PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

Machine Learning (CS60050) - Assignment 2 Report

Group 3

Kanchi Mohan Krishna - 20CS10030
Lav Jharwal - 20CS30031

Dataset:

Dataset Description:

The data describes 3 types of pathological lung cancers.

Attribute Information:

Label: Column 1

All predictive attributes are nominal, taking on integer values 0-3

Tasks

Unsupervised Learning

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).

PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

Unsupervised learning is a type of algorithm that learns patterns from untagged data. The hope is that through mimicry, which is an important mode of learning in people, the machine is forced to build a concise representation of its world and then generate imaginative content from it.

The given data has some missing values with '?' in it, these are taken care of by filling them with the mean of that column in Q1 and mode in Q2.

Principal Component Analysis

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D. PCA helps in finding a sequence of linear combinations of variables.

K-means Clustering:

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of predefined clusters that need to be created in the process.
It is a centroid-based algorithm, where each cluster is associated with a centroid.
The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.
Determines the best value for K center points or centroids by an iterative process.Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.
The functions mainly used are to recalculate centroid and clusters until there is no change in the centroids positions.
For determining the quality of clustering Normalized mutual information (NMI) is used, Since it's normalized we can measure and compare the NMI between different clusterings having different numbers of clusters.

PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

THE VARIANCE EXPLAINED BY PRINCIPAL COMPONENTS:

```
-------------------------------The variance explained by the principal components-------
Variance explained by principal component 1 : 0.15355519967801542
Variance explained by principal component 2 : 0.10766808336892833
Variance explained by principal component 3 : 0.08271753321100543
Variance explained by principal component 4 : 0.06675996633159514
Variance explained by principal component 5 : 0.062452214054500105
Variance explained by principal component 6 : 0.05959449987418653
Variance explained by principal component 7 : 0.04881807637651834
Variance explained by principal component 8 : 0.04623212480564807
Variance explained by principal component 9 : 0.041965884305189605
Variance explained by principal component 10 : 0.039532720662418114
Variance explained by principal component 11 : 0.03288634096833254
Variance explained by principal component 12 : 0.03091411425721769
Variance explained by principal component 13 : 0.030056348099540928
Variance explained by principal component 14 : 0.027461670449845713
Variance explained by principal component 15 : 0.022773792960126528
Variance explained by principal component 16 : 0.020707692277199662
Variance explained by principal component 17 : 0.018365650400085407
Variance explained by principal component 18 : 0.017342020897199672
Variance explained by principal component 19 : 0.016075991335655626
Variance explained by principal component 20 : 0.015107835338752602
Variance explained by principal component 21 : 0.01240686333046393
```

EXPLAINED VARIANCE vs NO. OF COMPONENTS:

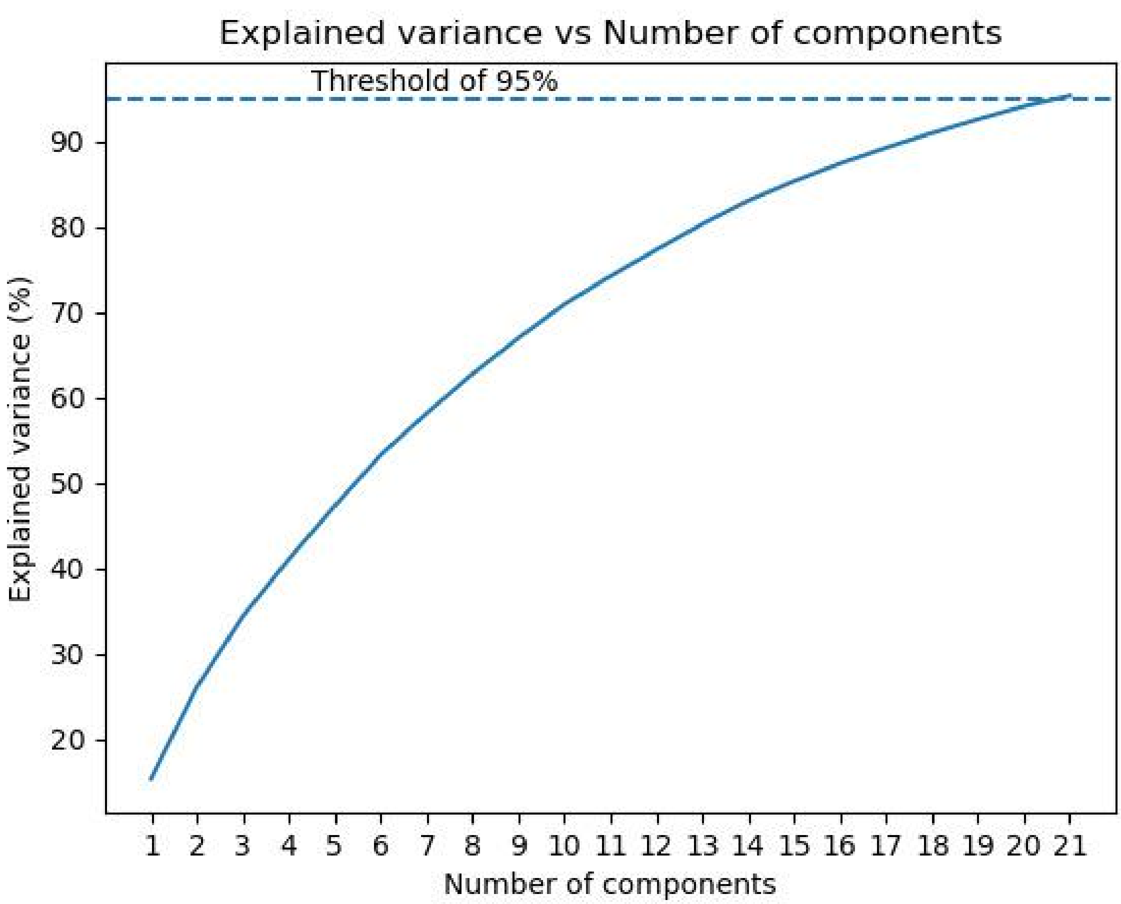
```
---------------------------------------------------------------------------------------
----------------------------Plot of Explained variance vs Number of components-------
---------------------------------------------------------------------------------------
-----------------------------------Cluster Representatives/Centriods-----------------
1 --> [-0.39767543 -0.74791648  0.1526375  -0.02265066 -0.05764312 -0.11282964
 -0.16341364  0.1033714  -0.03475558  0.10923647  0.05520867  0.10153733
  0.00686043  0.00717167  0.02728807 -0.0031937  -0.07238955  0.08161547
  0.02680294  0.06639954  0.00317327]
2 --> [ 2.1474473   4.038749   -0.8242425   0.12231358  0.31127284  0.60928007
  0.88243368 -0.55820557  0.18768012 -0.58987692 -0.29812683 -0.5483016
 -0.03704634 -0.03872703 -0.14735559  0.01724599  0.39090356 -0.44072353
 -0.14473586 -0.35855751 -0.01713568]
3 --> [-0.67147707  6.80456561 -1.14852203 -1.21586985  2.24679787  0.60303793
  0.45075518 -0.30094282  3.61091733 -1.52830185  0.75372288 -0.67561102
 -1.51912804 -1.07644797 -0.68347574 -1.24322267 -1.69751751  1.65434951
  0.09937032 -0.12518937  0.45862467]
---------------------------------------------------------------------------------------
```

PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

PLOT OF K vs NORMALIZED MUTUAL INFORMATION:

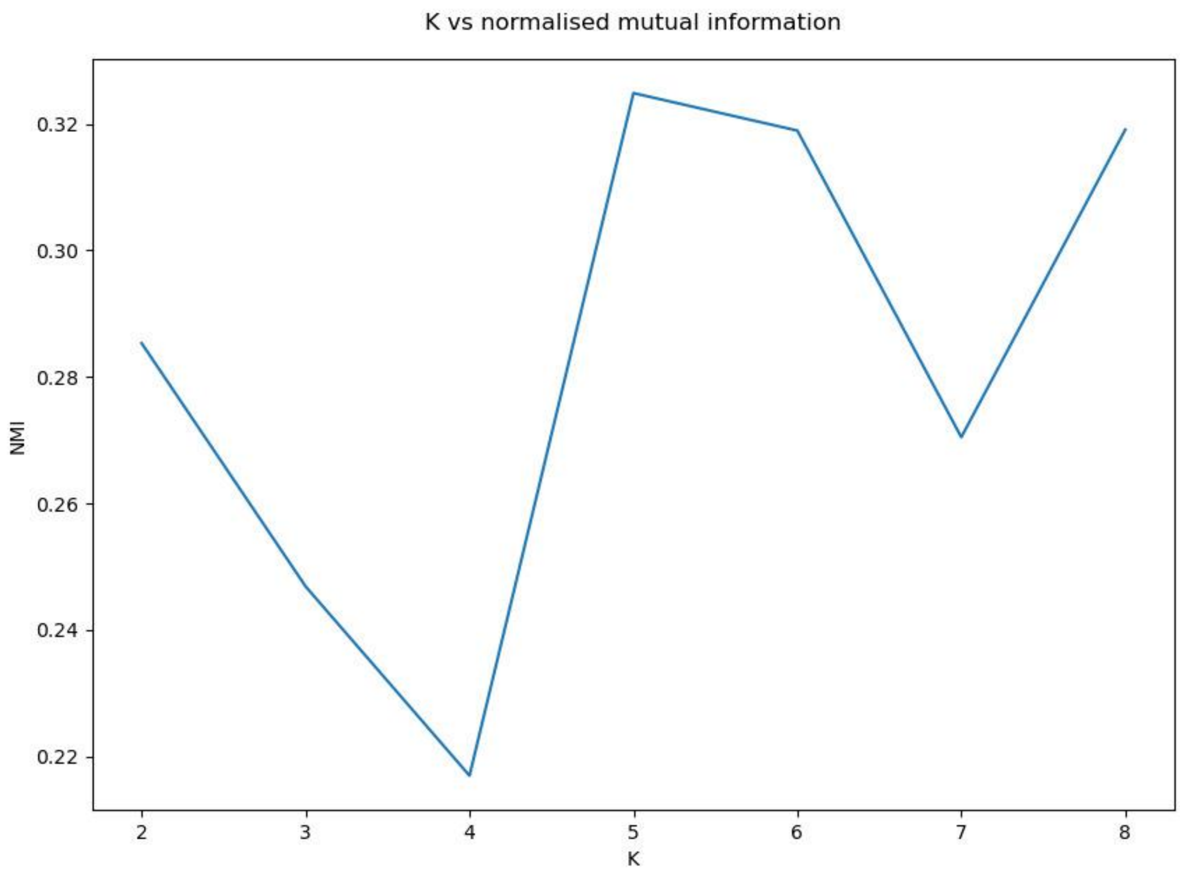
```
---------------------------------------------------------------------------------
---------------------------------Plot of K vs normalised mutual information---
---------------------------------------------------------------------------------
-----------------------------------------Optimal value of K-----------------
Optimal value of K: 6
---------------------------------------------------------------------------------
```

GRAPH FOR PCA:



PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

GRAPH FOR K vs NORMALIZED MUTUAL INFORMATION (NMI):



PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

Supervised Learning

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 methods (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 a 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.

PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

Standard Scalar Normalization

It scales each input variable separately by subtracting the mean and dividing by the standard deviation to shift the distribution to have a mean of zero and a standard deviation of one.

X = (x – mean) / std or we can use max_min normalization = X = (x-min)/(max-min)

Sampling

We divide data into testing and training, testing 20%.

Procedure

We have used both Binary SVM classifiers (with Linear, Quadratic, and Radial Basis Function as kernels) and Multi-Layer Perceptron classifiers with different learning rates for the assignment. The data used for training was in .csv format. We have shuffled the data. We considered a split of 80:20 for the Train and Test set.

Support Vector Classifier

- Binary SVM Classifier is a linear discriminant classifier.
- **Uses Vapnik's principle**: to never solve a more complex problem as a first step before the actual problem.
- After training the weight vector can be written in terms of training samples lying in class boundaries.
- The Primal problem for soft-margin hyperplanes and kernel function: Minimize w.r.t 'w'

$$L_p = \frac{1}{2}\|w\|^2 + C * \Sigma_t s^t - \Sigma_t \alpha^t[r^t (w^T \phi(x^t) + w_0) - 1 - s^t] - \Sigma_t \mu^t s^t$$

PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

Multi-Layer Perceptron

- It is a network of perceptrons arranged in a particular fashion, called the architecture of the model.
- We consider a multi-layered feed-forward network, this means there is no feed-back or loop in the network.

OUTPUT:

```
********************************************* PART 1 *********************************************
Performing standard scalar normalisation...
Randomly dividing the dataset into 80% training and 20% testing...
```

```
********************************************* PART 2 *********************************************
Training Linear Support Vector Machine
Accuracy of Linear SVM:  0.2857142857142857
Training Quadratic Support Vector Machine
Accuracy of Quadratic SVM:  0.5714285714285714
Training Radial Basis Function Support Vector Machine
Accuracy of Radial Basis SVM:  0.14285714285714285
Done!
```

```
********************************************* PART 3 *********************************************
Training MLP Classifier
With optimizer:  sgd  learning rate:  0.001  batch size:  32
Training MLP Classifier with 1 hidden layer of 16 neurons
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of MLP Classifier with 1 hidden layer of 16 neurons:  0.14285714285714285
Training MLP Classifier with 2 hidden layer of 256 neurons and 16 neurons
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of MLP Classifier with 2 hidden layer of 256 neurons and 16 neurons:  0.14285714285714285
Done!
```

PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY:

```
*********************************************** PART 4 ***********************************************
Best Model: MLP Classifier with 2 hidden layer of 256 neurons and 16 neurons
Training the above model with different learning rates
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.py:621:
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of MLP Classifier with learning rate:  0.1  is:  0.14285714285714285
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.py:621:
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of MLP Classifier with learning rate:  0.01  is:  0.2857142857142857
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.py:621:
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of MLP Classifier with learning rate:  0.001  is:  0.2857142857142857
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.py:621:
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of MLP Classifier with learning rate:  0.0001  is:  0.2857142857142857
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptron.py:621:
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of MLP Classifier with learning rate:  1e-05  is:  0.42857142857142855
Done!
```

```
  warnings.warn(
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perce
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perce
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Best Features are:  [5]
```

```
*********************************************** PART 6 ***********************************************
Applying Ensemble learning (Max voting technique) using SVM with Quadratic, SVM with Radial Basis Func
n layer of 16 neurons
/home/mohankrishna/miniconda3/lib/python3.9/site-packages/sklearn/neural_network/_multilayer_perceptro
ze` less than 1 or larger than sample size. It is going to be clipped
  warnings.warn(
Accuracy of Ensemble learning (Max voting technique) using SVM with Quadratic, SVM with Radial Basis F
dden layer of 16 neurons:  0.5714285714285714
```

Using SVM with Quadratic, SVM with Radial Basis Function and MLP Classifier with 1 hidden layer of 16 neurons: **0.2857142857142857**

PLOT THE GRAPH OF LEARNING RATE v/s ACCURACY: