

1. Implementation of ID3 algorithm using iris dataset and the parameter for attribute selection is set to Gini index.

```
import pandas as pd
import numpy as np
data = pd.read_csv('iris.data.csv')
data
```

	sepal_length	sepal_width	petal_length	petal_width	Class_name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

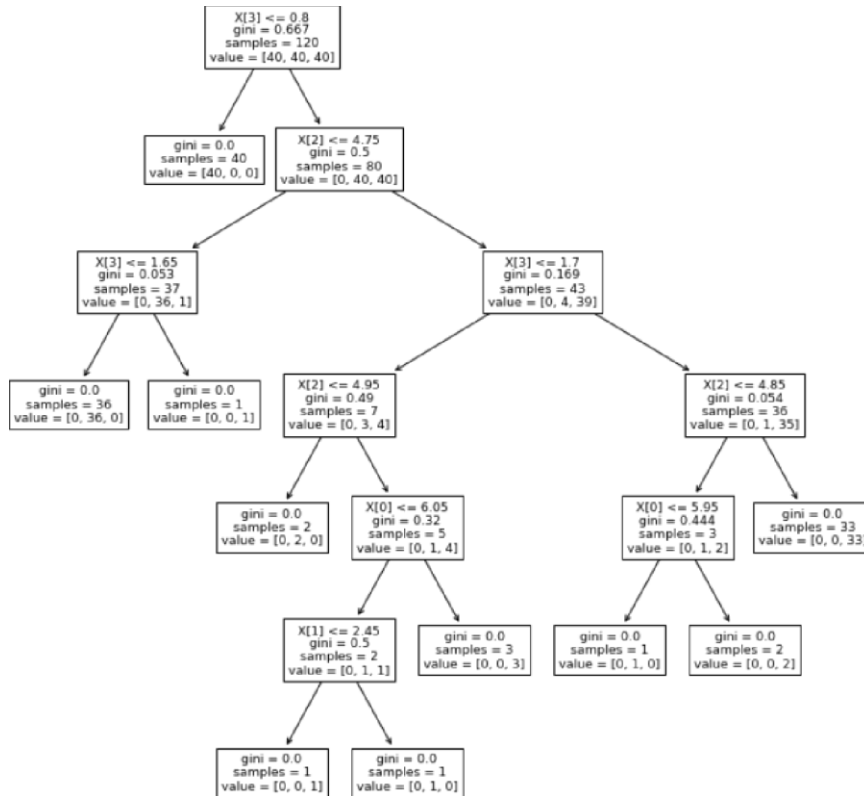
X = data.iloc[:, :-1]
y = data.iloc[:, -1]

datasets = train_test_split(X, y, test_size=0.2)

X_train, X_test, y_train, y_test = datasets

model = DecisionTreeClassifier(criterion="gini")
model.fit(X_train, y_train)

plt.figure(figsize=(12,12))
tree.plot_tree(model)
plt.show()
```



```

y_pred = model.predict(X_test)

from sklearn.metrics import confusion matrix
from sklearn.metrics import classification report
from sklearn.metrics import accuracy score

y_true = ytest

print('Accuracy : ',accuracy score(y_pred, y_test))

print('\nConfusion Matrix: \n', confusion matrix(y_true, y_pred))

matrix = classification report(y_true, y_pred)
print('\nClassification report : \n', matrix)

```

Accuracy : 0.9666666666666667

Confusion Matrix:

```
[[10  0  0]
```

```
[ 0  0 10]]
```

ClassifIcatiDn	precision	recall	f1-score	support
Iris-setDsa	1.00	1.00	1.00	10
Iris-versicolor	1.00	0.90	0.95	10
Iris-virginica	0.91	1.00	0.95	10
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

2. Implementation of ID3 algorithm using party dataset and the parameter for attribute selection is set to Gini index.

```
import pandas as pd
from sklearn import metrics
df = pd.read_csv('party.csv')
df
```

	Deadline	Party	Lazy	Activity
0	Urgent	Yes	Yes	Party
1	Urgent	No	Yes	Study
2	Near	Yes	Yes	Party
3	None	Yes	No	Party
4	None	No	Yes	Pub
5	None	Yes	No	Party
6	Near	No	No	Study
7	Near	No	Yes	TV
8	Near	Yes	Yes	Party
9	Urgent	No	No	Study

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt

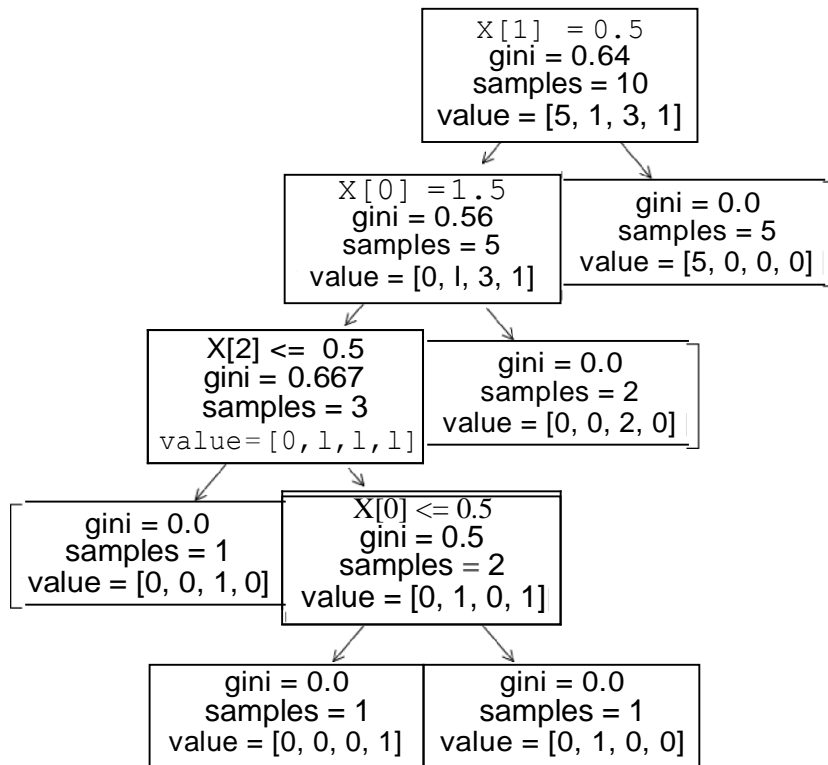
from sklearn import preprocessing
le = preprocessing.LabelEncoder()

df = df.apply(le.fit_transform)

X = df.iloc[:, :-1]
y = df.iloc[:, -1]
```

```
model = DecisionTreeClassifier(criterion="gini")
model.fit(X, y)

plt.figure(figsize=(12,12))
tree.plot_tree(model)
plt.show()
```



```

y_pred = model.predict(X)

from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

y_true = y

print('Accuracy:', accuracy_score(y_pred, y))

print('Confusion Matrix: \n', confusion_matrix(y_true, y_pred))

matrix = classification_report(y_true, y_pred)
print('Classification report: \n', matrix)

```

Accuracy : 1.0

Confusion matrix:

[[5 0 0 0]

[0 0 3 0]

[0 0 0 1]]

Classification report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	1
2	1.00	1.00	1.00	3
3	1.00	1.00	1.00	1
accuracy			1.00	10
macro avg	1.00	1.00	1.00	10
weighted avg	1.00	1.00	1.00	10