Implementing PCA from Scratch and Applying it to Car Data

Objective: The objective of this documentation is to delve into the process of PCA application on the 'Car_data' dataset, aiming to reduce the dimensions of the dataset while retaining crucial information and visualize the principal components' significance.

1.Introduction:

Dataset Overview: The 'Car_data' dataset comprises information related to various car models and their attributes such as model,year,price,transmission,mileage,fuel type,tax,MPG and engine size.

Purpose: This Python code conducts Principal Component Analysis (PCA) on the 'Car_data' dataset to reduce its dimensionality and visualize the principal components.

2.Data Understanding

-Loads the dataset and examines its structure to identify features.

-Extracts numeric features from the dataset using Pandas.

3.Implementing PCA through Covariance Matrix

-Calculates feature means and centers the dataset by subtracting means from each feature.

-Computes the covariance matrix of the centered dataset

-This matrix represents the relationships between different features and serves as a basis for identifying principal components.

4.Eigen Values and Eigen Vectors

-Eigenvalues and eigenvectors of the covariance matrix are computed using NumPy's linear algebra functions.

-These eigenvalues represent the variance captured by each eigenvector (principal component) and are essential in determining the most significant components.

5.Principal Components

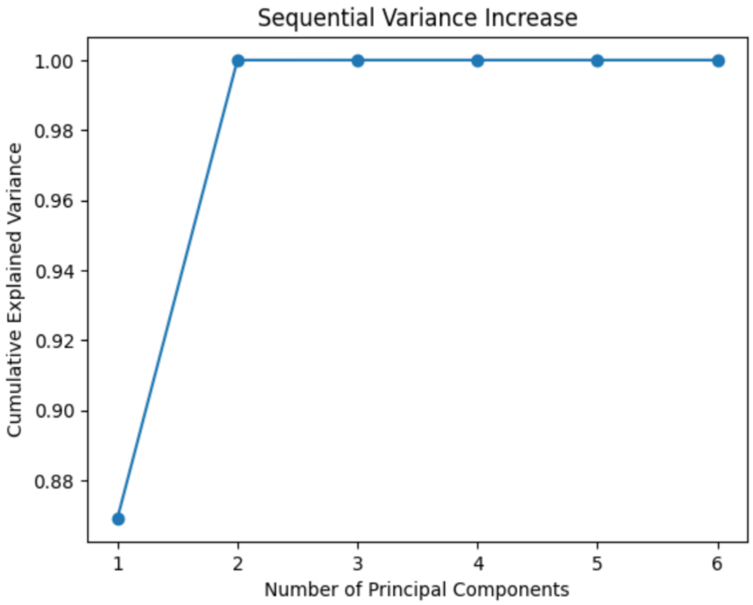
-Sorts eigenvalues and eigenvectors in descending order.

-Enables the selection of the top k eigenvectors, corresponding to the largest eigenvalues. These top eigenvectors form the principal components that capture the most variance within the dataset.

6.Explained Variance

-Calculates the variance covered by each principal component and cumulative explained variance.

-Understanding the variance covered by each component helps in assessing how much information is retained in the reduced-dimensional space.



7.Visualisation using Pair Plots

-Standardizes the data for uniformity

-Performs PCA on the standardized data to obtain principal components.

-Creates a DataFrame with projected data and merges it with original features.

- Create pair plots with principal components as vectors to visualize their directions and importance.

8.Conclusion

-The selection of top principal components significantly impacted the dataset's dimensionality reduction, emphasizing the principal components with the highest variance

-Understanding the variance covered by each principal component facilitated informed decisions on the number of components required to capture essential dataset information.

-The implementation highlighted PCA's relevance in extracting meaningful information from multi-dimensional datasets, facilitating better comprehension and analysis of complex datasets like 'Car_data'.