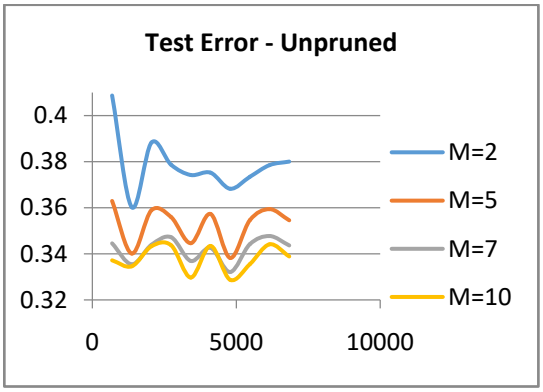
Classification Problem 1

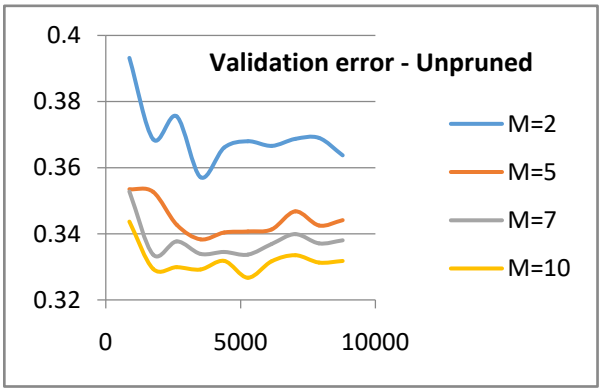
Prediction task is to determine whether a person makes over 50K a year.

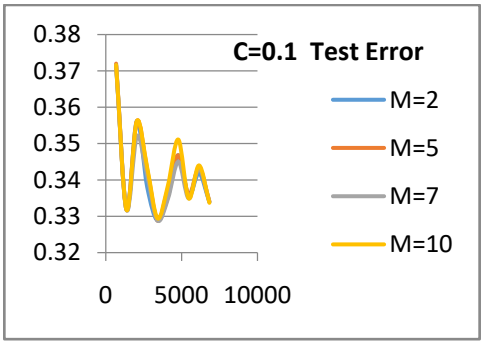
Decision Trees

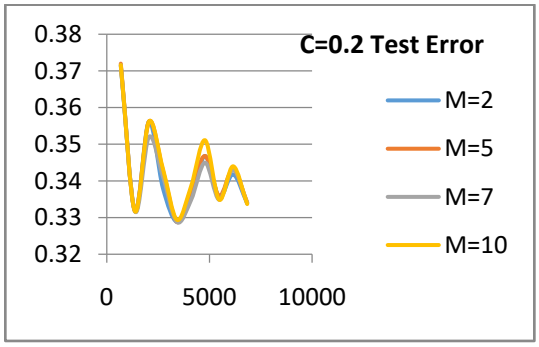
J48 is a Weka Class for generating a pruned or unpruned C4.5 decision tree.

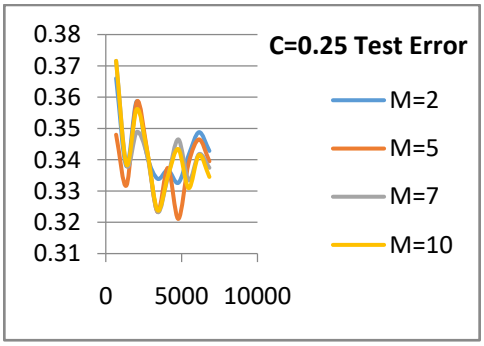
We get the following results for hyper parameters, Min. No. of instances per leaf={2,5,7,10} and confidence={0.1,0.2,0.25,0.3,0.4,0.5} for both pruned and unpruned trees. Confidence value indicates the amount of pruning.

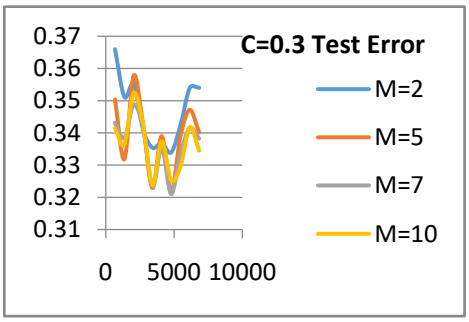


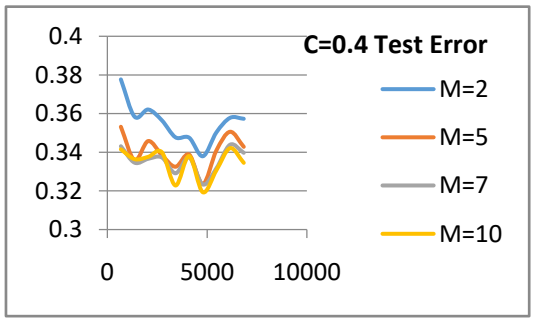


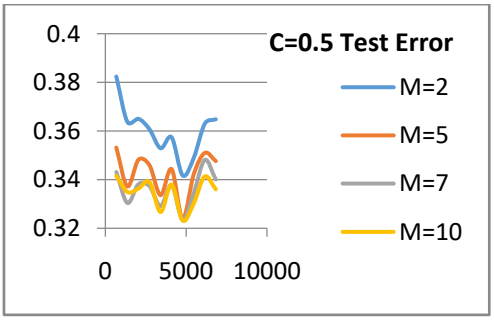


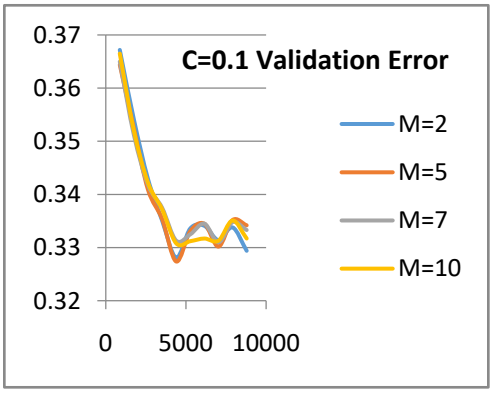


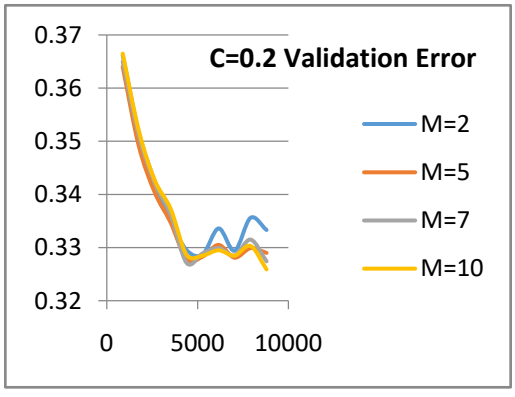


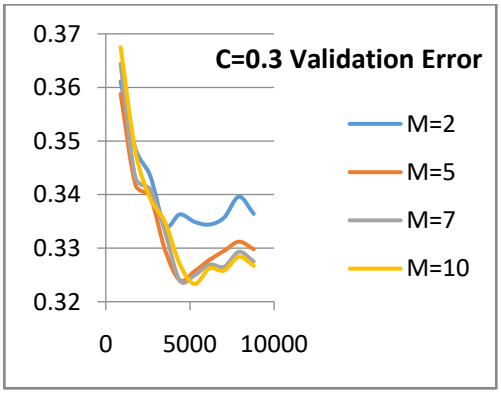


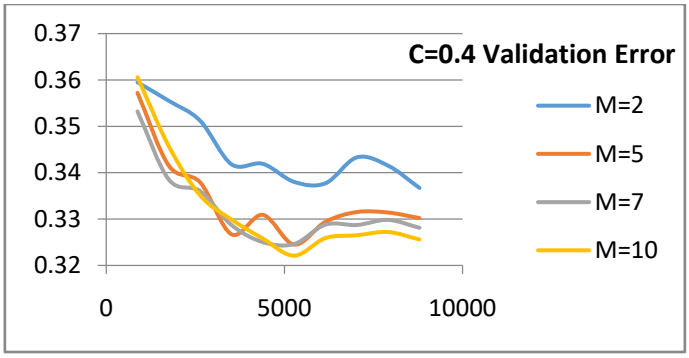


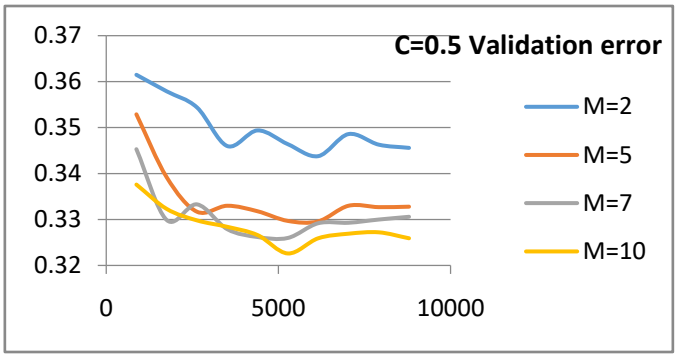




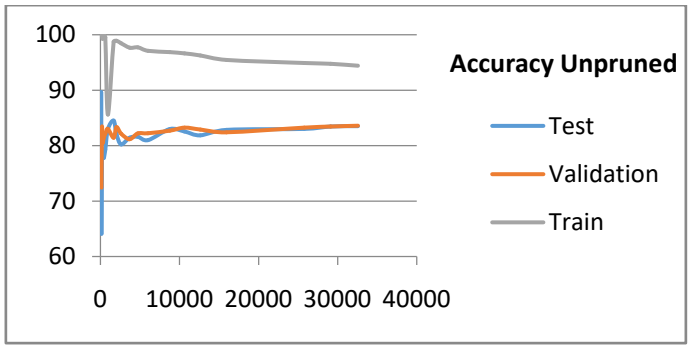


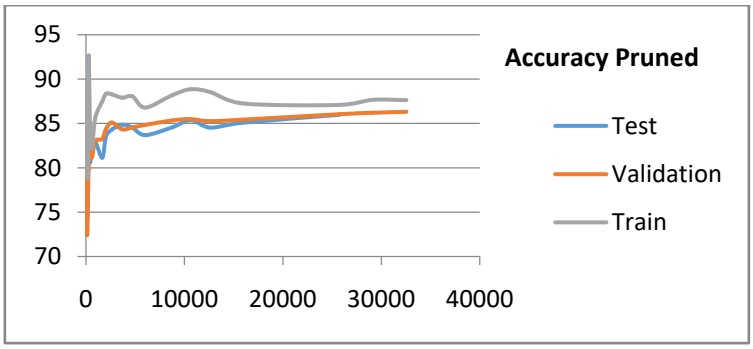


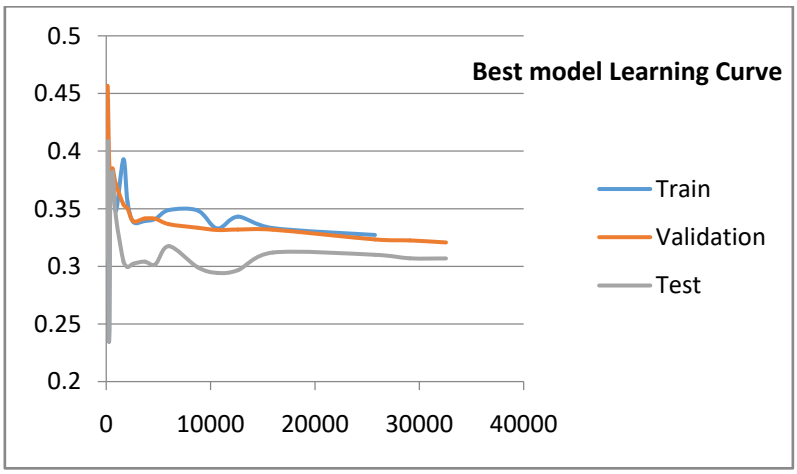




By comparing the accuracy and RMSE, we find that C=0.1, M=2 as the best model which fits the data.







Adaboost

Tried models with the following Confidence and Momentum values.

Table for fig 1

Series No.	Confidence	Momentum
1	0.1	1
2	0.2	1
3	0.3	1
4	0.1	2

Table for fig 2

Series No.	Confidence	Momentum
1	0.2	2
2	0.3	2
3	0.1	5
4	0.2	5
5	0.3	5

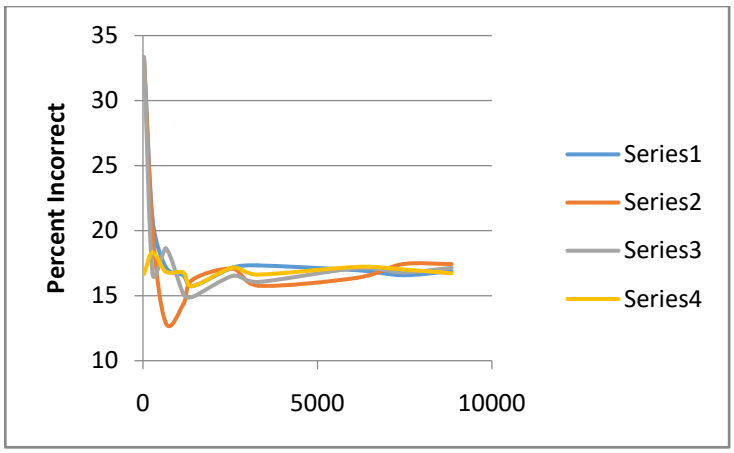


Fig 1

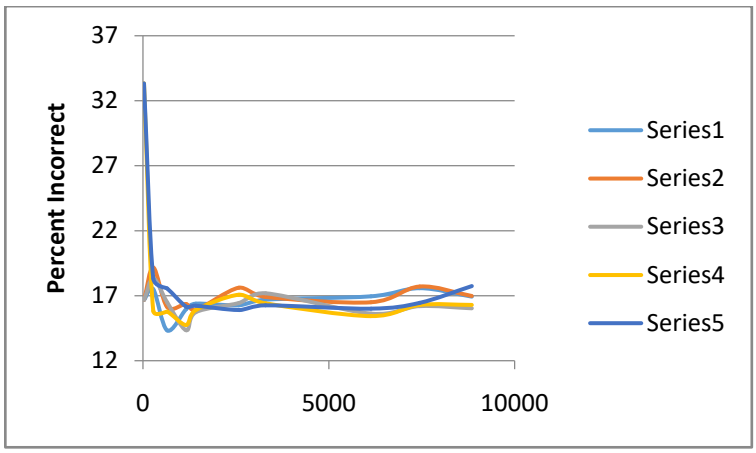
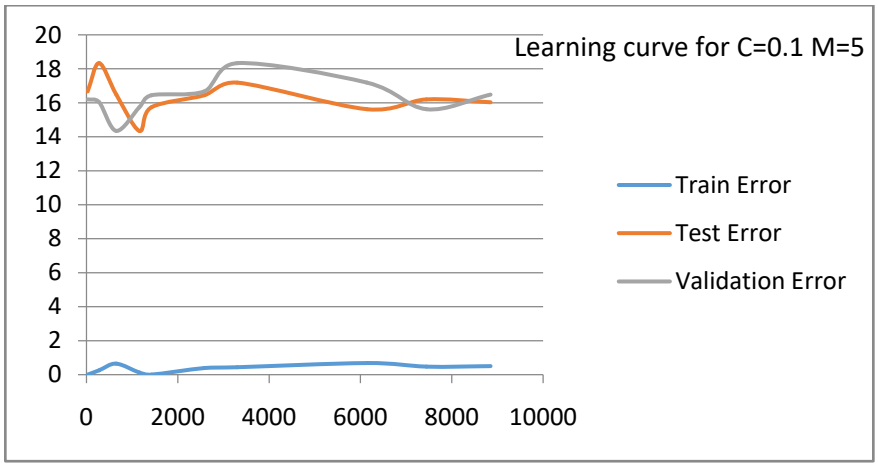


Fig 2

Found the best model with C=0.1 and M=5



Artificial neural networks

A Classifier that uses Backpropagation to classify instances. The nodes in this network are all sigmoid.

After trying various 1, 2, 3 layer network models with the following values.

Training Model	Learning Rate	Momentum	Hidden Layers	Error Rate
1	0.3	0.2	8	20.91
2	0.1	0.2	14, 3, 2	16.35
3	0.2	0.2	16	18.23
4	0.4	0.2	8	17.15
5	0.7	0.2	2, 16	16.89
6	0.7	0.4	8	17.42
7	0.5	0.6	8, 14	18.76676
8	0.3	0.8	8	24.12869
9	0.4	0.7	8, 2, 14	24.12869
10	0.5	0.7	8, 2	23.59249

Found the best models in 1, 2, 3 hidden layers as Fig 3. And the learning curve of the best model among all is with 3 hidden layers (L=0.1, M=0.2, Hidden Layers = {14,3,2}) is shown in Fig 4.

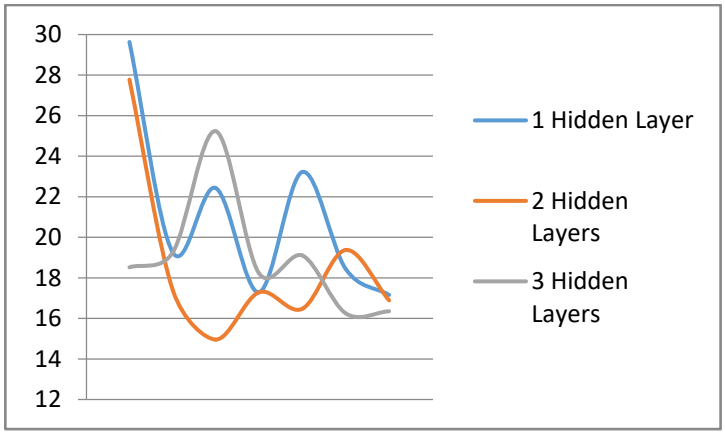


Fig 3

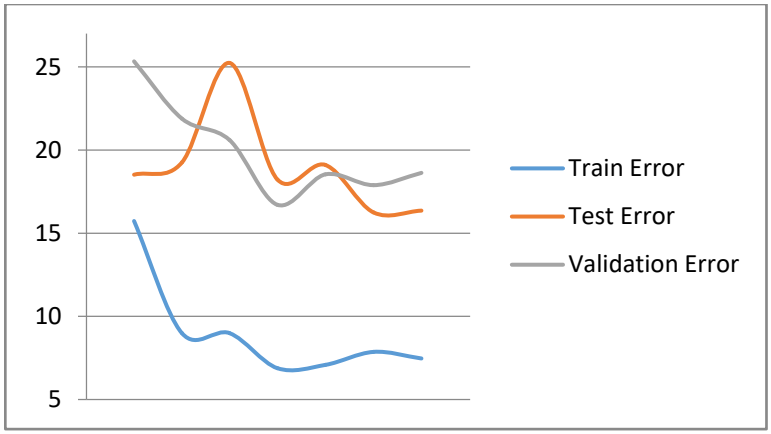
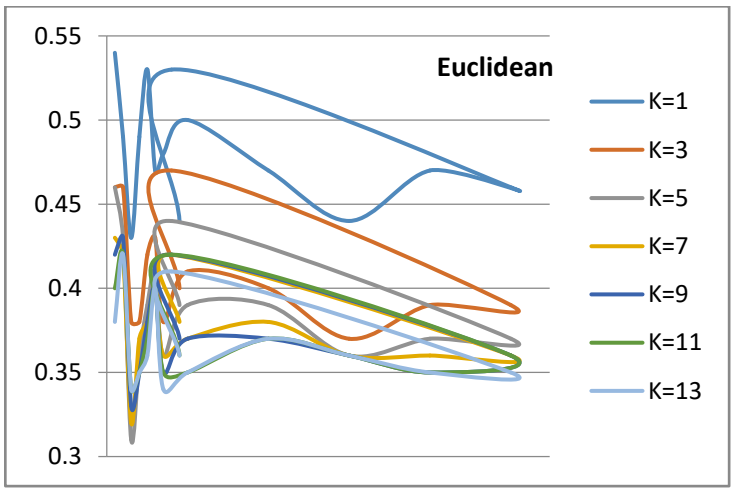


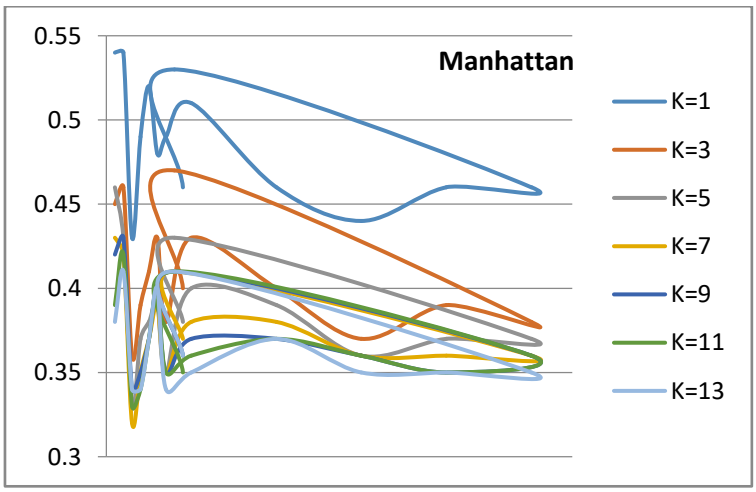
Fig 4

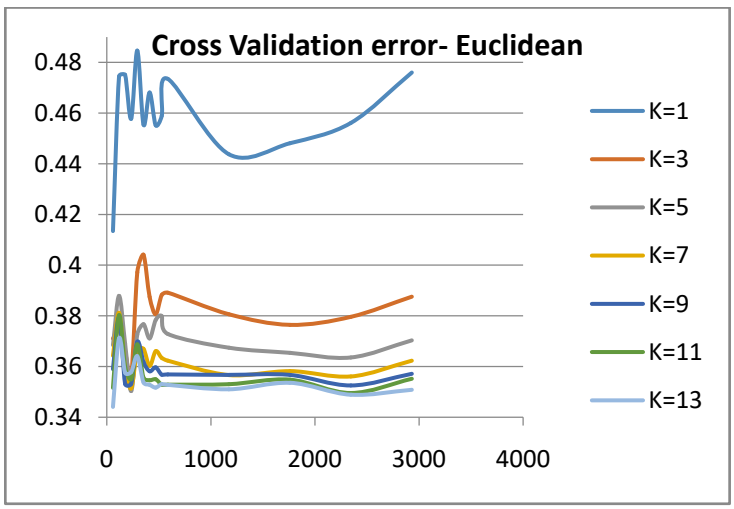
K Nearest Neighbors

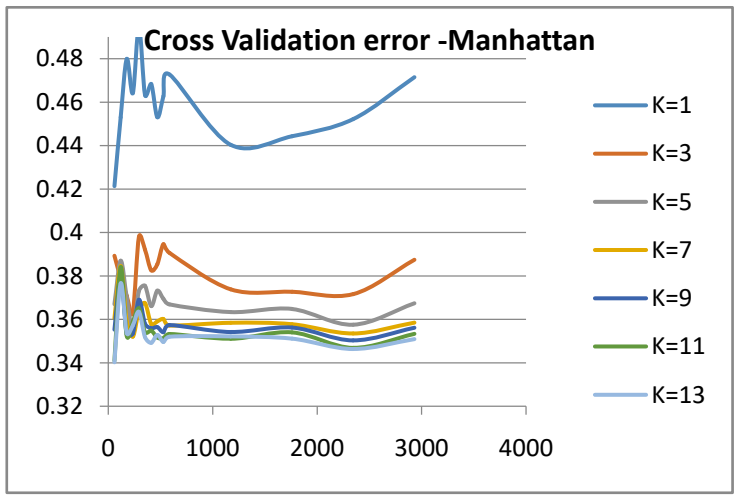
With K=1, 3, 5, 7, 9, 11, 13 and Euclidean and Manhattan distance measures, we get the following results with 70-30 split data.

X-axis: Training Size. Y- axis: RMSE



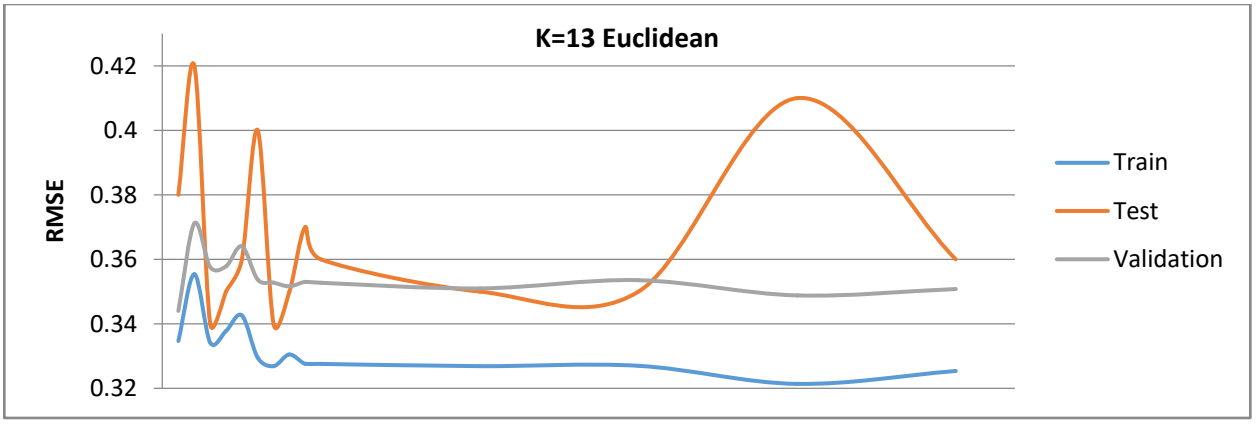






K=13 has done well with both Euclidean and Manhattan with Euclidean being slightly better.

Learning Curve of best model



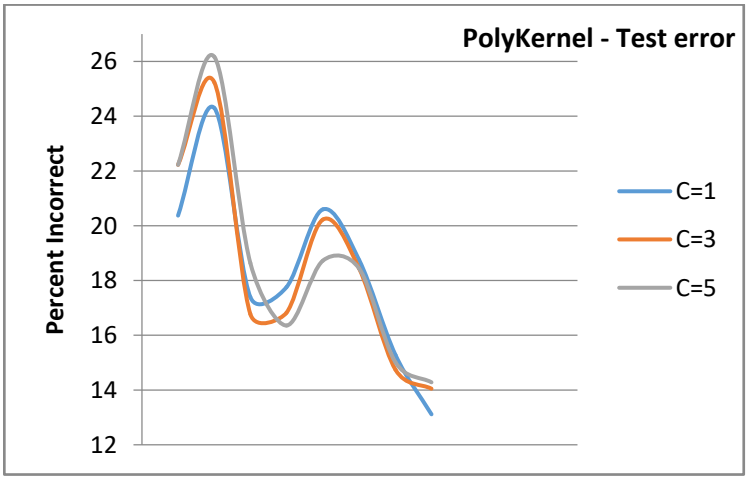
Support Vector Machines

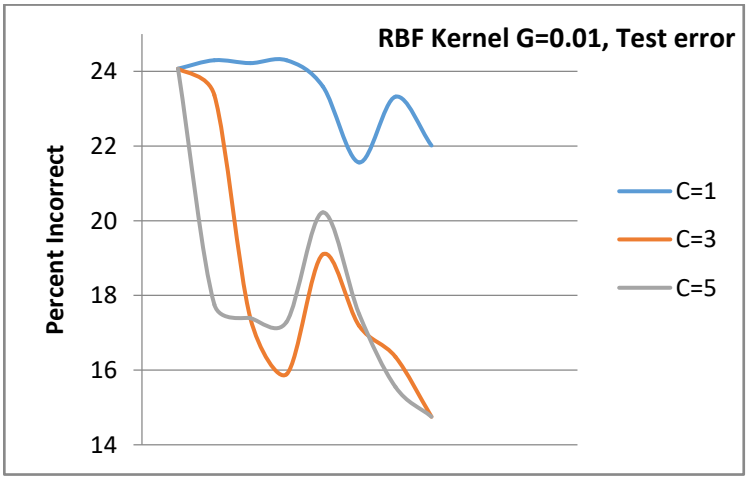
Used two kernel functions poly kernel and RBF kernel.

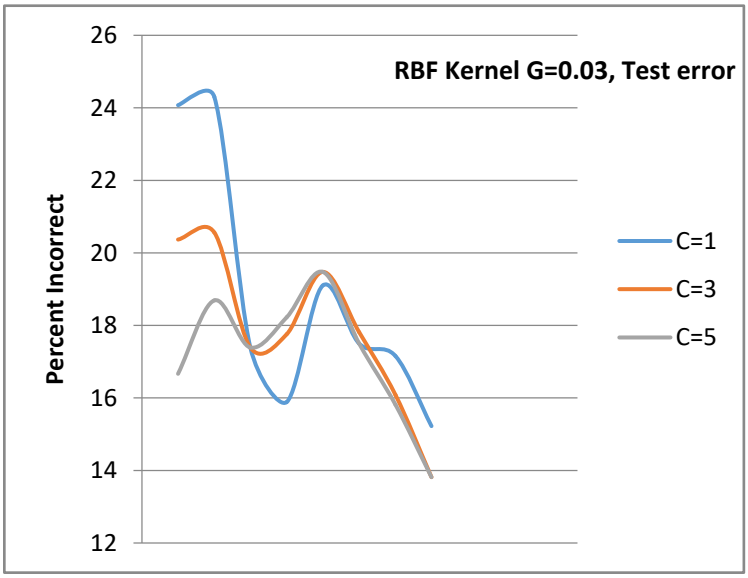
The polynomial kernel : K(x, y) = <x, y>^p or K(x, y) = (<x, y>+1)^p

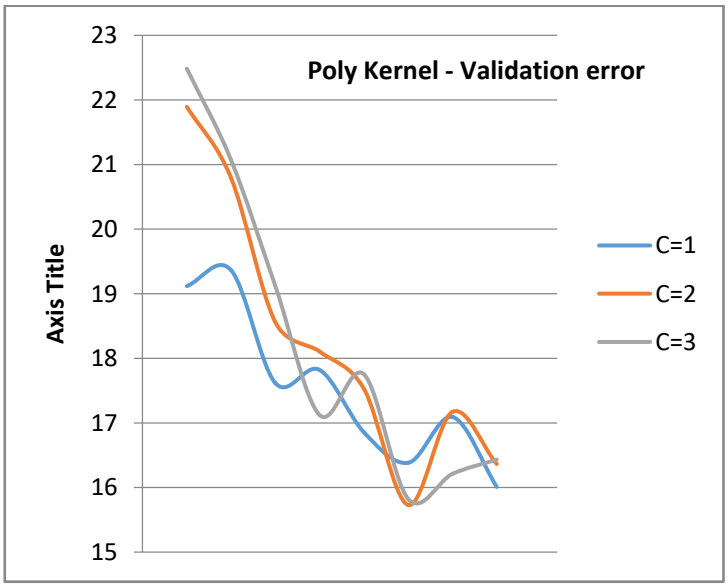
The RBF kernel: K(x, y) = e^-(gamma * <x-y, x-y>)

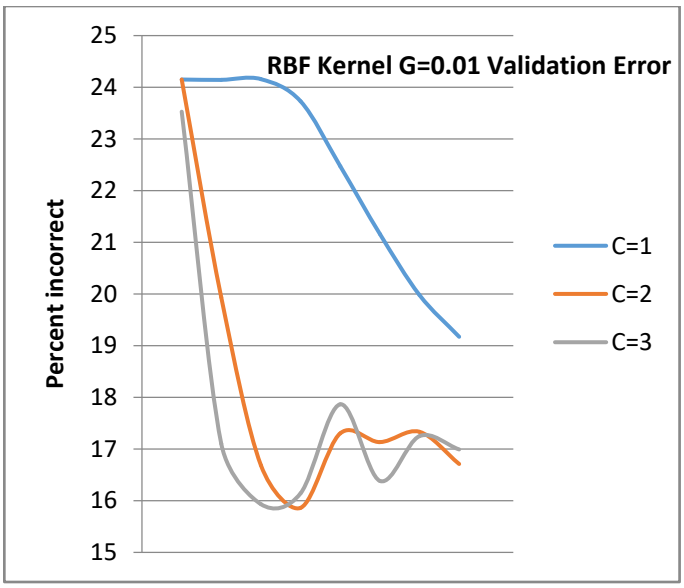
Found polykernel with c=1 as the best one among all the 9 models using both test and validation errors.

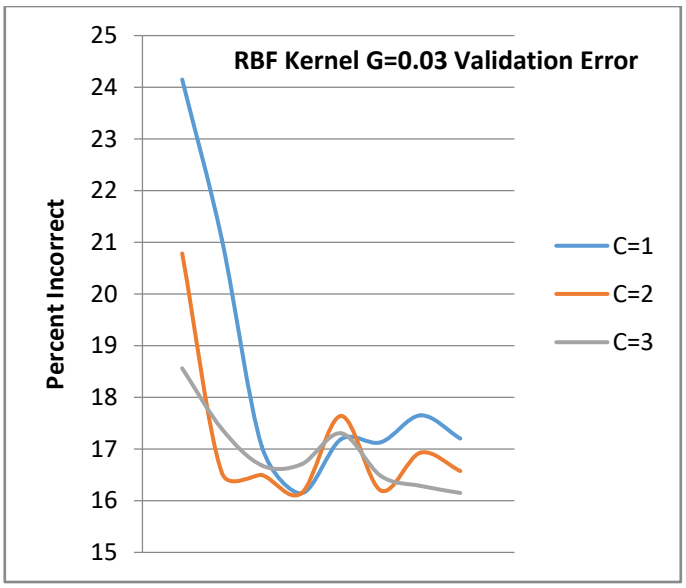


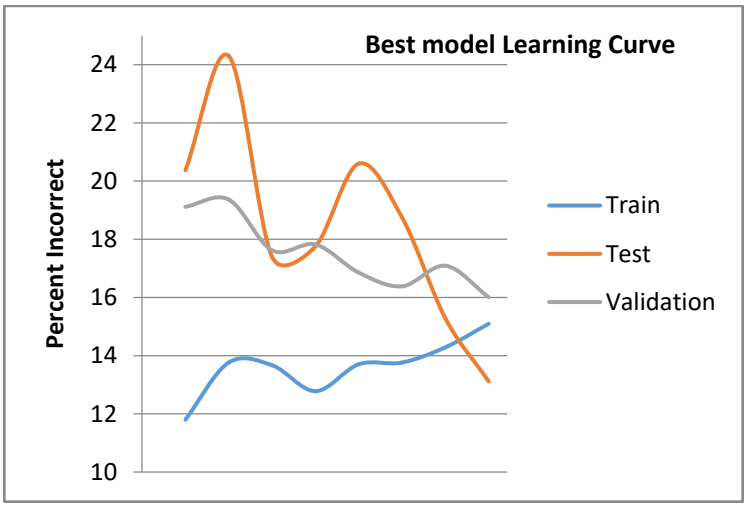




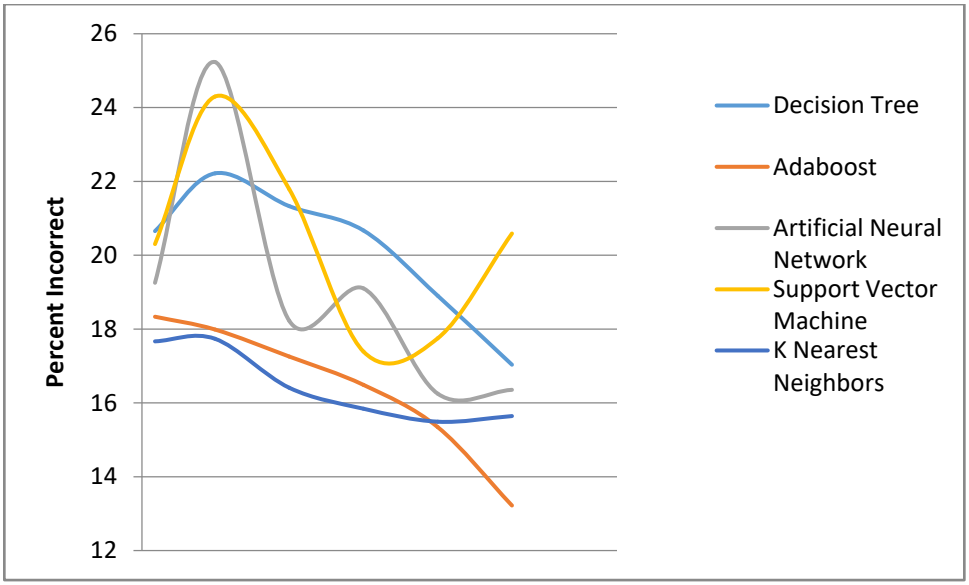








Comparison of Algorithms



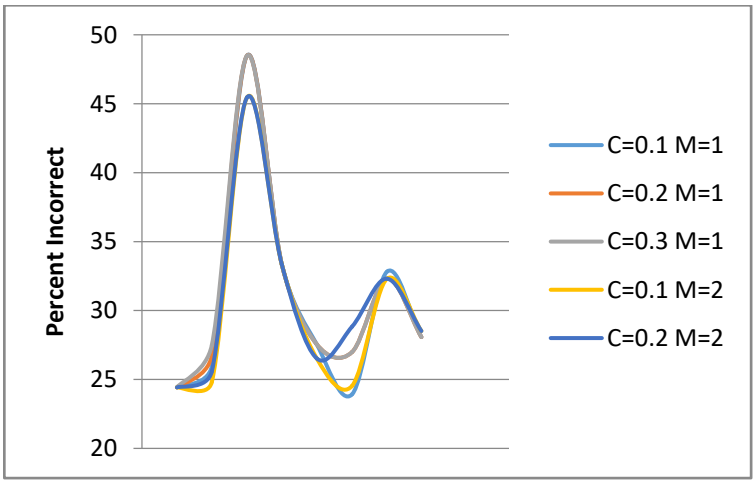
As we can see that boosted version of the decision tree C4.8 has done well over the dataset. This is due to the modifications made by incorporating extra pruning.

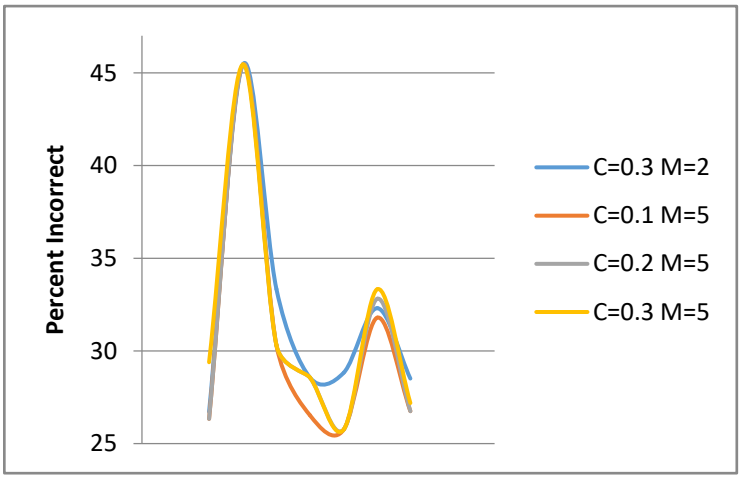
Classification Problem 2

Prediction task is to determine whether the wine quality is good or bad.

Decision Trees

After trying models with Confidence={0.1,0.2,0.3} and Momentum={1,2,5} we get the best model for C=0.1 M=5 with pruning shown in fig 5.





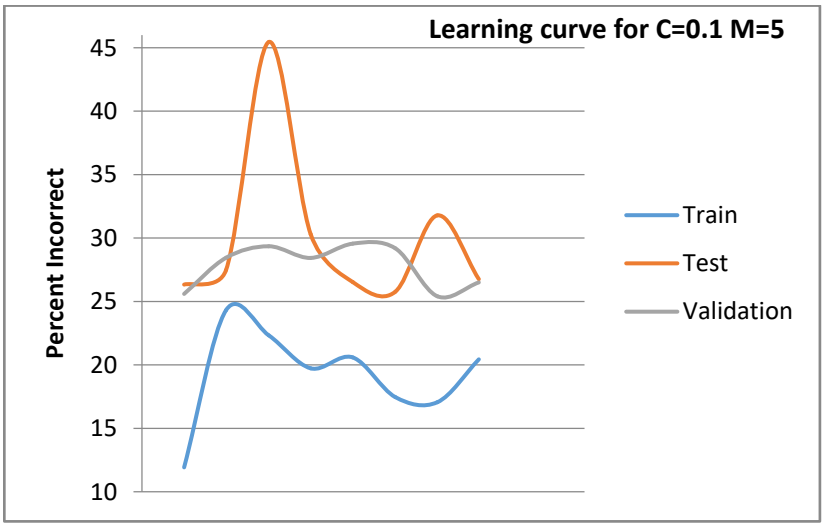
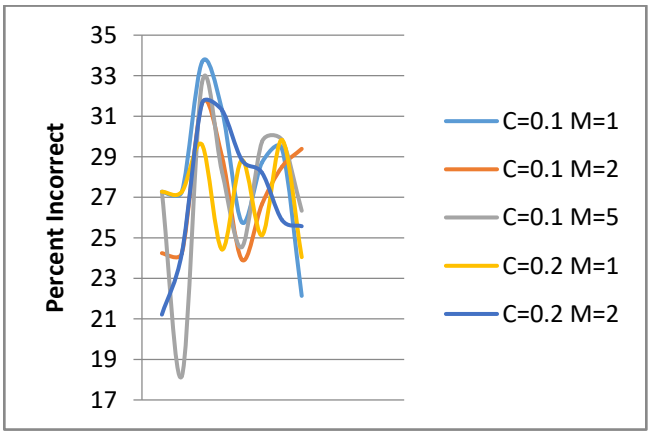
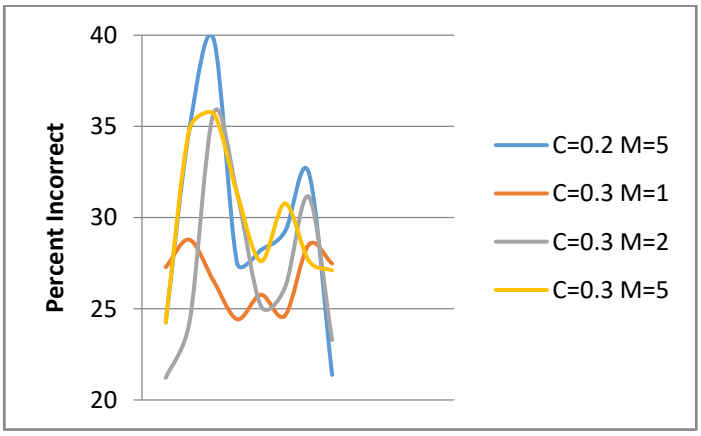


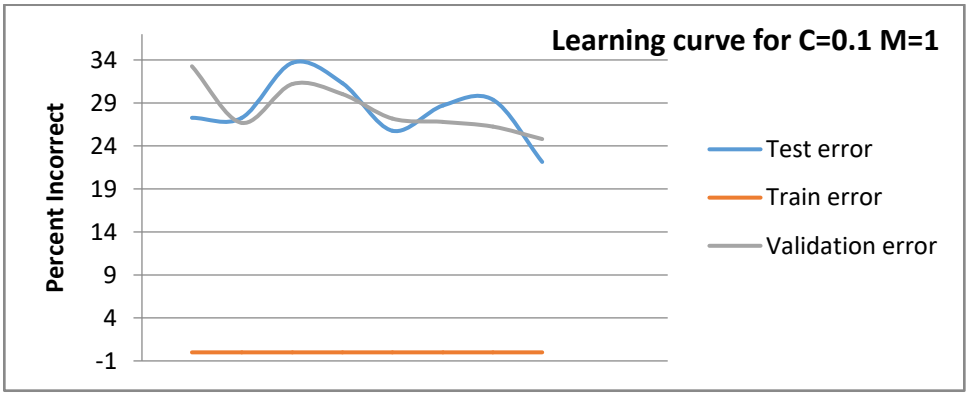
Fig 5

Adaboost

After trying models with C={0.1, 0.2, 0.3} and M={1,2,5} we get the best model for C=0.1 M=1







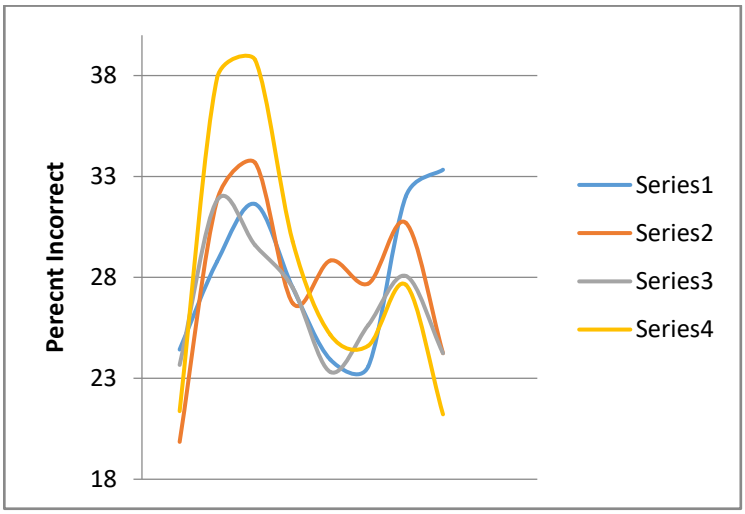
Artificial Neural Network

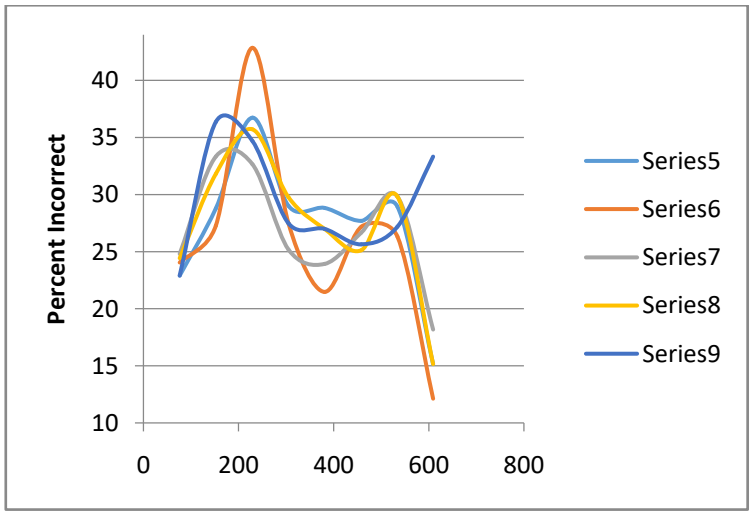
A Classifier that uses Backpropagation to classify instances. The nodes in this network are all sigmoid.

After trying various 1, 2, 3 layer network models.

Training Attempt	Learning Rate	Momentum	Hidden Layers	Error Rate
1	0.3	0.2	7	33.33333
2	0.1	0.4	14	24.24242
3	0.5	0.6	2	24.24242
4	0.5	0.4	5	21.21212
5	0.4	0.5	12,3,2	15.15152
6	0.1	0.3	7,2,12	12.12121
7	0.1	0.2	2,14	18.18182
8	0.3	0.2	7,12	15.15152
9	0.1	0.4	7,2	33.33333

The learning curve of the best model among all is with (L=0.1, M=0.3, Hidden Layer = {7,2,12}) is shown in Fig 6.





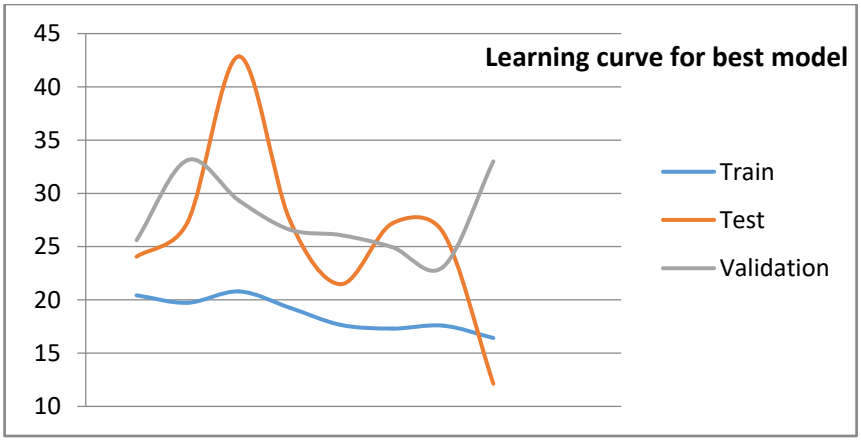
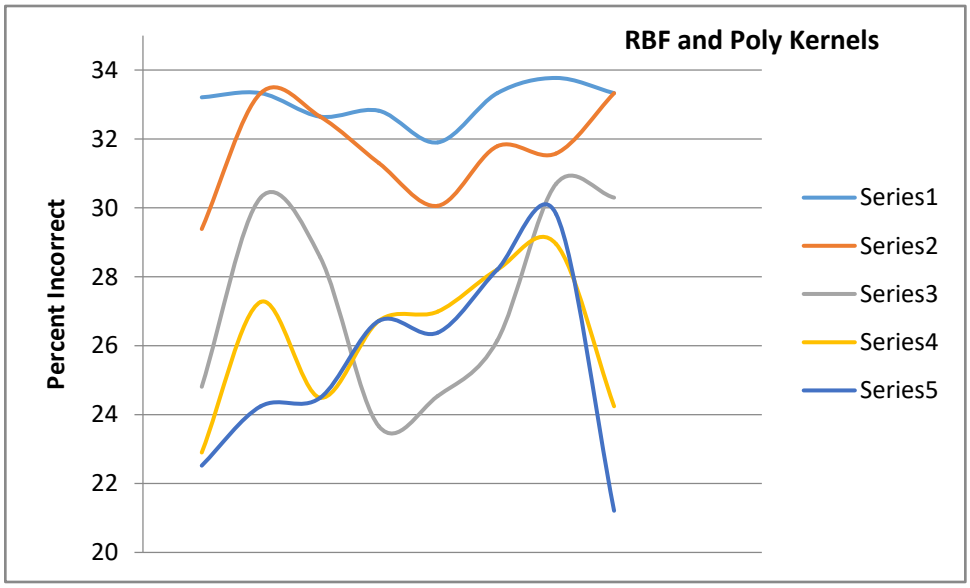


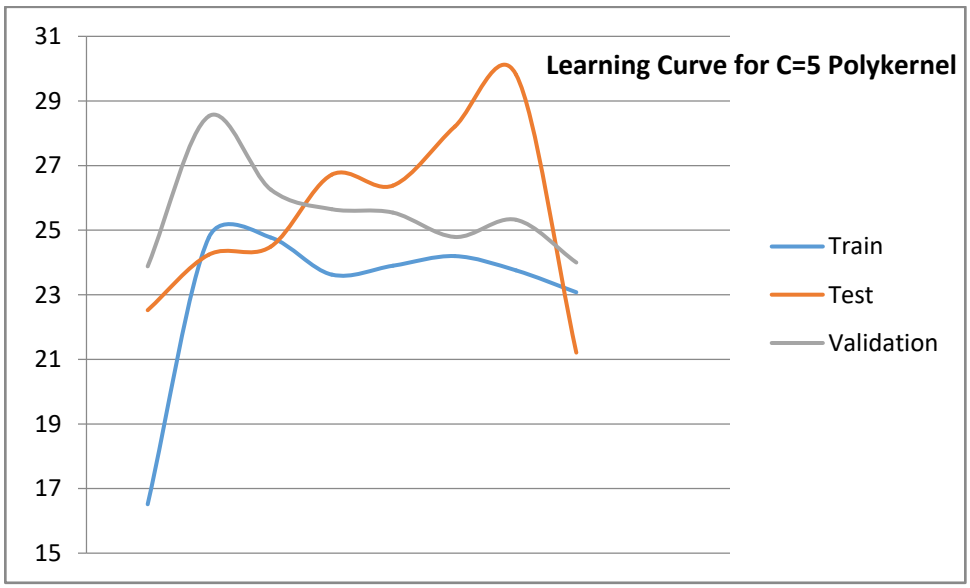
Fig 6

Support Vector Machines

Used two kernel functions. The polynomial kernel : K(x, y) = <x, y>^p or K(x, y) = (<x, y>+1)^p. The RBF kernel. K(x, y) = e^-(gamma * <x-y, x-y>). Tried RBF Kernel with Gamma = {0.01, 0.03, 0.05} and C={1,3,5} and Polykernel with C={1,3,5}.

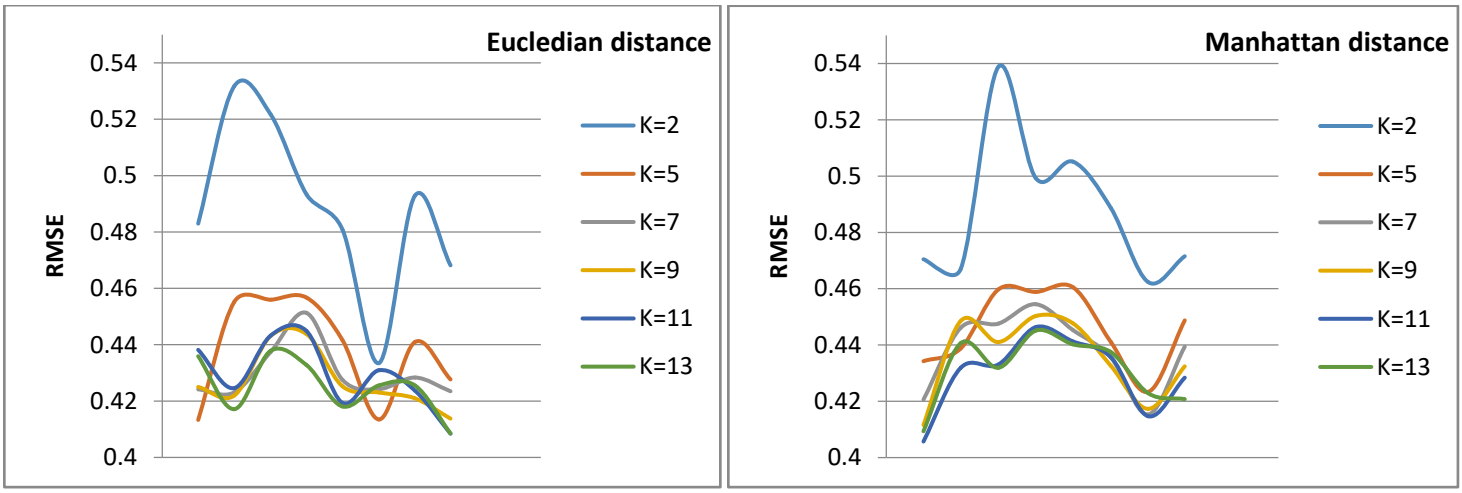


Found the best model with Polykernel function and C=5.



K Nearest Neighbors

Tried K={2,5,7,9,11,13} with Euclidean and Manhattan distance measures.



Found the best model with Euclidean distance measure and K=13 in Fig 7

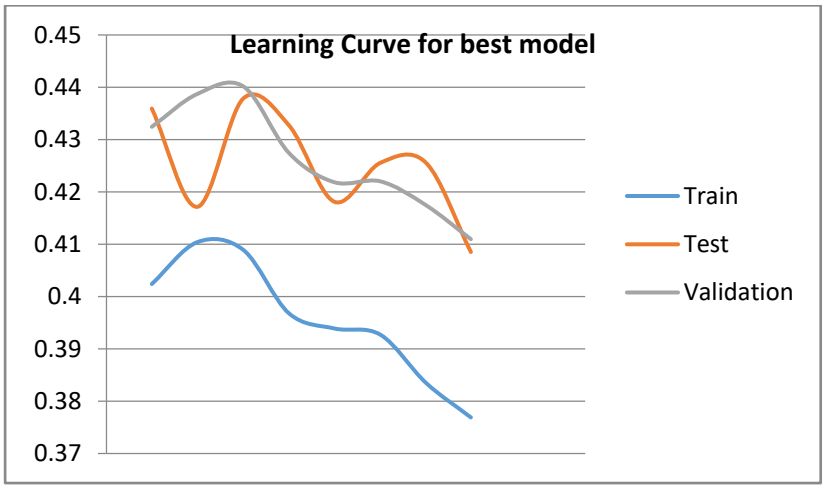
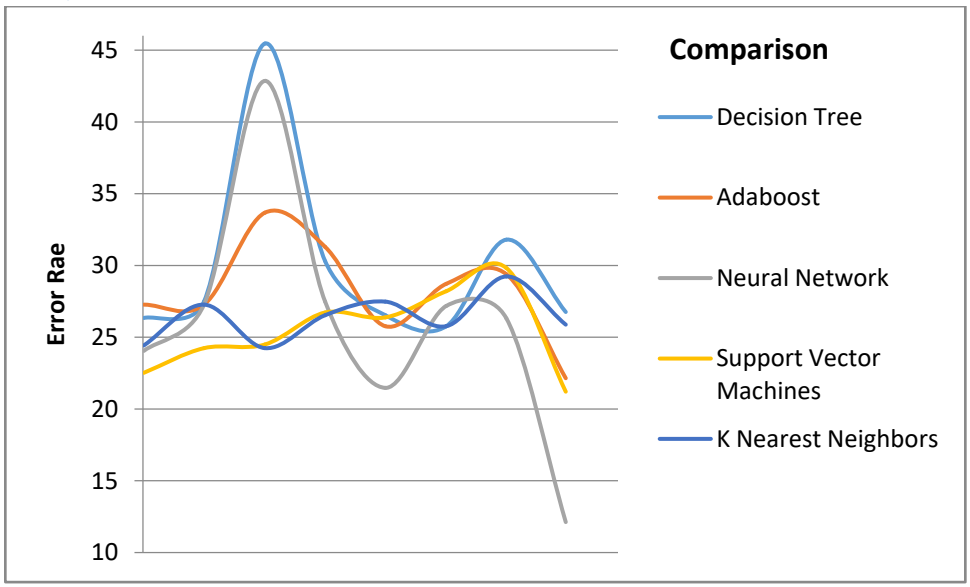


Fig 7

Comparison



As we can see that Artificial Neural Network model has done significantly well over the data.

Why the datasets are interesting?

The datasets are interesting due to the nature of the inherent attribute and their dependencies.

Interesting quality of classification problem 1: The inherent nature of the type of data which has all nominal attributes which describes an adult helps the decision tree to predict if his income is above or below 50K/year.

Interesting quality of classification problem 2: The inherent nature of the data set where the normalized wine characteristic numeric attributes form excellent candidates for a neural network.

Since these two datasets can be used to explain the characteristics of two algorithms, they are interesting.

Elapsed Time training and testing model on test data

Algorithm	Average Time for each training size	Average Time for each model	Average time to test different hyperparameter models using weka experimenter
Decision Tree	0.1s	1.5s	20min
Adaboost	5.27s	59s	1hr
Artificial Neural Network	95.9s	1000s	4hrs
Support Vector Machine	8s	89s	1hr
K Nearest Neighbors	0.9s	98s	30min

Elapsed time to test the models on training data

This took a lot of time as done manually since there is no option in weka experimenter to test with train data.

Elapsed time to cross validate the models

This took more than triple amount of average time of test data

Changes made to avoid over fitting

When there is noise in the data or when number of training examples is too small to produce true target function, trees that overfit the training examples are produced. To avoid this, we can stop growing the tree before it reaches the point where it perfectly classifies training data. Or we can allow the tree overfit the data and then post-prune the tree. (Mitchell et al., 1997).

To analyse this, both pruned and unpruned trees are generated and tested using test set and cross validation.

Bias: Bias are the simplifying assumptions made by a model to make the target function easier to learn. (website1). Less assumptions about the form of the target function lead to Low Bias, whereas, more assumptions lead to High Bias

Variance Error: Variance is the amount that the estimate of the target function will change if different training data was used. (website). Small changes to the target function with changes to the training dataset lead to Low Variance, whereas, large changes lead to High Variance.

The best model is the level of complexity at which the increase in bias is equivalent to the reduction in variance(website2)

Cross validation is used to pick the hyper parameters which are the best fit to the model to reduce bias due to the data.

Choosing the best model

Testing only on training data leads to picking a high complexity model. Whereas testing on the test data, increases error after certain point of model complexity. Cross validation helps in reduce the bias due to the data and predicts a view on unseen data error. Hence, in order to pick the best model, different hyper parameters are used to generate models. Each of the models is tested on trained, test and cross validation folds and the model which has less error is picked as the best model.

To this best model in order to pick the minimum data needed to train the model, learning curves have been plotted against the data sizes by testing on train test and validation folds.

References

Book

Mitchell Tom, M. (2013). Machine Learning. McGraw Hill Education, India. ISBN-13: 978-1-25-909695-2. Retrieved on September 24th, 2017.

Web

Blake and Merz. (1998). Adult Dataset. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Jason, B. (2016). Gentle Introduction to the Bias-Variance Trade-Off in Machine Learning. Retrieved on September 24th, 2017 from https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/

Scott Fortmann, R. (2012). Understanding the Bias-Variance Tradeoff. Retrieved on September 24th, 2017 from http://scott.fortmann-roe.com/docs/BiasVariance.html