or "bad" based on past data.

Pre-processing

First a general descriptive analysis was conducted in the dataset in order to understand the different types of variables that exist, their distributions and what might need to be changed for the convenience of the model input.

The distribution of the classifications' goal shows signs of being a problem, as the percentage of "good" in the dataset is 70% and only 30% is "bad", this might create a prediction bias [Figure 1].

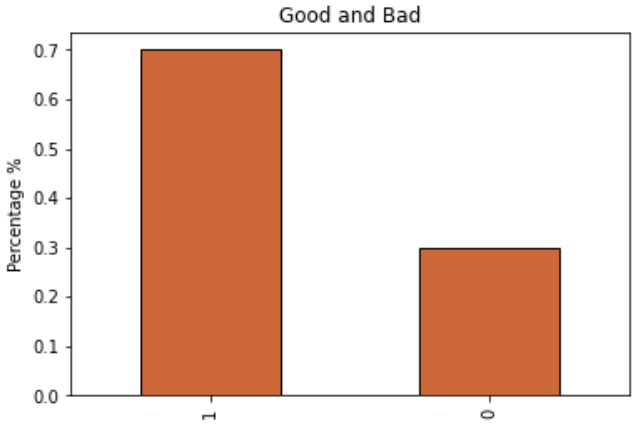

Figure 1: Percentage of "Good"=1 and "Bad"=0

The column "Bad" from the original dataset it dropped as all the information that "Bad" provides already exist in "Good", making that column unnecessary, and that is valid for all the categorical features mentioned to avoid the dummy variable trap, meaning to avoid 2 or more variables being perfectly correlated and cause the model to make incorrect calculations. Moreover, the "X" in column purpose is replaced by "10" as that would help in later stages for the binning procedure. Also, a search for empty and "NaN" spaces was conducted, finding none.

Finishing the pre-processing, the dataset was then split into 2 subsets, one where all the applicants have "Checking"<=2 and one with "Checking">2.

Train and Test set

The training set contains a sample of the original dataset that is used to fit the model and the test data is usually a smaller sample from the same dataset, used to test the model on data that it had never worked with before, so the creator can see how well the model works with new data, and it can also help avoid circumstances of over-fitting or under-fitting. The training set for this model is 70% of the total subset randomly selected, and thus the test set is the remaining 30%.

After having established the training and test set, two bar plots were created that show the percentage of “good” and “bad” in each training set. Subset 1 seems to have a good proportion of “good” at 56% and “bad” at 44%, while subset 2 seems to have a huge problem because it only has 13% of “bad” which will in turn push the model to have bias on the predictions [Figure 2].

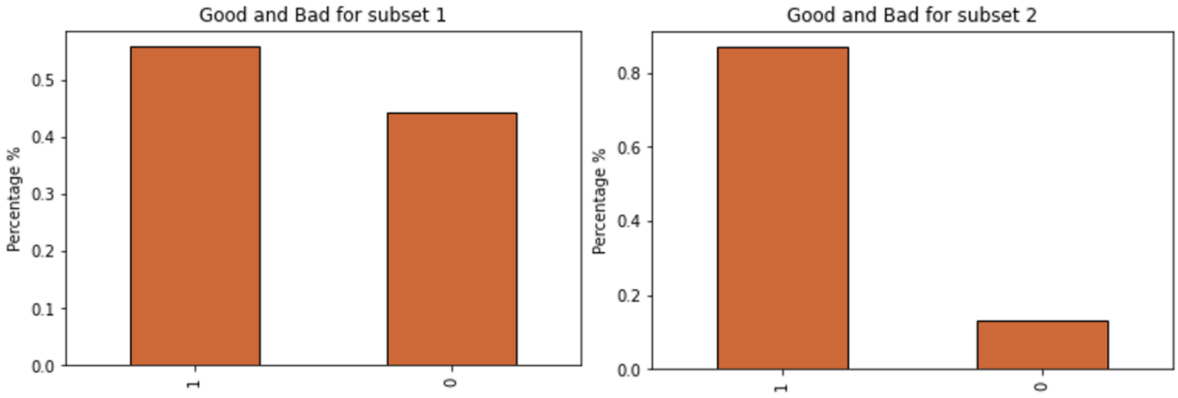

Figure 2: Percentage of Good and Bad for the 2 Subsets

Choosing the 4 most important variables

To help with the selection of the 4 most important variables for this predictive model, a very useful technique called Information Value was used [Appendix A]. IV is calculated by the sum of the percentage of non-events minus the percentage of events and that multiplied by the weight of evidence value. As a rule of thumb, any value below 0.1 is considered to have very weak relationship with the “good”/”bad” ratio, 0.1-0.3 is pretty good, 0.3-0.5 has strong predictive power and anything above 0.5 has a very suspicious predictive power and must be investigated.

For the training set of Subset 1, the most important variables were 'Age', 'History', 'Duration' and 'Property', and for Subset 2 the most important variables were ‘Age', 'Other', 'Duration' and 'Purpose’, but ‘Duration’ had a suspiciously high IV value, most likely because of the uneven distribution of “good” and “bad” of Subset 2. So ‘Duration’ was changed to ‘Emploed’ in order to avoid ‘Duration’ becoming the main weight in the model and the other variables to just take a corrective role.

Model Creation

For the creation of the models, Python was used. For each subset, two models were created, one using logistic regression and one using linear regression with a cut-off rate of 50% to transform the linear regressions’ predictions into binary. The “sklearn” python library was used to help with the creation of the model and for the creation of the scorecard another library called “scorecardpy” was used.

Subset 1 Logistic Regression

The coefficients of the Logistic Regression are logical and the most important variable seems to be ‘Age’, followed by the other 3 as shown below on Table 1.

Coefficients			
Age	Property	Duration	History
1.19972	0.822376	0.854162	0.706516

Table 1: Coefficients of Logistic Regression Subset 1

Looking at the Figure 3 below, the ROC for this model has a “fat curve” and shows it performs better than a random model would by 17.76% more according to AUC. That means that the proportion of

correctly classified “good” samples are more than the incorrectly classified samples that are not “good”.

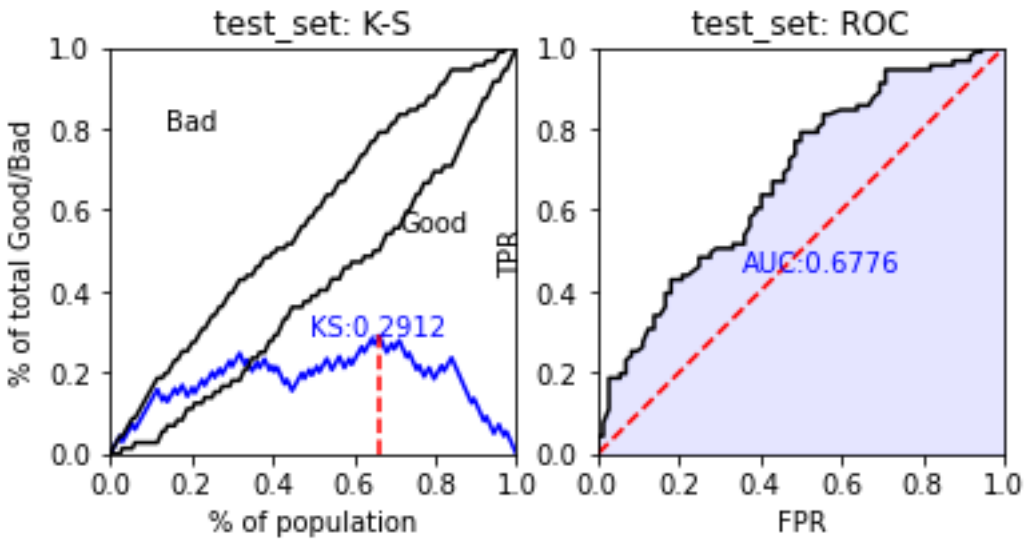

Figure 3: KS and ROC for Logistic Regression model, Subset 1

The KS value as shown in Figure 3, is a performance indicator for classification models and measures the degree of separation between the positive and negative distribution functions of the two classifications “good” and “bad”. The bigger the separation between the two functions the better it is (usually between 0-1), and the biggest value is considered the threshold value for accepting or rejecting. In this case the KS value is 0.29 indicating that the model is not bad but also not very good considering that the KS value of 0 indicates a random model.

The Gini value as seen by the calculation below, is directly correlated to the AUROC and as such shows the same information as the AUC. If the Gini coefficient is 0 then that shows that the model cannot discriminate between “good” and “bad”, and if the coefficient is 1 shows perfect discrimination [Table 2].

$$Gini = 2 * ROAUC - 1 = 2 * 0.6776 - 1 = 0.3552$$

Gini and KS value	
Gini	**KS**
0.3552	0.2912

Table 2: Gini and KS value for Logistic Regression model, Subset 1

The confusion matrix, and some results from that (such as specifity and sensitivity), as well as the bins and score allocation can be found on [Appendix B].

Subset 1 Linear Regression

The coefficients of the Logistic Regression are very logical again, as the most important variable follows the decision of the “IV” feature selection and is ‘Duration’, followed by the other 3 as shown below on Table 3.

Coefficients			
Duration	Age	History	Property
0.239231	0.1686	0.172955	0.143548

Table 3: Coefficients of Linear Regression Subset 1

The ROC for this model as can be seen on Figure 4, shows identical results as the Logistic Regression model, with a very slight insignificant difference on the AUC.

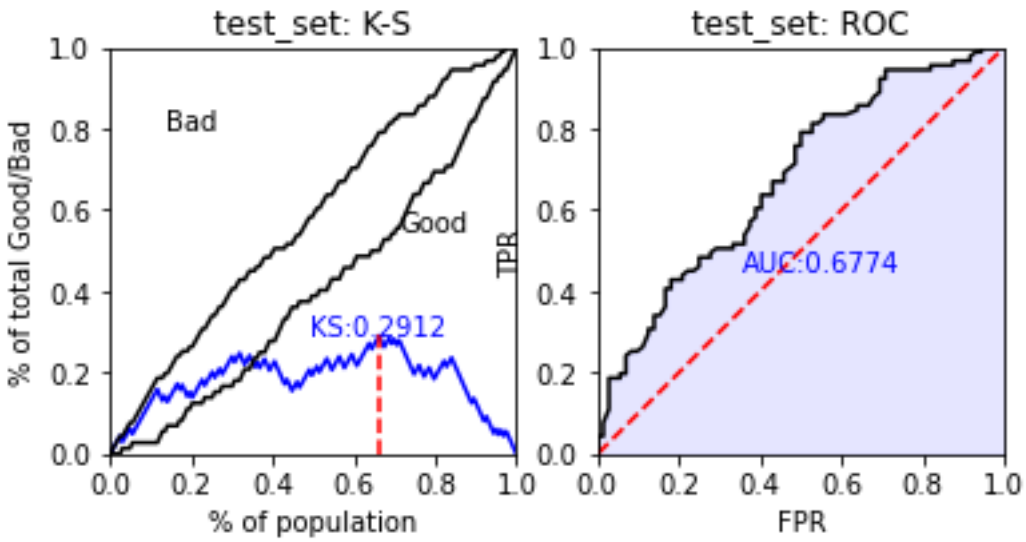

Figure 4: KS and ROC for Linear Regression model, Subset 1

As for the Gini and KS value, both are again identical with a slight difference of 0.08% on the Gini coefficient because of the change on AUC.

Gini and KS value	
Gini	**KS**
0.3552	0.2912

Table 4: Gini and KS value for Linear Regression model, Subset 1

The confusion matrix, and some results from that, such as specifity and sensitivity can be found on [Appendix B]. From these results we can conclude that both models work very similarly with almost no difference between them in regards to the results, except on the way they work, and that can also be concluded by seeing the difference of coefficients.

Subset 2 Logistic Regression

Even though the variable 'Age' was shown to be the most important feature through IV, it seems that every other variable was given more gravity than 'Age' as seen below on Table 5, which is interesting and maybe has something to do with the heavily imbalanced dataset.

Coefficients			
Age	Other	Purpose	Employed
0.701136	1.36068	0.817983	0.84459

Table 5: Coefficients of Logistic Regression Subset 2

The ROC curve of this model as can be seen in Figure 5, shows a high performing model at a glance, with AUC being at 72.% and Gini 0.44 [Table 6] and KS value being 0.3674, a pretty good value, but looking at other metrics such as the confusion matrix [Appendix D] it becomes clear that indeed the model is biased.

The model at glance looks to be good, having high accuracy of 83%, but taking a closer look, due to the low number of "bad" of the dataset, the model has not created a good idea on how a customer is considered to be "bad" and this is clear by seeing the recall rate for the "bad", which is 0.06, meaning

that of all the points of Real Negative, the model has correctly predicted only 6%. And due to the imbalance, having only 13% of "bad", the model gives 94% wrong prediction on a bad customer. In the case that such a model is deployed, it would give the "green light" to the bank to give loans to people with bad credit that would most likely create financial damage to the bank or organization.

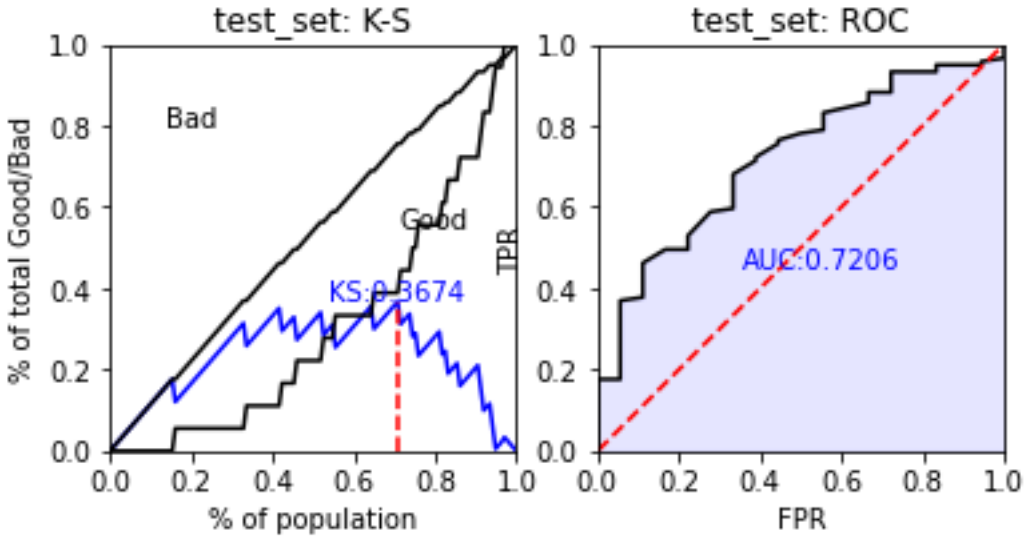

Figure 5: KS and ROC for Logistic Regression model, Subset 2

Gini and KS value	
Gini	**KS**
0.4412	0.3674

Table 6: Gini and KS value for Logistic Regression model, Subset 2

This concludes that this model, even though at first appears to be a decent one, with further investigation on the confusion matrix light is shined and shows a real problem that would have devastating effects on the bank.

Subset 2 Linear Regression

The coefficients of this Linear Regression model as seen in Table 7, appear to be normal, with the model giving most importance to 'Duration' and others not so much. The imbalanced dataset might again be the reason for this.

Coefficients			
Duration	Age	History	Property
0.197165	0.050762	0.090959	0.0716944

Table 7: Coefficients of Linear Regression Subset 2

The ROC curve's fatness as seen on Figure 6 seems good, with an AUC of 68.79% is decent and the Gini coefficient is at 0.55 indicating a good model [Table 8].

Going a step further and looking at the confusion matrix features [Appendix E], even though the model shows to be working properly, it can be seen that in actuality the model has not predicted even one correct "bad" customer. That means that every customer that wanted a loan was able to take it, with a bigger potential loss for the Bank. The recall rate for "bad" shows 0, while the recall rate for "good" shows 100%. This is not a good scenario, as the probability of losing money is usually a worse scenario than the probability of earning money.

This Linear Regression model is slightly worse than the Logistic Regression one for Subset 2, but in the bigger picture, both models are very bad and would be damaging to any organization deployed.

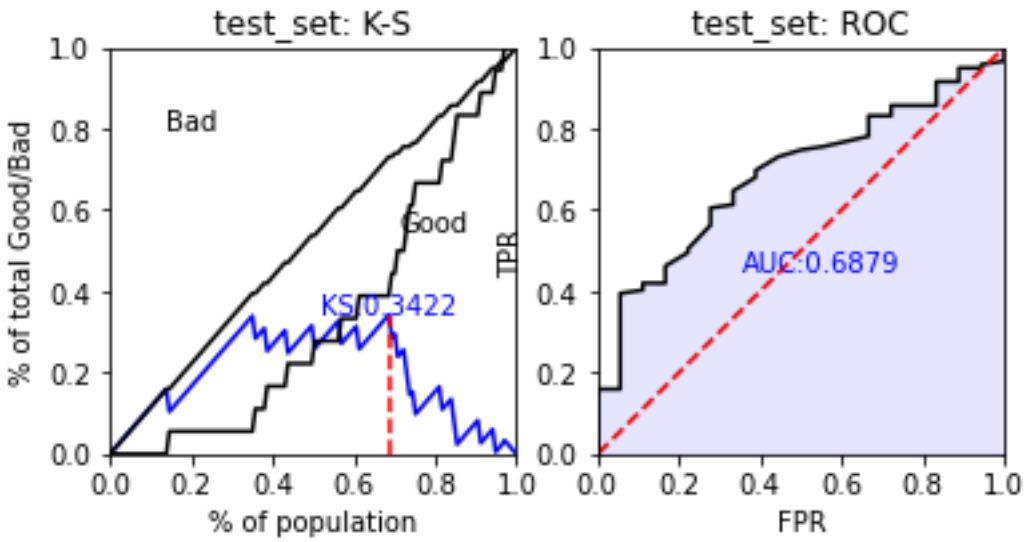

Figure 6: KS and ROC for Linear Regression model, Subset 2

Gini and KS value	
Gini	**KS**
0.5545	0.2912

Table 8: Gini and KS value for Linear Regression model, Subset 2

Conclusion

It appears that Logistic and Linear Regression perform similarly for this task, as can be seen especially through the results of Subset 1. The models of Subset 2, had a huge bias created by the imbalanced dataset and that could be seen through a closer look of their performance indexes, mainly an analysis on the confusion matrix, as the ROC curve and KS value were misleading. All in all, the importance of having a big quantity of varied data is shown clearly if the desired outcome is a good performing model.

APPENDIXES

APPENDIX A

Subset 1	
variable	**info_value**
Duration	0.3590413035930415
Age	0.3317427285042358
History	0.22404906302122768
Property	0.1963275385825361
Savings	0.15024214469288635
Purpose	0.12240403243614646
Coapp	0.09271757717479952
housing	0.06470541685906331
Installp	0.045971006514793966
marital	0.04435471495325051
Checking	0.04323506234656138
Foreign	0.0430282219693907
Emploed	0.03836481641663457
Resident	0.03172770085119685
Job	0.031217348749879085
Amount	0.022740426080794525
Existcr	0.016858987652347605
Other	0.014571905211650213
Depends	0.0031277563605213513
Telephone	0.000883005284804694

Subset 2	
variable	**info_value**
Duration	0.5679806902163772
Age	0.42149936482519207
Purpose	0.3953981574317986
Other	0.3100094640026229
Emploed	0.2761623253390293
History	0.21631919064518237
Checking	0.08471660220488032
Job	0.07541439905881175
Resident	0.06485546705808894
Savings	0.06398046571496506
marital	0.045835025035419724
Installp	0.038667993231183494
Amount	0.03723273975484364
Coapp	0.03621398218719188
housing	0.03223029932370784
Foreign	0.030643726997152388
Existcr	0.01875854320782458
Depends	0.016587440141060922
Telephone	0.011382010073799728

APPENDIX B

Logistic Regression Subset 1

Logistic Regression / Subset 1				
	precision	**recall**	**f1-score**	**support**
0	0.58	0.54	0.56	72
1	0.66	0.69	0.67	91
accuracy	0.63	163		
macro avg	0.62	0.62	0.62	163
weighted avg	0.62	0.63	0.62	163

Confusion Matrix	**Actual values**		
Predicted values		**Positive**	**Negative**
	Positive	TP->63	FP->28
	Negative	FN->33	TN->39

Recall (sensitivity) = TP/(TP+FN) = 63/(63+33) = 0.66

Specificity = TN/(FP+TN) = 39/(28+39) = 0.58

Bins and score			
	Variable	**Bin**	**Score**
0	Age	[-inf,23.0)	-6.0
1	Age	[23.0,26.0)	47.0
2	Age	[26.0,33.0)	-6.0
3	Age	[33.0,35.0)	53.0
4	Age	[35.0,42.0)	-55.0
5	Age	[42.0,48.0)	20.0
6	Age	[48.0,inf)	-11.0
7	Property	[-inf,2.0)	-39.0
8	Property	[2.0,3.0)	1.0
9	Property	[3.0,4.0)	8.0
10	Property	[4.0,inf)	40.0
11	Duration	[-inf,12.0)	-63.0
12	Duration	[12.0,22.0)	-12.0
13	Duration	[22.0,34.0)	15.0
14	Duration	[34.0,44.0)	34.0
15	Duration	[44.0,inf)	68.0
16	History	[-inf,2.0)	53.0
17	History	[2.0,3.0)	3.0
18	History	[3.0,4.0)	-12.0
19	History	[4.0,inf)	-33.0

APPENDIX C

Linear Regression Subset 1

Linear Regression / Subset 1				
	precision	**recall**	**f1-score**	**support**
0	0.58	0.54	0.56	72
1	0.66	0.69	0.67	91
accuracy	0.63	163		
macro avg	0.62	0.62	0.62	163
weighted avg	0.62	0.63	0.62	163

Confusion Matrix	**Actual values**		
Predicted values		**Positive**	**Negative**
	Positive	TP->63	FP->28
	Negative	FN->33	TN->39

Recall (sensitivity) = TP/(TP+FN) = 63/(63+33) = 0.66

Specificity = TN/(FP+TN) = 33/(28+33) = 0.58

APPENDIX D

Logistic Regression Subset 1

Logistic Regression / Subset 2				
	precision	**recall**	**f1-score**	**support**
0	0.17	0.06	0.08	18
1	0.87	0.96	0.91	119
accuracy	0.84	137		
macro avg	0.52	0.51	0.50	137
weighted avg	0.78	0.84	0.80	137

Confusion Matrix	**Actual values**		
Predicted values		**Positive**	**Negative**
	Positive	TP->114	FP->5
	Negative	FN->17	TN->1

Recall (sensitivity) = TP/(TP+FN) = 114/(114+17) = 0.87

Specificity = TN/(FP+TN) = 1/(5+1) = 0.17

Bins and score			
	Variable	**Bin**	**Score**
0	Other	[-inf,3.0)	97.0
1	Other	[3.0,inf)	-30.0
2	Age	[-inf,31.0)	18.0
3	Age	[31.0,35.0)	-2.0
4	Age	[35.0,37.0)	-56.0
5	Age	[37.0,46.0)	9.0
6	Age	[46.0,51.0)	-53.0
7	Age	[51.0,inf)	-21.0
8	Emploed	[-inf,3.0)	43.0
9	Emploed	[3.0,4.0)	11.0
10	Emploed	[4.0,5.0)	-40.0
11	Emploed	[5.0,inf)	-37.0
12	Purpose	[-inf,1.0)	17.0
13	Purpose	[1.0,2.0)	-83.0
14	Purpose	[2.0,3.0)	6.0
15	Purpose	[3.0,6.0)	-24.0
16	Purpose	[6.0,inf)	43.0

APPENDIX E

Linear Regression Subset 2

Linear Regression / Subset 2					
	precision	**recall**	**f1-score**	**support**	
0	0.00	0.00	0.00	18	
1	0.87	1.00	0.93	119	
accuracy	0.87	137			
macro avg	0.43	0.50	0.46	137	
weighted avg	0.75	0.87	0.81	137	

Confusion Matrix	**Actual values**		
Predicted values		**Positive**	**Negative**
	Positive	TP->119	FP->0
	Negative	FN->18	TN->0

Recall (sensitivity) = TP/(TP+FN) = 119/(119+18) = 0.868

Specificity = TN/(FP+TN) = 0/(0+0) = inf