Bharath Kumar Natarajan (BXN180005)
6341.003 – Applied Machine Learning (Assignment 3)

Objective:

To implement the following three learning algorithms on two datasets and to perform various experiments, compare and interpret the results: -

1. Artificial Neural Networks (ANN)
2. K Nearest Neighbours

Dataset Details:

Two datasets have been used to implement the above-mentioned algorithms: -

Dataset 1: Appliances Energy Prediction dataset

Dataset 2: Absenteeism at Work dataset

MODEL IMPLEMENTATION:

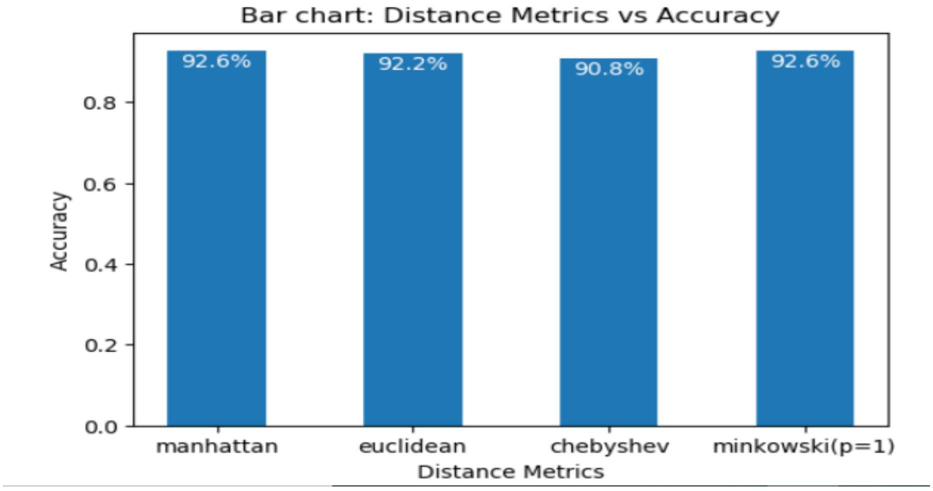
K Nearest Neighbours

The package used for implementing K Nearest Neighbours is KNeighborsClassifier which was downloaded from SciKit learn. The KNN model was implemented and the following experiments were performed :-

Experiments on Dataset 1 (Energy Dataset):-

Experiment on the Distance Metric:

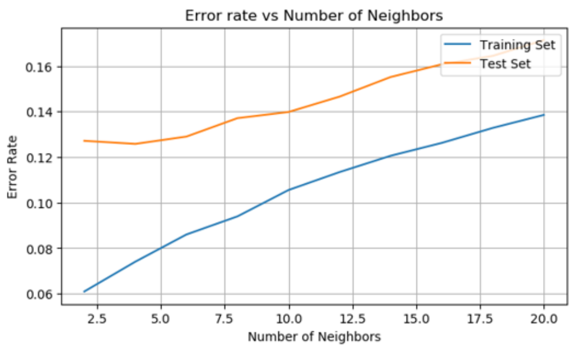
In order to start implementing KNN model for the Energy dataset, first we need to know the best distance parameter for which the given dataset performs well, i.e, for which distance metric the model gives the highest accuracy. For this different distance metrics such as Manhattan, Euclidean, Chebyshev and Minkowski (p=1) distance parameters were considered for the model implementation keeping the default number of clusters as in the package used. The following bar graph shows the accuracy obtained for the models for each of these distance parameters:

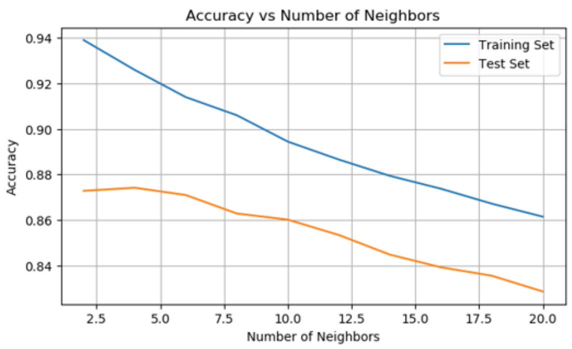


From the above bar graph, we can see that the data performs well for Manhattan distance with an accuracy of 92.6%. the Minkowski with p=1 is same as Manhattan while Minkowski with p=2 is same as Euclidean. So, we will consider Manhattan distance for the other experiments in this assignment.

Experiment on the Number of Neighbors:

For the Energy dataset, in order to implement KNN model, we need know how many clusters is optimal for this model implementation. For this an experiment was performed to find out the minimum error for each number of clusters formed and below output was obtained:-

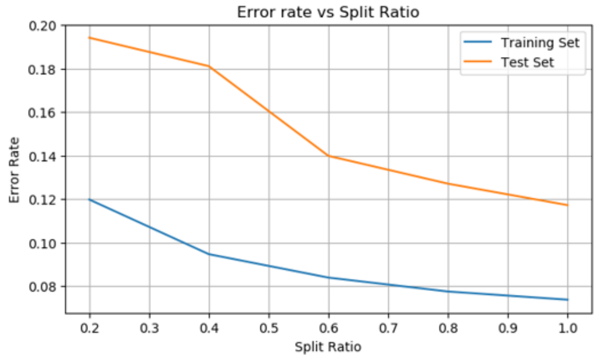


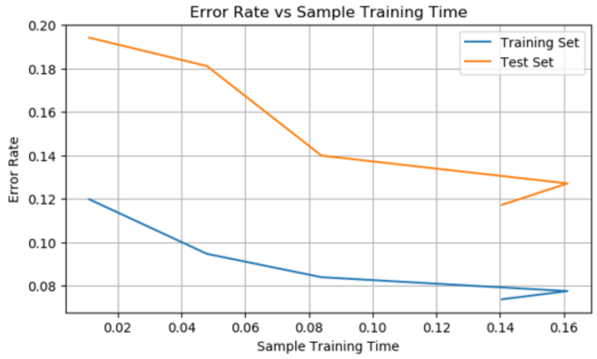


In the above plot of Training and Testing dataset Errors VS Number of clusters, it can be understood that with increase in the number of neighbours, the error rate increases. Now we want to take an optimal value of the number of neighbours to be considered for conducting various experiments. Let's consider 92% accuracy to be optimal for the model. For this accuracy, the number of neighbours in the above plot is 5. The other experiments in this assignment are performed fixing the number of neighbours as 5.

<u>Experiment on the Sample Size and Clock Time: -</u>

Now we have seen that for given distance metric (Manhattan) and for the given number of Neighbors (5 clusters), we are going to perform an experiment to estimate the error of the implemented model with different sample sizes and the clock time for training the data. The following plot was obtained for training and test dataset for different sample sizes plotted against each of its error rate:



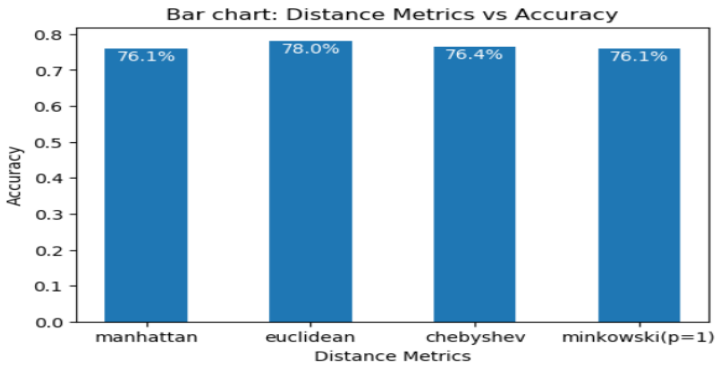


From the above plot we can see that the error rate of the model tends to decrease with increase in the sample size and in the other graph the error rate tends to decrease with increase in the training time. The training accuracy of the full sample size of the training Energy dataset is 92.63%.

<u>Experiments on Dataset 2 (Adult Dataset):-</u>

<u>Experiment on the Distance Metrics:</u>

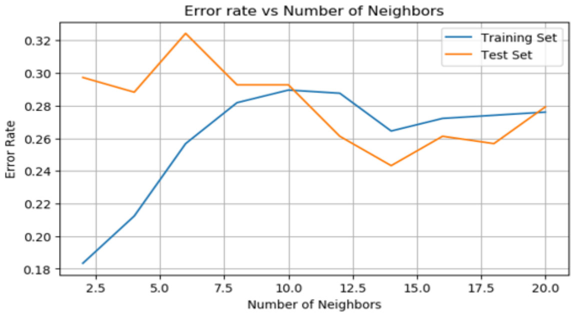
To start experimenting with the given Adult dataset, we need to find the best distance metric for the Adult dataset. For this different distance metrics such as Manhattan, Euclidean, Chebyshev and Minkowski (p=1) distance parameters were considered for the model implementation keeping the default number of clusters as in the package used. The following bar graph shows the accuracy obtained for the models for each of these distance parameters:

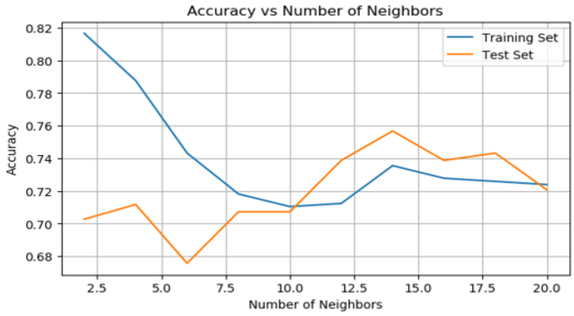


From the above bar chart, it can be understood that the model with highest accuracy is for the Euclidean distance metrics. Therefore, we are going to consider Euclidean distance as the distance metrics for the upcoming experiments.

Experiment on the Number of Neighbors:

For Adult dataset, we need to find out the ideal number of neighbours for which we have the ideal error rate so that the same can be used for the upcoming experiments. The below plot was obtained for different number of neighbours keeping Euclidean distance as the ideal distance metrics for the Adult dataset:

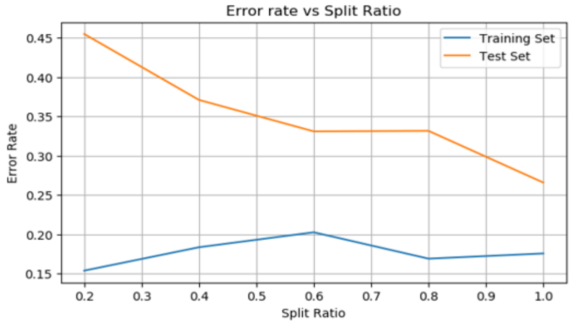


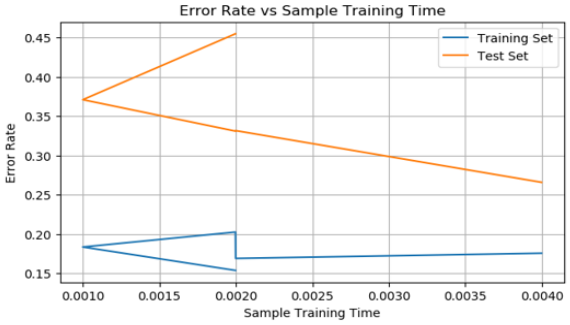


From the above plot with increase in the number of neighbors, the error rate increases. Taking 80 percentage of accuracy into consideration, we get 3 as an ideal number of neighbors for Adult dataset and the same will be used for the upcoming experiments.

Experiment on the Sample Size and Clock Time: -

Now we have seen that for given distance metric (Euclidean) and for the given number of Neighbors (3 clusters) for Adult dataset, we are going to perform an experiment to estimate the error of the implemented model with different sample sizes and the clock time for training the data. The following plot was obtained for training and test dataset for different sample sizes plotted against each of its error rate:





The above plot, when compared with the Energy dataset, looks a little inconsistent. But as we know that the performance of the model implemented depends on the dataset. The above plot's error rate varies with the different sample training sizes and clock times. The training accuracy for the full size of the Adult training dataset was determined to be 82.4%.

Artificial Neural Networks

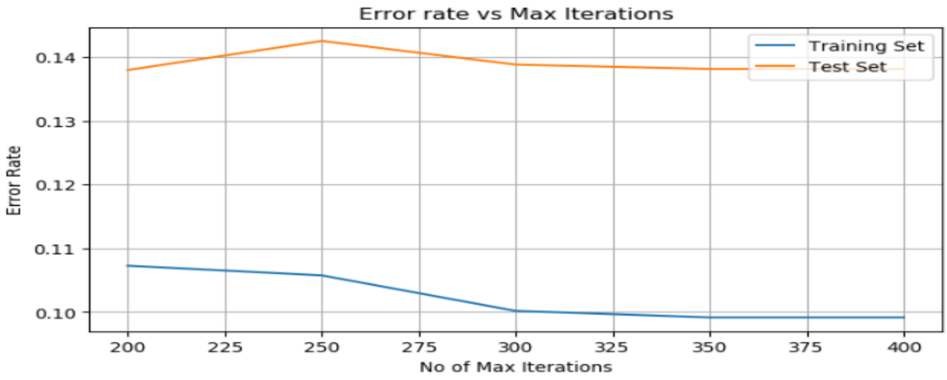
For implementing Artificial Neural Networks "MLPClassifier" package was downloaded from Sci-kit Learn.

Artificial Neural Networks using MLPClassifier was implemented and the following experiments were performed:

Experiments on Dataset 1 (Energy Dataset): -

Experiment with max_iter variable in MLPClassifier:

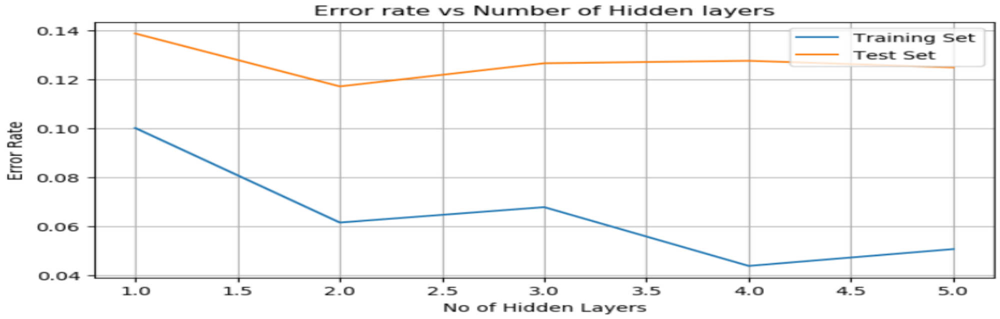
To begin implementing ANN using MLPClassifier, an optimal value for the parameter max_iter was required for which this experiment was done. Max_iter is the maximum number of iterations. The solver iterates until convergence (determined by 'tol') or this number of iterations. The default value for this is 200. However, error rate was calculated for various values of this variable and the following plot was obtained to determine the optimal value of max_iter variable:



From the above graph it can be inferred that the error rate is decreasing with increase in the number of iterations for the training dataset. The minimum error in the above graph is obtained at 300 iterations where the convergence has occurred which will be considered for further analysis

Experiment with Number of Hidden Layers:

Another important parameter here in MLPClassifier is the number of hidden layers for the ANN model built. For this, after setting the maximum number of iterations to 300, an experiment was conducted by varying the number of hidden layers in the model and this was plotted against the error rate for each hidden layer built and the following plot was obtained:

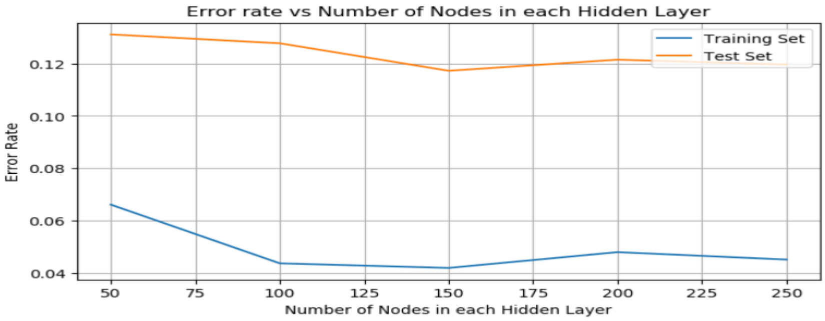


In the above plot the minimum error rate was determined at 4 hidden layers for the training data set. Hence, for the given dataset, the number of hidden layers will be taken as 4 for further analysis

Experiment on the Number of Neurons in each hidden layer: -

Now that we have determined the optimal number of hidden layers in the neural network to be 4 with the given dataset, we must determine the number of neurons in each of these hidden layers.

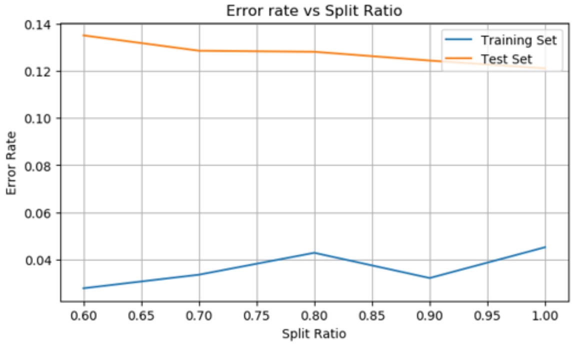
For this an experiment was conducted with different values of nodes that will be parsed to MLPClassifier to determine the optimal number of neurons in each hidden layer with minimal error rate and the following plot was obtained:

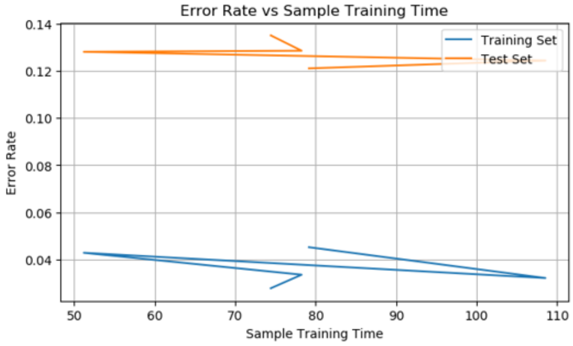


From the above plot we have determined the minimal error rate for 150 nodes in each hidden layer of the Neural Network which can be considered as optimal for conducting further experiments.

Experiment on the Length of the sample and the clock time: -

With the calculated values of maximum iterations (300), Number of Hidden Layers in the Neural Network (4) and the number of neurons in each hidden layer (150), an experiment was conducted to determine different error rates with different sample sizes and the training time for each of these samples and the following plot was obtained:



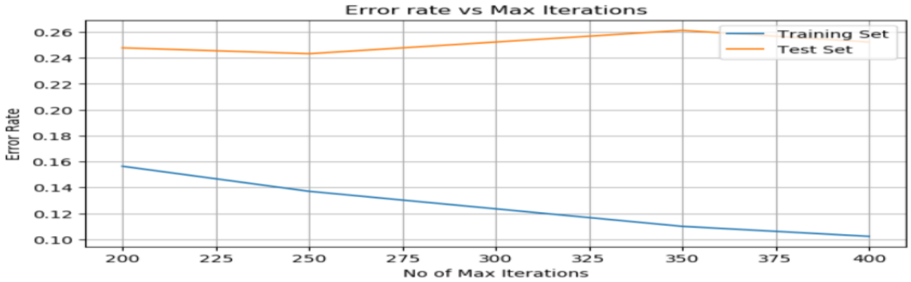


The above two graph shows the error rate plotted against each of the sample sizes and the clock time taken for training these samples. For different sample sizes, it can be seen that the error does not exceed 4% for the training dataset. For the full sample size of the training data, the accuracy is around 95.59% for the implemented ANN model.

Experiments on Dataset 2 (Adult Dataset): -

Experiment with max_iter variable in MLPClassifier:
For the adult dataset, an optimal value for the parameter max_iter was required for which this experiment was done. Max_iter is the maximum number of iterations. The solver iterates until convergence (determined by 'tol') or this number of iterations. The default value for this is 200.

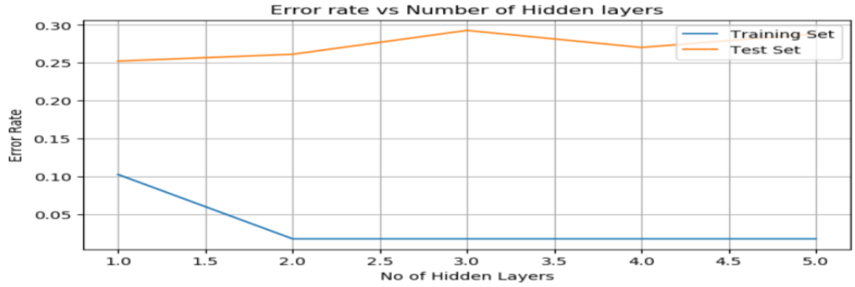
However, error rate was calculated for various values of this variable and the following plot was obtained to determine the optimal value of max_iter variable:



From the above graph it can be inferred that the error rate is decreasing with increase in the number of iterations for the training dataset. The minimum error in the above graph is 400 iterations where the convergence has occurred. Therefore max_iter variable is set to 400 for further experiments.

Experiment with Number of Hidden Layers:

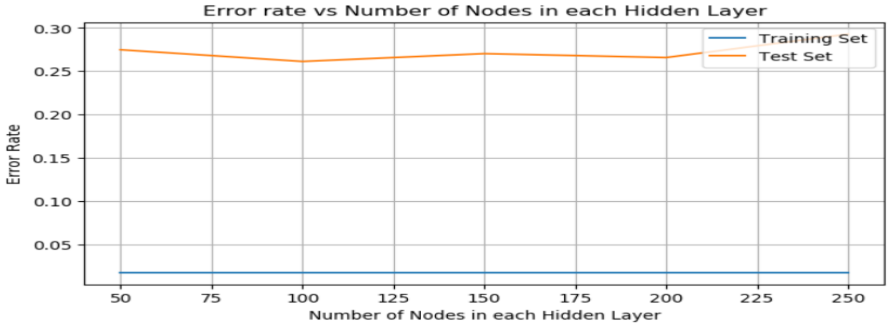
For this, after setting the maximum number of iterations to 400, an experiment was conducted by varying the number of hidden layers in the model and this was plotted against the error rate for each hidden layers built and the following plot was obtained:



In the above plot the minimum error rate was determined at 2 hidden layers for the training data set. Hence, for the given dataset, the number of hidden layers will be taken as 2 for further analysis.

Experiment on the Number of Neurons in each hidden layer: -

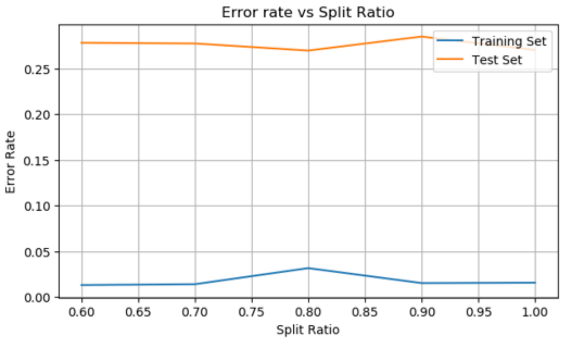
Now that we have determined the optimal number of hidden layers in the neural network to be 2 with the given dataset, we must determine the number of neurons in each of these hidden layers. For this an experiment was conducted with different values of nodes that will be parsed to MLPClassifier to determine the optimal number of neurons in each hidden layer with minimal error rate and the following plot was obtained:

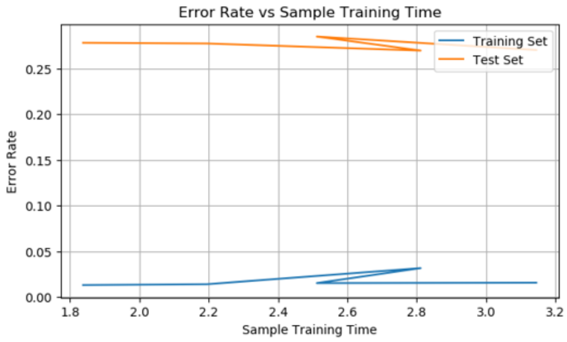


From the above plot it can be understood from the error plot of training data that the error rate almost remains constant for the given number of nodes. Hence, we will consider the default value of the number of nodes, which is 100, for further analysis.

Experiment on the Length of the sample and the clock time: -

With the calculated values of maximum iterations (400), Number of Hidden Layers in the Neural Network (2) and the number of neurons in each hidden layer (100), an experiment was conducted to determine different error rates with different sample sizes and the training time for each of these samples and the following plot was obtained:



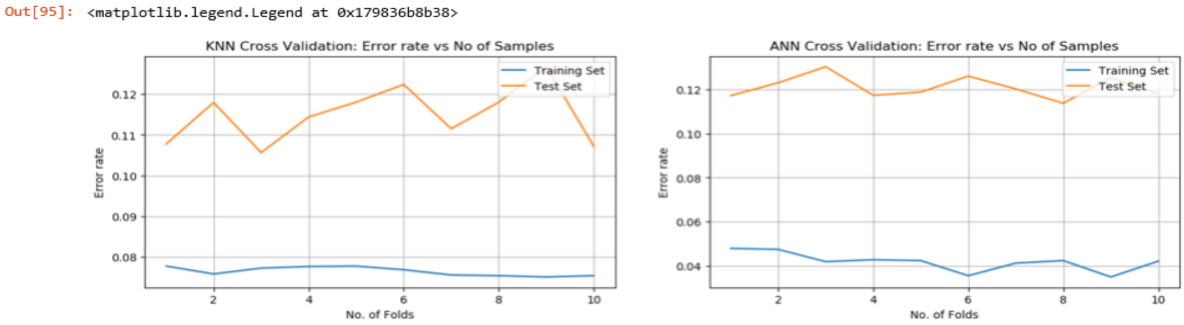


The above two graph shows the error rate plotted against each of the sample sizes and the clock time taken for training these samples. For all the values of sample sizes taken into consideration, the error is not exceeding 0.04. For the entire sample size of the training data the accuracy is estimated to be around 98% for the implemented ANN model.

Cross Validation of KNN and ANN models for dataset1 and dataset2

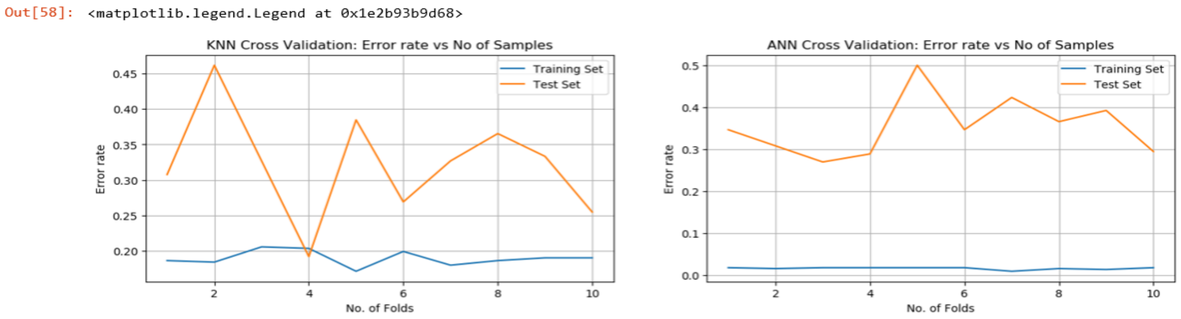
Now that we have implemented the best KNN and ANN models implemented for dataset 1 and dataset 2 by performing various experiments and parsing best values for the respective parameters, lets do Cross Validation for the given datasets with the implemented models. The package that is used for Cross Validation is "cross_validate" imported from sklearn.model selection. Cross Validation was implemented and the following plots for the errors was obtained:

Cross Validation for Dataset 1 (Left-side graph for KNN and right-side for ANN):-



```
Out[95]: <matplotlib.legend.Legend at 0x179836b8b38>
```

Cross Validation for Dataset 2 (Left-side graph for KNN and right-side for ANN):-



```
Out[58]: <matplotlib.legend.Legend at 0x1e2b93b9d68>
```

The above Cross Validation was performed with an ideal value of k=10. For dataset 1, it can be seen that the error rate is not exceeding 6% for KNN model, while with the same dataset the error rate is not exceeding 5% for ANN model. For dataset 2, we can see that the error rate is not exceeding 20% for the implemented KNN model, while with the same dataset the error rate is not exceeding 2% for the implemented ANN model.

Model Comparison, Interpretation and Conclusion

In this assignment, KNN and ANN models were implemented for the two datasets: Energy dataset (dataset 1) and Adult dataset (dataset 2). Accuracy of the model well describes how good a model is. The best models developed, that were obtained by conducting various experiments on the given dataset, using which the following table with the respective accuracies was formed: -

Training Accuracy	KNN	ANN	SVM (rbf)	Decision Tree	Boosting
Dataset 1(Energy)	92.63	95.59	85.34	86.01	97.95
Dataset 2(Adult)	82.4	98	95.55	79.34	98.45

Best model within KNN and ANN models for dataset1 and dataset2:

From the above table, for dataset 1, it can be seen that the accuracy for ANN is 95.59 which is better when compared to the accuracy of KNN model for dataset 1.

For dataset 2, the accuracy for ANN is 98 which is far better when compared to KNN model for dataset 2 Therefore in general, ANN model performs better when compared KNN model for both dataset 1 and dataset 2.

The best model as per the experiments that was conducted in this assignment is the ANN model for dataset 2 with a training accuracy of 98 when compared to the accuracy of other models in this assignment.

Best model within SVM, Decision Tree, Boosting, KNN and ANN models for dataset1 and dataset2:

From the above table, for dataset 1, it is clear that the accuracy for Boosting model has the highest accuracy of 97.95 when compared to the accuracies of the other models for dataset1 . Therefore, this model performs the best as per the implementation for dataset 1.

For dataset 2, the accuracy for the Boosting model is the highest with an accuracy of 98.45 when compared to the accuracies of other models in the dataset 2.

Overall taking into account the accuracies of the five models that was implemented (KNN, ANN, SVM, Decision Tree and Boosting) for dataset1 and dataset2, the best model is estimated as Boosting with an Accuracy of 98.45 for dataset 2.

Additional things that can be done to get better results:

As we know that the accuracy of the model depends on the data that we chose for analysis, still the following steps can be considered in order to get some better results:

Bharath Kumar Natarajan (BXN180005)
6341.003 – Applied Machine Learning (Assignment 3)

1. Firstly, by build simple models and using many independent variables need not necessarily mean that your model is good. Next step is to try and build many regression models with different combination of variables. Then you can take an ensemble of all these models.

2. To understand the relationship between dependent variable and all the independent variables and whether they have a linear trend. Only then we can afford to use them in the model to get a good output.

3. It's also important to check and treat the extreme values or outliers in your variables. This could be one reason why your predicted estimate values might vary as they are getting skewed by the outlier values.

Dataset Description and Data Pre-processing:

Appliances Energy Prediction dataset (Energy Dataset):

• The dataset is downloaded from UCI Machine Learning repository. Here is the link to download the dataset: https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

• The Dataset consists of 19735 observations on 29 variables.

• None of the column contains any missing value, so no missing value imputation is required.

The target column for the Appliances Energy Prediction dataset is “Appliances”. The Appliances column here is a continuous variable containing values ranging from 10Wh to 1080Wh with e median value of 60Wh.

Since our objective is to implement the classification learning algorithms that the problem statement requires the target variable needs to be a binary categorical variable, our target variable has been converted to a binary categorical variable with a threshold of its median values 60Wh. The column “date” has been removed from the input variables as this column is of no use for our analysis.

Absenteeism at Work dataset (Adult Dataset):

• The dataset is downloaded from UCI Machine Learning repository. Here is the link to download the dataset: https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work

• The Dataset consists of 740 observations on 21 variables.

• None of the column contains any missing value, so no missing value imputation is required.

The target column for the Absenteeism at Work dataset is “Absenteeism time in hours”. The Absenteeism time in hours column here is a continuous variable containing values ranging from 0 hours to 120 hours with a median value of 3 hours.

Since our objective is to implement the classification learning algorithms and that the problem statement requires the target variable needs to be a binary categorical variable, our target variable has been converted to a binary categorical variable with a threshold of its median values 3 hours. The column ID has been removed from the dataset since this column is not required for analysis. Both the datasets were partitioned into training and test datasets with split percentage of 70 as training data and 30 as testing data.