Mrs I.Sravani, 2025, 13:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

IRIS Flower Species Prediction Using Machine Learning and Web Based Interactive Tool for Non Technical Users

Assistant Professor Mrs I.Sravani, R.S.MD.Sahil, Y Jaya Krishna, G Murari, M Murali, P Raghu
Department of Computer Science Engineering
Sai Rajeswari Institute of Engineering & Technology

Abstract- The Iris flower species prediction tool is a web-based application that leverages machine learning to classify Iris flowers into one of three species (Setosa, Versicolor, or Virginica) based on their physical measurements: sepal length, sepal width, petal length, and petal width. Using a classification model trained on the famous Iris dataset, the tool predicts the species of a flower given these inputs. The system is designed to be user-friendly and accessible for non-technical users. A simple web interface, built with Flask, allows users to input flower measurements and receive predictions in real-time. This web tool integrates the power of machine learning with an intuitive user experience, making it easy for anyone, regardless of their technical background, to interact with and benefit from the model1. The system also includes validation features and visualizations to further enhance user engagement and understanding. By deploying the model on cloud platforms like Heroku, this tool can be accessed globally, serving educational purposes or assisting botanists and enthusiasts in identifying Iris species based on simple measurements.

Keywords- Iris Flower Classification, Machine Learning, Species Prediction, Web Application, Flask, Interactive Tool

I. INTRODUCTION

The Iris flower dataset is one of the most well-known datasets in machine learning, frequently used to demonstrate classification techniques. It contains measurements of various features of Iris flowers—specifically, sepal length, sepal width, petal length, and petal width—and uses these features to classify the flowers into one of three species: Setosa, Versicolor, or Virginica. The challenge of classifying flowers based on these features has been widely used to showcase the power of machine learning algorithms, such as decision trees, support vector machines, and random forests. Despite its prominence in the machine learning community, the process of classifying Iris flowers based on these attributes may not be accessible to individuals without a technical background. This is where an interactive, web-based tool can provide immense value. A non-technical user, such as a botanist, educator, or student, could benefit from a simple and intuitive interface that allows them to input the measurements of a flower and get an immediate species prediction. This project seeks to bridge the gap between complex machine learning techniques and everyday users by developing a web-based tool for Iris flower species prediction. The tool will employ a pre-trained machine learning model to classify the Iris species based on four key measurements. It aims to be easy to use, requiring no prior knowledge of machine learning or programming2. By integrating machine learning with web technologies, this system allows anyone to classify Iris flowers quickly and easily, contributing to educational efforts and assisting those in fields like botany, horticulture, and environmental science. Moreover, the tool's deployment on a cloud platform ensures that it is accessible worldwide, making it a powerful tool for students, researchers, and nature enthusiasts alike.

II. EXPERIMENTAL DATA

In this project, we use the well-known Iris flower dataset to train and evaluate the machine learning model for predicting the species of an Iris flower based on its physical attributes. Below is a description of the dataset, including the features, target variables, and some sample data.

Dataset Overview

- Number of Instances (Samples): 150
- Number of Features (Attributes): 4
- Target Variable (Species): 3 classes (Setosa,

Versicolor, Virginica)

Feature Types

- Sepal Length (in cm)
- Sepal Width (in cm)
- Petal Length (in cm)
- Petal Width (in cm)

Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Species
5.1	3.5	1.4	0.2	Setosa
4.9	3.0	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5.0	3.6	1.4	0.2	Setosa
7.0	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
5.5	2.3	4.0	1.3	Versicolor
6.5	2.8	4.6	1.5	Versicolor
6.3	3.3	6.0	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3.0	5.9	2.1	Virginica
6.3	2.9	5.6	1.8	Virginica
6.5	3.0	5.8	2.2	Virginica

Dataset Structure

The Iris dataset is typically organized into a table with rows representing different flowers and columns representing the flower's features and target species. Below is an example of the dataset format:

Explanation of Features

- **Sepal Length (cm):** The length of the sepal of the flower.
- **Sepal Width (cm):** The width of the sepal of the flower.
- **Petal Length (cm):** The length of the petal of the flower.
- **Petal Width (cm):** The width of the petal of the flower.

Target Variable (Species)

- **Setosa:** This species typically has smaller petals and sepals compared to the other two species.
- **Versicolor:** This species has medium-sized petals and sepals.
- **Virginica:** This species has the largest petals and sepals..

Example of Data Preprocessing Code:

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Load the Iris dataset
iris = datasets.load_iris()
X = iris.data  # Feature matrix
y = iris.target  # Target labels (species)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

# Standardize the features (scaling to mean=0, variance=1)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Check the shapes of the datasets
print("Training data shape:", X_train.shape)
```

III. MODEL TRAINING AND EVALUATION

After preprocessing the data, the next step is to train the machine learning model using an algorithm like Random Forest or Support Vector Machines (SVM). The model is then evaluated using the test set, and the accuracy of the predictions is assessed.

Evaluation Metrics

- **Accuracy:** The percentage of correctly classified instances in the test set.
- **Confusion Matrix:** A table that describes the performance of the model in terms of true positives, false positives, true negatives, and false negatives.

Sample Model Evaluation Code

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix

# Initialize and train the RandomForest model
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100}%")

# Display confusion matrix
cm = confusion_matrix(y_test, y_pred)
```

IV. RESEARCH METHODOLOGY

The research methodology for the Iris Flower Species Prediction project combines machine learning, web development, and user experience design to create a predictive model and a user-friendly tool for non-technical users. The goal is to develop an accessible web-based application that predicts the species of Iris flowers based on four key physical attributes (sepal length, sepal width, petal length, and petal width)3. Below is a detailed description of the steps and methodologies used in the research process.

1. Problem Definition

The first step in this research is identifying the problem to solve: predicting the species of Iris flowers based on their physical attributes. The target audience is non-technical users, and the goal is to provide a simple, accessible interface where anyone can input measurements and receive a prediction without needing expertise in machine learning.

2. Data Collection

The dataset used in this study is the Iris dataset from the UCI Machine Learning Repository, which is a well-known dataset in machine learning. It contains 150 samples of Iris flowers from three different species: Setosa, Versicolor, and Virginica.

The dataset includes the following features:

- Sepal Length (in cm)
- Sepal Width (in cm)
- Petal Length (in cm)
- Petal Width (in cm)

These features are used to train and evaluate machine learning models to predict the species.

3. Data Preprocessing

Data preprocessing is an essential part of the methodology to ensure that the data is suitable for machine learning. The preprocessing steps include:

Data Cleaning: Checking for missing or inconsistent values in the dataset. If any missing data is found, appropriate handling (such as imputation or removal) will be performed.

Normalization/Standardization: The features (sepal length, petal length, etc.) are scaled to ensure that the model does not give disproportionate weight to any single feature. This is important for algorithms like K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) that are sensitive to the scale of features.

Data Splitting: The dataset is split into training and testing sets (commonly 80% for training and 20% for testing) to allow the model to be trained on one portion and evaluated on another, preventing overfitting4.

4. Model Selection

Several classification algorithms were considered for the task of predicting the Iris species. These include:

- **Random Forest Classifier:** A popular ensemble method that creates a number of decision trees and combines their results to make a final prediction.

- **Support Vector Machine (SVM):** A powerful classifier that finds the hyperplane that best separates the data into different classes.
- **K-Nearest Neighbors (KNN):** A simple yet effective classification algorithm that classifies a data point based on the majority class of its nearest neighbors.
- **Logistic Regression:** A basic and interpretable classifier that estimates probabilities and predicts class labels.

The Random Forest Classifier was chosen for this project due to its high performance, ease of use, and robustness to overfitting.

5. Model Training and Evaluation

After splitting the dataset into training and testing sets, the following steps were carried out:

Training the Model: The chosen machine learning model (Random Forest Classifier) is trained on the training dataset. The model learns to map the four input features (sepal length, sepal width, petal length, and petal width) to the correct Iris species.

Model Evaluation: Once the model is trained, it is tested on the testing set to evaluate its performance. Key metrics for evaluating the model include:

- **Accuracy:** The percentage of correct predictions made by the model on the testing set.
- **Confusion Matrix:** A table used to evaluate the performance of a classification algorithm by showing the number of true positives, false positives, true negatives, and false negatives.
- **Precision, Recall, and F1-Score:** These metrics provide additional insights into how well the model performs, especially when dealing with imbalanced datasets.

6. Development of Web Interface

A web-based interface is developed using the Flask framework, which allows non-technical users to interact with the model. The web application allows users to input the sepal and petal measurements of a flower and receive a prediction about the species. The interface is designed to be simple and intuitive, with the following steps:

- **Frontend Design:** HTML, CSS, and JavaScript are used to create a clean and responsive user interface. Bootstrap is used for layout styling to ensure that the application is mobile-friendly.
- **Backend Design:** The Flask framework handles user input and routes it to the pre-trained machine learning model, which makes predictions based on the input data.
- **Deployment:** The Flask web application is deployed on a cloud platform (e.g., Heroku) to ensure accessibility. Users can access the web tool from anywhere via a web browser.

7. User Testing

To ensure that the web tool is user-friendly and accessible, user testing is conducted with non-technical users. The aim is to:

- Gather feedback on the user interface and usability.
- Ensure that the instructions for inputting data are clear and easy to understand.
- Assess the accuracy and helpfulness of the species prediction results.

This step helps in fine-tuning the user interface, improving the design, and addressing any challenges faced by non-technical users when interacting with the system.

8. Results and Evaluation

After testing and deploying the web tool, the following results are evaluated:

- **Model Performance:** The accuracy of the trained machine learning model, based on the testing data.
- **User Feedback:** Feedback from non-technical users on the ease of use and functionality of the web tool.
- **Tool Deployment:** Evaluation of the web tool’s accessibility, including response times, usability, and real-time predictions6.

Future Scope

The future scope of the Iris Flower Species Prediction project using machine learning and a web-based interactive tool for non-technical users is vast and promising. Enhancing the model's accuracy through deep learning techniques, such as CNNs and ensemble methods, can significantly improve species classification. Transitioning from numerical input-based predictions to real-time image-based classification using computer vision can make the tool more intuitive. Additionally, integrating the system into mobile applications will increase accessibility, while incorporating multilingual support and speech-to-text capabilities can further broaden its reach. Cloud-based data storage and API integration will enable scalability, allowing seamless data management and interoperability with other applications. The introduction of self-learning algorithms will help the model adapt over time based on user feedback, ensuring continuous improvement. Implementing an AI chatbot can enhance user experience by providing instant guidance and species-related information. Furthermore, the integration of IoT devices could allow real-time analysis of flower characteristics for research and smart gardening applications. Encouraging crowdsourced data collection and fostering a community-driven platform for botany enthusiasts will enhance dataset diversity and engagement. Lastly, this tool can be leveraged in educational and research fields, assisting students and scientists in botanical studies. With these advancements, the system can evolve into a comprehensive, intelligent, and user-friendly platform for plant identification and research7.

V. CONCLUSION

The Iris Flower Species Prediction project successfully demonstrates how machine learning can be applied to a real-world problem, allowing for the classification of Iris flower species based on four key features: sepal length, sepal width, petal length, and petal width. By leveraging the power of the Iris dataset, this tool provides an accessible and user-friendly web application that enables non-technical users to predict the species of Iris flowers with ease.

Key conclusions from this project include:

1. Effective Use of Machine Learning

The project utilized the Random Forest Classifier to achieve high accuracy in predicting the species of Iris flowers. The model performed well on the dataset, demonstrating the power of ensemble methods in classification tasks. It shows that even relatively simple machine learning algorithms can produce accurate results with well-defined datasets.

2. User Accessibility

The development of a web-based tool ensures that the application is accessible to a wide range of users, including those without technical expertise. The tool enables users to easily input flower measurements and receive predictions about the species, making it a valuable resource for botany enthusiasts, researchers, and students alike.

3. Integration of Machine Learning with Web Development

By integrating machine learning with a Flask-based web interface, the project highlights the importance of combining these technologies to create practical, real-world applications. The easy-to-use interface, combined with powerful backend algorithms, demonstrates how technology can simplify complex processes and make them accessible to a broad audience.

4. Future Potential

The scope for future improvements and expansion is vast. The system can be extended to predict other plant species, incorporate more features (such as environmental data), and even be developed into a mobile application or augmented reality tool. The inclusion of educational resources and real-time updates could also enhance the overall user experience, promoting learning and exploration9.

5. Impact on Botanical Research and Education

This tool can serve as an educational platform for learning about plant species and their classification. It can also assist in various practical applications, including conservation efforts, agriculture, and gardening. Its ability to classify flowers based on measurable physical attributes opens up possibilities for more precise and rapid identification of species in diverse environments.

6. Challenges and Learnings

One of the challenges faced during the development was ensuring the model's accuracy and usability. The simplicity of the Iris dataset presented both an opportunity and a limitation; while the model is effective for this dataset, future versions of the tool must be expanded to handle more complex or varied data. Ensuring the model's generalization to a wider variety of species would be key to its scalability and effectiveness8.

REFERENCES

1. **Fisher, R. A. (1936).** The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7(2), 179-188. This is the original paper by R. A. Fisher, where he introduced the Iris dataset and used it for developing linear discriminant analysis.
2. **UCI Machine Learning Repository. (2021).** Iris Data Set. Retrieved from. The Iris dataset was sourced from the UCI Machine Learning Repository, one of the most popular datasets for classification problems in machine learning.
3. **Breiman, L. (2001).** Random forests. Machine Learning, 45(1), 5-32. This paper introduced the concept of the Random Forest algorithm, which was used in this project for classifying the Iris species based on sepal and petal measurements.
4. **Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011).** Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. This paper discusses scikit-learn, the machine learning library used in this project for model training and evaluation.
5. **Flask Documentation. (2021).** Flask web framework. Retrieved from https://flask.palletsprojects.com/ The Flask documentation was referenced to build and deploy the web application that interacts with the machine learning model for flower species prediction.
6. **Bootstrap Documentation. (2021).** Bootstrap, The Most Popular HTML, CSS, and JavaScript Framework for Developing Responsive, Mobile-First Projects on the Web. Retrieved from .Bootstrap was used to design the responsive front-end of the web tool, ensuring ease of use and accessibility for non-technical users.
7. **Kingma, D. P., & Ba, J. (2015).** Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR). This paper discusses the Adam optimization algorithm which can be implemented in more advanced models for optimizing machine learning models, such as deep learning.
8. **He, K., Zhang, X., Ren, S., & Sun, J. (2016).** Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). This paper could be referenced if the project extends to image classification, showing how deep learning models like CNNs could enhance the tool's capabilities for recognizing flowers from images.
9. **Chollet, F. (2015).** Keras. GitHub Repository. Retrieved from. Keras was considered for developing deep learning models in future iterations of the tool. It is a high-level neural networks API, written in Python and capable of running on top of TensorFlow.