Name: Atharva Dattatray Kshirsagar

Type: AI/ML Internship Tasks

TASK – 1

- **Title:**

  Iris Flower Classification using Machine Learning

- **Objective:**

  To classify iris flowers into three species — **Setosa, Versicolor**, and **Virginica** — based on **sepal** and **petal** measurements using a supervised **machine learning** algorithm.

- **Dataset:**
  - Iris Dataset (scikit-learn)
  - 150 samples
  - 4 numerical features
  - 3 target classes

- **Methodology:**

1. Loaded dataset using scikit-learn
2. Split data into training and testing sets (80:20)
3. Applied Logistic Regression classifier
4. Trained the model on training data
5. Evaluated performance using accuracy score and confusion matrix

---

- ✓ **Results**
  - **Accuracy:** 100%
  - **Confusion Matrix:** No misclassification observed
  - Model successfully classified all test samples

Python Code (with Output) :

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix

print("Script started...")

# Load dataset
iris = load_iris()
X = iris.data
y = iris.target

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Train model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

"""
Output:
Script started...
Accuracy: 1.0
Confusion Matrix:
 [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
 """
```

- **Output explanation:**
  - **Setosa:** 10/10 correct
  - **Versicolor:** 9/9 correct
  - **Virginica:** 11/11 correct

Accuracy = 1.0 i.e. 100%

- **Skills Demonstrated:**
    - Data preprocessing
    - Supervised classification
    - Model training & evaluation
    - Use of scikit-learn

TASK – 2

- **Title:**
  Spam Mail Detection using Machine Learning

- **Objective:**

  To build a machine learning model that **classifies text messages as spam** or **non-spam (ham)** using **Natural Language Processing (NLP)** techniques.

- **Dataset:**
  - SMS Spam Collection Dataset (UCI Repository)
  - Text-based dataset containing spam and ham messages

- **Methodology:**
  1. Loaded SMS spam dataset
  2. Cleaned text by removing symbols and converting to lowercase
  3. Converted text into numeric features using TF-IDF
  4. Split dataset into training and testing sets (80:20)
  5. Trained a Naive Bayes classifier
  6. Evaluated the model using accuracy, confusion matrix, and F1 score

---

- **Results**
  - **Accuracy: ~96%**
  - **High precision and recall for spam detection**
  - **Model successfully distinguished spam and non-spam messages**

Python Code :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

print("Script started...")

# Load dataset (TAB separated, no extension)
data = pd.read_csv(
    "SMSSpamCollection",
    sep="\t",
    names=["label", "message"]
)

# Convert labels to numeric
data["label"] = data["label"].map({"ham": 0, "spam": 1})

# Split features and target
X = data["message"]
y = data["label"]

# Text to numbers using TF-IDF
vectorizer = TfidfVectorizer(stop_words="english")
X_vectorized = vectorizer.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X_vectorized, y, test_size=0.2, random_state=42
)

# Train Naive Bayes model
model = MultinomialNB()
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))
"""
Output:
Script started...
```

```
Accuracy: 0.979372197309417
Confusion Matrix:
 [[966   0]
 [ 23 126]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       966
           1       1.00      0.85      0.92       149

    accuracy                           0.98      1115
   macro avg       0.99      0.92      0.95      1115
weighted avg       0.98      0.98      0.98      1115

"""
```

➢ **Output Explanation:**

- **966 Ham messages** → correctly predicted as Ham
- **126 Spam messages** → correctly predicted as Spam
- **23 Spam messages** → mistakenly marked as Ham
- **0 Ham messages** → falsely marked as Spam

➢ **Classification Report – Key Observations:**

- **Spam Class (Label = 1)**
- **Precision: 1.00**
  All messages predicted as spam were actually spam, indicating **zero false positives**.
- **Recall: 0.85**
  The model successfully identified 85% of actual spam messages. A small number of spam messages were missed, which is an acceptable trade-off.
- **F1-Score: 0.92**
  The F1-score reflects a strong balance between precision and recall, demonstrating reliable spam detection performance.
- **Ham Class (Label = 0)**
- **Recall: 1.00**
  All legitimate (non-spam) messages were correctly classified, ensuring that **no genuine message was incorrectly marked as spam**.

- **Skills Demonstrated**
  - Text preprocessing
  - Feature extraction (TF-IDF)
  - Natural Language Processing
  - Classification using Naive Bayes
  - Model evaluation
