Decision Tree Classifier

CART dataset

Data Loaded

In [52]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [53]:
```
dfpd.read_excel('C:/Users/vaitheeswaran/Downloads/CART (1).xlsx')
```

In [54]:
```
df.head()
```

Out[54]:

	RID	age	income	student	credit_rating	buys_computer
0	1	youth	high	no	fair	no
1	2	youth	high	no	excellent	no
2	3	middle_aged	high	no	fair	yes
3	4	senior	medium	no	fair	yes
4	5	senior	low	yes	fair	yes

Preprocessing the data

In [55]:
```
##Preprocessing
df.isnull().sum()
```

Out[55]:
```
RID              0
age              0
income           0
student          0
credit_rating    0
buys_computer    0
dtype: int64
```

In [56]:
```
df
```

Out[56]:

	RID	age	income	student	credit_rating	buys_computer
0	1	youth	high	no	fair	no
1	2	youth	high	no	excellent	no
2	3	middle_aged	high	no	fair	yes
3	4	senior	medium	no	fair	yes
4	5	senior	low	yes	fair	yes
5	6	senior	low	yes	excellent	no
6	7	middle_aged	low	yes	excellent	yes
7	8	youth	medium	no	fair	no
8	9	youth	low	yes	fair	yes
9	10	senior	medium	yes	fair	yes
10	11	youth	medium	yes	excellent	yes
11	12	middle_aged	medium	no	excellent	yes
12	13	middle_aged	high	yes	fair	yes
13	14	senior	medium	no	excellent	no

In [57]:
```
import sklearn
from sklearn.preprocessing import LabelEncoder
```

In [58]:
```
l_age=LabelEncoder()
l_income=LabelEncoder()
l_student=LabelEncoder()
l_credit_rating=LabelEncoder()
l_buys_computer=LabelEncoder()
```

In [59]:
```
df['ag']=l_age.fit_transform(df['age'])
df['inco']=l_income.fit_transform(df['income'])
df['stu']=l_student.fit_transform(df['student'])
df['cr']=l_credit_rating.fit_transform(df['credit_rating'])
df['bc']=l_buys_computer.fit_transform(df['buys_computer'])
```

In [60]:
```
df
```

Out[60]:

	RID	age	income	student	credit_rating	buys_computer	ag	inco	stu	cr	bc
0	1	youth	high	no	fair	no	2	0	0	1	0
1	2	youth	high	no	excellent	no	2	0	0	0	0
2	3	middle_aged	high	no	fair	yes	0	0	0	1	1
3	4	senior	medium	no	fair	yes	1	2	0	1	1
4	5	senior	low	yes	fair	yes	1	1	1	1	1
5	6	senior	low	yes	excellent	no	1	1	1	0	0
6	7	middle_aged	low	yes	excellent	yes	0	1	1	0	1
7	8	youth	medium	no	fair	no	2	2	0	1	0
8	9	youth	low	yes	fair	yes	2	1	1	1	1
9	10	senior	medium	yes	fair	yes	1	2	1	1	1
10	11	youth	medium	yes	excellent	yes	2	2	1	0	1
11	12	middle_aged	medium	no	excellent	yes	0	2	0	0	1
12	13	middle_aged	high	yes	fair	yes	0	0	1	1	1
13	14	senior	medium	no	excellent	no	1	2	0	0	0

In [61]:
```
df1=df.drop(['RID','age','income','student','credit_rating','buys_computer'],axis='columns')
```

In [62]:
```
df1
```

Out[62]:

	ag	inco	stu	cr	bc
0	2	0	0	1	0
1	2	0	0	0	0
2	0	0	0	1	1
3	1	2	0	1	1
4	1	1	1	1	1
5	1	1	1	0	0
6	0	1	1	0	1
7	2	2	0	1	0
8	2	1	1	1	1
9	1	2	1	1	1
10	2	2	1	0	1
11	0	2	0	0	1
12	0	0	1	1	1
13	1	2	0	0	0

Training and Testing data

In [64]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=42)
```

In [65]:
```
from sklearn.tree import DecisionTreeClassifier
DTC=DecisionTreeClassifier()
DTC.fit(x_train,y_train)
```

Out[65]:
```
DecisionTreeClassifier()
```

In [66]:
```
df1.corr()
```

Out[66]:

	ag	inco	stu	cr	bc
ag	1.000000	0.092336	-8.944272e-02	7.745967e-02	-0.493333
inco	0.092336	1.000000	0.000000e+00	-1.992295e-01	0.127027
stu	-0.089443	0.000000	1.000000e+00	-6.409879e-17	0.447214
cr	0.077460	-0.198030	-6.409879e-17	1.000000e+00	0.258199
bc	-0.493333	0.127027	4.472136e-01	2.581989e-01	1.000000

In [67]:
```
df1.cov()
```

Out[67]:

	ag	inco	stu	cr	bc
ag	0.666613	0.065934	-0.039462	0.032967	-0.202297
inco	0.065934	0.747253	0.000000	-0.087912	0.054945
stu	-0.039462	0.000000	0.250231	0.000000	0.115385
cr	0.032967	-0.087912	0.000000	0.263736	0.065934
bc	-0.202297	0.054945	0.115385	0.065934	0.247253

In [68]:
```
sns.heatmap(df1.corr(),annot=True)
```

Out[68]:
```
<AxesSubplot:>
```

In [69]:
```
sns.heatmap(df.cov(),annot=True)
```

Out[69]:
```
<AxesSubplot:>
```


Exploratory Data Analysis

In [70]:
```
sns.pairplot(df1)
```

Out[70]:
```
<seaborn.axisgrid.PairGrid at 0x288e631ac70>
```


In [71]:
```
import six
from six import StringIO
from sklearn.tree import export_graphviz
import pydotplus
import graphviz
from sklearn import tree
```

In [72]:
```
dot_data =StringIO()
dot_data=tree.export_graphviz(DTC,
                              out_file=None,filled=True,rounded=True,
                              feature_names=feature_names,class_names=['0','1'])

graph=graphviz.Source(dot_data)
```

In [73]:
```
graph
```

Out[73]:

In [74]:
```
ypred=DTC.predict(x_test)
```

Evaluation

In [75]:
```
from sklearn.metrics import classification_report
```

In [76]:
```
classification_report(y_test,ypred)
```

Out[76]:
```
'              precision    recall  f1-score   support\n\n           0       0.50      1.00      0.67         1\n           1       1.00      0.67      0.80         3\n\n    accuracy                           0.75         4\n   macro avg       0.75      0.83      0.73         4\nweighted avg       0.88      0.75      0.77         4\n'
```

In [77]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
```

In [78]:
```
print("Confusion Matrix:",confusion_matrix(y_test,ypred))
```

Out[78]:
```
Confusion Matrix: [[1 0]
 [1 2]]
```

In [79]:
```
print("Accuracy:",accuracy_score(y_test,ypred))
```

Out[79]:
```
Accuracy: 0.75
```

In [ ]:

In [ ]: