Practical

- Shivam Tawari

Aim: Write a python program to evaluate a decision tree on given dataset.

Theory:

Decision Tree Algorithm:

The Decision tree algorithm one such algorithm that is used to solve both regression and classification problems.

A decision tree is supervised machine learning algorithm which looks like an inverted tree where in each node represents a predictor variable, the link between the nodes represents a decision and each leaf node represents an outcome (response variable).

A Decision Tree has the following structure:

① Root Node:
The root node is the starting point of a tree at this point the first split is performed.

② Internal Nodes:
Each internal node represents a decision point (predicator variable) that eventually leads to the prediction of the outcomes.

③ Leaf / Terimal Nodes:
Leaf nodes represent the final class of the outcome and therefore they're also called terminating nodes.

④ Branches:
The branches are connections between nodes, they're represented as arrows. Each branch represents a response such as yes or no.

The Decision Tree algorithm follows:

Step 1: Select the feature that best classifies the data set into the desired classes and assign that features to the root node.

Step 2: Traverse down from the root node, whilst making relevant decisions at each internal node such

that each internal node such each internal node best classifies the data.

Step 3: Route back to step 2 & repeat until you assign a class to the input data.

ID3 Algorithm:

ID3 or the iterative dichotomiser 3 algorithm is one of the most effective algorithm used to build a decision tree.

The ID3 Algorithm follows the below workflow in order to build Decision tree:

1. Select Best Attribute (A)
2. Assign A as a decision variable for the root node.
3. For each value of A, build a descendant of the node.
4. Assign classification labels to the leaf node.
5. If data is correctly classified stop.
6. Else: Iterate over the tree.

Code :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
% matplotlib inline

data = pd.read_csv("petrol_consumption.csv")
data.head()
data.describe()
X = data.drop('Petrol_Consumption', axis = 1)
y = data['petrol_consumption']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split
                    (X, y, test_size = 0.20, random
                                        _state = 0)
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(X_train, y_train)

y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test,
                   'Predicted': y_pred})
df
```

Code & Output:

Practical 7.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
# Shivam Tawari A-58
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]
```
data = pd.read_csv("/content/petrol_consumption.csv")
```

[3]
```
data.head()
```

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consumption
0	9.0	3571	1976	0.525	541
1	9.0	4092	1250	0.572	524
2	9.0	3865	1586	0.580	561
3	7.5	4870	2351	0.529	414
4	8.0	4399	431	0.544	410

[4]
```
data.describe()
```

[4]

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consumption
count	48.000000	48.000000	48.000000	48.000000	48.000000
mean	7.668333	4241.833333	5565.416667	0.570333	576.770833
std	0.950770	573.623768	3491.507166	0.055470	111.885816
min	5.000000	3063.000000	431.000000	0.451000	344.000000
25%	7.000000	3739.000000	3110.250000	0.529750	509.500000
50%	7.500000	4298.000000	4735.500000	0.564500	568.500000
75%	8.125000	4578.750000	7156.000000	0.595250	632.750000
max	10.000000	5342.000000	17782.000000	0.724000	968.000000

[5]
```
X = data.drop('Petrol_Consumption',  axis = 1)
y = data['Petrol_Consumption']
```

[6]
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
```

[7]
```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(X_train, y_train)
```

```
DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None,
                      max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort='deprecated',
                      random_state=None, splitter='best')
```

```
y_pred = regressor.predict(X_test)
```

+ Code + Text

```
[9] df = pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
    df
```

	Actual	Predicted
29	534	547.0
4	410	414.0
26	577	574.0
30	571	554.0
32	577	631.0
37	704	644.0
34	487	648.0

Example 2

```
import pandas
from sklearn import tree
import pydotplus
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import matplotlib.image as pltimg

df = pandas.read_csv("/content/shows.csv")

print(df)
```

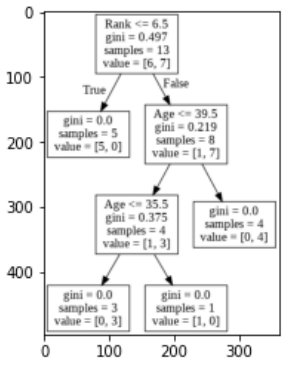
```
    Age  Experience  Rank Nationality   Go
0    36          10     9          UK   NO
1    42          12     4         USA   NO
2    23           4     6           N   NO
3    52           4     4         USA   NO
4    43          21     8         USA  YES
5    44          14     5          UK   NO
6    66           3     7           N  YES
7    35          14     9          UK  YES
8    52          13     7           N  YES
9    35           5     9           N  YES
10   24           3     5         USA   NO
11   18           3     7          UK  YES
12   45           9     9          UK  YES
```

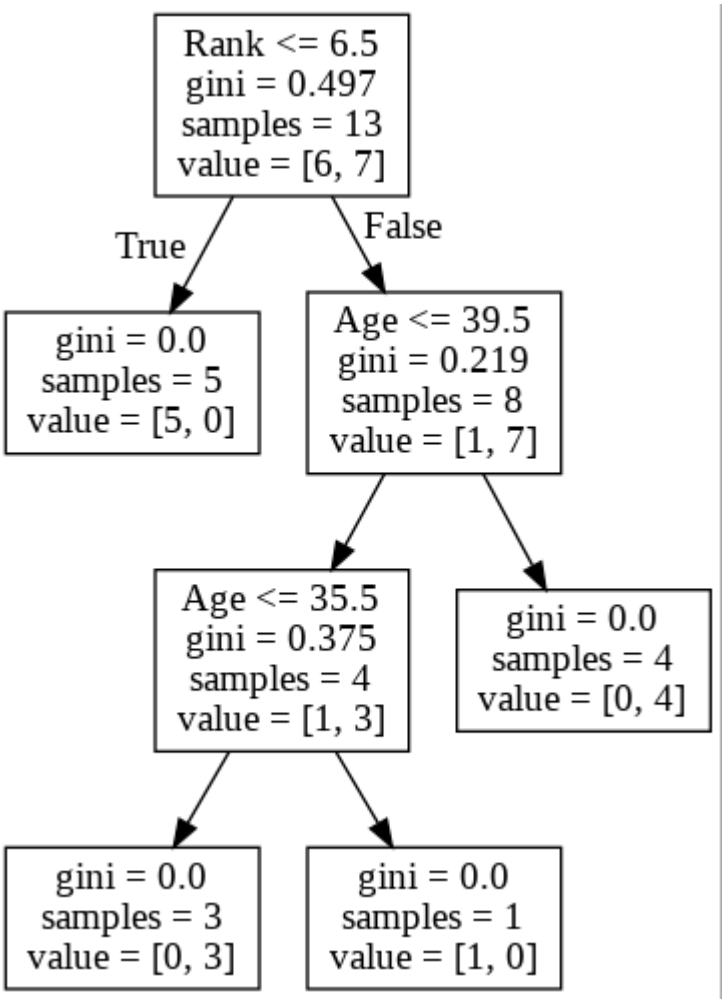
```
[11] d = {'UK': 0, 'USA': 1, 'N': 2}
     df['Nationality'] = df['Nationality'].map(d)
     d = {'YES': 1, 'NO': 0}
     df['Go'] = df['Go'].map(d)

     print(df)
```

```
    Age  Experience  Rank  Nationality  Go
0    36          10     9            0   0
1    42          12     4            1   0
2    23           4     6            2   0
3    52           4     4            1   0
4    43          21     8            1   1
5    44          14     5            0   0
6    66           3     7            2   1
7    35          14     9            0   1
8    52          13     7            2   1
9    35           5     9            2   1
10   24           3     5            1   0
11   18           3     7            0   1
12   45           9     9            0   1
```

```
[12] features = ['Age', 'Experience', 'Rank', 'Nationality']

     X = df[features]
     y = df['Go']

     print(X)
     print(y)
```

```
[13] dtree = DecisionTreeClassifier()
     dtree = dtree.fit(X, y)
     data = tree.export_graphviz(dtree, out_file=None, feature_names=features)
     graph = pydotplus.graph_from_dot_data(data)
     graph.write_png('mydecisiontree.png')

     img=pltimg.imread('mydecisiontree.png')
     imgplot = plt.imshow(img)
     plt.show()
```




Rank <= 6.5
gini = 0.497
samples = 13
value = [6, 7]
True
False
gini = 0.0
samples = 5
value = [5, 0]
Age <= 39.5
gini = 0.219
samples = 8
value = [1, 7]
Age <= 35.5
gini = 0.375
samples = 4
value = [1, 3]
gini = 0.0
samples = 4
value = [0, 4]
gini = 0.0
samples = 3
value = [0, 3]
gini = 0.0
samples = 1
value = [1, 0]