B365 HW6 (Fall 2024)

# Submission instructions

1. Dataset: For this assignment, use the provided dataset B365-HW6.csv (available on both the GitHub assignment repository and Canvas). Although it is a labeled dataset, you will treat it as unlabeled for the purpose of cluster analysis by drop- ping the ‘class’ column. The goal of this assignment is to give you hands-on expe- rience using sklearn for clustering and to familiarize you with the key parameters for various clustering algorithms.
2. Reference Materials: You may use the demo notebooks including [the one for com-](https://www.kaggle.com/code/morecoding/clustering-basic) [parison of the different clustering algorithms using simulated datasets](https://www.kaggle.com/code/morecoding/clustering-basic) and [the one](https://www.kaggle.com/code/morecoding/hierarchical-clustering/edit/run/29343525) [on hierarchical clustering with dendrogram visualizations](https://www.kaggle.com/code/morecoding/hierarchical-clustering/edit/run/29343525) for reference. Check out the lecture slides if you need to review the methods.
3. Submission Requirements: Submit a PDF file with your answers (including the plots) to Canvas, and upload your code/notebook to your B365 github repo under HW6.

# Questions

1. K-means Clustering (10 points)

Objective: Apply the k-means algorithm using sklearn.cluster.KMeans on the dataset. Instructions:

* + Experiment with different values of *k*, ranging from 1 to 20 (inclusive).
  + For each value of *k*, run the algorithm 10 times with different initializations to observe variations in performance.
  + Use init = ”random” and n init = ”auto” for other settings. Tasks:

1. Elbow Plot: Create an elbow plot where the performance for each *k* is repre- sented as a boxplot (instead of a single point) based on the 10 runs. Be sure to

label the axes properly.

1. Choosing *k*: Based on the elbow plot, select the optimal value for *k* and explain your choice.
2. Hierarchical Clustering (5 points)

Objective: Perform hierarchical clustering using sklearn.cluster.AgglomerativeClustering. Instructions: Test two linkage methods: ”average” and ”complete”.

Tasks:

1. Dendrogram Plots: Generate dendrograms for both linkage methods to visu- alize the clustering results.
2. Cluster Selection: Based on the dendrograms, determine the optimal number of clusters and briefly explain your reasoning.
3. DBSCAN Clustering (10 points)

Objective: Apply the DBSCAN algorithm using sklearn.cluster.DBSCAN.

Instructions: Estimate the eps parameter using the *k*th-nearest neighbor distance plot. For this problem, use *k* = 5.

Tasks:

1. k-Nearest Neighbor Plot: Generate a plot of the *k*th nearest neighbor distance for all data points. Assume *X* is the variable that holds the data. You may use the following code snippet to calculate the *k*th nearest neighbor distance for all the data points in *X*:

from sklearn . neighbors import Nearest Neighbors k = 5

neigh = Nearest Neighbors ( n neighbors=k ) nbrs = neigh . f i t ( X)

distances , = nbrs . kneighbors ( X) kt h di s = di s t a n c e s [ : , k − 1 ]

1. Choosing eps: Based on your plot, select an appropriate value for eps and justify your choice.
2. DBSCAN Results: Apply DBSCAN using your chosen eps value and report: the fraction of data points labeled as noise (i.e., assigned to cluster -1), and the total number of clusters inferred by the algorithm.