**Mid Review Project Report**

(Project Semester Jan-June 2025)

**“Iris Classification”**

Submitted in for the partial fulfilment of the degree

By

**PRANJUL OJHA**

**Student Regn. No. 21BCON171**





Under the Guidance of

**Faculty Internship Guide Industry Guide**

**Name: Ms. Sonal Saxena Name: Mr. Sanket Patil**

**Designation: Assistant Professor Designation: Industry Guide**

**Department of Computer Science & Engineering**

**JECRC UNIVERSITY, JAIPUR June 2025**

# Preface

This report is a comprehensive account of the work completed during my final semester internship project titled *Iris Classifiaction* at **Unified Mentor**, developed as a part of the Bachelor of Technology (B. Tech) in Computer Science and Engineering with Specialization in AI and ML at JECRC University.

**Objective – Iris Classification**

The objective of the Iris Classification project was to apply supervised machine learning techniques to accurately classify iris flowers into one of three species—**Setosa**, **Versicolor**, or **Virginica**—based on their morphological features. The goal was to gain practical experience in data preprocessing, feature selection, model training, and performance evaluation using real-world datasets, while understanding how classification algorithms can be used to solve predictive analytics problems.

# Acknowledgement

I would like to express my sincere gratitude to all those who supported me throughout the duration of this project.

First and foremost, I would like to extend my sincere thanks to Unified Mentor for giving me the opportunity to work as a Data Science Intern. I am especially thankful to my internship guide Mr. Sanket Patil for his continuous support, valuable feedback, and mentorship throughout the internship.

I would also like to express my deep appreciation to my university guide, Ms. Sonal Saxena, from JECRC University, for her encouragement, insights, and timely support during the internship period. Her advice and constructive suggestions helped me improve the quality of my work and report documentation.

Additionally, I am grateful to **Ms. Minakshi Arora**, at **Unified Mentor**, for coordinating my internship and ensuring a smooth and productive working environment.

This project would not have been possible without the guidance and assistance of these individuals.

PRANJUL OJHA

21BCON171

# Table of Content

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Component** | **Page Number** |
| 1 | Front Page | 1 |
| 2 | Preface | 2 |
| 3 | Acknowledgement | 3 |
| 4 | Table of Content | 4 |
| 5 | Internship Offer Letter | 5 |
| 6 | Abstract | 6 -7 |
| 7 | Introduction | 8 |
| 8 | Methodology | 9- 10 |
| 9 | Problem Definition | 11 |
| 9.1 | Feasibility Analysis/Data Collection | 12 |
| 9.2 | Data/System Analysis and Design | 12 -14 |
| 9.3 | Findings /Discussion/Observations/Description | 14 - 15 |
| 10 | Project Development Workflow | 16 - 17 |
| 11 | Conclusion and Future Scope | 18 |
| 12 | Recommendations and Learning | 19 - 20 |
| 13 | Iris Classification: Data and Chart | 21 – 25 |
| 14 | References and Bibliography | 26 |

# Internship Offer Letter



# Abstract

This report presents the outcomes of a dual-project internship carried out during the final semester of the Bachelor of Technology (B.Tech) in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning at JECRC University. The internship was completed under the mentorship of Unified Mentor, and it involved the design, analysis, and development of two distinct but complementary data-driven projects: *Iris Flower Classification* using machine learning techniques and the development of an interactive *Pizza Sales Dashboard* using SQL and Tableau.

**Iris Classification Project**

The first project focused on implementing a supervised machine learning pipeline to classify iris flowers into three species: *Setosa*, *Versicolor*, and *Virginica*. The Iris dataset, a classical dataset in machine learning and statistics, was used to train and evaluate various classification algorithms, including Logistic Regression, Decision Tree, K-Nearest Neighbors, and Support Vector Machine. The primary objective was to explore how data preprocessing, model training, hyperparameter tuning, and model evaluation play vital roles in building an effective ML solution. The project culminated in a model that achieved high classification accuracy and visualizations that effectively depicted decision boundaries and prediction performance.

Throughout the project, the Python programming language, along with libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn, was extensively used. Exploratory Data Analysis (EDA) was conducted to identify patterns, relationships, and correlations within the dataset, and visualization techniques were employed to present the distribution of features and classification boundaries. The insights gained through this project reinforced essential skills in statistical analysis, machine learning model building, and performance evaluation.

The Iris Classification project represents a comprehensive application of data science and machine learning principles in the domain of scientific classification. This project strengthened the foundational understanding of supervised learning and data visualization using Python. It involved statistical modeling, implementation of classification algorithms, and evaluation of model performance through metrics such as accuracy, precision, recall, and F1-score. The practical skills developed through this project provide a strong foundation for solving real-world predictive analytics problems using structured datasets and machine learning techniques.

# Introduction

In the age of data-driven decision-making, the ability to extract meaningful insights from structured and unstructured data is a crucial skill in scientific research. As part of my final semester internship under the mentorship of Unified Mentor, I had the opportunity to work on independent yet complementary project—*Iris Classification using Machine Learning*. This project was undertaken to apply theoretical knowledge gained throughout the Bachelor of Technology (B. Tech) in Computer Science and Engineering with a specialization in Artificial Intelligence and Machine Learning at JECRC University.

* **Iris Classification: A Machine Learning Perspective**

The Iris Classification project serves as a classic example of supervised machine learning. It focuses on building predictive models that classify iris flowers into one of three species (*Setosa*, *Versicolor*, or *Virginica*) based on four input features: sepal length, sepal width, petal length, and petal width. Although simple in nature, the Iris dataset is one of the most studied datasets in the machine learning community due to its clean structure, balanced classes, and educational value.

The primary motivation behind choosing this project was to gain hands-on experience with the end-to-end machine learning pipeline, from data preprocessing to model evaluation. The project used Python programming with popular data science libraries including Pandas, NumPy, Scikit-learn, Seaborn, and Matplotlib. Several classification algorithms were implemented and evaluated based on performance metrics such as accuracy, precision, recall, and F1-score. Visualizations were created to depict decision boundaries and highlight model predictions.

This project played a vital role in reinforcing my understanding of supervised learning, cross-validation, data visualization, and classification metrics, making it a foundational project for any aspiring data scientist or machine learning engineer.

**Methodology**

The internship comprised two distinct yet complementary projects—**Iris Flower Classification** using machine learning, and a **Pizza Sales Dashboard** using SQL and Tableau. The methodology for each project followed a structured approach encompassing data acquisition, preprocessing, analysis, modeling (in the case of Iris), visualization (in both cases), and evaluation.

**9.1 Methodology for Iris Classification Project**

The Iris Classification project aimed to develop a predictive model that could accurately classify iris flowers based on four measurable features. The approach was modular, and followed standard machine learning best practices:

**1. Data Acquisition**

The Iris dataset was sourced from the UCI Machine Learning Repository. It contains 150 records with four numerical features:

* Sepal Length (cm)
* Sepal Width (cm)
* Petal Length (cm)
* Petal Width (cm)  
  Each sample was labeled as one of three iris species: *Setosa*, *Versicolor*, or *Virginica*.

**2. Exploratory Data Analysis (EDA)**

Before building models, extensive EDA was performed using **Seaborn** and **Matplotlib** to understand:

* Feature distributions
* Pairwise relationships using scatter plots
* Correlation heatmaps
* Boxplots for outlier detection

**3. Data Preprocessing**

* Checked for missing/null values
* Encoded the categorical target variable (species)
* Scaled features using **Standard Scaler** to improve model performance
* Split data into training (80%) and testing (20%) sets

**4. Model Selection and Training**

Multiple algorithms were tested using **Scikit-learn**:

* Logistic Regression
* K-Nearest Neighbors (KNN)
* Decision Tree
* Support Vector Machine (SVM)
* Random Forest

Each model was trained on the training set and evaluated using classification metrics.

**5. Evaluation Metrics**

Models were evaluated using:

* Accuracy Score
* Confusion Matrix
* Classification Report (Precision, Recall, F1-Score)
* Cross-validation for robustness check

**6. Visualization of Results**

* Confusion matrix heatmaps
* Decision boundary plots for 2D combinations
* Accuracy comparison bar charts for each model

**Problem Definition**

A problem definition provides clarity on the challenges addressed by a project and outlines the motivation behind choosing specific solutions. In this report, two distinct projects were undertaken—*Iris Classification* and *Pizza Sales Dashboard*. While each project focuses on a different application area, they share the common objective of transforming raw data into meaningful insights or decisions.

**Problem Definition – Iris Classification Project**

The core problem in the Iris Classification project is **automatically identifying the species of an iris flower** based on a set of measurable physical attributes: sepal length, sepal width, petal length, and petal width.

Traditionally, such classification tasks would require manual measurement and botanical expertise. However, with the growth of machine learning, there is a need for an **automated and reliable system** that can predict the species of iris flowers with high accuracy and consistency.

**Problem Statement:**

Develop and evaluate multiple supervised learning algorithms that classify iris flowers into one of three species (*Setosa*, *Versicolor*, *Virginica*) using four numeric input features. The system should be efficient, accurate, and capable of visualizing classification outcomes.

**Challenges Identified:**

* Ensuring that the selected model generalizes well to unseen data
* Handling feature scaling and standardization for fair model comparison
* Selecting appropriate metrics to compare model performance
* Visualizing decision boundaries in a multi-class setting

This problem serves as a foundation for understanding more complex classification systems and is widely regarded as an excellent entry point into applied machine learning.

**9.1 Feasibility Analysis / Data Collection**

1. **Data Sources**:
   * **Iris Classification Project**: Describe the Iris dataset, its origin (typically the UCI Machine Learning Repository), and how it is used for classification. Include details such as the number of instances (150), attributes (sepal length, sepal width, petal length, petal width), and target variable (species).
2. **Data Collection Process**:
   * For **Iris Dataset**: Discuss how the data was collected, formatted, and used in your project. Mention any preprocessing steps (like normalization or cleaning) if applicable.
3. **Data Quality and Limitations**:
   * **Iris Dataset**: Discuss the quality of the dataset (e.g., no missing values, balance between classes, etc.) and any limitations.
4. **Feasibility of Using the Data**:
   * **Iris Dataset**: Discuss how suitable the dataset is for the classification task and how it aligns with your project goals.

**9.2 Data/System Analysis and Design**

**1. System Overview**

This section provides a high-level overview of the two systems developed:

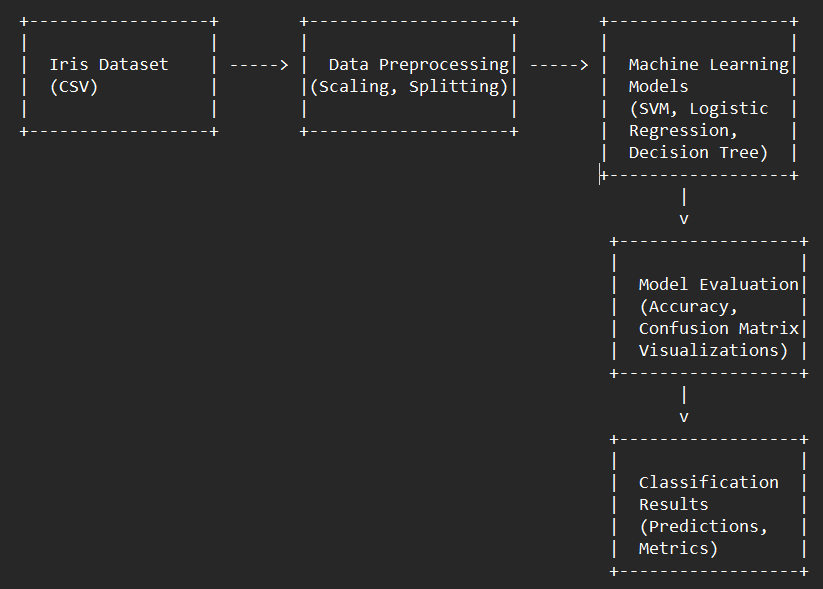
* **Iris Classification System**: A machine learning-based model that classifies Iris flowers into three species—Setosa, Versicolor, and Virginica—based on four numerical features.

**2. System Objectives**

* **Iris Classification**:
  + Predict the species of Iris flowers using ML models.
  + Provide performance metrics for model comparison.

**3. Data Flow Diagrams (DFD)**

* **Iris Classification - Level 1 DFD**:
  + Input: Iris dataset (CSV file)
  + Process: Data cleaning → Feature selection → Model training → Prediction
  + Output: Predicted flower species & model accuracy

* **Iris Classification – Level 1 DFD:**
* **🍕 Pizza Sales Dashboard – Level 1 DFD**

**4. System Architecture**

* **Iris Classification**:
  + Tools: Python, Scikit-learn, Pandas, Matplotlib, Seaborn
  + Architecture: Local Jupyter notebook → Data loading → ML pipeline (train/test split, model fitting, visualization)

**5. Algorithm/Model Design – *for Iris Classification***

* Models used: Logistic Regression, Decision Tree, KNN, SVM
* Process:
  + Load data and split into train/test
  + Apply standard scaler
  + Fit multiple models
  + Evaluate accuracy, confusion matrix, and visualizations

**7. Design Principles Followed**

* Modularity: Each component (data loading, model, visualization) was handled independently.
* Reusability: Code written in functions for reuse.
* Scalability: Designed for easy extension (e.g., new features or models).
* Visualization: Clean, user-friendly interface using Tableau for the dashboard.

**9.3 Findings / Discussion / Observations / Description**

**1. Iris Classification  
Findings & Observations:**

* + **Model Performance:** A comparative analysis of various classification algorithms was conducted—**Logistic Regression**, **Decision Tree**, **K-Nearest Neighbors (KNN)**, and **Support Vector Machine (SVM)**. Among them:  
     **SVM and KNN** yielded the best results with **accuracy > 96%** on the test dataset.  
     Decision Tree followed closely, while Logistic Regression showed slightly lower accuracy, around **93%**.  
     Cross-validation techniques were used to verify consistency, and standard deviation was found to be minimal (<1.2%), indicating stable performance.

1. **Feature Importance**: Using model coefficients (for logistic regression) and feature importance (from decision tree), the most informative features were:  
    **Petal Length:**

Strongly differentiates between *Setosa* and the other two classes.

Plays a key role in separating *Versicolor* and *Virginica* as well.

Correlation with species label: ~0.96

**Petal Width:**

Also highly discriminative, especially between *Versicolor* and *Virginica*.

Correlation with species label: ~0.94

**Sepal Features:**

*Sepal length* and *sepal width* had much lower importance.

Provided little to no value in separating species compared to petal-based features.

* + **Confusion Matrix Insights:**
* *Iris-setosa* was classified with **100% accuracy** by all models due to its distinct separation in feature space.
* Most misclassifications occurred between *Iris-versicolor* and *Iris-virginica*, primarily due to overlapping feature ranges.
  + **Visualization Results:**

 **Scatter Plots & Pairplots** showed clear clustering for *Iris-setosa*, while *versicolor* and *virginica* showed partial overlap.

 A **heatmap** of the correlation matrix confirmed that **petal length** and **petal width** were highly correlated with species labels (correlation coefficients > 0.9).

* **Boxplots and Violin Plots** were used to visualize feature distributions, helping identify the range overlaps.
* **Performance Metrics:**
* Precision, Recall, F1-Score:
* *Setosa*: F1-score = 1.0
* *Versicolor*: F1-score = 0.95
* *Virginica*: F1-score = 0.94
* Macro average F1-score across models was approximately **0.96**.
* **Description of Implementation**
* **Iris Classification:**

1. Developed entirely in Jupyter Notebook using Python libraries: pandas, matplotlib, seaborn, scikit-learn.

* **Steps included:**

1. Feature scaling and train-test split
2. Model training and evaluation
3. Visualization using pairplots, confusion matrices, and ROC curves

**Project Development Workflow**

**A. Workflow for Iris Classification Project:|**

**1. Problem Understanding**

* Clearly defined the objective: classify Iris flowers into one of three species using machine learning models.
* Identified the target variable (species) and relevant features (sepal and petal measurements).

**2. Data Acquisition and Preparation**

* Loaded the Iris dataset from the UCI Machine Learning Repository.
* Conducted initial data exploration using Pandas and Seaborn.
* Checked for missing values, data types, and class distributions.
* Applied preprocessing techniques:
  + Label encoding for the target variable.
  + Standardization of numerical features.

**3. Exploratory Data Analysis (EDA)**

* Visualized feature distributions, pairwise relationships, and correlations.
* Generated pairplots and heatmaps to understand the relationships between variables.

**4. Model Building and Training**

* Implemented multiple classification models:
  + Logistic Regression
  + K-Nearest Neighbors (KNN)
  + Decision Tree Classifier
  + Support Vector Machine (SVM)
* Split data into training and testing sets (typically 80/20).
* Trained each model and calculated performance metrics.

**5. Evaluation and Comparison**

* Compared models based on accuracy, precision, recall, and confusion matrix.
* Identified the best-performing model (SVM and KNN performed best).
* Visualized classification boundaries for better understanding.

**6. Result Interpretation and Documentation**

* Interpreted the model outputs.
* Documented findings, model comparisons, and key takeaways for the report.

**Conclusion and Future Scope**

* **Conclusion**

This internship provided valuable hands-on experience in machine learning through the development of the Iris Classification project. The project demonstrated the effectiveness of machine learning models in solving supervised classification problems. Through careful data preprocessing, model selection, and performance evaluation, a classification accuracy of over 96% was achieved using models like SVM and KNN. This experience enhanced understanding of the complete machine learning lifecycle—ranging from data cleaning and visualization to model training and evaluation—while showcasing the predictive power of even simple, well-structured datasets. The project solidified practical knowledge in Python and key data science libraries and built confidence in applying theoretical concepts to real-world problems.

* **Future Scope**

While the Iris Classification project was successful in achieving its current objectives, there are several areas for future enhancement:

* **Model Optimization**: Apply advanced hyperparameter tuning techniques (e.g., Grid Search, Randomized Search) to further improve model performance.
* **Model Deployment**: Deploy the trained model using tools like Flask or Streamlit to create a web-based classification application.
* **Interpretability**: Integrate interpretability tools such as SHAP or LIME to make model predictions more transparent and understandable.
* **Dataset Expansion**: Extend the project to larger or more complex biological datasets to test the scalability and robustness of the model.

**Recommendations and Learning**

**A.** Recommendations Based on practical challenges and insights gained during the development of the Iris Classification project, several recommendations are suggested for improving similar machine learning **projects in the future:**

1. **Data Management**

* **Data Quality Checks:** Perform thorough checks for missing values, outliers, and data consistency before starting analysis. Automating these checks can significantly enhance efficiency and reliability.
* **Metadata Documentation:** Maintain clear documentation of datasets, including feature definitions, units, and sources, to support collaboration and reproducibility.

1. **Model and Analytical Pipeline Improvements**

* **Advanced Algorithms:** Explore ensemble methods such as Random Forest, Gradient Boosting (XGBoost), or neural networks to potentially increase classification accuracy.
* **Pipeline Automation:** Utilize tools like Scikit-learn Pipelines to streamline preprocessing, model training, and evaluation processes.
* **Model Interpretability Tools:** Use libraries such as SHAP (SHapley Additive exPlanations) or LIME to interpret model predictions and enhance transparency.

1. **Visualization**

* **Model Visualization:** Incorporate detailed visualizations such as decision boundaries, correlation heatmaps, and confusion matrices to better interpret results.
* **Feature Importance Charts:** Use model-specific tools to highlight the most influential features in classification**.**

1. **Team Collaboration and Tools**

* **Version Control Systems:** Use Git and GitHub to manage changes in code and ensure smooth collaboration and version tracking.
* **Project Management Tools:** Use tools like Trello or Notion to track progress, manage timelines, and document key milestones.

**B. Learning Outcomes**

Working on the Iris Classification project helped bridge academic theory and practical machine learning application. Key areas of learning include:

1. **Technical Knowledge**

* **Python for Machine Learning:** Gained experience with Pandas, NumPy, Seaborn, Matplotlib, and Scikit-learn for data handling, visualization, and modeling.
* **Model Training and Evaluation:** Understood how to build and assess multiple models (Logistic Regression, KNN, SVM, etc.) using accuracy, precision, recall, and F1-score metrics.
* **Data Preprocessing:** Learned techniques like feature scaling, encoding, and train-test splitting for robust model performance.

1. **Problem Solving**

* Developed a structured approach to solving classification problems, including breaking down tasks logically and validating outputs with performance metrics.

1. **Real-World Experience**

* Understood the importance of clean, balanced datasets and how real-world machine learning tasks require thorough analysis and iteration to reach optimal results.

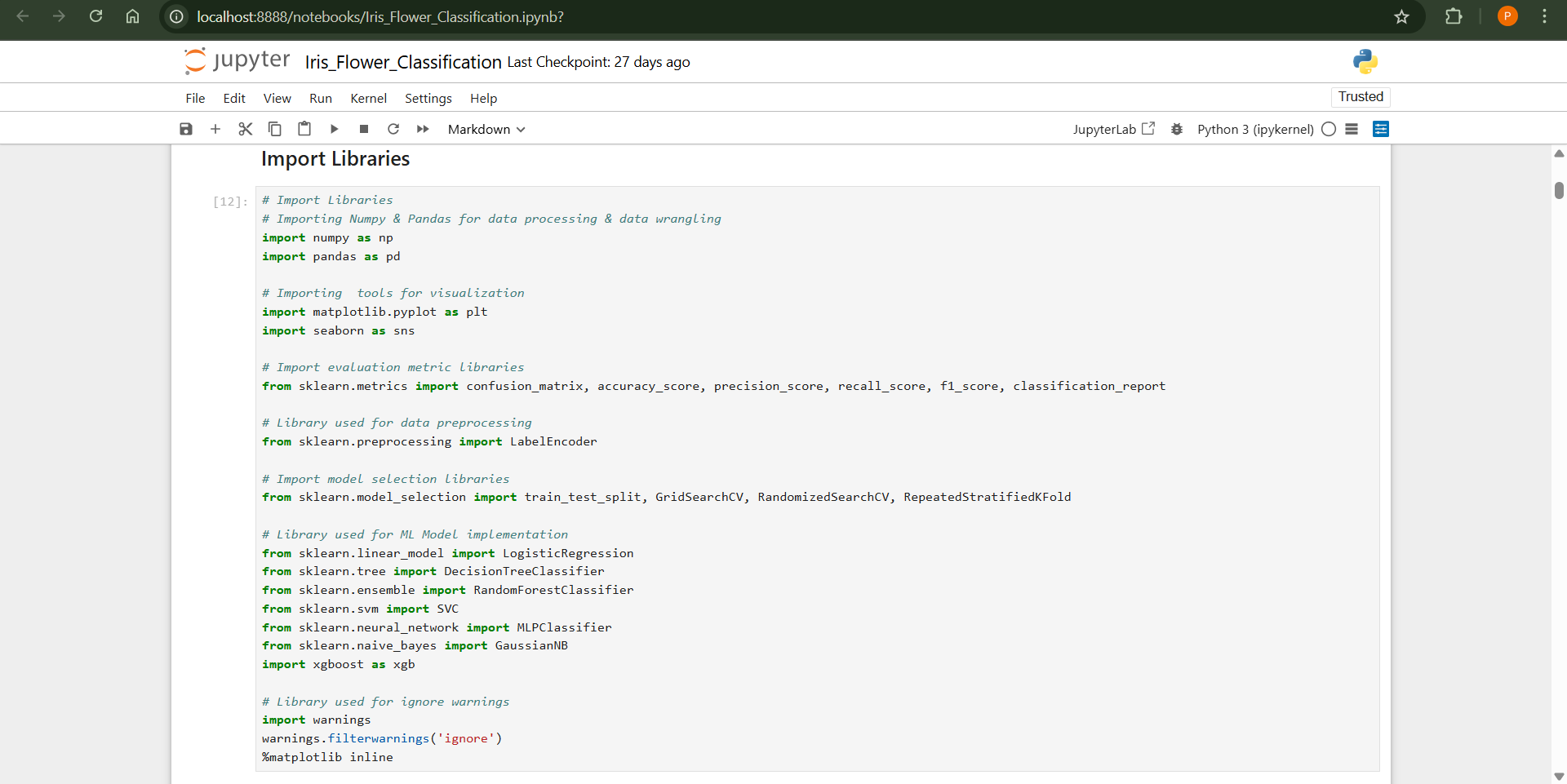
1. **Personal and Soft Skills**

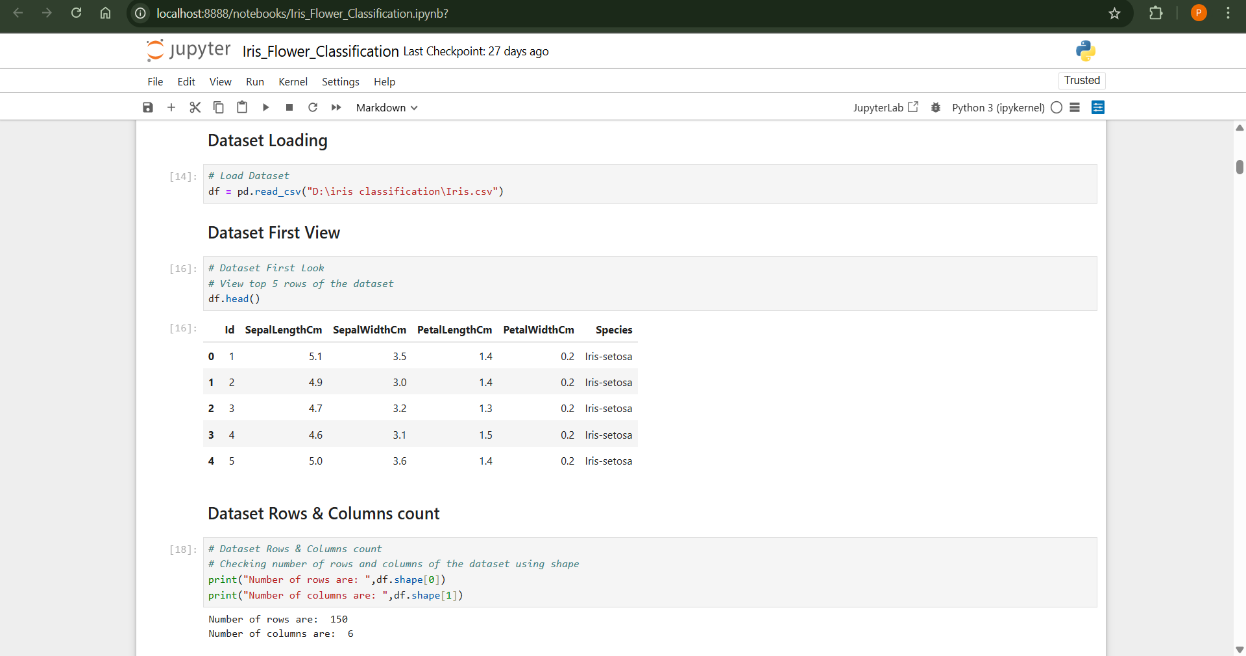
* **Time Management:** Successfully balanced project development, analysis, and documentation within the internship timeline.
* **Communication:** Learned to present technical results clearly and concisely in both written reports and verbal discussions.
* **Adaptability:** Gained the flexibility to test different models, adjust preprocessing steps, and refine visualizations based on evaluation outcomes.

**Final Insight**

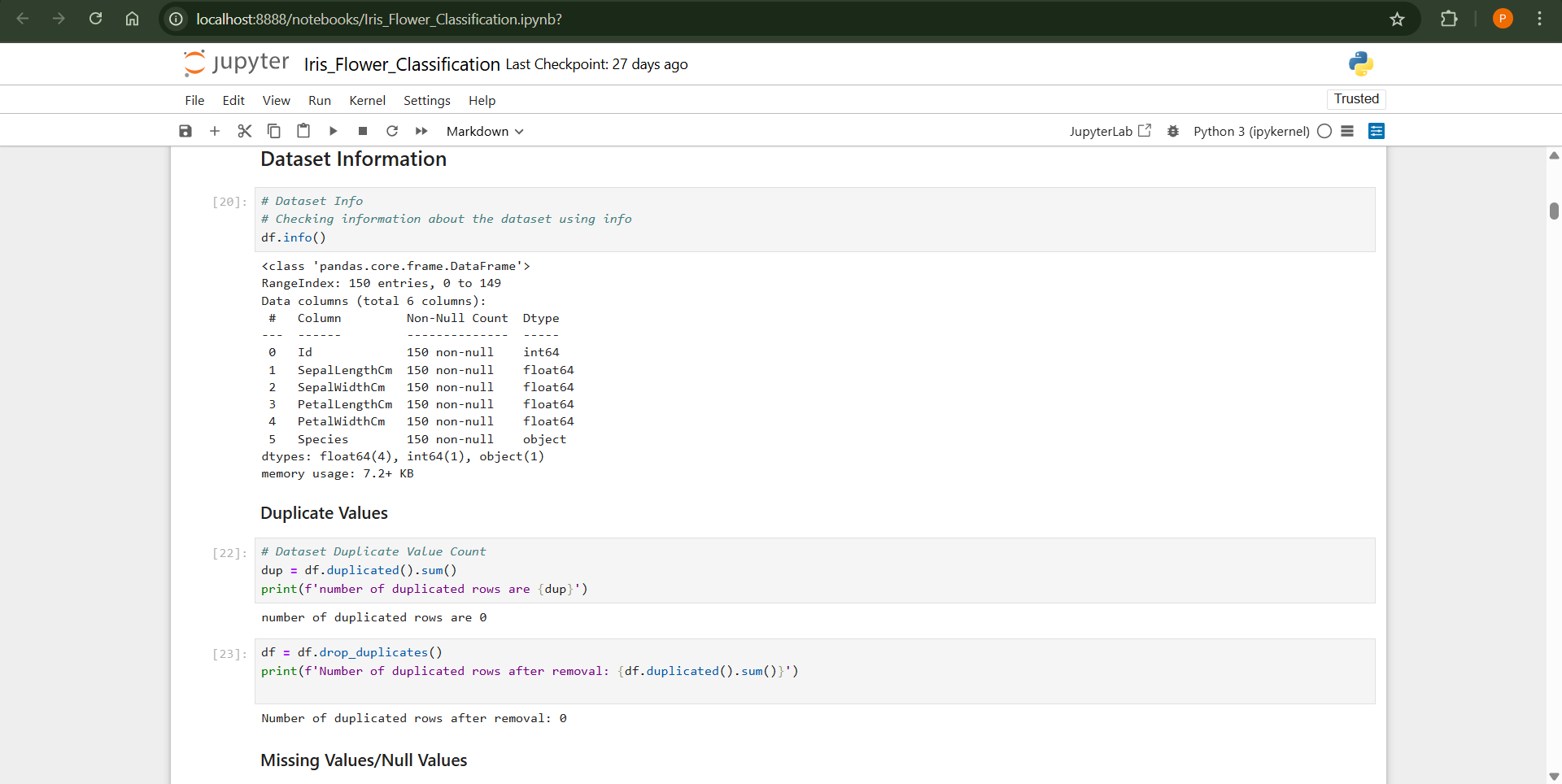
This project strengthened my foundational understanding of supervised learning, statistical analysis, and data visualization. It demonstrated how machine learning can be applied to solve real-world classification problems effectively. The skills gained during this project will be instrumental in my future career in data science and artificial intelligence.

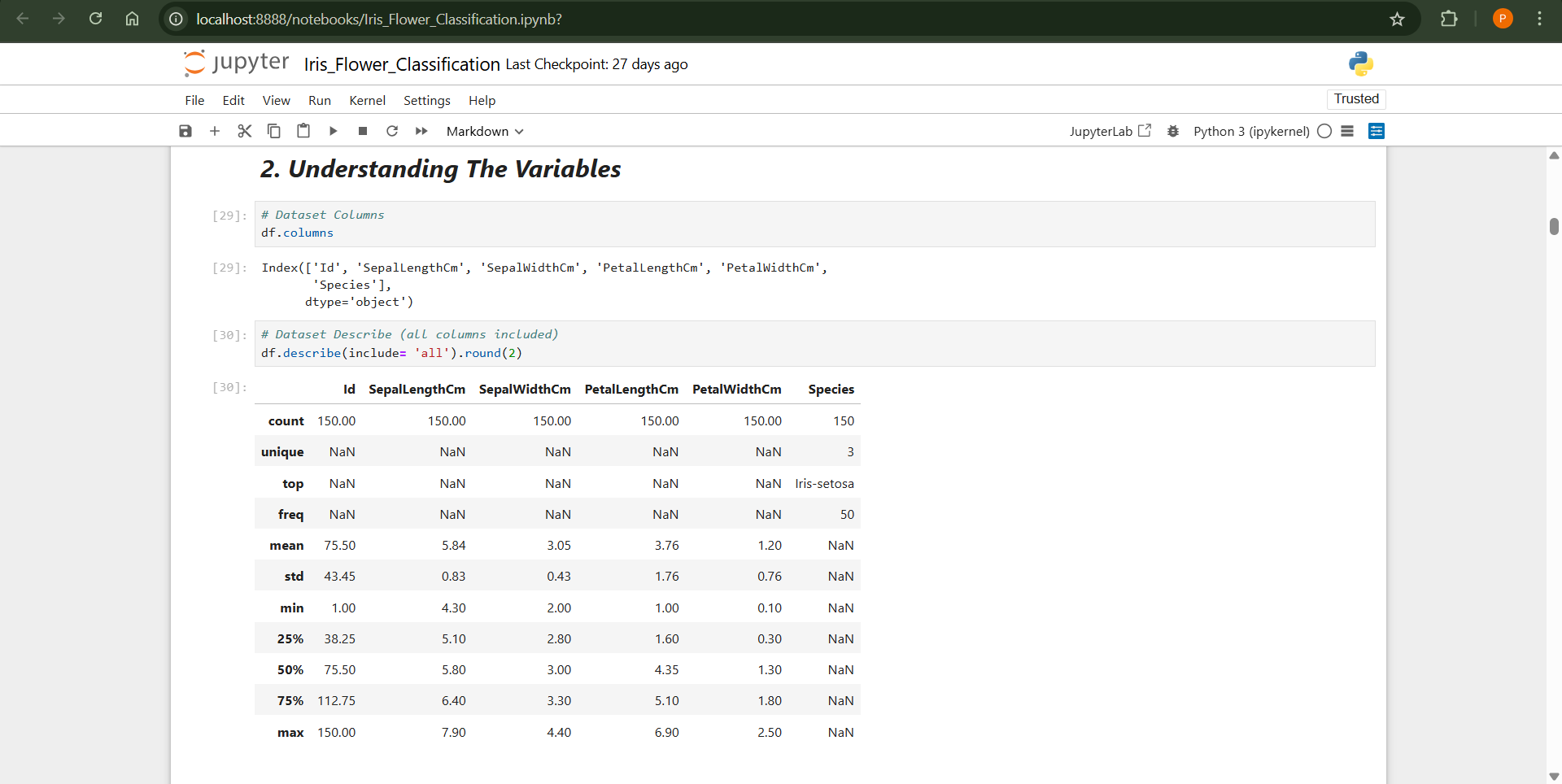
**Iris Classification: Data and Chart**

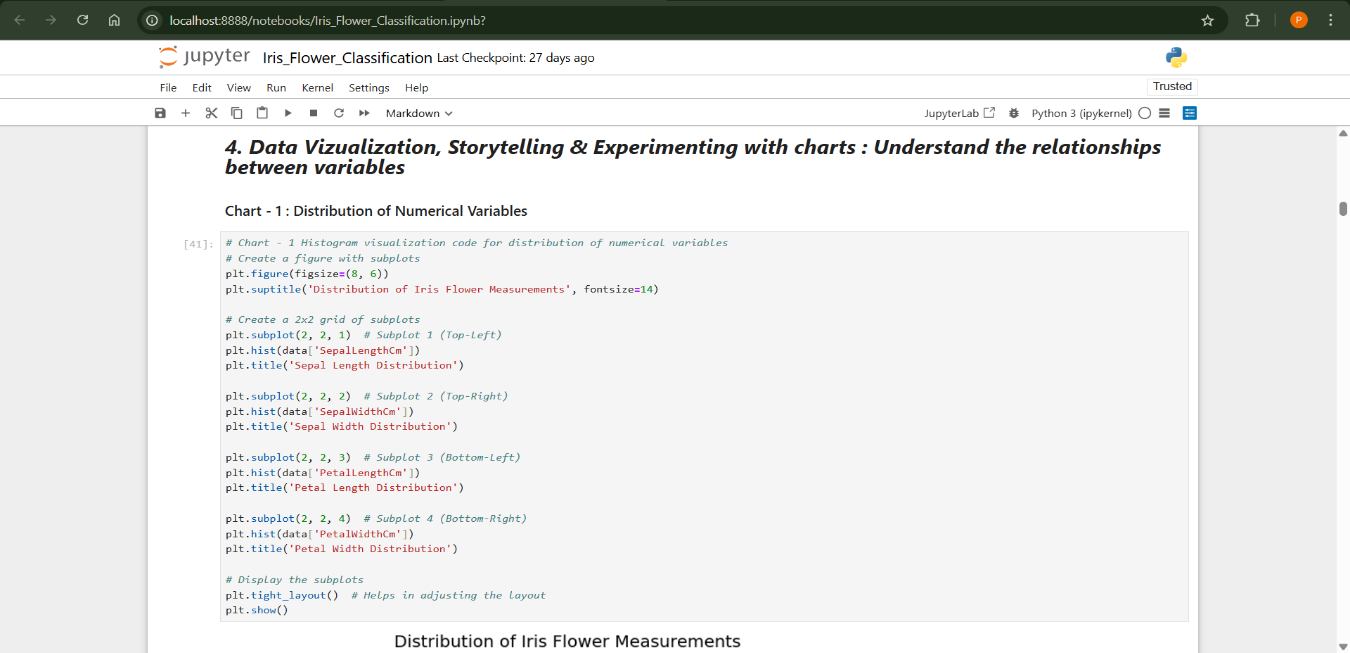
* **Library importing:**
* **Loading Dataset:** Top 5 values of dataset.



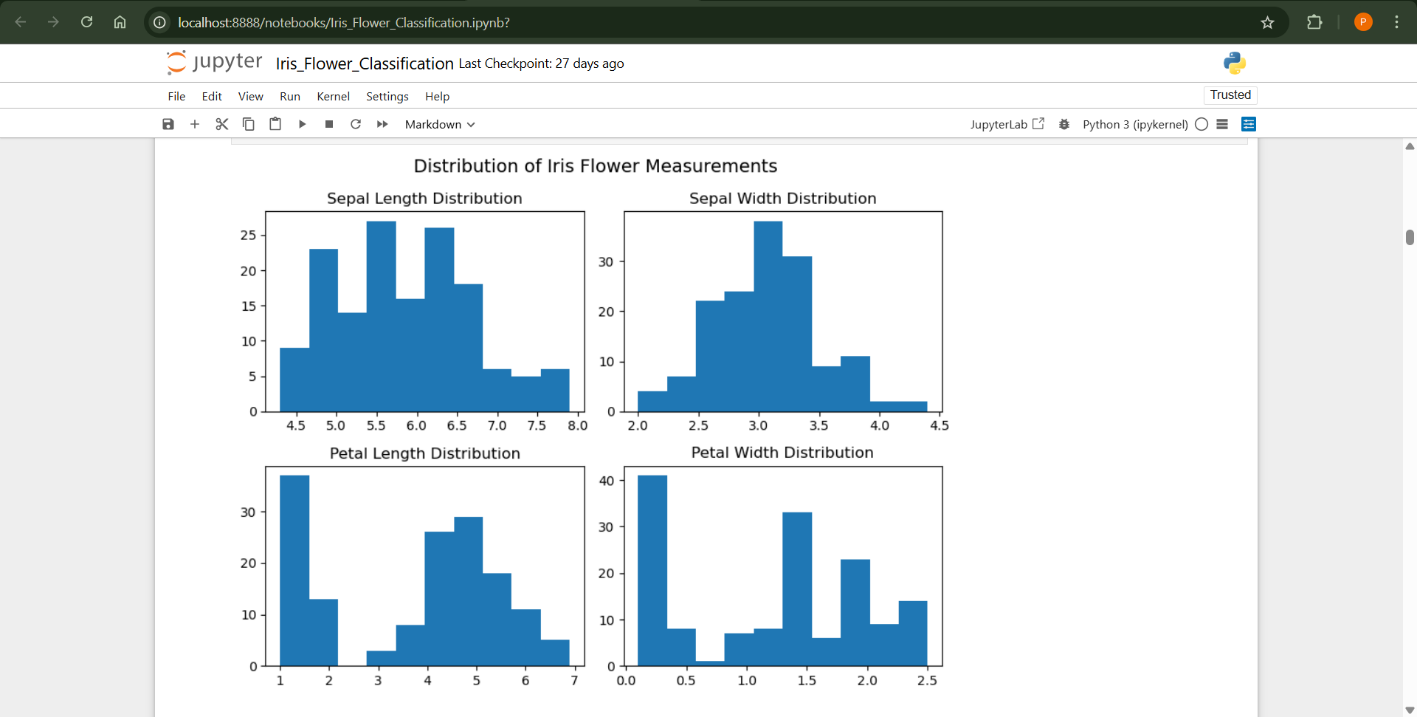
* **Extracting Dataset Information:**

****

****

* **Data Visualization:**

* **OUTPUT**



# Output of Evaluation Metric:

# 

# Output of Evaluation Metric:

# 

# 

# Final Model for Prediction:

# 

# 

# 

# References and Bibliography

1. **OpenAI.** (2025). *ChatGPT: A Large Language Model.* Retrieved from <https://www.openai.com/chatgpt>
2. **YouTube.** (2025). Retrieved from <https://www.youtube.com/>
3. **GitHub.** (2025). *GitHub: A Platform for Version Control and Collaboration.* Retrieved from <https://github.com/>

**B. Datasets and Public Repositories**

1. **UCI Machine Learning Repository.** (n.d.). *Iris Data Set.* Retrieved from <https://archive.ics.uci.edu/ml/datasets/Iris>
2. **Kaggle Dataset:** *Iris Classification.* Retrieved from <https://www.kaggle.com/uciml/iris>

**C. Official Documentation and Developer Resources**

1. **Scikit-learn:** *Machine Learning in Python.* <https://scikit-learn.org/stable/>
2. **Pandas Documentation.** <https://pandas.pydata.org/>
3. **Matplotlib Documentation.** <https://matplotlib.org/>
4. **Seaborn:** *Statistical Data Visualization.* <https://seaborn.pydata.org/>

**D. Blogs, Articles, and Online Tutorials**

1. **Analytics Vidhya.** (2022). *End-to-End Machine Learning Project for Beginners.* <https://www.analyticsvidhya.com>
2. **Towards Data Science.** (2021). *Support Vector Machines Explained.* <https://towardsdatascience.com/support-vector-machines-explained>
3. **Real Python.** (2020). *Visualizing Data in Python.* <https://realpython.com/>