WEB-X CA

Laksh Sodhai: D15A-57

FLOWER PREDICTION

Introduction:

The Flower Prediction project is a machine learning-based web application designed to classify flowers into their respective species based on physical characteristics such as sepal length, sepal width, petal length, and petal width. Leveraging the popular Iris dataset, this system utilizes supervised learning algorithms like Decision Tree or K-Nearest Neighbors to build a predictive model with high accuracy. The primary objective is to demonstrate the practical implementation of machine learning in the field of botany and data science, while offering a hands-on experience in integrating a trained model within a real-time web environment.

Project Description and Requirement Gathering

Project Description

The Flower Prediction project classifies flowers using machine learning. It takes sepal and petal dimensions as input and predicts the flower species using algorithms like Decision Tree or K-Nearest Neighbors.

The backend is built with Flask in Python, integrating a pre-trained model using scikit-learn. The frontend is interactive, allowing users to input measurements and view the predicted species instantly.

Requirement Gathering

Functional requirements include allowing users to input sepal length, sepal width, petal length, and petal width. The system must process these inputs through a trained ML model and return the predicted flower species.

Non-functional requirements include a clean interface, Python with Flask and scikit-learn support, smooth local machine operation, and accurate, fast predictions.

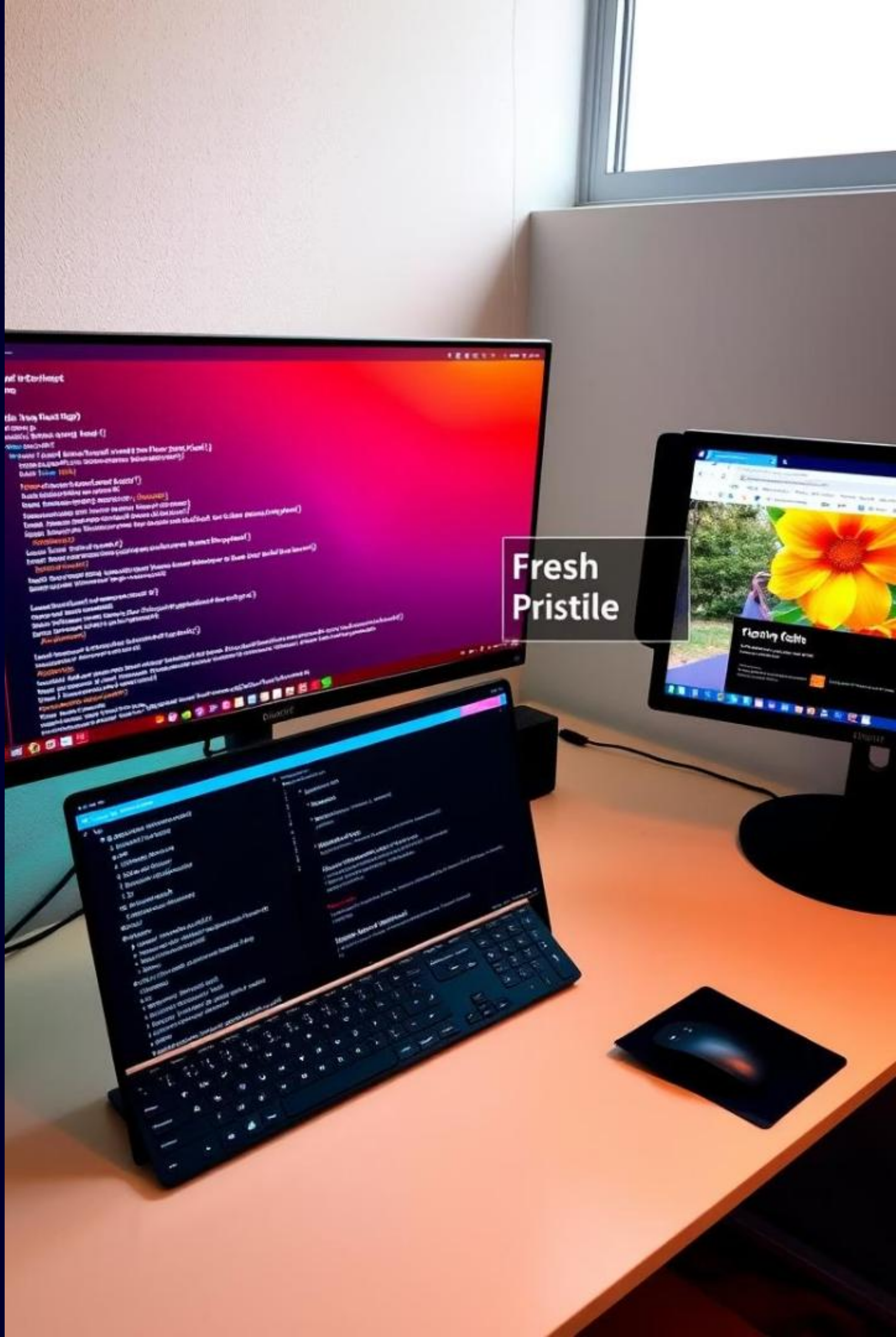
System Requirements

Hardware Requirements

- Processor: Intel Core i5 / AMD Ryzen 5 or higher
- RAM: Minimum 8GB (16GB recommended)
- Storage: At least 1GB free space (256GB SSD recommended)
- Network: Stable internet connection

Software Requirements

- Operating System: Windows 10/11, macOS 10.15+, or Ubuntu 20.04+
- Code Editor: Visual Studio Code or compatible IDE
- Version Control: Git 2.25+
- Python: Version 3.8 or higher



Technologies Used

Development

VS Code, Postman, Git

Frontend

HTML/CSS/Typescript (or Streamlit/Flask Templates)

Backend

Flask (Python 3.8+)

ML Model

Scikit-learn

The Flower Prediction project utilizes a variety of technologies for development, frontend, backend, and machine learning model implementation. These technologies ensure a robust and efficient system for flower species classification.

Setup Instructions

1

Install Python 3.8+

Install Python 3.8 or higher from python.org. Add Python to PATH during installation. Verify with python --version and pip --version.

2

Flask and Dependencies

Install necessary packages by running pip install -r requirements.txt in your project directory. Ensure requirements.txt is up to date.

3

ML Model Setup

Ensure your trained ML model (e.g., a .pkl file) is saved in the appropriate directory. The Flask backend will load this model at runtime.

4

Backend Setup

Navigate to backend folder, create a virtual environment, install dependencies, and start the Flask server.

Follow these steps to set up the Flower Prediction project. Ensure Python, Flask, and the ML model are correctly installed and configured for seamless operation.

Project Structure and Architectural Diagrams

Project Structure

The project structure includes folders for the backend (Flask API), frontend (HTML/CSS/JS), and ML model. This modular design ensures scalability and easy maintenance.

Architectural Diagrams

The class-based diagram illustrates the relationships between different classes and components in the system, providing a clear overview of the system's architecture.

Future Scope:

The Flower Prediction system can be expanded in several ways. Integration with mobile applications could make it accessible on-the-go for botanists, researchers, and students. Advanced models like deep learning (e.g., CNNs) could be used to predict species from flower images, enhancing accuracy and usability. The system can be extended to include a wider variety of plant species beyond the Iris dataset. Additionally, real-time data collection through IoT sensors in smart gardens can be incorporated. Integration with cloud platforms can allow for scalable deployment and analytics.

Conclusion:

The Flower Prediction project demonstrates how machine learning can be effectively applied to classify flower species based on measurable features. By utilizing the Iris dataset and implementing a user-friendly web interface, the system provides quick and accurate predictions. It bridges the gap between data science and real-world applications in botany and education. The integration of Flask as a backend framework ensures lightweight, efficient performance. This project not only highlights the power of predictive analytics but also emphasizes the importance of user-centric design. Through modular development, the system is easy to maintain and extend. With further enhancements, such as image-based predictions and mobile compatibility, the project holds strong potential for future growth. Overall, it serves as a practical introduction to applied machine learning.