MACHINE LEARINING

LAB ASSESSMENT – V

15BCE0082 VOLETI RAVI

Implement Principle Component Analysis for Dimensionality Reduction.

CODE:

```
import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns; sns.set()

from sklearn.decomposition import PCA


rng = np.random.RandomState(1)

X = np.dot(rng.rand(2, 2), rng.randn(2, 200)).T

plt.scatter(X[:, 0], X[:, 1])

plt.axis('equal');


pca = PCA(n_components=2)

pca.fit(X)


print(pca.components_)


print(pca.explained_variance_)


def draw_vector(v0, v1, ax=None):

    ax = ax or plt.gca()
```

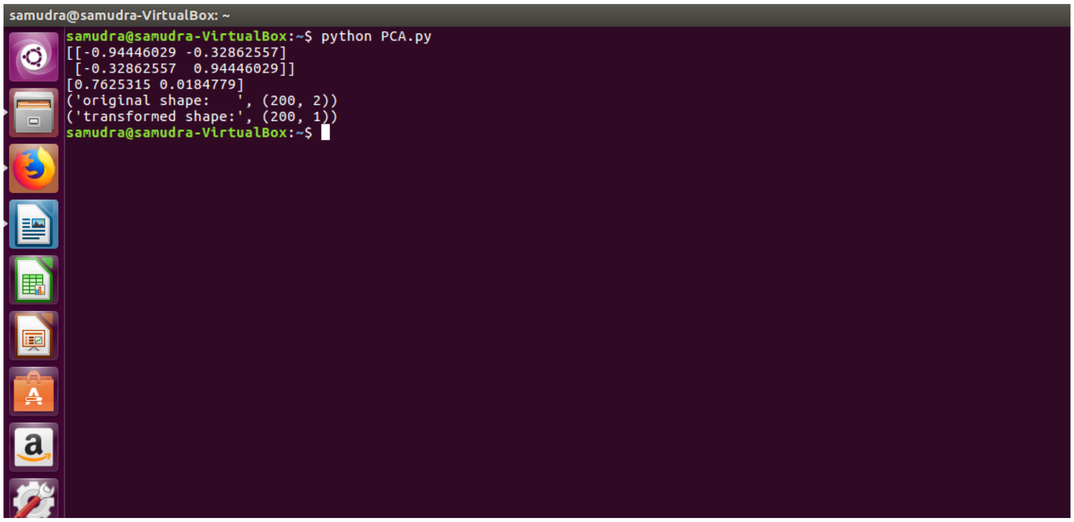
```
    arrowprops=dict(arrowstyle='->', linewidth=2, shrinkA=0, shrinkB=0)
    ax.annotate('', v1, v0, arrowprops=arrowprops)

# plot data
plt.scatter(X[:, 0], X[:, 1], alpha=0.2)
for length, vector in zip(pca.explained_variance_, pca.components_):
    v = vector * 3 * np.sqrt(length)
    draw_vector(pca.mean_, pca.mean_ + v)
plt.axis('equal');

pca = PCA(n_components=1)
pca.fit(X)
X_pca = pca.transform(X)
print("original shape:   ", X.shape)
print("transformed shape:", X_pca.shape)

X_new = pca.inverse_transform(X_pca)
plt.scatter(X[:, 0], X[:, 1], alpha=0.2)
plt.scatter(X_new[:, 0], X_new[:, 1], alpha=0.8)
plt.axis('equal');
```

OUTPUT:


samudra@samudra-VirtualBox: ~
samudra@samudra-VirtualBox:~$ python PCA.py
[[-0.94446029 -0.32862557]
 [-0.32862557 0.94446029]]
[0.7625315 0.0184779]
('original shape: ', (200, 2))
('transformed shape:', (200, 1))
samudra@samudra-VirtualBox:~$

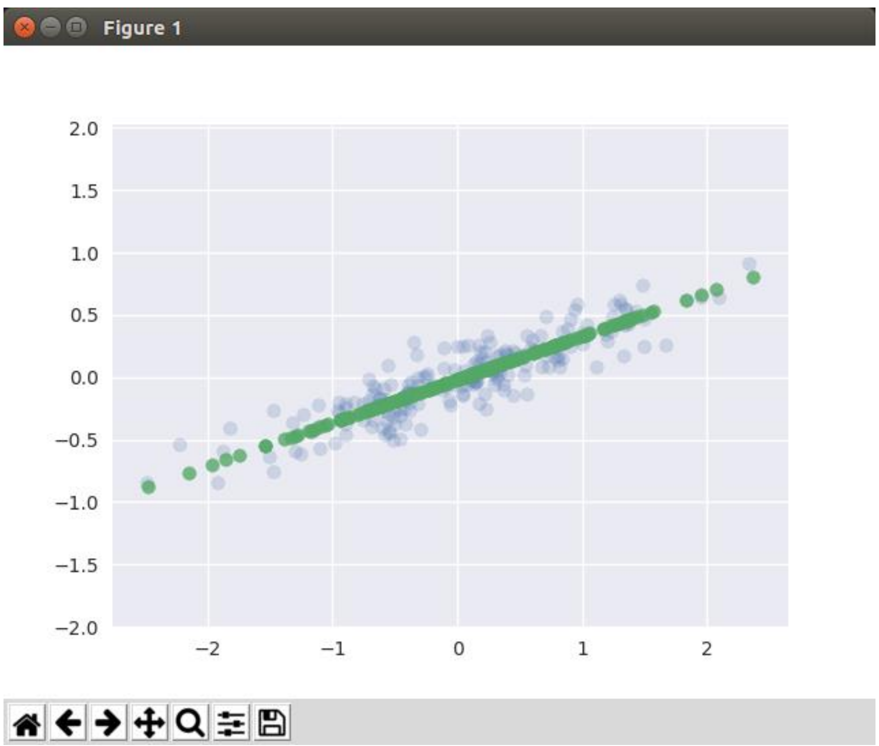


Figure 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

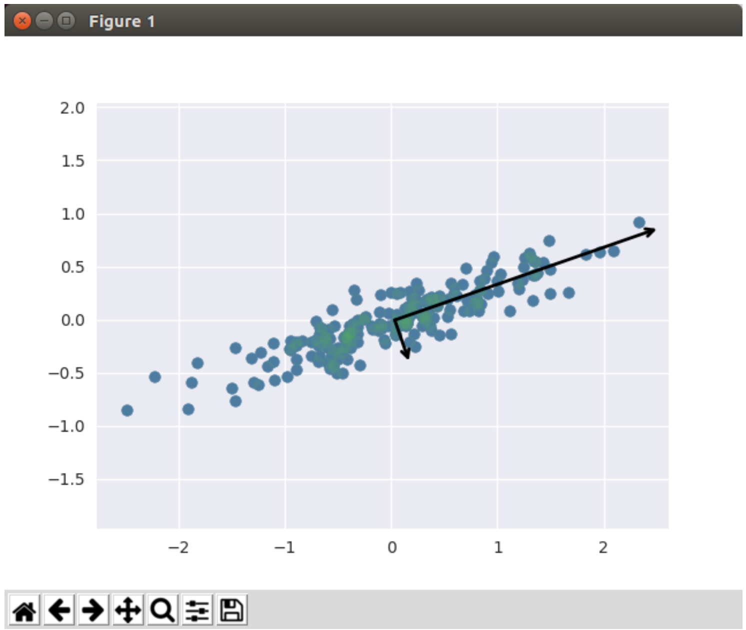


Figure 1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
−1
0
1
2