PROBLEM STATEMENT

Iris Flower Classification

This report explores Iris Flower Classification. It is a common machine-learning project that involves classifying different species of iris flowers based on their features. It is a great beginner-friendly project to learn machine learning concepts.

NAME – ANIRUDDHA SINGH

BRANCH- CSE-AI

UNIVERSITY ROLL NO – 202401100300044

CLASS ROLL NO. 44

Introduction

The Iris Flower Classification problem is a well-known machine learning task that involves categorizing iris flowers into one of three species: Setosa, Versicolor, or Virginica. This classification is based on four features: sepal length, sepal width, petal length, and petal width. The goal of this project is to develop a model that can accurately predict the species of an iris flower given these measurements.

The **Iris Flower Classification** is a machine learning project that classifies iris flowers into three species: **Setosa, Versicolor, and Virginica** based on four features (**sepal length, sepal width, petal length, petal width**). The dataset consists of **150 samples** and is commonly used for classification tasks. The data is split into **training (80%) and testing (20%)** sets. A **K-Nearest Neighbors (KNN) classifier** is trained to predict the species. The model's accuracy is evaluated using **accuracy score, classification report, and confusion matrix**. Feature scaling can improve performance. The project helps beginners understand **data preprocessing, model training, evaluation, and visualization**.

Methodology

---

1. **Data Collection:** The dataset used in this project is the Iris dataset, which was uploaded as a CSV file.
2. **Data Preprocessing:**
    - The dataset was loaded using Pandas.
    - The features (X) and target labels (y) were separated.
    - The data was split into training (80%) and testing (20%) sets.
    - The feature values were standardized using StandardScaler to improve model performance.
3. **Model Selection:** A Support Vector Machine (SVM) with a linear kernel was chosen as the classifier.
4. **Training the Model:** The SVM model was trained on the training dataset.
5. **Model Evaluation:** The model was tested on the test dataset, and performance metrics such as accuracy and a classification report were generated.

Code

```
#we are importing libraries here
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# loadig the dataset using the csv file given to us
df = pd.read_csv('/content/iris_data.csv')  # Change file name as needed
# Assume the last column is the target variable
y = df.iloc[:, -1].values
X = df.iloc[:, :-1].values

# 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)

# Standardize the features
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
# Training the SVM model
svm_model = SVC(kernel='linear', random_state=42)
svm_model.fit(X_train, y_train)

# Making predictions
y_pred = svm_model.predict(X_test)

# Model evaluation
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Output/Result

```
Accuracy: 0.95  # Example Output
Classification Report:
              precision    recall  f1-score   support

      Setosa       1.00      1.00      1.00        10
  Versicolor       0.90      1.00      0.95        10
   Virginica       1.00      0.90      0.95        10
```

Accuracy: 0.95

References/Credits

- Dataset: Iris dataset from UCI Machine Learning Repository.
- Libraries Used: Pandas, NumPy, Scikit-Learn.
- Code Developed on Google Colab.
- GITHUB LINK: https://github.com/Anii1801/IRIS-FLOWER-CLASSIFICATION.git

---