Machine Learning Foundations

Lab 3

Today’s Agenda

Icebreaker (15 mins)

Week 3 Overview + Q&A (30 mins)

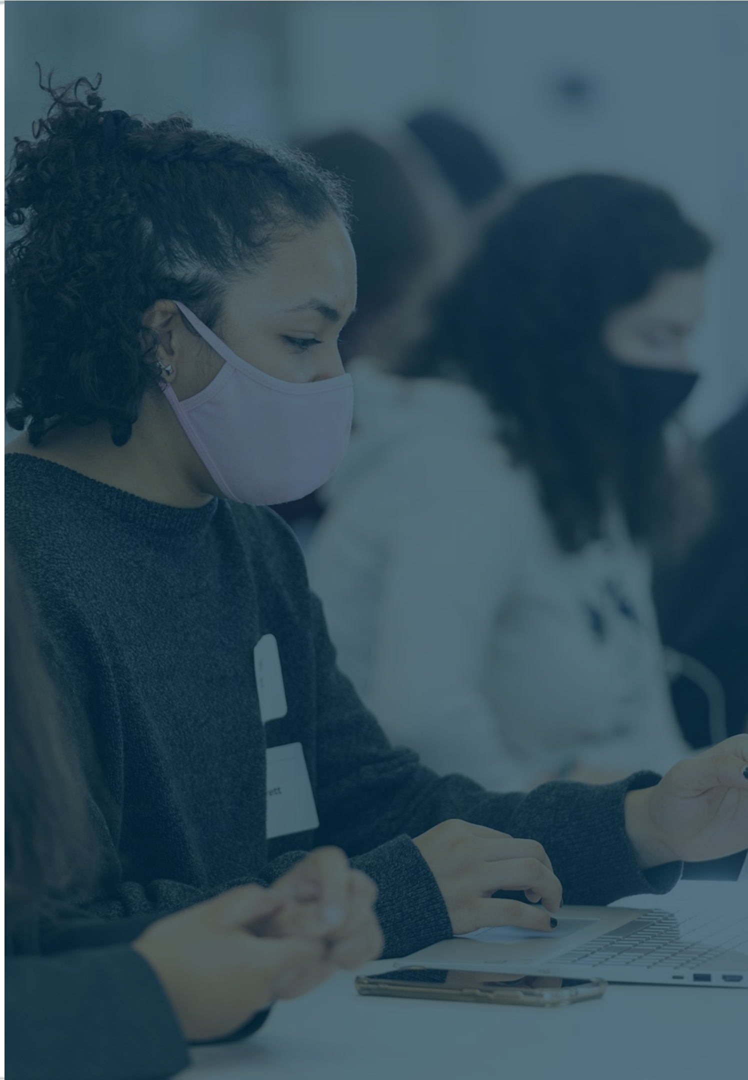
Breakout Group: Big Picture Questions (20 mins)

Class discussion (10 mins)

Break (10 mins)

Breakout Groups: Lab Assignment Working (80 mins)

Concluding Remarks and Survey (15 mins)



Icebreaker: “Pride Month”

Icebreaker: Pride Month

Objectives:

- Acknowledge the contributions of marginalized folks committed to inclusion and tech equity
- Share something about yourself that you're proud of
- Recognize your peers' strengths and accomplishments

Icebreaker: Pride Month

Dr. Timnit Gebru – co-founder of Black in AI and founder of the Distributed Artificial Intelligence Research Institute

Os Keyes – trans computer scientist and researcher who focuses on how recognition systems deal with gender/race, the role of science in authenticating trans existences, and the implications that AI has for disability/autism

AI can be used for good and bad! (video on next page.)

Icebreaker: Pride Month

Icebreaker: Pride Month

Students: Turn to your neighbor and discuss the “gaydar” video (thoughts, opinions, comments, concerns) and/or share what makes you proud.

Week 3 Overview + Q&A

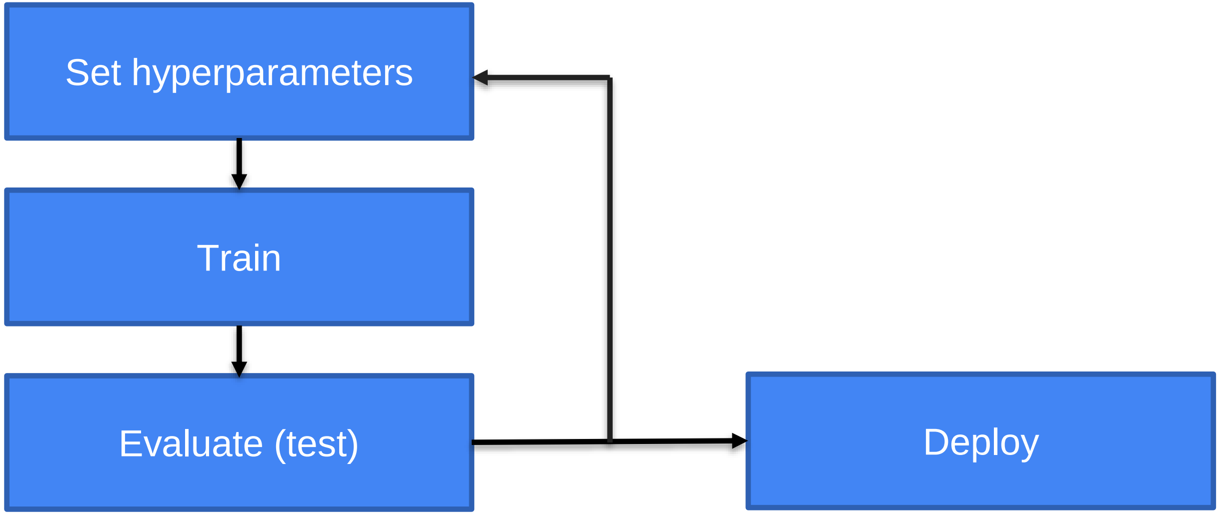
Week 3 Overview

This week you explored a number of topics. To refresh your memory, your goals were to:

- Define the core foundational elements of model training and evaluation
- Develop intuition for different classes of algorithms.
- Analyze the mechanics of two popular supervised learning algorithms: decision trees and k-nearest neighbors.
- Develop intuition on trade-offs between different algorithmic choices.

Hyperparameters

Hyperparameters are the settings of an algorithm that can be adjusted to optimize performance.



Set hyperparameters

Train

Evaluate (test)

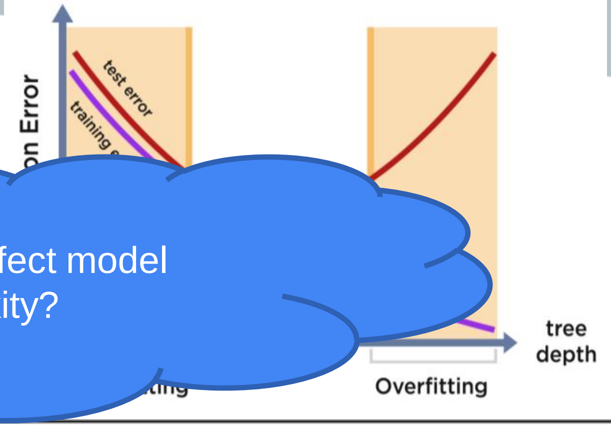
Deploy

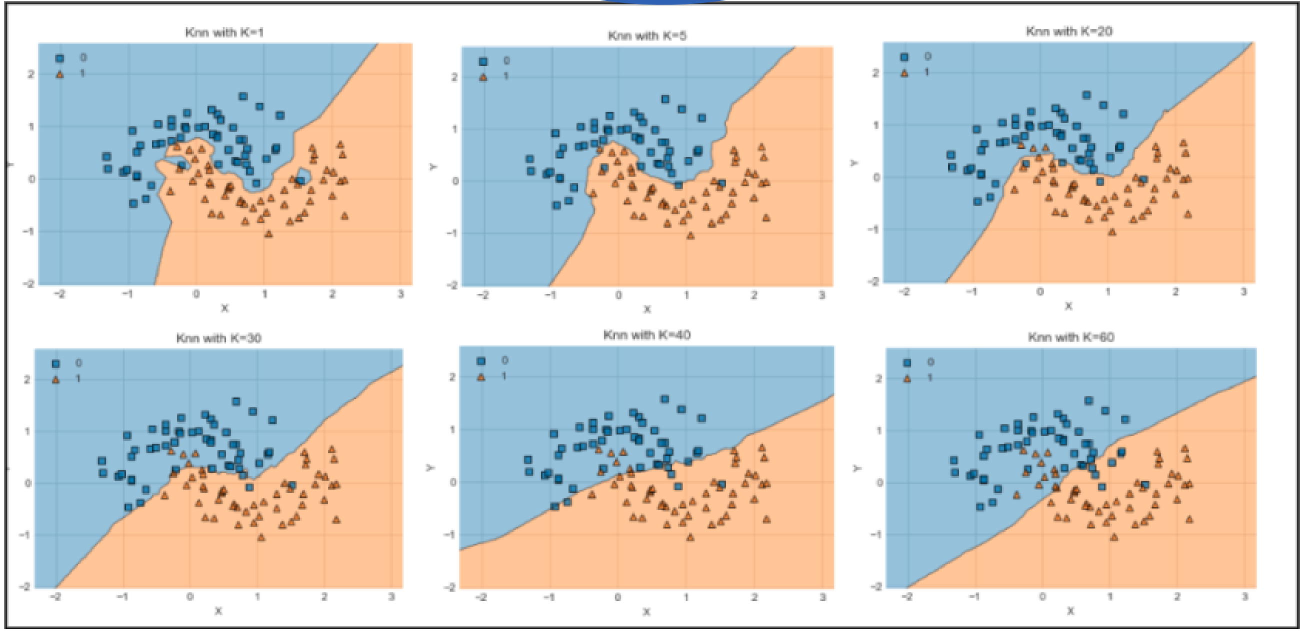
What are the hyperparameters of KNN and Decision Trees?

Model Complexity

- A model that overfits is t
- A model that under

How does K affect model complexity?





Model complexity versus model generalizability

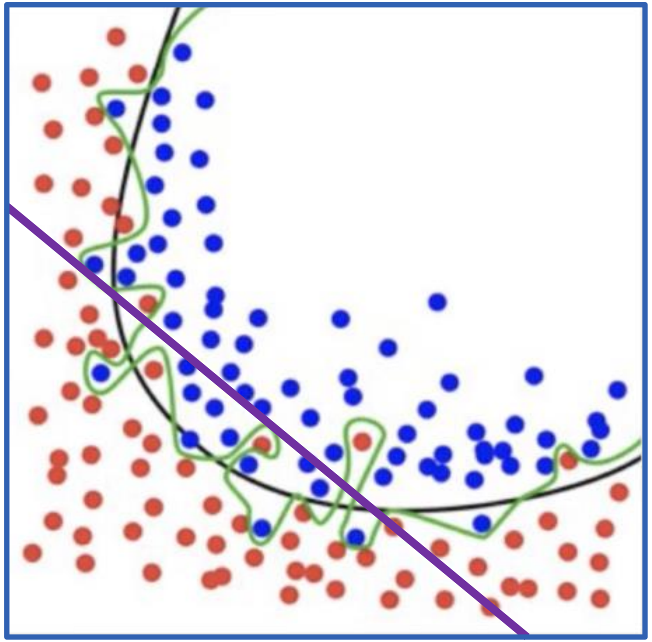
Q1: Which model has a smaller training loss?
(a) black line
[✓ (b) green line]
(c) purple line

Q2: Which model has a better generalizability?
[✓ (a) black line]
(b) green line
(c) purple line

Q3: Which model has a high complexity?
(a) black line
[✓ (b) green line]
(c) purple line

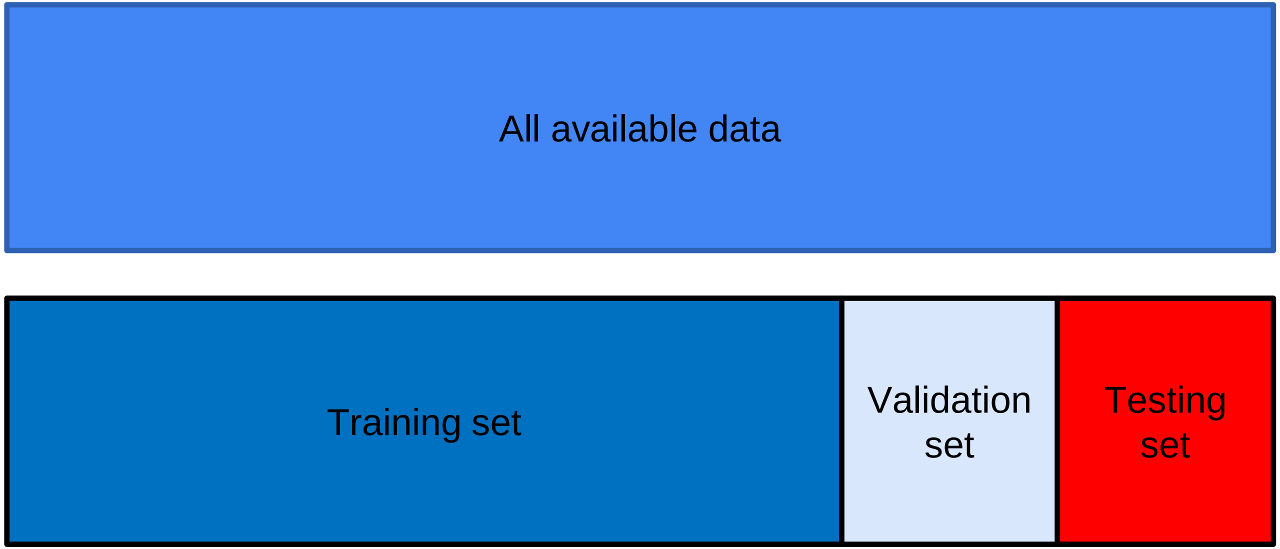
Q4: Is the green line overfitting or underfitting the training data?
[✓ (a) Overfitting]
(b) Underfitting

Q5: Is the purple line overfitting or underfitting the training data?
(a) Overfitting
[✓ (b) Underfitting]

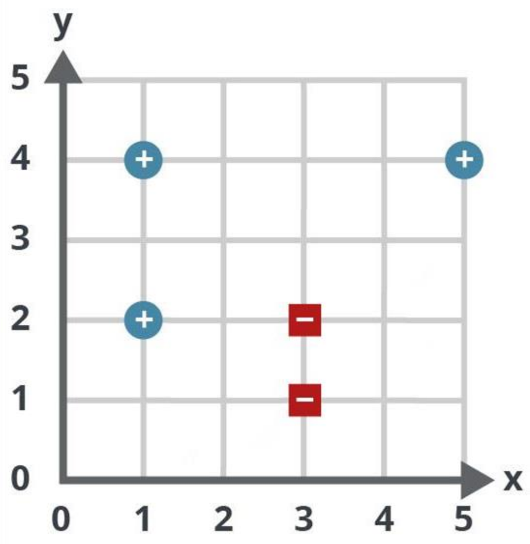


Training, validation, and testing set

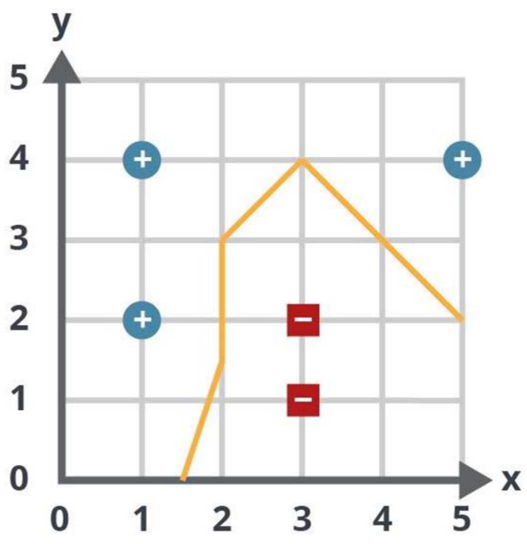
- Training set: Given a training dataset, learn a model
- Validation set: Using a separate validation dataset to do model selection (hyperparameter tuning)
- Testing set: Apply the model to a test dataset, to evaluate the performance of the model



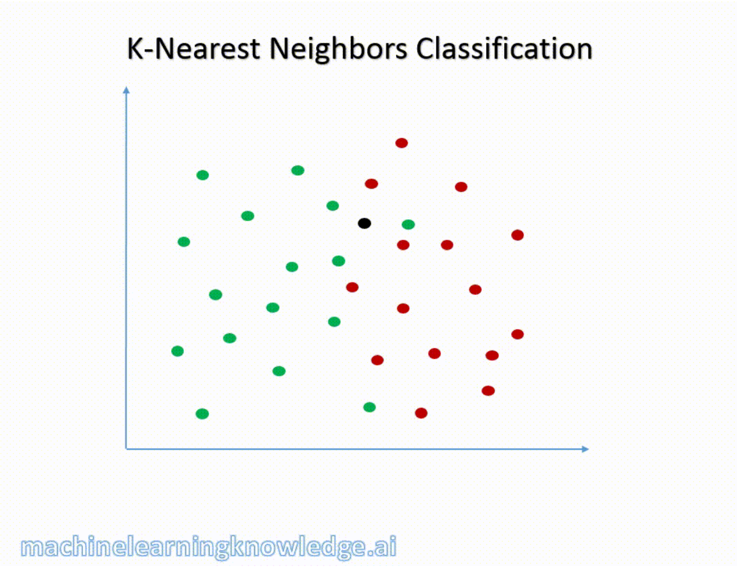
K-Nearest Neighbors (K = 1) Boundaries



Original data



For KNNs, we can answer the question: for any point in space, to which class does that point belong?

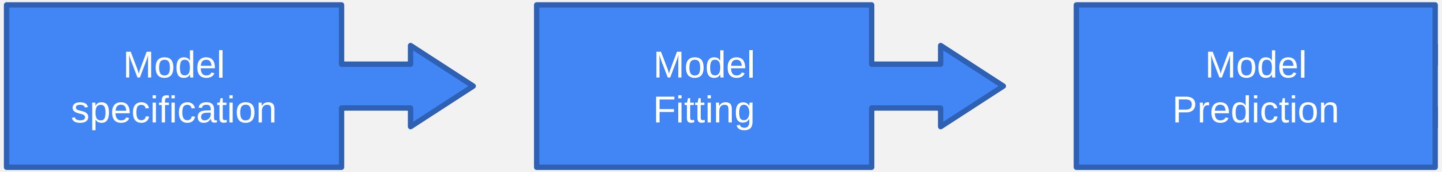


K-Nearest Neighbors Classification

machinelearningknowledge.ai

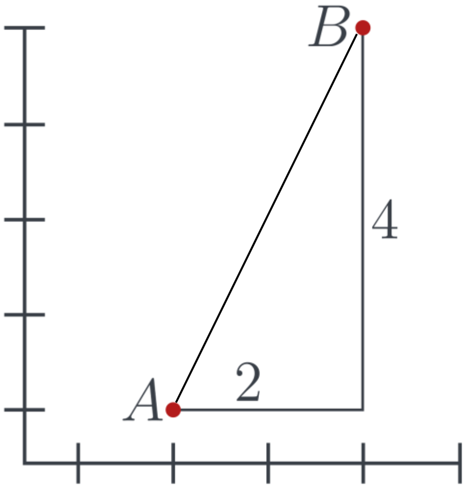
K-Nearest Neighbors in code

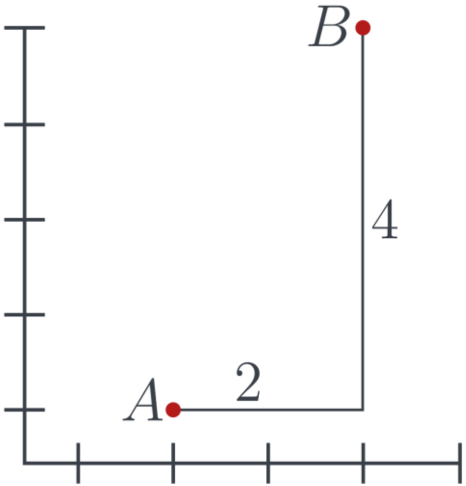
Scikit-learn’s 3 Steps



```
% Initialize the model
model = KNeighborsClassifier(n_neighbors = 3)

% Train the model using training sets
model.fit(X_train, y_train)

# Make predictions on the test set
prediction = model.predict(X_test)

# Compute and print model’s accuracy score
score = accuracy_score(y_test, prediction)
Print('Accuracy score of the model: ' + str(score))
```

Euclidean versus Manhattan distance





Euclidean distance:

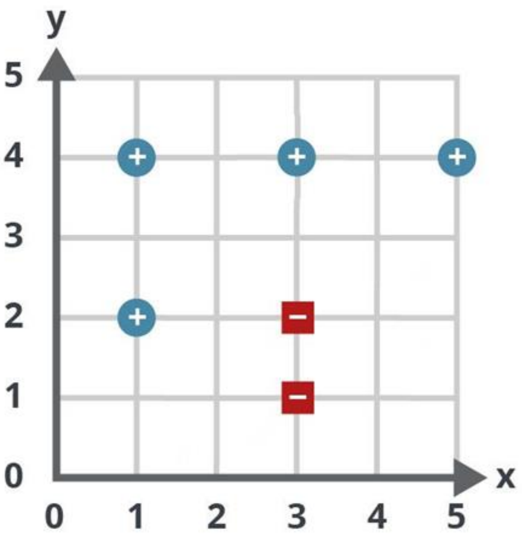
$$d(A,B) = \sqrt{\sum_{i=1}^{n}(x_i^b - x_i^a)^2} = \sqrt{2^2 + 4^2}$$

Manhattan distance:

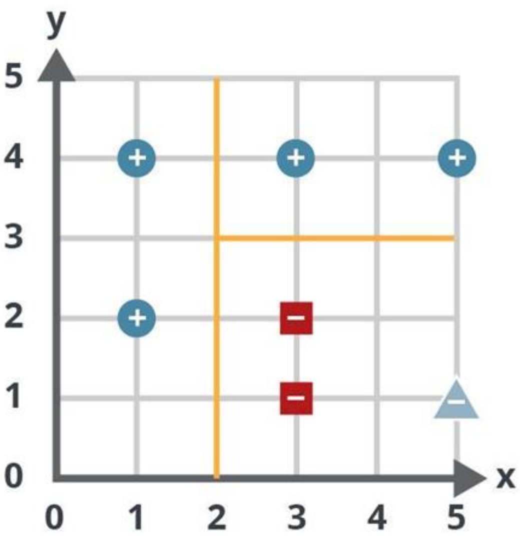
$$d(A,B) = \sum_{i=1}^{n} \left| x_i^b - x_i^a \right| = 2 + 4$$

Give examples of when you would use Euclidean distance and when you would use Manhattan distance

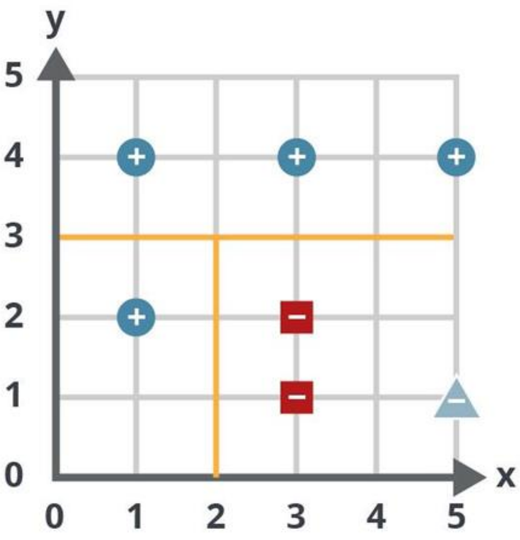
Decision Trees: Boundaries



Data set

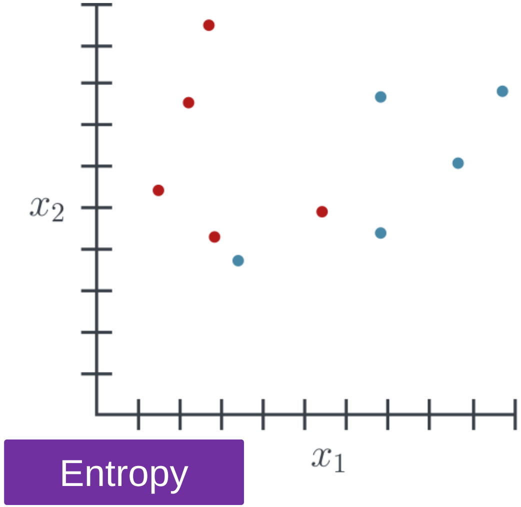


Split x > 2 , y > 3



Split y > 3 , x > 2

Training a decision tree



Entropy

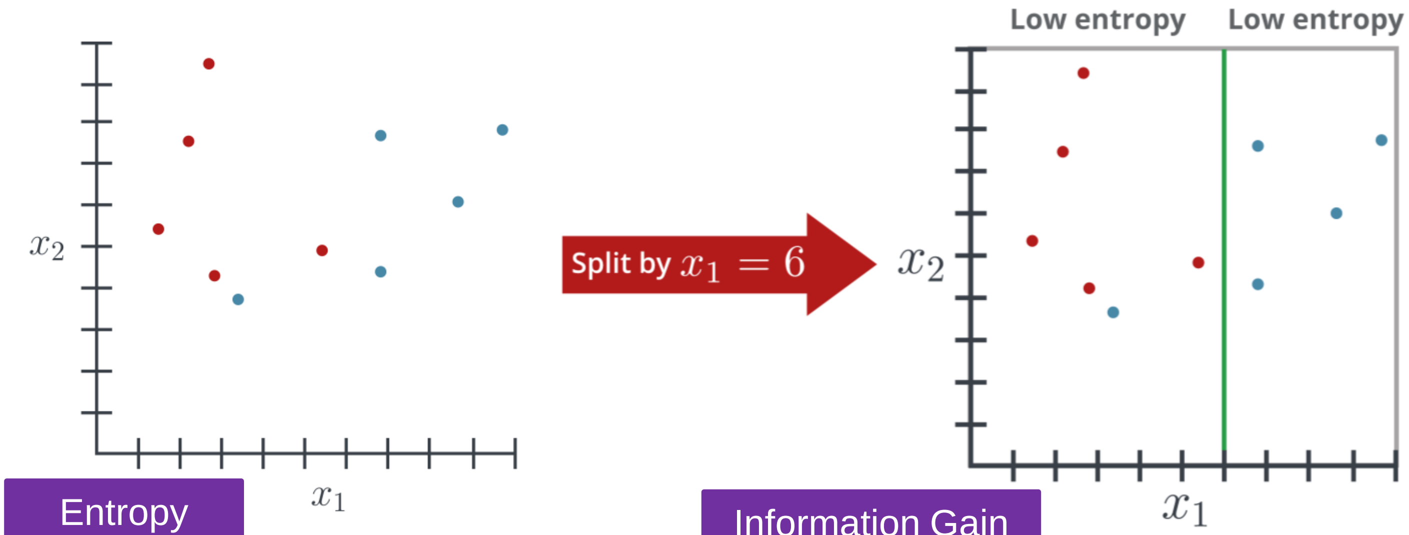
$$H(y) = -\sum_{i=1}^{n} P(y_i)\log_2(P(y_i))$$

$$= -[P(red)\log_2 P(red) + P(blue)\log_2 P(blue)]$$

$$= -\left[\frac{5}{10}\log_2\left(\frac{5}{10}\right) + \frac{5}{10}\log_2\left(\frac{5}{10}\right)\right]$$

$$= 1$$

Training a decision tree



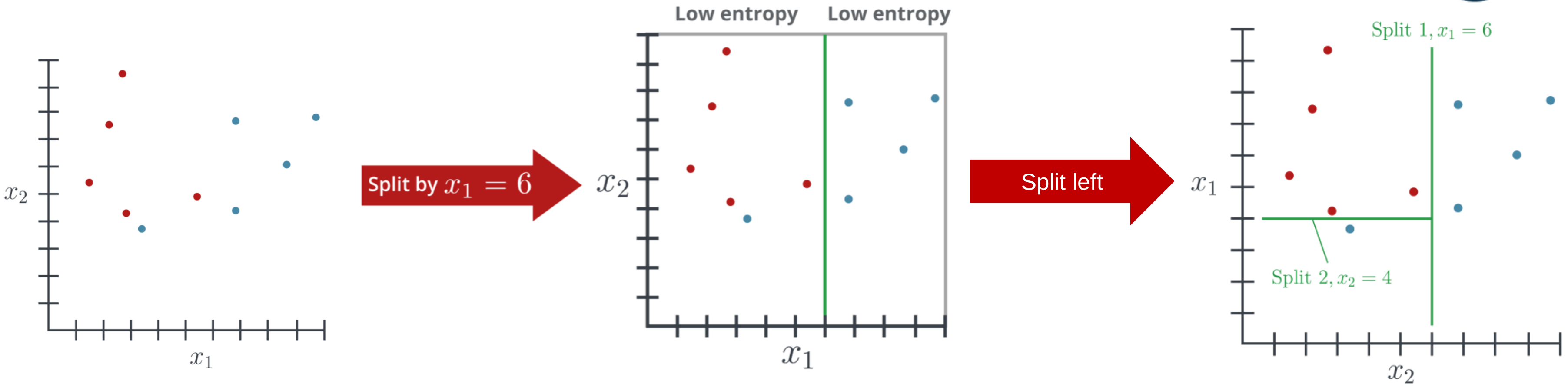
Entropy

$$H(y) = -\sum_{i=1}^{n} P(y_i)\log_2(P(y_i))$$
$$= 1$$

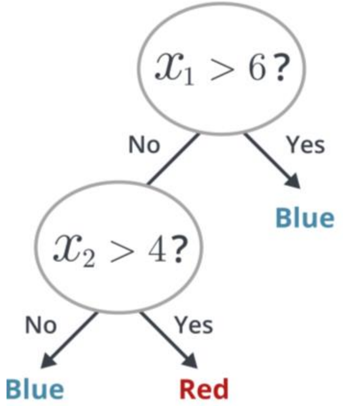
Information Gain

$$IG = H(y|parent) - \sum_{c \in C} r(c)H(y|c)$$
$$= 1 - \frac{6}{10}\left(-\left[\left(\frac{1}{6}\right)\log_2\frac{1}{6} + \left(\frac{5}{6}\right)\log_2\frac{5}{6}\right]\right) - \frac{4}{10}\left[-\frac{4}{4}\log_2\frac{4}{4}\right]$$
$$\approx 1 - 0.39 - 0$$
$$IG \approx 0.61$$

Training a decision tree

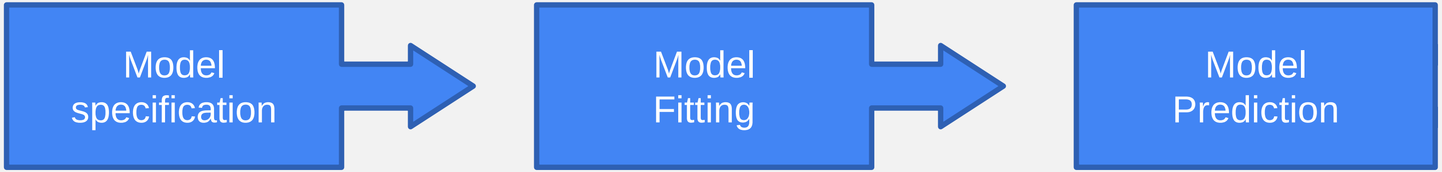


Final decision tree!



Decision Trees in code

Scikit-learn’s 3 Steps



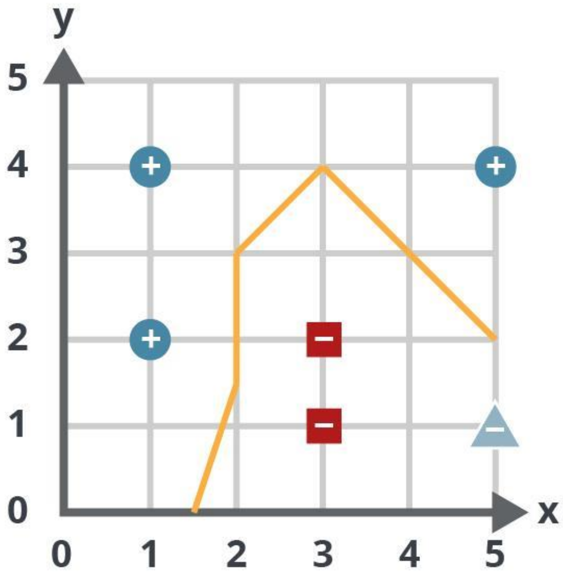
Model specification

Model Fitting

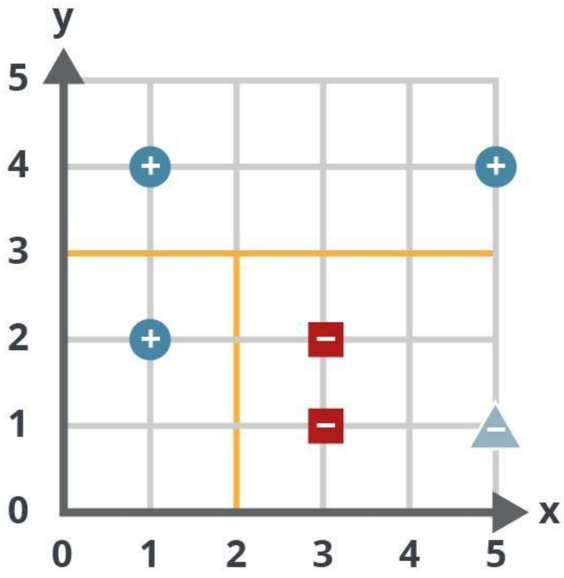
Model Prediction

```
% Initialize the model
model = DecisionTreeClassifier(criterion = crit,
                               max_depth = 5)

% Train the model using training sets
model.fit(X_train, y_train)

# Make predictions on the test set
prediction = model.predict(X_test)

# Compute and print model’s accuracy score
score = accuracy_score(y_test, prediction)
Print('Accuracy score of the model: ' + str(score))
```

KNN and Decision Trees



KNN, K=1 (1-NN)

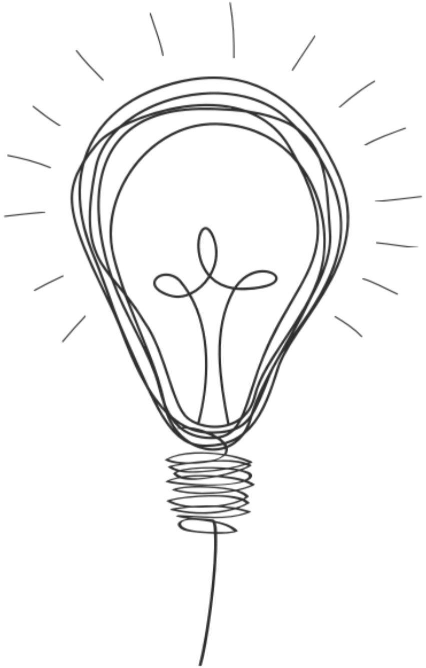


Decision Tree: y > 3, x > 2

KNN and Decision Trees can be used for classification problems and regression problems!

Questions & Answers

What questions do you have about the online content this week?



Breakout Groups: Big Picture Questions

Big-Picture Questions

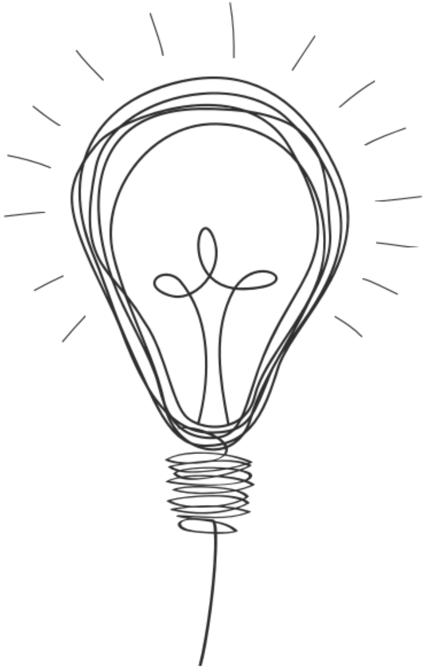
You have 20 minutes to discuss the following questions within your breakout groups:

- How would you explain model complexity to a non machine learning person?
- What are the hyperparameters in KNN and decision trees? How do they impact the respective model's complexity?
- In less-technical terms, why do you think KNN and decision trees work? In other words, what is special about them that enables them to make accurate predictions on new data?
- How can you tell if a model is overfitting the data?
- How can you tell if a model is underfitting the data?

Class Discussion

Class Discussion: Responses to Big Picture Questions

Let’s hear your classmates’ responses.



Break! (10 minutes)

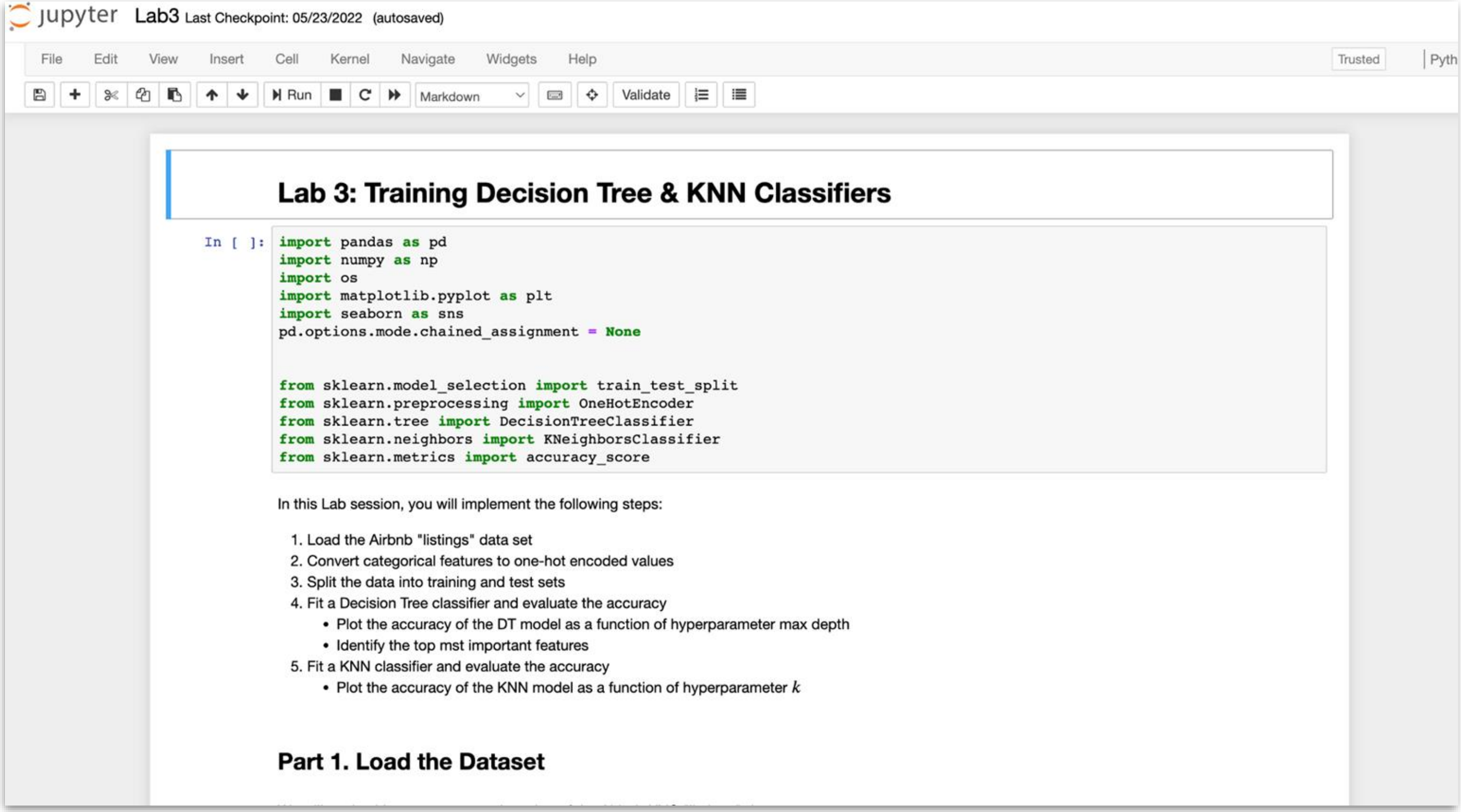
Breakout Groups:
Lab Assignment

Lab 3

In this lab, you will:

- Convert categorical features to one-hot encoded values.
- Train decision tree classifiers with various hyperparameter values.
- Train KNN classifiers with various hyperparameter values.
- Visualize the models' accuracies.

Lab 3



Lab 3: Training Decision Tree & KNN Classifiers

In [ ]:

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
pd.options.mode.chained_assignment = None


from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

In this Lab session, you will implement the following steps:

1. Load the Airbnb "listings" data set
2. Convert categorical features to one-hot encoded values
3. Split the data into training and test sets
4. Fit a Decision Tree classifier and evaluate the accuracy
    - Plot the accuracy of the DT model as a function of hyperparameter max depth
    - Identify the top mst important features
5. Fit a KNN classifier and evaluate the accuracy
    - Plot the accuracy of the KNN model as a function of hyperparameter k

Part 1. Load the Dataset

Working Session Debrief

Lab Debrief

- What did you enjoy about this lab?
- What did you find hard about this lab?
- What questions do you still have about this lab?
- How did you approach problem-solving during the exercise?
- What would you do differently if you were to repeat the exercise?

Concluding Remarks

Concluding Remarks

- You want the right balance of complexity in your models to avoid overfitting and underfitting (hyperparameter tuning)
- Know how to describe KNN and Decision Trees
- ML/AI can be use for good, but can also be used for bad...

Next week

In the following week, you will:

- Analyze the mechanics of logistic regression
- Understand the purpose of using gradient descent and loss functions
- Explore common hyperparameters for logistic regression
- Define the core math concepts required to solve common machine learning problems
- Use NumPy to perform vector and matrix operations
- Explore how linear regression works to solve real world regression problems

And in the lab, you will:

- Load and split the data into training and test sets
- Write a Python class that will train a logistic regression model
- Compare your implementation to scikit-learn's implementation

Content + Lab Feedback Survey

Weekly Survey + Early Program Survey

To complete your lab, please answer the following questions about BOTH your online modules and your lab experience. Your input will help pay it forward to the Break Through Tech student community by enabling us to continuously improve the learning experience that we provide to our community.

Thank you for your thoughtful feedback!

Weekly Content + Lab feedback: https://forms.gle/xdfN3Vy1BYMUHFvu8

Early-Program Feedback: https://forms.gle/ZypUgtAGAtzGDREK9