experiment no 01

name - sumit kumbar

roll no -02

div tycse B

PRN - 22SC114281069

Title : Implementation of feature selection and extraction algorithm

```
import numpy as np
import pandas as pd
data = {
"First" : [10,20,30,np.nan],
"Second" : [np.nan,90,40,55],
"Third" : [10,60,np.nan,50]
}
print(data)
```

```
{'First': [10, 20, 30, nan], 'Second': [nan, 90, 40, 55], 'Third': [10, 60, nan, 50]}
```

```
df = pd.DataFrame(data)
print(df)
```

```
   First  Second  Third
0   10.0     NaN   10.0
1   20.0    90.0   60.0
2   30.0    40.0    NaN
3    NaN    55.0   50.0
```

```
print(df.isnull())
```

```
   First  Second  Third
0  False    True  False
1  False   False  False
2  False   False   True
3   True   False  False
```

```
print(df.notnull())
```

```
   First  Second  Third
0   True   False   True
1   True    True   True
2   True    True  False
3  False    True   True
```

```
df.fillna(0)
```

	First	Second	Third
0	10.0	0.0	10.0
1	20.0	90.0	60.0
2	30.0	40.0	0.0
3	0.0	55.0	50.0

```
iris = pd.read_csv('/content/iris.csv')
print(iris)
```

```
     sepal.length  sepal.width  petal.length  petal.width    variety
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
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

[150 rows x 5 columns]
```

```
x = iris[['sepal.length','variety']]
print(x)
```

```
     sepal.length    variety
0             5.1     Setosa
1             4.9     Setosa
2             4.7     Setosa
3             4.6     Setosa
4             5.0     Setosa
..            ...        ...
145           6.7  Virginica
146           6.3  Virginica
147           6.5  Virginica
148           6.2  Virginica
149           5.9  Virginica

[150 rows x 2 columns]
```

```
print(x.shape)
```

```
(150, 2)
```

```
print(x.size)
```

```
300
```

PCA witout using library

```
import numpy as np
input = np.array([[2.5,2.4],[0.5,0.7],[2.2,2.9],[1.9,2.2],[3.1,3.0],[2.3,2.7],[2,1.6],[1,1.6],[1,1.1],[1.5,1.6],[1.1,0.9]])
print("Input Values \n")
print(input)
```

```
Input Values

[[2.5 2.4]
 [0.5 0.7]
 [2.2 2.9]
 [1.9 2.2]
 [3.1 3. ]
 [2.3 2.7]
 [2.  1.6]
 [1.  1.6]
 [1.  1.1]
 [1.5 1.6]
 [1.1 0.9]]
```

```
mean_values = input.mean(axis = 0)
print("Mean Values \n")
print(mean_values)
```

```
Mean Values

[1.73636364 1.88181818]
```

```
zero_mean_data = input - mean_values
print("Zero Mean Data \n")
print(zero_mean_data)
```

```
Zero Mean Data

[[ 0.76363636  0.51818182]
 [-1.23636364 -1.18181818]
 [ 0.46363636  1.01818182]
 [ 0.16363636  0.31818182]
 [ 1.36363636  1.11818182]
 [ 0.56363636  0.81818182]
 [ 0.26363636 -0.28181818]
 [-0.73636364 -0.28181818]
 [-0.73636364 -0.78181818]
 [-0.23636364 -0.28181818]
 [-0.63636364 -0.98181818]]
```

```
zero_mean_data = input - mean_values
print("Zero Mean Data \n")
print(zero_mean_data)
```

```
Zero Mean Data

[[ 0.76363636  0.51818182]
```

```
       [-1.23636364 -1.18181818]
       [ 0.46363636  1.01818182]
       [ 0.16363636  0.31818182]
       [ 1.36363636  1.11818182]
       [ 0.56363636  0.81818182]
       [ 0.26363636 -0.28181818]
       [-0.73636364 -0.28181818]
       [-0.73636364 -0.78181818]
       [-0.23636364 -0.28181818]
       [-0.63636364 -0.98181818]]
```

```
cov = np.cov(zero_mean_data.T)
print("Covariance Matrix \n")
print(cov)
```

```
Covariance Matrix

[[0.61454545 0.57672727]
 [0.57672727 0.65363636]]
```

```
eigen_values,eigen_vectors = np.linalg.eig(cov)
print("Eigen Values \n")
print(eigen_values)
print("Eigen Vectors \n")
print(eigen_vectors)
```

```
Eigen Values

[0.05703253 1.21114929]
Eigen Vectors

[[-0.71898221 -0.69502847]
 [ 0.69502847 -0.71898221]]
```

```
idx = eigen_values.argsort()[::-1]
print(idx)
eigen_vectors = eigen_vectors[:,idx]
print("Sorted Eigen Vectors \n")
print(eigen_vectors)
```

```
[1 0]
Sorted Eigen Vectors

[[-0.69502847 -0.71898221]
 [-0.71898221  0.69502847]]
```

```
row_feature_vector = eigen_vectors.T
row_zero_mean_data = zero_mean_data.T
final_data = row_feature_vector.dot(row_zero_mean_data)
print("PCA \n")
print(final_data)
```

```
PCA

[[-0.90331253  1.70901418 -1.05429509 -0.342499   -1.75171894 -0.98000149
   0.01938748  0.71441595  1.07390706  0.36690172  1.14820066]
 [-0.18888984  0.06752618  0.37431906  0.10349379 -0.20326209  0.16341514
  -0.38542152  0.3335607  -0.01395354 -0.02593041 -0.22485746]]
```

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
iris = load_iris()
x = iris.data
pca = PCA(n_components = 2)
principalComponents = pca.fit_transform(x)
print(principalComponents)
```

```
[[-2.68412563  0.31939725]
 [-2.71414169 -0.17700123]
 [-2.88899057 -0.14494943]
 [-2.74534286 -0.31829898]
 [-2.72871654  0.32675451]
 [-2.28085963  0.74133045]
 [-2.82053775 -0.08946138]
 [-2.62614497  0.16338496]
 [-2.88638273 -0.57831175]
 [-2.6727558  -0.11377425]
 [-2.50694709  0.6450689 ]
 [-2.61275523  0.01472994]
 [-2.78610927 -0.235112  ]
 [-3.22380374 -0.51139459]
```

```
[-2.64475039  1.17876464]
[-2.38603903  1.33806233]
[-2.62352788  0.81067951]
[-2.64829671  0.31184914]
[-2.19982032  0.87283904]
[-2.5879864   0.51356031]
[-2.31025622  0.39134594]
[-2.54370523  0.43299606]
[-3.21593942  0.13346807]
[-2.30273318  0.09870885]
[-2.35575405 -0.03728186]
[-2.50666891 -0.14601688]
[-2.46882007  0.13095149]
[-2.56231991  0.36771886]
[-2.63953472  0.31203998]
[-2.63198939 -0.19696122]
[-2.58739848 -0.20431849]
[-2.4099325   0.41092426]
[-2.64886233  0.81336382]
[-2.59873675  1.09314576]
[-2.63692688 -0.12132235]
[-2.86624165  0.06936447]
[-2.62523805  0.59937002]
[-2.80068412  0.26864374]
[-2.98050204 -0.48795834]
[-2.59000631  0.22904384]
[-2.77010243  0.26352753]
[-2.84936871 -0.94096057]
[-2.99740655 -0.34192606]
[-2.40561449  0.18887143]
[-2.20948924  0.43666314]
[-2.71445143 -0.2502082 ]
[-2.53814826  0.50377114]
[-2.83946217 -0.22794557]
[-2.54308575  0.57941002]
[-2.70335978  0.10770608]
[ 1.28482569  0.68516047]
[ 0.93248853  0.31833364]
[ 1.46430232  0.50426282]
[ 0.18331772 -0.82795901]
[ 1.08810326  0.07459068]
[ 0.64166908 -0.41824687]
[ 1.09506066  0.28346827]
[-0.74912267 -1.00489096]
```