Practical on Decision Tree

Toy Dataset: The problem and solution with a toy dataset are already well known.

This is a dataset that is available as part of the ML package itself such scikit-learn.
It is used to study a classification model such that it has a small size and has been used so extensively that might not have the case based relevance.

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

Iris Dataset For Studying Classification

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

Feature Variable and Target Variable

Feature Variable

Also known as:

* Independent Variables

* Knowns

* Attributes

Target Variable

Also known as:

* Dependent variable

* Unknown

* Label

A feature is something we have and know that we study in our research.

Aim of the research is to understand the target.

A decision tree (a supervised machine learning algorithm) uses historical data to learn patterns and uncover relationships between the features of your dataset and the target.

Problem

Question:

You are asked what is the rough estimate of the weight of the person whose height, age, gender you know.

What kind of problem is this: classification, clustering, or regression?

What will be your features and what will be your target?

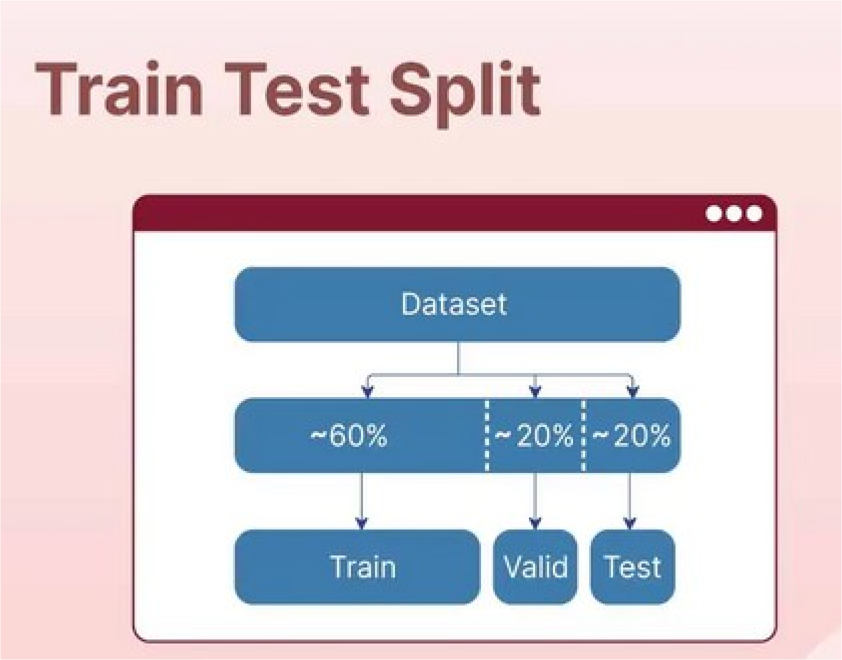
Answer

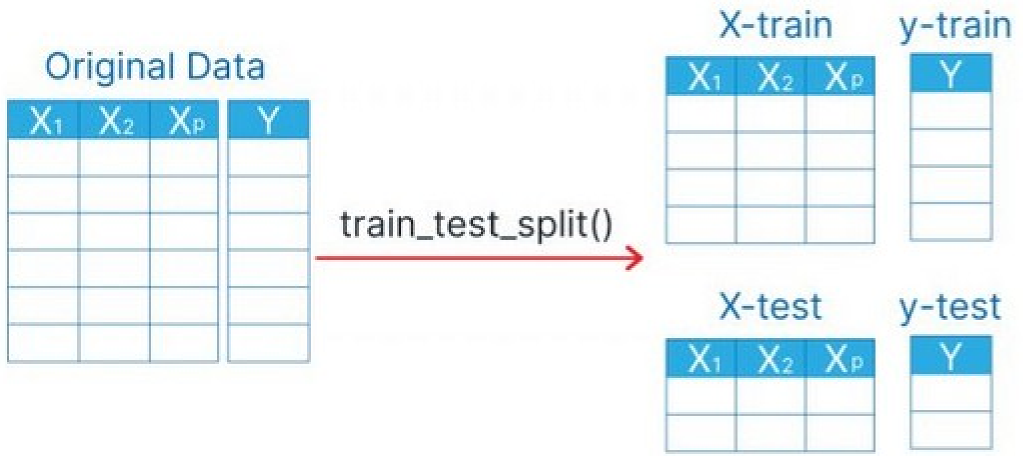
Answer:

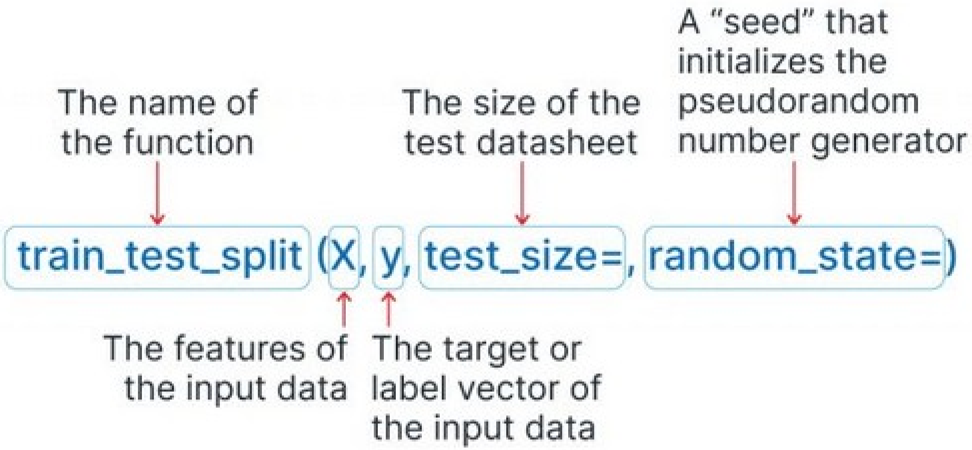
The problem is of regression.
Estimation of a real-valued valued number (weight).

Weight: target / dependent variable
Height, age, gender: are your features.

Splitting the data into train set and test set

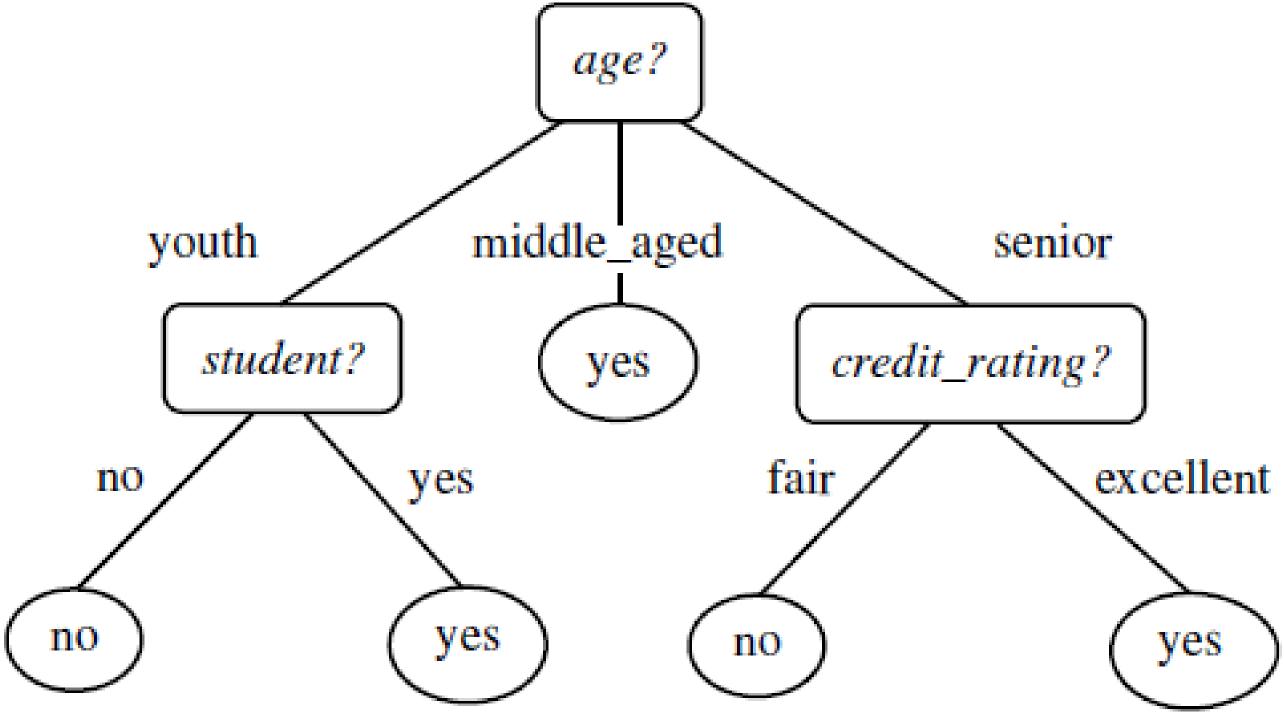



Original Data
X1
X2
Xp
Y
train_test_split()
X-train
y-train
X-test
y-test


The name of the function
The size of the test datasheet
A "seed" that initializes the pseudorandom number generator
train_test_split (X, y, test_size=, random_state=)
The features of the input data
The target or label vector of the input data

In a decision tree, we make decisions based on the attributes we study at the nodes (except leaf nodes).
As we move down from the root node (from top) to the bottom, we keep evaluting features (positioned at the nodes) for their values and decide whether we would like to go left or at the right down the tree.

At leaf nodes, we get to know what will be value of our target variable.



Attribute Selection Measures

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

Turn to slide 10: for solved problem

The Gini index measures the impurity of D, a data partition or set of training tuples

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

where p_i is the probability that a tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. The sum is computed over m classes.

Attribute Selection Measures

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

if a binary split on A partitions D into D_1 and D_2,

gini index of D given that partitioning is

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

The Gini index measures the impurity of D, a data partition or set of training tuples

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

where p_i is the probability that a tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. The sum is computed over m classes.

Gini Index

Gini index measures the impurity of data, D.

Formula: Gini(D) = 1 – sum(p(i) ** 2)

Probability of event, E = # favorable outcomes / # sample space

	buys_computer
1	no
2	no
3	yes
4	yes
5	yes
6	no
7	yes
8	no
9	yes
10	yes
11	yes
12	yes
13	yes
14	no

How many classes do we have?
Two classes.

p(no) = 5/14
p(yes) = 9/14

gini(D) = 1 – sum(p(i) ** 2)
= 1 - ((5/14) ** 2 + (9/14) ** 2)

= 0.4591836734693877

attr
alpha
alpha
alpha
alpha
alpha
alpha
alpha
alpha
alpha
alpha

Gini impurity is 0 when column contains single value throughout.

Gini = 1 – sum(p(i) ** 2)

p(i == alpha) = # alpha / # total = 1

Gini = 1 – 1 = 0

attr
alpha
alpha
alpha
alpha
alpha
alpha
alpha
alpha
alpha
beta

In next couple slides, we will be introducing impurity in the otherwise ‘alpha’ valued column and see the gini impurity coefficient increase.

Gini = 1 – sum(p(i) ** 2)

```
>>> 1- ((9/10)**2 + (1/10)**2 )
0.17999999999999994
```

attr
alpha
alpha
alpha
alpha
alpha
alpha
alpha
alpha
beta
beta

Gini = 1 – sum(p(i) ** 2)

```
>>> 1- ((8/10)**2 + (2/10)**2 )
0.319999
```

attr
alpha
alpha
alpha
alpha
alpha
alpha
alpha
beta
beta
beta

Gini = 1 – sum(p(i) ** 2)

```
>>> 1- ((7/10)**2 + (3/10)**2 )
0.42000000
```

attr
alpha
alpha
alpha
alpha
alpha
alpha
alpha
beta
beta
gamma

Gini = 1 – sum(p(i) ** 2)

```
>>> 1- ((7/10)**2 + (2/10)**2 + (1/10)**2)
0.460000000000
```

attr
alpha
alpha
alpha
alpha
alpha
beta
beta
beta
beta
beta

Gini = 1 – sum(p(i) ** 2)

```
>>> 1- ((5/10)**2 + (5/10)**2)
0.5
```

attr
alpha
beta
gamma
delta
epsilon
zeta
eta
theta
iota
kappa

Gini = 1 – sum(p(i) ** 2)

```
>>> 1- ((1/10)**2 * 10)
0.9
```

If we want to split on age: what will be the gini index?

Gini(split Data on Attribute age) = sum(p(age == ?) * Gini for Data(age == ?))

= p(age == youth) * Gini for Data(attr == youth)
+ p(age == middle_aged) * Gini for Data(attr == middle_aged)
+ p(age == seniod) * Gini for Data(attr == senior)

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

If we want to split on student: what will be the gini index?

Gini(split Data on Attribute attr) = sum(p(Data(attr)) * Gini (Data(attr)))

= p(Data(student == yes)) * Gini(Data(student == yes))
+ p(Data(student == no)) * Gini(Data(student == no))

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

	A	B
1	**age**	**buys_computer**
2	youth	no
3	youth	no
4	middle_aged	yes
5	senior	yes
6	senior	yes
7	senior	no
8	middle_aged	yes
9	youth	no
10	youth	yes
11	senior	yes
12	youth	yes
13	middle_aged	yes
14	middle_aged	yes
15	senior	no

	attr	#	p(attr)	p(no|attr)	p(yes|attr)	gini
0	youth	5	0.357143	0.6	0.4	0.48
1	middle_aged	4	0.285714	0.0	1.0	0.0
2	senior	5	0.357143	0.4	0.6	0.48

For records where age == youth:
These are yellow records.

p(yes | age == youth) = # yes / (# yes + # no)

yes | age == youth = 2
no | age == youth = 3

yes / # total = 2 / 5 = 0.4

no / # total = 3 / 5 = 0.6

Gini (age == youth) = ?

yes = 2
no = 3

Gini (age == youth) = 1 - ((2/5) ** 2 + (3/5) ** 2) = 0.48

	A	B
1	**age**	**buys_computer**
2	youth	no
3	youth	no
4	middle_aged	yes
5	senior	yes
6	senior	yes
7	senior	no
8	middle_aged	yes
9	youth	no
10	youth	yes
11	senior	yes
12	youth	yes
13	middle_aged	yes
14	middle_aged	yes
15	senior	no

For records where age == middle_aged:
These are orange records.

p(yes | age == middle_aged) = # yes / # total

yes | age == middle_aged = 4
no | age == middle_aged = 0

yes / # total = 4 / 4 = 1

no / # total = 0 / 4 = 0

Gini impurity is 0 when column contains single value throughout.

Algorithm for building a decision tree:

Step 1: Find overall gini coefficient. If the gini is zero. Stop the branching at this node.

Step 2: Find the attribute with which lowest gini impurity is left.

Step 3: Split on that attribute

Step 4: Repeat steps 1 to 3 to create more branches on splitted data.

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

RID	age	income	student	credit_rating	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

1: age == middle_aged or not

2: If true: buys computer

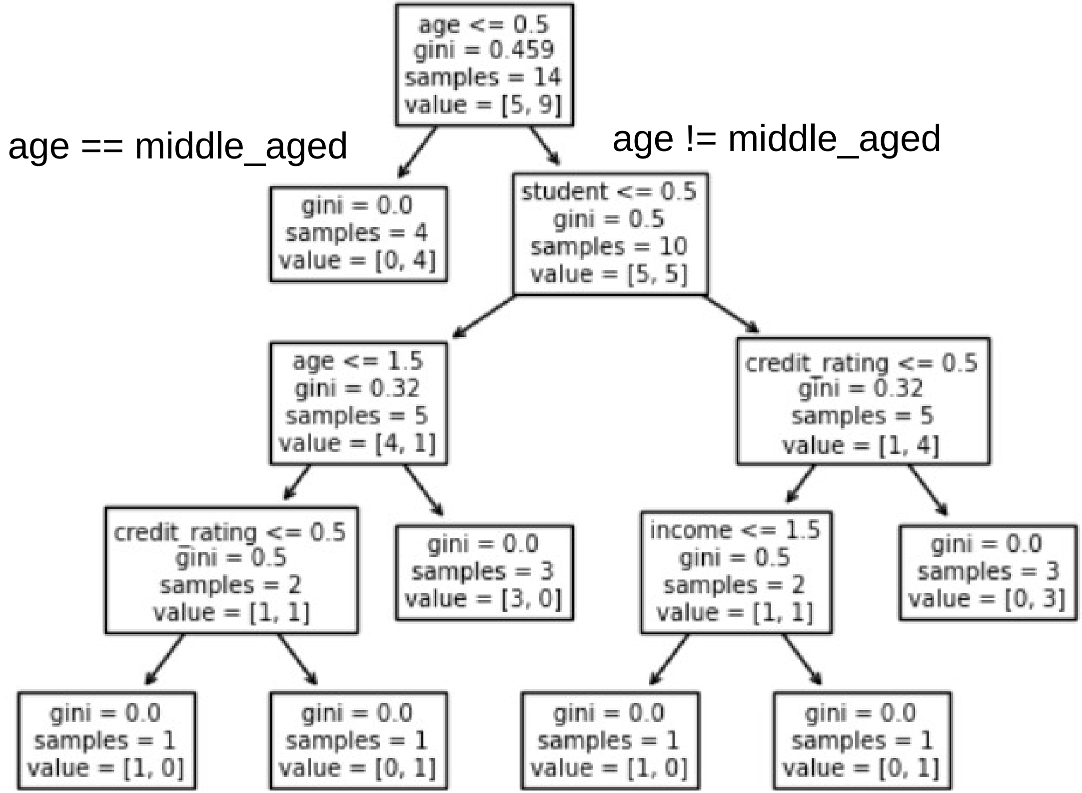
3: If false. Split.
Student == ‘no’: True or False.

4: If true: split on age.
Age == senior or not

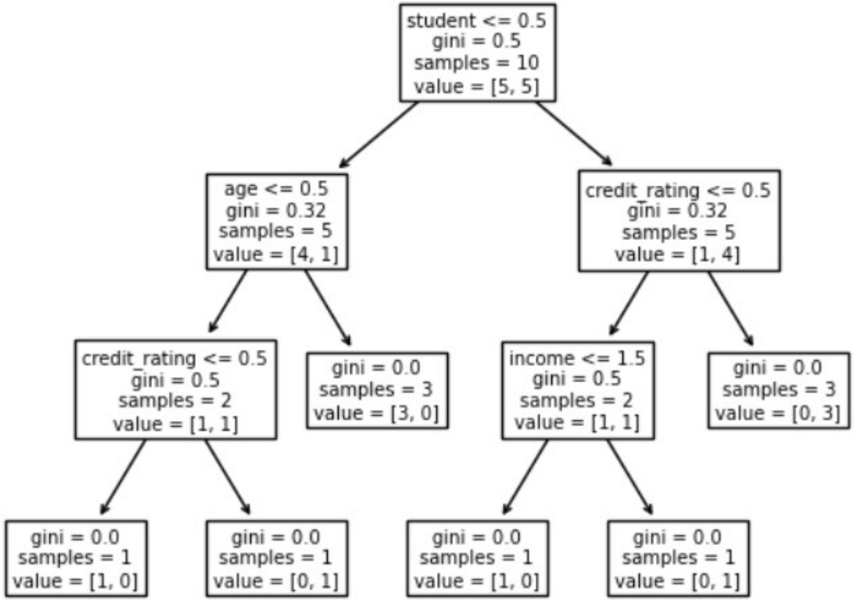
5: If false:
Condition: not a student
And (not middle_aged and not senior)

Result: does not buy a computer

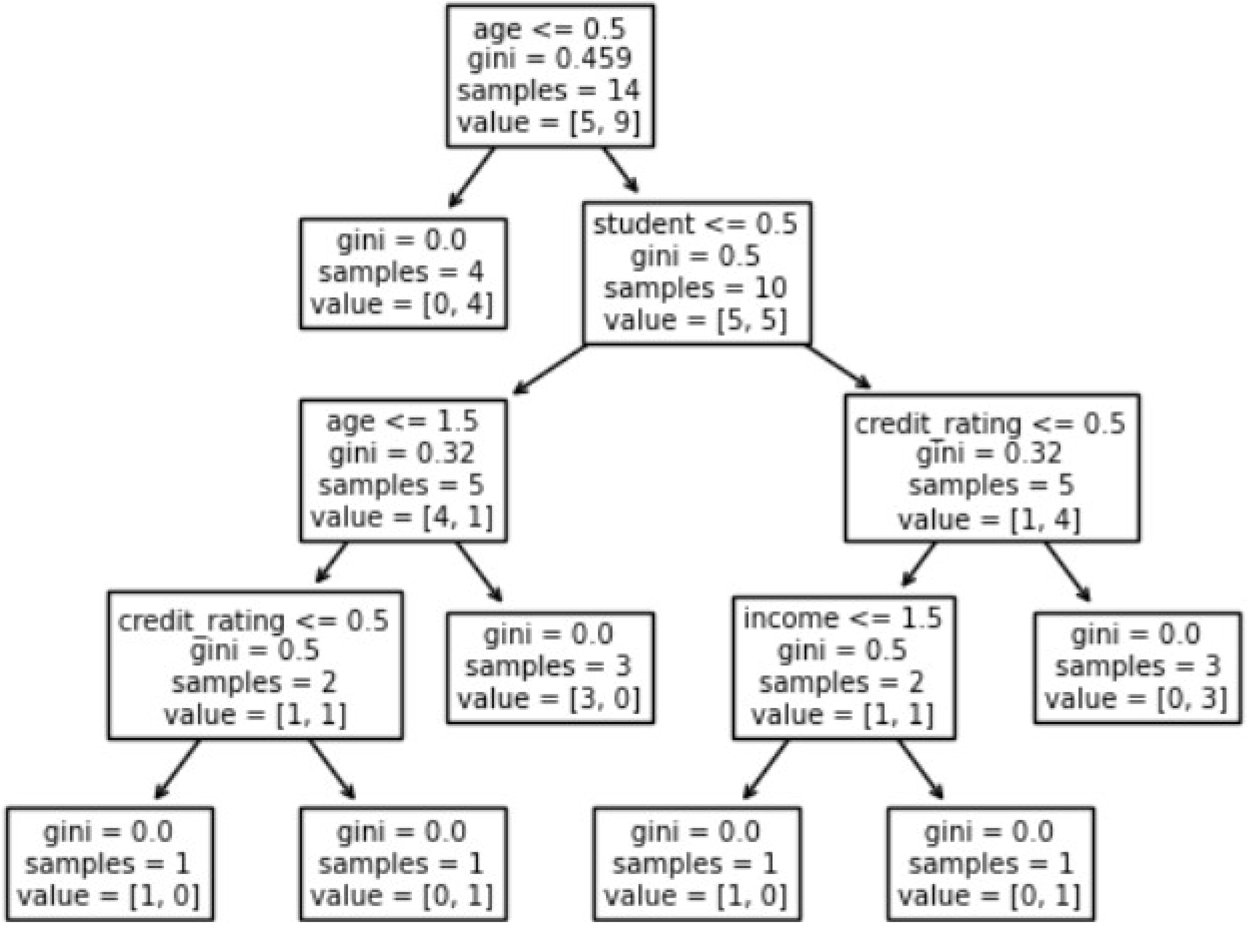
We split on age and on the right you are seeing the decision that we have yet to build.

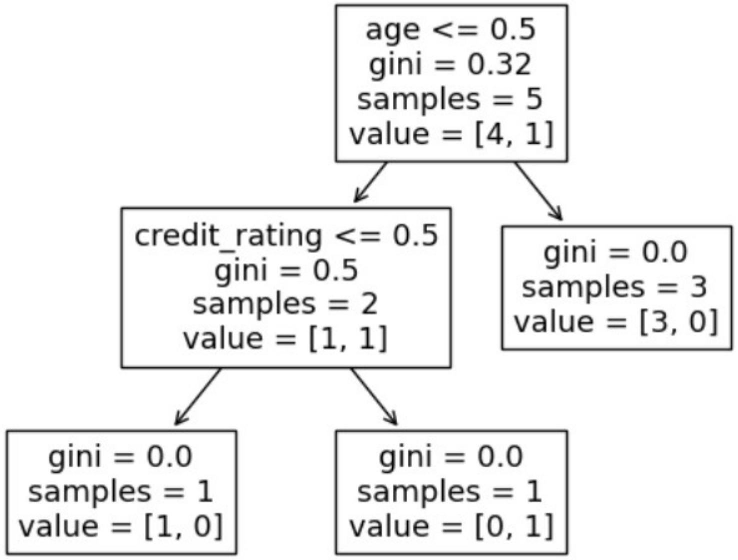


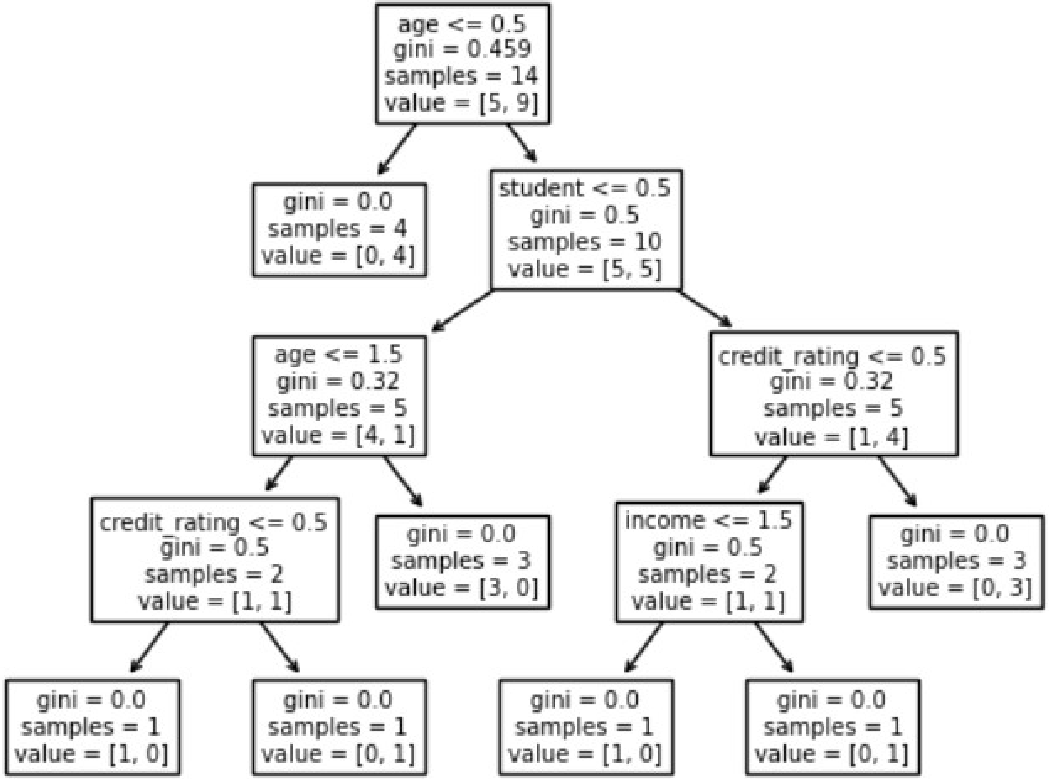
age != middle_aged

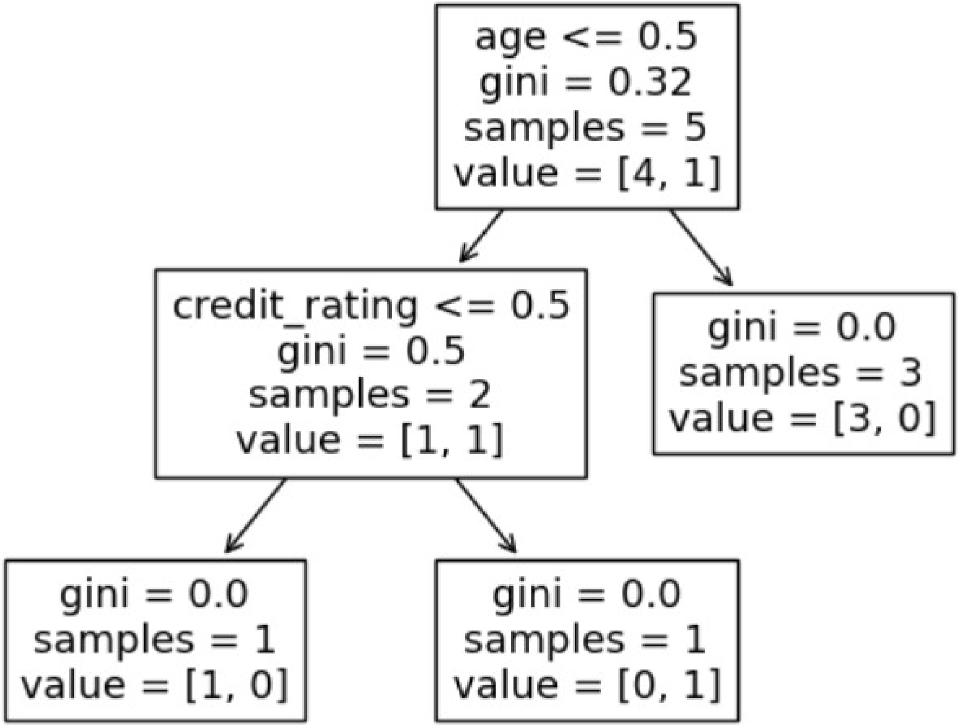


After we split on student and we are left building the following decision tree on the left branch.

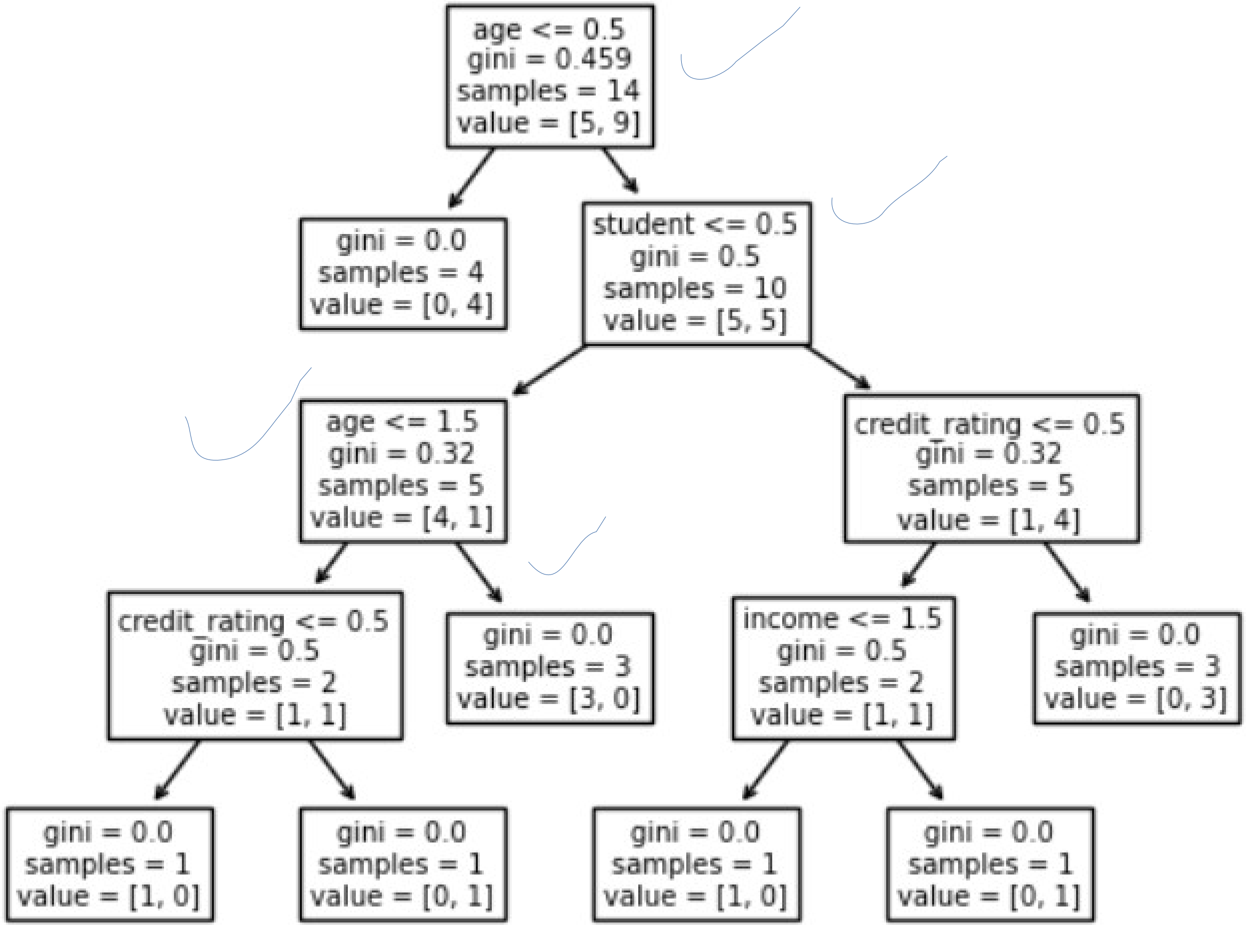








After we split on student and we are left building the following decision tree on the right branch.



Record: (
Age == youth | 2,
Income == high | 0,
Student == no | 0,
Credit_rating == fair | 1
)

Checks age <= 0.5 == False
Checks student <= 0.5 == True
Check age <= 1.5 == False
We get [3 (no's), 0 (yes's)]

RID	age	income	student	credit_rating	buys_computer	Step 1: age >= 0.5 (!= middle_aged)	Step 2: student <= 0.5 (no)	Step 3: age == 'youth' or age > 1.5
1	youth	high	no	fair	no	TRUE	TRUE	TRUE
2	youth	high	no	excellent	no	TRUE	TRUE	TRUE
3	middle_aged	high	no	fair	yes	FALSE	FALSE	FALSE
4	senior	medium	no	fair	yes	TRUE	TRUE	FALSE
5	senior	low	yes	fair	yes	TRUE	FALSE	FALSE
6	senior	low	yes	excellent	no	TRUE	FALSE	FALSE
7	middle_aged	low	yes	excellent	yes	FALSE	FALSE	FALSE
8	youth	medium	no	fair	no	TRUE	TRUE	TRUE
9	youth	low	yes	fair	yes	TRUE	FALSE	FALSE
10	senior	medium	yes	fair	yes	TRUE	FALSE	FALSE
11	youth	medium	yes	excellent	yes	TRUE	FALSE	FALSE
12	middle_aged	medium	no	excellent	yes	FALSE	FALSE	FALSE
13	middle_aged	high	yes	fair	yes	FALSE	FALSE	FALSE
14	senior	medium	no	excellent	no	TRUE	TRUE	FALSE
						Not all 'buys_computer' are same	Not all 'buys_computer' are same	All 'buys_computer' are same: no