Supervised learning- Decision tree(2)

Parcours Progis

Etudes, Medias, communication, Marketing

Bahareh Afshinpour.

02.12.2024

Université Grenoble Alpes

Sciences Po Grenoble

References

- https://www.geeksforgeeks.org/k-nearest-neighbours/
- https://www.youtube.com/watch?v=pR-Of1ua6Dc

- There are two main methods that are commonly used to split the data:
  a) Gini impurity and
  b) entropy information gain.

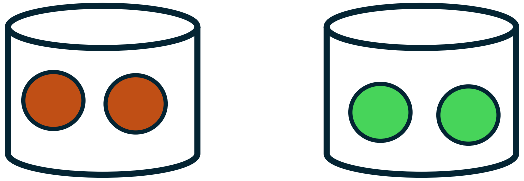
Example of Desision Tree- visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

Which of these columns(features) best splits these labels into the largest purest buckets?

We have two rows less that 50k and two more than 50k



No

Yes

Feature x

<=50k

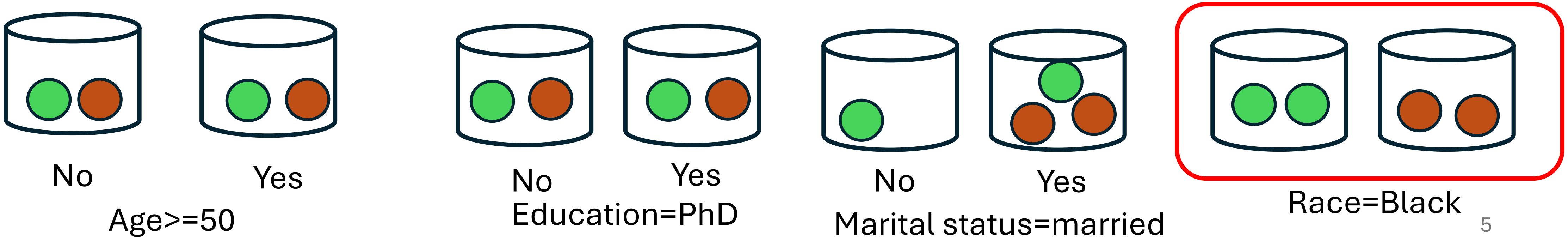
>50 k

Example of Desision Tree- visual representation

Target variable

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50k
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

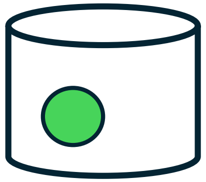
Race is a best one 100% pure

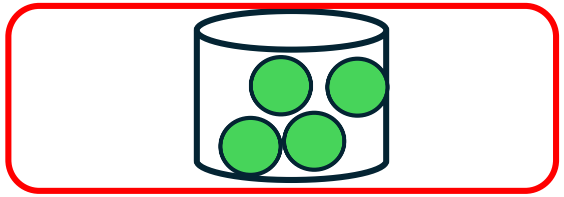


Gini impurity

- The probability that decision tree made a mistake.
  - High Gini ipmurity is bad
  - Low Gini impurity is good
- The algorithm goes to check a features one by one (like we just saw), and it calculates this gini impurity score for each one of the features.
- One that it picks is the one with the best that is the **lowest** gini impurity score.
- Gini consider bothe the **purity** and the **weight** of the leaves.

Not much weight





We have much weight.

Binning

- We need to convert the numeric feature into multiple classes (like age>50)
- Finding a cut off (finding the rule for a numeric column is a non-trivial task)
- We are going to create a rule (hypothetical decision)
- How does efficiently the algorithm find these thresholds for the rules
  -age <30 or age>50 or ……

✓It finds split point

✓It takes a copy of that numeric data and then it sorts it(ascending order)

Binning example

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

30 48 55 61

39 51.5 58

<40 <52 <59

We are going to find the split points :

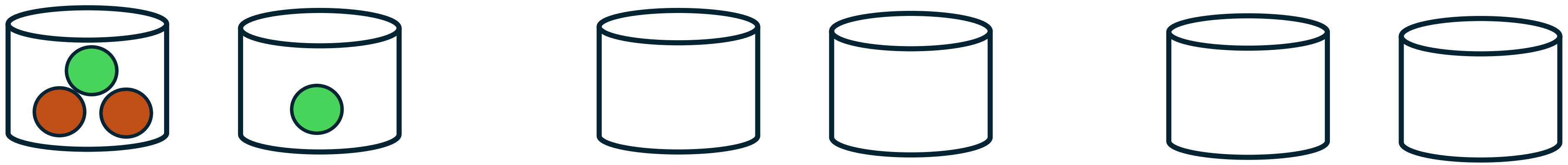
A bench of split points are calculated based on the differences between these numbers

What is the spilt point? The midpoint between adjacent values.

Binning example

- Which one has the best overall gini impurity score?

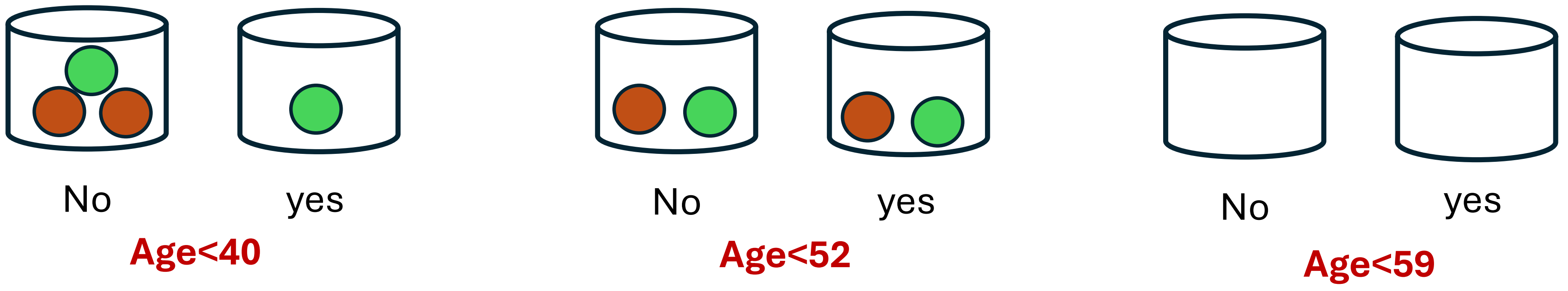
Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k



Binning example

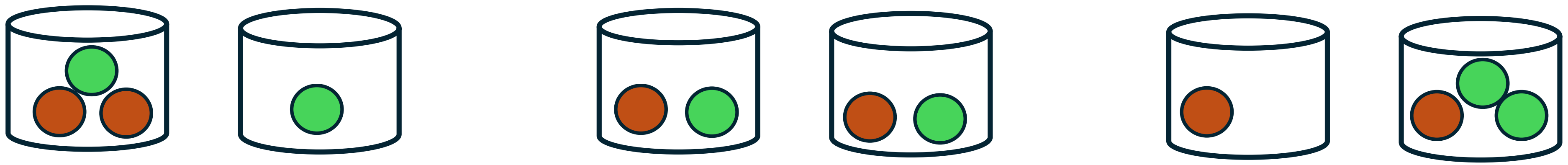
- Which one has the best overall gini impurity score?

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k



- Which one has the best overall gini impurity score?

Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k

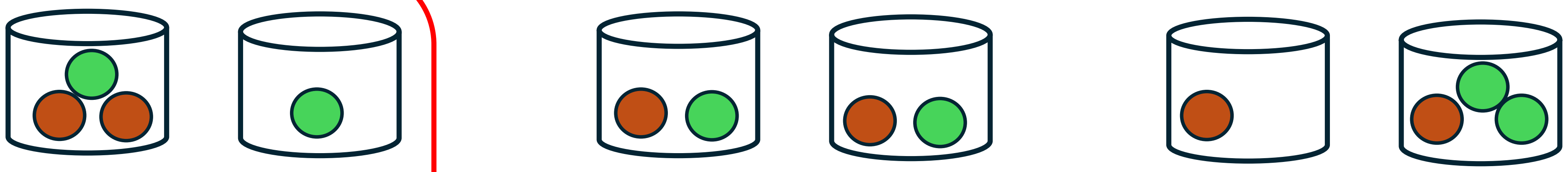


No yes
Age<40

No yes
Age<52

No yes
Age<59

- Which one has the best overall gini impurity score?



No yes
Age<40 **100%**
The first best one is chosen

No yes
Age<52
Not good 50/50

No yes
Age<59

Age<40

Imagine we have many rows (records) in our dataset.

Race=black

yes

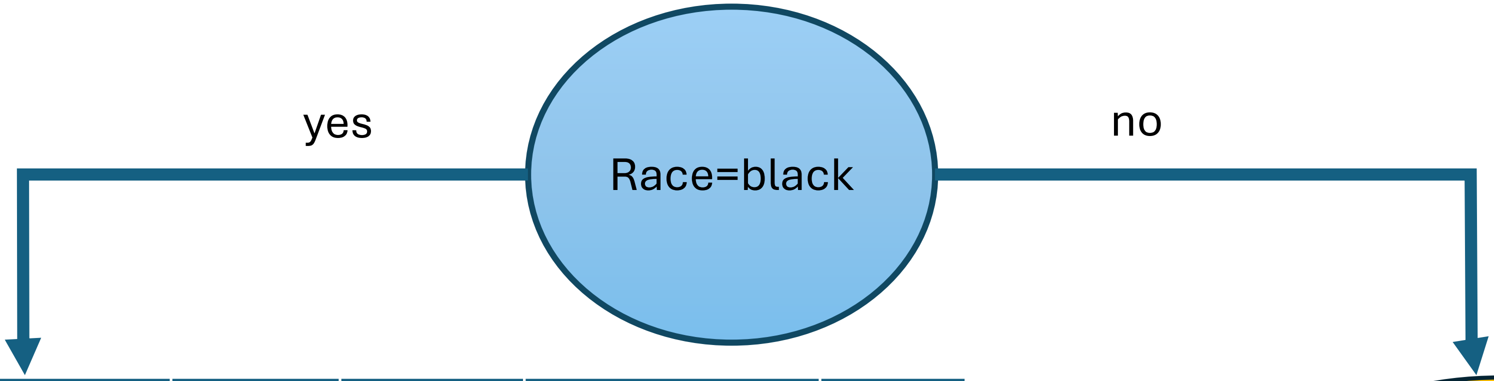
Age	Education	Marital status	Race	Sex	Hours Per Week	Label
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k
50	master	married	Black	Female	55	<=50
⋮						⋮

no

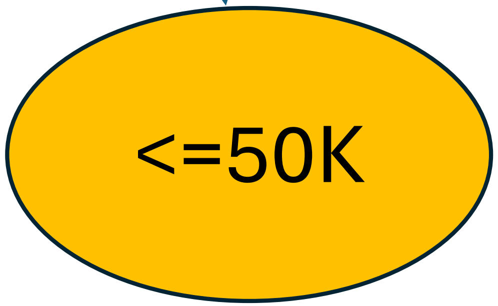
Age	Education	Marital status	Race	Sex	Hours Per Week	Label
61	master	maried	White	Male	40	<=50k
48	PhD	divorse	White	Female	16	<=50

100% pure

Because we have all the same label value, it is pure

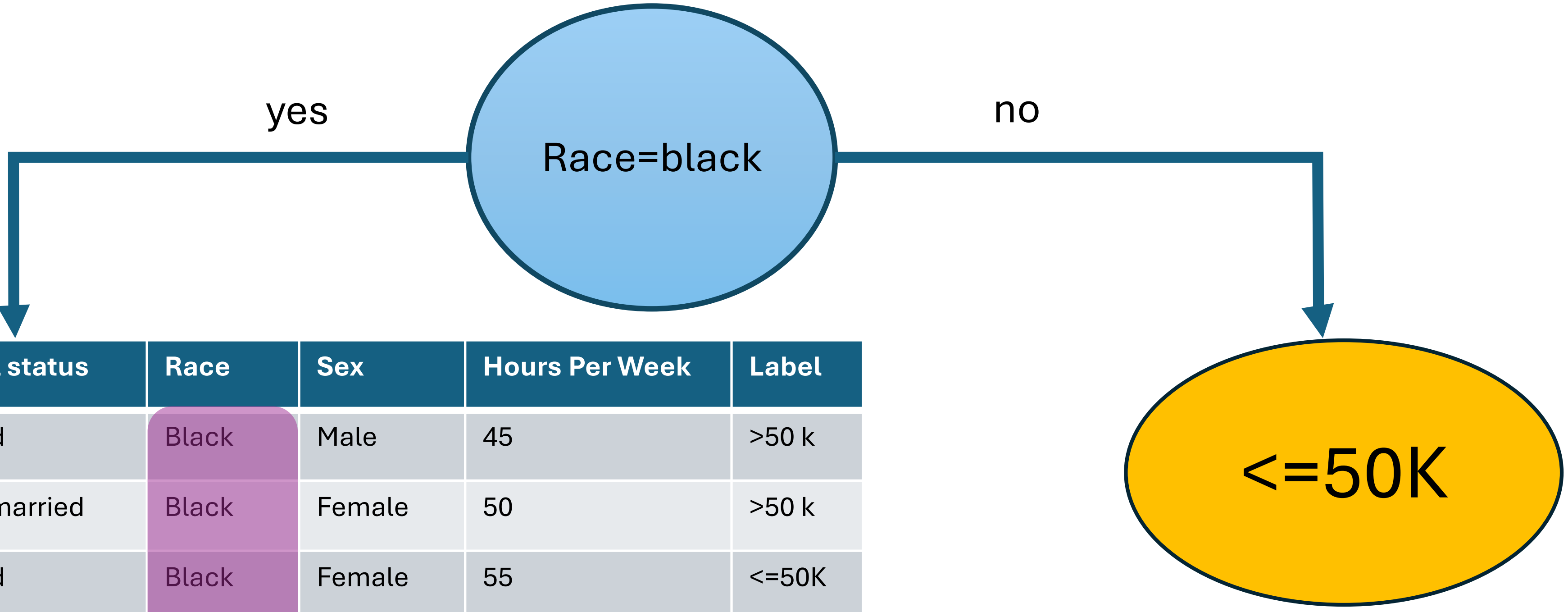


Age	Education	Marital status	Race	Sex	Hours Per Week	Label
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k
50	master	married	Black	Female	55	<=50K



Left side: we do not have purity
So, the algorithm try to split it again

A Leaf node
With a prediction
label

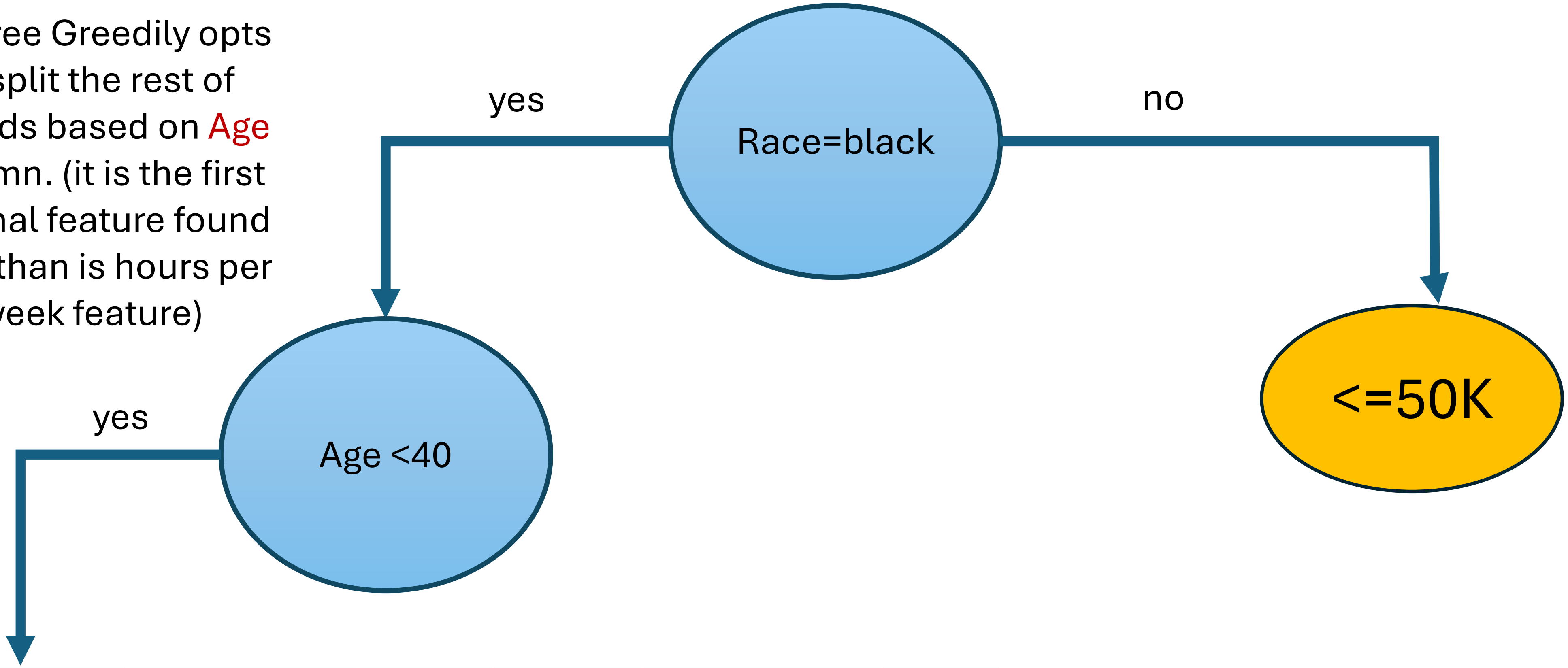


Age	Education	Marital status	Race	Sex	Hours Per Week	Label
55	PhD	married	Black	Male	45	>50 k
30	master	Never married	Black	Female	50	>50 k
50	master	married	Black	Female	55	<=50K

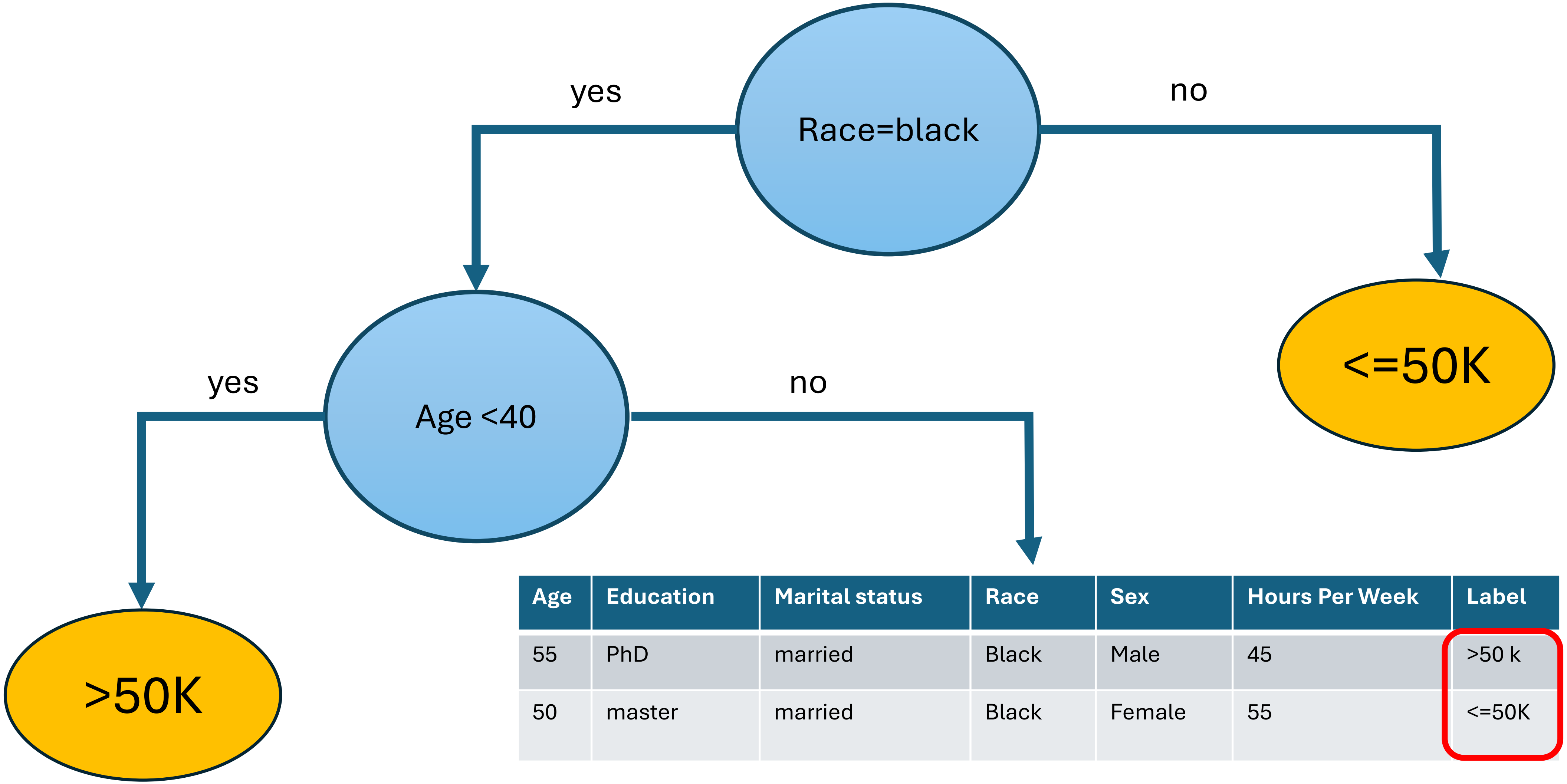
- But all the values in the Race column are the same.
- The algorithm **masks** them since they have no useful information.
- The algorithm starts to find the best column for the next condition.

A Leaf node
With a prediction
label

The tree Greedily opts to split the rest of records based on Age column. (it is the first optimal feature found after than is hours per week feature)



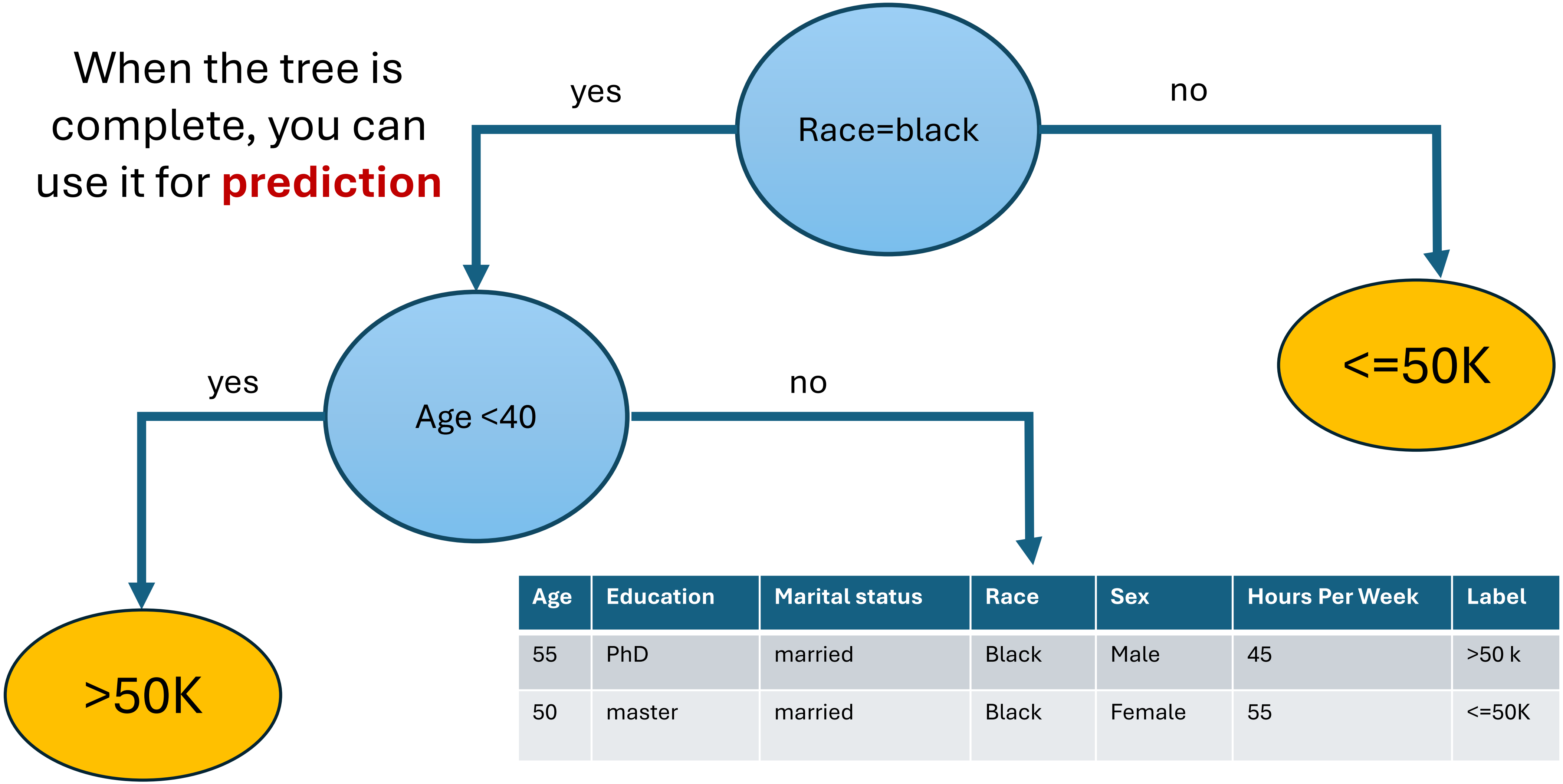
Age	Education	Marital status	Race	Sex	Hours Per Week	Label
30	master	Never married	Black	Female	50	>50 k



Age	Education	Marital status	Race	Sex	Hours Per Week	Label
55	PhD	married	Black	Male	45	>50 k
50	master	married	Black	Female	55	<=50K

We should continue

When the tree is complete, you can use it for **prediction**



Age	Education	Marital status	Race	Sex	Hours Per Week	Label
55	PhD	married	Black	Male	45	>50 k
50	master	married	Black	Female	55	<=50K

When the algorithm stop to split

- When the node is 100% pure.
- Based on Hyperparameters
  - You can set the thresholds for such things qs:
    - The max dept of the tree
    - The min number of record that fall into a leaf node
    - ....

A **hyperparameter**, on the other hand, is a variable that is set before the training process begins.

Hyperparameters are not learned from the data but are instead set by the user or determined through a process known as hyperparameter optimization.

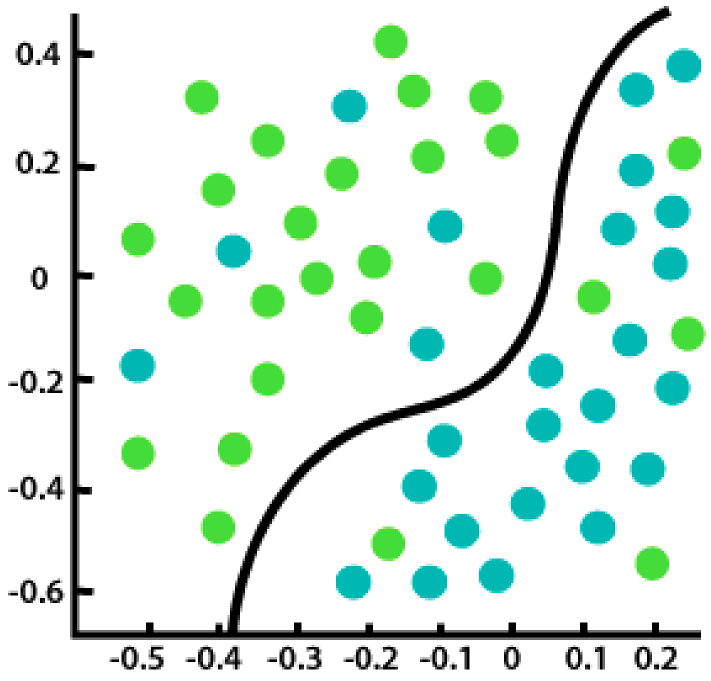
Arbres de décision

- Les arbres de décision sont utilisables pour faire de la régression. Au lieu d’associer une classe à chaque feuille, c’est la valeur moyenne de la variable cible des éléments dans cette feuille qui sera utilisée.

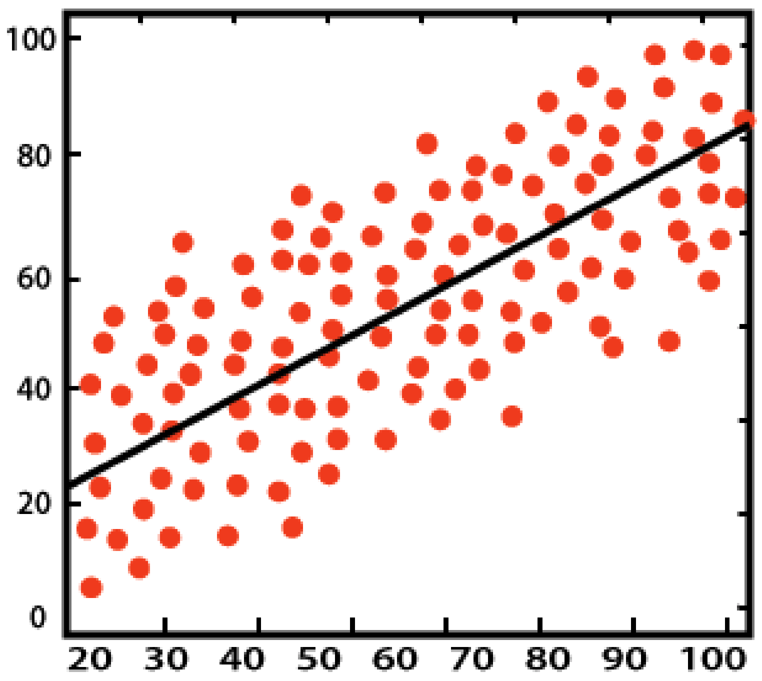
- En scikit-learn, la classe à utiliser est un DecisionTreeRegressor.

```
from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor(max_depth=2)

regressor.fit(X, y)

y_pred = regressor.predict(X_test)
```

Supervised learning- Decision tree(2)



Classification

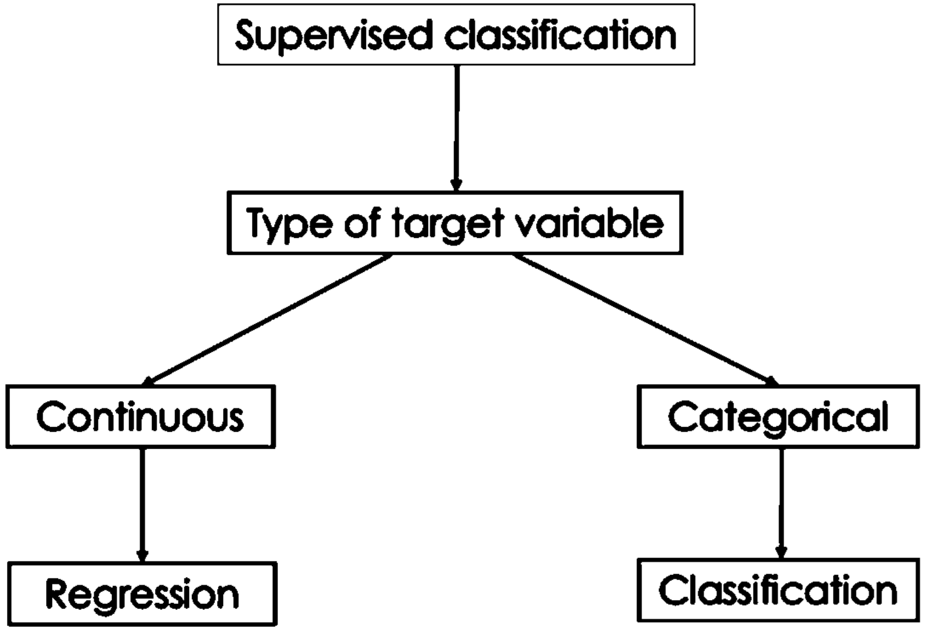


Regression

https://www.javatpoint.com/regression-vs-classification-in-machine-learning

What is Classification in Machine Learning?

- Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data.
- In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

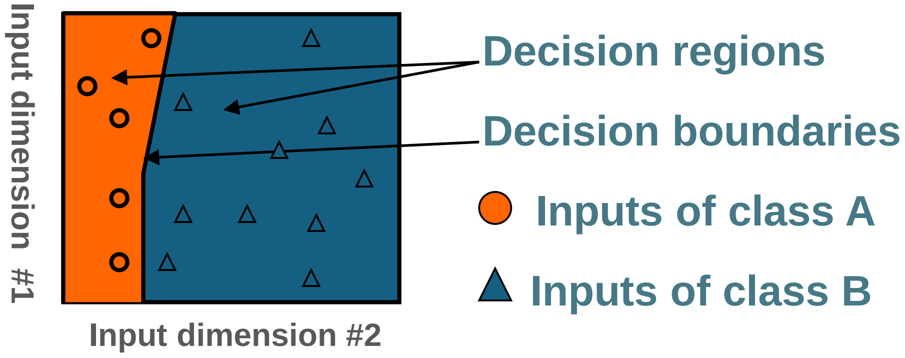


https://www.datacamp.com/blog/classification-machine-learning

Classification: Terminology

- A *classifier* can be viewed as a function of block.
- A classifier assigns one class to each point of the input space.
- The input space is thus partitioned into disjoint subsets, called *decision regions*, each associated with a class.

Classification: Terminology (cont.)



- The way a classifier classifies inputs is defined by its decision regions.
- The borderlines between decision regions are called *decision-region boundaries* or simply *decision boundaries*.