National Institute of Electronics & Information Technology

राष्ट्रीय इलेक्ट्रॉनिकी एवं सूचना प्रौद्योगिकी संस्थान

MACHINE LEARNING INTERNSHIP REPORT

IRIS FLOWER CLASSIFICATION

Submitted By:
Name: Aditya Prasad Kanhar
Program: Foundation Course on Machine Learning using Python
Institution: Utkal University
Reg. Number: 1782765

Submitted To

Organization: NIELIT,BHUBANESWAR

Acknowledgment

I take this opportunity to express my profound gratitude to UTKAL UNIVERSITY and the DEPARTMENT OF 5 YEARS OF INTEGRATED MCA for facilitating the successful completion of my internship titled "Foundation Course on Machine Learning using Python."
I am espeially thankful to HARIHAR DASH , whose expert guidance, constructive feedback, and consistent encouragement were invaluable throughout the course of this internship. Their mentorship played a critical role in enhancing both my theoretical understanding and practical application of machine learning techniques using Python.
I would also like to extend my sincere appreciation to the coordinators and faculty members associated with the internship programme for designing a structured, industry-relevant learning experience that effectively bridged the gap between academic knowledge and practical implementation.

NAME: Aditya Prasad Kanhar
REG. NO: 1782765

Certificate of Completion

This is to certify that Mr. ADITYA PRASAD KANHAR, a student of the 5 YEARS OF INTEGRATED M.C.A, UTKAL UNIVERSITY, has successfully completed the internship titled "Foundation Course on Machine Learning using Python", conducted during the period 15th May 2025 to 05th July 2025.This internship was carried out under the guidance and supervision of HARIHAR DASH, SCIENTIST-C,at NIELIT,BHUBANESWAR.We acknowledge the intern's active participation, consistent effort, and sincere contribution throughout the internship duration. The student has demonstrated a good understanding of machine learning concepts and hands-on proficiency in Python.

Signature of Guide

Table of Contents

Introduction

1.1 Problem Definition and Objective(s) :

Classification of flowers based on their physical features is an essential task in the field of botany and pattern recognition. The IRIS dataset is a classic dataset used in machine learning and statistics, containing 150 samples of iris flowers divided into three species (*Setosa*, *Versicolor*, and *Virginica*), with four numerical features:

- Sepal length (cm)
- Sepal width (cm)
- Petal length (cm)
- Petal width (cm)

The primary problem addressed in this project is: "To build a machine learning model that accurately classifies an iris flower into its respective species based on its measurable attributes."

The objectives of this project are:

- To utilize a Random Forest Classifier for effective multiclass classification.
- To develop an interactive web application using Streamlit for live predictions and easy usage.
- To visualize the dataset, predictions, and feature importances to enhance user understanding.
- To bridge the gap between machine learning model development and practical deployment for real-time applications.

1.2 Motivation(s) :

Several motivations inspired the implementation of this project:

- Educational Clarity: The IRIS dataset is often the first dataset introduced to students of machine learning due to its simplicity and well-labeled data, making it suitable for building intuition about classification problems.
- Practical Deployment Skills: While many learn to train models in Jupyter notebooks, deploying them interactively in a web interface like Streamlit allows for instant testing, increasing practical understanding.
- Visualization and Explainability: Visualizing feature importance helps understand how different attributes contribute to the classification decision, aligning with the goal of explainable AI (XAI).
- Foundation for Advanced Projects: Successfully implementing and deploying a simple yet complete ML pipeline forms a solid foundation for handling complex datasets and projects in the future.

1.3 Project Overview / Specifications :

In this project:

- The IRIS dataset from scikit-learn is loaded, and a Random Forest Classifier is trained to predict the flower species based on the four features.

- Streamlit is used to create an interactive web application where:
    - Users can explore the dataset.
    - Users can provide custom feature values using sliders.
    - The application displays predicted species and prediction probabilities instantly.
    - A bar plot of feature importances is displayed to understand which features influence the classification most.

This workflow demonstrates a complete machine learning pipeline from data preprocessing, model training, and testing, to deployment in a live application for practical demonstration.

1.4 Hardware Specification :

To run this project, the following hardware specifications are sufficient:

- Processor: Intel i3/i5/i7 or AMD equivalent
- RAM: 4 GB minimum (8 GB recommended for smoother experience)
- Disk: At least 500 MB of free space
- Internet connection (optional, required for installing dependencies)

1.5 Software Specification :

- Operating System: Windows 10/Linux/macOS
- Python 3.11+
- Required Libraries:
    - pandas
    - matplotlib
    - seaborn
    - scikit-learn
    - streamlit

1.6 Organisation of the Project :

- Chapter 1: Introduction
- Chapter 2: Literature Survey
- Chapter 3: Python Streamlit Codes and Project Output
- Chapter 4: Conclusions

Literature Survey

2.1 Existing System :

The IRIS dataset, introduced by Ronald A. Fisher in 1936, is one of the most widely used datasets for demonstrating multiclass classification in machine learning. Traditionally, it has been used to:

- Implement and understand algorithms such as K-Nearest Neighbors (KNN), Decision Trees, Naive Bayes, Logistic Regression, and Support Vector Machines (SVM).
- Evaluate accuracy, confusion matrices, and precision-recall metrics on well-labeled, clean datasets.
- Perform data visualization using scatter plots and pair plots to observe feature separability.

However, these systems are typically:

- Implemented in Jupyter Notebooks or static Python scripts, requiring the user to have programming knowledge to execute and modify the predictions.
- Focused only on model training and evaluation without practical deployment interfaces for live input and interaction.
- Limited in helping non-technical users understand the practical utility of ML models.

2.2 Proposed System :

The proposed system improves upon the existing systems in several ways:

- User-Friendly Deployment: By using Streamlit, the system transforms a static classification problem into an interactive web application where users can input values using sliders without writing code.
- Live Prediction: Users can instantly receive predictions for flower species and the associated probabilities, providing an engaging learning environment.
- Visualization: Feature importance from the Random Forest Classifier is displayed using Seaborn bar plots, helping users understand how each attribute affects the prediction.
- Self-Contained Application: All functionalities, including dataset exploration, input, prediction, and feature analysis, are included in a single interface, making it practical for students, researchers, and hobbyists.
- Effective Model: Using Random Forest improves prediction stability and accuracy, handling multiclass classification efficiently while providing interpretability via feature importance.

Thus, the proposed system addresses the limitations of static script-based implementations and bridges the gap between learning machine learning and deploying it practically for real-world-like testing and experimentation.

2.3 Feasibility Study :

2.3.1 Technical Feasibility

- The project uses Python and popular open-source libraries, making it easily implementable on standard systems without specialized hardware.
- Streamlit enables rapid web deployment without the need for advanced backend or frontend frameworks, making it technically simple yet effective.

2.3.2 Operational Feasibility

- The interface uses sliders, checkboxes, and clear outputs, requiring no technical background to operate.
- Users can immediately see how changes in input features impact the predicted species and understand the classifier's behavior.

2.3.3 Economic Feasibility

- The project incurs no additional financial costs aside from hardware already available with the user.
- All libraries used (streamlit, scikit-learn, pandas, matplotlib, seaborn) are free and open-source, ensuring zero software cost for development and deployment.

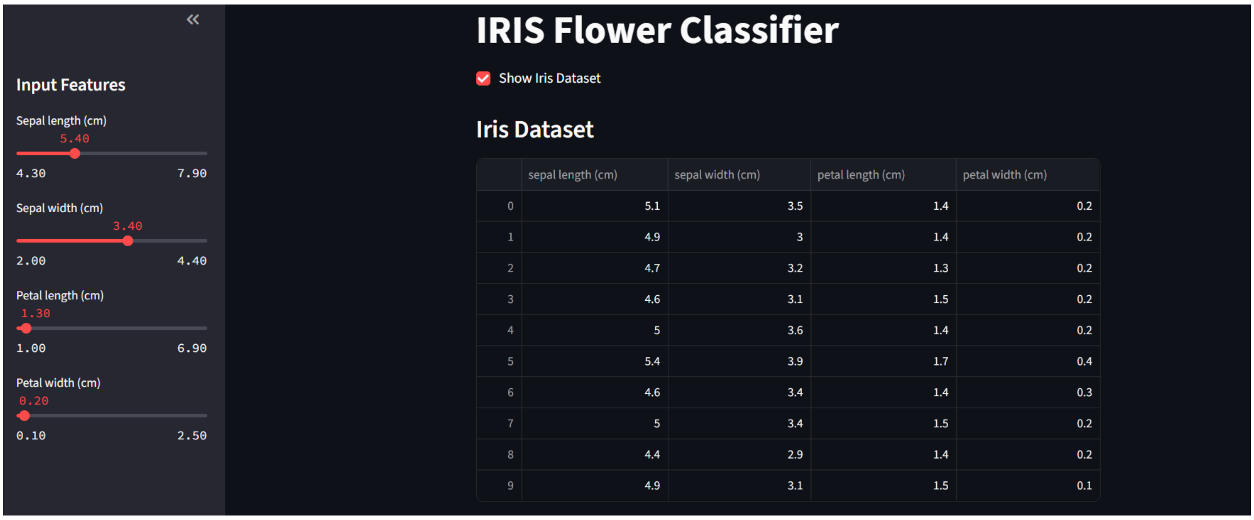
Thus, the proposed system is technically sound, operationally practical, and economically viable for educational deployment, student learning, and showcasing practical machine learning application skills.

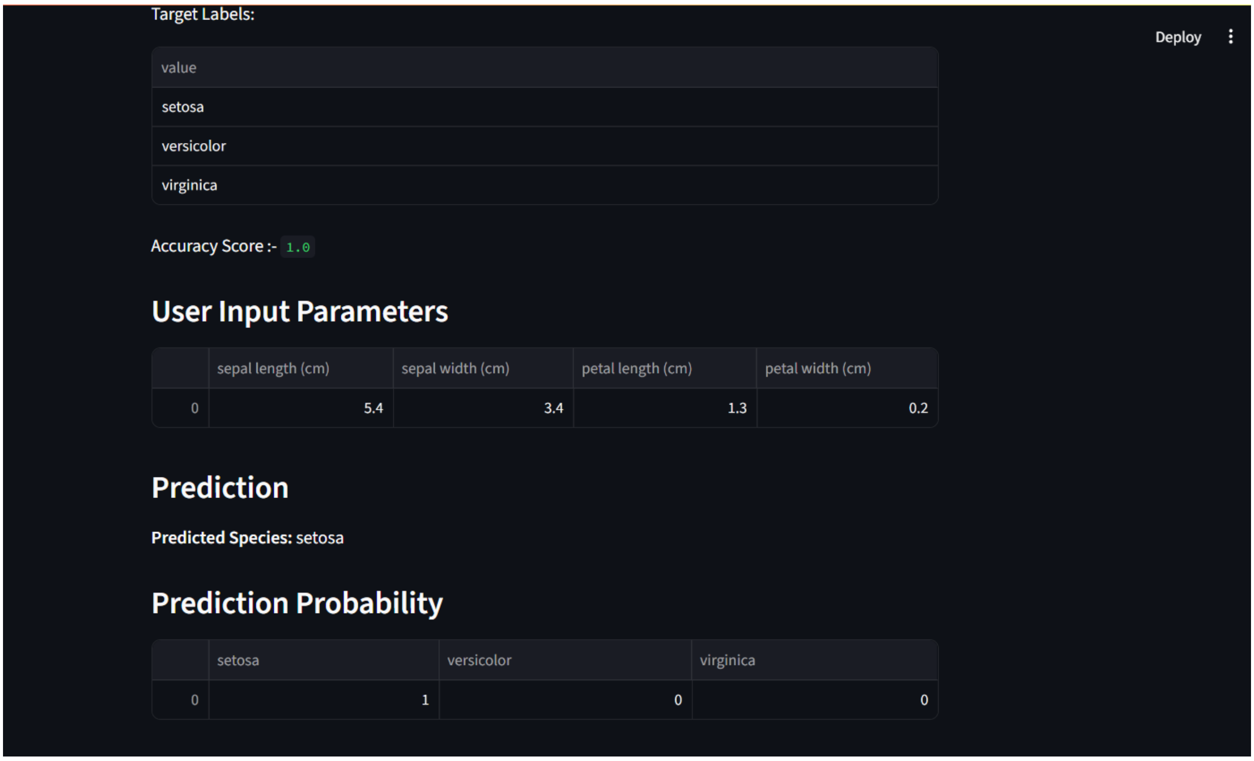
Python Streamlit Codes

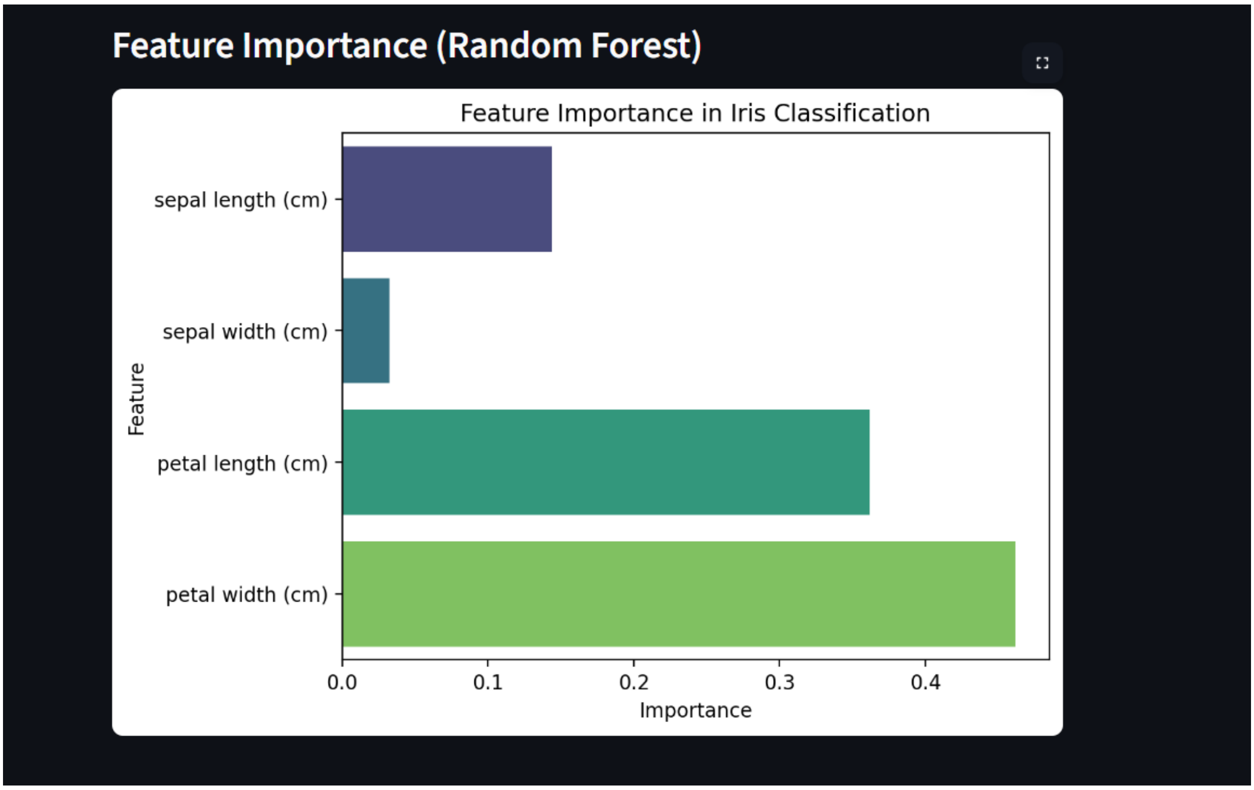
```
import streamlit as st
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier

st.title("IRIS Flower Classifier")

iris = load_iris()
x = pd.DataFrame(iris.data, columns=iris.feature_names)
y = iris.target

if st.checkbox("Show Iris Dataset"):
    st.subheader("Iris Dataset")
    st.write(x.head(10))
    st.write("Target Labels:", iris.target_names)

st.sidebar.header("Input Features")

def user_input_features():
    sepal_length = st.sidebar.slider('Sepal length (cm)', float(x['sepal length (cm)'].min()), float(x['sepal length (cm)'].max()), 5.4)
    sepal_width = st.sidebar.slider('Sepal width (cm)', float(x['sepal width (cm)'].min()), float(x['sepal width (cm)'].max()), 3.4)
    petal_length = st.sidebar.slider('Petal length (cm)', float(x['petal length (cm)'].min()), float(x['petal length (cm)'].max()), 1.3)
    petal_width = st.sidebar.slider('Petal width (cm)', float(x['petal width (cm)'].min()), float(x['petal width (cm)'].max()), 0.2)

    data = {
        'sepal length (cm)': sepal_length,
        'sepal width (cm)': sepal_width,
        'petal length (cm)': petal_length,
        'petal width (cm)': petal_width
    }
    features = pd.DataFrame(data, index=[0])
    return features

input_df = user_input_features()

model = RandomForestClassifier()
model.fit(x, y)

prediction = model.predict(input_df)
```

```
prediction_proba = model.predict_proba(input_df)

st.write("Accuracy Score :- " , model.score(x,y))

st.subheader("User Input Parameters")
st.write(input_df)

st.subheader("Prediction")
st.write(f"**Predicted Species:** {iris.target_names[prediction][0]}")

st.subheader("Prediction Probability")
proba_df = pd.DataFrame(prediction_proba, columns=iris.target_names)
st.write(proba_df)

st.subheader("Feature Importance (Random Forest)")
fig1, ax1 = plt.subplots()
feature_importances = model.feature_importances_
sns.barplot(x=feature_importances, y=iris.feature_names, palette="viridis", ax=ax1)
ax1.set_xlabel("Importance")
ax1.set_ylabel("Feature")
ax1.set_title("Feature Importance in Iris Classification")
st.pyplot(fig1)
```

Project Output






Feature Importance (Random Forest)
Feature Importance in Iris Classification
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
Feature
0.0
0.1
0.2
0.3
0.4
Importance

Conclusions

The IRIS Flower Classification Project using Streamlit and Random Forest demonstrates a complete machine learning pipeline, transitioning from traditional model building to practical, user-friendly deployment.

Through this project, the following key outcomes were achieved:

- Successful Model Implementation: A Random Forest Classifier was trained on the IRIS dataset, achieving high accuracy in classifying iris flowers into *Setosa, Versicolor,* and *Virginica* based on sepal and petal dimensions.

- Interactive Web Deployment: Using Streamlit, the model was deployed in an interactive environment allowing users to:

    - Explore and understand the IRIS dataset.
    - Input feature values using intuitive sliders.
    - Instantly receive species predictions and associated probabilities.
    - Visualize feature importances to gain insight into which features most influence classification.

- Enhanced Learning and Practical Skills: The project bridges the gap between theoretical machine learning concepts and practical application development, helping learners understand:

    - Dataset handling
    - Model training and evaluation
    - Deployment of ML models in a web application without complex frameworks

- Explainability: Feature importance visualization and immediate feedback help users understand and trust the model's predictions, aligning with explainable AI (XAI) practices.

In summary, this project not only provided practical experience in machine learning model development but also demonstrated the process of deploying a live, user-friendly application, enhancing accessibility to machine learning predictions for users without programming knowledge.

References

- **Scikit-Learn Documentation** – For understanding Random Forest and IRIS dataset utilities. https://scikit-learn.org/stable/documentation.html
- W3Schools. https://www.w3schools.com
- Streamlit Documentation. https://streamlit.io
- Python Official Documentation. https://docs.python.org
- Rajender Kumar. (2020). *Python Machine Learning*. IK International Publishing House Pvt. Ltd..