SMALL BUSINESS ADMINISTRATION LOAN DATA – AN ANALYSIS ON THE RISK OF NON-PAYMENT

Kevin Arnold, Brittany Roberts, and Sebastian Rolett

November 18, 2023

INTRODUCTION

This report focuses on the analysis of United States Small Business Administration (SBA) 7(a) loan data[i], extracted from the SBA 504 FOIA and 7(a) Data provided by the United States Small Business Administration. The dataset encompasses data divided into decade-long intervals, with the most recent update as of 09/30/2023, and quarterly updates expected. The objective of this analysis is to answer critical research questions and understand borrower characteristics, risk factors, and lending practices to enhance risk management and loan allocation, ultimately benefiting various stakeholders.

BUSINESS PROBLEM/HYPOTHESIS

Our hypothesis is that it may be possible to determine which loans have a higher probability of default based on certain indicators contained within the data captured by the SBA. The intention of this analysis is to answer critical research questions and understand borrower characteristics, risk factors, and lending practices to enhance risk management and loan allocation, ultimately benefiting various stakeholders.

METHODS/ANALYSIS

Our goal is to determine who will default on these SBA loans versus who will be able to pay them off in full. As such, this is a binary classification problem where we will use the paid in full and charged off data as our target with the remaining relevant data points as our features.

The first step involves merging and cleansing the data downloaded from the SBA website to rid it any unnecessary data and unify the dataset. This step also involves transforming certain datapoints like the legal entity type and the age of the business applying for these loans from categorical to numerical to make them usable in our model of choice.

The next step is to run the features and target through an initial logistic regression model and analysis the results.

RESULTS

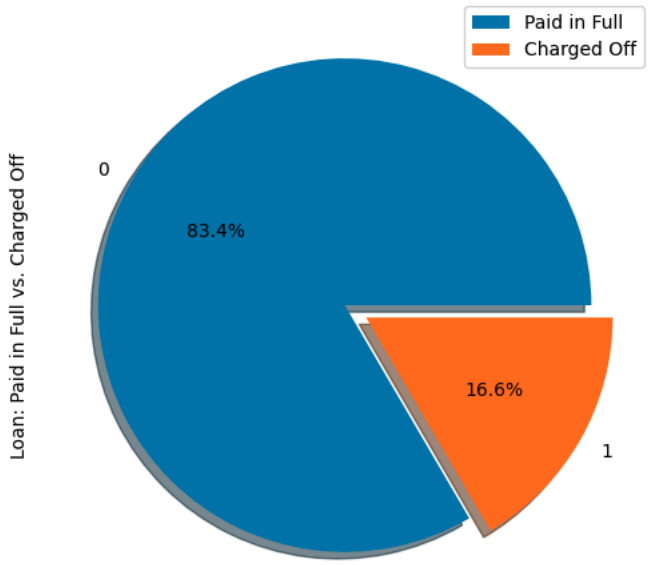
Below is the feature set of data points used in the model:

- Gross Approval amount
- Fiscal Year of the approval
- Initial Interest Rate
- Loan Term (in months)
- Age of the Business
- The number of jobs the business supports
- Business Type

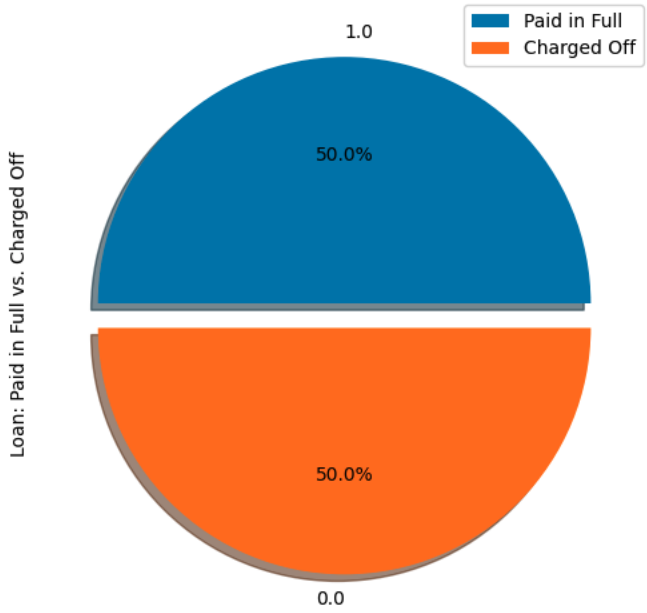
The initial run of the model with a train/test split of .3 on the entirety of the data presents an accuracy score of 83%.

	precision	recall	f1-score	support
0	0.40	0.02	0.04	63526
1	0.84	0.99	0.91	318330
accuracy			0.83	381856
macro avg	0.62	0.51	0.47	381856
weighted avg	0.76	0.83	0.76	381856

While this score is excellent to begin with, the related precision and recall scores don't perform the best. This is most likely due to the imbalance in our training data. Below is the chart showing the split in our data used for training the model where the loan resulted in being paid in full versus being charged off.



Seeing this imbalance, two different techniques to solve for this are deployed. The first is called 'Synthetic Minority Over-sampling Technique which will balance out the data used to train our model. When using the SMOTE package from the 'imblearn.over_sampling' library[ii], our training data is transformed to be evenly distributed as shown below.



When running this newly distributed data to train our model, the resulting output from the test data is as follows:

	precision	recall	f1-score	support
0	0.35	0.79	0.49	63526
1	0.94	0.71	0.81	318330
accuracy			0.72	381856
macro avg	0.65	0.75	0.65	381856
weighted avg	0.85	0.72	0.76	381856

The precision, recall, and f1-scores all increase all while maintaining a relatively high accuracy score of 72%.

The same process is now run with the under-sampling module called 'NearMiss' which again comes from the 'imblearn' library[iii]. This under-sampling of the training data again results in an even 50/50 distribution of loans that are paid in full versus loans that are charged off, however, the increases in precision, recall, and f1-score aren't as prominent all while this version of the model's accuracy is further decreased to 59%:

	precision	recall	f1-score	support
0	0.23	0.63	0.34	63526
1	0.89	0.58	0.70	318330
accuracy			0.59	381856
macro avg	0.56	0.61	0.52	381856
weighted avg	0.78	0.59	0.64	381856

As a result, the second model resulting from an over-sampling using SMOTE proves to be our best model

CONCLUSION

While the results of our second model prove to promising to begin with, multiple issues have arise during the process of examining the data. As seen in both the data preprocessing and model building workbooks, there is not much correlation between any of the features. While this helps to disprove issues around the presence of multicollinearity, there is still not much in the way of statistical inference that may be drawn between any of the features and why an

individual or corporation with a SBA loan defaults on the loan itself. After running a quick logarithmic comparison on the importance of our features, the resulting dataframe appears which confirms that no feature stands out as playing a large role in our model.

	Features	Importance
3	TermInMonths	1.018808
5	JobsSupported	1.005174
6	BusinessType_CORPORATION	1.000484
7	BusinessType_INDIVIDUAL	1.000133
4	BusinessAge	1.000133
8	BusinessType_PARTNERSHIP	1.000096
0	GrossApproval	0.999999
2	InitialInterestRate	0.999511
1	ApprovalFiscalYear	0.999317

A conclusion can be derived that building a model to accurately predict whether or holder of a SBA will default on their payment based solely on the historical data found on the SBA's website may prove difficult. Without joining additional profile data on the individuals and companies obtained elsewhere and further feature engineering that would require a significant time commitment, the performance of our model may be hindered and thus rendered untrustworthy by stakeholders looking to utilize it.

REFERENCES

[i] https://data.sba.gov/dataset/7-a-504-foia

[ii] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

[iii] https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.NearMiss.html