Name: **Date:** / /

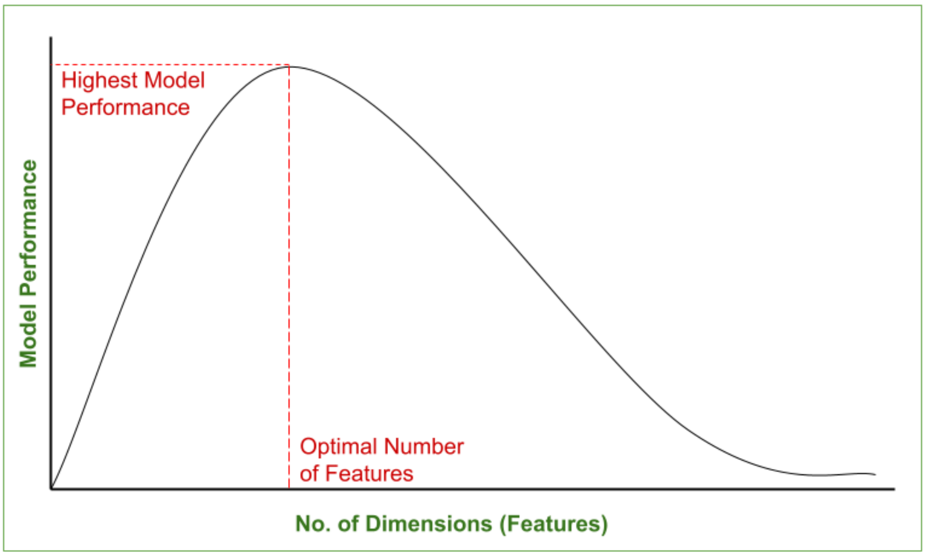
Experiment No: 10

Aim: Implementation of Dimension Reduction: PCA.

Theory:

Dimensionality Reduction is a statistical/ML-based technique wherein we try to reduce the number of features in our dataset and obtain a dataset with an optimal number of dimensions.

One of the most common ways to accomplish Dimensionality Reduction is Feature Extraction, wherein we reduce the number of dimensions by mapping a higher dimensional feature space to a lower-dimensional feature space. The most popular technique of Feature Extraction is Principal Component Analysis (PCA)



Principal Component Analysis (PCA)

As stated earlier, Principal Component Analysis is a technique of feature extraction that maps a higher dimensional feature space to a lower-dimensional feature space. While reducing the number of dimensions, PCA ensures that maximum information of the original dataset is retained in the dataset with the reduced no. of dimensions and the co-relation between the newly obtained Principal Components is minimum. The new features obtained after applying PCA are called Principal Components and are denoted as PCi (i=1,2,3…n). Here, (Principal Component-1) PC1 captures the maximum information of the original dataset, followed by PC2, then PC3 and so on.

The following bar graph depicts the amount of Explained Variance captured by various Principal Components. (The Explained Variance defines the amount of information captured by the Principal Components).

Steps to perform PCA:

1. Data Standardization: Standardization is all about scaling the data in such a way that all the values/variables are in a similar range. Standardization means rescaling data to have a mean of 0 and a standard deviation of 1

2. Computing covariance matrix: The covariance matrix is computed after data standardization, and it is used to find correlated features in the dataset. Each element of the covariance matrix represents the relation between two features.

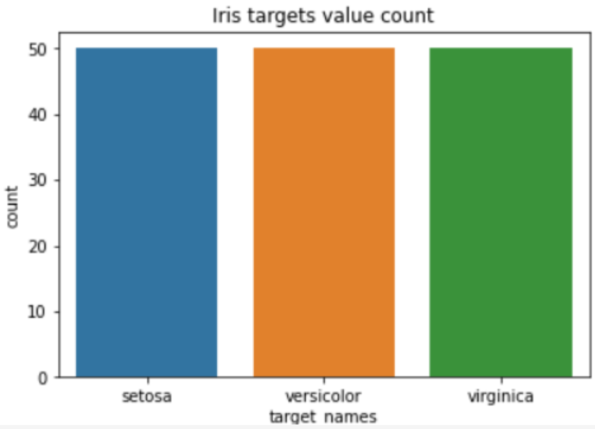
3. Determining eigenvalues and eigenvectors: Eigenvalues and eigenvectors are the mathematical constructs that must be computed from the covariance matrix in order to determine the principal components of the dataset.

4. Principal components are the new set of variables/features that are obtained from the initial set of features. They compress and possess most of the useful information that was scattered among the initial features. Eigenvectors are those vectors when a linear transformation is performed on them then their direction does not change. Eigenvalues simplify denote the scalars of the respective eigenvectors.

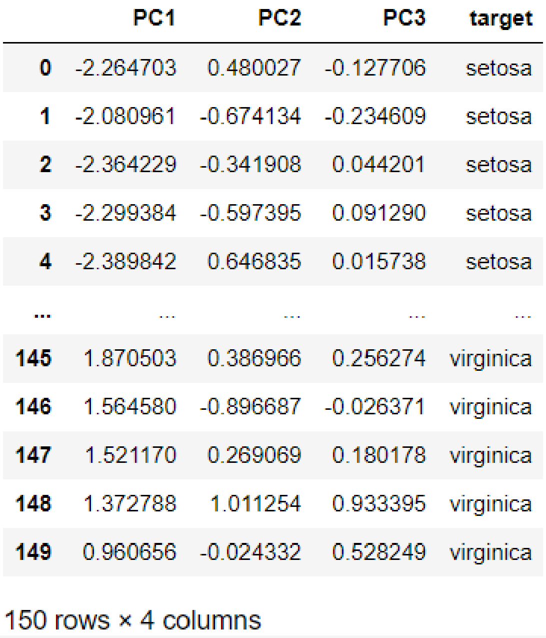
5. Computing Principal Components: Now, these principal components would be used as input to train the model and for data visualization.

Code:

```
import pandas as pd
from sklearn import datasets
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# Load the Iris dataset
iris = datasets.load_iris()
target_names = {0: 'setosa', 1: 'versicolor', 2: 'virginica'}

# Create a DataFrame from the Iris dataset
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target
df['target_names'] = df['target'].map(target_names)

# Countplot of target names
sns.countplot(x='target_names', data=df)
plt.title('Iris targets value count')
plt.show()
```



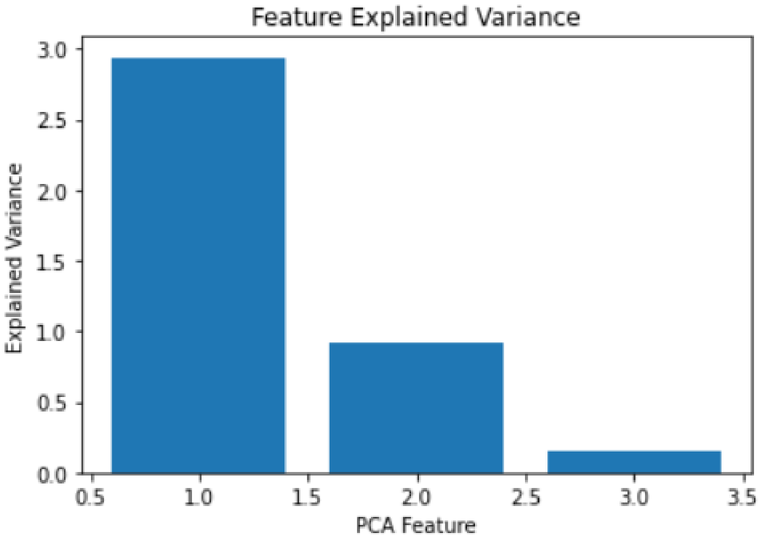
```
# Standardize the feature data
x_scaled = StandardScaler().fit_transform(df[iris.feature_names])

# Apply PCA with 3 components
pca = PCA(n_components=3)
pca_features = pca.fit_transform(x_scaled)

# Create a DataFrame from the PCA features
pca_df = pd.DataFrame(data=pca_features, columns=['PC1', 'PC2', 'PC3'])
pca_df['target'] = iris.target
pca_df['target'] = pca_df['target'].map(target_names)
pca_df
```



	PC1	PC2	PC3	target
0	-2.264703	0.480027	-0.127706	setosa
1	-2.080961	-0.674134	-0.234609	setosa
2	-2.364229	-0.341908	0.044201	setosa
3	-2.299384	-0.597395	0.091290	setosa
4	-2.389842	0.646835	0.015738	setosa
...	...	...	...	...
145	1.870503	0.386966	0.256274	virginica
146	1.564580	-0.896687	-0.026371	virginica
147	1.521170	0.269069	0.180178	virginica
148	1.372788	1.011254	0.933395	virginica
149	0.960656	-0.024332	0.528249	virginica

150 rows × 4 columns

```
# Bar plot of explained variance by each PCA feature
plt.bar(range(1, len(pca.explained_variance_) + 1), pca.explained_variance_)
plt.xlabel('PCA Feature')
plt.ylabel('Explained Variance')
plt.title('Feature Explained Variance')
plt.show()
```



Conclusion: I have implemented Dimension Reduction: PCA.