INTRODUCTION TO AI

PROJECT REPORT

PROBLEM STATEMENT: - Iris Flower Classification – Classify flower species based on petal and sepal dimensionsusing the Iris dataset.

NAME: - KANISHAK TYAGI
BRANCH: - CSE(AIML)
SECTION: - B
ROLL NO.: - 202401100400104

Introduction

Problem Description:

The goal of this project is to develop a machine learning model capable of accurately classifying iris flowers into one of three species: Setosa, Versicolour, or Virginica. The classification is based on four numerical features describing the flowers: sepal length, sepal width, petal length, and petal width.

This is a supervised learning problem where the model is trained on a labeled dataset to predict the class of new, unseen data. The challenge lies in finding the optimal decision boundaries that separate the three species effectively using a simple yet robust classification algorithm.

Objective

The objective of this project is to build a machine learning model capable of classifying iris flowers into three species (Setosa, Versicolour, Virginica) based on their physical attributes: sepal length, sepal width, petal length, and petal width. The model aims to achieve high accuracy and generalization using Logistic Regression while also providing useful visualizations to understand the classification process.

Dataset Description

The Iris dataset is a well-known dataset introduced by Ronald Fisher in 1936. It contains 150 samples of iris flowers, equally distributed among three classes:

- Setosa (0)
- Versicolour (1)
- Virginica (2)

Features:

- Sepal Length (cm)
- Sepal Width (cm)
- Petal Length (cm)
- Petal Width (cm)

Target:

Class labels indicating the species of the flower.

How The Code Works

Data Loading & Preprocessing:

The dataset is loaded using the sklearn.datasets library.

Features (X) and labels (y) are extracted.

The data is split into training and testing sets (80% training, 20% testing).

Feature scaling is applied using StandardScaler to standardize the data for better model performance.

1.Model Training:

A Logistic Regression model is created using LogisticRegression() from sklearn.

The model is trained using the training dataset via the fit() method.

2.Prediction & Evaluation:

Predictions are made on the testing set using the predict() method.

The model's performance is evaluated using metrics like Accuracy, Classification Report, and Confusion Matrix.

3.Visualization:

A pairplot is generated to visualize the relationships between features.

A heatmap of the Confusion Matrix is displayed to show prediction performance.

4. AI Strategy

Data Preprocessing: Ensuring high-quality input through scaling and splitting of data.

Algorithm Selection: Logistic Regression was chosen for its simplicity and effectiveness for linear classification tasks.

Evaluation: Performance was measured using accuracy, classification report, and confusion matrix.

Optimization: Hyperparameter tuning and trying different algorithms can be done to improve performance further.

Conclusion:

The Logistic Regression model achieved good accuracy in classifying the Iris dataset. However, further improvements can be made by trying other algorithms such as K-Nearest Neighbors, Decision Trees, or Support Vector Machines. Additionally, cross-validation techniques can be applied to enhance generalization performance.

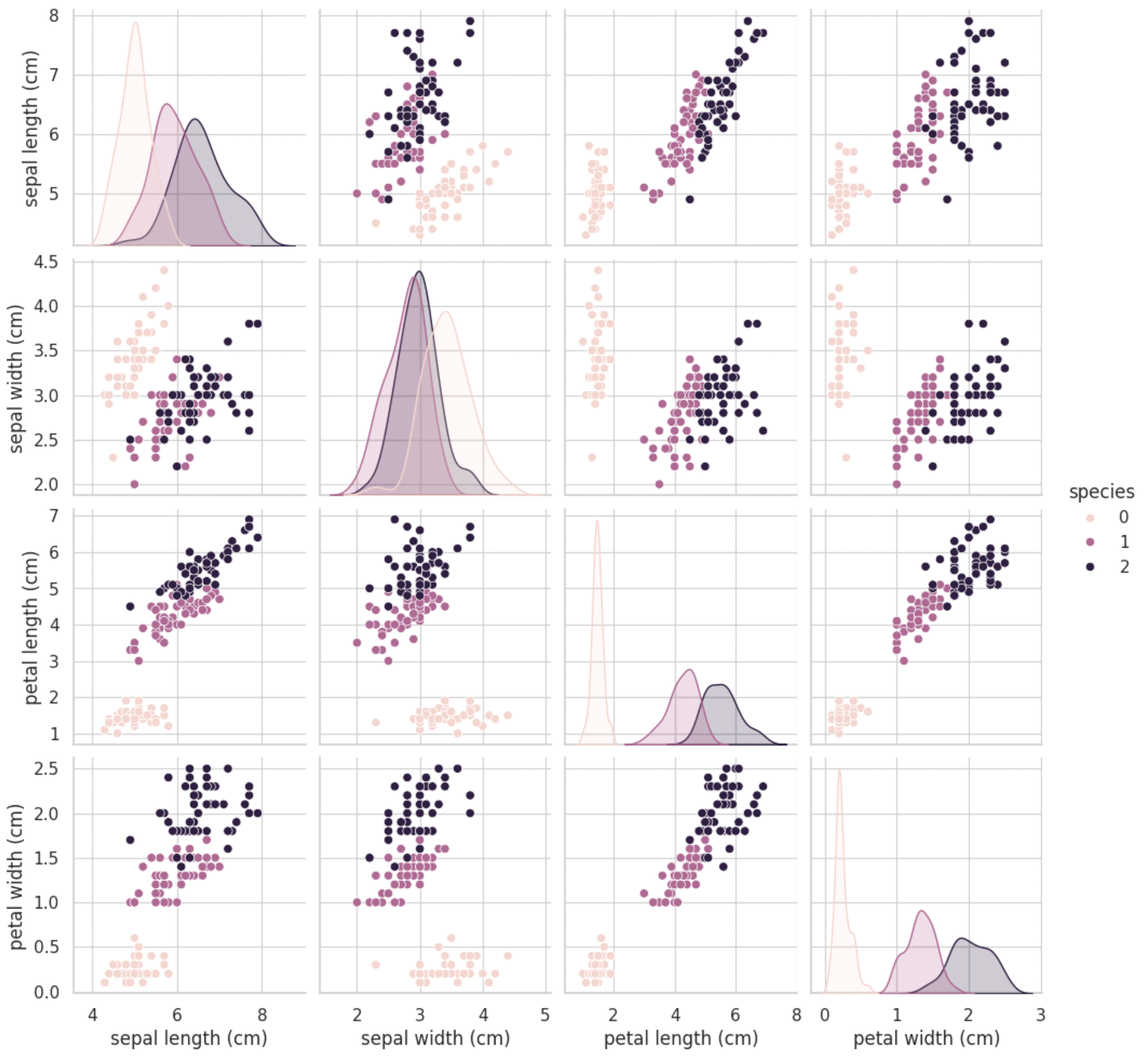
Code:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import datasets

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix


# Load the Iris dataset

iris = datasets.load_iris()

X = iris.data  # Features: Sepal length, Sepal width, Petal length, Petal width

y = iris.target  # Labels: 0 = Setosa, 1 = Versicolour, 2 = Virginica


# Convert to DataFrame for visualization

iris_df = pd.DataFrame(X, columns=iris.feature_names)
```

```
iris_df['species'] = iris.target

# Visualize pairplot
sns.pairplot(iris_df, hue='species')
plt.show()

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

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

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

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, target_names=iris.target_names)

# Display the results
print(f"Accuracy: {accuracy * 100:.2f}%")
print("\nClassification Report:\n", report)
```

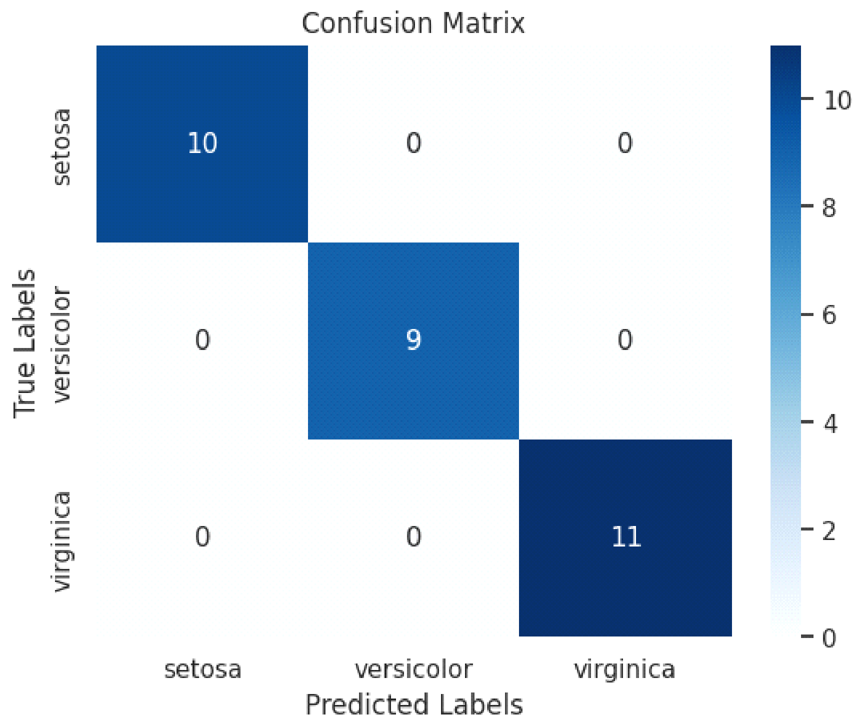
```
# Plot Confusion Matrix

cm = confusion_matrix(y_test, y_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=iris.target_names,
yticklabels=iris.target_names)

plt.title('Confusion Matrix')

plt.xlabel('Predicted Labels')

plt.ylabel('True Labels')

plt.show()
```

OUTPUT



Accuracy: 100.00%

```
Classification Report
   precision    recall  f1-score   support


    setosa       1.00      1.00      1.00        10

 versicolor       1.00      1.00      1.00         9

  virginica       1.00      1.00      1.00        11


   accuracy                          1.00        30
  macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```


Confusion Matrix
True Labels
Predicted Labels
setosa
versicolor
virginica
setosa
versicolor
virginica