INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Machine Learning Assignment - 2

Venkata Sai Suvvari 20CS10067
Shashank Goud Boorgu 20CS30013

1. Unsupervised Learning

Unsupervised Learning

Unsupervised learning refers to the use of artificial intelligence (AI) algorithms to identify patterns in data sets containing data points that are neither classified nor labeled.
The algorithms are thus allowed to classify, label and/or group the data points contained within the data sets without having any external guidance in performing that task.
In other words, unsupervised learning allows the system to identify patterns within data sets on its own.
In unsupervised learning, an AI system will group unsorted information according to similarities and differences even though there are no categories provided.
Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. Additionally, subjecting a system to unsupervised learning is one way of testing AI.

Principal Component Analysis (PCA)

Principal component analysis (PCA) is a popular technique for analyzing large datasets containing a high number of dimensions/features per observation, increasing the interpretability of data while preserving the maximum amount of information, and enabling the visualization of multidimensional data.
Formally, PCA is a statistical technique for reducing the dimensionality of a dataset.
This is accomplished by linearly transforming the data into a new coordinate system where (most of) the variation in the data can be described with fewer dimensions than the initial data.

K-Means Clustering

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.
This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances).

Normalized Mutual Information

Normalized Mutual Information (NMI) is a normalization of the Mutual Information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation).
Mutual information is a quantity that measures a relationship between two random variables that are sampled simultaneously. In particular, it measures how much information is communicated, on average, in one random variable about another.

$NMI(Y,C) = \frac{2*I(Y;C)}{[H(Y)+H(C)]}$
where,

i. Y = Class Labels

ii. C = Cluster Labels

iii. H(.) = Entropy

iv. I(Y;C) = Mutual information between Y and C → H(Y) - H(Y|C)

v. Note: All to the log base-2

Procedure

i. The given data is dimensions are reduced using PCA preserving 95% variance.

ii. Using the extracted features from PCA, K-Means clustering applied and the data is clustered for different values of varying from 2 to 8.

iii. The normalized mutual information is calculated for each value of k during k-means clustering.

iv. The plots are generated and the results are printed into the q1.txt.

Dataset Description

1. 3 classes of 50 instances of each plant is given.
2. Predicted attribute: class of iris plant
3. Label: Column 5
4. Attribute Information:

    i sepal length in cm

    ii sepal width in cm

    iii petal length in cm

    iv petal width in cm

    v Class:
    –Iris Serota
    –Iris Versicolor
    –Iris Virginica

Major Functions

i. **PCA():** An instance of PCA class with variance 95%.

ii. **fit():** This a class method of PCA to fit the feature data.

iii. **transform():** This is a class method of PCA to extract dimensions preserving the variance given in initialization.

iv. **K-Means:** This is a class.

  a. **fit():** This trains the model from the data and given k, to form k clusters.

  b. **predict():** To find the nearest centroid to the example and assign the class to it.

c. **compute_centroids():** The centroids are updated from the classes assigned.

Helper Functions

i. **k_means_clustering():** This trains model and try to predict values to find the normalized mutual information.

ii. **K-Means:**

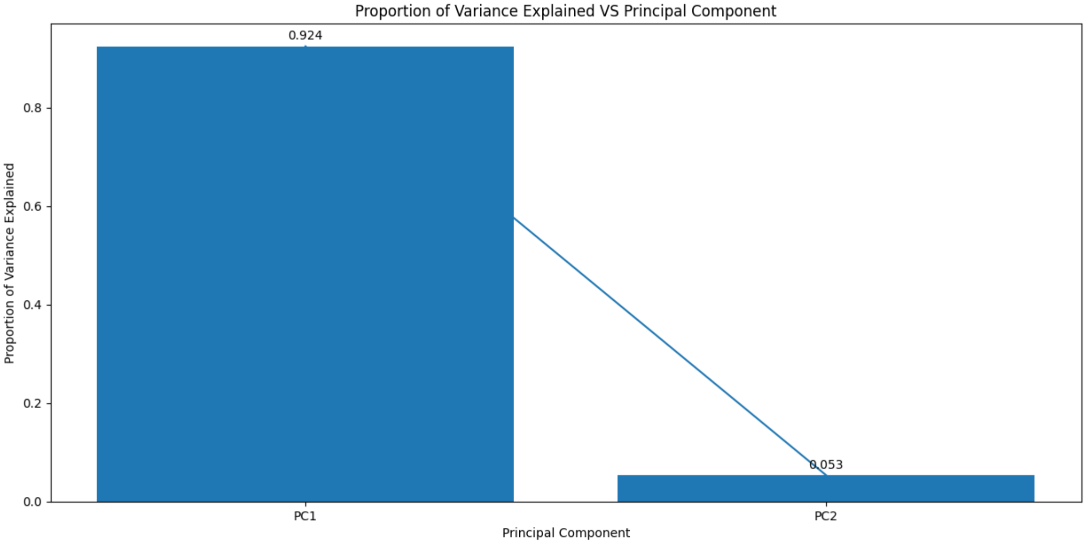
a. **initialise_centroids():** Randomly iniitalizes the centroids from the data given.

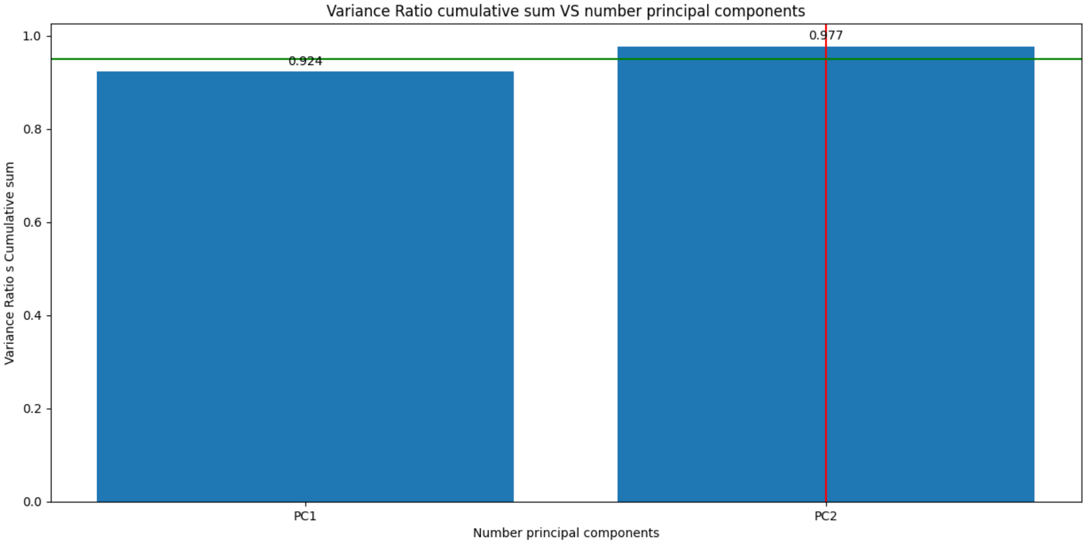
b. **compute_distance():** Computes the distance from all the clusters centroids.

c. **find_closest_cluster():** Gets the class of the cluster with minimum distance.

d. **compute_sse():** This computes the sum of squared errors with the predicted class and the original class.

Results for PCA





Results for K-Means vs NMI

i. NMI for k = 2: 0.653903316320765

ii. NMI for k = 3: 0.5871300165520078

iii. NMI for k = 4: 0.5678530958500374

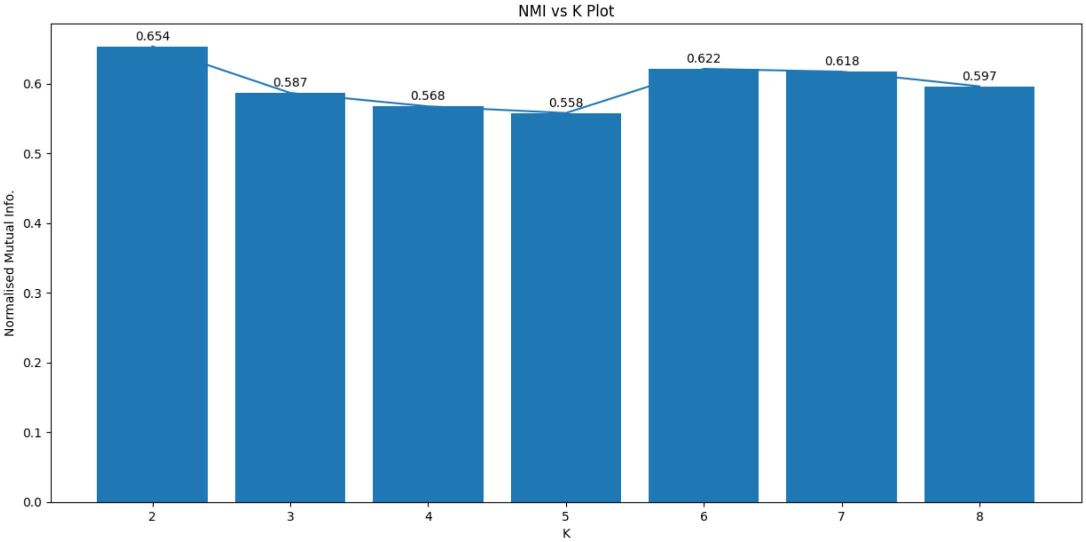
iv. NMI for k = 5: 0.5582521941880481

v. NMI for k = 6: 0.6216166326197388

vi. NMI for k = 7: 0.6176225403429995

vii. NMI for k = 8: 0.5965567445744009

Value of **k** for which the value of **NMI is maximum** is **2** with NMI **0.653903316320765**.



2. Supervised Learning

Procedure

1. The data set we used for this project is iris.data. The data is in csv format.
2. We have implemented a Binary SVM classifiers using Linear, Quadratic and Radial Basis Function kernels.
3. We have built an MLP classifier and used stochastic gradient descent as optimiser.
4. We used backward Elimination method to find the best set of features.
5. We applied Ensemble learning using max voting technique.

Major Functions

i. **standard_scalar_normalization():** It Standardizes the input data by subtracting the mean and then dividing by standard deviation. The output data will now have mean as 0 and variance as 1.

$$X' = \frac{X-\mu}{\sigma}$$

ii. **binary_SVM_Classifier():** This function takes training data and testing data as inputs. It then trains the SVM classifier using the training data. Then it tests the testing data and computes accuracy. It finally returns this accuracy.

iii. **MLP_classifier :** This function takes training data and testing data as inputs. It then trains the MLP classifier using the training data. Then it tests the testing data and computes the accuracy. It finally returns this accuracy.

iv. **backward_elimination():** This function takes a model, training data and testing data as inputs. It then trains the classifier using the training data by removing one feature at a time. Then it tests the testing data and computes accuracy. If it is above a threshold then we remove the feature obtained for which we get maximum accuracy. We call again the function with this modified set of features. Finally it returns the best set of features.

v. **ensemble_learning_max_voting():** This function takes as inputs a set of models and training and testing data. It trains these models using training data. Then it tests the test data and uses max voting technique and computes the accuracy. Finally it returns the accuracy.

Helper Functions

i. **categorical_encoding():** It converts categorical data into integer format.

ii. **sample_split():** It takes data and split size as inputs. It splits the data into two parts one with split size and remaining into other part.

iii. **seperate_X_Y():** It seperates the data into two parts X and Y. It then applies standard scalar normalization on X.

iv. **compute_Accuracy():** It takes as inputs Y_prediction and Y_test. It computes the number of correct predictions and returns the accuracy.

Support Vector Machine

1. A Support vector machine tries to find out a hyperplane with maximum margin in an N dimensional space that distinctly classifies the data.
2. It uses Vapnik's Principle : to never solve a more complex problem as a first step before the actual problem
3. It is a linear discriminant classifier
4. After training the weight vector can be written in terms of training samples lying in class boundaries

5. The Primary problem is to minimize the below equation w.r.t 'w'

$$L_p = \frac{1}{2}||w||^2 + C\sum_t s^t - \sum_{t=1}^{N} \alpha^t[r^t(\mathbf{w}^T\mathbf{x}^t + w_0) - 1 + s^t] - \sum_t \mu^t s^t$$

Multi Layer Perceptron

1. It is fully connected dense layers, which transform any input dimension to the desired dimension.
2. A multi-layer perception is a neural network that has multiple layers.
3. We are considering multi layered feed forward network, this means there are no feed backs or loops in the network
4. In this we define an Error function and try to minimize it w.r.t **W**
5. We can use Stochastic Gradient Descent technique for the minimization.

$$J_n(\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N} ||O_i - F(X_i;\mathbf{W})||^2$$

$$\mathbf{W}_i = \mathbf{W}_{i-1} + \eta(i)(O_k - F(X_k;\mathbf{W}_{i-1}))\nabla F(X_k;\mathbf{W}_{i-1})$$

Ensemble Learning

1. It combines the predictions from two or more models
2. Max Voting, Averaging and Weighted Averaging are few ensemble techniques.
3. In Max Voting, each base model makes a prediction and votes for each sample. Only the sample class with the highest votes is included in the final predictive class.
4. In Averaging, we take the average of all the predictions made by each model. With this value we make the final prediction
5. In Averaging, we take the weighted average of all the predictions made by each model. With this value we make the final prediction

Results

i. **Binary SVM Classifier**

   a. Accuracy using Linear Kernel : **0.9333333333333333**

   b. Accuracy using Quadratic Kernel : **0.8333333333333334**

   c. Accuracy using Radial Basis Function Kernel : **0.8666666666666667**

ii. **MLP Classifier**

   a. Accuracy with 1 hidden layer with 16 nodes **0.8333333333333334**

   b. Accuracy with 2 hidden layer with 256 and 16 nodes **0.8666666666666667**

   c. The best accuracy model has 2 hidden layers with 256 and 16 nodes respectively.

iii. **Learning rate vs Accuracy**

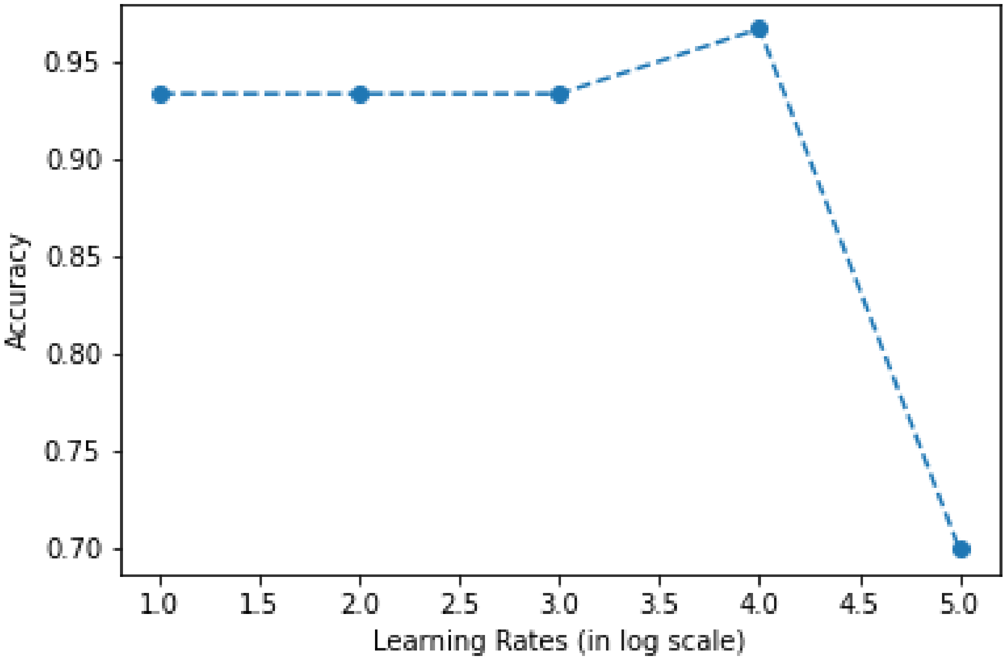
   a. Accuracy with learning rate as 0.100000 using 2 hidden layer is **0.8666666666666667**

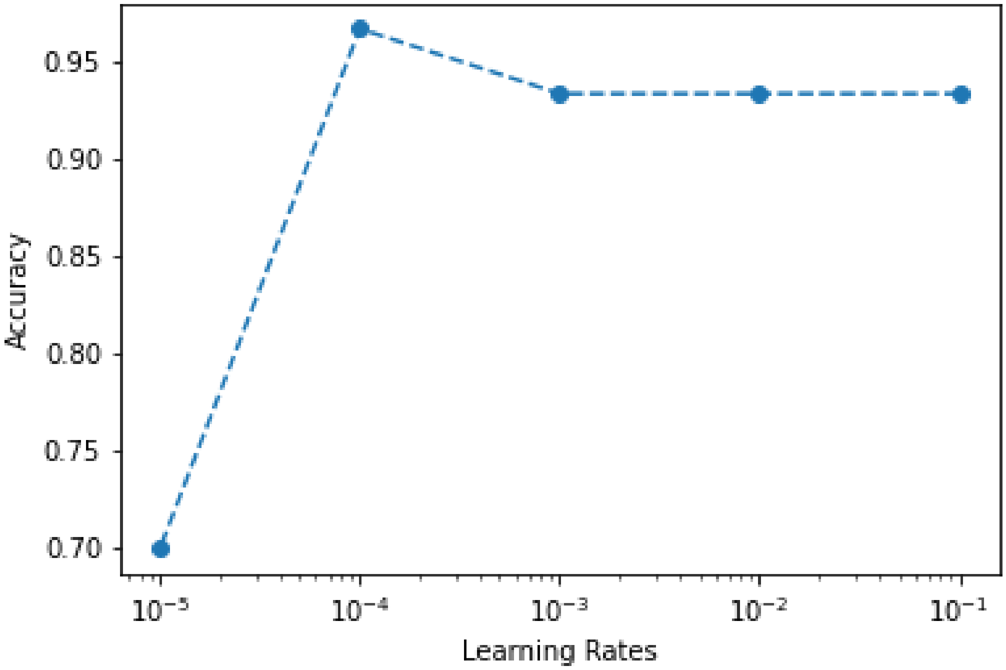
   b. Accuracy with learning rate as 0.010000 using 2 hidden layer is **0.8666666666666667**

   c. Accuracy with learning rate as 0.001000 using 2 hidden layer is **0.9**

   d. Accuracy with learning rate as 0.000100 using 2 hidden layer is **0.8333333333333334**

   e. Accuracy with learning rate as 0.000010 using 2 hidden layer is **0.7**


Accuracy
0.95
0.90
0.85
0.80
0.75
0.70
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Learning Rates (in log scale)


Accuracy
0.95
0.90
0.85
0.80
0.75
0.70
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
Learning Rates

iv. **Backward Elimination Method**

   a. The best set of features is Petal Width

   b. Accuracy with the above features is **0.6666666666666666**

v. **Ensemble Learning**

   a. Accuracy using ensemble learning with the models SVM Quadratic , SVM Radial Basis Function , MLP with 2 hidden layers is **0.8666666666666667**