Should This Loan Be Approved Or Denied: An Exploration Of Risk Classification Using SBA Loan Data

Matt Holsten
Tufts University
matthew.holsten@tufts.edu

Robert Pitkin
Tufts University
robert.pitkin@tufts.edu

Abstract

The United States Small Business Administration ("SBA") has created a large dataset of SBA-backed loans stretching back almost 60 years documenting the information and outcomes of their loans. Due to the nature of the SBA and the successes they've helped create, there is an ever-growing desire to have additional risk information for deciding the whether or not loans should be granted. To address this desire, we employed a feature analysis to discover which aspects of SBA-backed loans contribute most to the risk of defaulting and then confirmed our analysis with two classification methods, one linear (a logistic regression model) and one non-linear (a feed-forward neural network), to predict the outcomes of loans before they are granted. We find that both linear and non-linear models produce accuracies around 70% and 80% respectively and thereby serve as viable tools for SBA Loan Officers and small businesses alike.

1 Introduction

The Small Business Administration has created a loan dataset with loans stretching from 1967 to 2014. In addition to the details of the loan itself, such as loan amount, SBA backed amount, and term of the loan, each sample in the dataset also includes the outcome of the loan: whether it was paid in full or it was charged off (defaulted). Due to this addition, the SBA dataset serves as a prime example of a binary classification problem. However, the dataset also presents many obstacles, including missing data, input errors, a heavy imbalance of the loan results, and a number of text-based features which don't lend easily to classification methods. These obstacles create additional complexity in the dataset, which is already saturated by the unpredictability of the loans themselves.

Despite these obstacles, the overarching goal of risk classification is extremely valuable for the work of the SBA. If the SBA can easily predict the outcome of a loan, they can be extremely efficient with the funds they possess and contribute even more to the health of the U.S. economy. In addition, identifying significant features can aid future small businesses as they seek SBA-backed loans. If small businesses knew what aspects of loans consistently led to past failures or successes, they would be able to make more informed business decisions as they navigate the loan seeking process. Thus, if the aforementioned obstacles were overcome, a thorough exploration of risk classification methods could contribute significantly to the efficiency and success of both the SBA and the businesses they support.

In this work, we first employ a variety of feature selection methods, including mutual information tests, F-tests, and correlation matrices in order to inform the features for our classification models as well as gain some intuition of how successful each model approach will be. Then, for the models themselves, we use two types of classification models studied in class – logistic regression and neural networks – as an attempt to map out the feature space created by the SBA loans. Lastly, we record and visualize our results to evaluate each of our models and create additional infrastructure to enable future querying of our models.

We find that with a selected list of eleven features and using country-wide data, our logistic regression approach achieves a validation accuracy of about 69% and our feed-forward neural network achieves a validation accuracy of about 81%. We also used the two models on loans only for the state of California and only for the state of Massachusetts, which resulted in a validation accuracy of about 72% (LR) and 84% (NN) and a validation accuracy of about 76% (LR) and 81% (NN) respectively.

2 Background and Related Work

In this section we discuss the background of this project, the paper associated with the dataset, and our overarching motivation.

2.1 Li et, al.

In the paper associated with this dataset, "Should This Loan Be Approved Or Denied?"[4], by Li, Mickel, and Taylor, the authors characterize small businesses as a "primary source of job creation in the United States" that can provide numerous "social benefits by creating job opportunities and reducing unemployment". Thus, for an organization such as the SBA, it is in their best interest to foster the "formation and growth" of small businesses as doing so could increase the overall health of the U.S. Economy. One of the main methods the SBA accomplishes this goal is through its loan backing program, where the SBA guarantees significant portions of requested loans in order to encourage banks to approve said loans to small businesses. Such loans have resulted in many success stories, notably including "FedEx and Apple Computer", but they have also resulted in many failures. Therefore, in order for the SBA to achieve their goal of reducing unemployment and supporting small businesses, they need to be efficient and accurate with which loans they're willing to back. Thus, the crux of the problem becomes a matter of risk classification.

From a machine learning perspective, classification, specifically binary classification, can be done with a variety of models. Examples include logistic regression, neural networks, support vector machines, k-nearest neighbors, decision trees, and others. Li et, al.[4] choose to implement the first two, logistic regression and neural networks, in R. To do so, the authors perform feature selection first based off of discussion and intuition to form a list of seven significant features and then refine this list to five after experimentation with their implemented models. However, the authors only achieve a misclassification rate of about 32% for a data subset of only California-based loans, which is too high to be considered as a legitimate tool for the SBA. Hence, our motivation for this project is an attempt to take a more formal approach to feature selection and a more rigorous approach to model implementation in order to improve upon the results found in Li et, al. and thereby prove the legitimacy of risk classification models in the SBA Loan domain.

3 Methods

In this section we describe our data cleaning methods, our feature selection methods, our classification model constructions, our model evaluations, and the construction of our web application and API.

3.1 Data Cleaning and Feature Selection

The SBA dataset contains twenty-seven total features. These features are:

Loan ID Number	Borrower Name	Borrower City	Borrower State	Borrower Zip Code
Bank Name	Bank State	North American Industry Classification Code	Loan Approval Date	Loan Approval Year
Loan Term (Months)	Number of Employees	New vs. Existing Business (1 or 0)	Number of Jobs Created	Number of Jobs Retained
Franchise Code	Urban vs. Rural (1 if Urban, 2 if Rural, 0 if undefined)	Revolving Line of Credit (Yes or No)	Part of the LowDoc Loan Program (Yes or No)	Default Date
Disbursement Date	Gross Disbursement	Gross Amount Outstanding	Loan Status (Paid in Full or Charged Off)	Charged-Off Amount
Gross Amount Approved by Bank	Gross Amount Approved by SBA			

3.1.1 Feature Exploration

We began our feature exploration with the seven features selected in the associated paper (Li, et al.): Location (State), Industry Code, Gross Disbursement, New vs. Established Business, Loans Backed by Real Estate (Loan Term >= 240 Months), Economic Recession (if the loan occurred during the 2008 Recession), and the SBA's guaranteed portion of the approved loan. The motivation behind each of these features is explained in Li, et al. (besides the intuitive reasoning), but our larger motivation for beginning with these was obtaining a starting point for our feature selection.

Next, we created additional features from the original data (similar to Loans Backed by Real Estate or Economic Recession) or added features excluded by the paper based on our own intuition. These additional features were: SBA backed proportion (the proportion of the loan backed by the SBA), Administration Party (party in the White House at the time of the loan - 1 if Democrat, 0 if Republican), Urban vs. Rural (an original feature), Jobs Created (an original feature), Jobs Retained (an original feature), and the loan term (an original feature).

Our motivation behind the SBA backed proportion was to add an additional feature that connects two other features (the total loan amount and the SBA backed amount) due to their significant correlation with each other. Because the total loan amount and the SBA backed proportion are just dollar values, there's limited information about the risk of the loan gained without comparing the two values. Thus, by finding the proportion of the loan backed by the SBA, we are creating a new feature that represents how much of the total loan the SBA is willing to cover, thereby decreasing the risk of the loan for the bank. Because the SBA does its own evaluation of businesses before committing a loan amount, a higher SBA-backed proportion would indicate a lower risk loan overall since they're willing to cover a large amount of the loan.

Administration Party was added due to thr suspected external effect politics has on the SBA. Because different administrations have different policies when it comes to economic reform and budget allocation, the political party in power could have a significant impact in how much the SBA is able to loan out, which in turn would affect how much risk the SBA is willing to take. To create this feature, a dictionary mapping years to political party was created (1 for Democrat, 0 for Republican) and used to assign a value for each sample based on the approval date of the loan.

Urban vs. Rural was added with a similar line of reasoning as Administration Party; essentially, due to different industries as well as their locations, whether or not the business is urban or rural can have an impact on how successful the business will be in their local market. Thus, it was suspected to also have an effect on the riskiness of the loan.

Jobs Created and Jobs Retained were selected as additional potential measures of how successful the small business has been thus far, since the ability to both create and retain jobs for a business can directly correlate to a business' ability to grow, which, in turn, would decrease the chances of defaulting on a loan.

Loan term was added to represent the potential "pressure" the businesses have when it comes to the approved loan. If two loans are the same total amount, but one has a longer loan term, the business has more time to pay off the loan and thus, has a lower risk of defaulting.

Finally, we made the location (state) feature optional since it would be constant if we only look at a specific state's data.

3.1.2 Data Cleaning

Before any feature selection methods could be employed, we first had to clean up errors in the data and convert any strings into numerical values (continuous or discrete).

The first feature that required reworking was approval date. Since this feature is used to generate the Recession feature, we had to ensure that we could do comparisons on the dates with the start date (December 2007) and the end date (June 2009) of the Great Recession. However, the approval date feature is formatted as 'Day-Month-Year' instead of a number. Thus, a new data column was created, ApprovalUnixTime, that is made up of each approval date converted to its corresponding Unix time. We initially thought this might cause an error due to some loans occurring before the start of Unix time (1970), but then realized the resulting negative values wouldn't show up in our two-sided inequality for the recession period. Once this new column was created, the Recession feature column was then able to be created.

The next feature addressed was approval year, since it is used to create the AdminParty feature. We just had to ensure that there were no missing values (or account for them), and fix any error values. Fortunately, there was only one error in the feature column, which was an entered approval year of '1976A' instead of '1976'. So, this value was simply replaced with the right number.

After that the industry code feature was addressed. The industry code, which typically contains six digits, was shortened to the first two digits since the first two represent the larger industry category (Li, et al.).

The Li, et al. strategy was also used for creating the RealEstate column (i.e. loan term >= 240 months = Yes (1), No (0) otherwise).

For the total loan amount, the SBA backed amount, and the SBA backed proportion features, the former two columns were converted from string dollar values to floats.

Lastly, states were mapped to integers based on alphabetical order and the loan status column was converted from paid in full or charged off to a 1 or 0 respectively.

3.1.3 Selecting Features

To actually evaluate our selected features, three metrics were employed - a mutual information test, an F-test, and a correlation matrix. The former two were visualized with bar graphs to see which features stood out the most and the correlation matrix was visualized with a heat map to easily see which features had large positive or negative correlations with both each other and the target variable (paid in full or defaulted).

After employing these three methods, the New vs. Established Business and Created Job features were removed due to each feature's poor performance in the respective tests. Both Retained Job and Recession also had poor performances, but only in one of the two tests for either feature, so we felt there wasn't enough evidence to disqualify them.

The correlation matrix also proved the relationships between a few sets of features including real estate and loan term (due to how real estate is calculated) and SBA backed proportion (as it is the ratio of the SBA backed loan amount to the total loan amount).

3.2 Creation of Models

3.2.1 Logistic Regression Classifier

The logistic regression classifier was constructed using pytorch libraries. It has ten (or eleven if including location/state) input features and a single binary output node. Since logistic regression models can also be thought of as just a feed forward network with an input layer and a single output node, we were able to leverage the pytorch neural network module for our model class. In addition, we chose 10000 iterations for our training length (20000 if working with state-level datasets) and settled on a learning rate of 0.001 after trying out some larger values (0.05, 0.01, 0.1). We think 0.001 worked the best in the end because logistic regression is trying to find a linear decision boundary and using a large learning rate made it too hard to find a local (or global) minimum for our loss. For our loss function, we chose binary cross entropy as it is well known and researched for binary classification models. Within our loss functions we also passed class weights to remedy the imbalanced nature of the dataset (heavily skewed towards paid in full loans as these are the most common outcome). Lastly, we chose to use stochastic gradient descent for our model optimizer.

3.2.2 Feed-forward Neural Network

The feed-forward neural network classifier was also constructed using pytorch libraries. Since logistic regression was shown to be a reasonably successful approach in the associated paper, we believed that the relationship between the two classes may not be linearly separable, but also isn't extremely complex (i.e. could be easily non-linearly separable). Thus, the network was constructed with only a single hidden layer to allow for approximation of any continuous function (Universal Approximation Theorem) and to keep the overall network small and efficient. To confirm this choice, trials were held with two hidden layers, but didn't result in any noticeable gain in performance. Similar to the logistic regression model, both binary cross entropy with class weights and stochastic gradient descent were chosen for the loss function and model optimizer respectively. However, a momentum parameter of 0.9 was added to the model optimizer due to the addition of the hidden layer.

Once the initial network decisions were made (and tested with an arbitrary hidden layer size and learning rate to confirm the approach) hyper parameter tuning was employed to determine the best learning rate, hidden layer size, and number of iterations for the model. Learning rate was varied over a log space ranging from 0.001 to 0.1 with five values, the hidden layer size ranged over powers of two from two to thirty-two, and the number of iterations ranged over 5000, 10000, 20000, and 40000, which were chosen arbitrarily.

The best results from the hyper parameter tuning were then chosen to create the final model, which was then run on the dataset to generate the results.

3.3 Evaluation of Models

For both models, the total dataset was split into three groups: training (about 65%), testing (about 25%), and validation (about 10%). The training and testing sets were used during the training process, to train the model and to test every 1000 iterations, and the validation set was used on the final model to confirm model performance and accuracy.

To evaluate the two models, three metrics were employed - overall model accuracy on the validation set, a confusion matrix and an ROC curve. The confusion matrix was only generated from the last set of predicted values for each model (i.e. the fully-trained model), whereas the ROC curve was created from a list of all predicted values during testing.

The confusion matrix was used to ensure that the model wasn't just predicting one class or the other and was actually learning the correct labels for each sample. In fact, early on in the project process, utilizing the confusion matrix helped to debug an issue where the model would only return labels of 1.0. Additionally, it is only created from the predicted labels from the fully trained model because running it using all of the predicted labels would heavily dilute the performance of the final model.

The ROC curve was used since it can visually represent the classification ability of binary classifiers and do so over the entire training process. The resulting shape of the ROC curve (i.e. smooth vs. bumpy) can indicate important aspects of the model, such as how fast the model is learning, which, in particular, helped during the hyper-parameter tuning of the neural network.

3.4 Evaluation of Features

Once both the logistic regression model and the neural network model were built and trained on the full set of selected features, the next step was to evaluate the significance of different features. To do so, one of the significant features from feature selection was removed from the data set, had a new model trained on the data and evaluated, and then replaced. This process was done for three of the selected features due to time constraints: loan term, administration party, and SBA backed loan proportion.

3.5 API & Web Application

In order to have our trained neural network model available for public use, we created both an API and a website to interface with that API which could receive samples of loan proposals and return the predicted risk-levels according to a potential lender.

3.5.1 API

We began by creating a publicly-available API to allow our PyTorch model to be queried directly by users or within client applications. To do this, we used the open-source, Python-based framework Flask, and launched it for free on a Python cloud hosting service. Much like other APIs, ours accepts specific keyword arguments entered into the URL and processes them as input values for later use. These inputs are then validated and used to query our PyTorch model, and the results are served in JSON format back to the client on that webpage. The results include values such as a list of the input fields it received for client validation, a list of fields which had values that were in an incorrect format and an associated error message, and the result of our PyTorch model evaluated on the input values.

3.5.2 Web Application

As stated, a client can directly query our API in any browser. This client, however, could itself be a program or web application with the capability of fetching online data. We exploited this inherent abstraction layer to create a separately hosted web application which interfaces with our API so that users can have a more easy-to-use graphic user interface. Our web application uses the open-source, JavaScript-based framework React, as well as some other open source, React-based libraries as detailed in our GitHub repository. The website works by accepting user input in form fields, constructing the appropriate URL string with keyword arguments and values from those inputs, querying our API with that URL, and lastly displaying the results in an easy to view manner. The website allows users to input their own fields, as well as generate a random sample of fields based on the average value for each field.

4 Results

4.1 Feature Selection

We first provide the results of our feature selection methods. As mentioned in section three, we employed a mutual information test, an F-test, and a correlation matrix to investigate the relationships between our features and our target as well as within the features themselves. All tests were performed on country-wide data and included state/location to gain intuition for our generalized model.

4.1.1 Mutual Information

As seen in Fig. 1, loan term resulted in the highest mutual information value out of all features selected, more than tripling the next highest feature, urban vs. rural. Other notable features included total loan amount, backed by real estate, SBA loan amount, administration party, SBA proportion, industry, and retained job. State (location) had a surprisingly low value, lending itself to potentially being eliminated altogether.

The lowest performing features for the mutual information test were recession, new vs. old business, and jobs created.

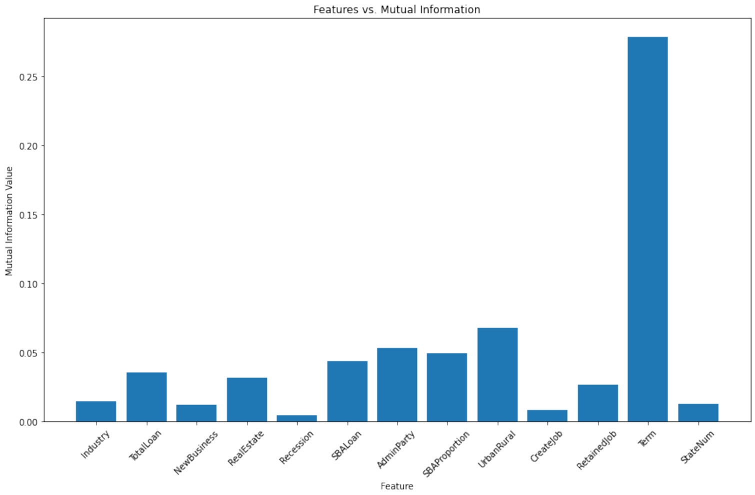


Figure 1: Features vs. Mutual Information

4.1.2 F-Test

The results of the F-Test differed little from the mutual information test. As seen in Fig. 2, loan term still dominates when compared to other features, thereby cementing its usefulness in creating our decision boundary. Aside from loan term, backed by real estate, administration party, SBA proportion, and urban vs. rural all performed well, similar to the mutual information test. Notably, retained jobs dropped significantly, alongside jobs created and new vs. old business, whereas recession actually increased relative to other features.

The poor performances from the new vs. old business and jobs created for both the mutual information test and the F-Test led us to believe they contribute little to forming the eventual decision boundary. Such beliefs were later confirmed with thorough testing.

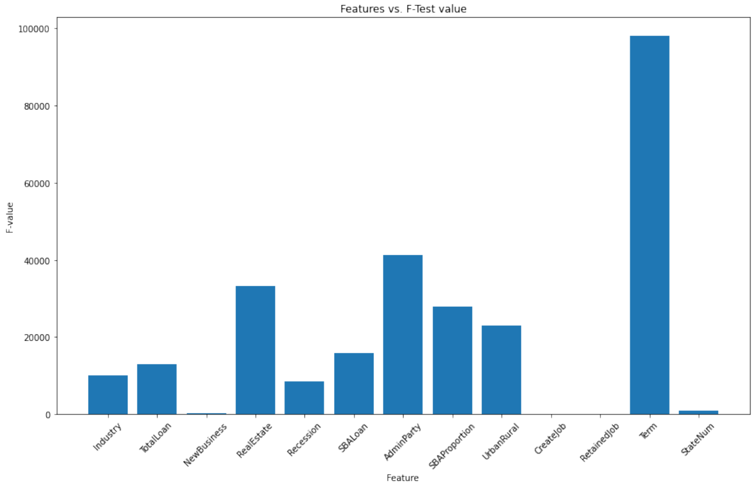
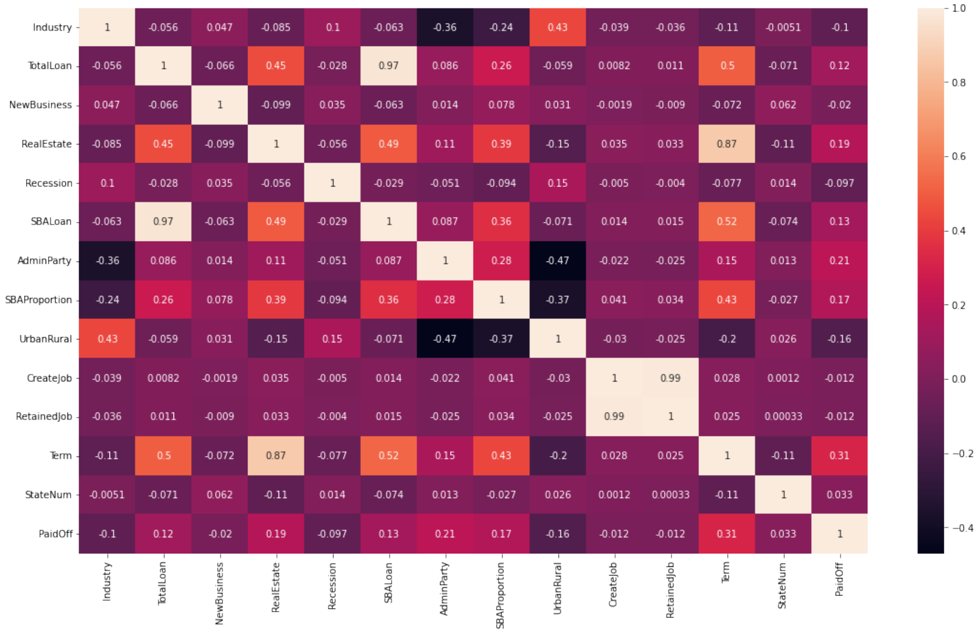


Figure 2: Features vs. Mutual Information

4.1.3 Correlation Matrix

The constructed correlation matrix was formed as a heat map to easily demonstrate the differences between positive and negative correlations. In Fig. 3, one can see the confirmation of loan term having the largest magnitude correlation with our target feature, paid off, out of all features compared. For the rest of the correlations with the target variable, the matrix generally reflected the results from the previous two tests. However, what we gain from the correlation matrix is interesting correlations and trends between features, such as administration party and urban vs. rural having a significantly negative correlation, likely due to differences in political priorities. Administration party also had significant correlations (negative and positive) with both industry and the SBA backed proportion of the loan. These correlations help to confirm its usage as a useful feature for our models. Another extremely significant correlation was between backed by real estate and loan term, both of which performed highly on previous tests. The fact that these two features were also highly correlated with each other established them both as extremely significant features for later testing.



	Industry	TotalLoan	NewBusiness	RealEstate	Recession	SBALoan	AdminParty	SBAProportion	UrbanRural	CreateJob	RetainedJob	Term	StateNum	PaidOff
Industry	1	-0.056	0.047	-0.085	0.1	-0.063	-0.36	-0.24	0.43	-0.039	-0.036	-0.11	-0.0051	-0.1
TotalLoan	-0.056	1	-0.066	0.45	-0.028	0.97	0.086	0.26	-0.059	0.0082	0.011	0.5	-0.071	0.12
NewBusiness	0.047	-0.066	1	-0.099	0.035	-0.063	0.014	0.078	0.031	-0.0019	-0.009	-0.072	0.062	-0.02
RealEstate	-0.085	0.45	-0.099	1	-0.056	0.49	0.11	0.39	-0.15	0.035	0.033	0.87	-0.11	0.19
Recession	0.1	-0.028	0.035	-0.056	1	-0.029	-0.051	-0.094	0.15	-0.005	-0.004	-0.077	0.014	-0.097
SBALoan	-0.063	0.97	-0.063	0.49	-0.029	1	0.087	0.36	-0.071	0.014	0.015	0.52	-0.074	0.13
AdminParty	-0.36	0.086	0.014	0.11	-0.051	0.087	1	0.28	-0.47	-0.022	-0.025	0.15	0.013	0.21
SBAProportion	-0.24	0.26	0.078	0.39	-0.094	0.36	0.28	1	-0.37	0.041	0.034	0.43	-0.027	0.17
UrbanRural	0.43	-0.059	0.031	-0.15	0.15	-0.071	-0.47	-0.37	1	-0.03	-0.025	-0.2	0.026	-0.16
CreateJob	-0.039	0.0082	-0.0019	0.035	-0.005	0.014	-0.022	0.041	-0.03	1	0.99	0.028	0.0012	-0.012
RetainedJob	-0.036	0.011	-0.009	0.033	-0.004	0.015	-0.025	0.034	-0.025	0.99	1	0.025	0.00033	-0.012
Term	-0.11	0.5	-0.072	0.87	-0.077	0.52	0.15	0.43	-0.2	0.028	0.025	1	-0.11	0.31
StateNum	-0.0051	-0.071	0.062	-0.11	0.014	-0.074	0.013	-0.027	0.026	0.0012	0.00033	-0.11	1	0.033
PaidOff	-0.1	0.12	-0.02	0.19	-0.097	0.13	0.21	0.17	-0.16	-0.012	-0.012	0.31	0.033	1

Figure 3: Features vs. Mutual Information

4.2 Logistic Regression Approach

We now evaluate the performance of the Logistic Regression model. After 50000 iterations with all selected features (including state/location) on country-wide data, the Logistic Regression model achieved an accuracy of 68.84% on the validation set. The associated confusion matrix (Fig. 4) and ROC curve (Fig. 5) for the model both confirm the legitimacy of this result.

We then shifted to state-level analysis for California, which was tested in Li, et al., and Massachusetts, to ensure that our approach can be generalized on a state-level.

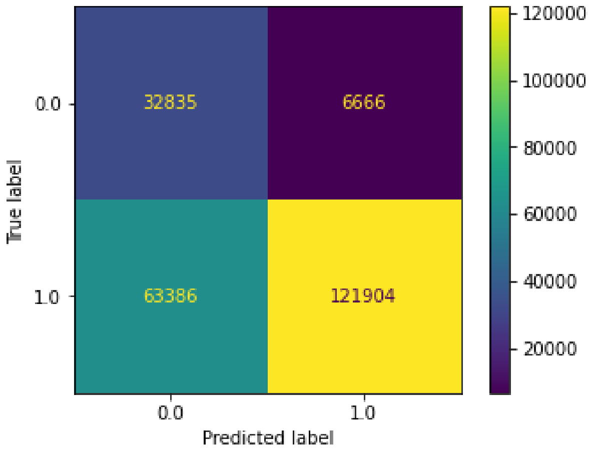


Figure 4: Confusion Matrix for Logistic Regression (Country-wide)

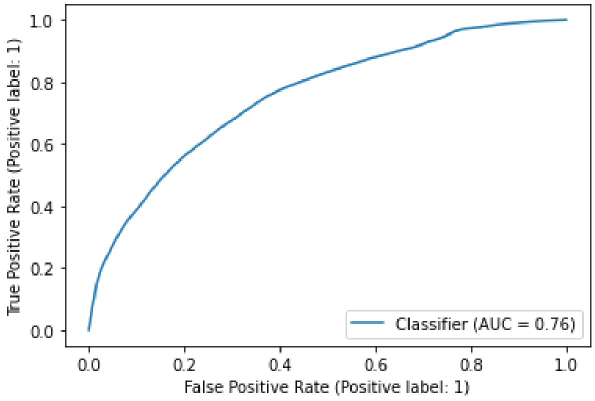


Figure 5: ROC Curve for Logistic Regression (Country-wide)

4.2.1 California

For California-only data, Li et, al.[4] reported a misclassification rate of 32.16%, which is the same as an accuracy of 67.84%. Our logistic regression model reported an accuracy of 72.43% after 50,000 iterations, showing a marked improvement of 4.59% over the associated paper. Both the confusion matrix and ROC curve can be found in Appendix A.

4.2.2 Massachusetts

For Massachusetts-only data, which is a significantly smaller dataset (25,000 samples compared to 900,000 country-wide or 130,000 in California), our logistic regression model reported an accuracy of 76.35% after 50,000 iterations of training. The confusion matrix and the ROC curve for the model can be found in Appendix B.

4.3 Neural Network Approach

Now we turn to our neural network model. We first review the results of hyper-parameter tuning and then use the best hyper-parameters to construct the most effective model for testing and evaluation.

4.3.1 Hyper-parameter Tuning

The three hyper-parameters that were varied were number of training iterations, number of nodes in the hidden layer, and learning rate. By varying iterations over three values, nodes over five values, and learning rate over five values, we trained and tested 75 total models. Table 1 shows the accuracy results for the top ten models.

Table 1: Hyper-parameter Tuning Results (Top 10)

Accuracy	Iterations	No. of Nodes	Learning Rate
81.87%	20000	16	0.01
81.76%	10000	32	0.01
81.69%	5000	32	0.01
81.11%	10000	32	0.0316
80.83%	20000	8	0.01
80.78%	20000	4	0.01
80.74%	20000	2	0.01
80.60%	10000	8	0.01
80.51%	20000	16	0.0316
80.46%	10000	16	0.01

From these results, we decided to use the best performing model - 16 hidden nodes, 20,000 iterations, and a learning rate of 0.01 - for our evaluation and submission.

4.3.2 Final Neural Network Model

With the optimal hyper-parameters selected, our final neural network classifier reported a validation accuracy of 80.93% on country-wide data, over 12% higher than our logistic regression model. Fig. 6 and 7 show the confusion matrix and ROC curve for the country-wide model.

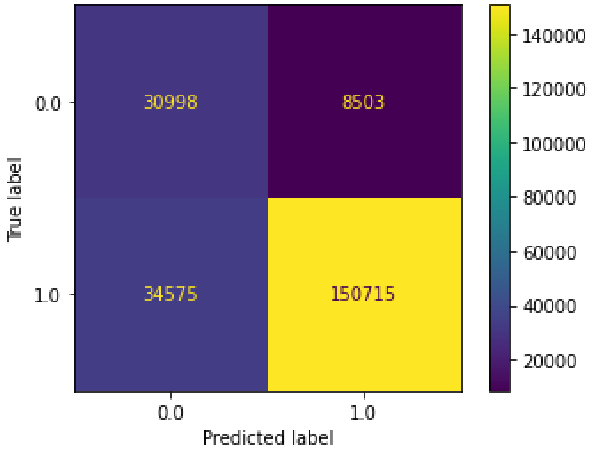


Figure 6: Confusion Matrix for Neural Network (Country)

We also constructed models for both California and Massachusetts data using the optimal hyper-parameters. Our California-only model reported a validation accuracy of 84.81%, which is 16% higher than the associated paper's model and 12% higher than our logistic regression model. On the other hand, our Massachusetts-only model reported a validation accuracy of 80.85%, which is only 4% higher than our logistic regression model. The associated confusion matrices and ROC curves can be found in Appendix C and D respectively.

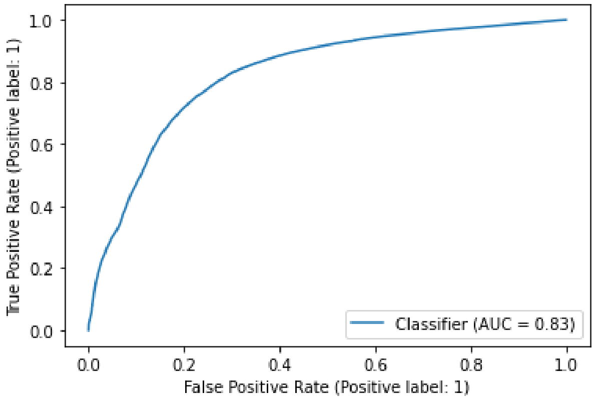


Figure 7: ROC Curve for Neural Network (Country)

4.4 Evaluating Significant Features

The three features we tried removing, thereby evaluating their significance, were loan term, administration party, and SBA backed loan proportion.

By removing loan term as a feature, our logistic regression model reported an accuracy of 61.61%, which was about 7% lower than including it. this is a decrease of over 10% in performance. Our neural network model fared even worse as it reported an accuracy of 62.16%, about 18% less than when we included loan term as a feature. Both of these results more than confirm our original hypothesis that loan term is a significant feature for forming our decision boundary.

Next, removing administration party as a feature had little to no impact on our logistic regression model, as it still reported an accuracy of about 68% after 50,000 iterations of training. However, it did decrease the neural network performance from 80% to 77%. While it didn't make as significant of a difference as loan term did in terms of accuracy, administration party as a feature still improves the model enough to warrant its inclusion.

A similar result occurred when removing SBA backed proportion. Both the logistic regression model and the neural network model reported no decrease in accuracy. This stands contrary to the results of our mutual information and F-tests. However, it's more than possible that the models gain equivalent information by having both the total loan amount and the SBA backed loan amount as additional features. Thus, we shifted to investigating if removing the loan amounts and leaving in the loan proportion could result in an identical accuracy, but with two fewer features. Somewhat surprisingly, our results were identical to our original models, 68% accuracy for our logistic regression model and 80% accuracy for our neural network model. This result confirmed our suspicion that the main information to be gained from the loan data is the proportion backed by the SBA, rather than the actual sizes of the loans themselves. That said, we leave in both total loan amount and SBA backed loan amount for any additional accuracy they may provide.

5 Discussion

In this section we address some of the ideas and questions that were raised during experimentation.

5.1 Effectiveness of Neural Network over Logistic Regression

Given our results, and the consistent improvement in performance that our Neural Network model had over our Logistic Regression model, it's clear that the decision boundary between low and high risk loans isn't linearly separable, or at least not easily. That said, there was also a trend of the neural network models predicting more positive labels (i.e. low risk loan) than their logistic regression counter-parts. Although the neural networks produce overall more accurate results, such trends warrant further investigation and we leave this as future work.

5.2 Translating hyper-parameter tuning to the local level

Even though we trained over seventy-five models in an attempt to identify the correct hyper-parameters, we only did so using country-wide data. While we assumed such parameters would also work for state-level data, they didn't necessarily work as we hoped. For example, our Massachusetts-only neural network only reported an accuracy increase of 4% from the logistic regression model. While this could be due to Massachusetts data being more linearly-separable than other states, we suspect a much stronger reason is the significant decrease in size (900k samples to about 60k samples). Because there are so fewer samples to train on, a smaller network, such as one with only eight hidden nodes, or a larger learning rate, such as 0.03, could've generated a more effective model. Furthermore, even though the California-only neural network model had a much more significant increase in accuracy compared to its logistic regression counterpart, it, like the Massachusetts model, reported somewhat shaky ROC curves (Appendix C and D), indicating potential errors in our hyper-parameters. Thus, for future experimentation, we look to performing identical hyper-parameter optimization, but at a local level, in an attempt to better translate the ability of each state's model.

5.3 Other Approaches

Another approach we could've taken given our data set and our problem was a Support Vector Machine. Li et, al.[4] states that an SVM approach resulted in little to no improvement over the other approaches, but they also left out a significant number of features that we ended up using. An SVM approach would be interesting compared to the other two due to its ability to find an optimal margin classifier, rather than just approximating a decision boundary. As a result, we leave such an approach as potentially fruitful future work.

6 Conclusion

In this project, we attempt to discover significant features for classifying the risk of small business loans backed by the SBA. Through feature analysis and selection, as well as model implementation for testing and evaluation, we develop a list of eleven significant features: loan term, administration party, urban vs. rural, SBA backed proportion, total loan amount, SBA backed loan amount, industry code, whether or not the loan is backed by real estate, whether or not the loan was approved during the recession, number of jobs retained, and, finally, location. When these features were utilized in a straightforward neural network, they were able to achieve a classification accuracy of over 80% for country-wide data and even up to 84% for state-level data (see section 4.3.2). Through our experimentation, we hope that our results prove the legitimacy of such an approach in the SBA loan domain and can aid the SBA and small businesses alike as they navigate the loan approval process.

Acknowledgements

We are grateful to Professor Sang Chin and the entire Teaching Assistant staff for Tufts University's CS135 - Introduction to Machine Learning for the instruction of the technical skills required for this project and for inspiring a current and future interest in the field of machine learning.

References

[1] "5. Visualizations." Scikit-Learn, https://scikit-learn/stable/visualizations.html. Accessed 7 May 2022.

[2] Feedforward Neural Networks (FNN) - Deep Learning Wizard. https://www.deeplearningwizard.com/deep_learning/practical_pytorch/pytorch_feedforward_neuralnetwork/. Accessed 7 May 2022.

[3] Jain, Vandit. "Handling Class Imbalanced Data Using a Loss Specifically Made for It." Medium, 14 Sept. 2019, https://towardsdatascience.com/handling-class-imbalanced-data-using-a-loss-specifically-made-for-it-6e58fd65ffab.

[4] Li, Min, et al. "'Should This Loan Be Approved or Denied?': A Large Dataset with Class Assignment Guidelines." Journal of Statistics Education, vol. 26, no. 1, Jan. 2018, pp. 55–66. DOI.org (Crossref), https://doi.org/10.1080/10691898.2018.1434342.

[5] Loevlie, Denny. “Logistic Regression with PyTorch.” Medium, 23 Jan. 2022, https://towardsdatascience.com/logistic-regression-with-pytorch-3c8bbea594be.

[6] Should This Loan Be Approved or Denied? https://www.kaggle.com/mirbektoktogaraev/should-this-loan-be-approved-or-denied. Accessed 7 May 2022.

[7] Training with PyTorch — PyTorch Tutorials 1.11.0+cu102 Documentation. https://pytorch.org/tutorials/beginner/introyt/trainingyt.html. Accessed 7 May 2022.

[8] Verma, Akshaj. “Pytorch [Tabular] — Binary Classification.” Medium, 27 Oct. 2021, https://towardsdatascience.com/pytorch-tabular-binary-classification-a0368da5bb89.

A Appendix

A.1 Appendix A

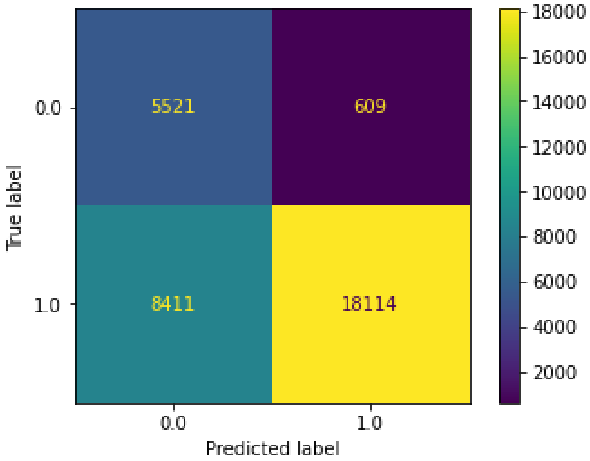


Figure 8: Confusion Matrix for Logistic Regression (CA)

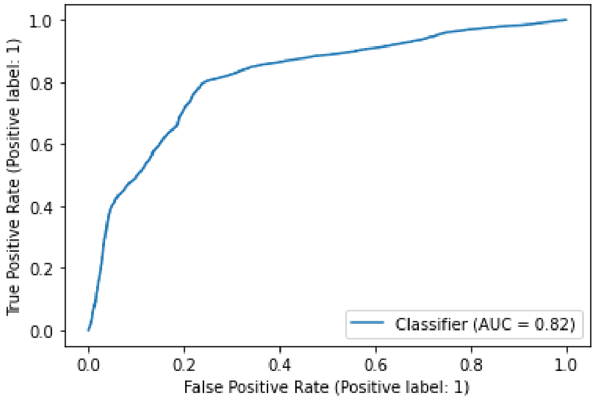


Figure 9: ROC Curve for Logistic Regression (CA)

A.2 Appendix B

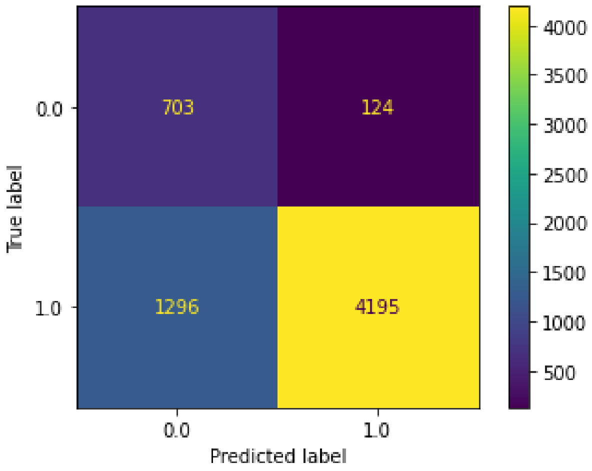


Figure 10: Confusion Matrix for Logistic Regression (MA)

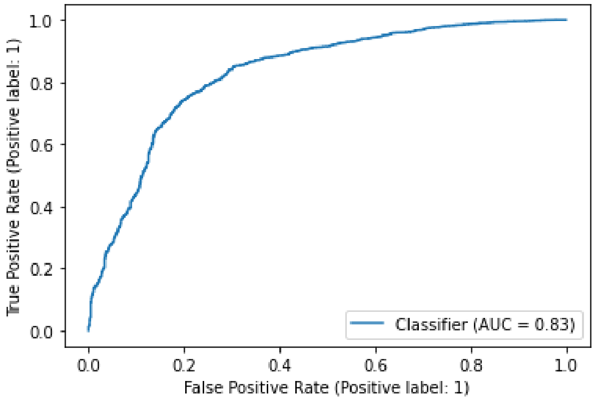


Figure 11: ROC Curve for Logistic Regression (MA)

A.3 Appendix C

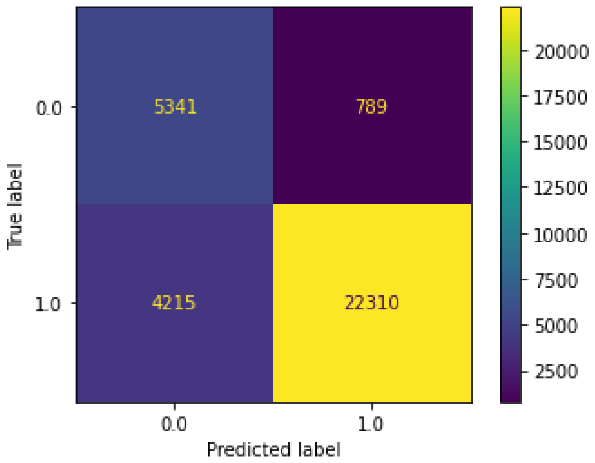


Figure 12: Confusion Matrix for Neural Network (CA)

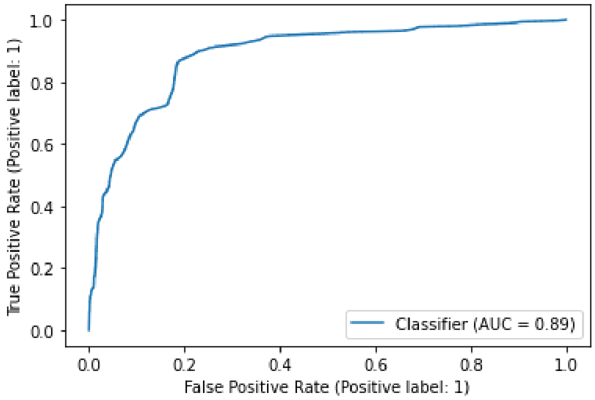


Figure 13: ROC Curve for Neural Network (CA)

A.4 Appendix D

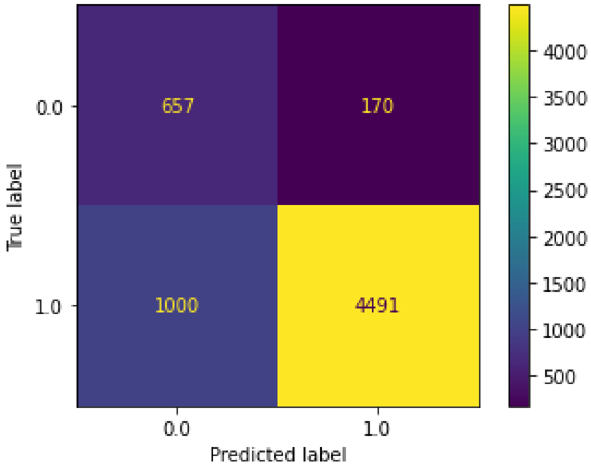


Figure 14: Confusion Matrix for Neural Network (MA)

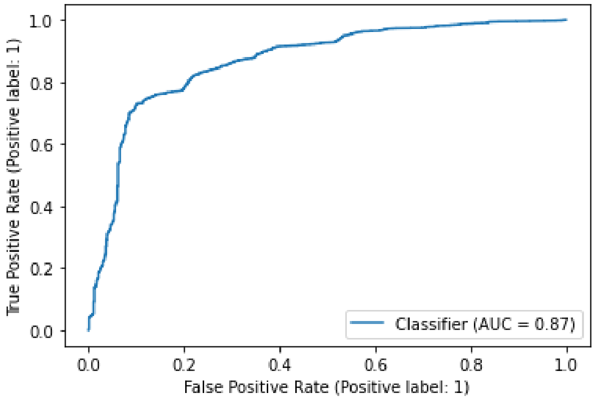


Figure 15: ROC Curve for Neural Network (MA)