Assignment 3

Evan Jones Boddu, ejb180003@utdallas.edu

In this report, I have implemented two of the learning algorithms which are Artificial Neural Networks (ANN) and K Nearest Neighbors (KNN) on two distinct datasets. The main goal is to use the classification dataset and experiment with the number of layers and the number of nodes for the neural networks and use different activation functions such as sigmoid, tanh, etc. Similarly, I have performed classification on KNN and have experimented with the number of neighbors as well as different distance metrics. The KNN algorithm is performed using Sckitlearn library in python language (Jupyter notebook). For implementing ANN, Keras library and TensorFlow are used.

Dataset 1: Appliances Energy Prediction Dataset

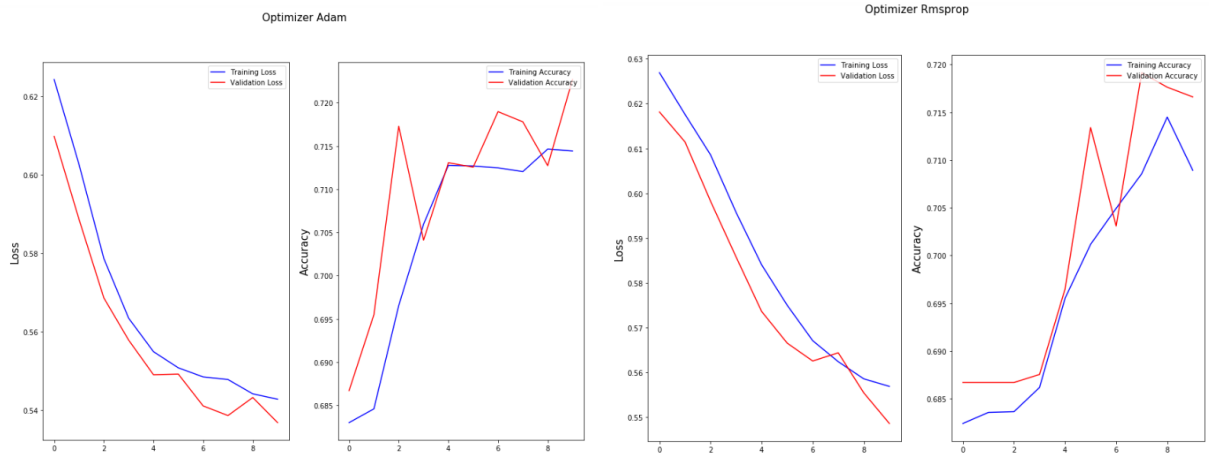
The dataset that has been used in this assignment is taken from the previous assignment (which has been used in assignment 1 as well as assignment 2) and the link to the dataset can be found here. Since the dataset is familiar, I will go ahead and show the implementations and the plots that have been performed using this dataset. The dataset is not a classification dataset, it has a continuous variable as its target variable, hence the following preliminary steps have been performed on the dataset before building and running the models:

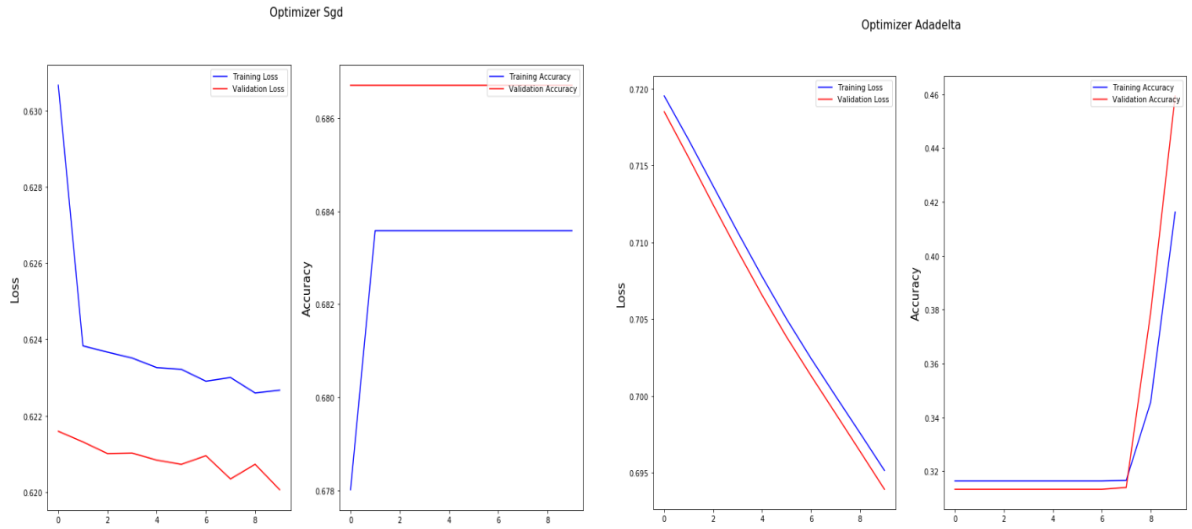
- Converted the target variable (“Appliances”), which is a continuous variable into a binary classification problem with values 0 and 1.
- I have put a threshold for the feature Appliances to convert it into a binary classification model. If Appliances is greater than 85, then it is considered as 1 or else it is considered as 0.
- Split the dataset into 70:30 ratio.
- Trained the dataset on different learning algorithms using different optimizers and picked the best optimizer.

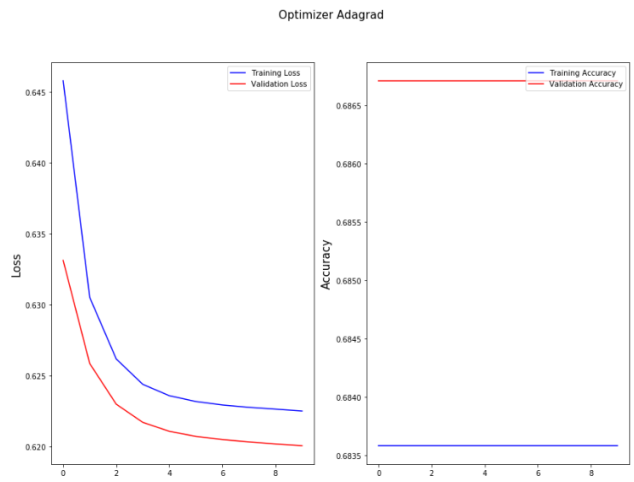
Furthermore, I have selected all the features from the dataset so that the data will as raw as possible and to get the best results when the dataset has been tweaked very little. I have assigned the target to ‘y’ and all the other independent variables to ‘x’. The dataset consists of 19735 observations and 28 features.

Artificial Neural Network (ANN)

ANN is implemented with various activation functions such as Adam, RMSProp, SGD, AdaDelta and AdaGrad. I have also tested for accuracy and losses for training set and validation set as well. A plot has been created in order to understand the performance of the model using different optimizers when made run in iterations. The following plots are the results that have been obtained for the optimizers that have been mentioned above.



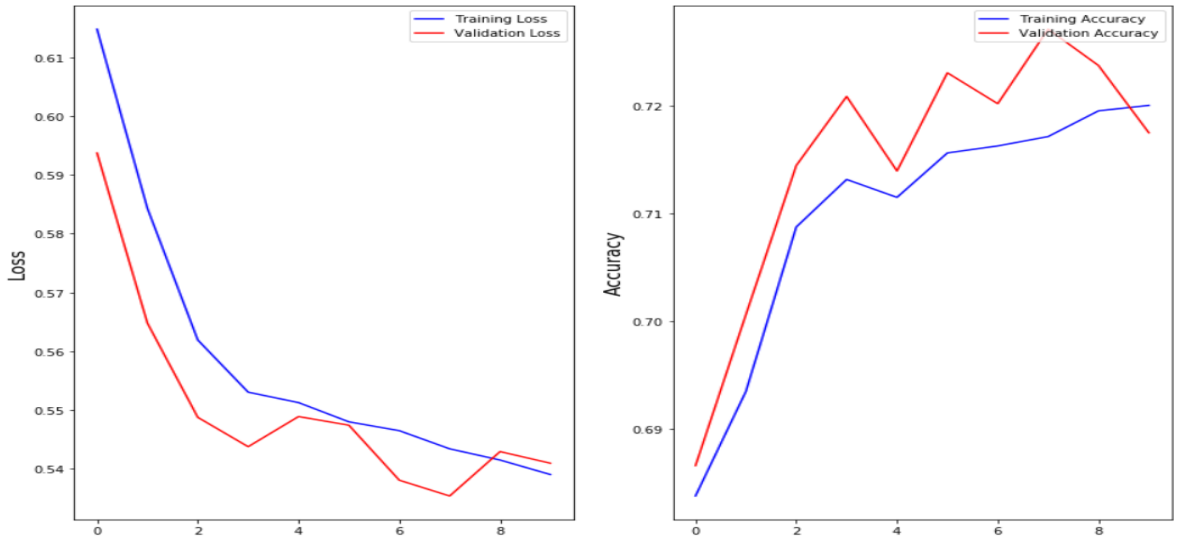




From the plots above, we can depict that there is no effect of increasing the nodes in the hidden layers and hence the results are not shown. The best model performance we got is by using one dense layer.

The plots show the training and validation losses as I mentioned before. Among all the optimizers that have been used, the optimizer Adam provided with convincing results on our dataset with an overall accuracy of 71.15% with a confusion matrix of ([3749, 317],[1391, 464]).

Similarly, I have also used different activation tanh for the neural net model using Adam optimizer. The following results have been observed:



We can observed from the above plots that as the number of iterations increase the loss kept on decreasing and the accuracy kept on increasing (until iteration 6 and then it started to fall down, this is when the model started to memorize the values and not learning. Hence, as the number of iterations increase after 6 the validation accuracy falls). For this model, I have used Adam optimizer using tanh activation which resulted in an accuracy of 71.75% with a confusion matrix of ([3844, 222],[1451, 404]).

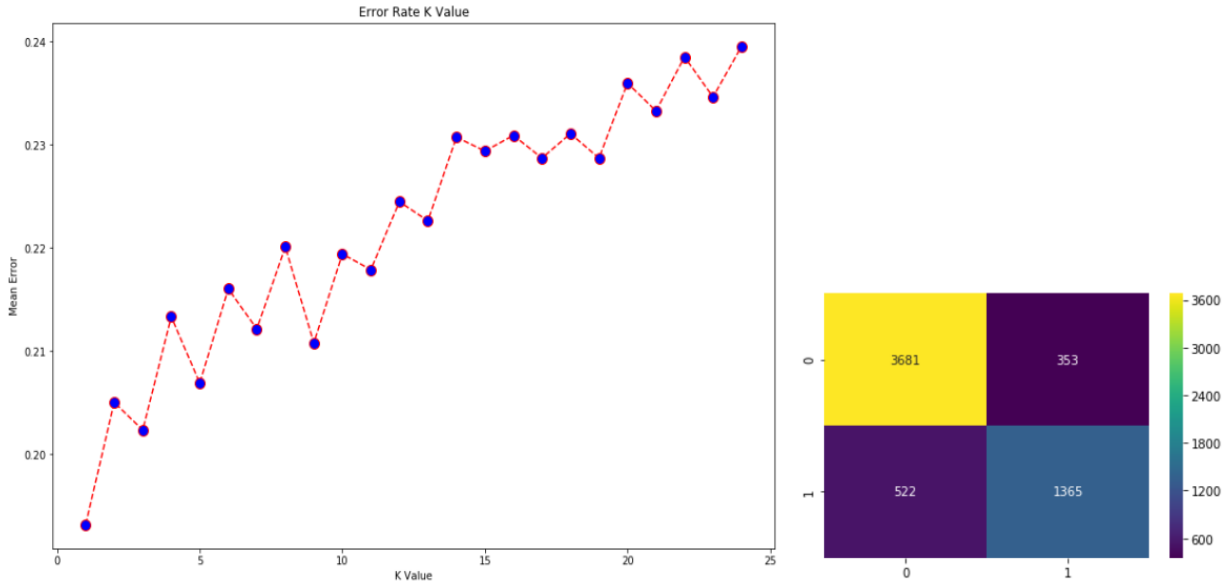
Accuracy for the performance of different algorithms:

SUPERVISED LEARNING ALGORITHMS	ACCURACY (%)
Linear SVM (C = 1)	79.48%
Decision Tree (Depth = 15)	86.37%
AdaBoost (Depth = 15)	91.95%
KNN (Hamming)	73.62%
ANN (tanh)	53.85%

K Nearest Neighbors (KNN)

The K Nearest Neighbors have been implemented over different values of K to check which number works best for the model and to understand the effect of change of neighbors on the dataset. I have also used different distance metrics to understand how the model reacts to different distance parameters.

To begin with, various values of K, from 1 to 25 have been used and the error values have been recorded and plotted. The main inference that we can make out of the plot that is shown below is that as the number of K values increase, the error mean also increases significantly. This means that the model's accuracy is varying a lot for changing values of K. From the plot, we can see that there are two values of K for which the error is minimum, which are K = 1 and K = 3. Hence, we will choose K = 3 for predicting the testing set because going with K = 1 is not a good choice. After training the model with K = 3, the model gave an accuracy of 85.22 % and I have also plotted a heatmap for the confusion matrix.

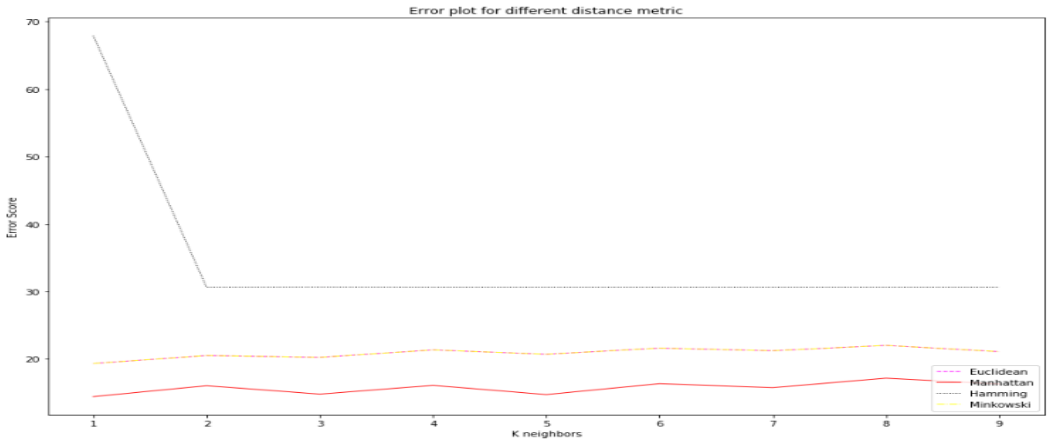


So initially, I have checked the errors for different values of K in order to find the best value of K for the dataset. From the above plot, we can see that K = 1 and K = 3 provide the least values of error. We can take K = 1 as it is not a good choice and it is likely to rely on one input to decide the classification. Next, I have made a KNN model which uses different distance parameters in order to find out which distance parameter gives the best results. The different parameters that I have used are Euclidean, Manhattan, Hamming and Minkowski. From the results, it is deduced that Manhattan gives the best results for the dataset.

I have created another KNN model which the best value of K (which is K =3) and the best distance parameter, which is Manhattan, which resulted in an accuracy of 85.22% and also the heatmap which provides us information about the confusion matrix.

Comparison with different distance parameters:

A plot which compare different distance parameter has been made and from the plot we can say that the Manhattan distance provides least error for increasing values of K.

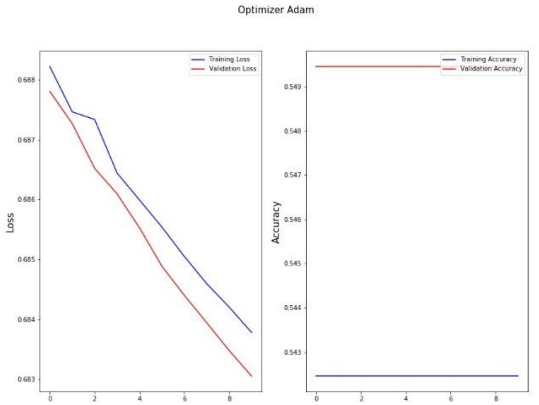


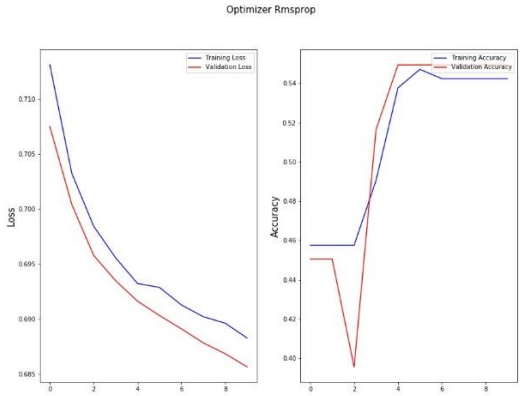
Dataset 2: Heart Disease Dataset

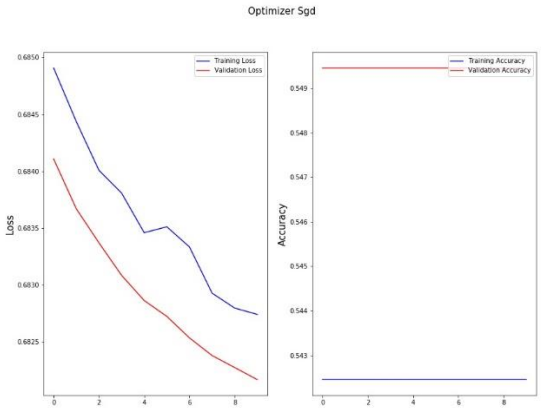
This dataset consists of 76 attributes, but all published experiments refer to using a subset of 14 features. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The goal field refers to the presence of heart disease in the patient. You can find the link to this dataset here.

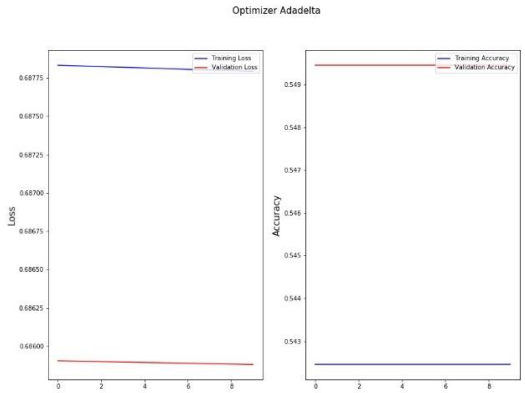
Artificial Neural Networks (ANN)

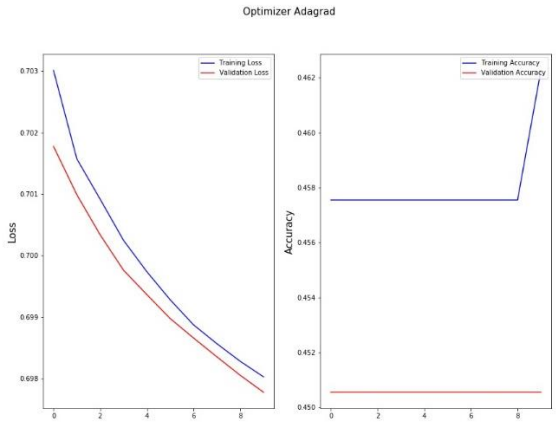
Neural Net is implemented on the second dataset which is Heart Disease Dataset and the same plots have been made for different optimizers. This is done in order to observe the performance of the model using different dataset. I have used the same set of optimizers that are used in dataset 1 (which is Appliances Energy Prediction Dataset) and the plots for different optimizers using a neural network model is shown below:





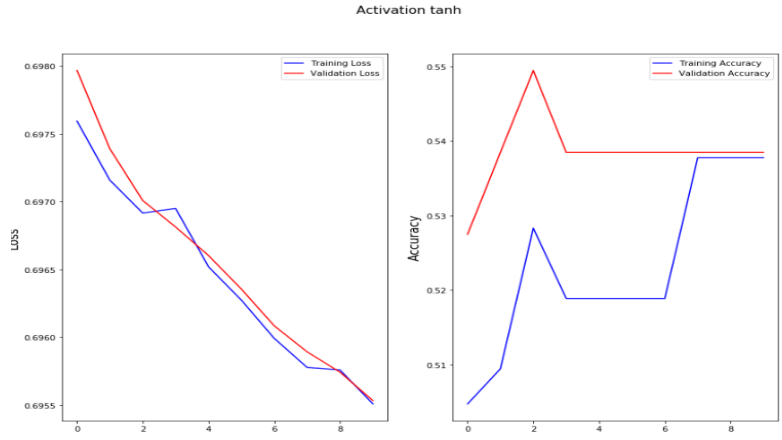






From the plots above, we can depict that there is no effect of increasing the nodes in the hidden layers and hence the results are not shown. The best model performance we got is by using one dense layer. The plots show the training and validation losses as I mentioned before. Among all the optimizers that have been used, the optimizer Adam provided with convincing results on our dataset with an overall accuracy of 51.65% with a confusion matrix of ([34, 7],[37, 13]).

Similarly, I have also used different activation tanh for the neural net model using Adam optimizer. The following results have been observed:



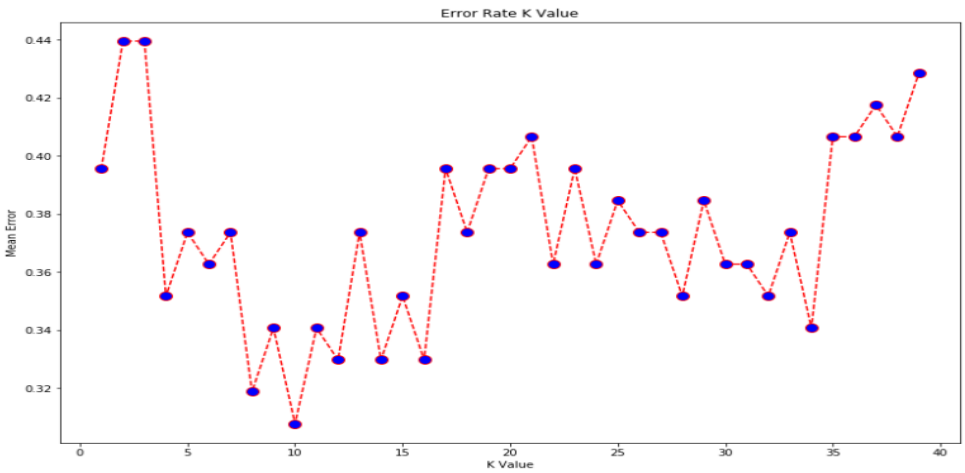
From the above plot, we can notice that as the number of iterations increases, the loss decreases, and the accuracy increases till 3rd iteration and then becomes saturate when the accuracy hits 0.54. The accuracy for this activation is 53.85% with a confusion matrix of ([0, 41],[1, 49]).

Accuracy for the performance of different algorithms:

SUPERVISED LEARNING ALGORITHMS	ACCURACY (%)
Linear SVM (rbf, C = 1)	59.34%
Decision Tree (Depth = 5)	76.93%
AdaBoost (Depth = 5)	75.82%
KNN (Manhattan)	85.22%
ANN (tanh)	71.75%

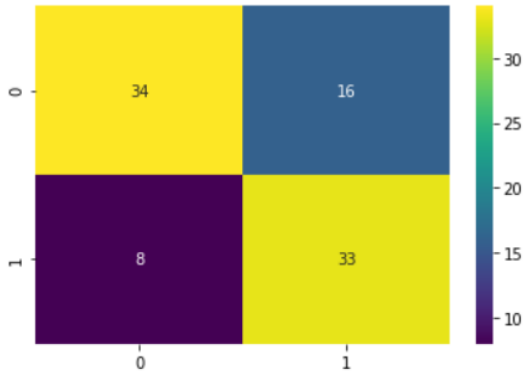
K Nearest Neighbors (KNN)

Initially, calculated the error for different values of K in order to find the best value for K under which the model performs well. So, I ran the KNN model for different values of K ranging from 1 to 40. From the plot below we can see that for K = 10, the mean error is less than 0.32.



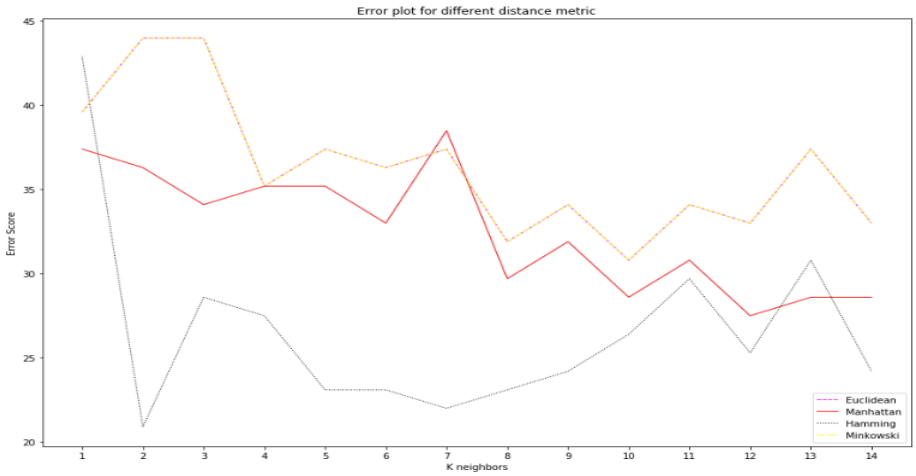
Next, I have created a KNN model with different distance parameters to find the best parameter to use for the model. The different types of distance parameters are Euclidean, Manhattan, Hamming and Minkowski. After running the model through the distance parameters and for different values of K from 1 to 15 (because the best value of K we found from the error plot to be 10), we found out that Hamming provides the best values when taken as the distance parameter.

After using the value of K = 10 and the distance metric as Hamming, I ran another KNN model in order to predict the accuracy as in how good it is performing with the given dataset, and the accuracy of the KNN model with the above said parameters is 73.62%. I have also plotted a heatmap of confusion matrix and it is shown below.



Comparison of different distance parameters:

From the plot below, we can see the error plots for different distance metrics. We can infer that Hamming provides the best results for the Heart disease dataset.



Conclusion:

We conclude that the models KNN and ANN perform must better on these datasets than the previous ones. This assignment has made me think on datasets from a broader aspect regarding data handling, feature selection and thinking from different point of views. Overall, it was a great practice to work on this report. Both the datasets were very diverse in terms of the information provided and it was really challenging to work on both datasets .