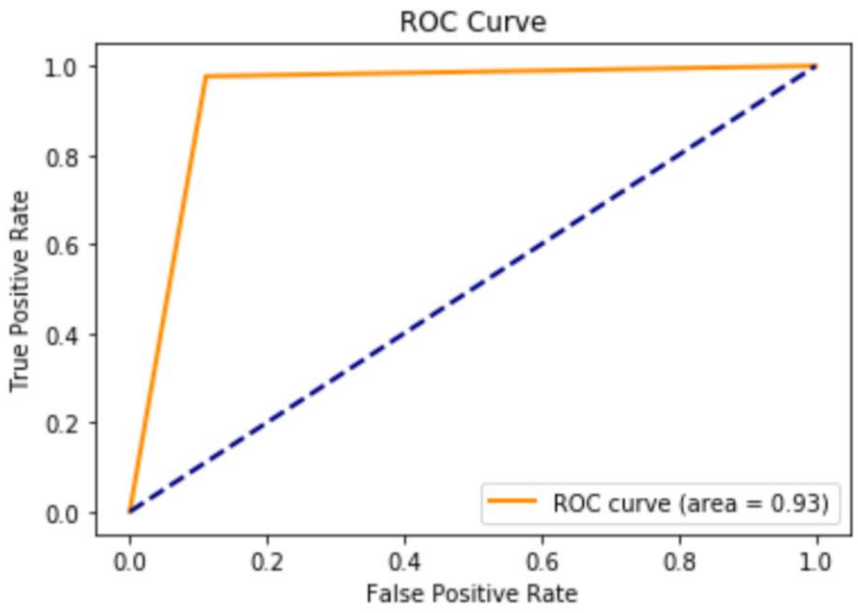
USC ID: 9907399097

Name: Subhiksha Rani

Homework-6 Report

1. Supervised, Semi-Supervised, and Unsupervised Learning
    a. Downloading the dataset and choosing 20% of Positive class (Malignant) and 20% of Negative class (Benign) as test set.

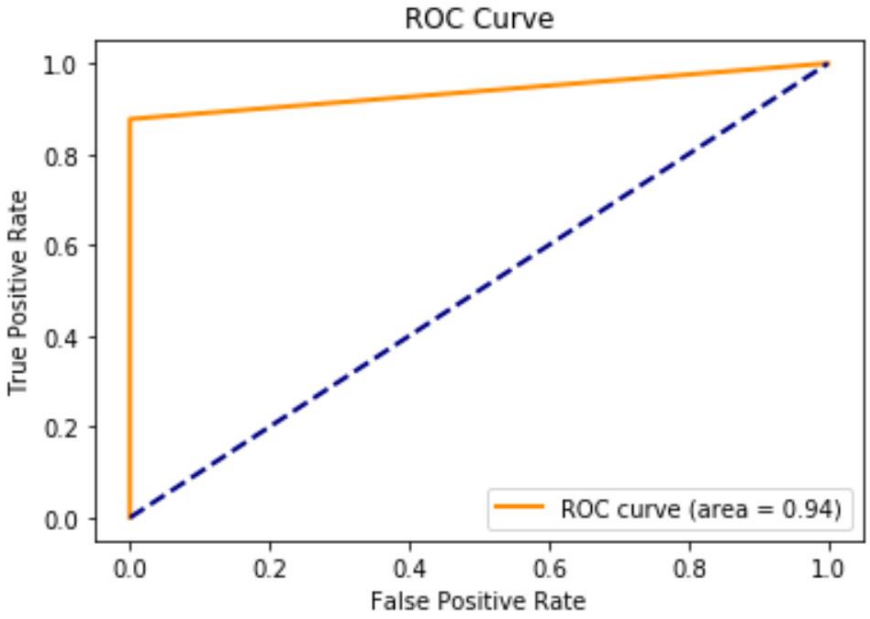
    b. **Monte-Carlo Simulation:**
        i. **Supervised Learning**:
        Average Accuracy Score: 0.9220289855072463
        Average Precision: 0.9220289855072463
        Average Recall: 0.9220289855072463
        Average F1-score: 0.9220289855072464
        Average AUC: 0.9319552110249785



Confusion Matrix:
[[64 8]
[1 42]]

ii. **Semi-Supervised Learning/ Self-training:**

Average Accuracy Score: 0.9526315789473684
Average Precision: 0.9526315789473684
Average Recall: 0.9526315789473684
Average F1-score: 0.9526315789473684
Average AUC: 0.9441437405572538



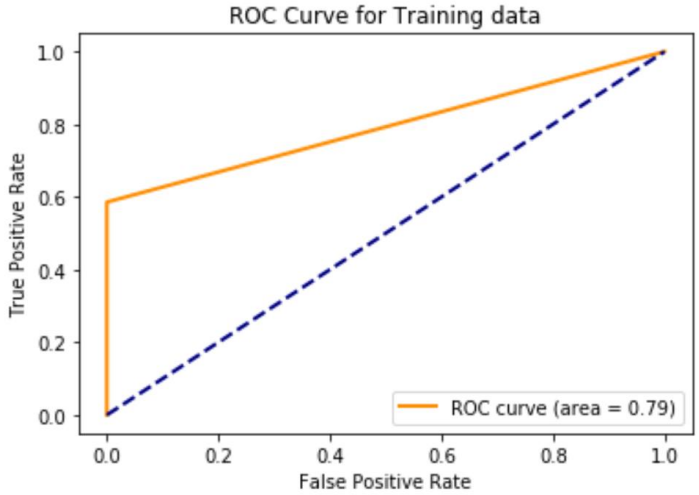
Confusion Matrix:
[[179 0]
[13 93]]

iii. **Unsupervised Learning:**

A. K-means algorithm is sensitive to the initial placement of cluster centers. With a better initial centers, the algorithm will generate a much better solution as compared with badly placed cluster center, the algorithm can converge into a local minima.
We can avoid K-means algorithm from getting stuck in local minima by using the following techniques:

- We could try many random starting points
- We could try non-local split-and-merge moves: Simultaneously merge two nearby clusters and split a big cluster into two.
- We could try using the largest distance algorithm to determine K initial cluster focal points. We can then combine it with traditional K-means algorithm. Finally, we can accomplish the classification of pattern congregation. This improved K-means algorithm is better than the traditional one in aspects such as: Precision of cluster, Speed of Cluster, Stability etc.

B. Scores of Training Set:
Average Accuracy Score: 0.8497797356828194
Average Precision: 0.8497797356828194
Average Recall: 0.8497797356828194
Average F1-score: 0.8497797356828194
Average AUC: 0.7991079276099519



Confusion Matrix for Training data:
[[285 0]
[70 99]]

C. Scores of Test Set:

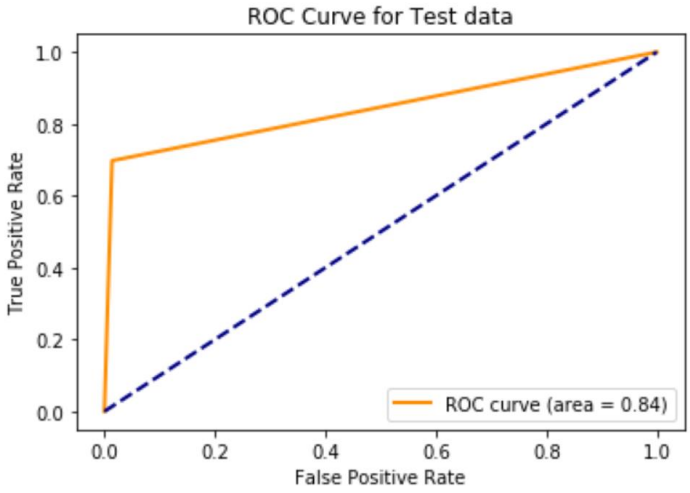
Average Accuracy Score: 0.852463768115942

Average Precision: 0.852463768115942

Average Recall: 0.852463768115942

Average F1-score: 0.852463768115942

Average AUC: 0.803805986218777



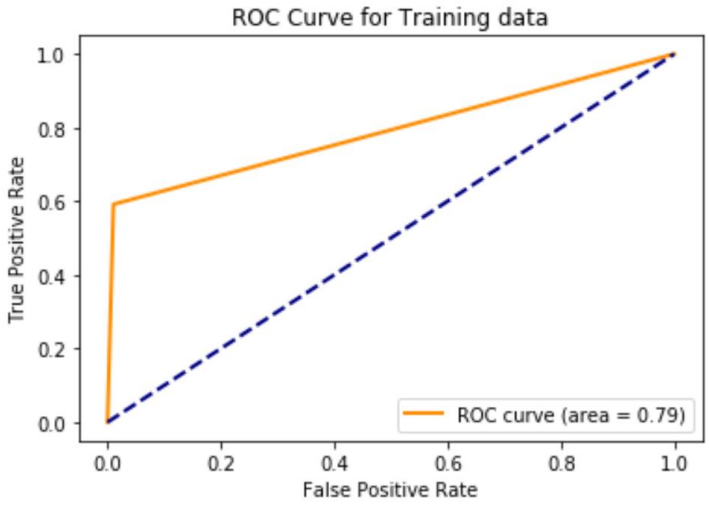
Confusion Matrix for Test data:

[[71 1]

[13 30]]

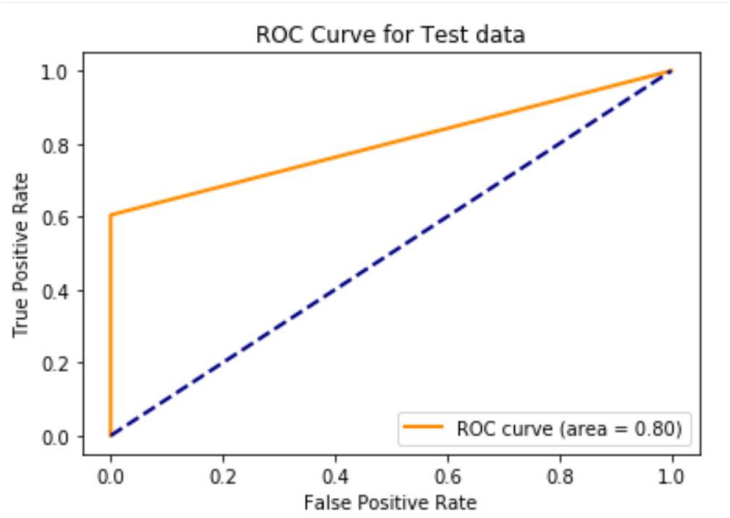
iv. **Spectral Clustering:** In multivariate statistics and the clustering of data, spectral clustering techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset.

Scores of Training Set:
Average Accuracy Score: 0.8566813509544787
Average Precision: 0.8566813509544787
Average Recall: 0.8566813509544787
Average F1-score: 0.8566813509544787
Average AUC: 0.8097428976781204



Confusion Matrix for Training data:
[[282 3]
[69 100]]

Scores of Test Set:
Average Accuracy Score: 0.8272463768115942
Average Precision: 0.8272463768115942
Average Recall: 0.8272463768115942
Average F1-score: 0.8272463768115942
Average AUC: 0.7705534022394488



Confusion Matrix for Test data:
[[72 0]
[17 26]]

v. Comparing the results of the above methods, we can see that Semi-Supervised Learning/ Self-training method performs the best for this dataset with an accuracy score of 0.9526315789473684. Supervised Learning method takes the second place with the accuracy score of 0.9220289855072463. Next would be Spectral Clustering for training set with an accuracy score of 0.8566813509544787. Next would be Unsupervised learning for Test set with an accuracy score of 0.852463768115942. Next would be Unsupervised Learning for Training set with an accuracy score of 0.8497797356828194. The worst performance is by Spectral Clustering on Test set with an accuracy score of 0.8272463768115942.

2. **Active Learning Using Support Vector Machines**

   a. Downloading the banknote authentication Data Set and choosing 472 data points randomly as the test set, and the remaining 900 points as the training set.

   b. Repeat each of the following two procedures 50 times.

      i. **Passive learning with L1-penalized SVM:**

      As the code was taking too long to run for outer loop(i) = 50 & inner loop(j) = 90, I have run the loop for i=5 and j=9.

      Test error for iteration 0: [0.833, 0.896, 0.985, 0.985, 0.985, 0.987, 0.985, 0.983, 0.983]

      Test error for iteration 1: [0.979, 0.979, 0.979, 0.979, 0.979, 0.962, 0.962, 0.962, 0.962]

      Test error for iteration 2: [0.977, 0.977, 0.983, 0.983, 0.983, 0.983, 0.983, 0.983, 0.983]

      Test error for iteration 3: [0.983, 0.983, 0.983, 0.983, 0.983, 0.983, 0.987, 0.987, 0.987]

      Test error for iteration 4: [0.987, 0.989, 0.989, 0.987, 0.987, 0.992, 0.992, 0.989, 0.992]

      List of best parameters: [{'C': 21.544346900318846}, {'C': 1000.0}, {'C': 1000.0}, {'C': 1000.0}, {'C': 1000.0}, {'C': 0.464158883361278}, {'C': 0.464158883361278}, {'C': 1000.0}, {'C': 0.464158883361278}]

      ii. **Active learning with L1-penalized SVM:**

      As the code was taking too long to run for outer loop(i) = 50 & inner loop(j) = 90, I have run the loop for i=5 and j=9

      Test error for iteration 0: [0.782, 0.909, 0.987, 0.951, 0.975, 0.975, 0.987, 0.977, 0.979]

      Test error for iteration 1: [0.981, 0.989, 0.987, 0.981, 0.989, 0.981, 0.989, 0.981, 0.979]

      Test error for iteration 2: [0.989, 0.985, 0.981, 0.979, 0.981, 0.985, 0.983, 0.981, 0.989]

      Test error for iteration 3: [0.979, 0.985, 0.981, 0.987, 0.873, 0.983, 0.869, 0.975, 0.979]

      Test error for iteration 4: [0.979, 0.983, 0.977, 0.975, 0.917, 0.875, 0.977, 0.856, 0.917]

      List of best parameters are: [{'C': 10000.0}, {'C': 1000.0}, {'C': 100000.0}, {'C': 100000000.0}, {'C': 0.1}, {'C': 0.1}, {'C': 1.0}, {'C': 0.1}, {'C': 0.1}]

c. **Monte Carlo Simulation:**

Mean Test error for Passive learning: [0.958, 0.9714444444444443, 0.9816666666666666, 0.9843333333333333, 0.9893333333333333]

Mean Test error for Active learning: [0.9468888888888889, 0.984111111111111, 0.9836666666666667, 0.9567777777777776, 0.9395555555555557]