Clustering Project: Human Activity Recognition using smartphones
Pattadon Naksuwan

Main Objective: To segment the data into group without data labels and to improve classification algorithm's performance with clustering technique

Description: The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING*UPSTAIRS, WALKING*DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% the test data. The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain. There are 562 columns and 10299 rows. There is only object data type which is the activity column, and others are all float.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10299 entries, 0 to 10298
Columns: 562 entries, tBodyAcc-mean()-X to Activity
dtypes: float64(561), object(1)
memory usage: 44.2+ MB
```

Data cleaning and features engineering:

1. The activity column will be removed, since we want to predict it using classification models and also we want to segment the data without the labels. Therefore, we will have 561 features, which is quite many.
2. The data is not normalized yet, so it needs standard scaling to be normalized.
3. Principal component analysis (PCA) can be applied to help reduce the number of features to avoid the curse of dimensionality and overfitting. The goal of PCA is to reduce the dimension of the data and still keep the variance. After looping through the numbers, the model PCA(n_components=100) is chosen because it has the explained variance of 94.63% and the number of dimensions is reduced from 561 to only 100.

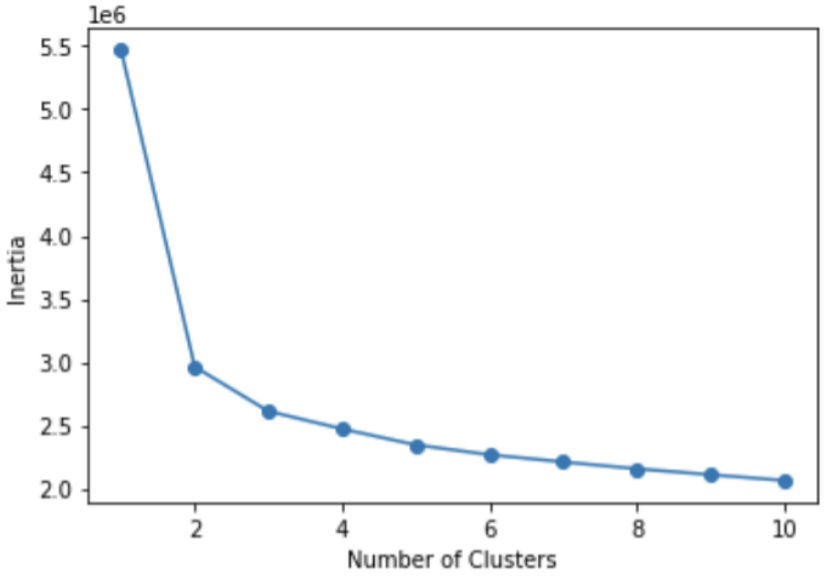
n	model	var
25	PCA(n_components=25)	0.792441
50	PCA(n_components=50)	0.870869
100	PCA(n_components=100)	0.946348
200	PCA(n_components=200)	0.993222
300	PCA(n_components=300)	0.999027
400	PCA(n_components=400)	0.999971
500	PCA(n_components=500)	1.0

```
data.shape
```

```
(10299, 100)
```

Model: There are 2 clustering models used in this project, which are KMeans and AgglomerativeClustering.

1. KMeans: We can use the elbow method to choose the number of clusters for the KMeans algorithm. We can see that number of clusters = 2 is clearly the elbow point.



Since KMeans is the unsupervised learning algorithm, it doesn't have metrics to measure how well the model can do. Luckily, the dataset provides the activity classes for us, then we can compare how well the model did with the real labels.

		number
Activity	**km1**	
LAYING	**0**	12
	1	1932
SITTING	**0**	3
	1	1774
STANDING	**1**	1906
WALKING	**0**	1722
WALKING_DOWNSTAIRS	**0**	1406
WALKING_UPSTAIRS	**0**	1536
	1	8

2. Agglomerative Clustering: From KMeans, we have our number of clusters to be 2, so we will use that in agglomerative clustering too.

		number
Activity	**ag1**	
LAYING	**1**	1944
SITTING	**1**	1777
STANDING	**1**	1906
WALKING	**0**	1722
WALKING_DOWNSTAIRS	**0**	1406
WALKING_UPSTAIRS	**0**	1544

We can see that agglomerative clustering did perfectly to cluster the data into 2 clusters. And you can easily see that cluster1 is not moving and cluster is moving.

Now we will use agglomerative clustering to help improve the performance of the classification algorithm, namely logistic regression.

However, in this dataset, both using KMeans and not using KMeans get the same in all scores.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	194
1	0.95	0.97	0.96	178
2	0.97	0.95	0.96	191
3	1.00	1.00	1.00	172
4	1.00	1.00	1.00	141
5	1.00	1.00	1.00	154
accuracy			0.99	1030
macro avg	0.99	0.99	0.99	1030
weighted avg	0.99	0.99	0.99	1030