**Instructions**

- This homework assignment is worth 80 points.
- Please submit a **.ipynb** file to Blackboard.
- One submission per team.
- **Please strive for clarity and organization.**
- **Due Date: October 13, 2023 by 11:59 pm.**

# Exercise 1

(4 points) What is "$k$" in the $k$-NN algorithm? Be specific.

# Exercise 2

(4 points) Which is the following is **true** about the $k$-NN algorithm?

(a) When you increase $k$, the bias will increase as well.

(b) When you decrease $k$, the bias will increase.

(c) All of the above.

(d) None of the above.

# Exercise 3

(4 points) Why is important to standardize the data before $k$-NN?

# Exercise 4

(4 points) Given the following two statements, find which one of these options is **true** in the case of $k$-NN?

(a) In case of very large value of $k$, we may include points from other classes into the neighborhood.

(b) In case of too small value of $k$ the algorithm is very sensitive to noise.

(c) (a) and (b)

(d) None of the above.

# Exercise 5

Consider the very popular iris dataset. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The goal of this exercise is to predict the iris class. **In Python**, answer the following:

(a) (3 points) Using the `pandas` library, read the csv file and create a data-frame called `iris`.

(b) (5 points) Create the scatter-plot between `petal_length` and `petal_width`. Comment on the plot. Do you see any natural grouping in the data?

(c) (6 points) Calculate the $z$-score standardized values of the four variables and store them in new columns named `z_sepal_lenght`, `z_sepal_width`, `z_petal_length` and `z_petal_width`.

(d) (6 points) Split the `iris` data-frame into training and testing. Select the first 120 observations and the standardized variables for the training dataset, and the remaining observations for the testing dataset. Before you create your `X_train`, `Y_train`, `X_test` and `Y_test`, run the line of code shown below.

```
## Random shuffle of the observations
iris = iris.sample(frac = 1, random_state = 453).reset_index(drop = True)
```

Make sure that `X_train` and `X_test` contain only the standardized variables. On the other hand, `Y_train` and `Y_test` contain only the target variable (`class`).

(e) (4 points) Build a $k$-NN classifier using the training dataset and 4 neighbors.

(f) (4 points) Using the model from part (e), predict `class` on the testing set.

(g) (4 points) Compare the predictions against the actuals. Comment on the results.

# Exercise 6

Consider the `diamonds.csv` datafile. This datafile contains information related to almost 54,000 diamonds (including their prices). Here is a description of each the variables in the diamonds datafile:

- **price:** price in US dollars ($326–$18,823)
- **carat:** weight of the diamond (0.2–5.01)
- **cut:** quality of the cut (Fair, Good, Very Good, Premium, Ideal)
- **color:** diamond colour, from J (worst) to D (best)
- **clarity:** a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))

- **x:** length in mm (0–10.74)
- **y:** width in mm (0–58.9)
- **z:** depth in mm (0–31.8)
- **depth:** total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43–79)
- **table:** width of top of diamond relative to widest point (43–95)

The goal is to predict the price of diamonds. **In Python**, answer the following:

(a) (3 points) Using the `pandas` library, read the csv file and create a data-frame called `diamonds`.

(b) (5 points) Create the scatter-plot of `carat` and `price`. Comment on the plot.

(c) (6 points) Calculate the $z$-score standardized values of `carat`, `depth`, `table`, `x`, `y` and `z`, and store them in columns named `z_carat`, `z_depth`, `z_table`, `z_x`, `z_y` and `z_z`.

(d) (6 points) Split the `diamonds` data-frame into training and testing. Select the first 43,000 observations and the standardized variables for the training dataset, and the remaining observations for the testing dataset. Before you create your `X_train`, `Y_train`, `X_test` and `Y_test`, run the line of code shown below.

```
## Random shuffle of the observations
diamonds = diamonds.sample(frac = 1, random_state = 823).reset_index(drop = True)
```

Make sure that `X_train` and `X_test` contain only the standardized variables. On the other hand, `Y_train` and `Y_test` contain only the target variable (`price`).

(e) (4 points) Build a $k$-NN regressor using the training dataset and 10 neighbors.

(f) (4 points) Using the model from part (e), predict `price` on the testing set.

(g) (4 points) Compare the predictions against the actuals by creating a scatter plot. Comment on the results.