Lab 2

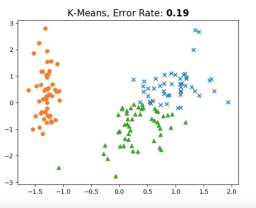
It is an individual programming assignment. This lab assignment is graded based on 100 points and is an individual effort (e.g, no teamwork is allowed).

**Part 1 (10x6 = 60 pts) KNN & K-mean**

For the datasets provided, complete all the steps requested: **NOTE: You should use both your own NumPy implementation of the algorithms and sklearn.**

**Step 1:** Load the data points and ground truth values (Some ground truth values may be in a different file)

**Step 2 (1 pts)**: Plot the points according to the ground truth value classes. For example if there are 3 different classes, expected output:



**Step 3:** Split data into train and test.

**Step 4 (2 pts):** Use KNN **(own NumPy code & sklearn library)** to clusters the data. **Step 5 (1 pts):** Plot your result in different colors similar to Step 2.

**Step 6 (1 pts):** Print the confusion matrix to check model performance.

**Step 7 (4 pts):** Repeat Steps 3 - 6 and use K-means **(own NumPy code & sklearn library)** to perform the same. Additionally, print the centroids.

**Step 8 (1pts):** Compare the results between KNN and K-means

Datasets:

1. **Dataset\_1/S1.txt:** N=5000 vectors and k=15 Gaussian cluster

2. **Dataset\_1/S2.txt:** N=5000 vectors and k=15 Gaussian cluster

3. **Dataset\_1/S3.txt**: N=5000 vectors and k=15 Gaussian cluster

4. **Dataset\_1/S4.txt:** N=5000 vectors and k=15 Gaussian cluster

NOTE: Ground Truth and original centroid for S1-4 are present in the **dataset\_1/s-originals** folder.

5. **Dataset\_2/dim32.txt**: 32 dimensions;

6. **Dataset\_3/spiral.txt:** N=312, k=3, D=2; Ground truth is the third column present in the same file.

**Part 2 (10x2 = 20 pts) Use Neural Networks + minimum 3 hidden layers**

For the dataset, hcc-data-complete-balanced.csv, do the following:

1. Data clean up – feature engineering
2. Impute the missing values with mean, median and mode. You need to evaluate which method is better based on the F1 score.
3. Build a neural network with at least 3 hidden layers and 4 neurons each. Train and test.
4. Which activation functions are you using? Why?
5. Tune the hyperparameters using cross-validation and see what precision you can achieve
6. Is using Adam optimization and early stopping helpful in this problem? Why?
7. Now try adding Batch Normalization and compare the learning curves: is it converging faster than before? Does it produce a better model?
8. Is the model overfitting the training set? Try adding dropout to every layer and try again. Does it help?
9. What is the final model you’ve arrived? Draw the neural network to explain your solution.
10. Mention your F-1 score for each development in your model

**Part 3 (20 pts) Use XGBoost (Highlighted questions carry 5 points each).**

Using XGBoost, predict avocado price.

Load the dataset

1. Train and test the data
2. Perform feature engineering
3. What features are the most correlated?
4. Build a model
5. Use XGBoost
6. Fine tune the parameters – explain each and every step in detail. Which parameter? Why this value?
7. Evaluate the performance of your model
8. Explain in detail what is happening inside your model? How have you built this model?