# Programming Assignment #3 Clustering

## Student Details

When submitting, fill your name and ID in this cell. Note that this is a markdown cell.

Student Name and ID:

## Rules

1. Work is to be done individually
2. Any cheating including plagiarism, collusion will be reported to the corresponding UTA' s instance
3. If using any resource (books, internet), please make sure that you cite it.

## Assignment Details

The purpose of this assignment is to cluster drug users using K-means clustering and Hierarchical Agglomerative clustering models and to visualize clusters for predicted and actual cluster labels.

You will be using the same dataset that you used in assignment 2. However, this time we provided the class labels and the dataset is given in a csv file.

You need to submit this ipython file after renaming it with your NetId.

## Required Python Packages

In [1]:

```
# Import required Python packages here
```

## TASK 1: K-Means Clustering

### Task 1-a: Determine "k" value from the elbow method

In this task, you will be using the elbow method to determine the optimal number of clusters for k-means clustering.

We need some way to determine whether we are using the right number of clusters when using k-means clustering. One method to validate the number of clusters is the elbow method.

The idea of the elbow method is to run k-means clustering on the dataset for a range of values of k (k will be from 1 to 10 in this task), and for each value of k calculate the sum of squared errors (SSE). Then, plot a line chart of the SSE for each value of k. If the line chart looks like an arm, then the "elbow" on the arm is the value of k that is the best. The idea is that we want a small SSE, but that the SSE tends to decrease toward 0 as we increase k (the SSE is 0 when k is equal to the number of data points in the dataset, because then each data point is its own cluster, and there is no error between it and the center of its cluster). So our goal is to choose a small value of k that still has a low SSE, and the elbow usually represents where we start to have diminishing returns by increasing k.

For this task, you need to perform the elbow method for k from 1 to 10, and plot a line chart of the SSE for each value of k, and determine the best k (the number of clusters). Note that you need to use the whole dataset in this task.

In [1]:

```
#########################begin code for Task 1-a



#########################end code for Task 1-a
```

## Task 1-b: Visualization for K-Means Clustering

In this task, you will be performing k-means clustering for k=2 and visualize the predicted training samples and actual training samples on scatter plots. Use 75% of the dataset for training and 25% of the dataset for testing. Perform kmeans for clustering samples in your training set.

Use two subplots for visualizing the predicted training samples and actual training samples on two scatter plots.

Since your dataset has multiple features(dimensions), you won't be able to plot your data on a scatter plot. Thus, you’re going to visualize your data with the help of one of the Dimensionality Reduction techniques, namely Principal Component Analysis (PCA). The idea in PCA is to find a linear combination of the two variables that contains most of the information. This new variable or “principal component” can replace the two original variables. You can easily apply PCA to your data with the help of scikit-learn.

In [2]:

```
###################begin code for Task 1-b-1: Split the dataset 75% for training and 25% for testing

### Important!!!
### Created training and testing datasets will be used for the rest of this assignment except Task 3.



###################end code for Task 1-b-1
```

In [ ]:

```
###################begin code for Task 1-b-2: Visualize the predicted training labels vs actual training labels

# Import PCA
from sklearn.decomposition import PCA



# Create the KMeans model

# Compute cluster centers and predict cluster index for each sample

# Model and fit the data to the PCA model
X_train_pca = None

# Visualize the predicted training labels vs actual training labels.
### scatter(x, y, your_data)
x = X_train_pca[:, 0]
y = X_train_pca[:, 1]


###################end code for Task 1-b-2
```

Now, you need to visualize the predicted testing labels versus actual testing labels. Use the trained model in previous step.

```
In [4]:
```

```
###################begin code for Task 1-b-3: Visualize the predicted testing la
bels vs actual testing labels



# predict cluster index for each sample

# Model and fit the data to the PCA model
X_test_pca = None

# Visualize the predicted testing labels vs actual testing labels.
### scatter(x, y, your_data)
x = X_test_pca[:, 0]
y = X_test_pca[:, 1]



###################end code for Task 1-b-3
```

In this step, you need to provide the evaluation of your clustering model. Print out a confusion matrix.

```
In [ ]:
```

```
###################begin code for Task 1-b-4: Print out a confusion matrix



###################end code for Task 1-b-4
```

# TASK 2: Hierarchical Agglomerative Clustering

## Task 2-a: Find the best Hierarchical Agglomerative Clustering Model

In this task, you will be performing Hierarchical Agglomerative clustering with different linkage methods (complete and average) and different similarity measures (cosine, euclidean, and manhattan) in order to find the best pair of linkage method and similarity measure. Use F1 score for evaluation and take n_clusters = 2.

In [ ]:

```
###################begin code for Task 2-a: Print out a confusion matrix
# Import AgglomerativeClustering
from sklearn.cluster import AgglomerativeClustering
# Import pairwise_distances for calculating pairwise distance matrix
from sklearn.metrics.pairwise import pairwise_distances
# Import f1_score
from sklearn.metrics import f1_score


## Calculate pairwise distance matrix for X_train
pdm_train = None

## Model and fit the training data to the AgglomerativeClustering model
## complete linkage + cosine


## Model and fit the training data to the AgglomerativeClustering model
## complete linkage + euclidean


## Model and fit the training data to the AgglomerativeClustering model
## complete linkage + manhattan


## Model and fit the training data to the AgglomerativeClustering model
## average linkage + cosine


## Model and fit the training data to the AgglomerativeClustering model
## average linkage + euclidean


## Model and fit the training data to the AgglomerativeClustering model
## average linkage + manhattan


print("F1-score for complete linkage + cosine", None)
print("F1-score for complete linkage + euclidean", None)
print("F1-score for complete linkage + manhattan", None)
print("F1-score for average linkage + cosine", None)
print("F1-score for average linkage + euclidean", None)
print("F1-score for average linkage + manhattan", None)

###################end code for Task 2-a
```

## Task 2-b: Visualization for Hierarchical Agglomerative Clustering

Find the best performed model from the previous step and use that model for visualizing the predicted training samples and actual training samples on scatter plots. Use PCA model for visualizing your data (use X_train_pca from Task 1-b-2).

```
In [ ]:
```

```
###################begin code for Task 2-b: Visualize the predicted training labels vs actual training labels


# Visualize the predicted training labels versus actual training labels.


###################end code for Task 2-b
```

# TASK 3: Compare K-Means Clustering and Hierarchical Agglomerative Clustering

## Task 3-a: Visualize Clusters

In this task, use whole dataset for training k-means cluster and hierarchical agglomerative clustering. Use the best model for agglomerative clustering. Visualize the predicted labels from k-means clustering and agglomerative clustering versus actual labels. Basically, you need to plot three scatter plots as subplots.

In [ ]:

```
###################begin code for Task 3-a: Visualize the predicted training labels vs actual training labels

### Kmeans Clustering
# Model and fit the data to the Kmeans (use fit_predict : Performs clustering on X and returns cluster labels.)


### Agglomerative Clustering
# Calculate pairwise distance matrix for X

# Model and fit the data to the Agglomerative (use fit_predict : Performs clustering on X and returns cluster labels.)


### Visualize Clusters
#  Model and fit the data to the PCA model
X_pca = None

# Visualize the predicted Kmeans labels versus  the predicted Agglomerative labels versus Actual labels.


###################end code for Task 3-a
```

## Task 3-b: Compare K-Means Clustering & Hierarchical Agglomerative Clustering¶

Print out confusion matrices for kmeans and agglomerative clustering. Also, compare precision, recall, and F1-score for both model.

In [ ]:

```
###################begin code for Task 3-b



###################end code for Task 3-b
```

## Grading

Task 1: 40 points

--> Task 1-a: 10 points

--> Task 1-b: 30 points

```
    Task 1-b-1: 5 points

    Task 1-b-2: 10 points

    Task 1-b-3: 10 points

    Task 1-b-4: 5 points
```

Task 2: 40 points

--> Task 2-a: 30 points

--> Task 2-b: 10 points

Task 3: 20 points

--> Task 3-a: 10 points

--> Task 3-b: 10 points