

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

```
df=pd.read_csv('/content/drive/MyDrive/Iris.csv')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 150 entries, 0 to 149
```

```
Data columns (total 6 columns):
```

#	Column	Non-Null Count	Dtype
0	Id	150 non-null	int64
1	SepalLengthCm	150 non-null	float64
2	SepalWidthCm	150 non-null	float64
3	PetalLengthCm	150 non-null	float64
4	PetalWidthCm	150 non-null	float64
5	Species	150 non-null	object

```
dtypes: float64(4), int64(1), object(1)
```

```
memory usage: 7.2+ KB
```

```
df.head()
```

```
{
  "summary": {
    "name": "df",
    "rows": 150,
    "fields": [
      {
        "column": "Id",
        "properties": {
          "dtype": "number",
          "std": 43,
          "min": 1,
          "max": 150,
          "num_unique_values": 150,
          "samples": [74, 19, 119]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "SepalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 0.8280661279778629,
          "min": 4.3,
          "max": 7.9,
          "num_unique_values": 35,
          "samples": [6.2, 4.5, 5.6]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "SepalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 0.4335943113621737,
          "min": 2.0,
          "max": 4.4,
          "num_unique_values": 23,
          "samples": [2.3, 4.0, 3.5]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "PetalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 1.7644204199522617,
          "min": 1.0,
          "max": 6.9,
          "num_unique_values": 43,
          "samples": [6.7, 3.8, 3.7]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "PetalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 0.7631607417008414,
          "min": 0.1,
          "max": 2.5,
          "samples": [0.1, 2.5, 0.1]
        },
        "semantic_type": "",
        "description": ""
      }
    ]
  }
}
```

```

\"num_unique_values\": 22,\n          \"samples\": [\n          0.2,\n1.2,\n          1.3\n          ],\n          \"semantic_type\": \"\",\n\"description\": \"\"\n          }\n          },\n          {\n          \"column\":\n\"Species\",\n          \"properties\": {\n          \"dtype\":\n\"category\",\n          \"num_unique_values\": 3,\n          \"samples\":\n[\n          \"Iris-setosa\",\n          \"Iris-versicolor\",\n          \"Iris-virginica\"\n          ],\n          \"semantic_type\": \"\",\n\"description\": \"\"\n          }\n          }\n          ]\n          }", "type": "dataframe", "variable_name": "df"}

```

```

from sklearn.preprocessing import LabelEncoder
LE=LabelEncoder()
df["Species"]=LE.fit_transform(df["Species"])

x_train,x_test,y_train,y_test=train_test_split(
    df.drop("Species",axis=1),
    df["Species"],
    test_size=0.2
)

```

```

Dt=DecisionTreeClassifier()
Dt.fit(x_train,y_train)

```

```

DecisionTreeClassifier()

```

```

from sklearn.tree import plot_tree
plot_tree(Dt,filled=True,max_depth=5)

```

```

[Text(0.6, 0.8333333333333334, 'x[0] <= 100.5\ngini = 0.665\nsamples = 120\nvalue = [37, 39, 44]'),
 Text(0.4, 0.5, 'x[4] <= 0.7\ngini = 0.5\nsamples = 76\nvalue = [37, 39, 0]'),
 Text(0.5, 0.6666666666666667, 'True '),
 Text(0.2, 0.16666666666666666, 'gini = 0.0\nsamples = 37\nvalue = [37, 0, 0]'),
 Text(0.6, 0.16666666666666666, 'gini = 0.0\nsamples = 39\nvalue = [0, 39, 0]'),
 Text(0.8, 0.5, 'gini = 0.0\nsamples = 44\nvalue = [0, 0, 44]'),
 Text(0.7, 0.6666666666666667, ' False')]

```

