Data Science Module 4

Day 1 - Decision Trees

Today we are going to learn to...

1. Explain how a decision tree is created
2. Build a decision tree model in scikit-learn
3. Tune a decision tree model and explain how tuning impacts the model
4. Interpret a tree diagram
5. Describe the key differences between regression and classification trees

...and then we are done!

Before we start...

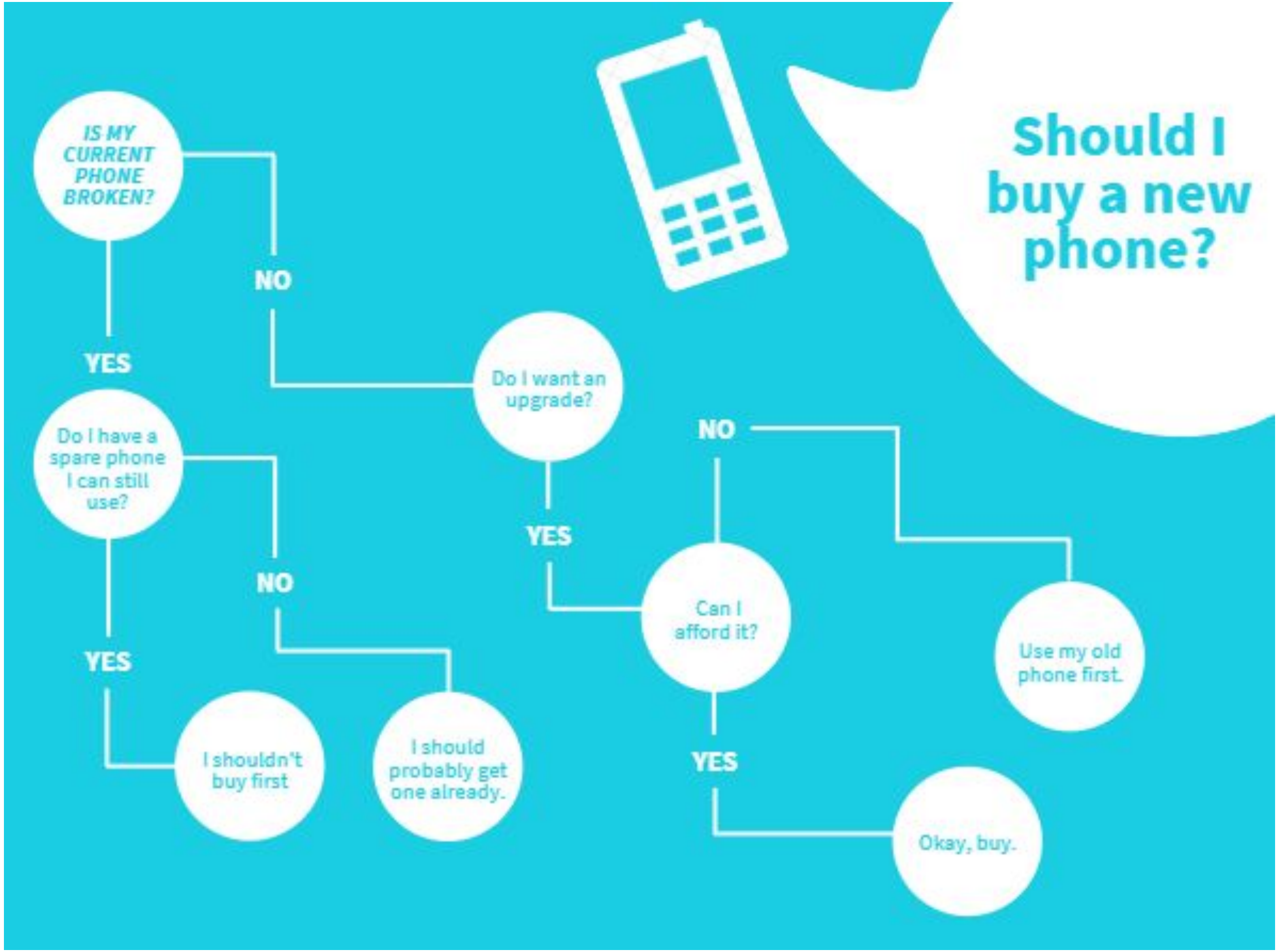
1. Make sure you are comfortable
2. Have water and maybe a strong coffee handy
3. If you need a break...take it!
4. If you need a stretch - please go ahead!
5. Please mute yourselves if you are not talking
6. Have your video on at all times

...and let's get started!

Decision Trees

Should you buy a new phone?

- Yes/No?
- How did you arrive to the answer?



Should I buy a new phone?
IS MY CURRENT PHONE BROKEN?
NO
YES
Do I want an upgrade?
Do I have a spare phone I can still use?
NO
YES
YES
NO
Can I afford it?
Use my old phone first.
YES
I shouldn't buy first
I should probably get one already.
Okay, buy.

Decision Trees

Decision Trees are a machine learning model for **regression and classification** that develops a series of **yes/no rules** to explain the differences present in the outcome variable.

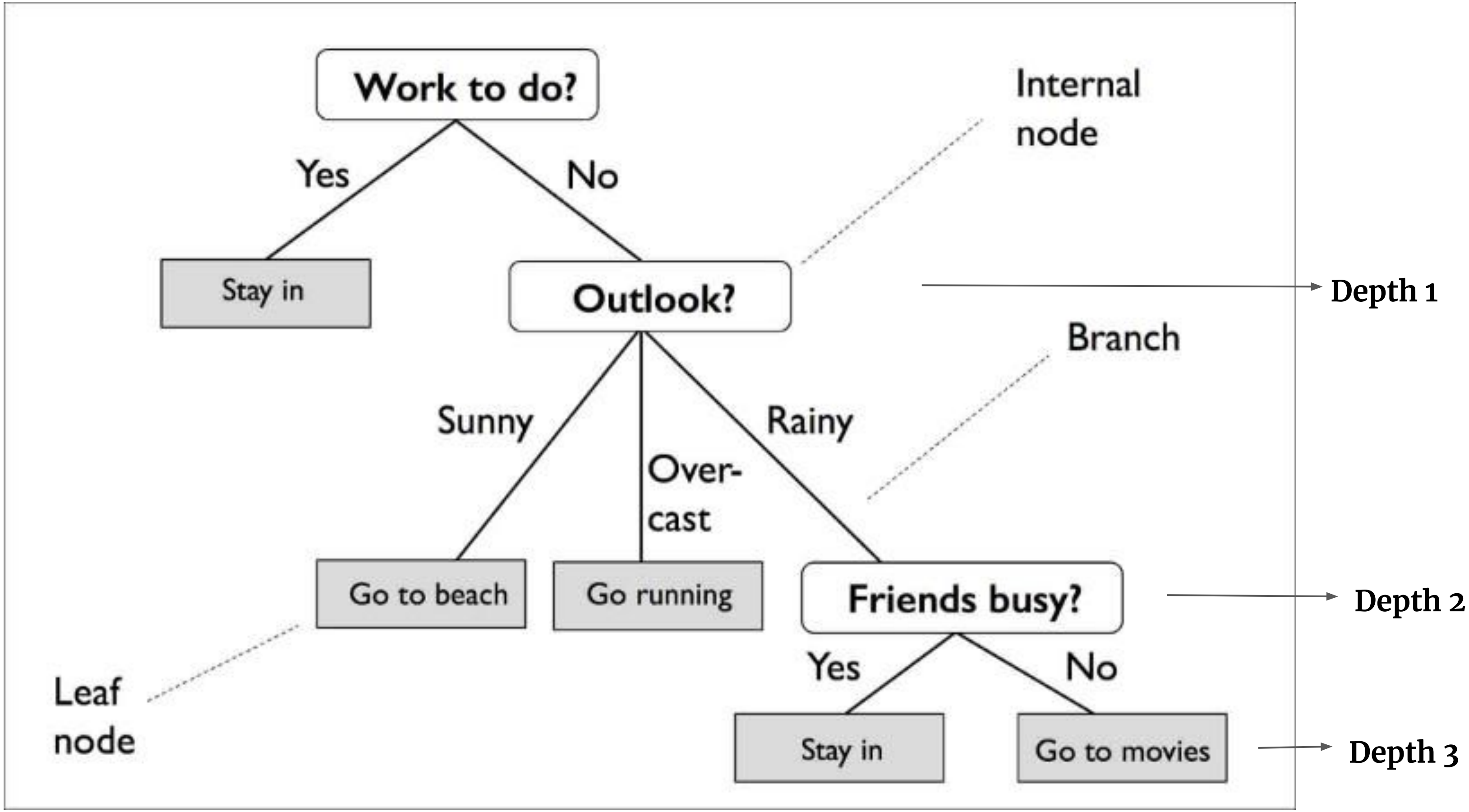
Decision Trees

Decision Trees are a machine learning model for **regression and classification** that develops a series of **yes/no rules** to explain the differences present in the outcome variable.

Why Decision Trees?

- They can be applied to both regression and classification problems.
- They are easy to explain to others (interpretable).
- They are very popular among data scientists.
- They are the basis for more sophisticated models.
- They have a different way of "thinking" than the other models we have studied.

Decision Trees - Terminology



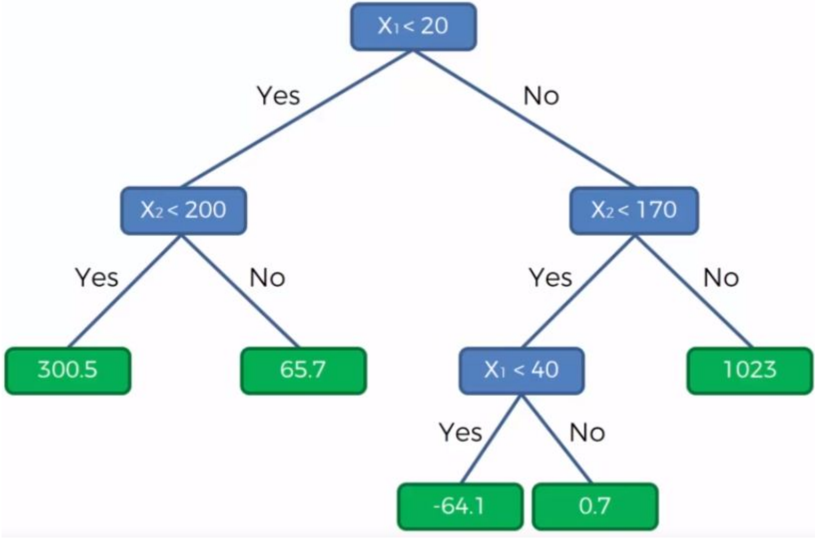
Decision Tree - Types

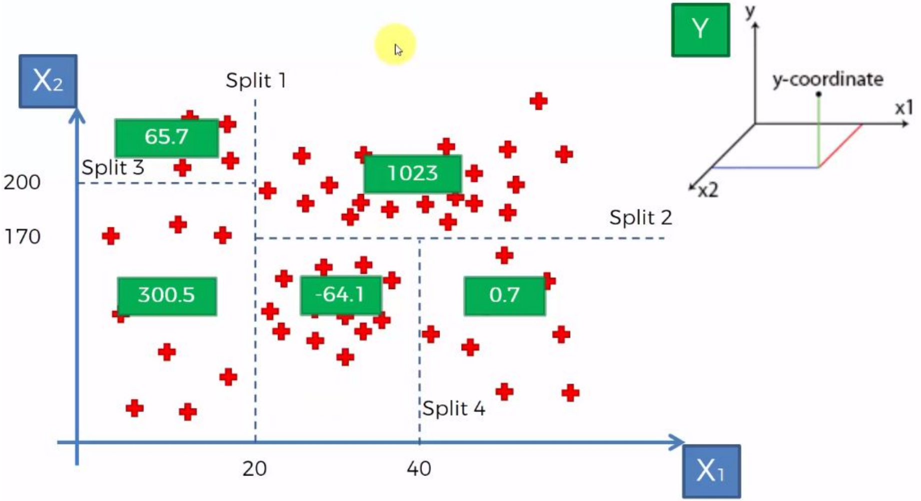
Regression Trees	Classification Trees
Predict a **continuous response**	Predict a **categorical response**
Predict using **mean response of each leaf**	Predict using most **commonly occurring class of each leaf**
Splits are chosen to minimize MSE	Splits are chosen to **minimize Gini index** (we'll discuss later)

Regression Trees

Regression Trees

Predict numeric data (equivalent to a Linear regression Model)





Regression Trees with `scikit-learn`

```
from sklearn.tree import DecisionTreeRegressor

treereg = DecisionTreeRegressor(random_state=42)
```

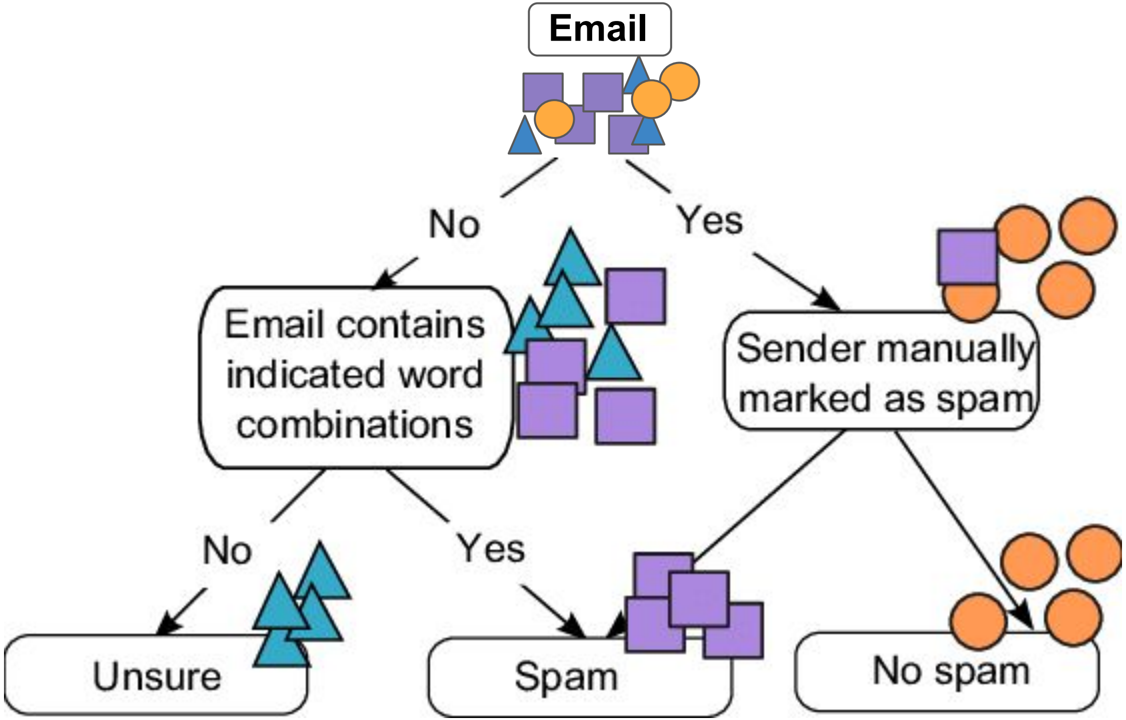
```
DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None,
                      max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort='deprecated',
                      random_state=42, splitter='best')
```

Key Parameters for Optimizing Decision Tree Performance

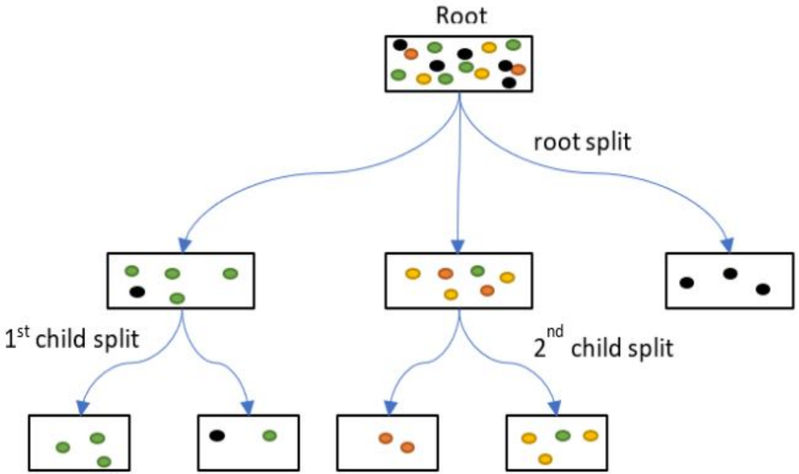
- **criterion : optional (default="gini") or Choose attribute selection measure**: This parameter allows us to use the different-different attribute selection measure. Supported criteria are "gini" for the Gini index and "entropy" for the information gain.
- **splitter : string, optional (default="best") or Split Strategy**: This parameter allows us to choose the split strategy. Supported strategies are "best" to choose the best split and "random" to choose the best random split.
- **max_depth : int or None, optional (default=None) or Maximum Depth of a Tree**: The maximum depth of the tree. If None, then nodes are expanded until all the leaves contain less than min_samples_split samples. The higher value of maximum depth causes overfitting, and a lower value causes underfitting.

Classification Trees

Classification Trees



How is splitting decided?



The data is repeatedly split according to predictor variables so that child nodes are **more “pure” (i.e., homogeneous)** in terms of the outcome variable

Classification Trees splitting Criteria

- **Classification Error Rate:** The fraction of training observations in a region that don't belong to the most common class
- **Gini Index:** The measure of total variance across classes in a region

Classification Error Rate

Task:

Predicting whether or not someone will buy an iPhone or an Android

Observation at node *n*:

- **25** Observations
- **15** bought androids
- **10** bought iPhones

Classification Error Rate:

If we assume all 25 is predicted for the majority class (Android), then we have

10 wrong predictions (iPhones), and thus ***CER = 10/25 = 40%***

Our goal in making splits is to *reduce the classification error rate*

Classification Error Rate

Task:

Predicting whether or not someone will buy an iPhone or an Android

Observation at node *n +1*:

- **Males:** 2 iPhones and 12 Androids, thus the predicted class is Android.
- **Females:** 8 iPhones and 3 Androids, thus the predicted class is iPhone

Classification Error Rate:

CER = 5/25 = 20%

Our goal in making splits is to *<u>reduce the classification error rate</u>*

Classification Error Rate

Task:

Predicting whether or not someone will buy an iPhone or an Android

Observation at node *n +1*:

- **30 or younger:** Four iPhones and eight Androids, thus the predicted class is Android.
- **31 or older:** Six iPhones and seven Androids, thus the predicted class is Android

Classification Error Rate:

CER = 10/25 = 40%

Our goal in making splits is to *<u>reduce the classification error rate</u>*

Classification Gini Index

1. It works with categorical target variable “Success” or “Failure”
2. It performs only Binary splits
3. The **maximum value** of the Gini index is 0.5 and occurs when the classes are perfectly balanced in a node.
4. The **minimum value** of the Gini index is 0 and occurs when there is only one class represented in a node. The node is **pure.**
5. CART (Classification and Regression Tree) uses Gini method to create binary splits

Classification Gini Index

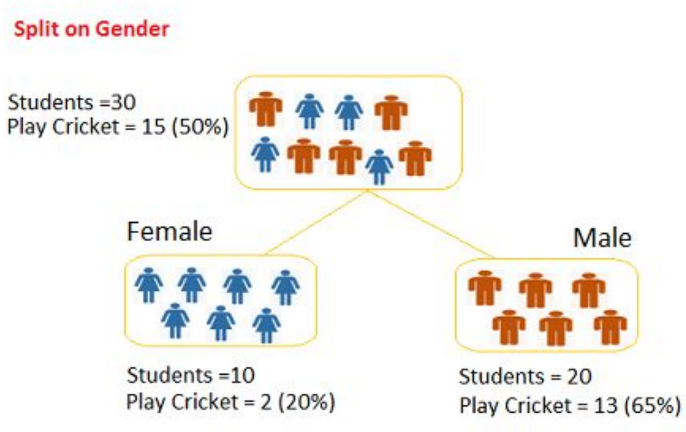
Steps to Calculate Gini for a split

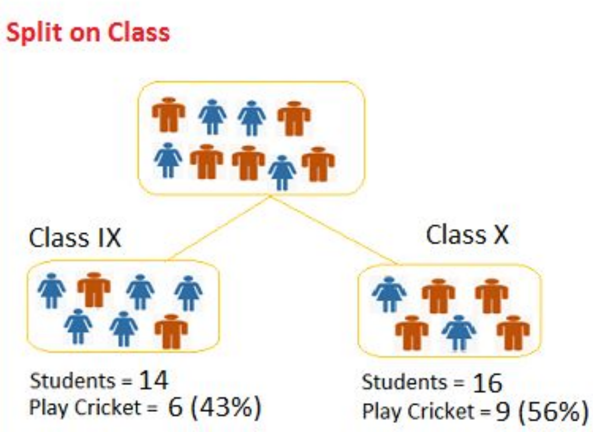
1. Calculate Gini for sub-nodes

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

2. Calculate Gini for split using weighted Gini score of each node of that split

Classification Gini Index





Gini for sub-node Female =1 - (0.2)*(0.2)+(0.8)*(0.8)=0.32

Gini for sub-node Male = 1 - (0.65)*(0.65)+(0.35)*(0.35)=0.45

Weighted Gini for Split Gender = (10/30)*0.32+(20/30)*0.45 = **0.41**

Gini for sub-node Class IX = 1- (0.43)*(0.43)+(0.57)*(0.57)=0.49

Gini for sub-node Class X = 1- (0.56)*(0.56)+(0.44)*(0.44)=0.49

Weighted Gini for Split Class = (14/30)*0.49+(16/30)*0.49 = **0.49**

Classification Trees with `scikit-learn`

```
from sklearn.tree import DecisionTreeClassifier
treeclf = DecisionTreeClassifier(max_depth=3, random_state=42)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=3, 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='deprecated',
                       random_state=42, splitter='best')
```

jupyter