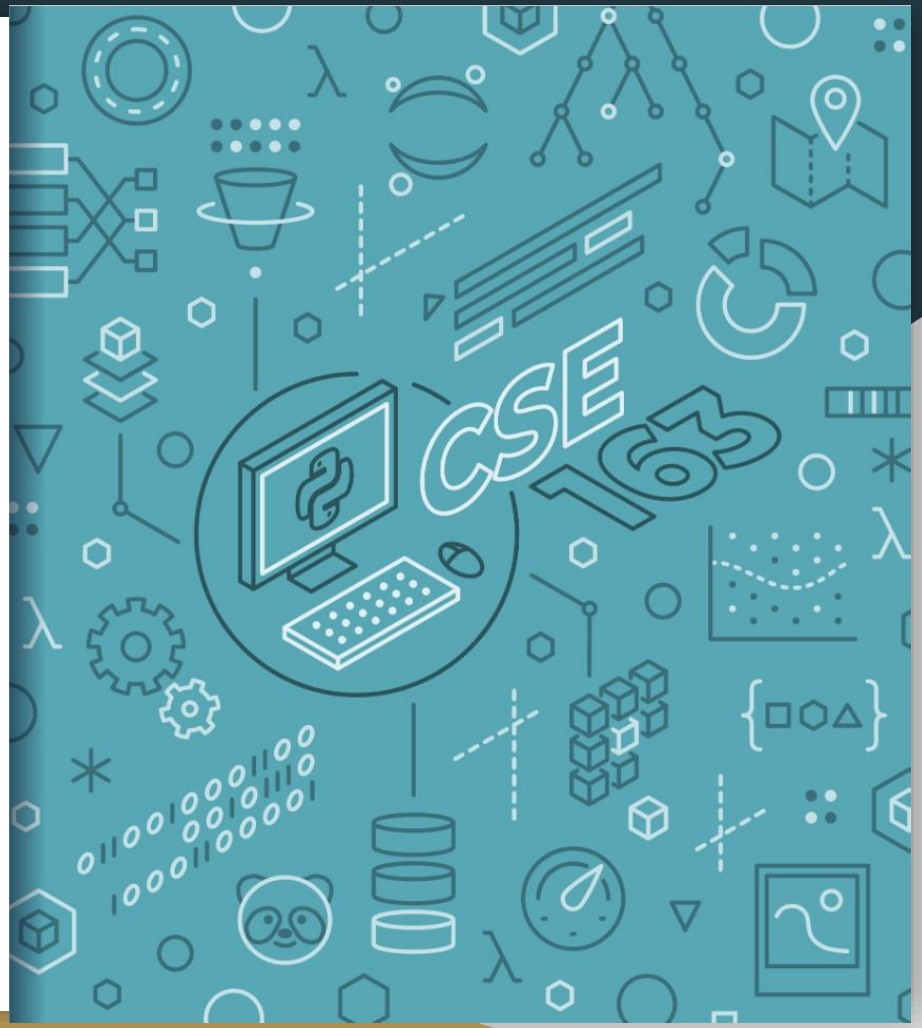
CSE 163
Model Evaluation

Adrian Salguero
Spring 2026

💭 Icebreaker (discuss with neighbors): ‘
Which fictional creature would you like as a pet?
Add to our Slido!

slido.com
#cse163



Announcements

- **Take Home Assessment 4: Education** due tomorrow at 11:59pm!
- **Peer Reviews for THA 3** due tonight at 11:59pm!
- **Lesson 16 Canvas Quiz** due tonight at 11:59pm!
- **Reading Assignment 4** now available, due May 11th at 11:59pm!
- **Project Part 2** now available, due May 14th at 11:59pm!
  - EDA/Milestone
  - Group Projects: only one person needs to submit but add your teammates to your submission using these instructions

One-Hot Encoding

- Mapping categories to numerical numbers can cause unintended consequences
- Transform categorical column into multiple columns of binary values (either 0 or 1)

	age	gender
0	22	male
1	38	female
2	26	non-binary
3	35	female
4	35	male

Categories → Numbers

	age	gender
0	22	1
1	38	2
2	26	3
3	35	2
4	35	1

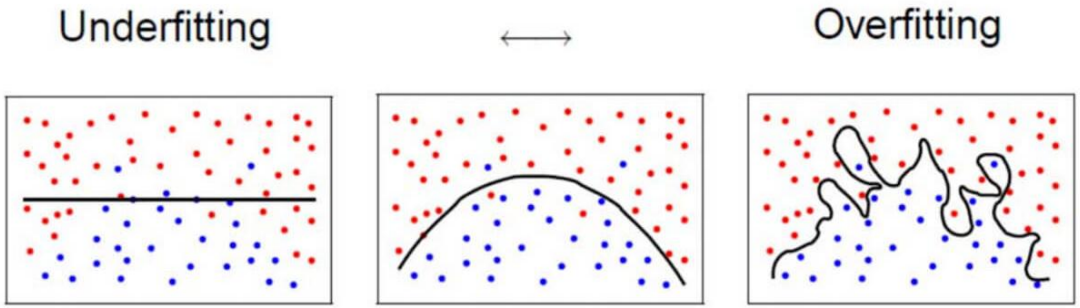
Categories → One-Hot Encoding

	age	gender_female	gender_male	gender_non-binary
0	22	0	1	0
1	38	1	0	0
2	26	0	0	1
3	35	1	0	0
4	35	0	1	0

Overfitting

- ***Overfitting***: when your model matches the training set so well, that it fails to generalize
  - *Example: Memorizing answers on a multiple choice test*

- Tall trees are likely to overfit if you don't have enough data
  - *Can learn very complex boundaries*
  - *Very few points at the leaves*



Assessing Performance

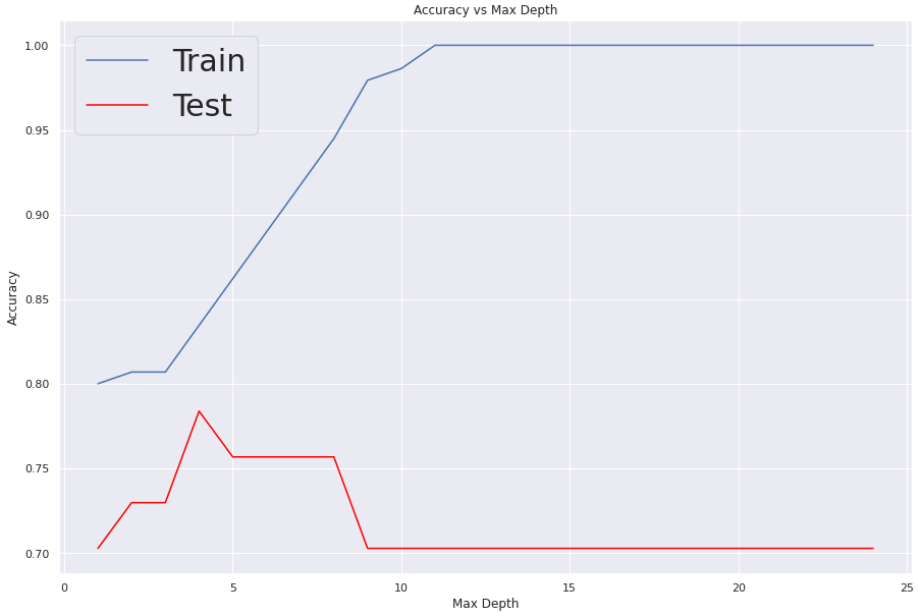
- Why is it bad practice to evaluate your data on your training set?
- The purpose of your models are to ***generalize to new data***
  - *Data your model has not seen*
- Set aside a ***test set*** to evaluate your model
- Never train or make decisions based on your test set!

Classification ML Pipeline (w/ categories)

```
# Separate data
features = data.loc[:, data.columns != 'target']
features = pd.get_dummies(features)
labels = data['target']
# Train/test split
feat_train, feat_test, lab_train, lab_test = \
   train_test_split(features, labels, test_size=0.2)
# Create and train model on train set
model = DecisionTreeClassifier()
model.fit(feat_train, lab_train)
# Predict on test data
predictions = model.predict(feat_test)
accuracy_score(lab_test, predictions)
```

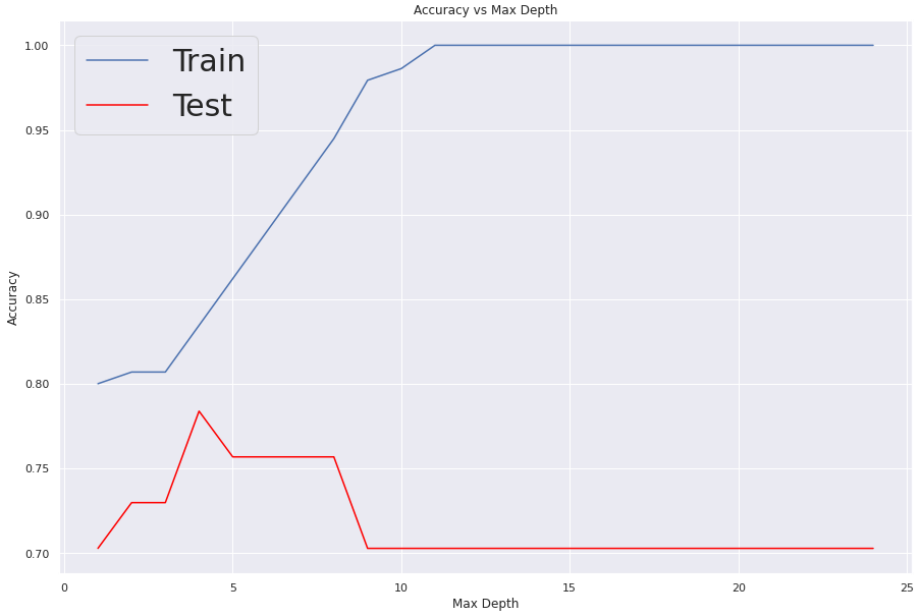
Model Complexity

- One hyperparameter to control complexity of decision tree is the max depth (or height) of tree

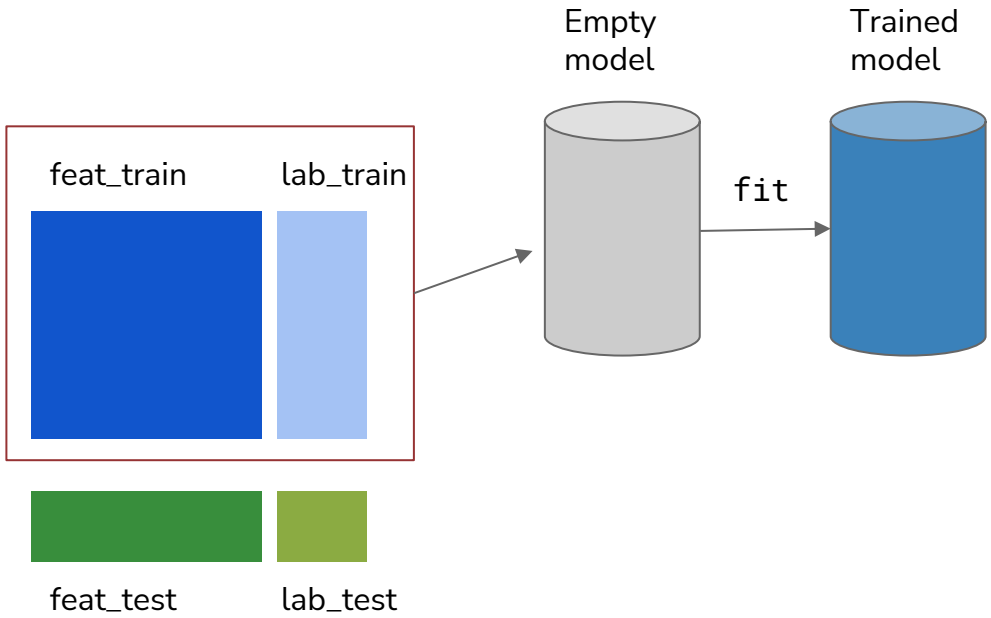


Model Complexity

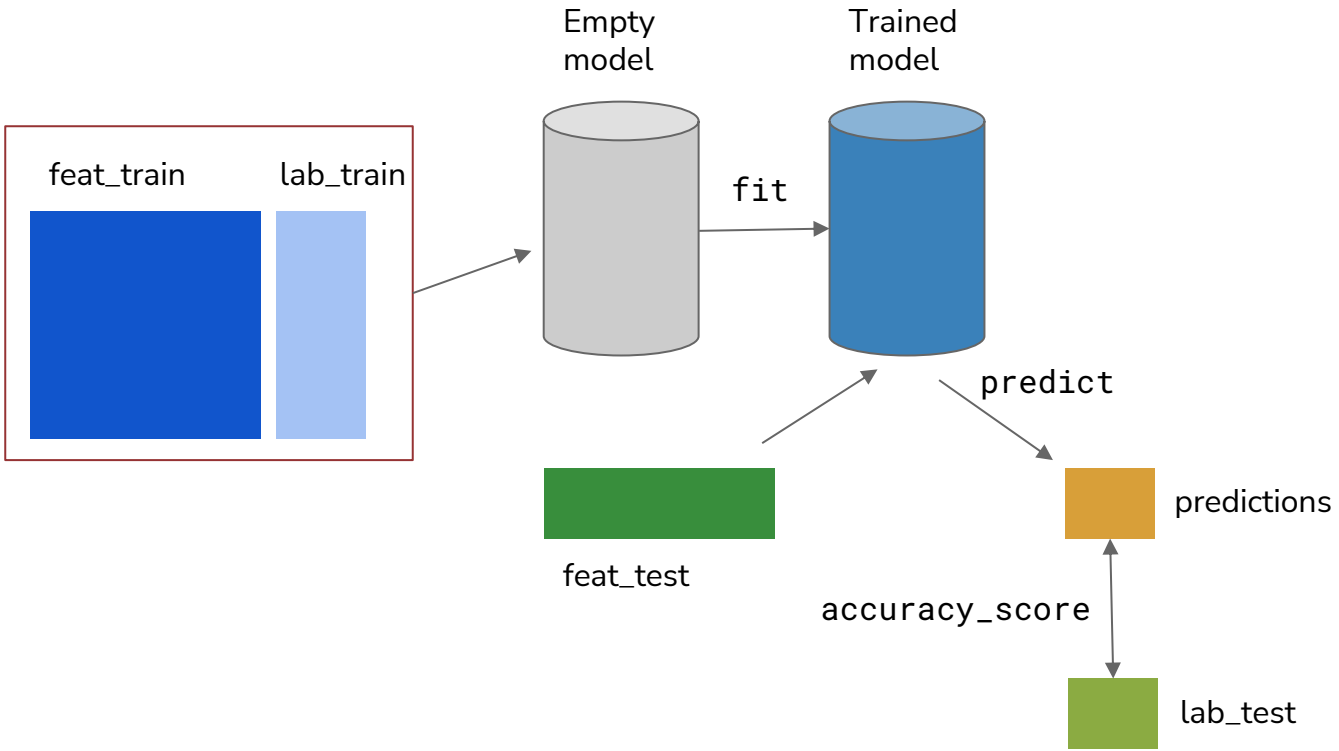
- One hyperparameter to control complexity of decision tree is the max depth (or height) of tree



Visualize the split



Visualize the split



Ground Rules

- Talking about social impact and ethics in data science can be challenging since its effects can be deeply personal or harmful
  - *Many reasonable people have differing opinions on how to draw the line between okay/not okay, there isn't always an easy yes/no answer*

- Productive Discussions
  - *Listen with intention to understand first and forming an opinion only after you fully understand*
  - *Take responsibility for the intended and unintended effects of your words and actions on others*
  - *Mindfully respond to others' ideas by acknowledging the unique value of each contribution*

Discussion – Lesson Quiz

1. Consider the case of our credit card churn predictor. Suppose we were using it in our first case use of predicting whether a current customer is likely to churn, and if they are, **provide them with special offers to incentivize them to stay.**

2. Consider the case of our credit card churn predictor. Suppose we were using it in our second use case of predicting whether a new customer is likely to churn or not, and if they are, **don't provide them with a credit card in the first place.**

Would you endorse using either system? Why or why not? Justify what concerns you might have about either system or why you think some potential concerns do not outweigh the benefit of the model.