International Journal for Modern Trends in Science and Technology, 6(10): 134-136, 2020

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST061024
Available online at: http://www.ijmtst.com/vol6issue10.html

Principal Component Analysis on the Breast Cancer – Python

Subash Kumar

Bachelor of Computer Science Engineering, Anna University

To Cite this Article
Subash Kumar, "Principal Component Analysis on the Breast Cancer – Python", *International Journal for Modern Trends in Science and Technology*, 6(10): 134-136, 2020.
Article Info
Received on 02-October-2020, Revised on 16-October-2020, Accepted on 18-October-2020, Published on 26-October-2020.

ABSTRACT

Timely diagnosis of any disease is critical in medical field, with increasing population of breast cancer patients, this paper is dedicated to all medical professionals who are trying to save many lives, let us discuss the importance of principal component analysis on the breast cancer in this journal.

KEYWORDS:*Machine Learning, Deep Learning, Artificial Intelligence, Medicine, Breast Cancer, Cancerprognosis and prediction, Data Science, Principal Component Analysis, Python*

INTRODUCTION

Principal Component Analysis is an unsupervised statistical technique used to examine the interrelations among a set of variables in order to identify the underlying structure of those variables, it is also known as factor analysis, when regression determines a line of best fit to a dataset, a factor analysis determines several orthogonal lines of the best fit to the dataset. The term "Orthogonal" means "at right angles", actually the lines are perpendicular to each other in n-dimensional space, n-dimensional space is the variable sample space, there are many dimensions as there are many variables, so for example in a dataset with 4 variables the sample space is 4 dimensional.

Principal Component Analysis is just a transformation of the data and it attempts to find out what features explain the most variance in the data, for example if we have an original dataset of two components, we try to get rid of the components that don't explain the much of the variance in the data.

In this paper we will use 12 attributes, out of which 10 real valued features from each cell nucleus obtained from the Wisconsin diagnostic breast cancer dataset that explains about the stage of breast cancer M (Malignant) and B (Benign).

Features explanations:

1. ID: Patient id
2. Diagnosis (M = Malignant, B = Benign)
3. Radius(mean of distances from center to points on the perimeter) (worst). Worst texture. Texture (standard deviation of gray-scale values) (worst). Worst perimeter. perimeter (worst)
4. Texture (Breast cancer can cause changes and inflammation in skin cells that can lead to texture changes), here we can take the standard deviation of grey scaled values
5. Perimeter: Size of the core tumor
6. Area: Area of the core tumor
7. Smoothness: Local variation in radius length
8. Compactness: (perimeter^2 / area - 1.0)
9. Concavity (severity of concave portions of the contour)
10. Concave points (Number of concave portions of the contour)
11. Symmetry

12. Fractal Dimension ("coastline approximation" - 1)

METHODOLOGY

Python is a very powerful tool in Machine Learning, let us discuss steps involved in performing the principal component analysis on the Wisconsin datasets in python.

1. The very first step is to import the required machine learning libraries

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
```

2. Now, we need to import the Breast Cancer Datasets from the sklearn library

```
from sklearn.datasets import load_breast_cancer
```

3. We are assigning the dataset to a variable in this step

```
cancer = load_breast_cancer()
```

4. Let us fetch the keys from this dataset, the keys are basically a dictionary datatype, which represents the context of the dataset.
   Input:

```
cancer.keys()
```

   Output:

```
Out[10]: dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename'])
```

5. We are going to create a dataframe based on the features which is “data” and “feature_names”
   Input:

```
df = pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
```

   Output:

Out[14]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	184.60
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	158.80
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	98.87
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	152.20

6. Let us identify the target feature from the dataset.

Input:

```
# for target
cancer['target']
```

Output:

```
Out[15]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
                0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
                1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
                1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
                1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
                0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1,
                1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
                0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
                1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,
                1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
```

7. The explanation of the target feature is shown below

Input:

```
cancer['target_names']
```

Output:

```
array(['malignant', 'benign'], dtype='<U9')
```

8. It is difficult to visualize the high dimensional data, in this dataset we have 30 dimensions and what we can do is to use the PCA concept to find the first two principal components and visualize the data in this new 2 dimensional space, now we are going to do is scale our data so that each feature.

```
from sklearn.preprocessing import StandardScaler
```

9. In this step we will be instantiating a PCA object and find the principal components using the fit method and then apply the rotation in dimensionality reduction and then we can specify the how many components we want to keep

Input:

```
plt.figure(figsize=(8,6))
plt.scatter(x_pca[:,0],x_pca[:,1],c=cancer['target'],cmap='plasma')
#[:0] All the rows from column 0 against [:1] all the rows from column 1
plt.xlabel('First Principal Component')
plt.ylabel('Second Principal Component')
# Now we see the data plotted by the first principal component and the second principle components , interpreting these co
# actually not that easy to understand but what I want to show you is just how powerful this technique is we are going to
# and add one more argument colour
```

```
#PCA
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components=2)
# we are going to visualize the enitire data set just by the two PCA
```

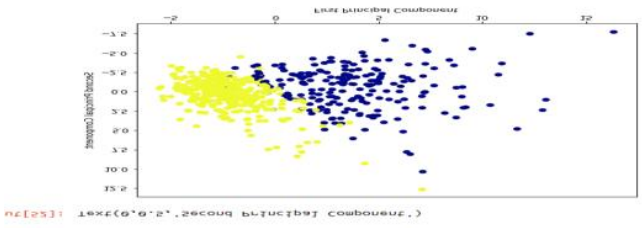
```
pca.fit(Scaled_data)
```

```
PCA(n_components=2)
```

```
x_pca = pca.transform(Scaled_data)
```

10. Now we are going to plot the data based on the first principal component and second principle component, interpreting these components is actually not that easy to understand but what we want is to show, how powerful the machine learning plot that describes the first and second principal components

Output:



11. We can see visually what these principal components represents through heat map, the below heat map shows the relationship between the correlation of the various features and the principal components

Input:

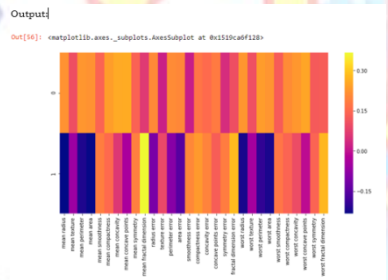
```
df_comp = pd.DataFrame(pca.components_,columns=cancer['feature_names'])
```

Output:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter
0	0.218902	0.103725	0.227537	0.220995	0.142590	0.239285	0.258400	0.260854	0.138167	0.064363	...	0.227997	0.104469	0.236640
1	-0.233857	-0.059706	-0.215181	-0.231077	0.186113	0.151892	0.060165	-0.034768	0.190349	0.366575	...	-0.219866	-0.045467	-0.199878

```
plt.figure(figsize=(12,6))
sns.heatmap(df_comp,cmap='plasma')
```

Output



RESULTS

The Wisconsin Breast cancer data with 30 features was analyzed using PCA component analysis basically each principal component shows here as a row and the higher the number are the dark looking colors or it is yellow which is more correlated to a specific feature in the columns and that is really the best explanation for what these principal components represents their combinations of all these features and we can see what features were specifically important to one of the principal components versus the other

REFERENCES

[1] Author: Ian Joliffe Reference work entry First Online: 02 December 2014 DOI: https://doi.org/10.1007/978-3-642-04898-2_455

[2] Karl Pearson F.R.S. , 1901. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), pp.559–572.

[3] DOI:10.1016/0169-7439(87)80084-9 Corpus ID: 54979942 Principal component analysis S. Wold, K. Esbensen, P. Geladi Published 1987 Computer Science Chemometrics and Intelligent Laboratory Systems

[4] Principal components analysis. Groth D,Hartmann S, Klie S, Selbig J Author information Methods in Molecular Biology (Clifton, N.J.), 31 Dec 2012, 930:527-547 DOI: 10.1007/978-1-62703-059-5_22 PMID: 23086856