Cairo University
Faculty of Computers and Artificial Intelligence
Machine Learning Course

Assignment 2: Decision Tree, kNN and Naïve Bayes

The Fifth Assessment Report from the Intergovernmental Panel on Climate Change (IPCC) confirms that our climate and its extreme events are changing. To reduce the risks and damage caused by these weather and climate extremes, accurate predictions are essential for both short-term and long-term planning. Understanding, modeling, and predicting these extremes is a key area of climate research, and it has been identified as one of the World Climate Research Program's (WCRP) Grand Challenges, known as the Extremes Grand Challenge. Although weather forecasting presents many challenges, we have created a simple task as an introduction to this area. Using our dataset, we aim to predict whether it will rain.

Dataset:

Our dataset is a modified version of a weather forecast dataset obtained from Kaggle. It includes 6 features and 2,500 observations. Your task is to analyze and work with this data to answer the following questions, using Python code:

Requirements:

Task 1: Preprocessing

1. Does the dataset contain any missing data? Identify them.
2. Apply the two techniques to handle missing data, dropping missing values and replacing them with the average of the feature.
3. Does our data have the same scale? If not, you should apply feature scaling on them.
4. Splitting our data to training and testing for training and evaluating our models

Task 2: Implement Decision Tree, k-Nearest Neighbors (kNN) and naïve Bayes

1. Using scikit-learn implement Decision Tree, kNN and Naïve Bayes
2. Compare the performance of your implementations by evaluating accuracy, precision, and recall metrics.
3. Implement k-Nearest Neighbors (kNN) algorithm **from scratch**.
4. Report the results and compare the performance of your custom k-Nearest Neighbors (kNN) implementation with the pre-built kNN algorithms in scikit-learn, using the evaluation metrics mentioned in point 2. Using any missing handling techniques, you chose from task 1.2.

Task 3: Interpreting the Decision Tree and Evaluation Metrics Report

1. **The effect of different data handling**
    - Provide a detailed report evaluating the performance of scikit-learn implementations of the Decision Tree, k-Nearest Neighbors (kNN) and naïve Bayes with respect to the different handling missing data technique.
2. **Decision Tree Explanation Report**
    - Create a well-formatted report that includes a plot of the decision tree and a detailed explanation of how the tree makes predictions.
    - Discuss the criteria and splitting logic used at each node of the tree.
3. **Performance Metrics Report**
    - Provide a detailed report evaluating the performance of your implementations of the k-Nearest Neighbors (kNN) from scratch with different k values at least 5 values.
    - Include the accuracy, precision, and recall metrics for models.
    - Compare these results with the performance of the corresponding algorithms implemented using scikit-learn.

Remarks:

- You can use functions from **data analysis and computing libraries** (e.g. Pandas and NumPy) as you please throughout the entire code.

- You can use **machine learning libraries** such as Scikit-learn for preprocessing and metrics **but NOT for "from scratch" requirements.**

- The **train/test split** has to be **performed before** the feature scaling step.

- **The numeric features of the test set should be scaled using the statistics of the train set that were used to scale it.**

Deliverables:

- **You are required to submit ONE zip file containing the following:**
  - Your **code (.py)** file.

    If you have a (.ipynb) file, you have to save/download it as (.py) before submitting.

  - A **report (.pdf)** containing the team members' names and IDs, and the code of each requirement with screenshots of the output of each part.

- **The zip file MUST follow this naming convention:**
  Group_A2_ID1_ID2_ID3_ID4

Submission Remarks:

- The **maximum** number of students in a team is **5** and the **minimum is 3**.
- Team members must be from the **same lab** (or have the same TA).
- All team members must understand all parts of the code.
- **No late submission** is allowed.
- Stick to uploading **ONLY** the **required files** following the **naming convention:** Group_A2_ID1_ID2_ID3_ID4
- A **penalty** will be **imposed for violating** any of the assignment rules.
- **Cheaters will get ZERO** and no excuses will be accepted as per the **"Plagiarism Scope"** document.

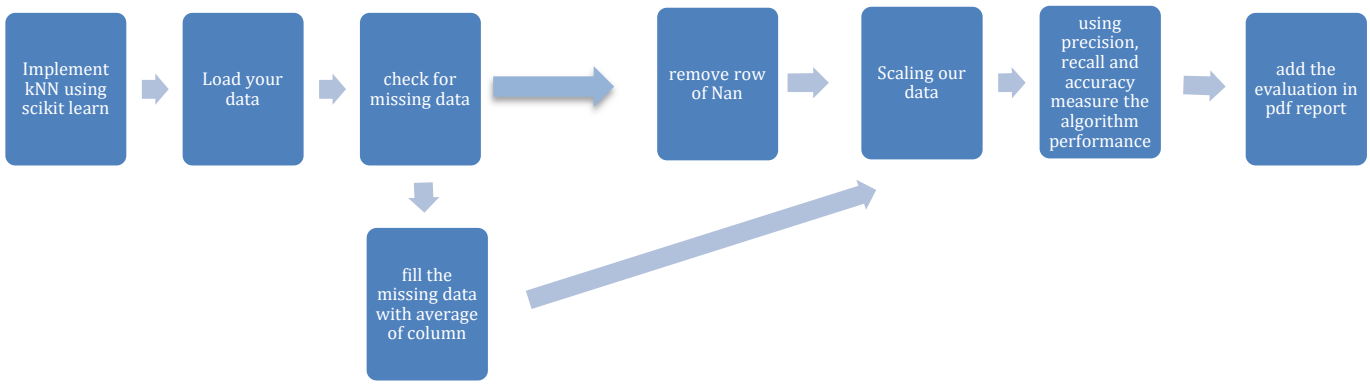
3 Metrics:
Accuracy, Precision, and Recall metrics.

Grading Criteria:

Both the code and the report must include:	
Discussion	**3 marks**
Preprocessing	
Handle missing data by 2 techniques	**1.5 marks**
Scaling	**0.5 mark**
Splitting our data	**0.5 mark**
Naïve Bayes	
Implementation (scikit-learn)	**1 mark**
Reporting performance using 3 metrics	**1.5 marks**
Decision Tree	
Implementation (scikit-learn)	**1 mark**
Reporting performance using 3 metrics	**1.5 marks**
KNN	
Implementation (scikit-learn)	**1 mark**
Implementation (scratch)	**2 marks**
Comparison using 3 metrics Task 2.4	**1.5 marks**
Post Processing	
Plotting the scikit-learn Decision Tree Task 3.2	**1 mark**
Reporting	
Comparison of kNN Scratch vs the scikit-learn	**1 mark**
Comparing two missing data handling techniques with scikit-learn implementations Task 3.1	**1 mark**
Plotting Decision Tree with clarifying each node	**2 marks**
The total is 20 marks (will be scaled to 6 marks)	

PLOT Decision Tree and save plot output in pdf with clarify by you own words what check node check and do

Same Flow applied for Naïve Bayes and Decision Tree



PLOT Decision Tree and save plot output in pdf with clarify by you own words what check node check and do