```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn import datasets
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

data = pd.read_csv('Downloads/iris.csv')
data
```

	Unnamed: 0	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	1	5.1	3.5	1.4	0.2	setosa
1	2	4.9	3.0	1.4	0.2	setosa
2	3	4.7	3.2	1.3	0.2	setosa
3	4	4.6	3.1	1.5	0.2	setosa
4	5	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	virginica
146	147	6.3	2.5	5.0	1.9	virginica
147	148	6.5	3.0	5.2	2.0	virginica
148	149	6.2	3.4	5.4	2.3	virginica
149	150	5.9	3.0	5.1	1.8	virginica

150 rows × 6 columns

```
data1=preprocessing.LabelEncoder()
data['Species']=data1.fit_transform(data['Species'])
data
```

	Unnamed: 0	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	1	5.1	3.5	1.4	0.2	0
1	2	4.9	3.0	1.4	0.2	0
2	3	4.7	3.2	1.3	0.2	0
3	4	4.6	3.1	1.5	0.2	0
4	5	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	2
146	147	6.3	2.5	5.0	1.9	2
147	148	6.5	3.0	5.2	2.0	2
148	149	6.2	3.4	5.4	2.3	2
149	150	5.9	3.0	5.1	1.8	2

150 rows × 6 columns

```
x=data.iloc[:,0:5]
y=data['Species']
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=30)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

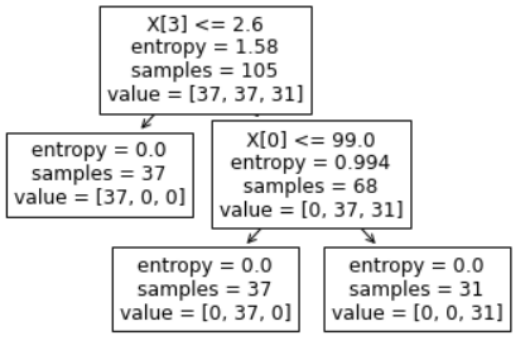
```
((105, 5), (45, 5), (105,), (45,))
```

```
model = DecisionTreeClassifier(criterion='entropy',max_depth=30)
model.fit(x_train,y_train)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=30)
```

```
tree.plot_tree(model)
```

```
[Text(133.92000000000002, 181.2, 'X[3] <= 2.6\nentropy = 1.58\nsamples = 105\nvalue = [37, 37, 31]'),
 Text(66.96000000000001, 108.72, 'entropy = 0.0\nsamples = 37\nvalue = [37, 0, 0]'),
 Text(200.88000000000002, 108.72, 'X[0] <= 99.0\nentropy = 0.994\nsamples = 68\nvalue = [0, 37, 31]'),
 Text(133.92000000000002, 36.23999999999998, 'entropy = 0.0\nsamples = 37\nvalue = [0, 37, 0]'),
 Text(267.84000000000003, 36.23999999999998, 'entropy = 0.0\nsamples = 31\nvalue = [0, 0, 31]')]
```




```
fn=['sepal length','sepal width','petal length','petal width']
cn=['setosa','versicolor','virginica']
fig,axes=plt.subplots(nrows=1,ncols=2,figsize=(4,4),dpi=300)
tree.plot_tree(model,feature_names=fn,class_names=cn,filled=True)
```

predict=model.predict(x_test)
predict

```
array([0, 0, 0, 2, 1, 1, 2, 2, 1, 2, 0, 2, 1, 2, 0, 1, 0, 0, 0, 1, 2, 0,
       0, 0, 2, 2, 1, 2, 0, 1, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 0, 1,
       2])
```

pd.crosstab(predict,y_test)

Species	0	1	2
row_0			
0	13	0	0
1	0	12	0
2	0	1	19

np.mean(predict==y_test)

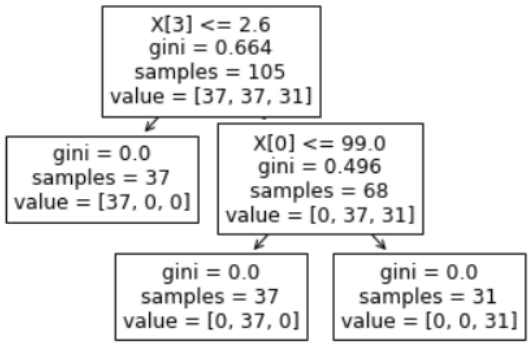
```
0.9777777777777777
```

model = DecisionTreeClassifier(criterion='gini',max_depth=30)
model.fit(x_train,y_train)

```
DecisionTreeClassifier(max_depth=30)
```

tree.plot_tree(model)

```
[Text(133.92000000000002, 181.2, 'X[3] <= 2.6\ngini = 0.664\nsamples = 105\nvalue = [37, 37, 31]'),
 Text(66.96000000000001, 108.72, 'gini = 0.0\nsamples = 37\nvalue = [37, 0, 0]'),
 Text(200.88000000000002, 108.72, 'X[0] <= 99.0\ngini = 0.496\nsamples = 68\nvalue = [0, 37, 31]'),
 Text(133.92000000000002, 36.23999999999998, 'gini = 0.0\nsamples = 37\nvalue = [0, 37, 0]'),
 Text(267.84000000000003, 36.23999999999998, 'gini = 0.0\nsamples = 31\nvalue = [0, 0, 31]')]
```



predict=model.predict(x_test)
predict

```
array([0, 0, 0, 2, 1, 1, 2, 2, 1, 2, 0, 2, 1, 2, 0, 1, 0, 0, 0, 1, 2, 0,
       0, 0, 2, 2, 1, 2, 0, 1, 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 0, 1,
       2])
```

---

pd.crosstab(predict,y_test)

Species	0	1	2
row_0			
0	13	0	0
1	0	12	0
2	0	1	19

np.mean(predict==y_test)

```
0.9777777777777777
```

data.describe

```
<bound method NDFrame.describe of      Unnamed: 0  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width  Species
0             1           5.1          3.5           1.4          0.2        0
1             2           4.9          3.0           1.4          0.2        0
2             3           4.7          3.2           1.3          0.2        0
3             4           4.6          3.1           1.5          0.2        0
4             5           5.0          3.6           1.4          0.2        0
..          ...           ...          ...           ...          ...      ...
145         146           6.7          3.0           5.2          2.3        2
146         147           6.3          2.5           5.0          1.9        2
147         148           6.5          3.0           5.2          2.0        2
148         149           6.2          3.4           5.4          2.3        2
149         150           5.9          3.0           5.1          1.8        2

[150 rows x 6 columns]>
```