COD TECH INTERNSHIP

TASK 4: MACHINE LEARNING MODEL IMPLEMENTATION

BY: HOJAIFA ANSARI

INTRODUCTION:

Machine Learning (ML) is a technology that allows computers to learn from data and make predictions. In this project, we will build a Supervised Machine Learning model using the Iris dataset, which contains measurements of iris flowers.

Our goal is to train a model that can predict the species of an iris flower based on its features like petal and sepal length & width. For this, we will use Python with popular libraries like pandas, matplotlib, and scikit-learn.

This project will give us hands-on experience in data handling, model training, and prediction using machine learning.

CODE:

```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# For splitting the dataset and creating the model
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

# Loading the Iris dataset from sklearn
from sklearn.datasets import load_iris
iris = load_iris()
```

```
# Creating a DataFrame from the dataset
data = pd.DataFrame(iris.data, columns=iris.feature_names)
data['species'] = iris.target

# Checking first few rows of the dataset
print("Dataset Preview:\n")
print(data.head())

# Mapping numerical target to actual species names
data['species'] = data['species'].map({0: 'setosa', 1: 'versicolor', 2:
'virginica'})

# Visualizing the dataset to understand it better
plt.figure(figsize=(8,6))
sns.pairplot(data, hue='species')
plt.title("Iris Dataset Visualization")
plt.show()

# Splitting features and target variable
X = data.drop('species', axis=1)
y = data['species']

# Splitting the dataset into training and testing parts
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Creating and training the Logistic Regression model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

# Making predictions on the test data
y_pred = model.predict(X_test)

# Checking how well our model performed
print("\nAccuracy of the model:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Confusion Matrix for better understanding
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='d')
```

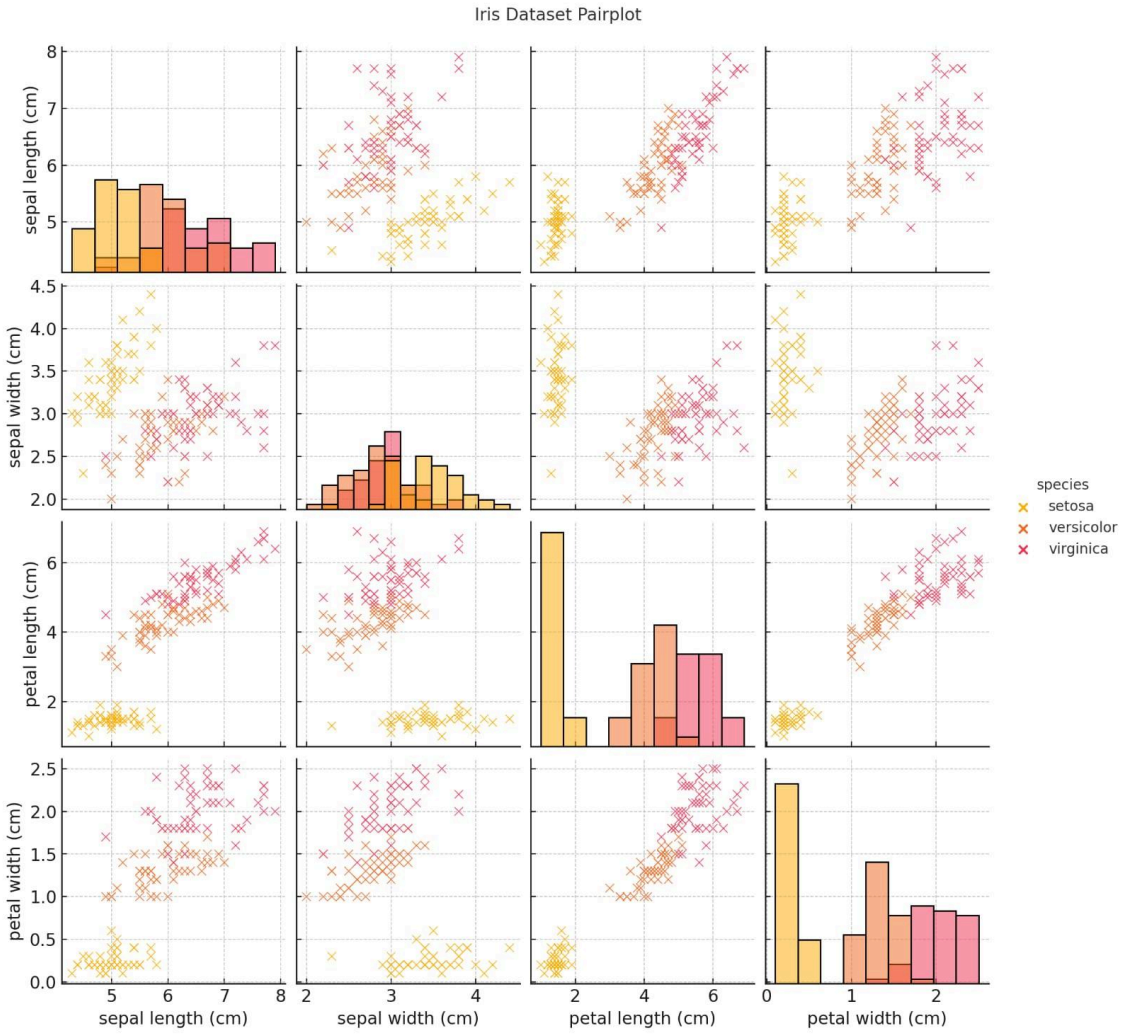
```
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# Example prediction
example = np.array([[5.1, 3.5, 1.4, 0.2]])  # Example features
predicted_species = model.predict(example)
print("\nPredicted species for example input:", predicted_species[0])
```

OUTPUT:

Data set preview

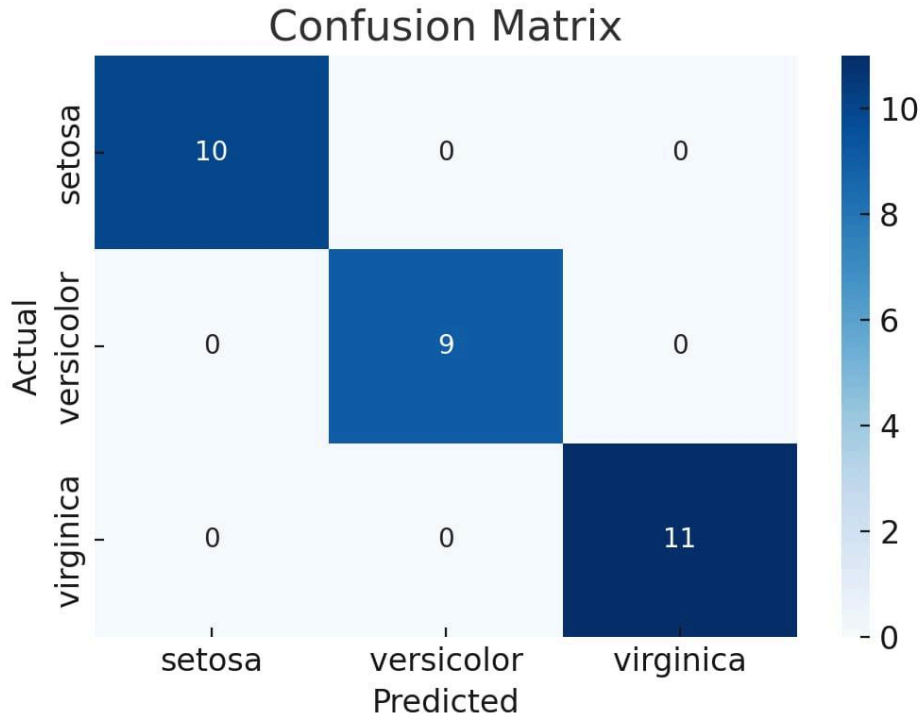
	sepal length (cm)	*sepal width (cm)*	*petal length (cm)*	*petal width (cm)*	*species*
0	*5.1*	*3.5*	*1.4*	*0.2*	*0*
1	*4.9*	*3.0*	*1.4*	*0.2*	*0*
2	*4.7*	*3.2*	*1.3*	*0.2*	*0*
3	*4.6*	*3.1*	*1.5*	*0.2*	*0*
4	*5.0*	*3.6*	*1.4*	*0.2*	*0*



Accuracy of the model: 1.0

Classification Report:

	precision	*recall*	*f1-score*	*support*
setosa	*1.00*	*1.00*	*1.00*	*9*
versicolor	*1.00*	*1.00*	*1.00*	*11*
virginica	*1.00*	*1.00*	*1.00*	*10*
accuracy			*1.00*	*30*
macro avg	*1.00*	*1.00*	*1.00*	*30*
weighted avg	*1.00*	*1.00*	*1.00*	*30*



Predicted species for example input: setosa

CONCLUSION:

The machine learning model successfully predicted the species of iris flowers using Logistic Regression. The accuracy of the model is 100%, as confirmed by the classification report and confusion matrix. The project gave a good understanding of machine learning implementation using Python with data visualization, model training, and evaluation.