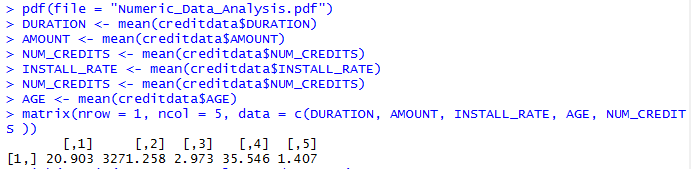
1. Data Exploration and Visualization :

The analysis of the data set has been performed in R. All graphs have been generated in R. The code and the output snippets have been shown for reference and reader’s clear understanding.

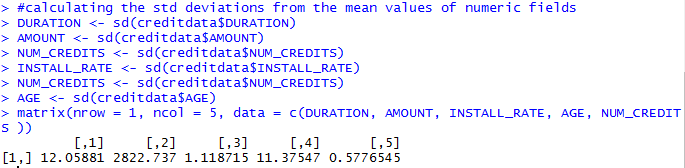
The GermanCredit dataset has 5 numerical fields namely, Duration, Amount, install\_rate, Age and Num\_creadits.

1. Obsevation using the mean values :

The output matrix of the mean values of each of these variable was calculate to establish a relationship between the variables. It can be observed that the average age of a pesron in the data set is 36 years and the average credit amount is approx. $3270. The recorded average duration of credit has been 20 months and average installment rate of 3% of disposable income.

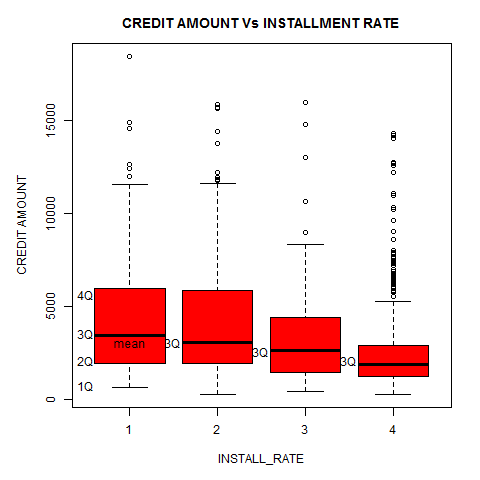
1. Observations using Standard Deviation values:

The standard deviation shows a large deviation for amount = $2822 and the average credit amount = $3270. This can be explained looking at the large deviation of number of average age => 36 - 11 = 25, and duration => 20 – 12 = 8 as well, which means that the data contains information of people over a wide range of age groups and duration of credit. Thus we can say, the variation in credit amount is a resulf of it being directly dependent on the age of a person and also on the duration for which he/she has been holding the credit account.

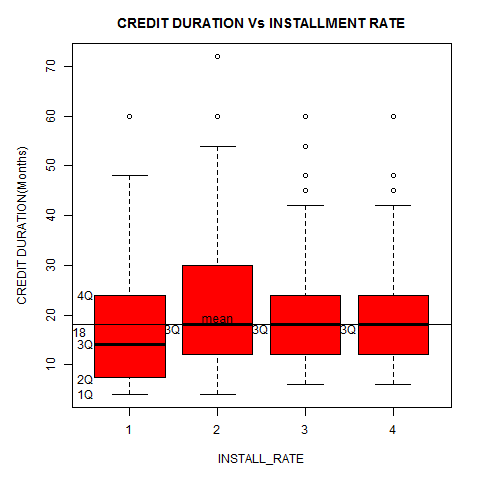


1. Observations using boxplots:

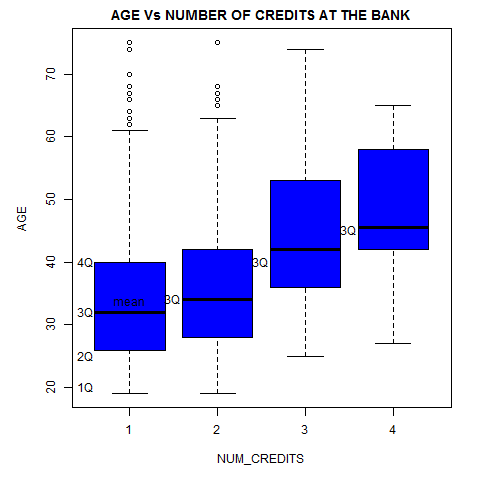
Boxplots have been utilized to compare two variables and analyse an existing relationship between them.

**Age Vs Number of Credits with the Bank:**

The X-asix shows the Installment Rates as integer values and the Y-axis shows the Amount in numerical. Its evident from the graph, for higher values of rate reflects lower values of credit amount. Therefore, it can be inferred, a person having minimal rate tends to spend more as can be seen that the average credit amount is highest in boxplot having rate = 1, whereas, a person with higher rate will tend to spend less. However, there are a number of outliers which account for abnormal cases.

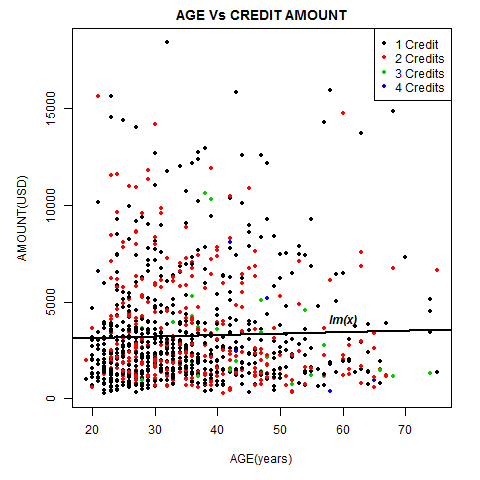
**Credit Duration Vs Installment Rates:**

The X-aixs shows Credit Amount and the Y-axis shows the Installment Rates as integer values. There is no significant change in the rates for an increase in credit duration. Although, the credit rates are the lowest when average duration is less than 20 and becomes an inconsiderable factor for deciding the rates thereafter.

**Number of Credits Vs Age:**

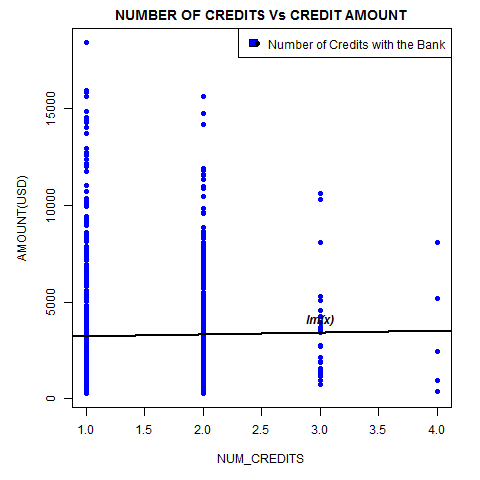
The X-axis shows Number of Credits with the bank and the Y-axis shows the Age of the person. The number of credits a person might have with a bank helps us in detemining the age of a person. From the plot we can observe that people lying between the age group of 40 to 60 have more credits than those who are in the age group of 20 to 40.

1. Scatter plots using numeric predictor variable to learn the relationships between them



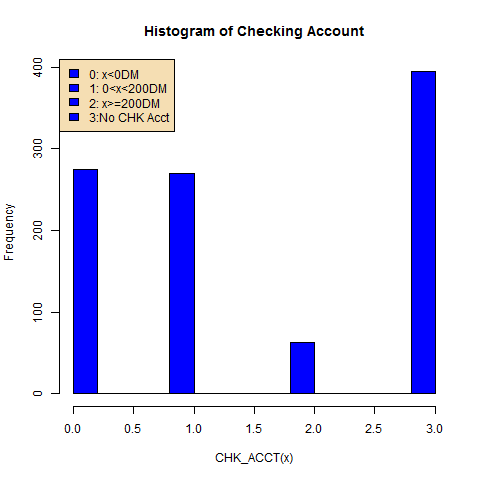
**Relationship between Age and credit Amount:**

The scatterplot shows the variation of the credit amount over the age factor and it can be observed that majority of the people have a credit amount in the range of $0 - $5000, belong to the age group of 20 to 40 and have 1 credit with that bank. The rest are scattered evenly across different points.

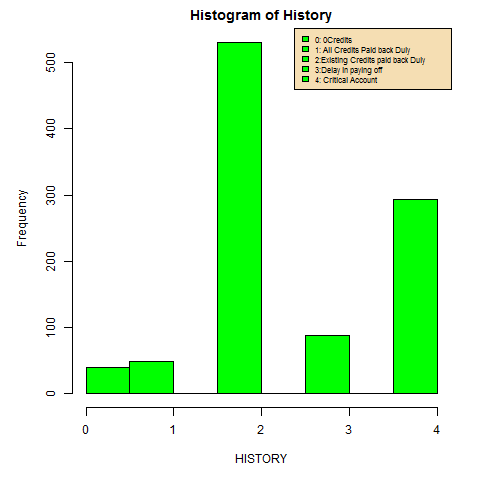
**Relationship between the number of credits and the credit amount:**

This plot tries shows that as the number of credits with the bank keeps on increasing the credit amount range keeps narrowing down. Very few as good as four out of 1000 persons have more than 3 credits with a bank. It can as well be inferred that the spending capacity of a person with multiple credits reduce.

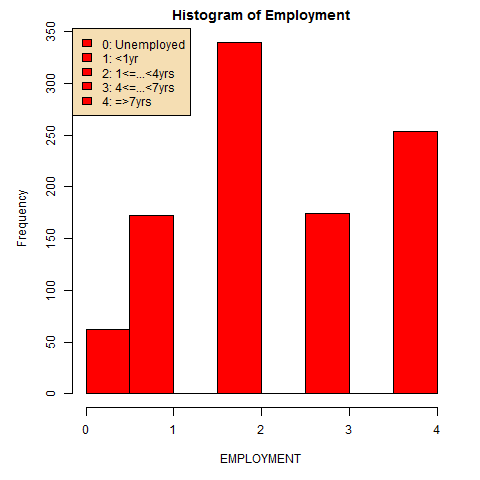
1. Categorical Variable are bucketed and are utilized to analyze with the help of histograms.

**Histogram of Checking Account:**

This graph show that more than a quarter population doesn’t have a checking account and 1/10th of the people having a checking account have a balance of =>200DM. The rest 9/10th of the people either have a balance of <0DM or between 0 to 200DM.

**Histogram of History:**

This graph shows the plot of categorical variable History with the help of a histogram. It is observable that a considerable accounts which have paid their credits duly and have a clear credit history as of the existing accounts with the bank and there are significant accounts as well which have been termed as critical.

**Histogram of Employment:**

This graph shows the plot of Employment where it can be seen that majority of those who are credit card holders have been employed and have worked from 1 to 4 years at the least. There are very few of them approx. 3/10ths who are either unemployed or have less than one years of experience.

1. **Neural Networks**:

The dataset was partitioned as follows –

* 50% Training data and 50% Validation data
* 60% Training data and 40% Validation data

1. **Neural Network Analysis of using 50% Training Data and 50% Validation Data:**

This dataset contains 1000 records and 32 variable. We are trying to develop a neural network model for a classification problem.

Credit Rating Problem: Classify the Credit Rating good or bad.

XL-Miner was used to partition the data. The partitioned data was then applied as neural network as input.

**Input Variables**: CHK\_ACCT, DURATION, HISTORY, NEW\_CAR, USED\_CAR, FURNITURE, RADIO/TV, EDUCATION, RETRAINING, AMOUNT, SAV\_ACCT, EMPLOYMENT, INSTALL\_RATE, MALE\_DIV, MALE\_SINGLE, MALE\_MAR\_WID, CO-APPLICANT, GUARANTOR, PRESENT\_RESIDENT, REAL\_ESTATE, PROP\_UNKN\_NONE, AGE, OTHER\_INSTALL, RENT, OWN\_RES, NUM\_CREDITS, JOB, NUM\_DEPENDENTS, TELEPHONE, FOREIGN

**Response Variable:** RESPONSE

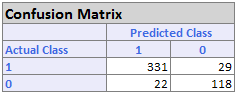
**Unused Variables:** OBS#

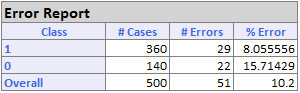
The initial cutoff probability for success was set as: 0.5 and automatic Neural Network was created using Automatic Network setting in the XL-Miner keeping all other values such as the No. of epochs, gradient Descent as default values.

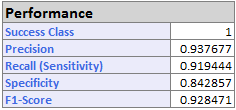
The data set contains 500 records in the training data set and 500 records in validation data set. The Auto network generates 100 possible matrices. The network with best possible network accuracy was chosen and the neural network algorithm was performed on it.

For simplicity the values such as epochs, gradient descent and error gradient were set to defaults. The number of hidden layers was chosen as 2 and number of nodes were chosen to be 20 and 10 in successive layers.

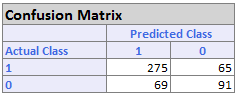
Training Data Scoring – Summary Report

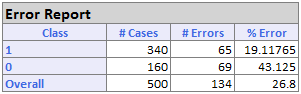
The confusion matrix shows the matrix of Actual and Predicted Outcomes. It can be explained as the confusion in predicting the actual output as 1 was 29 times wrong and the actual output as 0 the algorithm was 22 times wrong when it misinterprets them to be right. Therefore, the error in predicting the outcome variable can be calculated as (22 + 29)/ (331+118+29+22) = 10.2%. 

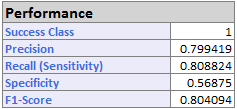
Thus the error report shows the total number of misses and the total number of hits and produces a percentage error of 10.2%.

This is the performance matrix which shows the performance of the algorithm over the training data set. It can be seen that the algorithm shows a good precision over the training data set. Although, Very high precisions can also lead to overfitting of the algorithm over the training data.

Validation Data Scoring – Summary Report

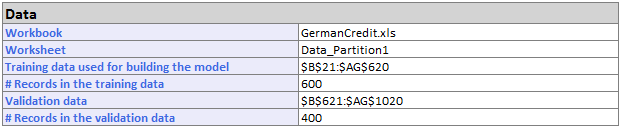
The confusion matrix shows the matrix of Actual and Predicted Outcomes. It can be explained as the confusion in predicting the actual output as 1 was 65 times wrong and the actual output as 0 the algorithm was 69 times wrong when it misinterprets them to be right. Therefore, the error in predicting the outcome variable can be calculated as (65 + 69)/ (275+91+69+65) = 26.8%.

Thus the error report shows the total number of misses and the total number of hits and produces a percentage error of 26.8%.

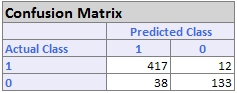
This is the performance matrix which shows the performance of the algorithm over the validation data set which significantly differs from the training data set by approx. 15%.

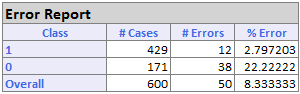
1. **Neural Network Analysis using 60% training data and 40% validation data:**

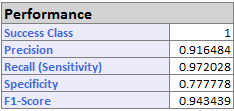
The image below shows the input data information we used for neural network analysis on the GermanCredit.xls dataset. The Partition used is 60 and 40 percent respectively. All other parameters are same as above.



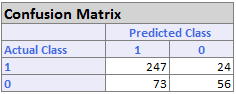
Training data Scoring – Scoring Report

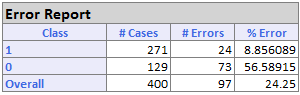
 The confusion matrix shows the matrix of Actual and Predicted Outcomes. It can be explained as the confusion in predicting the actual output as 1 was 12 times wrong and the actual output as 0 the algorithm was 38 times wrong when it misinterprets them to be right. Therefore, the error in predicting the outcome variable can be calculated as (12 + 38)/ (417+38+29+133+12) = 8.33%.

Thus the error report shows the total number of misses = 50 and the total number of hits = 550 produce a total percentage error of 8.33%.

This is the performance matrix which shows the performance of the algorithm over the training data set. It can be seen that the algorithm shows a good precision over the training data set. 

Validation Data Scoring – Summary Report

The confusion matrix shows the matrix of Actual and Predicted Outcomes. It can be explained as the confusion in predicting the actual output as 1 was 24 times wrong and the actual output as 0 the algorithm was 73 times wrong when it misinterprets them to be right which is 3 time higher than in training data. Therefore, the error in predicting the outcome variable can be calculated as (24 + 73)/ (247+56+73+24) = 24.25%.

Thus the error report shows the total number of misses = 97 and the total number of hits = 303 produce a total percentage error of 24.25%

Comparing cases A. and B. we made some observations as given below -

1. Neural nets produce more accurate results when given more number of inputs. This can be demonstrated by comparing the %Error produced by the training datasets in each partition. The partition where we employ 60 training data to train the neural net, the error% reduces by 2% => 10.2 - 8.2%.
2. Neural nets are prone to overfitting and are sometimes not the best way to simulate a generic model to deal with the problem. If too many artificial neurons are used the training set will be memorized, not generalized, and the network will be useless on new data sets. This can be demonstrated by looking at the significant difference in the %errors of the training and the validation data in both cases keeping all other parameters as same.

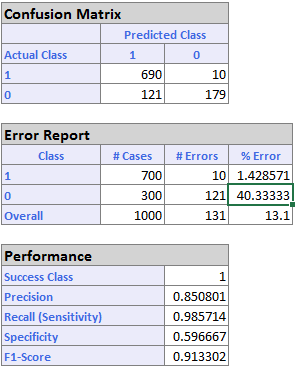
1. Each of the parameters were changed one by one to see the effect of each parameter on the resulting error percentage. Here, the full data set is used for better performance of the neural net.
2. **Effect of changing the number of layers:** The number of layers was increased from 2 to 3 having 20, 15 and 12 nodes respectively.

Calculating the number of nodes in each successive hidden layer:

UB1 = FLOOR(30 + 1) \* 2/3 = 20.6 ~ 20

UB2 = FLOOR(20 + 3) \* 2/3 = 15.3 ~ 15

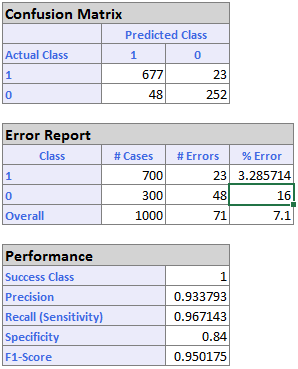
UB3 = FLOOR(15 + 3) \* 2/3 = 12 = 12

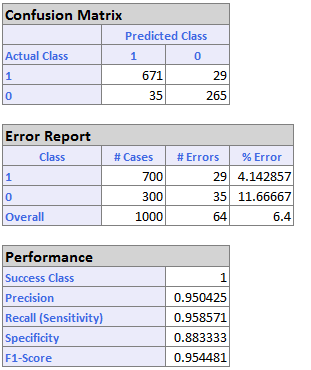
The Error Report table shows that error% in predicting the correct response has increased and the net error% is 13.1% which is => 13.1 – 10.2 = 3.1% higher than with training network using 2 layers.

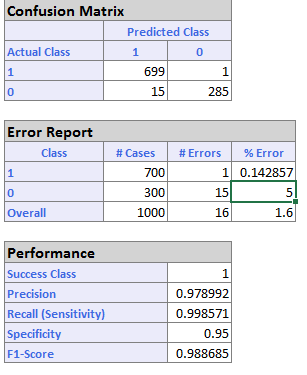
Thus it can be observed that as the complexity of the network increases, which is done by adding additional hidden layers to the neural network, the performance reduces. Also, as the complexity in the relationship between the input data and the

desired output increases, the number of the processing elements in the hidden layer should also increase.

1. **Effect of Increasing the gradient descent:**

The Gradient Descent was observed using various different values and it was observed that if the descent is increased by 0.1 at every step the resulting error starts reducing. The minimized error% was received at Gradient Descent at 0.4. Thus the chart shows the lowest error% as 7.1% as we go on increasing the gradient descent.

1. **Effect of changing the number of epochs:**



The number of Epochs are now increased from 30 to 35 keeping the rest of the properties unchanged. The response output shows that the error is reduced from 7.1% to 6.4% (as can be seen in the Fig. on the left hand side) which is significant given the fewer number of epochs we added to the analysis. As we go on increasing the number of epochs the network shows even better precision. The error% obtained with number of epochs = 80 is 1.6% which is remarkable (refer to the Fig. on the right).

1. **What is Neural Networks Methodology?**

Artificial neural networks are relatively crude electronic networks of “neurons” based on the neural structure of the brain. They process records one at a time, and “learn” by comparing their classification of the record with the known actual classification of the record.

Therefore, we can say that a neuron in an artificial network is

1. A set of input values (X*i*) and associated weights
2. A function “g” that sums the inputs and maps the result to an output

Neurons are organized into layers: input, hidden and output. The input layers is not composed of the full set of neuron, but rather consists of those record values which are inputs to the next layer of neurons. The next is the hidden layer. Several hidden layers can exist in one neural network. Final layer is the output layer, where there is one node for each class. A single sweep forward to the network results in assignment of a value to each output node and the record is assigned to the class node with the highest value.

In the training phase, the correct class for each record is known, and the output nodes can therefore be assigned “correct” values = “1” for the node corresponding to the correct class and “0” for the rest. It is thus possible to compare the network's calculated values for the output nodes to these "correct" values, and calculate an error term for each node (the "Delta" rule). These error terms are then used to adjust the weights in the hidden layers so that, hopefully, during the next iteration the output values will be closer to the "correct" values.

The function “g” explained as above is called the **Transfer Function.**

The neural network can be considered as more general form for complex computations.

**Is Neural Networks a recommended approach?**

**Pros**

- The most prominent advantage of neural networks is their good predictive performance.

- They are known to have high tolerance to noisy data and the ability to capture highly complicated relationships between the predictors and a response.

**Cons**

- Their weakest point is in providing insight into the structure of the relationship, hence their black-box reputation.

- Several consideration should be kept in mind when using neural networks. Even though they are capable of generalizing from a set of examples, extrapolation is still a serious danger. If the network sees only cases in a certain range, its predictions outside this range can be completely invalid.

- Neural networks don’t have a built in variable selection mechanism which means there is a need for careful consideration of predictors.

- Extreme flexibility of the neural networks relies heavily on having sufficient data for training purposes. A neural network works poorly if the training size is insufficient. Though, this can be achieved from oversampling.

- A technical problem is the risk of obtaining weights that lead to a local optimum rather than a global optimum.

- A practical consideration that can determine the usefulness of a neural network is the timeliness of the computation. They are relatively heavy on computation time.

1. **Analysis:**
2. **Your client currently doesn’t have a model and gives credit to whoever approaches him. If he approves credit to everyone, what is the profit for your current data set(1000 records)**

Let us calculate the probability of him getting a success. Here there is no algorithm has been used to attribute weights to a given input and Octoberfest is randomly assigning credit cards. Thus, there is an equal chance that a person who is given the credit could be a “credit worthy person” or “not”.

Therefore, P (E) = 0.5, E is the event of a selecting a person from a pool to be credit worthy.

Now according to the matrix given in the question, Octoberfest understands that there is a cost of misclassification which influences the profit over time. If a person is credit worthy, he adds to the profit of +100 DM and if a person is not credit worthy he adds to the negative profit of -500 DM.

The number of records = 1000

The number of people in the data set who can be credit worthy => 0.5 \* 1000 = 500

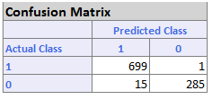
The number of people in the data set who might not credit worthy => 0.5 \* 1000 = 500

The total profit he can earn = [- (500\*500) + (500 \* 100)] DM = - 20 DB (Billion $)

**Thus, the Octoberfest will have a negative profit of -20 DB.**

1. **Using Neural Network to Calculate the profit**

We obtained a neural network with best results having the %error of 1.6 and used the same for our analysis. Refer to the confusion matrix as given below –

We consider column one of matrix:

Actual Class (1) = Predicted Class (1) = 699

Actual Class (0) = Predicted Class (1) = 15

As per the profit calculation Matrix we get the following results,

The number of people in the data set who can be credit worthy => 699

The number of people in the data set who may not credit worthy be => 15

The net profit the Octoberfest might incur => [(699\*100) – (15\*500)] DM = +62.4 DB

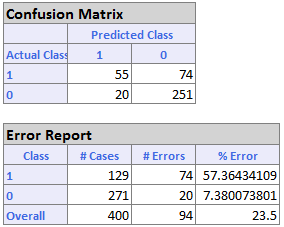
The error percentage of this model is [(1+15) / (699+16+285)] \* 100 = 1.6%

Thus the profit may vary + or – between this error.

**Thus, the Octoberfest will have a profit of (62.40 ± 0.99) DB.**

1. **Additional steps to improve the model**

We can consider some additional improvement to enhance the model which we created until now.

1. One of the problems arising during neural network analysis is **Overfitting** which means the error on the data is very small, but when the new data is presented the error is very large. The network has memorized the training examples but it hasn’t learned to generalize to new situations. There are various ways increase the generalization of the neural network one of which is using network just large enough to provide an adequate fit.
2. It is a good idea to train several networks to ensure that a network with good generalization is found. Boosting and Bagging could be the techniques applied for the same.
3. Another simple way to improve generalization, especially when caused by noisy data or a small dataset, is to train multiple neural networks and average their outputs.
4. The default method for improving generalization is called early stopping. We can limit the number epochs to iterate the neural net and reach a local optimum.
5. Increase the number of records in the training data set. As the neural networks learn better with more data available for training purpose. For example, the data set should have more records than the input variables to have enough information to train the neural network. We could recommend Octoberfest to provide us with more training data (>1000 records).
6. The logistic model for the German dataset was created using 60% and 40% division of the training and the validation data as a basis for selecting the best credit risk first followed by the poor risk applicants. For this a new column RESPONSE\_NEW was created which is a binary output variable where “1” – best risk person and “0”- represents low risk person. Our results are based on the validation data.
7. The logistic regression model can be seen in the file named GermanCredit\_LogisticRegreesion.xls
8. The logic model was created and following are the results we got on the validation data –

The confusion Matrix shows the value of FP- False Positives and the FN- False Negatives in the system.

From here we can calculate the net profit as follows –

Number of actual risk incorrectly predicted as no risk = 74

Actual poor risk predicted correctly in numbers = 251

Actual risk incorrectly predicted as no risk = 74

The profit in this case = [(251 \* 100) – (74 \* 500)] = -11.9 DB

**Therefore, we conclude that there is a net negative net profit of -11.9 DB.**

P.S. The profit calculation for each case was demonstrated in the excel sheet called “LR\_Validation\_Cases” of GermanCredit\_LogisticRegression.xls

1. From the analysis we calculated cumulative net profit as –

∑ (P(Success) \* Profit) = **6.151 DB**

1. The choice to specify the initial cutoff probability for success lies with the person who is building the model. If the Probability of success (probability of the output variable = 1) is less than this value, then a 0 will be entered for the class value, otherwise a 1 will be entered for the class value. As the cutoff is increased, the error percentage increases as well thus we will stick to a default of 0.5 because we have 2 output base values and we divide these two having equal contribution of 0.5.

**Executive Summary**

The document goes through the analysis on GermanCredit dataset containing 1000 records of potential credit applicants. The initial step of the study shows relationships between various parameters contributing towards predictive analysis of credit worthiness of an applicant. The later part of the study deals with a problem of classification modeling, we try to develop generic classification model for Octoberfest to classify an applicant as credit worthy thereby maximizing their profit with our predictive analysis using the neural networks methodology. The same was done using Logistic Regression modelling technique to come up with a model which is constructed on the basis for selecting the best credit risk first followed by the poor risk applicants thereby helping Octoberfest to select a best fit model for its bank.