	LetsGrowMore Name: V.Varsha Reddy Task:1-Iris Flower Classification ML Project
In [1]:	Dataset: http://archive.ics.uci.edu/ml/datasets/Iris Importing Data Manipulation and Data Visualization Libraries #importing the libraries required for the Data Manipulation import numpy as np
In [2]:	<pre>import pandas as pd #importing the libraries required for the Data Visualization import seaborn as sns import matplotlib.pyplot as plt import plotly.express as px</pre>
In [3]: In [4]: In [5]:	<pre>Importing the Dataset filename = ("iris.csv") df = pd.read_csv(filename) df</pre>
Out[5]:	Id SepalLength SepalWidth PetalWidth Species 0 1 5.1 3.5 1.4 0.2 Iris-setosa 1 2 4.9 3.0 1.4 0.2 Iris-setosa 2 3 4.7 3.2 1.3 0.2 Iris-setosa 3 4 4.6 3.1 1.5 0.2 Iris-setosa
	4 5 5.0 3.6 1.4 0.2 Iris-setosa 145 147 6.7 3.0 5.2 2.3 Iris-virginica 146 148 6.3 2.5 5.0 1.9 Iris-virginica 147 149 6.5 3.0 5.2 2.0 Iris-virginica
	148 150 6.2 3.4 5.4 2.3 Iris-virginica 149 151 5.9 3.0 5.1 1.8 Iris-virginica 150 rows × 6 columns
In [6]:	<pre>Processing the Data df.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 150 entries, 0 to 149 Data columns (total 6 columns): # Column Non-Null Count Dtype</class></pre>
	0 Id 150 non-null int64 1 SepalLength 150 non-null float64 2 SepalWidth 150 non-null float64 3 PetalLength 150 non-null float64 4 PetalWidth 150 non-null float64 5 Species 150 non-null object dtypes: float64(4), int64(1), object(1) memory usage: 7.2+ KB
In [7]: Out[7]:	SepalWidth 0 PetalLength 0 PetalWidth 0 Species 0
Out[8]:	<pre>dtype: int64 df.columns Index(['Id', 'SepalLength', 'SepalWidth', 'PetalLength', 'PetalWidth',</pre>
Out[9]:	count 150.000000 150.000000 150.000000 150.000000 150.000000 150.000000 150.00000
	25% 38.250000 5.100000 2.800000 1.600000 0.300000 50% 75.500000 5.800000 3.00000 4.350000 1.300000 75% 112.750000 6.400000 3.300000 5.100000 1.800000 max 151.000000 7.900000 4.400000 6.900000 2.500000
<pre>In [10]: In [11]: Out[11]:</pre>	
	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
	145 6.7 3.0 5.2 2.3 Iris-virginica 146 6.3 2.5 5.0 1.9 Iris-virginica 147 6.5 3.0 5.2 2.0 Iris-virginica 148 6.2 3.4 5.4 2.3 Iris-virginica 149 5.9 3.0 5.1 1.8 Iris-virginica
In [12]: In [13]:	150 rows × 5 columns print(df.duplicated().sum()) df.SepalLength.value_counts()
Out[13]:	5.0 10 5.1 9 6.3 9 5.7 8 6.7 8 5.8 7 5.5 7 6.4 7 4.9 6
	5.4 6 6.1 6 6.0 6 5.6 6 4.8 5 6.5 5 6.2 4 7.7 4 6.9 4
	4.6 4 5.2 4 5.9 3 4.4 3 7.2 3 6.8 3 6.6 2 4.7 2
	7.6 1 7.4 1 7.3 1 7.0 1 7.1 1 5.3 1 4.3 1 4.5 1 7.9 1
In [14]: Out[14]:	2.8 14 3.2 13 3.1 12 3.4 12
	2.9 10 2.7 9 2.5 8 3.5 6 3.3 6 3.8 6 2.6 5 2.3 4 3.7 3
	2.4 3 2.2 3 3.6 3 3.9 2 4.4 1 4.0 1 4.1 1 4.2 1 2.0 1
In [15]: Out[15]:	3.2
	2.9
	2.4 0.020000 2.2 0.020000 3.6 0.020000 3.9 0.013333 4.4 0.006667 4.1 0.006667 4.1 0.006667 4.2 0.006667 2.0 0.006667
In [16]: Out[16]:	<pre>Name: SepalWidth, dtype: float64 df.describe(include=["object", "bool"])</pre>
In [17]: Out[17]:	
	118 7.7 2.6 6.9 2.3 Iris-virginica 122 7.7 2.8 6.7 2.0 Iris-virginica 117 7.7 3.8 6.7 2.2 Iris-virginica 105 7.6 3.0 6.6 2.1 Iris-virginica 131 7.9 3.8 6.4 2.0 Iris-virginica
	df.dropna(axis=0,inplace=True) sepal = df.groupby ('SepalLength').SepalWidth.sum().to_frame().reset_index() sepal.columns = ['SepalLength', 'SepalWidth'] px.bar(data_frame = sepal, x='SepalLength', y='SepalWidth', color = 'SepalWidth') • Q • • • • • • • • • • • • • • • • •
	SepalWidth 30 25
	Separate to the second
	10 5 4.5 5 5.5 6 6.5 7 7.5
In [20]:	<pre>petal = df.groupby ('PetalLength').PetalWidth.sum().to_frame().reset_index() petal.columns = ['PetalLength', 'PetalWidth'] px.bar(data_frame =petal, x='PetalLength', y='PetalWidth', color = 'PetalWidth')</pre>
	PetalWidth 14
	O 1 2 3 4 5 6 PetalLength Define X and Y
In [21]: In [22]: Out[22]:	<pre># defining x and y x = df.iloc[:,:4] y = df.iloc[:,4]</pre> x
	0 5.1 3.5 1.4 0.2 1 4.9 3.0 1.4 0.2 2 4.7 3.2 1.3 0.2 3 4.6 3.1 1.5 0.2 4 5.0 3.6 1.4 0.2
	145 6.7 3.0 5.2 2.3 146 6.3 2.5 5.0 1.9 147 6.5 3.0 5.2 2.0 148 6.2 3.4 5.4 2.3
In [23]: Out[23]:	
	Iris-setosa Iris-setosa Iris-setosa Iris-setosa Iris-setosa Iris-setosa Iris-virginica Iris-virginica Iris-virginica Iris-virginica Iris-virginica Iris-virginica
In [24]:	149 Iris-virginica Name: Species, Length: 150, dtype: object splitting the data into train and test datasets from sklearn.model_selection import train_test_split x_train,x_test,y_train,y_test=train_test_split(x,y,random_state = 0)
	<pre>print("x_train:", x_train.shape) x_train: (112, 4) print("x_test:", x_test.shape) x_test: (38, 4) print("y_train:", y_train_shape)</pre>
	<pre>print("y_train:",y_train.shape) y_train: (112,) print("y_test:",y_train.shape) y_test: (112,) Create The Model</pre>
In [29]:	#creating the model from sklearn.linear_model import LogisticRegression clf = LogisticRegression() Train the Model
Out[30]:	Predict The Results
	<pre>clf.predict(x_test) array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',</pre>
In [32]:	'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-virginica', 'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-virginica'], dtype=object) y_test
Out[32]:	Iris-versicolor Iris-versicolor Iris-setosa Iris-setosa Inis-setosa Iris-setosa Iris-setosa Iris-versicolor Iris-versicolor
	Tris-versicolor
	Iris-setosa Iris-setosa Iris-virginica Iris-versicolor Iris-setosa Iris-setosa Iris-setosa Iris-setosa Iris-setosa Iris-setosa Iris-setosa
	Iris-setosa Iris-setosa Iris-versicolor Iris-versicolor Iris-versicolor Iris-virginica Iris-virginica Iris-virginica Iris-virginica Iris-virginica Iris-virginica Iris-virginica Iris-virginica
	Iris-virginica Firs-versicolor
In []:	