Machine Learning (CS60050)

# Assignment - 2

Unsupervised and Supervised Learning

Group 43
Umang Singla - 20CS10068
Utsav Mehta - 20CS10069

## DATASET DESCRIPTION -

1. Title: Lung Cancer Data

2. Number of Instances: 32

3. Number of Attributes: 57 (1 class attribute, 56 predictive)

4. Attribute Information:

- Attribute 1 is the class label.
- All predictive attributes are nominal, taking on integer values between 0-3

5. Missing Attribute Values: Attributes 5 and 39 (marked as ?)

6. Class Distribution: There are 3 classes with the following count of observations -

| | | |
|---|---|---|
| 1) | Class - 1: | 9 observations |
| 2) | Class - 2: | 13 observations |
| 3) | Class - 3: | 10 observations |

7. Dataset file - “lung-cancer.data”

8. Dataset download link - https://archive.ics.uci.edu/ml/datasets/Lung+Cancer

## 1. Unsupervised Learning

**Tasks:**

1. Apply PCA (select number of components by preserving 95% of total variance) (in-built function allowed for PCA).
2. Plot the graph for PCA.
3. Using the features extracted from PCA, apply K-Means Clustering. Vary the value of K from 2 to 8. Plot the graph of K vs normalized mutual information (NMI). Report the value of K for which the NMI is maximum. (in-built function not allowed for K-Means).

**Algorithms and Explanations:**

- Principal Component Analysis -
    - Is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.
    - In simpler terms, it reduces the attribute space by choosing a new set of attributes while maintaining the variance of the previous data.
    - Here, we use the inbuilt function PCA from the sklearn library to compute the new set of attributes and transform the dataset.

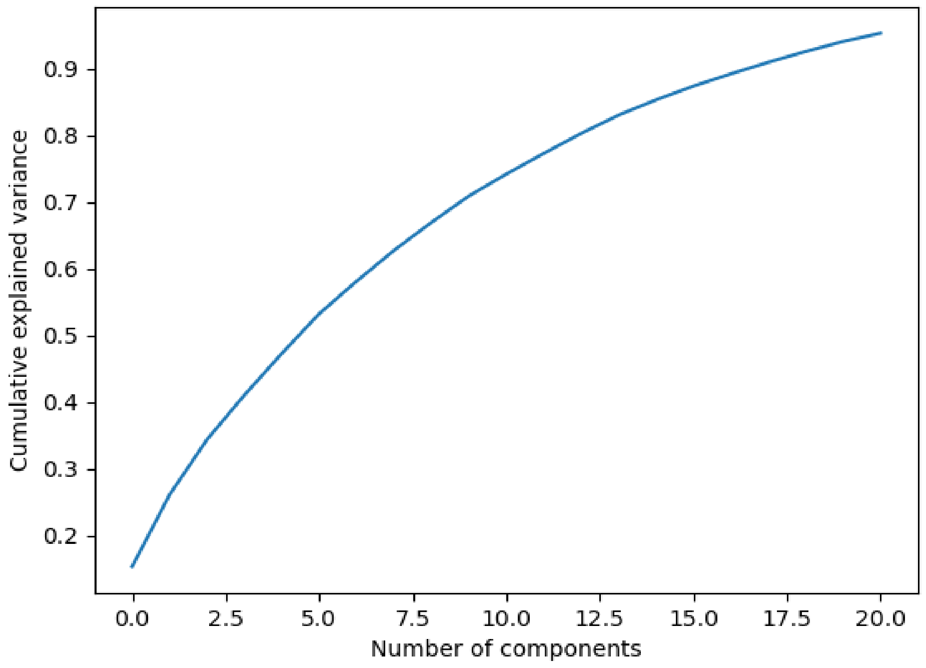
- K-Means Clustering -
    - Given a data set of items and features, we need to cluster the values into k-groups, and assign appropriate labels to each of the clusters.
    - In short, the algorithm will categorize the items into k groups or clusters of similarity.
    - Here, we use Euclidean Distance as a measure of similarity.

- The algorithm used is –
  - For a certain number of iterations (maximum 100), we repeat the following (the initial cluster representatives are randomly chosen) :
    1. Update the cluster assignment of the data values based on the new cluster representatives
    2. Update the cluster representatives as means of the clustered data
    3. Update the cluster labels as the mode of the labels of clustered data, repeat
  - The above loop repeats till the convergence criterion is met, or maximum number of iterations are reached.
  - The convergence criterion used is - If the average Euclidean Distance between cluster representatives and cluster members in two consecutive iterations is less than 0.0001%, mark convergence.

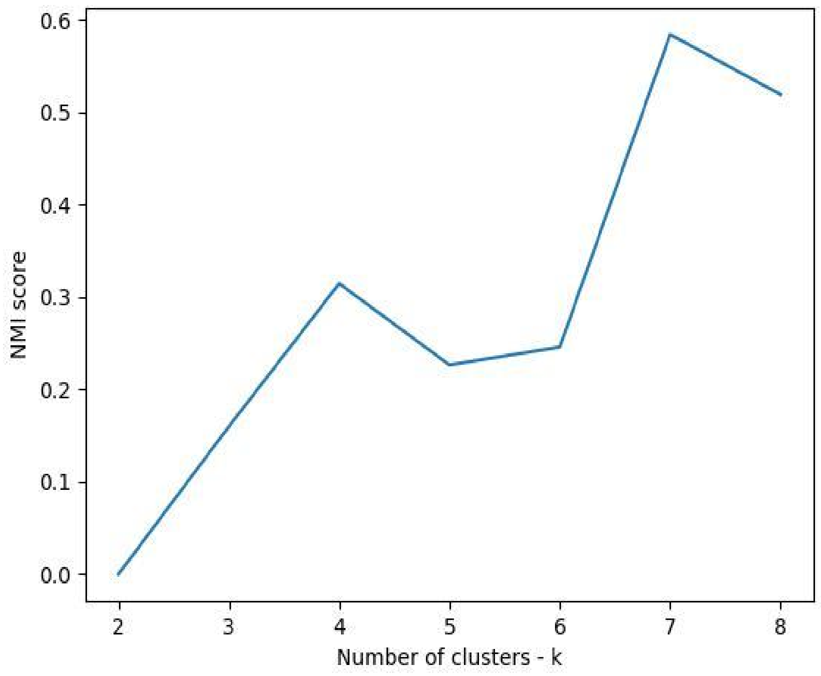
- Normalized Mutual Information -
  - Considered as a good measure to compute the quality of clustering.
  - It is an external measure because we need the class labels of the instances to determine the NMI.
  - Since it's normalized we can measure and compare the NMI between different clustering distributions having different numbers of clusters.
  - The algorithm used for computing NMI is:
    - Compute the individual cluster [ H(C) ] and label [ H(Y) ] probabilities
    - Compute the entropy of class labels within each cluster [ H(Y|C) ]
    - Compute the Mutual Information [ I(Y; C) = H(Y) - H(Y|C) ]
    - Finally, compute the normalized mutual information as - NMI = 2 x I(Y; C) / [ H(Y) + H(C) ]

**Results:**

1. **The graph of Principal Component Analysis -**



2. **The graph of K vs normalized mutual information (NMI)**



3. **The values of NMI for various values of k -**

| K(number of clusters) | NMI |
|---|---|
| **2** | **0.0007** |
| **3** | **0.1488** |
| **4** | **0.3202** |
| **5** | **0.2219** |
| **6** | **0.2694** |
| **7** | **0.5838** |
| **8** | **0.5210** |

Clearly, the maximum Normalized Mutual Information is obtained for a value of K = 7.

**Libraries Used:**

- Numpy - for computations
- Pandas - for data extraction from dataset
- Sklearn - for PCA
- Matplotlib - for plotting graphs

**Corresponding Files:**

q1.py, q1_Figure1.png, q1_Figure2.png, q1_output.txt

## 2. Supervised Learning

**Tasks:**

1. Normalize the data using Standard Scalar Normalisation. Randomly divide the Dataset into 80% for training and 20% for testing. Encode categorical variables using appropriate encoding method (in-built function not allowed for normalization, sampling and encoding).
2. Implement the binary SVM classifier using the following kernels: Linear, Quadratic, Radial Basis function. Report the accuracy for each. (in-built function allowed).
3. Build an MLP classifier (in-built function allowed). for the given dataset. Use stochastic gradient descent optimiser. Keep learning rate as 0.001 and batch size of 32. Vary the number of hidden layers and number of nodes in each hidden layer as follows and report the accuracy of each:

   a. 1 hidden layer with 16 nodes

   b. 2 hidden layers with 256 and 16 nodes respectively.

4. Using the best accuracy model from part 3, vary the learning rate as 0.1, 0.01, 0.001, 0.0001 and 0.00001. Plot the learning rate vs accuracy graph.
5. Use forward selection method on the best model found in part 3 to select the best set of features. Print the features.
6. Apply ensemble learning (max voting technique) using SVM with quadratic, SVM with radial basis function and the best accuracy model from part 3. Report the accuracy.
7. Prepare a report including all your results.

**Algorithms and Explanations:**

- **Standard Scaler Normalization:**

StandardScaler scales a feature to unit variance after subtracting the mean to standardize it. By the standard deviation, we may calculate the unit variance of all the values.

$$z = \frac{x - \mu}{\sigma}$$

$$where\ \mu = mean\ and\ \sigma = standard\ deviation$$

- **Binary SVM Classifier:**

  Support Vector Machines are machine learning algorithms that are employed for regression and classification tasks. One of the effective machine learning methods for outlier detection, regression, and classification is the SVM. An SVM classifier creates a model by categorizing fresh data points into one of the predetermined groups. It can be thought of as a non-probabilistic binary linear classifier as a result.

  To implement SVM we have used an inbuilt function of the sklearn library, svm.SVC which takes a type of kernel as input.

- **MLP Classifier:**

  Multi-layer Perceptron classifier, or MLPClassifier, is connected to a neural network by the name itself. MLPClassifier uses an underlying Neural Network to carry out the classification task, unlike other classification methods like Support Vectors or Naive Bayes Classifier.

  To implement MLP Classifier we have used an inbuilt function from **sklearn.neural_network** library which takes the size of the hidden layers as an input.

- **Forward Selection Method:**

  An empty set of features is used as the starting point for the iterative process of forward selection. It keeps adding features and evaluating performance after each iteration to see if the performance is getting better or not. The procedure is repeated until the model's performance is not improved by the inclusion of a new variable or feature.

- **Ensemble Learning (Max Voting Technique):**

  For categorization issues, the max voting approach is typically utilised. With this method, predictions are made for each data point using a variety of models. Each model's predictions are regarded as a "vote." The majority of the models' forecasts serve as the basis for the final projection.

**Results:**

- **Binary SVM Classifier:**

| **Kernel Type** | **Accuracy (in %)** |
|---|---|
| Linear Function | 57.14 |
| Quadratic Function | 28.57 |
| Radial Basis Function | 14.28 |

- **MLP Classifier:**

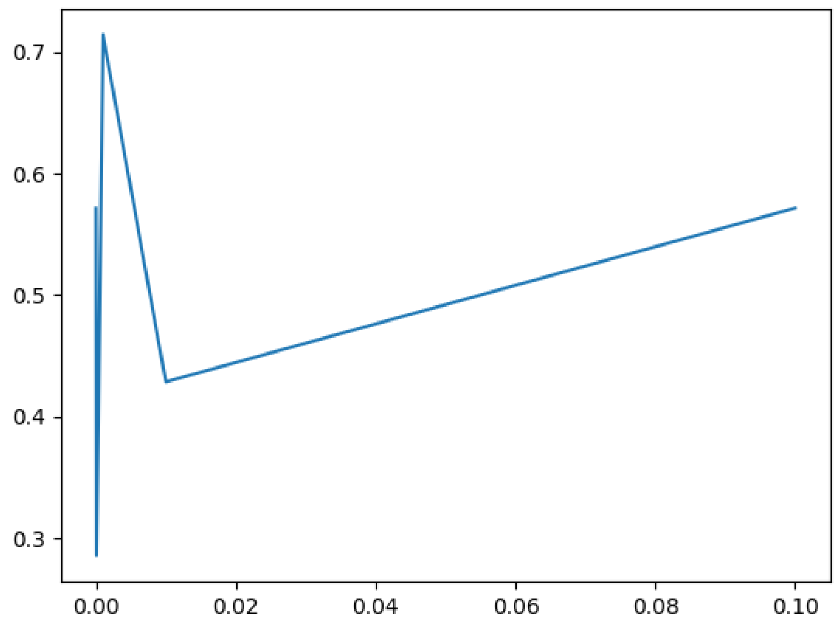
  **Results for MLP Classifier with learning rate as 0.001 and batch size of 32:**

| **No of hidden layers** | **Accuracy (in %)** |
|---|---|
| 1 hidden layer with 16 nodes | 85.71 |
| 2 hidden layers with 256 and 16 nodes | 42.85 |

**Results for MLP Classifier with different learning rates:**

| Learning Rate | Accuracy (in %) |
|---|---|
| 0.1 | 57.14 |
| 0.01 | 42.85 |
| 0.001 | 71.42 |
| 0.0001 | 28.57 |
| 0.00001 | 57.14 |

**Learning Rate vs Accuracy Curve**



- **Forward Feature Selection**

  Best Features are: [39, 36, 18, 22]

- **Ensemble Learning (Max Voting Technique)**

Accuracy of Ensemble learning (Max voting technique) using SVM with Quadratic, SVM with Radial Basis Function and MLP Classifier with 1 hidden layer of 16 neurons:  28.57%.

**Libraries Used:**

- Numpy - for computations
- Pandas - for data extraction from dataset
- Sklearn - for MLP Classifier and binary SVM Classifier
- Matplotlib - for plotting graphs

**Corresponding Files:**

q2.py,   q2_Figure.png,  q2_output.txt