Assignment

Modules: Pandas | NumPy | Metrics

Part 1: NumPy – Numerical Computing (30 Marks)

Task 1: Array Creation and Operations (15 Marks)

1. Create the following arrays:
    - A 1D array of integers from 0 to 20.
    - A 2D array of shape (4, 5) filled with random integers from 10 to 99.
    - A 3x3 identity matrix.
2. Perform the following operations:
    - Find the mean, median, and standard deviation of the 2D array.
    - Slice and print the second row and third column of the 2D array.
    - Multiply two arrays element-wise.

Deliverable: Python code and printed outputs with comments

Task 2: Broadcasting and Reshaping (15 Marks)

1. Reshape a 1D array of size 16 into a 4x4 matrix.
2. Add a 1D array `[1, 2, 3, 4]` to each row of the matrix using broadcasting.
3. Flatten the matrix back to a 1D array.

Deliverable: Code with explanations

Part 2: Pandas – Data Manipulation (40 Marks)

Task 3: Working with DataFrames (20 Marks)

1. Create a DataFrame with the following data:

Name	Age	Department	Salary
Alice	25	IT	50000
Bob	30	HR	45000
Charlie	28	IT	72000
Diana	35	Marketing	58000
Evan	40	HR	60000

2. Perform these operations:
    - Add a new column `Bonus` = 10% of Salary.
    - Filter and display employees in the `HR` department.
    - Calculate average salary by department.
    - Save the DataFrame to a CSV file.

Deliverable: Python code and resulting DataFrame

Task 4: Data Cleaning & Analysis (20 Marks)

Use any public CSV dataset (e.g., from Kaggle or UCI ML Repository)

1. Load the dataset using `pandas.read_csv()`
2. Show:
    - First 5 and last 5 rows
    - Shape of dataset
    - Columns and their data types
3. Handle missing values (e.g., fill, drop)

4. Generate basic statistics using `.describe()`

Deliverable: Cleaned dataset and code

Part 3: Classification Evaluation with Confusion Matrix (30 Marks)

Task 5: Dataset-Based Classification Evaluation (30 Marks)

🔹 **Dataset:**

Use the **Iris Dataset** from `sklearn.datasets`, but modify it into a **binary classification** problem:

```
from sklearn.datasets import load_iris
import pandas as pd

# Load Iris dataset
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

# Convert to binary classification (classify Versicolor vs not-Versicolor)
df['binary_target'] = (df['target'] == 1).astype(int)
```

🔹 **Documentation:**
https://scikit-learn.org/stable/api/sklearn.metrics.html#classification-metrics

🔹 **Steps to Perform:**

1. **Split the data** into training and testing sets (e.g., 70/30 split).
2. Train a simple **Logistic Regression** classifier.
3. Use the model to make predictions on the test set.
4. **Generate the Confusion Matrix** using `sklearn.metrics.confusion_matrix`.
5. **Calculate and display the following metrics**:
    - Accuracy
    - Precision

- Recall
- F1-Score

6. Print or display results clearly, and add a brief explanation of what each metric means.

Sample Code Reference (for guidance):

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score, f1_score

# Train-test split
X = df[iris.feature_names]
y = df['binary_target']
[todo]

# Train model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
[todo]
```

📤 **Deliverables:**

- Python code and output for:
  - Confusion matrix (as a table or printed array)
  - All 4 metrics
  - Short explanation of each metric

Submission Guidelines

- **Format**: Notebook Link (Google Colab/Kaggle)
- **Instruction**: Outputs must be visible

Grading Breakdown

Section	Marks
NumPy Operations	30
Pandas Data Manipulation	40
Classification Evaluation	30
Total	**100**