Report on Project:-

Iris Flowers Classification ML Project

Submitted By:-Keshav Kumar

Reg No. :- 12014465

Section:- KM096

Roll No. :- 29

Submitted to:- Md.Imran Hussain(26819)

Contents

Team Members

1.Keshav Kumar

2.Rishabh Gupta

Introduction

Machine Learning

Machine learning is a process of feeding a machine enough data to train and predict a possible outcome using the algorithms. the more the processed or useful data is fed to the machine the more efficient the machine will become. When the data is complicated it learns the data and builds the prediction model. It is state that more the data, better the model, higher will be the accuracy. There are many ways for machine learning i.e. supervised learning, unsupervised learning and reinforcement learning.

Supervised Learning

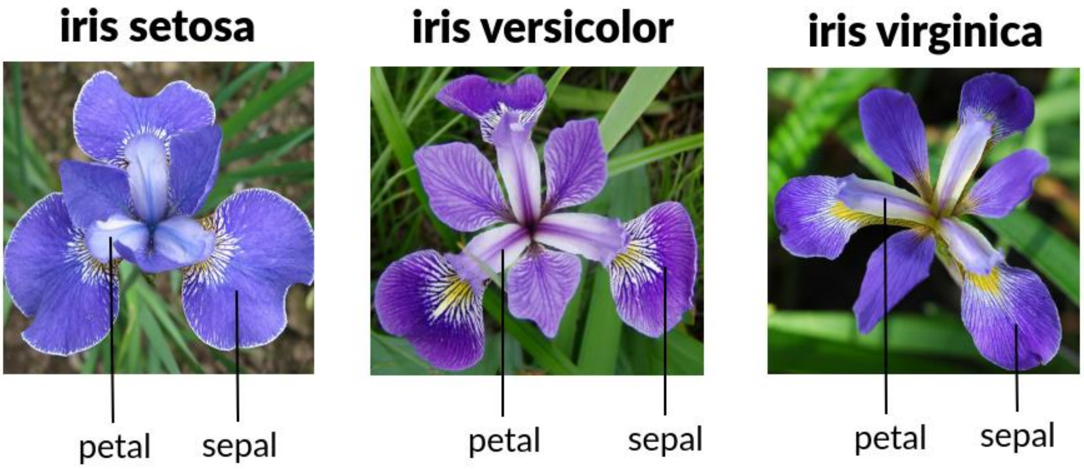
In supervised learning machine learning model learns through the feature and labels of the object. Supervised learning uses labeled data to train the model here, the machine knew the features of the object and labels associated with those features or we can say that the supervised learning uses the set of data where the labels or the desired outcomes are already known. It is allowed to prediction about the unseen or future data.

Classification

Classification is one of the major data mining processes which maps data into predefined groups. It comes under supervised learning method as the classes are determined before examining the data. For applying all approaches to performing classification it is required to have some knowledge of the data. Usually, the knowledge of the data helps to find some unknown patterns. The aim of pattern classification is to building a function that provides output of two or more than two classes from the input feature.

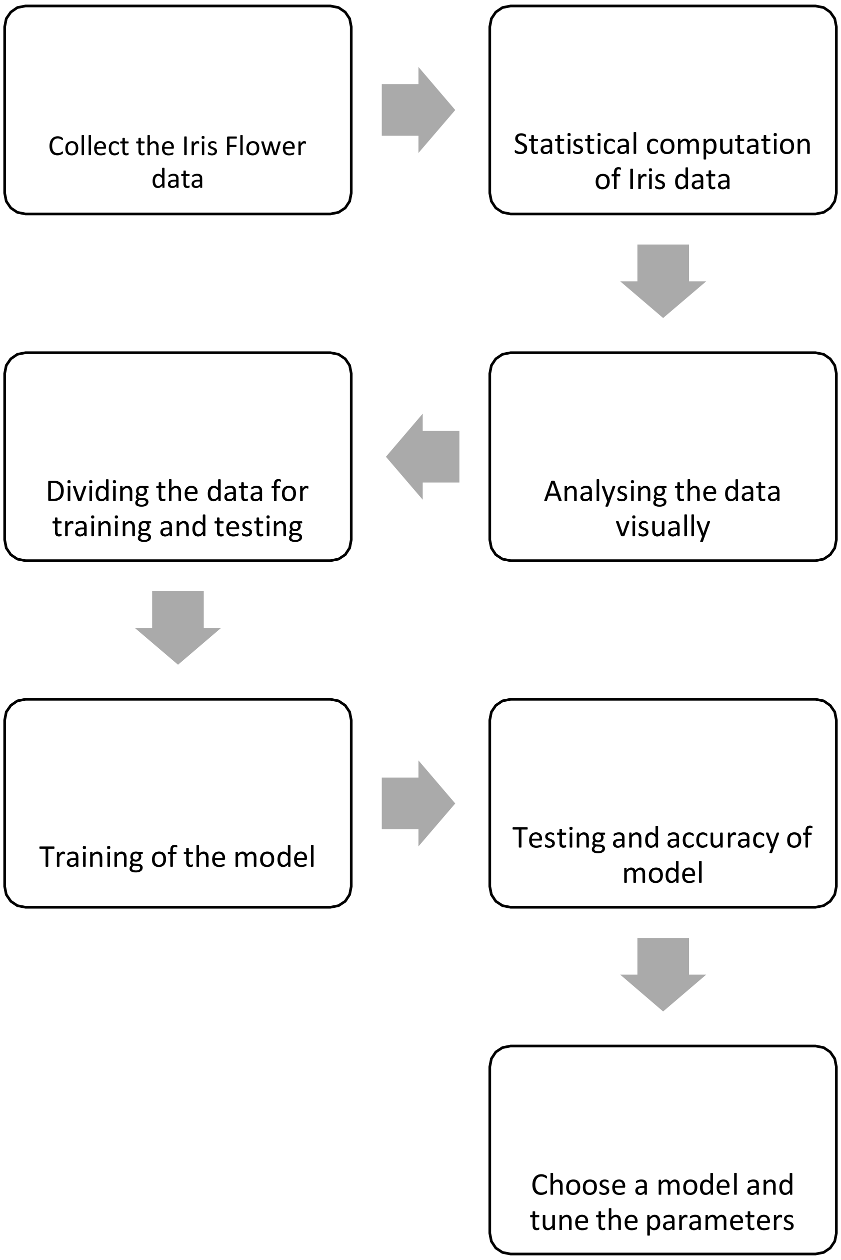
The Iris flower data set introduced by the British statistician and biologist Ronald Fisher that’s why is also known by Fisher's Iris data set and it is a multivariate data set. The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis.

The expectation from mining iris data set would be discovering patterns from examining sepal and petal size of the iris plant and how the prediction was made from analyzing the pattern to predict the class of iris plant. In upcoming years, using the classification and pattern recognition other flowers can be individually distinguish to each other. It is unmistakably expressed that the sort of relationship that being mined utilizing iris dataset would be a classification model.



PROPOSED MODEL

BLOCK DIAGRAM


Collect the Iris Flower data
Statistical computation of Iris data
Analysing the data visually
Dividing the data for training and testing
Training of the model
Testing and accuracy of model
Choose a model and tune the parameters

Data Set

A multivariant dataset used for machine learning purposes. The following dataset contains a set of 150 records under five attributes

- sepal length
- sepal width
- petal length
- petal width
- species In this data set we analyze three species of Iris flower, i-e Iris setosa , Iris versicolor , and Iris verginica.

Library Used

Pandas:-

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. Data analysis requires lots of processing, such as **restructuring,cleaning** or **merging**, etc. There are different tools are available for fast data processing, such as **NumPy, SciPy, Cython**, and **Panda**. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools.

Numpy:-

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Matplot:-

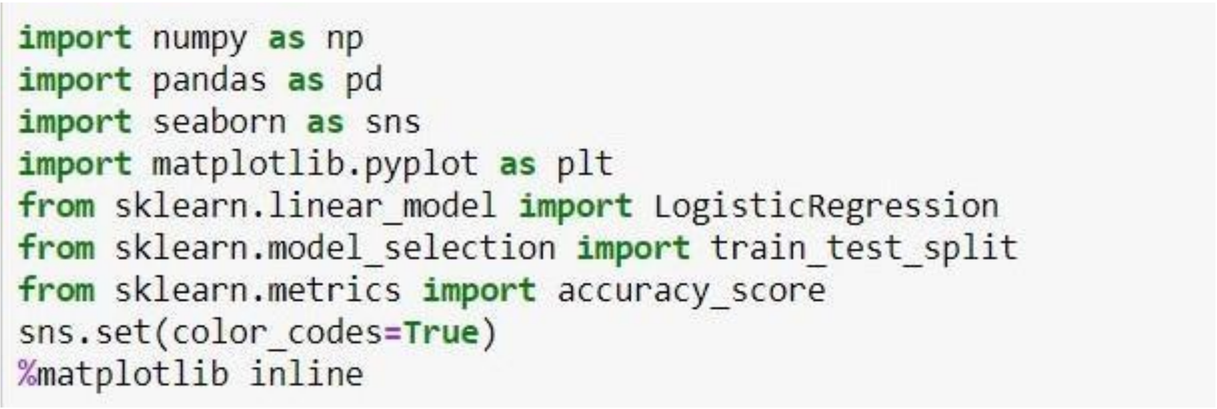
Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides a n object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Seaborn:-

Seaborn is one of an amazing library for visualization of the graphical statistical plotting in Python. Seaborn provides many color palettes and defaults beautiful styles to make the creation of many statistical plots in Python more attractive.

IMPORTING LIBRARIES AND DOWNLOAD THE DATA

The following libraries are required for this project:



```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
sns.set(color_codes=True)
%matplotlib inline
```

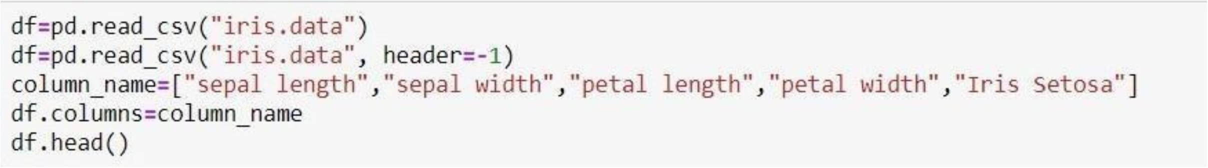
Fig.2 Imported Libraries

Here, we are importing numpy, pandas, seaborn, matplotlib and sklearn libraries. Where, numpy is an array processing package which is used in scientific computing with arrays. Pandas is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow us to store and manipulate tabular data in rows of observations and columns of variables. Seaborn is a library for statistical graphical representation and data visualization which is based on matplotlib. Matplotlib is a visualization library or plotting library used to generate plot, histogram, bar-chart, pie-chart, etc. Scikit-learn provides a range of machine learning algorithms which contains both unsupervised and supervised learning algorithms via a consistent interface in Python.

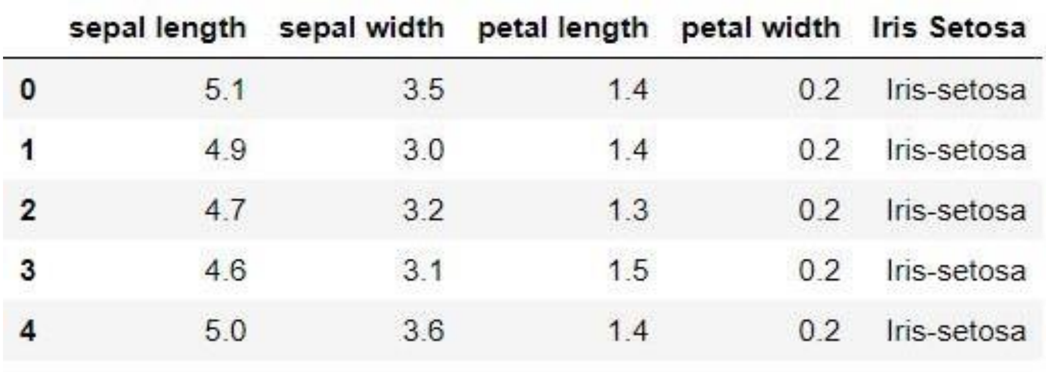
The iris dataset can be downloaded from the UCI Machine Learning Repository. Characteristics of data set is multivariate. This data set contain four attributes i.e. sepal length, sepal width, petal length, petal width in cm and it also contain three classes i.e. iris setosa, iris versicolour and iris virginica. The dataset downloaded from the UCI Machine Learning Repository is in the form of CSV (Comma Separated Values) file and the file name is ‘iris.data’ and save the file in the same directory as our project contain.

DATA EXPLORATION

Now we are going to move into data exploration as well as analysis using the iris data. Let’s import our data set using 'pandas' library, which will convert our data into the tabular format from the CSV format. The beauty of using pandas library is just that we can read the csv files. For converted our data into the understandable format we have to add column to the imported dataset which contain the attributes (sepal length, sepal width, petal length, petal width), it gives heading for the imported data.



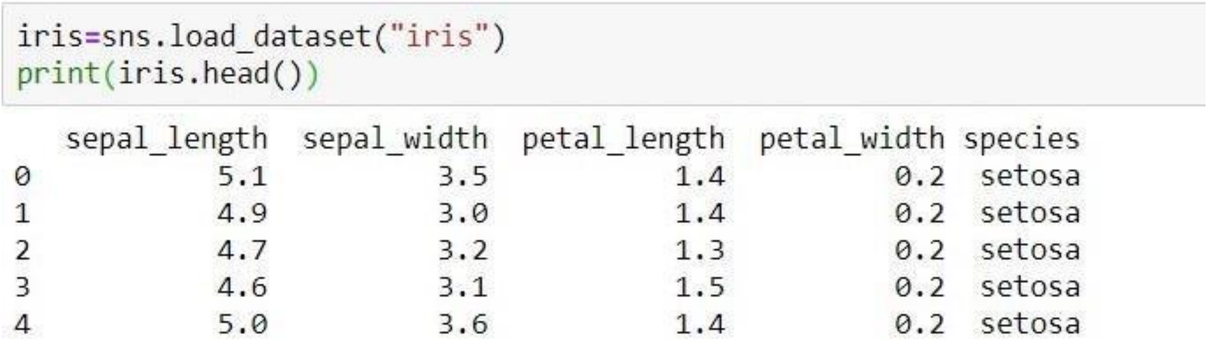
```
df=pd.read_csv("iris.data")
df=pd.read_csv("iris.data", header=-1)
column_name=["sepal length","sepal width","petal length","petal width","Iris Setosa"]
df.columns=column_name
df.head()
```



	sepal length	sepal width	petal length	petal width	Iris Setosa
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Table 1: Showing Iris Dataset using pandas Library

Or we can use seaborn instead of pandas as:



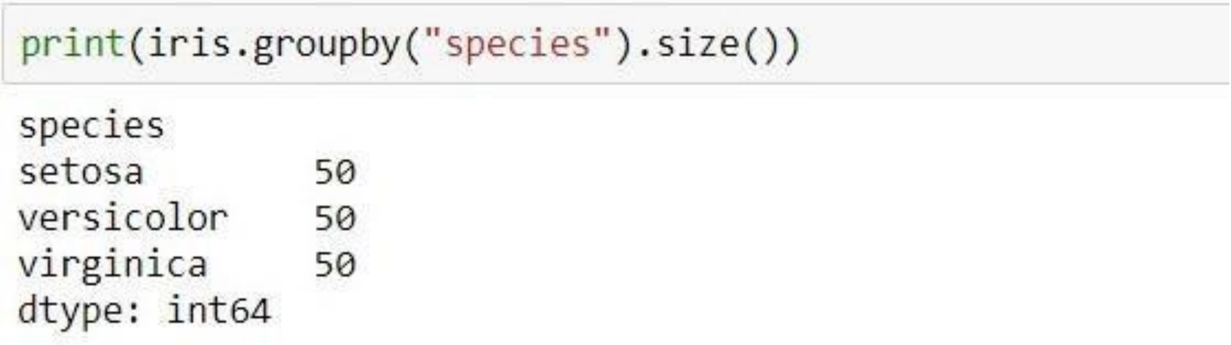
```
iris=sns.load_dataset("iris")
print(iris.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

Table 2: Showing Iris Dataset using seaborn Library

DATA ANALYSIS

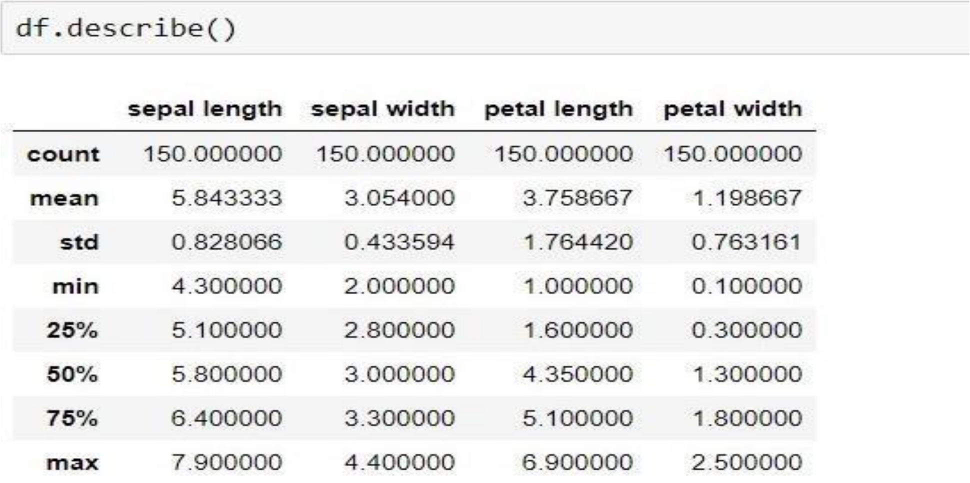
This dataset contains 150 samples Since the dataframe has four features (Petal length, petal width, sepal length and sepal width) with 150 samples belonging to either of the three target classes, and each class has distributed equally. Species in Iris Dataset



```
print(iris.groupby("species").size())
```

```
species
setosa        50
versicolor    50
virginica     50
dtype: int64
```

By using ‘df.describe()’ we can see the mathematics of the dataset, which helps to find out the standard deviation, mean, minimum value and the four quartile percentile of the data.

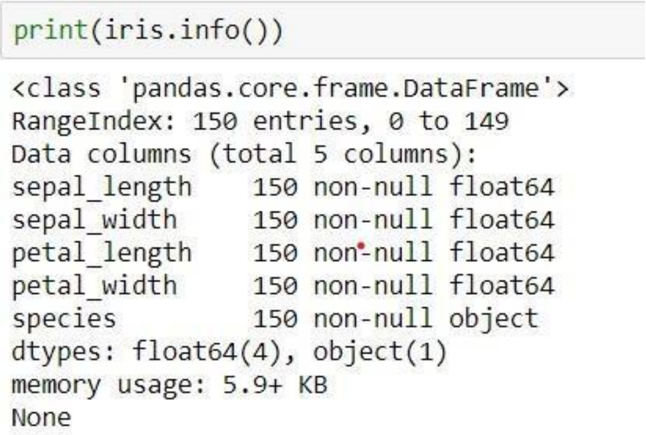


```
df.describe()
```

	sepal length	sepal width	petal length	petal width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Table 4: Statistical description of iris dataset

We can analyze some more information about our dataset, that it contains four non-null columns and one object-based column. We can also see memory usage by the iris dataset.



```
print(iris.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
sepal_length    150 non-null float64
sepal_width     150 non-null float64
petal_length    150 non-null float64
petal_width     150 non-null float64
species         150 non-null object
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
None
```

Information of iris dataset

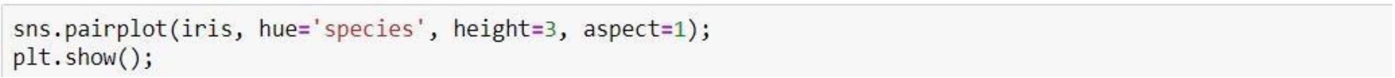
DATA VISUALIZATION

In the previous section what we gone through is the exploration of all the data where we did some preliminary analysis of the data and get a few of it, but to progress further and to dive into the data a little bit more we are going to do some visualization. Visualization is a great way to develop a better understanding of your data and python and has a lot of great tools for specifically that purpose.

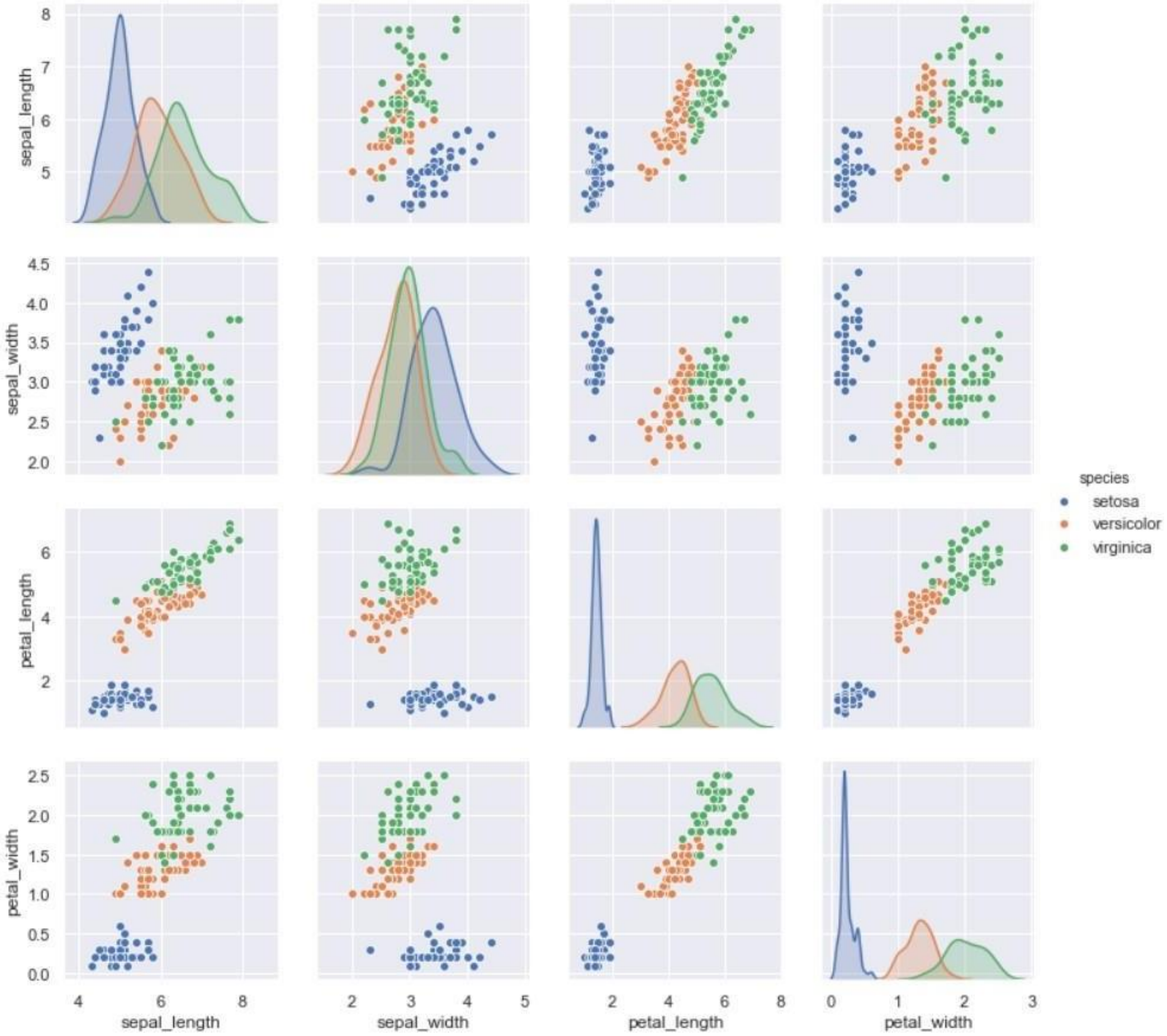
Pair-plot:

As we already import the seaborn so we just have to perform the pair-plot using the iris dataset we have. To understand how each feature accounts for classification of the data, we can build a pair-plotwhich shows us the correlation with respect to other features.

In the below picture if we look carefully, we can see that all the attributes are plotted against each other and the three different colours shows the distribution of three individual species (setosa, versicolor and verginica). It shows the distinctive relationships between the attributes.



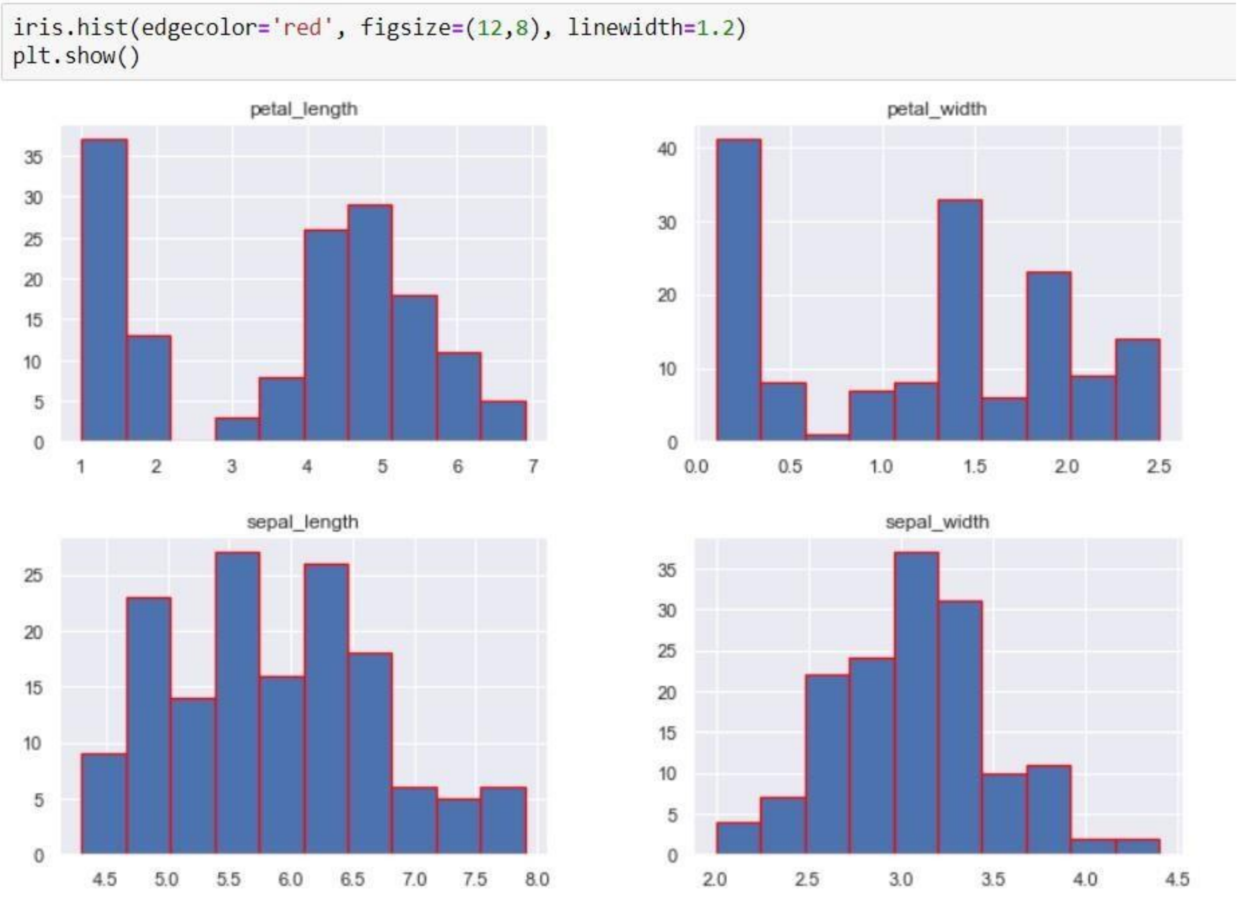
```
sns.pairplot(iris, hue='species', height=3, aspect=1);
plt.show();
```



Pair-plot

Histogram

Historical representation is basically the pure distribution off all three combined species and from this it's not really all that informative because it just tells us overall distribution.

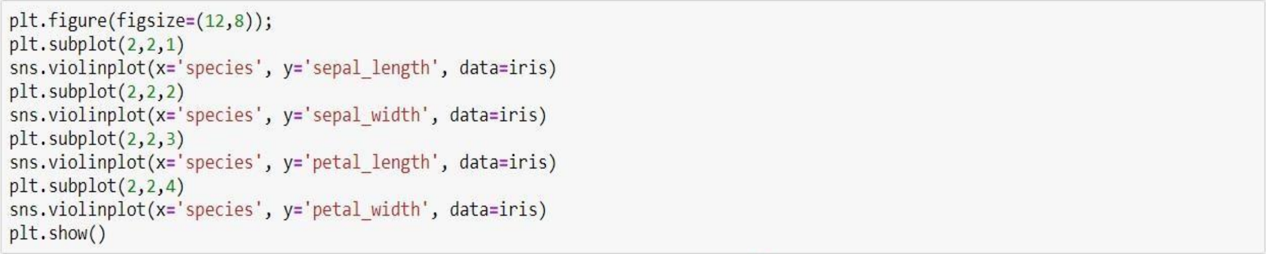


```
iris.hist(edgecolor='red', figsize=(12,8), linewidth=1.2)
plt.show()
```

Fig.5 Histogram

Violin-plot:

Let us look at the violin-plot which is as similar to the box-plot, Violin plot shows us the visual representation of how our data is scattered over the plane. Here we can conclude from the below picture that in sepal length we can see this the distribution in setosa is much smaller thanthe versicolor and verginica. In sepal width we can examine that the distribution of setosa is widest and also the longest sepal width and longest petal length in comparisons to the other attributes.



```
plt.figure(figsize=(12,8));
plt.subplot(2,2,1)
sns.violinplot(x='species', y='sepal_length', data=iris)
plt.subplot(2,2,2)
sns.violinplot(x='species', y='sepal_width', data=iris)
plt.subplot(2,2,3)
sns.violinplot(x='species', y='petal_length', data=iris)
plt.subplot(2,2,4)
sns.violinplot(x='species', y='petal_width', data=iris)
plt.show()
```

Output:

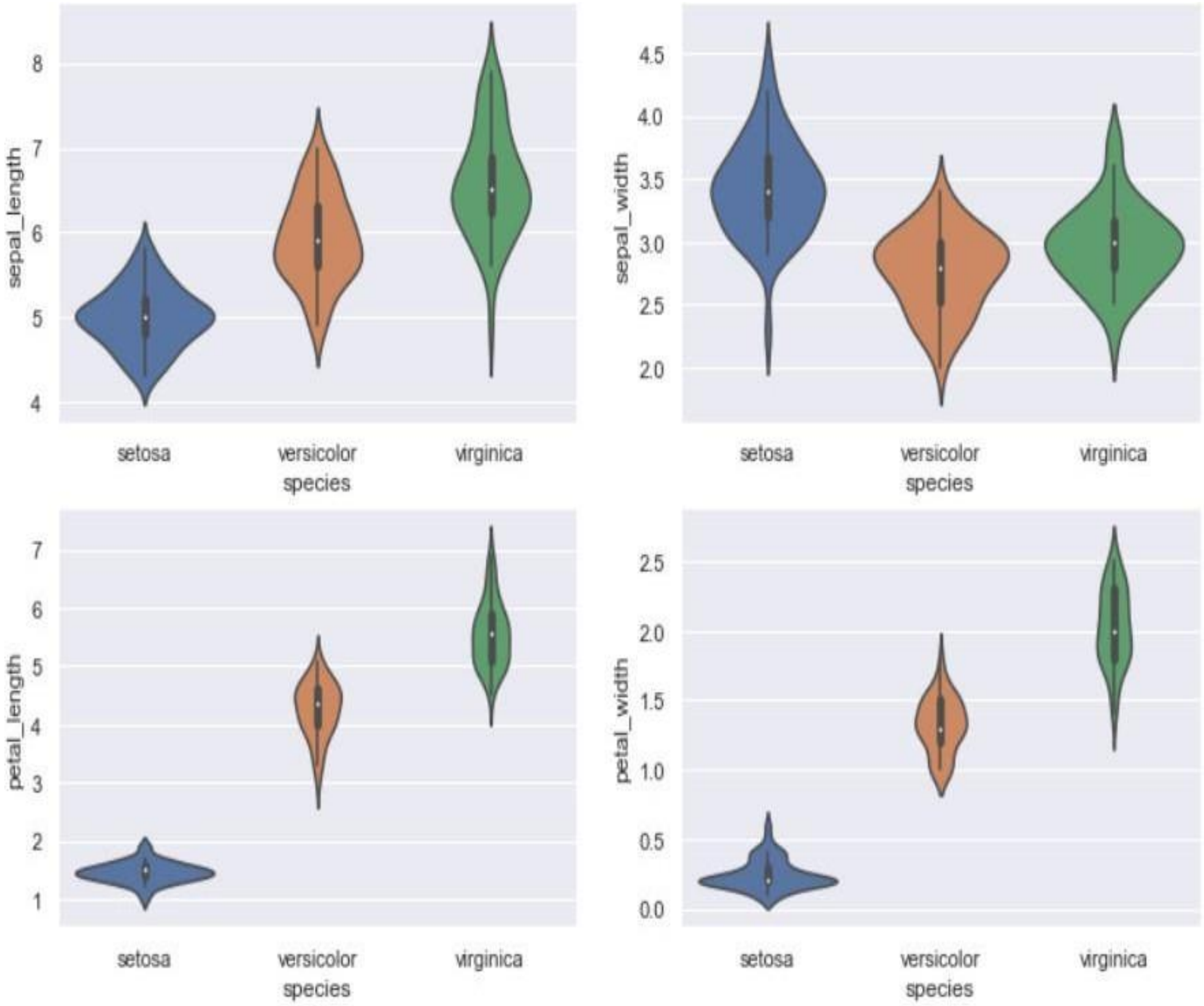
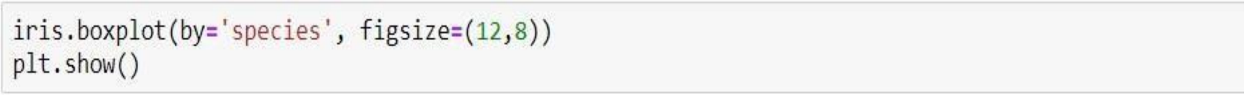


Fig.6 Violin-plot

Box-plot:

Box plot is a graph which is based on percentile, which divides the data into four quartiles of 25% each. This method is numerously used in statistical analysis to understand various measures such as max, min, mean, median and deviation.



```
iris.boxplot(by='species', figsize=(12,8))
plt.show()
```

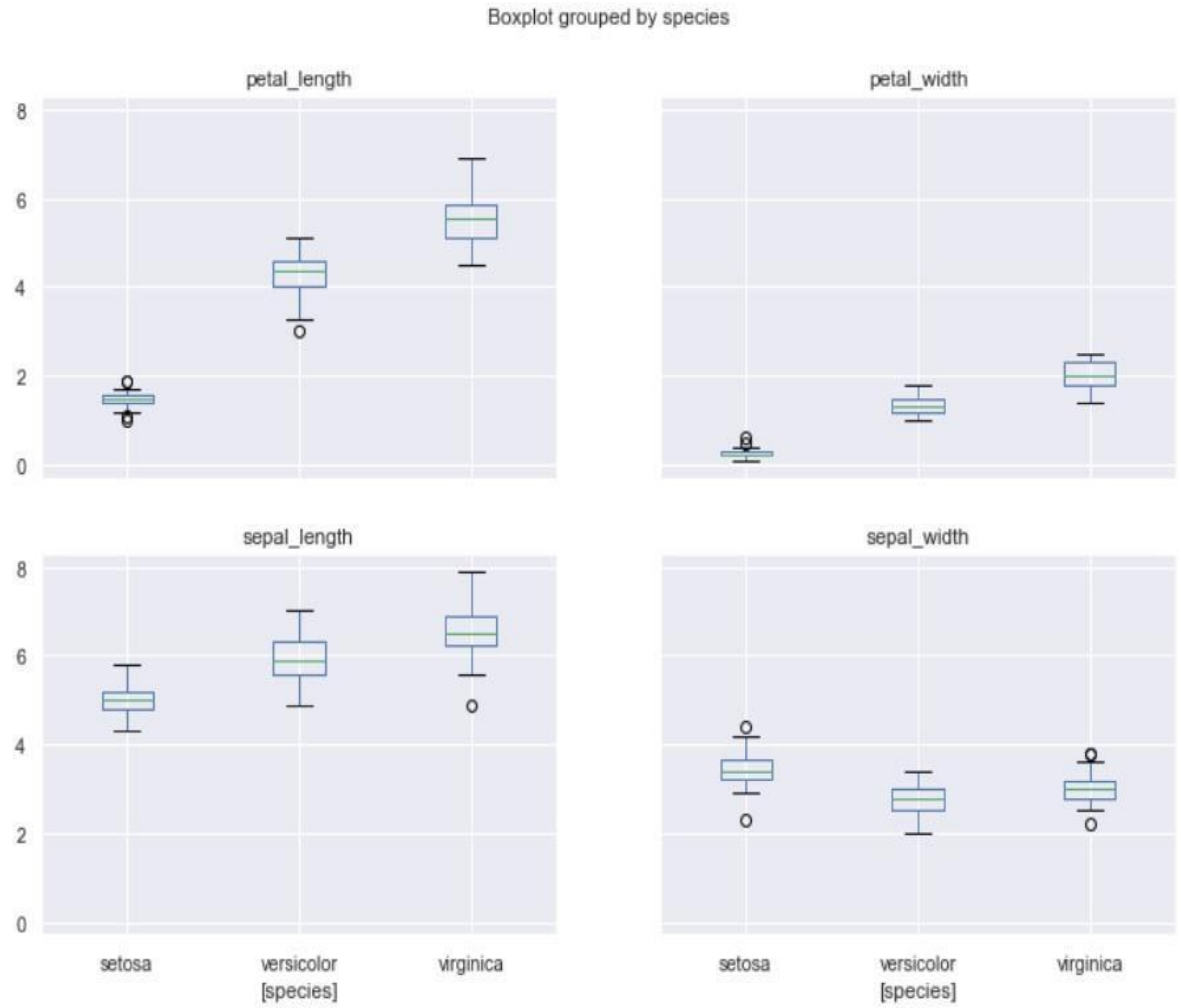
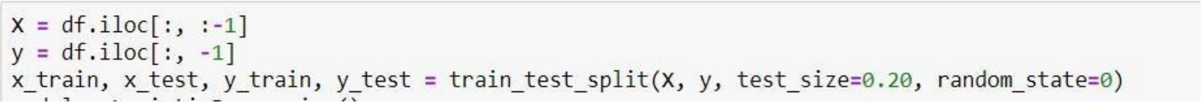


Fig.7 Box-plot

DIVIDING THE DATA FOR TRAINING AND TESTING

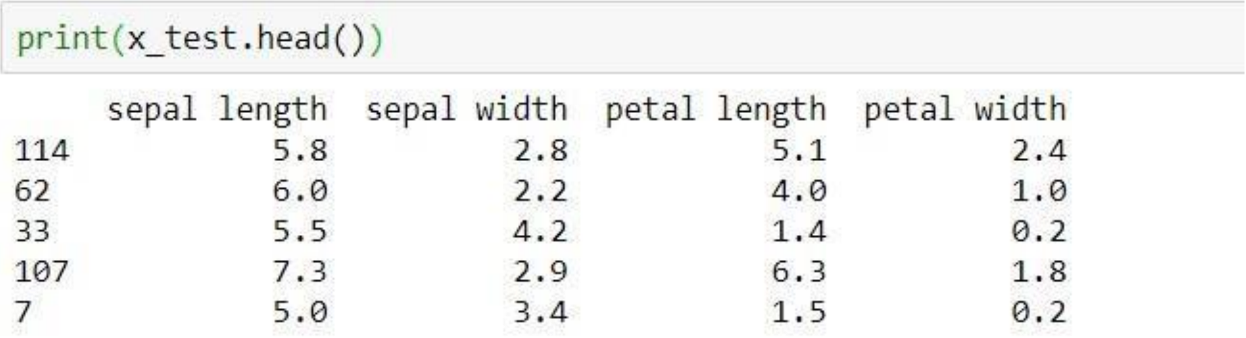
When we will understand what the dataset is about, we can start training our model based on the algorithms. First, we have to train our model with some of the samples. Here, we will be using scikit-learn library method called ‘train_test_split’ which divides our data set into a ratio of 80:20, in which 80% data will be using for training and 20% data will be using for testing. This process can be done by the following code:



```
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
```

Fig.8 Code for splitting dataset in training and testing dataset

Let’s see our test data,



```
print(x_test.head())
```

```
     sepal length  sepal width  petal length  petal width
114           5.8          2.8           5.1          2.4
62            6.0          2.2           4.0          1.0
33            5.5          4.2           1.4          0.2
107           7.3          2.9           6.3          1.8
7             5.0          3.4           1.5          0.2
```

Table 5: Test Data

ALGORITHM FOR TRAINING THE MODEL

LOGISTIC REGRESSION

Logistic Regression is a type of regression that predicts the probability of occurrence of an event by fitting the appropriate or cleaned data to a logistic function. Like several forms of regression analysis, it makes use of many predictor variables that will be either numerical or categorical. For instance, the probability that a email received is spam or not might be predicted from knowledge of the type of data or the history of sender. This regression is quite used in several scenarios such as prediction of customer's propensity of purchasing a product or used in market analysis to improve the business and use to predict the unpredictable scenarios.

What is logistic regression?

Logistic Regression, also known as Logit Model or Logit Regression, is a mathematical model used in statistics to estimate (guess) the probability of an event occurring having been given some previous data. Logistic Regression works with binary data, where either the event happens (True or 1) or the event does not happen (False or 0). So, by giving some feature x it tries to find out whether some event y happens or not. So, y can either be 0 or 1. In the case if the event happens, y is given the value 1. If the event does not happen, then y is given the value of 0. For example, if y represents whether a coin gives head, then y will be 1 if after tossing the coin we get head or y will be 0 if we get tail. This is known as Binomial Logistic Regression. Logistic Regression can also be used when there is use of multiple values for the variable y. This form of Logistic Regression is known as Multinomial Logistic Regression.

Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also called the Sigmoid function was developed by statisticians it’s an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1.

$$\frac{1}{1+e^{-x}}$$

Eq. 1

Where,

e: base of the natural logarithms

x: value that you want to transform via the logistic function The logistic regression equation has a very similar representation like linear regression. Thedifference is that the output value being modelled is binary in nature.

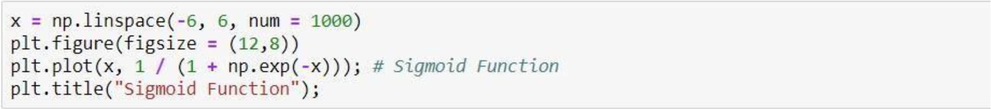
$$\hat{y} = \frac{e^{\beta_0+\beta_1 x_1}}{1+\beta_0+\beta_1 x_1}$$

or

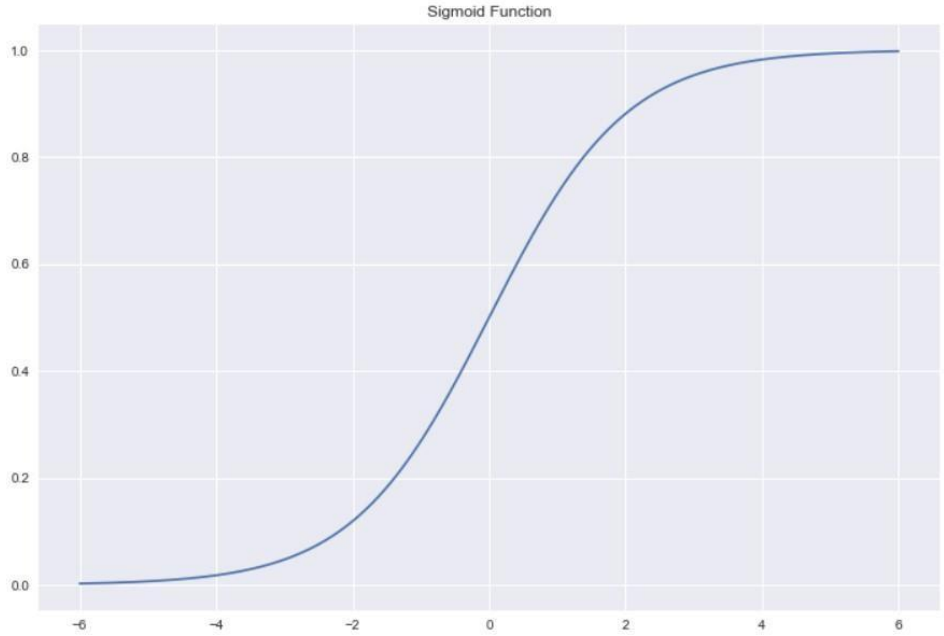
$$\hat{y} = \frac{1.0}{1.0+e^{-\beta_0-\beta_1 x_1}}$$

Eq. 2

β0: intercept term
β1: coefficient for x^y: predicted output with real value between 0 and 1



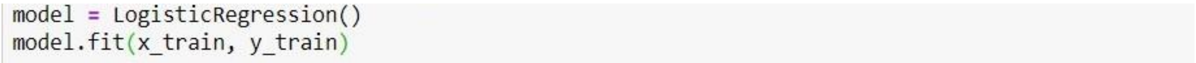
```
x = np.linspace(-6, 6, num = 1000)
plt.figure(figsize = (12,8))
plt.plot(x, 1 / (1 + np.exp(-x))); # Sigmoid Function
plt.title("Sigmoid Function");
```



TRAINING OF THE MODEL

For training of the data fit the 80% trained data into the model. By the below codes we can train our data.

By Logistic regression

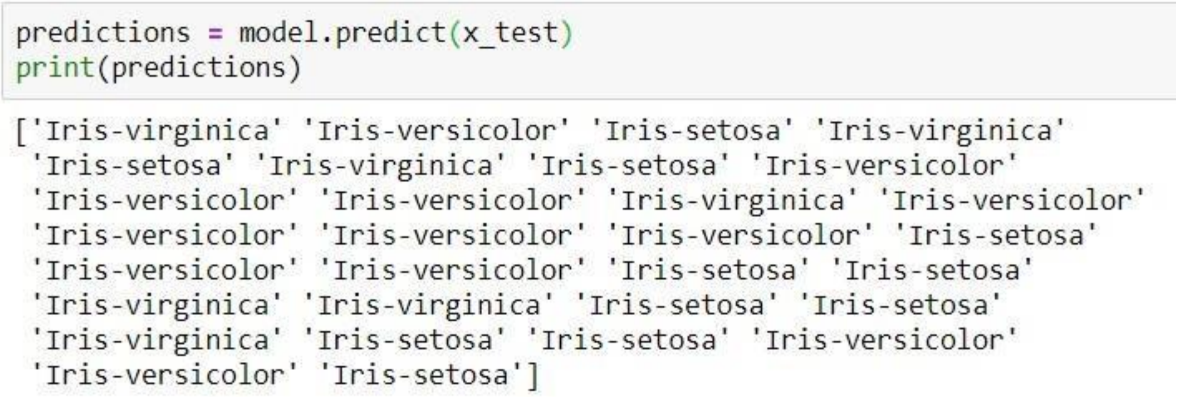


```
model = LogisticRegression()
model.fit(x_train, y_train)
```

PREDICT THE DATA AND ACCURACY OF THE MODEL

In this step we will predict iris species of our test data and also find the iris species of unknowndata i.e. data out of the box or can say that, data outside the taken dataset based on what it haslearnt in previous step and get the result

For test data:



```
predictions = model.predict(x_test)
print(predictions)
```

```
['Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa']
```

Create the Model (Classification)

So here we are going to classify the Iris flowers dataset using logistic regression. For creating the model, import LogisticRegression from the sci-kit learn library.

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

Now train the model using the fit method. In the fit method, pass training datasets in it. x_train and y_train are the training datasets.

```
model.fit(x_train,y_train)
```

Logistical Regression()

Now predict the results using predict method.

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

View the results now,

```
y_pred
array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-
setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica'], dtype=object)
```

It will give results like this. It contains species names in the form of an array.

Find the accuracy of the model and view the confusion matrix. The accuracy score tells us how accurately the model we build will predict and the confusion matrix has a matrix with Actual values and predicted values. For that, import accuracy_score and confusion_matrix from the sci-kit learn metric library.

```
from sklearn.metrics import accuracy_score,confusion_matrix
confusion_matrix(y_test,y_pred)
```

```
array([[13, 0, 0],
[ 0, 15, 1],
[ 0, 0, 9]], dtype=int64)
```

```
accuracy=accuracy_score(y_test,y_pred)*100
print("Accuracy of the model is {:.2f}".format(accuracy))
```

```
Accuracy of the model is 97.37
```

We can see that accuracy of the model is 97.37 percent which is very accurate.

Conclusion on Classification

This classification can be done by many classification algorithms in machine learning but in our article, we used logistic regression. Overall in this article, we have seen

- Mainly we focused on Logistic Regression
- We took Iris Flowers dataset and performed a logistic regression algorithm
- Finally, it classified flowers into their species.
- And we got an accuracy of 97.37%, which shows that the model we built is very accurate.

References:

- https://archive.ics.uci.edu/ml/datasets/iris
- https://www.neuraldesigner.com/learning/examples/iris-flowers-classification