```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
df=pd.read_csv('/content/iris.csv')

df.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

```
df.tail()
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
df.shape
```

```
(150, 5)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df.isnull().sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

```
df.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000

```
df['species'].unique()
```

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

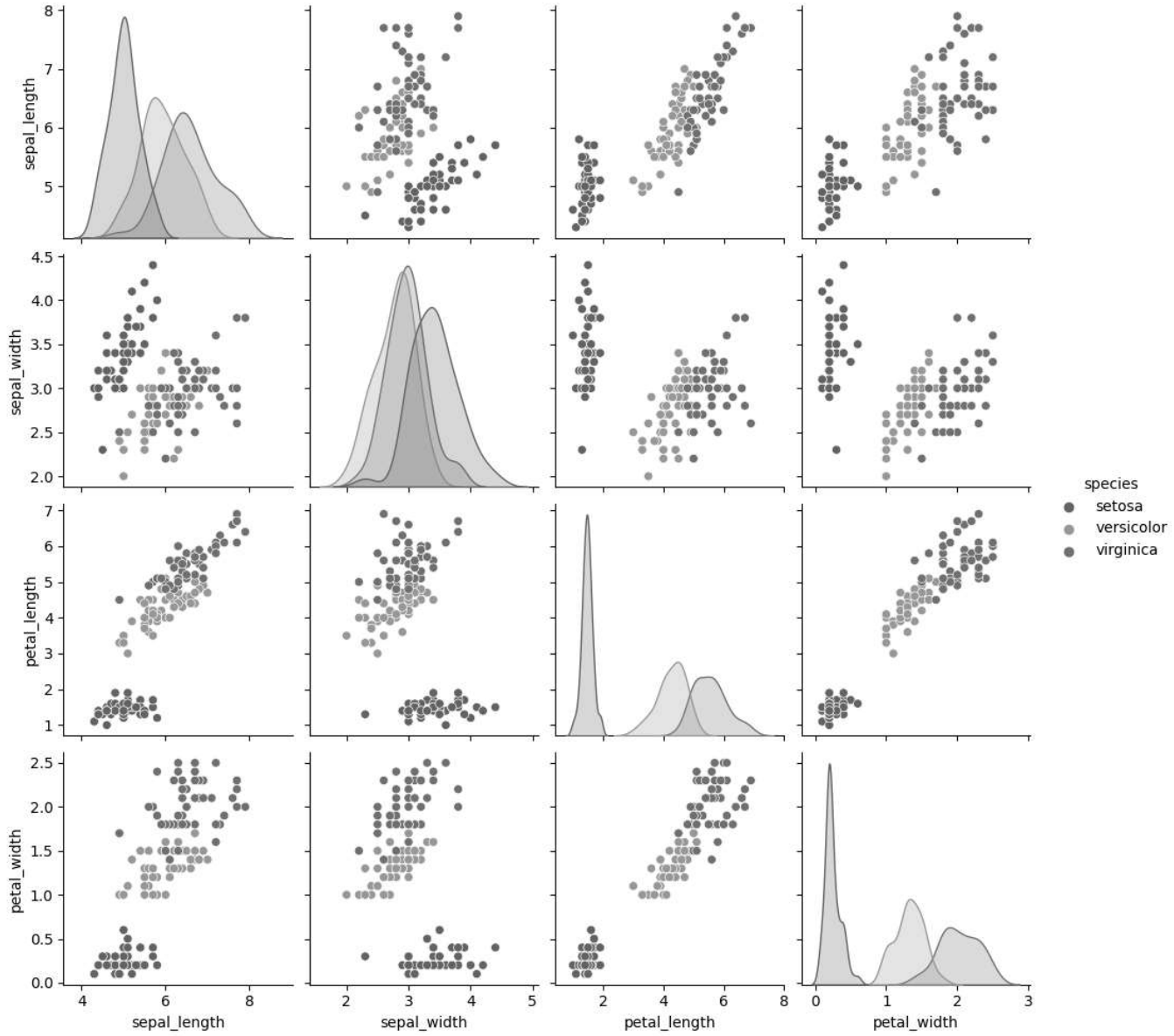
```
df['species'].value_counts()
```

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

max	7.900000	4.400000	6.900000	2.500000

```
sns.pairplot(df,hue='species')
```

```
<seaborn.axisgrid.PairGrid at 0x7e4ab16bb3d0>
```

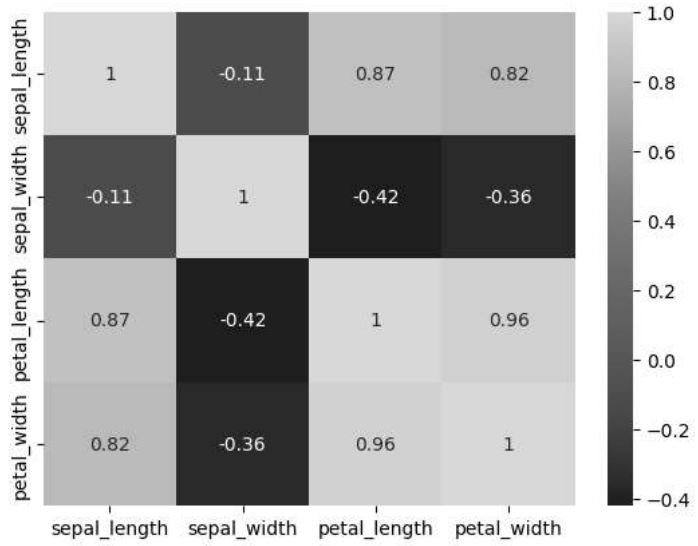



```
df.corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

```
sns.heatmap(df.corr(),annot=True,cmap='viridis')
```

```
<Axes: >
```




```
# X contains feature columns
X=df.drop(['species'],axis=1)
# y contain target column
y=df['species']
```

```
X.shape
```

```
(150, 4)
```

```
y.shape
```

```
(150,)
```

```
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=42)
```

```
X_train.shape
```

```
(120, 4)
```

```
y_train.shape
```

```
(120,)
```

```
X_test.shape
```

```
(30, 4)
```

```
y_test.shape
```

```
(30,)
```

```
from sklearn.tree import DecisionTreeClassifier

DTC=DecisionTreeClassifier ()
```

```
DTC.fit(X_train,y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
prediction=DTC.predict(X_test)
```

```
compare=pd.DataFrame({'Actual':y_test,'Prediction':prediction})
compare
```

	Actual	Prediction
73	versicolor	versicolor
18	setosa	setosa
118	virginica	virginica
78	versicolor	versicolor
76	versicolor	versicolor
31	setosa	setosa
64	versicolor	versicolor
141	virginica	virginica
68	versicolor	versicolor
82	versicolor	versicolor
110	virginica	virginica
12	setosa	setosa
36	setosa	setosa
9	setosa	setosa
19	setosa	setosa
56	versicolor	versicolor
104	virginica	virginica
69	versicolor	versicolor
55	versicolor	versicolor
132	virginica	virginica
29	setosa	setosa
127	virginica	virginica
26	setosa	setosa
128	virginica	virginica
131	virginica	virginica
145	virginica	virginica
108	virginica	virginica
143	virginica	virginica
45	setosa	setosa
30	setosa	setosa

```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
```

```
print(classification_report(y_test,prediction))
```

```
                precision    recall  f1-score   support

           setosa       1.00      1.00      1.00        10
       versicolor       1.00      1.00      1.00         9
        virginica       1.00      1.00      1.00        11

         accuracy                           1.00        30
        macro avg       1.00      1.00      1.00        30
```

```
weighted avg       1.00      1.00      1.00        30
```

```
print(confusion_matrix(y_test,prediction))
```

```
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

```
Accuracy = accuracy_score(y_test,prediction)
```

```
Precision =precision_score(y_test, prediction,average='weighted')
Precision
```

```
1.0
```

```
Sensitivity_recall = recall_score(y_test,prediction,average='weighted')
Sensitivity_recall
```

```
1.0
```

```
Specificity = recall_score(y_test, prediction, average='weighted')

Specificity
```

```
1.0
```

```
from sklearn.tree import plot_tree

plt.figure(figsize=(20,10))
tree=plot_tree(DTC,feature_names=X.columns,precision=2,rounded=True,filled=True,class_names=y.values)
```