Predicting Loan Default

Banks and credit card companies constantly face the problem of who to lend money too. They need to lend money to make a return on the money they hold, and often the opportunity to lend comes in the form of personal loans. Not only are these loans risky, because personal loans are not backed by collateral, but they can be costly to underwrite and maintain, because large amounts of data is needed for underwriters to assess the risk of the loans. It is therefore even more important to make sure the people they are lending too can and will pay back. And if possible the amount of data gathered ought to be limited to only what is necessary to assess the risk.

For this project, I will be addressing these issues by building a model to predict loan default on an individual basis. I will be looking at data provided by Lending Club on 850,000+ people taking out a personal loan. The data includes information about a person's credit history, and current standing with the loan (i.e. Fully paid, current,delinquent, default, etc.). The end goal of the project will be to predict the probability a particular person will go delinquent or default on there loan. Having an accurate prediction of a particular person going delinquent on their loan is especially useful for banks and credit card companies to improve the performance of future loan underwriting and get a more accurate view of their future income stream.

Data Wrangling

I acquired data gathered from Lending Club on customers with personal loans of all types. The data was provided as a csv file containing 71 variables of lendee information. After reading the file into Python, I immediately dropped 3 columns not needed for the analysis ('url', 'id', 'member_id'). Member ID and ID are unique individual identifiers with no predictive power. URL is a web address containing each person's ID and isn't useful for prediction.

Next, I looked at determining the percentage of missing values in each of the columns of the dataset. Based on a predetermined percentage of missing values (50% or more), I dropped 21 columns that didn't meet the missing value criteria. I then found all columns that contained continuous numeric data and filled in all missing values with the median for that column. I decided to use the median because we don't know the true underlying distribution and for skewed distributions the mean will be skewed as well and for symmetrical distributions the median and mean are the same.

Below is a list of all columns, with their corresponding percentage of missing values:

'loan_amnt': 0.0,
'funded_amnt': 0.0,
'funded_amnt_inv': 0.0,
'term': 0.0,
'int_rate': 0.0,
'installment': 0.0,
'grade': 0.0,
'sub_grade': 0.0,
'emp_title': 0.06,
'emp_length': 0.05,
'home_ownership': 0.0,
'annual_inc': 0.0,
'verification_status': 0.0,
'issue_d': 0.0,
'loan_status': 0.0,
'pymnt_plan': 0.0,
'desc': 0.86,
'purpose': 0.0,
'title': 0.0,
'zip_code': 0.0,
'addr_state': 0.0,
'dti': 0.0,
'delinq_2yrs': 0.0,
'earliest_cr_line': 0.0,
'inq_last_6mths': 0.0,
'mths_since_last_delinq': 0.51,
'mths_since_last_record': 0.85,
'open_acc': 0.0,
'pub_rec': 0.0,
'revol_bal': 0.0,
'revol_util': 0.0,
'total_acc': 0.0,
'initial_list_status': 0.0,

'out_prncp': 0.0,
'out_prncp_inv': 0.0,
'total_pymnt': 0.0,
'total_pymnt_inv': 0.0,
'total_rec_prncp': 0.0,
'total_rec_int': 0.0,
'total_rec_late_fee': 0.0,
'recoveries': 0.0,
'collection_recovery_fee': 0.0,
'last_pymnt_d': 0.02,
'last_pymnt_amnt': 0.0,
'next_pymnt_d': 0.29,
'last_credit_pull_d': 0.0,
'collections_12_mths_ex_med': 0.0,
'mths_since_last_major_derog': 0.75,
'policy_code': 0.0,
'application_type': 0.0,
'annual_inc_joint': 1.0,
'dti_joint': 1.0,
'verification_status_joint': 1.0,
'acc_now_delinq': 0.0,
'tot_coll_amt': 0.08,
'tot_cur_bal': 0.08,
'open_acc_6m': 0.98,
'open_il_6m': 0.98,
'open_il_12m': 0.98,
'open_il_24m': 0.98,
'mths_since_rcnt_il': 0.98,
'total_bal_il': 0.98,
'il_util': 0.98,
'open_rv_12m': 0.98,
'open_rv_24m': 0.98,
'max_bal_bc': 0.98,
'all_util': 0.98,
'total_rev_hi_lim': 0.08,
'inq_fi': 0.98,

'total_cu_tl': 0.98,
'inq_last_12m': 0.98

Some of the columns contained numeric data in string format, for example, the column ‘term’ contains the length of the loan in months but is shown as ‘36 months’. For this and the column ‘issue_d’, I created a function that would strip off letters and convert the data to an integer. I then created a new column with the integers. Below are the original columns:

issue_d	term
Dec-2011	36 months
Dec-2011	60 months
Dec-2011	36 months

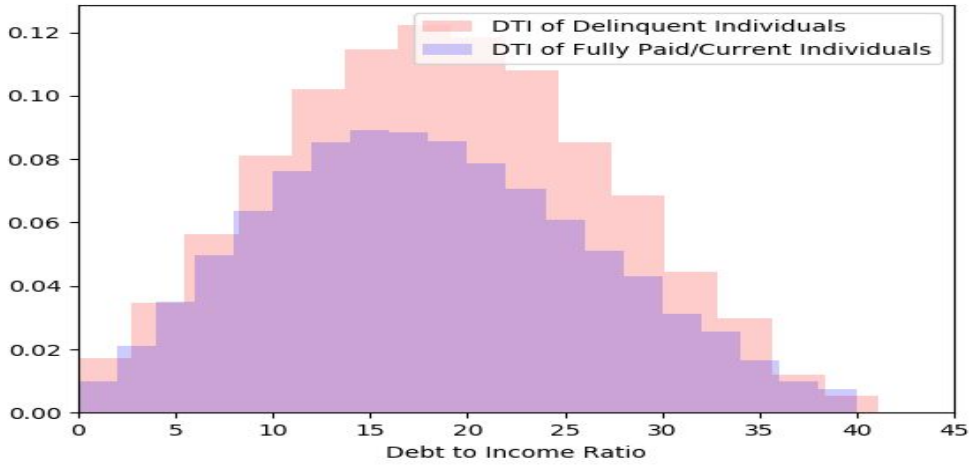
Here are the new columns created from stripping out the numbers:

issue_year	term_int
2011	36
2011	60
2011	36

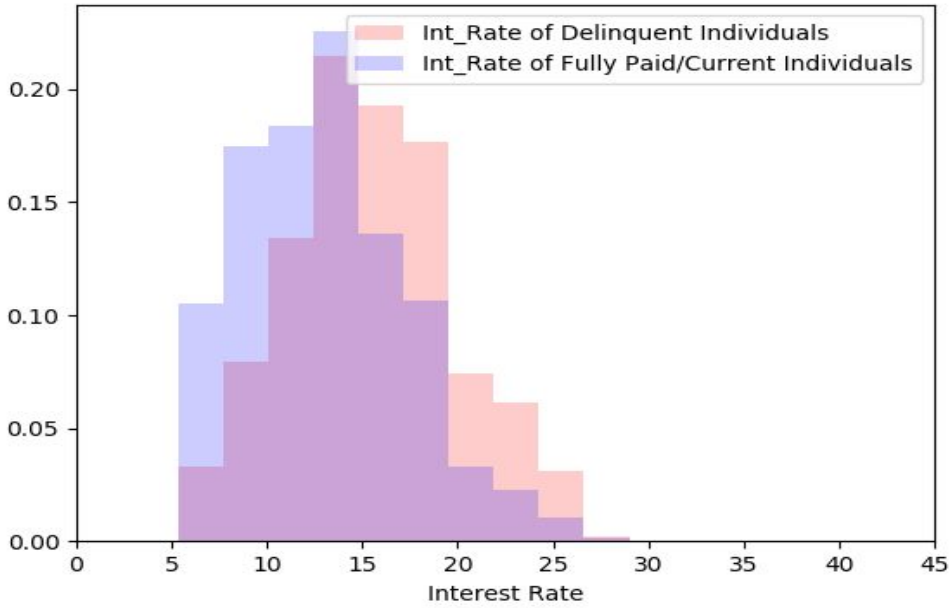
After deciding on the end goal of predicting lendee delinquency, I created a new indicator column called ‘paid_unpaid_ind’ that is 0 if an individual is delinquent or 1 if an individual is fully paid or current on their payments.

For the last step of wrangling the data, I wanted to visualize the variables and correlations within the dataset. To do this, I created multiple graphs and pivot tables. I started with variables that I thought would be predictive of future delinquency such as: debt to income ratio, annual income, Lending Club credit rating, employment length, number of open credit accounts, interest rate on loan, home ownership, and categorical purpose of the loan. Below are a few examples of these graphs:

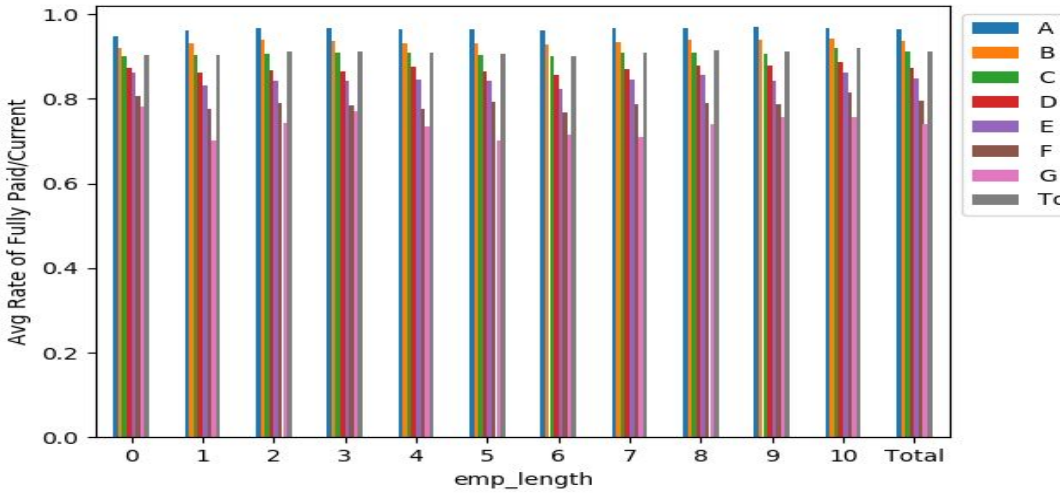
Histograms of Debt to Income Ratios for Delinquent and Fully Paid Groups



Histogram of Interest Rates for Delinquent and Fully Paid Groups



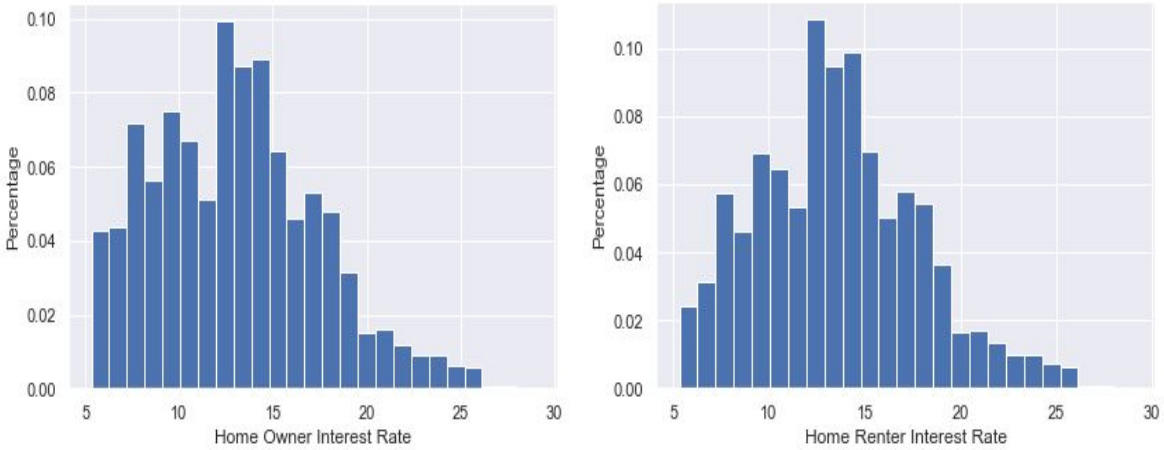
Paid Rate by Employment Length and Credit Rating



Inferential Statistical Analysis On Lending Club Data

In this section, I want to cover my analysis that went into discovering other useful insights that might aid in assessing the best risks for lending money.

After fitting a Random Forest Model, interest Rate showed to be a significant predictor in determining default rate. I want to further investigate, using inferential statistics, any relationships or other findings between borrower groups. First, I looked at comparing homeowners and home renters to see if there were any differences in the interest rates charged. Interest rates for home owners average 13.026% and 13.576% for renters. Let's look at some graphs to get a good look at the distributions.



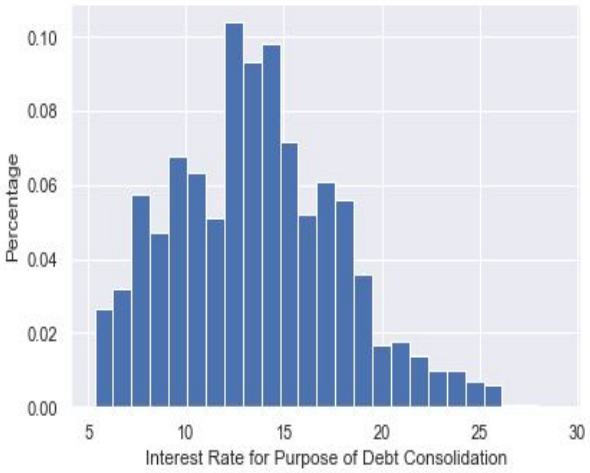
Just looking at the graphs and data it is difficult to tell if these groups are really different. To determine if these are significantly different I will use the following hypotheses using alpha=0.01:

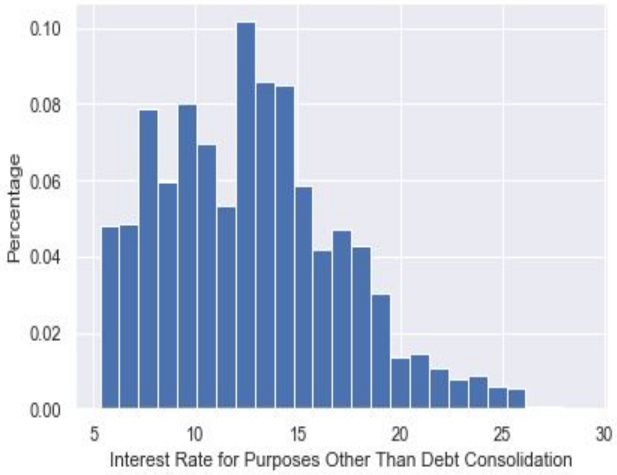
H_o: The mean difference between the interest rate for home owners and home renters is 0

H_a: The mean difference between the interest rates are not 0

First it's important to note that the distributions, based on the graphs above, they don't appear to fit a normal distribution so a t-test may not be appropriate. Instead I will use bootstrapping to test the null hypothesis. To do a bootstrap hypothesis test, I took the interest rates for both groups and subtracted out the mean for each respective group. Then I added back in the mean of interest rates for both groups together. This shifts the mean for each group to be equal to the overall mean. Now I can bootstrap sample from each of these groups (with shifted means) and compare the difference in means to the observed difference. Assuming the null hypothesis is true, I wouldn't expect to find a significant amount of samples as or more extreme than the observed difference. From this test, I found that none of the 10,000 samples were more extreme than what was observed, so I conclude that there is a statistically significant difference between the means of each group. On average, home owners will pay 0.55% lower interest rate than home renters.

Next, I looked at interest rates between individuals requesting a loan for purposes of debt consolidation vs. any other reason. Below are histograms of the two groups.





As with the previous analysis, I used bootstrapping to compare the means of each group with the hypothesis:

H_o: The mean difference between the interest rate for purposes of debt consolidation and anything else is 0

H_a: The mean difference between the interest rate for purposes of debt consolidation and anything else is not 0

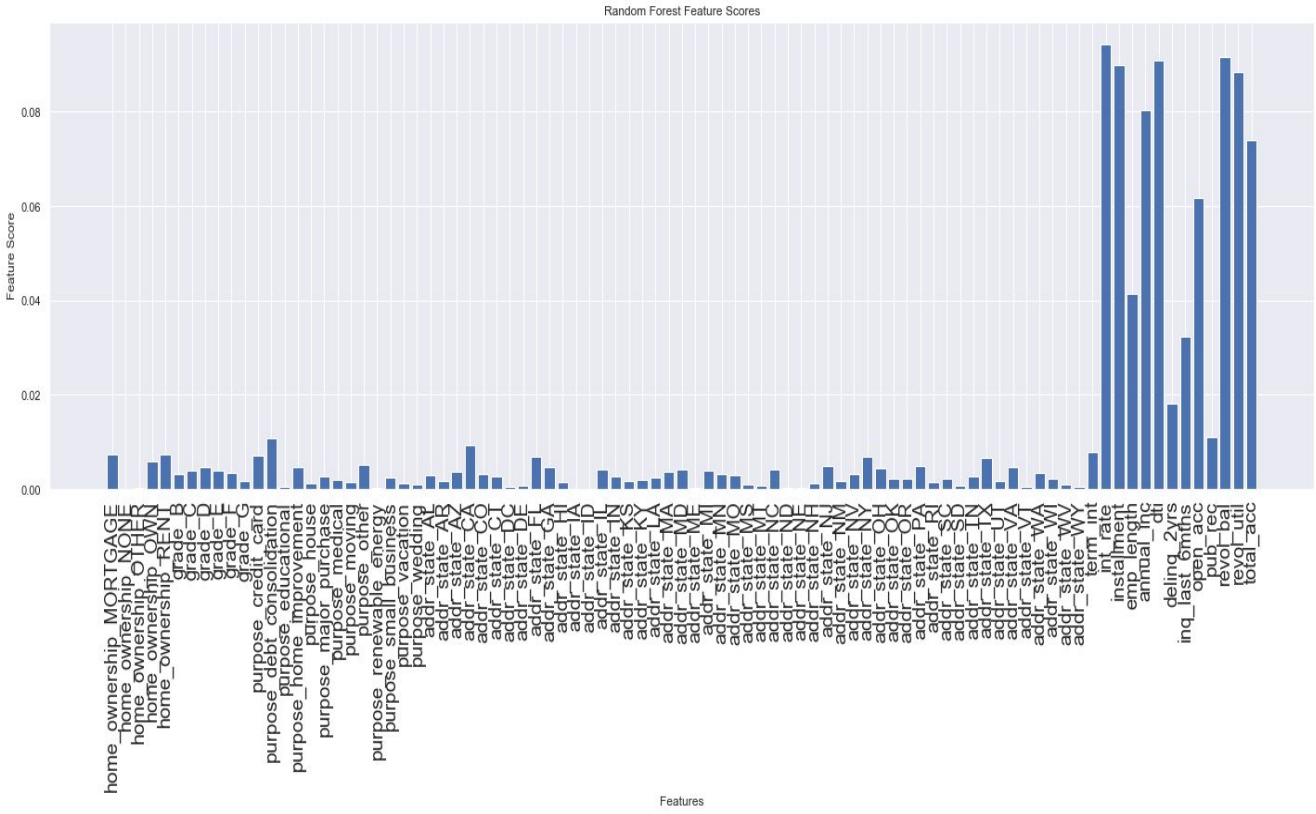
The p-value came in well below our stated alpha of .01, indicating there is a significant difference between mean interest rates for people who request a loan for debt consolidation and other purposes. On average, people who request a loan for debt consolidation pay a 0.87% higher interest rate.

Machine Learning

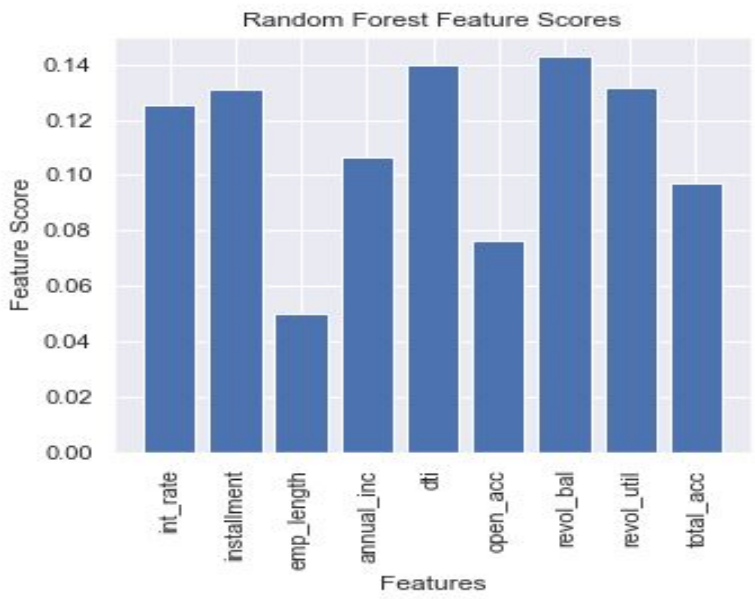
To predict loan default, I trained multiple models to the data gathered from the Lending Club data set. In this section, I want to cover the process of fitting and evaluating each of the models and any insights gained from the process.

I began with fitting a Random Forest Classifier to the data and looking at the most predictive features in the model. Interest rate, installment amount, employment length, annual income,

debt to income ratio, number of open credit accounts, revolving balance, revolving utility, and total credit accounts are the most predictive of loan default.



I trained the first model with with eighty six features, however, most of these don’t add predictive power to the model. We can simplify the model by just keeping the most predictive features. Doing this will improve processing speed of each model run and won't diminish predictive power significantly. The Random Forest Classifier correctly predicted 91% of the outcomes in the test data, using all the variables. Below are the remaining features to be used in in each of the models trained on the Lending Club data, this model maintained 91% prediction accuracy.



Now that we have the features we want to keep, I then looked at comparing different models to see if we can get better performance. For performance measurements, I looked at a confusion matrix, 5-fold cross validation ROC, and a Lift Graph for each of the models.

The Confusion Matrix looks at the model accuracy rates for true positive, true negative and the inaccuracy rates for false positive and false negative. In other words, the Confusion Matrix shows the rate at which the model predicted the lendee would pay on time (1) and they did, as well as the rate at which the model predicted the lendee would default (0) and they did default.

ROC stands for Receiver Operating Characteristic, the ROC curve contains true positive rate on the Y axis, and false positive rate on the X axis. This means that the top left corner of the indicates a true positive rate of 1 and a false positive rate of 0. Therefore a curve with a larger area under the curve (AUC) is better. For evaluating multiples models, I used ROC AUC with 5-fold cross validation. This means that I took the training data set and separated it into five pieces, where four chunks are used to train the model and the last chunk is used to validate the model and calculate the ROC AUC. This is then done five times, each time using a different chunk for the validation set. Then I calculated the average and standard deviation of the five scores to use as a comparison metric.

The Lift Graph shows how much better the model predictions are above randomly guessing at each of the probability percentiles. The Y axis is the lift measurement, and means if you used predictions at X percentile you would achieve Y times better accuracy than random guessing.

Below are each of the performance measures.

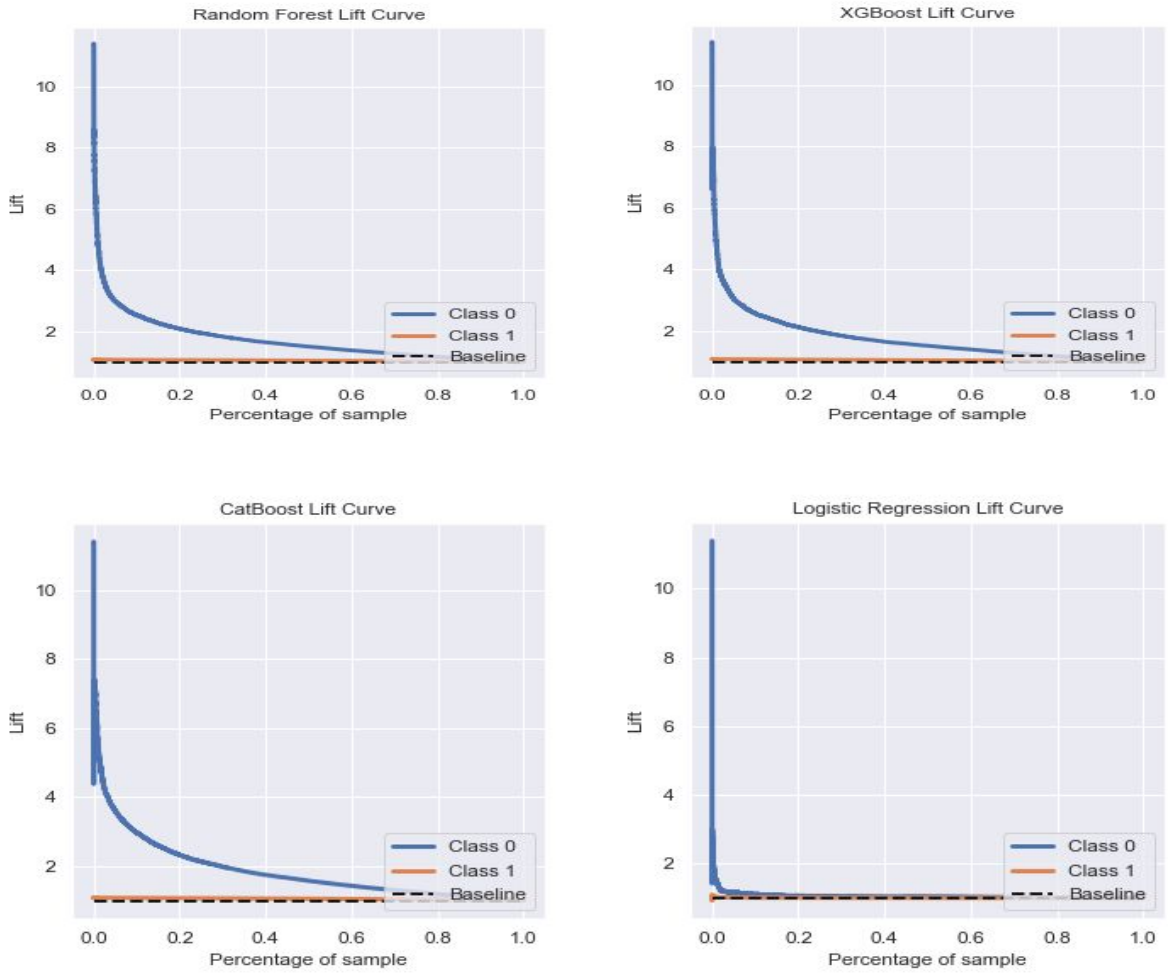
Confusion Matrices

Predicted								
	Random Forest		XGBoost		CatBoost		Logistic Reg	
Actual	0	1	0	1	0	1	0	1
0	0.000699	0.087119	0.000059	0.087759	0.001947	0.085871	0.000009	0.087809
1	0.000212	0.911970	0.000009	0.912173	0.000960	0.911222	0.000036	0.912146

Area Under the ROC Curve:

	Classifier			
ROC AUC	Random Forest	XGBoost	CatBoost	Log Regression
Mean	0.6981	0.7046	0.7285	0.5209
Std. Deviation	0.8355	0.8394	0.8535	0.7217

Lift Curves



Scores from the confusion matrices are very close, but looking at the AUC and Lift graphs, the CatBoost Model performs the best.

Conclusion

We are able to predict loan default with 91% accuracy and save on costs by limiting the data gathered to 9 variables: Interest rate, installment amount, employment length, annual income, debt to income ratio, number of open credit accounts, revolving balance, revolving utility, and total credit accounts. In addition, this model doesn't require obtaining a FICO score for each individual which saves dramatically on costs.