

SIT384 Cyber security analytics

Pass Task 8.1P: PCA dimensionality reduction

Task description:

PCA (Principle Component Analysis) is a dimensionality reduction technique that projects the data into a lower dimensional space. It can be used to reduce high dimensional data into 2 or 3 dimensions so that we can visualize and hopefully understand the data better.

In this task, you use PCA to reduce the dimensionality of a given dataset and visualize the data.

You are given:

- Breast cancer dataset which can be retrieved from:
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()
detailed info available at: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html
- PCA(n_components=2)
- 3D plot settings: (Please refer to prac7 for 3D plot examples)
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(10, 8))
cmap = plt.cm.get_cmap("Spectral")
ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=10, azim=10)
ax.scatter(x,y,z, c=cancer.target, cmap=cmap)
- Other settings of your choice

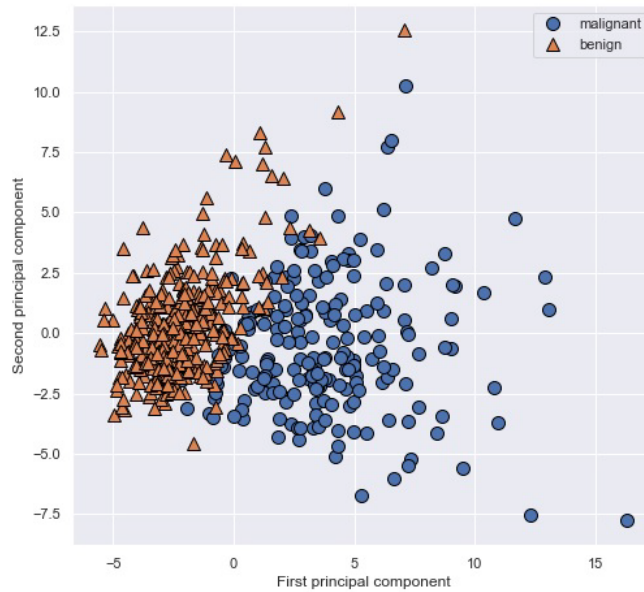
You are asked to:

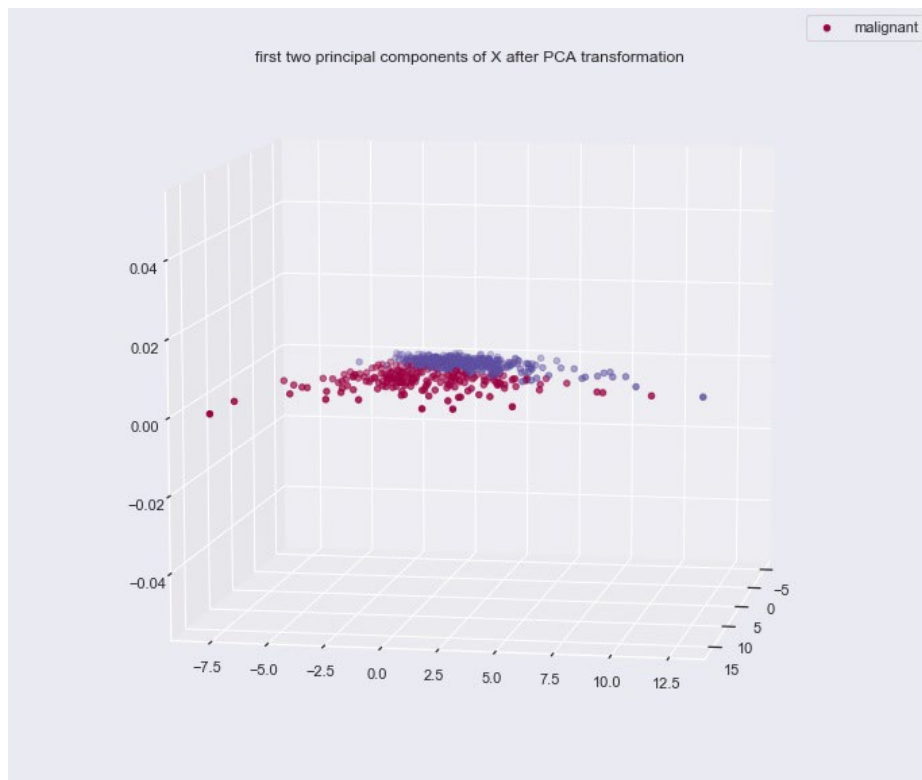
- use StandardScaler() to first fit and transform the cancer.data,
- apply PCA (n_components=2) to fit and transform the scaled cancer.data set
- print the scaled dataset shape and PCA transformed dataset shape for comparison
- create 2D plot with the first principal component as x axis and the second principal component as y axis
- set proper xlabel, ylabel for the 2D plot
- print the PCA component shape and component values
- create a 3D plot with the first 3 features (as x,y and z) of the scaled cancer.data set
- create a 3D plot with the first principal component as x axis and the second principal component as y axis, no value for z axis
- set proper title for the two 3D plots

Sample output as shown in the following figures are **for demonstration purposes only**. Yours might be different from the provided.

Original shape: (569, 30)
Reduced shape: (569, 2)
PCA component shape: (2, 30)
PCA components:

```
[[ 0.21890244 0.10372458 0.22753729 0.22099499 0.14258969 0.23928535
  0.25840048 0.26085376 0.13816696 0.06436335 0.20597878 0.01742803
  0.21132592 0.20286964 0.01453145 0.17039345 0.15358979 0.1834174
  0.04249842 0.10256832 0.22799663 0.10446933 0.23663968 0.22487053
  0.12795256 0.21009588 0.22876753 0.25088597 0.12290456 0.13178394]
 [-0.23385713 -0.05970609 -0.21518136 -0.23107671 0.18611302 0.15189161
  0.06016536 -0.0347675 0.19034877 0.36657547 -0.10555215 0.08997968
 -0.08945723 -0.15229263 0.20443045 0.2327159 0.19720728 0.13032156
  0.183848 0.28009203 -0.21986638 -0.0454673 -0.19987843 -0.21935186
  0.17230435 0.14359317 0.09796411 -0.00825724 0.14188335 0.27533947]]
```





Submission:

Submit the following files to OnTrack:

1. Your program source code (e.g. task8_1.py)
2. A screen shot of your program running

Check the following things before submitting:

1. Add proper comments to your code