Classifying Credit Card Loan Repayment

Ahilan Subbaian and Ahmed Bawla

Stevens Institute of Technology

Author Note

Thank you to Professor German Creamer for helping us on the project

Table of Contents

Classifying Credit Card Loan Repayment

In the digital age, the definition of money has moved past physical coins and bills to numbers on a computer. By convention the Federal Reserve increases the money supply by minting new coins and bills. However, with the automation of the financial market, the Federal Reserve creates money by simply digitally debiting and crediting to major banks. This whole process can happen in mere seconds.

Purchasing and debt holding has come a long way from storing cash in wallet or collecting change in piggy banks. Credit Cards have widely substituted these archaic forms of money transfer/collecting. Conventional money exchanges occur on the spot with each purchase following the transfer of cash. Credit Cards keep all purchases on debt and are paid to the bank at the end of the month. By paying at the end of the month, Banks take on the risk of the purchase on to themselves.

With thousands of credit card holders, Banks can end up taking significant risk. In the current market, United States owes almost $1 Trillion in credit card debt. Banks need to establish a method of classifying individuals that are trustworthy to lend their money to. Banks need to evaluate prior history and the current financial situation of applicants to make their choices. This problem is an opportunity to see if machine learning techniques can efficiently classify individuals that or trustworthy or not trustworthy.

Business Understanding

Banks lend out trillions every year to customers to purchase goods and items. Effective modeling strategies are required to make sure they get their money back. To apply for credit cards, customers must go through a rigorous screening process and provide financial information and personal information. I propose using a classification machine learning model that can

efficiently classify individuals as credit card worthy or not worthy. By creating a model, Banks have a quantitative means of selecting candidates that may be more reliable than interviews or qualitative screens.

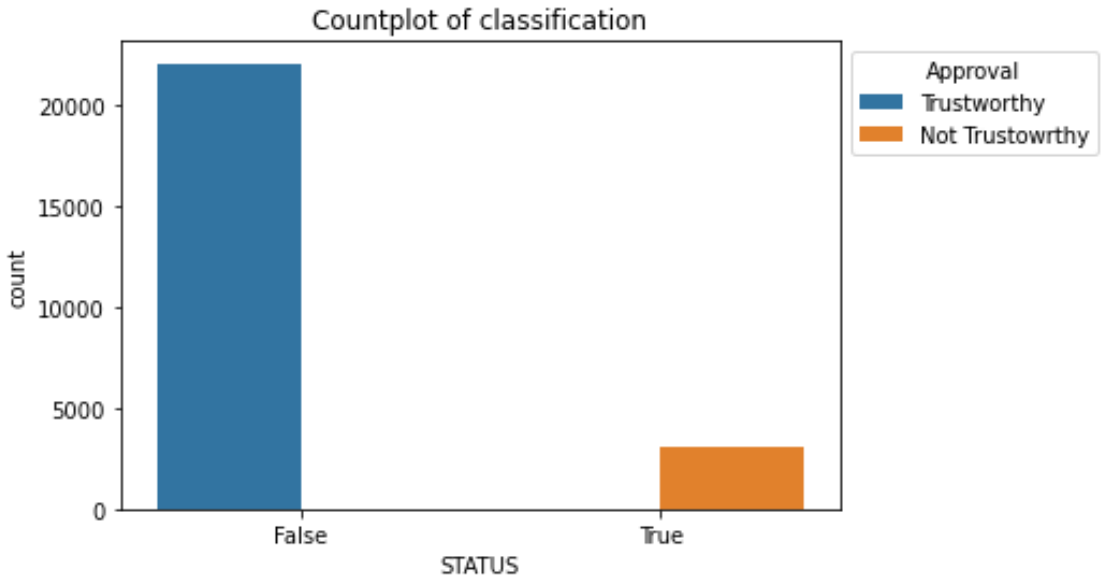
Data

Understanding the Data

Access to client data is clearly confidential, for this project I will be using data pulled from Kaggle. The data is published by Xiao Song on Kaggle, titled Credit Card Approval Prediction, the link can be found in references [1]. The data is provided as two csv files, application_record.csv and credit_record.csv. 'application_record.csv' provides information on applicants personal and financial information and 'credit_record.csv' maps Applicant Identification to their payback history.

Data Preparation

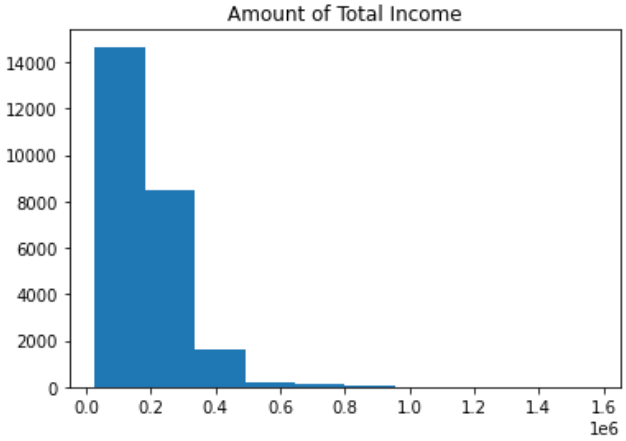
My preparation for 'credit_record.csv' entailed converting the multi-class problem of classifying how long it took applicants to pay back their credit card loan to if they ever took extra time to pay back the loan. This csv maps application ID to each month they used the credit card and how long it took to pay back the loan. Since this information is multi-class it was necessary to screen the data by classifying each applicant as True if they ever took longer than the posted deadline to pay back the loan and false if they always paid on time. I did this using by creating a new column that classified each payment as True or False by the aforementioned rules and then grouping the data frame by applicant ID.



My preparation for ‘application_record.csv’ was done by checking for any NaN values in the dataframe. The column ‘Occupation_type’ had several NaNs in the column. While this may mean the applicant did not have a job it may also be that the applicant chose not to provide this information. The author neglects to inform what the true meaning of this value is, because it is ambiguous, I chose to drop all rows that had NaNs.

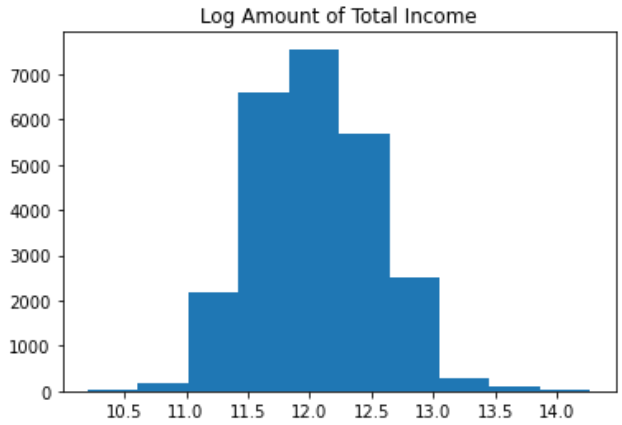
The two csv files need to be combined into one file that holds information for each applicant and the label. This can efficiently be done by merging the two data frames as on applicant ID.

Feature Engineering



The initial data did require modifications of features to perform more accurate calculations. By graphing the 'AMT_INCOME_TOTAL' we can visually see that the range goes from 0 – 1.6 e6, with a right tail.

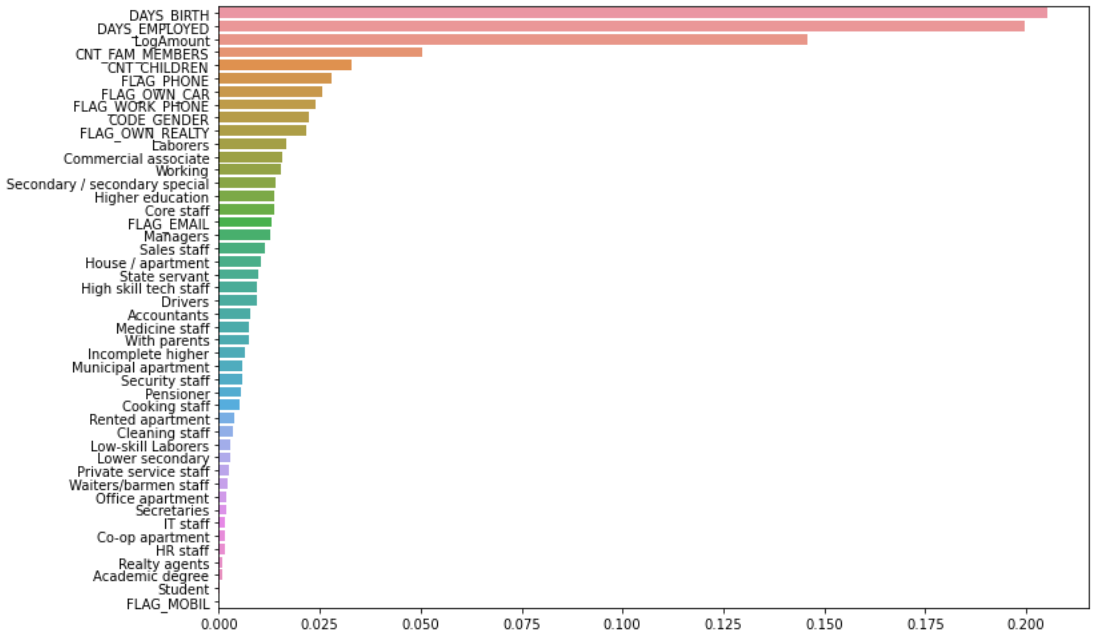
A distribution that is very skewed can significantly effect how models scale that feature. It is necessary to reformat the total income as log(total income). The new plot, shown below, has a cleaner graph with a normal distribution.



Several columns from the original dataset are categorical. Models like SVC, Logistic Regression and many others are unable to use qualitative information. These features need to be transformed into one-hot vectors that converts the qualitative information into separate columns that represent distinct values. The columns that need to be refitted are Income Type, Education Type, Family, Housing, Occupation. This transformation can be simply achieved by calling pandas.get_dummies().

Feature Selection

The next step is to select features that will contribute the most to out model. By using a Random Forest Classifier, I can identify which features are the most important in classifying applicants as trustworthy or not trustworthy. The model is able to classify which features are the most useful by calculating the change in entropy provided by each feature.



By examining the graph, we can identify that 'FLAG_MOBILE' which represents if the applicant has a mobile device or does not own a mobile device is not very significant. The feature 'FLAG_EMAIL' is similar to 'FLAG_MOBILE'. This feature is able to provide ~2%; however, since an email is easy obtained for free and may not be the most predictive, I chose to remove this feature. For ethical reasons I chose to remove 'CODE_GENDER' from the model because an applicant's gender does not correlate to their ability to meet deadlines.

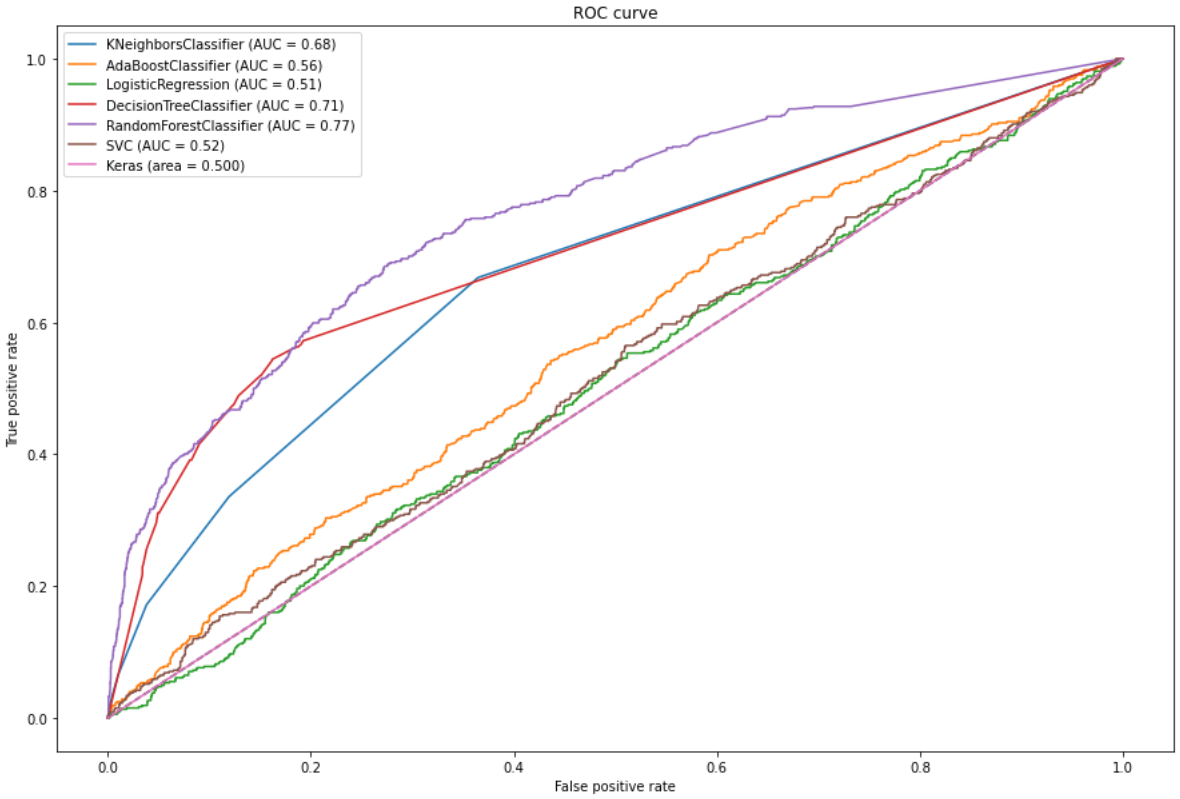
Modeling

The models that I chose is Logistic Regression, Decision Tree, Support Vector Machine (SVM), Ada Boost Classifier, K Nearest Neighbors Classifier and Tensorflow Sequential.

- Logistic Regression - statistical model that in its basic form uses a logistic function to model a binary dependent variable, estimates the parameters of a logistic model
- Decision Tree - flowchart-like structure in which each internal node represents a "test" on an attribute each
- SVM - supervised machine learning model that uses classification algorithms for two-group classification problems.
- Ada Boost Classifier - meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset
- KNeighborsClassifier - Classifier implementing the k-nearest neighbor's vote
- Tensorflow Sequential – Deep Learning method that uses layers and nodes to perform a binary classification

Evaluation

To finalize the model that is able to perform the best on the data we must use cross validation and establish a model that performs the best on average. This allows us to select the model that performs the best with significant reason. Below I have graphed the Receiver Operating Characteristic (ROC) which shows the relationship between True Positive Rate and False Positive Rate, the Area under the Curve (AUC) is shown at the key. The information is graphed on the validation set.



The three models that appear to perform the best are K Nearest Neighbors, Decision Tree and Random Forest. The lines for the other model are near to the diagonal line and have AUC values close to .5. These models (Ada Boost Classifier, Logistic Regression, SVC and Keras) do not appear to perform very well on the validation data. Distinguishing between K Nearest Neighbors, Decision Tree and Random Forest requires using examining the average and standard deviation of the performance of the models.

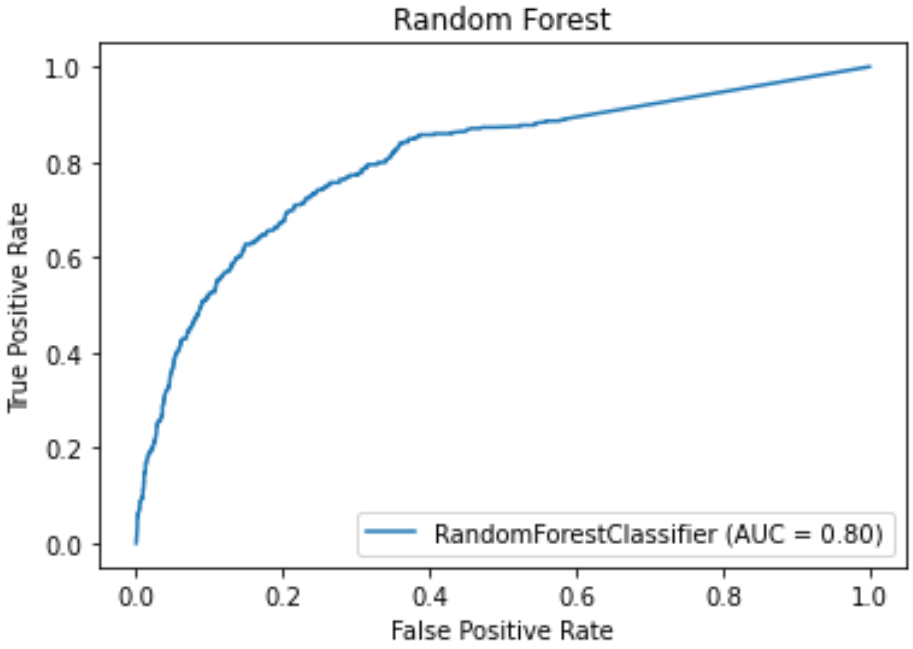
	Accuracy Mean	Accuracy Standard Deviation	AUC Mean	AUC Standard Deviation
Logistic	0.874064	0.002867	0.500000	0.000000
Decision Tree	0.865051	0.005342	0.625289	0.009935
Random Forest	0.875819	0.003666	0.624666	0.009766
SVM	0.874064	0.002867	0.500000	0.000000
Adaboost	0.874356	0.002529	0.502734	0.002055
KNN	0.864466	0.003607	0.565242	0.009561
Sequential	0.874064	0.002867	0.500000	0.000000

By examining each model's AUC Mean we can eliminate KNN as an effective model. The two models with the highest AUC Mean are Decision Tree and Random Forest. Random Forest has a higher average, but by the AUC Standard Deviation this difference is not conclusive. However, if we examine the Accuracy Mean for these two models, the Random Forest is significantly higher than the Decision Tree. In addition, the Accuracy Standard Deviation of the Random Forest is very minimal. Knowledge of the Random Forest algorithm confirms that it may perform better than a Decision Tree since Random Forest is simply a list of multiple Decision Trees. By looking at the ROC Curve, Accuracy and AUC we can conclude that Random Forest is the most predictive model.

Conclusion

The final model has been selected as Random Forest because it outperformed the other models in ROC curve, Accuracy and AUC. Finally, the Random Forest model must be tested on the testing data that has been put aside. This can be useful for evaluators who can use the Random Forest to look at which features are most predictive.

Model Performance:

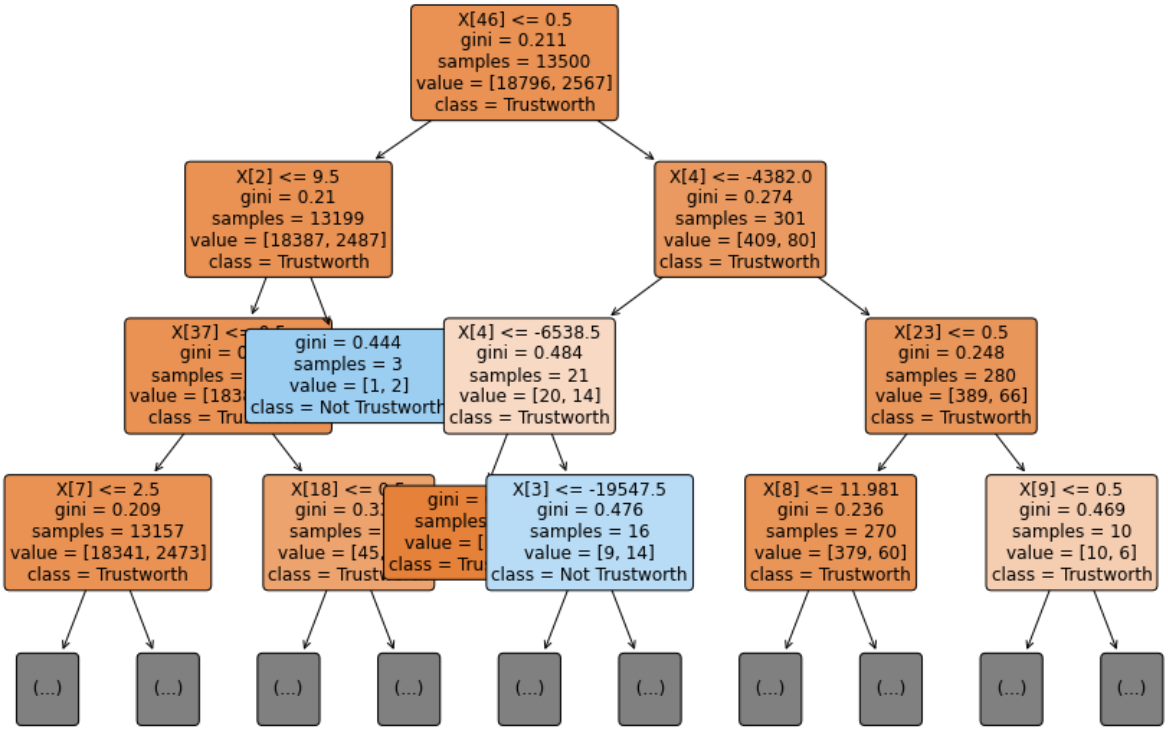
- Accuracy – 88.07%
- Area Under Curve – 64.26%
- 

This model can be deployed by banks to perform a cursory screening for identifying if credit card applicants will be able to meet deadlines. The model is only able to achieve an AUC of 63.30% and cannot be relied on as a final screening. The model can be improved provided more data and more efficient models for unbalanced data as we can see in this data set.

Ethical Considerations that must be made is Gender, which is represented as 'CODE_GENDER'. Credit Card Applicants should not be discriminated for their gender as it

does not pertain to their ability to meet deadlines. The risks of the model are its inaccuracies with classifying applicants. This can be solved by only using the model as an initial screening and the bank should continue to screen applicants.

As the final model is a Random Forest Classifier, it is composed of multiple Decision Trees. I have plotted four layers of the first decision tree below for further visualization.



References

Comoreanu, Alina. “2020 Credit Card Debt Study: Trends & Insights.” *WalletHub*, 8 Dec. 2020, wallethub.com/edu/cc/credit-card-debt-study/24400.

Ross, Sean. “Understanding How the Federal Reserve Creates Money.” *Investopedia*, Investopedia, 22 Sept. 2020, www.investopedia.com/articles/investing/081415/understanding-how-federal-reserve-creates-money.asp.

Song, Xiao. “Credit Card Approval Prediction.” *Kaggle*, Mar. 2020, www.kaggle.com/rikdifos/credit-card-approval-prediction.

Appendix

Ahilan Subbaian – Completed code and worked on paper

Ahmed Bawla – Worked on code and paper