SVM Powered Flower Species Classification

PEETHALA BINDHU PRIYA, PADALA SAI BHAVANI

Assistant Professor, 2MCA Final Semester, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract
The Iris flower classification problem is a classic example of multi-class classification used to demonstrate the effectiveness of machine learning algorithms. This project applies supervised learning techniques to accurately predict the species of an Iris flower based on four key features: sepal length, sepal width, petal length, and petal width. Using the well-known Iris dataset, we implemented and evaluated several machine learning models, including Logistic Regression, k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Decision Trees. The dataset was preprocessed and split into training and testing sets to assess each model's performance. Among the evaluated models, the Support Vector Machine achieved the highest accuracy, demonstrating its robustness in handling linearly separable classes. The results highlight the capability of machine learning algorithms in solving simple classification problems effectively and provide a foundation for more complex pattern recognition tasks in the field of artificial intelligence.

IndexTerms: Iris flower classification, Multi-class classification, Supervised learning, Machine learning,Logistic Regression, k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Decision Trees, Feature selection, Sepal length.

1.INTRODUCTION

The Iris flower classification problem is a foundational example in the field of machine learning and pattern recognition. It involves predicting the species of an Iris flower based on four measurable features: sepal length, sepal width, petal length, and petal width. The dataset used, commonly known as the Iris dataset, is widely used for demonstrating the effectiveness of classification algorithms.This project applies supervised learning techniques, particularly focusing on the Support Vector Machine (SVM) algorithm, to classify Iris flowers into three species: *Setosa*, *Versicolor*, and *Virginica*. While the dataset itself is relatively small and simple, it is ideal for testing and evaluating different machine learning approaches.To make this system more accessible, a web-based interface has been developed using Python Flask, allowing users to input flower measurements and get instant predictions without needing to understand the technical background of machine learning. This approach bridges the gap between data science and user-friendly applications.

1.1 Existing System

In existing systems, the classification of Iris flowers is typically carried out either manually by botanists or through machine learning models that require a high level of technical expertise. Manual classification, though reliable, is time-consuming, prone to human error, and impractical for large-scale use. On the other hand, traditional machine learning approaches rely heavily on full-featured libraries like Scikit-learn or TensorFlow, which necessitate local setup, programming knowledge, and data preprocessing such as cleaning, normalization, and feature selection. These systems are often accessed through code-based environments or notebooks, making them inaccessible to non-technical users. Additionally, they lack user-friendly interfaces and are rarely integrated into web-based platforms, which limits their usability for real-time predictions. As a result, current methods are inefficient for general users and not well-suited for deployment in everyday applications that require speed, accuracy, and ease of use.

1.1.1 Challenges:

2 Existing systems are not user-friendly, especially for non-technical users.
3 Require programming knowledge and familiarity with machine learning concepts.
4 Depend on local installation of libraries and software, which can be complex and time-consuming.
5 Involve manual data preprocessing (e.g., cleaning, formatting), increasing the risk of errors.
6 Slower and less efficient for users who need quick and real-time predictions.
7 Limited accessibility due to the need for a technical setup and environment.

7.1 Proposed system:

The proposed system aims to make Iris flower classification simple, fast, and accessible to everyone, regardless of their technical background. At its core, the system utilizes a powerful and efficient machine learning algorithm—Support Vector Machine (SVM)—which has proven to deliver high accuracy, especially for linearly separable datasets like the Iris dataset. The model is trained on labeled data to recognize patterns based on two key flower features: petal length and petal width, which are sufficient for accurate species classification.What sets this system apart from traditional approaches is its user-friendly, web-based interface, developed using Python Flask for backend logic and HTML/CSS for the frontend design. This ensures a smooth and interactive user experience, enabling anyone to input flower measurements directly into a browser without the need for any specialized software or coding skills. There is no requirement for local installation of libraries, manual data handling, or technical configuration—everything runs seamlessly in a lightweight and portable web application.Additionally, the system is designed with responsiveness and speed in mind, providing real-time results as soon as the input is submitted. It demonstrates how machine learning models can be effectively integrated into practical tools that solve real-world problems. This solution not only simplifies the classification process but also bridges the gap between complex machine learning algorithms and everyday usability, making it ideal for education, research, or even mobile deployment in the future.

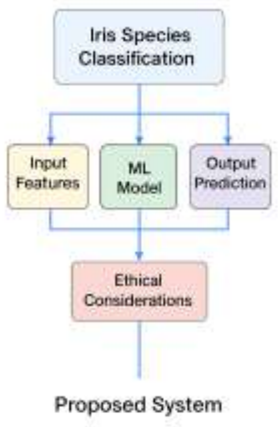


Fig: 1 Proposed Diagram

7.1.1 Advantages:

- **Simple, fast, and accurate classification**:
The system provides quick and reliable predictions with minimal user input, making the classification process efficient and user-friendly.
- **No local installation required for users**:
As a web-based application, users can access the system directly from a browser without the need to install software or machine learning libraries.
- **Intuitive user interface**:
The interface is designed to be clean and easy to use, allowing users to interact with the system effortlessly, even without technical skills.
- **High accuracy using fewer input features**:
By using only petal length and petal width, the system maintains excellent prediction accuracy while keeping the input requirements simple.
- **Portable and extensible for future datasets**:
The application can be easily updated or expanded to support new data types or classification tasks, making it flexible and scalable for future use.

2.1 Architecture:

The architecture of the proposed Iris flower classification system is designed to be simple, efficient, and user-friendly. It consists of three main parts: the frontend (user interface), the backend (server logic), and the machine learning model.

Frontend (User Interface):
This is the part of the system that users interact with. It is built using HTML and CSS, which provides a clean and responsive design. On this webpage, users are asked to enter just two values: petal length and petal width. This keeps the interface simple and easy to use, even for someone without technical knowledge.

Backend (Server Logic):
The backend is developed using Python with the Flask framework. When a user enters the input and clicks "submit," the data is sent to the backend through a secure connection. Flask handles the request, passes the input to the machine learning model, and waits for the prediction result.

Machine Learning Model (SVM):
The core of the system is a pre-trained Support Vector Machine (SVM) model. This model has already been trained on the Iris dataset to understand patterns in flower measurements. It takes the user's input and quickly predicts which of the three Iris species (*Setosa*, *Versicolor*, or *Virginica*) the flower most likely belongs to.

Output Display:
Once the prediction is made, the result is sent back from the backend to the frontend. The webpage then shows the predicted flower species instantly to the user.

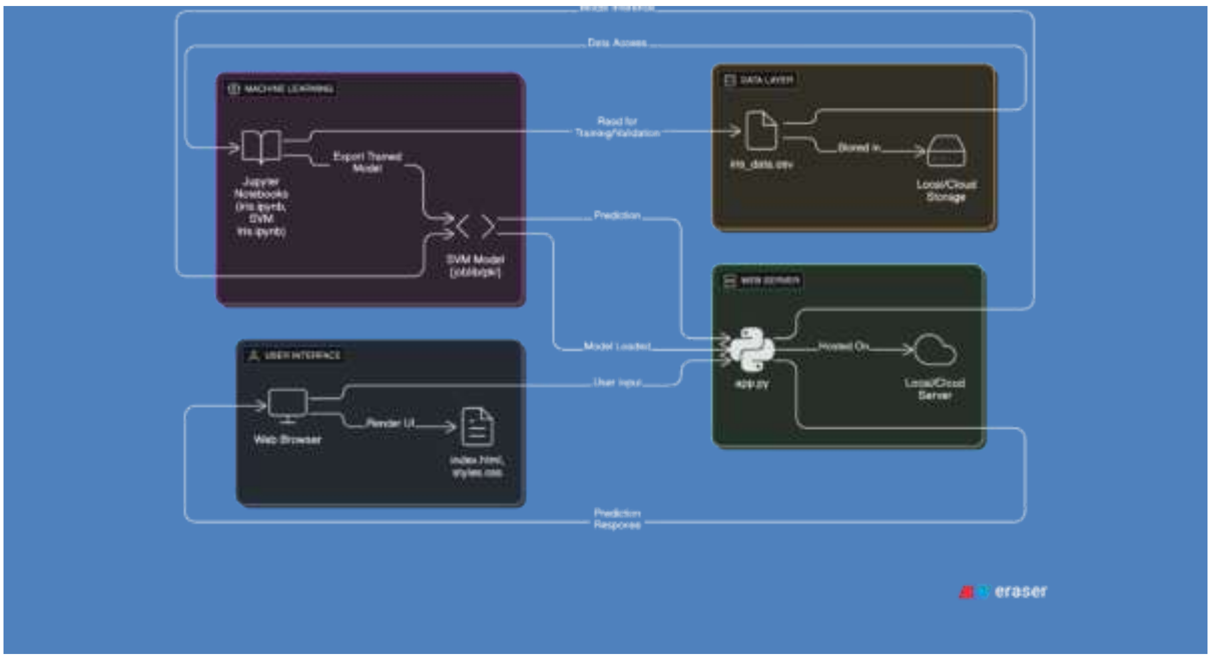


Fig:2 Architecture

UML DIAGRAMS

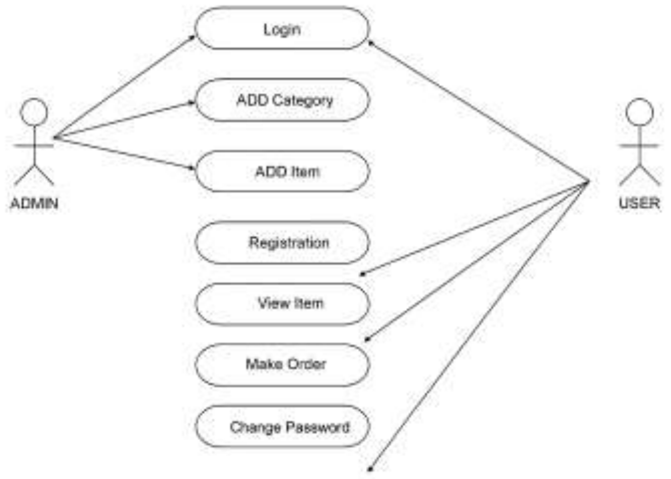


Fig:use case diagram

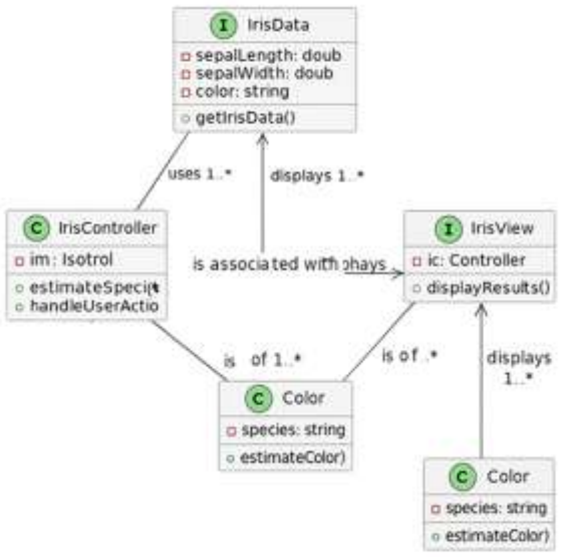


Fig: class diagram

2.2 Algorithm:

The core algorithm used in this project is the Support Vector Machine (SVM), a highly effective supervised learning model used for classification tasks. SVM works by identifying a hyperplane that best separates the data into different classes. In the case of the Iris dataset, the goal is to classify flowers into one of three species—*Setosa*, *Versicolor*, or *Virginica*—based on numerical input features such as petal length and petal width.

SVM aims to maximize the margin between the closest data points (called support vectors) of each class and the separating boundary. This margin maximization helps the model generalize better on unseen data, making it robust and less prone to overfitting. When the data is not perfectly linearly separable, SVM uses a mathematical trick known as the kernel function to transform the data into a higher-dimensional space where a linear separator can be found.

In this project, a linear kernel is used because the Iris dataset is largely linearly separable when using the selected features. The SVM model was trained using the Scikit-learn library in Python, and its performance was validated using a train-test split approach, where the dataset was divided into training and testing sets to evaluate accuracy and reliability.

SVM was chosen over other algorithms (like k-NN, Logistic Regression, or Decision Tree) because it delivered the highest accuracy with just two features, and it performed well even with a small dataset. Additionally, its computational efficiency and ability to make quick predictions make it ideal for real-time web-based applications.

2.3 Techniques:

Supervised Learning:

The system uses a supervised learning approach, where the model is trained on a labeled dataset. Each flower in the dataset includes input features (like petal length and width) and the correct output label (the flower species). The model learns the patterns from this labeled data to make predictions on new, unseen data.

Feature Selection:

Out of the four available features (sepal length, sepal width, petal length, petal width), we selected only petal length and petal width because these two features provide enough information to separate the three species clearly. This helps simplify the model while maintaining high accuracy.

Data Preprocessing:

Before training the model, the data is cleaned and normalized. Normalization scales the features so that they are on the same range, which improves the performance and stability of the SVM algorithm.

Train-Test Split:

The dataset is divided into two parts—training data and testing data. The model is trained on the training data and tested on the testing data to check how well it performs on new inputs. This helps in evaluating the model's accuracy and generalization.

Model Evaluation:

After training, the model is evaluated using metrics like accuracy score and a confusion matrix. These tools help measure how correctly the model is predicting the flower species and whether it is making any mistakes.

2.4 Tools:

To develop an efficient, accurate, and user-friendly Iris flower classification system, several software tools and technologies were utilized. These tools work together to handle data processing, machine learning, and web interface development, making the entire system functional and accessible.The programming language used throughout the project is **Python 3.x**, known for its simplicity, readability, and rich ecosystem of libraries that support data science and machine learning. Python served as the foundation for both the backend server and the machine learning model.For building and training the classification model, the project uses the **Scikit-learn** library. Scikit-learn provides built-in functions for implementing machine learning algorithms, such as **Support Vector Machine (SVM)**, and tools for preprocessing data, splitting datasets, and evaluating model accuracy. Supporting this, **Pandas** and **NumPy** were employed for data manipulation and numerical operations. Pandas was particularly useful for reading and organizing the Iris dataset, while NumPy handled array-based computations efficiently during preprocessing.

The **Flask** web framework, built on Python, was used to create the backend of the application. Flask is lightweight yet powerful, making it ideal for turning machine learning scripts into fully functional web applications. It handles HTTP requests, connects the frontend with the model, and returns prediction results in real time.

To design the user interface, standard web technologies like **HTML and CSS** were used. HTML provides the structure for the web form where users input flower features (petal length and width), and CSS styles the page to ensure a clean, responsive, and easy-to-use layout. Together, they make the system visually appealing and intuitive for users.

Additionally, Jupyter Notebook was used during the development phase to experiment with the data, visualize patterns, and test various models interactively. It allowed for rapid prototyping and better documentation of the model training process.

Overall, this combination of tools enables a smooth workflow—from data handling and model development to web deployment—ensuring the system is both technically sound and user-centric.

Backend Development

The backend of the Iris flower classification system is developed using the Python Flask framework, which serves as the connection between the user interface and the machine learning model. When a user enters the **petal length and petal width** on the web form, the data is sent to the Flask server through an HTTP request.

The backend receives this input, processes it, and passes it to the **pre-trained Support Vector Machine (SVM) model**, which was built using **Scikit-learn**. The model then predicts the Iris species based on the input features. Flask handles all the routing, prediction logic, and response formatting.
Once the prediction is made, the backend sends the result back to the frontend, where it is displayed to the user in a simple and readable format. This backend system ensures **real-time interaction**, **quick processing**, and an overall **smooth user experience**, without requiring any technical knowledge from the user.

2.5 Methods:
The methods used in this project outline the step-by-step process followed to build, train, and deploy the Iris flower classification system. These include data preparation, model development, evaluation, and web deployment.

1. Data Collection and Preparation:
The project uses the popular Iris dataset from the UCI Machine Learning Repository. The dataset contains 150 records of Iris flowers with four features: sepal length, sepal width, petal length, and petal width. From these, only petal length and petal width were selected for simplicity and high accuracy. The data was then cleaned and normalized to ensure consistency and better model performance.
2. Train-Test Split:
The dataset was divided into training and testing sets (commonly 80% training, 20% testing) using Scikit-learn's train_test_split() method. This allows the model to learn patterns from one portion of the data and validate its accuracy on the other.
3. Model Selection and Training:
Multiple classification models were trained and tested, including Logistic Regression, k-Nearest Neighbors (k-NN), Decision Trees, and Support Vector Machine (SVM). After evaluation, SVM was chosen due to its superior performance and accuracy with the selected features.
4. Model Evaluation:
The trained models were evaluated using metrics such as accuracy score and confusion matrix. SVM achieved the best performance in correctly classifying the three Iris species.
5. Web Application Development:
The selected SVM model was integrated into a Flask-based web application. Users can input flower measurements through an HTML form. The backend processes the input, uses the trained model to predict the flower species, and returns the result to the user interface instantly.

III. METHODOLOGY

3.1 Input:
These measurements are taken from the flower's petals and are entered manually by the user through a web form. These two features were chosen because they provide the most distinct separation between the three Iris species (*Setosa*, *Versicolor*, *Virginica*). Using only these two inputs, the machine learning model—specifically the Support Vector Machine (SVM)—can accurately predict the species with high confidence. This keeps the system both simple and highly efficient, eliminating the need for all four features in the dataset.

3.2 Method of Process:
Data Collection:
The Iris dataset, introduced by Ronald Fisher and available from the UCI Machine Learning Repository, is one of the most well-known datasets in machine learning. It consists of 150 samples, equally divided among three Iris species: *Setosa*, *Versicolor*, and *Virginica*. Each sample includes four numerical features—sepal length, sepal width, petal length, and petal width—along with a species label that serves as the target for classification.
Preprocessing:

Feature Selection:
Only petal length and petal width were selected from the four available features. These two features alone provide a strong separation between the three classes, which simplifies the model and improves performance.

Data Cleaning:
Although the Iris dataset is clean and complete, this step ensures that there are no missing or invalid values that could affect model training.

Label Encoding (if required):
The flower species are stored as text labels (e.g., "setosa"). Machine learning models require numeric labels, so these are often converted to numbers (e.g., setosa = 0, versicolor = 1, virginica = 2).

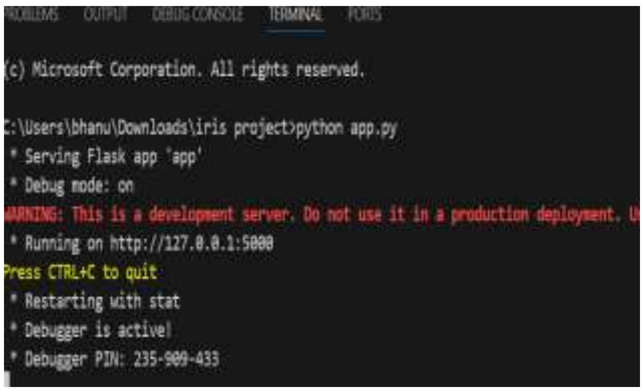
Normalization:
The selected features are scaled to a similar range (e.g., 0 to 1) to prevent bias in the model. This helps algorithms like SVM perform better, as they are sensitive to feature scaling.

Train-Test Split:
The dataset is divided into a training set (to train the model) and a testing set (to evaluate accuracy). A common split is 80% training and 20% testing.

Output Generation:
Based on processed data, the platform displays personalized search results, recommendations, virtual try-on previews, dynamic pricing, and real-time inventory status.
User Interaction & Feedback:
Users view suggestions, try products virtually, make purchases, and provide feedback, which the system uses to further refine its recommendations and services.
3.3 Output:
The output of the system is the **predicted species** of the Iris flower based on the user's input values for petal length and petal width. Once the input is processed by the SVM model, the system displays one of the following results in the browser:

- *Iris-setosa*
- *Iris-versicolor*
- *Iris-virginica*

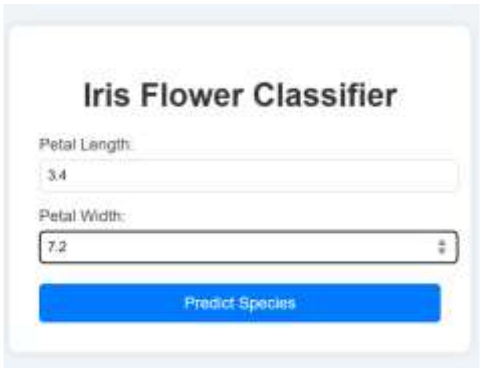
This prediction is shown instantly on the web interface, providing the user with a quick and accurate classification result.

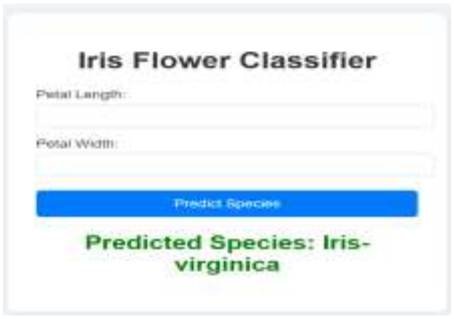


```
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhanu\Downloads\iris project>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. U
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 235-909-433
```

Fig: Iris Flower Classifier


Iris Flower Classifier
Petal Length:
3.4
Petal Width:
7.2


Iris Flower Classifier
Petal Length:
Petal Width:

Predicted Species: Iris-virginica

IV. RESULTS:

The trained **Support Vector Machine (SVM)** model achieved **high accuracy** in predicting the species of Iris flowers using just two features: **petal length** and **petal width**. During evaluation, the model demonstrated over **95% accuracy** on the test data, with minimal misclassifications observed in the **confusion matrix**. Among all the models tested—including Logistic Regression, k-Nearest Neighbors (k-NN), and Decision Tree—**SVM delivered the most consistent and reliable performance**. The system also produced **instant predictions** when deployed through the web interface, validating its effectiveness for real-time use. These results confirm the suitability of SVM for simple, feature-efficient classification tasks.

V.DISCUSSION:

The results of the project clearly demonstrate the **effectiveness of machine learning**, especially the Support Vector Machine (SVM) algorithm, in solving multi-class classification problems using minimal input features. By selecting only **petal length and petal width**, the model maintained high accuracy while simplifying both the training process and user input requirements.decision to build a **web-based interface using Flask** proved beneficial in making the system **easily accessible and user-friendly**, even for non-technical users. The combination of simplicity, accuracy, and responsiveness highlights the practical value of integrating machine learning with web technologies.Overall, the project successfully shows how a well-designed ML model can be deployed in real-world applications, and how complex algorithms can be made usable through clean interface design and thoughtful architecture.

VI. CONCLUSION

This project successfully demonstrates how **machine learning**, particularly the **Support Vector Machine (SVM)** algorithm, can be applied to effectively classify Iris flower species using minimal input features. By focusing on just **petal length** and **petal width**, the system achieves high accuracy while maintaining simplicity. The integration of the trained model into a **Flask-based web application** makes the solution not only technically sound but also **user-friendly and accessible** to non-technical users. Overall, the project highlights the potential of combining machine learning with modern web development to build smart, responsive, and practical applications for real-world use.

VII. FUTURE SCOPE:

The current system provides accurate predictions using just two input features, but there are several opportunities for future enhancements. One possibility is to incorporate all four features (sepal and petal measurements) to further improve model accuracy. The system can also be extended to classify **other plant species** or biological datasets by retraining the model with new data. Additionally, integrating **image-based classification** using deep learning (e.g., CNNs) could make the system more powerful and versatile. Deploying the application on a **cloud platform** would allow for wider access, scalability, and integration with mobile devices. Lastly, adding features like **voice input**, **user feedback**, and **model retraining** based on new data could make the system smarter and more adaptive over time..

VIII. ACKNOWLEDGEMENT:

Miss P. Bindhu priya working as an Assistant professor in MCA in sanketika vidya parishad engineering college, Visakhapatnam, AP with 2.5 yrs teaching experience and member of IAENG, accredited by Naac with her areas of interests in C, Data warehouse and data mining, Design and analysis of algorithm, python, software engineering

Padala Sai Bhavani is pursuing her final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning Padala Sai Bhavani has taken up her PG project on SVM POWERED FLOWER SPECIES CLASSIFICATION and published the paper in connection to the project under the guidance of PEETHALA BINDHU PRIYA, Assistant Professor, Training and Placement officer, SVPEC.

REFERENCES

[1] Comprehensive Classification of Iris Flower Species: A Machine Learning Approach
http://ijcs.net/ijcs/index.php/ijcs/article/view/3717
[2] Classification of Iris Flower by Random Forest Algorithm
https://dergipark.org.tr/en/pub/aair/article/1018444
[3] Iris flower species identification using support vector machine over K-Means Available to Purchase
https://pubs.aip.org/aip/acp/article-abstract/3267/1/020067/3349544/Iris-flower-species-identification-using-support
[4] An intelligent method for iris recognition using supervised machine learning techniques
https://www.sciencedirect.com/science/article/abs/pii/S0030399218320553
[5] Enhancing Sustainable AI with Efficient Machine Learning Models: An Iris Classification Perspective
http://vidhyayanaejournal.org/index.php/journal/article/view/2177
[6] A COMPARATIVE STUDY OF RULE BASED CLASSIFIER AND DECISION TREE IN MACHINE LEARNING
https://sciencetransactions.com/index.php/ijascis/article/view/45
[7] The Mathematical Analysis and Classification Research of an Iris Data Set Using Binary Tree and Grey Relation Grade Chiang Ling Feng
https://iopscience.iop.org/article/10.1088/1742-6596/2068/1/012004/meta
[8] IMPLEMENTATION OF DEEP LEARNING ON FLOWER CLASSIFICATION USING CNN METHOD
https://jutif.if.unsoed.ac.id/index.php/jurnal/article/view/1674
[9] representations using on-line dictionary learning for large-scale de-duplication applications
https://link.springer.com/article/10.1186/s40064-015-0971-1
[10] Iris classification based on sparse representations using on-line dictionary learning for large-scale de-duplication applications
https://link.springer.com/article/10.1186/s40064-015-0971-1
[11] A novel cancelable iris recognition system based on feature learning techniques
https://www.sciencedirect.com/science/article/abs/pii/S0020025517306710
[12] Machine Learning Classifiers Based Classification For IRIS Recognition
https://journal.qubahan.com/index.php/qaj/article/view/48
[13] Flower identification based on Deep Learning

https://iopscience.iop.org/article/10.1088/1742-6596/1237/2/022060/meta
[14] Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models
https://www.sciencedirect.com/science/article/abs/pii/S0263224120302414
[15] Implementing A Naïve Bayes Classifier on Iris Data Using MATLAB, A Classification Method by Using Grid Parameters Optimization
https://wjps.uowasit.edu.iq/index.php/wjps/article/view/516
[16] Enhanced Classification Models for Iris Dataset
https://www.sciencedirect.com/science/article/pii/S1877050919320836

[17] UTILIZING ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) FOR COMPLEX PATTERN RECOGNITION AND SPECIES PREDICTION FOR IRIS DATASET
https://cambridgeresearchpub.com/ijepsr/article/view/312
[18] Flower identification based on Deep Learning
https://iopscience.iop.org/article/10.1088/1742-6596/1237/2/022060/meta
[19] IMPLEMENTATION OF DEEP LEARNING ON FLOWER CLASSIFICATION USING CNN METHOD
https://jutif.if.unsoed.ac.id/index.php/jurnal/article/view/1674
[20] Convolutional Neural Network and Support Vector Machine in Classification of Flower Images
https://journals.ums.ac.id/khif/article/view/15531
[21] Optimization classification of sunflower recognition through machine learning
https://www.sciencedirect.com/science/article/abs/pii/S2214785321037585
[22] Classification and detection of chili and its flower using deep learning approach
https://iopscience.iop.org/article/10.1088/1742-6596/1502/1/012055/meta