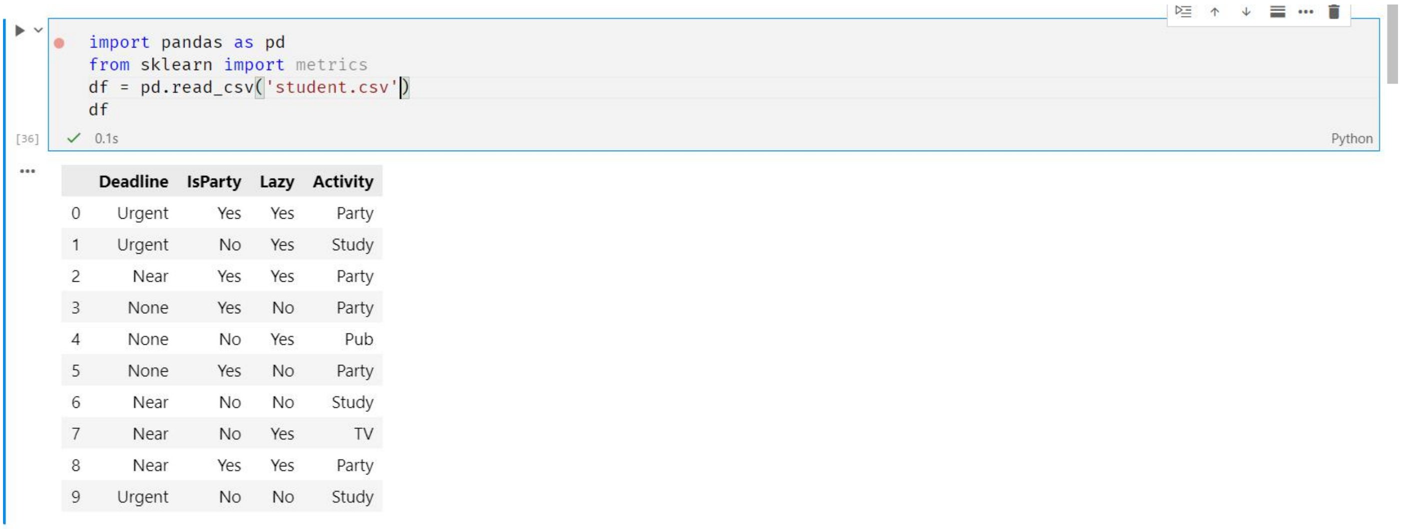
2019103555 **PRANAVA RAMAN B M S** **22/11/2021**

MACHINE LEARNING - LAB 12 - SPOT

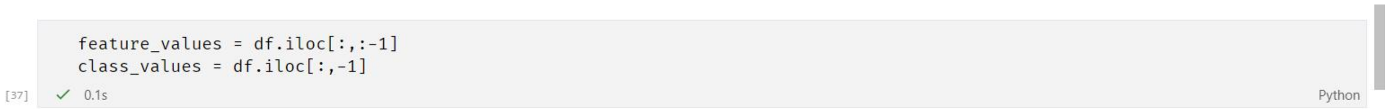
Implementation of ID3 algorithm and the parameter for attribute selection is set to Gini index.

PARTY DATASET:-

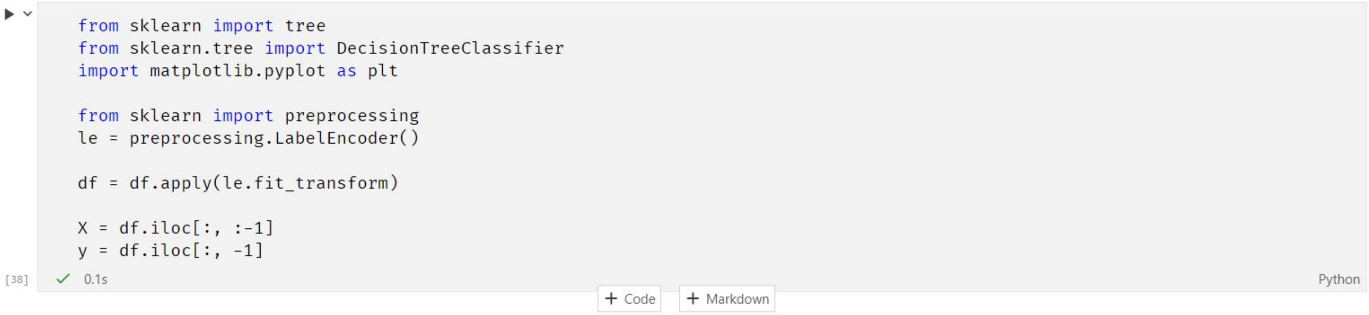


```
import pandas as pd
from sklearn import metrics
df = pd.read_csv('student.csv')
df
```

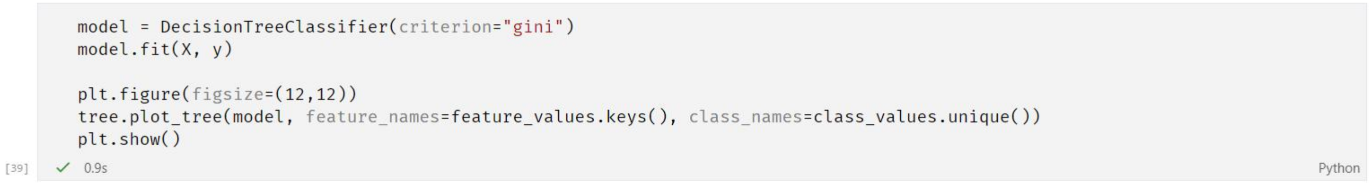
	Deadline	IsParty	Lazy	Activity
0	Urgent	Yes	Yes	Party
1	Urgent	No	Yes	Study
2	Near	Yes	Yes	Party
3	None	Yes	No	Party
4	None	No	Yes	Pub
5	None	Yes	No	Party
6	Near	No	No	Study
7	Near	No	Yes	TV
8	Near	Yes	Yes	Party
9	Urgent	No	No	Study



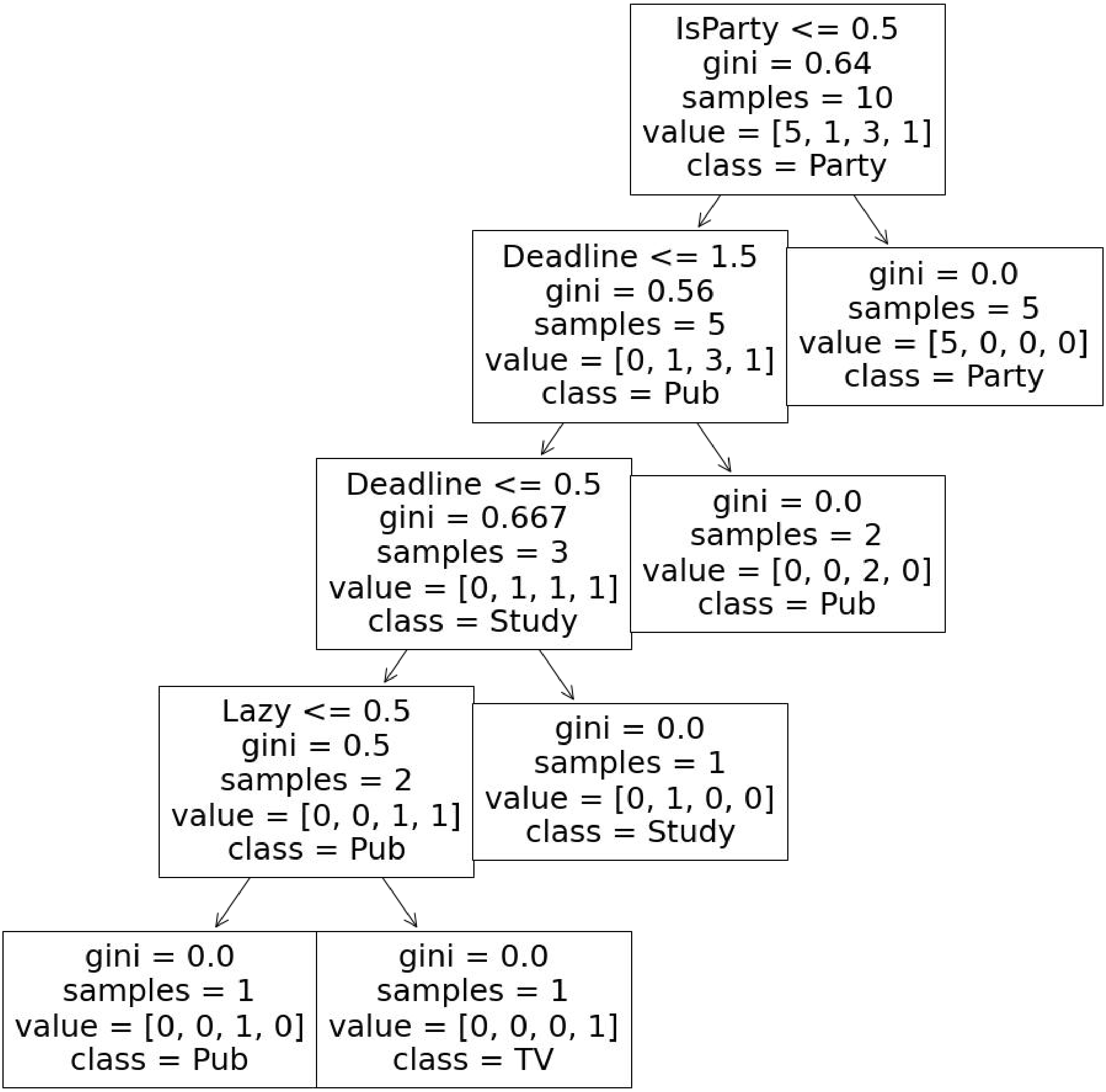
```
feature_values = df.iloc[:,:-1]
class_values = df.iloc[:,-1]
```



```
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, feature_names=feature_values.keys(), class_names=class_values.unique())
plt.show()
```


IsParty <= 0.5
gini = 0.64
samples = 10
value = [5, 1, 3, 1]
class = Party
Deadline <= 1.5
gini = 0.56
samples = 5
value = [0, 1, 3, 1]
class = Pub
gini = 0.0
samples = 5
value = [5, 0, 0, 0]
class = Party
Deadline <= 0.5
gini = 0.667
samples = 3
value = [0, 1, 1, 1]
class = Study
gini = 0.0
samples = 2
value = [0, 0, 2, 0]
class = Pub
Lazy <= 0.5
gini = 0.5
samples = 2
value = [0, 0, 1, 1]
class = Pub
gini = 0.0
samples = 1
value = [0, 1, 0, 0]
class = Study
gini = 0.0
samples = 1
value = [0, 0, 1, 0]
class = Pub
gini = 0.0
samples = 1
value = [0, 0, 0, 1]
class = TV

```python
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('\nConfusion Matrix: \n', confusion_matrix(y_true, y_pred))
```

[69] ✓ 0.6s Python

```
Accuracy :  1.0

Confusion Matrix:
 [[5 0 0 0]
 [0 1 0 0]
 [0 0 3 0]
 [0 0 0 1]]
```

```python
matrix = classification_report(y_true,y_pred)
print('\nClassification report : \n',matrix)
```

[70] ✓ 0.2s Python

```
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
```

Using IRIS dataset

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

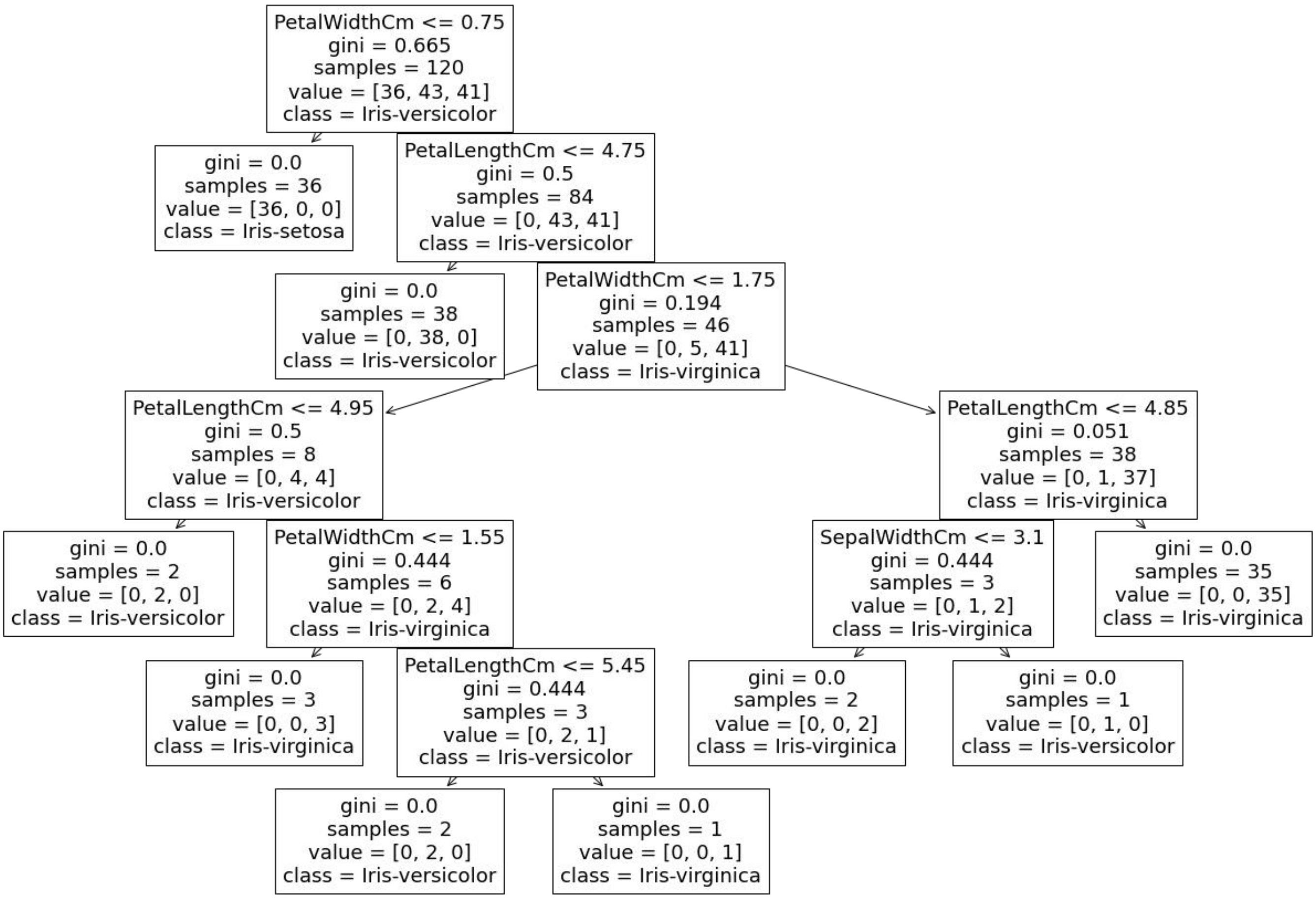
✓ 0.9s Python

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```python
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[:, 2:-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=(22,15))
tree.plot_tree(model, feature_names=X.keys(), class_names=y.unique())
plt.show()
```

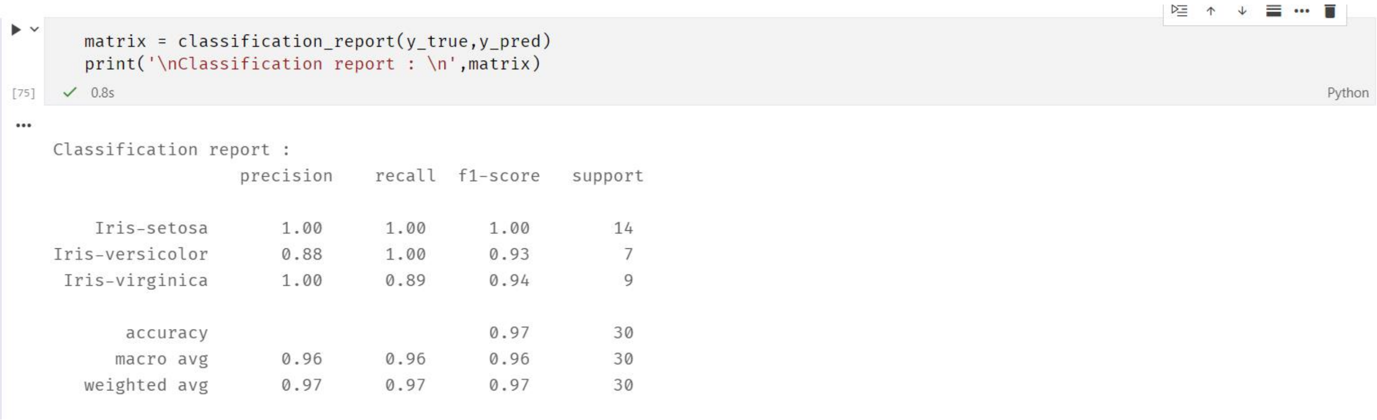
[72] ✓ 1.7s Python



```python
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 = y_test

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

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

[74] ✓ 0.1s Python

```
Accuracy :  0.9666666666666667

Confusion Matrix: 
 [[14  0  0]
 [ 0  7  0]
 [ 0  1  8]]
```



```python
matrix = classification_report(y_true,y_pred)
print('\nClassification report : \n',matrix)
```

```
Classification report :
                  precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        14
Iris-versicolor       0.88      1.00      0.93         7
 Iris-virginica       1.00      0.89      0.94         9

       accuracy                           0.97        30
      macro avg       0.96      0.96      0.96        30
   weighted avg       0.97      0.97      0.97        30
```