IIT Kanpur

Data Mining Project

Give Me Some Credit

Anupreet Porwal
Aakash Ghosh
Anshuman Mitra

Supervisor:
Prof. Amit Mitra

August 17, 2015

Give Me Some Credit

Aakash Ghosh Anupreet Porwal Anshuman Mitra

Abstract

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.In the following project we looked at different credit scoring algorithms and our aim was 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.

Keywords: Credit Scoring

1. Introduction

We have used a historical data set available at Kaggle.com provided under a contest "Give me some credit".A labelled raw data set(with missing entries) consisting of 150,000 borrowers is provided.We divided the cleaned data set into two part : Train and test data set with around 80,000 and 35,000 entries.Our aim was to train a classification algorithm using train data set and we labelled test dataset by assigning by assigning probabilities to each borrower on their chance on defaulting on their loans in the next two years and check the accuracy of the algorithm by looking at AUC values and ROC curves and confusion matrices. The information of the various features associated with a borrower is given below:

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

In the following sub sections we will look at data visualisation using Principal Component Analysis for the data set and cleaning of the raw dataset.This will then be followed by the Methodology of various algorithms used in section two with conclusion and results in section three.

1.1. Data Visualisation

We use Principal Component Analysis to visualise the data and see rough clusters present if any .The data given to us consist of 11 features but we can not plot them to see if there are any cluster or a pattern being formed .So we use Principal component Analysis to reduce the dimension of data to three.The first three principal components accounted just 47% of the total variance in the data.The three dimensional plot of the same is shown below.The red points corresponds to defaulters while black corresponds to non defaulters As evident from the graph it does not have very distinct clusters but at the same time we know that the three pc's capture only 47% of the total variance and so it does not explain the exact picture here.

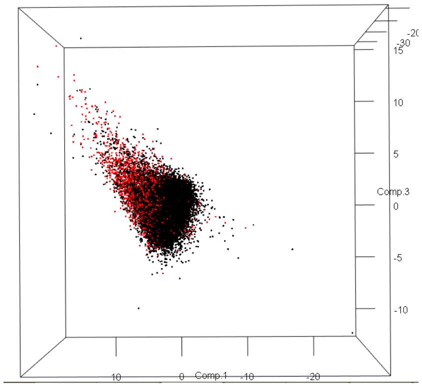


Figure 1: PCA plot of the data set

1.2. Cleaning and Organizing the data

The data provided by Kaggle was anonymous data taken from a real-world source and hence, it is expected that the input contains errors.Some of the quantitative values within the data set were actually coded values that had qualitative meanings. For example, under the column NumberOfTime30-59DaysPastDueNotWorse, a value of 96 represents Others, while a value of 98 represented Refused to say. The observation with these as entries were deleted from the data set. Similarly, only complete rows were taken into account. Rows with any entry as NA were deleted. After the following procedures were performed we were left with still about 115,000 observations. These were then divided into train and test dataset containing 70% and 30% of the datapoints respectively.

2. Methodology

2.1. Performance Measurement

We used ROC plots and the corresponding AUC score as a measure of the classification performance. ROC plot is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings.The area under a ROC curve (AUC) quantifies the overall ability of the test to discriminate between those individuals who are defaulters and those who are not defaulters.

2.2. ***Support Vector Machine***

We initially implemented linear SVM model on the given dataset, but we found out that the data was not linearly separable. Even after the maximum number of iterations, a suitable hyperplane was not found. We then searched for non-linear SVM techniques, and decided to use the standard RBF kernel function. We found out that along with a high number of true positives, it generated a unusually high number of false positives, that is, almost everyone was considered worthy of credit. We then tried to artificially balance the dataset by repeating points from the minority class, but performance remained unchanged. We later found out that the problem was in the skewness of our dataset (there were 14 times more zeros than ones), and that SVM classification performance drops significantly for imbalanced datasets. Since support vectors remained unchanged, performance also remained unchanged on increasing data points.

	Observed=0	observed=1
Score=0	32341	2120
Score=1	82	63

After some searching, we found that a popular approach towards solving these problems is to bias the classifier so that it pays more attention to the positive instances. This can be done, for instance, by increasing the penalty associated with misclassifying the positive class relative to the negative class. Another approach is to preprocess the data by oversampling the majority class or undersampling the minority class in order to create a balanced dataset. As both of these approaches required significant changes in the dataset we had, we decided to move ahead with other classification methods.

2.3. ***Trees Without Prior***

We then used the basic model of decision trees to get a set of rules that can be used to classify the two classes.Trees can be used to clean variables, find splits in cut offs of other variables, break data in segments, and offer simple insights. They are perfect for generating straw credit policies for rule based systems for quick and dirty needs.We fit a decision tree without any prior probabilities.The decision tree generated is shown below :

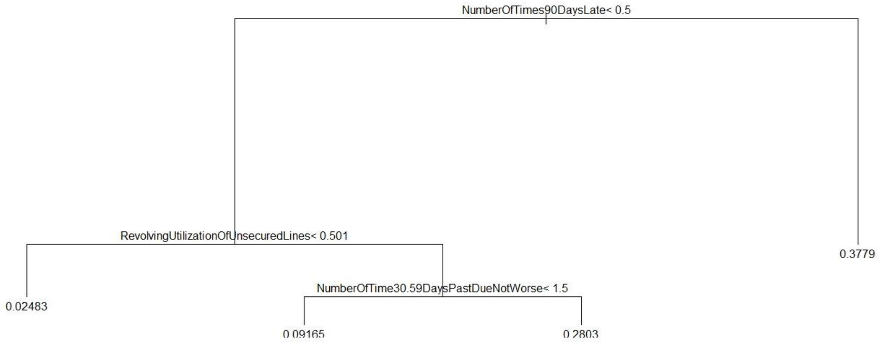

NumberOfTimes90DaysLate< 0.5
RevolvingUtilizationOfUnsecuredLines< 0.501
0.3779
NumberOfTime30.59DaysPastDueNotWorse< 1.5
0.02483
0.09165
0.2803

Figure 2: Decision Tree without prior

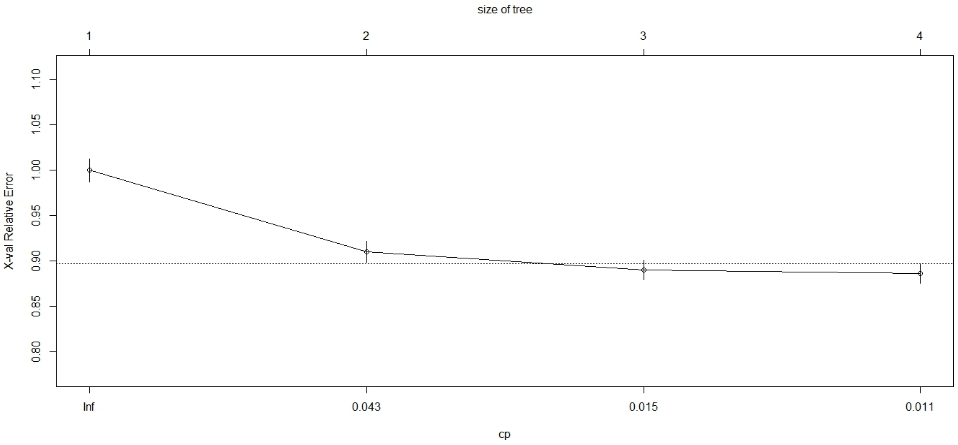


Figure 3: Cost Complexity Plot

The AUC score of the above tree was 0.7643.In terms of modeling rare events like fraud or low default credit portfolios or this problem using prior probabilities to configure trees can help improve performance. Therefore we next look at Trees with initial prior probabilities given.

2.4. ***Trees with Prior***

Prior probabilities can improve classification results by helping to resolve confusion among classes that are poorly separable, and by reducing bias when the training sample is not representative of the population being classified Here the training sample has almost 14 times zero entries as compared to entries with one.Therefore we take initial priors as 90/10 for constructing more high performance trees.The AUC of the following tree came out to be 0.786 which was greater than AUC score of the tree without prior,as expected.The tree was prunned with cost complexity parameter as 0.004.The tree that was generated as a result is shown below.

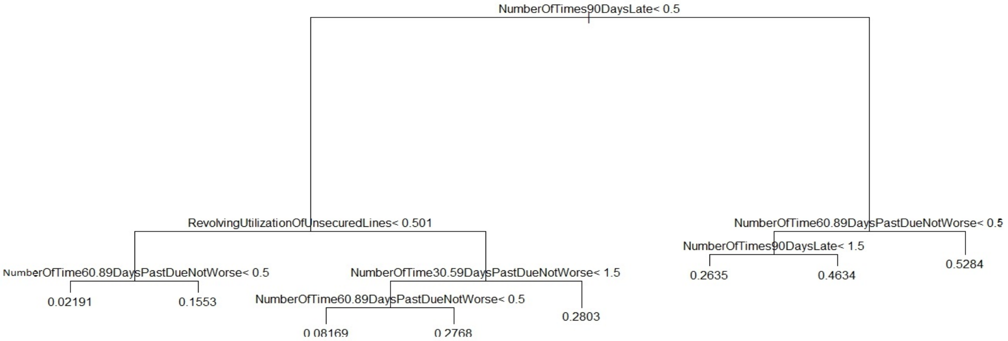

NumberOfTimes90DaysLate< 0.5
RevolvingUtilizationOfUnsecuredLines< 0.501
NumberOfTime60.89DaysPastDueNotWorse< 0.5
NumberOfTimes90DaysLate< 1.5
NumberOfTime60.89DaysPastDueNotWorse< 0.5
NumberOfTime30.59DaysPastDueNotWorse< 1.5
0.2635
0.4634
0.5284
0.02191
0.1553
NumberOfTime60.89DaysPastDueNotWorse< 0.5
0.2803
0.08169
0.2768

Figure 4: Decision Tree with prior as 90/10

Thus the comparison of the two trees can be done by looking at their ROC plots shown below.Clearly,The tree with prior outperforms the tree without prior.

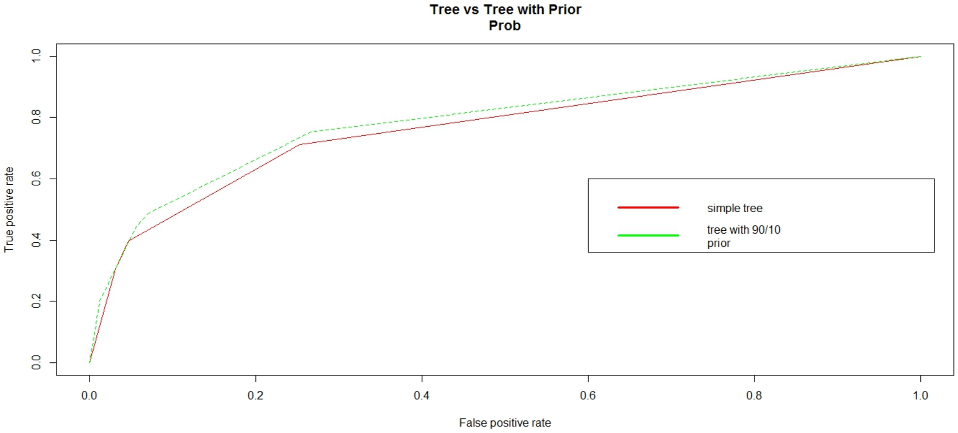


Figure 5: ROC plot of Tree with prior vs without prior

The main advantage of the trees is that they can be easily interpreted.Being non parametric in nature we did not need to worry about outliers or whether the data is linearly seperable.However,a closer look at the two trees help us to see one of the main disadvantages of decision trees i.e. they easily tend to overfit the data .For example,NumberOfTimes90DaysLate ¡0.5 does not make much sense as the only value less than 0.5 for this variable is 0. So the rules given by the following decision tree are not very useful to classify the customers.That's where ensemble methods like random forests (or boosted trees) come in.we will now look at the Random forest algorithm and its analysis.

2.5. ***Random Forest***

Random forest (Breiman, 2001) is an ensemble of unpruned classification or regression trees, induced from bootstrap samples of the training data, using random feature selection in the tree induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble. Random forest generally exhibits a substantial performance improvement over the single tree classifier such as CART.We now apply Random Forest on out data set with 300 trees,choosing 3 variables at random for any tree building and we get an AUC score of 0.826 which is significantly better as compared to simple decision tree thus It now do not over fit the data and with the increase in number of trees,it has better generalisation.The out of bag estimate of the error rate was 6.1% and the confusion matrix is given below:

	Observed=0	observed=1
Score=0	75177	582
Score=1	4342	644

The class one and class two error are 0.007 and 0.87respectively.Clearly it does not classify the one's entries nicely. Also,an important observation to note is that the OOB estimate of error is 6.1% which also happens to be the percentage of ones observation with respect to the total number of observations.RF suffer from the curse of learning from an extremely imbalanced training data set. As it is constructed to minimize the overall error rate, it will tend to focus more on the prediction accuracy of the majority class, which often results in poor accuracy for the minority class.

Several ways are used to handle this problem.For the Random Forest, artificially making class priors equal either by down-sampling the majority class or over-sampling the minority class is usually more effective with respect to a given performance measurement.However,down-sampling the majority class may result in loss of information, as a large part of the majority class is not used.

Here we use over sampling techniques by replicating the minority class observation many times so that we have equal number of observations of the two classes.This does not increase the information given to us but attaches more weight to the minority class and the reduces the problem of skewness of the dataset .In our case we replicate observations with one as the response roughly 14 times to balance the data set.The OOB estimate of the error is 1.43 while the confusion matrix is as follows:

	Observed=0	observed=1
Score=0	73611	2148
Score=1	0	74790

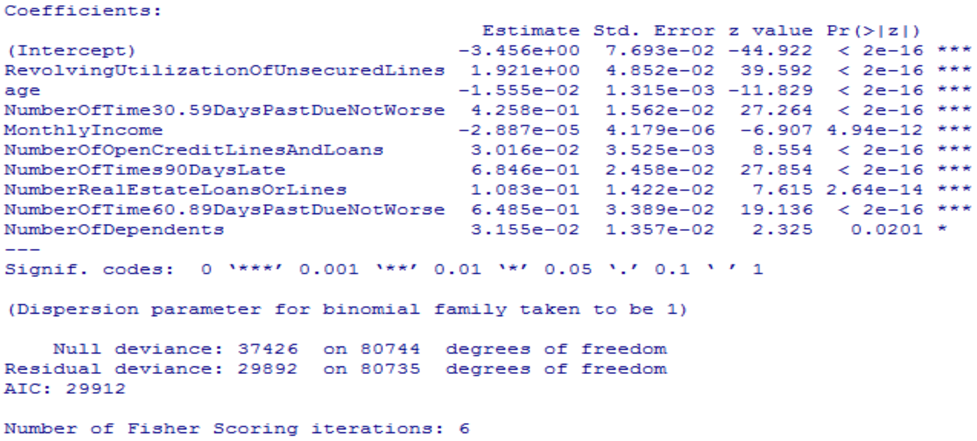
Here the class error reduces drastically to 0.028 and 0 respectively.However,oversampling has its own disadvantages and it is not computationally efficient for large data set.Several new methods like Weighted Random Forest or Balanced Random Forest can be used which do not suffer from this problem.

2.6. ***Logistic Regression Model***

We aim is to classify the data points correctly into the two categories: defaulters and non defaulters. As the response variable SeriousDlqin2yrs is categorical therefore logistic regression model is a useful model .Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables, which are usually (but not necessarily) continuous, by estimating probabilities. We first fitted the generalised linear model on with all the regressors. Note that the Wald test for significance of the coefficients for debt ratio yield p-value of p = 0.4888 indicating that the regressor appear to be redundant in the full model.

$$X^2 = 2[log - L(fullmodel)log - L(reducedmodel)] \tag{1}$$

So Using the likelihood ratio's test.Since the full and reduced models differ by 1 parameter, we can compare this test statistic to a chi-squared distribution on 1 degrees of freedom.The p value came out to be 0.49 and therefore null hypothesis was accepted debt ratio was dropped. The final model is as defined below :



```
Coefficients:
                                       Estimate Std. Error z value Pr(>|z|)    
(Intercept)                          -3.456e+00  7.693e-02 -44.922  < 2e-16 ***
RevolvingUtilizationOfUnsecuredLines  1.921e+00  4.852e-02  39.592  < 2e-16 ***
age                                  -1.555e-02  1.315e-03 -11.829  < 2e-16 ***
NumberOfTime30.59DaysPastDueNotWorse  4.258e-01  1.562e-02  27.264  < 2e-16 ***
MonthlyIncome                        -2.887e-05  4.179e-06  -6.907 4.94e-12 ***
NumberOfOpenCreditLinesAndLoans       3.016e-02  3.525e-03   8.554  < 2e-16 ***
NumberOfTimes90DaysLate               6.846e-01  2.458e-02  27.854  < 2e-16 ***
NumberRealEstateLoansOrLines          1.083e-01  1.422e-02   7.615 2.64e-14 ***
NumberOfTime60.89DaysPastDueNotWorse  6.485e-01  3.389e-02  19.136  < 2e-16 ***
NumberOfDependents                    3.155e-02  1.357e-02   2.325   0.0201 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 37426  on 80744  degrees of freedom
Residual deviance: 29892  on 80735  degrees of freedom
AIC: 29912

Number of Fisher Scoring iterations: 6
```

Figure 6: Description of the final logistic regression model

We then used it for prediction on our test dataset and we looked at the AUC value and ROC curve and also at the confusion matrix for analysis of the model.The AUC value for the final logistic regression model come out be 0.833 and the ROC plot of Logistic regression and random forest on imbalanced data set are shown below.

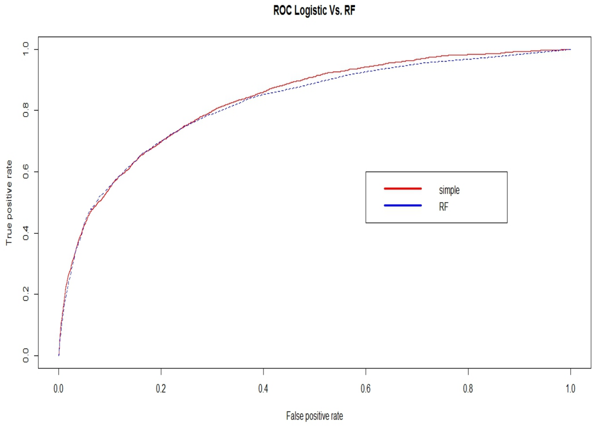


Figure 7: ROC Plot for the logistic regression model vs Random Forest algorithm

The confusion matrix for the logistic regression is given below -

	Observed=0	observed=1
Score=0	32213	210
Score=1	1903	280

The sensitivity and specificity measures are 57% and 94.4% respectively. It is evident that specificity measure is considerably better than the sensitivity measure.But, at the same time its performance seems to be better than all other classification algorithms with respect to these measures.

3. Conclusion

Machine learning from imbalanced data sets is an important problem, both practically and for research. Based on the parameter of AUC Score, we found that logistic regression was giving the best performance in the given dataset.We got the best AUC score of 0.833 while the best performing algorithm which was the winner of prize money on kaggle had an AUC socre of 0.869. SVM performs poorly with imbalanced data, as is the case in our dataset, but performance can be improved slightly by incorporating some changes in the learning algorithm. The method of trees with prior probabilities gave a better performance than trees without prior probabilities, as it helps remove bias when the training sample is not representative of the population being classified. Random Forest technique gave a better performance than simple decision tree, and the performance increased with the number of decision trees chosen by the algorithm. Since we had limited computing power, we could go up to about 300 trees for this algorithm.

4. References

- Guide to Credit Scoring in R - By DS (ds5j@excite.com) (Interdisciplinary Independent Scholar with 9+ years experience in risk management)

- Junjie Liang. Predicting borrowers chance of defaulting on credit loans (2011)

- Leo, Breiman. "Random Forests." Machine Learning 45.1 (2001): 5-32. Print.

- Chao Chen, Andy Liaw and Leo Breiman ,Using Random Forest to Learn Imbalanced Data

- Rehan Akbani1, Stephen Kwek1, Nathalie Japkowicz2 .Applying Support Vector Machines to Imbalanced Datasets.

- "Description - Give Me Some Credit - Kaggle." Data mining, forecasting and bioinformatics competitions on Kaggle. N.p., n.d. Web. 17 Dec. 2011. ¡http://www.kaggle.com/c/GiveMeSomeCredit¿

- http://www.datasciencecentral.com/profiles/blogs/random-forests-algorithm