CREDIT CARD DEFAULT

Prashanth Manjunath
163050043

Devendra Kumar Verma
163050061

Sandeep Subramaninan
163050036

Kurian Jacob
163050039

Mentor: Suthirtha Das

November 19, 2016

Contents

Abstract

Defaulting credit card customers make loss to the credit card issuer. Defaulting means non-payment of liabilities for a stipulated amount of time. Mostly this is 180 days after the issue of bill. So, for a credit card issuer, it is useful to know whether a customer is going to default on their payments. This report discusses various methods of classification of customers, or applicants as credible or non-credible, given the required data.

1 Introduction

When credit card payments are defaulted, it might cause loss to the credit card issuer. Normally, when payments are defaulted, the credit card agency writes-off the debt and sells the outstanding amount to a collection agency. This incurs heavy loss to the issuers and they would like to minimize the risk of issuing credit cards to potential defaulters. This is the motivation for us to do this exercise, and this is a market hot-topic. During this age of big data, we can procure a lot of information about customers due to which we can use machine learning to come up with machine learning insights.

2 Document representation and modelling

2.1 Data set

The data had 30000 entries. We split the dataset in 2:1 ratio. The larger part was designated as training data set and the rest to test the model. The data is organized into 24 fields. The ordering is as follows.

ID A serial number

LIMIT_BAL Maximum balance limit

SEX 1 = male, 2 = female

EDUCATION 1 for graduate school, 2 for university, 3 for high school, 4 for others

MARRIAGE 1 = married; 2 = single; 3 = others.

AGE Age (year).

PAY_0 Repayment status in September, 2005

PAY_2 Repayment status in August, 2005

PAY_3 Repayment status in July, 2005;

PAY_4 Repayment status in June, 2005;

PAY_5 Repayment status in May, 2005;

PAY_6 Repayment status in April, 2005;

BILL_AMT1 Bill amount for September, 2

BILL_AMT2 Bill amount for August, 2005

BILL_AMT3 Bill amount for July, 2005;

BILL_AMT4 Bill amount for June, 2005;

BILL_AMT5 Bill amount for May, 2005;

BILL_AMT6 Bill amount for April, 2005;

PAY_AMT1 Amount paid in September, 2

PAY_AMT1 Amount paid in August, 2005

PAY_AMT1 Amount paid in July, 2005;

PAY_AMT1 Amount paid in June, 2005;

PAY_AMT1 Amount paid in May, 2005;

PAY_AMT1 Amount paid in April, 2005;

Defaulted 1 = Defaulted; 0 = Not defaulted

We want to predict the *defaulted* field. To make any prediction about the data, we need first preprocess the data into useful numerical information. Since our dataset was already processedm, we started off with feature selection.

2.2 Feature selection

2.2.1 Analysis using PCA

PCA is used for dimensionality reduction. It will return the top k dimension which has the significant amount of information. That is, PCA determines the direction in the hyperspace along which the spread of data is maximum. We tried PCA before we tested against our model. The accuracy of prediction was found to increase with each dimension added, so we concluded that all fields are important.

3 Model implementation

We implemented four supervised classification algorithms. The models that we implemented were

- KNN classifier
- SVM classifier
- Decision tree classifier
- Neural Network classifier

3.1 KNN classifier

KNN classifier attempts to find the nearest k neighbours to the given data point and make predictions based on the neighbouring points. It is a simple classificaition algorithm. The parameters were $k = 4$, the distance metric was *minkowski*. We used the mode, which will attempt to find the best relationship in the data.

3.2 Neural Network

Neural networks try to imitate the biological brain. These are constituted of components called neurons, which each one of them taking multiple inputs and produces an output based on the activation function and sum of weighted sum of inputs. However single perceptrons can only classify linearly seperable data. To work on data that are not lineraly seperable, we need these neurons stacked into multiple layers. The activations from the input layer is fed into the succeeding layer and so on.

We implemented a neural network with 3 hidden layers. The input layer has 83 perceptrons, the first has 160, second has 80, third has 23, and the final output layer has two nodes. We used the sigmoid function as the activation function.

We used one hot encoding for discrete features like education and gender.

3.3 SVC

SVC is a deterministic binary classifier. In this case, we need to find a hyperplane that seperates the data into two classes. The kernel was found to be RBF.

3.4 Decision trees

We used random forests to implement decision trees. There were 142 estimators. Entropy for impurity function. Classes were weighed. The number 142 was used because 142 is the square root of the length of dataset, which in this case is 20000.

4 Results and Observations

4.1 KNN Classifier

n neighbors	Accuracy
1	69
2	77
3	74
4	77
5	76
6	78
7	77

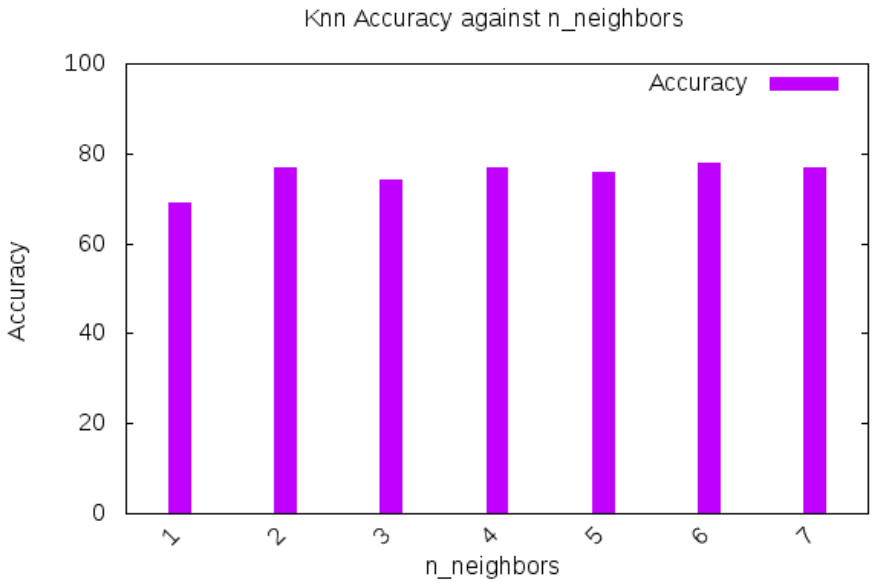

Figure 1: Graph between n neighbors and Accuracy

4.2 SVM Classifier

Kernel	Accuracy
RBF Kernel	79
Sigmoid Kernel	65
Linear Kernel	50

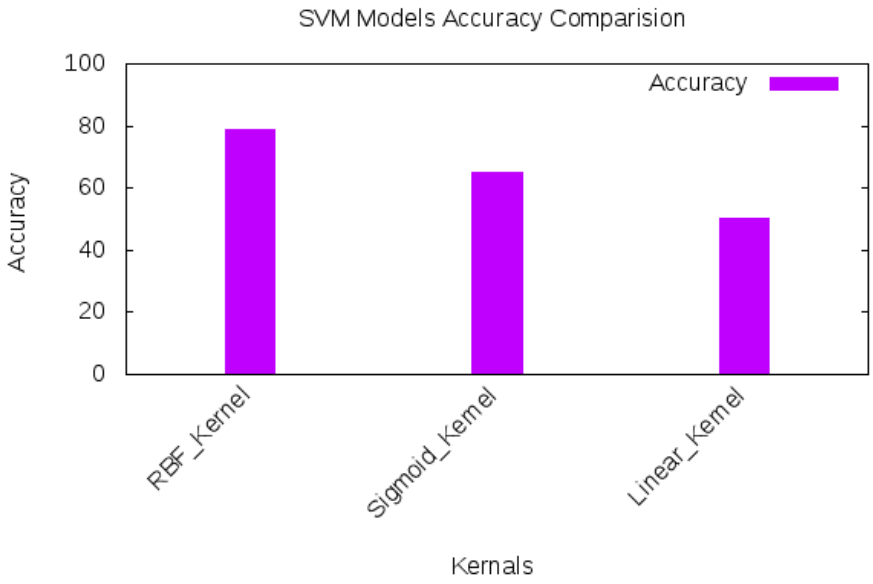

Figure 2: Graph between Kernel and Accuracy

4.3 Neural Network

Learning Rate	Accuracy
0.001	43
0.01	37
0.1	25
1	60

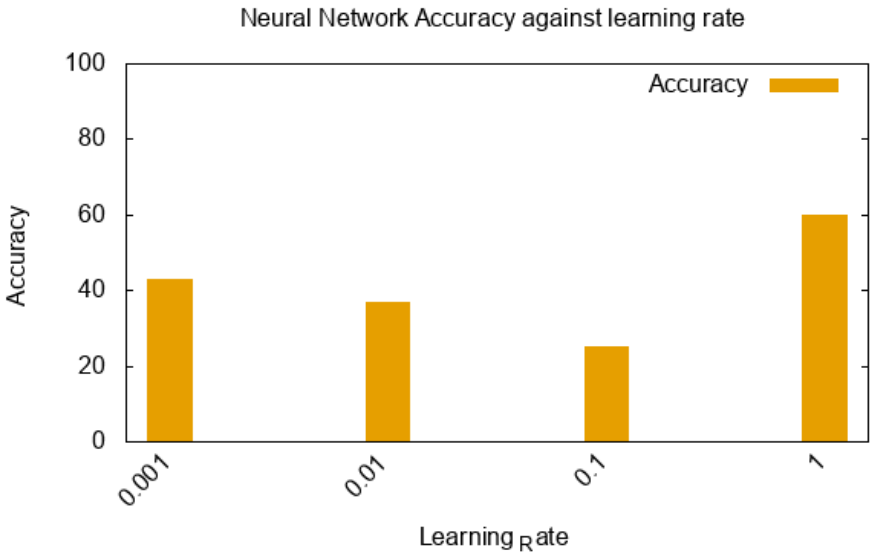

Figure 3: Graph between learning rate and Accuracy

4.4 Random forests

Random forests are ensemble of weakly learnt trees with equal weights. The impurity functions used for random forest were entropy and GINI index.

4.5 Decision tree with adaboost

Adaboost is similar to random forest, except the weakly learnt trees can have varying weights.

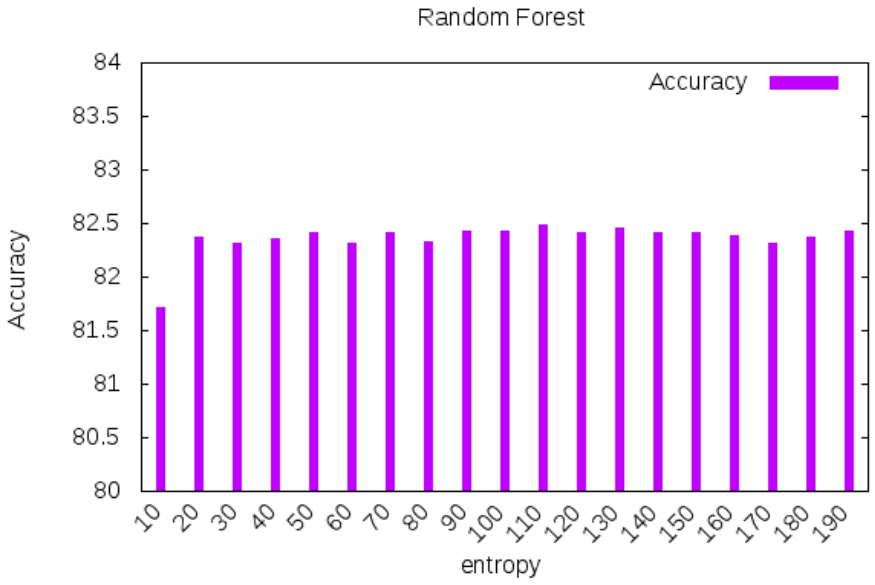

Figure 4: Accuracy with impurity function set as entropy

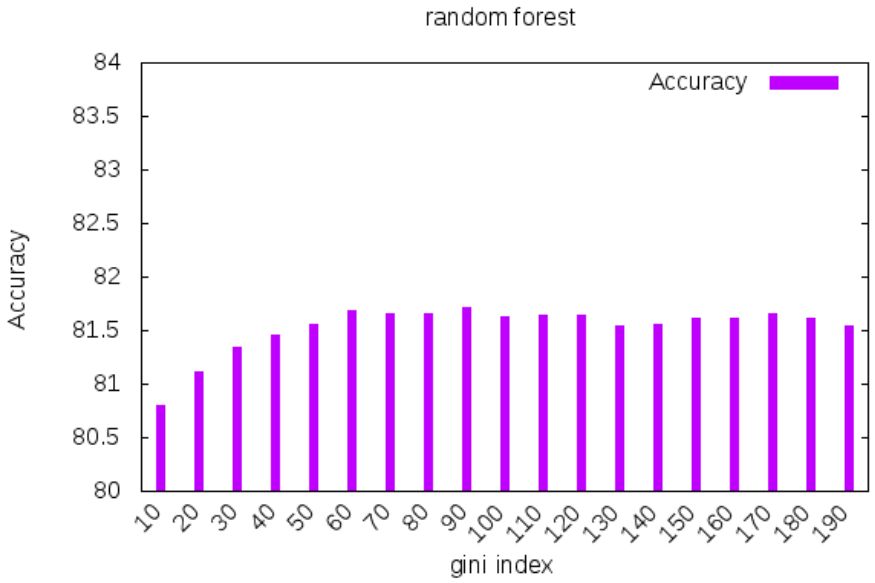

Figure 5: Accuracy with impurity function set as GINI

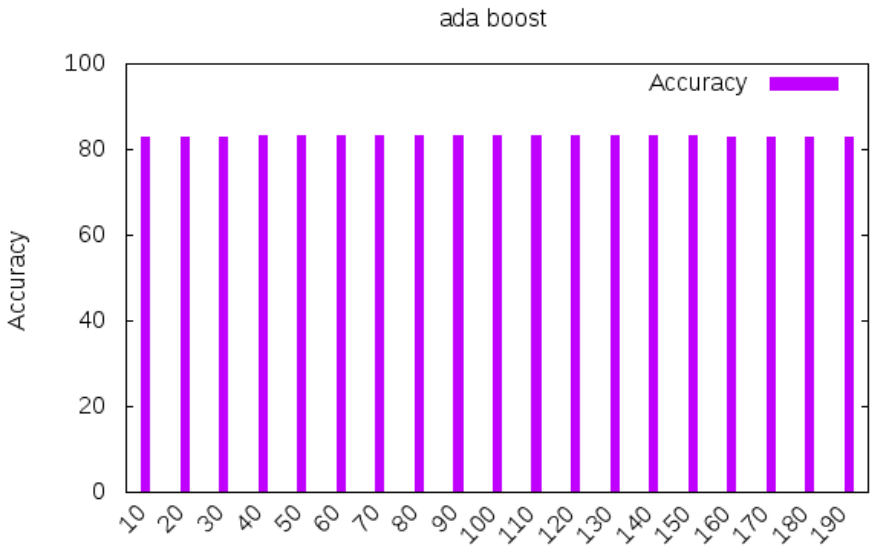

Figure 6: Accuracy with respect to number of decision trees

Number of trees	**Acuracy**
10	82.7
20	82.9
30	82.94
40	83.14
50	83.23
60	83.13
70	83.06
80	83.12
90	83.04
100	83.11
110	83.06
120	83.23
130	83.11
140	83.12
150	83.04
160	83.02
170	83.03
180	83.03
190	82.98

5 Detailed Analysis

5.1 Models Comparision

Model Name	Accuracy
SVM	79
Neural Network	79
KNN Classifier	78
Decision Tree	82

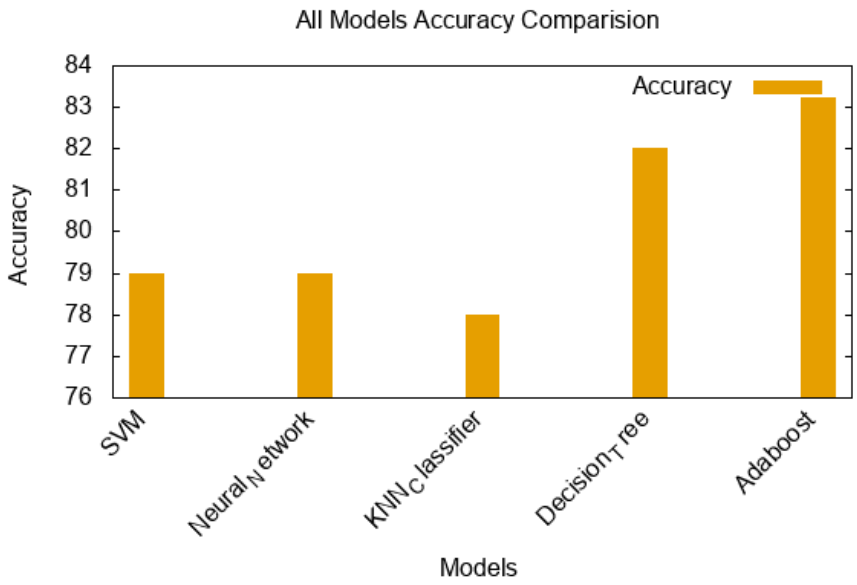

Figure 7: Graph showing Models Accuracy comparition

6 Conclusion and Further work

After comparing many models, we have seen that decison trees model has given the best accuracy. We could also have used LSTMs, but the required data to impelement LSTMs were not available. We have only data from April to September. If the history of payments were longer, we could use LSTM for a better predictive model.

References

[1] Yeh, I. C., & Lien, C. H. (2009).
The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.
http://www.sciencedirect.com/science/article/pii/S095741740700671

[2] Bart Baesens , Rudy Setiono , Christophe Mues , Jan Vanthienen.
Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation, Management Science, v.49 n.3, p.312-329, March 2003.
https://www.jstor.org/stable/pdf/4101754.pdf

[3] Machine learning with scikit-learn.
http://scikit-learn.org/stable/tutorial/basic/tutorial.html

[4] Neural network in 11 lines.
http://iamtrask.github.io/2015/07/12/basic-python-network/