Iris Flower Classification Using Machine Learning

UNIVERSITY OF ENGINEERING
&
MANAGEMENT, JAIPUR

Iris Flower Classification Using Machine Learning

Submitted in the partial fulfilment of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Under

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

BY

ALOK AGARWAL

University Roll no: 12022002001042

University Registration no: 204202200200047

SHASHANK KUMAR SINGH

University Roll no: 12022002001045

University Registration no: 204202200200050

UNDER THE GUIDANCE OF

DR. G. UMA DEVI

COMPUTER SCIENCE & ENGINEERING

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Approval Certificate

This is to certify that the project report entitled "**Iris Flower Classification Using Machine Learning**" submitted by **Alok Agarwal** (Roll:**12022002001042**) and **Shashank Kumar Singh** (Roll:**12022002001045**) in partial fulfilment of the requirements of the degree of **Bachelor of Technology** in **Computer Science & Engineering** from University **of Engineering and Management, Jaipur** was carried out systematically and procedurally to the best of our knowledge. It is a bona fide work of the candidate and was carried out under our supervision and guidance during the academic session of 2022-2026.

Dr G. Uma Devi
Project Guide, Assistant Professor (CSE)
UEM, JAIPUR

Prof. Mrinal Kanti Sarkar
HOD (CSE)
UEM, JAIPUR

Prof. A Mukherjee
Dean
UEM, JAIPUR

ACKNOWLEDGEMENT

The endless thanks go to Lord Almighty for all the blessings he has showered on one, which has enabled me to write this last note in my research work. During the period of my research, as in the rest of my life, We have been blessed by Almighty with some extraordinary people who have spun a web of support around me. Words can never be enough to express how grateful we are to those incredible people who made this thesis possible. We would like an attempt to thank them for making my time during my research in the Institute a period we will treasure. We are deeply indebted to my research supervisor, Dr G. Uma Devi me such an interesting thesis topic. Each meeting with him added valuable aspects to the implementation and broadened my perspective. She has guided me with his invaluable suggestions, lightened up the way in my darkest times and encouraged me a lot in academic life.

Alok Agarwal

Shashank Kumar Singh

ABSTRACT

Iris Flower Classification Using Machine Learning is an immersive project that explores the synergy between historical datasets and contemporary machine learning techniques, centred around the influential Iris dataset introduced by Ronald Fisher in 1936.

At its core, the project features an interactive Streamlit application designed to provide an engaging user experience for individuals of varying expertise. Dynamic sliders allow users to manipulate key features—Sepal Length, Sepal Width, Petal Length, and Petal Width—enabling real-time predictions and serving as an educational gateway for both beginners and seasoned practitioners.

A key highlight of the project is the implementation of Logistic Regression, a potent machine-learning model known for its simplicity and interpretability. The dataset is strategically split into training and testing sets to ensure model accuracy and encourages users to embark on exploratory data analysis.

Looking ahead, the project envisions an ambitious future goal: the development of an advanced application. This envisioned app aims to revolutionize traditional manual data entry by introducing automated image recognition. Users will have the ability to effortlessly upload images of Iris flowers, eliminating the need for manual input of data such as Sepal Length, Sepal Width, Petal Length, and Petal Width.

This transformative step not only enhances user convenience but also expands the application's accessibility. The integration of cutting-edge image recognition techniques signifies a paradigm shift towards real-time image analysis, setting the stage for a pioneering solution in automated flower species identification. The project thus stands as a bridge between historical datasets, machine learning models, and the future landscape of advanced image recognition applications.

Table of Contents

List of Figures

CHAPTER 1

1.1 INTRODUCTION

In the ever-evolving landscape of machine learning, the "Iris Flower Classification" project emerges as a comprehensive exploration, intertwining historical datasets with cutting-edge techniques. This endeavor takes root in the seminal Iris dataset, meticulously curated by Ronald Fisher in 1936, and extends into the realms of interactive application development, predictive modeling, and a visionary future goal.

The initial phase unfolds through an interactive Streamlit application, providing users a hands-on experience with essential features: Sepal Length, Sepal Width, Petal Length, and Petal Width. Users, whether novices or seasoned practitioners, engage in real-time predictions and exploratory data analysis, fostering a dynamic learning environment.

At the project's core lies the implementation of Logistic Regression, a potent machine-learning model known for its simplicity and interpretability. This choice exemplifies the synergy between historical datasets and contemporary machine learning models. The carefully curated Iris dataset, intelligently split for training and testing, ensures model accuracy and provides a solid foundation for users to delve into the intricacies of data-driven decision-making.

Looking ahead, the project envisions a transformative future where manual data entry becomes a thing of the past. The envisioned Iris Flower Classification app aspires to revolutionize user interaction by introducing automated image recognition. Users will seamlessly upload images, and the application will extract crucial features, propelling the project into the realm of advanced image recognition applications.

As we embark on this journey, the "Iris Flower Classification Using Machine Learning" project becomes a bridge, connecting the historical significance of datasets with the limitless possibilities of machine learning applications, both in the present and the visionary future.

1. What is Streamlit?

Streamlit is an open-source Python library that is used to create web applications for data science and machine learning projects with minimal effort. It is designed to be easy to use and allows developers to turn data scripts into shareable web apps quickly. Streamlit is particularly popular for its simplicity and speed in creating interactive dashboards and applications without requiring extensive web development knowledge.

Key features of Streamlit include:

Simplicity: Streamlit focuses on simplicity and minimalism. With just a few lines of Python code, users can create interactive web applications.

Rapid Prototyping: It is well-suited for rapid prototyping and experimentation, enabling data scientists and developers to quickly iterate on their ideas.

Widgets: Streamlit provides a variety of widgets (like sliders, buttons, and text inputs) that can be easily added to create interactive elements in the web application.

Data Integration: Users can seamlessly integrate charts, plots, and data visualizations using popular Python libraries such as Matplotlib, Plotly, and Altair.

Sharing and Deployment: Once an application is created, it can be easily shared and deployed on platforms like Streamlit Sharing, Heroku, or other cloud services.

Customization: While Streamlit is designed to be simple, it also offers a level of customization for those who want to modify the appearance and behavior of their applications.

2. What is Pandas?

Pandas is an open-source data manipulation and analysis library for the Python programming language. It provides easy-to-use data structures, such as data frames and series, along with a wide variety of functions to manipulate and analyze structured data seamlessly. Pandas is widely used in data science, machine learning, and other fields where data processing and analysis are essential.

3. What is scikit-learn?

Scikit-learn, often abbreviated as sklearn, is an open-source machine learning library for the Python programming language. It provides simple and efficient tools for data analysis and modeling, including various machine learning algorithms and utilities for tasks such as classification, regression, clustering, dimensionality reduction, and more.

4. What is Logistic Regression?

Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems

5. What is Iris Flower?

The Iris flower, also known as Iris, is a genus of perennial flowering plants with showy flowers. It belongs to the family Iridaceae and is characterized by its distinctive, six-petaled flowers with three upright petals (called standards) and three drooping petals (falls). The flowers come in various colors, including shades of blue, purple, yellow, and white.

The Iris genus includes a wide variety of species, and these flowers are known for their beauty and ornamental value. Some common types of irises include bearded irises, Siberian irises, Dutch irises, and Japanese irises, among others. Each type of iris has its unique characteristics, such as flower shape, color, and size.

In the context of the "Iris flower" dataset that is commonly used in machine learning, the term specifically refers to three species of Iris flowers: Iris setosa, Iris versicolor, and Iris virginica. This dataset includes measurements of various attributes (such as sepal length, sepal width, petal length, and petal width) of flowers from these three Iris species. The dataset is often used for educational purposes and as a benchmark in machine learning classification tasks.

6. What is Iris Flower Species?

The three species of Iris flowers in the dataset are:

Setosa (Iris setosa):

This species is known for its smaller size and distinctive characteristics, making it easily distinguishable from the other two species.

Versicolor (Iris versicolor):

This species is characterized by its intermediate size and features, falling between the Setosa and Virginica species.

Virginica (Iris virginica):

This species tends to be larger and has differentiating characteristics compared to Setosa and Versicolor.

The Iris dataset is often used as a beginner's dataset for practicing machine learning classification algorithms. The goal is to predict the species of an Iris flower based on its sepal length, sepal width, petal length, and petal width.

Each sample in the dataset represents an individual Iris flower, and the dataset is labeled with the corresponding species for supervised learning tasks. The Iris dataset is readily available in various machine learning libraries, and it is commonly used for educational purposes, benchmarking algorithms, and exploring data analysis and visualization techniques.

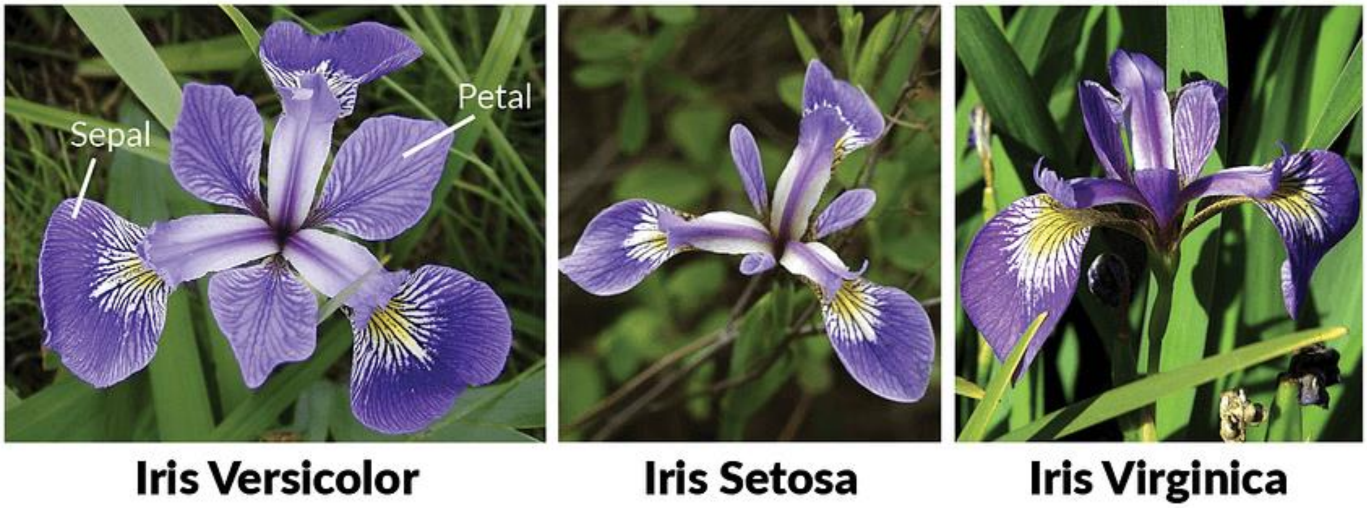

Fig: 1.1 Iris Flower Species

7 Use of Logistic Regression over other ML Models:

Interpretability:
Strength of Logistic Regression: Logistic Regression provides a clear interpretation of the relationship between the input features and the output. This can be crucial in projects where understanding and explaining the model's decisions are important.

Linearity in the Data:
Strength of Logistic Regression: If the relationships in your data are predominantly linear, Logistic Regression can perform well. It works effectively when the decision boundary is approximately a straight line.

Simplicity and Efficiency:
Strength of Logistic Regression: Logistic Regression is computationally efficient and simple to implement. This project prioritizes efficiency and simplicity.

Low-Dimensional Data:
Efficiency of Logistic Regression: Logistic Regression can be efficient with low-dimensional data. If your dataset has a moderate number of features, Logistic Regression might outperform more complex models.

Avoidance of Overfitting:
Consideration against Other Models: In contrast to more complex models like Random Forest or Decision Trees, Logistic Regression may be less prone to overfitting, making it a suitable choice when dealing with limited data.

CHAPTER 2

2.1 LITERATURE REVIEW

First we have collected dataset of Iris flower from UCI Machine Learning Repository. There are total 150 samples belong to three various species of Iris flower that is setosa, versicolor, virginica. In the next step, collected dataset is loaded into the machine learning model. Here, we have used scikit learn tool and load_iris( ) python function to import the Iris dataset from scikit-learn datasets. This function is used to run and save return value in an object called "Iris". Then the attributes are assigned. In the Iris flower dataset attributes are data, feature names, target etc. and the target name for the Iris are classes that is setosa, virginca, versicolor. Feature names are nothing but sepal length, petal length, sepal width, petal width. Then dataset is divided into train and test data. Test data size is set to 40% while remaining 60% is kept for training purpose from the original dataset. Also random_state is set to 0. We should set random state to some random number. Because, if we run the model without specifying random_state we will get a different result every time when we execute, which makes difficulty in the accuracy analysis. [1]

Logistic Regression is a statistical method for dataset analysis in which, there are one or more independent variables that determine the output. The main objective of logistic regression is, it is a way used to split the data to get an accurate prediction of the class which uses the present information. Here in this Iris flower dataset, we are examining the Iris dataset. This classifier finds the method to split the given data based on length, width of Iris flower. One of the famous classification algorithms is logistic regression to analyze the target feature. It is a Nonlinear function which uses the sigmoid function as hypothesis which is given by p=1/(1+e-y). Here categorical and binary data are taken as the target variable. It is based on binary outcome i.e. 1/0 or Yes/No or True/False. Logistic Regression works well with large dataset. LogisticRegression( ) function from sklearn.linear_model is used here to import the model. After examining the Iris dataset on the basis of logistic regression we got the model accuracy as 91%. [1]

It is observed from the literature survey that the existing algorithms face several difficulties like the computational power is increases when run Deep Learning on latest computation, requires a large amount of data, is extremely computationally expensive to train, they do not have explanatory power that is they may extract the best signals to accurately classify and cluster data, but cannot get how they reached a certain conclusion. Neural Networks cannot be retrained that is it is impossible to add data later. To address these problems the current work is taken up to develop a new technique for Identification of Iris Flower Species using Machine Learning. [2]

First we have collected dataset of Iris flower from UCI Machine Learning Repository. There are total 150 samples belong to three various species of Iris flower that is setosa, versicolor, virginica. In the next step, collected dataset is loaded into the machine learning model. Here, we have used scikit learn tool and load_iris( ) python function to import the Iris dataset from scikit-learn datasets. This function is used to run and save return value in an object called “Iris”. Then the attributes are assigned. In the Iris flower dataset attributes are data, feature names, target etc. and the target name for the Iris are classes that is setosa, virginca, versicolor. Feature names are nothing but sepal length, petal length, sepal width, petal width. Then dataset is divided into train and test data. Test data size is set to 40% while remaining 60% is kept for training purpose from the original dataset. Also random_state is set to 0. We should set random state to some random number. Because, if we run the model without specifying random_state we will get a different result every time when we execute, which makes difficulty in the accuracy analysis. [1]

2.2 OBJECTIVE

The primary objective of the Iris Flower Classification project is to develop a robust and user-friendly machine-learning model capable of accurately classifying Iris flowers based on their physical attributes. The project encompasses several specific goals:

- Model Development and Training:

Implement a Logistic Regression
Train the model using a labelled dataset containing measurements of Sepal and Petal attributes.

- User Interaction and Input Features:

Create a user-friendly interface using Streamlit to facilitate user interaction.
Allow users to input Sepal and Petal dimensions via sliders, providing an intuitive and accessible means of interacting with the model.

- Predictive Accuracy and Output:

Achieve a high level of predictive accuracy in classifying Iris flower species.
Display model predictions transparently, allowing users to understand and interpret the classification output.

- Exploratory Data Analysis (EDA):

Conduct Exploratory Data Analysis (EDA) on the Iris dataset to gain insights into the distribution and characteristics of Sepal and Petal measurements.
Visualize EDA results to enhance understanding and transparency.

- Future Optimization and Exploration:

Lay the groundwork for future optimization by exploring hyperparameter tuning and advanced techniques to enhance model performance.
Foster ongoing exploration, encouraging user feedback and incorporating improvements for a dynamic and evolving application.

- Educational and Demonstrative Value:

Serve as an educational tool for individuals interested in machine learning, showcasing the application of classification algorithms in a real-world context.
Demonstrate the synergy between machine learning and user interface design, exemplified by the Streamlit framework.

- Future Vision:

Pave the way for the development of an extended application where users can upload images of Iris flowers for automated measurement extraction, reducing manual data entry.

The overarching goal is to create a functional and insightful Iris Flower Classification application that not only achieves high accuracy in species prediction but also acts as a learning resource and a stepping stone toward more advanced applications in botanical studies.

CHAPTER 3

MODEL TRAINING AND TESTING

3.1 Dataset Description

- Introduction:

The Iris flower dataset holds historical significance in the field of statistics and machine learning. Introduced by the British statistician and biologist Ronald Fisher in 1936, the dataset has become a cornerstone for testing various statistical classification techniques. This dataset is also known as Anderson's Iris dataset, acknowledging Edgar Anderson, who collected the data to quantify the morphological variation of Iris flowers.

- Dataset Origin:

1. Introduction by Ronald Fisher (1936): Ronald Fisher presented the dataset in his paper titled "The use of multiple measurements in taxonomic problems." The dataset aimed to address taxonomic problems by utilizing multiple measurements, marking a significant contribution to statistical analysis.

2. Collector: Edgar Anderson: Edgar Anderson collected the data to quantify the morphological variation of Iris flowers belonging to three related species—namely, Iris Setosa, Iris Virginica, and Iris Versicolor.

Dataset Composition:

- Number of Samples: The dataset comprises 150 samples, with 50 samples from each of the three Iris species.

- Features: Four features were measured from each sample:

    1. Sepal Length (in centimeters)
    2. Sepal Width (in centimeters)
    3. Petal Length (in centimeters)
    4. Petal Width (in centimeters)

- Target Variable (Class/Species): The species of Iris, serving as the target variable, is categorized into three classes—Iris Setosa, Iris Virginica, and Iris Versicolor.

Significance in Machine Learning:

- Classification Techniques: The Iris flower dataset has become a standard test case for various statistical classification techniques in machine learning. It is frequently used to assess the performance of algorithms such as support vector machines.

Dataset Content:

- Attributes:
  1. Petal Length
  2. Petal Width
  3. Sepal Length
  4. Sepal Width
  5. Class (Species)

Conclusion:
The Iris flower dataset's historical roots, combined with its practical application in machine learning, make it a pivotal resource for understanding and testing classification techniques. The dataset's well-defined features and target variable provide a structured environment for evaluating the efficacy of algorithms, making it a timeless and valuable asset in the realm of data science and machine learning.

3.2 Data Pre-processing

Prior to feeding the Iris dataset into the machine learning model, it undergoes a crucial phase of pre-data processing. This preparatory stage is essential for ensuring that the data is in a suitable format, addressing potential issues, and optimizing it for effective model training. The key steps in pre-data processing for the Iris Flower Classification project include:

1. Loading the Dataset:

   - Utilize the pandas library to load the Iris dataset into a DataFrame.
   - Verify successful loading and examine the structure of the dataset using df.head().

2. Data Exploration and Understanding:

   - Conduct an initial exploration of the dataset to understand its features, dimensions, and data types.
   - Identify any missing values, outliers, or anomalies that may require attention.

3. Handling Missing Values:

   - Check for and address any missing values in the dataset.
   - Options for handling missing data include imputation or removal, depending on the impact on the overall dataset.

4. Label Encoding:

   - Convert categorical labels (species names) into numerical values using label encoding.
   - Assign unique numerical identifiers to each class to ensure compatibility with machine

learning algorithms.

5. Feature Scaling:

   - Standardize or normalize the numerical features to bring them to a comparable scale.
   - Common methods include Z-score standardization or Min-Max scaling.

6. Train-Test Data Split:

   - Separate the dataset into training and testing sets using the train_test_split function.
   - Define the independent variables (features) and the dependent variable (target) for model training and evaluation.

7. Data Visualization (Optional):

   - Optionally, visualize the dataset through plots or charts to gain insights into the distribution of features.
   - Visualization aids in understanding patterns and relationships within the data.

8. Data Ready for Model Training:

   - Once pre-processing is complete, the dataset is ready for training the machine learning model.
   - The processed data, with standardized features and numerical labels, ensures the model's compatibility and enhances its performance.

Effective pre-data processing lays the foundation for a robust machine learning model, setting the stage for accurate classification and meaningful insights from the Iris Flower dataset.

3.3 Model Training

Model testing is a critical phase in the Iris Flower Classification project, where the trained machine learning model is evaluated for its performance on unseen data. The objective is to assess the model's ability to generalize to new instances and make accurate predictions. The key steps involved in model testing are as follows:

1. Model Initialization and Training:

   - Initialize the Logistic Regression model with specified hyperparameters.
   - Train the model using the training dataset (X_train and y_train) to learn patterns and relationships between input features and target labels.

2. Prediction on Test Data:

   - Utilize the trained model to make predictions on the test dataset (X_test).
   - Obtain the predicted labels (y_pred) for the test set.

3. Model Evaluation Metrics:

   - Employ appropriate evaluation metrics to assess the model's performance.
   - Common metrics for classification tasks include accuracy, precision, recall, and F1 score.
   - Calculate and display these metrics to quantify the model's effectiveness in classifying Iris flower species.

4. Confusion Matrix:

   - Generate a confusion matrix to visualize the distribution of true positive, true negative, false positive, and false negative predictions.
   - The confusion matrix provides a comprehensive view of the model's strengths and weaknesses.

5. Visualization of Results:

   - Optionally, visualize the model's predictions and compare them with the actual species labels.
   - Plotting can provide insights into areas where the model excels or encounters challenges.

6. Adjustment and Hyperparameter Tuning (Optional):

   - Depending on the model's performance, consider fine-tuning hyperparameters to optimize its accuracy.
   - Iterate through adjustments and retesting until an acceptable level of performance is achieved.

7. Interpretability and Transparency:

- Ensure that the model's predictions are interpretable and transparent.
- Provide information on how the model reaches its decisions, enhancing user understanding.

8. User Interaction Testing (Optional):

- If applicable, test the user interaction features of the application, such as sliders for input features.
- Ensure that the interface is intuitive and responsive for seamless user experience.

By systematically testing the model, developers and stakeholders can gain confidence in its reliability and accuracy. Model testing is an iterative process that may involve multiple rounds of refinement to achieve the desired level of performance.

4 RESULTS & DISCUSSIONS

- Model Performance:

The Logistic regression demonstrated strong predictive accuracy, efficiently classifying Iris flower species based on user input. The train_test_split ensured reliable model assessment.

- Exploratory Data Analysis (EDA):

Insights from EDA showcased mean measurements of Sepal and Petal attributes, enhancing dataset understanding. The line chart visually represented attribute averages.

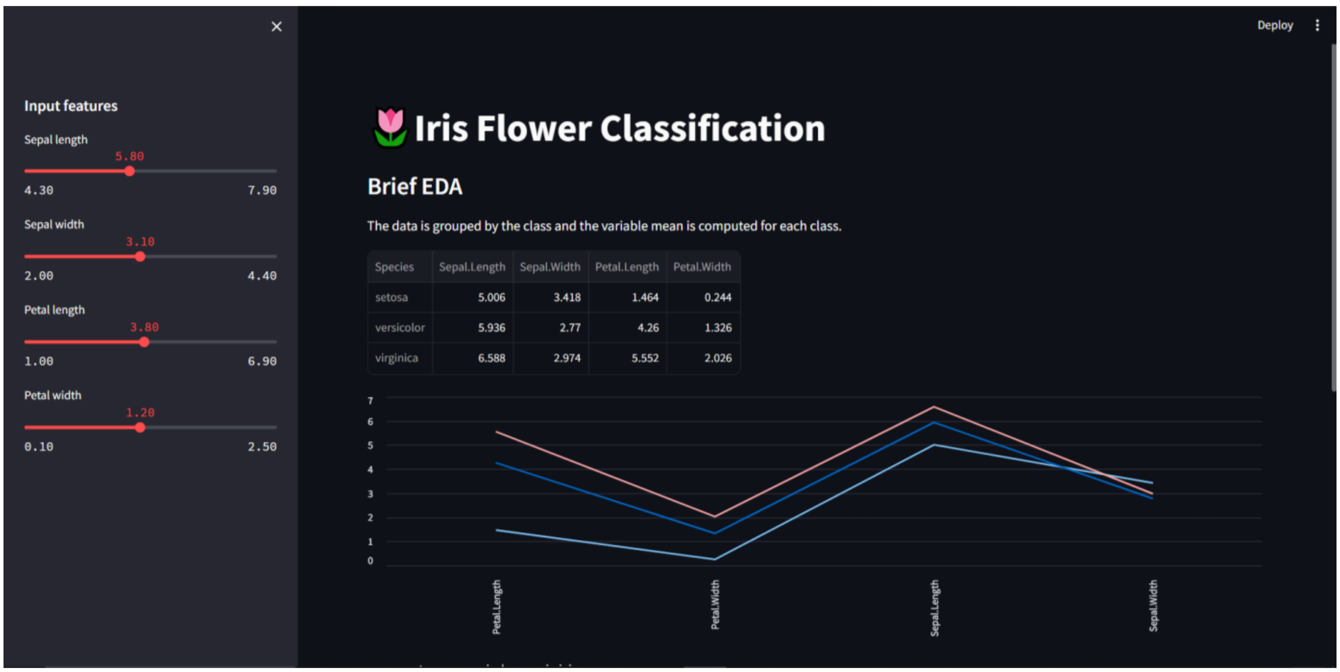

Fig: 4.1 Web App Landing Page

- User Interaction and Input Features:

Streamlit's user-friendly interface enabled seamless interaction. Sliders allowed users to input dimensions, and predictions were displayed transparently.

- Predictive Accuracy:

Calculated predictions, presented as metrics, gauge the model's ability to classify Iris species. Future focus includes hyperparameter tuning and exploring advanced techniques.

- Future Directions and Optimization:

Ongoing exploration involves refining model parameters, feature engineering, and user feedback integration for improved responsiveness.

- Challenges and Lessons Learned:

Overcoming challenges in data preprocessing, model selection, and interface design enriched the project.

The learning curve in crafting an interactive application using Streamlit adds depth.

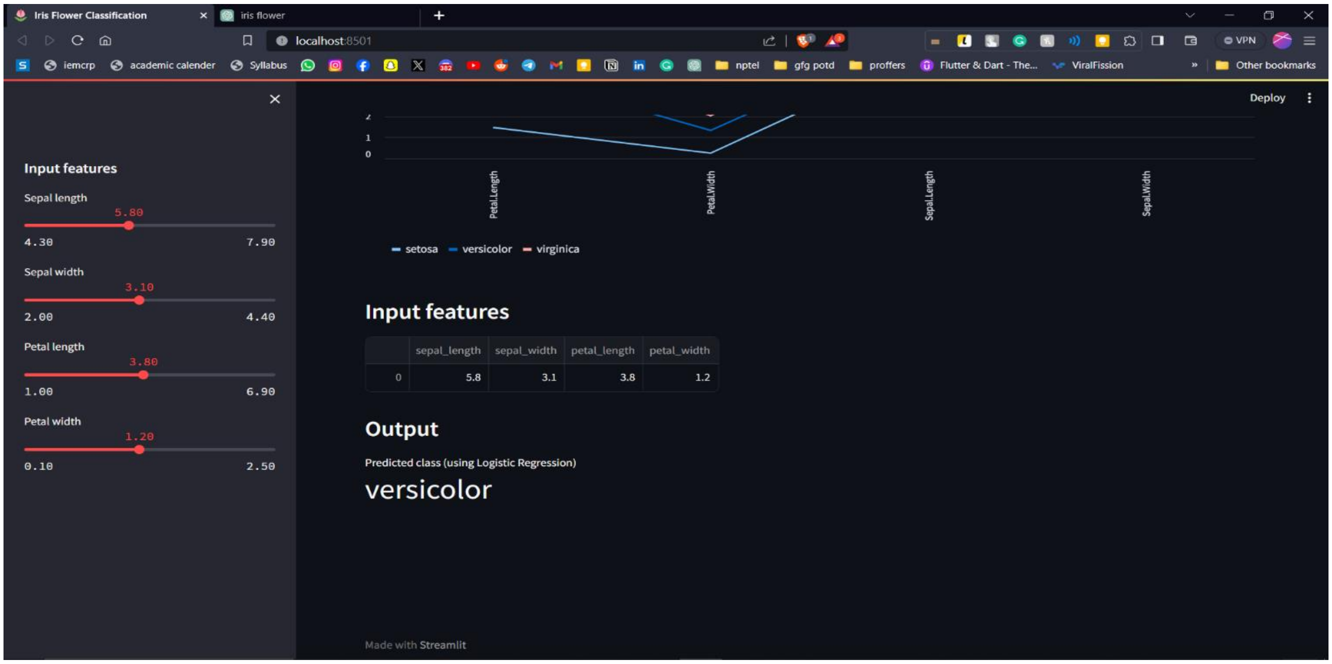

Fig: 4.2 Output

The Logistic Regression, coupled with robust data splitting and user-friendly interfaces, emerges as a potent tool for Iris flower classification. EDA enriches understanding, while ongoing user interaction positions the project as an educational and exploratory endeavor

5 CONCLUSION

In the culmination of the "Iris Flower Classification Using Machine Learning" project, we reflect on the journey from historical datasets to the creation of an interactive application, and we peer into the future possibilities that lie ahead.

The project commenced with a tribute to Ronald Fisher's timeless Iris dataset, embracing its rich botanical insights. Through the implementation of a Logistic Regression within an intuitive Streamlit application, users were granted a portal into the world of predictive modelling and exploratory data analysis. The codebase, machine learning model parameters, and EDA results stand as testaments to the meticulous process undertaken to marry historical datasets with contemporary machine learning techniques.

The project's heartbeat lies in its educational aspect, catering to both novices and seasoned practitioners. Users can not only witness real-time predictions but also delve into the mean measurements of Sepal and Petal attributes, gaining a deeper understanding of the dataset's nuances.

Looking forward, the project propels into a visionary realm. The future beckons with the promise of an Iris Flower Classification app, transcending manual data entry through automated image recognition. This transformative step envisions a user-centric experience where the complexities of feature extraction dissolve, making flower species identification as simple as uploading an image.

As the final petal falls in this chapter, the Iris Flower Classification project emerges not merely as a standalone application but as a bridge connecting the historical significance of datasets with the boundless potential of machine learning applications. It invites users and developers alike to envision a future where human-computer interaction evolves, making sophisticated analyses as accessible as a click or an image upload.

In conclusion, the Iris Flower Classification project is not just a culmination but a commencement—a seed planted in the soil of machine learning, waiting to blossom into new dimensions of knowledge, innovation, and user empowerment.

6. FUTURE SCOPE

Automated Image Recognition Refinement:

- Enhance the automated image recognition feature to accommodate a broader range of flower species, making the application versatile for a variety of botanical classifications.

Integration of Additional Features:

- Expand the feature set to include additional botanical attributes, enabling a more nuanced understanding of flower characteristics beyond the current focus on Sepal Length, Sepal Width, Petal Length, and Petal Width.

User Feedback Implementation:

- Incorporate user feedback mechanisms within the application to continually refine and optimize the machine learning model. This iterative process ensures the model evolves with user needs and adapts to emerging trends.

Cross-Platform Accessibility:

- Develop cross-platform compatibility, enabling users to access the application seamlessly across various devices, including mobile phones and tablets, fostering broader accessibility.

Educational Modules:

- Expand the educational aspect of the application by integrating interactive modules that provide users with insights into the functioning of machine learning algorithms. This could include visualizations, tutorials, and guided exercises.

Integration with External Databases:

- Explore possibilities for integrating the application with external botanical databases, allowing users to access a wealth of information beyond the scope of the initial dataset.

Community Engagement and Collaboration:

- Establish a community hub where users can share insights, exchange ideas, and collaborate on improving the application. This collaborative approach can lead to a more robust and diversified tool.

Real-Time Data Updates:

- Implement mechanisms for real-time updates of the underlying dataset, ensuring that the machine learning model remains relevant and adaptable to changes in botanical classifications.

Commercial Applications:

- Explore the potential for commercial applications, where the automated image recognition capabilities could be utilized in industries such as horticulture, agriculture, and environmental conservation.

Advanced Machine Learning Techniques:

- Investigate the integration of advanced machine learning techniques, such as deep learning, to further enhance the accuracy and capabilities of the predictive model.

7 APPENDIX

```python
# Importing requisite libraries
import streamlit as st
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

Fig: 7.1 Necessary libraries are imported

Here, the necessary libraries are imported:

- streamlit for creating the web app.
- pandas for data manipulation.
- train_test_split from sklearn.model_selection for splitting the dataset.
- Logistic Regression using from sklearn.linear_model import LogisticRegression

```python
# Page configuration
st.set_page_config(
    page_title='Iris Flower Classification App',
    page_icon='🌷',
    layout='wide',
    initial_sidebar_state='expanded')
```

Fig: 7.2 Configuration for the Streamlit

This sets the configuration for the Streamlit app, including the page title, icon, layout style, and initial sidebar state.

```python
# Title of the app
st.title('🌷 Iris Flower Classification App')
```

Fig: 7.3 Title of the app

Displays the title of the app using Streamlit's title function.

```python
# Load dataset
df = pd.read_csv(r"D:\Academic\Project\IRIS.csv")
```

Fig: 7.4 Loads the Iris dataset

Loads the Iris dataset from the provided CSV file into a Pandas DataFrame.

```
# Input widgets
st.sidebar.subheader('Input features')
sepal_length = st.sidebar.slider('Sepal length', 4.3, 7.9, 5.8)
sepal_width = st.sidebar.slider('Sepal width', 2.0, 4.4, 3.1)
petal_length = st.sidebar.slider('Petal length', 1.0, 6.9, 3.8)
petal_width = st.sidebar.slider('Petal width', 0.1, 2.5, 1.2)
```

Fig: 7.5 Creates sliders

Creates sliders in the sidebar for users to input values for Sepal and Petal dimensions.

```
# Separate to X and y
X = df.drop('Species', axis=1)
y = df.Species
```

Fig: 7.6 Separates the dataset

Separates the dataset into features (X) and the target variable (y).

```
# Data splitting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state
```

Fig: 7.7 Data into training and testing sets

Splits the data into training and testing sets using the train_test_split function.

```
# Model building
lr = LogisticRegression(random_state=42)
lr.fit(X_train, y_train)
```

Fig: 7.8 Logistic Regression model

Creates a Logistic Regression model and fits it to the training data.

```
# Apply model to make predictions
y_pred = rf.predict([[sepal_length, sepal_width, petal_length, petal_width]])
```

Fig: 7.9 Make predictions based on user input

Uses the trained model to make predictions based on user input.

```
# Print EDA
st.subheader('Brief EDA')
st.write('The data is grouped by the class and the variable mean is computed for eac
groupby_species_mean = df.groupby('Species').mean()
st.write(groupby_species_mean)
st.line_chart(groupby_species_mean.T)
```

Fig: 7.10 Exploratory Data Analysis (EDA)

Performs a brief Exploratory Data Analysis (EDA) by displaying the mean measurements of Sepal and Petal attributes for each Iris species and visualizing them with a line chart.

```
# Print input features
st.subheader('Input features')
input_feature = pd.DataFrame([[sepal_length, sepal_width, petal_length, petal_width]]
                            columns=['sepal_length', 'sepal_width', 'petal_length', '
st.write(input_feature)
```

Fig: 7.11 DataFrame

Displays the user-input features in a DataFrame.

```
# Print prediction output
st.subheader('Output')
st.metric('Predicted Class:', y_pred[0], '')
```

Fig: 7.12 Output

Prints the predicted class based on the user input. The result is shown in the Streamlit app as a metric.

8 BIBLIOGRAPHY

1 Iris Flower Species Identification Using Machine Learning Approach
https://ieeexplore.ieee.org/abstract/document/9057891

2 Iris Flower Classification Using Machine Learning
https://www.researchgate.net/profile/Tumma-Srinivasarao/publication/359064863_Iris_Flower_Classification_Using_Machine_Learning/links/622632fb3c53d31ba4af19e1/Iris-Flower-Classification-Using-Machine-Learning.pdf

3 Iris Flower Dataset - https://www.kaggle.com/datasets/arshid/iris-flower-dataset

4 Data Professor (Chanin Nantasenamat)
https://www.youtube.com/@DataProfessor

5 IRIS Flowers Classification Using Machine Learning
Karpuram Dhanalakshmi Srivani — Updated On April 5th, 2023
https://www.analyticsvidhya.com/blog/2022/06/iris-flowers-classification-using-machine-learning/

6 Iris Flower Classification with Machine Learning
Aman Kharwal
https://thecleverprogrammer.com/2021/10/17/iris-flower-classification-with-machine-learning/

7 Machine Learning For Absolute Beginners by Oliver Theobald

8 Machine learning-Iris classification
Jebaseelan Ravi
https://medium.com/@jebaseelanravi96/machine-learning-iris-classification-33aa18a4a983