Machine Learning Assignment-III

GPU Dataset

Data is taken from UCI machine learning repository and is collection of observations of different GPU run times under different machine configurations

- There are total 14 features, of which first 10 are ordinal while last 4 are binary and total number of observations is 241600
- There are no missing values in data
- There is no need for scaling the data as different features are approximately in same range

Data Preparation for Modelling

- Average run time has been converted into categorical variable based on median value of given runtime thus transforming the problem into classification problem
- Complete dataset is divided into training (80%) and testing dataset (20%).
- Used K-Fold Cross Validation and GridSearchCV for HyperParameter Tuning in K Nearest Neighbours and Artificial Neural Networks respectively.

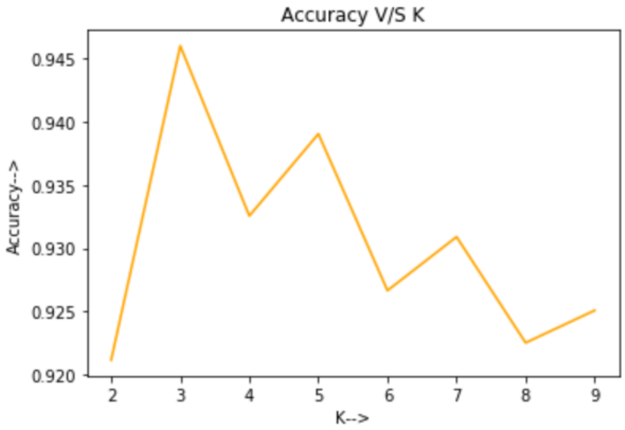
Model 1: K Nearest Neighbors

Experiment

- Classification of GPUs between high run time or low run time based optimal N_Neighbors value using KNN Classifier.

Observations

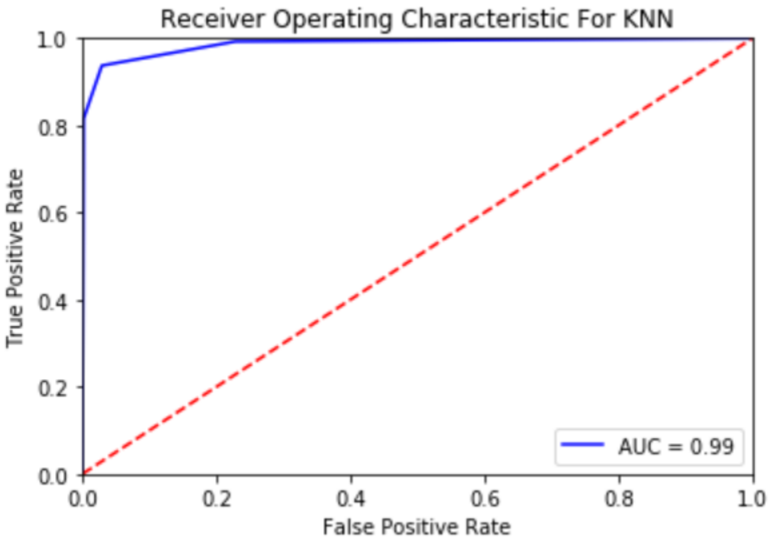
- While performing K-Fold CV on the training dataset to find out optimal value of K, K=3 gave the maximum Cross Validation Accuracy, whereas underfitting is observed as we increased the value of K as it was expected. A CV accuracy V/S K graph has been plotted to observe the change in accuracies with increasing values of K.
- The Training Accuracy came out to be 98.18% while the Test Accuracy came out to be 95.42%, which seems to be a good result as no overfitting and underfitting problem is observed at optimal K(K=3).



Training Accuracy	98.18%
Test Accuracy	95.42%

ROC CURVE:

The ROC is showing a good lift, which implies that the model is much better than the dumb model(Red Line) and AUC=0.99 implies that the binary classification model has good measure for separability.



Confusion Matrix:

TN:23542	FP:705
FN:1508	TP:22565

Conclusions

- As observed from the Accuracy V/S K graph and Training and Test Accuracies, the optimal value of K=3 gives the highest ***Training(98.18%) and Test(95.42%) accuracies.***

Model 2: Artificial Neural Network

Experiment

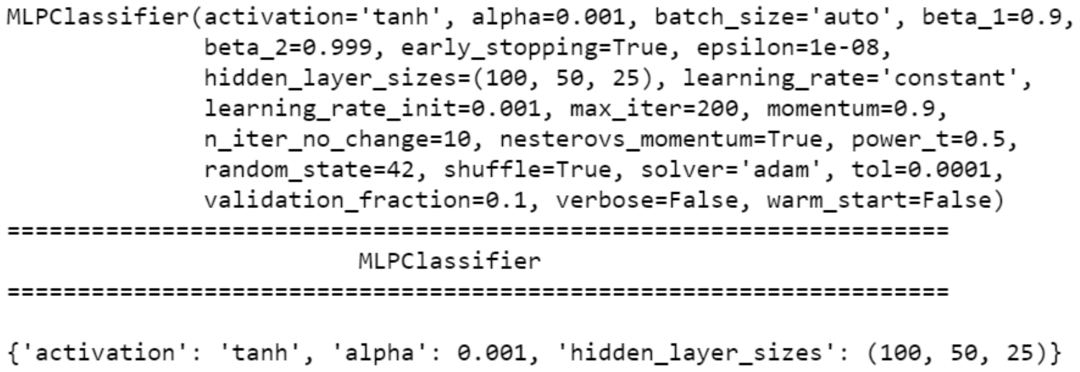
- Try to improve the accuracy using Artificial Neural Networks using multiple hidden layers.
- I have used stochastic gradient-based weight optimizer solver(Adam) and constant learning rate.
- I have also set early_stopping to true, to terminate training if the validation score is not improving.
- Used Grid SearchCV to obtain : Activation Function, Hidden Layer Sizes and L2 regularization term.

Observations

- Values checked for:
  Activation Function: Logistic, ReLu, tanh.
  Regularization Term: 0.1, 0.01, 0.001.
  Hidden Layer Sizes: (100,50,25) Three hidden layers with 100,50,25 neurons respectively and (100,50) Two hidden layers with 100,50 neurons respectively.

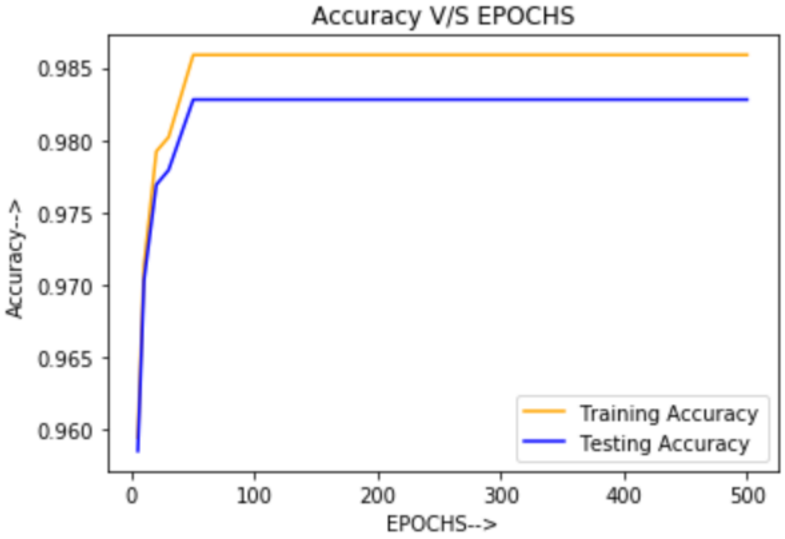
- Again checked training and test accuracies for different value of epochs keeping the best set of parameters obtained from GridSearchCV. A graph Train and Test Accuracy V/S Epochs has been plotted to check for the number of epochs which give highest train and test accuracies.

Result Of Grid Search:

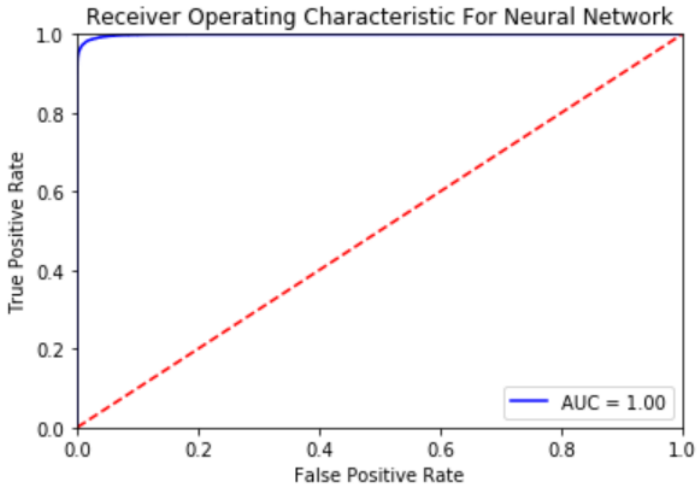


```
MLPClassifier(activation='tanh', alpha=0.001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=True, epsilon=1e-08,
              hidden_layer_sizes=(100, 50, 25), learning_rate='constant',
              learning_rate_init=0.001, max_iter=200, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=42, shuffle=True, solver='adam', tol=0.0001,
              validation_fraction=0.1, verbose=False, warm_start=False)
===================================================================
                         MLPClassifier
===================================================================

{'activation': 'tanh', 'alpha': 0.001, 'hidden_layer_sizes': (100, 50, 25)}
```



ROC-AUC:



The ROC is showing a good lift, which implies that the model is much better than the dumb model (Red Line) and AUC=1 implies that the binary classification model has good measure for separability.

Confusion Matrix:

TN:23939	FP:318
FN:513	TP:23560

Conclusions

- The highest train and test accuracies were found at 50 epochs and kept constant beyond that.
- Thus, final Neural Network model has:
  Activation Function: **tanh**
  Hidden Layers: **3 hidden layers with 100,50,25 neurons each.**
  L2 Regularization Term(Alpha) : **0.001**
  Epochs(Max Iter) : **50.**
- The final ANN Model has:
  Training Accuracy : 98.58%
  Test Accuracy: 98.28%

Final Conclusions for GPU Dataset :

- Below is the test accuracy of final models of support vector machines, decision trees and ensemble methods

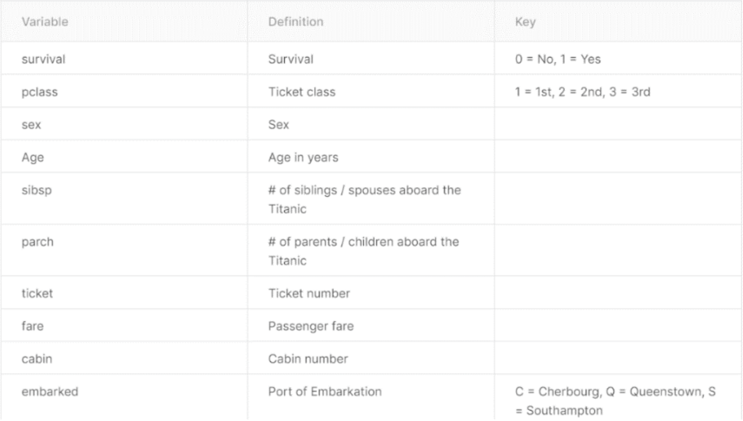
Models	Test Accuracy
Support Vector Machines	0.8760
Decision Tree Classifier	0.9653
AdaBoost Classifier	0.9617
Artificial Neural Network	0.9828
K Nearest Neighbors	0.9542

- Thus, looking at above table we can conclude, **Artificial Neural Network** gives best results for GPU Dataset 1**(98.28%).**
- Also, as we can say that nearly all the algorithms are performing better on this dataset expect SVM which has a slightly lower test accuracy but this lower accuracy can be accounted for the lesser training datapoints as high computation was required.

Titanic Dataset

The famous Titanic Dataset from Kaggle is used to classify Survival of the passengers who boarded the famous Titanic ship.

Features and relevant Information about the Dataset:



Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Data Preparation for Modelling

- Checked for the percentage of missing data points and dropped "Cabin" as it has 77% missing values, and then imputed Age and Embarked with mean and mode values respectively.
- Feature Engineered a new feature by combining Parch and SibSp into a feature named Family Size.
- Dropped SibSP,Parch,Name features.
- One-Hot Encoded Embarked and Sex features and dropped the existing Embarked and Sex columns to avoid the exact collinearity problem in the dataset.
- Complete dataset is divided into training (70%) and testing (30%) dataset
- Cross validation has been done for selection of hyperparameter while test dataset would be used for checking the accuracy of final model.

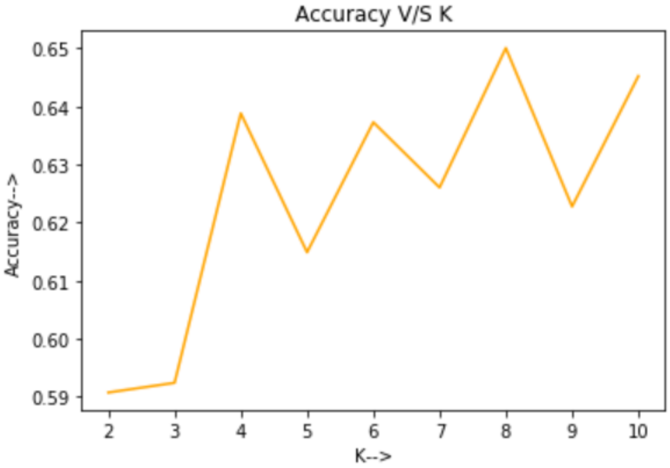
Model 1: K Nearest Neighbors

Experiment

- Classification of GPUs between high run time or low run time based optimal N_Neighbors value using KNN Classifier.

Observations

- While performing K-Fold CV on the training dataset to find out optimal value of K, K=8 gave the maximum Cross Validation Accuracy, where as underfitting is observed as we increased the value of K as it was expected. A CV accuracy V/S K graph has been plotted to observe the change in accuracies with increasing values of K.
- The Training Accuracy came out to be 69.02% while the Test Accuracy came out to be 64.18%, which seems to be a good result as no overfitting and underfitting problem is observed at optimal K(K=8).



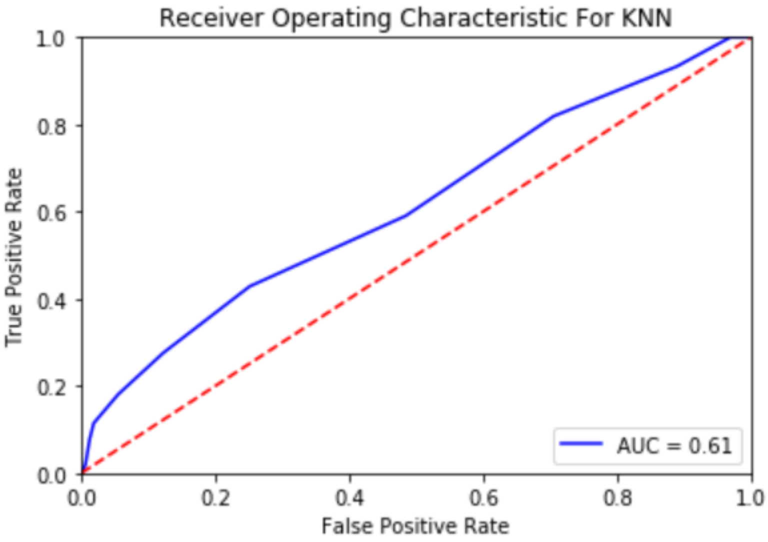
Training Accuracy	69.02%
Test Accuracy	64.18%

Confusion Matrix:

TN:143	FP:20
FN:76	TP:29

ROC CURVE:

The ROC is showing a decent lift, which implies that the model is only slightly better than the dumb model (Red Line) and AUC=0.61 implies that the binary classification model has better measure for separability when compared to the base model. However, this model does not give satisfactory result when compared to other models on this dataset.



Conclusions

- As observed from the Accuracy V/S K graph and Training and Test Accuracies, the optimal value of K=3 gives the highest ***Training(69.02%) and Test(64.18%) accuracies.***

Model 2: Artificial Neural Network

Experiment

- Try to improve the accuracy using Artificial Neural Networks using multiple hidden layers.
- I have used stochastic gradient-based weight optimizer solver(Adam) and constant learning rate.
- I have also set early_stopping to true, to terminate training if the validation score is not improving.
- Used Grid SearchCV to obtain : Activation Function, Hidden Layer Sizes and L2 regularization term.

Observations

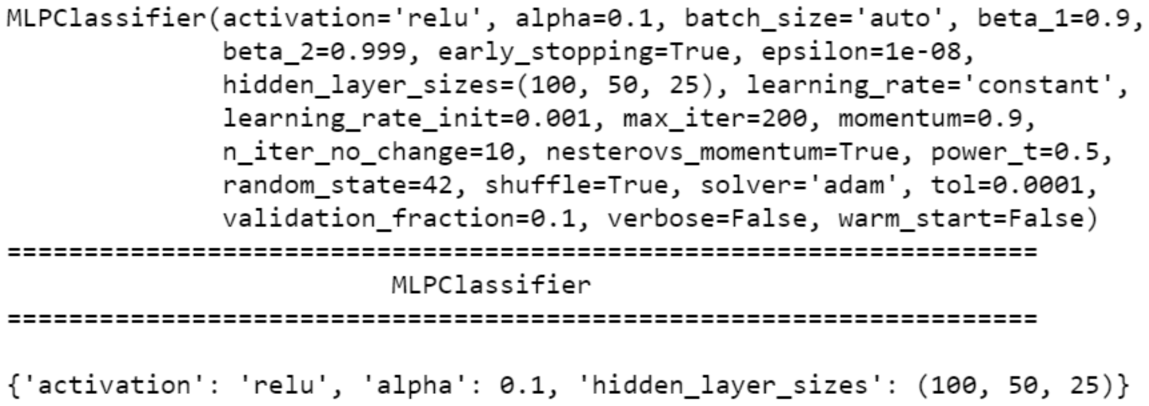
- Values checked for:
  Activation Function: Logistic, ReLu, tanh.
  Regularization Term: 0.1, 0.01, 0.001.

Hidden Layer Sizes: (100,75,50,25,10) Five hidden layers with 100,75,50,25,10 neurons respectively, (100,50,25) Three hidden layers with 100,50,25 neurons respectively and (100,50) Two hidden layers with 100,50 neurons respectively.

- Again checked training and test accuracies for different value of epochs keeping the best set of parameters obtained from GridSearchCV. A graph Train and Test Accuracy V/S Epochs has been plotted to check for the number of epochs which give highest train and test accuracies.

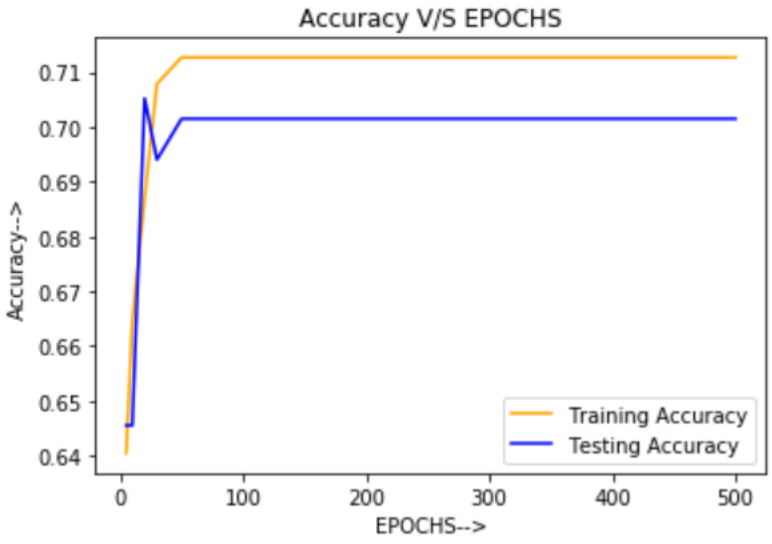
Observations:

#Result Of GridSearchCV



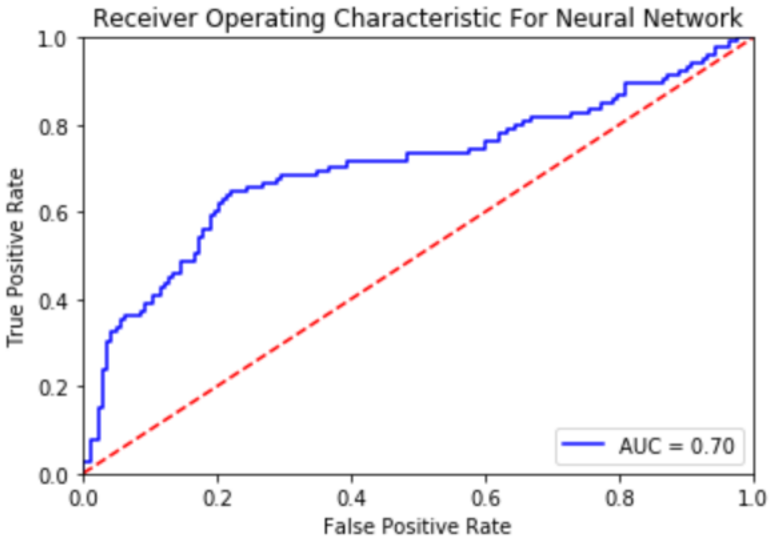
```
MLPClassifier(activation='relu', alpha=0.1, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=True, epsilon=1e-08,
              hidden_layer_sizes=(100, 50, 25), learning_rate='constant',
              learning_rate_init=0.001, max_iter=200, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=42, shuffle=True, solver='adam', tol=0.0001,
              validation_fraction=0.1, verbose=False, warm_start=False)
===================================================================
                         MLPClassifier
===================================================================

{'activation': 'relu', 'alpha': 0.1, 'hidden_layer_sizes': (100, 50, 25)}
```



ROC-AUC Curve:

The ROC is showing a decent lift, which implies that the model is only slightly better than the dumb model (Red Line) and AUC=0.70 implies that the binary classification model has better measure for separability when compared to the base model and KNN. However, this model does not give satisfactory result when compared to other models on this dataset.



Confusion Matrix:

TN:135	FP:28
FN:52	TP:53

Conclusions

- The highest train and test accuracies were found at 50 epochs and kept constant beyond that.
- Thus, final Neural Network model has:
  Activation Function: **RelU**
  Hidden Layers: **3 hidden layers with 100,50,25 neurons each.**
  L2 Regularization Term(Alpha) : **0.1**
  Epochs(Max_Iter) : **50.**
- The final ANN Model has:
  Training Accuracy : 71.23%
  Test Accuracy: 70.15%

Final Conclusions for Titanic Dataset:

- Below is the test accuracy of final models of support vector machines, decision trees and ensemble methods

Models	Test Accuracy
Support Vector Machines	0.7723
Decision Tree Classifier	0.8171
AdaBoost Classifier	0.8134
Artificial Neural Network	0.7015
K Nearest Neighbors	0.6418

- Thus, looking at above table we can conclude, Decision Tree Classifier and ensemble learning (XGBClassifier) gives best results for titanic dataset.