**k-Nearest Neighbor (kNN) classification**

**Tasks**

1. Identify and describe the characteristics of the dataset (given) for you to work with for this assignment (e.g., original resource information, dimension, etc.)
2. Execute a kNN classification algorithm (given) on the two platforms, including (1) **kNN.py** (on Python command-line) and (2) **kNN.ipynb** (on Google Colab).
3. Get an initial accuracy result without applying any sophisticated strategies.
4. Figure out what strategies you can apply to improve the performance.
5. Explain how you can specifically tailor the learning strategies for the data.
6. If possible, visualize your result in a couple of ways of visualizations, which you can find from the references (given) as well as from other resources.
7. Write a short report that describes your findings.

**Dataset**

1. Pima Indians Diabetes data (“diabetes.csv”)
2. The data file is given in the assignment directory so that you can run kNN.py with the data as given.
3. Where can you find the original data?
4. Briefly describe the characteristics of the original data. What is the shape? How many classes? How many instances, features, and classes?

**What should you do?**

1. Briefly explain about the data.
2. Notice that the minimal python code ((1) kNN.py) is given for you to start with. Therefore, you can just use it as it is. However, you are welcome to extend it as you wish for your experimental study.
3. Based on (1) kNN.py, (2) kNN.ipynb can be generated. You need to figure out how to read the data file from Google Colab. For example, you can read the data file directly from the original data repository. Another way is to first store the data file into Google Drive and read it using the file path in Google Drive.
4. Find out how you can improve the overall accuracy performance of kNN classification on the data set. Specifically, you are asked to investigate the classification accuracy performance with varied parameter values, which include feature engineering, the number of nearest neighbors (i.e., k), cross-validation split, etc.
5. Visualize your result if possible and briefly discuss the findings if you have any.
6. Record any heuristics or tips you learned from other resources and the experimental study.

**Due date**

Unless otherwise mentioned, every assignment is due one week after the posting.

**Grading**

The total point of this assignment is 100.

**Report format and structure**

The report length should be minimum three typed pages in Times New Roman (font size of 12) with references and possible figures. Specifically, when writing the report, make sure to include the following:

1. Abstract (optional)
2. Introduction
3. Related work (optional)
4. Methodology (including (1) kNN.py and (2) kNN.ipynb as well as other approaches you applied)
5. Data (brief description of the given data file)
6. Result
7. Discussion
8. Conclusion
9. Future work (if you have any, optional)
10. Reference (including the ones given below)

**Reference**

1. <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
2. <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
3. <https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a>
4. <https://www.kaggle.com/uciml/pima-indians-diabetes-database>
5. <https://www.kaggle.com/amolbhivarkar/knn-for-classification-using-scikit-learn>
6. <https://www.kaggle.com/shrutimechlearn/step-by-step-diabetes-classification-knn-detailed>
7. <https://www.kaggle.com/pouryaayria/a-complete-ml-pipeline-tutorial-acu-86>