Predicting Financial Distress

Sanjog Sharma

Aug 12, 2015

In the Kaggle competition called Give Me Some Credit, contestants were asked to predict if somebody would face financial distress in the next two years, this paper describes my solution which relies on cleaning data and imputing missing values, creating features and ensembling models.

1. Introduction to the problem

Accorfing to the competition’s introduction:

‘Banks play a crucial role in market economies. They decide who can get finance and on what terms and can make or break investment decisions. For markets and society to function, individuals and companies need access to credit.

Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted. This competition requires participants to improve on the state of the art in credit scoring, by predicting the probability that somebody will experience financial distress in the next two years.

The goal of this competition is to build a model that borrowers can use to help make the best financial decisions.’

The competition provided historical data for 250,000 borrowers. The data was broken into test and train data. Training data contains an additional feature: ‘Serious delinquency faced in the past two years’, the objective of the competition is to calculate the probability of the same for the borrowers in test data.

The table below lists the features provided by the competition:

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Description** | **Type** |
| **SeriousDlqin2yrs** | **Person experienced 90 days past due delinquency or worse** | **Y/N** |
| RevolvingUtilizationOfUnsecuredLines | Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits | percentage |
| age | Age of borrower in years | integer |
| NumberOfTime30-59DaysPastDueNotWorse | Number of times borrower has been 30-59 days past due but no worse in the last 2 years. | integer |
| DebtRatio | Monthly debt payments, alimony,living costs divided by monthy gross income | percentage |
| MonthlyIncome | Monthly income | real |
| NumberOfOpenCreditLinesAndLoans | Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards) | integer |
| NumberOfTimes90DaysLate | Number of times borrower has been 90 days or more past due. | integer |
| NumberRealEstateLoansOrLines | Number of mortgage and real estate loans including home equity lines of credit | integer |
| NumberOfTime60-89DaysPastDueNotWorse | Number of times borrower has been 60-89 days past due but no worse in the last 2 years. | integer |
| NumberOfDependents | Number of dependents in family excluding themselves (spouse, children etc.) | integer |

1. Preprocessing

Both data sets had multiple missing observations in the ‘Income’ and ‘Number of dependents’ fields. The average and the dependent in both sets was close to zero, and the most frequent dependent was also zero. I used Scikit learn’s preprocessor to impute the most frequent observation(Zero) to fill the values.

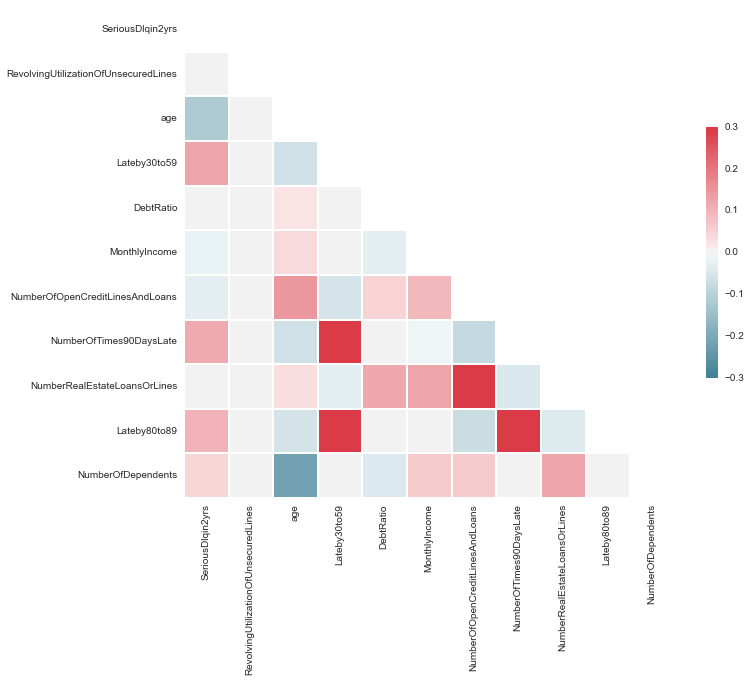
I used a Random Forest Regressor to impute income. I used existing features, and a Random Forest Regressor with the following parameters

RandomForestClassifier(n\_estimators = 100, max\_features=8, oob\_score=True, random\_state=1)

This imputed the income with a RMSE of 1533.6. Other models (unscaled KNN Regressor, for example, performed much worse RMSE of 14,454)

Other features, despite not having missing data, had a lot of nonsensical data. Total credit utilization ranged from 0-5,000,000%. Debt Ratio had similarly absurd percentages: (0-32966400%). There were 269 values where total late days in all of 30-59, 60-89, and 90+ days were filled with either 96 or 98.

This is the heatmap of correlation of features after imputing missing income and number of dependents:



Due to the high correlation between number of late days and default, and since default essentially is being 120 days past due, I had to impute these values thoughtfully.

To do so, I separated values with high late days. For those that did not contain the erroneous late days, I added the number of late days – and hence engineered my first feature. Additionally, I took the log of income, credit utilization, past30-59, past60-89, past 90 and sum of total late days. Once again, I used the Random Forest Regressor to impute late days with RMSE of 1.01. I fitted the regressor over the values with logical total late days and used to to fit total late days for those with high late days values.

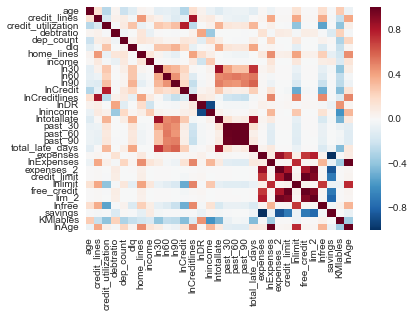
For credit utilization, I used the same approach. I separated borrowers with credit utilization percentage higher than 250% and used Random Forest Regressor. This time, the results were not as successful as I achieved an RMSE of about 28% (.28).

3) Feature Extraction:

Income and debtratio calculate total debt payment/month. Which, in turn could be used to calculate total free credit. The difference between debt payment/month and income can be a crude metric for total savings, and credit utilization% and total credit payment would give a crude idea of the credit limit of a borrower. I created all these features, and used their nautral log as another metric. I substituted .1 for values of zero and under while calculating natural log.

I also used K-means clustering to cluster borrowers into 5 clusters and used that as another feature.

Here’s an updated heatmap with new features:



-

3. Ensemble Selection:

For my first logistical regression using features provided by the competition I ended up with AUC of .692 using a 10-fold cross evaluation; a Random Forest Classifier with all competition provided features and two extra features (total late day, sum late days), I ended up with an AUC of .84955. For other models, I only evaluated on accuracy of predicted class, and here are the results:

|  |  |
| --- | --- |
| Model | Accuracy% |
| GradientBoostingClassifier | 93.8% |
| Logistical Regression | 93.5% |
| RandomForestClassifier | 93.6% |
| ADABoost | 93.7% |
| Stochastic Gradient Dissent | 93.55 |

The final model didn’t perform well (a AUC score of .850).

1. Your challenges and successes

The biggest challenge was cleaning the data sufficiently. I believe that the features I created, if calibrated well, would perform well. However, due to limited number of existing features and heavily interlinked features made it a difficult task. I intend to repeat the competition with better data.

1. Conclusions and key learnings

The key insights I derived from this project was the importance of have high quality data. Had I imputed my data with more thought and deliberation, I might gave ended with a better result using a few models instead an ensemble of many.