

```
In [1]: import pandas as pd
        from sklearn.tree import DecisionTreeClassifier
```

```
In [2]: df = pd.read_csv(r"C:\Users\hp\Documents\Datasets\IRIS.csv")
        df.head()
```

```
Out[2]:
```

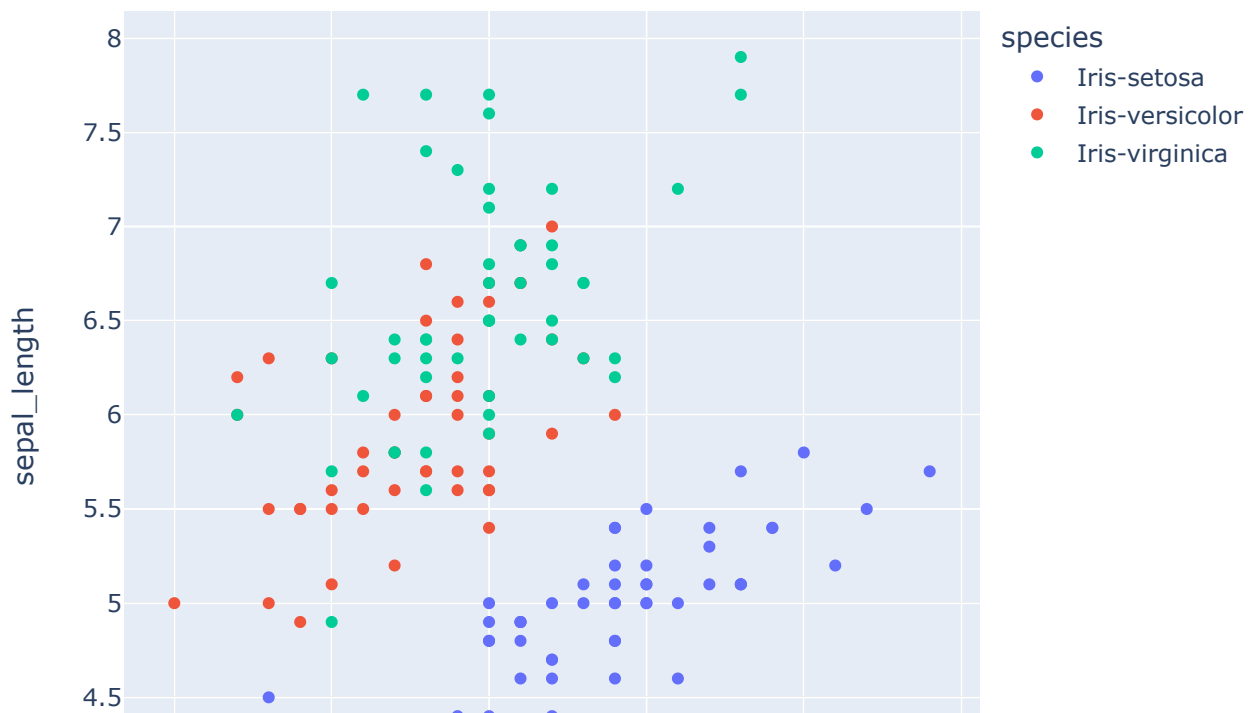
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
In [3]: ## Checking any missing values
        df.isnull().sum()
```

```
Out[3]: sepal_length    0
        sepal_width     0
        petal_length    0
        petal_width     0
        species         0
        dtype: int64
```

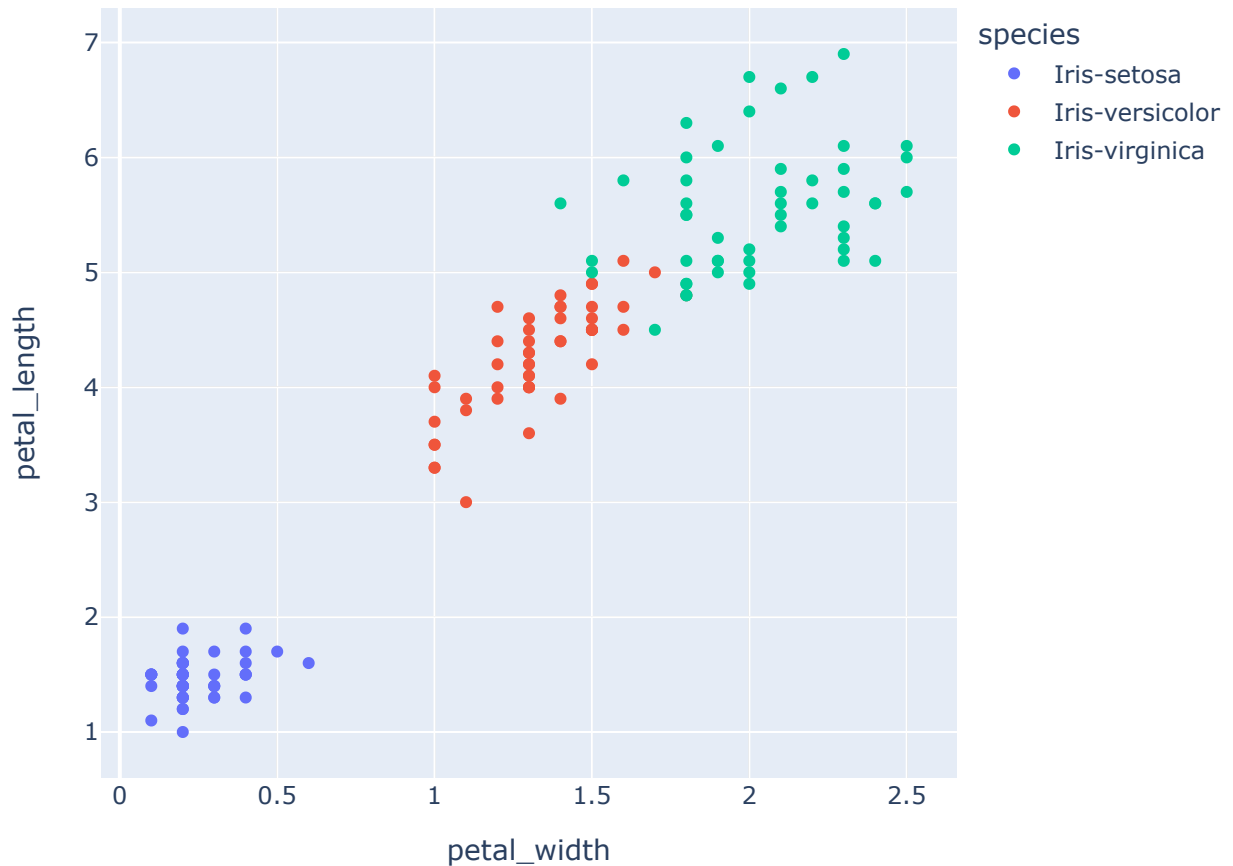
## Visualisation part

```
In [4]: import plotly.express as px
        fig = px.scatter(df, x="sepal_width", y="sepal_length", color="species")
        fig.show()
```





```
In [5]: import plotly.express as px
fig = px.scatter(df, x="petal_width", y="petal_length", color="species")
fig.show()
```



```
In [6]: ## Splitting Data
X = df.drop(["species"],axis=1)
Y = df["species"]
```

```
In [7]: from sklearn.model_selection import train_test_split
```

```
In [8]: X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=1)
```

```
In [9]: clf = DecisionTreeClassifier()
```

```
In [10]: model = clf.fit(X_train,Y_train)
```

```
In [13]: Y_pred = model.predict(X_test)
Y_pred
```

```
Out[13]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
        'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
        'Iris-setosa', 'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
        'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
        'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
        'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
        'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
        'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
        'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```
In [12]: model.score(X_test,Y_test)
```

```
Out[12]: 0.9666666666666667
```

```
In [14]: import matplotlib.pyplot as plt
from sklearn import tree
```

```
In [15]: fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(clf,
                  feature_names=X.columns,
                  class_names=Y.unique(),
                  filled=True)
```

