```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
iris= pd.read_csv("/content/IRIS.csv")

iris
```

	sepal_length	sepal_width	petal_length	petal_width	species
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
...	...	...	...	...	...
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

150 rows × 5 columns

Next steps: Generate code with `iris` View recommended plots

```
iris.head()#first five rows of the dataset
```

	sepal_length	sepal_width	petal_length	petal_width	species
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

Next steps: Generate code with `iris` View recommended plots

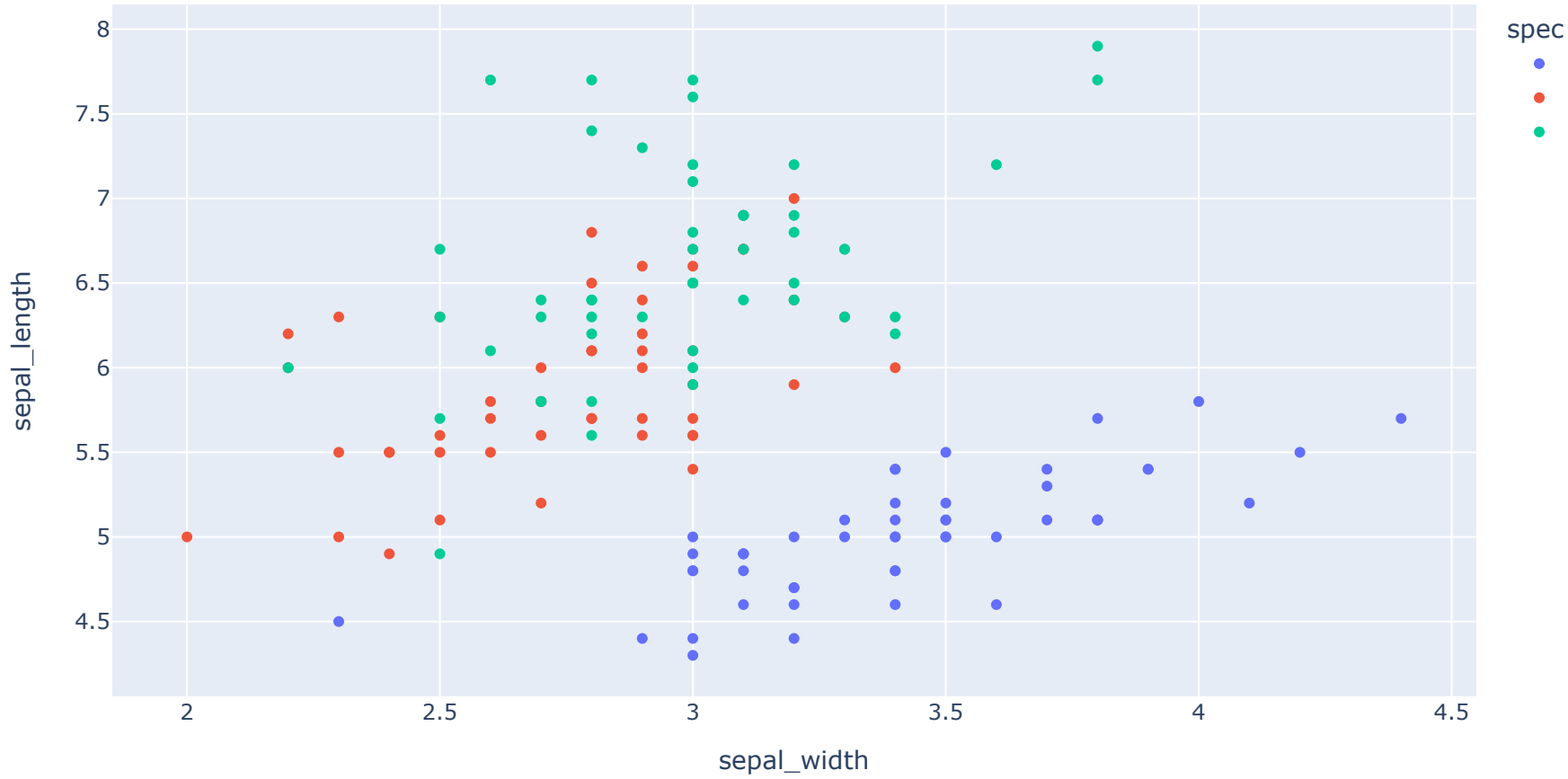
```
iris.describe()#last five rows of the dataset
```

	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

```
print("Target Labels",iris["species"].unique())  #target values
```

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

```
#scatter plot according to sepal length and sepal width
import plotly.express as px
fig=px.scatter(iris,x="sepal_width",y="sepal_length",color="species")
fig.show()
```




```
#KNN classification algorithm to train iris classification model

x= iris.drop("species",axis=1)
y= iris["species"]
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x,y,test_size=0.2, random_state=0)  #splitting data into training and
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier(n_neighbors=1)
knn.fit(x_train, y_train)
```

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

```
▾ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=1)
```

```
#input a set of measurements of iris flower and use the model to predict iris species
x_new = np.array([[5, 2.9, 1, 0.2]])
prediction = knn.predict(x_new)
print("Prediction: {}".format(prediction))
```

```
Prediction: ['Iris-setosa']
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning:

X does not have valid feature names, but KNeighborsClassifier was fitted with feature names
```

So this is how you can train a ML model for task of Iris Flo
Classification using Python.

So this is how you can train a ML model for task of Iris Flower Classification using Python.