Programming Assignment 2

For this assignment, we are going to be implementing three classifiers - **Decision Tree**, **Naïve Bayes**, and **Nearest Neighbor classifier.**

- Each classifier is using its own dataset.
- Each part is having its own tasks.
- Each part is presenting a challenge.

Decision Tree Classifier:

Dataset information

Load dataset_DT.csv
Refer to the assignment 1 to get the description of the dataset.

Tasks

1. Import the libraries and load the dataset (from the csv file).
2. Preprocessing step: you can use the result of your assignment 1 or if you have any other new approach you can apply it .
3. Determine Six most influential attributes on target attribute (with explanation). You do not necessarily need to drop the remaining features for the Decision Tree Classifier. Your task is just to determine and show the Six most influential attributes with detailed explanation.
4. Split your dataset 75% for training, and 25% for testing the classifier.
5. Use gini and entropy (play around with max_depth and min_samples_leaf parameters) to measure the quality of a split.
6. Use comments to explain your code and variable names.
7. Calculate and print the confusion matrix (use graphics instead showing a 2D array), and the classification Report (includes: precision, recall, f1-score, and support) for both criteria.
8. Compare the results of the classifier using gini and entropy.
9. Print the decision tree visualization with depth of 5.

Hints:

1. Categorized Data is preferable for decision trees. If needed, figure out how to convert continuous feature to categorical and implement it.
2. Unwanted data can reduce the model's accuracy

Naïve Bayes Classifier:

Dataset information

Features Description:

- email → text data of the actual email

Target variable:

- Label → spam(1) or not spam(0)

Tasks

1. Load the dataset as pandas dataframe.
2. You have textual data that you cannot feed into the model. Therefore, you need to extract features from the text (email) and transform the data.
3. Test train split, using 80% for training, rest for testing.
4. Train NB model (Gaussian) for classification, on the training data.
5. Predict on the test data.
6. Get the accuracy, plot the confusion matrix, report Accuracy Score(metrics.accuracy_score), and plot Confusion Matrix(metrics.confusion_matrix) plotted graphically.
7. Create a report file to include concise answers to the following questions -

    a. Briefly explain your approach, any preprocessing, explain the output, any visualization for explanation, any feature extraction, in same colab file (3-4 paragraphs max)

Hints

1. There are techniques to extract features, such as Bag of Words, n-grams, Tf-Idf, Word2Vec, CountVectorizer, and many others.
2. Know your data. Look at the data in dataset (Open the data file and see the data or use pandas to check the info).

Nearest Neighbor Classifier:

Dataset information

Load dataset_NN.csv

The data is ordered by age.

Features Description:

- Pregnancies → Number of times pregnant
- Glucose → Plasma glucose concentration 2 hours in an oral glucose tolerance test
- BloodPressure → Diastolic blood pressure (mm Hg)
- SkinThickness → Triceps skin fold thickness (mm)
- Insulin → 2-Hour serum insulin (mu U/ml)
- BMI → Body mass index (weight in kg/(height in m)^2)
- DiabetesPedigreeFunction → Diabetes pedigree function
- Age → Age (years)

Target Variable:

- Outcome - Class variable (0 or 1)

Tasks:

1. Load dataset_NN.csv dataset.
2. Data Pre-processing.
3. Using Pearson's Correlation Coefficient find out the relation between variables using Heat Map (Draw heat maps before and after cleaning data to find differences)
4. Scale the data and mention which scaling technique used.
5. Split your dataset 75% for training, and 25% for testing and do cross validation for the classifier.
6. Find the best K using elbow method.
7. Use Euclidean distance.
8. Select three best attributes and explain why you chose them.
9. Test the classifier with three different k values for neighbors and record the results.
10. Plot the ROC curve for best K value.
11. Use comments to explain your code at each step of all points.

12. Calculate and print the confusion matrix, and the classification Report (includes: precision, recall, f1-score, and support) for all three different numbers. Plot the Error rate vs. K-value.
13. Create a report file to include concise answers to the following questions:
    a. Describe the Nearest Neighbors method and why scaling is important in KNN.
    b. Explain what your criteria was for selecting the three attributes. What other 3 attribute can you choose? Visualizations of the target variable with three most significant attributes in a 2D projection, and write your observations in 4 - 5 lines in the same collab file
    c. Explain Pearson's Correlation Coefficient, write the observations from heatmaps drawn.
    d. Interpret and compare the results.

Hints:

1. Dataset consists of Nan values/ null values, to pre-process the data, you simply should not replace with mean/median, instead understand the data distribution, and do data preprocessing.
2. You can use libraries: NumPy, Pandas, Scikit-learn, Matplotlib and Seaborn
3. While choosing K-values, that should be meaningful, you cannot just simply choose and do analysis. Describe why you choose only those particular K values.
4. Models' accuracy depends on the first step data preprocessing

Programming Assignment Details:

- For this assignment use colab.

- For each part create a colab file.
- You can use libraries: NumPy, Pandas, Scikit-learn, Matplotlib and Seaborn.
- Make sure to explain any kind of visualization.
- Report file has to be with the same colab file.