Project: Creditworthiness – Jean Paul Barddal

Dear reviewer, please see the Jupyter Notebook that is located in the same repo as this report (https://github.com/jaycwb/udacity-bizanalytics-creditworthiness). There you will see the code I wrote to obtained the results presented here.

Step 1: Business and Data Understanding

Provide an explanation of the key decisions that need to be made. (250 word limit)

Key Decisions:

1. What decisions needs to be made? The primary goal of this project is to determine whether a customer qualifies or not to a loan.

2. What data is needed to inform those decisions? This is one of those cases where more data usually may lead to better insights onto whether customers should be trusted to pay their loans in due time. Initially, the bank provided us with a fair amount of attributes, that seem to encompass financial, work-related and housing regarding each customer. A detailed analysis will take place to determine if we have redundant and irrelevant data that should be removed, and otherwise, it will be used to train our predictive model.

3. What kind of model (Continuous, Binary, Non-Binary, Time-Series) do we need to use to help make these decisions? As stated in the project description, the manager is concerned to divide customers into two segments (classes): creditworthy and non-creditworthy. Therefore, we're dealing with a binary classification problem. Also, since we don't have a lot of historical data, I don't like the idea of working as a time-series (such as a data stream), since we don't have enough data to back up the claim that we might have concept drifts.

Step 2: Building the Training Set

Build your training set given the data provided to you. The data has been cleaned up for you already so you shouldn't ***need to convert any data fields to the appropriate data types.***

Here are some guidelines to help guide your data cleanup:

- *For numerical data fields, are there any fields that highly-correlate with each other? The correlation should be at least .70 to be considered "high". OK*
- *Are there any missing data for each of the data fields? Fields with a lot of missing data should be removed OK*
- *Are there only a few values in a subset of your data field? Does the data field look very uniform (there is only one value for the entire field?). This is called "low variability" and you should remove fields that have low variability. Refer to the "Tips" section to find examples of data fields with low-variability. OK*
- *Your clean data set should have 14 columns where the Average of Age Years should be 36 (rounded up) OK*

***Note**: For the sake of consistency in the data cleanup process, impute data using the average of the entire data field instead of removing a few data points. (100 word limit) OK*

***Note**: For students using software other than Alteryx, please format each variable as:*

Variable	**Data Type**
Credit-Application-Result	String
Account-Balance	String
Duration-of-Credit-Month	Double
Payment-Status-of-Previous-Credit	String
Purpose	String
Credit-Amount	Double
Value-Savings-Stocks	String
Length-of-	String

current-employment	
Instalment-per-cent	Double
Guarantors	String
Duration-in-Current-address	Double
Most-valuable-available-asset	Double
Age-years	Double
Concurrent-Credits	String
Type-of-apartment	Double
No-of-Credits-at-this-Bank	String
Occupation	Double
No-of-dependents	Double
Telephone	Double
Foreign-Worker	Double

To achieve consistent results reviewers expect.

Answer this question:

1. In your cleanup process, which fields did you remove or impute? Please justify why you removed or imputed these fields. Visualizations are encouraged.

Original statistics about the dataset

	Duration-of-Credit-Month	Credit-Amount	Instalment-per-cent	Duration-in-Current-address	Most-valuable-available-asset	Age-years	Type-of-apartment	Occupation	No-of-dependents	Telephone	Foreign-Worker
count	500.00000	500.000000	500.000000	156.000000	500.000000	488.000000	500.000000	500.0	500.00000	500.000000	500.000000
mean	21.43400	3199.980000	3.010000	2.660256	2.360000	35.637295	1.928000	1.0	1.14600	1.400000	1.038000
std	12.30742	2831.386861	1.113724	1.150017	1.064268	11.501522	0.539814	0.0	0.35346	0.490389	0.191388
min	4.00000	276.000000	1.000000	1.000000	1.000000	19.000000	1.000000	1.0	1.00000	1.000000	1.000000
25%	12.00000	1357.250000	2.000000	2.000000	1.000000	27.000000	2.000000	1.0	1.00000	1.000000	1.000000
50%	18.00000	2236.500000	3.000000	2.000000	3.000000	33.000000	2.000000	1.0	1.00000	1.000000	1.000000
75%	24.00000	3941.500000	4.000000	4.000000	3.000000	42.000000	2.000000	1.0	1.00000	2.000000	1.000000
max	60.00000	18424.000000	4.000000	4.000000	4.000000	75.000000	3.000000	1.0	2.00000	2.000000	2.000000

- The first thing I did was to remove features with low variability (< 25% to be specific). This process removed the following features ['Guarantors', 'Occupation', 'No-of-dependents', 'Foreign-Worker'].
- My strategy to remove attributes with a lot of missing data is simple: remove all attributes with more than 50% of the inputs are missing. This process removed the following features: ['Duration-in-Current-address', 'Age-years'].

```
Account-Balance                     0
Duration-of-Credit-Month            0
Payment-Status-of-Previous-Credit   0
Purpose                             0
Credit-Amount                       0
Value-Savings-Stocks                0
Length-of-current-employment        0
Instalment-per-cent                 0
Guarantors                          0
Duration-in-Current-address       344
Most-valuable-available-asset       0
Age-years                          12
Concurrent-Credits                  0
Type-of-apartment                   0
No-of-Credits-at-this-Bank          0
Occupation                          0
No-of-dependents                    0
Telephone                           0
Foreign-Worker                      0
Credit-Application-Result           0
dtype: int64
```

- All of the remainder numeric attributes are imputed used the mean of the existing values.

Correlation matrix for the cleaned data set

Step 3: Train your Classification Models

First, create your Estimation and Validation samples where 70% of your dataset should go to Estimation and 30% of your entire dataset should be reserved for Validation. Set the Random Seed to 1. (PLEASE SEE JUPYTER NOTEBOOK)

Create all of the following models: Logistic Regression, Decision Tree, Forest Model, Boosted Model

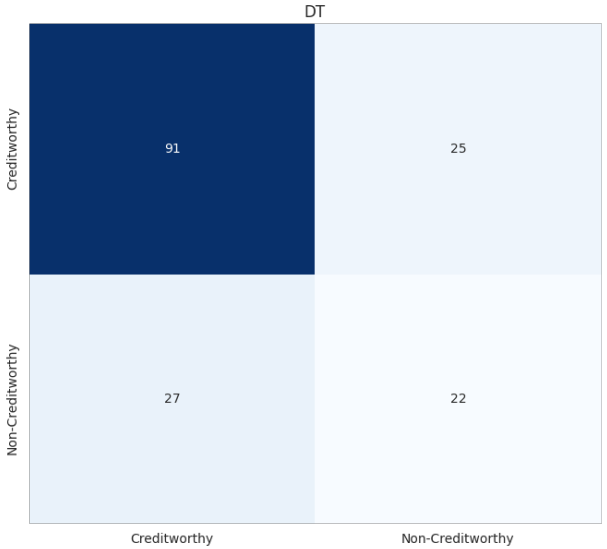
*Answer these questions for **each model** you created:*

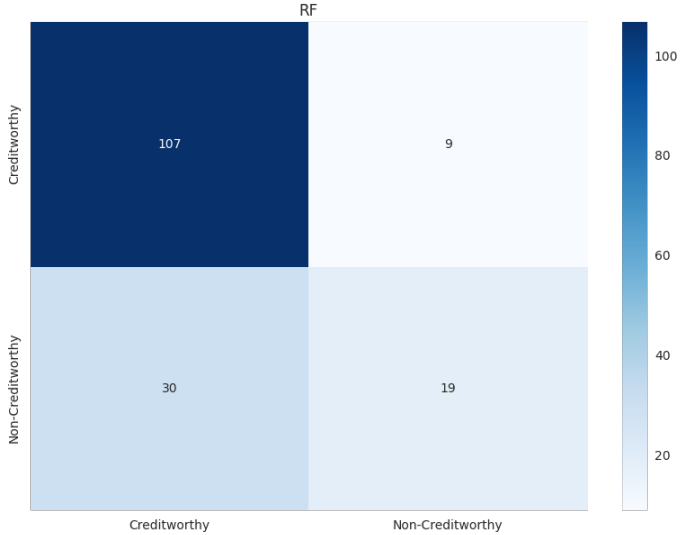
1. Which predictor variables are significant or the most important? Please show the p-values or variable importance charts for all of your predictor variables. The p-values for all features are presented in the following table with the exception of SGD, where I presented the coefficients. For both Decision Tree and Random Forests, all variables are important, since all have non-zero values, but Credit Amount is by far the most important. These results show the impact of classification bias: for instance, the tree-based models (Decision Tree and Random Forest) present almost the same importance values for all attributes, while the values presented for AdaBoost are somewhat different. This way, the same discussion on which attributes are important for the Decision Tree hold for the Random Forest classifier. By analyzing the coefficients of SGD, we can see that the biggest absolute value is also for credit amount, and thus, it should be considered the most important attribute. Finally, for Adaboost, it follows that only the credit-amount and the 'Duration-of-Credit-Month' variables were used, and thus, should be considered important. In my opinion, further exploration should be performed here to evaluate feature selection based on wrappers to find an optimal subset of features for this problem.

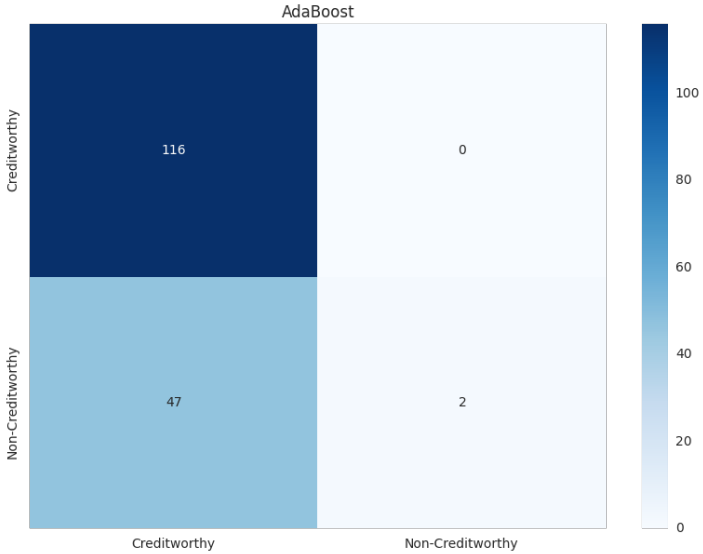
2. Validate your model against the Validation set. What was the overall percent accuracy? Show the confusion matrix. Are there any bias seen in the model's predictions? The accuracy obtained by the models are as follows: Decision Tree obtained 68.48%, Random Forest obtained 76.36%, AdaBoost obtained 71.51% and SGD obtained 70.30% (which is the ZeroR baseline since it predicted all instances as creditworthy (highly biased!) – please see the confusion matrix below.) The confusion matrices corroborate the results presented, where a deeper analysis between Random Forests and AdaBoost must occur to verify if the manager prefers a model that is more or less restrictive wrt false positives. Below the AUROC curves are presented, showing the trade-off between the true positive and false positive rates for all classifiers. Again, the Random Forest obtains the best results, followed by AdaBoost.

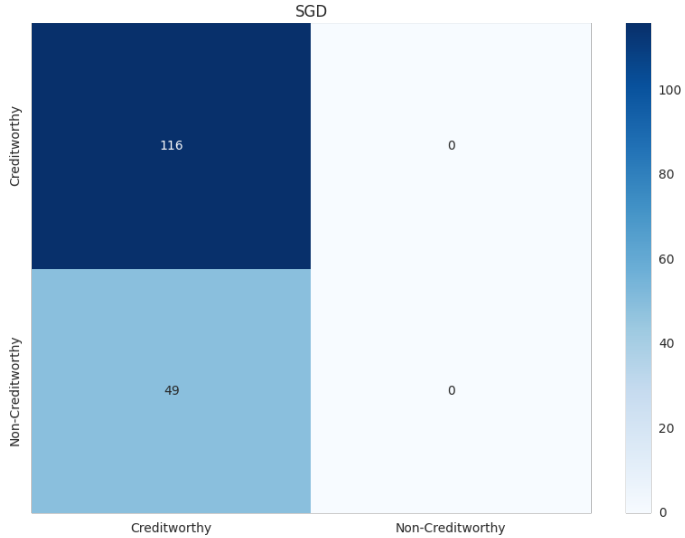
You should have four sets of questions answered. (500 word limit)

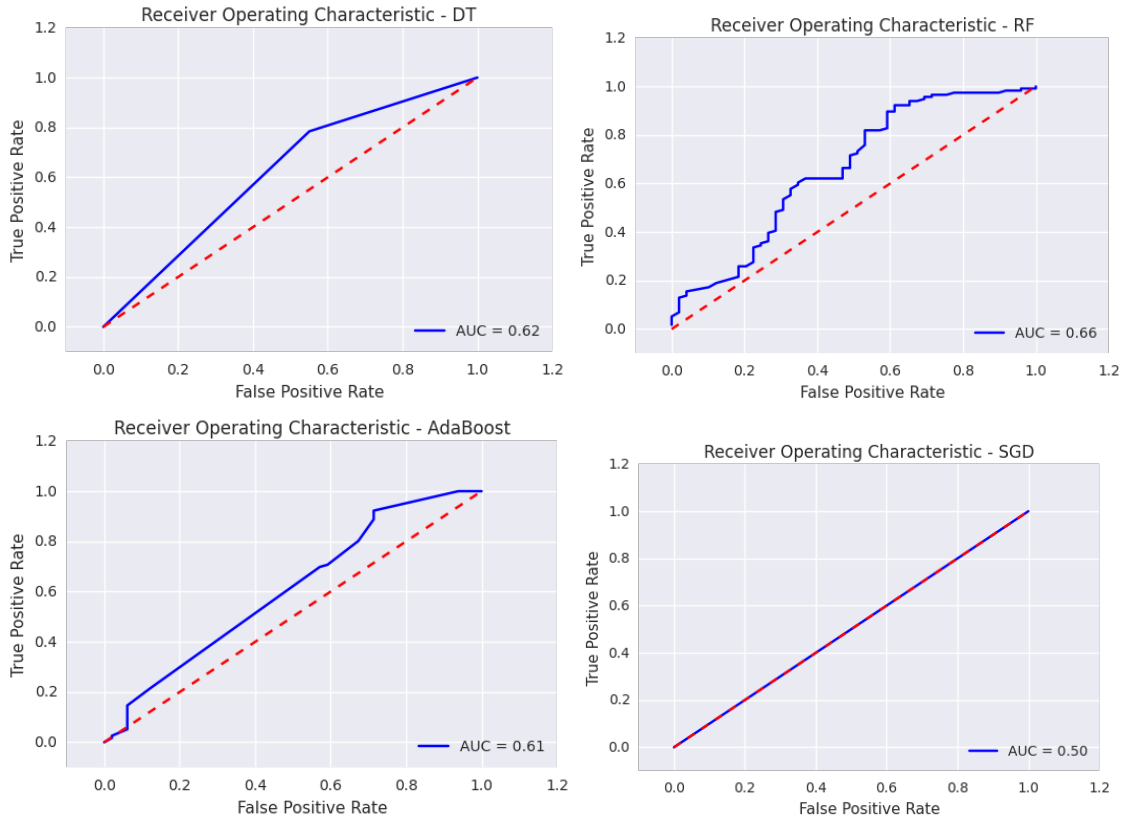
Topic		*Logistic Regression (SGD)*	*Decision Tree (DT)*	*Random Forest (RF)*	*AdaBoost*
Variables and their importance	*'Duration-of-Credit-Month'* *'Credit-Amount'* *'Instalment-per-cent'* *'Most-valuable-available-asset'* *'Type-of-apartment'* *'Telephone'*	*(Coefficients)* *-2385.93866866* *-9197.83096485* *-1185.48990277* *-779.73074046* *-905.01121915* *-643.23111444*	 *0.22007894* *0.55176798* *0.07542275* *0.11714853* *0.02746515* *0.00811664*	 *0.25295143* *0.46010023* *0.09215148* *0.09684417* *0.05331514* *0.04463756*	 *0.6* *0.4* *0.* *0.* *0.* *0.*
Accuracy obtained		*70.30%*	*68.48%*	***76.36%***	*71.51%*


DT
Creditworthy
Non-Creditworthy
91
25
27
22
Creditworthy
Non-Creditworthy
90
75
60
45
30


RF
Creditworthy
Non-Creditworthy
107
9
30
19
Creditworthy
Non-Creditworthy
100
80
60
40
20


AdaBoost
Creditworthy
Non-Creditworthy
116
0
47
2
Creditworthy
Non-Creditworthy
100
80
60
40
20
0


SGD
Creditworthy
Non-Creditworthy
116
0
49
0
Creditworthy
Non-Creditworthy
100
80
60
40
20
0



Step 4: Writeup

Decide on the best model and score your new customers. For reviewing consistency, if Score_Creditworthy is greater than Score_NonCreditworthy, the person should be labeled as "Creditworthy"

Write a brief report on how you came up with your classification model and write down how many of the new customers would qualify for a loan. (250 word limit)

Answer these questions:

1. Which model did you choose to use? Please justify your decision using only the following techniques:

a. Overall Accuracy against your Validation set
b. Accuracies within "Creditworthy" and "Non-Creditworthy" segments
c. ROC graph
d. Bias in the Confusion Matrices

The best accuracy was obtained by Random Forest with a value of 78.78%. However, if we analyze the results presented in the table and confusion matrices above, we will see that all classifiers have problems with predicting the minority class (non-creditworthy customers). This shows that all classifiers are strongly biased towards this class, being SGD the extreme case,

where all the instances in the validation set were classified as creditworthy. Naively, I would suggest going with the most accurate model (Random Forest), since it is able to correctly classify a few non-creditworthy customers. The validation is critical and should be performed with specialists, since the bank manager is the one that will be able to tell if the risk of providing loans for non-trustworthy customers surpasses more restrictive models, such as the single Decision Tree, which is able to correctly predict more non-trustworthy customers. A way to explain this to the manager would be through ROC curves, which are presented below. In this case, we can see that the Random Forests also achieves slightly greater AUC results, showing the best trade-off between true positive and false positive rates.

Note: Remember that your boss only cares about prediction accuracy for Crediyworth and Non-Creditworthy segments.

2. How many individuals are creditworthy?

Assuming the Random Forest classifier: the number of creditworthy customers in the validation set is 420 and non-creditworthy is 80.

Before you Submit

Please check your answers against the requirements of the project dictated by the rubric here. Reviewers will use this rubric to grade your project.