Introduction to Machine Learning

Assignment 3

Azure note book link – Datasets used for this assignment were too large to implement on local machine. So, during the execution, problems like long execution as well as lack of memory were occurred frequently. So, I have created three different azure notebook files in following fashion and I have added the links under the respective questions.

I. Notebook for Question 1
Filename - Omkar_Pandit_Lab_Assignment3_Q1

II. Notebook for Question 2 – BBC news dataset
Filename - Omkar_Pandit_Lab_Assignment3_Q2_bbc_1

III. Notebook for Question 2 – Cornell sentiment polarity dataset
Filename - Omkar_Pandit_Lab_Assignment3_Q2_reviews_2

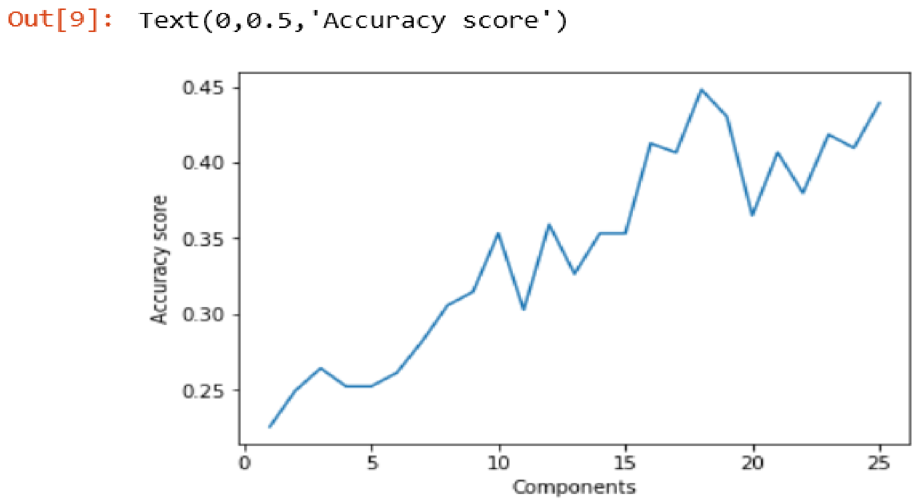
Q1. Azure note book link - https://omkarpandit-oap338.notebooks.azure.com/j/notebooks/Omkar_Pandit_Lab_Assignment3_Q1.ipynb

a. Create 75/25 train/test split of the dataset. (Code).

b. For each dataset, train and evaluate two classification algorithms.

c. Plots to show results for both classification algorithms on both datasets.

1. Decision tree on LFW dataset.

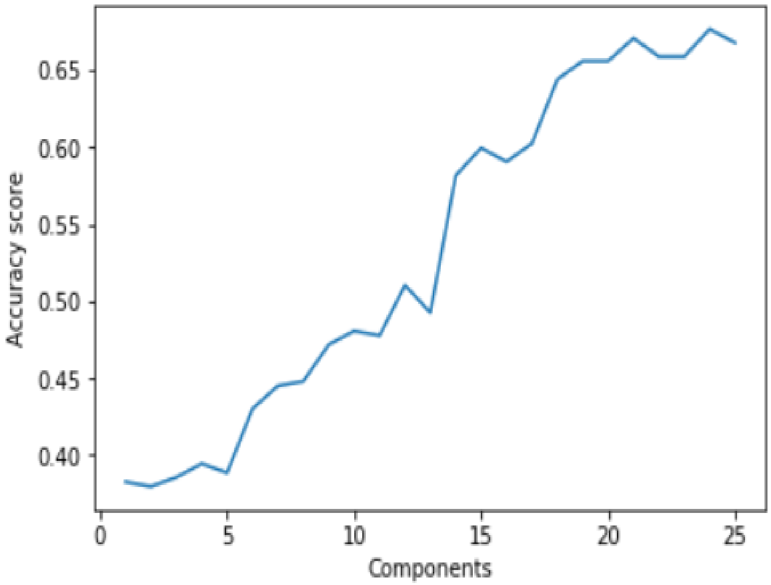


```
Out[9]: Text(0,0.5,'Accuracy score')
```

Highest accuracy achieved was 0.44807121661721067 at 18th component.

2. Naïve Bayes on LFW dataset.

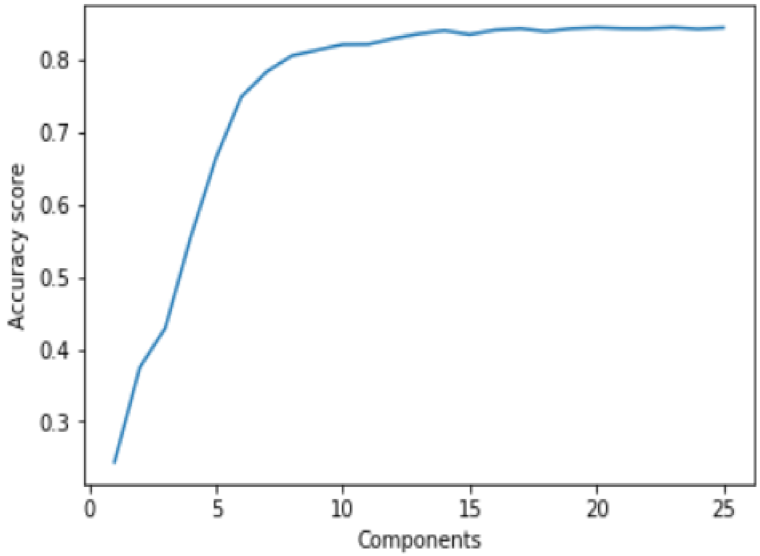
```
Out[10]:  Text(0,0.5,'Accuracy score')
```



Highest accuracy achieved was 0.6765578635014837 at 21st component.

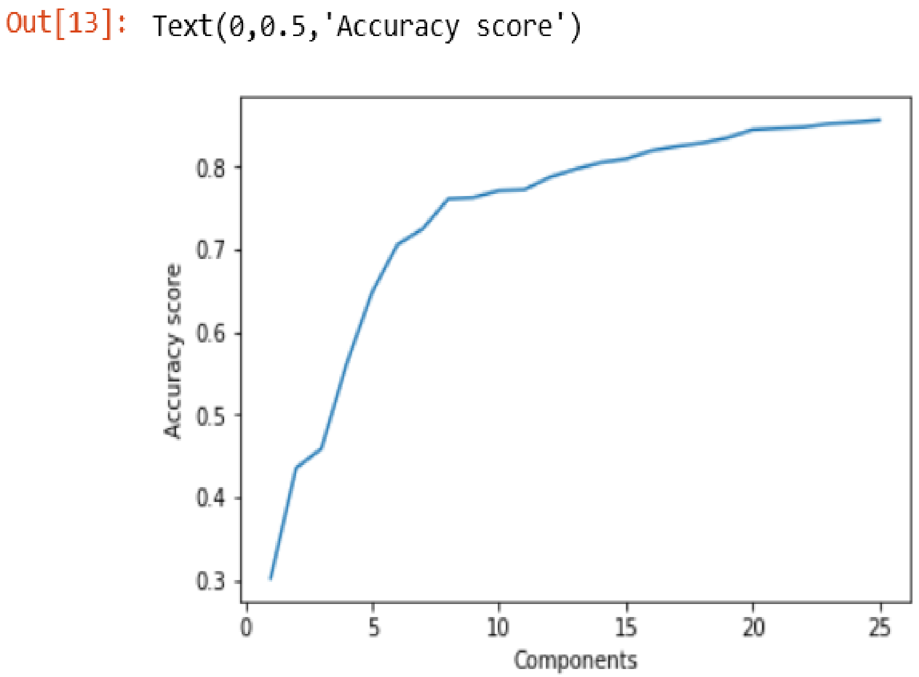
3. Decision tree on MNIST dataset.

```
Out[12]:  Text(0,0.5,'Accuracy score')
```



Highest accuracy achieved was 0.844047619047619 at 23rd component.

4. Naïve Bayes on MNIST dataset.



```
Out[13]: Text(0,0.5,'Accuracy score')
```

Highest accuracy achieved was 0.8562380952380952 at 25th component.

d. Principle Component Analysis (PCA) is a dimensionality reduction technique. So, the ideal situation to use PCA is, when you have too many numbers of predictors/ features compare to number of observations. So, in our case we have limit the number of features up to 25.
I have tried 25 variations in applying various number of PCA's on given classification problems. The overall observed trend is, as number of components/ reduced features increase eventually accuracy also increase. As PCA reduces the number of input features it also reduces the flexibility of the model. Therefore, for our datasets, accuracy increase with increase in the number of components, but this is applicable only up to the certain point.

MNIST – Both decision tree and Naïve Bayes classifiers are giving good accuracy measures around 85% with use of significant number of components i.e. number of components - 23, 25 respectively. But if we you observe, there is a slight increase in accuracies when we have applied 10-12 or more components.

Reason – As in MNIST dataset we have limited number of categories i.e. images can be any number between 0-9. So, as this has clear and limited categories to classify, classifiers performed better.

LFW – If we compare accuracy scores of both decision tree and Naïve Bayes classifiers then Naïve Bayes (67%) is performing better than Decision trees algorithm (44%) with use of significant number of components i.e. number of components - 21, 18 respectively.

Reason – Decision tree always try to interact all feature values together to perform classification. But in case of image classification problems there are lot many features to combine to build a tree. So, decision tree in case of image classification can be very complex so this might be the reason for poor results.

Q2. Optimize Hyperparameter(s) for Each Classification Model [60 points]:

When evaluating each model in this problem, perform stratified 10-fold cross-validation on the training dataset and use the "accuracy" measure to assess performance.

a. Load two text-based classification datasets (e.g., from NLTK or Kaggle) and pre-process as necessary to arrive at a numerical representation of the text.

1. BBC news dataset consists of 2225 documents related to 5 typical areas –
   1 – business, 2- entertainment, 3 - politics, 4- sport, 5 -tech
   So, Number of unique classes – 5
   Azure notebook link - https://omkarpandit-oap338.notebooks.azure.com/j/notebooks/Omkar_Pandit_Lab_Assignment3_Q2_bbc_1.ipynb

2. Cornell sentiment polarity dataset consist of 1000 positive and 1000 negative processed reviews.
   1 – negative, 0 – positive.
   So, Number of unique classes – 2
   Azure notebook link - https://omkarpandit-oap338.notebooks.azure.com/j/notebooks/Omkar_Pandit_Lab_Assignment3_Q2_reviews_2.ipynb

b. For each dataset, evaluate these classifiers using 10-fold cross validation:

Classifiers used – Decision Tree, Naïve Bayes, Logistic regression

c. Report in a table the mean accuracy of each classifier.

I. Accuracy report of BBC news dataset.

Classifier	Mean Accuracy Score
Naïve Bayes	0.92
Decision Tree	0.82
Logistic regression	0.97
Majority Vote	0.95
Bagging	0.96
Boosting	0.87

II. Accuracy report of Cornell sentiment polarity dataset.

Classifier	Mean Accuracy Score
Naïve Bayes	0.65
Decision Tree	0.65
Logistic regression	0.83
Majority Vote	0.77
Bagging	0.77
Boosting	0.66

d. Write a discussion about the performance.

Ensemble learning approach mainly deals with combining multiple algorithms to achieve better accuracy/ results. So, I have used Majority Vote classifier in which I have combined and compare three different classifiers namely Naïve Bayes, Decision Tree, Logistic regression. So, if we observe, above accuracy reports clearly indicate Ensemble classifier gives better results than some individual classifiers.

Also, along with the use of ensemble classifier, I have also implemented K-fold cross validation resampling methods to boost up the accuracy. But for me as I have used both cross validations and ensemble learning (which itself is a very time and resource consuming process) executions were time consuming. I have also faced 'Memory error' multiple times during these executions.

BBC news dataset – For this dataset, overall all the classifiers gave good accurate results. But if we compare the accuracy scores, Logistic regression (97%) performed better than the rest. But accuracy scores are so close to each other that I can say, not only Logistic regression but also Majority Vote performed best (95%). Here, I succeed in achieving better accuracy as I have implemented bagging on top of Majority Vote classifier (96%). Decision tree performed worst with an accuracy score of 82%.
Reason – To obtain better results using Decision Tree classifier, tree must be as deep as possible. SO, to build deep tree more lot of data covering all the given possibilities is required. But in our case, we have only around 2000 samples to do the text classification which is not enough for better Decision Tree classification.

Cornell sentiment polarity dataset – For this dataset, if we compare the accuracy scores, Logistic regression (83%) performed better than the rest. Majority Vote classifier gave 77% accuracy which is not bad if compare to all three individual accuracy score. Decision tree performed worst with an accuracy score of 65%.

Reason – To obtain better results using Decision Tree classifier, tree must be as deep as possible. SO, to build deep tree more lot of data covering all the given possibilities is required. But in our case, we have only around 2000 samples to do the text classification which is not enough for better Decision Tree classification.