

Introduction

Decision Tree

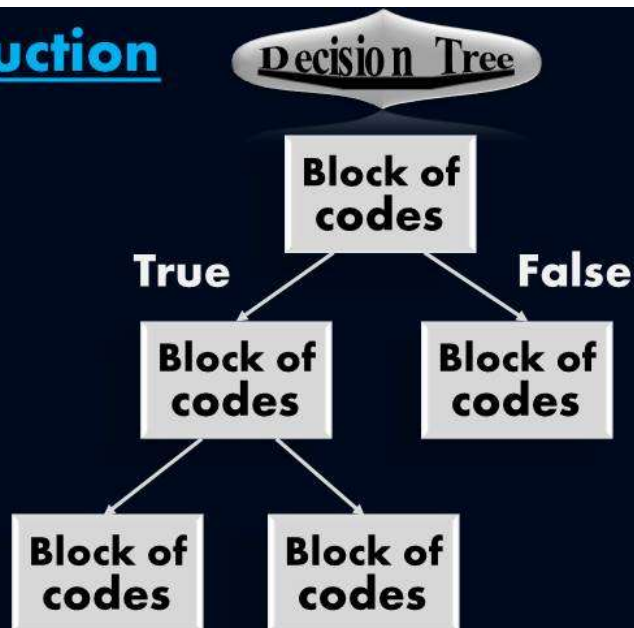
Decision tree is a classifier of machine learning. It's like a family tree, but in decision tree each node can have maximum two children.

Introduction

Decision Tree

Decision tree is helpful when the model has less number of features. The more features the model keeps the more complex it is.

Introduction



Project

Decision Tree

```
#Step1: Import all libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
#Step 2: Load and read the data
d = pd.read_csv('Books.csv')
#Now to explore the data
print(d)
```

Project

Decision Tree

>>

	age	gender	Books
0	18	1	Love story
1	19	1	Love story
2	20	1	Love story
3	27	1	Business
4	29	1	Business
5	31	1	Business
6	36	1	Politics
7	38	1	Politics
8	50	1	Politics
9	17	0	History
10	19	0	History
11	21	0	History
12	26	0	Geography
13	27	0	Geography
14	30	0	Geography
15	37	0	Politics
16	38	0	Politics
17	45	0	Politics

Project

Decision Tree

#Inputs of the data

```
x = d.drop(columns=['Books'])
print(x)
```

>>

	age	gender
0	18	1
1	19	1
2	20	1
3	27	1
4	29	1
5	31	1
6	36	1
7	38	1
8	50	1
9	17	0
10	19	0
11	21	0
12	26	0
13	27	0
14	30	0
15	37	0
16	38	0
17	45	0

Project

Decision Tree

#Outputs of the data

```
y = d['Books']
```

```
print(y)
```

```
>>
```

```
0    Books
1    Love story
2    Love story
3    Love story
4    Business
5    Business
6    Business
7    Politics
8    Politics
9    Politics
10   History
11   History
12   History
13   Geography
14   Geography
15   Politics
16   Politics
17   Politics
```

Project

Decision Tree

#Step3: Clean the data. Already cleaned

#Step4: Split the data in train test

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

#Step5: Create the machine model

```
model = DecisionTreeClassifier()
```

#Step6: Train the machine model

```
model.fit(x_train,y_train)
```

#Step7: Prediction of the machine model

```
prediction = model.predict(x_test)
```

#Step8: Final evaluation of the machine model

```
score = accuracy_score(y_test,prediction)
```

Project

Decision Tree

```
print(score)
>> 1.0
#Run it one more time by pressing ctrl+enter
>> 0.75
```

Project

Decision Tree

```
#Step3: Clean the data. Already cleaned
#Step4: Split the data in train test
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.8)
#Step5: Create the machine model
model = DecisionTressClassifier()
#Step6: Train the machine model
model.fit(x_train,y_train)
#Step7: Prediction of the machine model
prediction = model.predict(x_test)
#Step8: Final evaluation of the machine model
score = accuracy_score(y_test,prediction)
```


Project

Decision Tree

```
print(score)
>> 0.4
#Run it one more time by pressing ctrl+enter
```

Project

Decision Tree

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
d = pd.read_csv('Books.csv')
x = d.drop(columns=['Books'])
y = d['Books']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
model = DecisionTreeClassifier()
model.fit(x, y)
prediction = model.predict([25, 1], [20, 0])
print(prediction)
>> array(['Business', 'History'], dtype=object)
```

COMPLETE CODES ON ONE PAGE

Project

Decision Tree

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
d = pd.read_csv('Books.csv')
x = d.drop(columns=['Books'])
y = d['Books']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
model = DecisionTreeClassifier()
model.fit(x_train,y_train)
prediction = model.predict(x_test)
score = accuracy_score(y_test,prediction)
```