Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering

Department of Computer Engineering

Mini Project 1 Report

On

PCA Visualization using Streamlit for Dataset using ML

Group Members

Bhavansh Gupta	BECOA134
Saurabh Bomble	BECOA114
Onkar Gaikar	BECOA129
Shubham Chalekar	BECOA115

Guide

Prof. Sushma Vispute

INDEX

Sr. No.	Content	Page No.
1.	Introduction	3
2.	Requirements and Dataset Description	4
3.	Theory	9
4.	Source Code / Application	11
5.	Output Screenshots	12
6.	Conclusion	16

1) INTRODUCTION

1.1 Motivation

In general, We took two dimension data to understand any machine algorithm and to draw data exploration. That is easy too to understand . But in real life data , we can get almost all data in higher dimensional form . And that is curse for supervised machine learning algorithm . Machine learning can not perform very well in higher dimension, and the curse of dimensionality is a very crucial problem. What happens when the given data set has too many variables? Here are few possible situations which you might come across:

- You find that most of the variables are correlated on analysis.
- You lose patience and decide to run a model on the whole data. This returns poor accuracy and you feel terrible.
- You become indecisive about what to do
- You start thinking of some strategic method to find few important variables

In such a scenario, Principal Component Analysis(PCA) plays a major part in efficiently reducing the dimensionality of the data yet retaining as much as possible of the variation present in the data set.

1.2 Objectives

- For prediction of class labels of given data instances, build classifier models using different techniques
- To analyze the confusion matrix and compare these models.
- To apply cross validation while preparing the training and testing datasets.
- To apply PCA to dataset and perform the classification
- To compare the difference between the metrics before and after applying PCA

2) REQUIREMENTS & DATASET DESCRIPTION

Software:

- Python 3.8 or above
- Python Packages
    - Pandas
    - Numpy
    - Matplotlib
    - Seaborn
    - Plotly
    - Sklearn
    - Streamlit

Dataset:

```
Wine recognition dataset
------------------------

**Data Set Characteristics:**

    :Number of Instances: 178 (50 in each of three classes)
    :Number of Attributes: 13 numeric, predictive attributes and the cl
ass
    :Attribute Information:
                - Alcohol
                - Malic acid
                - Ash
                - Alcalinity of ash
                - Magnesium
                - Total phenols
                - Flavanoids
                - Nonflavanoid phenols
                - Proanthocyanins
                - Color intensity
                - Hue
                - OD280/OD315 of diluted wines
                - Proline

    - class:
            - class_0
            - class_1
            - class_2

    :Summary Statistics:

    ============================= ==== ===== ======= =====
                                   Min   Max   Mean     SD
    ============================= ==== ===== ======= =====
    Alcohol:                      11.0  14.8    13.0   0.8
    Malic Acid:                   0.74  5.80    2.34  1.12
    Ash:                          1.36  3.23    2.36  0.27
    Alcalinity of Ash:            10.6  30.0    19.5   3.3
    Magnesium:                    70.0 162.0    99.7  14.3
    Total Phenols:                0.98  3.88    2.29  0.63
    Flavanoids:                   0.34  5.08    2.03  1.00
    Nonflavanoid Phenols:         0.13  0.66    0.36  0.12
    Proanthocyanins:              0.41  3.58    1.59  0.57
    Colour Intensity:              1.3  13.0     5.1   2.3
    Hue:                          0.48  1.71    0.96  0.23
    OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71
    Proline:                       278  1680     746   315
    ============================= ==== ===== ======= =====

    :Missing Attribute Values: None
    :Class Distribution: class_0 (59), class_1 (71), class_2 (48)
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988
```

```
Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the cla
ss
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
    ============== ==== ==== ======= ===== ====================

    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988
```

```
Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the cl
ass

    :Attribute Information:
        - radius (mean of distances from center to points on the perime
ter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
        - fractal dimension ("coastline approximation" - 1)

        The mean, standard error, and "worst" or largest (mean of the t
hree
        worst/largest values) of these features were computed for each
image,
        resulting in 30 features.  For instance, field 0 is Mean Radius
, field
        10 is Radius SE, field 20 is Worst Radius.

        - class:
                - WDBC-Malignant
                - WDBC-Benign

    :Summary Statistics:

    ===================================== ====== ======
                                           Min    Max
    ===================================== ====== ======
    radius (mean):                        6.981  28.11
    texture (mean):                       9.71   39.28
    perimeter (mean):                     43.79  188.5
    area (mean):                          143.5  2501.0
    smoothness (mean):                    0.053  0.163
    compactness (mean):                   0.019  0.345
    concavity (mean):                     0.0    0.427
    concave points (mean):                0.0    0.201
    symmetry (mean):                      0.106  0.304
    fractal dimension (mean):             0.05   0.097
    radius (standard error):              0.112  2.873
    texture (standard error):             0.36   4.885
    perimeter (standard error):           0.757  21.98
    area (standard error):                6.802  542.2
    smoothness (standard error):          0.002  0.031
    compactness (standard error):         0.002  0.135
    concavity (standard error):           0.0    0.396
    concave points (standard error):      0.0    0.053
```

```
    symmetry (standard error):            0.008  0.079
    fractal dimension (standard error):   0.001  0.03
    radius (worst):                       7.93   36.04
    texture (worst):                      12.02  49.54
    perimeter (worst):                    50.41  251.2
    area (worst):                         185.2  4254.0
    smoothness (worst):                   0.071  0.223
    compactness (worst):                  0.027  1.058
    concavity (worst):                    0.0    1.252
    concave points (worst):               0.0    0.291
    symmetry (worst):                     0.156  0.664
    fractal dimension (worst):            0.055  0.208
    ===================================== ====== ======

    :Missing Attribute Values: None

    :Class Distribution: 212 - Malignant, 357 - Benign

    :Creator:  Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasar
ian

    :Donor: Nick Street

    :Date: November, 1995
```

3) THEORY

Principal Component Analysis (PCA): Principal Component Analysis (PCA) is one of famous techniques for dimension reduction, feature extraction, and data visualization. In general, PCA is defined by a transformation of a high dimensional vector space into a low dimensional space. Let's consider visualization of 10-dim data. It is barely possible to effectively show the shape of such high dimensional data distribution. PCA provides an efficient way to reduce the dimensionality (i.e., from 10 to 2), so it is much easier to visualize the shape of data distribution. PCA is also useful in the modeling of robust classifier where considerably small number of high dimensional training data is provided. By reducing the dimensions of learning data sets, PCA provides an effective and efficient method for data description and classification.

In simple words, PCA is a method of obtaining important variables (in form of components) from a large set of variables available in a data set. It extracts low dimensional set of features by taking a projection of irrelevant dimensions from a high dimensional data set with a motive to capture as much information as possible. With fewer variables obtained while minimizing the loss of information, visualization also becomes much more meaningful. PCA is more useful when dealing with 3 or higher dimensional data.

K-nearest-neighbours (kNN) algorithm: It is a simple supervised learning algorithm in pattern recognition. It is one of the most popular neighborhood classifiers due to its simplicity and efficiency in the field of machine learning. KNN algorithm stores all cases and classifies new cases based on similarity measures; it searches the pattern space for the k training tuples that are closest to the unknown tuples. The performance depends on the optimal number of neighbors (k) chosen, which is different from one data sample to another.

Logistic regression: In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one. Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable (target) is categorical.

4) SOURCE CODE / FUNCTIONS

The source code can be accessed on GitHub at this link -

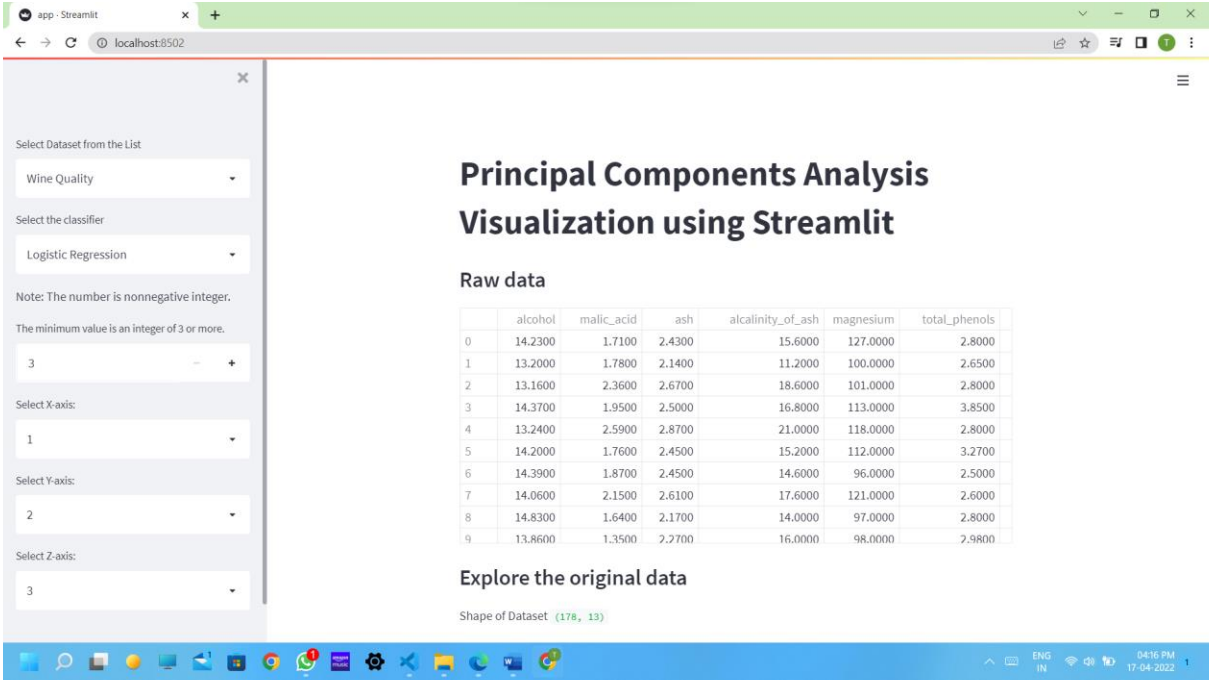
bhavansh/Visualize-PCA-using-Streamlit (github.com)

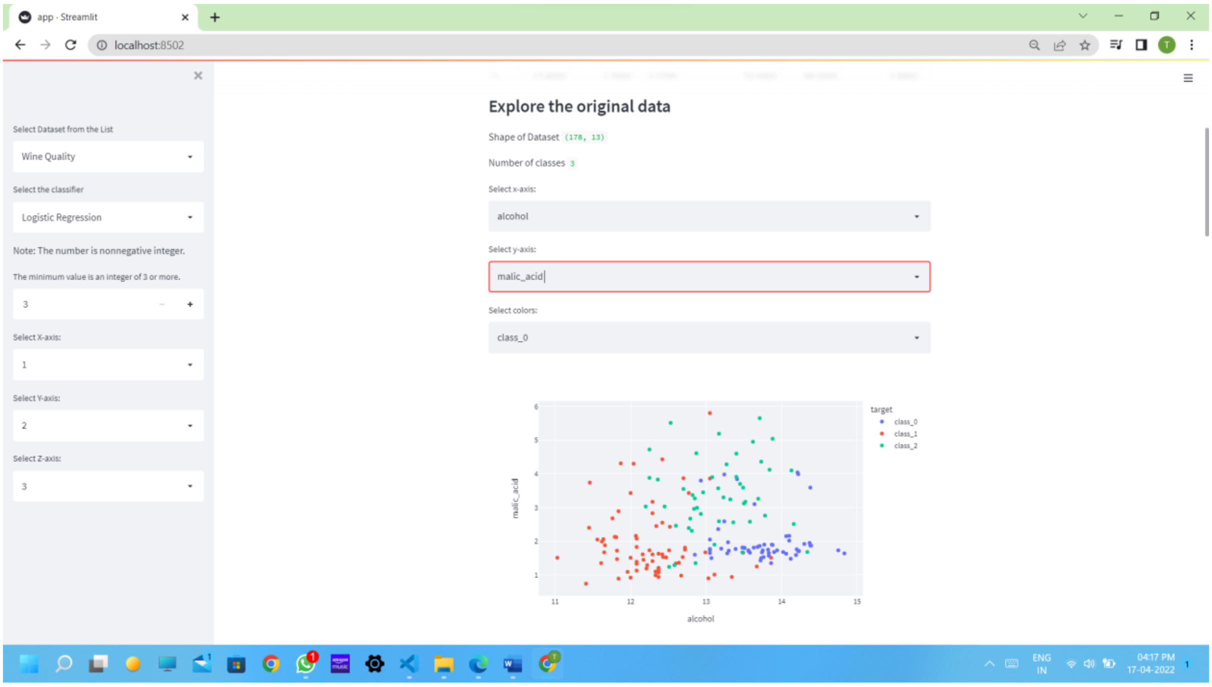
The following URL :
https://github.com/bhavansh/Visualize-PCA-using-Streamlit

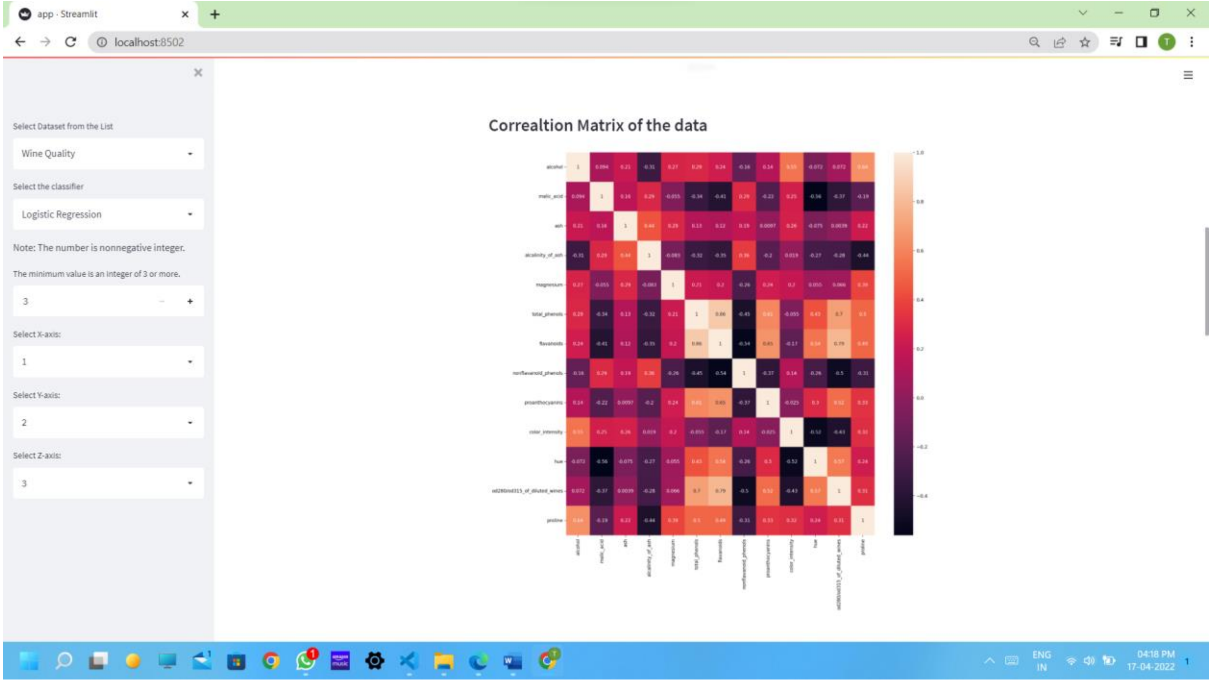
The hosted application can be accessed using the following URL –

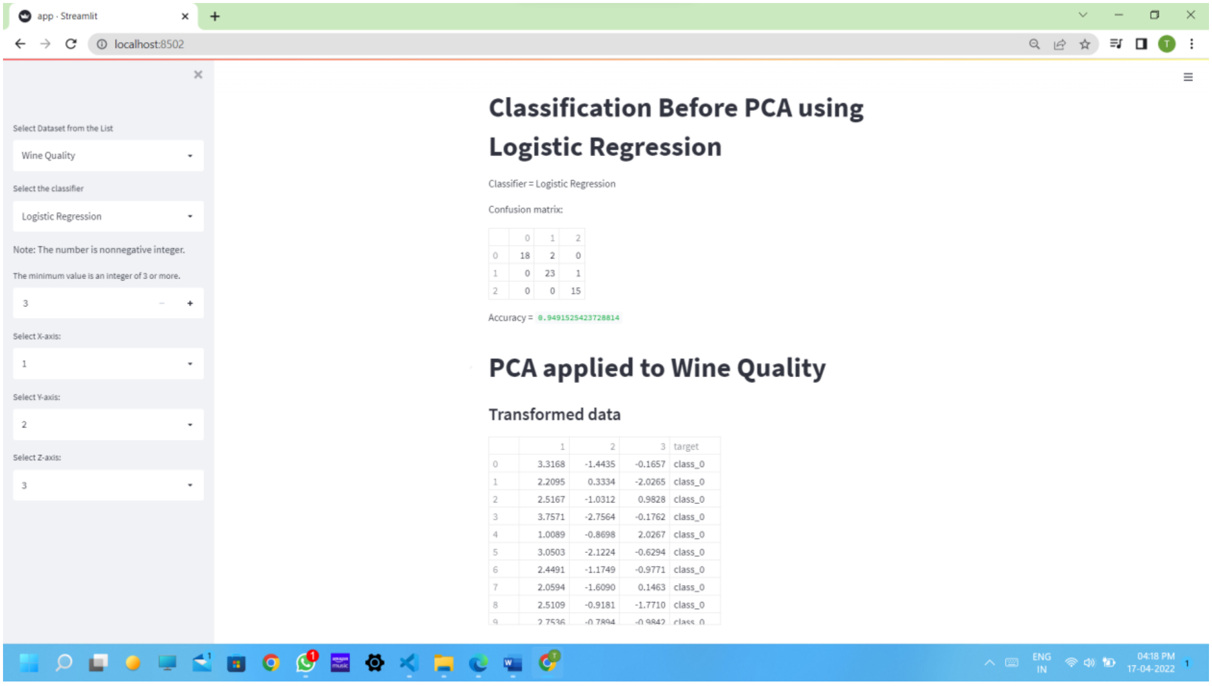
https://share.streamlit.io/bhavansh/visualize-pca-using-streamlit/app.py

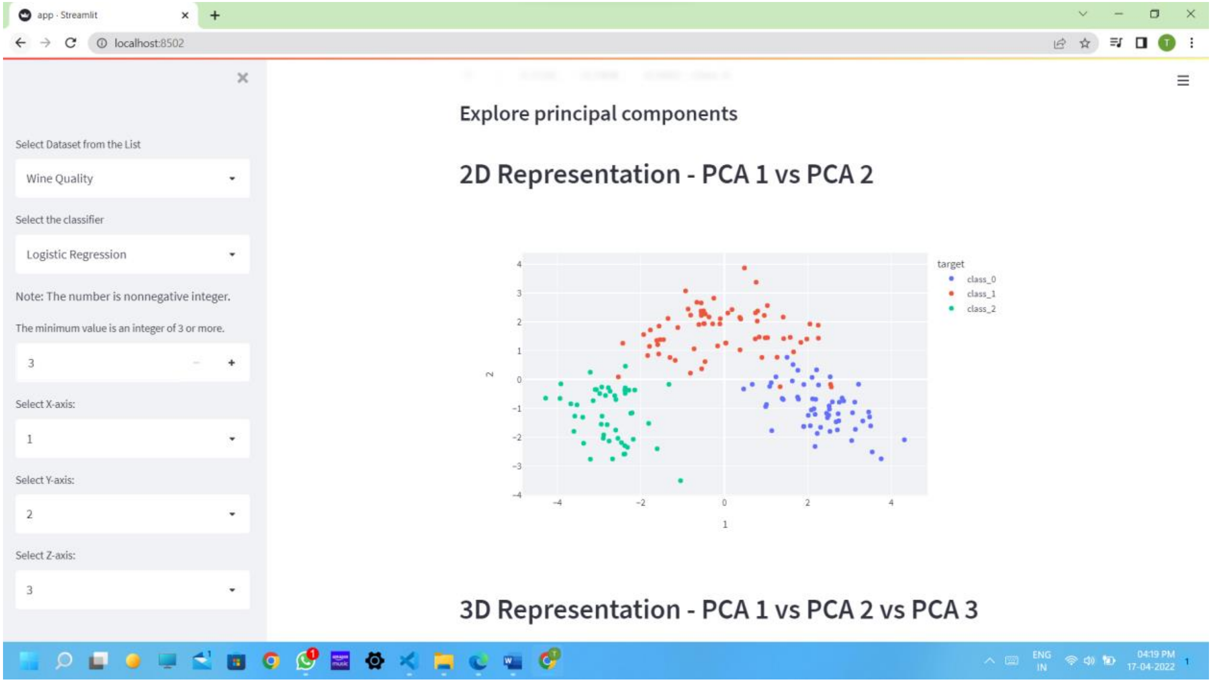
5) OUTPUT SCREENSHOTS

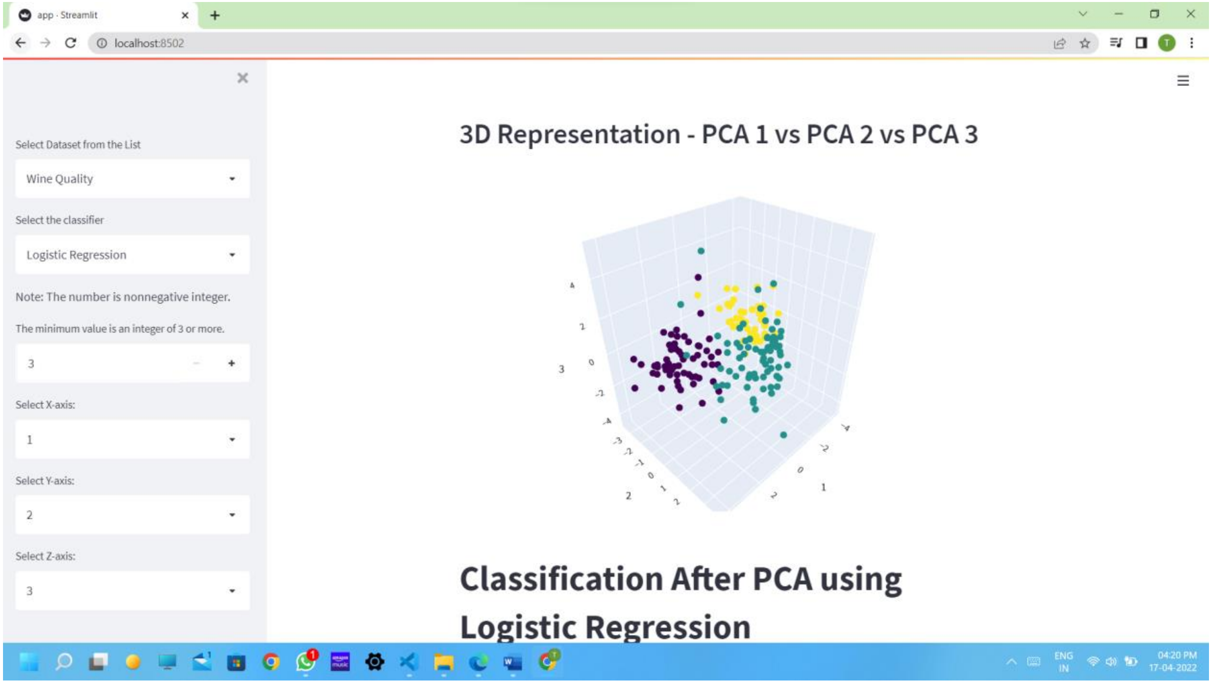


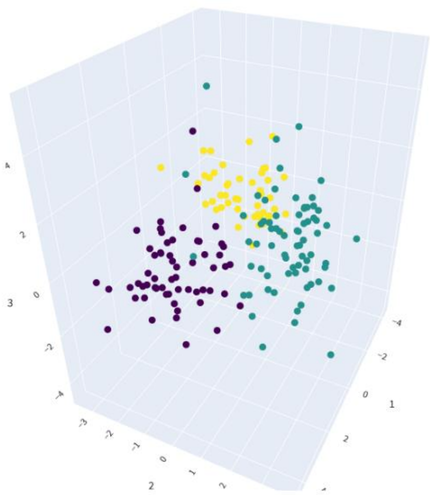


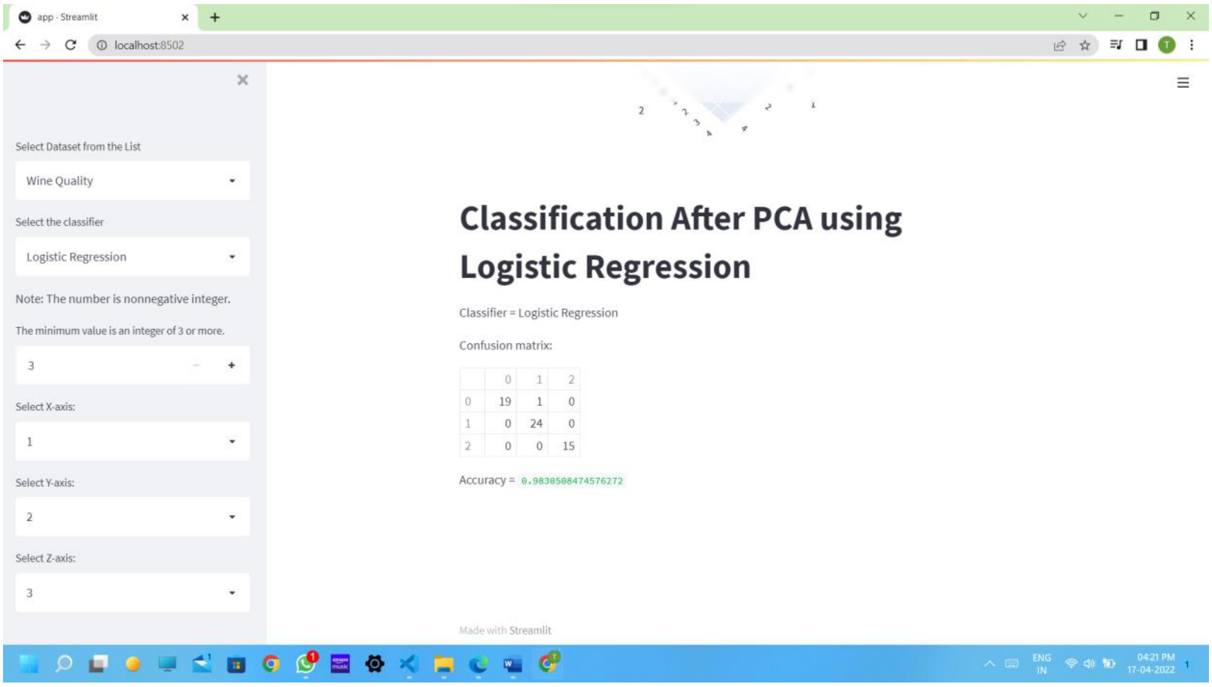

app · Streamlit
localhost:8502
Select Dataset from the List
Wine Quality
Select the classifier
Logistic Regression
Note: The number is nonnegative integer.
The minimum value is an integer of 3 or more.
3
Select X-axis:
1
Select Y-axis:
2
Select Z-axis:
3
Correaltion Matrix of the data
04:18 PM
17-04-2022


app · Streamlit
localhost:8502
Select Dataset from the List
Wine Quality
Select the classifier
Logistic Regression
Note: The number is nonnegative integer.
The minimum value is an integer of 3 or more.
3
Select X-axis:
1
Select Y-axis:
2
Select Z-axis:
3
Classification Before PCA using Logistic Regression
Classifier = Logistic Regression
Confusion matrix:
0 1 2
0 18 2 0
1 0 23 1
2 0 0 15
Accuracy = 0.9491525423728814
PCA applied to Wine Quality
Transformed data
1 2 3 target
0 3.3168 -1.4435 -0.1657 class_0
1 2.2095 0.3334 -2.0265 class_0
2 2.5167 -1.0312 0.9828 class_0
3 3.7571 -2.7564 -0.1762 class_0
4 1.0089 -0.8698 2.0267 class_0
5 3.0503 -2.1224 -0.6294 class_0
6 2.4491 -1.1749 -0.9771 class_0
7 2.0594 -1.6090 0.1463 class_0
8 2.5109 -0.9181 -1.7710 class_0
04:18 PM
17-04-2022


app - Streamlit
localhost:8502
Select Dataset from the List
Wine Quality
Select the classifier
Logistic Regression
Note: The number is nonnegative integer.
The minimum value is an integer of 3 or more.
3
Select X-axis:
1
Select Y-axis:
2
Select Z-axis:
3
Explore principal components
2D Representation - PCA 1 vs PCA 2
target
class_0
class_1
class_2
3D Representation - PCA 1 vs PCA 2 vs PCA 3
04:19 PM
17-04-2022


app - Streamlit
localhost:8502
Select Dataset from the List
Wine Quality
Select the classifier
Logistic Regression
Note: The number is nonnegative integer.
The minimum value is an integer of 3 or more.
3
Select X-axis:
1
Select Y-axis:
2
Select Z-axis:
3
3D Representation - PCA 1 vs PCA 2 vs PCA 3
Classification After PCA using
Logistic Regression
04:20 PM
17-04-2022




app · Streamlit
localhost:8502
Select Dataset from the List
Wine Quality
Select the classifier
Logistic Regression
Note: The number is nonnegative integer.
The minimum value is an integer of 3 or more.
3
Select X-axis:
1
Select Y-axis:
2
Select Z-axis:
3
Classification After PCA using Logistic Regression
Classifier = Logistic Regression
Confusion matrix:
	0	1	2
0	19	1	0
1	0	24	0
2	0	0	15
Accuracy = 0.9830508474576272
Made with Streamlit
ENG IN
04:21 PM
17-04-2022

6) CONCLUSION & FUTURE SCOPE

In this Python project, we have successfully implemented PCA (Principal Component Analysis) on three standard SKLEARN datasets. We discovered that on datasets with larger number of attributes PCA does have any significant effect on the output compared to the datasets with smaller number of attributes.
We have also visualized the PCA components generated in 2D and 3D for better understanding of the PCA.