ASSIGNMENT NO.4

NAME-: MORE PANKAJ SAMPAT

ROLL NO-: 054

CLASS-: TE COMP

SUB-: COMPUTATIONAL SATISTICS

COURSE-: AIML HONOUR COURSE

ACADEMIC YEAR-: 2022-23

Problem Satement-: Apply Basic PCA on the iris dataset. The data set is available at: https://raw.github.com/neurospin/pystatsml/master/datasets/iris.csv • Describe the data set. Should the dataset been standardized? • Describe the structure of correlations among variables. • Compute a PCA with the maximum number of components .• Compute the cumulative explained variance ratio. Determine the number of components$K$by your computed values. • Print the$K$principal components directions and correlations of the$K$principal compo-nents with the original variables. Interpret the contribution of the original variables into the PC. • Plot the samples projected into the$K$first PCs. • Color samples by their species.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.mixture import GaussianMixture
from sklearn.metrics.cluster import adjusted_rand_score
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files iris4.csv

- **iris4.csv**(text/csv) - 3858 bytes, last modified: 11/16/2022 - 100% done

```
Saving iris4.csv to iris4.csv
```

```
import pandas as pd
import io
```

```
3
4 iris_data = pd.read_csv(io.BytesIO(uploaded['iris4.csv']))
```

```
1 iris_data.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Standardization

Each feature in the dataset has different mean and SD hence it is necessary to scale data for comparision with other features.

Standardization sets mean to zero and SD of 1 for all features.

As seen in the data below the SD and is different for each datasetalso the variation is different.

```
1 print("Sepal length range: [%s, %s]" % (min(iris_data["sepal_length"]),max(iris_data["s
2 print("Sepal width range: [%s, %s]" % (min(iris_data["sepal_width"]),max(iris_data["sep
3 print("Petal length range: [%s, %s]" % (min(iris_data["petal_length"]),max(iris_data["p
4 print("Petal width range: [%s, %s]" % (min(iris_data["petal_width"]),max(iris_data["pet
5
6 print("Sepal Length standard deviation :\t %f" %np.std(iris_data["sepal_length"]))
7 print("Sepal Width standard deviation :\t %f" %np.std(iris_data["sepal_width"]))
8 print("Petal Length standard deviation :\t %f" %np.std(iris_data["petal_length"]))
9 print("Petal Width standard deviation :\t %f" %np.std(iris_data["petal_width"]))
```

```
Sepal length range: [4.3, 7.9]
Sepal width range: [2.0, 4.4]
Petal length range: [1.0, 6.9]
Petal width range: [0.1, 2.5]
Sepal Length standard deviation :	 0.825301
Sepal Width standard deviation :	 0.434411
Petal Length standard deviation :	 1.759404
Petal Width standard deviation :	 0.759693
```

Correlation Matrix :

```
1 corr = iris_data.corr(method ="pearson")
2 display(corr)
3 sns.heatmap(corr,cmap =sns.diverging_palette(10,0,as_cmap=True),annot= True,fmt = "f")
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f066f15f190>
```

** PCA (Principal Component Analysis):**

```
from sklearn.decomposition import PCA
pca = PCA()
x_new1 =pca.fit_transform(iris_data.drop(["species"],axis =1))
x_new1[:5]
```

```
array([[-2.68412563e+00,  3.19397247e-01, -2.79148276e-02,
        -2.26243707e-03],
       [-2.71414169e+00, -1.77001225e-01, -2.10464272e-01,
        -9.90265503e-02],
       [-2.88899057e+00, -1.44949426e-01,  1.79002563e-02,
        -1.99683897e-02],
       [-2.74534286e+00, -3.18298979e-01,  3.15593736e-02,
         7.55758166e-02],
       [-2.72871654e+00,  3.26754513e-01,  9.00792406e-02,
         6.12585926e-02]])
```

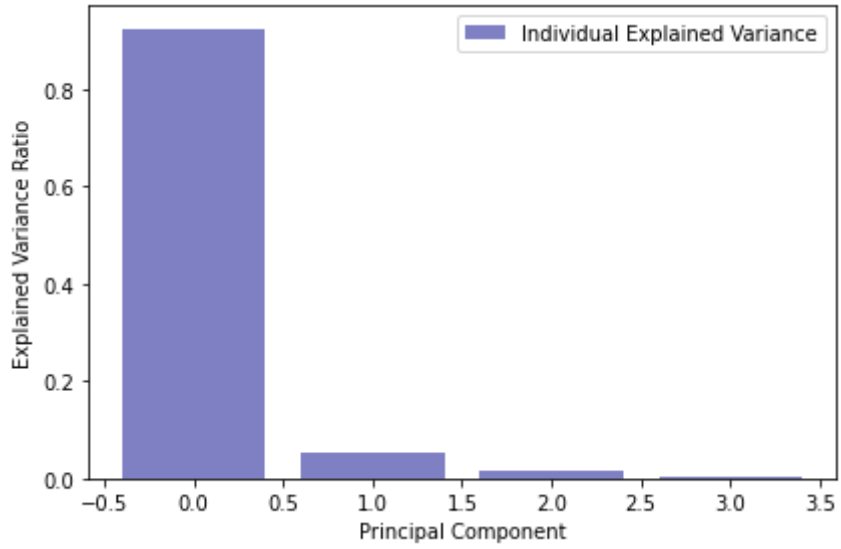
Explained Variance of R^2

```
explained_variance= pca.explained_variance_ratio_

explained_variance
```

```
array([0.92461872, 0.05306648, 0.01710261, 0.00521218])
```

```
plt.figure(figsize =(6,4))
plt.bar(range(4),explained_variance, alpha=0.5, align='center', label ="Individual Expl
plt.ylabel("Explained Variance Ratio")
plt.xlabel("Principal Component")
plt.legend(loc ="best")
plt.tight_layout()
```



There are 3 Principal Components.

Applying PCA

```
pca  = PCA(n_components=3)
x_new =pca.fit_transform(iris_data.drop(['species'],axis =1))
x_new[:5]
```

```
array([[-2.68412563,  0.31939725, -0.02791483],
       [-2.71414169, -0.17700123, -0.21046427],
       [-2.88899057, -0.14494943,  0.01790026],
       [-2.74534286, -0.31829898,  0.03155937],
       [-2.72871654,  0.32675451,  0.09007924]])
```

Correlation and Direction of PCA

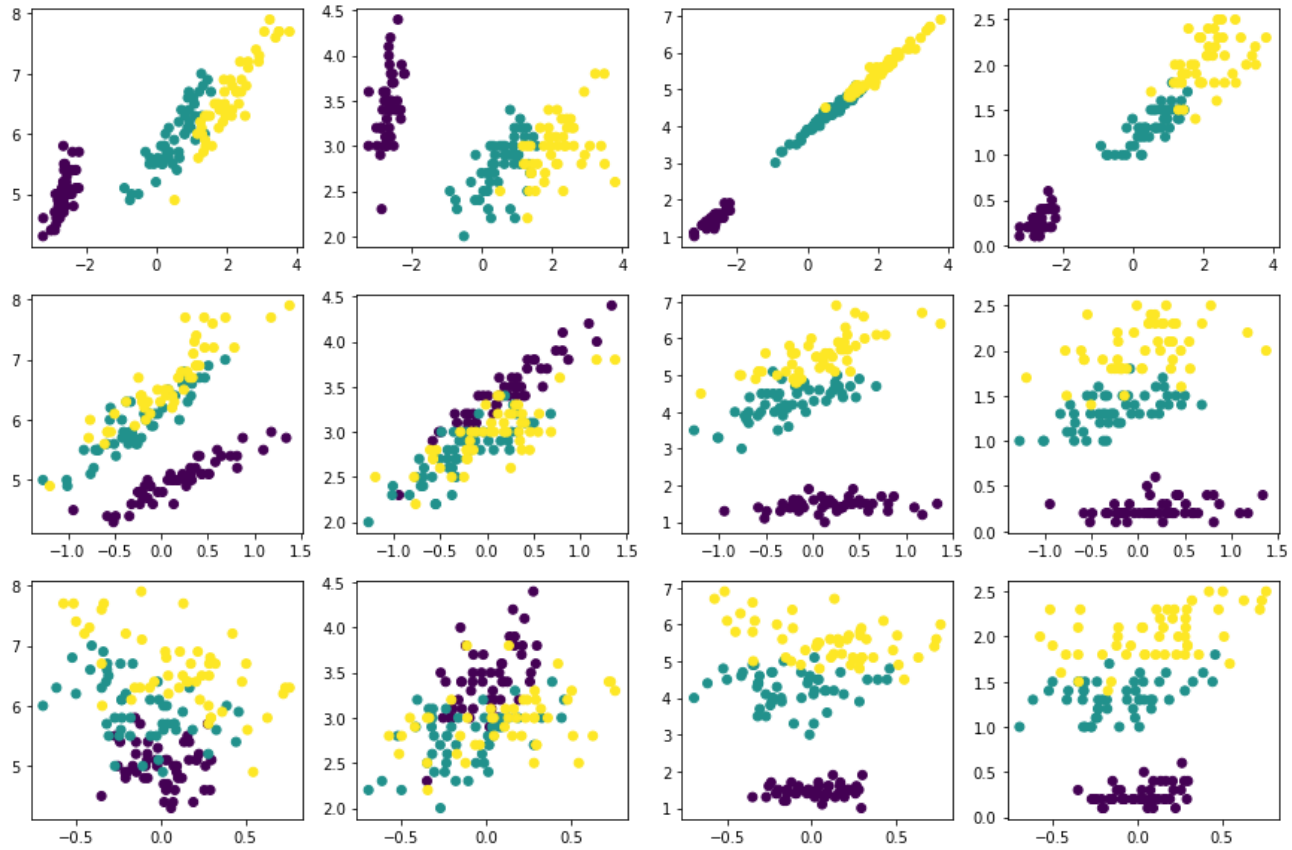
converting categorial data to numerical

```
categ_num ={"species":{"setosa":0,"versicolor":1,"virginica":2}}
iris_data1 = iris_data.replace(categ_num)
columns = list(iris_data.columns[:4])
```


```
categ_num ={"species":{"setosa":0,"versicolor":1,"virginica":2}}
iris_data1 = iris_data.replace(categ_num)
columns = list(iris_data.columns[:4])
```

Comparing Principal Components with original variables

```
fig, axes =plt.subplots(3,4,figsize =(15,10))
k= 0
for i in range(axes.shape[0]):
    for j in range(axes.shape[1]):
        axes[i,j].scatter(x_new[:,i],iris_data[columns[j]],c =iris_data1["species"])
plt.show()
```



Colab paid products - Cancel contracts here

✓ 2s completed at 5:17 PM