Springboard Data Analytics Course

Report for Capstone I:

SBA Loan Performance in Colorado

August 2020

David Olivero

Introduction

1.1 Problem Statement

The Small Business Administration (SA) is a fundamental way the US government promotes and provides liquidity and capital to small businesses around the country. The SBA's stated core strategic goals include:

1. Support small business revenue and job growth;
2. Build healthy entrepreneurial ecosystems and create business friendly environments;
3. Restore small businesses and communities after disasters; and
4. Strengthen SBA's ability to serve small businesses.

Concretely, the SBA supports local business endeavors by guaranteeing loans from area banks, up to 85% of the loan value, thus incentivizing loaning institutions to provide capital to a wide variety and great number of business interests.

As these are government-backed loans, much of the data associated with the loans is public domain. By examining records of past loans, we discover a significant statistic: roughly 15-18% of SBA-backed loans result in loan default. **The question is, can a model be built to predict whether or not a loan will end in default?**

To answer this, a dataset was constructed which focuses on a particular state: Colorado. Doing so allows the usage of numerous supplemental datasests made publicly available through an initiative called GoCodeColorado, which is "an initiative of the Secretary of State's Business Intelligence Center … (which) challenges multidisciplinary teams to turn public data into useful business insights, analyses, and tools." These data accord a variety of data not associated with the SBA-provided loan dataset; namely, population, income and health data thought to be of interest.

After combining the various data sources into a coherent dataset, the main objective was a binary classifier model capable of predicting loan default; specifically, to determine whether or not a specific loan will be paid in full.

With a binary classifier model, there are two possible forms of error: Firstly, the incorrect assignment of worth (i.e., believing a loan will be repaid when in fact it won't be), and secondly the incorrect flagging of a loan as a probable default (i.e., believing a loan will be defaulted on when in fact it won't be). It's worth considering the relative impact of each form of error.

On one hand, while a couple percent is probably acceptable, **getting something like 15% of our "good loan" predictions wrong would indicate we're not providing any meaningful business value**.

On the other hand, wrongfully flagging loans as likely defaults is not only contrary to the SBA strategic goals, but also will indicate some sort of inherent bias in the modeling approach; **such bias will likely disadvantage a particular (unknown) subset of business situations**; not a great situation either. Hence success must be defined as simultaneously minimizing both forms of error, i.e., **provide significant business value to loaning institutions while minimizing prediction errors resulting from modeling bias**.

1.2 Dataset Summary

The SBA loan dataset will be used for this project and originates from Kaggle dataset website (https://www.kaggle.com/larsen0966/sba-loans-case-data-set). While never formalized as a competition, this dataset provides a wide variety of features with the intent of training binary classifiers. For this study, the full dataset was subsetted for specifically Colorado loans. This

subset includes data for approximately 21,000 Colorado loans between the years 1970 and 2013, each loan with roughly 26 feature variables corresponding to it, shown in Figure 1.

Data from GoCodeColorado were obtained from their website, following proper registration (https://gocode.colorado.gov/data/current-data-sets/). Data pulled from GoCodeColorado included labor, income and population data which could be grouped by year and by county. In particular, county-level data was believed to be a useful level of data granularity, as there are 63 counties within the state of Colorado, widely ranging in terms of the variables mentioned above.

Some features of the GoCodeColorado dataset were intentionally omitted from the modeling effort; namely, the use of race and gender information, as well as crime and marijuana usage statistics, in an effort to prevent damaging model bias as mentioned earlier.

Lastly, mortality statistics by year and county were collected and wrangled from the CDC's WONDER database. (https://wonder.cdc.gov/controller/datarequest)

Figure 1: List of 26 Attributes found in the Kaggle SBA Loan Dataset.

#	Column	Description	Dtype
0	LoanNr_ChkDgt	Loan Number	int64
1	Name	Business Name	object
2	City	Business City	object
3	State	Business State	object
4	Zip	Business Zipcode	int64
5	Bank	Loaning Institution	object
6	BankState	State of Bank Branch	object
7	NAICS	Industry Sector (encoded)	int64
8	ApprovalDate	Date Loan Approved	object
9	ApprovalFY	Year Loan Approved	object
10	Term	Term of Loan (Mo)	int64
11	NoEmp	Number of Employees in Business	int64
12	NewExist	New or Existing Business	float64
13	CreateJob	Jobs created by business	int64
14	RetainedJob	Jobs retained by business	int64
15	FranchiseCode	Franchise Code	int64
16	UrbanRural	Urban or Rural idenifier	int64
17	RevLineCr	Line of Credit (yes or no)	object
18	LowDoc	Low-Documentation loan (yes or no)	object
19	ChgOffDate	Date loan charged off	object
20	DisbursementDate	Date funds disbursed	object
21	DisbursementGross	Loan Disbursement ($)	object
22	BalanceGross	Balance at end of loan	object
23	MIS_Status	Loan Paid Off (yes or no)	object
24	ChgOffPrinGr	Charge off principal	object
25	GrAppv	Gross approved loan amount	object
26	SBA_Appv	SBA approved loan amount	object

2.1 Data Wrangling & Exploratory Data Analysis

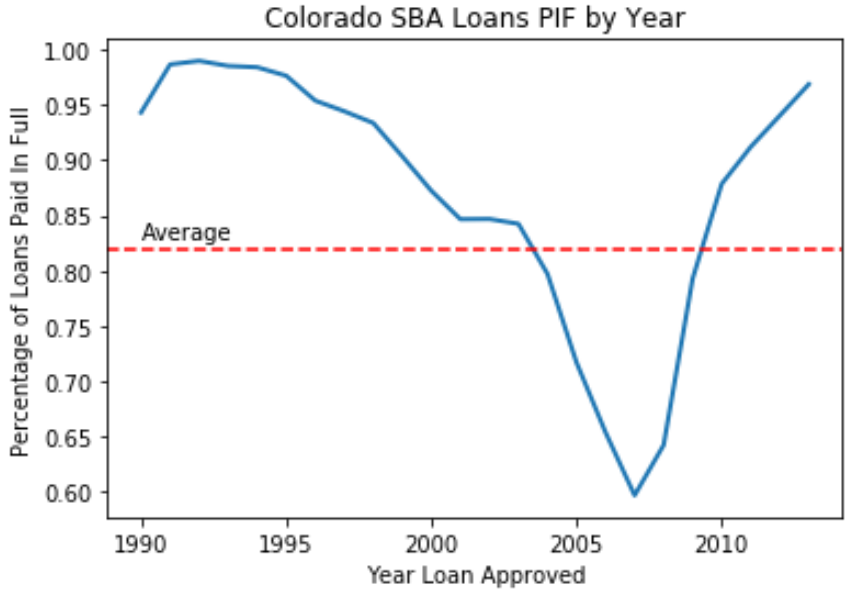
Consistent with the US-wide observation, approximately 85% of the entries in the SBA loan dataset were defaulted, indicated by the MIS_Status variable as “CHGOFF", meaning the loan was charged off. The primary issue with this dataset was its timespan; too much time and history might negatively affect the predictive power of features, or overfit the model. Figure 2

shows the number of loans for each year in the dataset. As indicated by the figure, subsetting the dataset to **only include loan years from 1990-2013** reduced the timespan of the dataset in half, while only removing 400-500 rows of data.

Figure 2: Loan Count for each year in SBA CO dataset.

Figure 3 shows the default rate of the remaining loans as a function of the year they were initiated, indicating a **sharp decline right before the Great Recession**:

Figure 3: Loan Paid Off Rate for each year in dataset.



To fold in the GoCodeColorado and CDC data, which were indexed by county, a separate list was imported consisting of the 500 largest towns in Colorado. This file included County name and FIPS (Federal Information Processing Standards) number for each county. This was merged into the dataset using the City feature. Limiting the dataset to only the 500 largest owns in the state only reduced the dataset size by 27 rows.

Numerous columns were eliminated from the dataset, as follows:

1 - Name: Name of Individual
2 - City: City of Business (redundant to county-specific information in the model)
3 - Bank: Assume that the bank the loan is with is not a significant factor.
4 - BankState: The state of the branch of bank used. Again, assumed to be not significant.
5 - UrbanRural: 40% of rows undefined make this not a useful category.
6 - ChgOffDate: Assume date of charge-off irrelevant to loan payment performance.
7 - BalanceGross: A redundant feature to the loan amount.
8 - County_Fips: Key column for adding other data, removed after merging data.
9 - County_Name: County name redundant to county-level data imported.
10- GrAppv: - Highly correlated to gross disbursement.
11- SBA_Appv: - Also highly correlated to gross disbursement.
12- ChgOffPrinGr: Mostly empty data.
13- ApprovalDate: Assume loan performance independent of calendar date.
14- DisbursementDate: Similar to approval date.

Before removing the ApprovalDate and DisbursementDate features, a new feature was generated called “PayDelay” that tallied the number of days it took for money to get disbursed (just in case this was a significant variable). Figure 4 shows a CDF of PayDelay, which was truncated to 1000 days. The result is an exponential distribution with a mean of 100 days or so:

Figure 4: Normalized CDF of PayDelay.

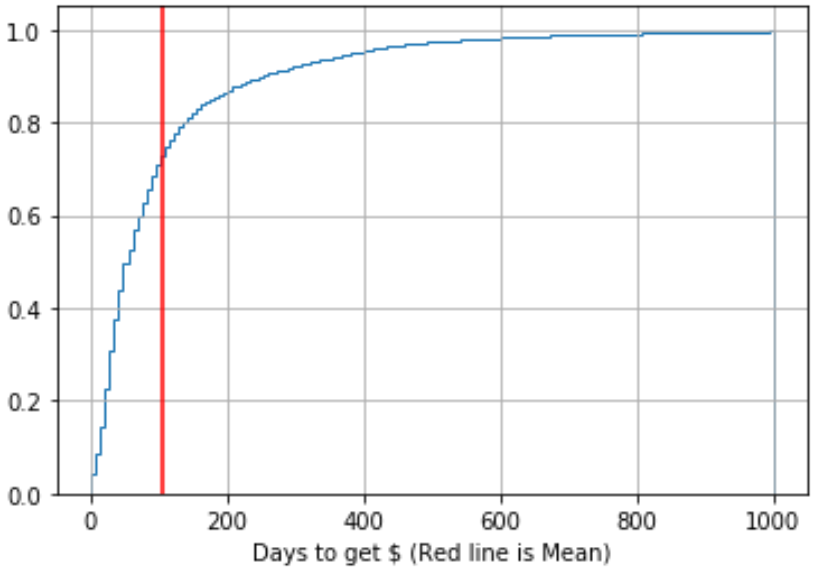


Figure 5 shows a sample of data pulled from the GoCodeColorado website, containing median income, labor and population data indexed by county and year. These were left-merged into the main dataset on both **ApprovalFY** and **County_Name**:

Figure 5: GoCodeColorado data (head of DataFrame).

	areaname	periodyear	incdesc	income	population
394	Adams County	1990	Per Capita Personal Income - Bureau of Economi...	16192	266629.0
397	Adams County	1991	Per Capita Personal Income - Bureau of Economi...	16647	274311.0
399	Adams County	1992	Per Capita Personal Income - Bureau of Economi...	17538	284362.0
401	Adams County	1993	Per Capita Personal Income - Bureau of Economi...	18393	293995.0
404	Adams County	1994	Per Capita Personal Income - Bureau of Economi...	19262	302197.0

	County	Year	LaborForceInCty	EmployedInCty	UnempInCty	UnempRateInCty
0	Adams	1990	140544	132385	8159	5.8
1	Adams	1991	142002	134516	7486	5.3
2	Adams	1992	145671	137195	8476	5.8
3	Adams	1993	150890	142831	8059	5.3
4	Adams	1994	158763	151849	6914	4.4
...	...	...	...	...	...	...
1915	Elbert	2018	15034	14628	406	2.7
1916	Lake	2016	4613	4481	132	2.9
1917	Kit Carson	2016	4534	4438	96	2.1
1918	Baca	2019	2175	2138	37	1.7
1919	Otero	2018	8401	7990	411	4.9

Using the Population and year data, a new feature called “PopChange” was created, which contains the change in population between 1995 and 2010, normalized against the 1995 population level, for a given county. The resulting variable PopChange distribution is shown in Figure 6, and ranges from 30% decreases to 60% increases in county population:

Figure 6: Distribution of PopChange for Colorado’s 63 counties.

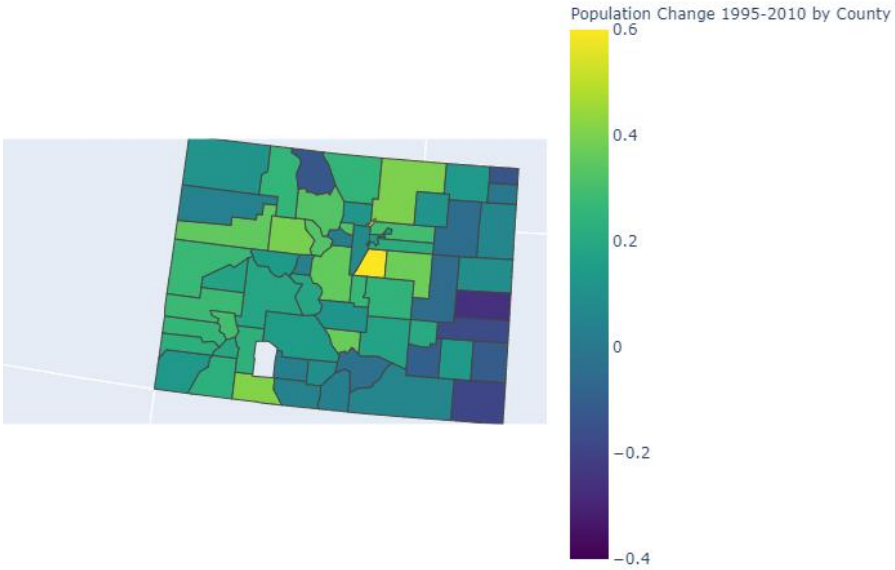


Figure 7: shows the county map of median household income, where for each county the median incomes for each year are compared and the maximum value selected:

Figure 7: Median Household Income for Colorado’s 63 counties.

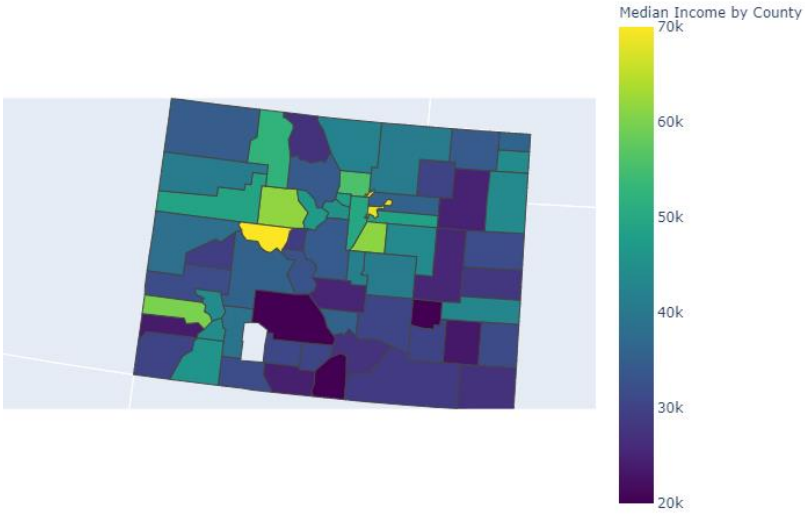
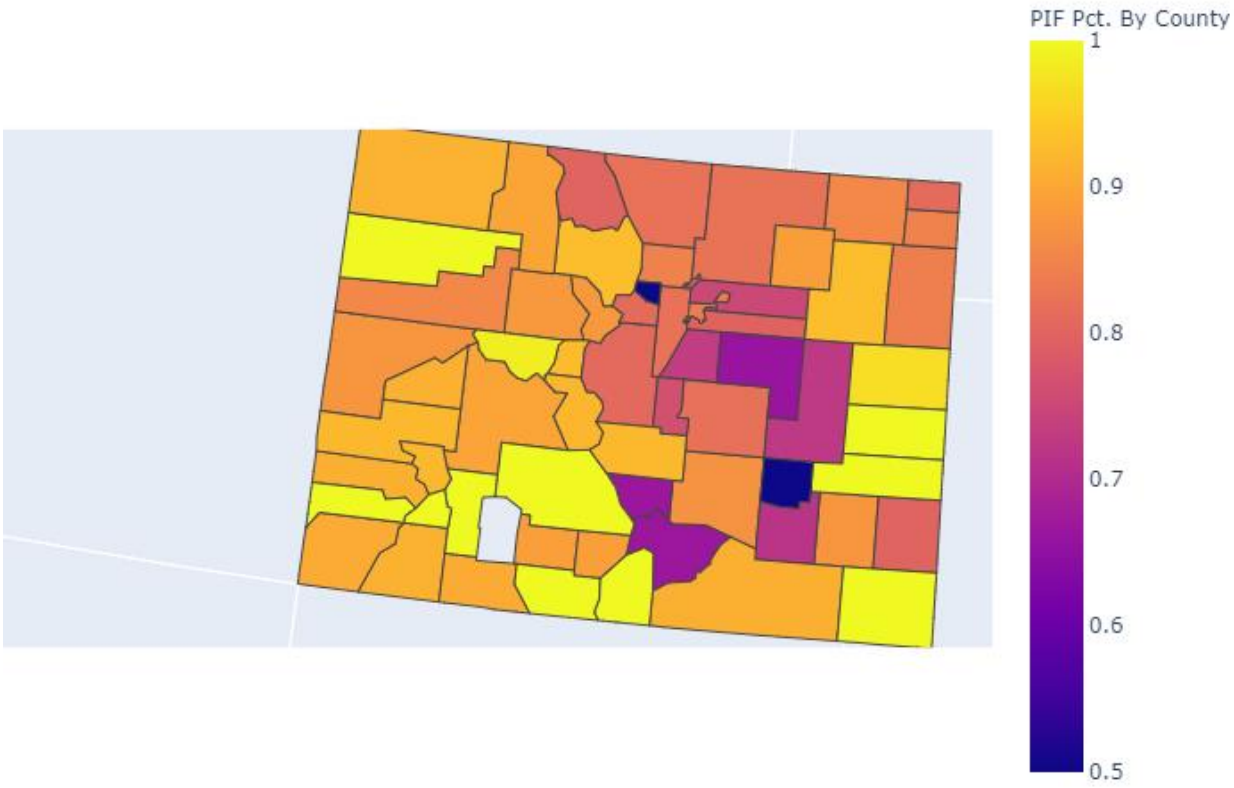


Figure 8 shows the Paid-In-Full Rate for each county, ranging from 50% to near 100%:

Figure 8: Paid-In-Full Rate for Colorado's 63 counties.



Additionally, there are numerous binary features in the SBA dataset, for example Low Documentation, RevLine Credit, Franchise Code, and New Business. To explore these features, they were first converted to binary encoding, and a script was written to calculate a two-proportion z statistic on the 2x2 matrix that results from comparing counts of 1 vs. 0 for the variable of interest, against the Paid vs. Default variable. Table 1 shows the z-stat for the variables, and based on these results FranchiseCode was dropped from the dataset:

Table 1: Z-statistic results for binary features:

Feature	Z-Stat
NewExist	-1.64
RevLineCr	-30.21
LowDoc	11.04
FranchiseCode	0.47

Figure 9 below shows the Paid-In-Full (PIF) rate vs. the loan amount, with 95% confidence intervals shaded in light blue. It's noteworthy that smaller loans have significantly lower PIF rates, and large loans seem to perform above-average:

Figure 9: Paid-In-Full Rate vs. Loan Amount in Colorado.

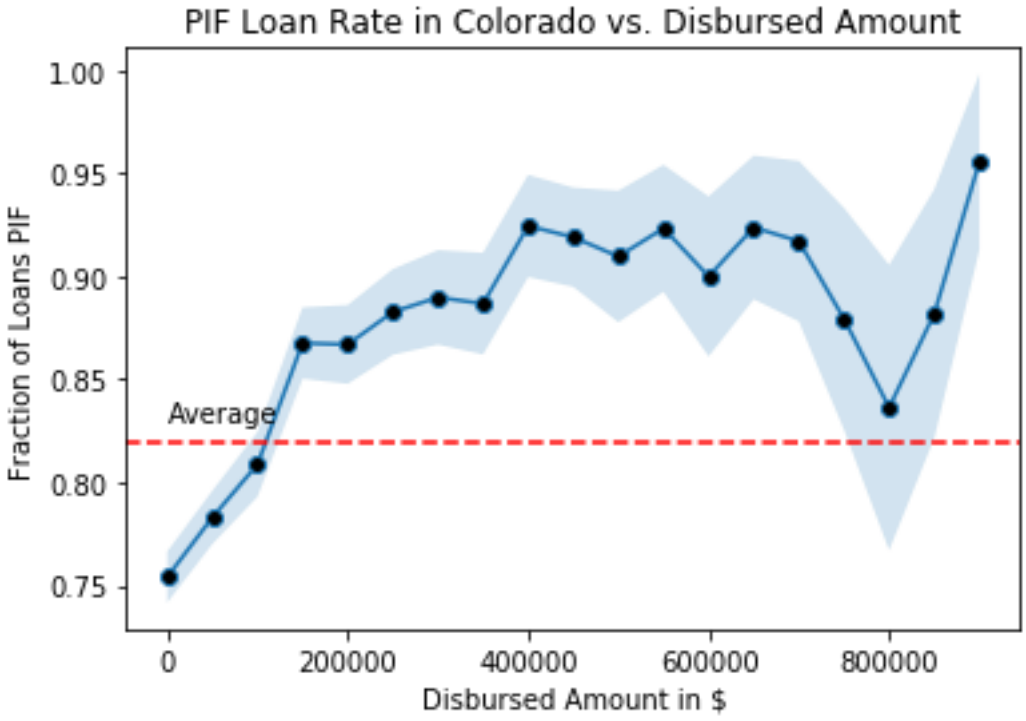
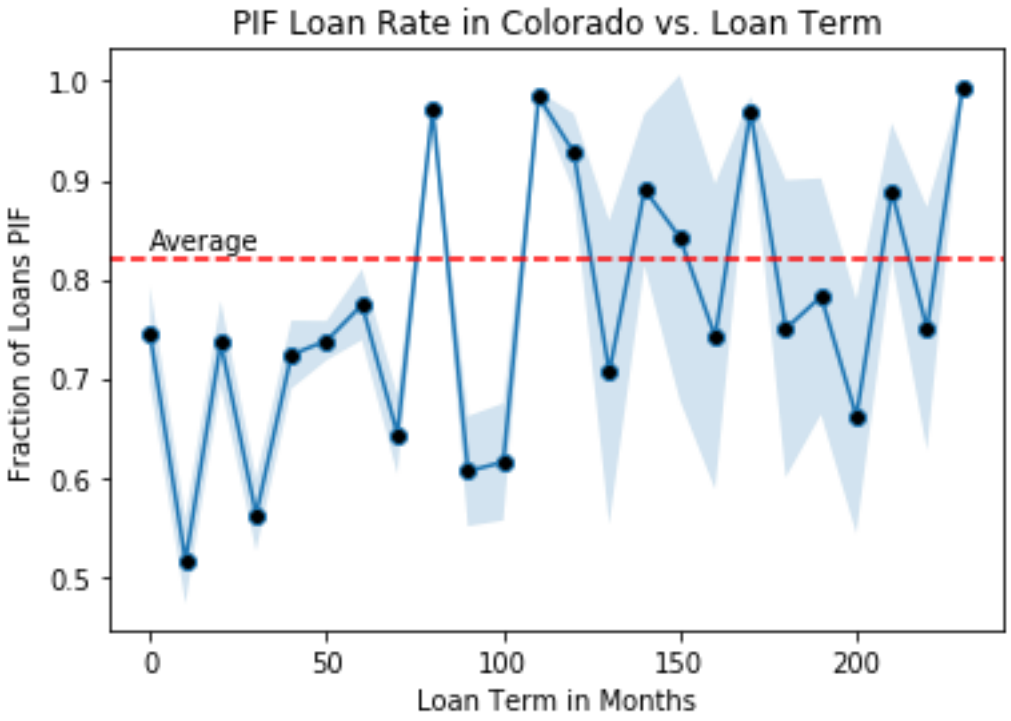


Figure 10 below shows the Paid-In-Full (PIF) rate vs. the loan term, again with 95% confidence intervals shaded in light blue. The highly erratic pattern of PIF rate with loan term seems to be statistically significant and not attributable to normal variation, although the reasons for such behavior are mysterious. Still, it's hoped that a classifier model will be able to make a decision tree that can address this behavior:

Figure 10: Paid-In-Full Rate vs. Loan Term in Colorado.



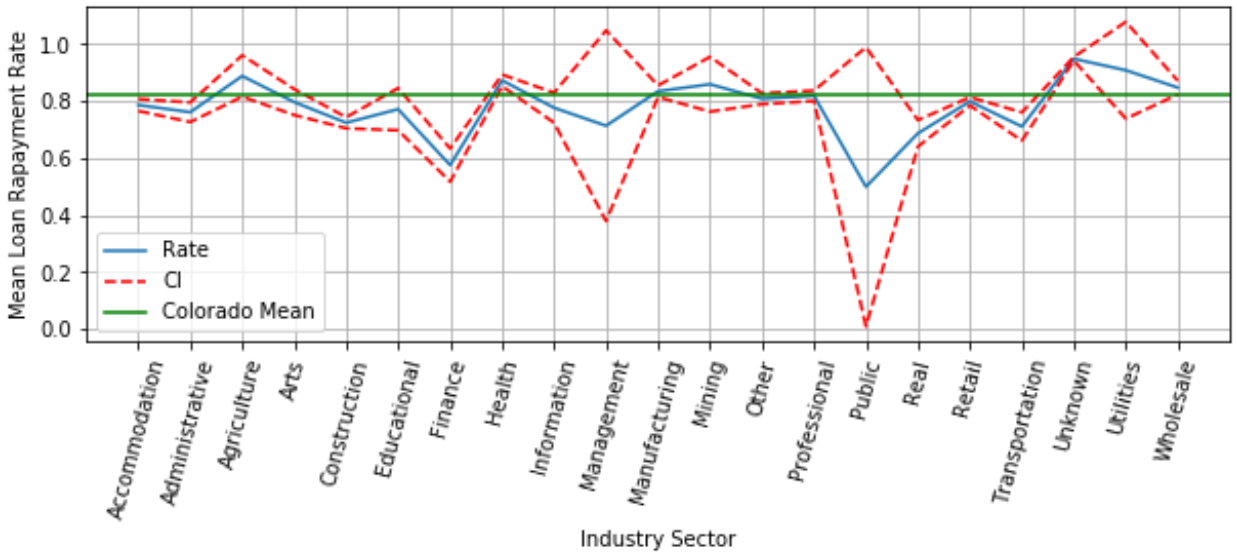
2.2 NAICS Codes

North American Industry Classification System (NAICS) codes are a six-digit categorical framework to describe the entire economy, and NAICS codes are included in the SBA loan dataset. Unfortunately, the wide range of businesses in the dataset makes for thousands of unique NAICS codes. To make this information useable in modeling efforts, the NAICS codes were truncated to the first two digits, then merged with a separate csv file containing the prefix codes for entire industrial sectors; This reduces the number of unique categories to less than 25, and converts the integer values to human-readable titles, which can then subsequently be one-hot encoded.

Figure 11 shows the Paid-In-Full (PIF) Rates from the dataset for each of the resulting categories, along with 95% confidence intervals. **Several industries outperform the average,**

such as agriculture, health, and utilities, while many are found to underperform, such as finance, transportation, construction, real estate and the public sector:

Figure 11: Paid-In-Full Rate for Colorado's principal industrial sectors.



2.3 Addressing feature correlation

Pearson's correlation coefficients were subsequently computed for the array of remaining features in the combined dataset. A very strong correlation was observed between Labor Force in County, Employed in County, and Population. Reasons for this are fairly obvious. **Population and Labor Force features were hence dropped from the dataset**. Figures 12a & 12b show the before and after correlation heatmaps (note that heatmaps are colored by absolute value of the Pearson's coefficient):

Figure 12a: Correlation heatmap before.

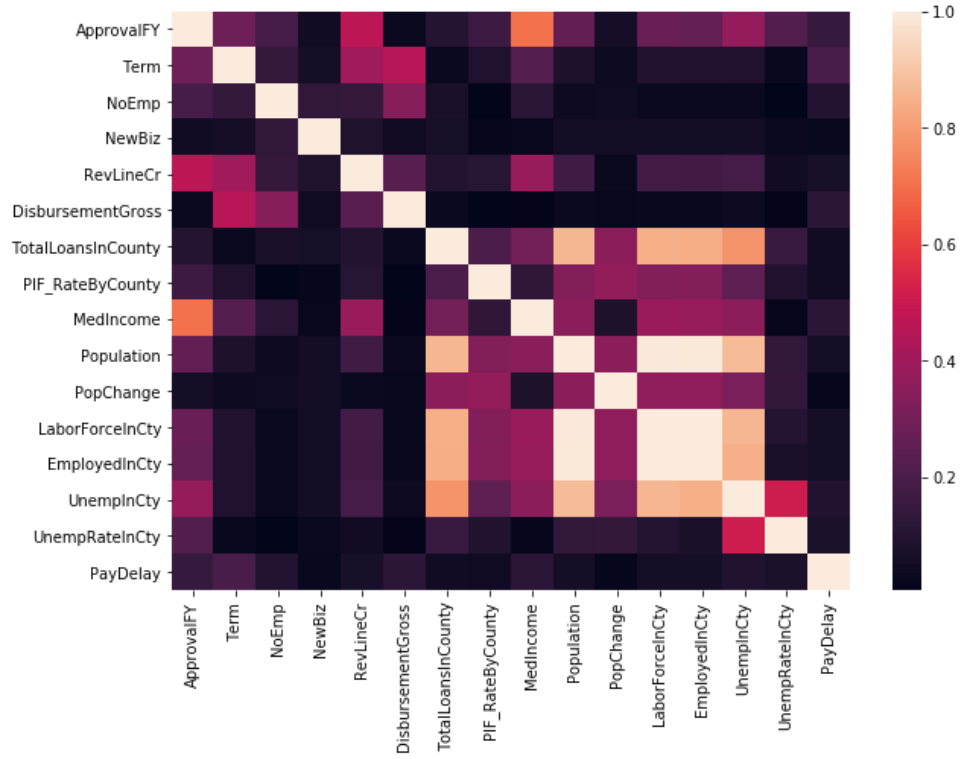
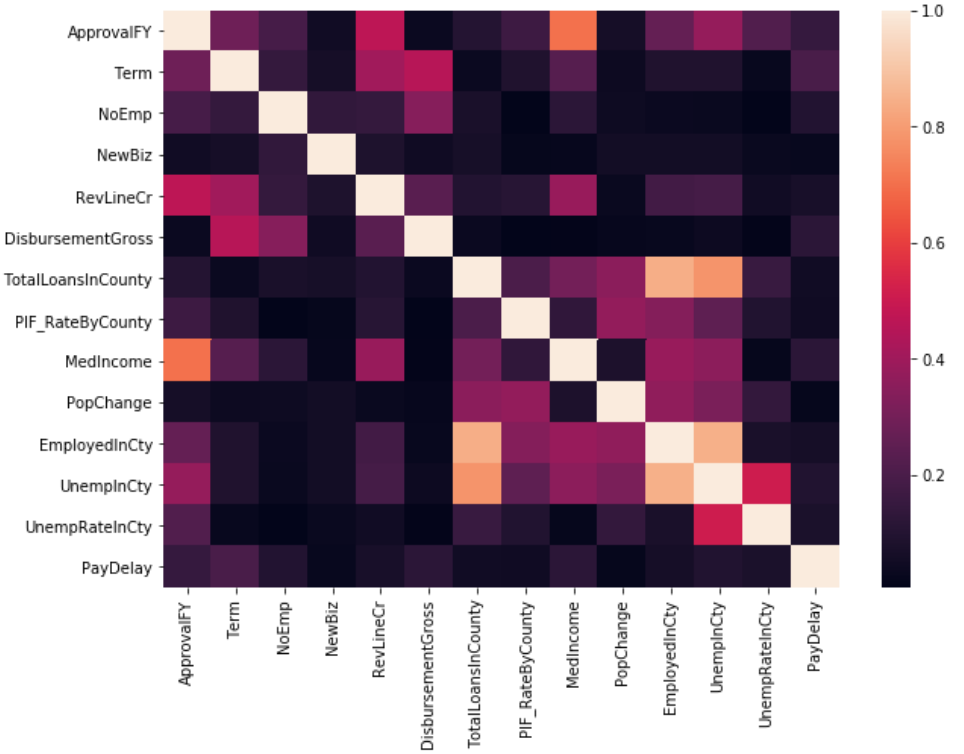


Figure 12b: Correlation heatmap after.



Results

3.1 Models and methods overview

Binary classification models were constructed based on feature column "MIS_Status", which contained the "response" of a committed loan (Paid in Full = 1, Defaulted = 0). As mentioned previously, the dataset contains roughly 83% Paid-off loans and 17% defaulted loans, making for a significantly unbalanced response variable.

A number of models were employed, ranging from two types of dummy classifiers, to logistic regression, to random forest, to Gradient Boost, to Extreme Gradient Boosting.

In each case, models were trained on a subset of data earmarked for training, which comprised roughly 78% of the full dataset. The remaining 22% was a randomly selected hold-out test set, with random seed variables held constant in each case. Since this hold-out test set is randomly selected from the entire dataset, it contains loans spread throughout the entire timespan of the dataset, from 1990 to 2013.

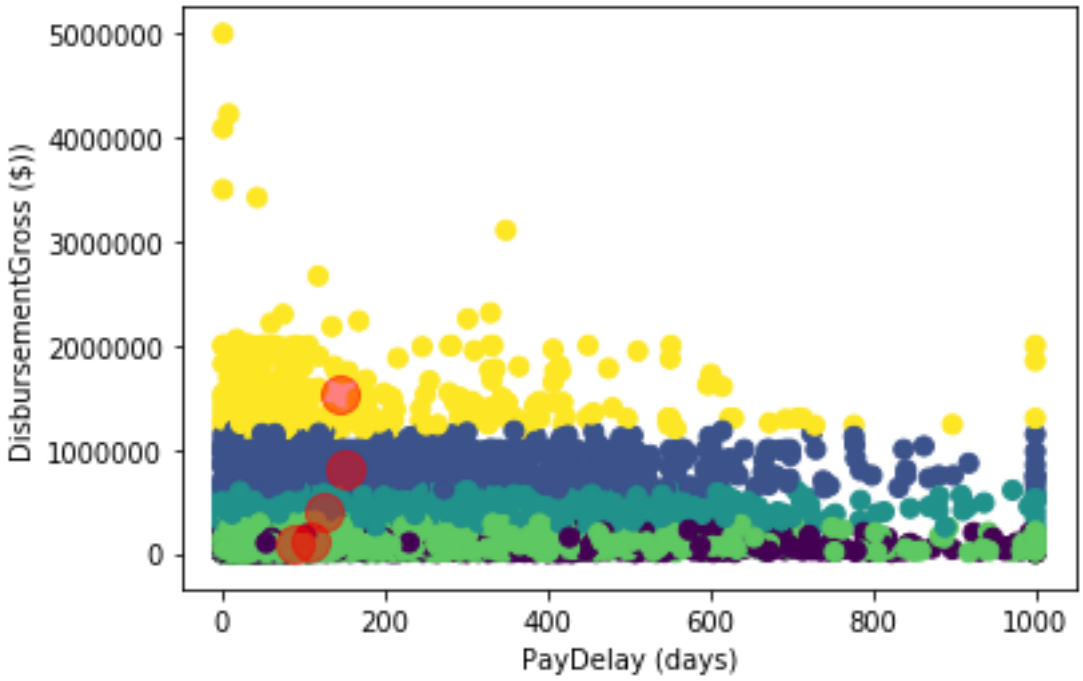
In reality, of course, a useful predictive model for loaning institutions would be trained on previous loans (i.e., data from the past), and would accurately predict the performance of current loan applications (i.e. data from the future, which we don't have yet). With this in mind, the best performing classifier model (based on traditional train-test-split methodologies) was tested by training it on data from 1990-2009 and testing it on data from 2010-2013, to simulate actual usage as much as possible.

3.2 KMeans Clustering

Prior to training on the feature dataset, unsupervised learning was used, specifically the KMeans clustering algorithm, to identify potentially useful clustering. KMeans seeks to

minimize a data point's "inertia" by assigning it to a cluster mean that it is closest to. Based on mean error vs. cluster count analysis, a total of five clusters were used. The potential utility of these clusters was subsequently verified by exploring the "pass/fail" ratios for each of the clusters, similar to exploring relevance of binary variables in the Data Wrangling section 2.1. Figure 13 below shows a scatter plot of PayDelay vs. DisbursementGross, color-coded by cluster, with red dots indicating cluster mean locations.

Figure 13: Scatterplot showing KMeans clusters on dataset (n = 5).



Cluster values, like Industry sector categories, were one-hot encoded immediately prior to modeling efforts, bringing the dataset size to 19237 rows and 41 columns. Finally, all features were run through a Standard Scaler.

3.3 Dummy Classifiers

For starters, model development began with the construction of two base model, using dummy classifiers from scikit-learn. In the first classifier, results are predicted based on a stratified model; that is, each predicted response has a probability of producing a positive result equal to the proportion of positive results in the training set (in this case, roughly 83%). In other words, each response has a 83% chance of being called a good loan, and a 17% chance of being called a default loan. This is equivalent to saying that loan defaults are a purely random event and there are no discoverable input variables. In the second baseline classifier, each response is assigned the dominant value in the dataset, meaning that each response is assigned the value of 1, i.e., a fully paid loan. This is equivalent to saying that "all loans are good loans".

Results for these two classifiers (predicting on randomly selected hold-out test set) are shown in figures 14 a through c. 14a shows the confusion matrix for the stratified classifier, and 14b shows the confusion matrix for the constant classifier. In each case we end up with an identical ROC curve, shown in 14c, which indicates that these models perform no better than a simple random assignment of value (not surprising, since that's what these models are doing).

Figure 14(a): Confusion Matrix for the Baseline Dummy Stratified Classifier.

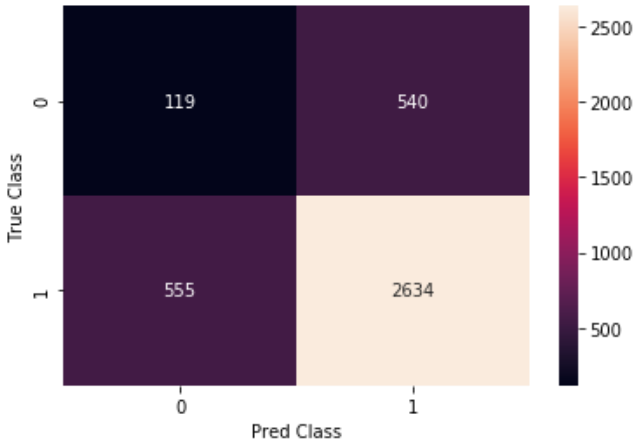


Figure 14(b): Confusion Matrix for the Baseline Dummy Constant-Value Classifier.

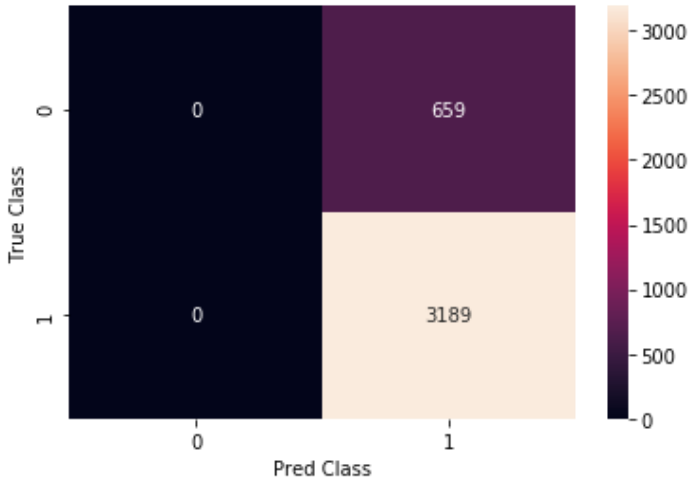
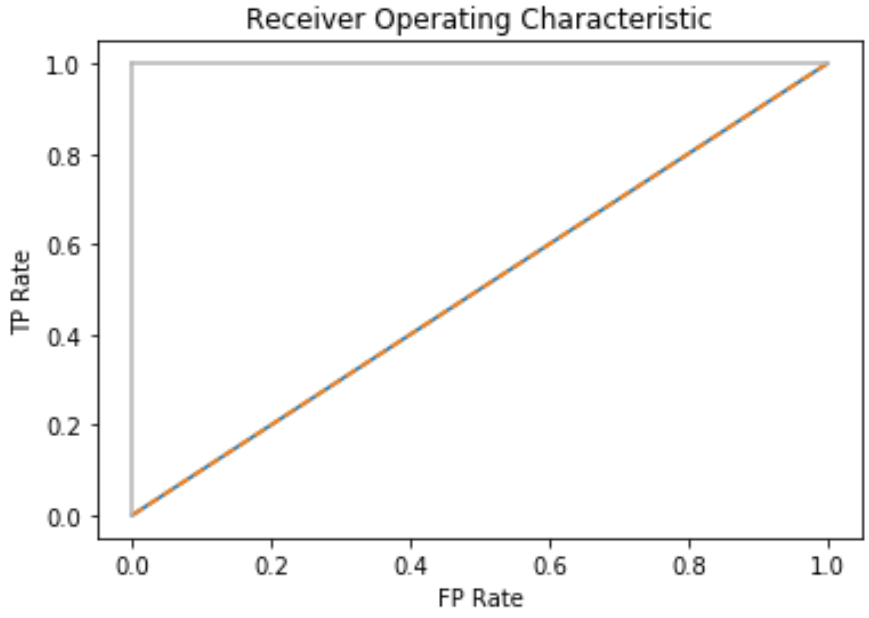


Figure 14(c): ROC Curve for both Dummy Classifiers.



3.4 Logistic Regression

As a first true attempt at predictive capability, a simple logistic regressor was trained on the training set. Figures 15a & b show the confusion matrix and ROC curve. The ROC curve had an Area Under Curve (AUC) score of 73.2%.

Figure 15a: Confusion Matrix for Logistic Regression Model.

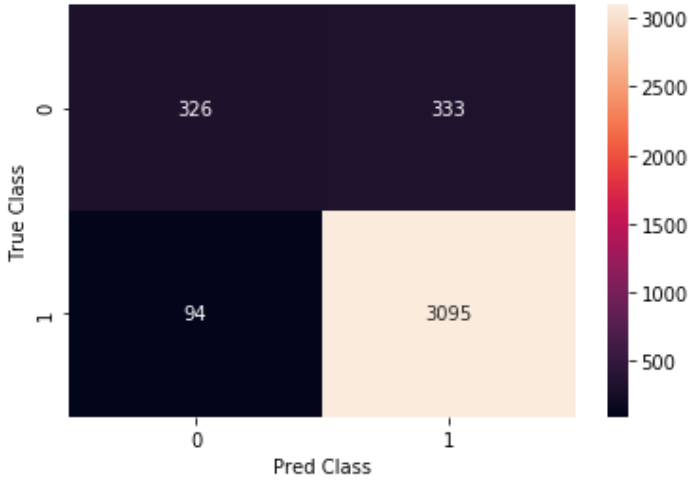
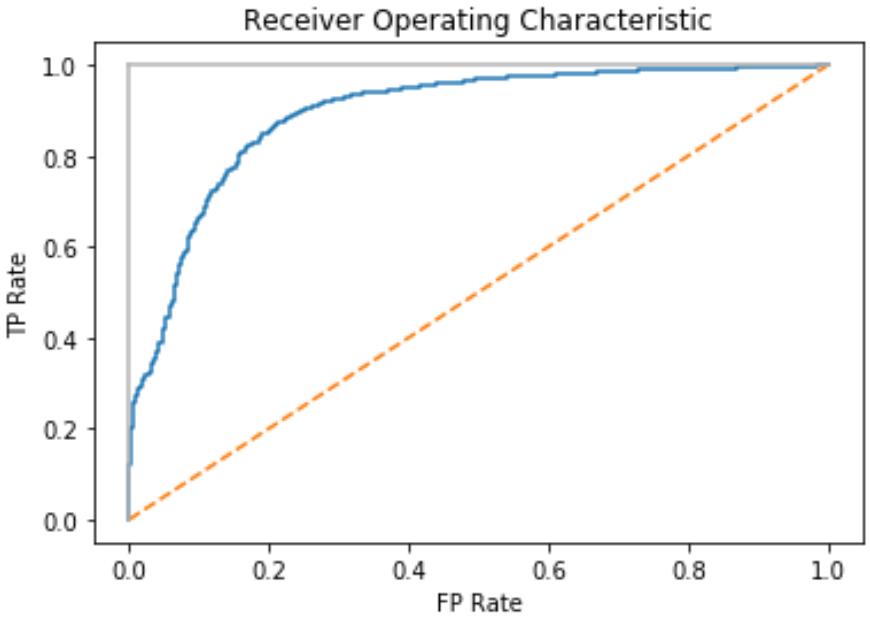


Figure 15b: ROC Curve for Logistic Regression.



3.5 Random Forest (Bagging) Classifier

Scikit-learn's Random Forest Classifier was used to explore the effectiveness of this technique. Random Forest results can vary substantially depending on choices made in modeling parameters. Specifically, the number of estimators was varied from 50 to 150, and both entropy and Gini criteria were explored as well. In the end, entropy criterion was selected combined with 100 estimators, as yielding the greatest accuracy score on the test set.

Random forest did substantially better at test set performance compared to Logistic Regression. Presumably RM is more adept at perceiving more nuanced features, such as Loan Term and Disbursed Amount (figs. 9 & 10, seen previously). Figures 16a & b show the resulting confusion matrix and ROC curve, respectively. RM's AUC score was 84.6%.

Figure 16a: Test set Confusion Matrix for Random Forest Classifier.

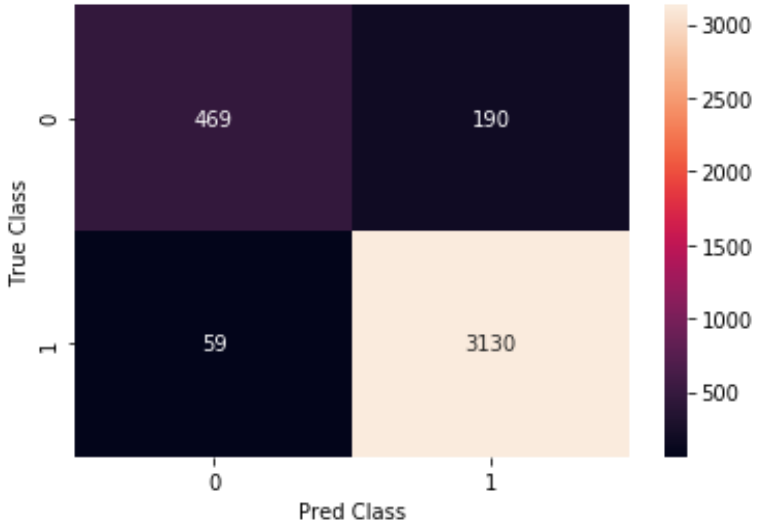


Figure 16b: ROC Curve for Random Forest Classifier.

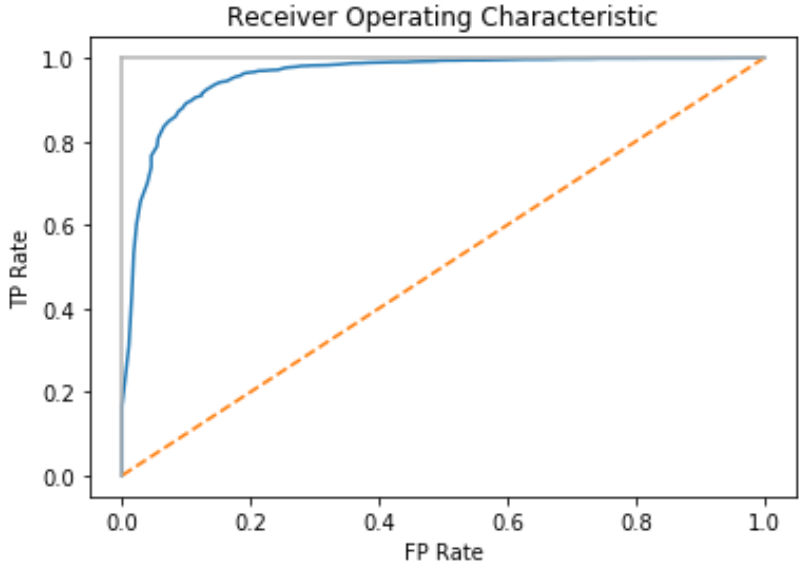
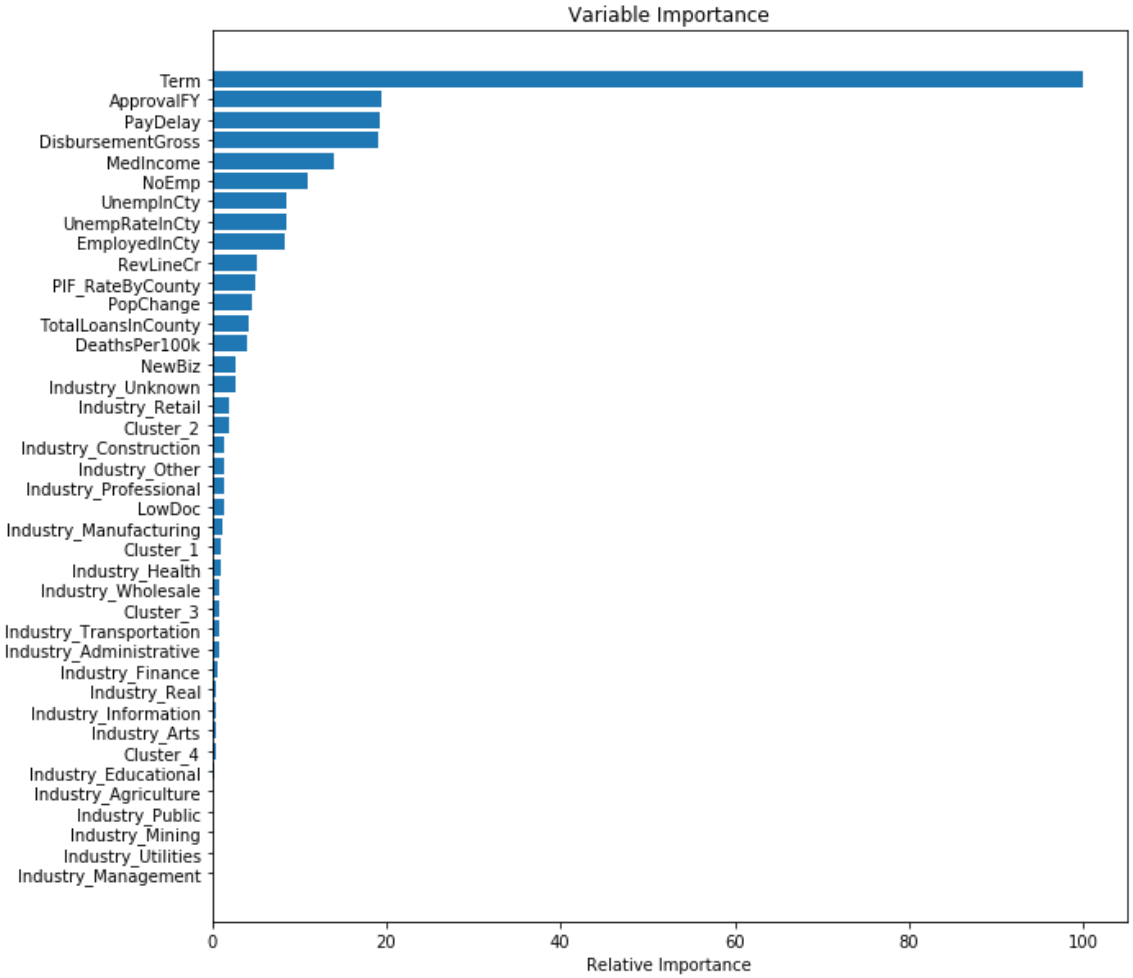


Figure 17: Normalized Pareto of feature importance: Random Forest Classifier.



From Figure 17, the pareto of feature importance, we see that the top five predictive features random forest used were **Loan Term, Approval Year, Pay Delay, Disbursement Gross, and Median Income in County**.

3.6 Gradient Boosting Classifier

Scikit-learn's Gradient Boosting Classifier was employed next. Similar to Random Forest, this algorithm has tunable parameters; These include the number of estimators, the learning rate, the maximum number of features in any tree iteration, and the max depth of each tree iteration. Tuning of these parameters resulted in the optimal set of parameters being 20 estimators, a learning rate of 0.5, max_features of 5 and max_depth also of 5. Figures 18a & b show confustion matrix and ROC curve results. AUC had an accuracy score of 85.1%.

Figure 18a: Test set Confusion Matrix for Gradient Boosting Classifier.

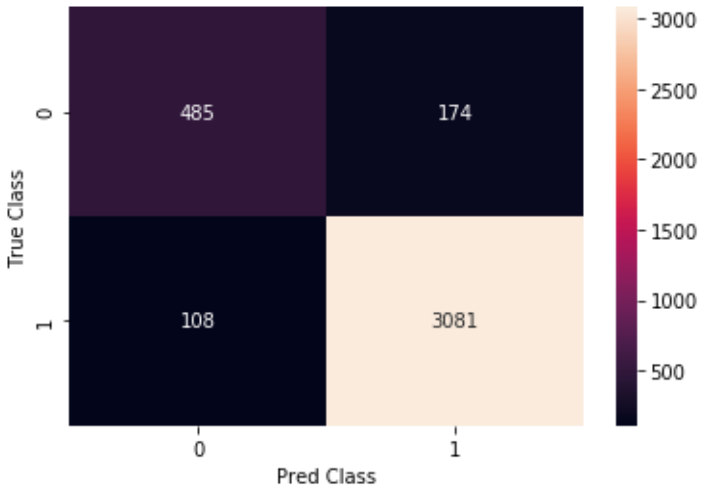


Figure 18b: ROC Curve for Gradient Boosting Classifier.

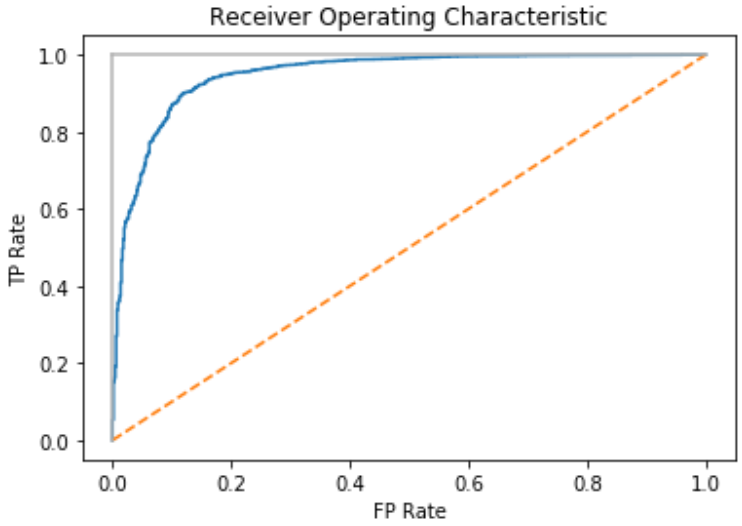
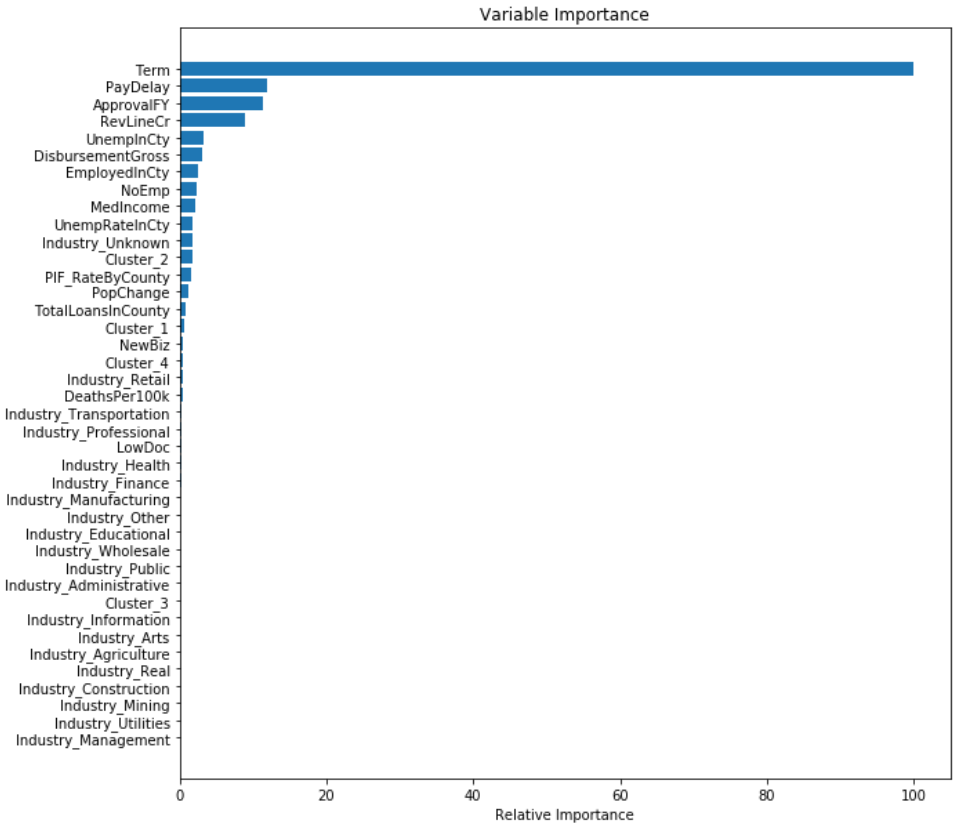


Figure 19: Normalized Pareto of feature importance: Gradient Boosting Classifier.



From Figure 19, the pareto of feature importance, we see that the top five predictive features gradient Boosting used were **Loan Term, Approval Year, Pay Delay, Line of Credit, and Unemployed in County:** The last two being different from Random Forest.

3.7 Extreme Gradient Boosting Classifier (XGBoost)

To explore one final model, XGBoost, an extreme gradient boosting algorithm recently popular in Kaggle competitions, was imported and used. As before parameters were tuned for optimal performance, based on a 3-fold cross validation scheme and a binary logistic objective. Maximum tree depth was set to 5 and the number of boosting rounds was set to 15, a permissible level of computational expense given the relatively small dataset. In all, this model takes less than 10 seconds to run on the training set with a typical laptop computer. Figures 20a& b show the confusion matrix and ROC curve for this model, which achieved an AUC of 90.4%.

Figure 20a: Test set Confusion Matrix for XGBoost Classifier.

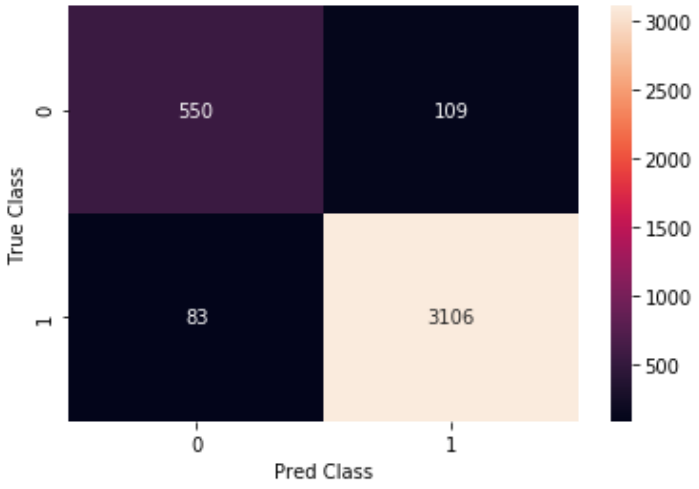


Figure 20b: ROC Curve for XGBoost Classifier.

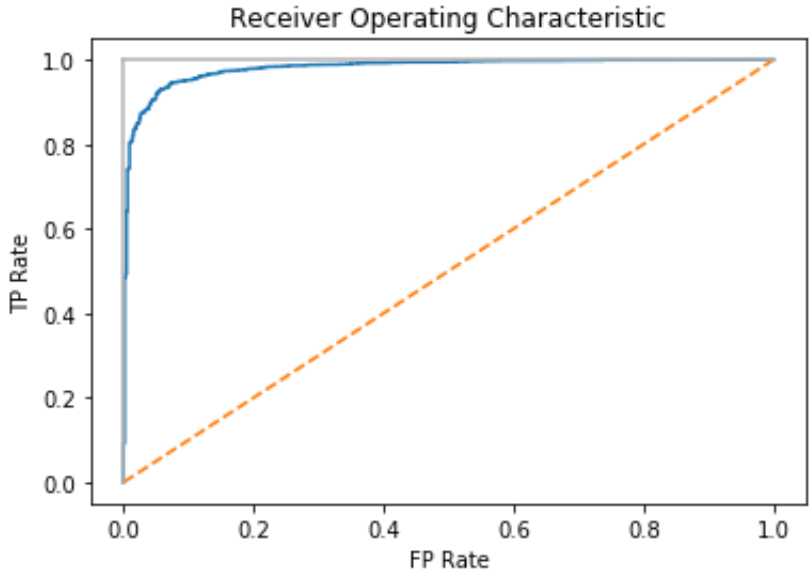
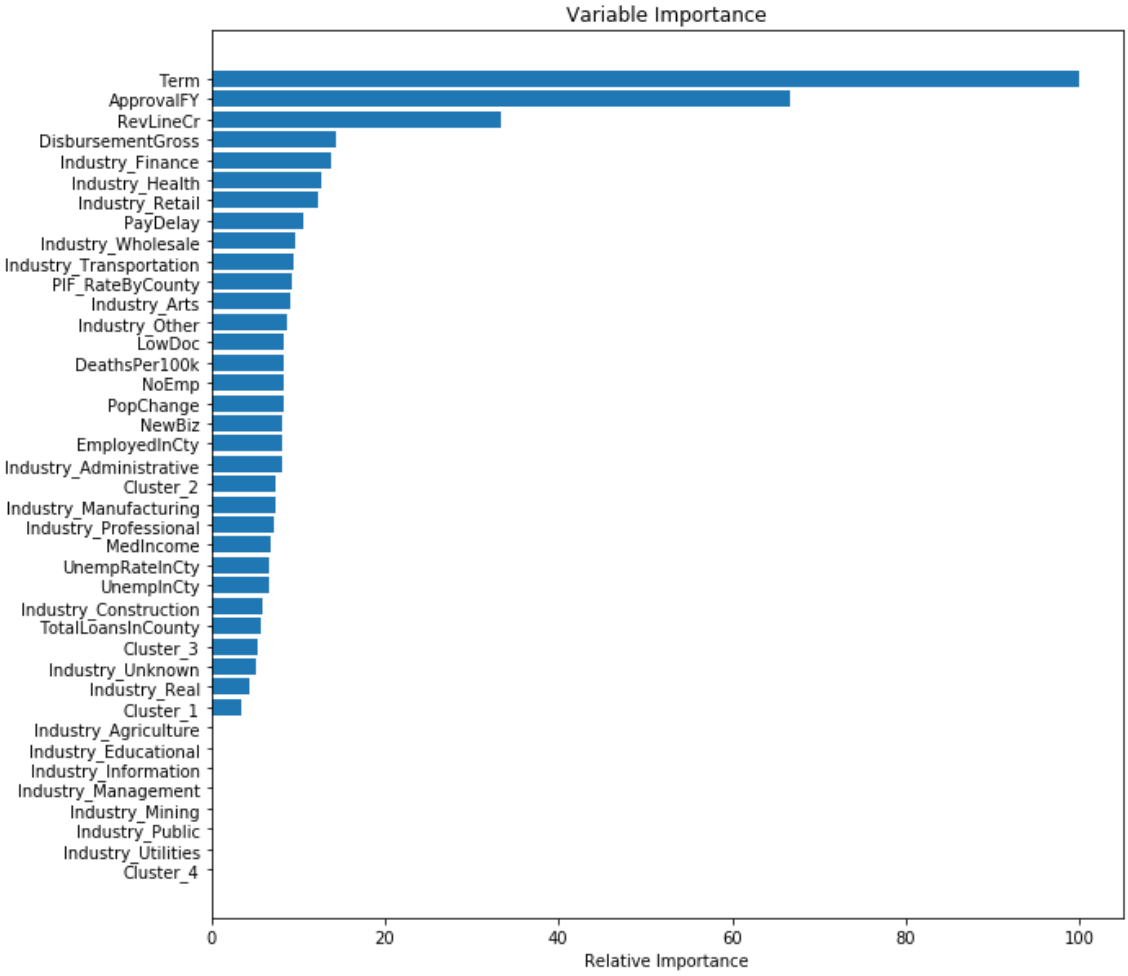


Figure 21: Normalized Pareto of feature importance: XGBoost Classifier.



From Figure 21, the pareto of feature importance, we see that the top five predictive features gradient Boosting used were **Loan Term, Approval Year, Line of Credit, and Disbursement Gross and Finance Industry:** While the first two factors are identical for all models, the prominence of Industry features with XGBoost is interesting. This is corroborated by recent reports of extremely large variation in default rates by industry, (https://www.nerdwallet.com/blog/small-business/study-1-in-6-sba-small-business-administration-loans-fail/) Also recall Figure 11 in which the Finance industry was identified as one of the highest defaulting industries in Colorado.

3.8 XGBoost Trained On Old Data, Tested On New

Splitting the dataset into loans prior to 2010 (Training Set) and loans 2010 and after (Test Set) resulted in slightly less than 1000 loans in the test set. Identical parameters were used for the XGBoost model. Figure 22 shows the confusion matrix for the test set predictions. This model had an AUC score of 90.6%, similar to the previous train-test split.

Figure 22: Latter Year Test set Confusion Matrix for XGBoost Classifier.

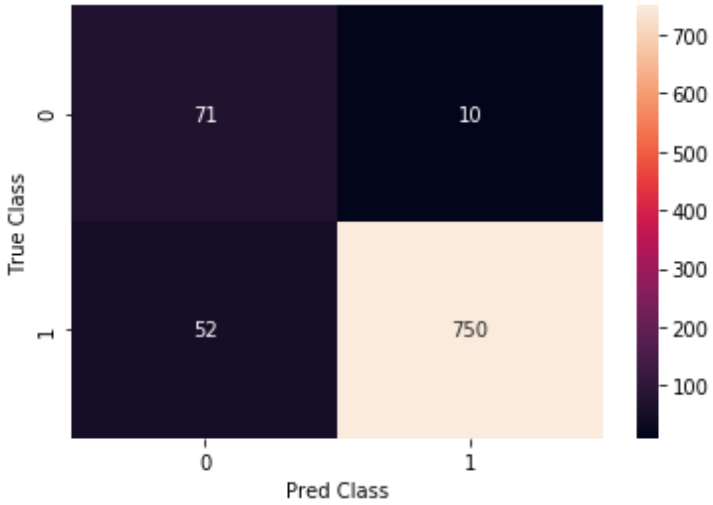
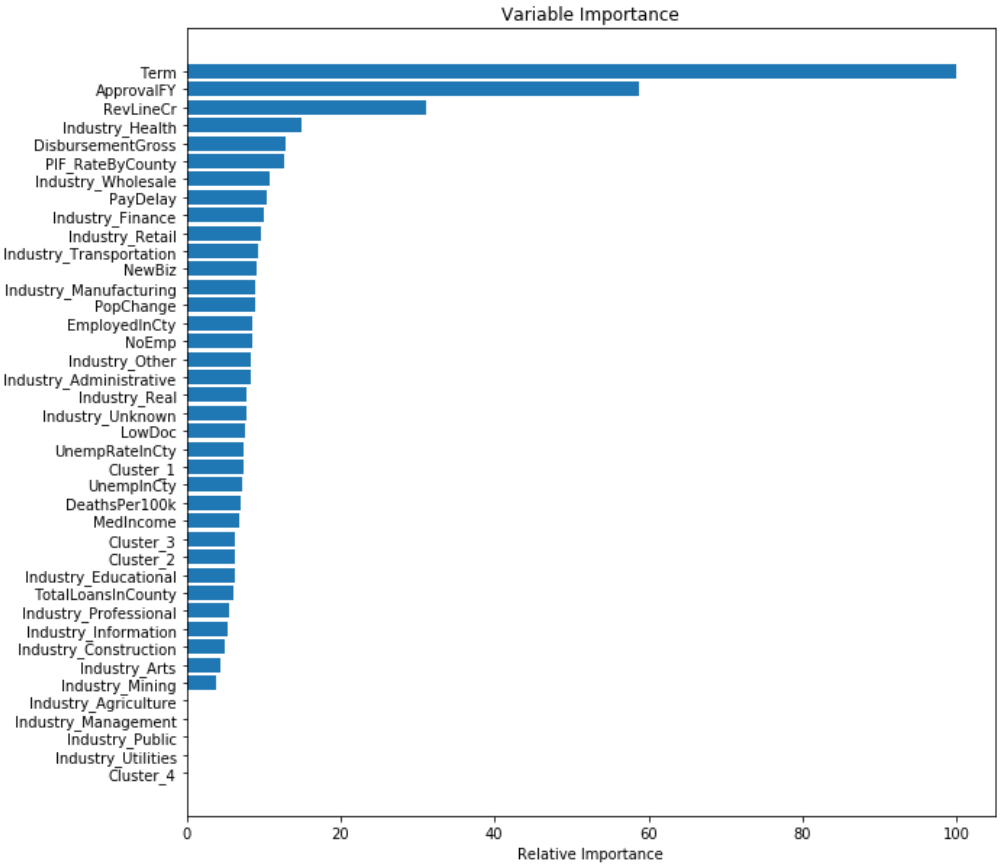


Figure 23: Normalized Pareto of feature importance: XGBoost Classifier (Old vs. New).



4.1 Discussion & Model Comparisons

Table 2 shows a comparison of the various models run in this project, comparing the quantity of True Negatives, False Positives, False Negatives, True Positives, and metrics such as Precision, Recall, f1 score, Accuracy, and Specificity. Due to the unbalanced nature of the dataset, none of the standard array of metric seems to do a satisfactory job of reflecting the goal; namely, the simultaneous minimization of both False Negatives and False Positives.

To address this, a metric called Matthews Correlation Coefficient was used (https://en.wikipedia.org/wiki/Matthews_correlation_coefficient) , and is calculated as follows:

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Matthews Correlation Coefficient (MCC) allows for a model to be properly penalized if either False Positives or False Negatives are excessive. In this way it provides a superior single metric for use in this dataset.

Table 2: Model Performance Comparison.

	tn	fp	fn	tp	Precision	Recall	F1	Accuracy	Specificity	MCC
XGBoost	619	128	95	3487	0.965	0.973	0.969	0.948	0.829	0.821
Gradient Boost	517	230	110	3472	0.938	0.969	0.953	0.921	0.692	0.720
Random Forest	524	223	73	3509	0.940	0.980	0.960	0.932	0.701	0.753
Logistic Regression	355	392	109	3473	0.899	0.970	0.933	0.884	0.475	0.563
Dummy Reg. (Constant)	0	747	0	3582	0.827	1.000	0.906	0.827	0.000	#DIV/0!
Dummy Reg. (Stratified)	129	618	612	2970	0.828	0.829	0.828	0.716	0.173	0.143
XGBoost old v new	74	7	54	748	0.991	0.933	0.961	0.931	0.914	0.698

Two observations become apparent from Table 2:

1. **Using MCC as an overall performance metric, we see that XGBoost out-performed all other models**, by a fairly wide margin (~7% better than the 2nd ranked model, Random Forest). It accomplished this by demonstrating the best ability to minimize both False categories at the same time.
2. In the XGBoost model run of old vs. new (i.e., trained on data before 2010 and tested on later data), the MCC score dropped significantly (~70%) compared to the random test set XGBoost model (~82%). While this model only allowed 7 loans that were destined to default, another 54 loans were incorrectly flagged as default, which cost the model in terms of MCC score.

5.1 Conclusion

By combining data from sources like GoCodeColorado and the CDC WONDER database into the Kaggle SBA Loan dataset, it was demonstrated that Classifier models could be successfully trained on the resulting dataset to yield significant predictive insight into the likelihood of SBA loan default. Models like XGBoost are able to identify loans destined for default with a minimum of error, hence to properly classify the vast majority of loans, and can yield significant business value to both government and loaning institutions.

Additional work could include exploring additional features such as relevant census and crime statistics, the pursuit of loan default data more recent than 2013, the use of more nuanced interpretation of NAICS codes and of course extrapolation of the model to other states.