Iris Classification

Internship Project Report

1. Introduction

The Iris Classification project is a classic machine learning problem that focuses on classifying iris flowers into three species based on sepal and petal measurements.

2. Objective

• Classify iris flowers into three species
• Train and evaluate classification models

3. Dataset Description

Dataset file name: **Iris.csv**. It contains 150 records with four numerical features and one categorical target.

4. Tools Used

Python, Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, VS Code

5. Data Preprocessing

Label encoding and feature scaling were applied before model training.

6. Exploratory Data Analysis

Scatter plots and count plots were used to analyze the dataset.

7. Machine Learning Models

Logistic Regression and KNN classifiers were used.

8. Results & Evaluation

Both models achieved high accuracy on the test dataset.

9. Conclusion

The Iris dataset is well-suited for classification problems.

10. Code Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

df = pd.read_csv("Iris.csv")

X = df.drop("Species", axis=1)
y = df["Species"]

le = LabelEncoder()
y = le.fit_transform(y)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

lr = LogisticRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)

print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred_lr))
print("KNN Accuracy:", accuracy_score(y_test, y_pred_knn))

cm = confusion_matrix(y_test, y_pred_knn)
sns.heatmap(cm, annot=True, cmap="Blues")
plt.show()
```