IIT ROORKEE

FREE ONLINE EDUCATION
swayam

NPTEL

NPTEL ONLINE
CERTIFICATION COURSE

Classification and Regression Trees (CART – III)

Dr A. RAMESH

DEPARTMENT OF MANAGEMENT STUDIES

Agenda

Python demo for CART model -

- Visualizing Decision Tree
- Interpretation of CART model

Example

Problem Description-

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

Han, J., Pei, J. and Kamber, M., 2011. *Data mining: concepts and techniques*. Elsevier.

Import Relevant Libraries and Loading Data File

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```python
data = pd.read_excel('CART.xlsx')
```

In [3]:
```python
data
```

Out[3]:

	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

Methods used in Data Encoding

- **LabelEncoder ():** This method is used to normalize labels. It can also be used to transform non-numerical labels to numerical labels.

- **Fit_transform ():** This method is used for Fitting label encoder and return encoded labels.

Data Encoding Procedure

In [4]:
```
import sklearn
from sklearn.preprocessing import LabelEncoder
```

In [5]:
```
le_age = LabelEncoder()
le_income = LabelEncoder()
le_student = LabelEncoder()
le_credit_rating = LabelEncoder()
le_buys_computer = LabelEncoder()
```

In [6]:
```
data['age_n'] = le_age.fit_transform(data['age'])
data['income_n'] = le_income.fit_transform(data['income'])
data['student_n'] = le_student.fit_transform(data['student'])
data['credit_rating_n'] = le_credit_rating.fit_transform(data['credit_rating'])
data['buys_computer_n'] = le_credit_rating.fit_transform(data['buys_computer'])
```

Data Encoding

In [7]:
```
data.head()
```

Out[7]:

	RID	age	income	student	credit_rating	buys_computer	age_n	income_n	student_n	credit_rating_n	buys_computer_n
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

Structuring Dataframe

drop(): This is used to **Remove** rows or columns by specifying label names and corresponding axis or by specifying directly index or **column** names.

In [8]:
```
data_new = data.drop(['age','income','student','credit_rating','buys_computer'], axis='columns')
data_new.head()
```

Out[8]:

	RID	age_n	income_n	student_n	credit_rating_n	buys_computer_n
0	1	2	0	0	1	0
1	2	2	0	0	0	0
2	3	0	0	0	1	1
3	4	1	2	0	1	1
4	5	1	1	1	1	1

Independent and Dependent Variables Selection

In [9]:
```
feature_cols = ['age_n', 'income_n', 'student_n', 'credit_rating_n']
x = data_new.drop(['buys_computer_n','RID'], axis='columns') #input
y = data_new['buys_computer_n'] #target
```

In [10]:
```
x.head()
```

Out[10]:

	age_n	income_n	student_n	credit_rating_n
0	2	0	0	1
1	2	0	0	0
2	0	0	0	1
3	1	2	0	1
4	1	1	1	1

In [11]:
```
y.head()
```

Out[11]:
```
0    0
1    0
2    1
3    1
4    1
Name: buys_computer_n, dtype: int32
```

Build the Decision Tree Model without Splitting

In [12]:
```python
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
dt = clf.fit(x,y)
dt
```

Out[12]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', 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=False, random_state=None,
            splitter='best')
```

Visualizing Decision Tree

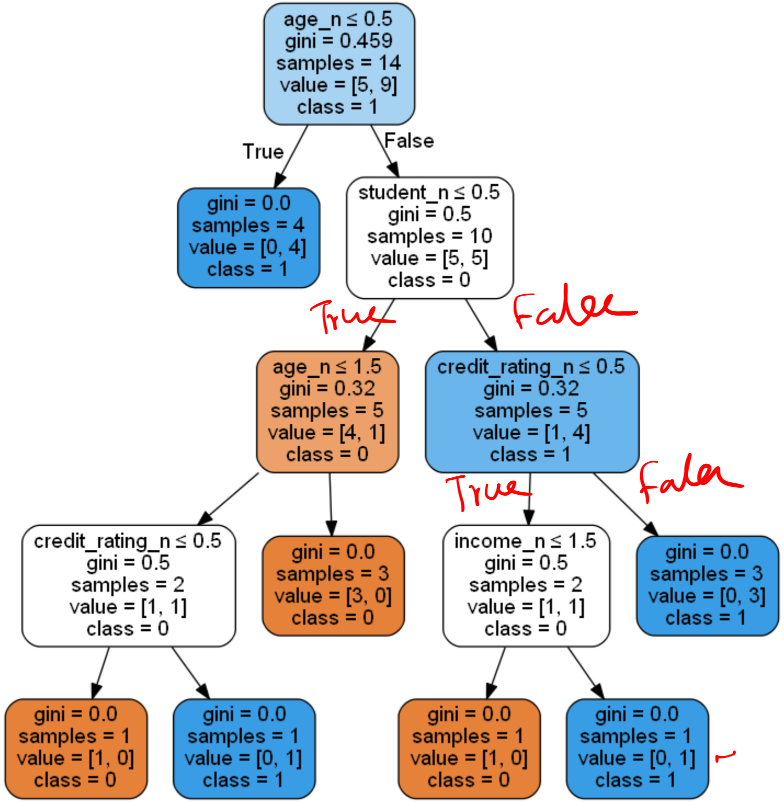
In [13]:
```python
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus
```

In [14]:
```python
dot_data = StringIO()
export_graphviz(dt, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('buys_computer.png')
```

Out[14]: True

In [15]:
```python
Image(graph.create_png())
```

Decision Tree Visualization



Interpretation of the CART Output

Calculation of Gini(D)

- We first use the following Equation for Gini index to compute the impurity of D:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

$$= \quad Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459.$$

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

Income Attribute

- Low, Medium, High
- Option 1: {Low, Medium}, {High}
- Option 2 : {High, Medium}, {low}
- Option 3 : {High, Low}, {Medium}

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

Tuples in partition D1

Low, medium
D1

High
D2

- Low + Medium:

Low + Medium	Class: buys computer
Yes	3+4 =7
No	1+ 2 = 3

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

Tuples in partition D2

- High :

High	Class: buys computer
Yes	2
No	2

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

Gini index for income attribute

- The Gini index value computed based on this partitioning is

$$Gini_{income \in \{low,medium\}}(D) = \frac{10}{14}Gini(D_1) + \frac{4}{14}Gini(D_2)$$

$= (10/14)\,(1- (7/10)^2 – (3/10)^2) +$

$(4/14)\,(1- (2/4)^2 – (2/4)^2)$

$= 0.443 = \text{Gini}_{\text{income} \in \{\text{high}\}}$

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

Gini index for income attribute

- The Gini index value computed based on this partitioning is

h, m   L

D_1 D_2

Gini $_{\text{income} \in \{\text{high, medium}\}}$

$$= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$

$= (10/14)\ (1 - (6/10)^2 - (4/10)^2) +$

$(4/14)\ (1 - (3/4)^2 - (1/4)^2)$

$= 0.45 =$ Gini $_{\text{income} \in \{\text{low}\}}$

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

Gini index for income attribute

h, L

m

- The Gini index value computed based on this partitioning is

$$\text{Gini}_{\text{income} \in \{\text{high, low}\}} = (8/14)\,(1 - (5/8)^2 - (3/8)^2) + (6/14)\,(1 - (2/6)^2 - (4/6)^2)$$

$$= 0.458 = \text{Gini}_{\text{income} \in \{\text{medium}\}}$$

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

Gini index for income attribute

- $\boxed{\text{Gini}_{\text{income} \in \{\text{low, medium}\}} = 0.443 = \text{Gini}_{\text{income} \in \{\text{high}\}}}$
- $\text{Gini}_{\text{income} \in \{\text{high, medium}\}} = 0.45 = \text{Gini}_{\text{income} \in \{\text{low}\}}$
- $\text{Gini}_{\text{income} \in \{\text{high, low}\}} = 0.458 = \text{Gini}_{\text{income} \in \{\text{medium}\}}$

Gini index for Age attribute

- The Gini index value computed based on this partitioning is

$$\text{Gini}_{\text{Age} \in \{\text{Youth, middle\_aged}\}} = 0.457 = \text{Gini}_{\text{Age} \in \{\text{senior}\}}$$

$$\boxed{\text{Gini}_{\text{Age} \in \{\text{Youth, Senior}\}} = 0.357 = \text{Gini}_{\text{Age} \in \{\text{middle\_aged}\}}}$$

$$\text{Gini}_{\text{Age} \in \{\text{senior, middle\_aged}\}} = 0.393 = \text{Gini}_{\text{Age} \in \{\text{Youth}\}}$$

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

Gini index for student attribute

- The Gini index value computed based on this partitioning is

$$Gini_{student \in \{Yes, No\}}$$

$$= 7/14\ (1-(6/7)^2 - (1/7)^2) + 7/14\ (1-(3/7)^2 - (4/7)^2)$$

$$= 0.367$$

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

Gini index for credit_rating attribute

- The Gini index value computed based on this partitioning is

$$Gini_{credit\ rating \in \{fair,\ Excellent\}}$$

$$= 8/14\ (1-(6/8)^2-(2/8)^2\ ) +$$

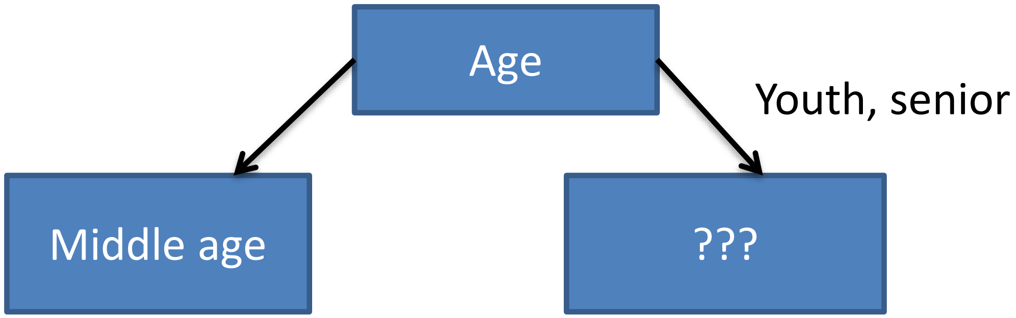
$$6/14\ (1-(3/6)^2-(3/6)^2\ )$$

$$= 0.428$$

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

Choosing the root node

The attribute with minimum Gini score will be taken, i.e. Age (Gini $_{Age \in \{Youth,\ Senior\}}$ = 0.357 = Gini $_{Age \in \{middle\_aged\}}$)



Attribute	Gini score
Age	0.357
Income	0.443
Student	0.367
Credit_rating	0.428

Gini index for different attributes for sample of 10

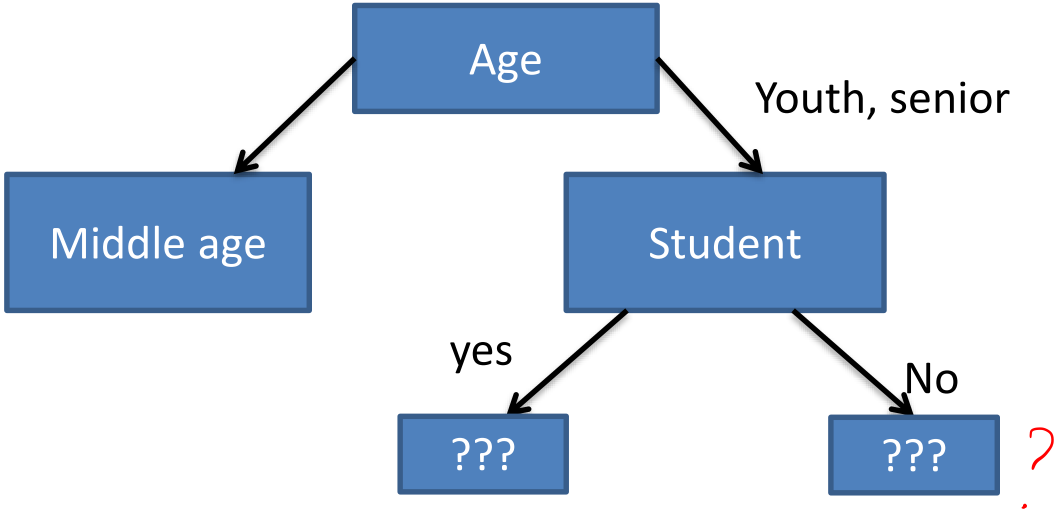
- After separating 4 samples belonging middle age, total 10 are remaining:

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

Gini index for different attributes for sample of 10

- Gini (D) = $(1- (5/10)^2 – (5/10)^2)) = 0.5$
- $\text{Gini}_{\text{Age}} = 0.48$
- $\text{Gini}_{\text{Credit Rating}} = 0.41$
- $\boxed{\text{Gini}_{\text{Student}} = 0.32}$
- $\text{Gini}_{\text{income}} = 0.375$
- Take student as node as it have mini. Gini Score

Drawing cart



For branch Student = No

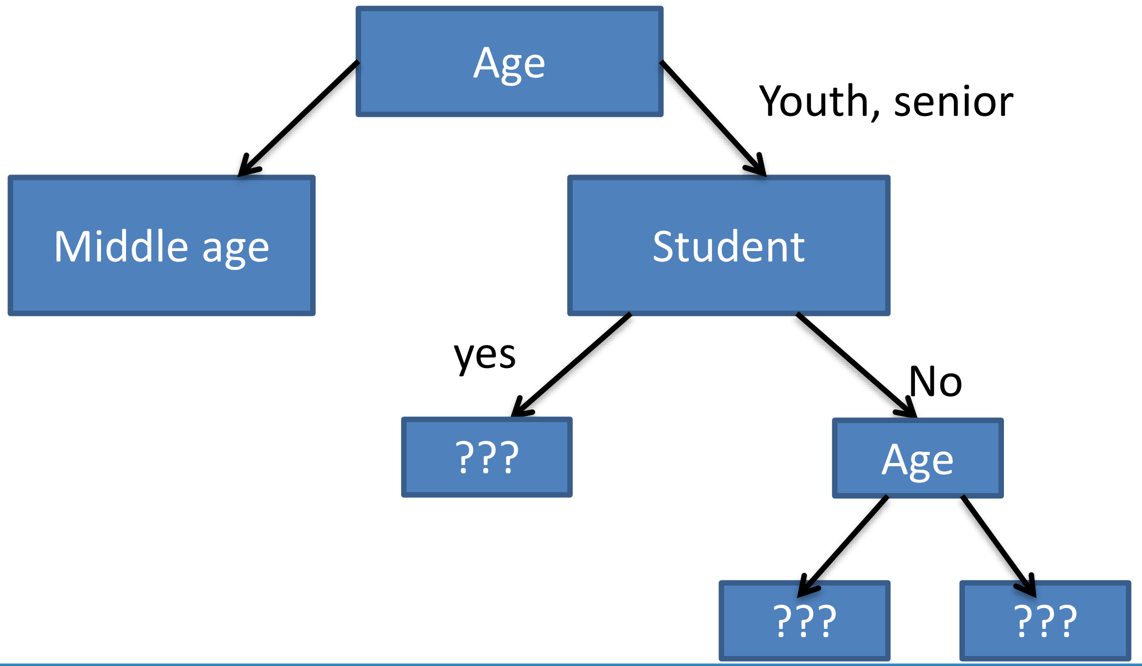
- Omit the marked rows (Data entry), either belonging Age = middle_aged or student = Yes
- Total 5 rows are remaining

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

Gini index for different attributes For branch Student = No

- Gini (D) = $(1- (4/5)^2 – (1/5)^2)) = 0.32$
- $\boxed{\text{Gini}_{Age} = 0.2}$
- $\text{Gini}_{Credit\ Rating} = 0.267$
- $\text{Gini}_{Student} = 0.32$
- $\text{Gini}_{income} = 0.267$
- Take age as node as it have mini. Gini Score

Drawing cart



For branch Student = Yes

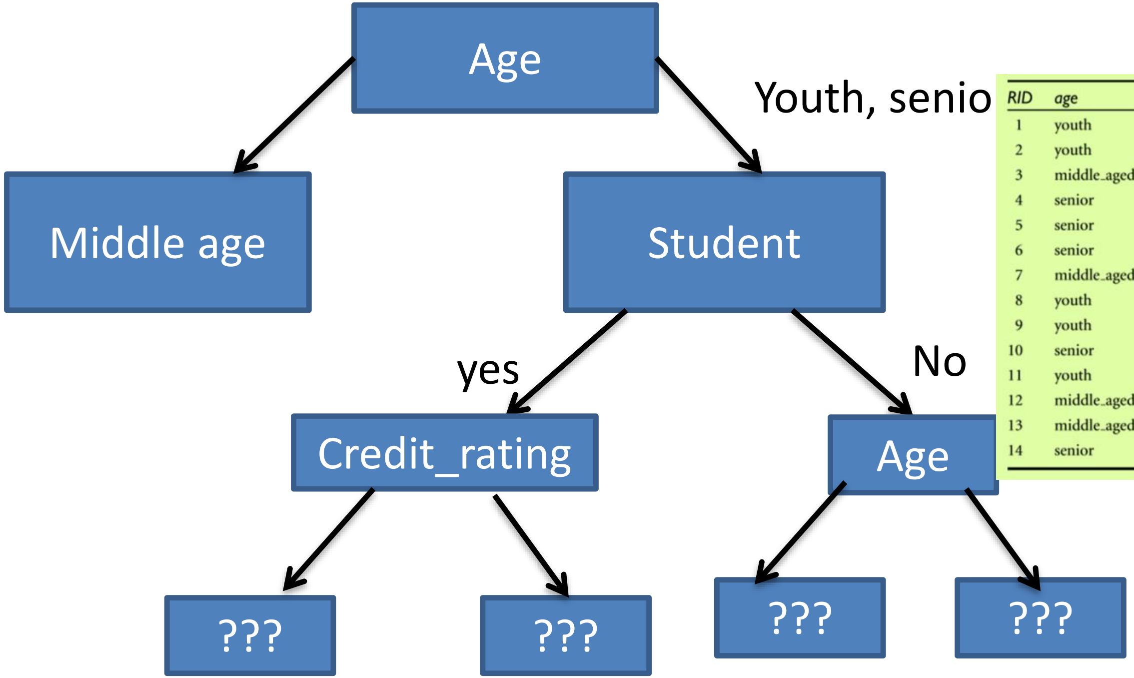
- Omit the marked rows (Data entry), either belonging Age = middle_aged or student = No
- Total 5 rows are remaining

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

Gini index for different attributes For branch Student = No

- Gini (D) = $(1- (4/5)^2 – (1/5)^2)) = 0.32$
- $Gini_{Age} = 0.267$
- $\boxed{Gini_{Credit\ Rating} = 0.2}$
- $Gini_{\ Student} = 0.32$
- $Gini_{\ income} = 0.267$
- Take credit rating as node as it have mini. Gini Score

Drawing cart



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

Coding scheme

Age	Code
Youth	2
Middle Age	0
senior	1

Student	Code
Yes	1
No	0

$n \leq 0.5$

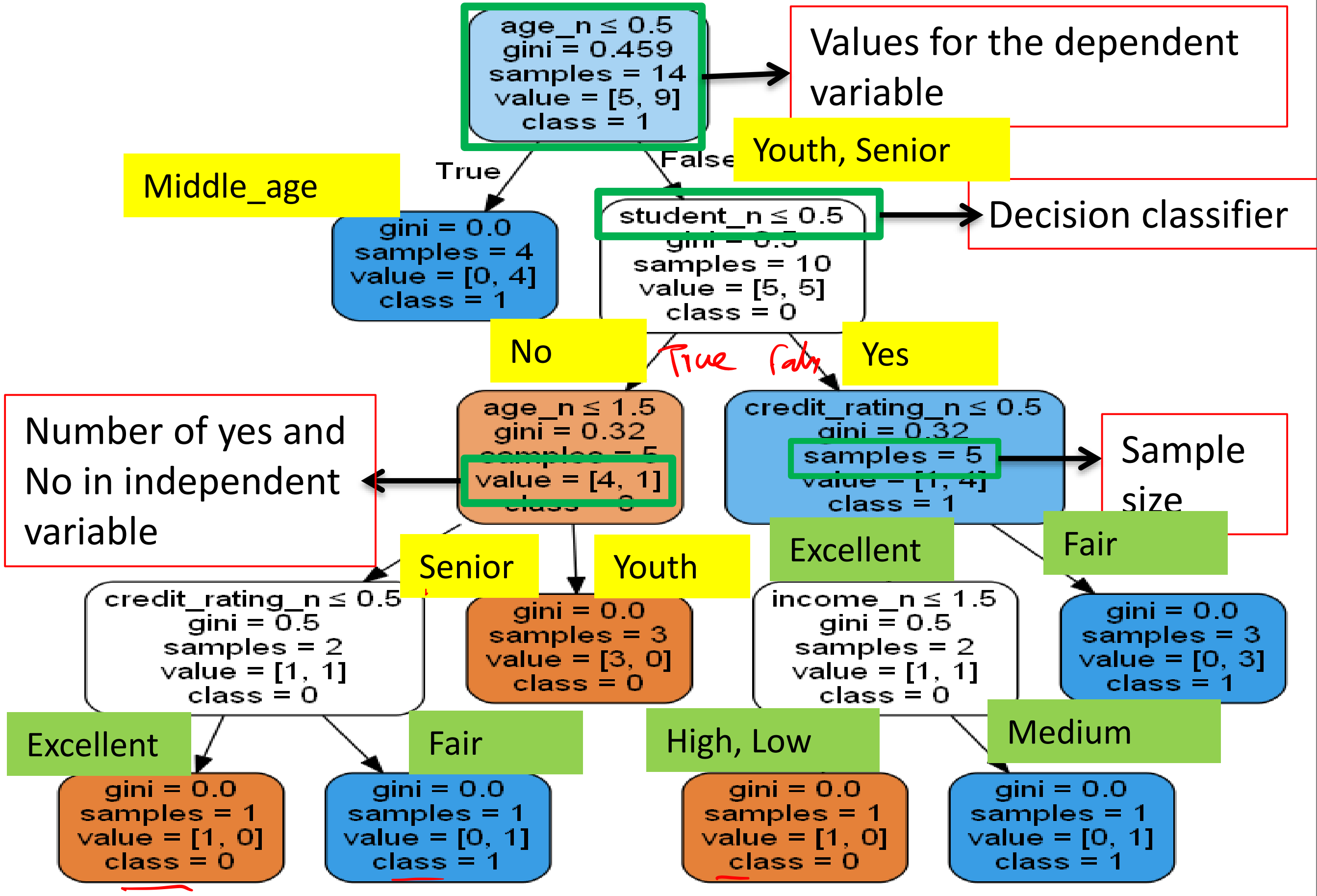
Income	Code
High	0
Low	1
Medium	2

Credit rating	Code
Fair	1
Excellent	0

Buys computer	Class
Yes	1
No	0

Decision tree

- Repeat the splitting process until we obtain all the leaf nodes, the final out - put:



Splitting Dataset

- Train_test_split(): This method is used for splitting dataset into training and testing data subsets.

```
In [12]:  from sklearn.model_selection import train_test_split

In [13]:  x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
```

Build the Decision Tree Model

In [14]:
```python
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
dt = clf.fit(x_train,y_train)
dt
```

Out[14]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', 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=False, random_state=None,
            splitter='best')
```

Evaluating the Model

In [16]:
```python
from sklearn import metrics
```

In [17]:
```python
y_pred = clf.predict(x_test)
```

In [18]:
```python
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.75
```

```
True:  [1 1 0 1]
pred:  [1 0 0 1]
```

Visualizing Decision Tree

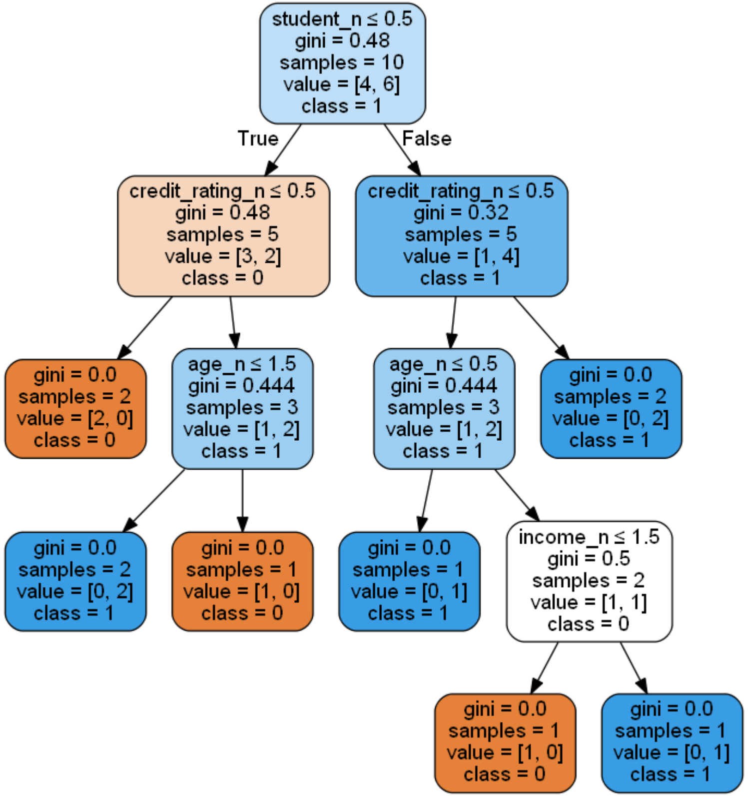
In [19]:
```
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus
```

In [20]:
```
dot_data = StringIO()
export_graphviz(dt, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('buys_computer.png')
```

Out[20]: True

In [21]:
```
Image(graph.create_png())
```

Decision Tree Visualization



Thank You