Report of Task Submission 2024- March-P2 Batch OIB-SIP

On

Iris Flower Classification

By Amruta Varsha Yadav Pigili

Contents:

1. Introduction

1.1 Machine Learning:

Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed for each specific task.

This training allows the machine learning model to make predictions, classify new data, or take actions based on the learned patterns. Various strategies are included in machine learning, such as reinforcement learning, unsupervised learning, and supervised learning. In supervised learning, the model picks up new information from data that has already been labelled and has associated target labels or outcomes. On the other hand, unsupervised learning includes identifying patterns or relationships in unlabelled data. Through trial and error, reinforcement learning focuses on teaching models to make decisions based on interactions with the environment.

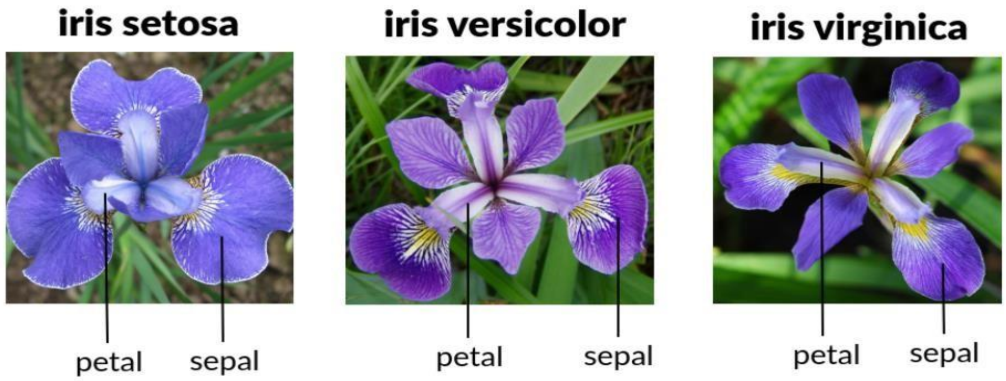
1.2 Classification:

Classification is a machine learning technique that involves assigning categorical labels or classes to input data based on patterns or relationships in the data. It is commonly used for tasks such as image recognition, spam filtering, sentiment analysis, fraud detection, and medical diagnosis, among others. The goal is to build a model that can accurately predict the class of new, unseen data.

Iris flower classification involves the task of categorizing iris flowers into different species based on their characteristics. The iris flower dataset is a multivariate data set used and was introduced by the British statistician and biologist Ronald Fisher in 1936. Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species. This dataset consists of measurements of four features, namely sepal length, sepal width, petal length, and petal width, taken from samples of three different iris species: Setosa, Versicolor, and Virginia. The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica, and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features.

Various machine learning algorithms can be used for iris flower classification, ranging from simple ones like k-nearest neighbors (KNN) and decision trees to more complex approaches like support vector machines (SVM) and neural networks. The choice of algorithm depends on factors such as the size of the dataset, the dimensionality of the feature space, and the desired accuracy.

Iris flower classification serves as a fundamental example for exploring concepts such as data preprocessing, feature selection, model training, and evaluation. It is often used to introduce concepts like supervised learning, classification algorithms, and model evaluation metrics in the field of Machine Learning.



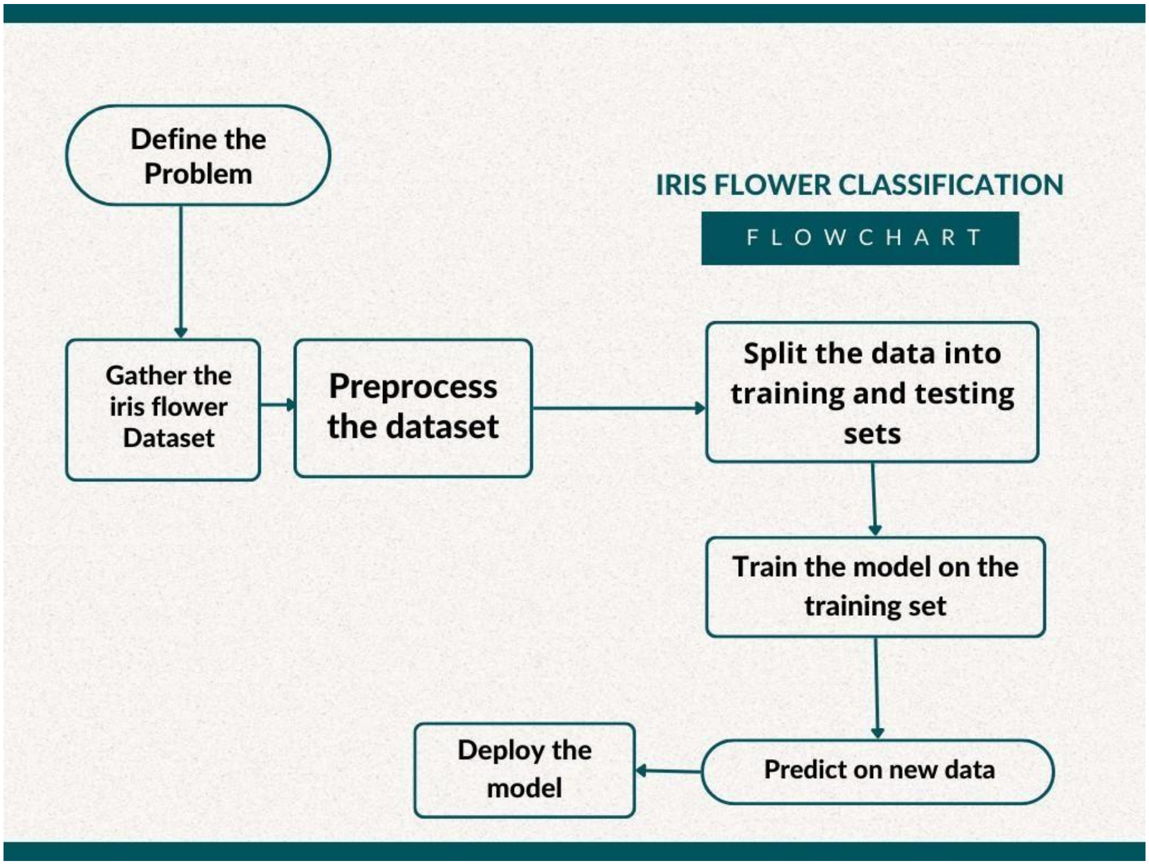
1.3 About the Task:

The task of iris flower classification involves building a machine learning model to accurately classify iris flowers into different species based on their measurements. Iris Dataset, which includes characteristics of various flower species. Sepal Length, Sepal Width, Petal Length, and Petal Width are independent features in this dataset. They were all measured in centimetres. Species is a dependent feature that will be the model's output. It includes the species name to which the specific flower with those measurements belongs.

2. Methodology

2.1 Use Case Diagram:

A flowchart is a diagrammatic representation of a process or workflow, typically using various shapes and arrows to indicate the flow of steps or decisions. Here's a Flow chart outlining the process of Iris Flower Classification.


IRIS FLOWER CLASSIFICATION
FLOWCHART
Define the Problem
Gather the iris flower Dataset
Preprocess the dataset
Split the data into training and testing sets
Train the model on the training set
Predict on new data
Deploy the model

2.2 Tech-Stack used:

1. Programming Language: I had used Python due to its extensive libraries for data analysis and machine learning.
2. Data Manipulation and Analysis: Libraries used are pandas and NumPy for data manipulation, preprocessing, and exploratory data analysis tasks.
3. Model Training and Evaluation: scikit-learn, for splitting the data into training and testing sets.
4. Visualization: Matplotlib and Seaborn - Python libraries was used for creating visualizations,

3. Implementation

3.1 Problem Statement:

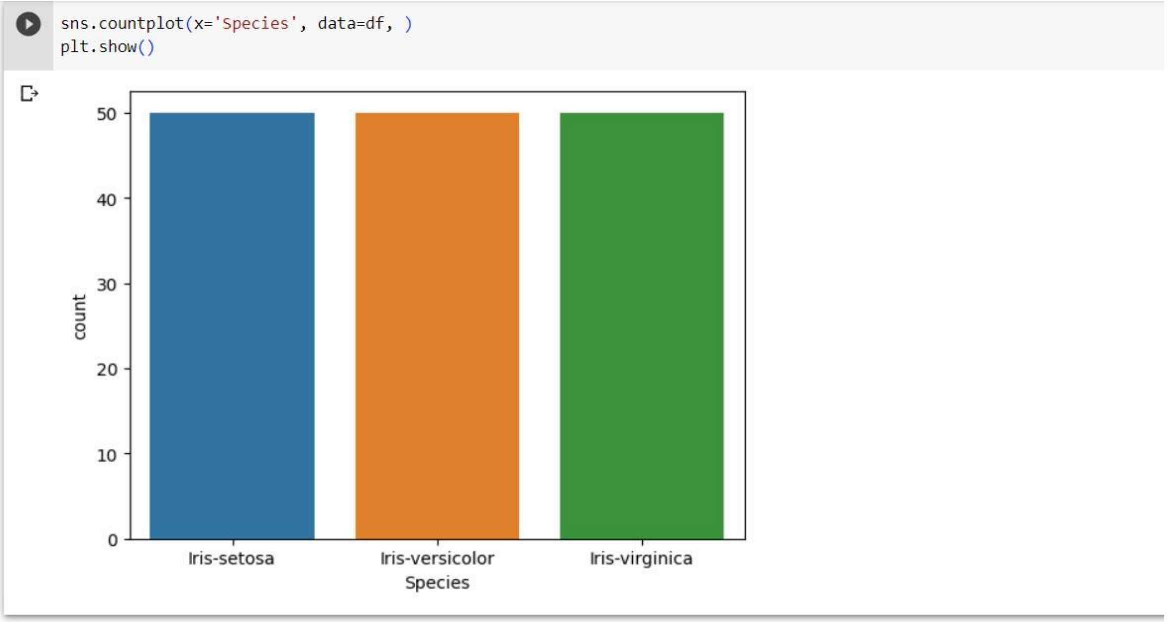
The objective is to develop a classification model that can correctly classify the different species of iris flowers using measurements of their sepal length, petal length, sepal width, and petal width. The dataset contains 150 examples of iris flowers from three different species—Setosa, Versicolor, and Virginica.

3.2 Data Collection:

The Iris dataset was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems, and can also be found on the UCI Machine Learning Repository.
It includes three iris species with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.

3.3 Data Exploration and Visualization

An exploratory data analysis was performed to gain the details of the data set prior to creating the classification models. This included analysing the statistical summary of the features, visualizing the distributions of the features using count plot, pair plot, histograms, box plots, and heat map, and exploring relationships between the features using scatter plots.



```
sns.countplot(x='Species', data=df, )
plt.show()
```

According to the Plot the total rows are 150 and on which 50 are Iris-Setosa, 50 are Iris-Versicolor and 50 are Iris-Virginica

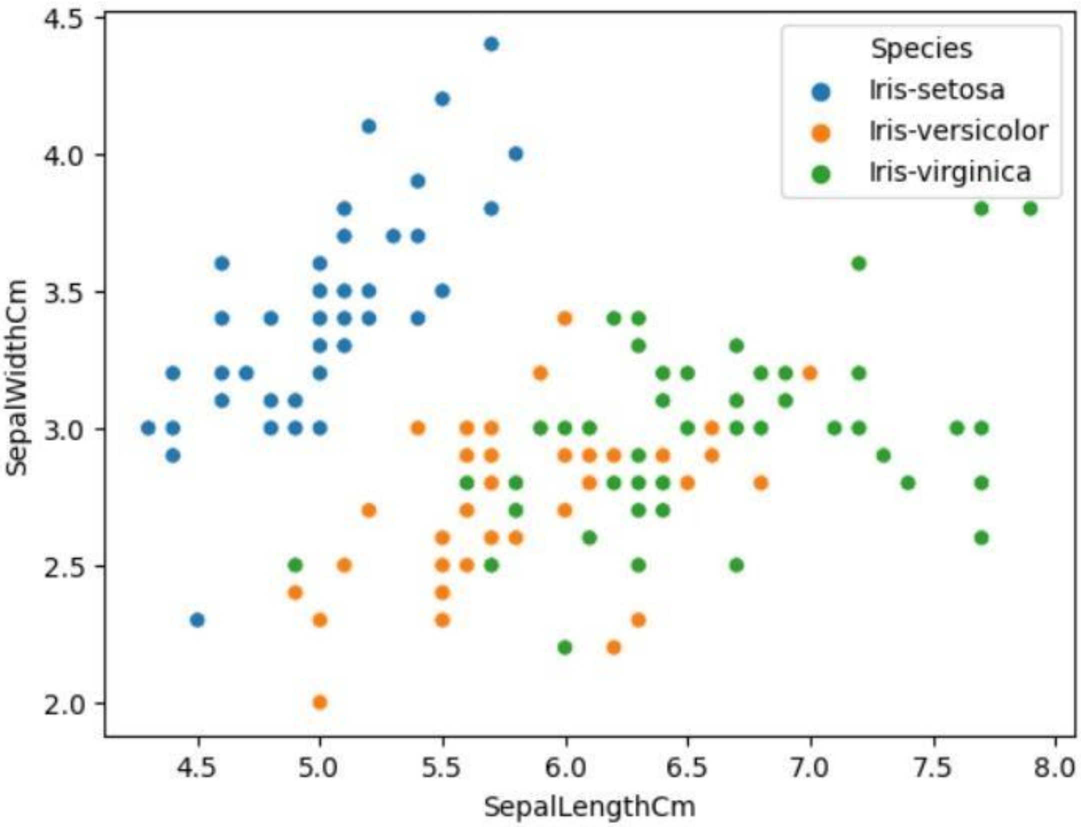


Fig 3.3.2 Relationship Graph between Sepal Length and Sepal Width

According to the above plot

1) Iris-Setosa has smaller Sepal Length and larger Sepal Widths

2) Iris-Versicolor is the medium range of Sepal Length and Sepal Width and, 3) Iris-Virginica has larger sepal lengths and larger sepal width

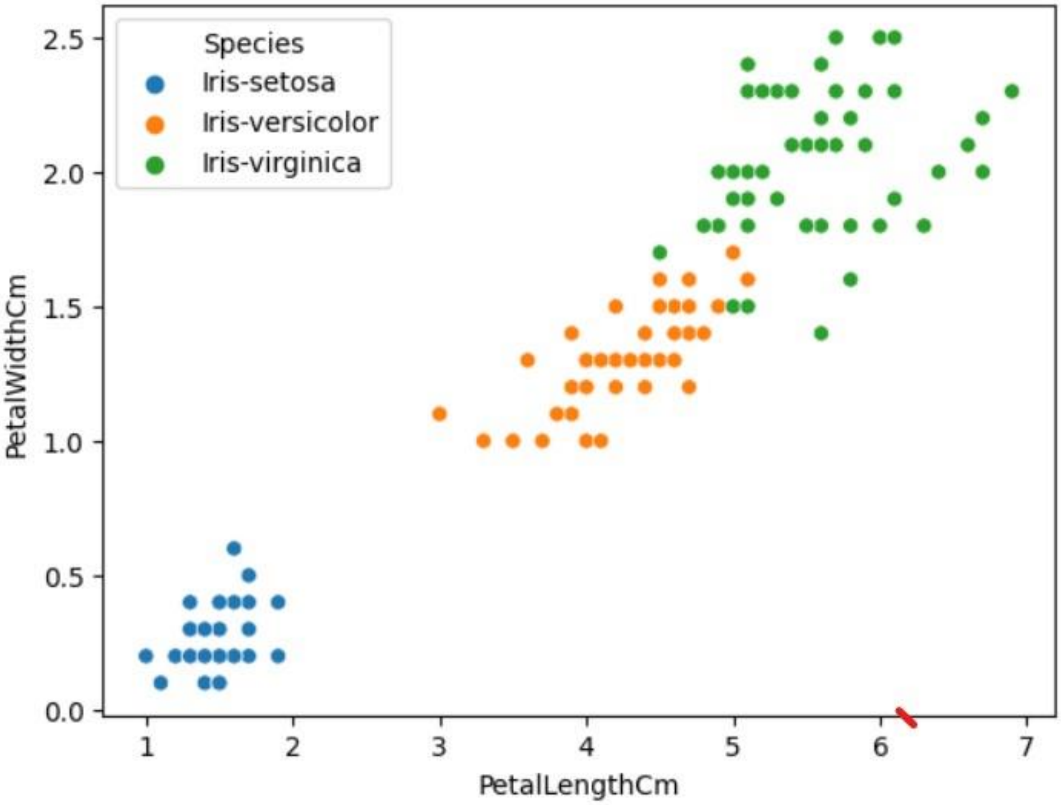


Fig 3.3.3 Relationship Graph between Petal Length and Petal Width

This Scatter plot explains that Iris Setosa has smaller Petal Length and Smaller Petal width, while Iris- Versicolor lies in the middle and Iris- Virginica has Larger Petal Width and Petal Length

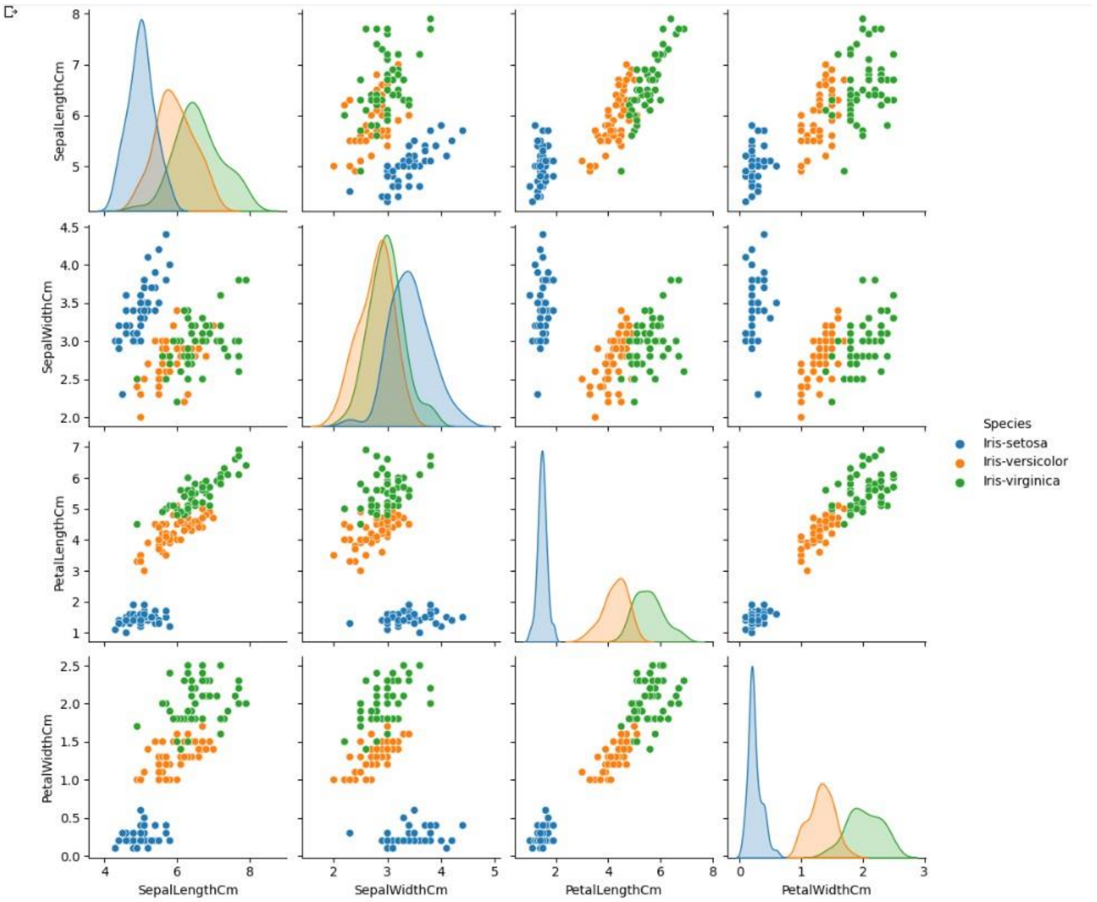


Fig 3.3.4 Pair Plot showing the relationship between Sepal Length, Sepal Width, Petal Length, and Petal Width of different Species.

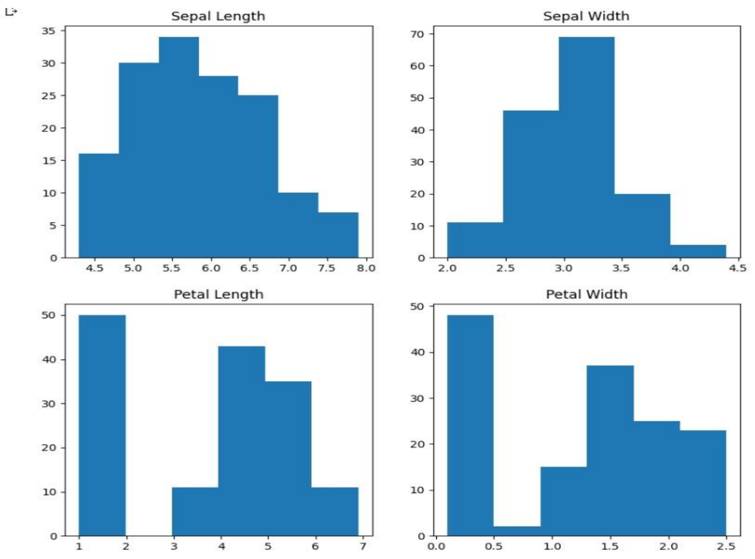


Fig 3.3.5 Histogram Plot

The above Histogram depicts the:

- The highest frequency of the sepal length is between 30 and 35 which is between 5.5 and 6
- The highest frequency of the sepal Width is around 70 which is between 3.0 and 3.5
- The highest frequency of the petal length is around 50 which is between 1 and 2

- The highest frequency of the petal width is between 40 and 50 which is between 0.0 and 0.5

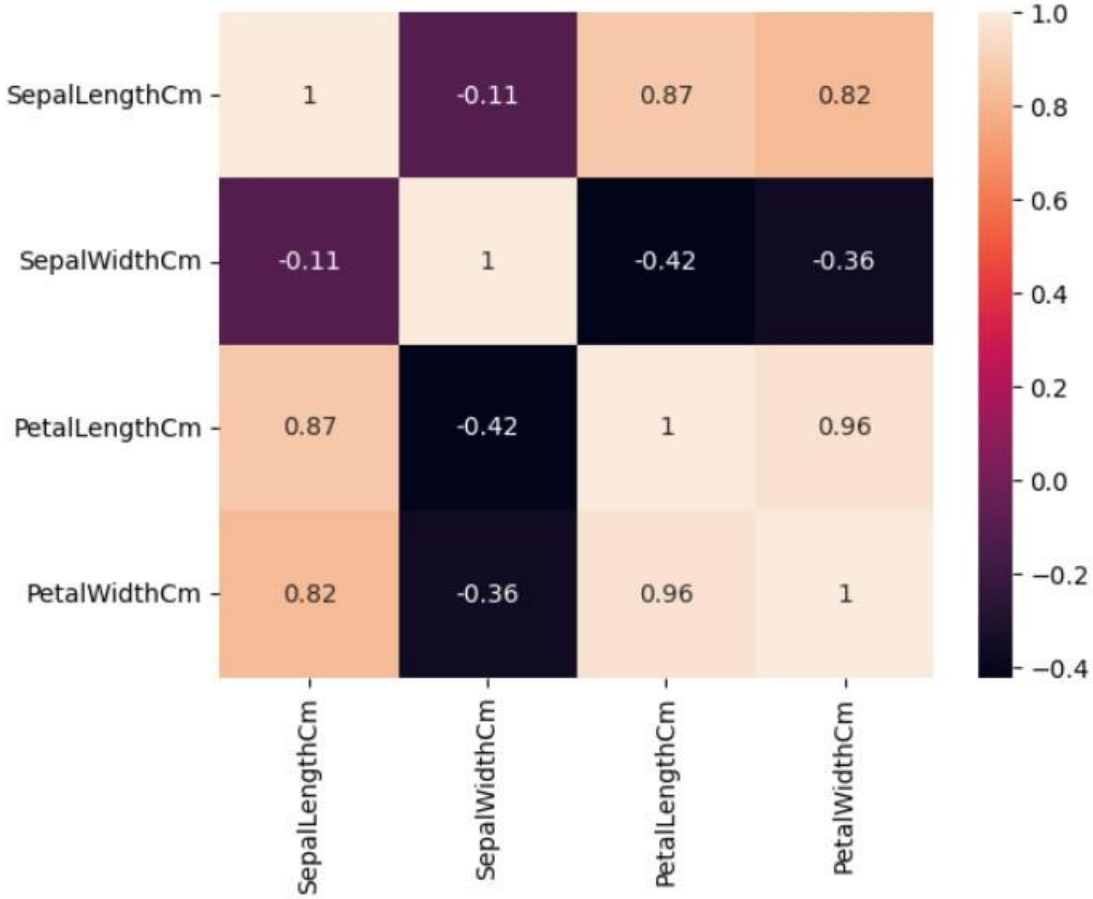


Fig 3.3.5 Heatmap

From the above graph:

- Petal width and petal length have high correlations.
- Petal length and sepal width have good correlations.
- Petal Width and Sepal length have good correlations.

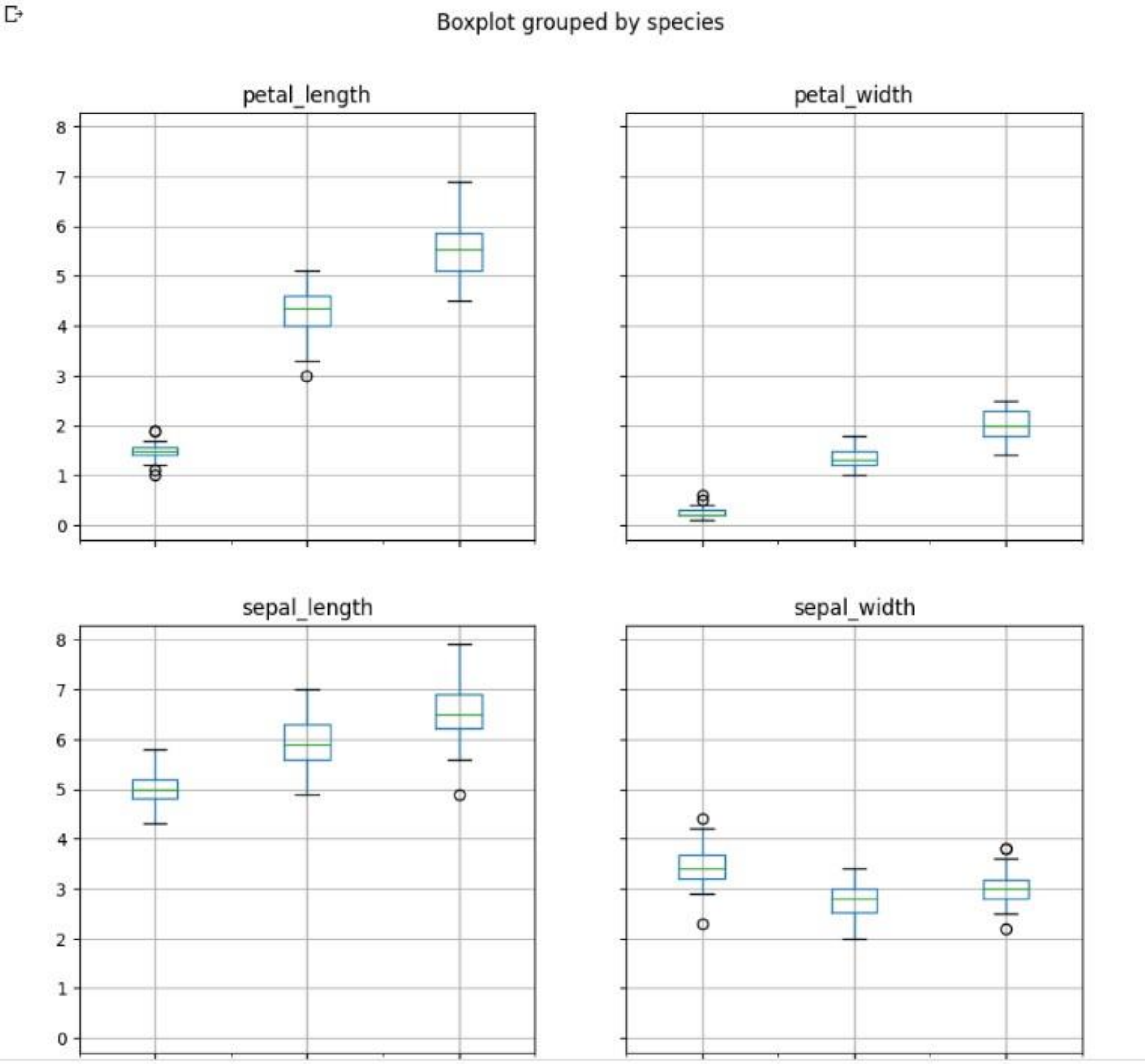


Fig 3.3.5 Box Plot

Box Plot is a graphical representation of a dataset that provides a visual summary of its distribution. It displays several key summary statistics, including the median, quartiles, and potential outliers.

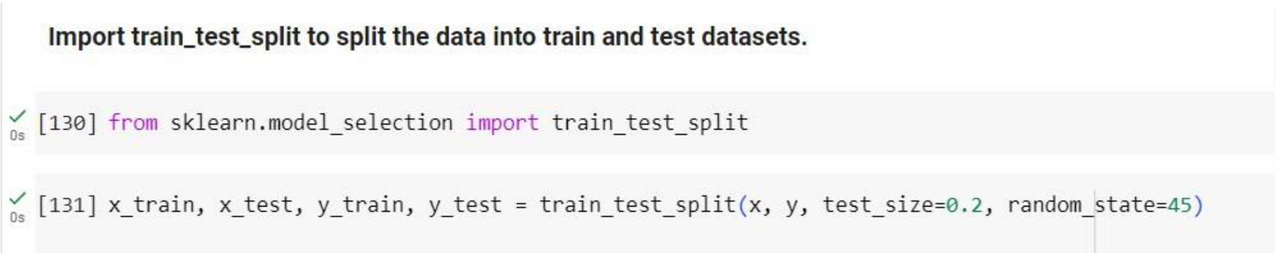
According to the graph:

- Species Setosa has the smallest features and less distributed with some outliers.
- Species Versicolor has the average features.
- Species Virginica has the highest features

3.4 Data Splitting:

First, we split the dataset into two subsets: a training set and a testing set. The training set will be used to train the model, while the testing set will be used to evaluate its performance. A common split is to use around 70-80% of the data for training and the remaining 20-30% for testing. Here we are utilising the **'train_test_split' scikit-learn** library method, which divides our data collection into an 80:20 ratio, with 80% of the data being used for training and 20% being used for testing.

Code snippet:



Import train_test_split to split the data into train and test datasets.

```
[130] from sklearn.model_selection import train_test_split
```

```
[131] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=45)
```

Fig 3.4.1 Code snippet of Training and Testing State

3.5 Training:

Training the iris dataset involves using a machine learning algorithm to learn patterns and relationships in the data that can help classify iris flowers into their respective species.

3.6 Testing:

Testing the iris flower classification model involves evaluating its performance on a separate dataset called the testing set, which contains iris flower samples that the model has not seen during training. The purpose of testing is to assess how well the trained model generalizes to new, unseen data and to estimate its real-world performance.

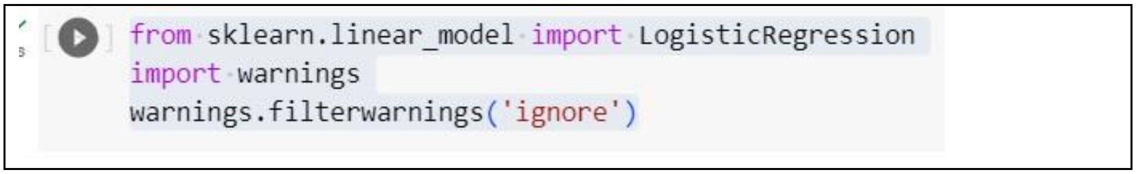
3.7 Feature Extraction:

The goal is to extract informative features that capture the important patterns and variations within the data, which can improve the performance of the classification model. In the case of the iris flower dataset, the features are typically the measurements of the flower's sepal length, sepal width, petal length, and petal width.

4. Algorithms used

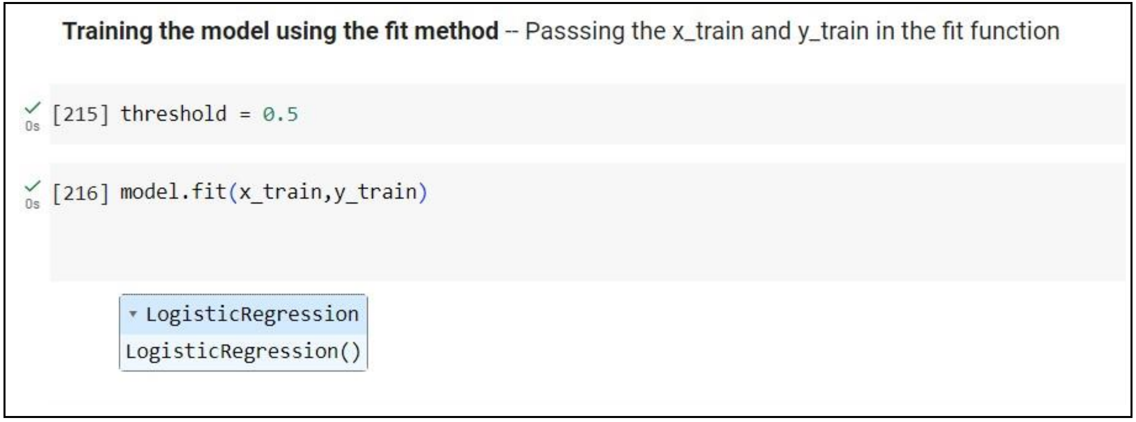
4.1 Logistic Regression:

Logistic regression is a supervised machine learning algorithm used for binary classification tasks. It is a linear model that predicts the probability of an input belonging to a particular class. Given a dataset with input features (X) and corresponding labels (y) for binary classification, logistic regression aims to learn a decision boundary that separates the two classes.



```
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
```

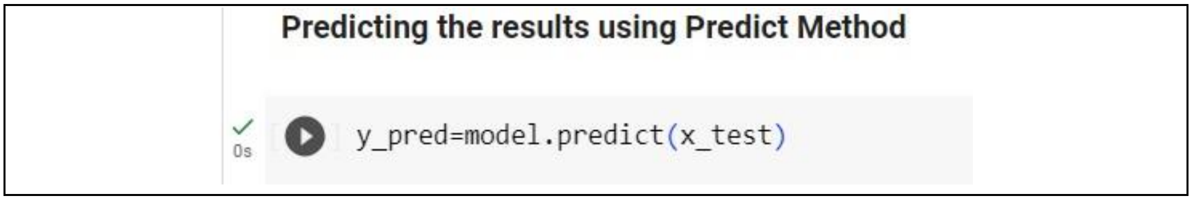
Fig 4.1.1 Code snippet of importing Logistic Regression from the sci-kit learn library.



Training the model using the fit method -- Passsing the x_train and y_train in the fit function

```
[215] threshold = 0.5

[216] model.fit(x_train,y_train)

      ▾ LogisticRegression
      LogisticRegression()
```

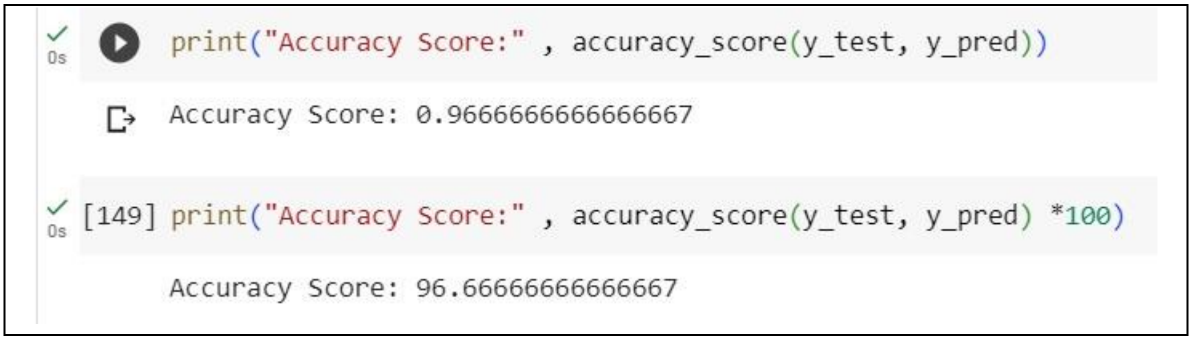
Fig 4.1.2 Code snippet of Training the Model



Predicting the results using Predict Method

```
y_pred=model.predict(x_test)
```

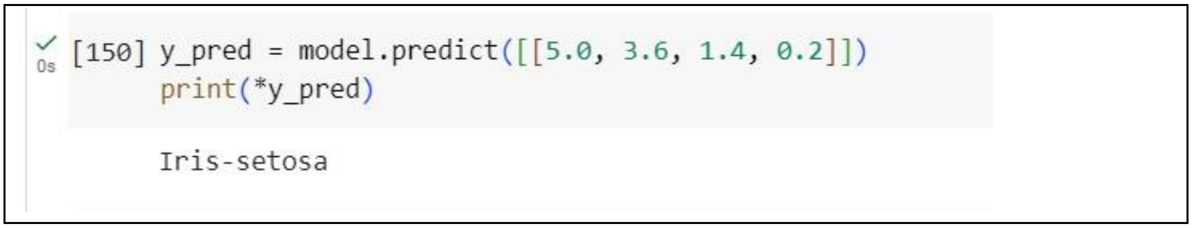
Fig 4.1.3 Code snippet of predicting data



```
print("Accuracy Score:" , accuracy_score(y_test, y_pred))

Accuracy Score: 0.9666666666666667

[149] print("Accuracy Score:" , accuracy_score(y_test, y_pred) *100)

Accuracy Score: 96.66666666666667
```

Fig 4.1.4 Code snippet of accuracy of the model

Accuracy of the model is 96.66, which is very accurate



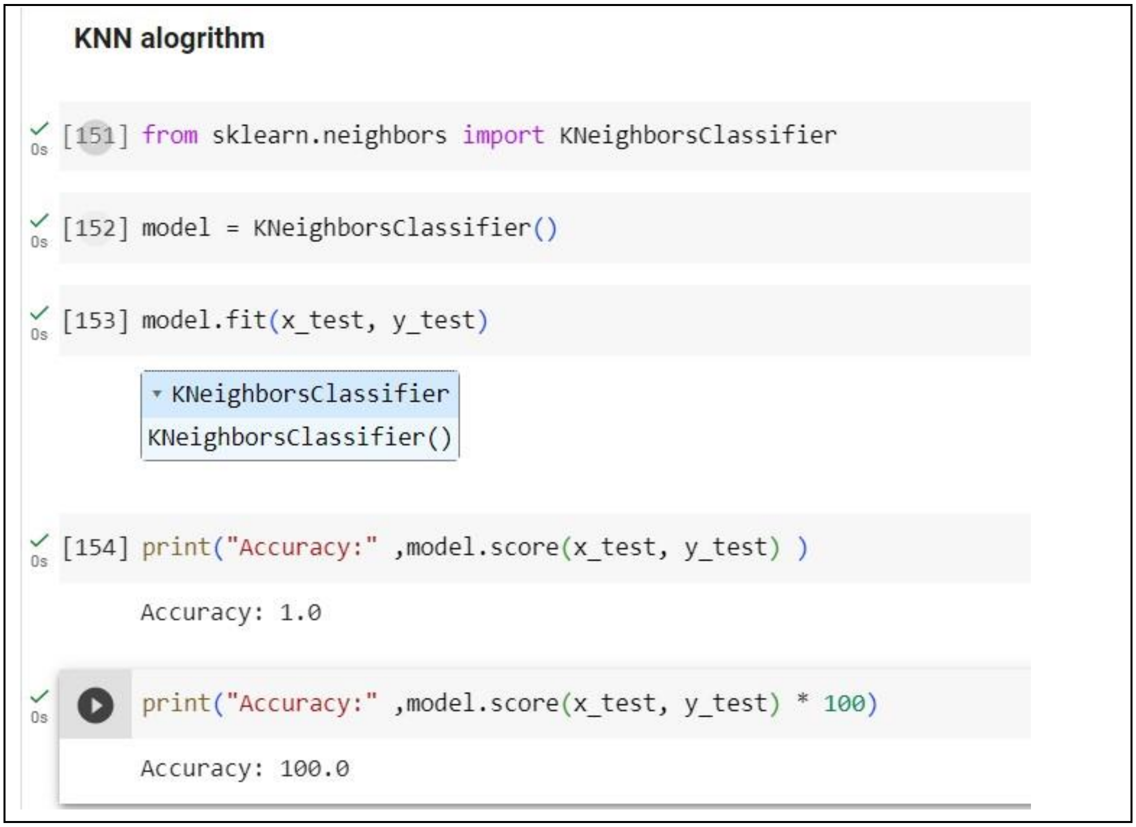
```
[150] y_pred = model.predict([[5.0, 3.6, 1.4, 0.2]])
      print(*y_pred)

      Iris-setosa
```

Fig 4.1.5 Prediction of the Iris Species

Predicted the Species of Iris Flower which is IRIS- SETOSA

4.2 . K-Nearest Neighbor Algorithm:

The k-Nearest Neighbors (KNN) algorithm is a supervised machine learning algorithm used for both classification and regression tasks. It is a non-parametric algorithm that makes predictions based on the similarity of data points in the feature space. Given a dataset with input features (X) and corresponding class labels (y), the KNN algorithm aims to classify new, unseen data points based on their similarity to the existing data points.

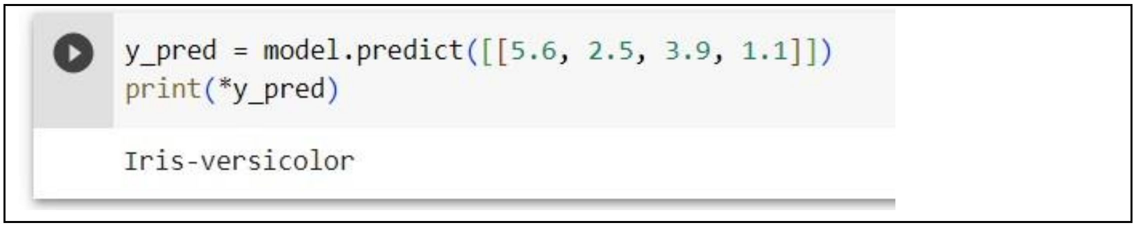


KNN alogrithm

```
[151] from sklearn.neighbors import KNeighborsClassifier

[152] model = KNeighborsClassifier()

[153] model.fit(x_test, y_test)

      ▾ KNeighborsClassifier
      KNeighborsClassifier()

[154] print("Accuracy:" ,model.score(x_test, y_test) )

      Accuracy: 1.0

      print("Accuracy:" ,model.score(x_test, y_test) * 100)

      Accuracy: 100.0
```

Fig 4.2.1 KNN Algorithm

Accuracy of the model is 100.0, which is very accurate



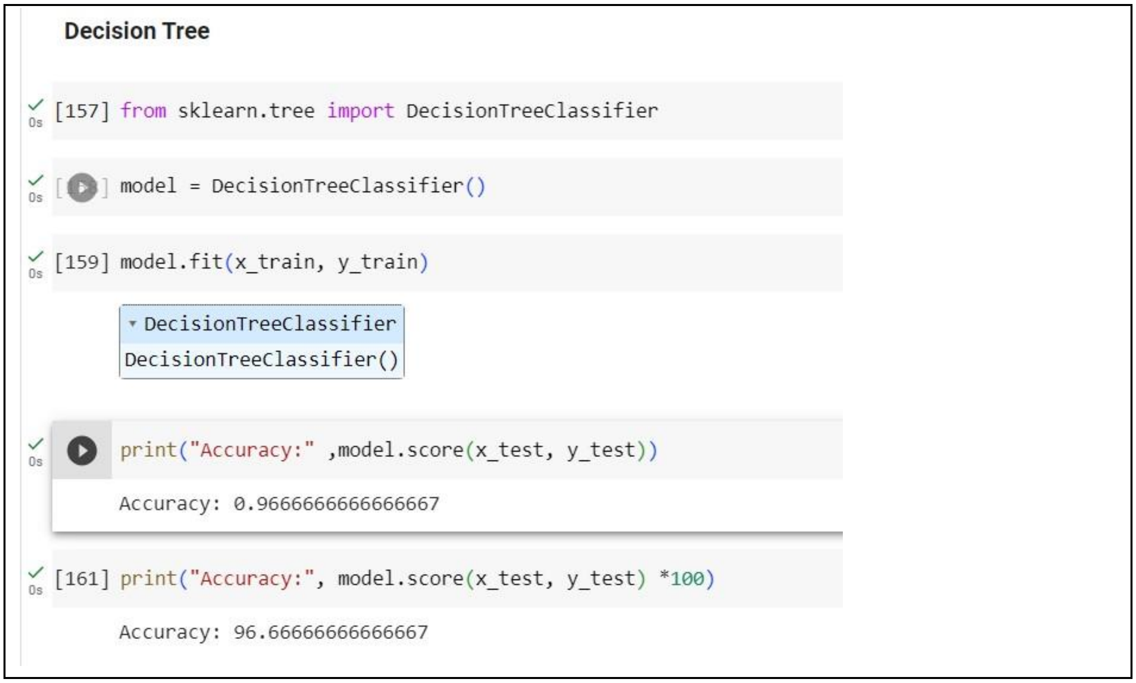
```
y_pred = model.predict([[5.6, 2.5, 3.9, 1.1]])
print(*y_pred)

Iris-versicolor
```

Fig 4.2.2 Prediction of the Iris Flower Species

Predicted the Species of Iris Flower which is IRIS- Versicolor

4.3 . Decision Tree:

A decision tree is a supervised machine learning algorithm that can be used for both classification and regression tasks. It creates a flowchart-like structure, where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the outcome or prediction.

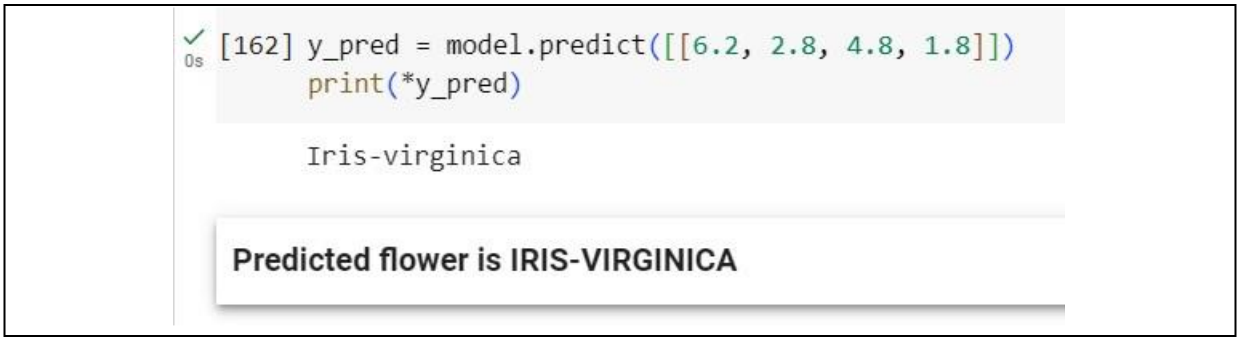


Decision Tree

```
[157] from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

[159] model.fit(x_train, y_train)

▾ DecisionTreeClassifier
DecisionTreeClassifier()

print("Accuracy:" ,model.score(x_test, y_test))

Accuracy: 0.9666666666666667

[161] print("Accuracy:", model.score(x_test, y_test) *100)

Accuracy: 96.66666666666667
```

Fig 4.3.1 Decision Tree Algorithm

Accuracy of the model is 96.66, which is very accurate



```
[162] y_pred = model.predict([[6.2, 2.8, 4.8, 1.8]])
      print(*y_pred)

      Iris-virginica
```

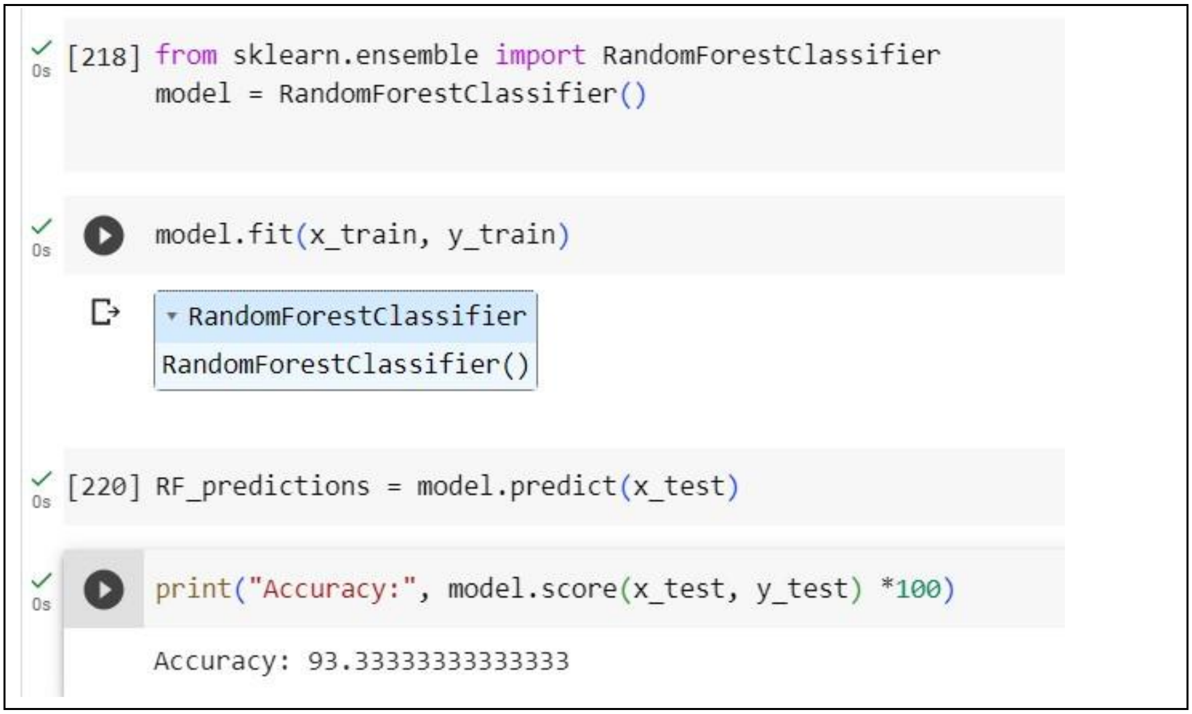
Predicted flower is IRIS-VIRGINICA

Fig 4.3.2 Prediction of Iris Flower Species

Predicted the Species of Iris Flower which is IRIS- Virginica

4.4 . Random Forest Algorithm:

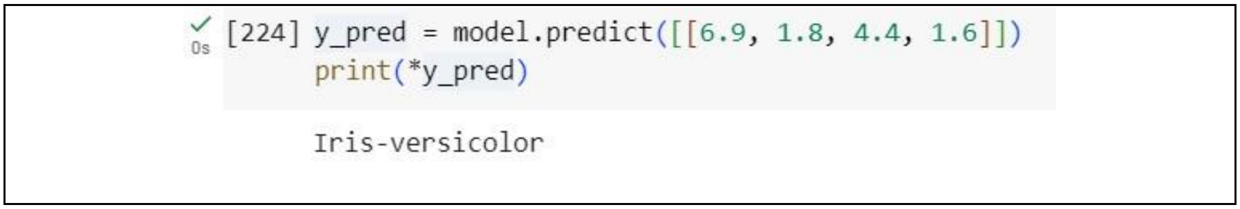
The Random Forest algorithm is an ensemble learning method that combines multiple decision trees to make predictions. It is a powerful algorithm widely used for classification tasks, including the classification of the iris flower dataset. Random Forests are known for their ability to handle complex datasets, handle both numerical and categorical features, and provide good generalization performance. They also offer built-in feature importance estimation, which can help identify the most informative features for classification



```
[218] from sklearn.ensemble import RandomForestClassifier
      model = RandomForestClassifier()

      model.fit(x_train, y_train)

      ▾ RandomForestClassifier
      RandomForestClassifier()

[220] RF_predictions = model.predict(x_test)

      print("Accuracy:", model.score(x_test, y_test) *100)

      Accuracy: 93.33333333333333
```

Fig 4.4.1 Random Forest Algorithm

Accuracy of the model is 93.33



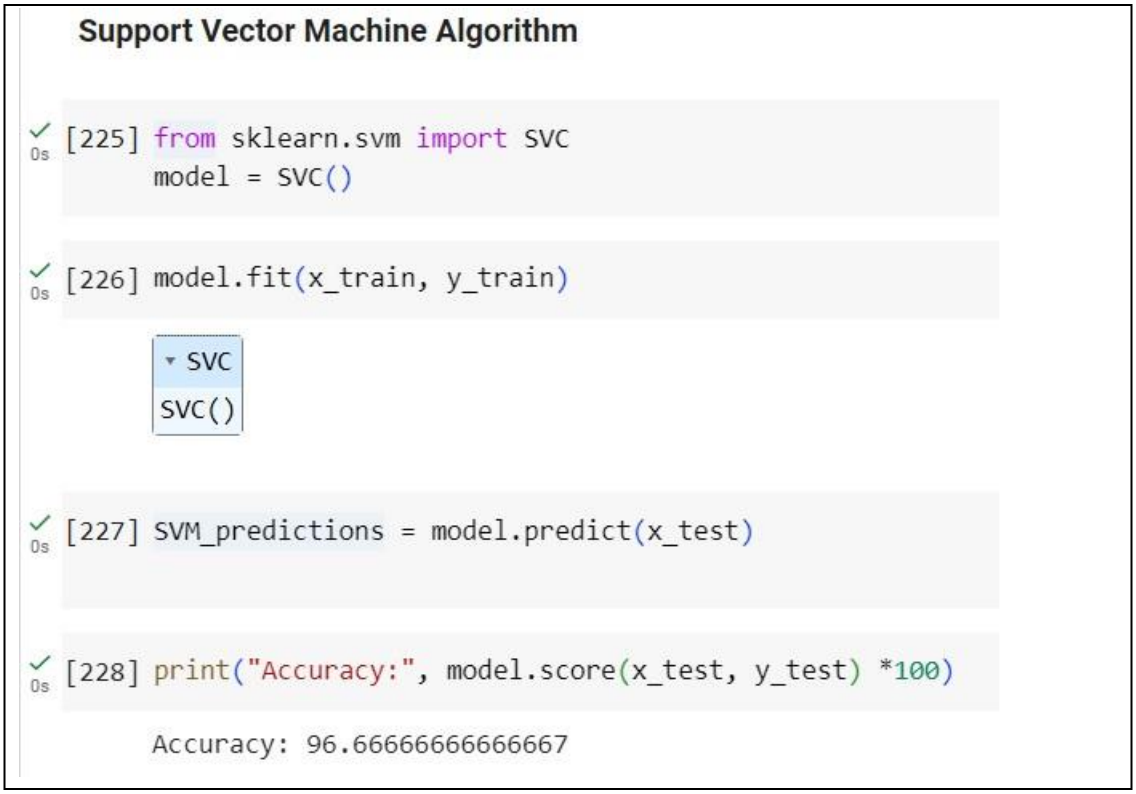
```
[224] y_pred = model.predict([[6.9, 1.8, 4.4, 1.6]])
      print(*y_pred)

      Iris-versicolor
```

Fig 4.4.2 Prediction of Iris Flower Species

Predicted the Species of Iris Flower which is IRIS- Versicolor

4.5 . Support Vector Machine Algorithm:

Support Vector Machines (SVM) is a supervised machine learning algorithm that can be used for classification tasks, including the classification of the iris flower dataset. SVM is known for its ability to handle complex datasets and handle both linear and non-linear decision boundaries. It can handle both numerical and categorical features through appropriate kernel functions. SVM also has a regularization parameter that can be tuned to control the trade-off between fitting the training data and generalization to unseen data.

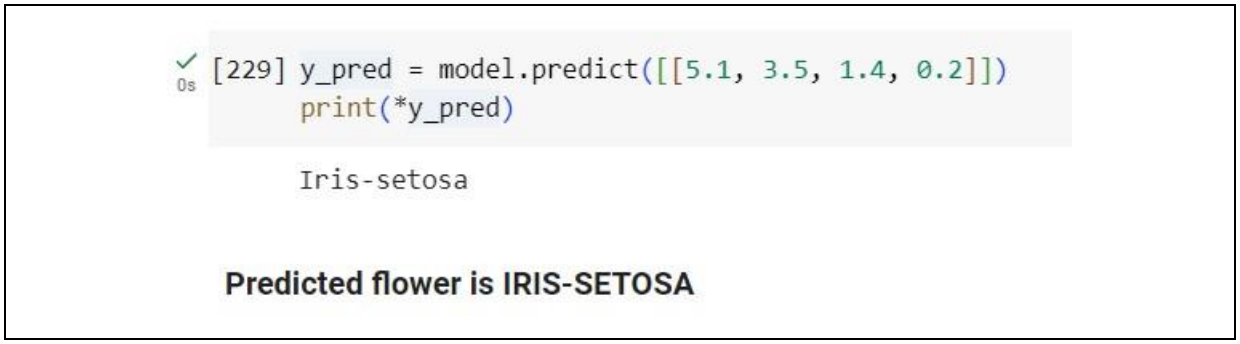


Support Vector Machine Algorithm

```
[225] from sklearn.svm import SVC
      model = SVC()

[226] model.fit(x_train, y_train)

      ▾ SVC
      SVC()

[227] SVM_predictions = model.predict(x_test)

[228] print("Accuracy:", model.score(x_test, y_test) *100)

      Accuracy: 96.66666666666667
```

Fig 4.5.1 Support Vector Machine Algorithm

Accuracy of the model is 96.66



```
[229] y_pred = model.predict([[5.1, 3.5, 1.4, 0.2]])
      print(*y_pred)

      Iris-setosa
```

Predicted flower is IRIS-SETOSA

Fig 4.5.1 Prediction of Iris Flower Species

Predicted the Species of Iris Flower which is IRIS- Setosa

Accuracy of **Logistic Regression Algorithm: 96.66**

Accuracy of **KNN Algorithm: 100**

Accuracy of **Decision Tree Algorithm: 96.66**

Accuracy of **Random Forest Algorithm**: **93.33**

Accuracy of **SVM Algorithm: 96.66**

There are two models which shows the highest accuracy i.e. Random Forest and SVM with accuracy score 1.0.

5. Conclusion

In conclusion, the iris flower dataset is a commonly used dataset in machine learning and is often used for classification tasks. I had used various machine learning algorithms to the iris flower dataset to classify the different species accurately. In this article, we covered five popular algorithms: Logistic Regression, Decision Tree, Support Vector Machine (SVM), KNN, and Random Forest. Overall, the iris flower dataset serves as an excellent benchmark dataset for exploring and comparing different machine learning algorithms. It provides a solid foundation for understanding classification algorithms and their applicability to real-world problems. The accuracy score of the above five models is very good and they can be used to predict the species of Iris Flower in five of the above models KNN Algorithm shows 100% accuracy.