Credit Score Analysis

Overview

The problem is we have to identify weather given person is able to get credit card or not using predictive modeling, problem type is classification modeling under the supervised learning domain.

We start with two models first one is SVM (Support vector machines) with linear kernel reason to selecting this tool is if data have more then 10-12 feature vectors and if they are separable in some hyperplane generally SVM will perform good in this situation, second is resilient backpropagation neural network if data have lots of noise and sparsity due to their many control factors neural nets are more robust.

Data visualization

Before discussing algorithms and their performance let's take a look on dataset, dataset contains a good mixture of continuous and discrete values following are the steps of data pre-processing.

Note: - the missing value in a given dataset were denoted as a "?"

- First we coerced discrete attributes as a factor and continuous as a numeric in R data frame.
- Then we identified rows having column input "?" and replace it with most frequent value of that column.
- Then we normalize the continuous attributes of dataset to scale them down by using following function

$$x_i = \frac{x_i - min(x)}{max(x) - min(x)}$$

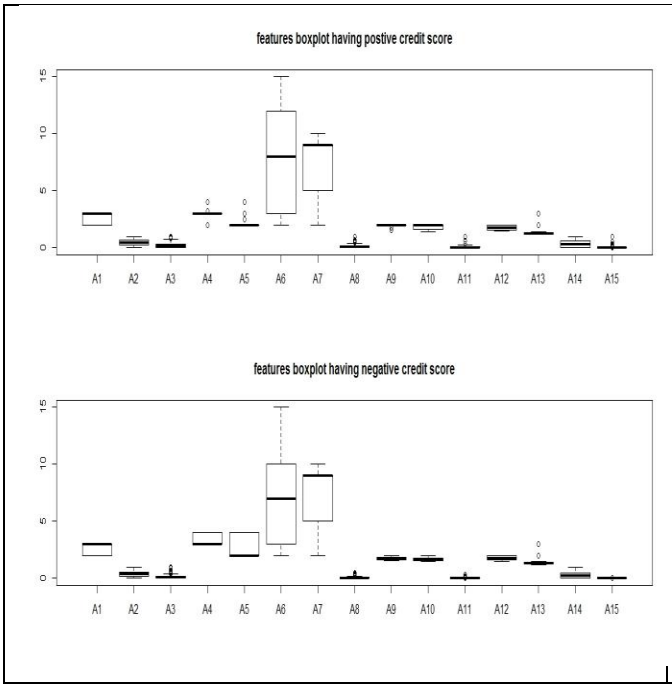


Figure 1: Boxplot for all 15 attributes

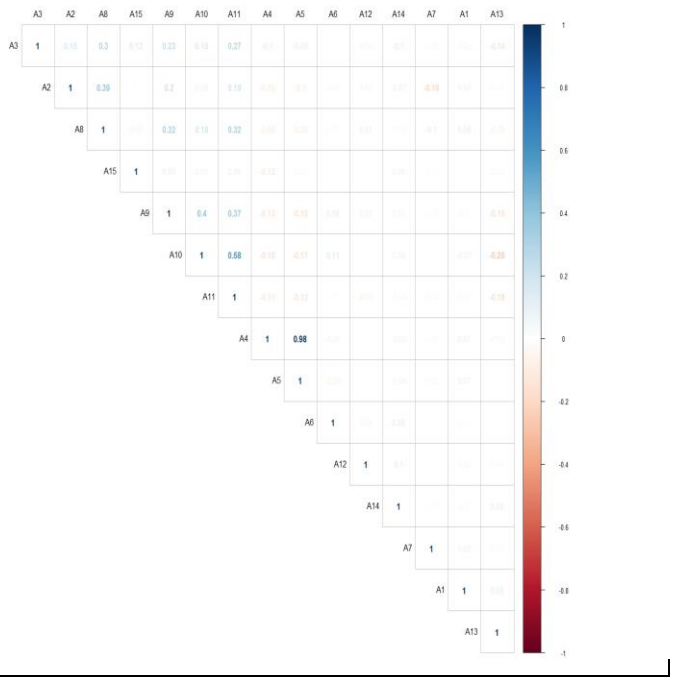


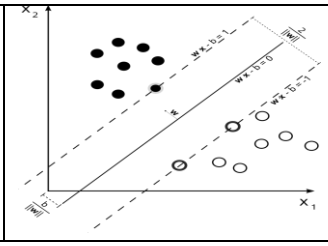
Figure 2: Correlation plot for all 15 attributes

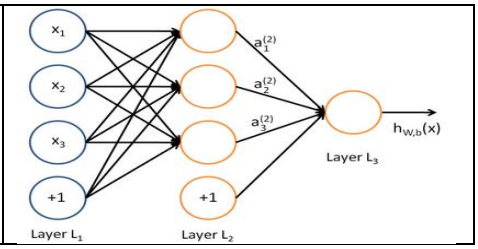
From both graphs we can draw some intuition about the dataset, the figure 1 is a boxplot for all the 15 attributes for both cases positive and negative credit score. In this graph attributes A4 and A5 shown significant difference between people having positive and negative credit score and small changes in A9 and A10 attributes.

While figure 2 is useful to represents the correlation between attributes, which might be used in dimensionality reduction and attributes identifying similar kind of feature vector it is more useful in continuous variables case.

Algorithm description

We don't go in detail of each algorithm just a very short overview on each then we move on to describing code files, control flow and performance measures.

SVM (Support Vector Machines) works on assuming that data points are separable in some hyperplane and try to find that plane having maximum possible boundary separation so we have to $minimize\ \|w\|\ subject\ to\ \ y_i(\vec{w}.\vec{x_i} - b) > 1\ for\ i = 1,2,\dots n$, given formula only hold for linear SVM.	

Neural network uses a property of multiple computing units, output from each neuron will be treated as input in next layer and in same fashion derivative of error function can be back propagated to up to input neurons using simple chain rule, like in a given figure layer 1 is input layer and layer2 is hidden layer and layer3 is a output layer, use of neural network preferred in very large dataset and heavy feature set like pictures and NLP.	

Code files and control flow

- Code are written in R language and divided into three sections credit_score.r will implement resilient backpropagation neural network with 15 input neurons, 7 hidden neurons and one output neuron.
- Credit_score_svm.r will implement linear SVM type class classification for predicting the values.
- Data_cleaning.r both code files use this script to perform data preprocessing steps as described in previous page.

Performance

We trained both of our algorithms on 60% of original dataset and rest all used for testing.

Linear SVM	Resilient Backpropagation
reference prediction 0 1 0 124 11 1 25 116 Accuracy : 0.8695652 95% CI : (0.8240214, 0.9069461) Sensitivity : 0.8322148 Specificity : 0.9133858 Pos Pred Value : 0.9185185 Neg Pred Value : 0.8226950 Prevalence : 0.5398551 Detection Rate : 0.4492754 Detection Prevalence : 0.4891304 Balanced Accuracy : 0.8728003	reference prediction 0 1 0 123 14 1 41 98 Accuracy : 0.8007246 95% CI : (0.7486565, 0.8462064) Sensitivity : 0.7500000 Specificity : 0.8750000 Pos Pred Value : 0.8978102 Neg Pred Value : 0.7050360 Prevalence : 0.5942029 Detection Rate : 0.4456522 Detection Prevalence : 0.4963768 Balanced Accuracy : 0.8125000

As we can see SVM is performing better then resilient backpropagation reason is neural network required much more data to train properly and there is some outliers also (refer to the boxplot figure).

Note: - for better visualization refer to graphs folder.