# A python program to implement Dimensionality Reduction using PCA

Aim:

To implement Dimentionality reduction using PCA in a python program.

Algorithm:

Step 1: Import Libraries

Import necessary libraries, including pandas, numpy, matplotlib.pyplot, and sklearn.decomposition.PCA.

Step 2: Load the dataset (iris dataset)

Load your dataset into a pandas DataFrame.

Step 3: Standardize the data.

* Standardize the features of the dataset using StandardScaler from sklearn.preprocessing.

Step 4: Apply PCA.

* Create an instance of PCA,
* Fit PCA to the Stemdardized data,
* Transform the data to its principle components using transform.

Step 5: Explained Variance Ratio

* calculate the explained variance ratios.

Step 6: Choose the number of components

* Based on the scree plot, choose the number of principle components that explain a significant amount of variance.

Step 7: Apply PCA with chose components.

* Apply PCA again with the chosen number of components.

Step 8: Visualize the reduced Data.

* Transform the original data to the reduced dimension using the fitted PCA.

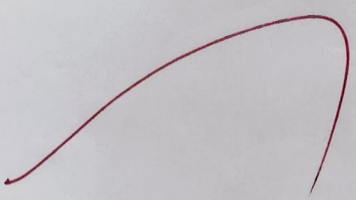
Step 9: Interpretation.

* Interpret the results, considering the trade-off's between dimensionality reduction and information loss.

Program:

```
from sklearn import datasets
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

```
import seaborn as sns
iris = datasets.load_iris()
df.head()
Scalar = StandardScaler()
Scaled_data = pd.DataFrame(Scalar.fit_transform(df))
Scaled_data
sns.heatmap(Scaled_data.corr())
pca = PCA(n_components = 3)
pca.fit(Scaled_data)
data_pca = pca.transform(Scaled_data)
data_pca = pd.DataFrame(data_pca, columns = ['PC1',
                                             'PC2', 'PC3'])
data_pca.head()
sns.heatmap(data_pca.corr())
```



Result:-

Thus Dimensionality Reduction has been implemented using PCA in a python program successfully and the results have been analyzed.