# CS60050: MACHINE LEARNING

**Assignment 2**

**Unsupervised Learning**
**Supervised Learning**

Group 14
Kulkarni Pranav Suryakant - 20CS30029
Sidharth Vishwakarma - 20CS10082

**Dataset**

This data set is the result of a chemical analysis of three different cultivars grown in the same region of Italy. The amount of 13 constituents found in each of the three types of wines were determined by the analysis.

Label: Column 1
Class Distribution:

| **Class Number** | **Number of instances** |
|---|---|
| 1<br>2<br>3 | 59<br>71<br>48 |
| | Total = 178 |

**Attributes:** There is a total of 13 attributes

1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

# Unupervised 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 normalised mutual information (NMI). Report the value of K for which the NMI is maximum. (in-built function not allowed for K-Means).
4. Prepare a report including all your results.

## Algorithm

### Principal Component Analysis (PCA)

- Standardize each variable to unit variance and zero mean.
- Calculate the covariance matrix.
- Calculate the eigenvectors and eigenvalues of the covariance matrix.
- Reduce dimensionality and form a feature vector.
- Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest.
- The according to choose the principal components corresponding to the variance that you want.

### K-Means Clustering

- Let $x_1, x_2, \ldots, x_N \in R^n$, be the feature vectors of the given data.
- Initialize the list of k cluster representatives $z_1, \ldots, z_k$ to some random vectors from the set of feature vectors.
- Repeat until convergence
    - Cluster assignment based on cluster representatives.
    - Update cluster representatives.
- Cluster assignment is based on the distance norm of the cluster representative from the feature vector.
- Cluster representatives are updated based on the mean of the feature vectors belonging to that cluster number.

## Pseudo Code

KMeans (*Feature_Attributes*)

Initialize the *Centroids* to some random features from the *Feature_Attributes*.

Iterate unti convergence.

Calculate the euclidean distance of the *Feature_Attributes* from the *cluster_representatives*.

Update the cluster assignment of the *Feature_Attributes* based on euclidean distance.

Update the *cluster_representatives* by taking the average over the *Feature_Attributes* belonging to the respective *cluster_represenatives*

## Important Terms & Definitions

- Normalized Mutual Information:

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

  Where,
  1) Y = class labels
  2) C = cluster labels
  3) H(.) = Entropy
  4) I(Y;C) = Mutual Information b/w Y and C

- Mutual information is given as:
  – $I(Y;C) = H(Y) - H(Y|C)$
  – We already know H(Y)
  – H(Y|C) is the entropy of class labels within each cluster

## Procedure

- Load the dataset into the memory using the padas read_csv function.
- Split the features and labels into different numpy arrays.
- Perform the standardization of the data to unit variance and zero mean.
- Apply the Principal Component Analysis on the standardized dataset while preserving 95% of total variance.
- Plot the Cumulative Explained Varaiance v/s Number of Principal Components and save the plot (as elbow_plot.png).
- Apply K-Means Clustering algorithm on the dataset with Principal Components derived.
- Plot the graph of K v/s Normalized Mutual Information for K varying from 2 to 8.
- Report the value of K where maximum NMI is obtained.

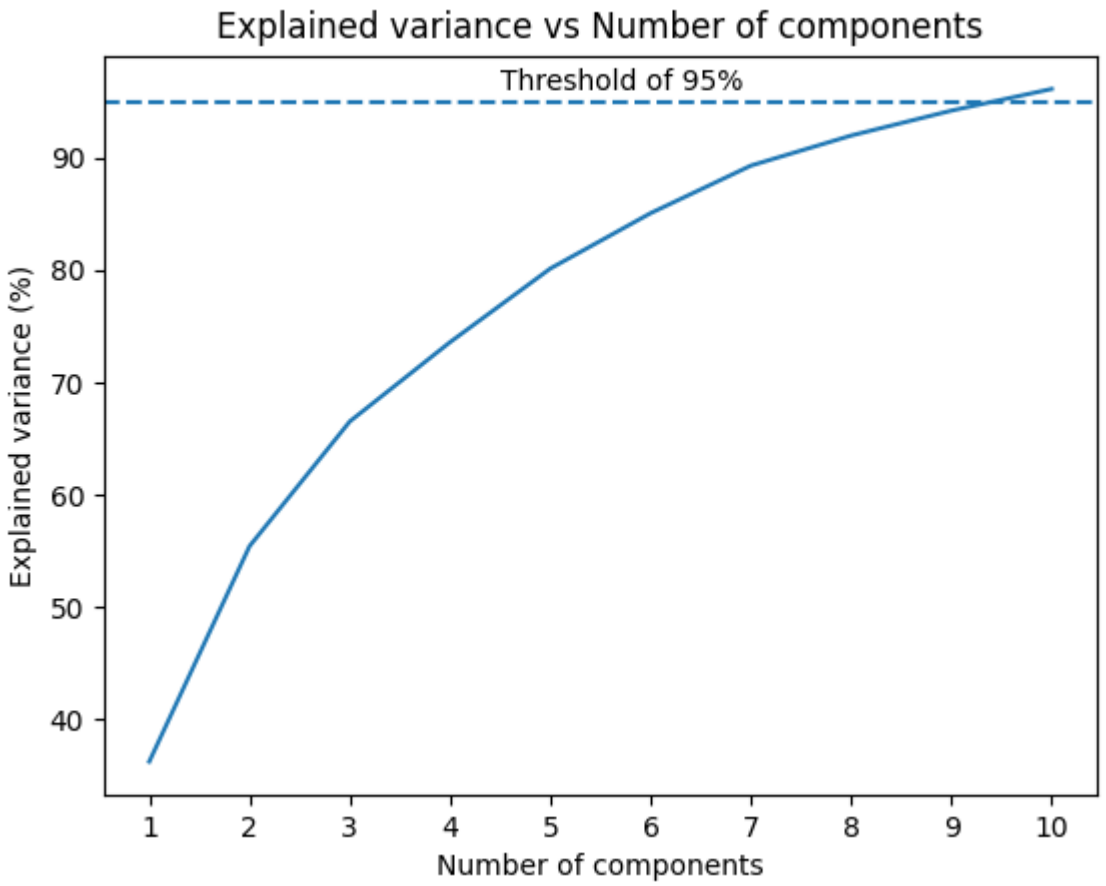
## Results

- Percentage of Explained Variance v/s Number of Principal Components:

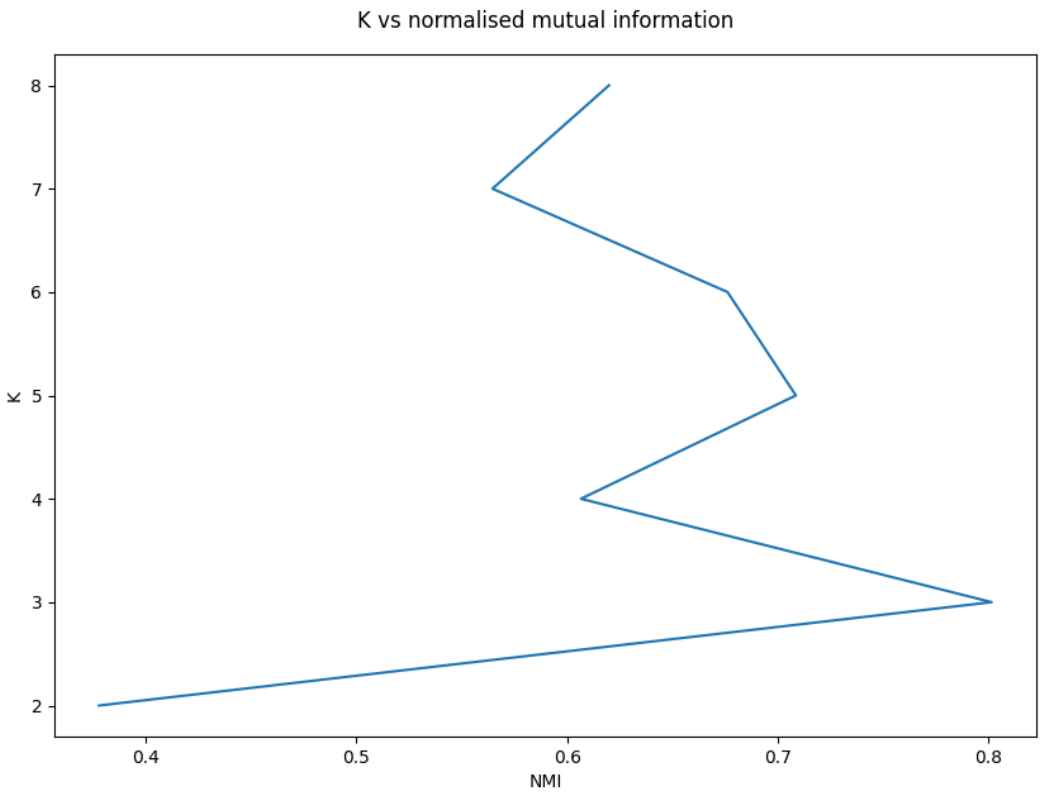
| No. of Principal Components | Cumulative Explained Variance (%) |
|---|---|
| 1 | 36.198848099926334 |
| 2 | 19.20749025700895 |
| 3 | 11.123630536249976 |
| 4 | 7.069030182714037 |
| 5 | 6.563293679648602 |
| 6 | 4.935823319222553 |
| 7 | 4.2386793226233184 |
| 8 | 2.6807489483788707 |

| | |
|---|---|
| 9 | 2.2221534047897163 |
| 10 | 1.9300190939440777 |

- Plot (Explained Variance v/s Number of Principal Components):



- Plot (K v/s Normalized Mutual Information):



# Supervised Learning

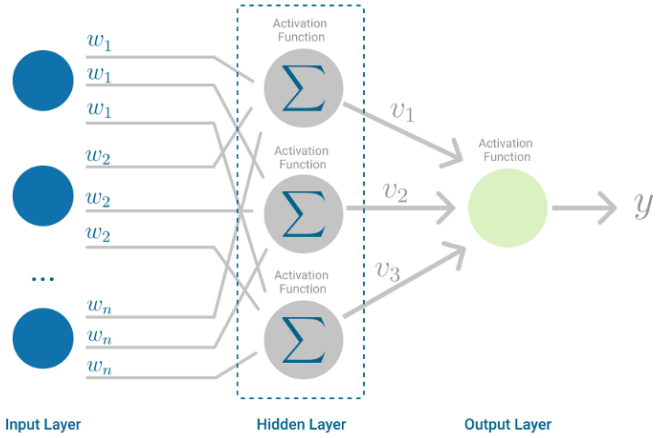
## A. Tasks

1. Normalize the data using **Standard Scalar Normalisation**. Randomly divide the Dataset into 80% for training and 20% for testing. Encode categorical variables using the 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, and Radial Basis functions**. Report the accuracy for each. (**in-built function allowed**).
3. Build an **MLP classifier** (in-built function allowed). for the given dataset. Use s**tochastic gradient descent** optimizer. **Keep the learning rate at 0.001 and a batch size of 32** Vary the number of hidden layers and nodes in each hidden layer as follows and report the **accuracy** of each:
    a. 1 hidden layer with 16 nodes
    b. 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 the **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.

## B. Algorithm:

One of the simplest methods of combining predictions from multiple machine learning algorithms is **max-voting**, which is commonly used for classification problems. Each base model predicts and votes for each sample in max-voting. The final predictive class includes only the sample class with the most votes.

- **SVM classifier** using linear, quadratic, and radial basis function kernels is implemented.
  Given a training dataset of n points of the form $(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)$, where $y_i$ is the class label to which the $x_i$ belongs.
  SVM finds the "**maximum-margin hyperplane**" that divides the group of points $x_i$ for which the $y_i$=class label is defined so that the distance between the hyperplane and the nearest point $x_i$ from either group is maximized.
- **MLP classifier** implemented using 1 hidden layer with 16 nodes and 2 hidden layers with 256 and 16 nodes. The accuracy of both models is compared and the best one is chosen for voting.



- **Ensemble learning** (max voting technique) applied using SVM with quadratic, SVM with radial basis function, and the best accuracy model MLP classifier.

## C. Procedure

### 1. Data Preprocessing:

a. There were no missing attribute values in the dataset.

b. The data is normalized using Standard Scalar Normalization. Every feature is standardized by subtracting the mean and then scaling to unit variance
If μ is the mean (average) and σ is the standard deviation from the mean, sample standard scores (also known as z scores) are calculated as follows:

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

c. The data are randomly divided into 80% for training and 20% for testing.

d. Categorial variables are encoded into 0, 1, and 2 for class labels 1, 2, and 3 respectively.

### 2. SVM Classifier:

a. To implement the SVM classifier with linear kernel **`sklearn.svm.LinearSVC`** function from the sci-kit-learn library is used.

b. For quadratic kernel **`sklearn.svm.SVC`** with parameters **`kernel=poly, degree=2`** is used.

c. For radial basis function kernel **`sklearn.svm.SVC`** with parameters **`kernel='rbf'`** is used.

### 3. MLP Classifier:

a. To implement the MLP classifier **`sklearn.neural_network.MLPClassifier`** function from the sci-kit-learn library is used with the following parameters:

i. **solver = 'sgd'**: stochastic gradient descent optimizer.

ii. **batch_size = 32**: Size of mini-batches for stochastic optimizers.

iii. **learning_rate = 0.001**: Learning rate schedule for weight updates.

iv. **hidden_layer_sizes = (16,)** for 1 hidden layer with 16 nodes and **(256, 16)** for 2 hidden layers with 256 and 16 nodes respectively.

b. After obtaining the MLP classifier with best-hidden layers learning rate is varied as 0.1, 0.01, 0.001, 0.0001, and 0.00001

### 4. Forward Selection Method:

Steps to perform forward feature selection:

a. Train n model using each feature (n) individually and check the performance

b. Choose the variable which gives the best performance

c. Repeat the process and add one variable at a time

d. Variable producing the highest improvement is retained

e. Repeat the entire process until there is no significant improvement in the model's performance.

### 5. Ensemble Learning (max voting technique):

a. Applied ensemble learning (max voting technique) using SVM with quadratic, SVM with radial basis function, and the best accuracy model from MLP Classifier with 2 hidden layers.

## D. Results

### 1. SVM Classifier:

Using **`sklearn.svm.LinearSVC`** for linear kernel and **`sklearn.svm.SVC`** for quadratic and radial basis function kennel following accuracies were observed:

| Kernel | Accuracy |
|---|---|
| 1. Linear | 97.22% |
| 2. Quadratic | 86.11% |
| 3. Radial Basis Function | 100.00% |

### 2. MLP Classifier:

Function: **`sklearn.neural_network.MLPClassifier`**

Parameters:

a. solver: **`sgd`**

b. batch_size: **`32`**

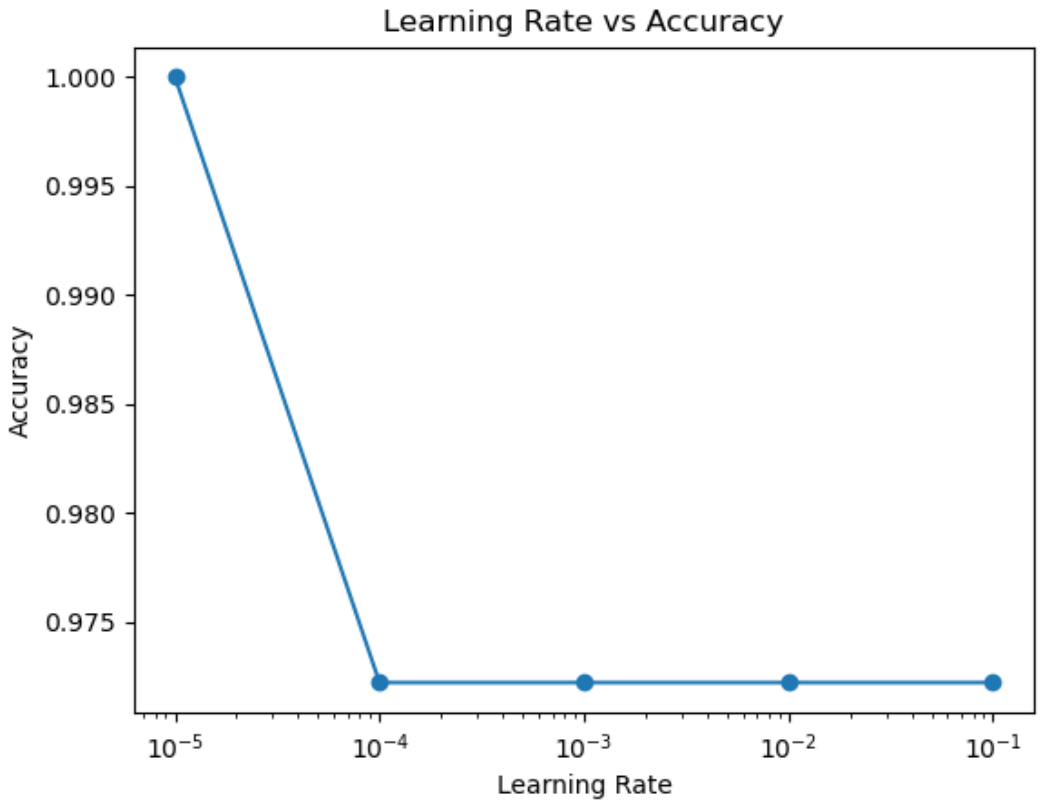
c. learning_rate: **`0.001`**

| Hidden_layer_sizes | Accuracy |
|---|---|
| 1. 1 hidden layer with 16 nodes | 94.44% |
| 2. 2 hidden layers with 256 and 16 nodes respectively | **97.22%** |

Model with 2 hidden layers with 256 and 16 nodes is better than the model with 1 hidden layer.

### 3. Variation of learning rates for MLP classifier:

For 2 hidden layer model learning rate varied as 0.1, 0.01, 0.001, 0.0001, and 0.00001.



4. **Forward Selection Method:**
   When the forward selection method is applied to the 2 hidden layer MLP classifier model following features are returned as the best features.

| Iteration | Best Features | Accuracy |
|---|---|---|
| 1st | Flavanoids | 75.00% |
| 2nd | Flavanoids, Color intensity | 94.44% |
| 3rd | Flavanoids, Color intensity, Alcohol | 97.22% |
| 4th | Flavanoids, Color intensity, Alcohol, Proline | 100.00% |

   Best features: **Flavanoids, Color intensity, Alcohol, Proline**

5. **Ensemble Learning:**
   After applying ensemble learning (max voting technique) using SVM with quadratic, SVM with radial basis function, and the MLP classifier with 2 hidden layers with 16 and 256 nodes following accuracy is obtained: **97.22%**.