Gopi Krishna Nowduri
May 07, 2024

Capstone 2 Final Report: Predicting Defaults for Small Business Loans

Problem

The Small Business Association (SBA) is a United States government agency which promotes small business by guaranteeing a portion of certain small business loans. If the borrower defaults on an SBA-guaranteed loan, the SBA will pay a portion of the balance. Lenders, however, still face serious risk when approving an SBA-backed loan. The goal of this project is to build a model which predicts whether an SBA-guaranteed loan will be repaid. The potential client is a bank which lends money to small businesses. Equipped with a data-driven risk assessment model, the client could maximize profit by focusing on lower-risk loans and avoiding financial losses from defaults.

The data for this project is from the SBA, and contains information on over 899,000 small business loans. The data was retrieved and partially cleaned by professors Min Li, Amy Mickel and Stanley Taylor, for use as a case study in undergraduate and graduate level statistics courses, and uploaded to Kaggle by Mirbek Toktogareav (Li et al., Toktogareav). I used this data to train random forest and gradient boosting models, and tuned hyperparameters for both algorithms using grid search. The final model was a gradient boosting classifier with optimized hyperparameter values, which gave an f1-score of 0.85 on the test set.

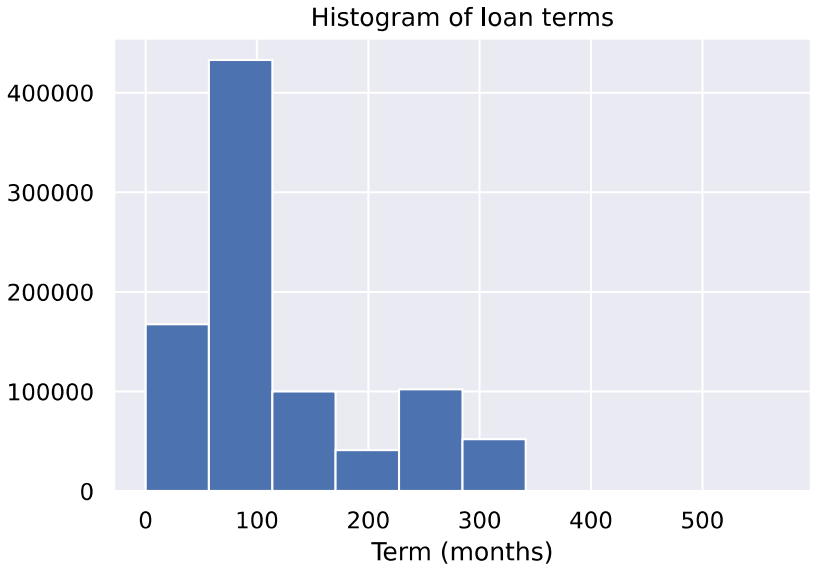
Data Wrangling

This data set contains twenty-seven variables, which describe the company applying for the SBA loan, and the specifics on the loan application itself. Several columns had entries that appeared to represent missing values or data entry errors. In most cases, I simply replaced these with NaN, Python's symbol for "missing value." The exception was the column RevLineCr, where about a quarter of the values were missing or ambiguous. I chose to drop this column. According to the code book, FranchiseCode values of 0 and 1 both mean "no franchise." To eliminate redundancy, I replaced 1's in this column with 0's.

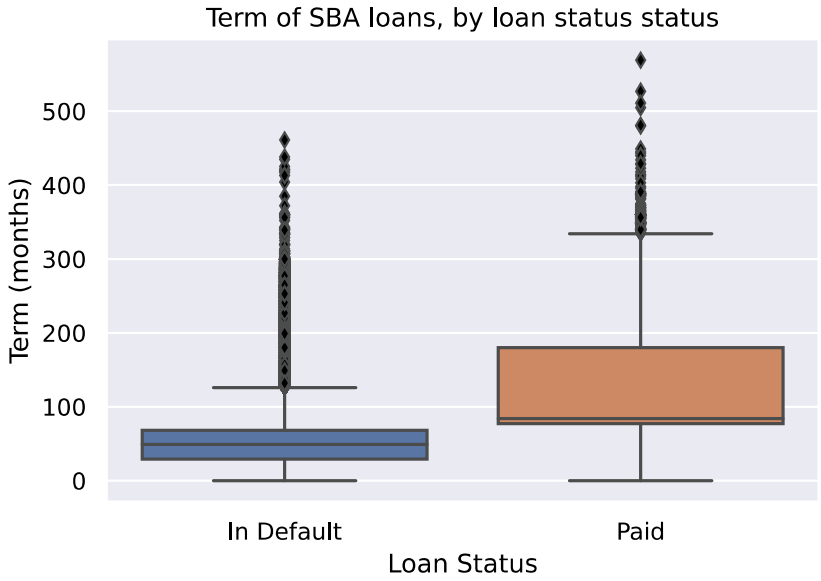
The earliest default date in the data set is in 1988, and the latest is in 2014. I discarded all data from before 1980, and after 2014. The resulting cleaned data frame had 898080 rows and 26 columns.

Exploratory Data Analysis

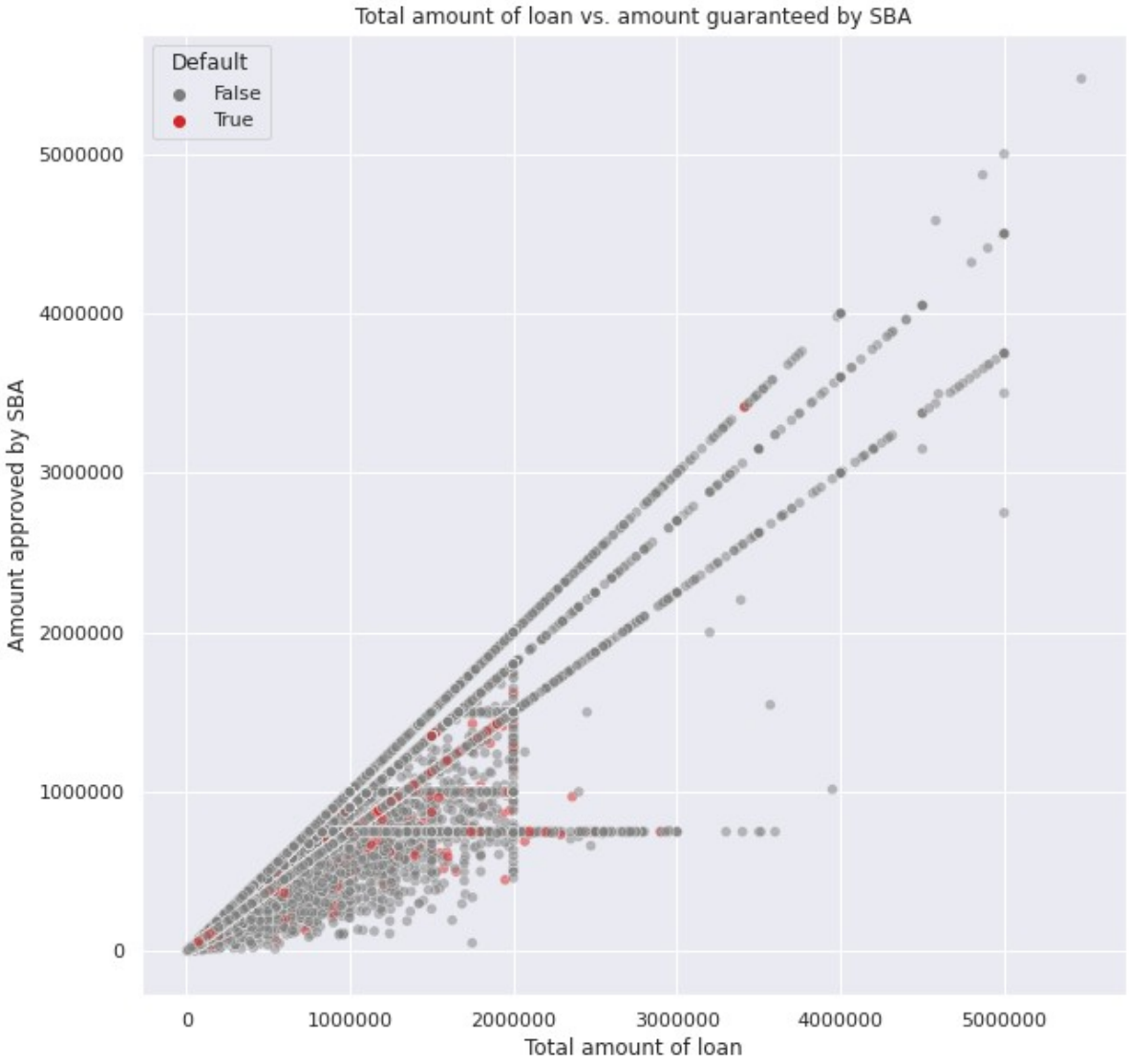
I compared default rates for different values of the categorical predictors using tables. I plotted histograms of numerical variables, and used boxplots to compare the distribution of values for loans that were paid in full, versus those that went into default. Most numerical variables were dramatically skewed right, with a small number of very high values. See the histogram of term below for an example. While it is possible that some of the large values of these variables represent errors, skewed distributions are not uncommon when dealing with financial data.



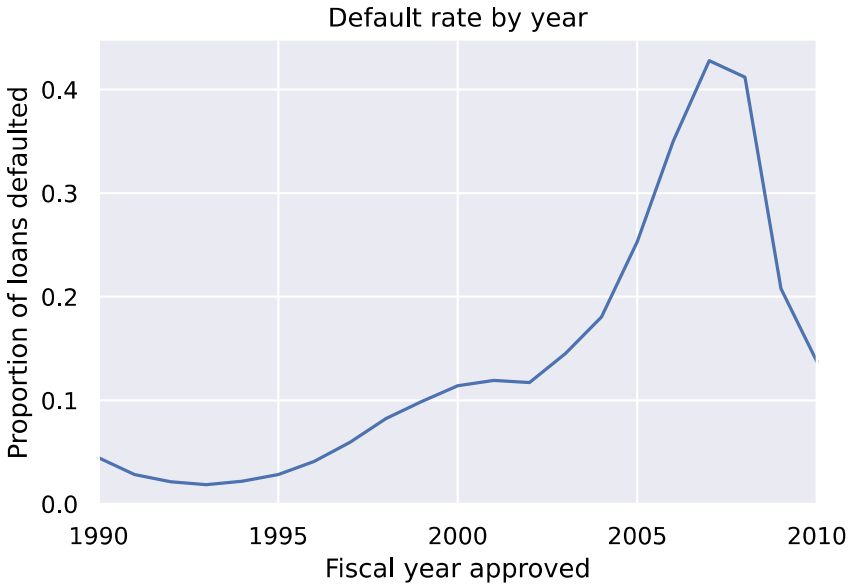
The variable with the strongest apparent relationship to default rate was term, which represents loan term in months. As seen in the plot, loans that were paid in full tended to have much larger values of term than ones that went into default.



I observed that paid-off loans tended to have larger total loan amounts, and larger amounts guaranteed by the SBA. When checking correlations between numerical variables, I found the correlation between total loan amount as SBA-guaranteed amount was 0.97, suggesting these two features might be redundant. However, the plot below tells a different story. When we plot total loan amount versus amount guaranteed by the SBA, we see that defaults are more likely as one moves away from the 45-degree line on the plot. That is, loans where the *fraction* guaranteed by the SBA is smaller are more likely to go into default.



Finally, I observed that default rates varied tremendously by year, as shown in the plot below. This is not surprising. During challenging economic times, I would expect to see more defaults. However, this presents a challenge for the model. The feature year is a proxy for overall economic conditions over the life of the loan. To achieve comparable accuracy on data from future years, I would need to incorporate some forecast of economic conditions into the model.



Feature Engineering

At this stage, I restricted to a subset of columns that are suitable predictors. I dropped high cardinality categorical features that would present challenges for dummy including, such as City and Bank. I replaced the column NAICS, which gives 6-digit industry classification codes, with a column Industry containing the first two digits from NAICS codes; these two digits give the general sector an industry belongs to. I also dropped features that pertain to events after the loan is approved, such as RetainedJob, as our goal is to create a model that can be applied to a new loan application. I restricted myself to observations with approval year between 1990 and 2010, noting that there is very little data from outside of that range.

The finalized data frame had 11 features and 847977 observations.

I dummy encoded all categorical variables, so that all data is in numeric form. I then split the data into training and test set, with 30% of the data in the test set. After performing the test-train split, I standardized the magnitude of all columns, so that all columns on the training set have a mean of 0 and standard deviation of 1.

Model Selection

I tried two different algorithms, and performed hyperparameter tuning for both. This data set has features with very skewed distributions, as well as highly correlated features. Decision-tree based algorithms tend to be robust to these issues, so I tried two tree-based models: random forest, and gradient boosting. The data is highly imbalanced; most loans are paid off, with only a small minority of loans going into default. Accuracy scores can be misleading for unbalanced data. Instead, I used f1-score as my primary metric instead. To choose between models with similar f1-scores, I also considered training time.

Out of the box, random forest and gradient boosting classifiers gave similar performance; the f1 score on the test set for random forest was 0.78 while the f1 score on the test set for gradient boosting was 0.79.

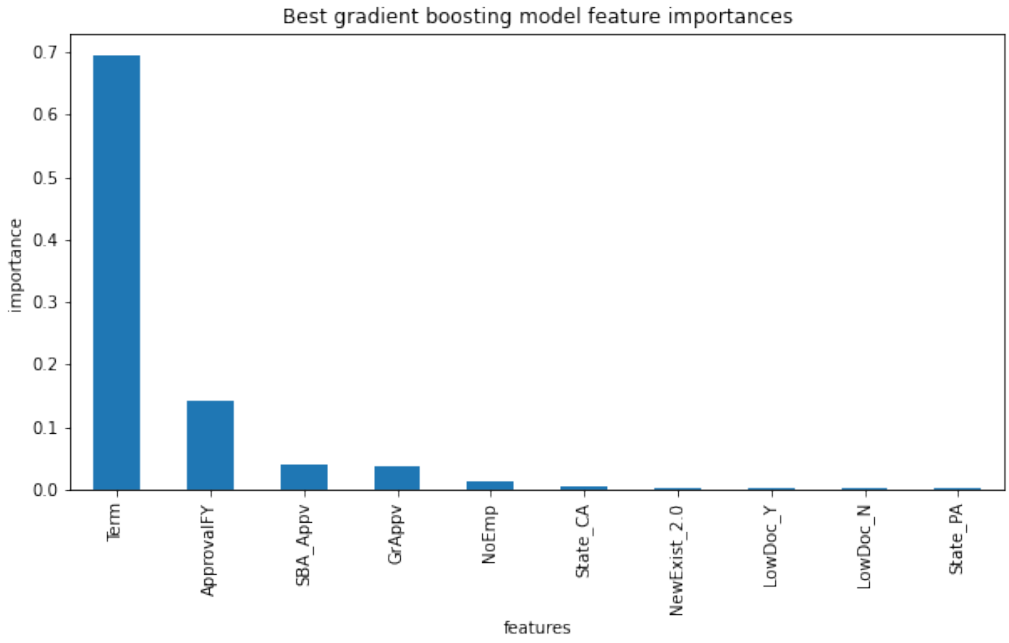
I tuned hyperparameters for both models using a grid search with three-fold cross-validation on the training set. Hyperparameter turning yielded only very slight improvements for random forest. For gradient boosting, tuning the max_depth parameter—which controls the depth of individual decision trees used in the model—gave performance gains. My final gradient boosting model, with a max_depth of 11 gave an f1-score of 0.85 on the test data.

For gradient boosting model, I was especially struck by the trade-off between performance and training time. The results of my grid-search showed that the second-best choice of hyperparameters (measured by f1 score on the test set) gave essentially identical result to the best choice, but with about half the training time. A shorter training time means a model can be easily re-trained on more recent data, so I selected the second most powerful gradient boosting model from the grid search as my final model. All hyperparameters for this model were equal to the defaults, except that max_depth was set to 11. The training time for this model was approximately 14 minutes on the entire training set.

Results and Conclusions

My final model performed reasonably well on the test data. This model could help a bank make a data-driven decision about whether to approve a small-business loan. The model has a precision of 0.83, meaning that 83% of loans which the model predicted would go into default did in fact do so. The recall of 0.86 means that 86% of loans which went into default were correctly predicted by the model. Hence if the model predicts that a loan will be paid in full, one can be reasonable confident that is actually the case.

Plotting feature importance for both random-forest and gradient boosting, I saw that both models identified the same top five features as important. It appears that only 4-5 features contribute meaningfully to our final models' predictions. These are loan term, year, total loan amount, amount guaranteed by the SBA, and perhaps number of employees. Of these, loan term was by far the most important, as seen in the plot below.



One possible take-away is that the SBA's process for vetting small businesses is very effective. One does not need an in-depth analysis of the individual details of a business to make reliable predictions about whether an SBA loan will go into default; knowing a few basic features of the loan itself seems to be enough.

Further Work

There are a number of possibilities for improving this model. I could try fitting a model using only the top predictors. The simplified model would likely achieve similar accuracy, but with less training time.

More fundamentally, the categorical variables in the data set are being largely ignored. This is a common issue in decision-tree based algorithms, especially when dealing with high cardinality features, such as the 50 U.S. states. In future iterations of the model, these features should be encoded using hashing, or another alternative to dummy encoding, to prevent their importance from getting lost. I might also replace state with some numerical proxy, such as state GDP per capita; or bin states into low, medium and high default risk based on training data.

Finally, care must be taken with the year variable. To create a model that is applicable to future data, this could be replaced with an indicator variable such as whether the loan was active in a recession year; or with the value of some economic index from the year in question. Combining the gadient boosting classifier with an economic forecasting model, while beyond the score of this project, would likely produce even more useful results.

References

Li, Min, Amy Mickel and Stanley Taylor. "'Should This Loan be Approved or Denied?': A Large Dataset with Class Assignment Guidelines," Journal of Statistics Education, vol. 26, no. 1, 2018, pp. 55-66. doi:10.1080/10691898.2018.1434342

Toktogaraev, Mirbek. Kaggle. "Should this loan be approved or denied?", March 17 2020. www.kaggle.com/mirbektoktogaraev/should-this-loan-be-approved-or-denied. Accessed September 6, 2021.