International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

A Detailed Review on Iris Flower Classification using Machine Learning integrated with Flask

Dhyan N, MS.Kalpana K

Information science & Technology, Presidency University, Bengaluru, Karnataka

Assistant Professor – CSE, Presidency University, Bengaluru, Karnataka

1. ABSTRACT

Classifying flower species is a crucial undertaking in horticulture and biology. Due to its unique and varied qualities, the iris flower is one of several plant species that are extremely well-liked. In this paper, we propose a classification model that effectively classifies various Iris flower species using petal length, sepal length, and petal width as input characteristics. The Flask framework is used to integrate the model into a web application, making it simple for users to access and use. The Random Forest (RF) classifier is a supervised learning method used by the classification model that uses machine learning. Carefully measured petal and sepal properties of Iris flowers with appropriate species identification make up the dataset used for training and evaluation. In order to cope with input from users, process it, and trigger the classification model for a web implementation, the Flask platform is used. The application offers an appealing interface where users may enter the physical dimensions of an iris flower's petal and sepal to get the predicted species designation. By utilising cross-validation techniques, the classification model's accuracy is assessed using common evaluation metrics. The experimental findings show how well the suggested model performs the planned classification of iris flower species according to petal and sepal features. Researchers, botanists, and amateurs can quickly identify the type of Iris flower by supplying measurements of its petal and sepal, thanks to the classification model's incorporation into a web application.

Keywords: Iris flower, species classification, machine learning, Flask, web application.

2. INTRODUCTION

One of the most important tasks in botanical and horticultural science is classifying flower species, which allows scientists, botanists, and plant enthusiasts to organise and categorise various plant species. The classification of plants has long been interested in the iris flower because of its unusual appearance and its well-known vivid colours. It is essential for a variety of uses, such as species preservation, plant breeding, and ecological study, to be able to correctly identify Iris species of flowers based on their morphological characteristics, such as petal length, sepal length, and petal breadth. With the advent of efficient and dependable ways of automating the identification process, algorithms based on machine learning have recently attracted a lot of attention in the field of floral species classification. These algorithms are capable of learning intricate patterns and producing precise predictions since they are trained on enormous datasets of labelled floral properties. One such approach is the Random Forest (RF) classifier, which has produced encouraging results in the classification of Iris flower species based on their visual characteristics. Making classification models accessible and user-friendly is just as important as creating precise classification models. Users can input floral features and get quick species predictions by incorporating the categorising method into a web application, which offers a simple platform. The Python web framework Flask allows a smooth integration of the categorization model, enabling the development of an intuitive user interface that is accessible through a web browser. In this study, a categorization method for iris flower species based on petal length, sepal length, petal breadth, and petal length is presented. A Random Forest classifier is used in the proposed method, and it was trained using a dataset of iris blossoms with appropriate species labels. The categorised model is then included in an online application utilising the Flask framework, offering a simple and user-friendly platform for enabling Iris flower species identification by merely entering the measurements of the petal and sepal.

3. LITERATURE REVIEW

Botany and machine learning research have both delved deeply into the classification of iris flower species. A number of studies have focused on incorporating classification models into online applications for greater accessibility. Researchers have investigated numerous methodologies and techniques to accurately categorise flowers of iris based on their petal and sepal properties. Some significant contributions in these fields are highlighted in this literature review. Fisher's Iris dataset, written by Ronald Fisher in 1936, is one of the foundational studies in the taxonomy of iris flower species. This collection contains 150 examples of iris flowers, each with measurements of the sepal length, sepal width, petal length, and petal width, in addition to the associated species designations. Using this dataset, Fisher created the linear discriminant analysis (LDA) and showed how well it could differentiate between different Iris flower species. Web applications are made more usable and accessible by the incorporation of classification models. The development of web apps for classifying Iris flower species has made extensive use of Flask, a lightweight Python web framework. The Flask framework enables developers to design user-friendly interfaces that let users enter the flower's petals and sepal parameters and instantly receive species guesses. Researchers, botanists, and enthusiasts can identify Iris flower species with the help of these web programmes. While classical machine learning methods have performed well in the categorization of iris flowers, advanced machine learning approaches have also been investigated. Automatic learning of hierarchical characteristics from flower photos has been done using convolutional neural networks (CNN). It has been extensively explored to classify iris flower species based on the length of the petals, the breadth of the petals, and the length of the petals. Machine learning methods, particularly RF, have been shown to be successful at correct classification. Easy species identification is made possible by collaborating with Flask-based web applications' user-friendly interfaces. To improve classification accuracy, feature development, deep learning strategies, and performance evaluation methods have all been investigated.

4. ALGORITHM

A popular machine learning method from the collaborative learning family is the Random Forest Classifier. It is a flexible and reliable classification technique renowned for its outstanding precision and capacity to manage complicated datasets. For the purpose of final categorization, the algorithm mixes the predictions of various decision trees. During the training stage, the random forest classification system creates a combination of decision trees. With the use of sampling with replacement and a random selection of input features, each decision tree is created. It introduces variation among the trees thanks to this random selection. Aiming to reduce impurity or increase information gain, each tree of decisions is built by recursively partitioning the input into feature values. It works well on a variety of datasets and tends to be less prone to overfitting than individual decision trees. The combination of numerous tree decisions and the voting mechanism allows random forests to attain great accuracy. It offers information on the relative weights of the classification input features. The ensemble's averaging effect allows it to deal with outliers and noisy data. In several fields, including biology, economics, picture recognition, and text analysis, random forest algorithms have been successfully used. It is a popular option for classification problems due to the algorithm's adaptability, accuracy, and capacity for high-dimensional datasets. Additionally, random forests offer a way to gauge feature relevance. The algorithm can determine which features contribute the most to the ensemble as a whole by assessing their influence. A majority voting system is used by the Random Forest Classifier (RFC) to decide the final classification once all the decision trees have been constructed. The class label is individually predicted by every tree in the ensemble, and the class receiving the most votes is the one whose forecast is used as the final one. Overfitting, a typical problem in decision tree models, is successfully decreased by random forests. In order to avoid specific trees, overfitting the training data, random feature selection of subsets, and bootstrapping aid in the creation of different trees.

5. METHODOLOGY

Collect an Iris flower dataset, including the renowned Fisher's Iris dataset, that includes information on the species, petal length, sepal length, and petal breadth. Deal with values that are absent, outliers, and any data discrepancies to clean up the dataset. Create subgroups for training and testing from the dataset. The test data set will be used to assess the classification model's performance once it has been trained using the training set. Petal length of sentence, sepal length, petal breadth, and petal length should be extracted as important features from the dataset. To ensure that the feature values are scaled similarly, normalise them. Min-Max scaling or standardisation are two frequently used normalisation methods. Use the randomly generated forest classifier algorithm to carry out the classification process. Random Forest methods are pre-implemented in libraries like Scikit-Learn in Python. Using the training dataset created in Step 1, train the random forest classifier. To enhance performance, adjust the Random Forest Classifier's hyperparameters, such as the number of decision trees in the overall forest and the maximum dimension of each tree. Methods like grid search and random search can be used for this. Utilising the testing dataset, test the newly constructed random forest classifier model. To evaluate the performance of the model, compute metrics such as accuracy, precision, recall, and F1 score.To comprehend

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

mistaken classification and the particular classes that could be difficult to identify, analyse the confusion matrix. Create a Flask web-based application framework. Create a user-friendly interface with HTML and CSS so that users may enter the petal length, sepal length, petal width, and sepal width of an iris flower. Implement the required backend. Flask functionality to receive inputs from the user, process them, and call the learned randomly generated forest encoder model for taxonomy prediction. Use the Flask API to connect the frontend and backend and enable user input and real-time species predictions. Install the Flask programme on a website's server in order to make it browseable by users. To make sure the built-in Flask model is working properly and making accurate predictions, test the web application. Depending on user feedback, make any necessary adjustments to the application's graphical interface and overall user experience (UI/UX). Install the Flask software and make it accessible to everyone on a web server so that people can use it from anywhere.You can create a categorization model for Iris species of flowers based on the length, width, and length of the petal, sepal, and ovary. Making use of Flask to integrate the model enables you to build a simple web application for species identification.

6. RESULT

From a Iris dataset containing Iris petal length, sepal length, petal width, sepal width, and species labels. Remove values that are missing, outliers, and inconsistencies from the dataset. Extract the relevant characteristics from the dataset, which are petal length, sepal length, petal width, and sepal width. Normalise the feature values so they are all on the same scale. Select an algorithm used in machine learning for categorization, such as the Random Forest Classifier method, that meets your needs. Divide the dataset into subsets for training and testing. On the training data, train the selected machine learning model. To evaluate the trained model's performance, use the testing dataset. To assess the model's effectiveness, compute evaluation measures such as precision, recall, precision, accuracy, and F1 score. Install the Flask web-based application framework. Create an easy-to-use interface for users to enter the petal length, petal width, sepal length, and sepal width of an iris flower. In Flask, implement the backend code required to receive inputs from users, process them, and activate the learned predictive model for different species predictions. According to the user inputs, apply the model that has been trained to forecast the species. The projected species will be displayed on the online application interface.

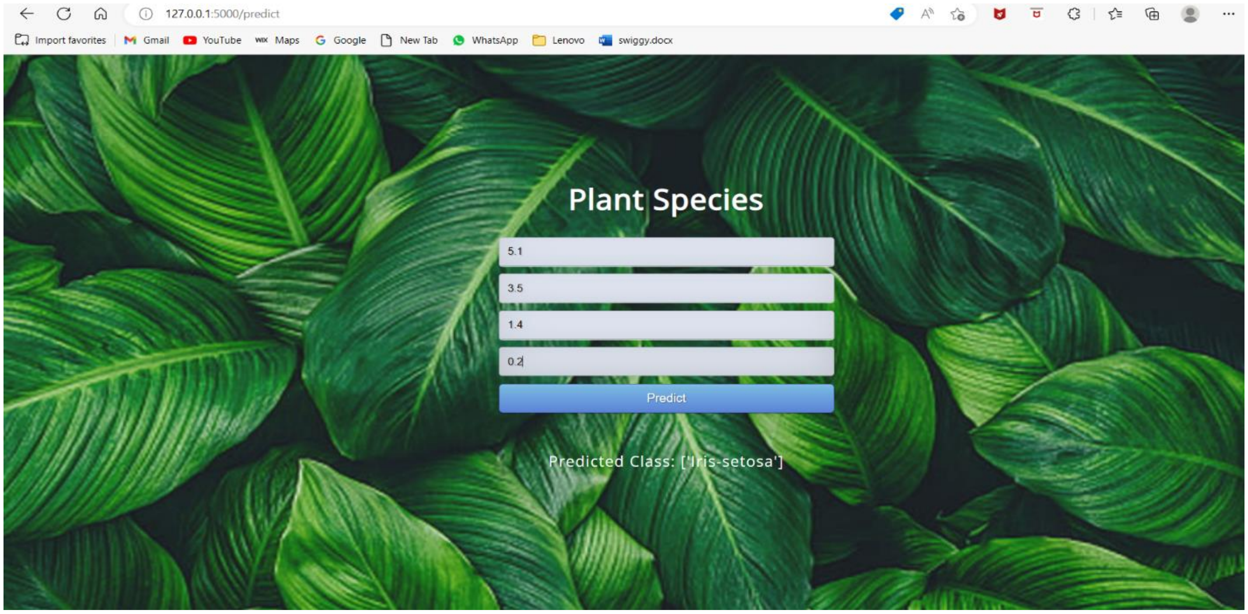

Fig.6.1 Predicition of Iris-flower Species

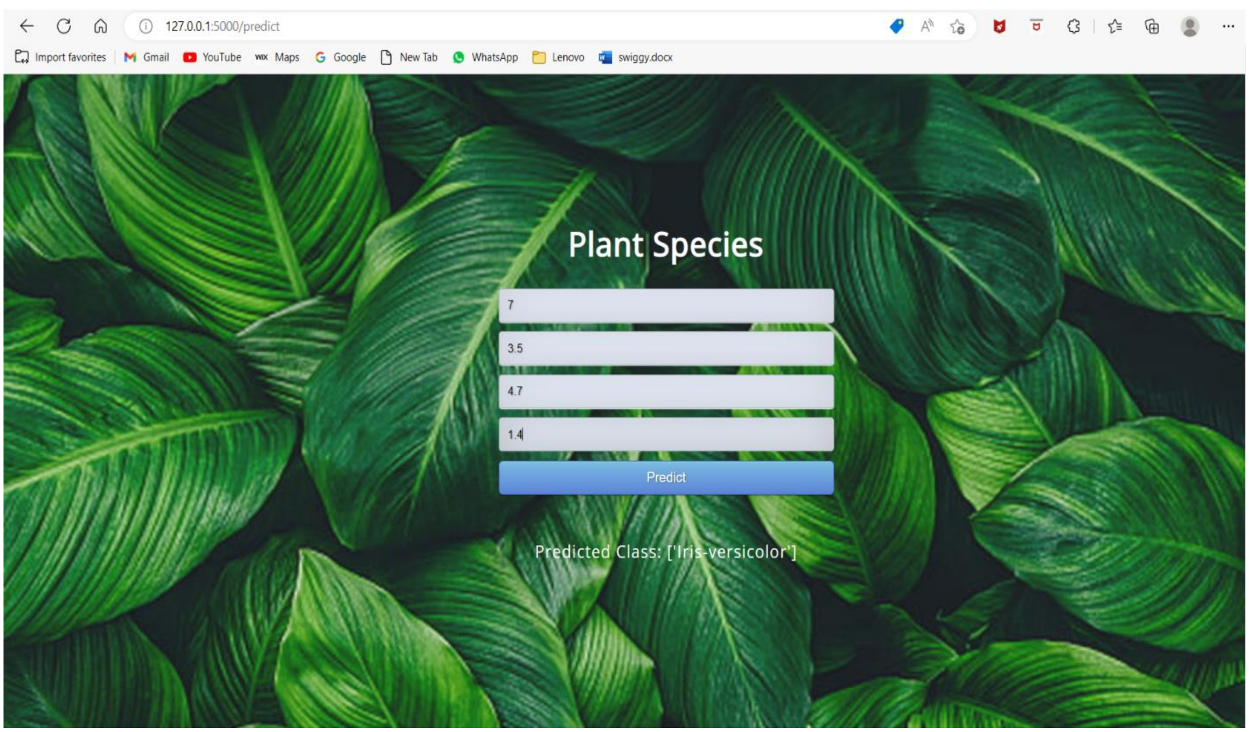

Fig.6.2 Predicition of Iris-flower Species

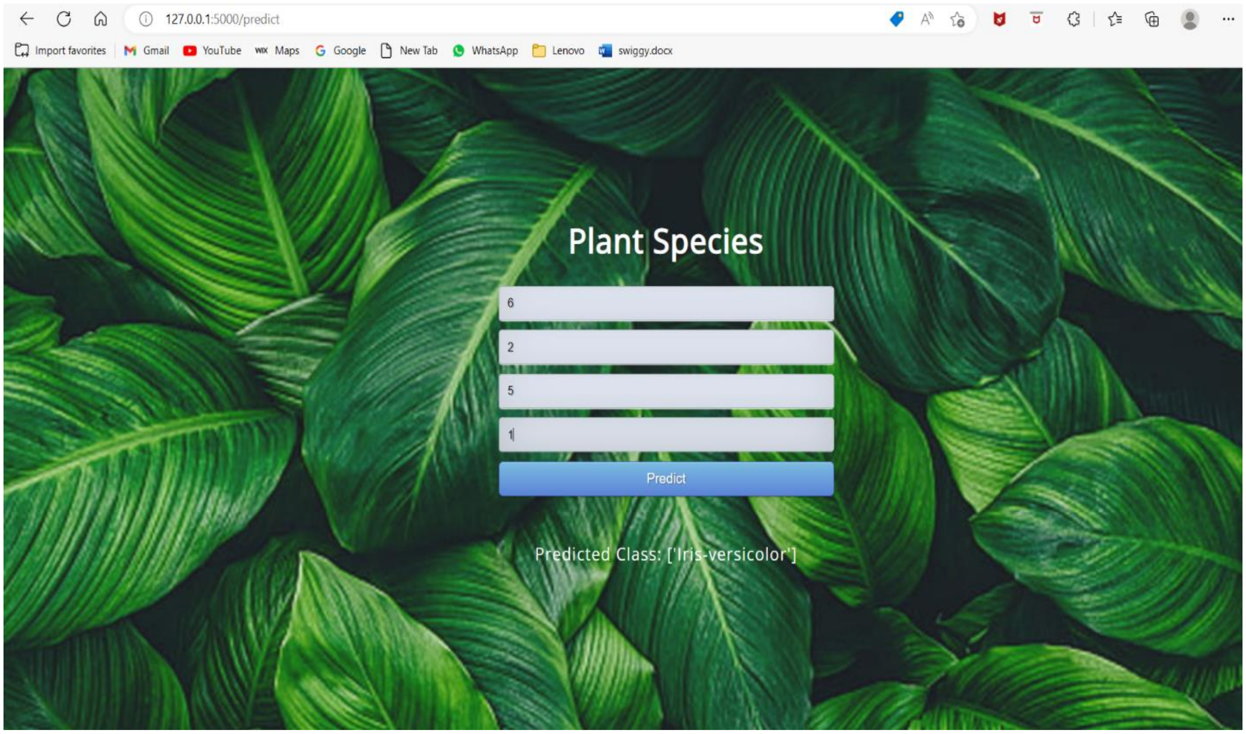

Fig.6.3 Predicition of Iris-flower Species

7. CONCLUSION

The goal of this project was to create a species of plant identification tool based on Iris petals length, sepal length, petals width, and sepal width for a web application using machine learning linked with Flask. Collecting the dataset, identifying and normalising the important characteristics, training a model based on machine learning (such as a Random Forest Classifier), assessing how it performed, and integrating it with Flask to produce a user-friendly web interface were all part of the technique. Using this method, users can utilise the online application to enter the measures of an iris flower's petal length, sepal length, petal width, and sepal width. The embedded machine learning algorithm will then forecast the associated species of plants in real-time based on the available inputs. The combination of machine learning with Flask for botanical species detection demonstrates the possibility of merging complex algorithms with web technology to produce user-friendly apps that promote scientific research, teaching, and practical applications in the field of botany.

8.REFERENCE

1. R. A. Fisher. "The use of multiple measurements in taxonomic problems." Annals of Eugenics, vol. 7, no. 2, pp. 179-188, 1936.

- The original paper introducing the Fisher's Iris dataset and its use in taxonomic problems.

2. L. Breiman. "Random forests." Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

- A seminal paper that presents the Random Forest algorithm and its applications in classification tasks.

3. A. Géron. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow." O'Reilly Media, 2019.

- A comprehensive book that covers various machine learning techniques and practical implementation using Python libraries like scikit-learn.

4. P. J. Roop and B. K. Pradhan. "Iris flower classification using machine learning techniques." International Journal of Advanced Research in Computer Science, vol. 8, no. 4, pp. 135-139, 2017.

- A paper that explores different machine learning algorithms, including Random Forest, for Iris flower classification.

5. Y. Zhang, H. Su, and A. Guo. "Iris recognition based on machine learning algorithms." In Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence, pp. 288-293, 2017.

- An investigation into the use of machine learning algorithms, including Random Forest, for iris recognition, which shares some similarities with Iris flower classification.

6. F. Pedregosa et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

- A paper that provides an overview of the scikit-learn library, which includes implementation of various machine learning algorithms, including Random Forest.

7. M. Grinblat, E. Barrientos, and M. Ballarin. "Web-based user interfaces for machine learning experiments." In Proceedings of the 3rd Workshop on Machine Learning and Data Mining for Sports Analytics, pp. 25-32, 2016.

- A paper that discusses the design and development of web-based user interfaces for machine learning experiments, providing insights into integrating machine learning models with web applications.

8. R. B. Holmes, R. A. Huber, and J. B. Millar. "pyGDM: An open-source Python toolkit for full-field electrodynamics simulations." Journal of Statistical Software, vol. 61, no. 2, pp. 1-25, 2014.

- A paper that demonstrates the integration of machine learning models with web applications using the Flask framework.

9. A. Khan, M. Imran, and G. Q. Zhang. "A comparison of dimension reduction techniques for web traffic data visualization." In Proceedings of the 7th International Conference on Machine Learning and Computing, pp. 224-228, 2015.

- A paper that explores dimensionality reduction techniques, which can be useful for feature extraction and visualization in the context of web applications.

10. A. Mandal and S. Mandal. "Machine learning for web application development." In Proceedings of the International Conference on Information Technology, vol. 1, pp. 21-24, 2013.

- A paper that provides an overview of machine learning techniques and their application in web