**Comp 3425 Data Mining**

**Assignment 2**

The Australian National University

Submitted by: -

Prateek Arora

U6742441

**1)** The evaluation of the dataset for mining is done windows as an operating system and Rattle was used as a software to evaluate the data. The CPU used is intel core i5, 8th generation and memory for the same is 1TB.

**2)**

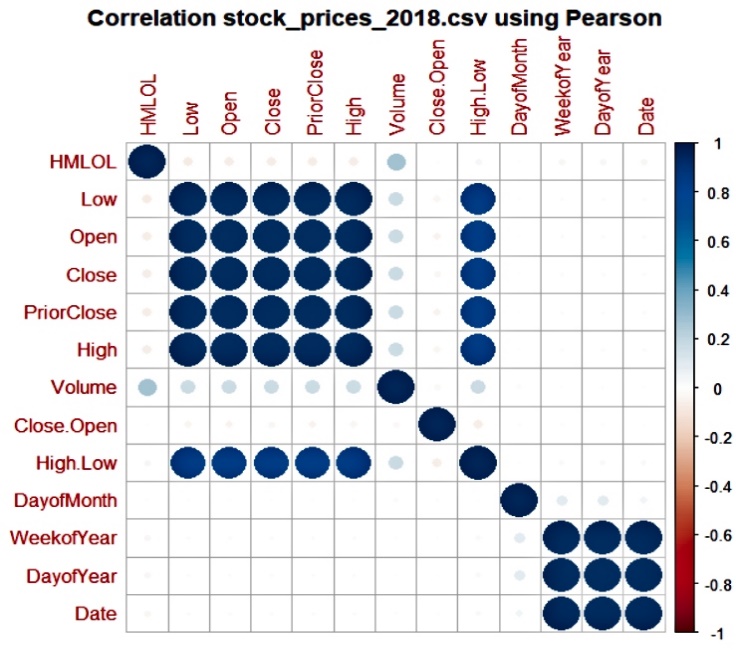


Fig 2

a) The Pearson correlation coefficient, r, takes a range of values from +1 to -1. Value of 0 indicates that there is no correlation between the two variables. A value greater than 0 indicates a positive correlation, i.e., as the value of one variable increases, the value of another variable will also increase. A value less than 0 indicates a negative correlation; that is, as the value of one variable increases, the value of the other variable decreases [1]. So, here DayofYear and WeekofYear tend to have value almost equal to 1 as it can be clearly deduced from the above figure. So, according to the pearson correlation coefficient, if r is greater than 0, then it is positively correlated. As r is close to 1, we can say that DayofYear and WeekofYear are highly correlated.

b) High, Low, Open, Close and Volume are highly correlated as they all have Pearson correlation coefficient, r nearer to 1 with comparison to each other except themselves as that will surely give r value equal to 1. E.g.: (High, Low), (High, Open), (Open, Close) are some of the pairs in which correlation is almost 1 and other different pairs can also be formed using High, Low, Open, Close and Volume whose result r would be almost 1 is shown in figure 2.

**3.**

a)

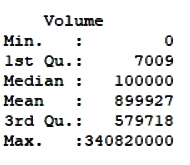
Fig 3 clearly represents the 5-number summary for the particular data. Here, min gives the minimum value whereas max gives the maximum. The difference between 3rd quantile and 1st quantile is known as IQR. Therefore, IQR = 572,709. So, through IQR, we can calculate are called Outliers. So, for the outliers, the condition is that the value should be below Q1 – 1.5\*IQR and above Q3 + 1.5\*IQR.

Fig 3 Q1 – 1.5\*IQR = 7009 – 859063.5 = -852054.5   
 Q3 + 1.5\*IQR = 1431772.5

So, the outliers for this data are from the range [1431772.5, 340820000]. Volume here qualitatively tells that there are some outliers that in the data. The number of outliers in this data is 4552 as given below in fig 3 b) i).

b)

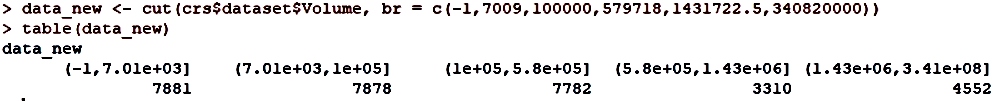


Fig 3 b) i)

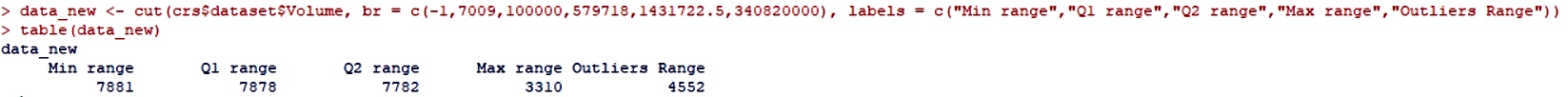


Fig 3 b) ii)

While dividing the categories, in the first range, the value is taken as -1 instead of 0. This is because in range, the first element is not included. So, if we write 0 instead of -1, then it will give the wrong answer as the attributes having volume =0 will be excluded. Quantile binning for association mining of volume is appropriate as the volume is defined as integer type data and this data cannot be included in the association rules as in the association rules , you can only calculate for categorical data, so, henceforth, data for the volume was binned in 5 quantiles so that it can become a categorical data and be used in association rules. This will also increase the association rules.

c) Rules are:

i) {Sector= Automotive, SubSector = Motor\_Vehicle\_Parts} => {Volume=Q2 range}

Support = 0.01248288, Confidence = 0.5219707, Lift = 2.1063282, Count = 392

ii) {SubSector= Investing/Securities\_Companies, Change=up} => {Volume=Min range}

Support = 0.01850142, Confidence = 0.5707269, Lift = 2.2741451, Count = 581

iii) {Subsector = Chemicals, Change=up} => {Volume=Min range}

Support = 0.01471197, Confidence = 0.5789474, Lift = 2.3069007, Count = 462

Under the objective measures of patterns interestingness, we can say that confidence for all rules is mostly equal to 0.5, i.e., all three could occur. Moreover, Lift for all three is greater than 1, so, all three rules are frequent and are independent.

Under the subjective measures of patterns interestingness, we can say that the rules derived for the volume are not interesting as support values are really low as well as confidence, i.e., the occurrence of that rule is also not much high and rules will not be mostly implemented as the values are too low .So, I think the rules made for the volumes are not interesting.(read it again , not sure)

d) Association mining is not useful technique for this dataset. The reason for this is that the rules formed in the dataset are not interesting as they have low confidence ,support as well as lift . The binning for volume also didn’t turn out to be good as the rules formed by volume are also not interesting. The general rule which has highest support and confidence is:

{SubSector=Mining\_&\_Metals} => {Change=up}

The support for the same is 0.14 and the confidence is 0.77. Though the confidence is high, the support is very low for a rule to be interesting. Moreover, this rule has highest confidence and support and we can deduce that the other rules would be less than this due to which this dataset cannot form interesting rules. So, we can, in general say, that the association mining is not useful technique for this data.

**4.**

a) It is a very easy task as change is a categorical data and has only two values, up and down which depends on the Close.Open. Change will be up if Close.Open is greater than 0 and will be down otherwise. So, it will be easy for the learner as there are only 2 variables being used on which algorithm has to be used for calculating accuracy and both are defined in such a way that they are interlinked and dependent on each other. Moreover, we can expect higher accuracy as well as there would be less computational time as variables to be evaluated are less and are correlated.

b)

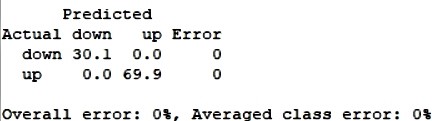
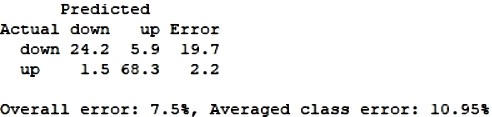
 

Fig 4.1(Error matrix for the decision tree) Fig 4.2(Error matrix for the SVM model)

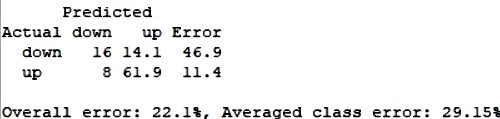
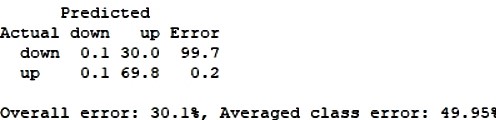
 

Fig 4.3(Error matrix for the Linear model) Fig 4.4(Error matrix for the Neural net model)

The above figures show the confusion matrix with the error for decision tree, SVM, linear, Neural net classification respectively. Decision tree shows that there is 0 percent error and its prediction made on testing and validation data is correct whereas others have 7.15, 22.1 and 30.1 percent error respectively. So, the accuracy for the models are 100 % - error got in every model. Therefore, accuracy here would 100, 92.75, 77.9, 69.9 for decision tree, SVM, linear, Neural net classifier respectively.

c)

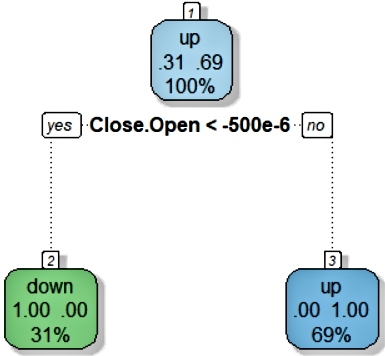


Fig 4 c)

(Represents the tree for the tree classifier as Change as the target)

So, over here, neural net gives an accuracy of 69.9 percent which is not good. It basically it uses the variables and then puts the hidden layers in front while calculating and adds bias to it and gives the accuracy. Over here, the number of hidden layers taken is 10. Now comes the linear classification model in which the model is linearly separated by a line and its accuracy is 77.9 percent which is better than the neural net accuracy. SVM model basically makes the line for dividing using the vector points nearby so that the distance between the line and the vector can be maximum and the kernel used for this is radial bias. So, after making the accurate line, if you test it on test set, the accuracy turns out to be 92.75 percent which better accuracy than that of neural net and linear classification.

The decision learner performed the best as it can be deduced from the Fig 4.1. The accuracy for the same is 100 percent which is best out of 4 classifiers. Though 100 percent accuracy at the first sound inappropriate as from seeing the result, we might conclude that it is overfitting. But it gave 100 percent accuracy because of how change is being defined. As we can see from the graph that if close.open satisfies is less than the 500e-6, then , Change is down or otherwise it is up. Change is defined in a dataset in a way that if close.open is negative then, change is down otherwise it is up. So, it obvious that if it is defined in such a way that they are interlinked, then surely the accuracy would be 100 percent. For the confirmation, algorithm is also tested on the training set to avoid overfitting but on training set also it gave the same accuracy.

**5)**

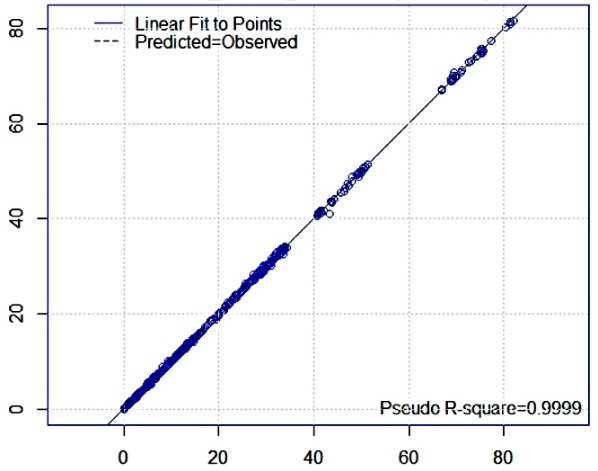
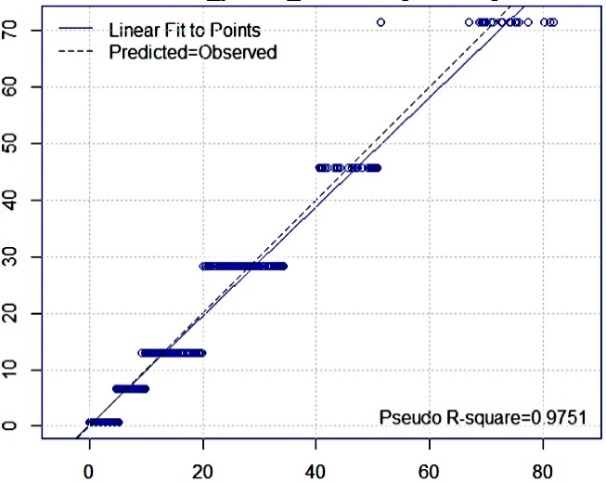
 

Fig 5.1(Pr vs Ob for neural net) Fig 5.2(Pr vs Ob for decision tree)

a) If you run the rattle using the variables given by the assignment, you will see that neural net has R squared value equal to 0.9999 and R squared value of tree is 0.9751. So, for checking whether there is an overfit or not, algorithm is tested on training dataset and the R-squared value for neural net comes out to be 0.9999 and for tree, it is 0.9767.So, for the second opinion , I calculated the MAE value .MAE is basically difference between actual value – predicted value. MAE having value 0 is considered as a best fit. This is because if the MAE is 0, then we can say that the prediction is 100 percent correct. In this scenario, neural net has MAE value equal to 0.0209171 whereas decision tree has MAE value equal to 0.9431273. As neural net has the lower value of MAE as compared to MAE value of decision tree as well as the R squared value for neural net is higher than that of decision tree, we can say that neural net regressor is good for this dataset.

b) Here, I varied the hidden layers in the neural net as well as changed the activation function of the neural net. Here, for calculating accuracy, MAE is used instead of Pseudo R squared values because changing the values for neural net on the GUI didn’t affect the neural net. So, the neural net was coded manually so as to get a MAE. So, the first hidden layer value is 5 and second-layer value is 3. These number of layers were selected so as to keep first hidden layer equal to the number of variables used. Then, varied the number of first hidden layer by decreasing its value to 3 as well as increasing its value to 9. But at the end, 5 layers at the first hidden layer and 3layers at the second hidden layer which is exactly the same assigned at starting turned out to be best because when the hidden layers are increased to 9, so there is a chance that there is overfitting in the algorithm and when tested and if 3 hidden layers are used, then the rattle shows the error as the data given is more complex due to which it cannot sum up in 3 hidden layers. I also varied the activation function. In the starting, I took “logistic” function for the activation function and then tried to use RELU in the neural net but couldn’t be successful in using as the rattle itself doesn’t provide ReLU activation function as default. In the end, I choose logistic for the neural net.

c) Over here, the MAE value is 0.02 which is close to zero. So, this means the accuracy is really good as MAE equal to 0 is a perfect effect. So, the values tried for the hidden layers in neural net has optimal value of 5 layers in the first and on second layer, there are 3 layers and if the value for first hidden layer is changed to 9, then there is an overfitting in the dataset and if the number is increased after 9 , then it is sure that overfitting would be there. If the value is decreased from 5 to 3 in first hidden layer, then it will show an error as the data is much more complex and algorithm cannot be evaluated using the 3 layers at the starting.

**6.**

a) HMLOL is defined as high – low/low. HMLOL is a continuous data and here, we have to find whether the stocks are high or low and a continuous data cannot do the job as we won’t be able to find whether the change is high or low. So, for that reason, HMLOL was neglected and instead of that HMLOL is binned into 2 clusters, high and low clusters using k means clustering which will easily define whether the stock is high or low and which point of the data belongs to which cluster.

The dataset wouldn’t matter much. Though, it is true that having more data for training will give algorithm more data points to make it a better algorithm. But, over here, after sometime, also there is a saturation point after which increase the percentage of training dataset doesn’t matter much. Over here also, after 20 percent, it reaches the saturation point, due to which it doesn’t matter much if the training set 20 percent of the data or 60 percent. So, over here, 60 percent of data is taken for training and 20 percent is for both validation as training set. This dataset is taken arbitrarily. Different dataset can also be taken.

b)

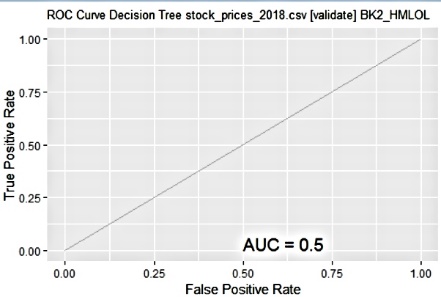


Fig 6.1(Area under curve for the decision tree(initial))

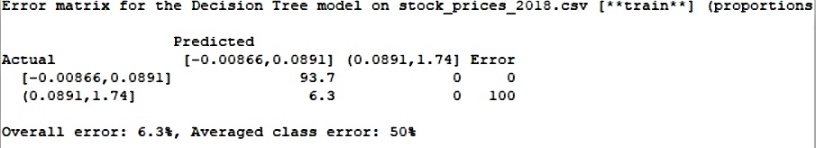


Fig 6.2(Error matrix for train dataset for decision tree(initial))

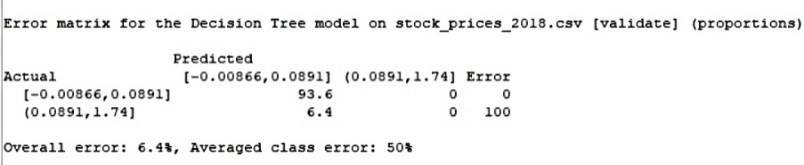


Fig6.3(Error matrix for validation dataset for the decision tree(initial))

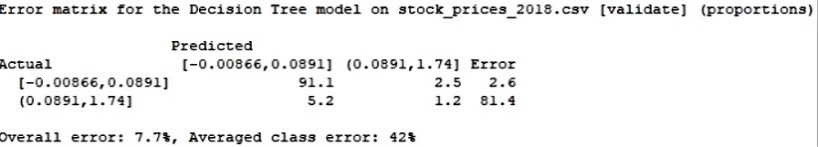


Fig6.4(Error matrix for validation dataset for the decision tree with prior probability of 0.8 for 1st cluster and 0.2 for 2nd cluster)

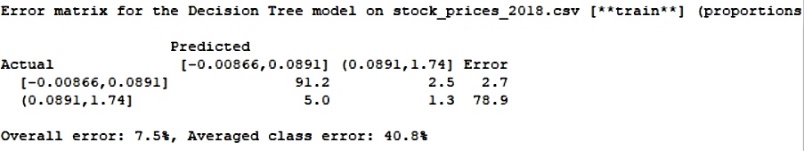


Fig6.5(Error matrix for training dataset for the decision tree with prior probability of 0.8 for 1st cluster and 0.2 for 2nd cluster)

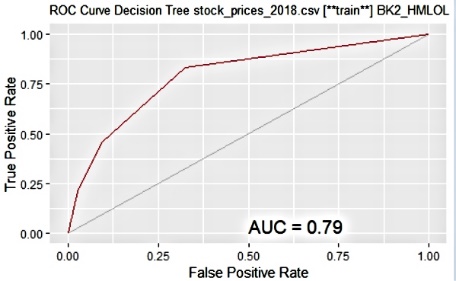


Fig6.6(Area under curve for the decision tree with prior probability of 0.8 for 1st cluster and 0.2 for 2nd cluster)

In this, decision tree is used to classify the data points into whether the HMLOL is high or low. So, in this one, it uses two clusters for the algorithm but when the algorithm is running, it is clearly seen that the algorithm only uses the first cluster and neglects the prediction for the second cluster which is shown in the figure above. Moreover, when you try to plot the tree, you will see the there is only one node. So, the output, we will be getting is bogus as it will predict every time that the stock change is always low. So, we have to make changes in our algorithm. Now, here prior probabilities come into play. Prior probabilities are the probabilities assigned to each cluster for their role in the algorithm. This will in a way force the decision tree algorithm to use both the clusters and here, the probability given to first cluster is 0.8 and 0.2 to other. The probabilities are assigned in this way as the first cluster contains 10 times the number of values of cluster second. So, after the assigning of probabilities, algorithm also starts predicting for the stock change that is high sometimes. So, area under the curve increases from 0.5 to 0.79 and average class error also decreases. This increase in ROC and decrease in the average class error is all because of the prior probabilities that are set for the use. However, in this, as we are forcing the algorithm to work in a certain way due to which it could be overfitting. So, to see for that issue, the algorithm is run on both validation and training set which is shown in the figure 6.5. So, the difference between accuracy as well as the training set is very less. So, we can also say that there is no overfitting for that model. So, we can say that the new algorithm is far better than the previous one as less error which means that less mistake would be made per class in the algorithm.

c)

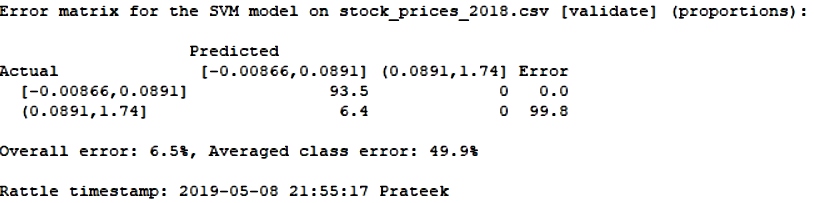


Fig 6.7(Error matrix for radial bias kernel with cost = 0)

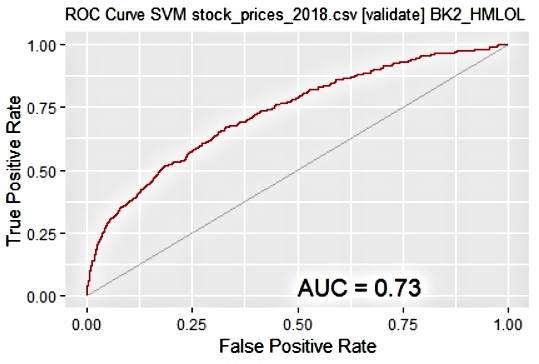


Fig 6.8(Area under curve with radial bias kernel with cost = 0)

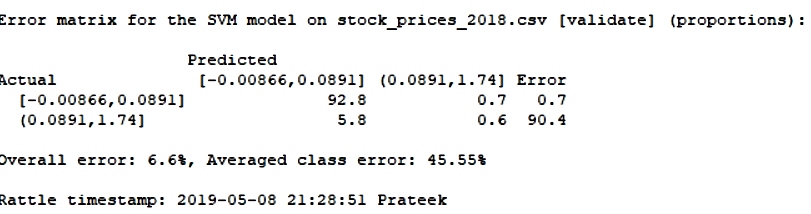


Fig6.9(Error matrix for validation dataset with radial bias with cost =30)

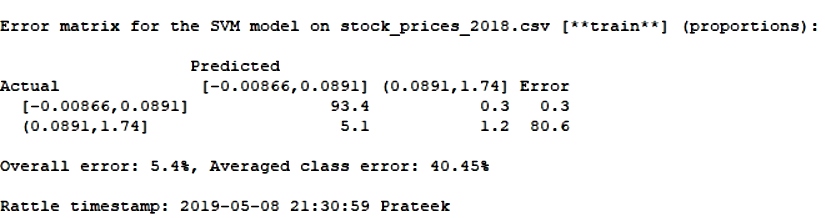


Fig 6.10(Error matrix for training dataset with radial bias with cost =30)

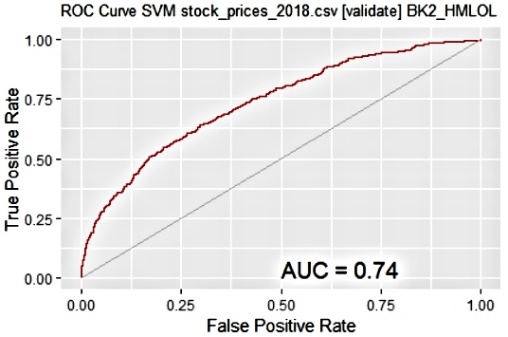


Fig 6.11(Area under the curve for the svm having radial bias kernel with cost = 30)

In svm, radial bias kernel is being used to classify that the points belong to high stock price change or low. So, if we run the code for the svm , then we will get the output as shown in figure 6.7 and it’s ROC can be seen in the figure 6.8. The algorithm used gives very poor results. The reason is that the algorithm doesn’t predict for the high stock change (2nd class) and predicts that every time for 1st class which is the lower one and the answer for the model is that the stock change is always low. Here, average class error is also 50 percent which is quite high and should be decreased. So, for improving the model, C, i.e., cost has been used in the algorithm for betterment. C is basically how much we penalize the SVM for data points within the margin (where "within" can also mean on the wrong side of the dividing hyper-plane) [2]. If we use cost =30, then average class error is decreased from 49.9 percent to 45.5 percent on validation. If you run the program on the training data set to check for the overfitting, then the average class error would be 40.45 percent which is not much difference. So, overfitting isn’t there for this model having cost = 30. The area under curve for this can be seen in fig 6.7. So, the area under the curve has been increased from 0.73 to 0.74. As we know that increase in the ROC means that the algorithm is good, i.e, more the area under the curve, better fit is the model. Moreover, the average class error has been decreased from 49.9 percent to 45.5 percent which means that it will commit fewer mistakes for each class. So, we can say that the cost = 30, works in our favour as class average is decreased and also ROC value is increased.

d)

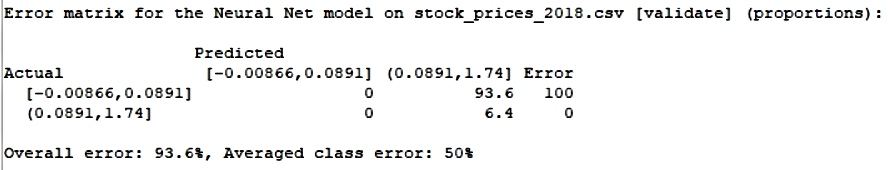


Fig 6.12 (Error matrix for the neural net predicting only class2)

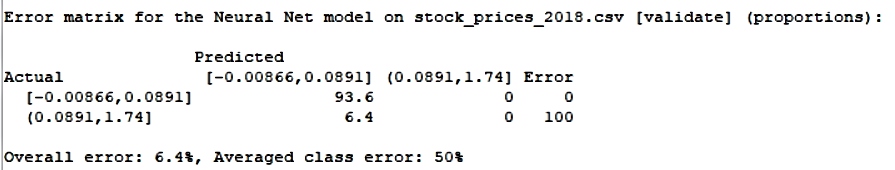


Fig 6.13 (Error matrix for the neural net predicting only class1)

So, in neural net, if the number of layers is changed, we get only these two-error matrices. The activation function used in the neural net is logistic. We can also use ReLU activation function which is far better than the logistic and tanh but in rattle, ReLU is not there in rattle. So, it cannot be used. So, in the neural net, if the hidden nodes equal to 8, we got the error matrix shown in the Fig 6.12 whereas when the number of hidden layers is equal to 10, then the error matrix produced is shown in the figure 6.13. Both of them are bogus in a way as both only predicts only for one class. But the second error matrix would be a better one as it always predicts that stock change is low. Though it’s not an optimal solution, still, it is better than the first error matrix. It is better than the first error matrix because in the first error matrix, accuracy is 6.4 percent whereas, for the second error matrix, the accuracy is 93.6 percent. Moreover, the reason , why second would be discarded is because it gives 93.6 percent false positive results which should be avoided at stock price as predicting high and get low change is more harmful in stock market than predicting low and getting high. So, accordingly, the accuracy for the second error matrix is 93.6 percent which is a good accuracy and can be good solution between these two.

**7)**

a) Clusters should be 5 because if elbow method is used to determine clusters, then according to that, where the bent is or the elbow is considered as the best cluster point and the bent is there at clusters equal to 5. So, we can say that the 5 clusters can be used. In k means clustering, we can just probably say that this number of clusters is possible for this particular dataset as the algorithm is itself NP hard due to which it is hard to compute the apt clusters for the algorithm. So, there is a limitation that there is no way that we can find the optimal clusters.

b)

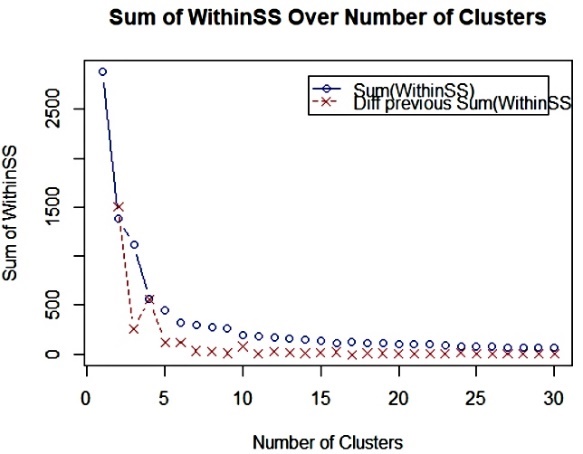


Fig 7(Represents the difference between the cluster points and its centroids square vs number of clusters)

The interesting about the measure is that the within-cluster-sum-of-square helps in finding the number of clusters appropriate for that dataset in k means clustering algorithm. Moreover, for the clusters, to be good, the within-cluster-sum-of-squares should be as small as possible because if the within-cluster-sum-of-squares distance is small for every cluster, then we can say that the cluster can be distinguished easily and clustering result would be good. In the graph, it is clearly seen that the within-clusters-sum-of-squares decrease exponentially for this dataset and gets saturated after some point. So, the within-cluster sum of squares for clusters are as follows:

48.16277 ,229.27448 , 54.29300 , 55.11675 , 56.74136

c) Cluster centres are:

R01\_Date R01\_Open R01\_High R01\_Low R01\_Volume

1 0.3683665 0.02147485 0.02167437 0.02144930 0.002044348

2 0.4817069 0.47262734 0.47455713 0.47426018 0.012118316

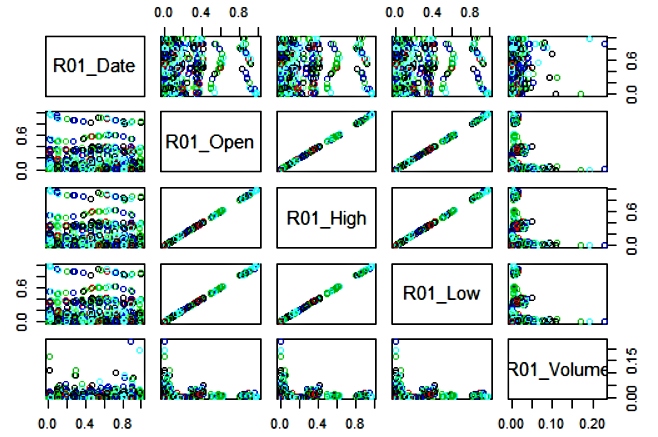
3 0.6300389 0.02157300 0.02177987 0.02152050 0.001775096

4 0.1002500 0.02292444 0.02313581 0.02288252 0.002386815

5 0.8922634 0.02187533 0.02210186 0.02180774 0.002388995

So, these points determine the 5 d centroid pints for 5 different clusters. The cluster centroids or centres are used to get the equal number of clusters to the number of centres. So, the centroid changes according to the points surrounded to it. So, for placing the point in cluster, Euclidian distance is used to find the minimum distance and the centroid having the minimum distance with point, gets that point of the data and centre of that cluster change accordingly as the new points have been added. So, in this each cluster has its own centre and it keeps the cluster different from the other as the distance from the points in the cluster have the minimum distance from its centre as compared to others.

d) The below graph shows the clustering using different variables, taking 2 at a time. If combined and compressed using PCA, we can see the 2 d cluster using 5 variables.



**8**) Association mining for this dataset turned about bad as the rules formed were not at all interesting as it had low support, confidence and value of lift were also small. So, association mining didn’t work here. The speed of calculation was fast but that wouldn’t matter in this case as the rules are not interesting. Clustering on this dataset turned out to be a bad idea as the clusters formed are really bad as we can see from the picture above. So, clustering also didn’t do well. Predicting Open price would be a good idea as the Rsquared for both neural net and tree were around 1 which means that both the algorithms worked well for the Open price. The algorithms used were time efficient as it just a few seconds to evaluate, though the training set was 70 percent of the data, training was 15 percent and 15 percent for validation set. Decision tree classifier worked wonderful for the dataset when change was the target. It gave 100 percent accuracy and moreover, it was time efficient. HMLOL as a target for classifier didn’t do wonders as all the classifiers predicted only one class and those algorithms were forced to train in way using cost function and prior probabilities to some extent because if we exceed the limit, then there the algorithm started showing overfitting on the dataset. In this, it also took around 4 to5 minutes for SVM and around 1 to 2 minutes for another algorithm. So, it was also not much time efficient. The dataset used was 60 percent for training, 20 percent for testing and validation set. So, after all the evaluation, the conclusion can be drawn that these results cannot be used for investment. The reason for this is though, some of the algorithm gave the wonderful results but our main focus is to make investment and we can only make investment, when we know that the change of HMLOL is high or low because when change is high, then only there is a reason to invest as our main motive is to make profit but over here, each and every algorithm didn’t work, though they had a good accuracy but average class error was high as all algorithm only predicted for only one class out of two due to which the results would always be wrong and the change would always be either high or low. So, we can conclude that the results we got is not good for the investment.

We can use High.Low for the evaluation and we can bin that into high or low and can test for the classification models and can get to know whether it is useful or not as at the profit matters and if the change is high, then only there is a point in investing.

**References:**

[1] <https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php>

[2] <https://www.quora.com/What-is-the-intuition-behind-the-Cost-and-Gamma-parameters-in-SVM>