FEATURE SELECTION TECHNIQUES ON AIR POLLUTION DATASET

Table of contents Page

INTRODUCTION

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve. Irrelevant or partially relevant features can negatively impact model performance. Feature selection and Data cleaning should be the first and most important step of your model designing. Below is some information on the algorithms used in the project.

CORRELATION SCORE

A correlation coefficient is a number between -1 and 1 that **tells you the strength and direction of a relationship between variables.** In other words, it reflects how similar the measurements of two or more variables are across a dataset.

There are three type of feature selection we used in our dataset-

- PCA
- LDA
- Correlation Method

PRINCIPAL COMPONENT ANALYSIS

The Principal Component Analysis is **a popular unsupervised learning technique for reducing the dimensionality of data**. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D.The steps to perform PCA are the following:

1- Standardize the data.

2- Compute the covariance matrix of the features from the dataset.

3- Perform Eigen decomposition on the covariance matrix.

4- Order the eigenvectors in decreasing order based on the magnitude of their corresponding eigenvalues.

5- Determine k, the number of top principal components to select.

6- Construct the projection matrix from the chosen number of top principal components.

7- Compute the new k-dimensional feature space.

LINEAR DISCRIMINANT ANALYSIS

Linear Discriminant Analysis (LDA) is one of the commonly used dimensionality reduction techniques in machine learning to solve more than two-class classification problems. It is also known as Normal Discriminant Analysis (NDA) or Discriminant Function Analysis (DFA).The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs.

DIFFERENCE BETWEEN LDA AND PCA

PCA is an unsupervised learning algorithm while LDA is a supervised learning algorithm. This means that PCA finds directions of maximum variance regardless of class labels while LDA finds directions of maximum class separability.

CORRELATION METHOD-

Correlation is **a statistical calculation that indicates that two variables are parallelly related** (which means that the variables change together at a constant rate). It is a simple and popularly used tool for defining relationships without delivering a statement concerning the cause and effect.Correlation is an indication about the changes between two variables.

DATASET DESCRIPTION

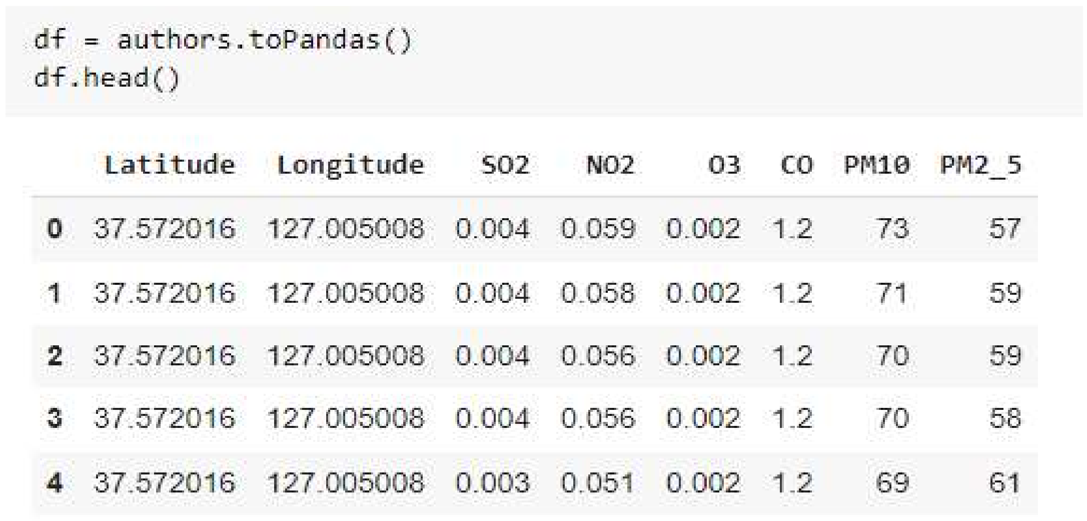
We took a dataset from kaggle called Air pollution in Seoul.This data provides average values for six pollutants (SO2, NO2, CO, O3, PM10, PM2.5).This dataset contains 8 columns and 647511 rows.

IMPLEMENTATION

Now we would talk about the implementation of the project, How it has been implemented and explanation of the steps taken.

Firstly, we do preprocessing and normalization in Pyspark to make it more legitimate to use for further data Analysis process, code snippet for same given below:-

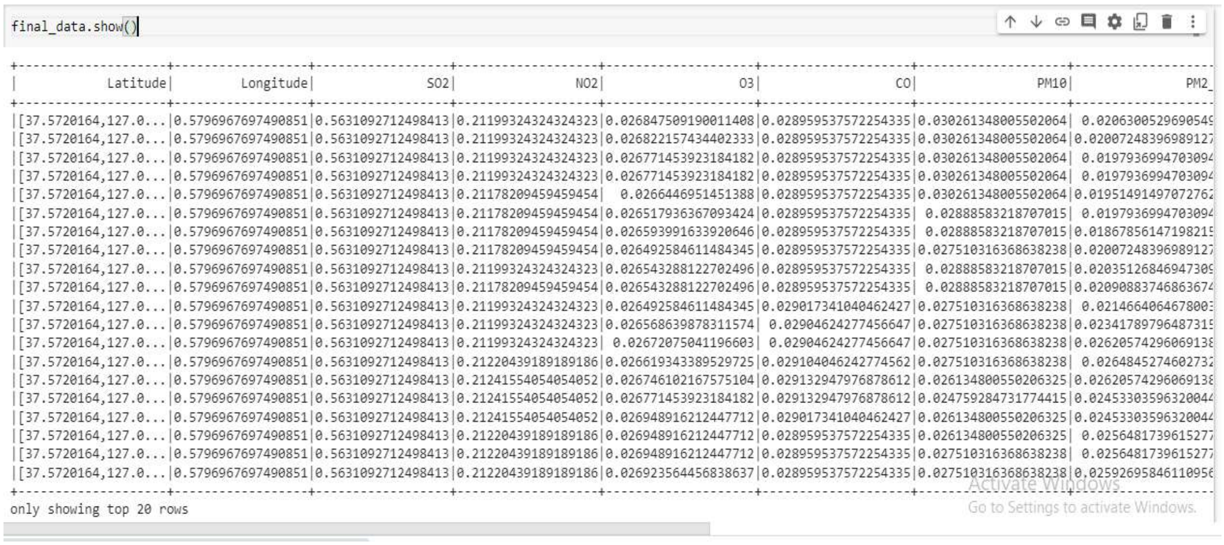
Dataframe before normalization:-



```
df = authors.toPandas()
df.head()
```

	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2_5
0	37.572016	127.005008	0.004	0.059	0.002	1.2	73	57
1	37.572016	127.005008	0.004	0.058	0.002	1.2	71	59
2	37.572016	127.005008	0.004	0.056	0.002	1.2	70	59
3	37.572016	127.005008	0.004	0.056	0.002	1.2	70	58
4	37.572016	127.005008	0.003	0.051	0.002	1.2	69	61

Dataframe after normalization:-



```
final_data.show()
```

```
+--------------------+------------------+------------------+-------------------+--------------------+--------------------+--------------------+-------------------
|            Latitude|         Longitude|               SO2|                NO2|                  O3|                  CO|                PM10|               PM2_
+--------------------+------------------+------------------+-------------------+--------------------+--------------------+--------------------+-------------------
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323|0.026847509190011408|0.028959537572254335|0.030261348005502064| 0.0206300529690549
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323|0.026822157434402333|0.028959537572254335|0.030261348005502064|0.0200724839698912
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323|0.026771453923184182|0.028959537572254335|0.030261348005502064| 0.019793699470309
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323|0.026771453923184182|0.028959537572254335|0.030261348005502064| 0.019793699470309
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21178209459459454|  0.0266446951451388|0.028959537572254335|0.030261348005502064|0.0195149149707276
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21178209459459454|0.026517936367093424|0.028959537572254335| 0.02888583218707015| 0.019793699470309
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21178209459459454|0.026593991633920646|0.028959537572254335| 0.02888583218707015|0.0186785614719821
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21178209459459454|0.026492584611484345|0.028959537572254335|0.027510316368638238|0.0200724839698912
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323|0.026543288122702496|0.028959537572254335| 0.02888583218707015|0.0203512684694730
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21178209459459454|0.026543288122702496|0.028959537572254335| 0.02888583218707015|0.0209088374686367
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323|0.026492584611484345|0.029017341040462427|0.027510316368638238| 0.021466406467800
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323|0.026568639878311574| 0.02904624277456647|0.027510316368638238|0.0234178979648731
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21199324324324323| 0.02672075041196603| 0.02904624277456647|0.027510316368638238|0.0262057429606913
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21220439189189186|0.026619343389529725|0.029104046242774562|0.027510316368638238| 0.026484527460273
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21241554054054052|0.026746102167575104|0.029132947976878612|0.026134800550206325|0.0262057429606913
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21241554054054052|0.026771453923184182|0.029132947976878612|0.024759284731774415|0.0245330359632004
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21241554054054052|0.026948916212447712|0.029017341040462427|0.026134800550206325|0.0245330359632004
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21220439189189186|0.026948916212447712|0.028959537572254335|0.026134800550206325| 0.025648173961527
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21220439189189186|0.026948916212447712|0.028959537572254335|0.027510316368638238| 0.025648173961527
|[37.5720164,127.0...|0.5796967697490851|0.5631092712498413|0.21220439189189186|0.026923564456838637|0.028959537572254335|0.027510316368638238|0.0259269584611095
+--------------------+------------------+------------------+-------------------+--------------------+--------------------+--------------------+-------------------
only showing top 20 rows
```

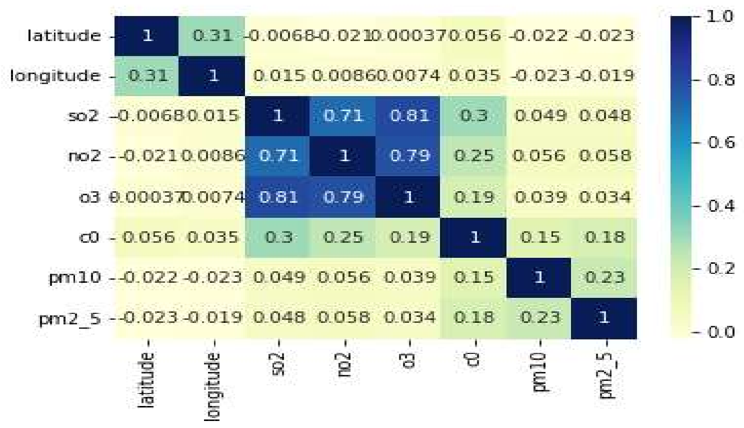
It is returning all the columns along with the normalized form of those columns. Other preprocessing has also been done in order to use the data for dimensionality reduction.

We do Dimensional Reduction of our dataset by the following three way-

1- Correlation Method

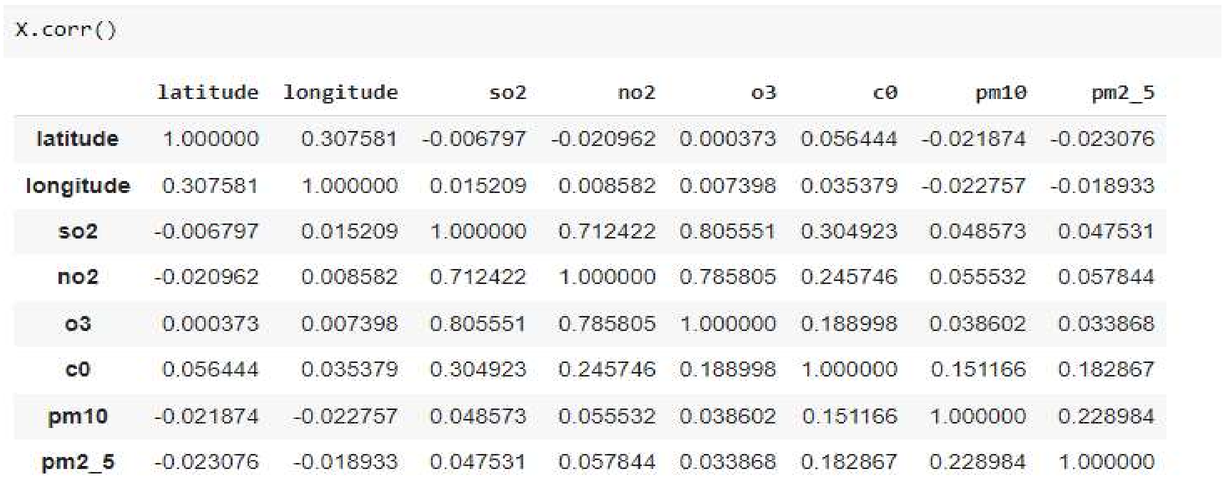
Firstly , we will find the correlation feature of our normalized dataset as below-

```
dataplot=sns.heatmap(X.corr(),cmap="YlGnBu", annot=True)
```



Above heat map shows the correlation value between each feature which would eventually help us in dimensionality reduction by eliminating most correlated features.

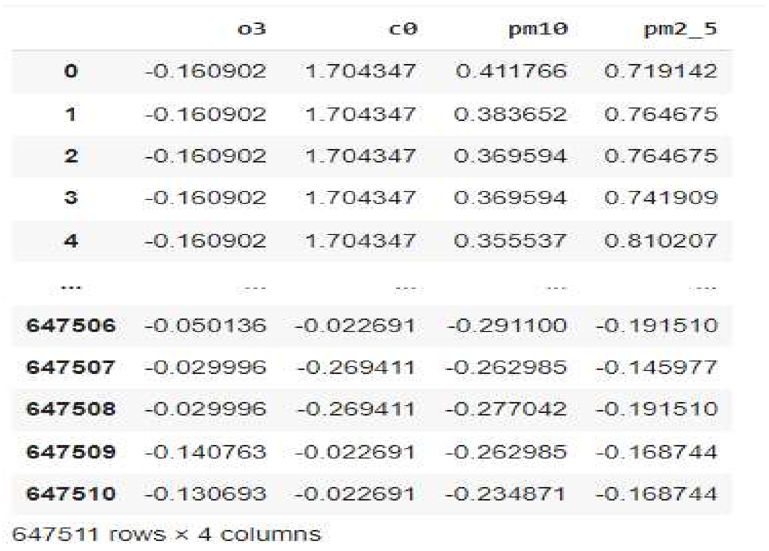
Below screenshot shows the simple correlation values between each feature without heatmap.



```
X.corr()
```

	latitude	longitude	so2	no2	o3	c0	pm10	pm2_5
latitude	1.000000	0.307581	-0.006797	-0.020962	0.000373	0.056444	-0.021874	-0.023076
longitude	0.307581	1.000000	0.015209	0.008582	0.007398	0.035379	-0.022757	-0.018933
so2	-0.006797	0.015209	1.000000	0.712422	0.805551	0.304923	0.048573	0.047531
no2	-0.020962	0.008582	0.712422	1.000000	0.785805	0.245746	0.055532	0.057844
o3	0.000373	0.007398	0.805551	0.785805	1.000000	0.188998	0.038602	0.033868
c0	0.056444	0.035379	0.304923	0.245746	0.188998	1.000000	0.151166	0.182867
pm10	-0.021874	-0.022757	0.048573	0.055532	0.038602	0.151166	1.000000	0.228984
pm2_5	-0.023076	-0.018933	0.047531	0.057844	0.033868	0.182867	0.228984	1.000000

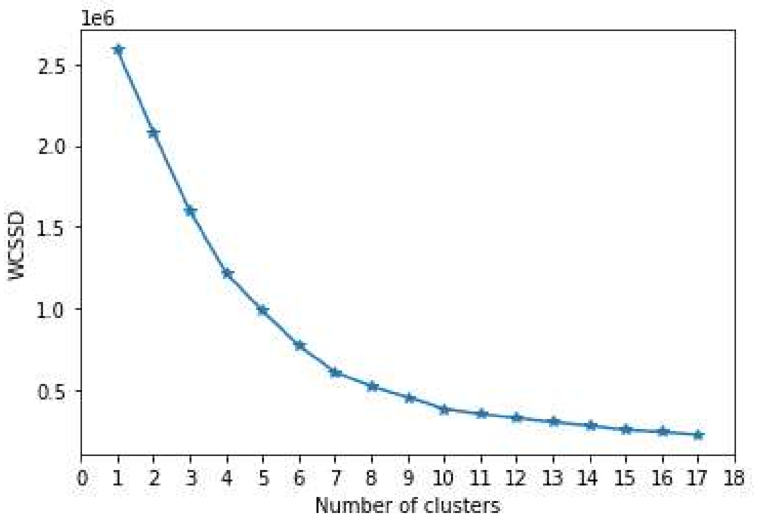
As we can see from above heatmap and correlation table the correlation between O3 and sO2 ,O3 and NO2 are high so we will take only one feature from them i.e O3 and drop other two feature i.e. NO2, SO2 and we also drop latitude and longitude columns as they do not give any useful information. So, after dropping these columns our dataset contains only 4 columns as shown below.



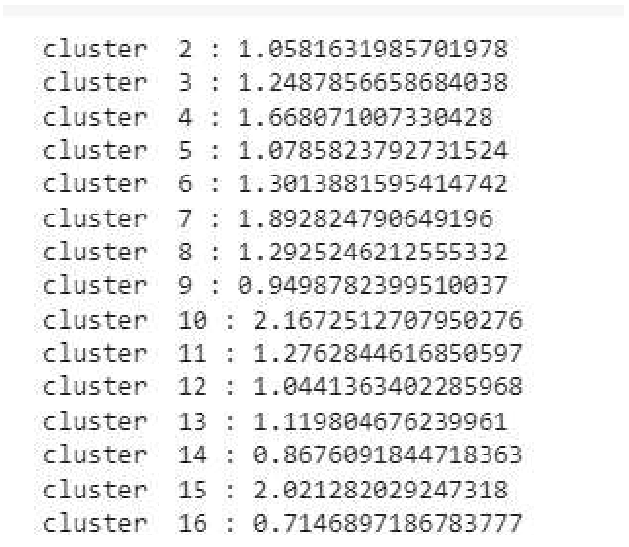
	o3	c0	pm10	pm2_5
0	-0.160902	1.704347	0.411766	0.719142
1	-0.160902	1.704347	0.383652	0.764675
2	-0.160902	1.704347	0.369594	0.764675
3	-0.160902	1.704347	0.369594	0.741909
4	-0.160902	1.704347	0.355537	0.810207
...	...	...	...	...
647506	-0.050136	-0.022691	-0.291100	-0.191510
647507	-0.029996	-0.269411	-0.262985	-0.145977
647508	-0.029996	-0.269411	-0.277042	-0.191510
647509	-0.140763	-0.022691	-0.262985	-0.168744
647510	-0.130693	-0.022691	-0.234871	-0.168744

647511 rows × 4 columns

In order to find accuracy of this method we have to label our extracted dataset by applying clustering algorithm i.e. K-Means algorithm. In order to find a suitable number of clusters we have to apply the elbow method and on the basis of WCSSD (weighted centroid sum of squared distance) i.e elbow method, we will choose the value for number of clusters on the basis of highest wcssd. So, here we choose the number of clusters as 10.

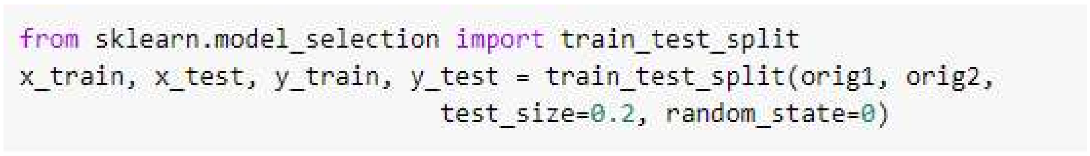


WCSSD Values for each clusters are-

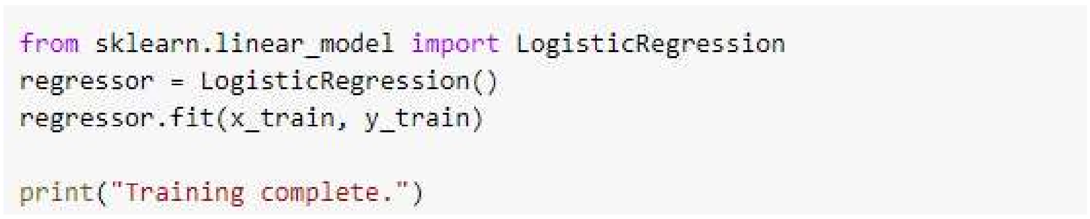


```
cluster  2 : 1.0581631985701978
cluster  3 : 1.2487856658684038
cluster  4 : 1.668071007330428
cluster  5 : 1.0785823792731524
cluster  6 : 1.3013881595414742
cluster  7 : 1.892824790649196
cluster  8 : 1.2925246212555332
cluster  9 : 0.9498782399510037
cluster  10 : 2.1672512707950276
cluster  11 : 1.2762844616850597
cluster  12 : 1.0441363402285968
cluster  13 : 1.119804676239961
cluster  14 : 0.8676091844718363
cluster  15 : 2.021282029247318
cluster  16 : 0.7146897186783777
```

Now we add one more column in our extracted dataset i.e. cluster which contains the labeled cluster for sample. Now, our unlabeled dataset gets converted into labeled dataset. For finding accuracy of our model we have to split our dataset into train and test. Here, the train part to train our model and test is used for testing and predicting our model. We are applying a Logistic regression model since target variables are categorical in nature.

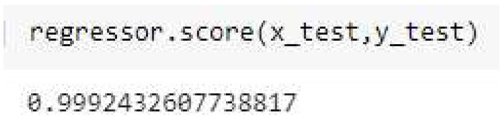


```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(orig1, orig2,
                                        test_size=0.2, random_state=0)
```



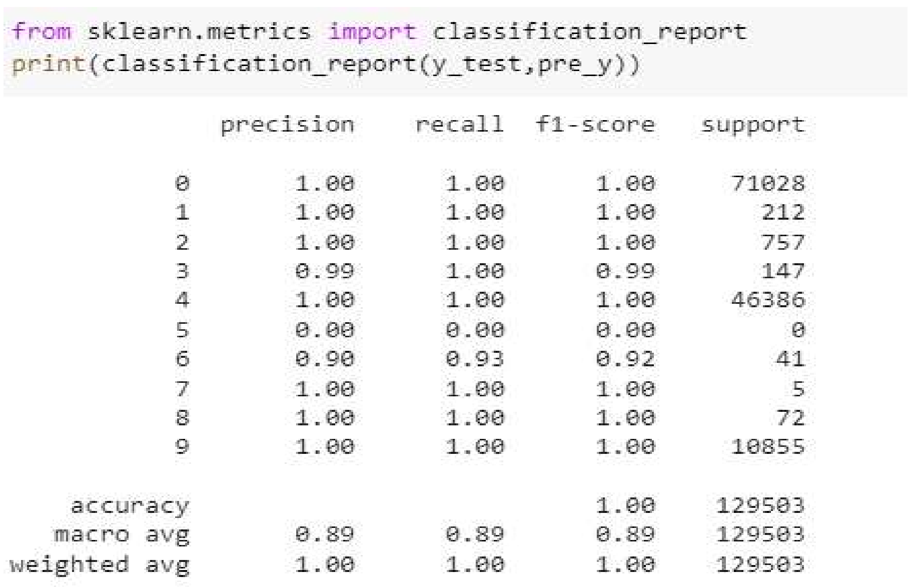
```
from sklearn.linear_model import LogisticRegression
regressor = LogisticRegression()
regressor.fit(x_train, y_train)

print("Training complete.")
```

After fitting this model in our dataset the accuracy of our method comes out to be 99.92% as shown in the below snippet.



```
regressor.score(x_test,y_test)

0.9992432607738817
```

Evaluation metrics for this method are as given below-



```
from sklearn.metrics import classification_report
print(classification_report(y_test,pre_y))
```

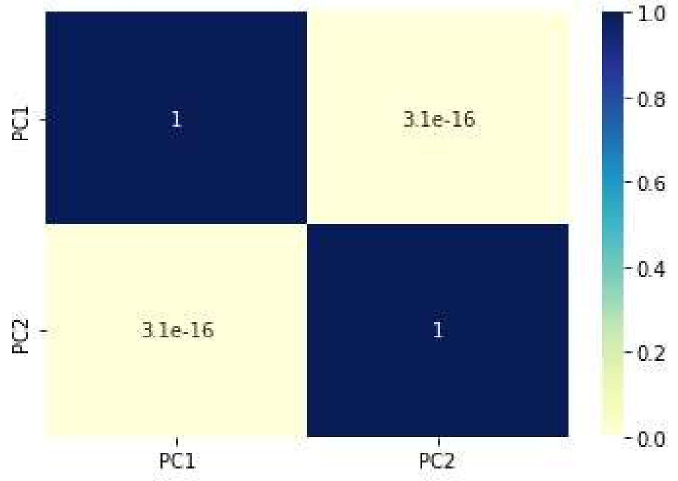
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     71028
           1       1.00      1.00      1.00       212
           2       1.00      1.00      1.00       757
           3       0.99      1.00      0.99       147
           4       1.00      1.00      1.00     46386
           5       0.00      0.00      0.00         0
           6       0.90      0.93      0.92        41
           7       1.00      1.00      1.00         5
           8       1.00      1.00      1.00        72
           9       1.00      1.00      1.00     10855

    accuracy                           1.00    129503
   macro avg       0.89      0.89      0.89    129503
weighted avg       1.00      1.00      1.00    129503
```

2- Principal Component Analysis

We take normalized data and apply PCA algorithm for dimensionality reduction and take top 2 columns which cover most of our data with less correlation between them as shown in heatmap

```
sns.heatmap(data_pca.corr(),cmap="YlGnBu", annot=True)
```

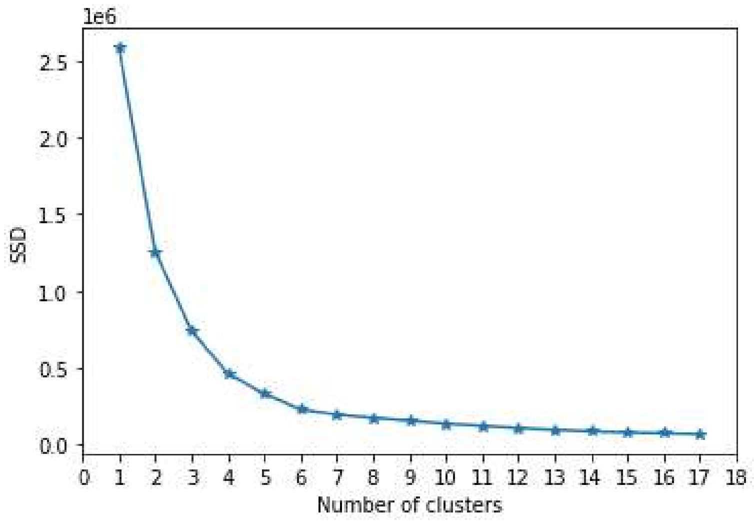
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa1eaecc910>
```



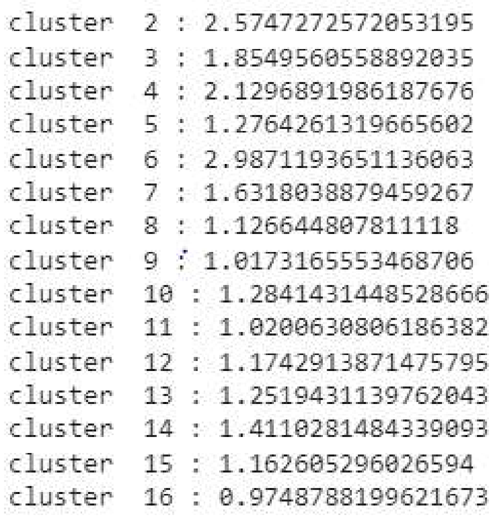
```
data_pca.corr()
```

	PC1	PC2
PC1	1.000000e+00	3.100067e-16
PC2	3.100067e-16	1.000000e+00

Now for predicting the accuracy of pca firstly ,we have to label our dataset by doing clustering. For this we have to apply the K-means algorithm. In order to find a suitable number of clusters we have to apply the elbow method and on the basis of WCSSD (weighted centroid sum of squared distance) i.e elbow method, we will choose the value for number of clusters on the basis of highest wcssd. So, here we choose the number of clusters as 6.

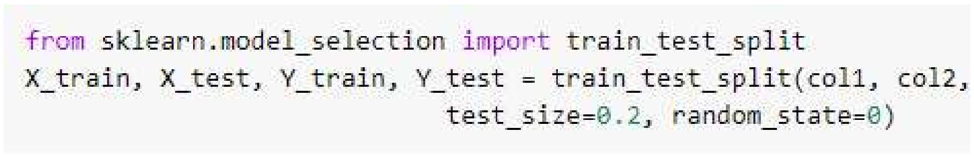


WCSSD Values for each clusters are-

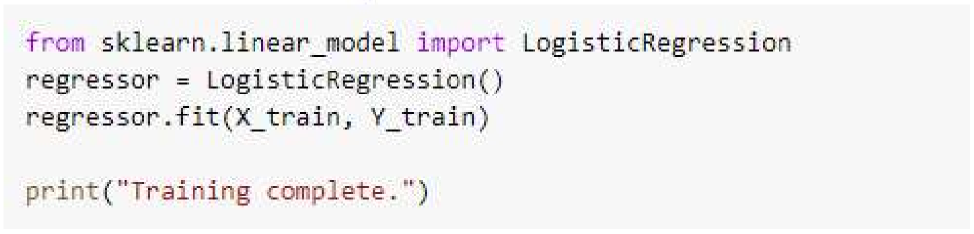


```
cluster  2 : 2.5747272572053195
cluster  3 : 1.8549560558892035
cluster  4 : 2.1296891986187676
cluster  5 : 1.2764261319665602
cluster  6 : 2.9871193651136063
cluster  7 : 1.6318038879459267
cluster  8 : 1.126644807811118
cluster  9 : 1.0173165553468706
cluster  10 : 1.2841431448528666
cluster  11 : 1.0200630806186382
cluster  12 : 1.1742913871475795
cluster  13 : 1.2519431139762043
cluster  14 : 1.4110281484339093
cluster  15 : 1.162605296026594
cluster  16 : 0.9748788199621673
```

Now we add one more column in our extracted dataset i.e. cluster which contains the labeled cluster for sample. Now, our unlabeled dataset gets converted into labeled dataset. For finding accuracy of our model we have to split our dataset into train and test. Here, the train part to train our model and test is used for testing and predicting our model. We are applying a Logistic regression model since target variables are categorical in nature.

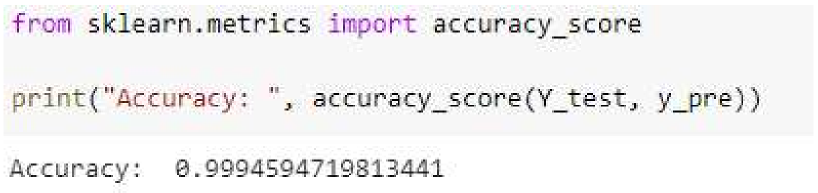


```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(col1, col2,
                                test_size=0.2, random_state=0)
```



```
from sklearn.linear_model import LogisticRegression
regressor = LogisticRegression()
regressor.fit(X_train, Y_train)

print("Training complete.")
```

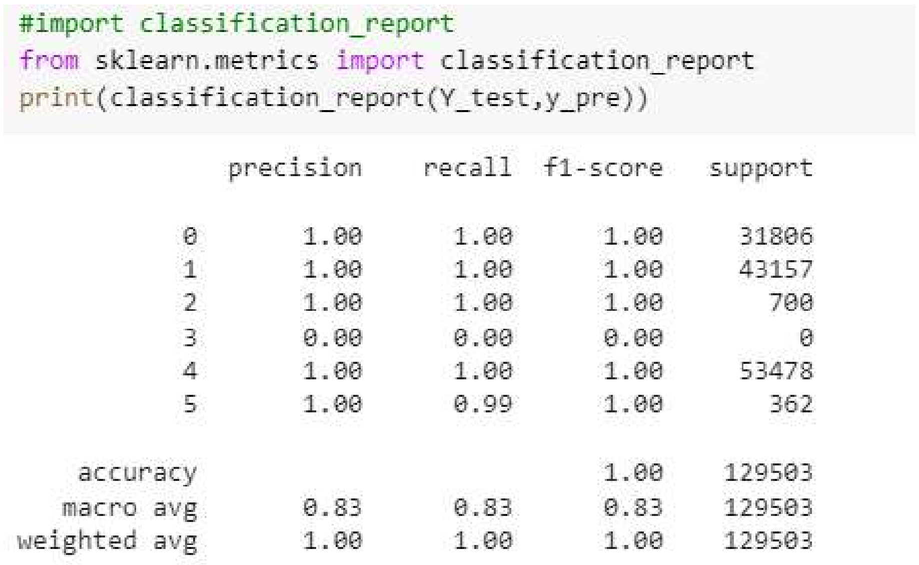
After fitting this model in our dataset the accuracy of our method comes out to be 99.94% as shown in the below snippet.



```
from sklearn.metrics import accuracy_score

print("Accuracy: ", accuracy_score(Y_test, y_pre))
```

```
Accuracy:  0.9994594719813441
```

Evaluation metrics for this method are as given below-

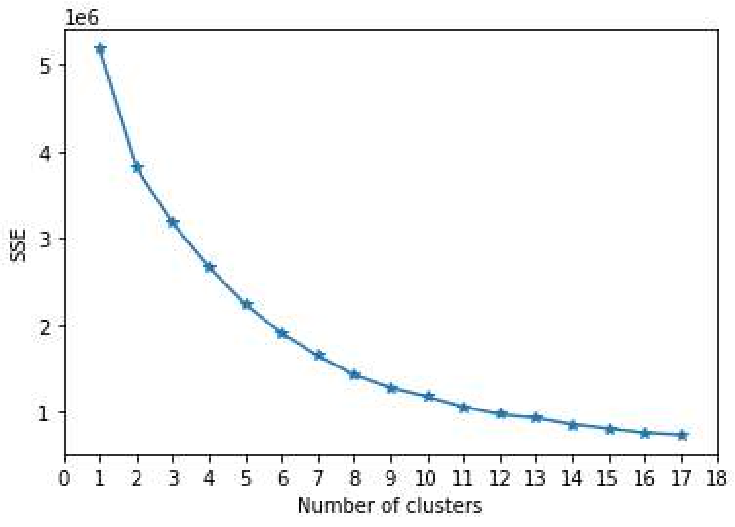


```
#import classification_report
from sklearn.metrics import classification_report
print(classification_report(Y_test,y_pre))
```

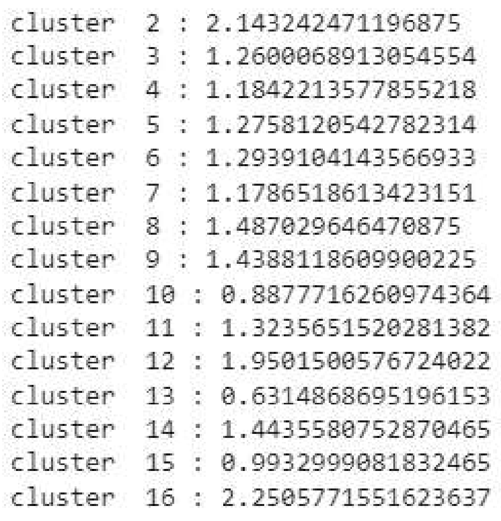
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     31806
           1       1.00      1.00      1.00     43157
           2       1.00      1.00      1.00       700
           3       0.00      0.00      0.00         0
           4       1.00      1.00      1.00     53478
           5       1.00      0.99      1.00       362

    accuracy                           1.00    129503
   macro avg       0.83      0.83      0.83    129503
weighted avg       1.00      1.00      1.00    129503
```

3- Linear Discriminant Analysis

The LDA algorithm is applied on a labeled dataset so for making our unlabeled dataset to labeled dataset ,we have to do clustering. For this we have to apply the K-means algorithm. In order to find a suitable number of clusters we have to apply the elbow method and on the basis of WCSSD (weighted centroid sum of squared distance) i.e elbow method, we will choose the value for number of clusters on the basis of highest wcssd. So, here we choose the number of clusters as 16.

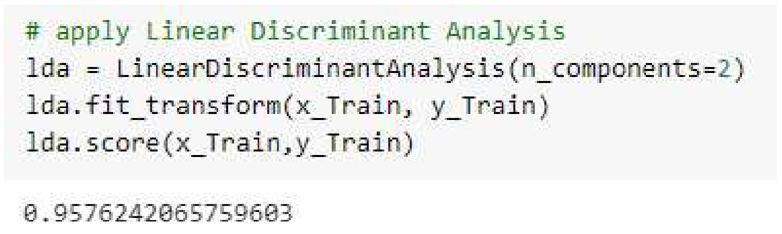


WCSSD Values for each clusters are-



```
cluster  2 : 2.143242471196875
cluster  3 : 1.2600068913054554
cluster  4 : 1.1842213577855218
cluster  5 : 1.2758120542782314
cluster  6 : 1.2939104143566933
cluster  7 : 1.1786518613423151
cluster  8 : 1.487029646470875
cluster  9 : 1.4388118609900225
cluster  10 : 0.8877716260974364
cluster  11 : 1.3235651520281382
cluster  12 : 1.9501500576724022
cluster  13 : 0.6314868695196153
cluster  14 : 1.4435580752870465
cluster  15 : 0.9932999081832465
cluster  16 : 2.2505771551623637
```

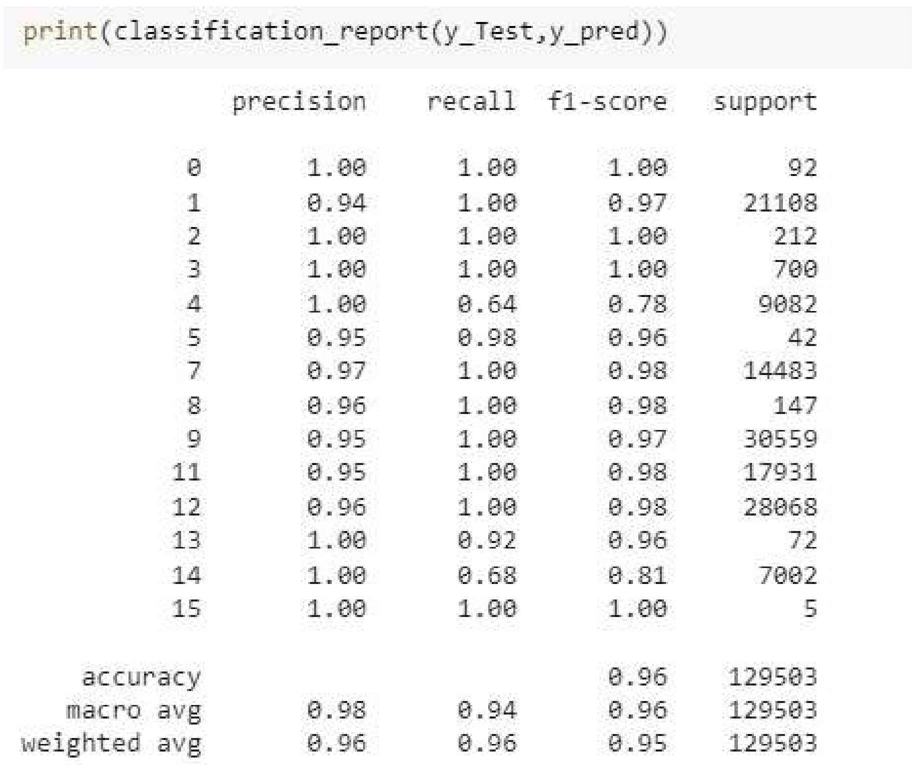
Now we add one more column in our original dataset i.e. cluster which contains the labeled cluster for sample. Now, our unlabeled dataset gets converted into labeled dataset. For finding accuracy of our model we have to split our dataset into train and test. Here, the train part to train our model and test is used for testing and predicting our model. We are applying a Linear discriminant analysis model. After fitting this model in our dataset the accuracy of our method comes out to be 95.76% as shown in the below snippet.



```
# apply Linear Discriminant Analysis
lda = LinearDiscriminantAnalysis(n_components=2)
lda.fit_transform(x_Train, y_Train)
lda.score(x_Train,y_Train)
```

```
0.9576242065759603
```

Evaluation metrics for this method are as given below-



```
print(classification_report(y_Test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        92
           1       0.94      1.00      0.97     21108
           2       1.00      1.00      1.00       212
           3       1.00      1.00      1.00       700
           4       1.00      0.64      0.78      9082
           5       0.95      0.98      0.96        42
           7       0.97      1.00      0.98     14483
           8       0.96      1.00      0.98       147
           9       0.95      1.00      0.97     30559
          11       0.95      1.00      0.98     17931
          12       0.96      1.00      0.98     28068
          13       1.00      0.92      0.96        72
          14       1.00      0.68      0.81      7002
          15       1.00      1.00      1.00         5

    accuracy                           0.96    129503
   macro avg       0.98      0.94      0.96    129503
weighted avg       0.96      0.96      0.95    129503
```

CONCLUSION

From the above three methods we found that Principal component analysis algorithm is best suited for feature selection or dimensionality reduction technique on the given dataset, PCA's accuracy is 99.94% on the other hand the accuracy score for LDA algorithm and correlation method are 95.76% and 99.24% respectively.

REFERENCES

1) https://github.com/krishnadulal/Feature-Selection-in-Machine-Learning-using-Python-All-Code/tree/master/Filtering%20Method
2) https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0
3) https://medium.com/@srishtisawla/linear-discriminant-analysis-d38decf48105