**ML ICP-5**

**Name: Sudheer Gajulapalli**

**Id: 700741485**

**GIT HUB URL: https://github.com/SudheerGajulapalli/ML\_ICP\_5**

**Video Recording: https://drive.google.com/file/d/11OS\_2anxniHO2O4P16p-2TDntQXok-Qy/view**

**1. Principal Component Analysis**

**a. Apply PCA on CC dataset.**

**b. Apply k-means algorithm on the PCA result and report your observation if the silhouette score has improved or not?**

**c. Perform Scaling+PCA+K-Means and report performance**

**a. Apply PCA on CC dataset.**

Graphical user interface, text, application

Description automatically generated

**Graphical user interface, text, application, email

Description automatically generated**

**b. Apply k-means algorithm on the PCA result and report your observation if the silhouette score has improved or not?**

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

**c. Perform Scaling+PCA+K-Means and report performance**

Graphical user interface, text, application, email

Description automatically generated

**Code Explanation:**

* Data Preparation: The code reads in a dataset ('CC GENERAL.csv'), drops some irrelevant columns, and fills the missing values with the column mean. Then it standardizes the data using the StandardScaler from scikit-learn.
* PCA: The code applies PCA to the standardized data, using 10 principal components. It then calculates and prints the explained variance ratios and the cumulative sum of explained variance.
* K-Means Clustering: The code applies K-Means clustering to the PCA-transformed data, using 5 clusters. It then calculates the silhouette score, a metric for assessing the quality of clustering, and prints the average silhouette score.
* Silhouette Scores for Multiple Cluster Sizes: The code applies PCA to the standardized data using 2 principal components. Then, it applies K-Means clustering for cluster sizes ranging from 2 to 10, calculates the silhouette score for each cluster size, and prints the results.
* Silhouette Score Plot: The code plots the silhouette scores against the number of clusters, to visualize the optimal number of clusters to use for the dataset.

**2. Use pd\_speech\_features.csv**

**a. Perform Scaling**

**b. Apply PCA (k=3)**

**c. Use SVM to report performance**

Graphical user interface, text, application, email

Description automatically generated

**Code Explanation:**

* The code loads a dataset and separates the input features from the target variable.
* The input features are standardized using Scikit-Learn's StandardScaler.
* Principal Component Analysis (PCA) is performed on the standardized features to reduce their dimensionality.
* The reduced features are split into training and testing sets using Scikit-Learn's train\_test\_split function.
* A Support Vector Machine (SVM) model is trained on the reduced training data.
* The trained SVM model is used to make predictions on the reduced testing data and the accuracy of the predictions is evaluated using Scikit-Learn's accuracy\_score function.

**3. Apply Linear Discriminant Analysis (LDA) on Iris.csv dataset to reduce dimensionality of data tok=2.**

Graphical user interface, text, application, email

Description automatically generated

**Code Explanation:**

* Import the necessary libraries - pandas and sklearn.
* Load the Iris dataset using the pandas library.
* Split the dataset into features (X) and labels (y).
* Scale the features using the StandardScaler from sklearn.
* Apply Linear Discriminant Analysis (LDA) to reduce the dimensionality of the feature space from the original number of features to 2.
* Fit the LDA model to the scaled data and transform it into the new 2-dimensional space.

**4. Briefly identify the difference between PCA and LDA**

* Polynomial Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are techniques used for dimensionality reduction. However, they have distinct objectives and are utilized in different circumstances.
* PCA is an unsupervised approach that attempts to identify the most critical characteristics or directions in the data that capture the highest amount of variance. It disregards the data labels and attempts to uncover a low-dimensional representation of the data that retains as much information as feasible. PCA is frequently employed for data visualization, noise reduction, and feature extraction.
* Conversely, LDA is a supervised approach that aims to identify the most distinctive features or directions in the data that maximize the separation between classes. It considers the data labels and strives to discover a low-dimensional representation of the data that maximizes the inter-class distance and minimizes the intra-class distance. LDA is often employed for classification and feature extraction.
* In summary, although both PCA and LDA are techniques for dimensionality reduction, PCA is an unsupervised approach that attempts to capture the highest amount of variance in the data, while LDA is a supervised approach that attempts to maximize the separation between classes in the data.