Source code:

# Import necessary libraries

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

# Problem Definition

# Goal: Predict iris species based on features

# Data Collection

df = sns.load\_dataset('iris')

# Data Preprocessing

df.dropna(inplace=True) # drop missing values if any

# Exploratory Data Analysis (EDA)

sns.pairplot(df, hue='species')

plt.show()

# Feature Engineering

X = df.drop('species', axis=1)

y = df['species']

# Train-test split

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\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Model Building

model = RandomForestClassifier(random\_state=42)

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

# Model Evaluation

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

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

# Result Visualization

feature\_importances = pd.Series(model.feature\_importances\_, index=X.columns)

feature\_importances.sort\_values().plot(kind='barh')

plt.title("Feature Importances")

plt.show()