Homework#4

ACS577 Knowledge Discovery and Data Mining, Summer II 2023

Due: August 2

- For your homework4 submission, prepare a single file, *YourLastName_FirstName*_ACS577_HW4.zip
- Organize your submission like P1, P2, etc. In each directory, include your program codes and outputs (answers) for each sub question (1), (2), …
- Clearly number your answer according to the question number.

- Homework #4 includes only hands-on practice problems for outlier detection.
- The following question problems are based on Ch 7 from Zhao, "R and Data Mining", but modified for this homework. You can use R or any programming language or library for this homework.

P1. **(Univariate outlier detection)** You will use randomly generated numbers for univariate outlier test.

(1) Generate 200 normally distributed random numbers
(2) Show the summary of the numbers using min, Q1, Median, Mean, Q3 and Max
(3) Draw the boxplot of this data
(4) Report the values of any data points which lie beyond the extremes of the whiskers

P2. **(Multivariate outlier detection)** We will generate 2-dimensioanl data for multivariate outlier test. Outliers can be detected separately from each attribute, and then take outliers as those data which are outers for both attributes.

(1) Generate 200 normally distributed random numbers for one attribute, ***x***,
and generate 200 normally distributed random numbers for another attribute, ***y***,
and then combine them for 2-dimensioanl data. For the vilification, display first 5 rows present in the data set.
(2) List the values of any data points which lie beyond the extremes of the whiskers in the boxplot of ***x***
(3) List the values of any data points which lie beyond the extremes of the whiskers in the boxplot of ***y***
(4) Report the values of outliers in both ***x*** and ***y***
(5) Draw a 2-dimensional boxplot with the input data and mark the outlier data points with "orange" color.

P3. **(Clustering-based outlier detection)** For this practice, you will use the iris data set.
By grouping data into clusters, those data not assigned to any clusters are taken as outliers. In this practice, you will detect outliers with *k-mean* algorithm.

(1) Conduct *k-mean* with k=3. In clustering, do not include the class attribute, "Species". The clustering-based detection is an unsupervised method.
(2) Show the centroid of each cluster
(3) Calculate distances between objects and the cluster centroids. Each object will have three distances to three centroids.
(4) Compute the local outlier factor value of each example (instance) as its outlier score. In this computation, do not include the class attribute, "Species".
(5) Report the data objects with top 5 largest distances as outliers
(6) Plot the clusters and point their centroids and the five outliers. Because we cannot plot all four attributes, make a 2-dimensional plot with two attributes, PetalLength and PetalWidth.

P4. **(Density-based local outlier detection)** For this practice, you will use the iris data set.
LOF (Local Outlier Factor) is an algorithm for identifying density-based local outliers.

(1) Calculate the local outlier factor value of each object as its outlier score. In this calculation, do not include the class attribute, "Species".
(2) Report the data objects of top 5 LOF values as outliers
(3) Compute kernel density estimates with the LOF values and generate the plot diagram for plotting of them.

P5. **(Distance-based outlier detection)** For this practice, you will use the iris data set, https://archive.ics.uci.edu/dataset/53/iris
For this distance-based outlier detection, you will use *k*-nearest neighbor information. In nearest neighbor search, do not include the class attribute, "Species".

(1) Compute the distance from each data object to its *k*-nearest neighbor (*k*=1)
(2) Report the data objects with top 5 largest distances from (1) as outliers
(3) Compute the distance from each data object to its *k*-nearest neighbor (*k*=3)
(4) Report the data objects with top 5 largest distances from (3) as outliers
(5) If any, list the data objects in both lists from (2) and (4) as outliers

Next, we will use the number of neighboring objects of each object in a given neighbor distance for outlier detection

(6) Count the number of neighboring objects per each object within a distance threshold, 2
(7) Report the data objects with top 5 smallest neighbors as outliers