Intro to Machine Learning - Week 3

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Classification (Part 2)

Decision Trees
Overfitting, Underfitting
Ensembles Learning

What will we focus on?

Concepts, Problems

1 hour

Google Colab Project

1 hour

Schedule

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Supervised Learning
Classification & Regression, Hypothesis Testing

Classification 1
Conditional Probability, Naive Bayes, Bayesian Learning

Classification 2
Information Gain, Decision Trees, Random Forest, Ensembles

Regression 1
Linear Regression, Loss Function, Gradient Descent, Back Propagation

Regression 2
Logistic Regression, Support Vector Machine, Model Tuning

Model Evaluation
Accuracy Metrics, Over- & Underfitting, Cross Validation

Classification

- Target responses are **categorical** in nature
- We are going to see decision tree based classification in this session

Decision Trees

- Class of algorithms that **learn decision rules** from a dataset.
- It generates a **tree-like model** of decisions and their possible consequences.
- Represented using a **flow-chart** like structure.
- Since it **maps all possible consequences** observed in your data, this would represent the **hypothesis space.**

Example Data

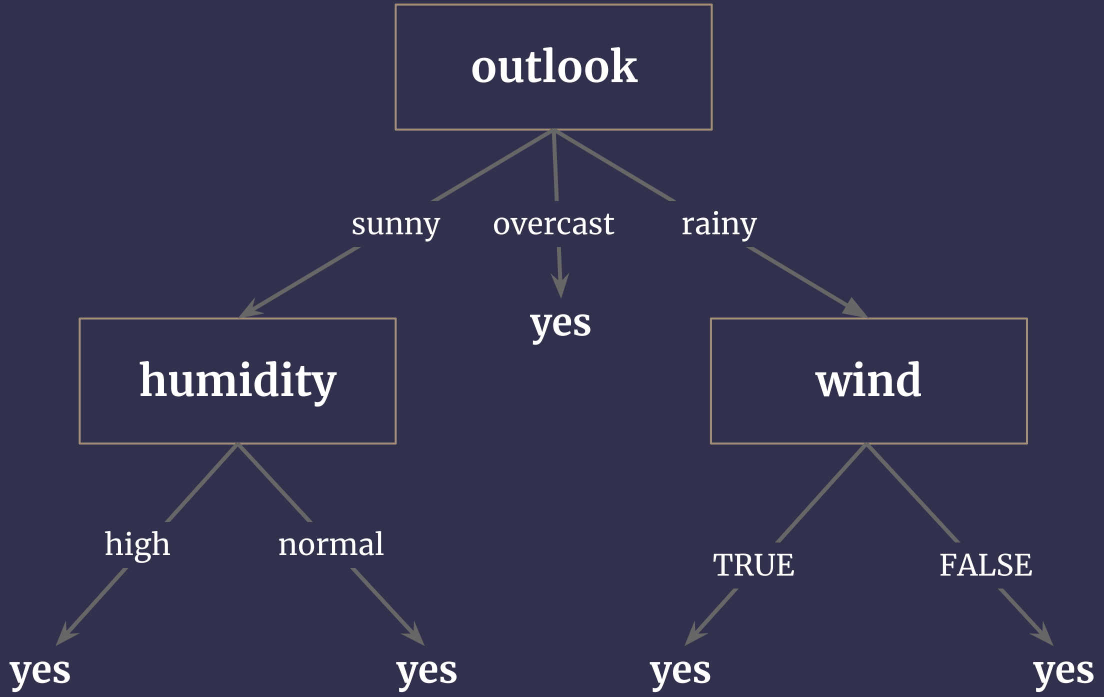
- 14 rows
- Class variable - play
- Features - outlook, temperature, humidity, windy

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

Decision Tree Example

How can we represent categorical values?

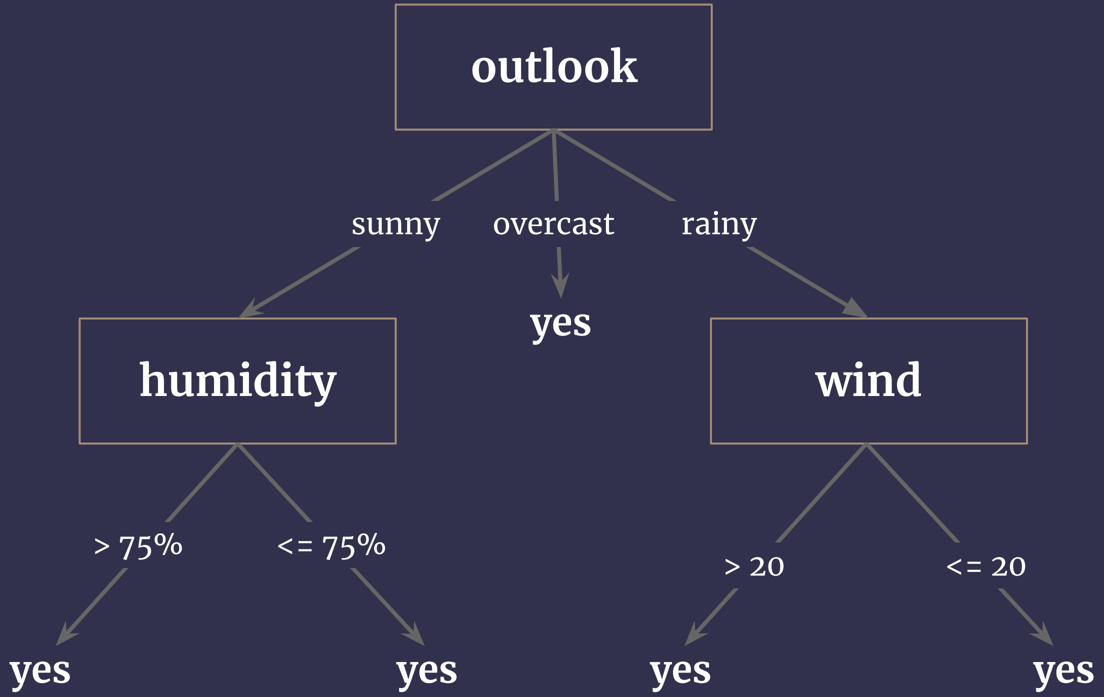
Let us ignore temperature for this.



Decision Tree Example

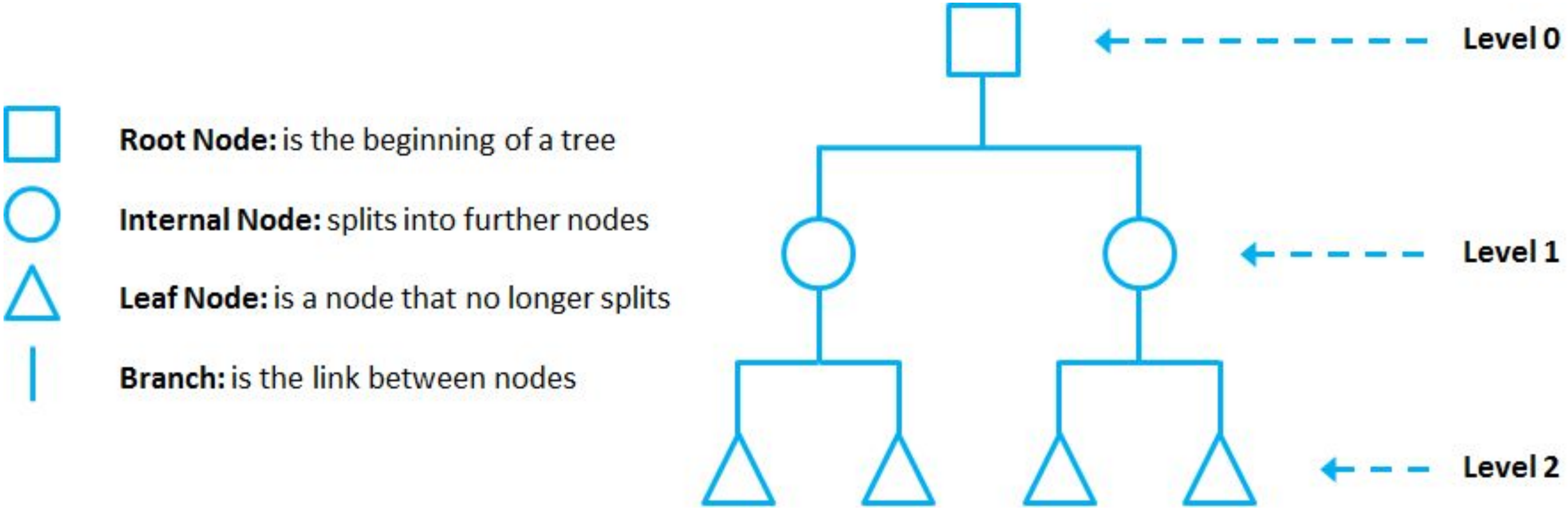
How can we represent continuous values?

Let us assume humidity and wind are continuous for this.



How does it work?

Elements in a Decision Tree



https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

How to build a Decision Tree

Decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous)

How do we decide which feature is the root of the tree?

Entropy

How do we decide which feature is the root of the tree?

- Entropy is used to **calculate homogeneity** of a data sample.
- Average rate at which **information** is produced by a stochastic data source
- Entropy value ranges from **0 - 1**
  - 0 = completely homogeneous
  - 1 = more uncertainity

https://en.wikipedia.org/wiki/Entropy_(information_theory)
https://www.saedsayad.com/decision_tree.htm

Example Data

- 14 rows
- Features
  - x_1 = outlook
  - x_2 = temperature
  - x_3 = humidity
  - x_4 = windy
- Target
  - y = play
    - c_1 = yes
    - C_2 = no

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

Entropy Formula

WOMEN WHO CODE SILICON VALLEY X WOMEN WHO CODE DATA SCIENCE

$$\underbrace{E(y)}_{\text{Entropy of Target Response}} = \sum_{i=(1,C)} \underbrace{-P(c_i)}_{\text{Probability of class } c_i} \times \log_2 P(c_i)$$

$$\underbrace{E(y,X)}_{\text{Entropy of target class for input feature } x_i \in X} = \sum_{c \in X} \underbrace{P(c_i)}_{\text{Probability of class c of } x_i} \times \underbrace{E(c_i)}_{\text{Entropy of class c of } x_i}$$

Decision Tree – Step 1

Entropy of the target response

Unique values of target y

y	count	probability
y_1		
y_2		
...		
y_Y		
total		

Decision Tree – Step 1

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Entropy of the feature class with respect to target response

1. For every feature

2. Find the Unique values of feature x_i

3. Counts

4. Calculate Probabilities

x_i	count	play = yes	play = no	P(play = yes)	P(play = no)
c_1					
c_2					
...					
c_C					

E(play)

play	count	probability
yes	9	0.64
no	5	0.36
total	14	1

$$\underbrace{E(y)}_{\text{Entropy of Target Response}} = \sum_{i=(1,C)} -\underbrace{P(c_i)}_{\text{Probability of class } c_i} \times \log_2 P(c_i)$$

$$E(play) = -0.64\log_2(0.64) - 0.34\log_2(0.34)$$

$$= 0.94$$

E(play,outlook)

$$E(y,X) = \sum_{c \in X} \underbrace{P(c_i)}_{\text{Probability of class c of } x_i} \times \underbrace{E(c_i)}_{\text{Entropy of class c of } x_i}$$

Entropy of target class for input feature $x_i \in X$

outlook	count	yes	no	P(yes)	P(no)
sunny	5	3	2	0.6	0.4
overcast	4	4	0	1	0
rainy	5	2	3	0.4	0.6

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

$$E(play,outlook) = P(sunny) \times E(sunny) + P(overcast) \times E(overcast) + P(sunny) \times E(sunny)$$

$$= 5/14 \times E(3,2) + 4/14 \times E(4,0) + 5/14 \times E(2,3)$$

$$E(3,2) = -\tfrac{3}{5}\log_2\tfrac{3}{5} - \tfrac{2}{5}\log\tfrac{2}{5} = -(0.6 \times 0.737)-(0.4 \times 0.529) = 0.971$$

$$E(4,0) = -1\log_2 1 - 0\log_2 0 = 0$$

$$E(2,3) = E(3,2) = 0.971$$

$$= 0.357 \times 0.971 + 0.286 \times 0 + 0.357 \times 0.971$$

$$= 0.693$$

$$\boxed{E(play, outlook) = 0.693}$$

E(play,temperature)

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

$$\underbrace{E(y,X)}_{\text{Entropy of target class for input feature } x_i \in X} = \sum_{c \in X} \underbrace{P(c_i)}_{\text{Probability of class c of } x_i} \times \underbrace{E(c_i)}_{\text{Entropy of class c of } x_i}$$

temperature	count	yes	no	P(yes)	P(no)
hot	4	2	2	0.5	0.5
mild	6	4	2	0.67	0.33
cool	4	3	1	0.75	0.25

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

$$E(play,temp) = P(hot) \times E(hot) + P(mild) \times E(mild) + P(cool) \times E(cool)$$

$$= 4/14 \times E(2,2) + 6/14 \times E(4,2) + 4/14 \times E(3,1)$$

$$E(2,2) = -\ 2/4 \log_2 2/4 - 2/4 \log_2 2/4 = 1.0$$
$$E(4,2) = -\ 4/6 \log_2 4/6 - 2/6 \log_2 2/6 = 1.811$$
$$E(3,1) = -\ 3/4 \log_2 3/4 - 1/4 \log_2 1/4 = 0.918$$

$$= 0.286 \times 1.0 + 0.429 \times 1.811 + 0.286 \times 0.918$$

$$= 0.911$$

E(play, temperature) = 0.911

E(play,humidity)

$$\underbrace{E(y,X)}_{\text{Entropy of target class for input feature } x_i \in X} = \sum_{c \in X} \underbrace{P(c_i)}_{\text{Probability of class c of } x_i} \times \underbrace{E(c_i)}_{\text{Entropy of class c of } x_i}$$

humidity	count	yes	no	P(yes)	P(no)
normal	7	6	1	0.86	0.14
high	7	3	4	0.43	0.57

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

$$E(\text{play},\text{humidity}) = P(\text{normal}) \times E(\text{normal}) + P(\text{high}) \times E(\text{high})$$

$$= 7/14 \times E(6,1) + 7/14 \times E(3,4)$$

$$E(3,2) = -\ 6/7 \log_2 6/7 - 1/7 \log 1/7 = 0.985$$

$$E(3,4) = -\ 3/7 \log_2 3/7 - 4/7 \log 4/7 = 0.592$$

$$= 0.5 \times 0.985 + 0.5 \times 0.592$$

$$= 0.788$$

E(play, humidity) = 0.788

E(play,windy)

WOMEN WHO CODE SILICON VALLEY X WOMEN WHO CODE DATA SCIENCE

$$\underbrace{E(y,X)}_{\text{Entropy of target class for input feature } x_i \in X} = \sum_{c \in X} \underbrace{P(c_i)}_{\text{Probability of class c of } x_i} \times \underbrace{E(c_i)}_{\text{Entropy of class c of } x_i}$$

windy	count	yes	no	P(yes)	P(no)
TRUE	6	3	3	0.5	0.5
FALSE	8	6	2	0.75	0.25

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

$E(play,windy) = P(TRUE) \times E(TRUE) + P(FALSE) \times E(FALSE)$

$= 6/14 \times E(3,3) + 8/14 \times E(6,2)$

$E(3,3) = -3/6 \log_2 3/6 - 3/6 \log_2 3/6 = 1.0$

$E(6,2) = -6/8 \log_2 6/8 - 2/8 \log_2 2/8 = 0.811$

$= 0.428 \times 1.0 + 0.571 \times 0.811$

$= 0.892$

E(play, windy) = 0.892

What do we have till now?

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

E(play) = 0.94

E(play, outlook) = 0.693

E(play, temperature) = 0.911

E(play, humidity) = 0.788

E(play, windy) = 0.892

Decision Tree – Step 2

Calculate the information gain of each input feature

$$\underbrace{\mathrm{Gain}(y,x_i)}_{\text{Information gain of input feature } x_i \in X} = \underbrace{E(y)}_{\text{Entropy of Target Response}} \times \underbrace{E(y,x_i)}_{\text{Entropy of target class for input feature } x_i \in X}$$

Decision Tree – Step 2

Calculate the information gain of each input feature

y = play
x_1 = outlook
x_2 = temperature
x_3 = humidity
x_4 = windy

E(play) = **0.94**
E(play, outlook) = **0.693**
E(play, temp) = **0.911**
E(play, humidity) = **0.788**
E(play, windy) = **0.892**

Gain(play, outlook) = **E(play) - E(play,outlook)**
= 0.94 - 0.693 = **0.247**

Gain(play, temp) = **E(play) - E(play,outlook)**
= 0.94 - 0.911 = **0.029**

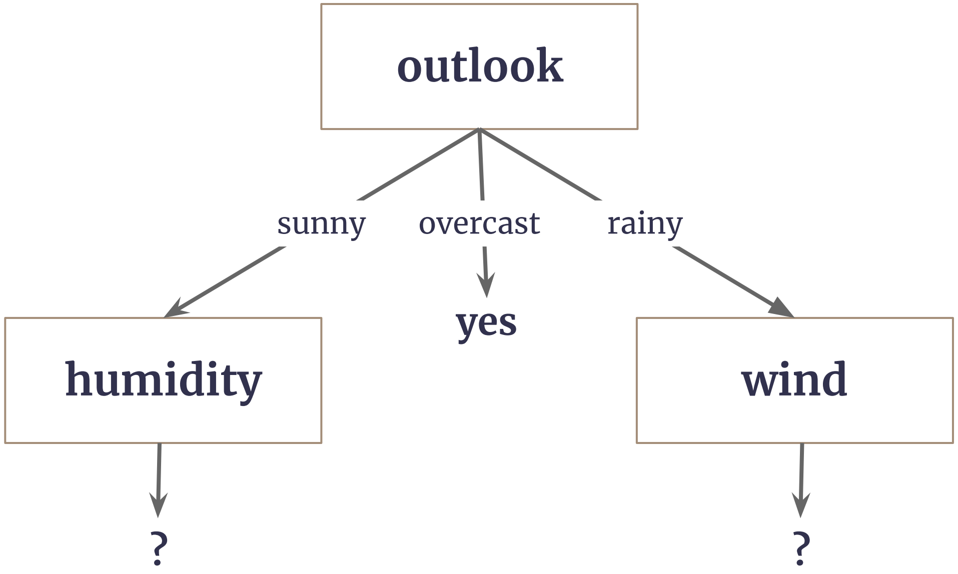
Gain(play, humidity) = **E(play) - E(play,humidity)**
= 0.94 - 0.788 = **0.152**

Gain(play, windy) = **E(play) - E(play,windy)**
= 0.94 - 0.892 = **0.048**

Decision Tree – Step 3

Pick the feature with highest information gain for first split

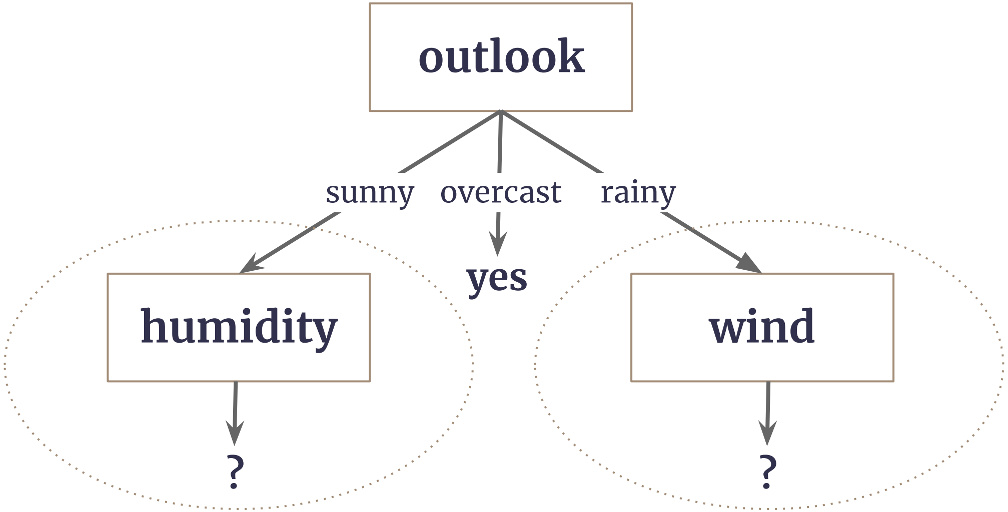
Gain(play, outlook) = 0.247
Gain(play, temp) = 0.029
Gain(play, humidity) = 0.152
Gain(play, windy) = 0.048



Decision Tree – Next Steps

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

- **Filter** the data samples based on previous decision.
- **Recursively** conduct Step 1 – 3 for every decision required.

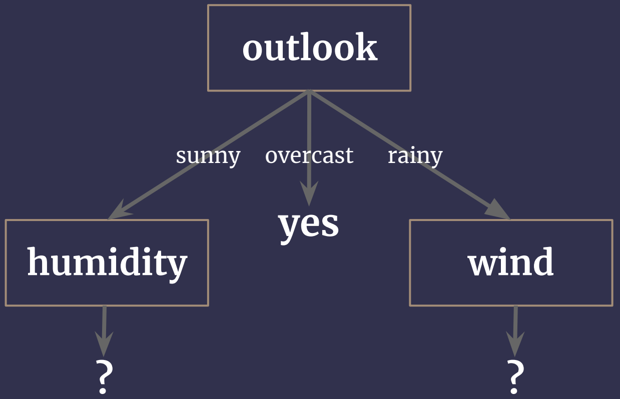


- Filter on outlook=sunny
- Conduct Step 1 – 3 on this data sample

- Filter on outlook=rainy
- Conduct Step 1 – 3 on this data sample

WOMEN WHO CODE SILICON VALLEY x WOMEN WHO CODE DATA SCIENCE

Homework #1



Complete the decision tree from the previous slide. Data sample listed below:

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

Decision Trees

Assumption

- Feature values are required to be categorical to calculate information gain
  - Continuous data can be discretized easily.
    - Eg. Bucketing ages, splitting based on quantile, etc

Decision Trees

Advantages

- Easy to understand/explain
  - **Intuitive** way to make decisions
- Features are **not** assumed to be **equally important** to the target response (hint: Naive Bayes)

Decision Trees

Variations

- Iterative Dichotomiser 3 (**ID3**)
- **C4.5**
- Chi-squared Automatic Interaction Detection (**CHAID**)
- **CART**

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

ID3

- Mainly used to produce Classification Trees
- Uses the **Information Gain** metric to select the most useful attributes for classification
  - The example we solved is ID3!

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

C4.5

- Can be used for both classification and regression trees
- Uses the **Gain Ratio** metric to select the most useful attributes for classification
  - Information gain that takes into account the number and size of the branches when choosing an attribute (IG shows unfair favoritism towards attributes with many outcomes)
  - Basically "normalizes" the Information Gain by using a split information value.

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

CHAID

- Chi-square tests check if there is a relationship between two variables
- Applied at each stage of the DT to ensure that each branch is significantly associated with a statistically significant predictor of the response variable.
- For classification trees, it uses the Chi squared test. For regression trees, it uses f-test
    - If the test fails, the nodes are merged into one

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

CART

- Produces binary classification/regression trees.
- Uses a metric called **Gini Impurity** to create decision points for classification tasks.
    - **Classification** - Measure of how mixed the classes are in groups created by the split. This has to be minimized.
    - **Regression** - minimizes the sum of the squared distances (or deviations) between the observed values and the predicted values (**L**east **S**quare **D**eviation)

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

Decision Trees

Disadvantages

- Calculations become complicated if there are **many class labels**
- Cannot estimate **missing data**
- **Overfitting**
  - Biased towards class value with more data samples (bias vs. variance)
  - Biased on data samples in the training data (pruning)

Decision Trees

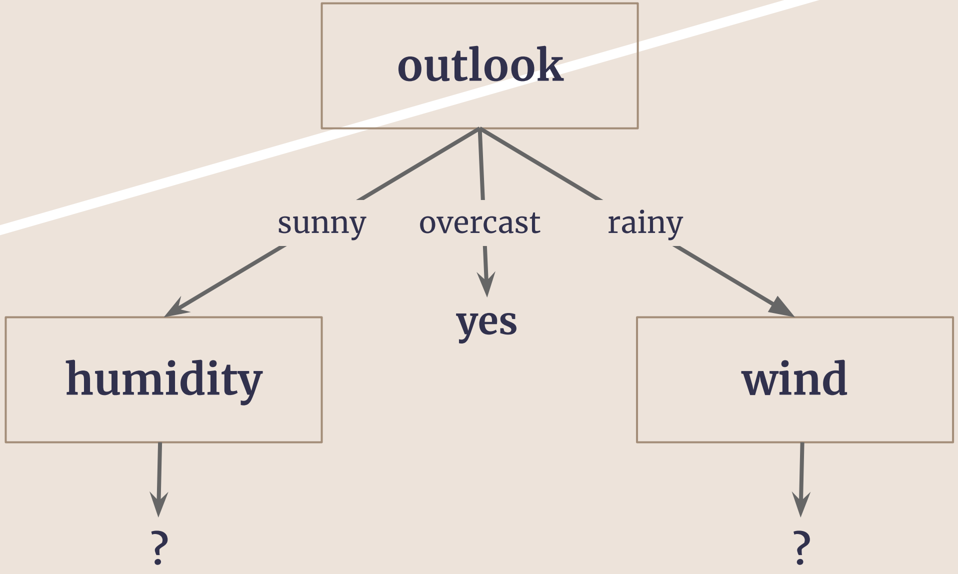
Overcoming Disadvantages

- **Understanding your data**
  - Bias vs. Variance Trade-Off
- **Understanding the algorithm**
  - Overfitting and underfitting
- **Understanding the Decision Tree**
  - Pruning
  - Ensemble Learning

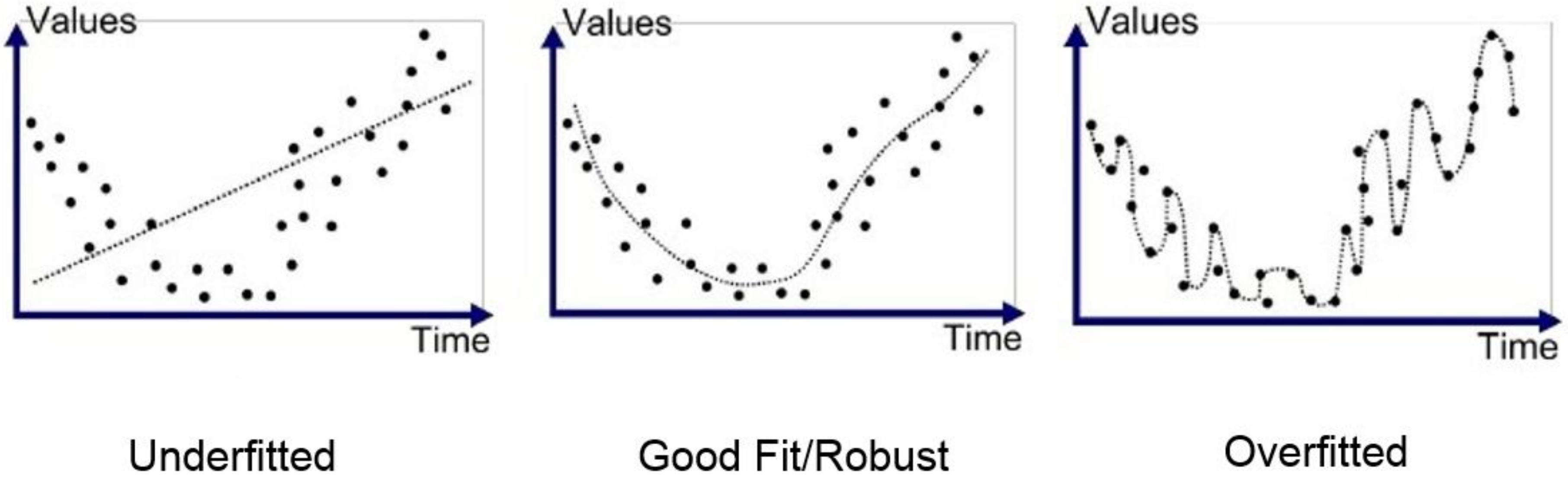
Terminology

Overfitting

- When the model isn't generalizable and contain biases from the dataset it was built on.
- eg. The relationship between play and outlook - there are no examples in the dataset of outlook=overcast and play=no.
  - A model is as good as its data



What is a good fit?



Terminology

Bias

- **Bias** is the tendency of a statistic to overestimate or underestimate a parameter.
- Assumptions made to make the target function easier to learn.
  - Low Bias: Suggests less assumptions about the form of the target function.
  - High-Bias: Suggests more assumptions about the form of the target function.

Terminology

Variance

- **Variance** measures how far a set of numbers are spread out from their average value.
- **Variance error** is the amount that the estimate of the target function will change if different training data was used.
    - High Variance: Machine learning algorithms that have a high variance are strongly influenced by the specifics of the training data.
    - Low Variance: Suggests small changes to the estimate of the target function with changes to the training dataset.

Bias vs. Variance

Bias and variance are 2 **sources of errors in** supervised learning problems.

A models goal is to **minimize the error**

- **Bias** is an erroneous assumption made by the algorithm
  - High bias can lead to missing relevant relations between features and targets (underfitting).
- **Variance** is an error from sensitivity to small fluctuations in the training set.
  - High variance can lead to modeling the random noise in the training data, rather than the intended outputs (overfitting).

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

Bias vs. Variance Trade-off

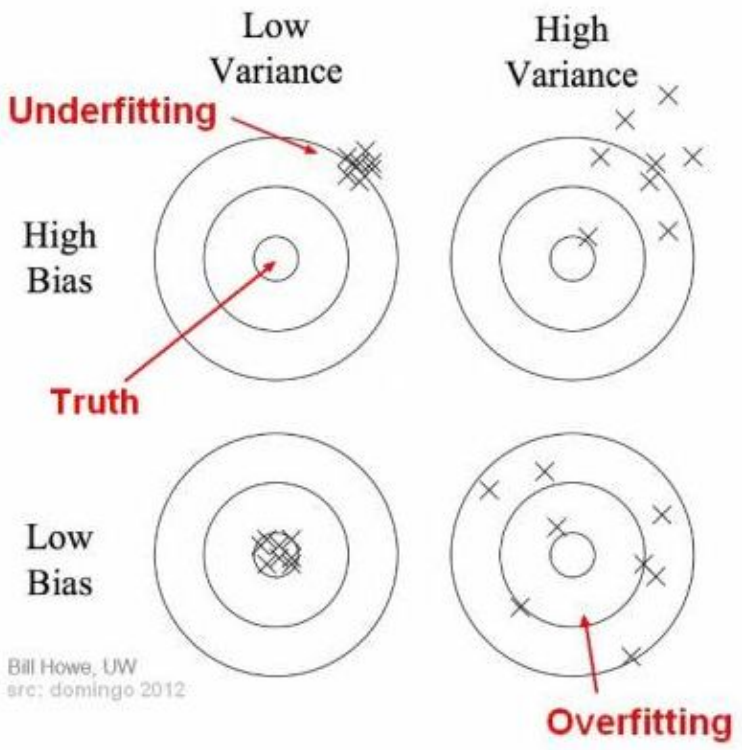
- Models with a **lower bias have a higher variance** across samples
- Models with a **higher bias have a lower variance** across samples

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

Bias vs. Variance Trade-off

Low Bias = High Variance

High Bias = Low Variance



https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

Bias vs. Variance Trade-off

Low Bias = High Variance

High Bias = Low Variance

- If our model is too simple and has very few parameters then it may have **high bias and low variance**.
- On the other hand if our model has large number of parameters then it's going to have **high variance and low bias**.
- This tradeoff in complexity is **why there is a tradeoff between bias and variance**. An algorithm can't be more complex and less complex at the same time.
- So we need to find the **right/good balance** without overfitting and underfitting the data.

https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

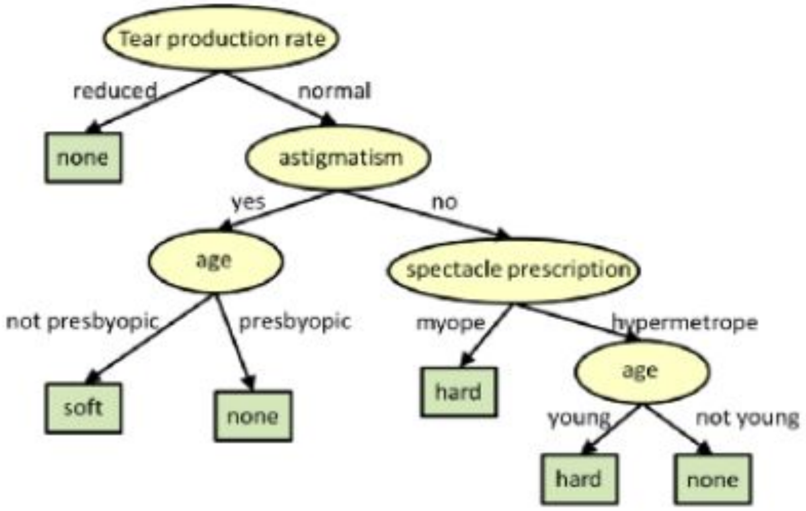
Pruning

How do we avoid overfitting in Decision Tree Learning?

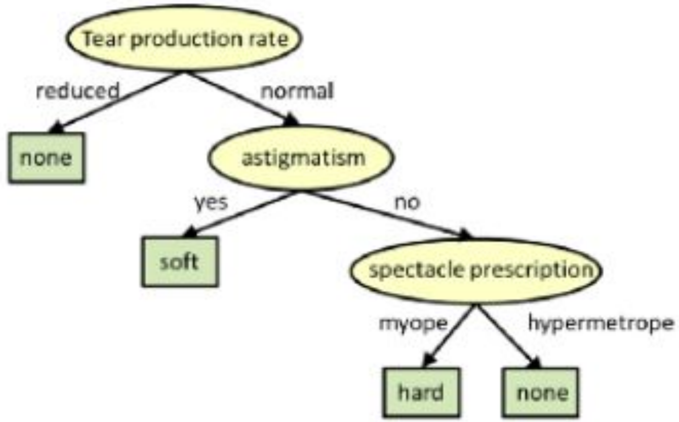
- **Pruning** is a technique used to deal with overfitting, that reduces the size of DTs by removing sections of the Tree that provide little predictive or classification power.
  - Reduces complexity of final classifier
  - Overcome overfitting → improve accuracy

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

Pruning Example



Original Tree



Pruned Tree

https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4

Pruning Strategies

- **Pre-pruning**
  - When you stop growing DT branches when information becomes **unreliable**.
- **Post-pruning**
  - In a fully grown DT, removing leaf nodes only if it results in a better model performance.

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14

Ensemble Learning

- **Ensemble learning** is a machine learning paradigm where multiple models (or "weak learners") are trained to solve the same problem and combined to **get better results**.
- The main hypothesis is that when **weak models are correctly combined** we can obtain more accurate and/or robust models.

https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Ensemble Learning

How to combine models?

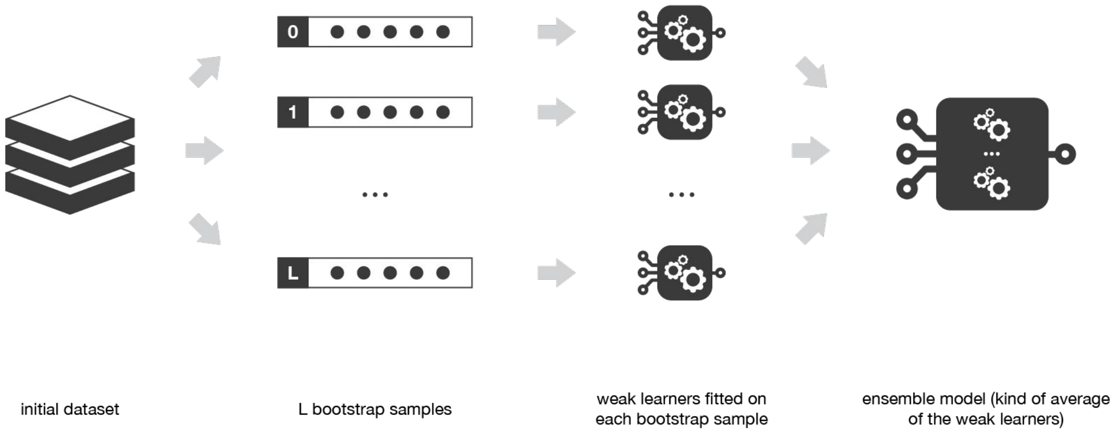
- **Bagging** considers homogeneous weak learners, learns them independently from each other and combines them following some kind of deterministic averaging process.
- **Boosting** considers homogeneous weak learners, learns them sequentially in a very adaptative way (a base model depends on the previous ones) and combines them following a deterministic strategy
- **Stacking** considers heterogeneous weak learners, learns them in parallel and combines them by training a meta-model to output a prediction based on the different weak models predictions

https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Ensemble Learning

Bagging

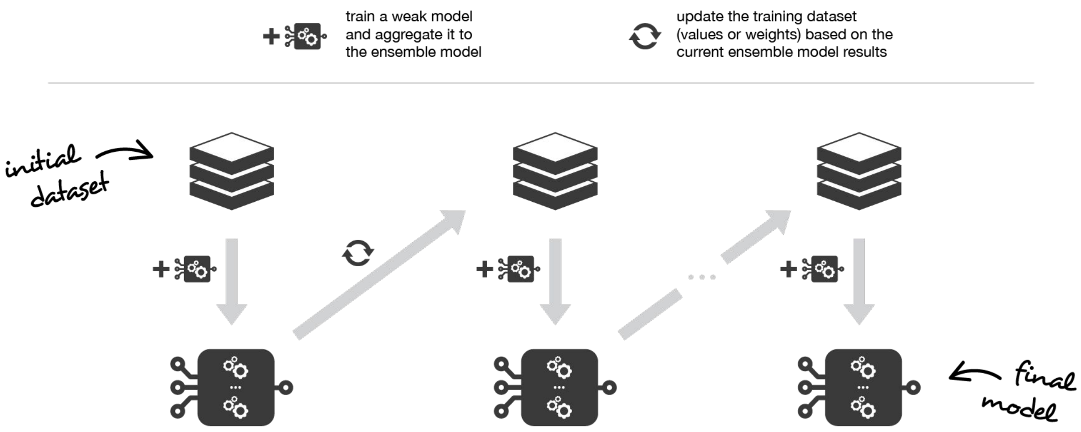
Bootstrap generates samples of size B from an dataset of size N by randomly drawing with replacement B observations.



https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Ensemble Learning

Boosting

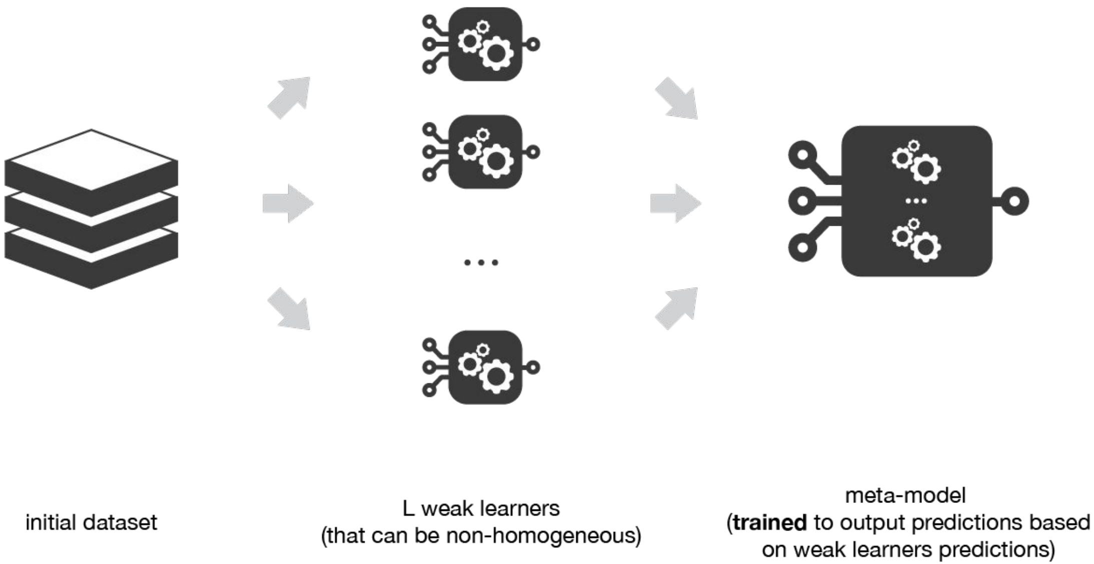


- **Adaboost** updates weights of the observations at each iteration.
  - Weights of **well classified observations decrease** relatively to weights of misclassified observations.
  - Models that perform better have higher weights in the final ensemble model.

https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

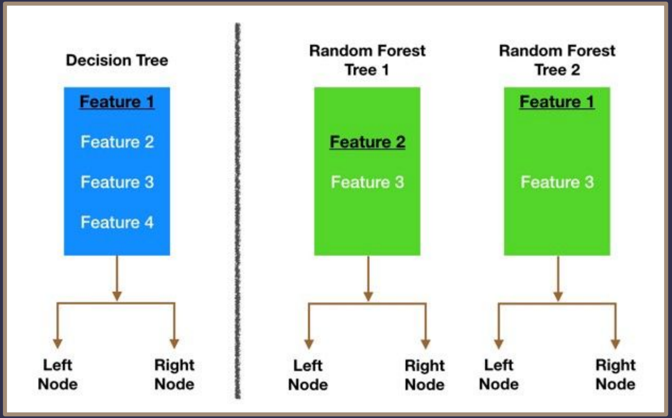
Ensemble Learning

Stacking



https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

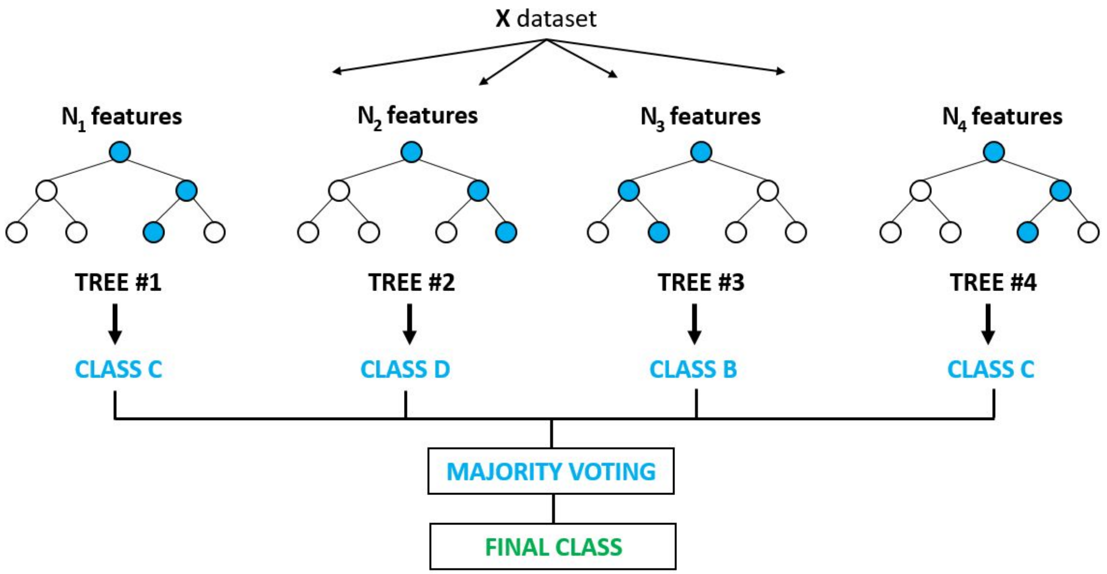
Random Forest



- **Random Forest** is an extension over **bagging**.
- It builds models on random subsets of data (bootstrapping)
  - Additionally, it also takes a **random selection of features** rather than using all to grow DTs.
- Each individual tree in the random forest outputs a class prediction and the **class with the most votes** becomes our model's prediction.

https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14
https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Random Forest



https://miro.medium.com/max/1170/1*58f1CZ8M4il0OZYg2oRN4w.png

Random Forest

Advantages

- This model **can estimate missing data**
  - maintains accuracy when large proportion of the data is missing
- Bagging **balances errors** in data sets where classes are imbalanced.
- **Powerful** due to the number of combinations it considers

https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Random Forest

Disadvantages

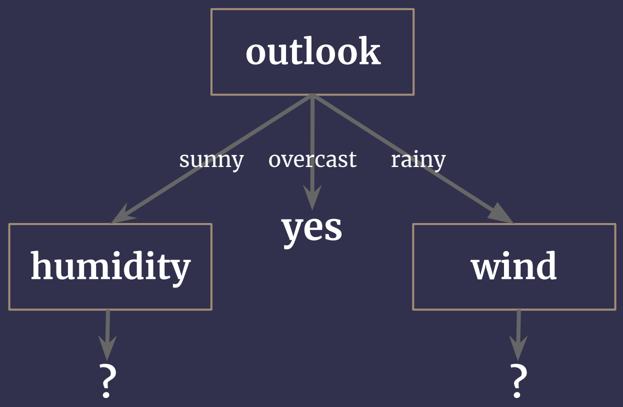
- **No interpretability**
  - Feels like a “black box” approach since we have very little control on what the model does
- **Time consuming** due to the number of combinations considers and computations it does.

Theory Recap

- **Decision Trees**
  - Entropy, Information Gain
  - Example
  - Advantages
  - Variations
  - Disadvantages
- **Overcoming Disadvantages**
  - Overfitting
  - Bias vs. Variance
  - Pruning
  - Ensemble Learning
- **Random Forest**

Google Colab Project

Homework #1



Complete the decision tree from the previous slide. Data sample listed below:

outlook	temperature	humidity	windy	play
overcast	cool	normal	TRUE	yes
overcast	hot	high	FALSE	yes
overcast	hot	normal	FALSE	yes
overcast	mild	high	TRUE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
rainy	mild	high	FALSE	yes
rainy	mild	high	TRUE	no
rainy	mild	normal	FALSE	yes
sunny	cool	normal	FALSE	yes
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
sunny	mild	high	FALSE	no
sunny	mild	normal	TRUE	yes

Homework #2

Note: The dataset has already been split into train and test

Try to solve an end-to-end project on the Titanic dataset. This table has multiple columns which can be used to predict if a passenger survived or not.

For your ease, we have uploaded the titanic dataset to our github and you can use the below URL as the parameter to read_csv to load the dataset in Google Colab

https://github.com/WomenWhoCode/WWCodeDataScience/tree/master/Intro_to_MachineLearning/data/titanic

https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/

Homework #3

Note: You have to download these datasets from links and host it on your Google Drive. This video will help you get set up!

Our leaders have curated 2 other datasets you can try out your classification skills on.

Dataset	Link	Task Description
Wids 2020 hospital Dataset	https://www.kaggle.com/c/widsdatathon2020/data	This dataset has patient details such as heart rate, temperature and few other essential measurements from 1st 24 hours in a Intensive care unit. The task is to build a model to predict patient survival
Credit card fraud Dataset	https://www.kaggle.com/mlg-ulb/creditcardfraud	This dataset has credit card transaction details which have been anonymized using PCA. The task is to identify fraudulent transactions from others

See you next week!

Questions?

Join us on slack (bit.ly/wwcodedatascience-slack) and post it on our **#help-me** channel.

Register?

Register for all sessions at linktr.ee/wwcodedatascience_registration