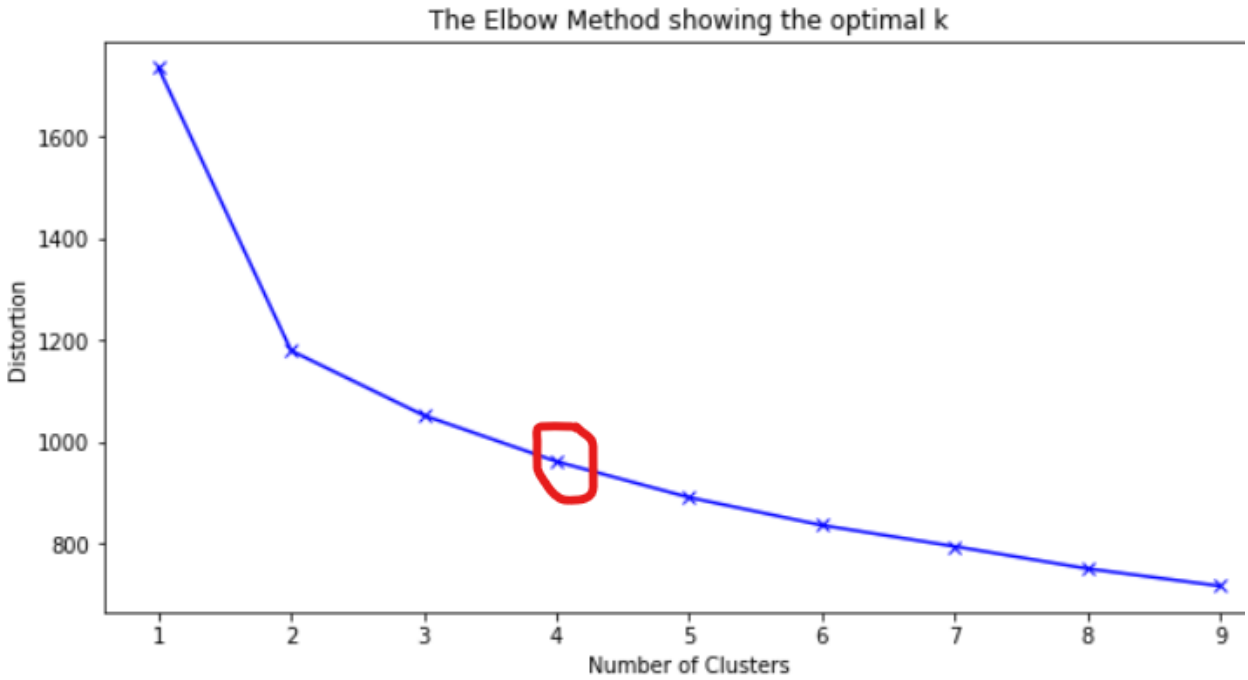
The curse of dimensionality refers to an extensive increase in data dimension and computation resources. The unlabeled data provided in TripAdvisor requires feature reduction to decrease the ETL (Extract/Transform/Load) time and compute resources with minimizing error loss. There are 10 features in the dataset, using all 10 features will result in poor clustering, it needs to reduce in certain trip categories.

Travels can be divided into different clusters by using unsupervised machine learning algorithms, which group the data into K clusters. The optimal number of K or centroid can be achieved via the Elbow method. To find the optimal clusters I am using the KMeans algorithm, in the figure below the Centroid value is 4.

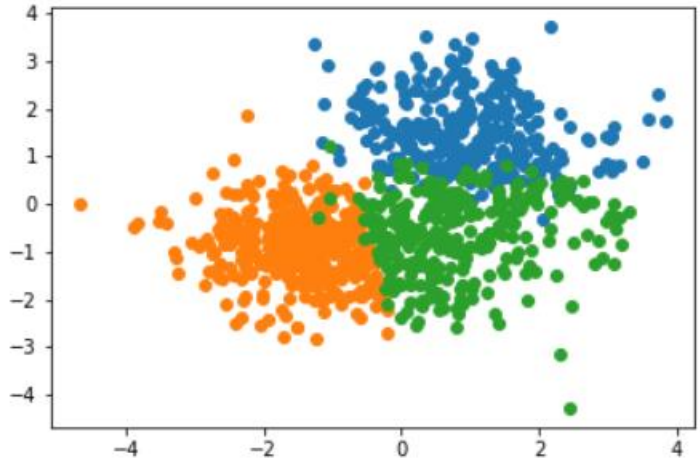


The first solution I have implemented uses KMeans unsupervised machine learning (ML) algorithm.

K-means is an unsupervised ML algorithm approach for vector quantization, which is used when we are dealing with unlabeled data. It groups the unlabeled set of input into several K clusters, the cluster centroids are then used to produce feature

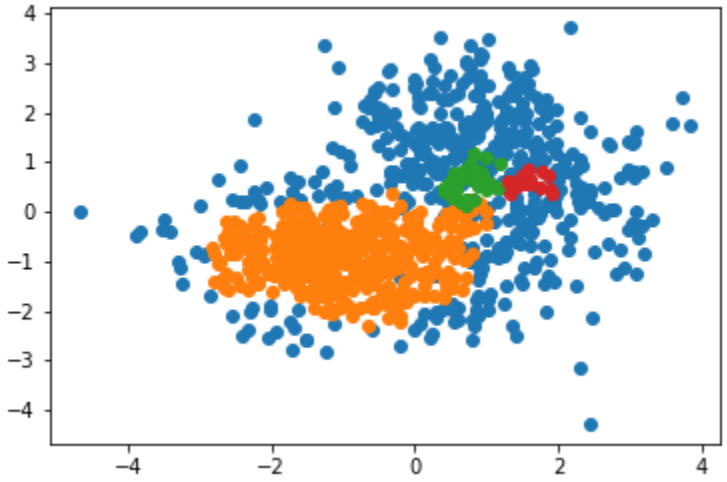
The algorithm aims to group an unlabeled set of inputs into K clusters and then use the centroids of these clusters to produce features.

In this solution I am using an elbow value of three or K value of three, since the solution requires feature reduction, I am using only three features with the sample data sets. Features above three will result in poor clustering. The three centroids are used in this clustering. For implementation details, please refer to the code in the ipynb file.


4
3
2
1
0
−1
−2
−3
−4
−4
−2
0
2
4

The second solution is DBScan (Density-Based Spatial Clustering of Application with Noise), which refers to unsupervised learning methods. The discovery of the arbitrary shapes and good efficiency is the critical requirements for large spatial database.

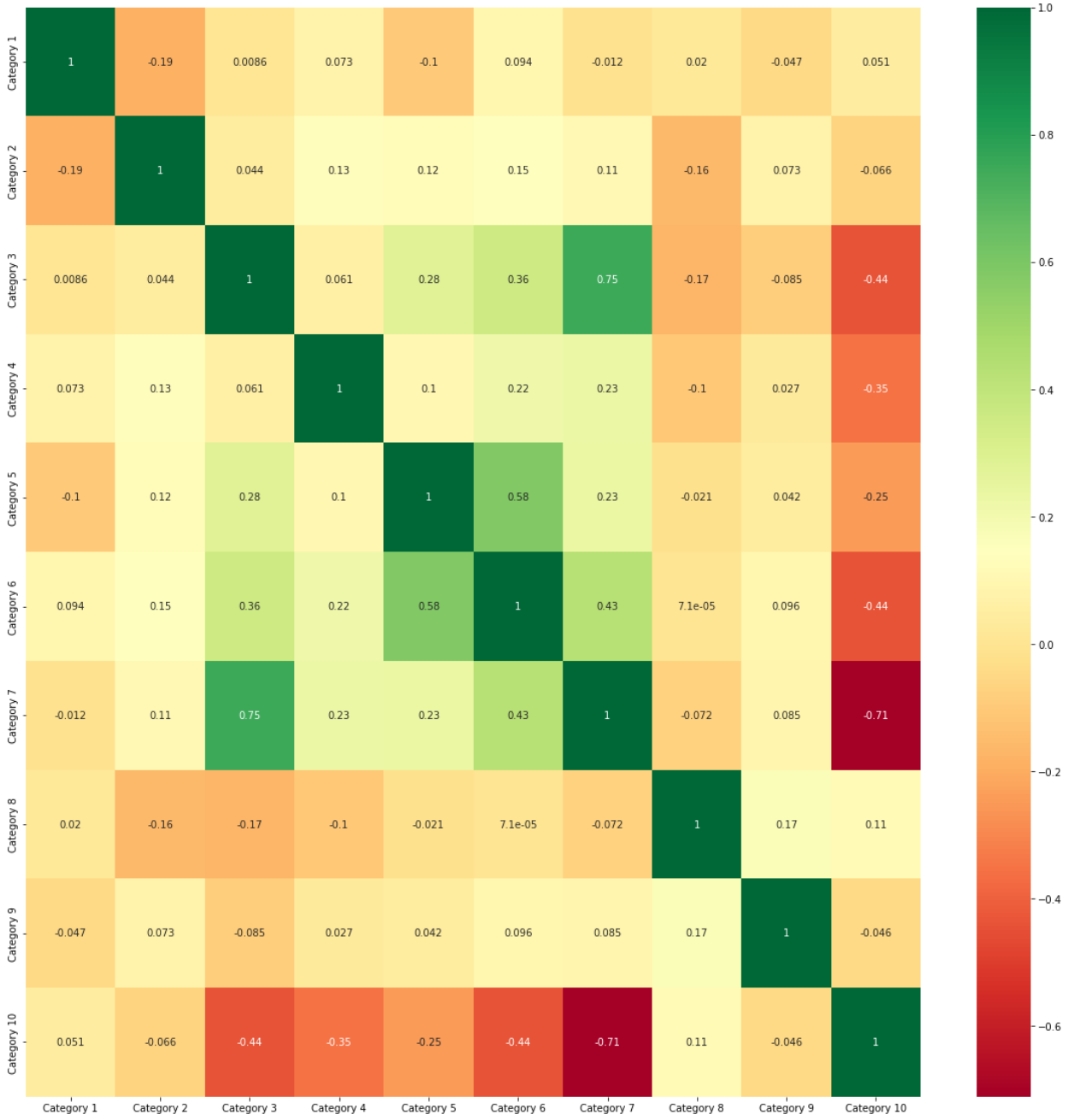
The following solution uses two input parameters epsilon and a minimum number of points in the cluster. The Epsilon is a distance measure to locate the points in the neighborhood of any point. For more information on the algorithm please refer to the ipynb file.



The K-Means algorithm clusters the loosely related data together. Every single element in the cluster plays a significant role since the cluster is the mean of the all the elements. Any change in the data point might have significant affect the clustering outcome. The main disadvantage of the K-Means is prior knowledge of the K value the number of clusters required, in most of the cases reasonable K value is unknown.

In contrast DBScan algorithm doesn't require prior knowledge of the number of clusters. It uses the function to calculate the distance between values and the guiding parameter which determines distance measurement to be considered close. It uses two parameters, minPts and eps.

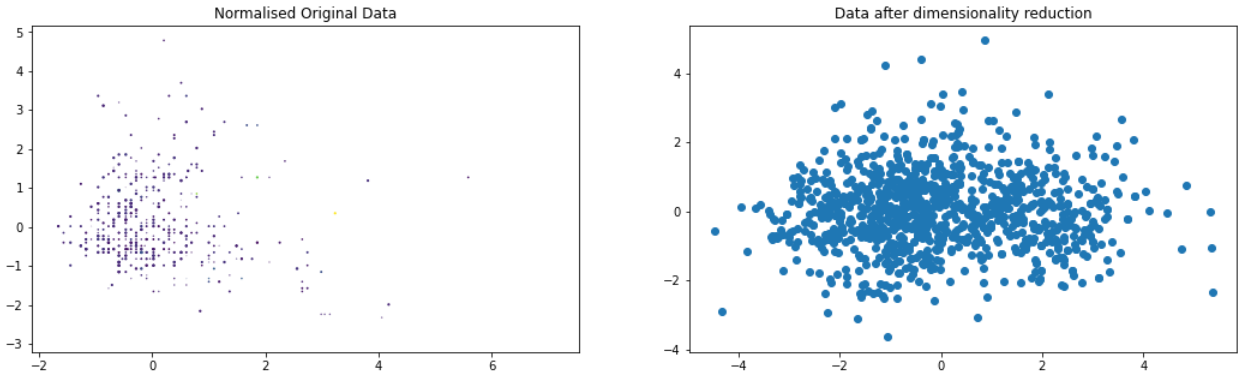
- minPts: minimum data points required to form a dense cluster.
- eps: A distance measure that will be used to locate the points in the neighborhood of any point.



The correlation heat map represents the data in two-dimension, it quantifies the correlation between variables with different color scheme. The color intensity measures the correlation of the data, which is the linear relationship between two variables.

The squares in heat map represents the correlation between the variables on each axis, and non-zero range from -1 to +1. The value of +1 represents that the correlation is stronger as they both increase by +1. The correlation of -1 is similar, but instead one of the variables has stronger correlation. The correlation of ZERO means there is no linear trend between the two variables. The darker green (+1) means they are correlating to itself.

Before we do dimension reduction, we need to normalize the data. The dataset contains features that highly vary in magnitudes, reduction of the features is obtained via PCA.



After the transformation of feature columns into two features and reverting back to the original data the reconstruction error is 0.76.

```
rec_error = np.linalg.norm(Xnorm-pca_orgfeat, 'fro')/np.linalg.norm(Xnorm, 'fro')

print(f'The reconstruction error is: {rec_error}')
```

The reconstruction error is: 0.7589269810329784

Reducing the number of features in high dimensional data is called dimensionality reduction.

Principal component analysis (PCA) is used for feature extraction, the technique entails building a new variable from the original dataset. The new variable is non-redundant and minimizes information loss.

Variance measures the spread of data in a given independent variable around its mean and is given by the formula

$$\mathrm{var}(x) = \frac{\sum_{i=1}^{n}(x_i - \bar{x}^2)}{n-1}$$

The eigenvalues of the corresponding eigenvector can tell us how much variance is associated with its vector. To calculate the percentage of the variance for each eigenvector, first, get the relative number by first summing up the eigenvalues then divide the eigenvalue by this sum.

In the case of PCA, “variance means summative variance” consider the Eigen covariance matrix

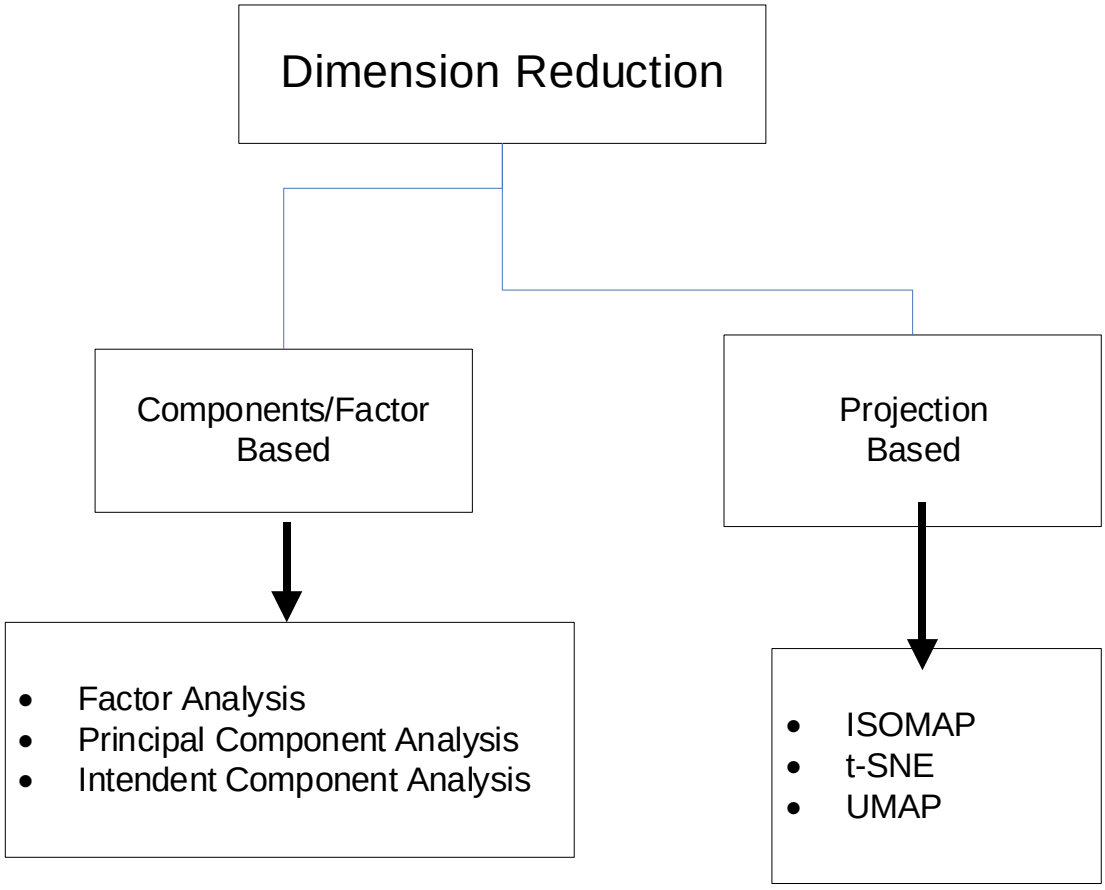
```
1.343730519  -.160152268   .186470243
 -.160152268   .619205620  -.126684273
  .186470243  -.126684273  1.485549631
```

Their variance is on the diagonal and the sum of the 3 values is 3.448. PCA replaces the original variable with a new orthogonal variance called eigenvalues, the values are placed in decreasing order. The new covariance matrix is

1.651354285 .000000000 .000000000
.000000000 1.220288343 .000000000
.000000000 .000000000 .576843142

The diagonal sum is still the same 3.448, the first principal component 1.651/3.448 = 47.9%, the second 1.220/3.448 = 35.4%, and finally the third one 577/3.448 = 16.7%. The largest variance out of all the variance is located first in the above example the largest variant is 1.651354285, followed by the dimension of the second variance and finally the smallest variance.

Dimension Reduction can be divided into two Components/Factor base or Projection based



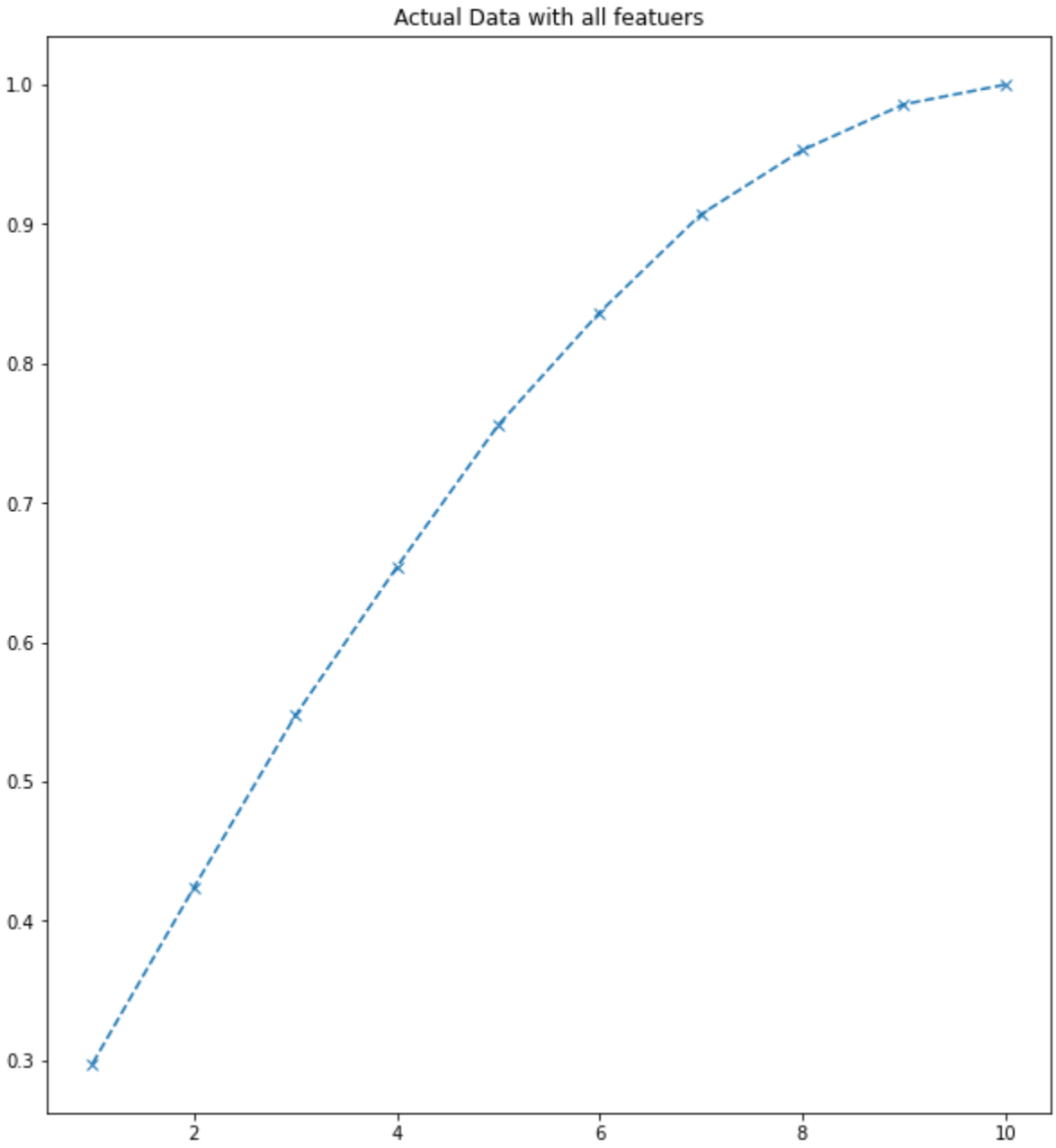
Component Based

Factor Analysis: The technique is best suited for highly correlated set of variables; the variables are divided based on their correlation into different groups.

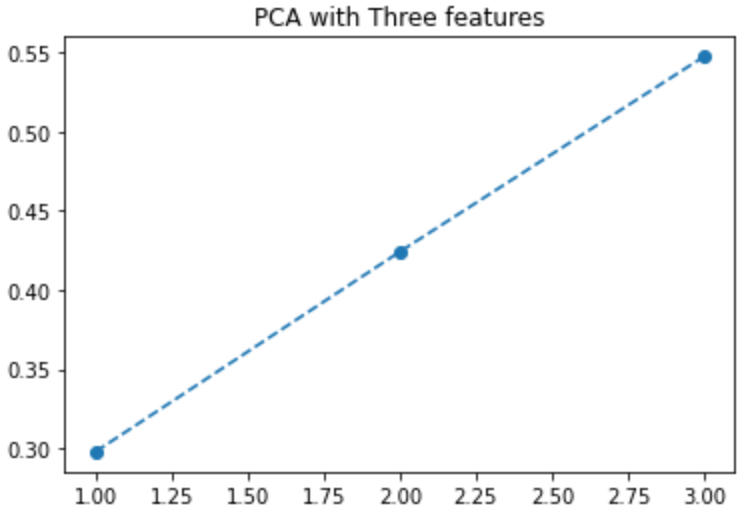
Principal Component Analysis: The technique is used for dealing with linear data, it divides the data in components with variance.

Independent Component Analysis: it transforms the data into individual component and requires a smaller number of components.

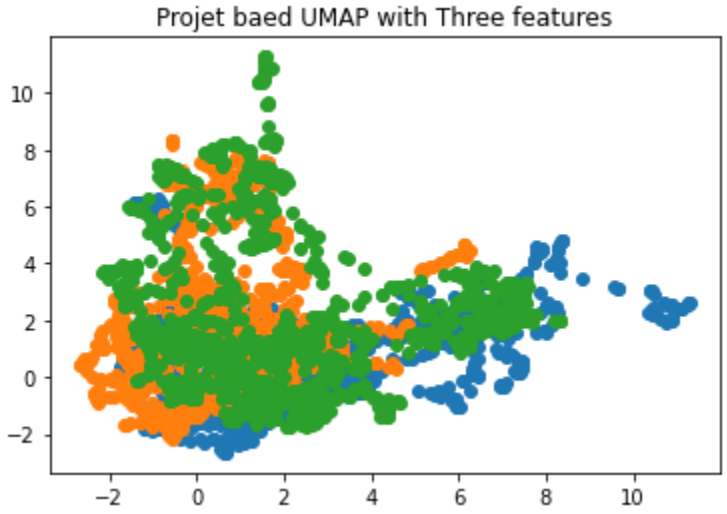
The first graph is actual data with all features



The second graph uses component/factor-based analysis PCA



Finally, the third graph uses project based UMAP analysis.



The PCA and projection based UMAP reduces the complexity of the compute required and also the storage requirement. The new variable is a non-redundant variable providing more compute efficiency with sparse matrix.

Different types of clustering algorithms

Cluster or pattern analysis is the unsurprised learning task, clustering algorithms are mainly used to discover the dense region of observation. They mainly use similarity or distance, and some requires predetermine number of clusters. Based on the class notes and the reference below here are the list of clustering algorithms

Affinity Propagation

Agglomerative Clustering

BIRCH

DBSCAN

K-Means

Mini-Batch K-Means

Mean Shift

OPTICS

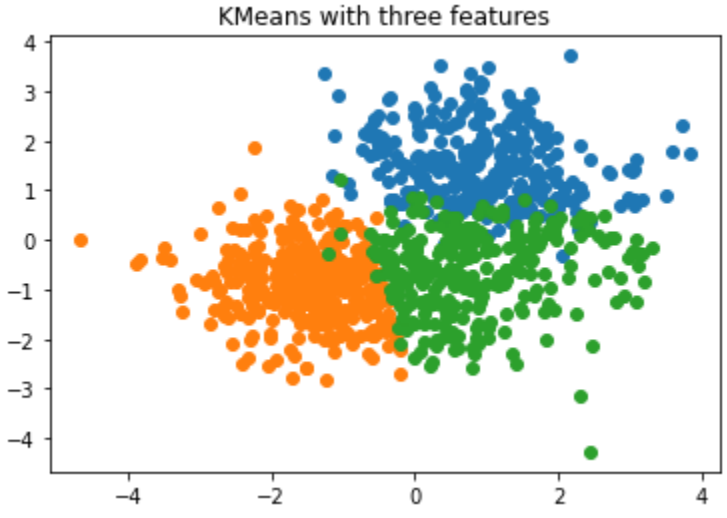
Spectral Clustering

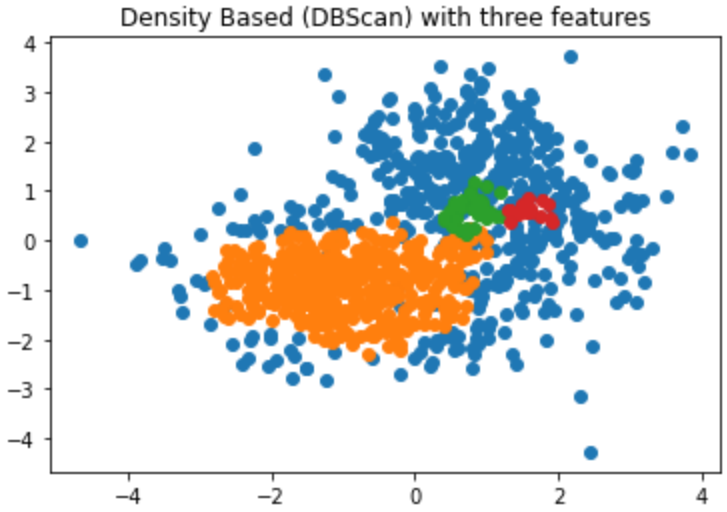
Mixture of Gaussians

Grid based clustering

Model based clustering

I have used the assignment dataset to test the KMeans (Ensemble) and Density based (DBScan).





References:

Overall PCA operation: https://towardsdatascience.com/principal-component-analysis-ac90b73f68f5#:~:text=Unlike%20variance%2C%20which%20measures%20spread,relationship%20between%20the%20two%20dimensions.

PCA Variance explains: https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579

Different Clustering algorithms: http://www.cs.utsa.edu/~bylander/cs6243/kotsiantis-clustering.pdf