Introduction:

In this assignment we are working on Artificial Neural Network and K Nearest neighbors. I have used two data sets:

1) Best GPU Processor Predictor

2)Brisbane weather predictor

DATASET 1

Best GPU Processor Predictor

In part 1 , We have applied classification algorithms on GPU Processor Predictor dataset and classified the combination of processors as good and bad. We have checked for null values and made a new column i.e average runtime. We have checked for outliers and have split the dataset into training set and testing set with the ratio of 30% testing set and 70% training set. Standard scaling is done on all the feature variables.

ARTIFICIAL NEURAL NETWORK

1) Download and use any neural network package to classify your classification problems. Experiment with number of layers and number of nodes , activation functions(sigmoid,tanh etc), and may be couple of other parameters.

EXPERIMENTS

A) Trying with different number of epochs:

NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
200	RELU	1	6	0.78516	[[39768 3416] [10430 10836]]
100	RELU	1	6	0.75351	[[39066 4118] [11768 9498]]
400	RELU	1	6	0.86844	[[7918 861] [1128 5212]]

B) Trying with different number of layers:

NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
400	RELU	2	6	0.86308	[[7983 796] [1274 5066]]
400	RELU	3	6	0.58066	[[8779 0] [6340 0]]
400	RELU	4	6	0.58066	[[8779 0] [6340 0]]

C) Trying with different Activation Function:

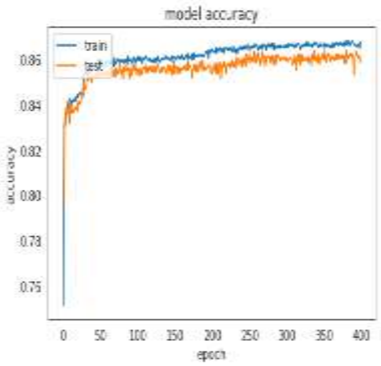
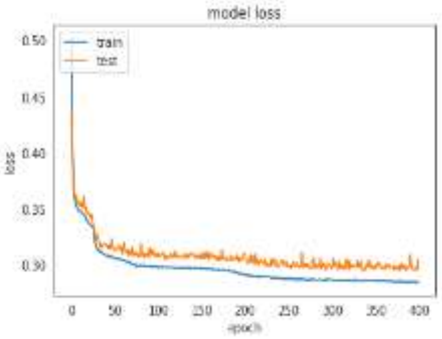
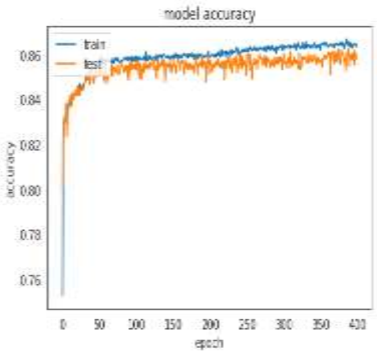
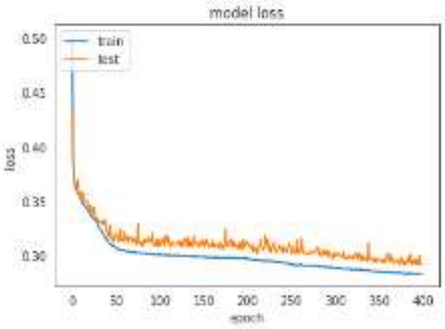
NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
400	RELU	2	6	0.86308	[[7983 796] [1274 5066]]
400	SIGMOID	2	6	0.86057	[[7717 1062] [1046 5294]]
400	SOFTMAX	2	6	0.41933	[[0 8779] [0 6340]]

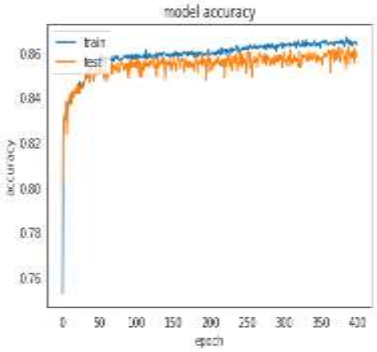
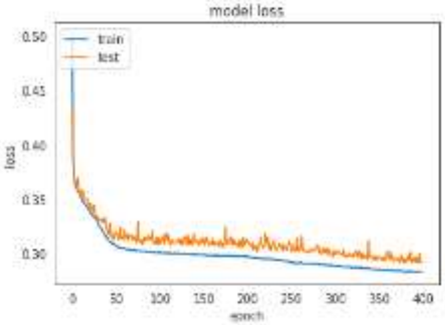
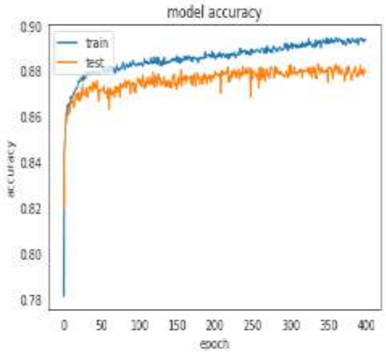
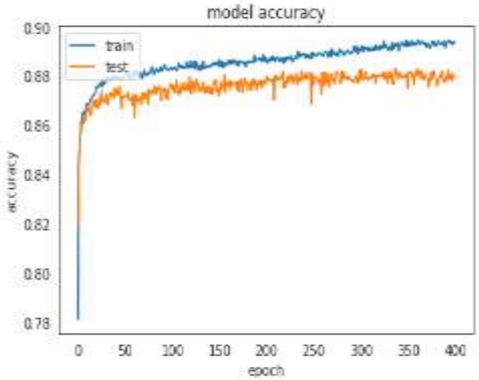
D) Changing the number of nodes to 16

NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
400	RELU	2	16	0.88385	[[7906 873] [883 5457]]

Hence the above combination is the best model with the appropriate parameters.
Below are the graphs for the above experiment.

GRPAHS:

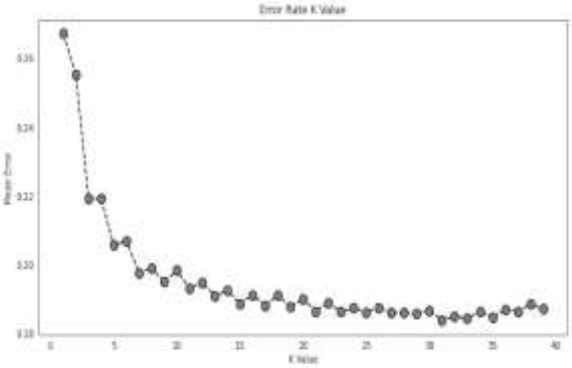
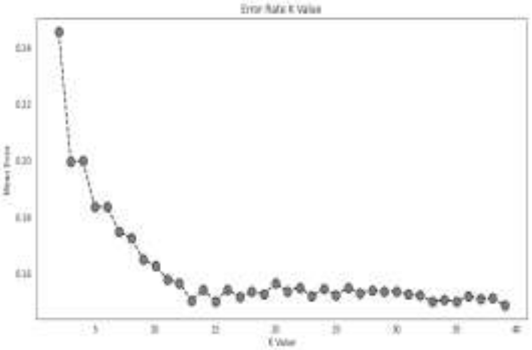
EXP.	Model Accuracy	Model Loss
A)		
B)		

C)		
D)		

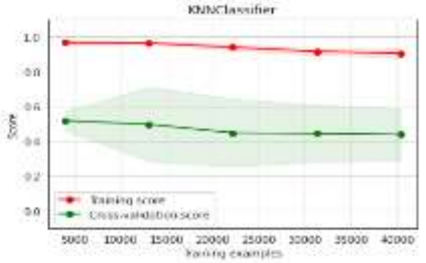
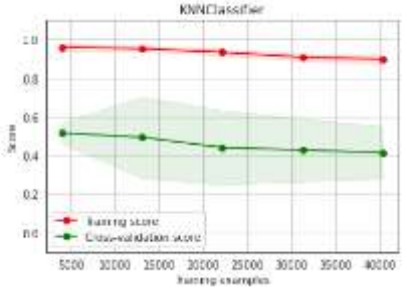
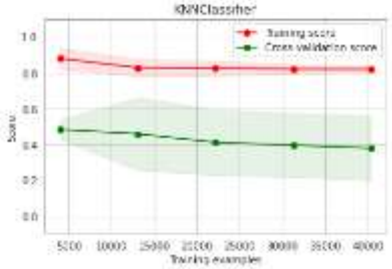
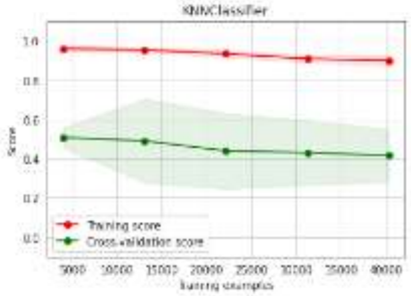
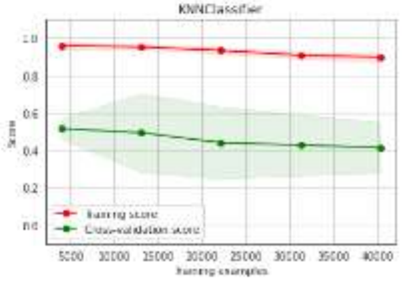
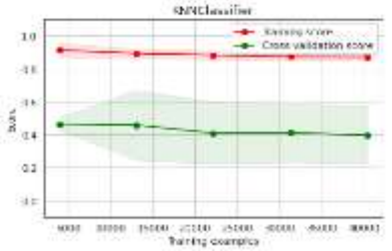
2) K NEAREST NEIGHBOURS

Download and use any KNN package to classify your classification problems. Experiment with number of neighbors. You can use any distance metric appropriate to your problem. Just be clear to explain why you used the metric that you used.

In this algorithm, I have tried with random values of k such as 5 and 3 and then applied grid search algorithm for Euclidean and Manhattan distance to find appropriate value of k. Below are the results and graphs.

Grid search for Euclidean	Grid search for Manhattan
	

Value of K	Confusion Matrix	Accuracy	Mean Accuracy with CV=5	Standard Deviation	Learning curve

5(Eucldiean)	[[7359 1420] [1691 4649]]	79.42324	78.85754	0.57531	
3	[[7218 1561] [1755 4585]]	78.06733	77.42026	0.31985	
32(Best value from grid search)	[[7823 956] [1842 4498]]	81.49348	80.43090	0.51173	
5(Manhattan)	[[7518 1261] [1514 4826]]	81.64561	81.16513	0.54125	
3	[[7347 1432] [1584 4756]]	80.0515	79.16654	0.34034	
15(best value from manhattan grid search)	[[7804 975] [1295 5045]]	84.9857	84.06236	0.389276	

DATASET 2

Brisbane weather dataset contains 24 variables. It contains 56240 rows. In this I have applied classification algorithm to decide whether it will rain tomorrow or not. There are total 7 categorical variables including date column. We will create dummy variables for categorical variables. We have checked for outliers and have split the dataset into training set and testing set with the ratio of 30% testing set and 70% training set. Standard scaling is done on all the feature variables.

ARTIFICIAL NEURAL NETWORK

EXPERIMENTS

A) Trying with different number of epochs:

NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
200	RELU	1	6	0.85655	[[12154 1004] [1424 2344]]
100	RELU	1	6	0.85708	[[12267 891] [1528 2240]]
400	RELU	1	6	0.85684	[[12353 805] [1618 2150]]

B) Trying with different number of layers:

NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
100	RELU	2	6	0.85749	[[12349 809] [1603 2165]]
100	RELU	3	6	0.85767	[[12320 838] [1571 2197]]
100	RELU	4	6	0.85773	[[12375 783] [1625 2143]]
100	RELU	10	6	0.85501	[[12181 977] [1477 2291]]

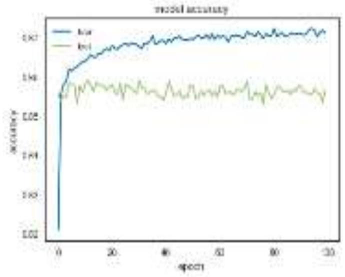
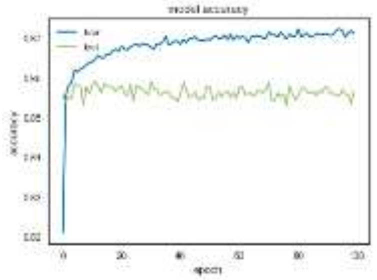
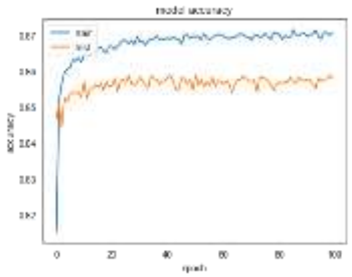
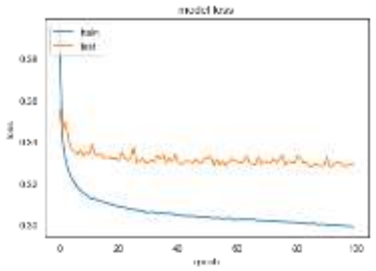
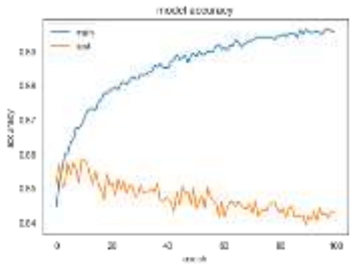
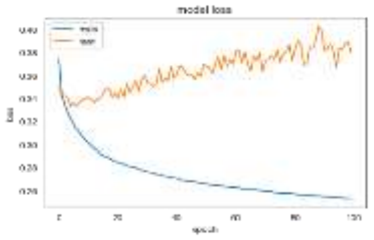
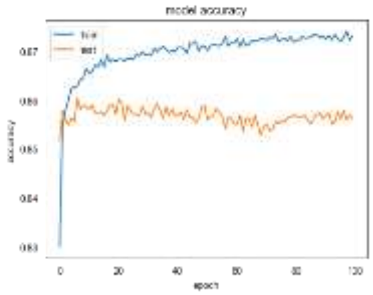
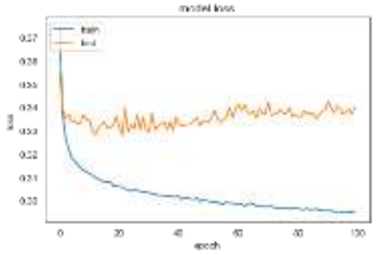
C) Trying with different Activation Function:

NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
100	SIGMOID	4	6	0.84012	[[11888 1270] [1436 2332]]
100	SOFTMAX	4	6	0.83752	[[12358 800] [1950 1818]]
100	TANH	4	6	0.84414	[[12008 1150] [1488 2280]]

D) Changing the number of nodes to 16

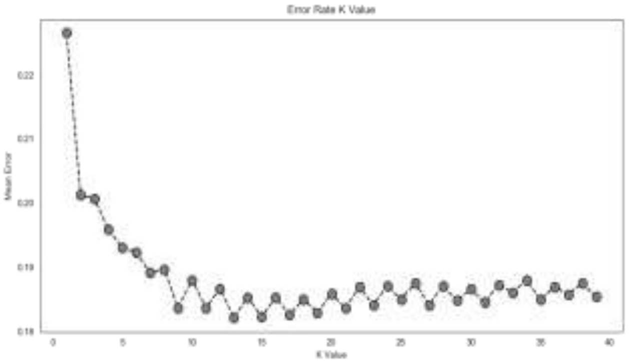
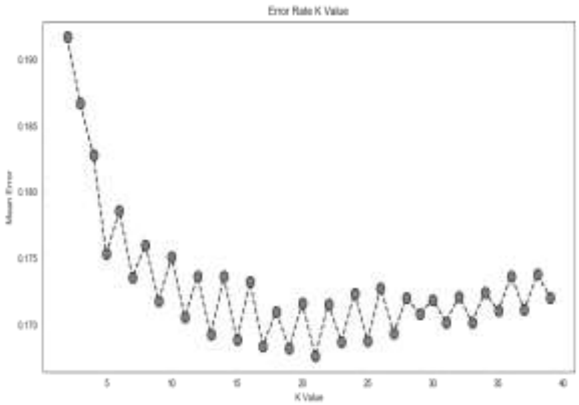
NUMBER OF EPOCHS	ACTIVATION FUNCTION	NO. OF HIDDEN LAYERS	NUMBER OF NODES	ACCURACY	CONFUSION MATRIX
100	RELU	2	16	0.85288	[[12248 910] [1580 2188]]

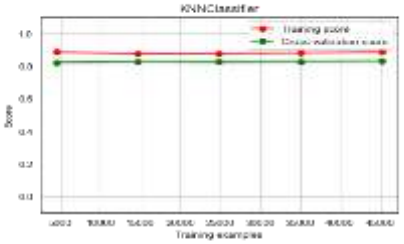
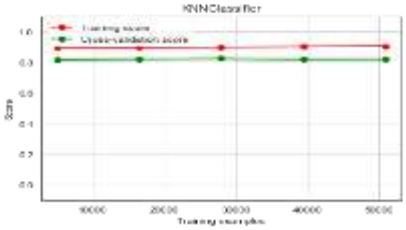
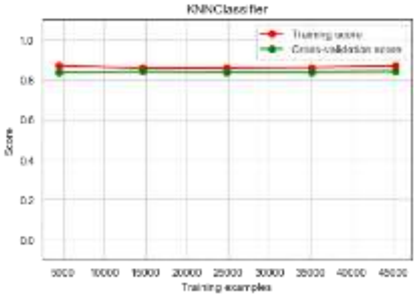
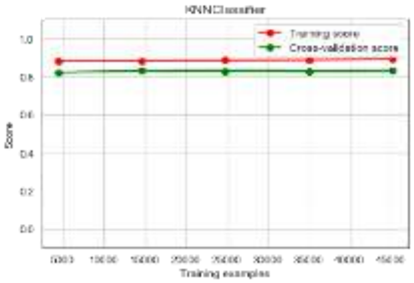
GRPAHS:

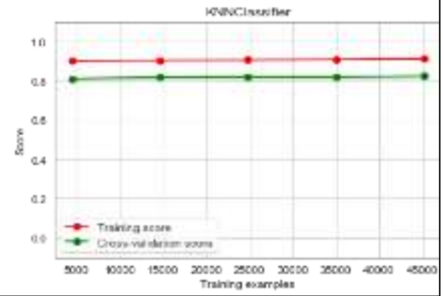
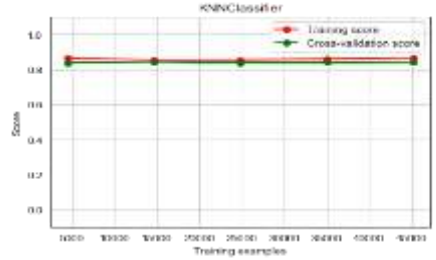
EXP.	Model Accuracy	Model Loss
A)		
B)		
C)		
D)		

3) K NEAREST NEIGHBOURS

Download and use any KNN package to classify your classification problems. Experiment with number of neighbors. You can use any distance metric appropriate to your problem. Just be clear to explain why you used the metric that you used.
In this algorithm, I have tried with random values of k such as 5 and 3 and then applied grid search algorithm for Euclidean and Manhattan distance to find appropriate value of k. Below are the results and graphs.

GRID Search for Euclidean	Grid Search for Manhattan
	

Value of K	Confusion Matrix	Accuracy	Mean Accuracy with CV=5	Standard Deviation	Learning curve
5(Eucldiean)	[[12387 771] [2495 1273]]	80.70424	81.12624	0.32931	
3	[[12130 1028] [2367 1401]]	79.94210	80.32359	0.30965	
13(Best value from grid search)	[[12727 431] [2650 1118]]	81.79723	81.936486	0.243646	
5(Manhattan)	[[12429 729] [2238 1530]]	82.47075	82.534052	0.3279131	

3	[[12187 971] [2189 1579]]	81.33049	81.7162	0.19626	
21(best value from grid search)	[[12787 371] [2466 1302]]	83.23880	83.2987	0.26177	

ACCURACY COMPARISON :

DATASETS	SVM	Decision Trees	Boosting	ANN(Relu)	KNN(Manhattan)
Dataset 1(GPU)	78.7323506594259	68.2932	75.4584	88.3854	84.9857
Dataset 2(Brisbane weather predictor)	86.44097837646225	80.1784	85.4011	085.7733	83.2388

ANN did better for dataset 1 and SVM did better for dataset 2.

IMPROVEMENTS AND SUGESSTIONS:

1) **More combinations of activation functions , number of layers and number of nodes could have been tried to achieve better accuracy and results.**
2) **Plotting accuracy curve for different algorithms could have helped us to compare the different algorithms better.**