**Application Of Principal Component Analysis In Medical Field**

A Project Submitted

In Partial Fulfillment of the Requirements

For the Degree of

**Bachelor of Technology**

In

**Computer Science**

By

**G.K.N. Kireet**

**G. Raghavendra Dinesh**

**Gelle Abhiram**

**Thogati Devaharsha**



**November, 2019**

**Abstract**

For our project on the application of principal component analysis in medical field, we obtained the cumulative data of Drug Poisoning Mortality Rate per state in South-Eastern America from the year 1999 to the year 2016. The dataset has 5 variables named ['Year', 'Deaths in the year', 'Population of the state', 'Crude Death Rate', 'Age-adjusted Rate'] to which we applied Principal Component Analysis and made it into 2 variables with the help of a Programming Language (Python) and observed the results on a plot.

**Introduction**

# Principal Component Analysis:

Principal component analysis is a statistical dimensionality reduction procedure which transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components that still contains most of the information of the data before transformation.

Usually, when we have a dataset having two variables (Example: A dataset of the marks of the students of a class in two subjects), we can visualize the data on a graph by considering one axis as a variable and another axis as the other variable but, when the number of variables of the dataset tend to increase in number, it is not practically possible to visualize the dataset on a graph. In such scenarios Principal Component Analysis is used to reduce the number of variables to a number which is small enough to be visualized without any problems.

Principal Component Analysis is a feature extraction type of dimensionality reduction technique in which, if we have 5 independent variables in the dataset, 5 new independent variables are created which are a combination of the previous 5 variables. In the new variables, the variables which effect the dataset best are kept and the rest are discarded. Hence, the number of variables are reduced without effecting the dataset in a vast way.

# Python Packages

We chose python for our project since the language has a wide flexibility due to which there are vast number of packages in the internet which allow statistical and scientifical methods to be done with ease in python.

Pandas Package:

It is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive.

Numpy Package:

It is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Matplotlib package:

Matplotlib is a python library which is used for visualization of graphs in python.

Sklearn Package:

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

It contains a method to directly convert data into principal components.

# Normalization of Variables

The principal components are supplied with normalized version of original factors. This is because, the original predictors may have different scales. For example: Imagine a data set with variables’ measuring units as gallons, kilometers, light years etc. It is definite that the scale of variances in these variables will be large.

Performing PCA on un-normalized variables will lead to insanely large loadings for variables with high variance. In turn, this will lead to dependence of a principal component on the variable with high variance. This is undesirable.

**Mathematical Details**

One Of the methods to do Principal Component Analysis is the “Singular Value Decomposition” method.

To explain Principal Component Analysis, lets consider a dataset with two variables. One in X and another in Y.

We’ll start PCA by plotting the graph with the two variables.

Now, we calculate the mean of all the values in X and the mean of data in Y and using the two means we find the center C of the plot.

Now, we shift the plot such that the center C is at the origin.

Now, by using Simple Linear Regression, we find a line which best fits the data plot. Or, we can find a line which has the largest sum of squared distances from to each of the point in the plot i.e the eigen value for Principal Component 1 whose square root is called as Singular value of Principal Component 1.

This Line is our Principal Component 1. Using the slope of the line, we can find out the ratio of X and Y in the Principal Component 1. This ratio is known as a “Linear Combination” of X and Y.

Principal Component 2 is the line through the origin and perpendicular to principal component 1 and best fitting the plot.

Project the points on PC1 and PC2 respectively.

Now we simply rotate the plot such that Principal component 1 is horizontal.

Now we use the previously projected points to find where the samples go in the PCA plot.

We can convert the Eigen values for PC1 and PC2 into variation around the origin by dividing them with sample size -1.

Now, for example, if variation for PC1 is 10 and the variation for PC2 is 5, we get 10+5 = 15 i.e. PC1 accounts for 10/15, i.e. 66.66% of total variation and PC2 is 33.33%.

Now, when we have more than two variables in our dataset, we simply find the Principal Components which are in equal number to the number of variables. i.e. if, the dataset has 4 variables, we find 4 Principal Components and after finding the % variation of each Principal component, we pick the ones which are highest and discard the rest. i.e. if we have 4 principal components and we only want 2 principal components, we take the 2 top principal components with the highest variation and plot the points using them.

**Procedure**

We obtained the dataset of the drug mortality rate per state in south eastern America from the internet and converted the dataset into CSV(Comma-separated values) format.

We then imported the dataset into our python code with the help of Pandas Library.

With the help of pandas methods, we separated the variables and sample in the imported csv file.

We then used SkLearn’s StandardScaler function to standardize the scale of our dataset to unit scale.

Now, we used SkLearn’s PCA function (Principal Component Analysis) on the imported dataset.

We now used matplotlib and numpy to visualize our data in the form of a graph.

# Python Program

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

%matplotlib

#loading dataset into Pandas DataFrame

df = pd.read\_csv(r'C:\Users\kingk\Desktop\PCA2.csv', names = ['Year', 'Deaths', 'Population', 'Crude Death Rate', 'Age-adjusted Rate', 'State'])

df.head()

features = ['Year', 'Deaths', 'Population', 'Crude Death Rate', 'Age-adjusted Rate']

x = df.loc[:, features].values

y = df.loc[:, ['State']].values

x = StandardScaler().fit\_transform(x)

pd.DataFrame(data = x, columns = features).head()

pca = PCA(n\_components = 2)

principalComponents = pca.fit\_transform(x)

principalDf = pd.DataFrame(data = principalComponents, columns = ['principal component 1', 'principal component 2'])

principalDf.head()

finalDf = pd.concat([principalDf, df[['State']]], axis = 1)

finalDf.head()

#VISUALIZING 2D PROJECTION

#Using a PCA projection to 2D to visualize the entire data set.

fig = plt.figure(figsize = (8,8))

ax = fig.add\_subplot(1,1,1)

ax.set\_xlabel('Principal Component 1', fontsize = 15)

ax.set\_ylabel('Principal Component 2', fontsize = 15)

ax.set\_title('2 Component PCA', fontsize = 20)

targets = ['Alabama', 'Arkansas', 'Florida', 'Georgia', 'Louisiana', 'Mississippi', 'North Carolina', 'Tennessee']

colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k', 'olive']

for target, color in zip(targets, colors):

indicesToKeep = finalDf['State'] == target

ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1'], finalDf.loc[indicesToKeep, 'principal component 2'],

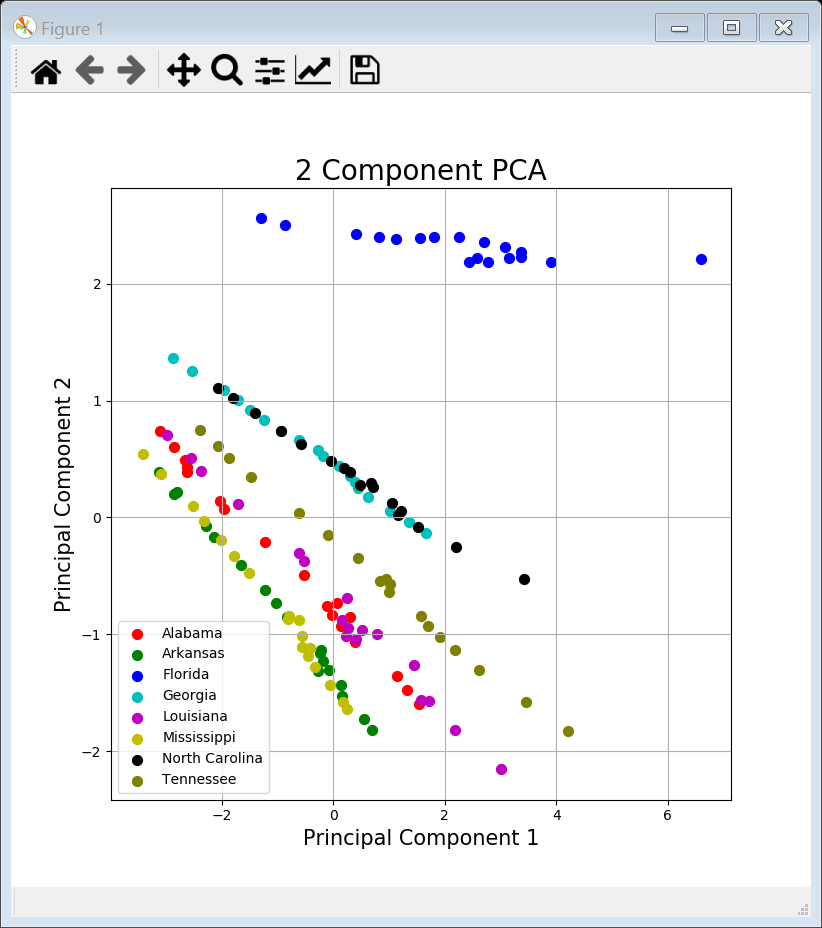
c = color, s = 50)

ax.legend(targets)

ax.grid()

**Results and Discussion**

The plot obtained after applying Principal Component Analysis Is,



From the final plot of the dataset, we can clearly state that the drug mortality rate of Alabama, Arkansas, Georgia, Louisiana, Mississippi, North Caroline and Tennessee are highly correlated to each other while Florida is least correlated to them.

We can also see the vast difference in the plots of Florida and rest of the states which indicates an alarming drug poisoning mortality rate in the state of Florida.

We can use this result to manage the distribution of the types of medicines to be supplied in these States. For example, we can reduce the supply of narcotic painkillers in the state of Florida or make the access to them restricted or we can increase the number of rehabs in the state of Florida.

In our result, the first principal component contained 65.8% of the variance of the original dataset and second principal component contained 27.7% of the variance. Together, the two components contained 93.5% of the information of the original dataset which makes it pretty accurate to the original data.

**References**

1.Dataset obtained from official data library of US government

url: <https://catalog.data.gov/dataset?_organization_limit=0&organization=hhs-gov&page=16>

2. Principal Component Analysis

url: <https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0>

url: <http://setosa.io/ev/principal-component-analysis/>

3.Pandas Library

url: <https://pandas.pydata.org/>

3.Scikit-learn Library

url: <https://scikit-learn.org/stable/>