NATIONAL INSTITUTIONAL RANKING FRAMEWORK
NIRF-2024
Engineering Rank Band (151-200)
Pharmacy Rank - 77
Innovation Rank Band (11-50)

KIET GROUP OF INSTITUTIONS
www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

CELEBRATING 25 YEARS OF EXCELLENCE

ACCREDITED WITH GRADE A+ NAAC
Accredited by
NAAC with Grade "A+"

Internal Quality Assurance Cell IQAC

Introduction to AI (ID-AI101B)
Even Semester
Session 2025-26

Iris Flower Classification

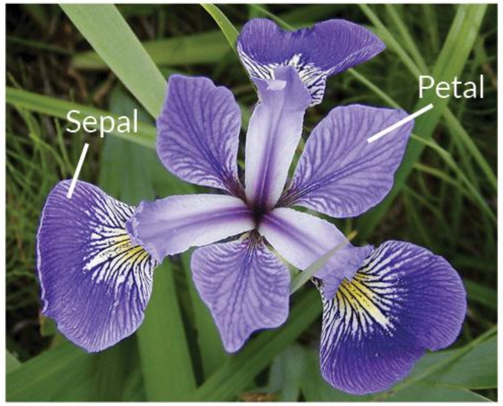
Manish Kumar Singh (202410116100114)
Divyanshu Mishra (202410116100116)
Divyansh Pathak (202410116100117)
Kartik Agarwal(202410116100096)

Project Supervisor:
Mr. Apoorv Jain
Designation-Assistant Professor.

Overview of the Iris Flower Dataset

- "The **Iris** flower dataset or Fisher's **Iris** dataset is a multivariate dataset introduced by Ronald Fisher in 1936."
- "**Edgar Anderson** collected the data to quantify the morphological variation of **Iris** flowers..."
- "It is a **multivariate** dataset (more than 2 dependent variables)..."
- "The dataset **contains** 50 samples of each species (**Iris setosa, Iris virginica, and Iris versicolor**)."

Clustering in the Iris Dataset



Iris Versicolor

Iris Setosa

Iris Virginica

- One group contains **Iris setosa**.
- The other group contains **Iris virginica** and **Iris versicolor**, which are difficult to separate without species labels.
- The dataset is **multivariate** (has more than two dependent variables).
- It includes **50 samples of each species**:
  - **Iris setosa**
  - **Iris virginica**
  - **Iris versicolor**

Feature List Formatting

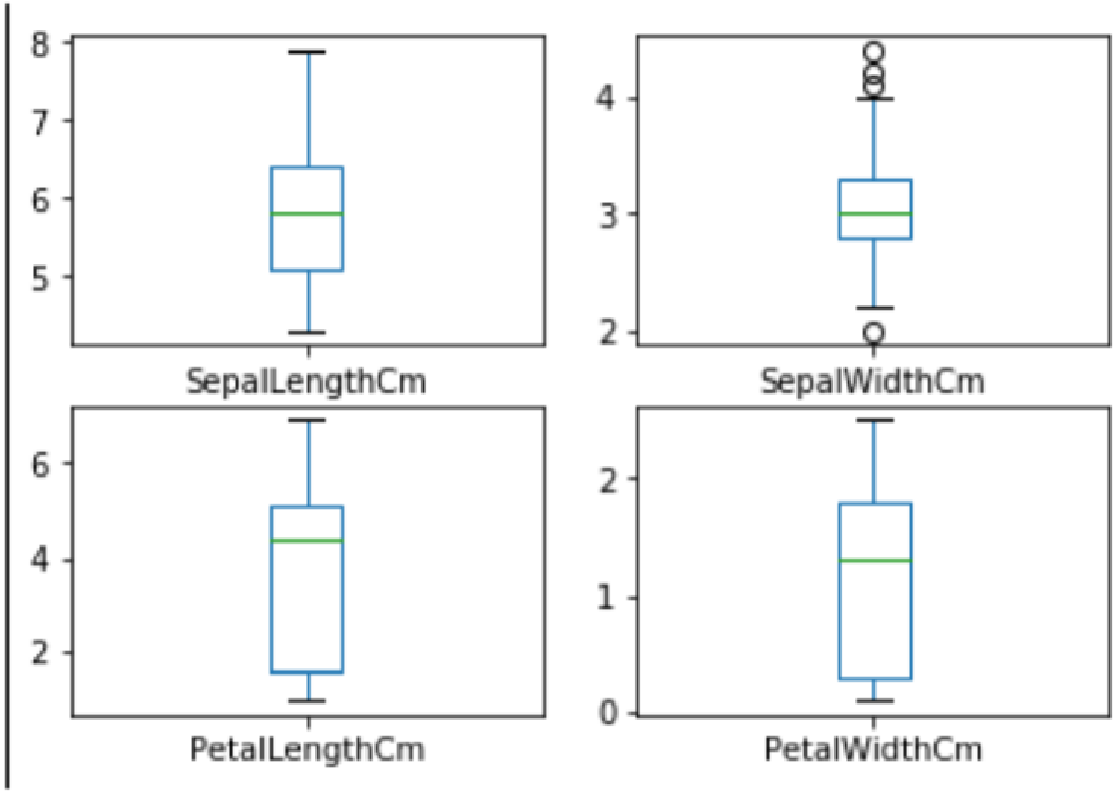
Features Used:

1. **Sepal Length** – The length of the outer part of the flower.
2. **Sepal Width** – The width of the outer part of the flower.
3. **Petal Length** – The length of the inner colorful part of the flower.
4. **Petal Width** – The width of the inner colorful part of the flower

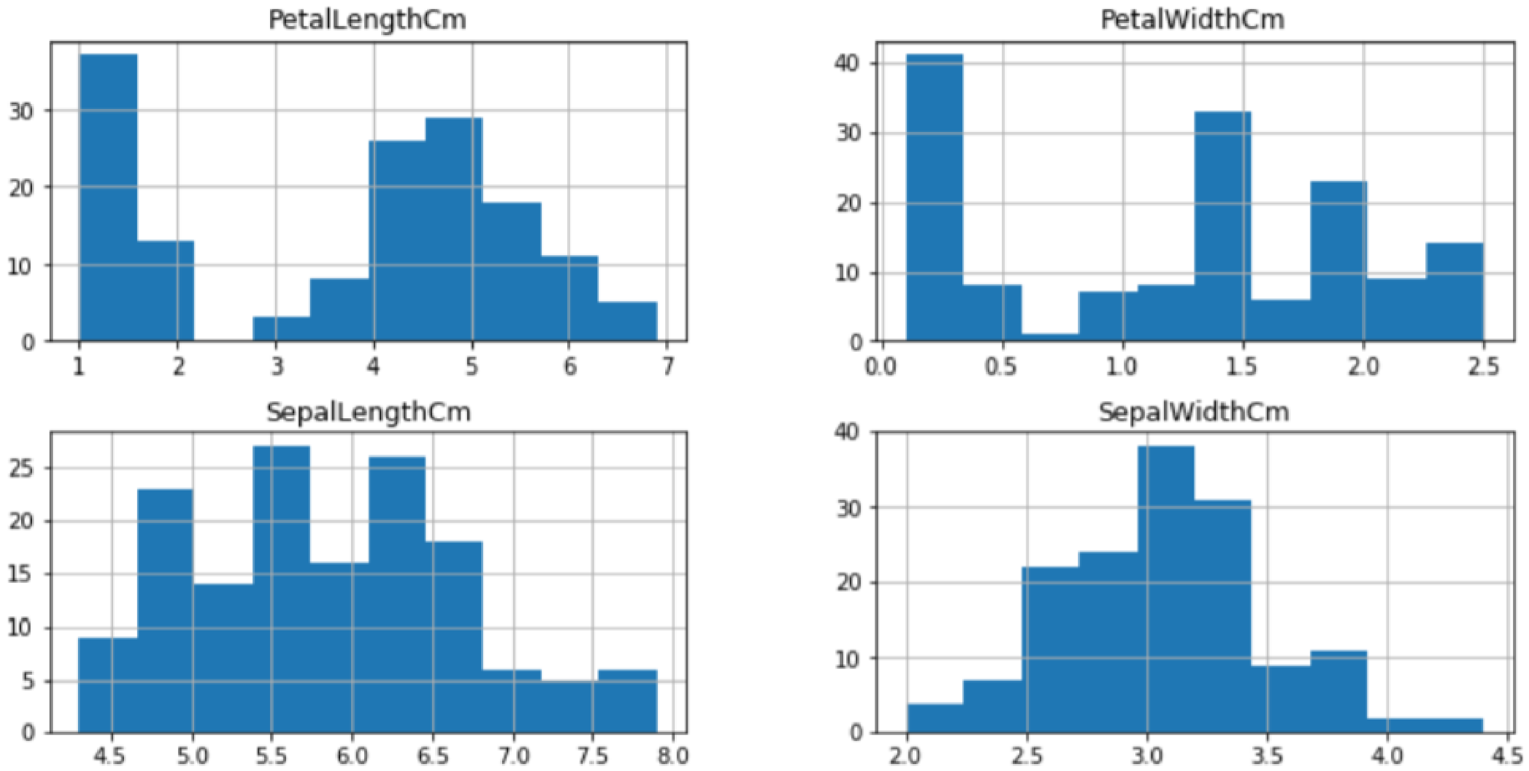
Data Analysis Formatting

- **Descriptive Statistics** – Summarizes data with values like **minimum, maximum, mean, and standard deviation** to understand variations.
- **Class Distribution** – Ensures each **Iris species** has an equal number of samples for fair model training.
- **Univariate Plots** – Uses **graphs** (histograms, box plots) to show how each feature is distributed among species.

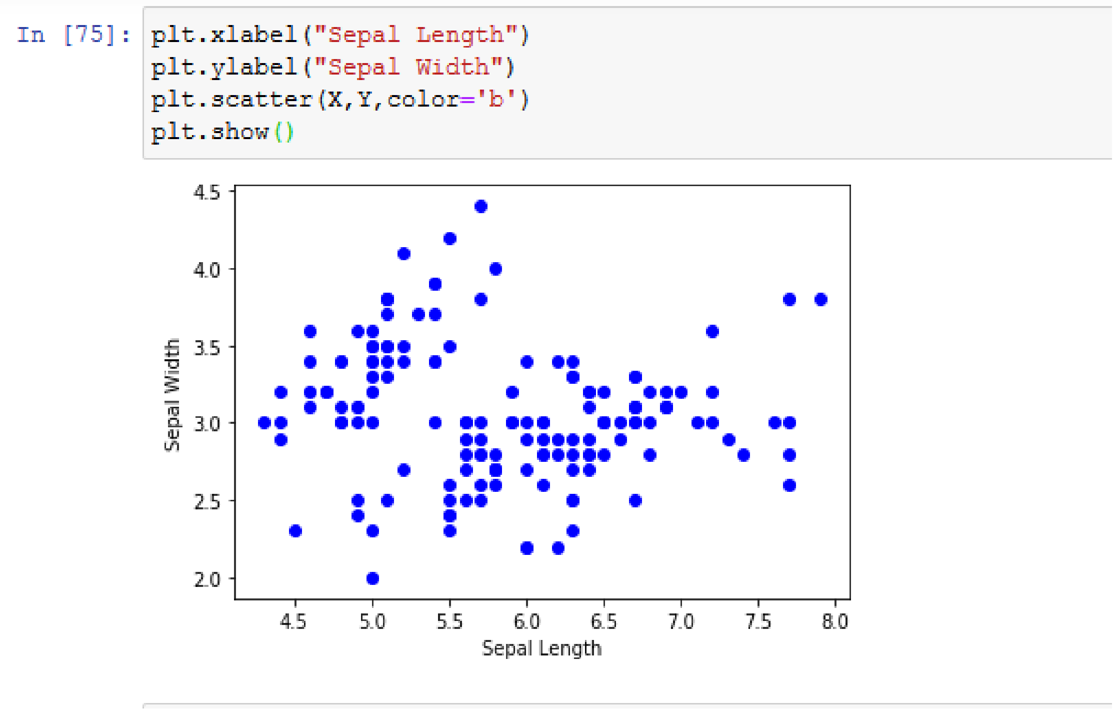
Box and whisker plots(Give idea about distribution of input attributes)



Plotting Histogram:



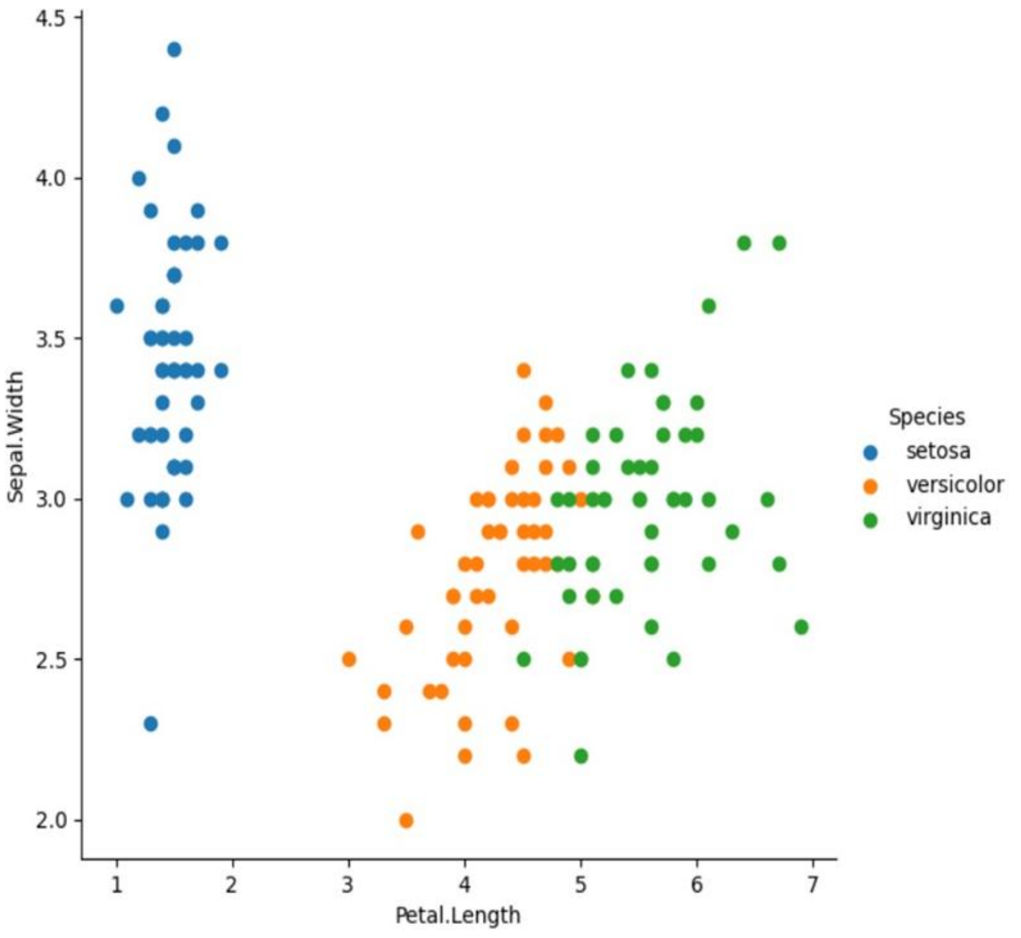
Plotting Scatter Graph Between Sepal Length and Sepal Width:



```
In [75]: plt.xlabel("Sepal Length")
         plt.ylabel("Sepal Width")
         plt.scatter(X,Y,color='b')
         plt.show()
```

Observation:

- Using **Sepal Length** & **Sepal Width**, we can only distinguish **Setosa** from other species.
- Separating **Versicolor** & **Virginica** is much harder due to their overlap.
- Therefore, **Sepal Length** & **Sepal Width** work best for **Setosa** only.



Machine Learning Implementation

"Steps to implement Machine Learning:“

1.Import Libraries – Load Pandas, NumPy, Matplotlib, and Scikit-Learn.

2.Analyze Data – Check missing values, visualize patterns, and understand distributions.

3.Split Data – Divide into **training (80%)** and **testing (20%)** sets for model evaluation.

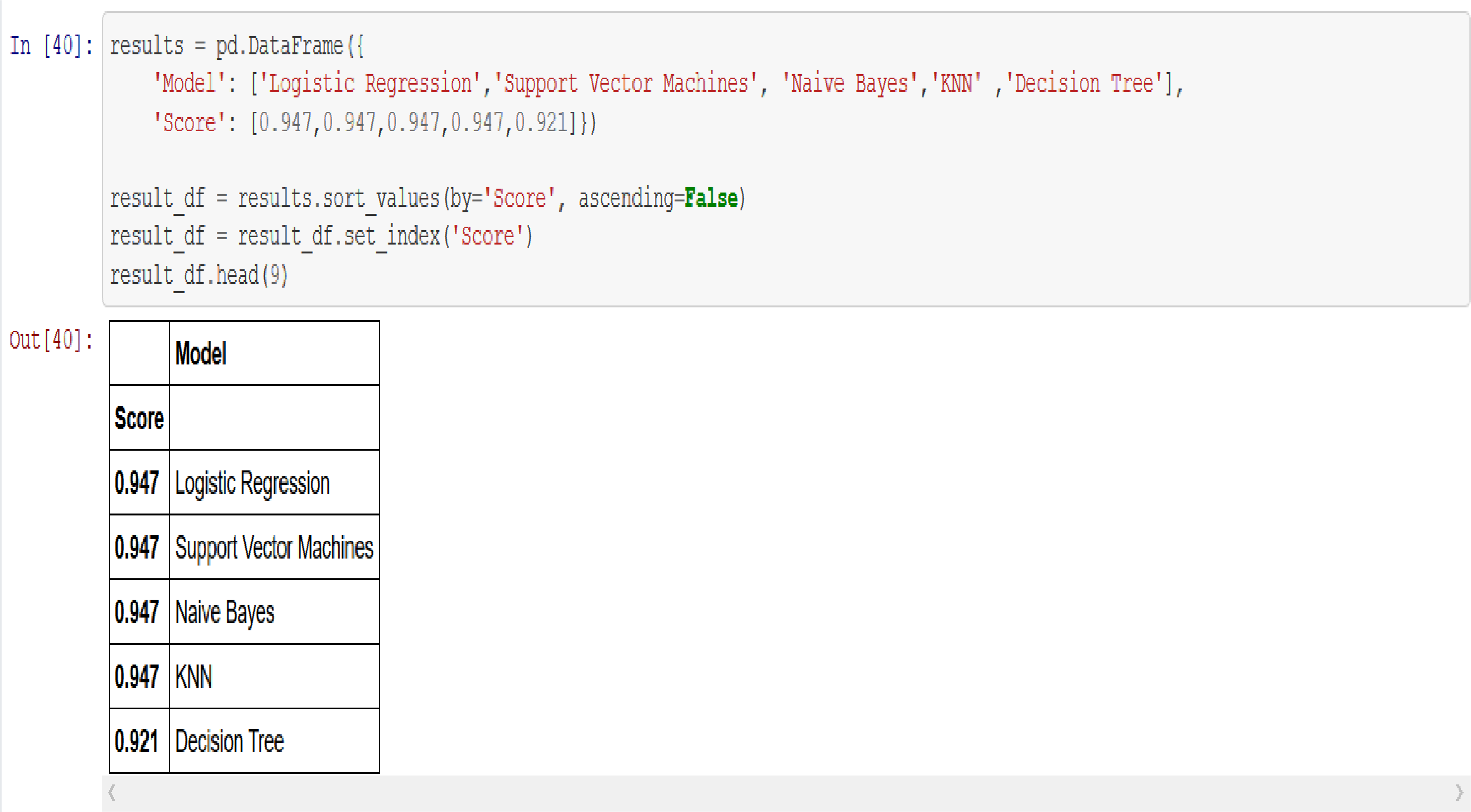
4.Choose Algorithm – Select models like **Logistic Regression, SVM, KNN, or Decision Tree**.

5.Test Model – Evaluate accuracy using test data, confusion matrix, and performance metrics.

Algorithms Used for Classification

1. **Logistic Regression** – Uses probability to classify data points.
2. **Support Vector Machine (SVM)** – Finds the optimal boundary between classes.
3. **Classification and Regression Tree (CART)** – Uses decision rules for classification.
4. **Gaussian Naïve Bayes (NB)** – Assumes feature independence for probabilistic classification.
5. **K-Nearest Neighbors (KNN)** – Classifies based on the majority of nearest neighbors.
6. **Decision Tree** – Splits data using conditions for easy interpretation.

Final Evaluation Of All Models:



In [40]:

```
results = pd.DataFrame({
    'Model': ['Logistic Regression','Support Vector Machines', 'Naive Bayes','KNN' ,'Decision Tree'],
    'Score': [0.947,0.947,0.947,0.947,0.921]})

result_df = results.sort_values(by='Score', ascending=False)
result_df = result_df.set_index('Score')
result_df.head(9)
```

Out[40]:

	Model
Score	
0.947	Logistic Regression
0.947	Support Vector Machines
0.947	Naive Bayes
0.947	KNN
0.921	Decision Tree

THANK YOU