DSCI-552 Section 32415D: Spring 2021

# Problem Set 3 : KMeans

Ananya Sharma

Department of Computer Science,

University of Southern California, Los Angeles, California 90089, USA

(Dated: March 4, 2021)

## Abstract

This report focuses on finding how similar certain groups of people are given their genetic fingerprints due to a variant of a virus. People are segregated into groups depending upon their similarities. A patient Z has been known to be immune to the virus. So, taking his example as a base case, further patients with similar genetics are found and reported.

## 1. INTRODUCTION

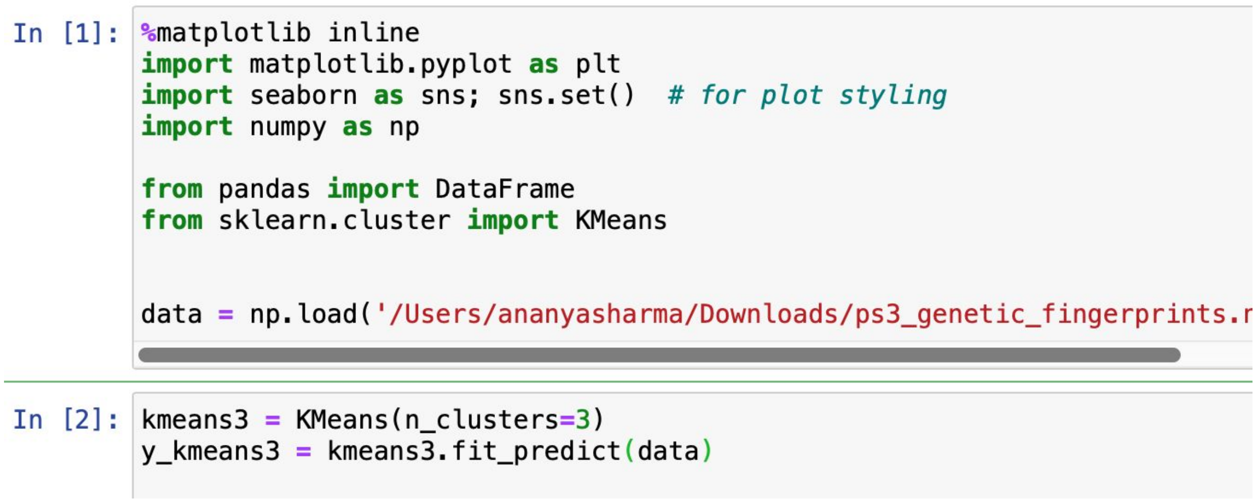
The data consists of a numpy array that has all the information about the patients. There is a separate file for patient z. First, in order to group together people we need to know how many groups can they be segregated into? Since each person has 512 components told about them, such a task can be extremely tedious. To do the grouping, the k-means algorithm is used.

*Number of clusters :*

The first step in this process is finding the number of clusters that will be formed.

The number of clusters that are optimum will be calculated by the model's inertia / sum of squared error. The lower it is, the better. Moreover, the number of clusters would be those after which the inertia of a model linearly decreases.

Considering how the basic divide would be into 2 groups as a baseline but then considering that the features of each person are 512, I decided to start with 3 clusters.



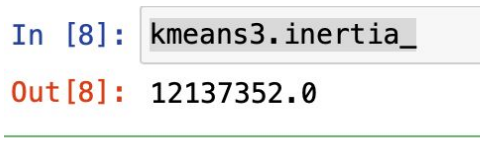
```
In [1]: %matplotlib inline
        import matplotlib.pyplot as plt
        import seaborn as sns; sns.set()  # for plot styling
        import numpy as np

        from pandas import DataFrame
        from sklearn.cluster import KMeans


        data = np.load('/Users/ananyasharma/Downloads/ps3_genetic_fingerprints.r
```

```
In [2]: kmeans3 = KMeans(n_clusters=3)
        y_kmeans3 = kmeans3.fit_predict(data)
```

FIG 1 : Loading libraries, data and running k means with k=3

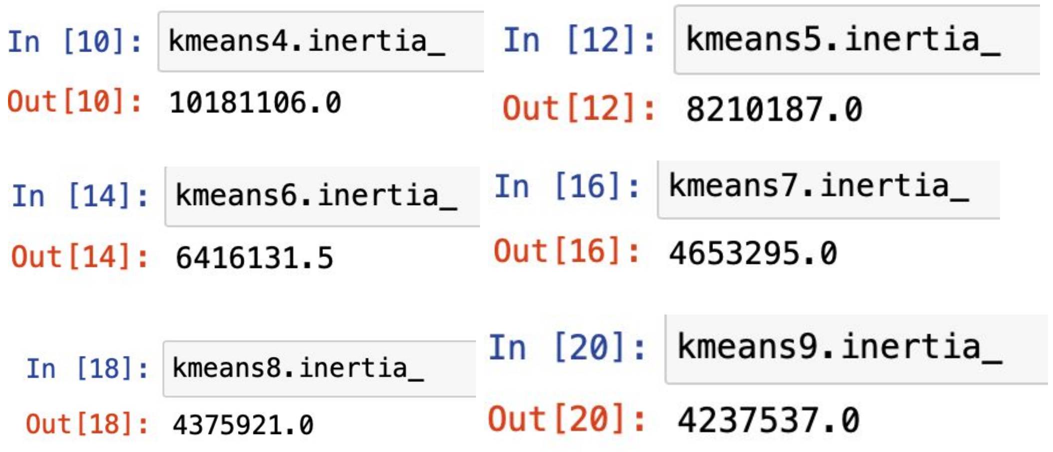


```
In [8]: kmeans3.inertia_
Out[8]: 12137352.0
```

FIG 2 : Finding its inertia

The same process is repeated for more times and its inertia is noted for each model with incremental clusters.

The following inertia values for models with clusters 4,5,6,7,8 and 9 are found:



```
In [10]: kmeans4.inertia_
Out[10]: 10181106.0
```

```
In [12]: kmeans5.inertia_
Out[12]: 8210187.0
```

```
In [14]: kmeans6.inertia_
Out[14]: 6416131.5
```

```
In [16]: kmeans7.inertia_
Out[16]: 4653295.0
```

```
In [18]: kmeans8.inertia_
Out[18]: 4375921.0
```

```
In [20]: kmeans9.inertia_
Out[20]: 4237537.0
```

FIG 3 : Inertia of models with clusters 4-9

Here we observe that uptil cluster 7, the inertia was dropping at a significant rate. However, post 7 clusters, the drop has been steep, almost negligible in comparison with the previous reductions.

Creating a graph of the same, to better understand and visualize:

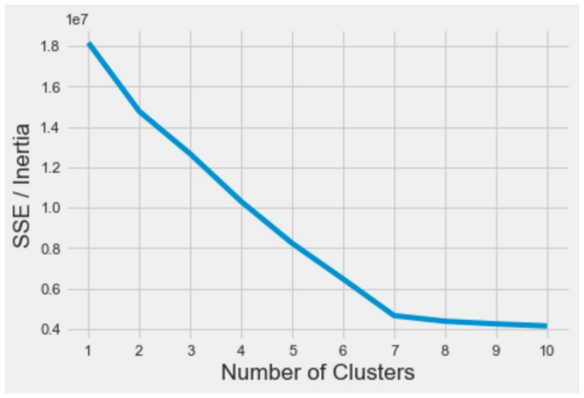


FIG 4 : Graph plotted number of clusters vs their Inertia with that number of clusters

From the graph it's visible that there's a steady decline in the inertia till the number of clusters is 7. Beyond that the decline is linear and negligible in comparison to previous reductions.

So, the optimal number of clusters is 7.

When the model has 7 clusters, its inertia is at a low beyond which inertia reduction is linear. So, even if the model has 8 clusters, it might be very possible that data points previously together till clusters 7 will now be unnecessarily separated.

## *PCA Algorithm for Visualization:*

Now that we know that the number of clusters needed is 7, we can now visualize the data.

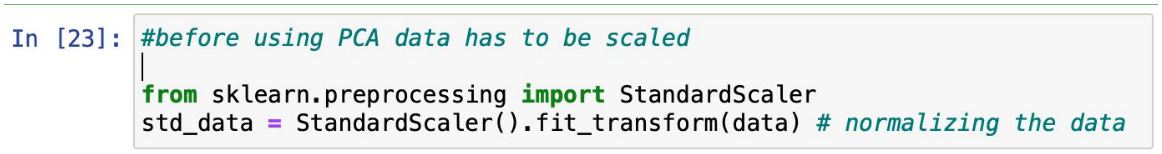
However, our data has 512 components. This is high dimensional data and needs to be reduced before visualization can be performed on it.

Principal Component Analysis (PCA) is used when data is of high dimension (has a large number of components/ features) and one wants to reduce the features. These components are basically new variables, derived from the original ones, and they are usually displayed in order of importance. At the end of the PCA analysis, we aim to choose only a few components, while preserving as much of the original information as possible.

PCA tries to find the best possible subspace which explains most of the variance in the data.

Data sets usually contain numerical features that have been measured in different units, such as height (in inches) and weight (in pounds). A machine learning algorithm would consider weight more important than height only because the values for weight are larger and have higher variability from person to person.

This is the reason why scaling of data is important, so that all features are on the same ground and do not vary greatly cumulatively.



```
In [23]: #before using PCA data has to be scaled

         from sklearn.preprocessing import StandardScaler
         std_data = StandardScaler().fit_transform(data) # normalizing the data
```

FIG 5 : This is how the data is scaled.

Using the StandardScaler class the data is scaled. Standardization scales /shifts, the values for each numerical feature, so that the features have a mean of 0 and standard deviation of 1.

Post scaling of data, now lets see how much each feature contributes to the variance in the dataset. This is extremely important as it shows us which features will then be selected for Principal Components.

Let's see a broad overview of how the features contribute to the model variance.

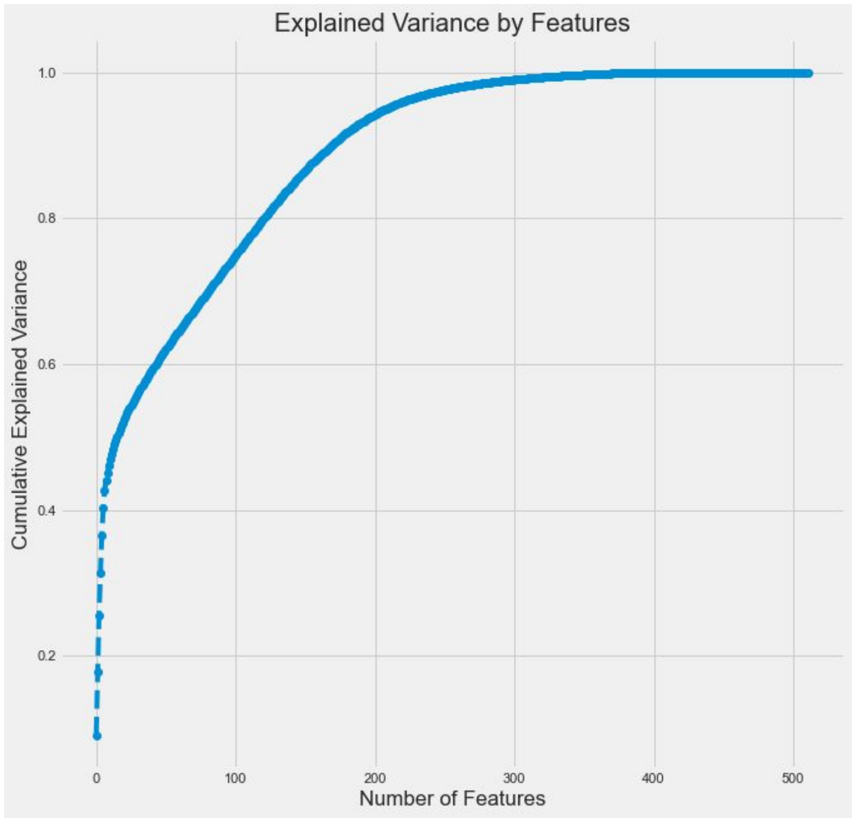


FIG 5 : Features vs their variance

Looking at the graph above it shows that atleast 200 out of 512 features contribute to the variance experienced by the model. Moreover, the dotted lines are present towards the y-axis.

Since, taking 200 components as Principal components wouldn't be feasible, i started with 20 to see what their variances are.

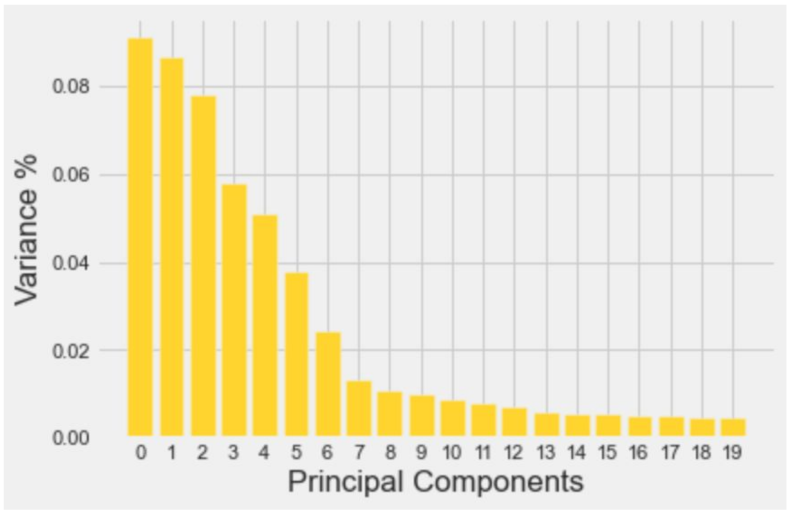


FIG 6 : The principal components and their corresponding variance

From the graph above we can see that the first three components contribute most to the variance of the model. Although components 4-6 do also contribute, their contributions in comparison are less. Further, components 7 onwards show an almost linear contribution to variance, which can be ignored while choosing Principal Components.

Now that we know that we have 3 principal components, the dimensionality of the dataset can be reduced and clusters can be visualized.

For this, I've taken the number of PCs to be 2 and visualized the clusters.

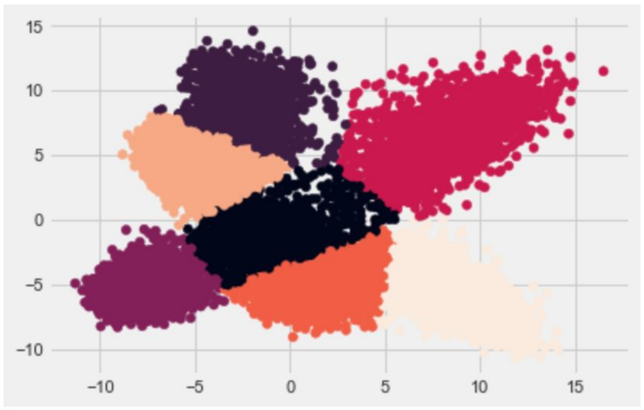
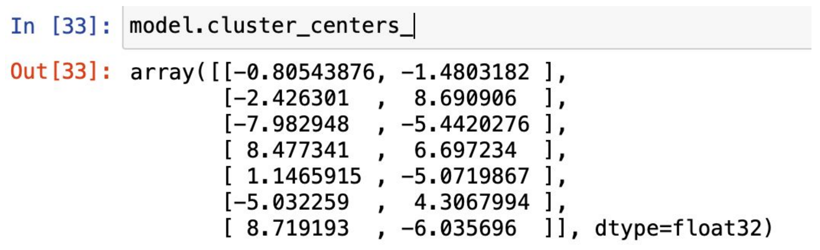


FIG 7 : The 7 clusters with different colors to differentiate between the same.

Cluster centres for the 7 clusters are :

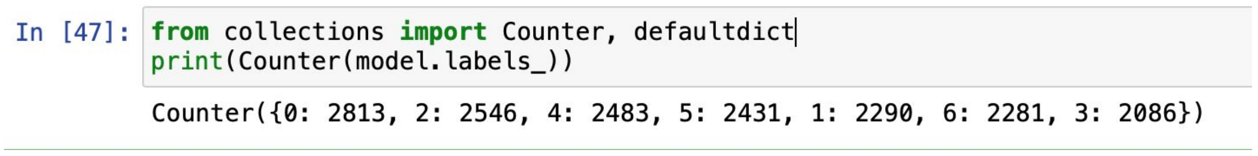


```
In [33]: model.cluster_centers_

Out[33]: array([[-0.80543876, -1.4803182 ],
               [-2.426301  ,  8.690906  ],
               [-7.982948  , -5.4420276 ],
               [ 8.477341  ,  6.697234  ],
               [ 1.1465915 , -5.0719867 ],
               [-5.032259  ,  4.3067994 ],
               [ 8.719193  , -6.035696  ]], dtype=float32)
```

FIG 8 : The 7 cluster centers

Now, lets see how many data points each cluster has.



```
In [47]: from collections import Counter, defaultdict
         print(Counter(model.labels_))

         Counter({0: 2813, 2: 2546, 4: 2483, 5: 2431, 1: 2290, 6: 2281, 3: 2086})
```

FIG 9 : Data Points in each cluster

We can see that each cluster has a label numbered from 0-6 for the 7 clusters. Each label holds similar types of data points and their whole individual aggregate is given in Fig 9.

Now, the patient Z data is combined with the general patient data (appended at the end) and the model is again trained with a cluster of 7 and visualised to see in which cluster does patient Z belong.

We observe that patient Z belongs to cluster 5.

## 2. CONCLUSIONS

The optimal number of clusters to have are 7. So the patients can be divided up into 7 groups in such a way that each person in the group has most similarities genetically with other people of the same group.

The Principal Components that are most important are 3. Components 4-7 can be included if wanted but for simplistic visualisation, even ⅔ are ample.

Patient Z belongs to cluster 5 and it is these groups of patients that should be started first on inoculations and trials. Conclude the report. Remember, your CEO will likely read only abstract and the con- clusions.

DATA AVAILABILITY

Data is available at
https://www.kaggle.com/c/usc-dsci552-section-32415d-spring-2021-ps3/overview

CODE AVAILABILITY

Code is available at
https://github.com/usc-dsci552-32415D-spring2021/problem-set-03-AnanyaSharma25/tree/main

[1]https://vitalflux.com/k-means-elbow-point-method-sse-inertia-plot-python/
[2]https://heartbeat.fritz.ai/k-means-clustering-using-sklearn-and-python-4a054d67b187
[3]https://realpython.com/k-means-clustering-python/
[4]https://365datascience.com/tutorials/python-tutorials/pca-k-means/