Applied Machine Learning in R

Pittsburgh Summer Methodology Series

Lecture 1-A July 19, 2021

Workshop Overview

Jeffrey Girard

Assistant Professor
University of Kansas

Research Areas

- Affective Science
- Clinical Psychology
- Computer Science

Machine Learning

- Recognition of Facial Expressions
- Prediction of Emotional States
- Prediction of Mental Health Status

www.jmgirard.com
jmgirard@ku.edu
@jeffreymgirard

Shirley Wang

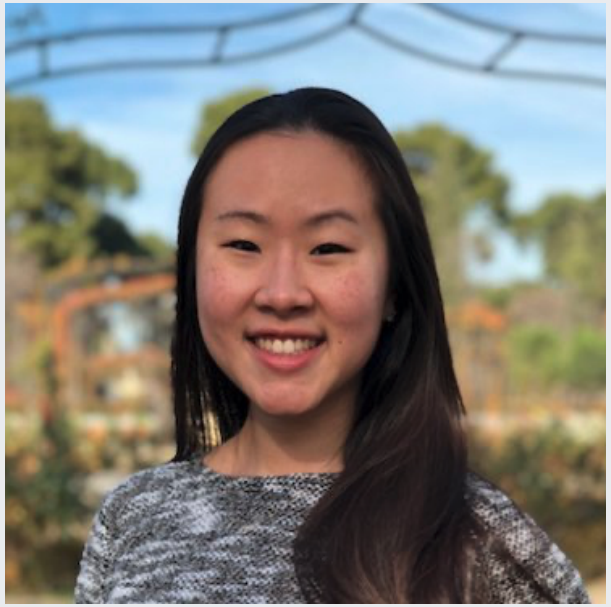
Doctoral Candidate
Harvard University

Research Areas

- Clinical Psychology
- Computational Psychiatry
- Mathematical Modeling

Machine Learning

- Prediction of suicide risk
- Prediction of longitudinal illness course
- Idiographic prediction



shirleywang.rbind.io
shirleywang@g.harvard.edu
@ShirleyBWang

Goals and Timeline

Build a foundation of concepts and skills

Describe every step from start to finish

Emphasize **practical and applied** aspects

Provide intuitions rather than lots of theory

Dive deeper into a few algorithms

Highlight algorithms good for beginners

Communicate the pros and cons of choices

Day	Topic	Lead
1-A	Conceptual introductions	JG
1-B	Logistics and data exploration	SW
2-A	Feature engineering	JG
2-B	Basic predictive modeling	JG
3-A	Regularized regression	SW
3-B	Decision trees and random forests	SW
4-A	Support vector machines	JG
4-B	Practical issues and advice	SW
5-A	Panel Q&A and discussion	Both
5-B	Hackathon and consultation	Both

Format and Materials

Each workshop day will have **two parts**

Most parts will have **lecture** and **live coding**

Most parts will have hands-on **activities**

We will take a **~10m break** after the first part

Course materials are on Github and OSF

- github.com/ShirleyBWang/pittmethods_ml
- osf.io/3qhc8

A few inspirations for this workshop include Applied Predictive Modeling, Tidy Modeling with R, and StatQuest.

Etiquette and Responsibilities

Behave professionally at all times

Stay on topic and **minimize distractions**

Stay muted unless talking to minimize noise

Ask **questions** in chat or use "Raise Hand"

Be **respectful** to everyone in the workshop

Be **patient** with yourself and others

You have the right to:

- Be **treated with respect** at all times
- Turn your **camera on or off**
- **Arrive and depart** whenever needed
- **Ask for help** with workshop content
- **Share your opinions** respectfully
- **Reuse materials** according to the license
- Receive **reasonable accommodations**
- **Contact the instructors** by email

Icebreakers

Icebreakers

We will randomly assign everyone to one of two breakout rooms

Each person will have *up to one minute* to introduce themselves

In addition to **sharing your name**, please answer these questions:

1. Where are you joining us from?
2. What field(s) do you work in?
3. What is one of your research interests?
4. What is one of your personal interests?

The instructor will go first and call on attendees to go next

If you would prefer not to share, please indicate that in chat

Conceptual Introduction

What is machine learning?

The field of machine learning (ML) is a **branch of computer science**

ML researchers **develop algorithms** with the capacity to **learn from data**

When algorithms learn from (i.e., are **trained on**) data, they create **models**[1]

This workshop is all about applying ML algorithms to create **predictive models**

The goal will be to **predict unknown values** of important variables **in new data**

[1] ML models are commonly used for prediction, data mining, and data generation.

Labels / Outcomes

Labels are variables we **want to predict** the values of (because they are unknown)

Labels tend to be expensive or difficult to measure in new data (though are known in some existing data that we can learn from)

AKA outcome, dependent, or y variables

Features / Predictors

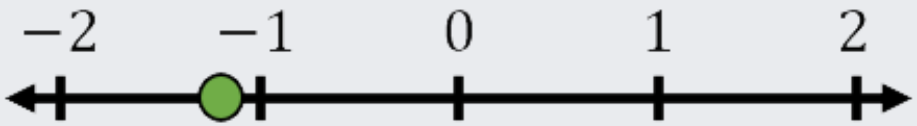
Features are variables we **use to predict** the unknown values of the label variables

Features tend to be relatively cheaper and easier to measure in new data than labels (and are also known in some existing data)

AKA predictor, independent, or x variables

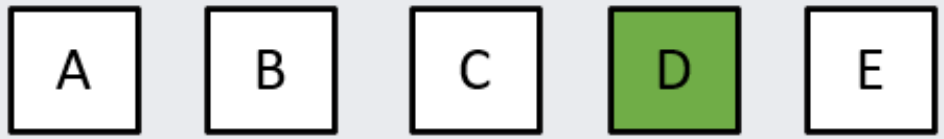
Modes of Predictive Modeling

When labels have continuous values, predicting them is called **regression**



- *How much will a customer spend?*
- *What GPA will a student achieve?*
- *How long will a patient be hospitalized?*

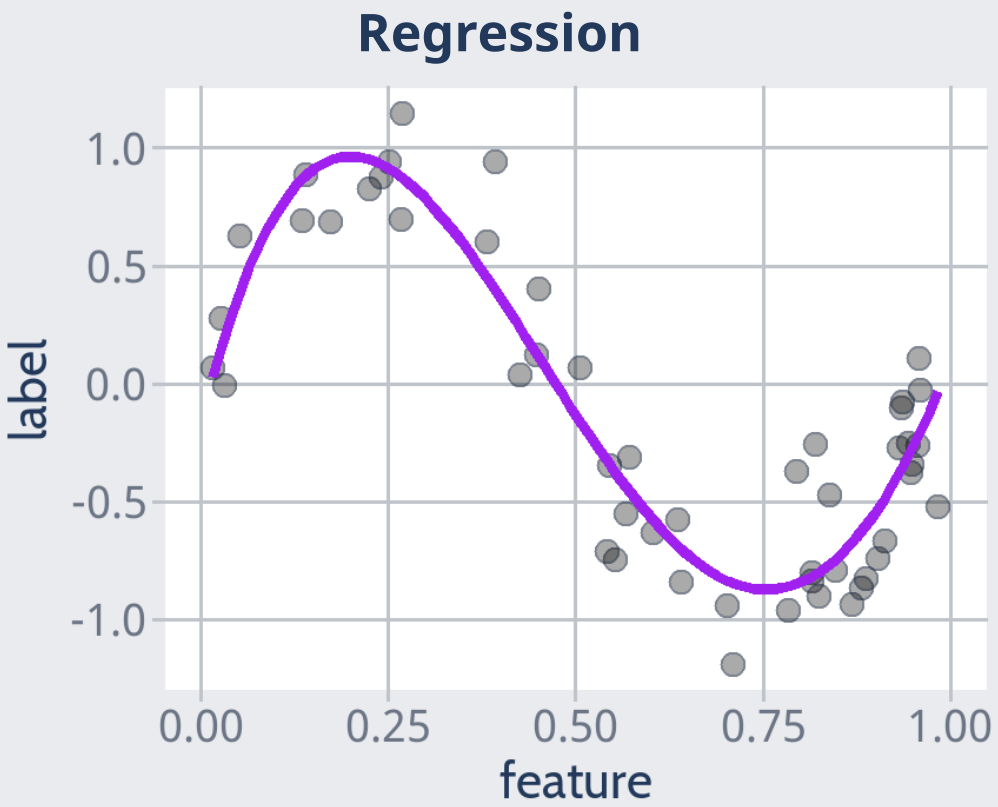
When labels have categorical values, predicting them is called **classification**

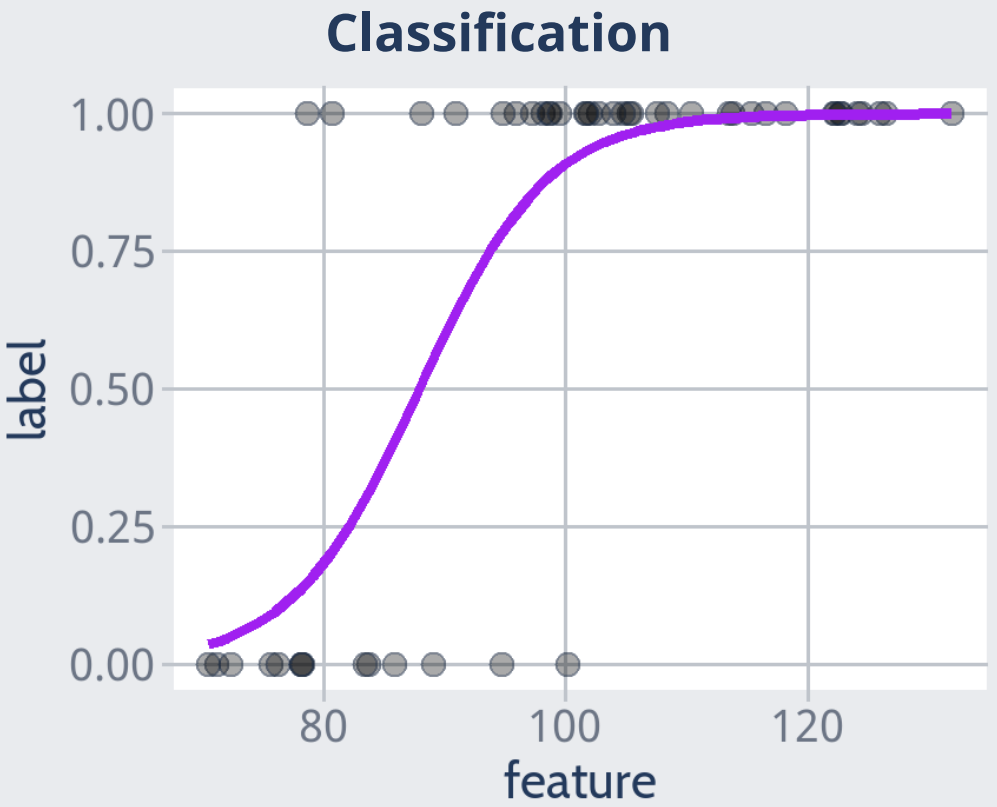


- *Is an email spam or non-spam?*
- *Which candidate will a user vote for?*
- *Is a patient's glucose low, normal, or high?*

Unsupervised learning (AKA data mining) has no explicit labels and just looks for patterns within the features.

Modes of Predictive Modeling





Comprehension Check #1

Ann has developed an ML system that looks at a patient's physiological signals and tries to determine whether they are having a micro-seizure.

Question 1

The features are _____ and the labels are _____.

a) physiological signals; physiological signals

b) physiological signals; micro-seizure (yes/no)

c) micro-seizure (yes/no); physiological signals

d) micro-seizure (yes/no); micro-seizure (yes/no)

Question 2

Which "mode" of predictive modeling is this?

a) Regression

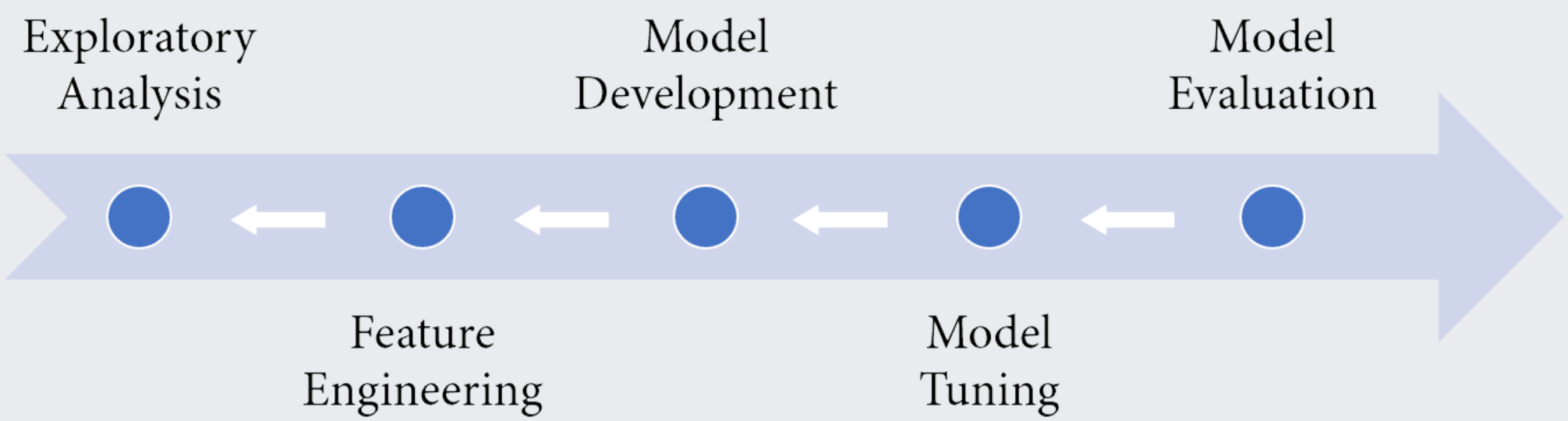
b) Classification

c) Unsupervised learning

d) All of the above

Modeling Workflow

Typical ML Workflow



Exploratory Analysis

Verify the quality of your variables

- Examine the distributions of feature and label variables
- Look for errors, outliers, missing data, etc.

Gain inspiration for your model

- Identify relevant features for a label
- Detect highly correlated features
- Determine the "shape" of relationships

Feature Engineering

Prepare the features for analysis

- *Extract* features
- *Transform* features
- *Re-encode* features
- *Combine* features
- *Reduce* feature dimensionality
- *Impute* missing feature values
- *Select* and drop features

Model Development

Choose algorithms, software, and architecture

- Elastic Net and/or Random Forest
- `caret` or `tidymodels`, `elasticnet` or `glmnet`
- Regression or classification

Train the model by estimating parameters

- Learn the nature of the feature-label relationships
- For instance, estimate the intercept and slopes

Model Tuning

Determine how complex the model can become

- How many features to include in the model
- How complex the shape of relationships can be
- How many features can interact together
- How much to penalize adding more complexity

Make other decisions in a data-driven manner

- Which of three algorithms should be preferred
- Which optimization method should be used

Model Evaluation

Decide how to quantify predictive performance

- In regression, performance is based on the errors/residuals
- In classification, performance is based on the confusion matrix

Determine how successful your predictive model was

- Compare predictions (i.e., predicted labels) to trusted labels
- Compare the performance of one model to another model

Comprehension Check #2

Yuki trained an algorithm to predict the number of "likes" a tweet will receive based on measures of the tweet's formatting and content.

Question 1

Calculating the length of each tweet is _____?

a) Feature Engineering

b) Model Development

c) Model Tuning

d) Model Evaluation

Question 2

When should problems with the data be found?

a) Model Evaluation

b) Model Tuning

c) Model Development

d) Exploratory Analysis

Signal and Noise

A Delicate Balance

Any data we collect will contain a mixture of **signal** and **noise**

- The "signal" represents informative patterns that generalize to new data
- The "noise" represents distracting patterns specific to the original data

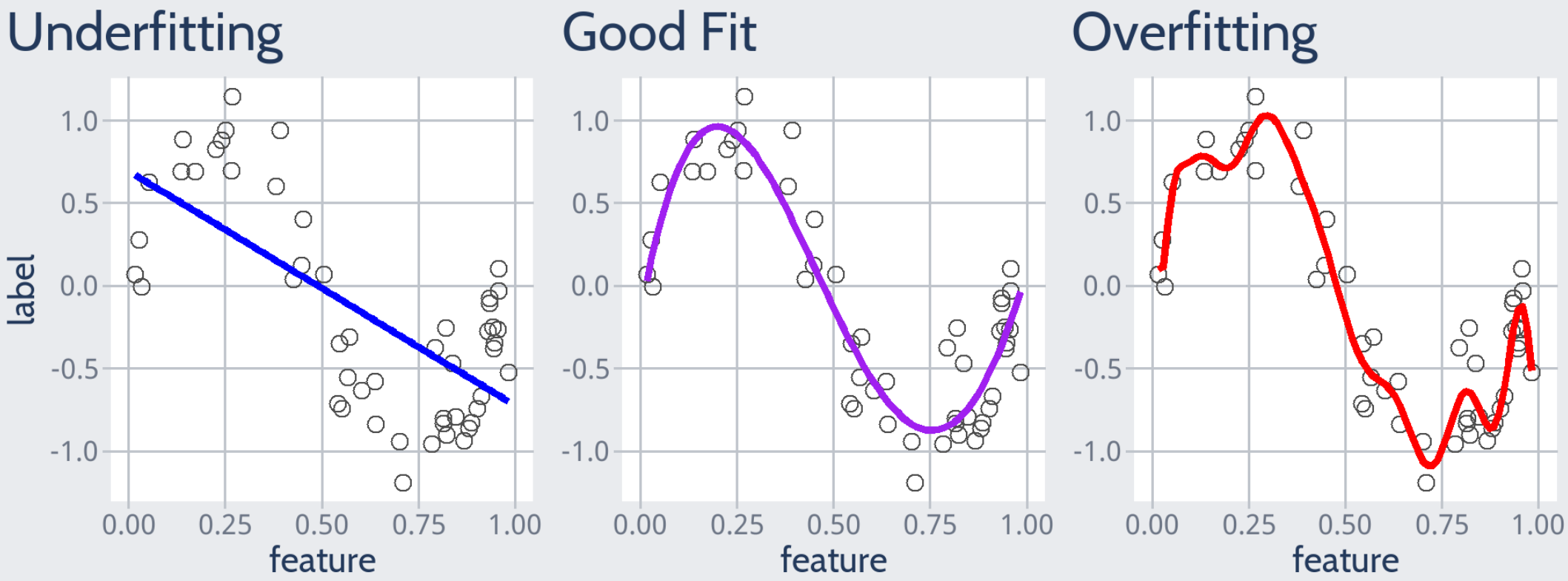
We want to capture as much signal and as little noise as possible

More complex models will allow us to capture **more signal** but also **more noise**

Overfitting: If our model is too complex, we will capture unwanted noise

Underfitting: If our model is too simple, we will miss important signal

Model Complexity



A Super Metaphor

What makes machine learning so amazing is its **ability to learn complex patterns**

However, with this great power and flexibility comes the looming **danger of overfitting**

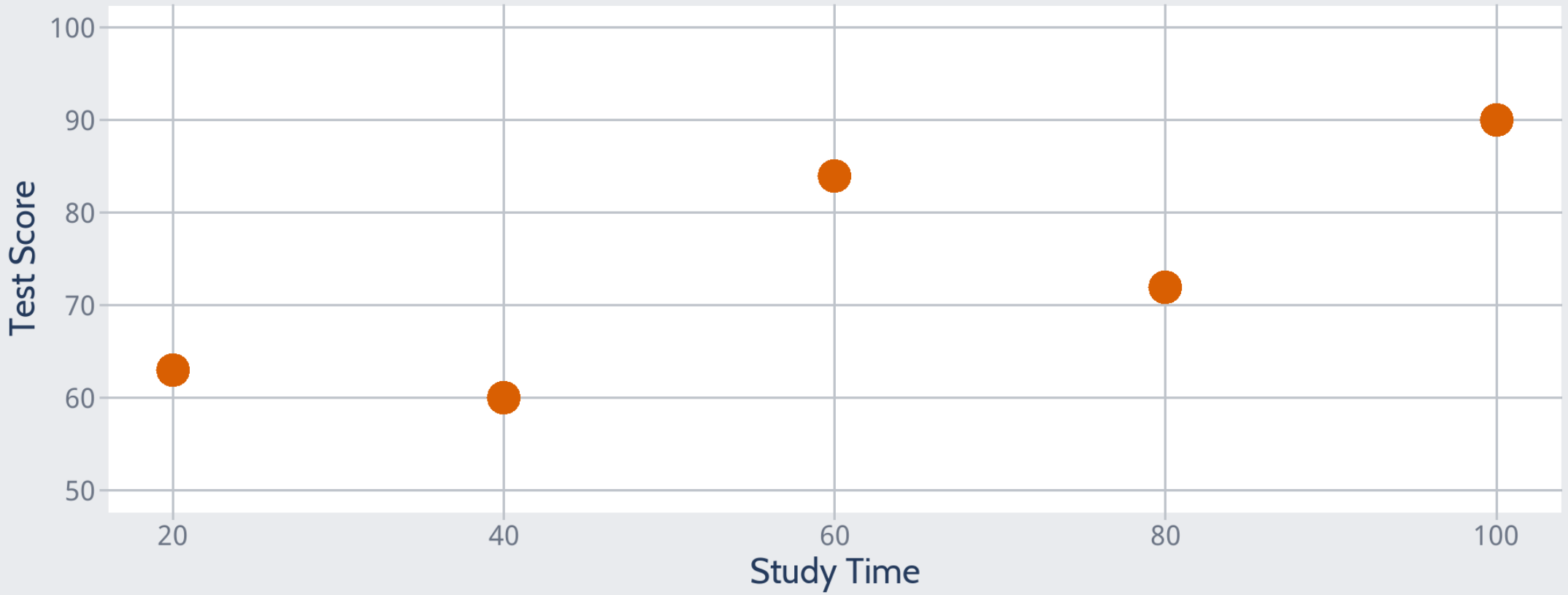
Thus, much of ML research is about finding ways to **detect** and **counteract** overfitting

For detection, we need two sets of data:

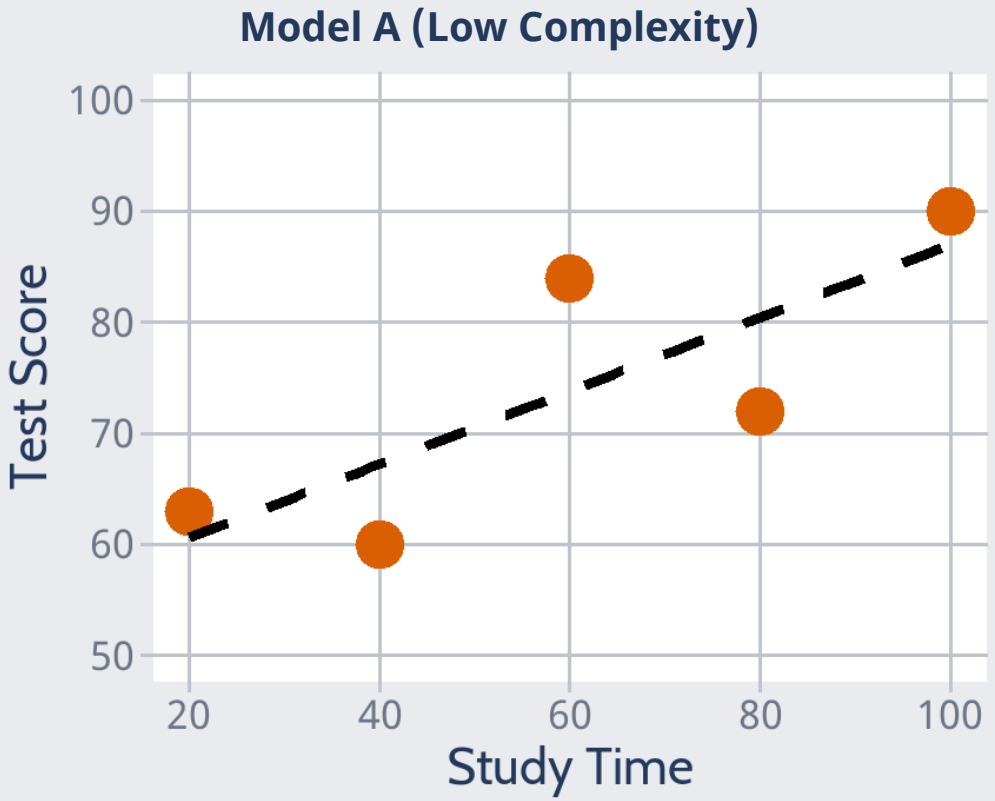
Training set: used to learn relationships

Testing set: used to evaluate performance

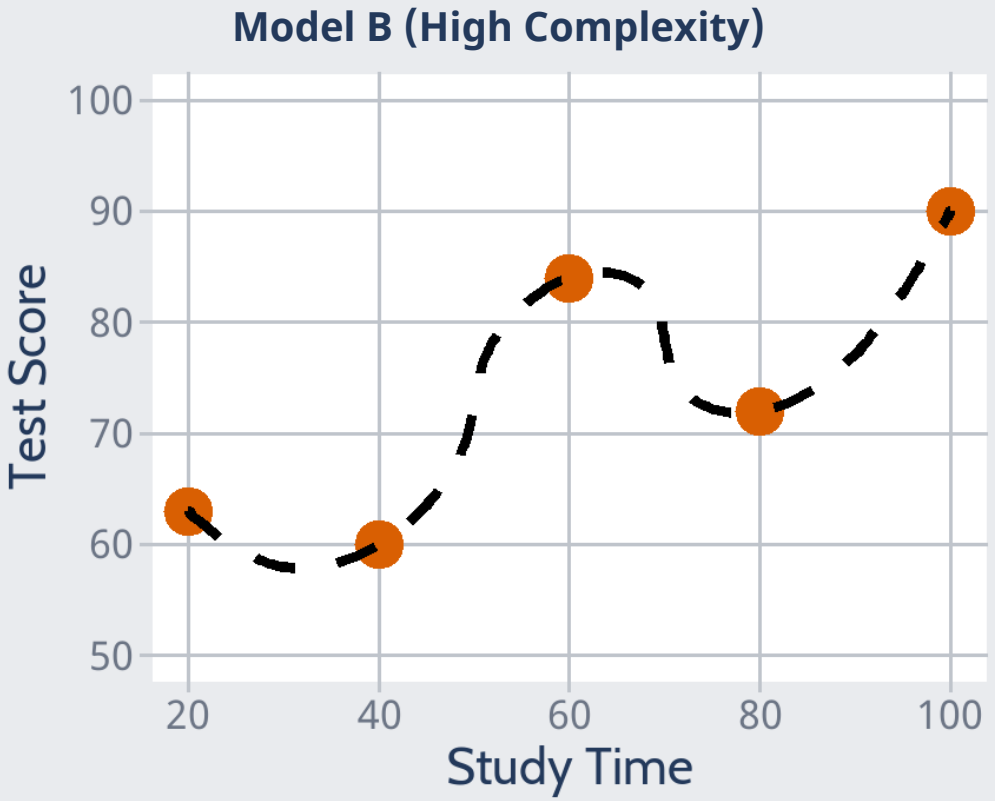
An Example of Overfitting



An Example of Overfitting

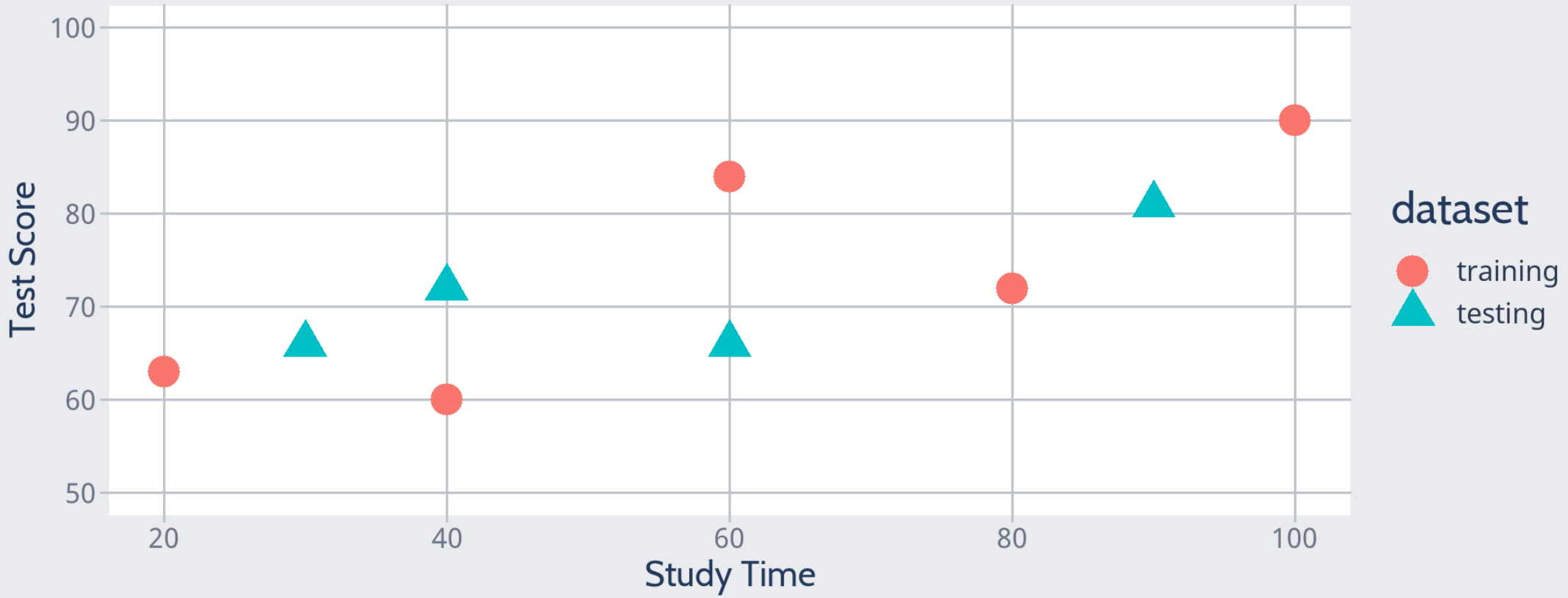


Total error on training data = 31.2 (**High Bias**)

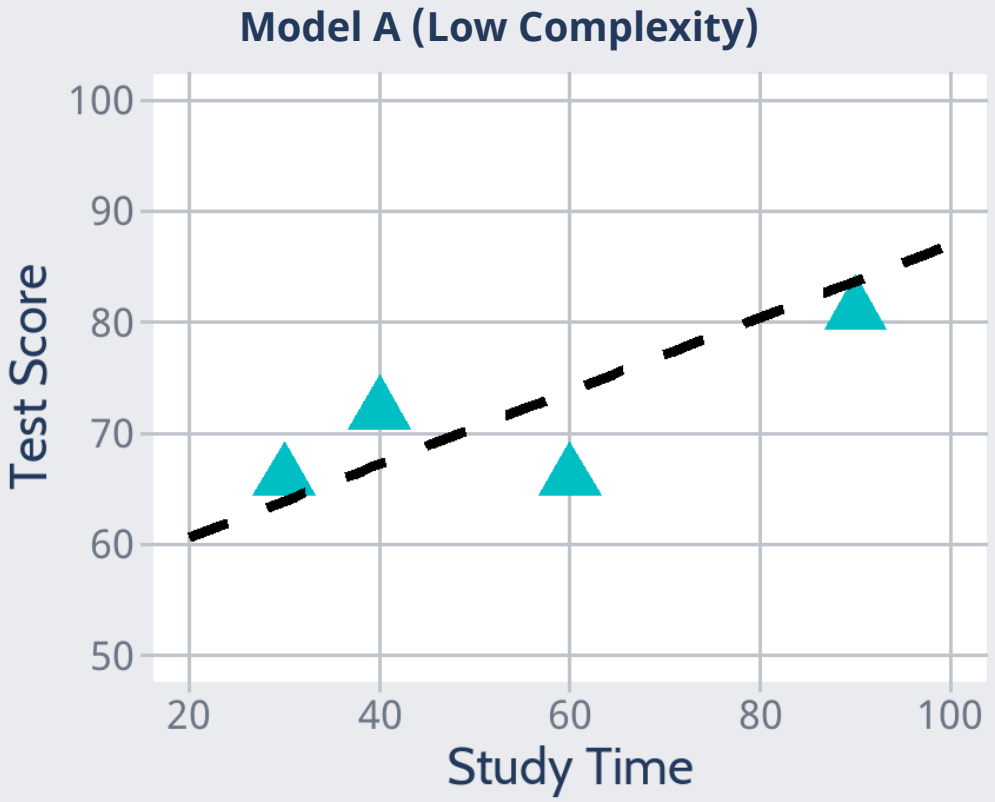


Total error on training data = 0.0 (**Low Bias**)

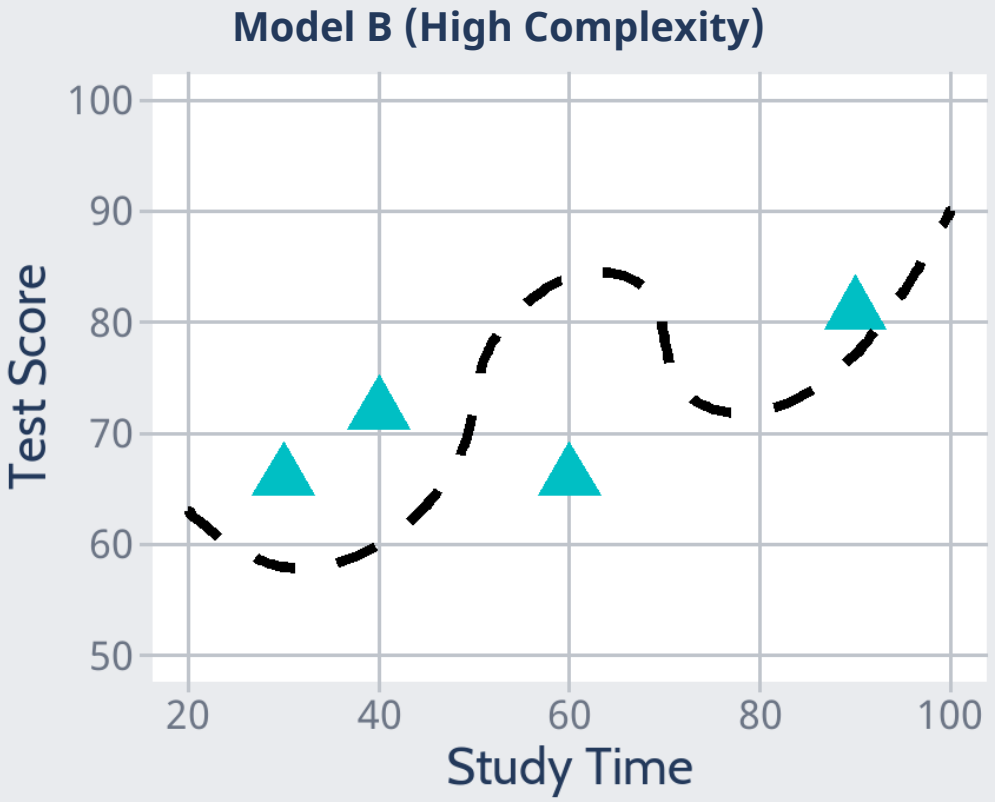
An Example of Overfitting



An Example of Overfitting



Total error on testing set = 17.4 (**Low Variance**)



Total error on testing set = 41.6 (**High Variance**)

Conclusions from Example

In ML, **bias** is a lack of predictive accuracy in the original data (the "training set")

In ML, **variance** is a lack of predictive accuracy in new data (the "testing set")

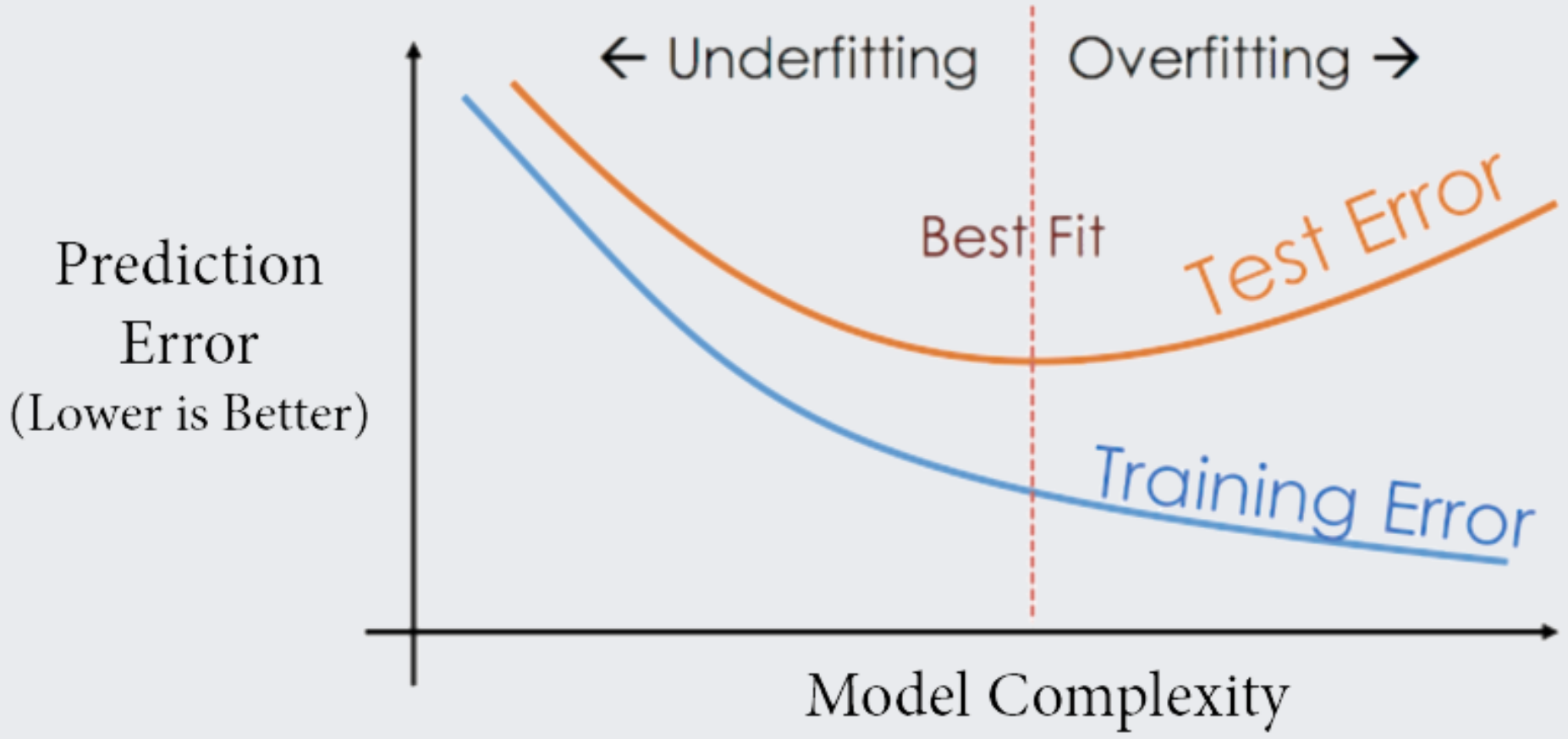
An ideal predictive model would have both low bias and low variance

However, there is often an inherent **trade-off between bias and variance**[1]

We want to find the model that is **as simple as possible** but **no simpler**

[1] To increase our testing set performance, we often need to worsen our performance in the training set.

A Graphical Explanation of Overfitting



A Meme-based Explanation of Overfitting

Comprehension Check #3

Sam used all emails in his inbox to create an ML model to classify emails as "work-related" or "personal." Its accuracy on these emails was 98%.

Question 1

Is Sam done with this ML project?

a) Yes, he should sell this model right now!

b) No, he needs to create a training set

c) No, he needs to test the model on new data

d) No, his model needs to capture more noise

Question 2

Which problems has Sam already addressed?

a) Overfitting

b) Underfitting

c) Variance

d) All of the above

Countering Overfitting

Cross-Validation

There are some clever algorithmic tricks to prevent overfitting

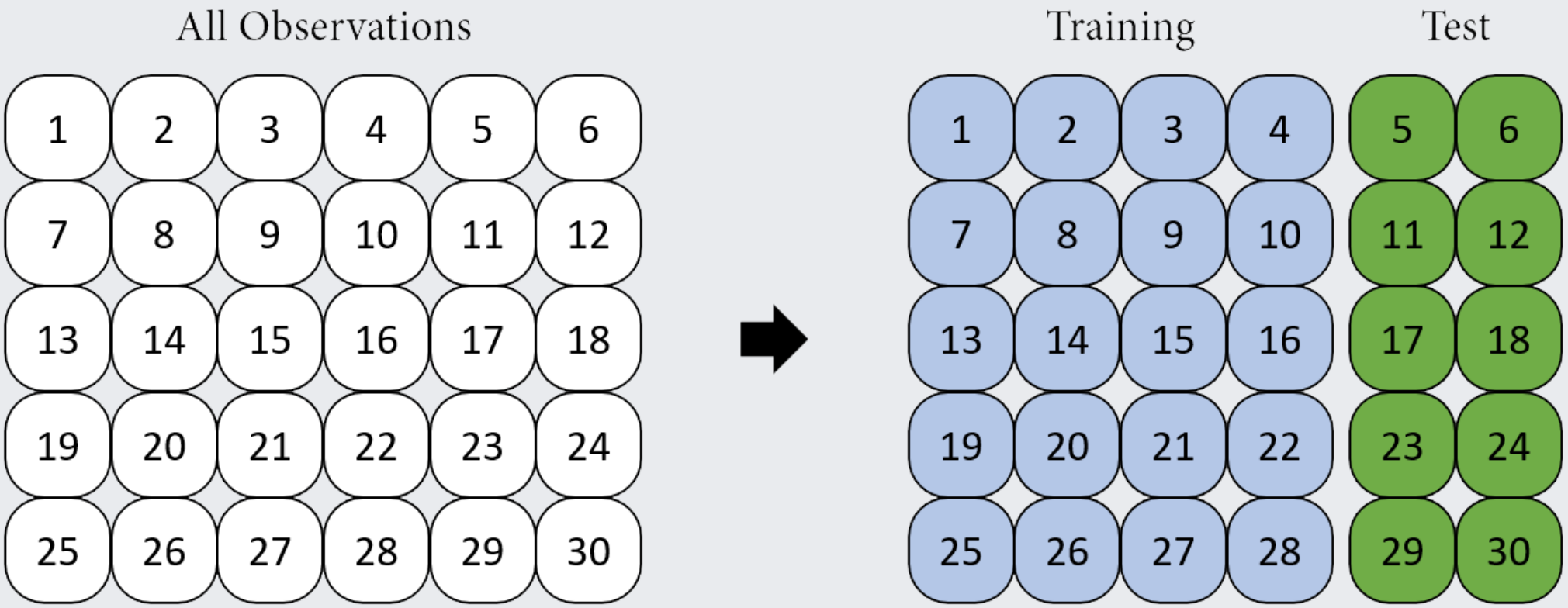
- For example, we can penalize the model for adding complexity

The main approach, however, is to use **cross-validation**:

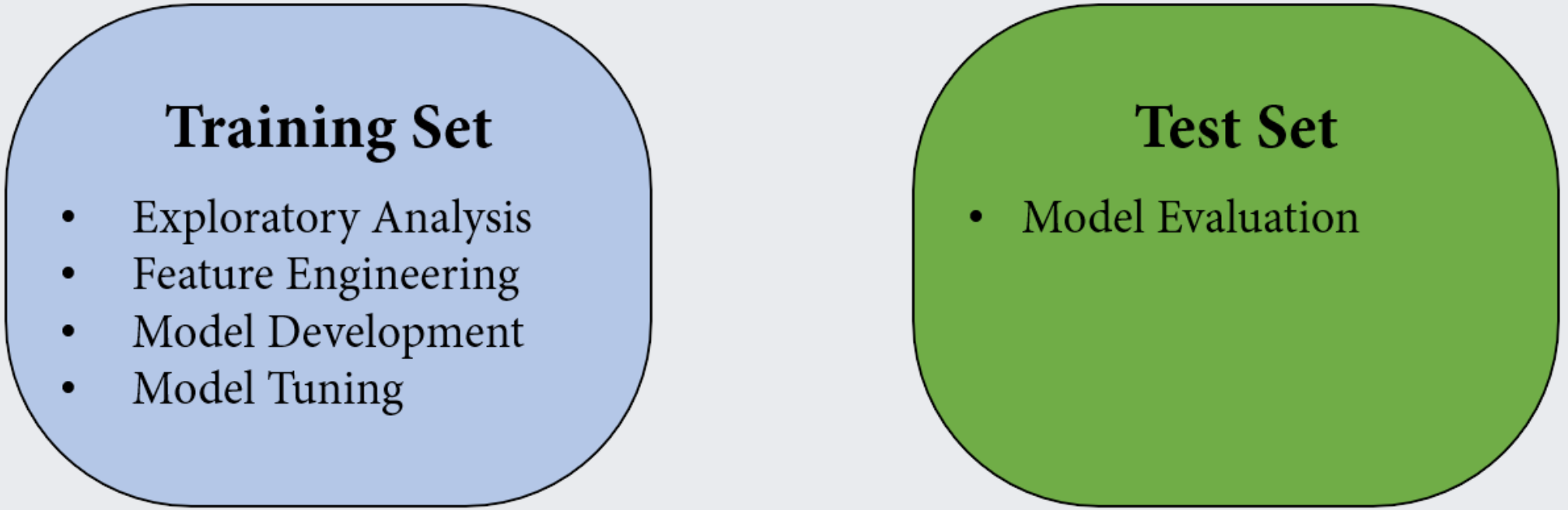
- Multiple **fully independent** sets of data are created (by subsetting or resampling)
- Some sets are used for training (and tuning) and other sets are used for testing
- **Model evaluation is always done on data that were not used to train the model**
- This way, if performance looks good, we can worry less about variance/overfitting[1]

[1] Although we always need to worry somewhat about whether the original data was itself representative.

Holdout Cross-Validation

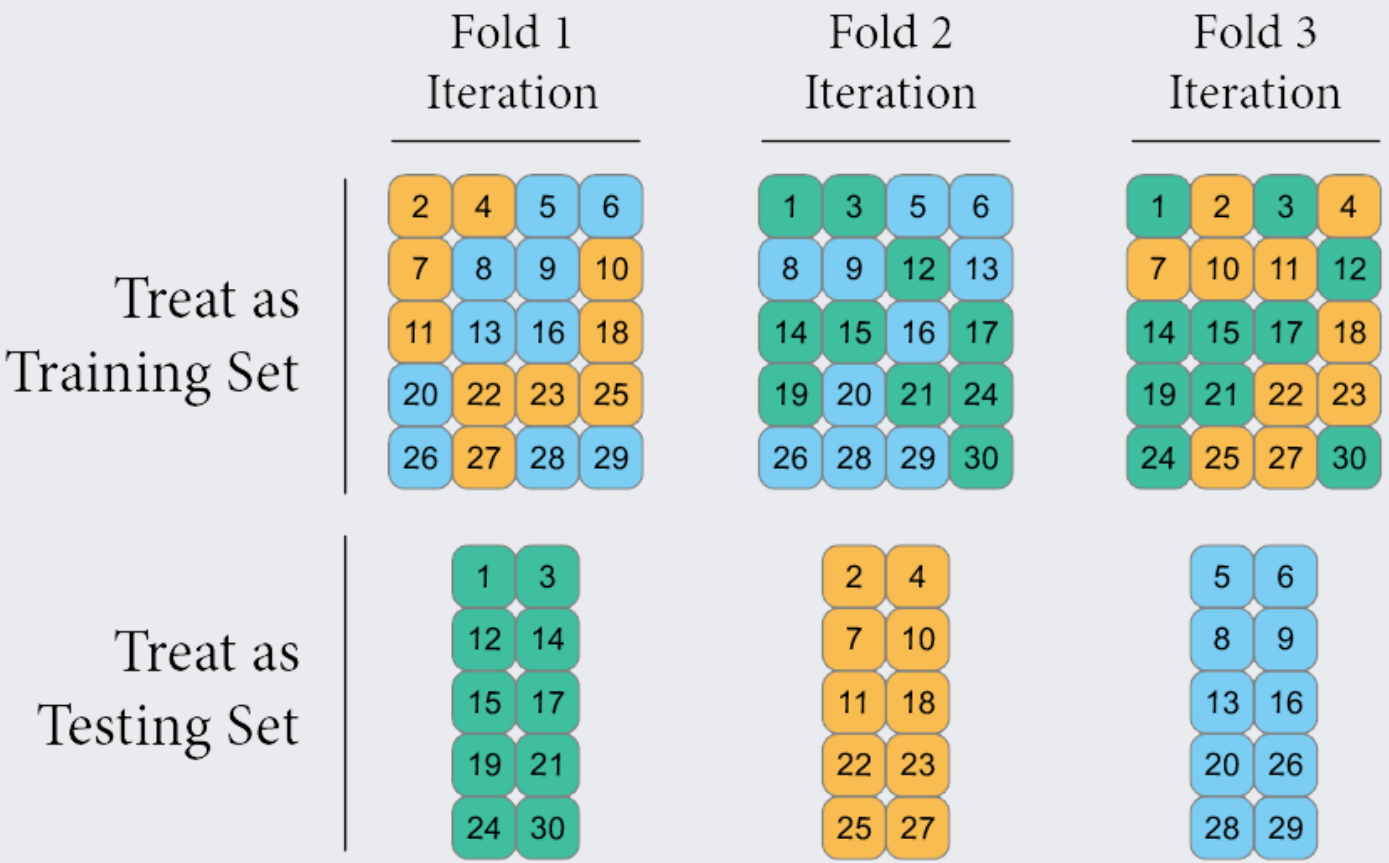


Holdout Cross-Validation



"Oh, East is East, and West is West, and never the twain shall meet"

k-fold Cross-Validation



Advanced Cross-Validation

Cross-validation can also be **nested** to let the model tune on unseen data:

- An outer loop (applied to the original data) is used for *model evaluation*
- An inner loop (applied to the training set) is used for *model tuning*

Cross-validation can also be **stratified** to keep the sets relatively similar

Cross-validation can also be **repeated** to avoid problems with any single split

A great default procedure is nested, stratified, and 3x repeated 10-fold cross-validation.

Comprehension Check #4

Bogdan collects data from 1000 patients. He assigns patients 1 to 800 to be in his training set and patients 700 to 1000 to be in his testing set.

Question 1

What major mistake did Bogdan make?

a) He used a testing set instead of a holdout set

b) Some patients are in both training and testing

c) The two subsets of data have different sizes

d) He did not use k-fold cross-validation

Question 2

Which step should not be done in the training set?

a) Exploratory Analysis

b) Feature Engineering

c) Model Development

d) Model Evaluation

Small Group Discussion

Small Group Discussion

We will randomly assign you to a small breakout room

We will jump between rooms to join discussions and answer questions

Introduce yourselves again and discuss the following topics

1. What types of labels and features would you like to work with?
2. What problems might predictive modeling help your field solve?
3. Do you have any questions or comments about the material so far?

Time for a Break!

10:00