

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn import metrics
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

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

df.head()

{"summary":{"\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\n    {\n      \"column\": \"Id\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 43,\n        \"min\": 1,\n        \"max\": 150,\n        \"num_unique_values\": 150,\n        \"samples\": [\n          74,\n          19,\n          119\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"SepalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.828066127977863,\n        \"min\": 4.3,\n        \"max\": 7.9,\n        \"num_unique_values\": 35,\n        \"samples\": [\n          6.2,\n          4.5,\n          5.6\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"SepalWidthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.4335943113621737,\n        \"min\": 2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n        \"samples\": [\n          2.3,\n          4.0,\n          3.5\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"PetalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 1.7644204199522626,\n        \"min\": 1.0,\n        \"max\": 6.9,\n        \"num_unique_values\": 43,\n        \"samples\": [\n          6.7,\n          3.8,\n          3.7\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"PetalWidthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.7631607417008411,\n        \"min\": 0.1,\n        \"max\": 2.5,\n        \"num_unique_values\": 22,\n        \"samples\": [\n          0.2,\n          1.2,\n          1.3\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Species\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 3,\n        \"samples\": [\n          \"Iris-setosa\",\n          \"Iris-versicolor\",\n          \"Iris-virginica\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    ]\n  ],\n  \"type\": \"dataframe\", \"variable_name\": \"df\"}

```

```
df.tail()
```

```
{
  "summary": {
    "name": "df",
    "rows": 5,
    "fields": [
      {
        "column": "Id",
        "properties": {
          "dtype": "number",
          "std": 1,
          "min": 146,
          "max": 150,
          "num_unique_values": 5,
          "samples": [147, 150, 148]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "SepalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 0.30331501776206193,
          "min": 5.9,
          "max": 6.7,
          "num_unique_values": 5,
          "samples": [6.3, 5.9, 6.5]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "SepalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 0.31937438845342625,
          "min": 2.5,
          "max": 3.4,
          "num_unique_values": 3,
          "samples": [3.0, 2.5, 3.4]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "PetalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 0.14832396974191348,
          "min": 5.0,
          "max": 5.4,
          "num_unique_values": 4,
          "samples": [5.0, 5.1, 5.2]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "PetalWidthCm",
        "properties": {
          "dtype": "number",
          "std": 0.23021728866442667,
          "min": 1.8,
          "max": 2.3,
          "num_unique_values": 4,
          "samples": [1.9, 1.8, 2.3]
        },
        "semantic_type": "",
        "description": ""
      },
      {
        "column": "Species",
        "properties": {
          "dtype": "category",
          "num_unique_values": 1,
          "samples": ["Iris-virginica"]
        },
        "semantic_type": "",
        "description": ""
      }
    ]
  },
  "type": "dataframe"
}
```

```
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.describe()
```

```
{ "summary": "{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n    {\n      \"column\": \"Id\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 53.756293020494844,\n        \"min\": 1.0,\n        \"max\": 150.0,\n        \"num_unique_values\": 6,\n        \"samples\": [\n          150.0,\n          75.5,\n          112.75\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"SepalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 51.24711349471842,\n        \"min\": 0.828066127977863,\n        \"max\": 150.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          5.843333333333334,\n          5.8,\n          150.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"SepalWidthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 52.08647211421483,\n        \"min\": 0.4335943113621737,\n        \"max\": 150.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          3.0540000000000003,\n          3.0,\n          150.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"PetalLengthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 51.835227940958106,\n        \"min\": 1.0,\n        \"max\": 150.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          3.7586666666666666,\n          4.35,\n          150.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"PetalWidthCm\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 52.636634243409915,\n        \"min\": 0.1,\n        \"max\": 150.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          1.1986666666666668,\n          1.3,\n          150.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    ]\n  },\n  \"type\": \"dataframe\" }
```

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

```
Id      0
SepalLengthCm  0
SepalWidthCm  0
PetalLengthCm  0
PetalWidthCm  0
Species   0
dtype: int64
```

```
df.isnull().count()
```

```

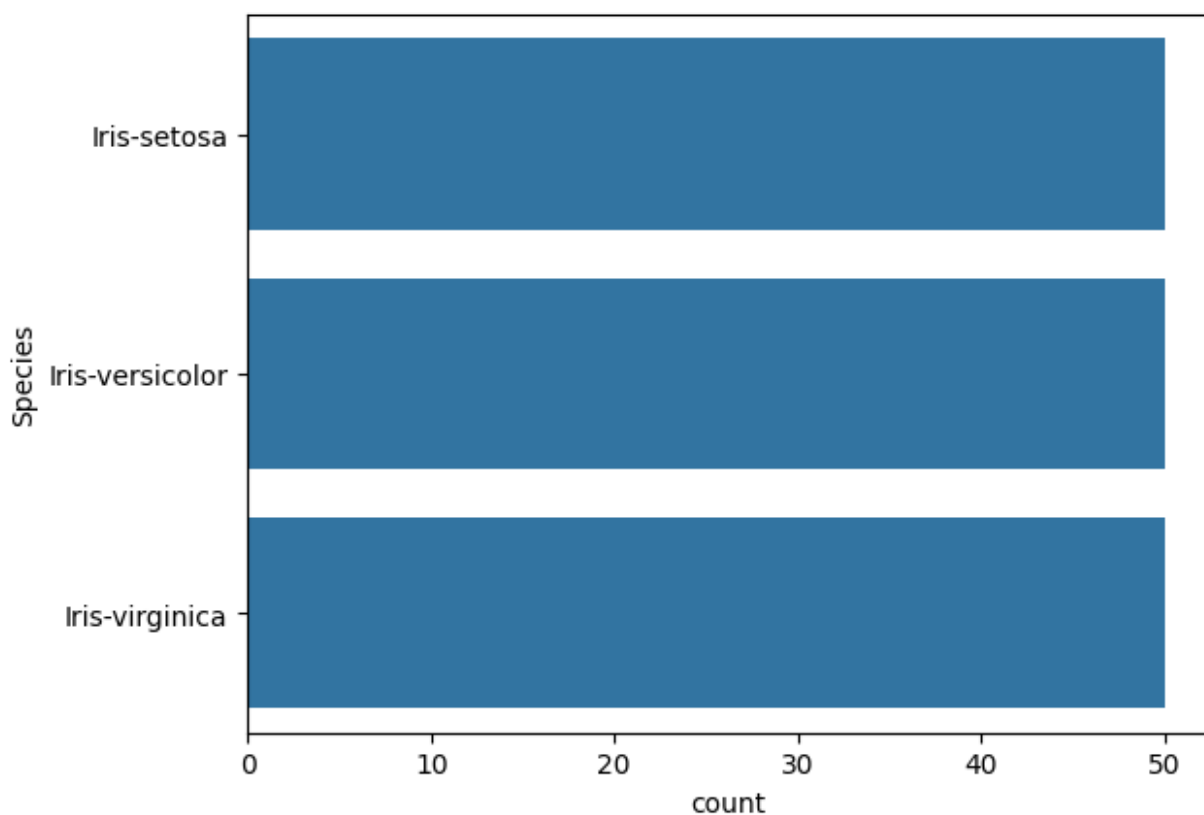
Id          150
SepalLengthCm  150
SepalWidthCm   150
PetalLengthCm  150
PetalWidthCm   150
Species       150
dtype: int64

print(df['Species'].value_counts())
sns.countplot(df['Species'])

Species
Iris-setosa      50
Iris-versicolor  50
Iris-virginica   50
Name: count, dtype: int64

<Axes: xlabel='count', ylabel='Species'>

```



```

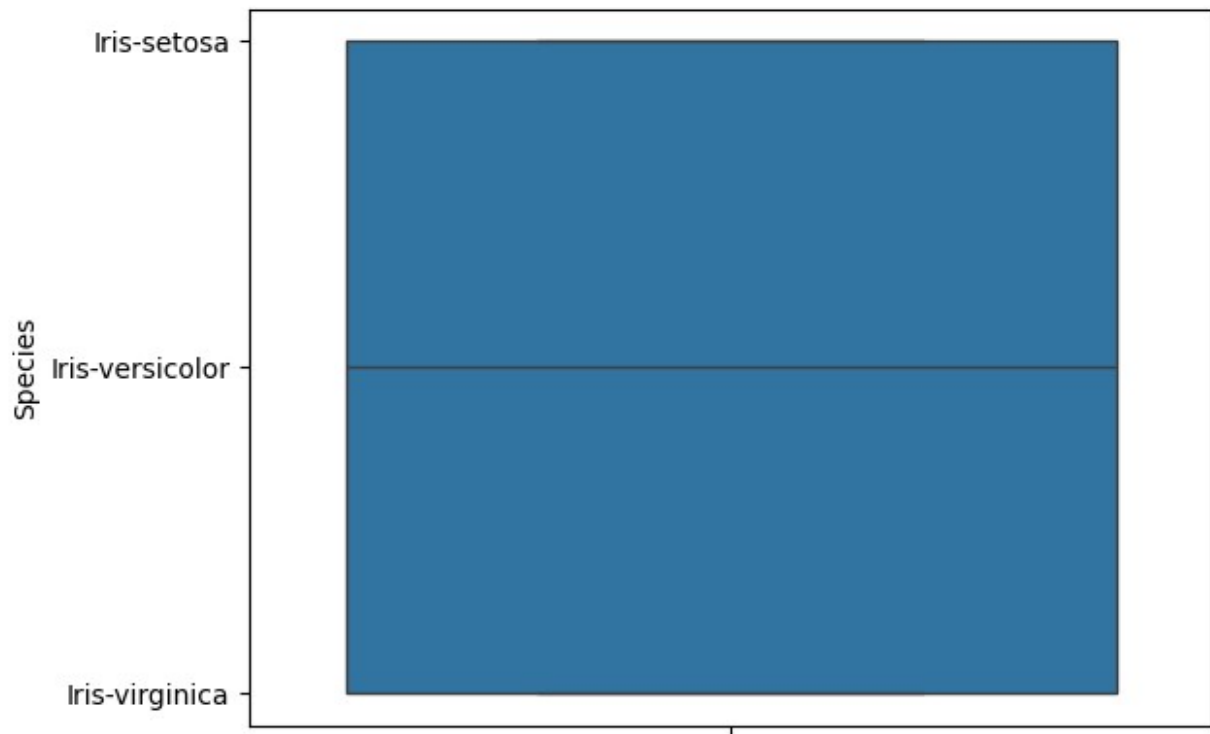
print(df['Species'].value_counts())
sns.boxplot(df['Species'])

Species
Iris-setosa      50

```

```
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

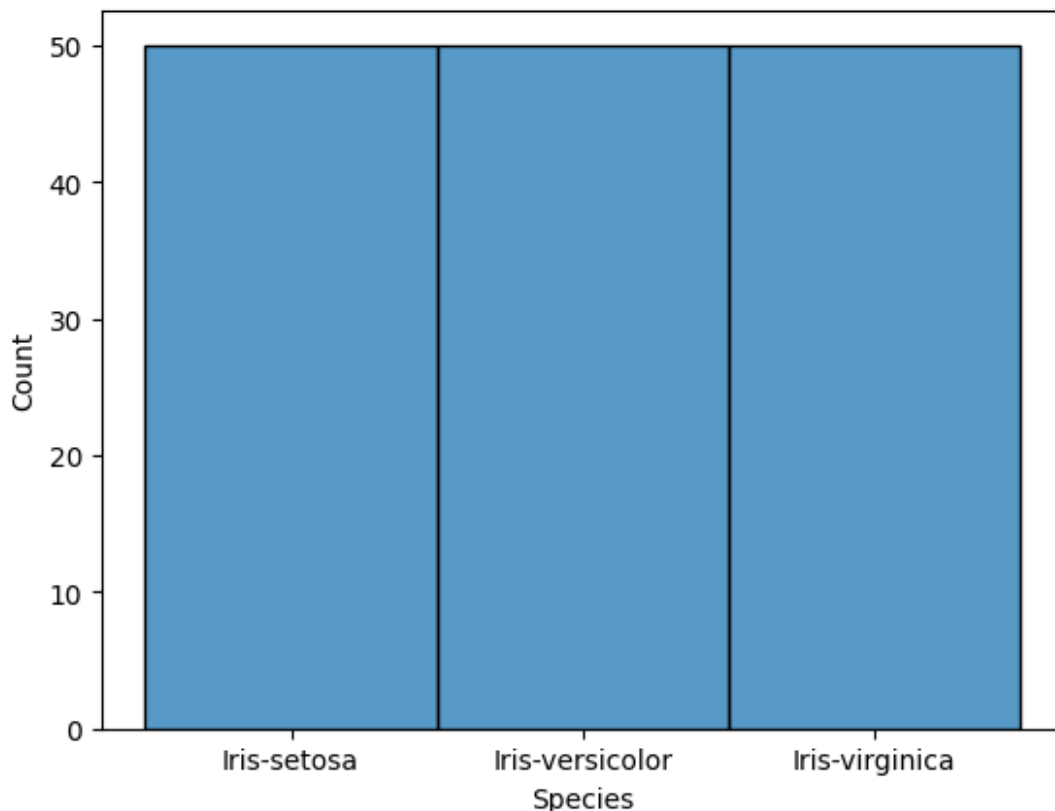
```
<Axes: ylabel='Species'>
```



```
print(df['Species'].value_counts())
sns.histplot(df['Species'])
```

```
Species
Iris-setosa    50
Iris-versicolor 50
Iris-virginica 50
Name: count, dtype: int64
```

```
<Axes: xlabel='Species', ylabel='Count'>
```



```
df.drop("Id", axis=1, inplace = True)
```

```
X = df.iloc[:,0:4]
```

```
y = df['Species']
```

```
X
```

```
{
  "summary": {
    "name": "X",
    "rows": 150,
    "fields": [
      {
        "column": "SepalLengthCm",
        "properties": {
          "dtype": "number",
          "std": 0.828066127977863,
          "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": 1.7644204199522626,
          "min": 1.0,
          "max": 6.9,
          "num_unique_values": 43,
          "samples": [
            6.7,
            3.8,
            3.7
          ],
          "semantic_type": ""
        },
        "description": ""
      }
    ]
  }
}
```

```

{"PetalWidthCm": 0.1, "std": 0.7631607417008411, "min": 0.1, "max": 2.5, "num_unique_values": 22, "samples": [0.2, 1.2, 1.3], "semantic_type": "", "description": ""}
{"type": "dataframe", "variable_name": "X"}

```

y

```

0      Iris-setosa
1      Iris-setosa
2      Iris-setosa
3      Iris-setosa
4      Iris-setosa

```

...

```

145     Iris-virginica
146     Iris-virginica
147     Iris-virginica
148     Iris-virginica
149     Iris-virginica

```

Name: Species, Length: 150, dtype: object

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.33, random_state=42)

```

X_train.shape

```
(100, 4)
```

X_test.shape

```
(50, 4)
```

y_train.shape

```
(100,)
```

y_test.shape

```
(50,)
```

DT Gini Index

```

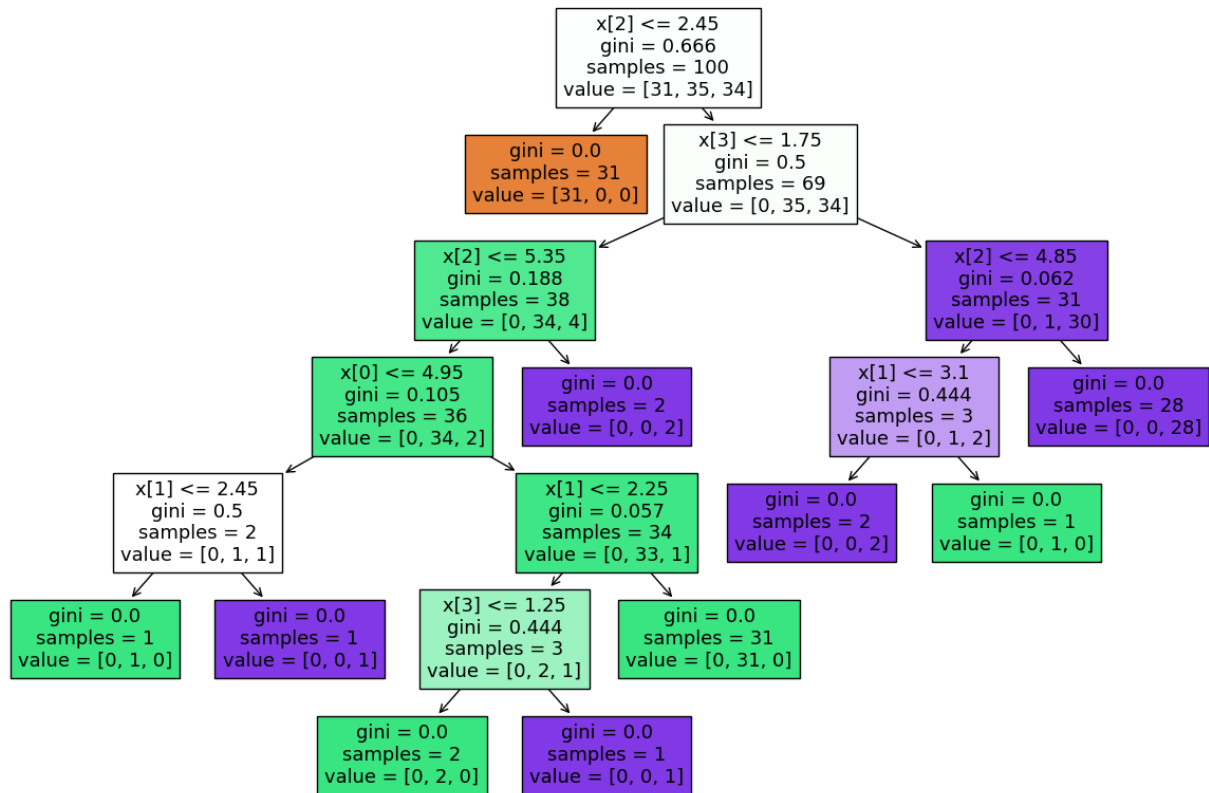
dt = DecisionTreeClassifier()
dt.fit(X_train,y_train)

DecisionTreeClassifier()

plt.figure(figsize=(15,10))
tree.plot_tree(dt,filled=True)

```

```
[Text(0.5416666666666666, 0.9285714285714286, 'x[2] <= 2.45\ngini = 0.666\nsamples = 100\nvalue = [31, 35, 34]'),  
Text(0.4583333333333333, 0.7857142857142857, 'gini = 0.0\nsamples = 31\nvalue = [31, 0, 0]'),  
Text(0.625, 0.7857142857142857, 'x[3] <= 1.75\ngini = 0.5\nsamples = 69\nvalue = [0, 35, 34]'),  
Text(0.4166666666666667, 0.6428571428571429, 'x[2] <= 5.35\ngini = 0.188\nsamples = 38\nvalue = [0, 34, 4]'),  
Text(0.3333333333333333, 0.5, 'x[0] <= 4.95\ngini = 0.105\nsamples = 36\nvalue = [0, 34, 2]'),  
Text(0.1666666666666666, 0.35714285714285715, 'x[1] <= 2.45\ngini = 0.5\nsamples = 2\nvalue = [0, 1, 1]'),  
Text(0.08333333333333333, 0.21428571428571427, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),  
Text(0.25, 0.21428571428571427, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),  
Text(0.5, 0.35714285714285715, 'x[1] <= 2.25\ngini = 0.057\nsamples = 34\nvalue = [0, 33, 1]'),  
Text(0.4166666666666667, 0.21428571428571427, 'x[3] <= 1.25\ngini = 0.444\nsamples = 3\nvalue = [0, 2, 1]'),  
Text(0.3333333333333333, 0.07142857142857142, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),  
Text(0.5, 0.07142857142857142, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),  
Text(0.5833333333333334, 0.21428571428571427, 'gini = 0.0\nsamples = 31\nvalue = [0, 31, 0]'),  
Text(0.5, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),  
Text(0.8333333333333334, 0.6428571428571429, 'x[2] <= 4.85\ngini = 0.062\nsamples = 31\nvalue = [0, 1, 30]'),  
Text(0.75, 0.5, 'x[1] <= 3.1\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),  
Text(0.6666666666666666, 0.35714285714285715, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),  
Text(0.8333333333333334, 0.35714285714285715, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),  
Text(0.9166666666666666, 0.5, 'gini = 0.0\nsamples = 28\nvalue = [0, 0, 28]')]
```

```

y_pred = dt.predict(X_test)
y_pred

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

metrics.accuracy_score(y_pred,y_test)

```

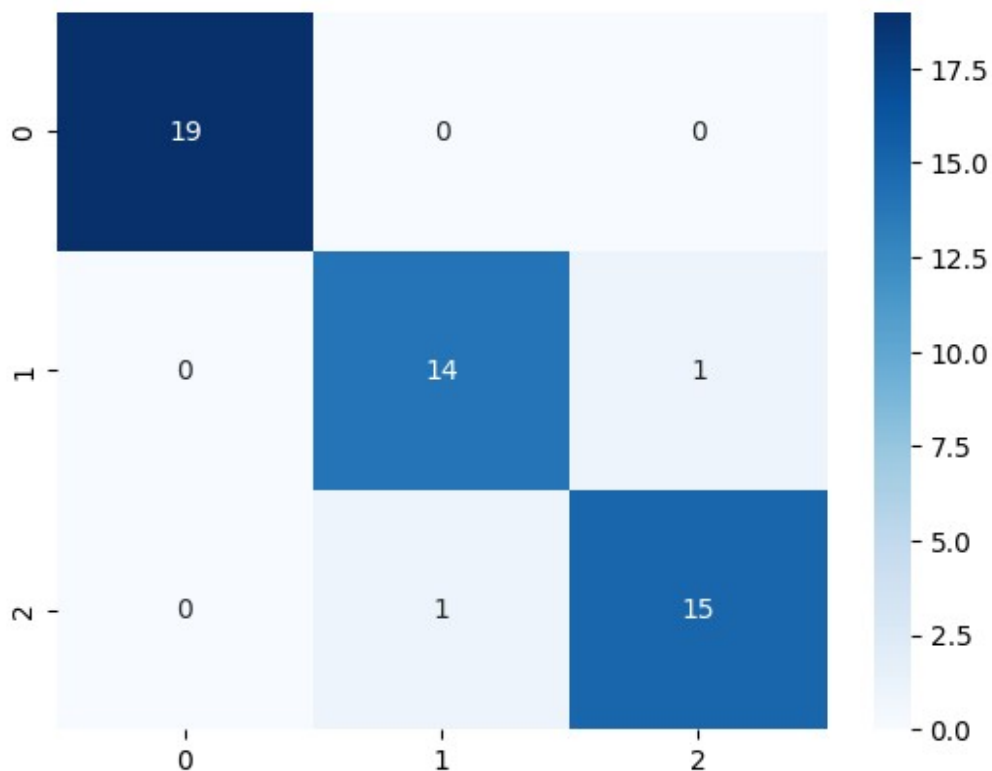
0.96

```
from sklearn.metrics import classification_report  
print(classification_report(y_pred, y_test))
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	19
Iris-versicolor	0.93	0.93	0.93	15
Iris-virginica	0.94	0.94	0.94	16
accuracy			0.96	50
macro avg	0.96	0.96	0.96	50
weighted avg	0.96	0.96	0.96	50

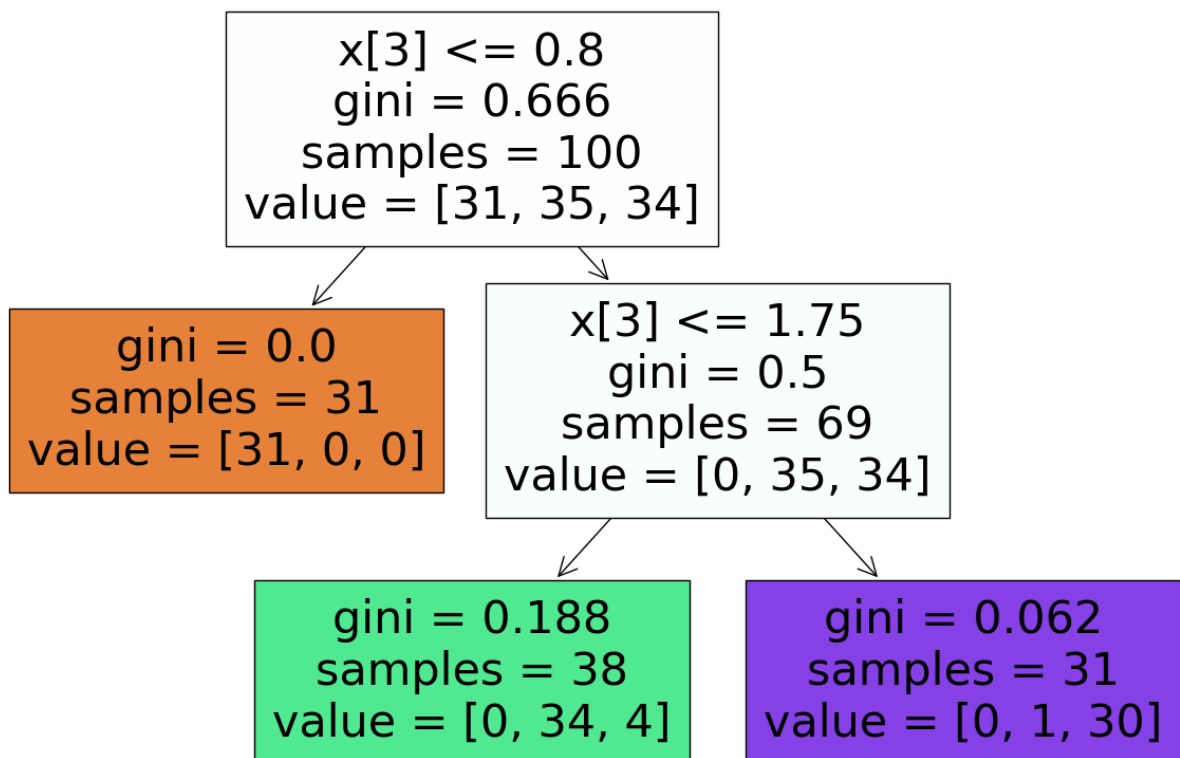
```
from sklearn.metrics import confusion_matrix  
cm = confusion_matrix(y_pred, y_test)  
sns.heatmap(cm, annot=True, cmap='Blues')
```

<Axes: >



```
dt1 = DecisionTreeClassifier(max_depth=2)  
dt1.fit(X_train, y_train)  
plt.figure(figsize=(15, 10))  
tree.plot_tree(dt1, filled=True)
```

```
[Text(0.4, 0.8333333333333334, 'x[3] <= 0.8\ngini = 0.666\nsamples = 100\nvalue = [31, 35, 34]'),
Text(0.2, 0.5, 'gini = 0.0\nsamples = 31\nvalue = [31, 0, 0]'),
Text(0.6, 0.5, 'x[3] <= 1.75\ngini = 0.5\nsamples = 69\nvalue = [0, 35, 34]'),
Text(0.4, 0.16666666666666666, 'gini = 0.188\nsamples = 38\nvalue = [0, 34, 4]'),
Text(0.8, 0.16666666666666666, 'gini = 0.062\nsamples = 31\nvalue = [0, 1, 30]')]
```



```
y_pred1 = dt1.predict(X_test)
y_pred1
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
      'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
      'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
      'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-
setosa',
      'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-
virginica',
      'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
      'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-
virginica',
      'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
```

```

'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-
versicolor',
'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
'Iris-virginica'], dtype=object)

```

```
print(classification_report(y_pred1, y_test))
```

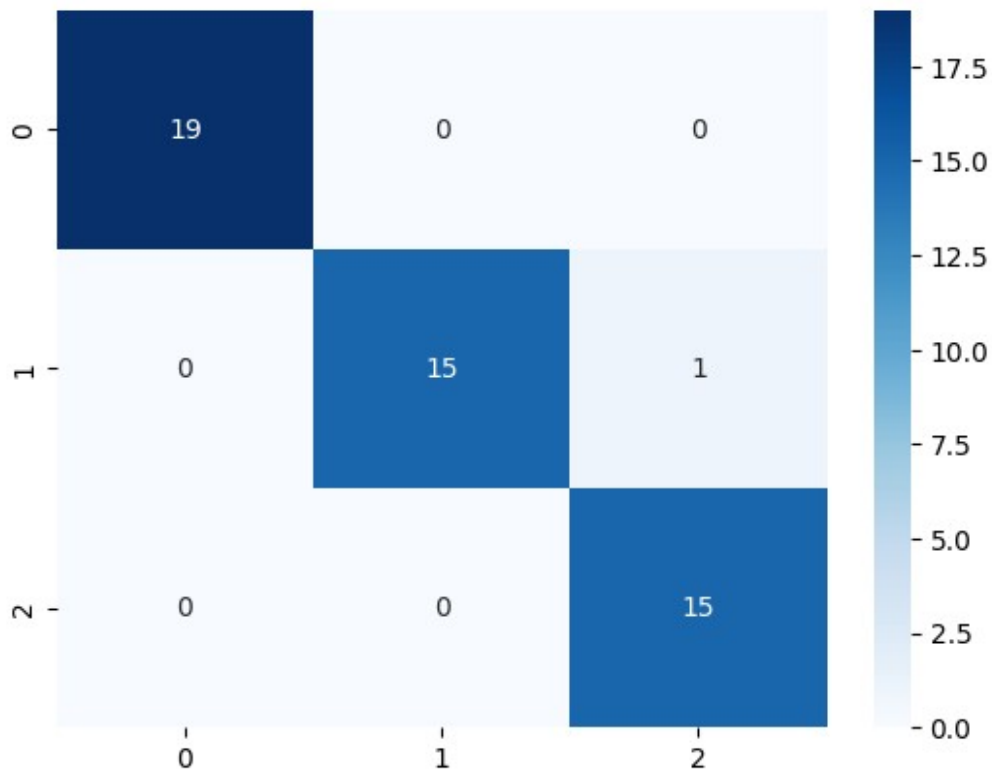
	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	19
Iris-versicolor	1.00	0.94	0.97	16
Iris-virginica	0.94	1.00	0.97	15
accuracy			0.98	50
macro avg	0.98	0.98	0.98	50
weighted avg	0.98	0.98	0.98	50

```

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_pred1,y_test)
sns.heatmap(cm,annot=True,cmap='Blues')

```

<Axes: >



```
parameter = {
    'criterion':['gini','entropy','log_loss'],
    'max_depth':[1,2,3,4,5],
    'max_features':['auto','sqrt','log2'],
    'splitter':['best','random']
}
```

#GridSearchCV and fine-tuning the parameters

```
from sklearn.model_selection import GridSearchCV
treemodel = DecisionTreeClassifier(max_depth=2)
grid =
GridSearchCV(treemodel,param_grid=parameter,cv=5,scoring='accuracy')
grid.fit(X_train,y_train)
```

/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py:425: FitFailedWarning:
150 fits failed out of a total of 450.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:

150 fits failed with the following error:

```

Traceback (most recent call last):
  File
"/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py", line 729, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "/usr/local/lib/python3.10/dist-packages/sklearn/base.py", line 1145, in wrapper
    estimator._validate_params()
  File "/usr/local/lib/python3.10/dist-packages/sklearn/base.py", line 638, in _validate_params
    validate_parameter_constraints(
  File
"/usr/local/lib/python3.10/dist-packages/sklearn/utils/_param_validation.py", line 96, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The
'max_features' parameter of DecisionTreeClassifier must be an int in
the range [1, inf), a float in the range (0.0, 1.0], a str among
{'sqrt', 'log2'} or None. Got 'auto' instead.

    warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:979: UserWarning: One or more of the test scores are non-finite:
[ nan nan 0.65 0.62 0.66 0.57 nan nan 0.84 0.79 0.91 0.76 nan nan
 0.91 0.88 0.92 0.73 nan nan 0.93 0.87 0.89 0.85 nan nan 0.92 0.91
 0.92 0.93 nan nan 0.66 0.53 0.66 0.62 nan nan 0.92 0.67 0.87 0.82
 nan nan 0.85 0.86 0.9 0.78 nan nan 0.87 0.89 0.87 0.87 nan nan
 0.92 0.86 0.92 0.93 nan nan 0.63 0.55 0.63 0.6 nan nan 0.86 0.68
 0.88 0.77 nan nan 0.9 0.88 0.83 0.87 nan nan 0.9 0.91 0.9 0.87
 nan nan 0.94 0.89 0.89 0.88]
    warnings.warn(

GridSearchCV(cv=5, estimator=DecisionTreeClassifier(max_depth=2),
             param_grid={'criterion': ['gini', 'entropy', 'log_loss'],
                          'max_depth': [1, 2, 3, 4, 5],
                          'max_features': ['auto', 'sqrt', 'log2'],
                          'splitter': ['best', 'random']},
             scoring='accuracy')

grid.best_params_

{'criterion': 'log_loss',
 'max_depth': 5,
 'max_features': 'sqrt',
 'splitter': 'best'}

y_pred2 = grid.predict(X_test)

y_pred2

```

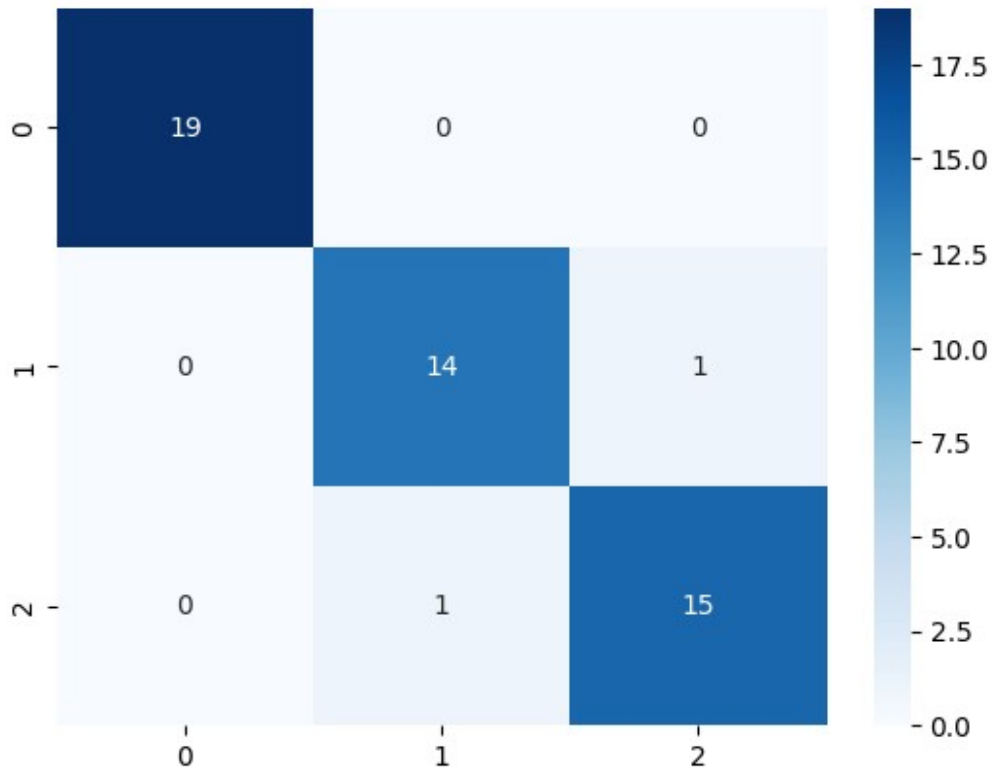
```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
      'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
      'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
      'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-
setosa',
      'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',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

```
print(classification_report(y_test,y_pred2))
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	19
Iris-versicolor	0.93	0.93	0.93	15
Iris-virginica	0.94	0.94	0.94	16
accuracy			0.96	50
macro avg	0.96	0.96	0.96	50
weighted avg	0.96	0.96	0.96	50

```
cm = confusion_matrix(y_test,y_pred2)
sns.heatmap(cm,annot=True,cmap='Blues')
```

```
<Axes: >
```



```
df1= pd.read_csv('/content/titanic_dataset (1).csv')
```

```
df1.head()
```

```
{
  "summary": {
    "name": "df1",
    "rows": 891,
    "fields": [
      {
        "column": "PassengerId",
        "properties": {
          "dtype": "number",
          "std": 257,
          "min": 1,
          "max": 891,
          "num_unique_values": 891,
          "samples": [
            710,
            440,
            841
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Survived",
        "properties": {
          "dtype": "number",
          "std": 0,
          "min": 0,
          "max": 1,
          "num_unique_values": 2,
          "samples": [
            1,
            0
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Pclass",
        "properties": {
          "dtype": "number",
          "std": 0,
          "min": 1,
          "max": 3,
          "num_unique_values": 3,
          "samples": [
            3,
            1
          ],
          "semantic_type": "",
          "description": ""
        },
        "column": "Name",
        "properties": {
          "dtype": "string",
          "num_unique_values": 891,
          "samples": [
            "Moubarek, Master. Halim Gonios (\\\\"William George\\\\")",
            "Kvillner, Mr. Johan Henrik Johannesson"
          ],
          "semantic_type": "",
          "description": ""
        }
      ]
    }
  }
}
```



```

}\n    },\n    {\n        \"column\": \"Sex\", \n        \"properties\": {\n            \"dtype\": \"category\", \n            \"num_unique_values\": 2, \n            \"samples\": [\n                \"female\", \n                \"male\" \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    }, \n    {\n        \"column\": \"Age\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 14.526497332334042, \n            \"min\": 0.42, \n            \"max\": 80.0, \n            \"num_unique_values\": 88, \n            \"samples\": [\n                0.75, \n                22.0 \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    }, \n    {\n        \"column\": \"SibSp\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 1, \n            \"min\": 0, \n            \"max\": 8, \n            \"num_unique_values\": 7, \n            \"samples\": [\n                1, \n                0 \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    }, \n    {\n        \"column\": \"Parch\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 0, \n            \"min\": 0, \n            \"max\": 6, \n            \"num_unique_values\": 7, \n            \"samples\": [\n                0, \n                1 \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    }, \n    {\n        \"column\": \"Ticket\", \n        \"properties\": {\n            \"dtype\": \"string\", \n            \"num_unique_values\": 681, \n            \"samples\": [\n                \"11774\", \n                \"248740\" \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    }, \n    {\n        \"column\": \"Fare\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 49.6934285971809, \n            \"min\": 0.0, \n            \"max\": 512.3292, \n            \"num_unique_values\": 248, \n            \"samples\": [\n                11.2417, \n                51.8625 \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    }, \n    {\n        \"column\": \"Cabin\", \n        \"properties\": {\n            \"dtype\": \"category\", \n            \"num_unique_values\": 147, \n            \"samples\": [\n                \"D45\", \n                \"B49\" \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    }, \n    {\n        \"column\": \"Embarked\", \n        \"properties\": {\n            \"dtype\": \"category\", \n            \"num_unique_values\": 3, \n            \"samples\": [\n                \"S\", \n                \"C\" \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    } \n    ] \n    }, \n    \"type\": \"dataframe\", \n    \"variable_name\": \"df1\" \n}

```

```
df1.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64

```

3	Name	891	non-null	object
4	Sex	891	non-null	object
5	Age	714	non-null	float64
6	SibSp	891	non-null	int64
7	Parch	891	non-null	int64
8	Ticket	891	non-null	object
9	Fare	891	non-null	float64
10	Cabin	204	non-null	object
11	Embarked	882	non-null	object

dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

df1.describe()

```
{
  "summary": {
    "name": "df1",
    "rows": 8,
    "fields": [
      {
        "column": "PassengerId",
        "properties": {
          "dtype": "number",
          "std": 320.8159711429855,
          "min": 1.0,
          "max": 891.0,
          "num_unique_values": 6,
          "samples": [
            891.0,
            446.0,
            668.5
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Survived",
        "properties": {
          "dtype": "number",
          "std": 314.8713661874558,
          "min": 0.0,
          "max": 891.0,
          "num_unique_values": 5,
          "samples": [
            0.3838383838383838,
            1.0,
            0.4865924542648575
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Pclass",
        "properties": {
          "dtype": "number",
          "std": 314.2523437079694,
          "min": 0.836071240977049,
          "max": 891.0,
          "num_unique_values": 6,
          "samples": [
            891.0,
            2.308641975308642,
            3.0
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Age",
        "properties": {
          "dtype": "number",
          "std": 242.9056731818781,
          "min": 0.42,
          "max": 714.0,
          "num_unique_values": 8,
          "samples": [
            29.69911764705882,
            28.0,
            714.0
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "SibSp",
        "properties": {
          "dtype": "number",
          "std": 314.4908277465442,
          "min": 0.0,
          "max": 891.0,
          "num_unique_values": 6,
          "samples": [
            891.0,
            8.0,
            0.5230078563411896
          ],
          "semantic_type": ""
        },
        "description": ""
      },
      {
        "column": "Parch",
        "properties": {
          "dtype": "number",
          "std": 314.65971717879,
          "min": 0.0,
          "max": 891.0,
          "num_unique_values": 5,
          "samples": [
            0.38159371492704824,
            6.0,
            0.8060572211299483
          ],
          "semantic_type": ""
        },
        "description": ""
      }
    ]
  }
}
```

```
{
  "column": "Fare",
  "dtype": "number",
  "min": 0.0,
  "max": 891.0,
  "samples": [
    32.204207968574636,
    14.4542,
    891.0
  ],
  "std": 330.6256632228578,
  "num_unique_values": 8,
  "semantic_type": "Fare",
  "description": "Fare paid by passenger or other person for travel on a ship or aircraft."
},
{
  "column": "Age",
  "dtype": "number",
  "min": 0.0,
  "max": 80.0,
  "samples": [
    32.204207968574636,
    14.4542,
    891.0
  ],
  "std": 330.6256632228578,
  "num_unique_values": 8,
  "semantic_type": "Age",
  "description": "Age of the passenger."
},
{
  "column": "Survived",
  "dtype": "boolean",
  "min": 0.0,
  "max": 1.0,
  "samples": [
    32.204207968574636,
    14.4542,
    891.0
  ],
  "std": 330.6256632228578,
  "num_unique_values": 2,
  "semantic_type": "Survived",
  "description": "Whether the passenger survived."
}
],
"type": "dataframe"
}
```

```
df1.shape
```

 $(891, 12)$

```
df1.isnull().sum()
```

PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	9
dtype:	int64

```
df1['Age'].fillna(df1['Age'].median(), inplace= True)
```

```
df1.isnull().sum()
```

```

PassengerId      0
Survived         0
Pclass           0
Sex              0
Age              0
SibSp            0
Parch            0
Fare             0
Embarked_Q       0
Embarked_S       0
dtype: int64

```

```
df1['Sex'] = df1['Sex'].map({'male':0, 'female':1})
```

```
df1.head()
```

```
{ "summary": "{\n  \"name\": \"df1\",\n  \"rows\": 891,\n  \"fields\": [\n    {\n      \"column\": \"PassengerId\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 257,\n
```

```

\"min\": 1,\n          \"max\": 891,\n          \"num_unique_values\": 891,\n          \"samples\": [\n          710,\n          440,\n          841\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Survived\",\n          \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 0,\n          \"min\": 0,\n          \"max\": 1,\n          \"num_unique_values\": 2,\n          \"samples\": [\n          1,\n          0\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Pclass\",\n          \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 0,\n          \"min\": 1,\n          \"max\": 3,\n          \"num_unique_values\": 3,\n          \"samples\": [\n          3,\n          1\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Name\",\n          \"properties\": {\n          \"dtype\": \"string\",\n          \"num_unique_values\": 891,\n          \"samples\": [\n          \"Moubarek, Master. Halim Gonios (\\\"William George\\\")\",\n          \"Kvillner, Mr. Johan Henrik Johannesson\"\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Sex\",\n          \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 0,\n          \"min\": 0,\n          \"max\": 1,\n          \"num_unique_values\": 2,\n          \"samples\": [\n          1,\n          0\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Age\",\n          \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 13.019696550973201,\n          \"min\": 0.42,\n          \"max\": 80.0,\n          \"num_unique_values\": 88,\n          \"samples\": [\n          0.75,\n          22.0\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"SibSp\",\n          \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 1,\n          \"min\": 0,\n          \"max\": 8,\n          \"num_unique_values\": 7,\n          \"samples\": [\n          1,\n          0\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Parch\",\n          \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 0,\n          \"min\": 0,\n          \"max\": 6,\n          \"num_unique_values\": 7,\n          \"samples\": [\n          0,\n          1\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Ticket\",\n          \"properties\": {\n          \"dtype\": \"string\",\n          \"num_unique_values\": 681,\n          \"samples\": [\n          \"11774\",\n          \"248740\"\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\",\n          },\n          {\n          \"column\": \"Fare\",\n          \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 49.6934285971809,\n          \"min\": 0.0,\n          \"max\": 512.3292,\n          \"num_unique_values\": 248,\n          \"samples\": [\n          11.2417,\n          51.8625\n          ],\n          \"semantic_type\":

```

```

\\",\n      \"description\": \"\"\n    }\n  },\n  {\n    \"column\": \"Cabin\", \n    \"properties\": {\n      \"dtype\": \n    \"category\", \n    \"num_unique_values\": 147, \n    \"samples\": [\n      \"D45\", \n      \"B49\" \n    ], \n    \"semantic_type\": \"\", \n    \"description\": \"\"\n  }, \n  {\n    \"column\": \"Embarked\", \n    \"properties\": \n    {\n      \"dtype\": \"category\", \n      \"num_unique_values\": \n    3, \n    \"samples\": [\n      \"S\", \n      \"C\" \n    ], \n    \"semantic_type\": \"\", \n    \"description\": \"\"\n  }\n  }\n  ], \"type\": \"dataframe\", \"variable_name\": \"df1\"}

```

```
print(df1['Sex'])
```

```

0      0
1      1
2      1
3      1
4      0
..
886    0
887    1
888    1
889    0
890    0

```

```
Name: Sex, Length: 891, dtype: int64
```

```
df1=pd.get_dummies(df1 , columns=['Embarked'],drop_first=True)
print(df1)
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	
..	
886	887	0	2	
887	888	1	1	
888	889	0	3	
889	890	1	1	
890	891	0	3	

	SibSp	\	Name	Sex	Age
0			Braund, Mr. Owen Harris	0	22.0
1					
1			Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0
1					
2			Heikkinen, Miss. Laina	1	26.0
0					

```

3      Futrelle, Mrs. Jacques Heath (Lily May Peel)    1  35.0
1
4      Allen, Mr. William Henry                        0  35.0
0
..      ...      ...      ...
..
886      Montvila, Rev. Juozas                        0  27.0
0
887      Graham, Miss. Margaret Edith                 1  19.0
0
888      Johnston, Miss. Catherine Helen "Carrie"     1  28.0
1
889      Behr, Mr. Karl Howell                        0  26.0
0
890      Dooley, Mr. Patrick                          0  32.0
0

```

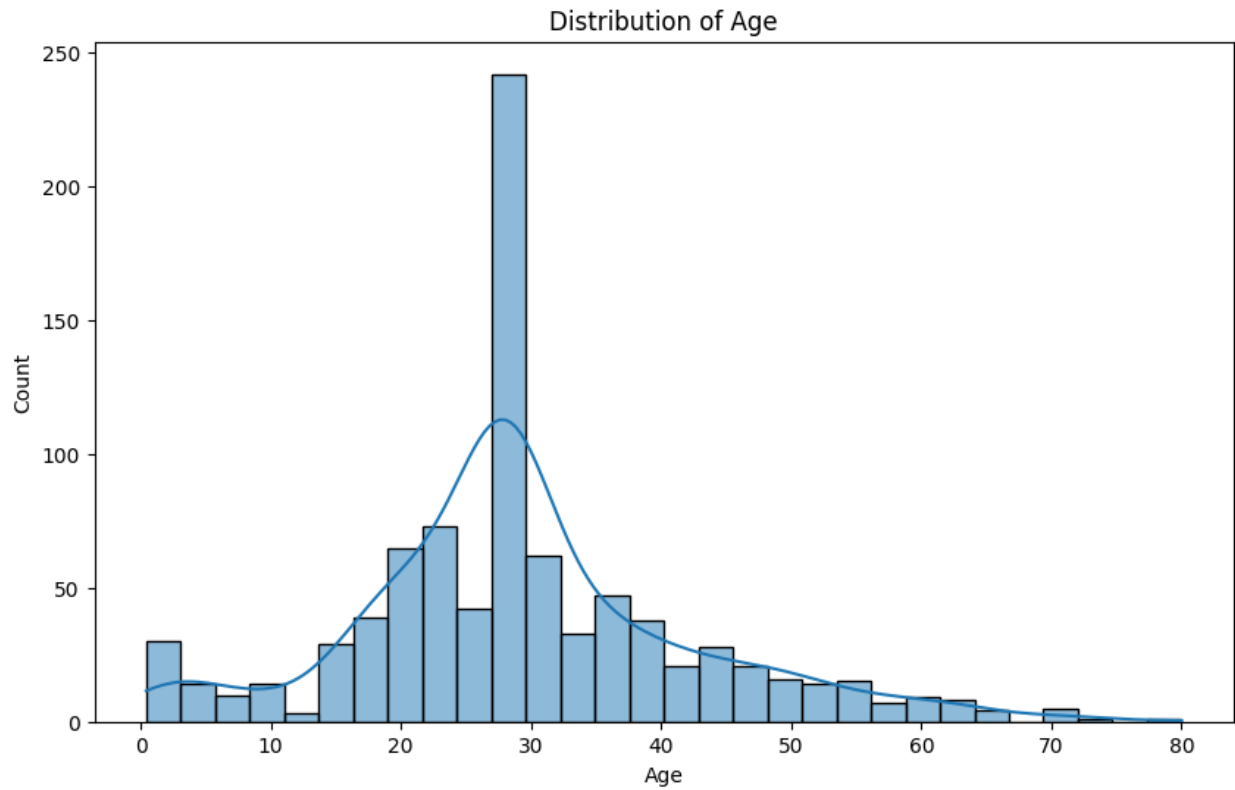
	Parch	Ticket	Fare	Cabin	Embarked_Q	Embarked_S
0	0	A/5 21171	7.2500	NaN	False	True
1	0	PC 17599	71.2833	C85	False	False
2	0	STON/O2. 3101282	7.9250	NaN	False	True
3	0	113803	53.1000	C123	False	True
4	0	373450	8.0500	NaN	False	True
..
886	0	211536	13.0000	NaN	False	True
887	0	112053	30.0000	B42	False	True
888	2	W./C. 6607	23.4500	NaN	False	True
889	0	111369	30.0000	C148	False	False
890	0	370376	7.7500	NaN	True	False

[891 rows x 13 columns]

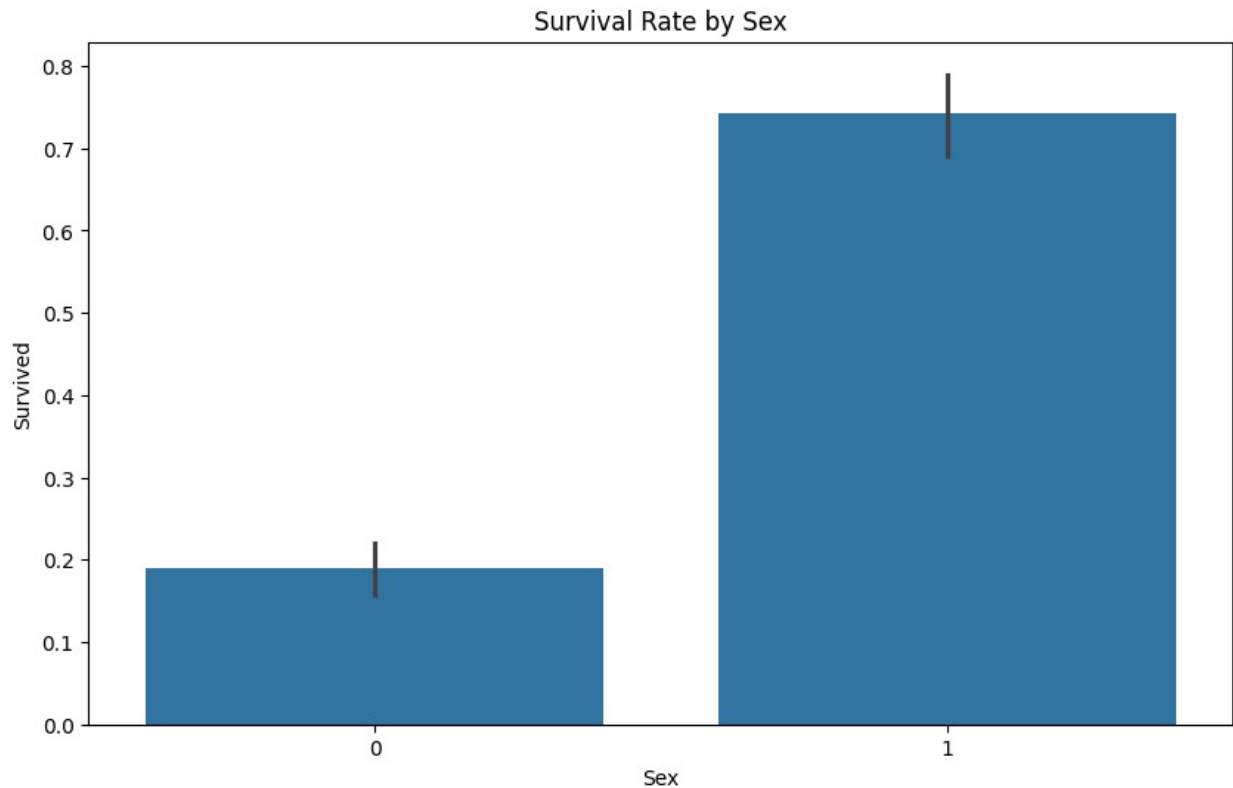
```

#Univariate
plt.figure(figsize=(10,6))
sns.histplot(df1['Age'],bins=30,kde=True)
plt.title('Distribution of Age')
plt.show()

```



```
#Bivariate  
plt.figure(figsize=(10,6))  
sns.barplot(x='Sex',y='Survived',data=df1)  
plt.title('Survival Rate by Sex')  
plt.show()
```



```
df1=df1.drop(['Name','Ticket','Cabin'],axis=1)
```

```
df1.head()
```

```
{
  "summary": {
    "\n  \"name\": \"df1\",
    "\n  \"rows\": 891,
    "\n  \"fields\": [
      {
        "\n    \"column\": \"PassengerId\",
        "\n    \"properties\": {
          "\n      \"dtype\": \"number\",
          "\n      \"std\": 257,
          "\n      \"min\": 1,
          "\n      \"max\": 891,
          "\n      \"num_unique_values\": 891,
          "\n      \"samples\": [
        710,
        440,
        841
      ],
          "\n      \"semantic_type\": \"\",
          "\n      \"description\": \"\"
        },
        "\n    },
        {
          "\n      \"column\": \"Survived\",
          "\n      \"properties\": {
        "\n      \"dtype\": \"number\",
        "\n      \"std\": 0,
        "\n      \"min\": 0,
        "\n      \"max\": 1,
        "\n      \"num_unique_values\": 2,
        "\n      \"samples\": [
        1,
        0
      ],
          "\n      \"semantic_type\": \"\",
          "\n      \"description\": \"\"
        },
        "\n    },
        {
          "\n      \"column\": \"Pclass\",
          "\n      \"properties\": {
        "\n      \"dtype\": \"number\",
        "\n      \"std\": 0,
        "\n      \"min\": 1,
        "\n      \"max\": 3,
        "\n      \"num_unique_values\": 3,
        "\n      \"samples\": [
        3,
        1
      ],
          "\n      \"semantic_type\": \"\",
          "\n      \"description\": \"\"
        },
        "\n    },
        {
          "\n      \"column\": \"Sex\",
          "\n      \"properties\": {
        "\n      \"dtype\": \"number\",
        "\n      \"std\": 0,
        "\n      \"min\": 0,
        "\n      \"max\": 1,
        "\n      \"num_unique_values\": 2,
        "\n      \"samples\": [
        1,
        0
      ],
          "\n      \"semantic_type\": \"\"
        }
      ]
    }
  }
}
```



```

\\",\n      \"description\": \"\\\"\n    },\n    {\n\n\"column\": \"Age\", \n    \"properties\": {\n      \"dtype\":\n\"number\", \n      \"std\": 13.019696550973201, \n      \"min\":\n0.42, \n      \"max\": 80.0, \n      \"num_unique_values\": 88, \n      \"samples\": [\n        0.75, \n        22.0\n      ], \n      \"semantic_type\": \"\\\", \n      \"description\": \"\\\"\n    }, \n    {\n      \"column\": \"SibSp\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 1, \n        \"min\":\n0, \n        \"max\": 8, \n        \"num_unique_values\": 7, \n        \"samples\": [\n          1, \n          0\n        ], \n        \"semantic_type\": \"\\\", \n        \"description\": \"\\\"\n    }, \n    {\n      \"column\": \"Parch\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 0, \n        \"min\":\n0, \n        \"max\": 6, \n        \"num_unique_values\": 7, \n        \"samples\": [\n          0, \n          1\n        ], \n        \"semantic_type\": \"\\\", \n        \"description\": \"\\\"\n    }, \n    {\n      \"column\": \"Fare\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 49.6934285971809, \n        \"min\": 0.0, \n        \"max\": 512.3292, \n        \"num_unique_values\": 248, \n        \"samples\": [\n          11.2417, \n          51.8625\n        ], \n        \"semantic_type\":\n\"\\\", \n        \"description\": \"\\\"\n    }, \n    {\n      \"column\": \"Embarked_Q\", \n      \"properties\": {\n        \"dtype\": \"boolean\", \n        \"num_unique_values\": 2, \n        \"samples\": [\n          true, \n          false\n        ], \n        \"semantic_type\": \"\\\", \n        \"description\": \"\\\"\n    }, \n    {\n      \"column\": \"Embarked_S\", \n      \"properties\": {\n        \"dtype\": \"boolean\", \n        \"num_unique_values\": 2, \n        \"samples\": [\n          true, \n          false\n        ], \n        \"semantic_type\": \"\\\", \n        \"description\": \"\\\"\n    }\n  ], \n  \"description\": \"\\\"\n}\n\nn} \", \"type\": \"dataframe\", \"variable_name\": \"df1\"}

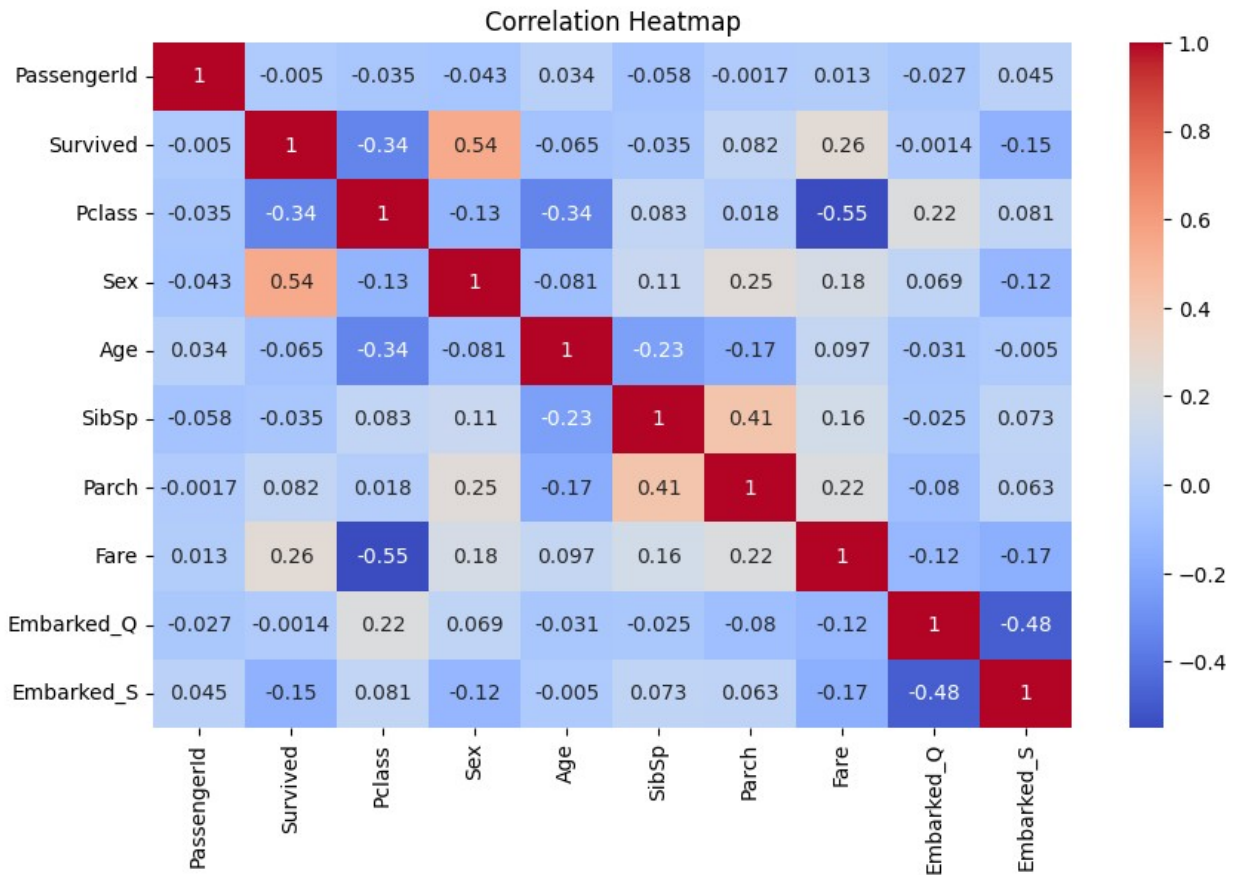
```

#Multi-variate

```

plt.figure(figsize=(10,6))
sns.heatmap(df1.corr(),annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

```



```
df =
df1.drop(['PassengerId', 'SibSp', 'Parch', 'Fare', 'Embarked_Q', 'Embarked_S'], axis=1)
print(df)
```

	Survived	Pclass	Sex	Age
0	0	3	0	22.0
1	1	1	1	38.0
2	1	3	1	26.0
3	1	1	1	35.0
4	0	3	0	35.0
...
886	0	2	0	27.0
887	1	1	1	19.0
888	0	3	1	28.0
889	1	1	0	26.0
890	0	3	0	32.0

[891 rows x 4 columns]

```
from sklearn.model_selection import train_test_split
X = df.drop('Survived', axis=1)
y = df['Survived']
```

```
print(X)
```

	Pclass	Sex	Age
0	3	0	22.0
1	1	1	38.0
2	3	1	26.0
3	1	1	35.0
4	3	0	35.0
...
886	2	0	27.0
887	1	1	19.0
888	3	1	28.0
889	1	0	26.0
890	3	0	32.0

```
[891 rows x 3 columns]
```

```
X_train,X_test,y_train,y_test =  
train_test_split(X,y,test_size=0.2,random_state=42)
```

```
X_train.shape
```

```
(712, 3)
```

```
X_test.shape
```

```
(179, 3)
```

```
y_train.shape
```

```
(712,)
```

```
y_test.shape
```

```
(179,)
```

```
from sklearn.tree import DecisionTreeClassifier  
dt=DecisionTreeClassifier()  
dt.fit(X_train,y_train)
```

```
DecisionTreeClassifier()
```

```
from sklearn import tree  
plt.figure(figsize=(10,6))  
tree.plot_tree(dt,filled=True)
```

```
[Text(0.48298104956268223, 0.9705882352941176, 'x[1] <= 0.5\ngini =  
0.469\nsamples = 712\nvalue = [444, 268]'),  
Text(0.155466472303207, 0.9117647058823529, 'x[2] <= 6.5\ngini =  
0.303\nsamples = 467\nvalue = [380, 87]'),  
Text(0.027988338192419825, 0.8529411764705882, 'x[0] <= 2.5\ngini =  
0.434\nsamples = 22\nvalue = [7, 15]'),
```

```
Text(0.018658892128279883, 0.7941176470588235, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
Text(0.037317784256559766, 0.7941176470588235, 'x[2] <= 2.5\ngini = 0.497\nsamples = 13\nvalue = [7, 6]'),
Text(0.018658892128279883, 0.7352941176470589, 'x[2] <= 0.71\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),
Text(0.009329446064139942, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.027988338192419825, 0.6764705882352942, 'x[2] <= 1.5\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.018658892128279883, 0.6176470588235294, 'gini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.037317784256559766, 0.6176470588235294, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.05597667638483965, 0.7352941176470589, 'x[2] <= 3.5\ngini = 0.444\nsamples = 6\nvalue = [2, 4]'),
Text(0.04664723032069971, 0.6764705882352942, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.0653061224489796, 0.6764705882352942, 'x[2] <= 5.0\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.05597667638483965, 0.6176470588235294, 'gini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.07463556851311953, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.28294460641399416, 0.8529411764705882, 'x[0] <= 1.5\ngini = 0.271\nsamples = 445\nvalue = [373, 72]'),
Text(0.09329446064139942, 0.7941176470588235, 'x[2] <= 17.5\ngini = 0.442\nsamples = 91\nvalue = [61, 30]'),
Text(0.08396501457725948, 0.7352941176470589, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.10262390670553936, 0.7352941176470589, 'x[2] <= 24.5\ngini = 0.437\nsamples = 90\nvalue = [61, 29]'),
Text(0.09329446064139942, 0.6764705882352942, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
Text(0.1119533527696793, 0.6764705882352942, 'x[2] <= 27.5\ngini = 0.452\nsamples = 84\nvalue = [55, 29]'),
Text(0.09329446064139942, 0.6176470588235294, 'x[2] <= 26.0\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),
Text(0.08396501457725948, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.10262390670553936, 0.5588235294117647, 'gini = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.1306122448979592, 0.6176470588235294, 'x[2] <= 75.0\ngini = 0.426\nsamples = 78\nvalue = [54, 24]'),
Text(0.12128279883381925, 0.5588235294117647, 'x[2] <= 60.5\ngini = 0.419\nsamples = 77\nvalue = [54, 23]'),
Text(0.1119533527696793, 0.5, 'x[2] <= 30.5\ngini = 0.438\nsamples = 71\nvalue = [48, 23]'),
Text(0.0717201166180758, 0.4411764705882353, 'x[2] <= 28.5\ngini =
```

```
0.34\nsamples = 23\nvalue = [18, 5]'),
Text(0.06239067055393586, 0.38235294117647056, 'gini = 0.375\nsamples =
= 20\nvalue = [15, 5]'),
Text(0.08104956268221575, 0.38235294117647056, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
Text(0.1521865889212828, 0.4411764705882353, 'x[2] <= 38.5\ngini =
0.469\nsamples = 48\nvalue = [30, 18]'),
Text(0.09970845481049563, 0.38235294117647056, 'x[2] <= 31.5\ngini =
0.498\nsamples = 15\nvalue = [7, 8]'),
Text(0.09037900874635568, 0.3235294117647059, 'gini = 0.444\nsamples
= 3\nvalue = [2, 1]'),
Text(0.10903790087463557, 0.3235294117647059, 'x[2] <= 32.5\ngini =
0.486\nsamples = 12\nvalue = [5, 7]'),
Text(0.09970845481049563, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.11836734693877551, 0.2647058823529412, 'x[2] <= 34.0\ngini =
0.496\nsamples = 11\nvalue = [5, 6]'),
Text(0.10903790087463557, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.12769679300291545, 0.20588235294117646, 'x[2] <= 35.5\ngini =
0.48\nsamples = 10\nvalue = [4, 6]'),
Text(0.11836734693877551, 0.14705882352941177, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.13702623906705538, 0.14705882352941177, 'x[2] <= 36.5\ngini =
0.5\nsamples = 8\nvalue = [4, 4]'),
Text(0.12769679300291545, 0.08823529411764706, 'gini = 0.5\nsamples =
4\nvalue = [2, 2]'),
Text(0.14635568513119535, 0.08823529411764706, 'x[2] <= 37.5\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.13702623906705538, 0.029411764705882353, 'gini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
Text(0.15568513119533528, 0.029411764705882353, 'gini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
Text(0.20466472303206998, 0.38235294117647056, 'x[2] <= 47.5\ngini =
0.422\nsamples = 33\nvalue = [23, 10]'),
Text(0.16501457725947521, 0.3235294117647059, 'x[2] <= 45.25\ngini =
0.245\nsamples = 14\nvalue = [12, 2]'),
Text(0.15568513119533528, 0.2647058823529412, 'x[2] <= 41.0\ngini =
0.346\nsamples = 9\nvalue = [7, 2]'),
Text(0.14635568513119535, 0.20588235294117646, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.16501457725947521, 0.20588235294117646, 'x[2] <= 43.0\ngini =
0.408\nsamples = 7\nvalue = [5, 2]'),
Text(0.15568513119533528, 0.14705882352941177, 'gini = 0.5\nsamples =
2\nvalue = [1, 1]'),
Text(0.17434402332361515, 0.14705882352941177, 'x[2] <= 44.5\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.16501457725947521, 0.08823529411764706, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
```

```
Text(0.1836734693877551, 0.08823529411764706, 'gini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.17434402332361515, 0.2647058823529412, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.24431486880466471, 0.3235294117647059, 'x[2] <= 48.5\ngini = 0.488\nsamples = 19\nvalue = [11, 8]'),
Text(0.23498542274052478, 0.2647058823529412, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.2536443148688047, 0.2647058823529412, 'x[2] <= 53.0\ngini = 0.457\nsamples = 17\nvalue = [11, 6]'),
Text(0.2303206997084548, 0.20588235294117646, 'x[2] <= 50.5\ngini = 0.494\nsamples = 9\nvalue = [5, 4]'),
Text(0.21166180758017492, 0.14705882352941177, 'x[2] <= 49.5\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),
Text(0.20233236151603498, 0.08823529411764706, 'gini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.22099125364431488, 0.08823529411764706, 'gini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.24897959183673468, 0.14705882352941177, 'x[2] <= 51.5\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.23965014577259475, 0.08823529411764706, 'gini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.2583090379008746, 0.08823529411764706, 'gini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.27696793002915454, 0.20588235294117646, 'x[2] <= 55.5\ngini = 0.375\nsamples = 8\nvalue = [6, 2]'),
Text(0.26763848396501455, 0.14705882352941177, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.2862973760932945, 0.14705882352941177, 'x[2] <= 57.0\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),
Text(0.27696793002915454, 0.08823529411764706, 'gini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.2956268221574344, 0.08823529411764706, 'x[2] <= 59.0\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.2862973760932945, 0.029411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.30495626822157434, 0.029411764705882353, 'gini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.1306122448979592, 0.5, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
Text(0.13994169096209913, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.4725947521865889, 0.7941176470588235, 'x[2] <= 32.25\ngini = 0.209\nsamples = 354\nvalue = [312, 42]'),
Text(0.3183673469387755, 0.7352941176470589, 'x[2] <= 31.5\ngini = 0.241\nsamples = 264\nvalue = [227, 37]'),
Text(0.278134110787172, 0.6764705882352942, 'x[2] <= 13.0\ngini = 0.217\nsamples = 251\nvalue = [220, 31]'),
Text(0.22565597667638485, 0.6176470588235294, 'x[0] <= 2.5\ngini =
```

```
0.42\nsamples = 10\nvalue = [7, 3]'),
Text(0.2163265306122449, 0.5588235294117647, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.23498542274052478, 0.5588235294117647, 'x[2] <= 11.5\ngini =
0.346\nsamples = 9\nvalue = [7, 2]'),
Text(0.22565597667638485, 0.5, 'x[2] <= 8.5\ngini = 0.219\nsamples =
8\nvalue = [7, 1]'),
Text(0.2163265306122449, 0.4411764705882353, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
Text(0.23498542274052478, 0.4411764705882353, 'x[2] <= 10.0\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.22565597667638485, 0.38235294117647056, 'gini = 0.444\nsamples
= 3\nvalue = [2, 1]'),
Text(0.24431486880466471, 0.38235294117647056, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.24431486880466471, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
Text(0.3306122448979592, 0.6176470588235294, 'x[2] <= 18.5\ngini =
0.205\nsamples = 241\nvalue = [213, 28]'),
Text(0.27230320699708455, 0.5588235294117647, 'x[2] <= 16.5\ngini =
0.077\nsamples = 25\nvalue = [24, 1]'),
Text(0.2629737609329446, 0.5, 'x[2] <= 15.5\ngini = 0.165\nsamples =
11\nvalue = [10, 1]'),
Text(0.2536443148688047, 0.4411764705882353, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
Text(0.27230320699708455, 0.4411764705882353, 'x[0] <= 2.5\ngini =
0.219\nsamples = 8\nvalue = [7, 1]'),
Text(0.2629737609329446, 0.38235294117647056, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.2816326530612245, 0.38235294117647056, 'gini = 0.278\nsamples
= 6\nvalue = [5, 1]'),
Text(0.2816326530612245, 0.5, 'gini = 0.0\nsamples = 14\nvalue = [14,
0]'),
Text(0.3889212827988338, 0.5588235294117647, 'x[2] <= 27.5\ngini =
0.219\nsamples = 216\nvalue = [189, 27]'),
Text(0.3422740524781341, 0.5, 'x[2] <= 26.5\ngini = 0.268\nsamples =
94\nvalue = [79, 15]'),
Text(0.3189504373177843, 0.4411764705882353, 'x[2] <= 20.5\ngini =
0.233\nsamples = 89\nvalue = [77, 12]'),
Text(0.30029154518950435, 0.38235294117647056, 'x[2] <= 19.5\ngini =
0.33\nsamples = 24\nvalue = [19, 5]'),
Text(0.2909620991253644, 0.3235294117647059, 'x[0] <= 2.5\ngini =
0.245\nsamples = 14\nvalue = [12, 2]'),
Text(0.2816326530612245, 0.2647058823529412, 'gini = 0.375\nsamples =
4\nvalue = [3, 1]'),
Text(0.30029154518950435, 0.2647058823529412, 'gini = 0.18\nsamples =
10\nvalue = [9, 1]'),
Text(0.30962099125364434, 0.3235294117647059, 'gini = 0.42\nsamples =
10\nvalue = [7, 3]'),
```

```
Text(0.33760932944606414, 0.38235294117647056, 'x[0] <= 2.5\ngini = 0.192\nsamples = 65\nvalue = [58, 7]'),
Text(0.3282798833819242, 0.3235294117647059, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
Text(0.3469387755102041, 0.3235294117647059, 'x[2] <= 23.5\ngini = 0.237\nsamples = 51\nvalue = [44, 7]'),
Text(0.3236151603498542, 0.2647058823529412, 'x[2] <= 22.5\ngini = 0.153\nsamples = 24\nvalue = [22, 2]'),
Text(0.3142857142857143, 0.20588235294117646, 'x[2] <= 21.5\ngini = 0.159\nsamples = 23\nvalue = [21, 2]'),
Text(0.30495626822157434, 0.14705882352941177, 'gini = 0.18\nsamples = 10\nvalue = [9, 1]'),
Text(0.3236151603498542, 0.14705882352941177, 'gini = 0.142\nsamples = 13\nvalue = [12, 1]'),
Text(0.33294460641399415, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.37026239067055394, 0.2647058823529412, 'x[2] <= 24.75\ngini = 0.302\nsamples = 27\nvalue = [22, 5]'),
Text(0.3516034985422741, 0.20588235294117646, 'x[2] <= 24.25\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
Text(0.3422740524781341, 0.14705882352941177, 'gini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.360932944606414, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.3889212827988338, 0.20588235294117646, 'x[2] <= 25.5\ngini = 0.32\nsamples = 20\nvalue = [16, 4]'),
Text(0.3795918367346939, 0.14705882352941177, 'gini = 0.346\nsamples = 9\nvalue = [7, 2]'),
Text(0.39825072886297375, 0.14705882352941177, 'gini = 0.298\nsamples = 11\nvalue = [9, 2]'),
Text(0.36559766763848395, 0.4411764705882353, 'x[0] <= 2.5\ngini = 0.48\nsamples = 5\nvalue = [2, 3]'),
Text(0.356268221574344, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.3749271137026239, 0.38235294117647056, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.43556851311953354, 0.5, 'x[2] <= 28.75\ngini = 0.177\nsamples = 122\nvalue = [110, 12]'),
Text(0.40291545189504374, 0.4411764705882353, 'x[0] <= 2.5\ngini = 0.149\nsamples = 99\nvalue = [91, 8]'),
Text(0.3935860058309038, 0.38235294117647056, 'gini = 0.298\nsamples = 11\nvalue = [9, 2]'),
Text(0.4122448979591837, 0.38235294117647056, 'x[2] <= 28.25\ngini = 0.127\nsamples = 88\nvalue = [82, 6]'),
Text(0.40291545189504374, 0.3235294117647059, 'gini = 0.13\nsamples = 86\nvalue = [80, 6]'),
Text(0.4215743440233236, 0.3235294117647059, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.46822157434402334, 0.4411764705882353, 'x[2] <= 29.5\ngini =
```



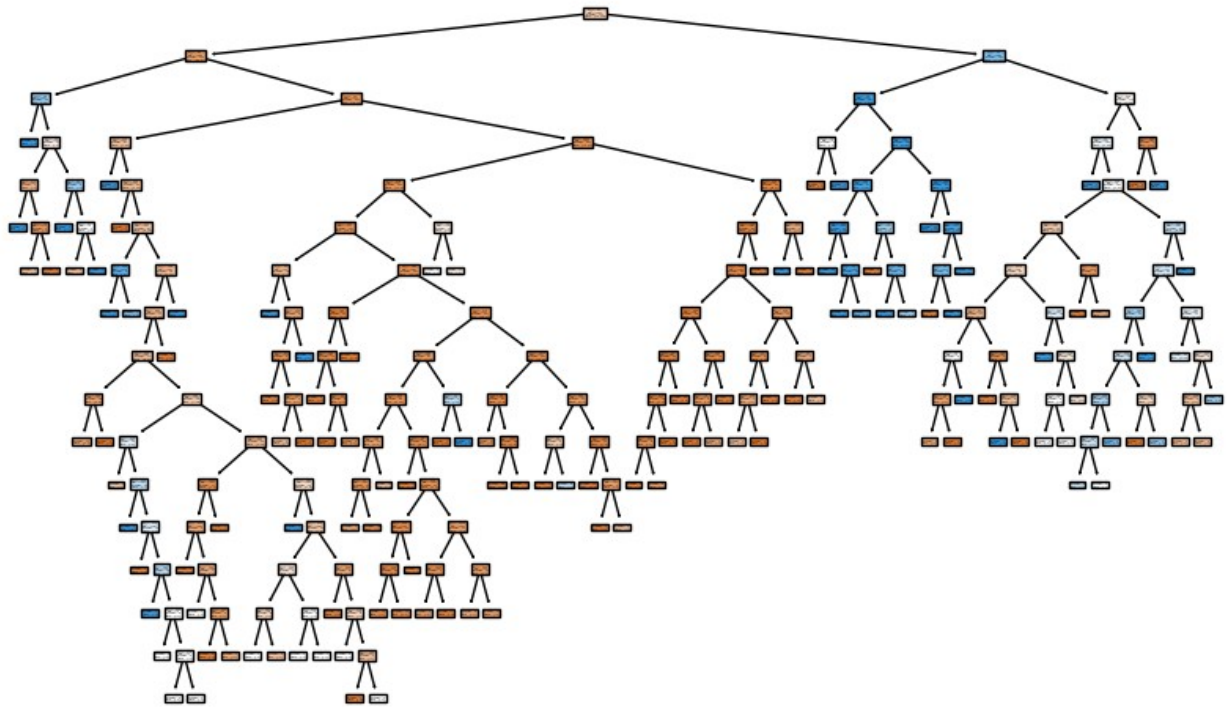
```
0.287\nsamples = 23\nvalue = [19, 4]'),
Text(0.4495626822157434, 0.38235294117647056, 'x[0] <= 2.5\ngini =
0.469\nsamples = 8\nvalue = [5, 3]'),
Text(0.4402332361516035, 0.3235294117647059, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
Text(0.4588921282798834, 0.3235294117647059, 'gini = 0.48\nsamples =
5\nvalue = [2, 3]'),
Text(0.4868804664723032, 0.38235294117647056, 'x[2] <= 30.5\ngini =
0.124\nsamples = 15\nvalue = [14, 1]'),
Text(0.4775510204081633, 0.3235294117647059, 'gini = 0.0\nsamples =
10\nvalue = [10, 0]'),
Text(0.49620991253644314, 0.3235294117647059, 'x[0] <= 2.5\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.4868804664723032, 0.2647058823529412, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.5055393586005831, 0.2647058823529412, 'gini = 0.444\nsamples =
3\nvalue = [2, 1]'),
Text(0.358600583090379, 0.6764705882352942, 'x[0] <= 2.5\ngini =
0.497\nsamples = 13\nvalue = [7, 6]'),
Text(0.3492711370262391, 0.6176470588235294, 'gini = 0.5\nsamples =
2\nvalue = [1, 1]'),
Text(0.36793002915451894, 0.6176470588235294, 'gini = 0.496\nsamples
= 11\nvalue = [6, 5]'),
Text(0.6268221574344023, 0.7352941176470589, 'x[2] <= 61.5\ngini =
0.105\nsamples = 90\nvalue = [85, 5]'),
Text(0.6081632653061224, 0.6764705882352942, 'x[2] <= 44.5\ngini =
0.089\nsamples = 86\nvalue = [82, 4]'),
Text(0.5988338192419825, 0.6176470588235294, 'x[2] <= 41.5\ngini =
0.124\nsamples = 60\nvalue = [56, 4]'),
Text(0.5615160349854227, 0.5588235294117647, 'x[2] <= 38.5\ngini =
0.08\nsamples = 48\nvalue = [46, 2]'),
Text(0.5428571428571428, 0.5, 'x[0] <= 2.5\ngini = 0.056\nsamples =
35\nvalue = [34, 1]'),
Text(0.5335276967930029, 0.4411764705882353, 'x[2] <= 34.5\ngini =
0.142\nsamples = 13\nvalue = [12, 1]'),
Text(0.524198250728863, 0.38235294117647056, 'x[2] <= 33.5\ngini =
0.219\nsamples = 8\nvalue = [7, 1]'),
Text(0.5148688046647231, 0.3235294117647059, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.5335276967930029, 0.3235294117647059, 'gini = 0.278\nsamples =
6\nvalue = [5, 1]'),
Text(0.5428571428571428, 0.38235294117647056, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
Text(0.5521865889212828, 0.4411764705882353, 'gini = 0.0\nsamples =
22\nvalue = [22, 0]'),
Text(0.5801749271137027, 0.5, 'x[2] <= 39.5\ngini = 0.142\nsamples =
13\nvalue = [12, 1]'),
Text(0.5708454810495627, 0.4411764705882353, 'x[0] <= 2.5\ngini =
0.278\nsamples = 6\nvalue = [5, 1]'),
```

```
Text(0.5615160349854227, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.5801749271137027, 0.38235294117647056, 'gini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.5895043731778425, 0.4411764705882353, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
Text(0.6361516034985423, 0.5588235294117647, 'x[2] <= 43.5\ngini = 0.278\nsamples = 12\nvalue = [10, 2]'),
Text(0.6174927113702624, 0.5, 'x[0] <= 2.5\ngini = 0.219\nsamples = 8\nvalue = [7, 1]'),
Text(0.6081632653061224, 0.4411764705882353, 'x[2] <= 42.5\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.5988338192419825, 0.38235294117647056, 'gini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.6174927113702624, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6268221574344023, 0.4411764705882353, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.6548104956268221, 0.5, 'x[0] <= 2.5\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.6454810495626823, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6641399416909621, 0.4411764705882353, 'gini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.6174927113702624, 0.6176470588235294, 'gini = 0.0\nsamples = 26\nvalue = [26, 0]'),
Text(0.6454810495626823, 0.6764705882352942, 'x[2] <= 66.0\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.6361516034985423, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6548104956268221, 0.6176470588235294, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.8104956268221575, 0.9117647058823529, 'x[0] <= 2.5\ngini = 0.386\nsamples = 245\nvalue = [64, 181]'),
Text(0.7037900874635569, 0.8529411764705882, 'x[2] <= 2.5\ngini = 0.074\nsamples = 130\nvalue = [5, 125]'),
Text(0.673469387755102, 0.7941176470588235, 'x[0] <= 1.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.6641399416909621, 0.7352941176470589, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.682798833819242, 0.7352941176470589, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7341107871720116, 0.7941176470588235, 'x[2] <= 27.5\ngini = 0.061\nsamples = 128\nvalue = [4, 124]'),
Text(0.7014577259475219, 0.7352941176470589, 'x[2] <= 24.5\ngini = 0.12\nsamples = 47\nvalue = [3, 44]'),
Text(0.682798833819242, 0.6764705882352942, 'x[2] <= 23.5\ngini = 0.048\nsamples = 41\nvalue = [1, 40]'),
Text(0.673469387755102, 0.6176470588235294, 'gini = 0.0\nsamples =
```

```
29\nvalue = [0, 29]'),
Text(0.6921282798833819, 0.6176470588235294, 'x[0] <= 1.5\ngini =
0.153\nsamples = 12\nvalue = [1, 11]'),
Text(0.682798833819242, 0.5588235294117647, 'gini = 0.0\nsamples = 5\
nvalue = [0, 5]'),
Text(0.7014577259475219, 0.5588235294117647, 'gini = 0.245\nsamples =
7\nvalue = [1, 6]'),
Text(0.7201166180758017, 0.6764705882352942, 'x[0] <= 1.5\ngini =
0.444\nsamples = 6\nvalue = [2, 4]'),
Text(0.7107871720116619, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.7294460641399417, 0.6176470588235294, 'x[2] <= 26.0\ngini =
0.32\nsamples = 5\nvalue = [1, 4]'),
Text(0.7201166180758017, 0.5588235294117647, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.7387755102040816, 0.5588235294117647, 'gini = 0.444\nsamples =
3\nvalue = [1, 2]'),
Text(0.7667638483965015, 0.7352941176470589, 'x[2] <= 49.5\ngini =
0.024\nsamples = 81\nvalue = [1, 80]'),
Text(0.7574344023323615, 0.6764705882352942, 'gini = 0.0\nsamples =
65\nvalue = [0, 65]'),
Text(0.7760932944606413, 0.6764705882352942, 'x[2] <= 50.5\ngini =
0.117\nsamples = 16\nvalue = [1, 15]'),
Text(0.7667638483965015, 0.6176470588235294, 'x[0] <= 1.5\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.7574344023323615, 0.5588235294117647, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.7760932944606413, 0.5588235294117647, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
Text(0.7854227405247813, 0.6176470588235294, 'gini = 0.0\nsamples =
12\nvalue = [0, 12]'),
Text(0.917201166180758, 0.8529411764705882, 'x[2] <= 36.5\ngini =
0.5\nsamples = 115\nvalue = [59, 56]'),
Text(0.8985422740524781, 0.7941176470588235, 'x[2] <= 1.5\ngini =
0.499\nsamples = 106\nvalue = [51, 55]'),
Text(0.8892128279883382, 0.7352941176470589, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
Text(0.9078717201166181, 0.7352941176470589, 'x[2] <= 21.5\ngini =
0.5\nsamples = 102\nvalue = [51, 51]'),
Text(0.8577259475218659, 0.6764705882352942, 'x[2] <= 19.0\ngini =
0.461\nsamples = 36\nvalue = [23, 13]'),
Text(0.8274052478134111, 0.6176470588235294, 'x[2] <= 14.75\ngini =
0.48\nsamples = 30\nvalue = [18, 12]'),
Text(0.7947521865889213, 0.5588235294117647, 'x[2] <= 5.5\ngini =
0.415\nsamples = 17\nvalue = [12, 5]'),
Text(0.7760932944606413, 0.5, 'x[2] <= 3.5\ngini = 0.5\nsamples = 8\
nvalue = [4, 4]'),
Text(0.7667638483965015, 0.4411764705882353, 'x[2] <= 2.5\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
```

```
Text(0.7574344023323615, 0.38235294117647056, 'gini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.7760932944606413, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7854227405247813, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.8134110787172012, 0.5, 'x[2] <= 12.0\ngini = 0.198\nsamples = 9\nvalue = [8, 1]'),
Text(0.8040816326530612, 0.4411764705882353, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
Text(0.8227405247813411, 0.4411764705882353, 'x[2] <= 13.5\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.8134110787172012, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.8320699708454811, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.8600583090379009, 0.5588235294117647, 'x[2] <= 15.5\ngini = 0.497\nsamples = 13\nvalue = [6, 7]'),
Text(0.8507288629737609, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.8693877551020408, 0.5, 'x[2] <= 17.5\ngini = 0.496\nsamples = 11\nvalue = [6, 5]'),
Text(0.8600583090379009, 0.4411764705882353, 'x[2] <= 16.5\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.8507288629737609, 0.38235294117647056, 'gini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.8693877551020408, 0.38235294117647056, 'gini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.8787172011661808, 0.4411764705882353, 'gini = 0.49\nsamples = 7\nvalue = [4, 3]'),
Text(0.8880466472303207, 0.6176470588235294, 'x[2] <= 20.5\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.8787172011661808, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.8973760932944607, 0.5588235294117647, 'gini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.9580174927113703, 0.6764705882352942, 'x[2] <= 32.0\ngini = 0.489\nsamples = 66\nvalue = [28, 38]'),
Text(0.9486880466472303, 0.6176470588235294, 'x[2] <= 27.5\ngini = 0.494\nsamples = 63\nvalue = [28, 35]'),
Text(0.9253644314868804, 0.5588235294117647, 'x[2] <= 26.5\ngini = 0.465\nsamples = 19\nvalue = [7, 12]'),
Text(0.9160349854227405, 0.5, 'x[2] <= 24.5\ngini = 0.492\nsamples = 16\nvalue = [7, 9]'),
Text(0.8973760932944607, 0.4411764705882353, 'x[2] <= 23.5\ngini = 0.463\nsamples = 11\nvalue = [4, 7]'),
Text(0.8880466472303207, 0.38235294117647056, 'x[2] <= 22.5\ngini = 0.49\nsamples = 7\nvalue = [3, 4]'),
Text(0.8787172011661808, 0.3235294117647059, 'gini = 0.48\nsamples =
```

```
5\nvalue = [2, 3]'),
  Text(0.8973760932944607, 0.3235294117647059, 'gini = 0.5\nsamples =
2\nvalue = [1, 1]'),
  Text(0.9067055393586005, 0.38235294117647056, 'gini = 0.375\nsamples
= 4\nvalue = [1, 3]'),
  Text(0.9346938775510204, 0.4411764705882353, 'x[2] <= 25.5\ngini =
0.48\nsamples = 5\nvalue = [3, 2]'),
  Text(0.9253644314868804, 0.38235294117647056, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
  Text(0.9440233236151604, 0.38235294117647056, 'gini = 0.444\nsamples
= 3\nvalue = [1, 2]'),
  Text(0.9346938775510204, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
  Text(0.9720116618075801, 0.5588235294117647, 'x[2] <= 28.5\ngini =
0.499\nsamples = 44\nvalue = [21, 23]'),
  Text(0.9626822157434403, 0.5, 'gini = 0.496\nsamples = 35\nvalue =
[16, 19]'),
  Text(0.9813411078717201, 0.5, 'x[2] <= 30.5\ngini = 0.494\nsamples =
9\nvalue = [5, 4]'),
  Text(0.9720116618075801, 0.4411764705882353, 'x[2] <= 29.5\ngini =
0.444\nsamples = 6\nvalue = [4, 2]'),
  Text(0.9626822157434403, 0.38235294117647056, 'gini = 0.444\nsamples
= 3\nvalue = [2, 1]'),
  Text(0.9813411078717201, 0.38235294117647056, 'gini = 0.444\nsamples
= 3\nvalue = [2, 1]'),
  Text(0.99067055393586, 0.4411764705882353, 'gini = 0.444\nsamples =
3\nvalue = [1, 2]'),
  Text(0.9673469387755103, 0.6176470588235294, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
  Text(0.9358600583090378, 0.7941176470588235, 'x[2] <= 55.0\ngini =
0.198\nsamples = 9\nvalue = [8, 1]'),
  Text(0.926530612244898, 0.7352941176470589, 'gini = 0.0\nsamples = 8\
nvalue = [8, 0]'),
  Text(0.9451895043731778, 0.7352941176470589, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]')]
```



```

y_pred3 = dt.predict(X_test)
y_pred3
array([0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0,
0,
      0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0,
0,
      0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
1,
      0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1,
1,
      0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0,
      1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1,
0,
      0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
1,
      0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
0,
      0, 1, 1])

print(classification_report(y_test,y_pred3))

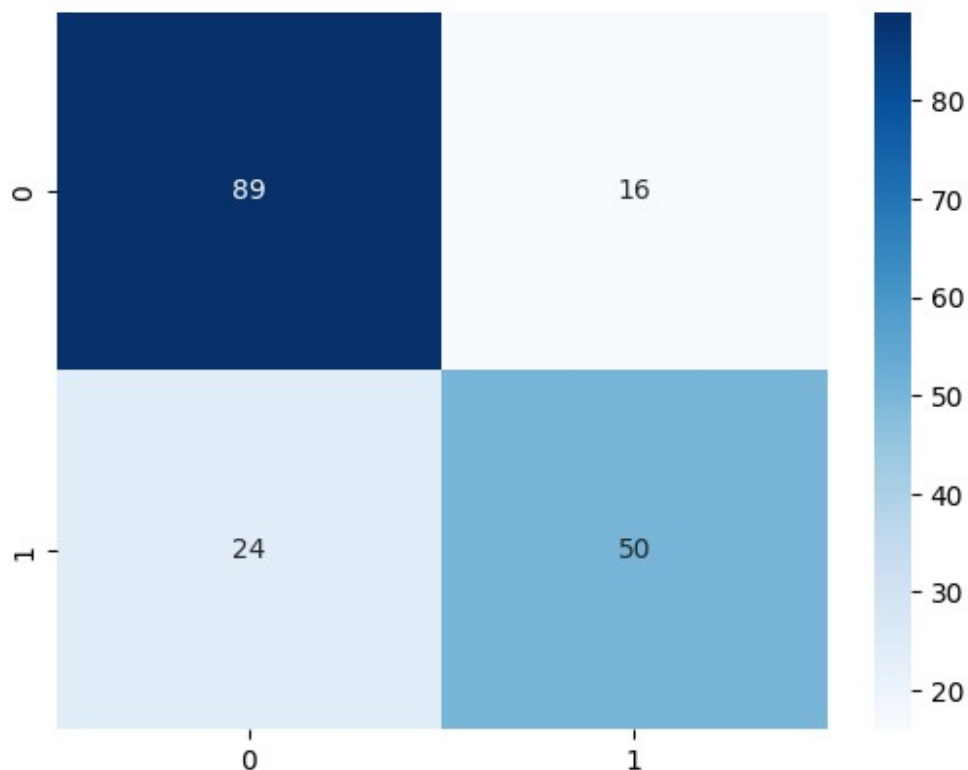
```

	precision	recall	f1-score	support
0	0.79	0.85	0.82	105
1	0.76	0.68	0.71	74

accuracy			0.78	179
macro avg	0.77	0.76	0.77	179
weighted avg	0.78	0.78	0.77	179

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred3)
sns.heatmap(cm,annot=True,cmap='Blues')
```

<Axes: >



```
dt1 = DecisionTreeClassifier(max_depth=5)
dt1.fit(X_train,y_train)
```

```
DecisionTreeClassifier(max_depth=5)
```

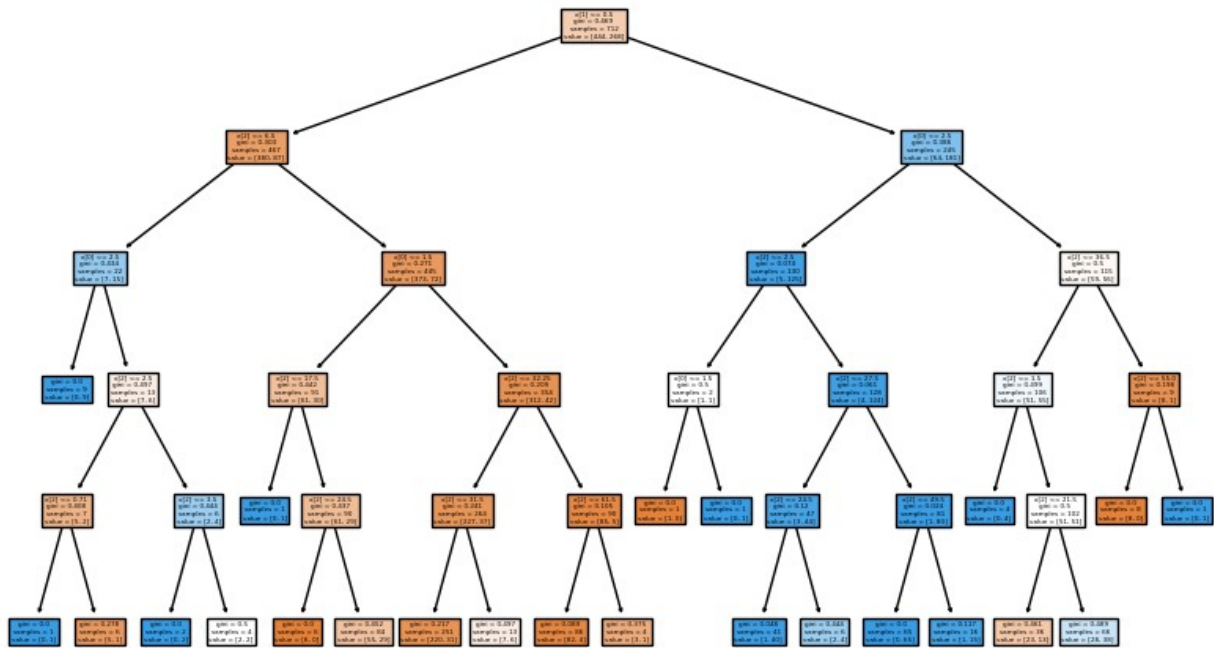
```
plt.figure(figsize=(10,6))
tree.plot_tree(dt1,filled=True)
```

```
[Text(0.4864864864864865, 0.9166666666666666, 'x[1] <= 0.5\ngini = 0.469\nsamples = 712\nvalue = [444, 268]'),
 Text(0.20945945945945946, 0.75, 'x[2] <= 6.5\ngini = 0.303\nsamples = 467\nvalue = [380, 87]'),
 Text(0.08108108108108109, 0.5833333333333333, 'x[0] <= 2.5\ngini = 0.434\nsamples = 22\nvalue = [7, 15]'),
```

```
Text(0.05405405405405406, 0.4166666666666667, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
Text(0.10810810810810811, 0.4166666666666667, 'x[2] <= 2.5\ngini = 0.497\nsamples = 13\nvalue = [7, 6]'),
Text(0.05405405405405406, 0.25, 'x[2] <= 0.71\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),
Text(0.02702702702702703, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.08108108108108109, 0.08333333333333333, 'gini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.16216216216216217, 0.25, 'x[2] <= 3.5\ngini = 0.444\nsamples = 6\nvalue = [2, 4]'),
Text(0.13513513513513514, 0.08333333333333333, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.1891891891891892, 0.08333333333333333, 'gini = 0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.33783783783783783, 0.5833333333333334, 'x[0] <= 1.5\ngini = 0.271\nsamples = 445\nvalue = [373, 72]'),
Text(0.24324324324324326, 0.4166666666666667, 'x[2] <= 17.5\ngini = 0.442\nsamples = 91\nvalue = [61, 30]'),
Text(0.21621621621621623, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.2702702702702703, 0.25, 'x[2] <= 24.5\ngini = 0.437\nsamples = 90\nvalue = [61, 29]'),
Text(0.24324324324324326, 0.08333333333333333, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
Text(0.2972972972972973, 0.08333333333333333, 'gini = 0.452\nsamples = 84\nvalue = [55, 29]'),
Text(0.43243243243243246, 0.4166666666666667, 'x[2] <= 32.25\ngini = 0.209\nsamples = 354\nvalue = [312, 42]'),
Text(0.3783783783783784, 0.25, 'x[2] <= 31.5\ngini = 0.241\nsamples = 264\nvalue = [227, 37]'),
Text(0.35135135135135137, 0.08333333333333333, 'gini = 0.217\nsamples = 251\nvalue = [220, 31]'),
Text(0.40540540540540543, 0.08333333333333333, 'gini = 0.497\nsamples = 13\nvalue = [7, 6]'),
Text(0.4864864864864865, 0.25, 'x[2] <= 61.5\ngini = 0.105\nsamples = 90\nvalue = [85, 5]'),
Text(0.4594594594594595, 0.08333333333333333, 'gini = 0.089\nsamples = 86\nvalue = [82, 4]'),
Text(0.5135135135135135, 0.08333333333333333, 'gini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.7635135135135135, 0.75, 'x[0] <= 2.5\ngini = 0.386\nsamples = 245\nvalue = [64, 181]'),
Text(0.6351351351351351, 0.5833333333333334, 'x[2] <= 2.5\ngini = 0.074\nsamples = 130\nvalue = [5, 125]'),
Text(0.5675675675675675, 0.4166666666666667, 'x[0] <= 1.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.5405405405405406, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1,
```



```
0]'),
  Text(0.5945945945945946, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
  Text(0.7027027027027027, 0.4166666666666667, 'x[2] <= 27.5\ngini =
0.061\nsamples = 128\nvalue = [4, 124]'),
  Text(0.6486486486486487, 0.25, 'x[2] <= 24.5\ngini = 0.12\nsamples =
47\nvalue = [3, 44]'),
  Text(0.6216216216216216, 0.08333333333333333, 'gini = 0.048\nsamples
= 41\nvalue = [1, 40]'),
  Text(0.6756756756756757, 0.08333333333333333, 'gini = 0.444\nsamples
= 6\nvalue = [2, 4]'),
  Text(0.7567567567567568, 0.25, 'x[2] <= 49.5\ngini = 0.024\nsamples =
81\nvalue = [1, 80]'),
  Text(0.7297297297297297, 0.08333333333333333, 'gini = 0.0\nsamples =
65\nvalue = [0, 65]'),
  Text(0.7837837837837838, 0.08333333333333333, 'gini = 0.117\nsamples
= 16\nvalue = [1, 15]'),
  Text(0.8918918918918919, 0.5833333333333334, 'x[2] <= 36.5\ngini =
0.5\nsamples = 115\nvalue = [59, 56]'),
  Text(0.8378378378378378, 0.4166666666666667, 'x[2] <= 1.5\ngini =
0.499\nsamples = 106\nvalue = [51, 55]'),
  Text(0.8108108108108109, 0.25, 'gini = 0.0\nsamples = 4\nvalue = [0,
4]'),
  Text(0.8648648648648649, 0.25, 'x[2] <= 21.5\ngini = 0.5\nsamples =
102\nvalue = [51, 51]'),
  Text(0.8378378378378378, 0.08333333333333333, 'gini = 0.461\nsamples
= 36\nvalue = [23, 13]'),
  Text(0.8918918918918919, 0.08333333333333333, 'gini = 0.489\nsamples
= 66\nvalue = [28, 38]'),
  Text(0.9459459459459459, 0.4166666666666667, 'x[2] <= 55.0\ngini =
0.198\nsamples = 9\nvalue = [8, 1]'),
  Text(0.918918918918919, 0.25, 'gini = 0.0\nsamples = 8\nvalue = [8,
0]'),
  Text(0.972972972972973, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]')]
```



```

y_pred4 = dt1.predict(X_test)
y_pred4

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

print(classification_report(y_test,y_pred4))

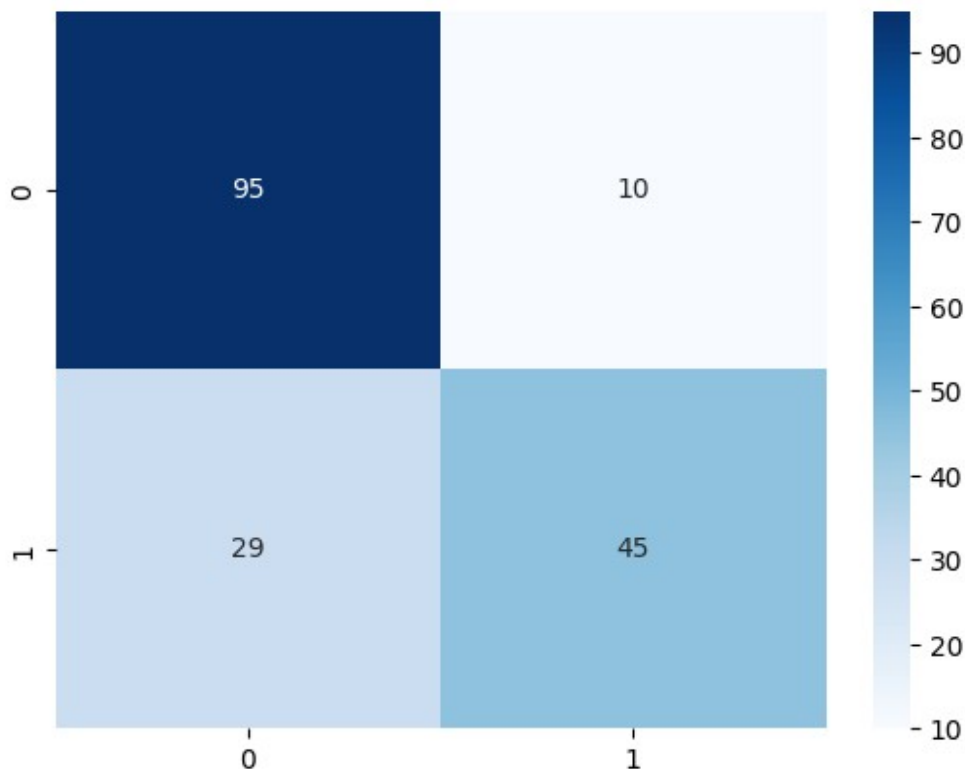
```

	precision	recall	f1-score	support
0	0.77	0.90	0.83	105
1	0.82	0.61	0.70	74

accuracy			0.78	179
macro avg	0.79	0.76	0.76	179
weighted avg	0.79	0.78	0.78	179

```
cm=confusion_matrix(y_test,y_pred4)
sns.heatmap(cm,annot=True,cmap='Blues')
```

<Axes: >



#Entropy gain

```
dt2=DecisionTreeClassifier(criterion='entropy')
dt2.fit(X_train,y_train)
```

```
DecisionTreeClassifier(criterion='entropy')
```

```
plt.figure(figsize=(10,6))
tree.plot_tree(dt2,filled=True)
```

```
[Text(0.4776408520214782, 0.9705882352941176, 'x[1] <= 0.5\nentropy = 0.955\nsamples = 712\nvalue = [444, 268]'),
 Text(0.1605880843335439, 0.9117647058823529, 'x[2] <= 6.5\nentropy = 0.694\nsamples = 467\nvalue = [380, 87]'),
 Text(0.030322173089071383, 0.8529411764705882, 'x[0] <= 2.5\nentropy = 0.902\nsamples = 22\nvalue = [7, 15]'),
```

```
Text(0.02021478205938092, 0.7941176470588235, 'entropy = 0.0\nsamples = 9\nvalue = [0, 9]'),
Text(0.04042956411876184, 0.7941176470588235, 'x[2] <= 2.5\nentropy = 0.996\nsamples = 13\nvalue = [7, 6]'),
Text(0.02021478205938092, 0.7352941176470589, 'x[2] <= 0.71\nentropy = 0.863\nsamples = 7\nvalue = [5, 2]'),
Text(0.01010739102969046, 0.6764705882352942, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.030322173089071383, 0.6764705882352942, 'x[2] <= 1.5\nentropy = 0.65\nsamples = 6\nvalue = [5, 1]'),
Text(0.02021478205938092, 0.6176470588235294, 'entropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
Text(0.04042956411876184, 0.6176470588235294, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.060644346178142766, 0.7352941176470589, 'x[2] <= 3.5\nentropy = 0.918\nsamples = 6\nvalue = [2, 4]'),
Text(0.05053695514845231, 0.6764705882352942, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.07075173720783323, 0.6764705882352942, 'x[2] <= 5.0\nentropy = 1.0\nsamples = 4\nvalue = [2, 2]'),
Text(0.060644346178142766, 0.6176470588235294, 'entropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
Text(0.08085912823752368, 0.6176470588235294, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.29085399557801644, 0.8529411764705882, 'x[0] <= 1.5\nentropy = 0.639\nsamples = 445\nvalue = [373, 72]'),
Text(0.10107391029690461, 0.7941176470588235, 'x[2] <= 17.5\nentropy = 0.915\nsamples = 91\nvalue = [61, 30]'),
Text(0.09096651926721415, 0.7352941176470589, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.11118130132659507, 0.7352941176470589, 'x[2] <= 24.5\nentropy = 0.907\nsamples = 90\nvalue = [61, 29]'),
Text(0.10107391029690461, 0.6764705882352942, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]'),
Text(0.12128869235628553, 0.6764705882352942, 'x[2] <= 27.5\nentropy = 0.93\nsamples = 84\nvalue = [55, 29]'),
Text(0.10107391029690461, 0.6176470588235294, 'x[2] <= 26.0\nentropy = 0.65\nsamples = 6\nvalue = [1, 5]'),
Text(0.09096651926721415, 0.5588235294117647, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.11118130132659507, 0.5588235294117647, 'entropy = 0.811\nsamples = 4\nvalue = [1, 3]'),
Text(0.14150347441566646, 0.6176470588235294, 'x[2] <= 75.0\nentropy = 0.89\nsamples = 78\nvalue = [54, 24]'),
Text(0.131396083385976, 0.5588235294117647, 'x[2] <= 60.5\nentropy = 0.88\nsamples = 77\nvalue = [54, 23]'),
Text(0.12128869235628553, 0.5, 'x[2] <= 30.5\nentropy = 0.909\nsamples = 71\nvalue = [48, 23]'),
Text(0.07770056854074542, 0.4411764705882353, 'x[2] <= 28.5\nentropy
```

```
= 0.755\nsamples = 23\nvalue = [18, 5]'),  
  Text(0.06759317751105495, 0.38235294117647056, 'entropy = 0.811\  
nsamples = 20\nvalue = [15, 5]'),  
  Text(0.08780795957043588, 0.38235294117647056, 'entropy = 0.0\  
nsamples = 3\nvalue = [3, 0]'),  
  Text(0.16487681617182565, 0.4411764705882353, 'x[2] <= 38.5\nentropy  
= 0.954\nsamples = 48\nvalue = [30, 18]'),  
  Text(0.1080227416298168, 0.38235294117647056, 'x[2] <= 31.5\nentropy  
= 0.997\nsamples = 15\nvalue = [7, 8]'),  
  Text(0.09791535060012634, 0.3235294117647059, 'entropy = 0.918\  
nsamples = 3\nvalue = [2, 1]'),  
  Text(0.11813013265950727, 0.3235294117647059, 'x[2] <= 32.5\nentropy  
= 0.98\nsamples = 12\nvalue = [5, 7]'),  
  Text(0.1080227416298168, 0.2647058823529412, 'entropy = 0.0\nsamples  
= 1\nvalue = [0, 1]'),  
  Text(0.12823752368919772, 0.2647058823529412, 'x[2] <= 34.0\nentropy  
= 0.994\nsamples = 11\nvalue = [5, 6]'),  
  Text(0.11813013265950727, 0.20588235294117646, 'entropy = 0.0\  
nsamples = 1\nvalue = [1, 0]'),  
  Text(0.13834491471888818, 0.20588235294117646, 'x[2] <= 35.5\nentropy  
= 0.971\nsamples = 10\nvalue = [4, 6]'),  
  Text(0.12823752368919772, 0.14705882352941177, 'entropy = 0.0\  
nsamples = 2\nvalue = [0, 2]'),  
  Text(0.14845230574857865, 0.14705882352941177, 'x[2] <= 36.5\nentropy  
= 1.0\nsamples = 8\nvalue = [4, 4]'),  
  Text(0.13834491471888818, 0.08823529411764706, 'entropy = 1.0\  
nsamples = 4\nvalue = [2, 2]'),  
  Text(0.15855969677826912, 0.08823529411764706, 'x[2] <= 37.5\nentropy  
= 1.0\nsamples = 4\nvalue = [2, 2]'),  
  Text(0.14845230574857865, 0.029411764705882353, 'entropy = 1.0\  
nsamples = 2\nvalue = [1, 1]'),  
  Text(0.16866708780795958, 0.029411764705882353, 'entropy = 1.0\  
nsamples = 2\nvalue = [1, 1]'),  
  Text(0.2217308907138345, 0.38235294117647056, 'x[2] <= 47.5\nentropy  
= 0.885\nsamples = 33\nvalue = [23, 10]'),  
  Text(0.17877447883765002, 0.3235294117647059, 'x[2] <= 45.25\nentropy  
= 0.592\nsamples = 14\nvalue = [12, 2]'),  
  Text(0.16866708780795958, 0.2647058823529412, 'x[2] <= 41.0\nentropy  
= 0.764\nsamples = 9\nvalue = [7, 2]'),  
  Text(0.15855969677826912, 0.20588235294117646, 'entropy = 0.0\  
nsamples = 2\nvalue = [2, 0]'),  
  Text(0.17877447883765002, 0.20588235294117646, 'x[2] <= 43.0\nentropy  
= 0.863\nsamples = 7\nvalue = [5, 2]'),  
  Text(0.16866708780795958, 0.14705882352941177, 'entropy = 1.0\  
nsamples = 2\nvalue = [1, 1]'),  
  Text(0.18888186986734049, 0.14705882352941177, 'x[2] <= 44.5\nentropy  
= 0.722\nsamples = 5\nvalue = [4, 1]'),  
  Text(0.17877447883765002, 0.08823529411764706, 'entropy = 0.0\  
nsamples = 1\nvalue = [1, 0]'),
```

```
Text(0.19898926089703095, 0.08823529411764706, 'entropy = 0.811\
nsamples = 4\nvalue = [3, 1]'),
Text(0.18888186986734049, 0.2647058823529412, 'entropy = 0.0\nsamples
= 5\nvalue = [5, 0]'),
Text(0.26468730259001894, 0.3235294117647059, 'x[2] <= 48.5\nentropy
= 0.982\nsamples = 19\nvalue = [11, 8]'),
Text(0.2545799115603285, 0.2647058823529412, 'entropy = 0.0\nsamples
= 2\nvalue = [0, 2]'),
Text(0.27479469361970943, 0.2647058823529412, 'x[2] <= 53.0\nentropy
= 0.937\nsamples = 17\nvalue = [11, 6]'),
Text(0.24952621604548325, 0.20588235294117646, 'x[2] <= 50.5\nentropy
= 0.991\nsamples = 9\nvalue = [5, 4]'),
Text(0.22931143398610235, 0.14705882352941177, 'x[2] <= 49.5\nentropy
= 0.971\nsamples = 5\nvalue = [3, 2]'),
Text(0.21920404295641188, 0.08823529411764706, 'entropy = 1.0\
nsamples = 2\nvalue = [1, 1]'),
Text(0.23941882501579279, 0.08823529411764706, 'entropy = 0.918\
nsamples = 3\nvalue = [2, 1]'),
Text(0.2697409981048642, 0.14705882352941177, 'x[2] <= 51.5\nentropy
= 1.0\nsamples = 4\nvalue = [2, 2]'),
Text(0.25963360707517374, 0.08823529411764706, 'entropy = 1.0\
nsamples = 2\nvalue = [1, 1]'),
Text(0.2798483891345546, 0.08823529411764706, 'entropy = 1.0\nsamples
= 2\nvalue = [1, 1]'),
Text(0.30006317119393555, 0.20588235294117646, 'x[2] <= 55.5\nentropy
= 0.811\nsamples = 8\nvalue = [6, 2]'),
Text(0.2899557801642451, 0.14705882352941177, 'entropy = 0.0\nsamples
= 3\nvalue = [3, 0]'),
Text(0.31017056222362605, 0.14705882352941177, 'x[2] <= 57.0\nentropy
= 0.971\nsamples = 5\nvalue = [3, 2]'),
Text(0.30006317119393555, 0.08823529411764706, 'entropy = 1.0\
nsamples = 2\nvalue = [1, 1]'),
Text(0.3202779532533165, 0.08823529411764706, 'x[2] <= 59.0\nentropy
= 0.918\nsamples = 3\nvalue = [2, 1]'),
Text(0.31017056222362605, 0.029411764705882353, 'entropy = 0.0\
nsamples = 1\nvalue = [1, 0]'),
Text(0.330385344283007, 0.029411764705882353, 'entropy = 1.0\nsamples
= 2\nvalue = [1, 1]'),
Text(0.14150347441566646, 0.5, 'entropy = 0.0\nsamples = 6\nvalue =
[6, 0]'),
Text(0.15161086544535693, 0.5588235294117647, 'entropy = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.48063408085912823, 0.7941176470588235, 'x[2] <= 32.25\nentropy
= 0.525\nsamples = 354\nvalue = [312, 42]'),
Text(0.345349020846494, 0.7352941176470589, 'x[2] <= 31.5\nentropy =
0.585\nsamples = 264\nvalue = [227, 37]'),
Text(0.3007738471257107, 0.6764705882352942, 'x[2] <= 13.0\nentropy =
0.539\nsamples = 251\nvalue = [220, 31]'),
Text(0.2419456727732154, 0.6176470588235294, 'x[2] <= 11.5\nentropy =
```

```
0.881\nsamples = 10\nvalue = [7, 3]'),
Text(0.23183828174352494, 0.5588235294117647, 'x[0] <= 2.5\nentropy =
0.764\nsamples = 9\nvalue = [7, 2]'),
Text(0.2217308907138345, 0.5, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]'),
Text(0.2419456727732154, 0.5, 'x[2] <= 8.5\nentropy = 0.544\nsamples
= 8\nvalue = [7, 1]'),
Text(0.23183828174352494, 0.4411764705882353, 'entropy = 0.0\nsamples
= 3\nvalue = [3, 0]'),
Text(0.2520530638029059, 0.4411764705882353, 'x[2] <= 10.0\nentropy =
0.722\nsamples = 5\nvalue = [4, 1]'),
Text(0.2419456727732154, 0.38235294117647056, 'entropy = 0.918\n
samples = 3\nvalue = [2, 1]'),
Text(0.2621604548325963, 0.38235294117647056, 'entropy = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.2520530638029059, 0.5588235294117647, 'entropy = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.3596020214782059, 0.6176470588235294, 'x[2] <= 18.5\nentropy =
0.518\nsamples = 241\nvalue = [213, 28]'),
Text(0.29248262792166774, 0.5588235294117647, 'x[2] <= 16.5\nentropy
= 0.242\nsamples = 25\nvalue = [24, 1]'),
Text(0.28237523689197724, 0.5, 'x[2] <= 15.5\nentropy = 0.439\n
samples = 11\nvalue = [10, 1]'),
Text(0.2722678458622868, 0.4411764705882353, 'entropy = 0.0\nsamples
= 3\nvalue = [3, 0]'),
Text(0.29248262792166774, 0.4411764705882353, 'x[0] <= 2.5\nentropy =
0.544\nsamples = 8\nvalue = [7, 1]'),
Text(0.28237523689197724, 0.38235294117647056, 'entropy = 0.0\n
samples = 2\nvalue = [2, 0]'),
Text(0.3025900189513582, 0.38235294117647056, 'entropy = 0.65\n
samples = 6\nvalue = [5, 1]'),
Text(0.3025900189513582, 0.5, 'entropy = 0.0\nsamples = 14\nvalue =
[14, 0]'),
Text(0.42672141503474414, 0.5588235294117647, 'x[2] <= 27.5\nentropy
= 0.544\nsamples = 216\nvalue = [189, 27]'),
Text(0.3815540113708149, 0.5, 'x[2] <= 26.5\nentropy = 0.633\nsamples
= 94\nvalue = [79, 15]'),
Text(0.35186355022109916, 0.4411764705882353, 'x[0] <= 2.5\nentropy =
0.571\nsamples = 89\nvalue = [77, 12]'),
Text(0.3228048010107391, 0.38235294117647056, 'x[2] <= 20.0\nentropy
= 0.31\nsamples = 18\nvalue = [17, 1]'),
Text(0.31269740998104867, 0.3235294117647059, 'entropy = 0.811\n
samples = 4\nvalue = [3, 1]'),
Text(0.33291219204042954, 0.3235294117647059, 'entropy = 0.0\nsamples
= 14\nvalue = [14, 0]'),
Text(0.3809222994314593, 0.38235294117647056, 'x[2] <= 20.5\nentropy
= 0.622\nsamples = 71\nvalue = [60, 11]'),
Text(0.3531269740998105, 0.3235294117647059, 'x[2] <= 19.5\nentropy =
0.722\nsamples = 20\nvalue = [16, 4]'),
```

```
Text(0.34301958307012004, 0.2647058823529412, 'entropy = 0.469\
nsamples = 10\nvalue = [9, 1]'),
Text(0.36323436512950097, 0.2647058823529412, 'entropy = 0.881\
nsamples = 10\nvalue = [7, 3]'),
Text(0.408717624763108, 0.3235294117647059, 'x[2] <= 23.5\nentropy =
0.577\nsamples = 51\nvalue = [44, 7]'),
Text(0.38344914718888184, 0.2647058823529412, 'x[2] <= 22.5\nentropy
= 0.414\nsamples = 24\nvalue = [22, 2]'),
Text(0.3733417561591914, 0.20588235294117646, 'x[2] <= 21.5\nentropy
= 0.426\nsamples = 23\nvalue = [21, 2]'),
Text(0.36323436512950097, 0.14705882352941177, 'entropy = 0.469\
nsamples = 10\nvalue = [9, 1]'),
Text(0.38344914718888184, 0.14705882352941177, 'entropy = 0.391\
nsamples = 13\nvalue = [12, 1]'),
Text(0.39355653821857234, 0.20588235294117646, 'entropy = 0.0\
nsamples = 1\nvalue = [1, 0]'),
Text(0.4339861023373342, 0.2647058823529412, 'x[2] <= 24.75\nentropy
= 0.691\nsamples = 27\nvalue = [22, 5]'),
Text(0.41377132027795327, 0.20588235294117646, 'x[2] <= 24.25\
entropy = 0.592\nsamples = 7\nvalue = [6, 1]'),
Text(0.4036639292482628, 0.14705882352941177, 'entropy = 0.65\
nsamples = 6\nvalue = [5, 1]'),
Text(0.4238787113076437, 0.14705882352941177, 'entropy = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.4542008843967151, 0.20588235294117646, 'x[2] <= 25.5\nentropy
= 0.722\nsamples = 20\nvalue = [16, 4]'),
Text(0.44409349336702464, 0.14705882352941177, 'entropy = 0.764\
nsamples = 9\nvalue = [7, 2]'),
Text(0.46430827542640557, 0.14705882352941177, 'entropy = 0.684\
nsamples = 11\nvalue = [9, 2]'),
Text(0.41124447252053065, 0.4411764705882353, 'x[0] <= 2.5\nentropy =
0.971\nsamples = 5\nvalue = [2, 3]'),
Text(0.40113708149084015, 0.38235294117647056, 'entropy = 0.0\
nsamples = 2\nvalue = [2, 0]'),
Text(0.4213518635502211, 0.38235294117647056, 'entropy = 0.0\nsamples
= 3\nvalue = [0, 3]'),
Text(0.4718888186986734, 0.5, 'x[2] <= 28.75\nentropy = 0.464\
nsamples = 122\nvalue = [110, 12]'),
Text(0.4516740366392925, 0.4411764705882353, 'x[0] <= 2.5\nentropy =
0.405\nsamples = 99\nvalue = [91, 8]'),
Text(0.441566645609602, 0.38235294117647056, 'entropy = 0.684\
nsamples = 11\nvalue = [9, 2]'),
Text(0.46178142766898295, 0.38235294117647056, 'x[2] <= 28.25\
entropy = 0.359\nsamples = 88\nvalue = [82, 6]'),
Text(0.4516740366392925, 0.3235294117647059, 'entropy = 0.365\
nsamples = 86\nvalue = [80, 6]'),
Text(0.4718888186986734, 0.3235294117647059, 'entropy = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.4921036007580543, 0.4411764705882353, 'x[0] <= 2.5\nentropy =
```



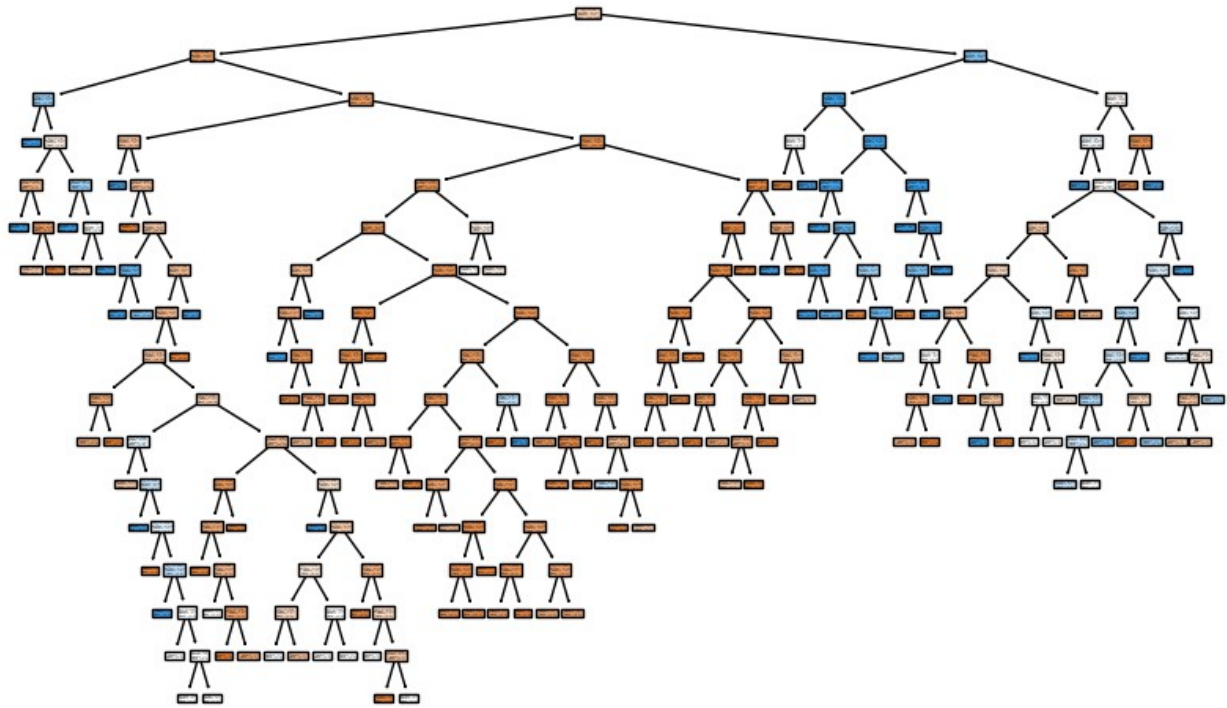
```
0.667\nsamples = 23\nvalue = [19, 4]'),
Text(0.4819962097283639, 0.38235294117647056, 'entropy = 0.0\nsamples
= 9\nvalue = [9, 0]'),
Text(0.5022109917877448, 0.38235294117647056, 'x[2] <= 29.5\nentropy
= 0.863\nsamples = 14\nvalue = [10, 4]'),
Text(0.4921036007580543, 0.3235294117647059, 'entropy = 0.971\n
samples = 5\nvalue = [2, 3]'),
Text(0.5123183828174352, 0.3235294117647059, 'x[2] <= 30.5\nentropy =
0.503\nsamples = 9\nvalue = [8, 1]'),
Text(0.5022109917877448, 0.2647058823529412, 'entropy = 0.0\nsamples
= 6\nvalue = [6, 0]'),
Text(0.5224257738471257, 0.2647058823529412, 'entropy = 0.918\n
samples = 3\nvalue = [2, 1]'),
Text(0.38992419456727734, 0.6764705882352942, 'x[0] <= 2.5\nentropy =
0.996\nsamples = 13\nvalue = [7, 6]'),
Text(0.37981680353758684, 0.6176470588235294, 'entropy = 1.0\nsamples
= 2\nvalue = [1, 1]'),
Text(0.4000315855969678, 0.6176470588235294, 'entropy = 0.994\n
samples = 11\nvalue = [6, 5]'),
Text(0.6159191408717625, 0.7352941176470589, 'x[2] <= 61.5\nentropy =
0.31\nsamples = 90\nvalue = [85, 5]'),
Text(0.5957043588123816, 0.6764705882352942, 'x[2] <= 44.5\nentropy =
0.271\nsamples = 86\nvalue = [82, 4]'),
Text(0.5855969677826911, 0.6176470588235294, 'x[2] <= 38.5\nentropy =
0.353\nsamples = 60\nvalue = [56, 4]'),
Text(0.5527479469361971, 0.5588235294117647, 'x[0] <= 2.5\nentropy =
0.187\nsamples = 35\nvalue = [34, 1]'),
Text(0.5426405559065066, 0.5, 'x[2] <= 34.5\nentropy = 0.391\nsamples
= 13\nvalue = [12, 1]'),
Text(0.5325331648768161, 0.4411764705882353, 'x[2] <= 33.5\nentropy =
0.544\nsamples = 8\nvalue = [7, 1]'),
Text(0.5224257738471257, 0.38235294117647056, 'entropy = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.5426405559065066, 0.38235294117647056, 'entropy = 0.65\n
samples = 6\nvalue = [5, 1]'),
Text(0.5527479469361971, 0.4411764705882353, 'entropy = 0.0\nsamples
= 5\nvalue = [5, 0]'),
Text(0.5628553379658876, 0.5, 'entropy = 0.0\nsamples = 22\nvalue =
[22, 0]'),
Text(0.6184459886291851, 0.5588235294117647, 'x[2] <= 43.5\nentropy =
0.529\nsamples = 25\nvalue = [22, 3]'),
Text(0.593177511054959, 0.5, 'x[2] <= 39.5\nentropy = 0.454\nsamples
= 21\nvalue = [19, 2]'),
Text(0.572962728995578, 0.4411764705882353, 'x[0] <= 2.5\nentropy =
0.65\nsamples = 6\nvalue = [5, 1]'),
Text(0.5628553379658876, 0.38235294117647056, 'entropy = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.5830701200252685, 0.38235294117647056, 'entropy = 0.811\n
samples = 4\nvalue = [3, 1]'),
```

```
Text(0.6133922931143398, 0.4411764705882353, 'x[0] <= 2.5\nentropy = 0.353\nsamples = 15\nvalue = [14, 1]'),
Text(0.6032849020846494, 0.38235294117647056, 'x[2] <= 42.5\nentropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
Text(0.593177511054959, 0.3235294117647059, 'entropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
Text(0.6133922931143398, 0.3235294117647059, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6234996841440303, 0.38235294117647056, 'entropy = 0.0\nsamples = 11\nvalue = [11, 0]'),
Text(0.6437144662034112, 0.5, 'x[0] <= 2.5\nentropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
Text(0.6336070751737208, 0.4411764705882353, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6538218572331017, 0.4411764705882353, 'entropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
Text(0.605811749842072, 0.6176470588235294, 'entropy = 0.0\nsamples = 26\nvalue = [26, 0]'),
Text(0.6361339229311433, 0.6764705882352942, 'x[2] <= 66.0\nentropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
Text(0.626026531901453, 0.6176470588235294, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6462413139608338, 0.6176470588235294, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.7946936197094125, 0.9117647058823529, 'x[0] <= 2.5\nentropy = 0.829\nsamples = 245\nvalue = [64, 181]'),
Text(0.6790903348073278, 0.8529411764705882, 'x[2] <= 2.5\nentropy = 0.235\nsamples = 130\nvalue = [5, 125]'),
Text(0.6462413139608338, 0.7941176470588235, 'x[0] <= 1.5\nentropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
Text(0.6361339229311433, 0.7352941176470589, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6563487049905243, 0.7352941176470589, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7119393556538218, 0.7941176470588235, 'x[2] <= 27.5\nentropy = 0.201\nsamples = 128\nvalue = [4, 124]'),
Text(0.6765634870499052, 0.7352941176470589, 'x[2] <= 23.5\nentropy = 0.342\nsamples = 47\nvalue = [3, 44]'),
Text(0.6664560960202148, 0.6764705882352942, 'entropy = 0.0\nsamples = 29\nvalue = [0, 29]'),
Text(0.6866708780795957, 0.6764705882352942, 'x[2] <= 24.5\nentropy = 0.65\nsamples = 18\nvalue = [3, 15]'),
Text(0.6664560960202148, 0.6176470588235294, 'x[0] <= 1.5\nentropy = 0.414\nsamples = 12\nvalue = [1, 11]'),
Text(0.6563487049905243, 0.5588235294117647, 'entropy = 0.0\nsamples = 5\nvalue = [0, 5]'),
Text(0.6765634870499052, 0.5588235294117647, 'entropy = 0.592\nsamples = 7\nvalue = [1, 6]'),
Text(0.7068856601389766, 0.6176470588235294, 'x[0] <= 1.5\nentropy =
```

```
0.918\nsamples = 6\nvalue = [2, 4]'),
Text(0.6967782691092862, 0.5588235294117647, 'entropy = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.7169930511686671, 0.5588235294117647, 'x[2] <= 26.0\nentropy =
0.722\nsamples = 5\nvalue = [1, 4]'),
Text(0.7068856601389766, 0.5, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]'),
Text(0.7271004421983576, 0.5, 'entropy = 0.918\nsamples = 3\nvalue =
[1, 2]'),
Text(0.7473152242577384, 0.7352941176470589, 'x[2] <= 49.5\nentropy =
0.096\nsamples = 81\nvalue = [1, 80]'),
Text(0.7372078332280481, 0.6764705882352942, 'entropy = 0.0\nsamples
= 65\nvalue = [0, 65]'),
Text(0.7574226152874289, 0.6764705882352942, 'x[2] <= 50.5\nentropy =
0.337\nsamples = 16\nvalue = [1, 15]'),
Text(0.7473152242577384, 0.6176470588235294, 'x[0] <= 1.5\nentropy =
0.811\nsamples = 4\nvalue = [1, 3]'),
Text(0.7372078332280481, 0.5588235294117647, 'entropy = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.7574226152874289, 0.5588235294117647, 'entropy = 0.0\nsamples
= 3\nvalue = [0, 3]'),
Text(0.7675300063171194, 0.6176470588235294, 'entropy = 0.0\nsamples
= 12\nvalue = [0, 12]'),
Text(0.9102969046114971, 0.8529411764705882, 'x[2] <= 36.5\nentropy =
1.0\nsamples = 115\nvalue = [59, 56]'),
Text(0.8900821225521163, 0.7941176470588235, 'x[2] <= 1.5\nentropy =
0.999\nsamples = 106\nvalue = [51, 55]'),
Text(0.8799747315224258, 0.7352941176470589, 'entropy = 0.0\nsamples
= 4\nvalue = [0, 4]'),
Text(0.9001895135818067, 0.7352941176470589, 'x[2] <= 21.5\nentropy =
1.0\nsamples = 102\nvalue = [51, 51]'),
Text(0.8458622867972204, 0.6764705882352942, 'x[2] <= 19.0\nentropy =
0.944\nsamples = 36\nvalue = [23, 13]'),
Text(0.8130132659507264, 0.6176470588235294, 'x[2] <= 14.75\nentropy
= 0.971\nsamples = 30\nvalue = [18, 12]'),
Text(0.7776373973468098, 0.5588235294117647, 'x[2] <= 5.5\nentropy =
0.874\nsamples = 17\nvalue = [12, 5]'),
Text(0.7574226152874289, 0.5, 'x[2] <= 3.5\nentropy = 1.0\nsamples =
8\nvalue = [4, 4]'),
Text(0.7473152242577384, 0.4411764705882353, 'x[2] <= 2.5\nentropy =
0.722\nsamples = 5\nvalue = [4, 1]'),
Text(0.7372078332280481, 0.38235294117647056, 'entropy = 0.811\n
samples = 4\nvalue = [3, 1]'),
Text(0.7574226152874289, 0.38235294117647056, 'entropy = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.7675300063171194, 0.4411764705882353, 'entropy = 0.0\nsamples
= 3\nvalue = [0, 3]'),
Text(0.7978521794061908, 0.5, 'x[2] <= 12.0\nentropy = 0.503\nsamples
= 9\nvalue = [8, 1]'),
```

```
Text(0.7877447883765003, 0.4411764705882353, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]'),
Text(0.8079595704358813, 0.4411764705882353, 'x[2] <= 13.5\nentropy = 0.918\nsamples = 3\nvalue = [2, 1]'),
Text(0.7978521794061908, 0.38235294117647056, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.8180669614655717, 0.38235294117647056, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.848389134554643, 0.5588235294117647, 'x[2] <= 15.5\nentropy = 0.996\nsamples = 13\nvalue = [6, 7]'),
Text(0.8382817435249527, 0.5, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.8584965255843335, 0.5, 'x[2] <= 17.5\nentropy = 0.994\nsamples = 11\nvalue = [6, 5]'),
Text(0.848389134554643, 0.4411764705882353, 'x[2] <= 16.5\nentropy = 1.0\nsamples = 4\nvalue = [2, 2]'),
Text(0.8382817435249527, 0.38235294117647056, 'entropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
Text(0.8584965255843335, 0.38235294117647056, 'entropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
Text(0.868603916614024, 0.4411764705882353, 'entropy = 0.985\nsamples = 7\nvalue = [4, 3]'),
Text(0.8787113076437144, 0.6176470588235294, 'x[2] <= 20.5\nentropy = 0.65\nsamples = 6\nvalue = [5, 1]'),
Text(0.868603916614024, 0.5588235294117647, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.8888186986734049, 0.5588235294117647, 'entropy = 0.811\nsamples = 4\nvalue = [3, 1]'),
Text(0.9545167403663929, 0.6764705882352942, 'x[2] <= 32.0\nentropy = 0.983\nsamples = 66\nvalue = [28, 38]'),
Text(0.9444093493367025, 0.6176470588235294, 'x[2] <= 27.5\nentropy = 0.991\nsamples = 63\nvalue = [28, 35]'),
Text(0.9191408717624763, 0.5588235294117647, 'x[2] <= 26.5\nentropy = 0.949\nsamples = 19\nvalue = [7, 12]'),
Text(0.9090334807327859, 0.5, 'x[2] <= 24.5\nentropy = 0.989\nsamples = 16\nvalue = [7, 9]'),
Text(0.8888186986734049, 0.4411764705882353, 'x[2] <= 23.5\nentropy = 0.946\nsamples = 11\nvalue = [4, 7]'),
Text(0.8787113076437144, 0.38235294117647056, 'x[2] <= 22.5\nentropy = 0.985\nsamples = 7\nvalue = [3, 4]'),
Text(0.868603916614024, 0.3235294117647059, 'entropy = 0.971\nsamples = 5\nvalue = [2, 3]'),
Text(0.8888186986734049, 0.3235294117647059, 'entropy = 1.0\nsamples = 2\nvalue = [1, 1]'),
Text(0.8989260897030954, 0.38235294117647056, 'entropy = 0.811\nsamples = 4\nvalue = [1, 3]'),
Text(0.9292482627921668, 0.4411764705882353, 'x[2] <= 25.5\nentropy = 0.971\nsamples = 5\nvalue = [3, 2]'),
Text(0.9191408717624763, 0.38235294117647056, 'entropy = 0.0\nsamples
```

```
= 2\nvalue = [2, 0]'),  
  Text(0.9393556538218573, 0.38235294117647056, 'entropy = 0.918\  
nsamples = 3\nvalue = [1, 2]'),  
  Text(0.9292482627921668, 0.5, 'entropy = 0.0\nsamples = 3\nvalue =  
[0, 3]'),  
  Text(0.9696778269109286, 0.5588235294117647, 'x[2] <= 28.5\nentropy =  
0.999\nsamples = 44\nvalue = [21, 23]'),  
  Text(0.9595704358812381, 0.5, 'entropy = 0.995\nsamples = 35\nvalue =  
[16, 19]'),  
  Text(0.9797852179406191, 0.5, 'x[2] <= 30.5\nentropy = 0.991\nsamples  
= 9\nvalue = [5, 4]'),  
  Text(0.9696778269109286, 0.4411764705882353, 'x[2] <= 29.5\nentropy =  
0.918\nsamples = 6\nvalue = [4, 2]'),  
  Text(0.9595704358812381, 0.38235294117647056, 'entropy = 0.918\  
nsamples = 3\nvalue = [2, 1]'),  
  Text(0.9797852179406191, 0.38235294117647056, 'entropy = 0.918\  
nsamples = 3\nvalue = [2, 1]'),  
  Text(0.9898926089703095, 0.4411764705882353, 'entropy = 0.918\  
nsamples = 3\nvalue = [1, 2]'),  
  Text(0.9646241313960834, 0.6176470588235294, 'entropy = 0.0\nsamples  
= 3\nvalue = [0, 3]'),  
  Text(0.930511686670878, 0.7941176470588235, 'x[2] <= 55.0\nentropy =  
0.503\nsamples = 9\nvalue = [8, 1]'),  
  Text(0.9204042956411876, 0.7352941176470589, 'entropy = 0.0\nsamples  
= 8\nvalue = [8, 0]'),  
  Text(0.9406190777005685, 0.7352941176470589, 'entropy = 0.0\nsamples  
= 1\nvalue = [0, 1]'))]
```



```

y_pred5 = dt2.predict(x_test)
y_pred5
array([0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0,
0,
      0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0,
0,
      0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
1,
      0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1,
1,
      0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0,
      1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1,
0,
      0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
1,
      0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
0,
      0, 1, 1])

print(classification_report(y_test,y_pred5))

```

	precision	recall	f1-score	support
0	0.79	0.85	0.82	105
1	0.76	0.68	0.71	74

accuracy			0.78	179
macro avg	0.77	0.76	0.77	179
weighted avg	0.78	0.78	0.77	179

```
dt3 = DecisionTreeClassifier(max_depth=5,criterion='entropy')  
dt3.fit(X_train,y_train)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=5)
```

```
plt.figure(figsize=(10,6))
```

```
<Figure size 1000x600 with 0 Axes>
```

```
<Figure size 1000x600 with 0 Axes>
```