Assignment 3: KNN, ANN

Dataset – Average GPU runtime:

The dataset (SGEMM GPU kernel performance) dataset can be downloaded at:
https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance#
There are 14 parameters. The first 4 are ordinal and the last four variables are binary. The dataset has total 241600 data entries and 18 features with the last four being the runtime measurement.

KNN - Experiment 1: Different K values

It simply calculates the distance of a new data point to all other training data points. It then selects the K-nearest data points, where K can be any integer. Finally, it assigns the data point to the class to which most of the K data points belong.

K-value: 3
Train accuracy: 91.82%
Test accuracy: 83.74%

Classification Report

	precision	recall	f1-score	support
0	0.83	0.86	0.84	12234
1	0.85	0.82	0.83	11961
accuracy			0.84	24195
macro avg	0.84	0.84	0.84	24195
weighted avg	0.84	0.84	0.84	24195

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)

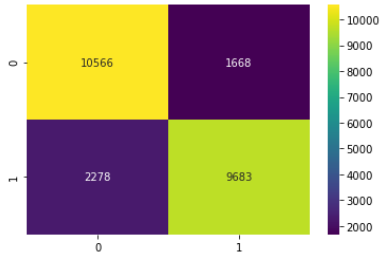
K-value: 5
Train accuracy: 89.6%
Test accuracy: 83.69%

Classification Report

	precision	recall	f1-score	support
0	0.82	0.86	0.84	12234
1	0.85	0.81	0.83	11961
accuracy			0.84	24195
macro avg	0.84	0.84	0.84	24195
weighted avg	0.84	0.84	0.84	24195

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



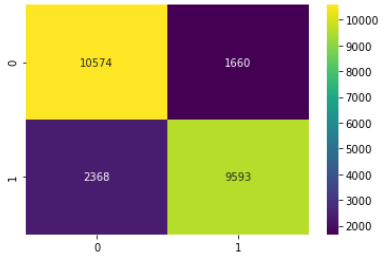
K-value: 9
Train accuracy: 87.46%
Test accuracy: 83.35%

Classification Report

	precision	recall	f1-score	support
0	0.82	0.86	0.84	12234
1	0.85	0.80	0.83	11961
accuracy			0.83	24195
macro avg	0.83	0.83	0.83	24195
weighted avg	0.83	0.83	0.83	24195

Confusion matrix

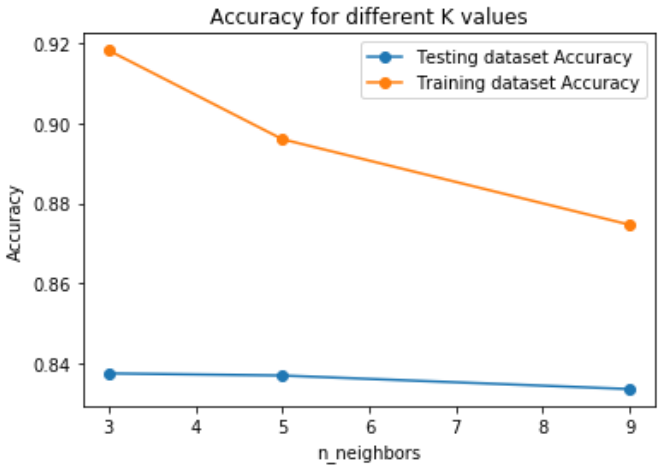
AxesSubplot(0.125,0.125;0.62x0.755)



Accuracy for different K values:

The above experiment with different K-values shows that with the k-value = 3 neighbors gives the best test data accuracy compared to 5 and 9 neighbors. K value = 3 also gives a better accuracy for the training set.

As the training dataset gives better result for k-value = 3, we are going to use this value for our next experiment for different distance metric.



KNN - Experiment 2: Different distance metric

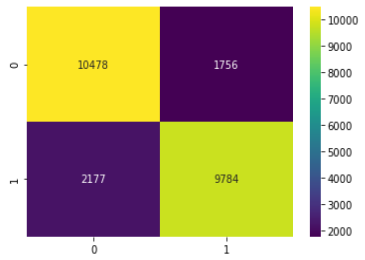
K-value: 3
Distance metric: Euclidean
Train accuracy: 91.82%
Test accuracy: 83.74%

Classification Report

	precision	recall	f1-score	support
0	0.83	0.86	0.84	12234
1	0.85	0.82	0.83	11961
accuracy			0.84	24195
macro avg	0.84	0.84	0.84	24195
weighted avg	0.84	0.84	0.84	24195

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



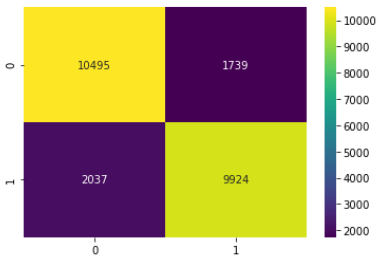
K-value: 3
Distance metric: Manhattan
Train accuracy: 92.32%
Test accuracy: 84.39%

Classification Report

	precision	recall	f1-score	support
0	0.84	0.86	0.85	12234
1	0.85	0.83	0.84	11961
accuracy			0.84	24195
macro avg	0.84	0.84	0.84	24195
weighted avg	0.84	0.84	0.84	24195

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



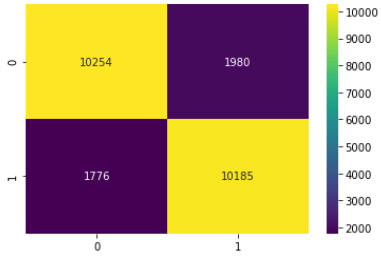
K-value: 3
Distance metric: Hamming
Train accuracy: 92.49%
Test accuracy: 84.47%

Classification Report

	precision	recall	f1-score	support
0	0.85	0.84	0.85	12234
1	0.84	0.85	0.84	11961
accuracy			0.84	24195
macro avg	0.84	0.84	0.84	24195
weighted avg	0.84	0.84	0.84	24195

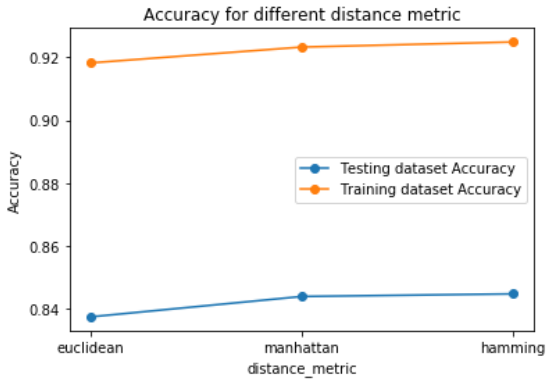
Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



Accuracy for different distance metric:
We can conclude from the above experiment that, with K value of 3 with distance metric 'Hamming' gives the best test data and train data accuracy compared to other distance metric (Euclidean, Manhattan).

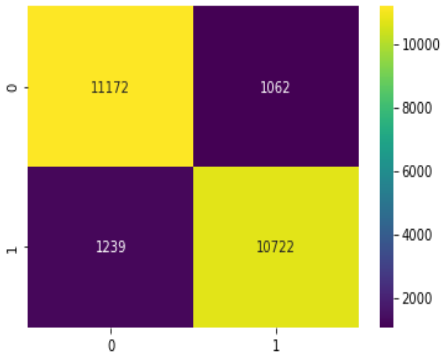
We do notice there is not much difference between the training and test data set for manhattan and hamming distance metric. But with closer analysis we select hamming as parameter which performs best with k-value = 3 for this dataset.



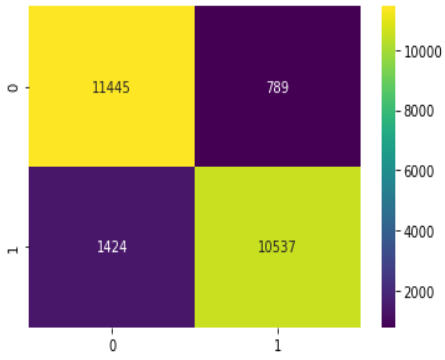
.

ANN - Experiment 1: Different number of layers

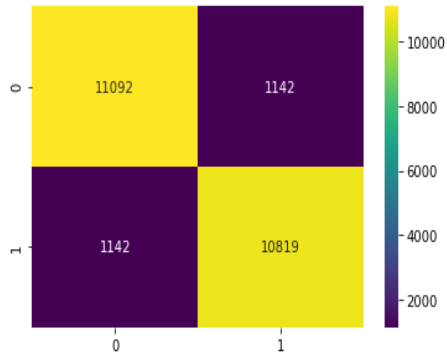
Layer: 3
Train accuracy: 90.48%
Test accuracy: 90.94%



Layer: 4
Train accuracy: 90.85%
Test accuracy: 91.27%

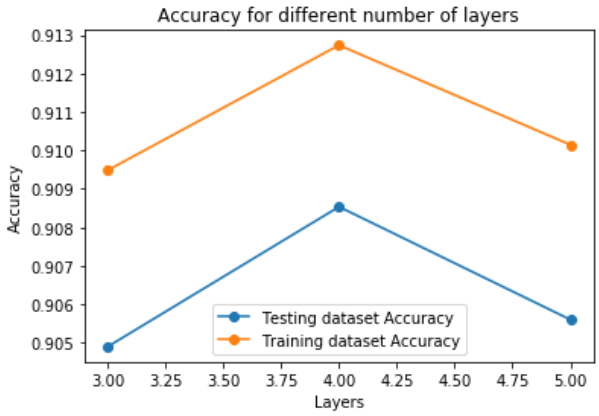


Layer: 5
Train accuracy: 90.56%
Test accuracy: 91.01%



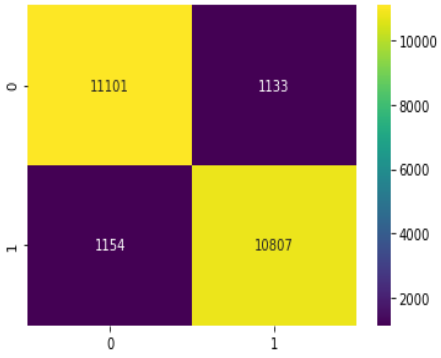
Accuracy for different number of layers:

From the above layers, we can see that both the train and the test accuracy have significant difference between layers = 4 and layers = 3 and 5. Therefor by looking at the train and test accuracy we can conclude that it performs best when using 4 layers.

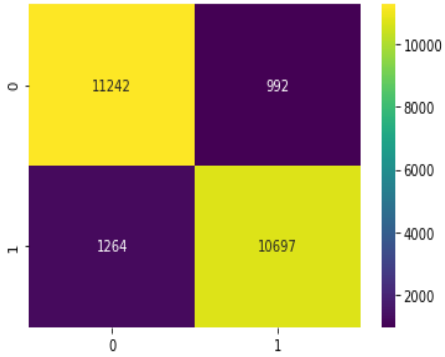


ANN - Experiment 2: Different activation functions:

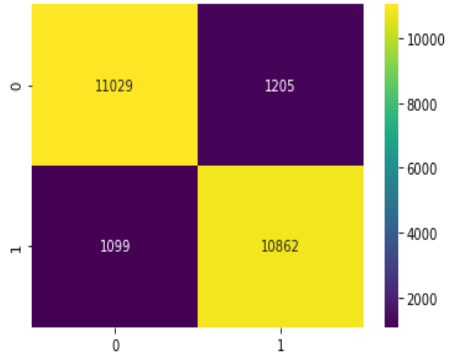
Activation fn1: 'relu', 'relu', 'relu', 'sigmoid'
Train accuracy: 90.54%
Test accuracy: 91.06%



Activation fn2: 'relu', 'tanh', 'tanh' 'sigmoid'
Train accuracy: 90.49%
Test accuracy: 91.00%



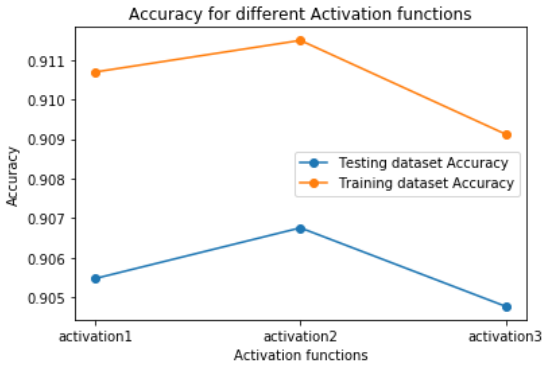
Activation fn3: 'tanh', 'relu', 'sigmoid', 'sigmoid'
Train accuracy: 90.47%
Test accuracy: 90.91%



Accuracy for different activation function:

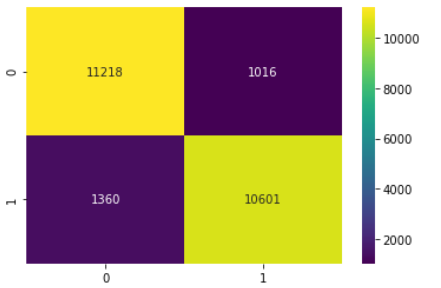
Combining both the outputs of different number of layers and different activation functions, we can see that 4 layers work best with one relu layers, two tanh layer and one sigmoid layer.

The activation set that gives the best result is activation set 2 which is one relu layers, two tanh layer and one sigmoid layer. Hence we will use that model to vary number of neurons for our next experiment.



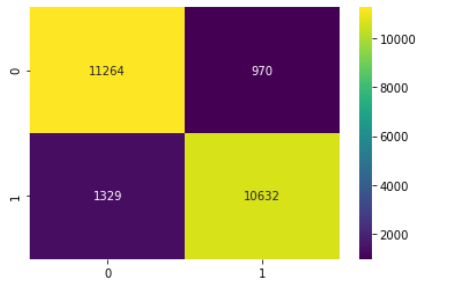
ANN - Experiment 3: Different number of nodes

Node set1: [10, 10, 7, 1]
Train accuracy: 90.17%
Test accuracy: 90.82%

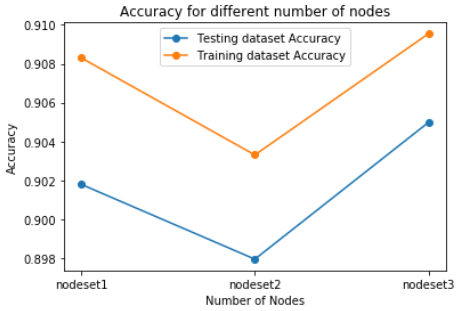


Node set2: [10, 8, 10, 1]
Train accuracy: 89.79%
Test accuracy: 90.33%

Node set3: [14, 7, 5, 1]
Train accuracy: 90.49%
Test accuracy: 90.95%



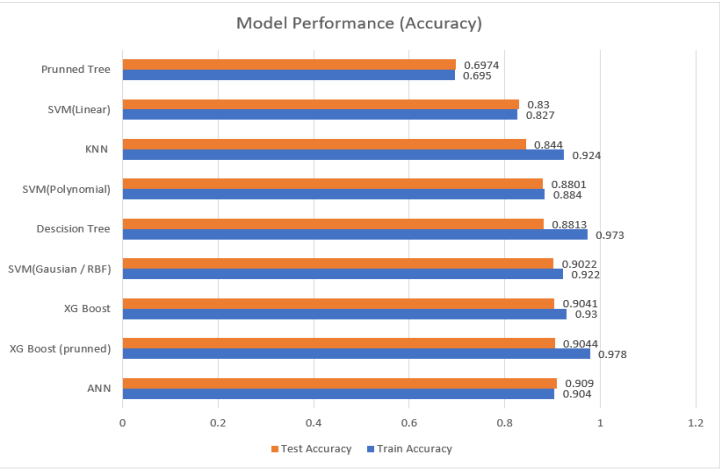
Accuracy of different number of nodes:
For both training and testing datasets, graph shows that using node set3 [14, 7, 5, 1] gives the best accuracy.

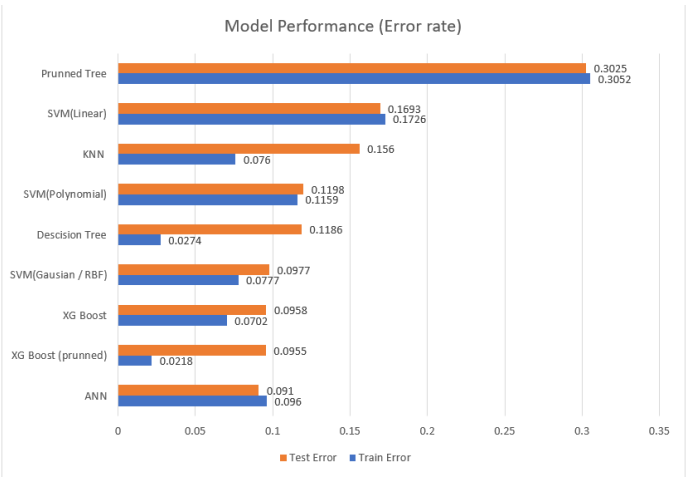


Hence for ANN 4 layers with different activation function [relu, tanh, tanh, sigmoid] with node set [14, 7, 5, 1] gives the best accuracy.

Model Performance:

Rank	Algorithm	Train Accuracy	Test Accuracy	Train Error	Test Error
1	ANN	0.904	0.909	0.096	0.091
2	XG Boost (prunned)	0.978	0.9044	0.0218	0.0955
3	XG Boost	0.93	0.9041	0.0702	0.0958
4	SVM(Gausian / RBF)	0.922	0.9022	0.0777	0.0977
5	Descision Tree	0.973	0.8813	0.0274	0.1186
6	SVM(Polynomial)	0.884	0.8801	0.1159	0.1198
7	KNN	0.924	0.844	0.076	0.156
8	SVM(Linear)	0.827	0.83	0.1726	0.1693
9	Prunned Tree	0.695	0.6974	0.3052	0.3025





Interpretation of the results:

- The models are ranked based on the test accuracy.
- If we compare ANN and KNN then, based on both the test and train accuracy ANN performs better than KNN.
- ANN performs the best and ranks 1st on the test data set and the train data set, than any of the previous algorithms used in assignment 2.
- KNN ranks 7th based on the accuracy on the test dataset.

Dataset – Rains in Australia:

The dataset is obtained from Kaggle. The following is the link to the dataset. https://www.kaggle.com/jsphyg/weather-dataset-rattle-package. The weather dataset contains 142,193 daily weather observations from 49 weather stations across Australia over the period November 2007 to June 2017 and with 24 features such as Rain tomorrow, min Temp, max Temp etc.

KNN – Experiment 1: Different K values

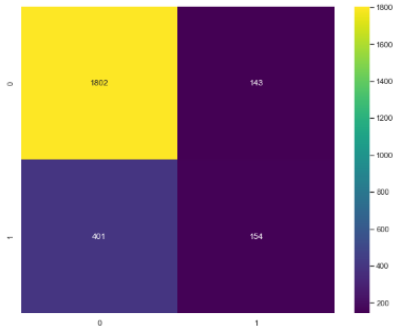
For K=3 Train accuracy– 87.30%, Test accuracy– 78.24%

Classification Report

	precision	recall	f1-score	support
0.0	0.82	0.93	0.87	1945
1.0	0.52	0.28	0.36	555
accuracy			0.78	2500
macro avg	0.67	0.60	0.62	2500
weighted avg	0.75	0.78	0.76	2500

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



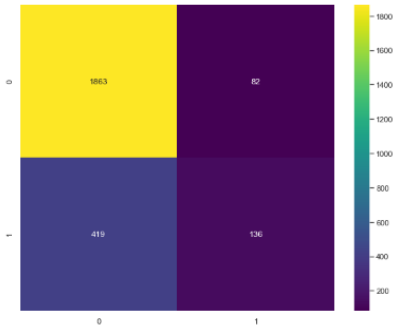
For K=5 Train accuracy– 84.48%, Test accuracy- 79.96%

Classification Report

	precision	recall	f1-score	support
0.0	0.82	0.96	0.88	1945
1.0	0.62	0.25	0.35	555
accuracy			0.80	2500
macro avg	0.72	0.60	0.62	2500
weighted avg	0.77	0.80	0.76	2500

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



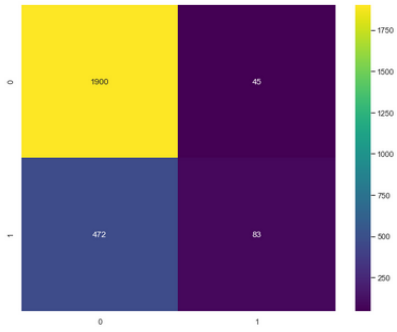
For K= 9 Train accuracy- 82.57%, Test accuracy- 79.32%

Classification Report

	precision	recall	f1-score	support
0.0	0.80	0.98	0.88	1945
1.0	0.65	0.15	0.24	555
accuracy			0.79	2500
macro avg	0.72	0.56	0.56	2500
weighted avg	0.77	0.79	0.74	2500

Confusion matrix

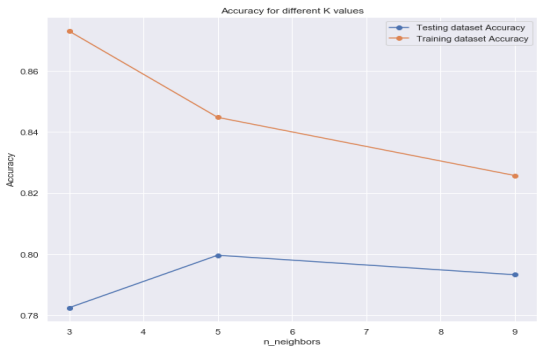
AxesSubplot(0.125,0.125;0.62x0.755)



Accuracy for different K values:

The above experiment with different K values show that for value of K = 3 gives best train accuracy but with k value = 5 gives the best test accuracy.

So, we can use k=5 for the next experiments. Both the training and the test results are good and close to each other which tells us that it generalizes it well.



KNN – Experiment 2: Different distance metric

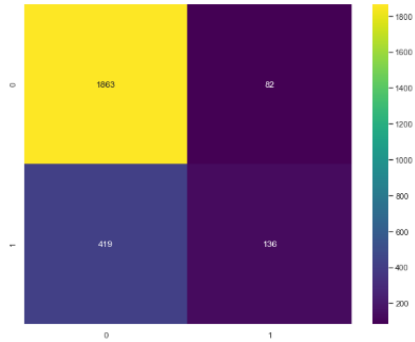
K-value: 5
Distance metric: Euclidean
Train accuracy: 82.57%
Test accuracy: 79.32%

Classification Report

	precision	recall	f1-score	support
0.0	0.82	0.96	0.88	1945
1.0	0.62	0.25	0.35	555
accuracy			0.80	2500
macro avg	0.72	0.60	0.62	2500
weighted avg	0.77	0.80	0.76	2500

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



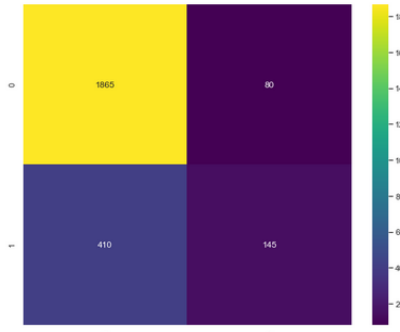
K-value: 5
Distance metric: Manhattan
Train accuracy: 83%
Test accuracy: 80.76%

Classification Report

	precision	recall	f1-score	support
0.0	0.82	0.96	0.88	1945
1.0	0.64	0.26	0.37	555
accuracy			0.80	2500
macro avg	0.73	0.61	0.63	2500
weighted avg	0.78	0.80	0.77	2500

Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)



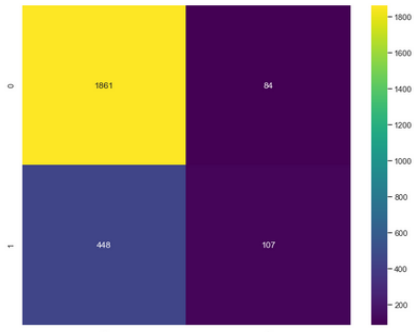
K-value: 5
Distance metric: Hamming
Train accuracy: 81.8%
Test accuracy: 78.92%

Classification Report

	precision	recall	f1-score	support
0.0	0.81	0.96	0.87	1945
1.0	0.56	0.19	0.29	555
accuracy			0.79	2500
macro avg	0.68	0.57	0.58	2500
weighted avg	0.75	0.79	0.74	2500

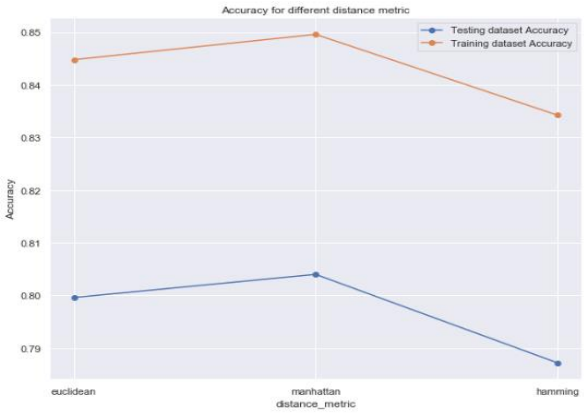
Confusion matrix

AxesSubplot(0.125,0.125;0.62x0.755)

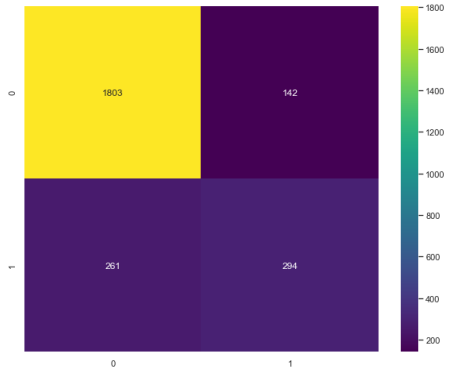
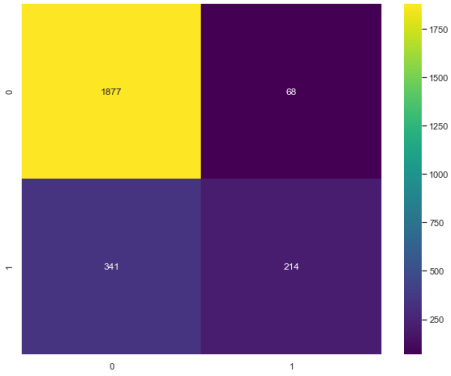
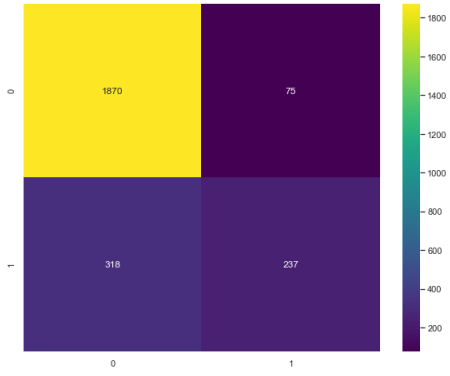


Accuracy for different distance metric:

We can conclude from the above experiment with different K values = 5 with distance metric = 'Manhattan' the best test data accuracy compared to using the other metrics ('Euclidean', 'Hamming').

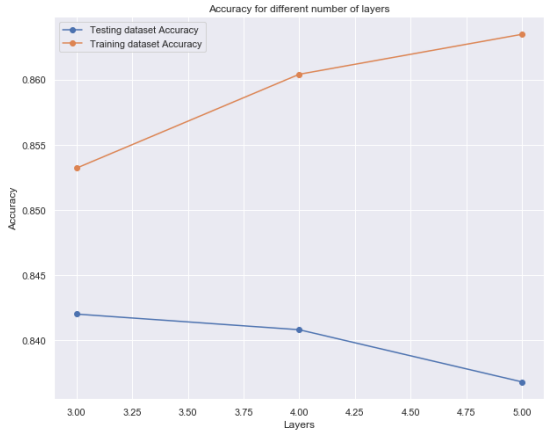


ANN - Experiment 1: Different number of layers

Layer: 3	Layer: 4	Layer: 5
Train accuracy: 83.88%	Train accuracy: 83.64%	Train accuracy: 84.28%
Test accuracy: 85.09%	Test accuracy: 85.92%	Test accuracy: 85.25%
		

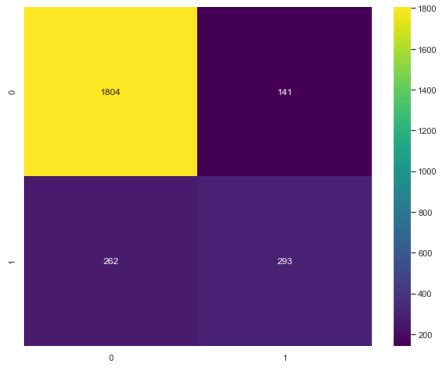
Accuracy for different number of layers:

From the above layers, we can see that both the train and the test accuracy are very close, there's only slight difference. By looking at the test accuracy we can conclude that it performs best using 4 layers.

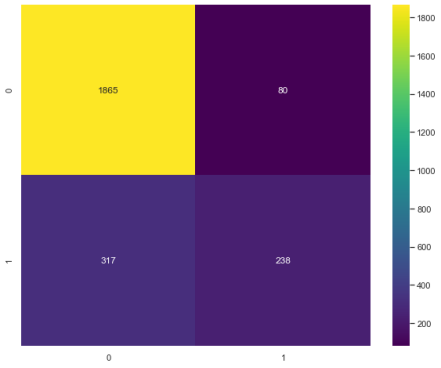


ANN - Experiment 2: Different activation functions

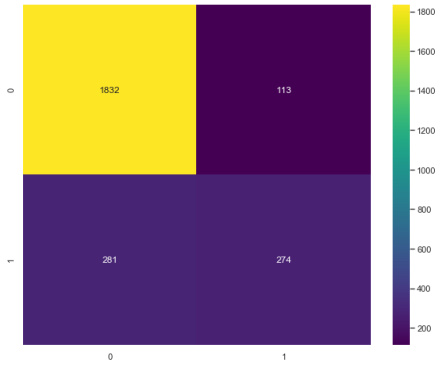
Activation set1: [‘relu’, ‘tanh’, ‘relu’, ‘sigmoid’]
Train accuracy: 83.88%
Test accuracy: 85.97%



Activation set2: [‘relu’, ‘tanh’, ‘tanh’, ‘sigmoid’]
Train accuracy: 84.12%
Test accuracy: 86.13%



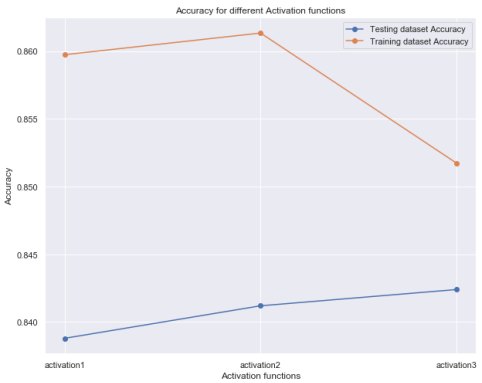
Activation set3: [‘tanh’, ‘relu’, ‘sigmoid’, ‘sigmoid’]
Train accuracy: 84.24%
Test accuracy: 85.17%



Accuracy for different activation function:

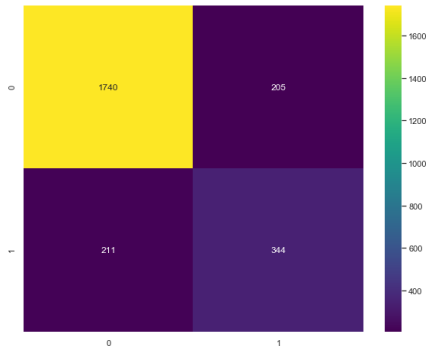
Combining both the outputs of different number of layers and different activation functions, we can see that 4 layers work best with one ‘tanh’ layers one ‘relu’ layer and two ‘sigmoid’ layer.

The activation set that gives the best result is activation set3 which is one ‘tanh’ layers one ‘relu’ layer and two ‘sigmoid’ layer, hence will use that model to vary number of neurons.

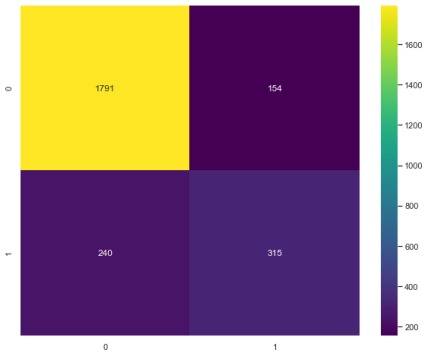


ANN - Experiment 3: Different number of nodes

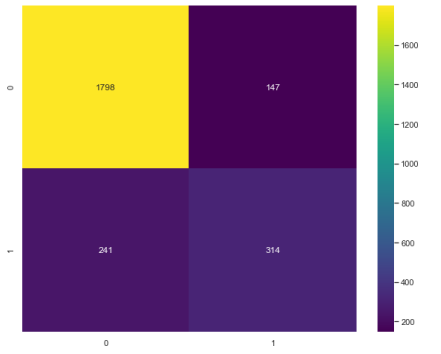
Node set1: [10, 8, 4, 1]
Train accuracy: 83.36%
Test accuracy: 83.92%



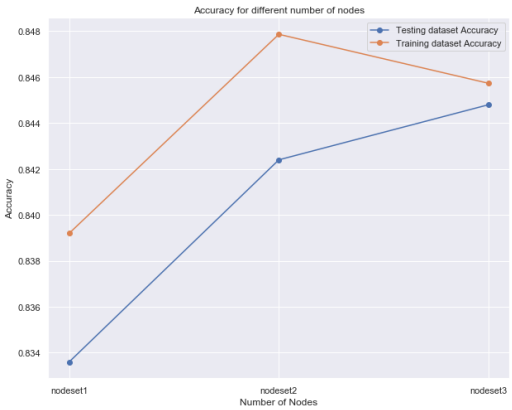
Node set2: [10, 7, 7, 1]
Train accuracy: 84.24%
Test accuracy: 84.78%



Node set3: [14, 7, 3, 1]
Train accuracy: 84.48%
Test accuracy: 84.57%



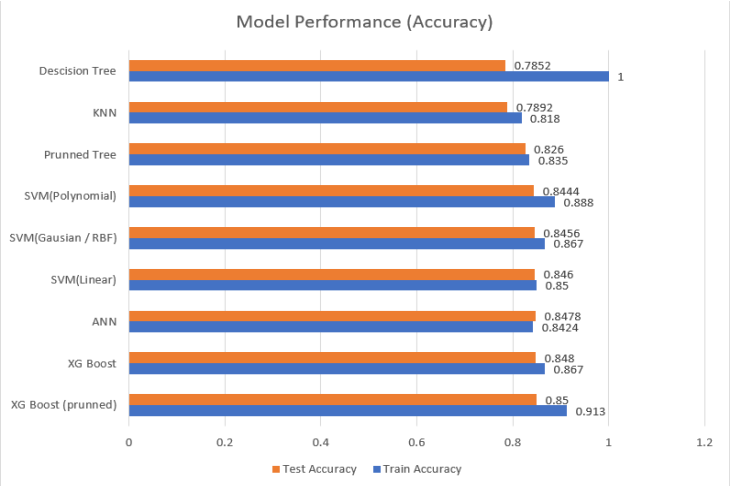
Accuracy of different number of nodes:
For both training and testing datasets, graph shows that using node set2 [10, 7, 7, 1] gives the best accuracy.



Hence for ANN 4 layers with different activation function [tanh, relu, sigmoid, sigmoid] with node set [10, 7, 7, 1] gives the best accuracy.

Model Performance:

Rank	Algorithm	Train Accuracy	Test Accuracy	Train Error	Test Error
1	XG Boost (prunned)	0.913	0.85	0.0866	0.15
2	XG Boost	0.867	0.848	0.1333	0.152
3	ANN	0.8424	0.8478	0.1576	0.1522
4	SVM(Linear)	0.85	0.846	0.1498	0.154
5	SVM(Gausian / RBF)	0.867	0.8456	0.1334	0.1543
6	SVM(Polynomial)	0.888	0.8444	0.1119	0.15559
7	Prunned Tree	0.835	0.826	0.1652	0.174
8	KNN	0.818	0.7892	0.182	0.2108
9	Descision Tree	1	0.7852	0	0.2148



Interpretation of the results:

- The models are ranked based on the test accuracy.
- If we compare KNN and ANN then ANN performs better in both test and train dataset.
- Neither KNN nor ANN perform better than any of the previous algorithms used in assignment 2.
- ANN ranks 3rd based on the accuracy of the test and train data set.
- KNN ranks 8th based on the accuracy of the test and train data set.