**NEXUS PROJECT - 1**

**Documentation for Iris Dataset Classification Project**

**Project Overview:**

The goal of this project is to develop a classification model for the Iris dataset using Logistic Regression. The Iris dataset is a popular dataset in machine learning and consists of measurements of 150 iris flowers from three different species: setosa,versicolor, and virginica.

**Project Structure:**

**Importing Libraries:**

Necessary libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn are imported for data manipulation, visualisation, and modelling.

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

import seaborn as sns

**Loading the Iris Dataset:**

The Iris dataset is loaded using the load\_iris function from Scikit-learn and converted into a Pandas DataFrame for easier analysis.

iris = load\_iris()

iris\_df = pd.DataFrame(data=np.c\_[iris['data'], iris['target']], columns=iris['feature\_names'] + ['target'])

**Exploratory Data Analysis (EDA):**

Basic statistics and visualisations are used to explore the characteristics of the Iris dataset.

print(iris\_df.describe())

iris\_df.hist(figsize=(10, 8))

plt.suptitle('Distribution of Iris Dataset Features')

plt.show()

sns.pairplot(iris\_df, hue='target')

plt.suptitle('Pairplot of Iris Dataset Features')

plt.show()

**Data Splitting:**

The dataset is split into training and testing sets using the train\_test\_split function.

X = iris.data

y = iris.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Model Training:**

A Logistic Regression model is chosen and trained on the training set.

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train, y\_train)

**Model Evaluation:**

The trained model is evaluated on the testing set using accuracy, precision, and recall metrics.

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='macro')

recall = recall\_score(y\_test, y\_pred, average='macro')

print(f"Accuracy: {accuracy:.4f}")

print(f"Precision: {precision:.4f}")

print(f"Recall: {recall:.4f}")

**Conclusion:**

This project successfully demonstrates the process of loading a dataset, performing exploratory data analysis, splitting the data, training a classification model (Logistic Regression), and evaluating its performance using various metrics. The model's accuracy, precision, and recall provide insights into its effectiveness in classifying Iris flowers.