ASSIGNMENT 1

PYTHON FOR DATA SCIENCE COURSE

NAME: Nikhil V

SRN: PES2UG22EC088

MCQs (1 mark each)

Q1. Which of the following libraries in Python is primarily used for data manipulation and analysis, offering data structures such as Series and DataFrame?
a) NumPy
b) Pandas
c) Matplotlib
d) Scikit-learn
Answer: b) Pandas

Q2. Let `X` be a NumPy array of shape `(n_samples, n_features)`. Which expression standardizes each feature column to zero mean and unit variance without scikit-learn?

a) `(X - X.mean(axis=1)) / X.std(axis=1)`
b) `(X - X.mean(axis=0)) / X.std(axis=0)`
c) `(X - X.mean()) / X.std()`
d) `(X - X.mean(axis=0, keepdims=True)) / X.std(axis=1, keepdims=True)`

Answer: b)
Why: Feature-wise standardization uses statistics along `axis=0` (columns).

Descriptive Questions (5–7 marks each)

Q3. (5 Marks) Explain the differences between **NumPy arrays** and **Python lists**. Why are NumPy arrays preferred in data science applications?
Answer:

- **Storage:** NumPy arrays store elements in contiguous memory blocks, while Python lists are collections of references, making NumPy more memory-efficient.
- **Performance:** NumPy arrays allow vectorized operations (element-wise addition, multiplication, etc.), which are faster compared to looping through lists.
- **Data type:** NumPy arrays enforce a single data type, whereas lists can store mixed types, leading to slower computations.
- **Functionality:** NumPy provides built-in functions for linear algebra, statistics, and broadcasting that are not available in Python lists.
  NumPy is preferred because of **speed, efficiency, and rich mathematical functions**, which are essential in data science.

Q4. (7 Marks) Describe the steps involved in building a **machine learning model pipeline** in Python using scikit-learn. Provide an example.
Answer:
The steps are:

1. **Data Collection** – Gather the dataset (CSV, database, API, etc.).
2. **Data Preprocessing** – Handle missing values, encode categorical variables, scale/normalize data.
3. **Train-Test Split** – Divide dataset into training and testing sets using `train_test_split`.
4. **Model Selection** – Choose an appropriate algorithm (e.g., Logistic Regression, kNN, Random Forest).
5. **Model Training** – Fit the model on training data.
6. **Model Evaluation** – Use accuracy, confusion matrix, RMSE, or R^2 score to check performance.
7. **Prediction** – Apply the model to test data or new inputs.