1. **Banknote authentication Dataset**

   Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.

   (a) Download the Skin Segmentation data from: `https://archive.ics.uci.edu/ml/datasets/banknote+authentication#`.

   (b) Pre-Processing and Exploratory data analysis:

      i. Make scatterplots of the independent variables in the dataset. Use color to show Classes 0 and 1.

      ii. Make boxplots for each of the independent variables. Use color to show Classes 0 and 1 (see ISLR p. 129).

      iii. Select the first 200 rows of Class 0 and the first 200 rows of Class 1 as the test set and the rest of the data as the training set.

   (c) Classification using KNN on Banknote authentication Dataset

      i. Write code for k-nearest neighbors with Euclidean metric (or use a software package).

      ii. Test all the data in the test database with k nearest neighbors. Take decisions by majority polling. Plot train and test errors in terms of $1/k$ for $k \in \{1, 4, 7, \ldots, 901\}$. You are welcome to use smaller increments of k. Which k^* is the most suitable k among those values? Calculate the confusion matrix, true positive rate, true negative rate, precision, and F-score when $k = k^*$.

      iii. Since the computation time depends on the size of the training set, one may only use a subset of the training set. Plot the *best error rate*, which is obtained by some value of k, against the size of training set, when the size of training set is $N \in \{50, 100, 150, \ldots, 900\}$.[1] Note: for each N, select your training set by choosing the first $N/2$ rows of Class 0 and the first $N/2$ rows of Class 1 in the training set you creatd in 1(b)iii. Also, for each N, select the optimal k from a set starting from $k = 1$, increasing by 40. For example, if $N = 250$, the optimal k is selected from $\{1, 41, 81, \ldots, 241\}$. This plot is called a *Learning Curve*.

   Let us further explore some variants of KNN.

   (d) Replace the Euclidean metric with the following metrics[2] and test them. Summarize the test errors (i.e., when $k = k^*$) in a table. Use all of your training data and select the best k when $k \in \{1, 11, 21, \ldots, 901\}$.

      i. Minkowski Distance:

         A. which becomes Manhattan Distance with $p = 1$.

[1] For extra practice, you are welcome to choose smaller increments of N.

[2] You can use sklearn.neighbors.DistanceMetric. Research what each distance means.

B. with $\log_{10}(p) \in \{0.1, 0.2, 0.3, \ldots, 1\}$. In this case, use the k^* you found for the Manhattan distance in 1(d)iA. What is the best $\log_{10}(p)$?

C. which becomes Chebyshev Distance with $p \to \infty$

ii. Mahalanobis Distance.

(e) The majority polling decision can be replaced by weighted decision, in which the weight of each point in voting is proportional to its distance from the query/test data point. In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away. Use weighted voting with Euclidean, Manhattan, and Chebyshev distances and report the best test errors when $k \in \{1, 11, 21, \ldots, 901\}$.

(f) What is the lowest training error rate you achieved in this exercise?