INFO 1998: Introduction to Machine Learning

Pull up Lecture 5 Demo from website as well!

CDS Education

We explore, learn, and educate big minds.

Lecture 5: Fundamentals of Machine Learning Pt. 2

INFO 1998: Introduction to Machine Learning

Tuning Models

CDS Education

We explore, learn, and educate big minds.

Announcements

Mid-Semester Check-in

Where you should be right now:

- Have an idea of what your problem statement/hypothesis is
- Have your group chosen
- Have your data set chosen and some progress

Drop Deadline: **October 16th**

What We'll Cover

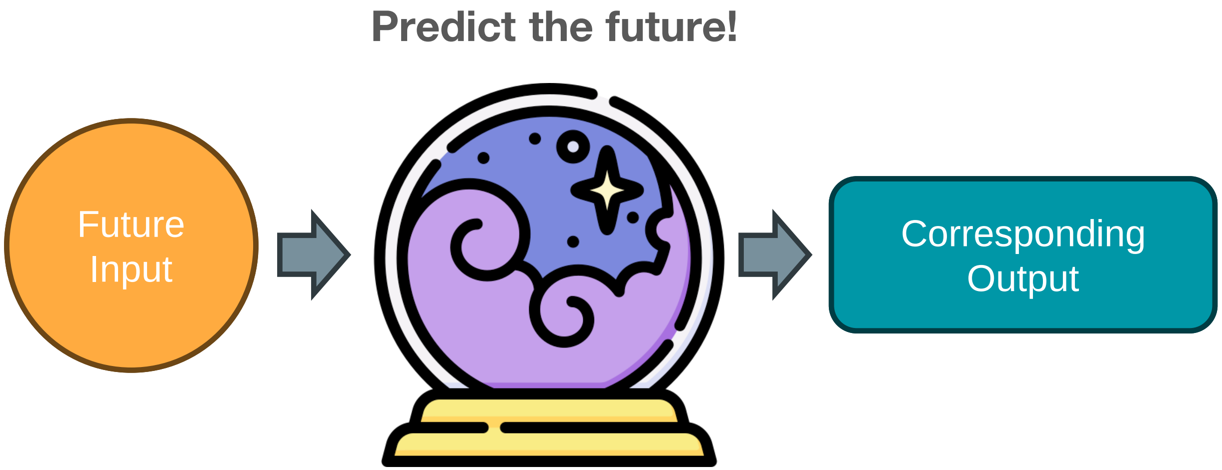
Last Time's Goal: identify what ML is and write ML code (to some extent)

This Time's Goal: how to tell if your ML model is *useful*

Agenda

1. Review
2. Types of Machine Learning
3. Measuring Accuracy
4. Bias-Variance trade-off
5. Feature Selection

Review: Model

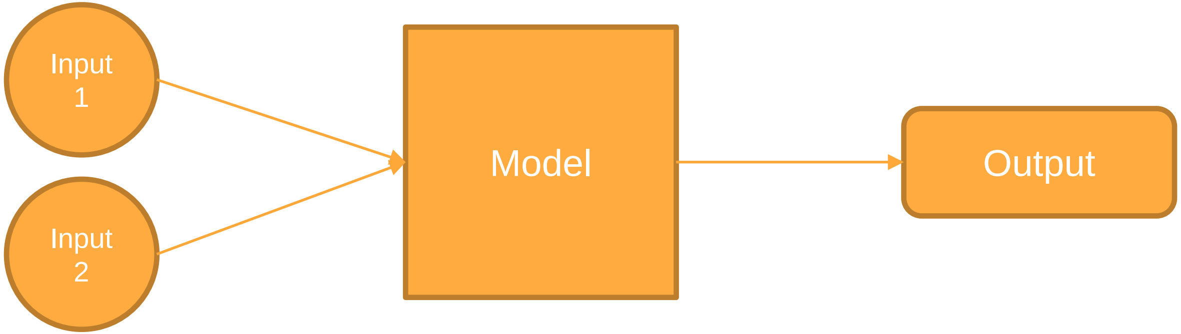


Use **known** inputs/outputs

Learn the **patterns from data**

Given a future input, **predict** the corresponding output

Review: Model



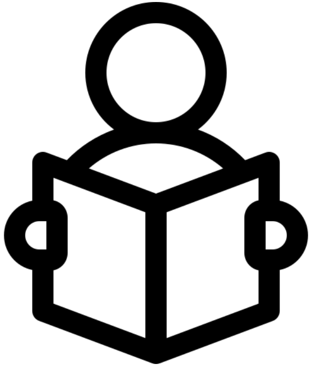
- Takes in input and output and learns the relationship
- Used to predict outputs

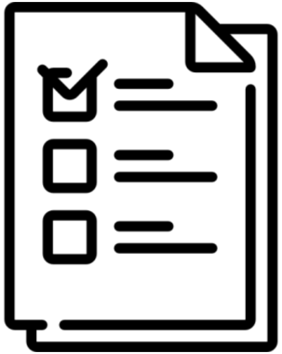
- “Model training” = learn a relationship/program
- “Model validation” = see if the learned relationship is accurate on other data
- “Model testing” = final model performance

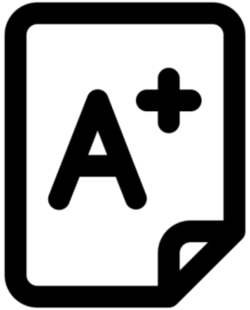
Review: Model

Ex: Cornell Students

Study Material
"Training" → Take Practice Prelims
"Validation" → Take the Exam
"Test"







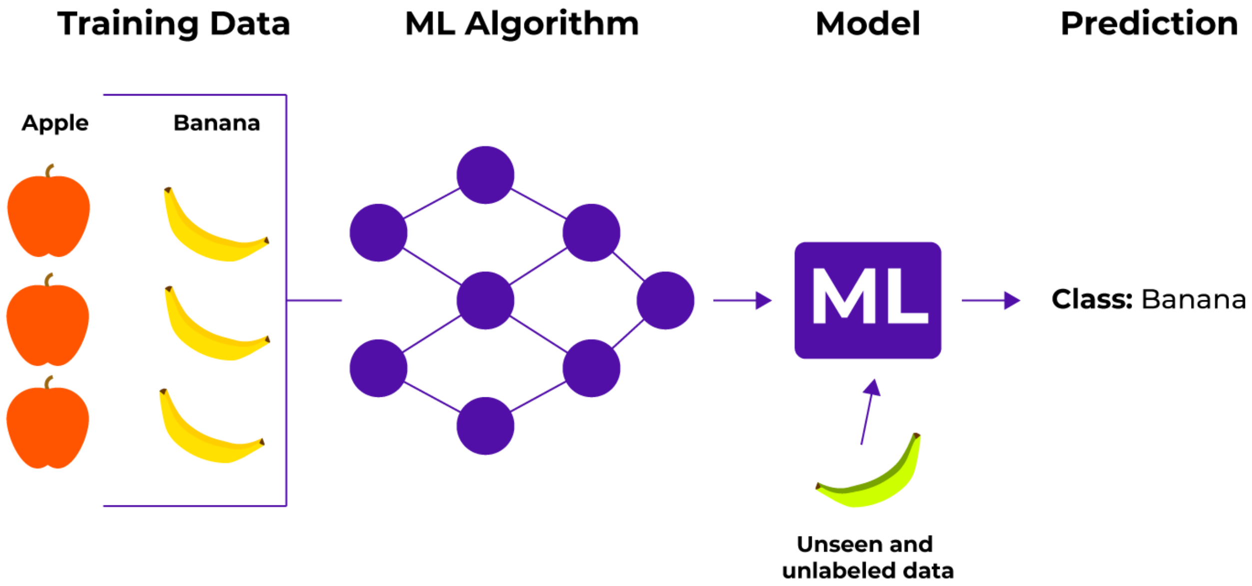
Different Types of ML

(supervised & unsupervised)
(classification & regression)

Supervised vs. Unsupervised

Supervised learning…

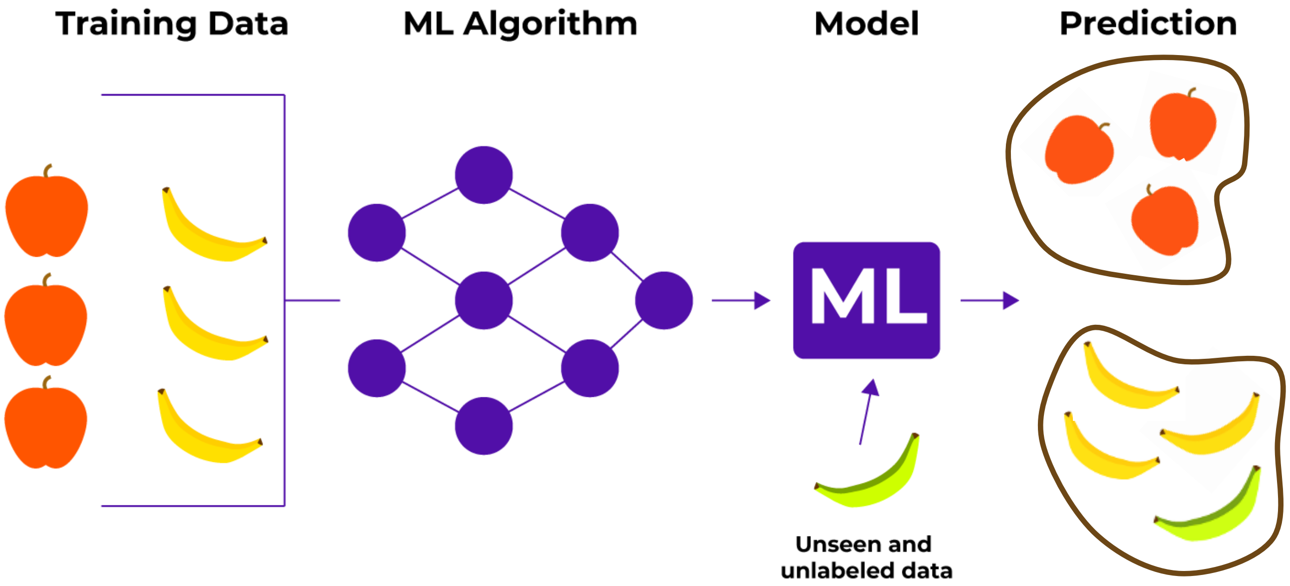
- Trained using labeled data
- Easy to validate



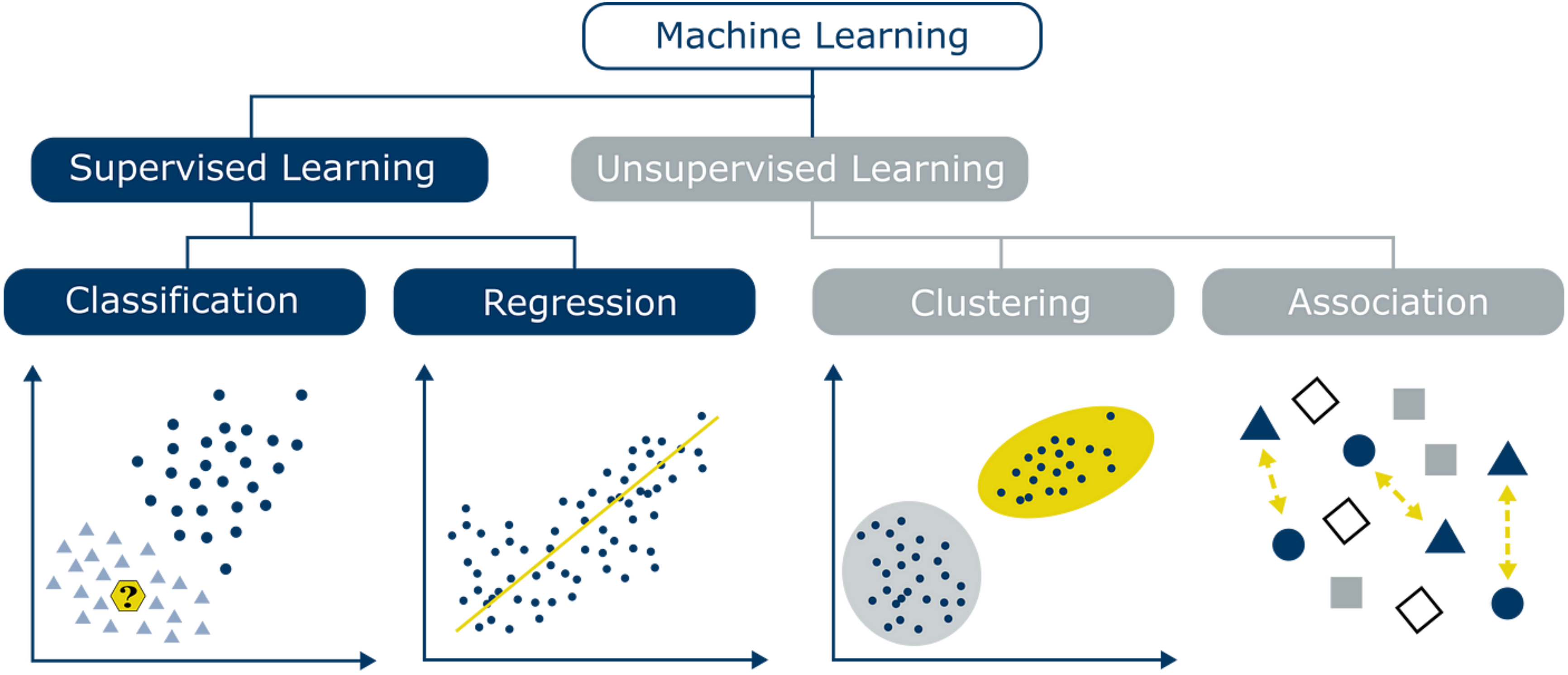
Supervised vs. Unsupervised

Unsupervised learning…

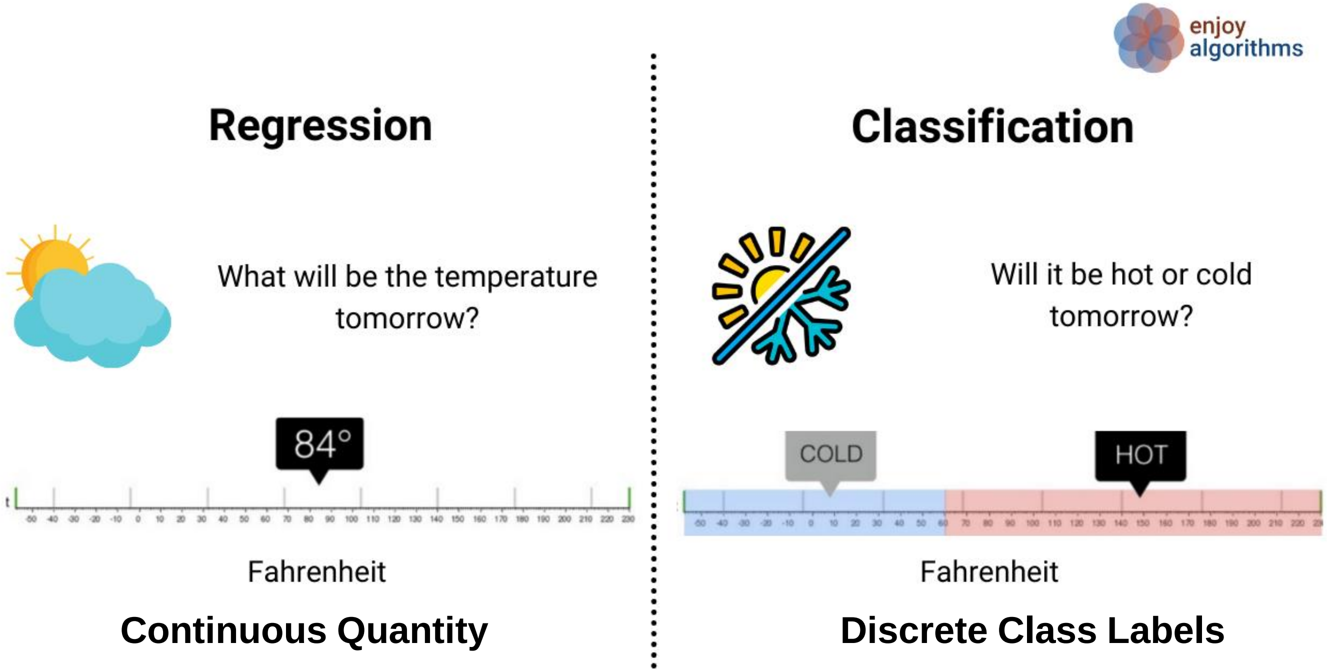
- Trained on unlabeled data
- Difficult to validate



Classification vs. Regression



Classification vs. Regression

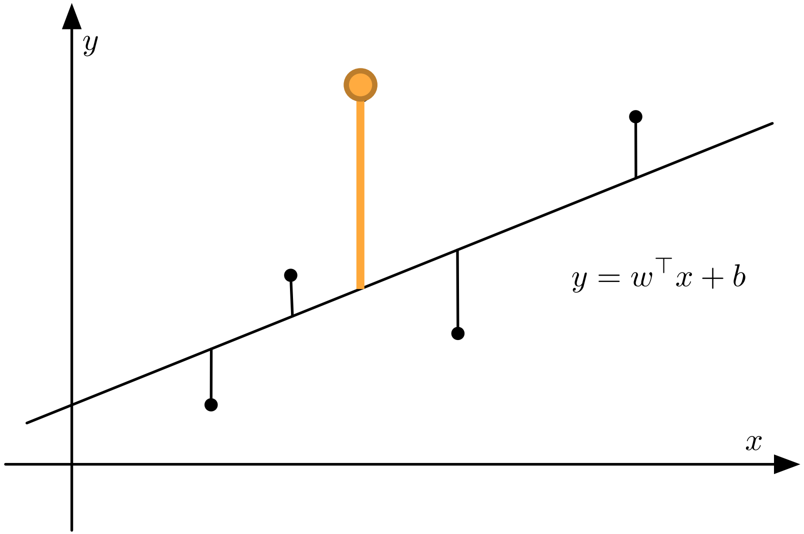


Measuring Bias / Loss

(training accuracy)

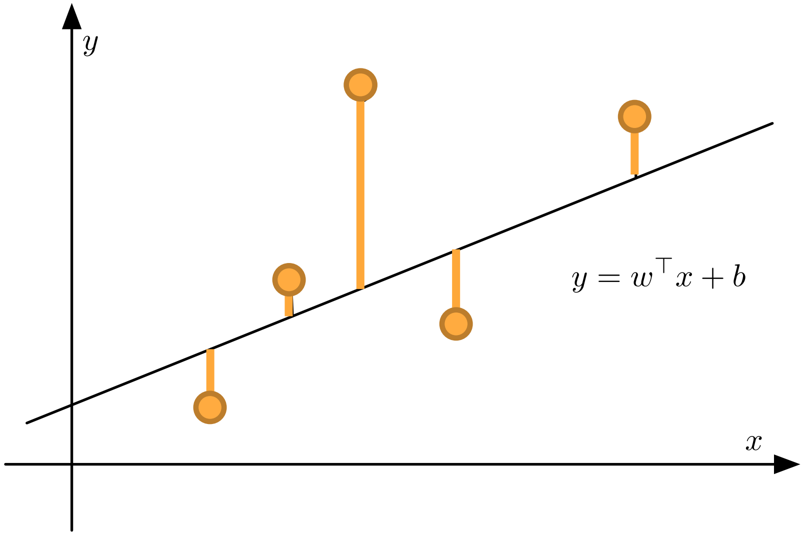
Loss, Cost, and Score Functions

Loss Function: Penalty for missing a single data point



Loss, Cost, and Score Functions

Cost Function: Indicates how bad the whole model is



- Applies loss function to each point, then combines that into a single number
  - ex: average of (loss from each point)

Loss, Cost, and Score Functions

Cost Function: Indicates how bad the whole model is

- Applies loss function to each point, then combines that into a single number
  - ex: average of (loss from each point)

Loss, Cost, and Score Functions

Cost Function: Indicates how bad the whole model is

Cost =

- Applies loss function to each point, then combines that into a single number
  - Ex:, Total Loss, Average Loss

Loss, Cost, and Score Functions

- **Score Function**
  - A more interpretable version of the cost function (how well we did)
  - Loss/Cost used in training to help a model learn, Score is just what we use for interpretability

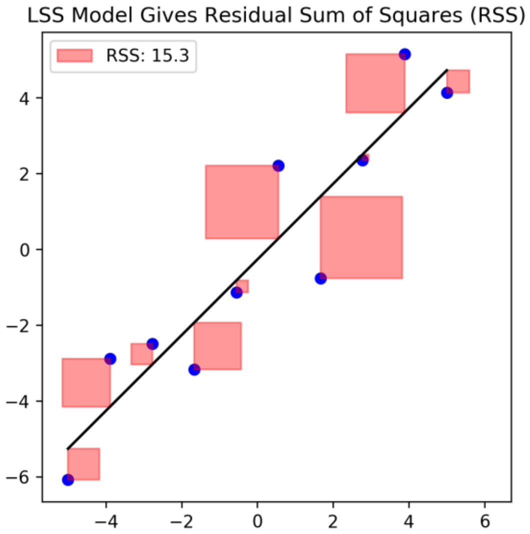
Cost = → Score = 50% Accuracy

Linear Regression Loss Formula: Euclidean Distance

$$\text{loss}(x, y^*) = (h(x) - y^*)^2$$

Two things to note about this loss function:

- Positives and negatives won't cancel
- Large errors are penalized to a power of 2 more



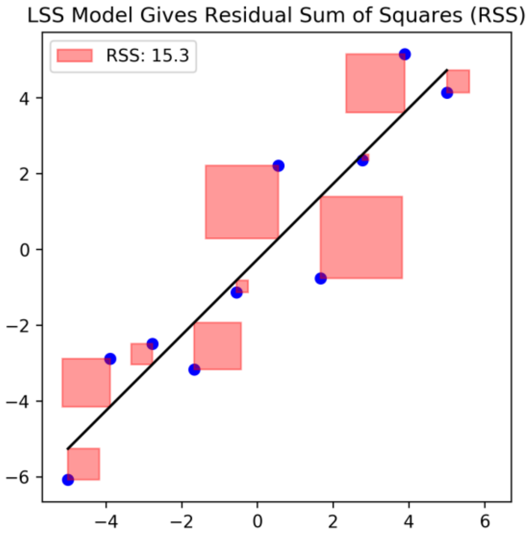
Cost Function - average of the loss function over all the points

Linear Regression Loss Formula: Euclidean Distance

$$\text{loss}(x, y^*) = (h(x) - y^*)^2$$

In what situations might you want a high penalty loss function as opposed to a lower penalty function?

- High stakes situations (Ex: Cancer Diagnosis)
- Data does not have many outliers



Solution: Compare to Baseline

- When determining accuracy, usually want to compare our model to a **baseline**
  - For regression, one baseline model is the model that predicts the **average** of the target value for every point
  - For our purposes: don't worry about the baseline *model*, just have a set of baseline *predictions*

Cost to Accuracy Score

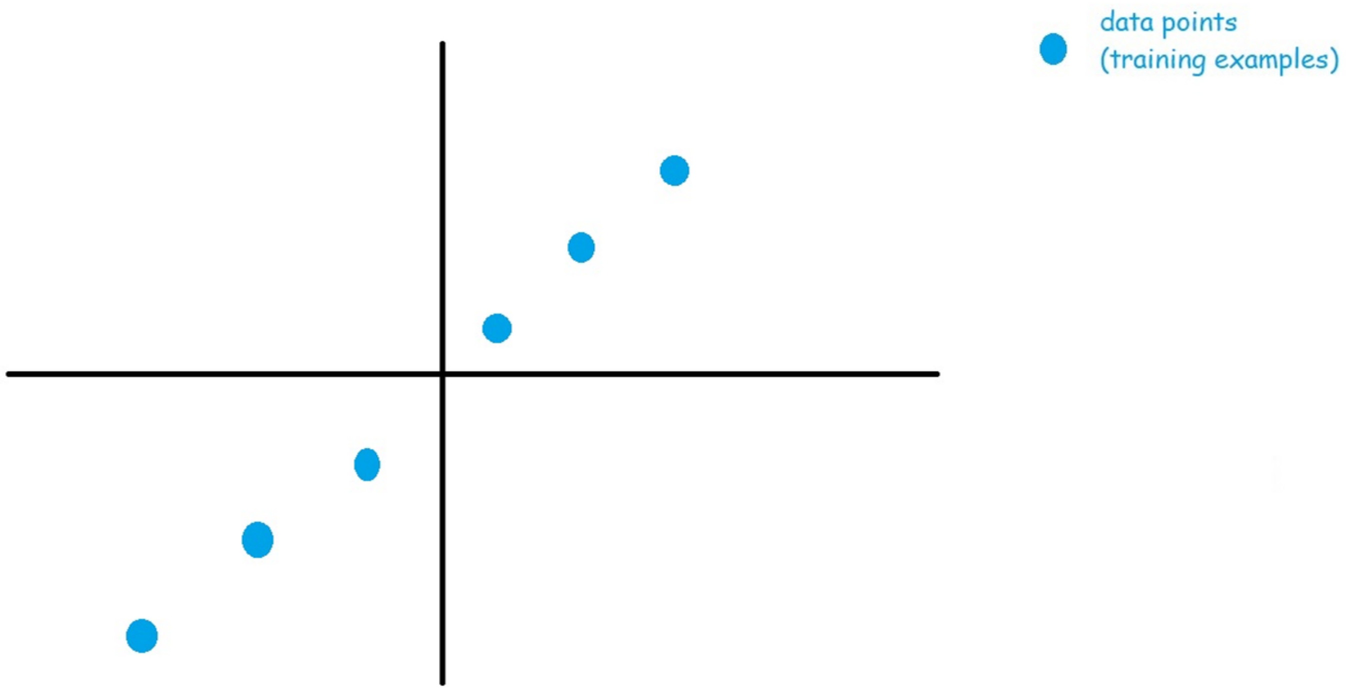
- sklearn's score function is:

  1 - ([Cost of model] / [Cost of baseline])

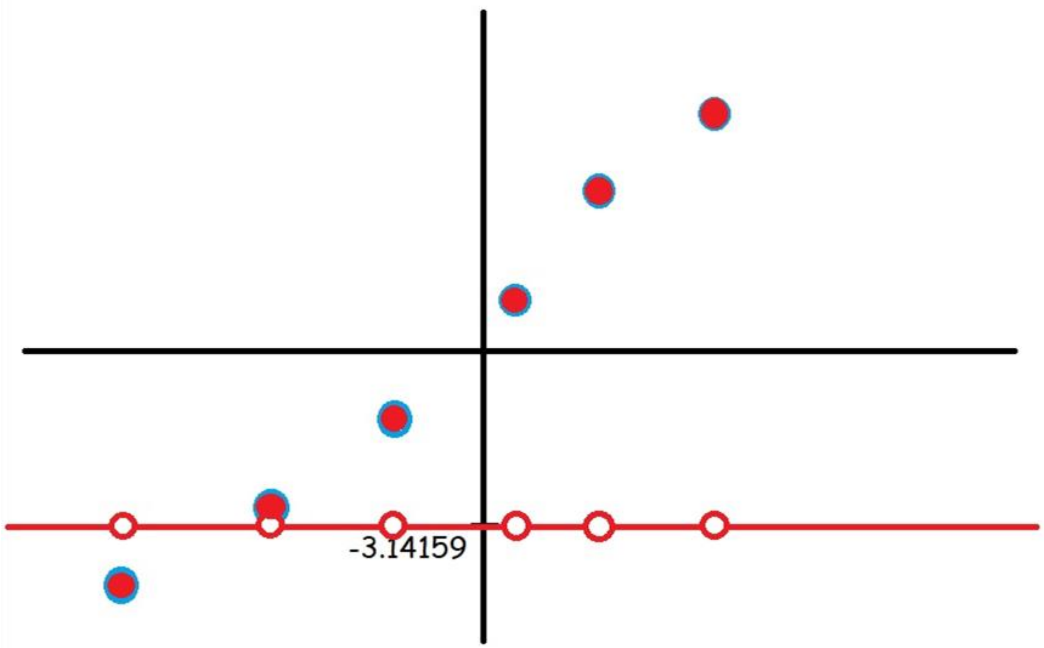
- **1** is very, very good
- **0** means you were as bad as the baseline
- **<0** means either your baseline predictions were accurate, or you really, really messed up

- **NOT USED IN TRAINING, For our interpretability**

Question!



Question!

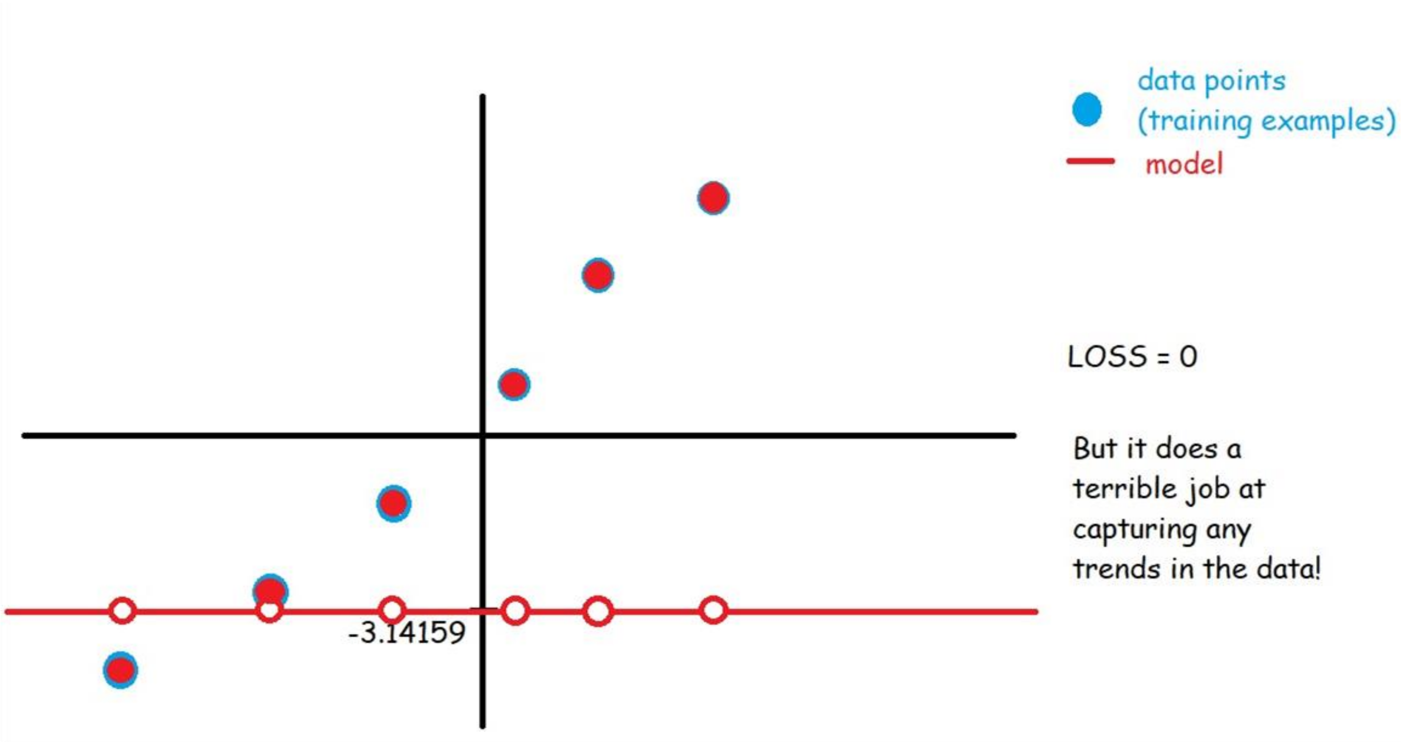


- Suppose our model outputs the following:
    - y_i, if $x = x_i$ for some i in {1,2,...,n}
    - -3.14159 otherwise

- What is the cost of this model (using Euclidean distance)? Hint: the answer is quite simple.

- Should we expect this algorithm to perform well in predicting outputs for new inputs?

Cost = 0, but model is horrible…

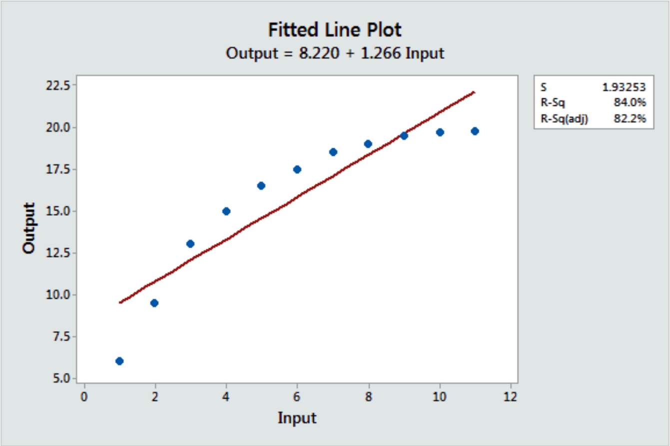


MORAL: Assumptions are important!

No Free Lunch Theorem

Every ML algorithm makes assumptions!

Ex: Linear regression assumes data has a linear relationship



MORAL: Assumptions are important!

Overfitting and Underfitting

(what makes a model good?)

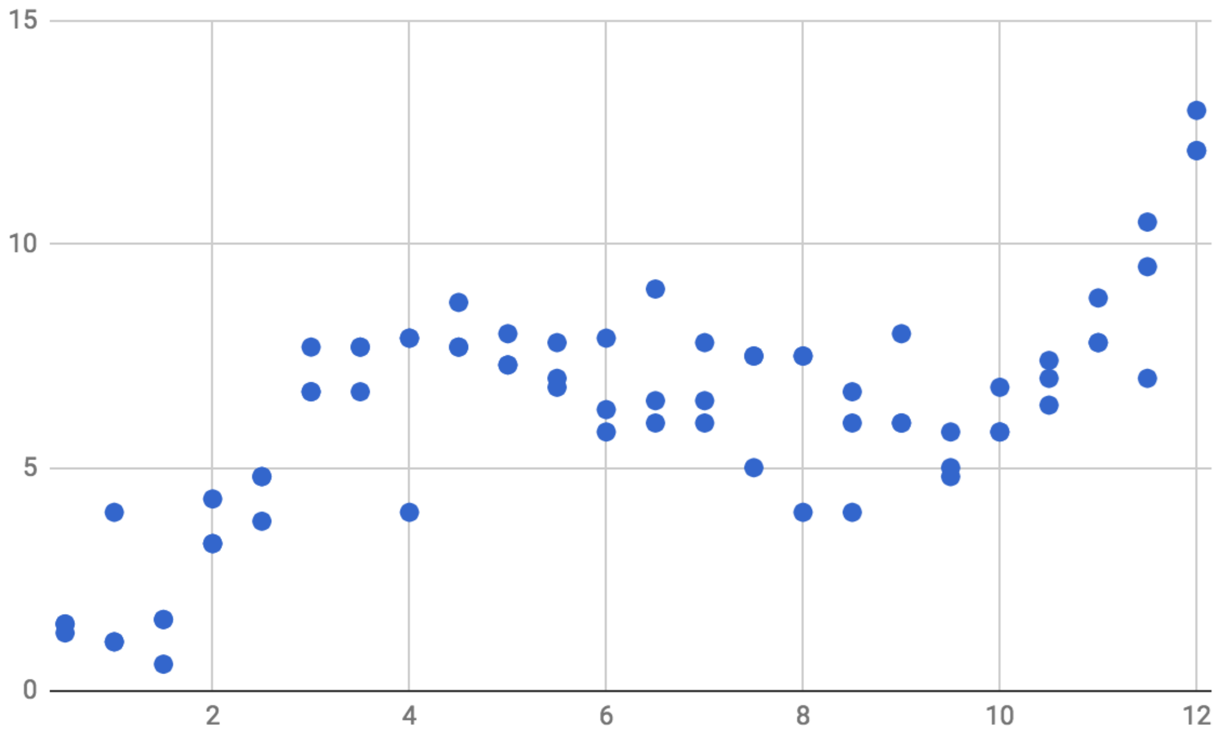
Model Goals

When training a model, we want our model to:

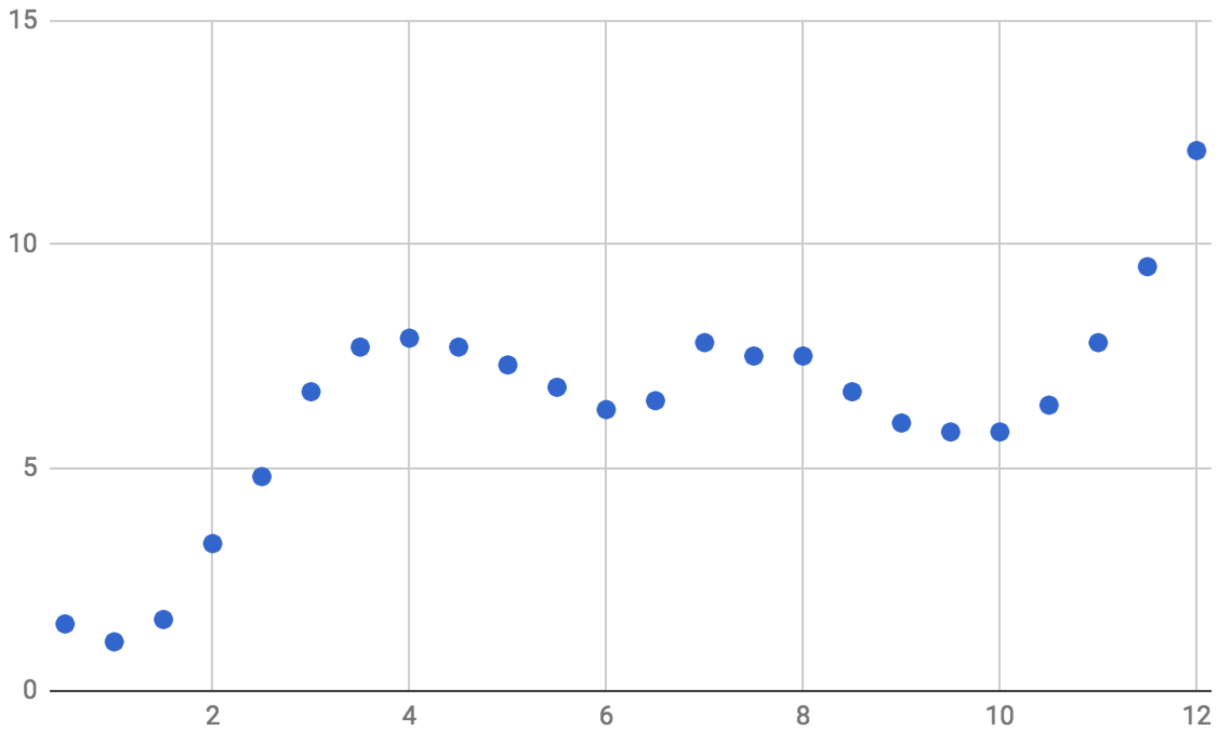
- Capture the trends of the training data
- Generalize well to other samples of the population
- Be moderately interpretable

The first two are especially difficult to do simultaneously!
The more sensitive the model, the less generalizable and vice versa.

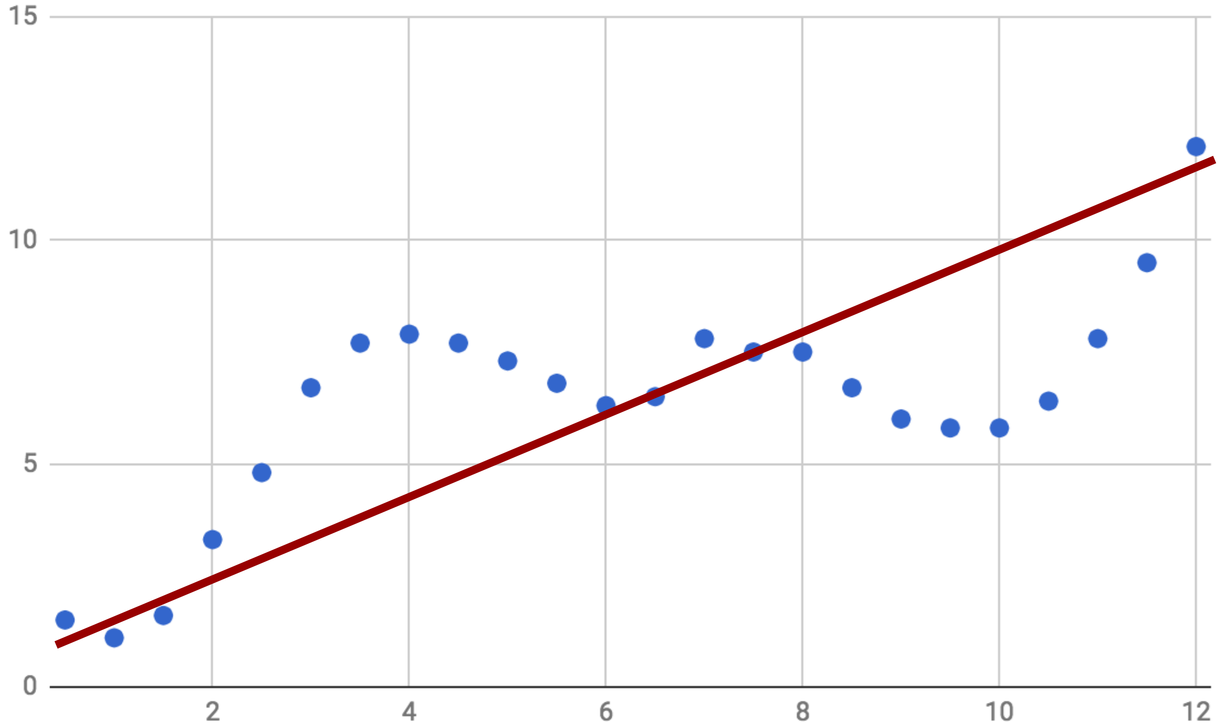
Underfitting: A situation when your model is **too simple** for your data.



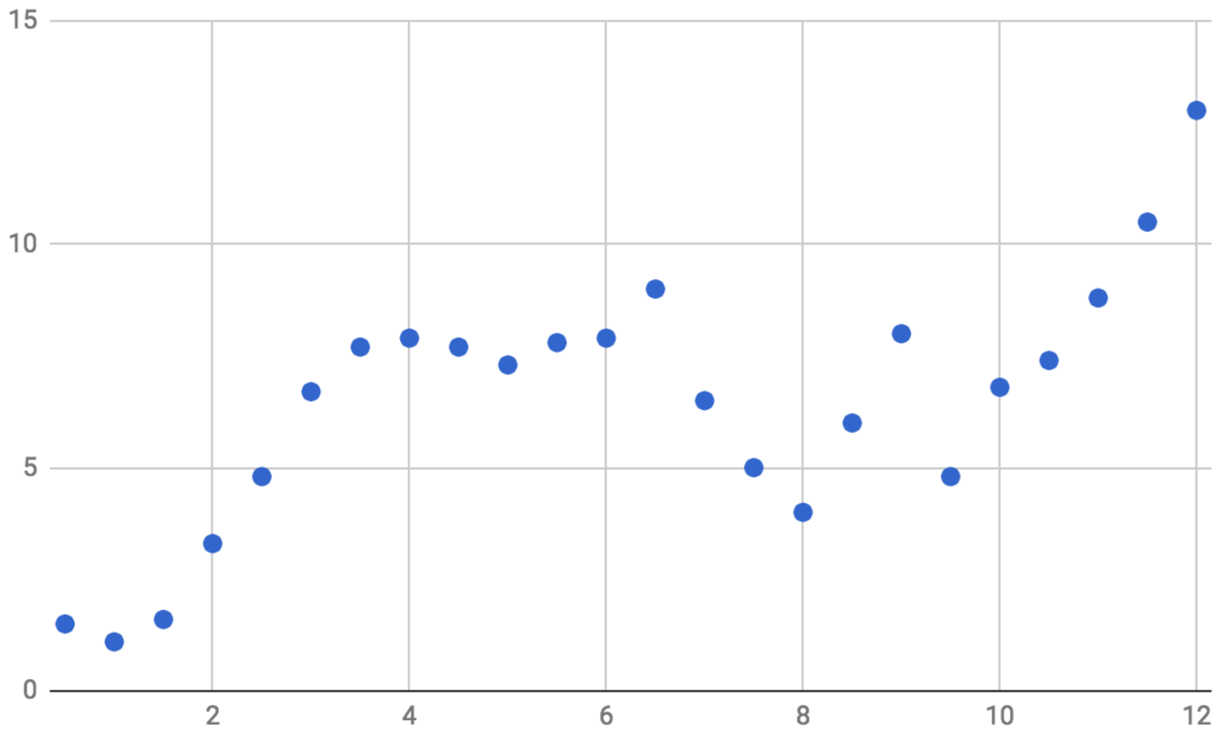
Underfitting



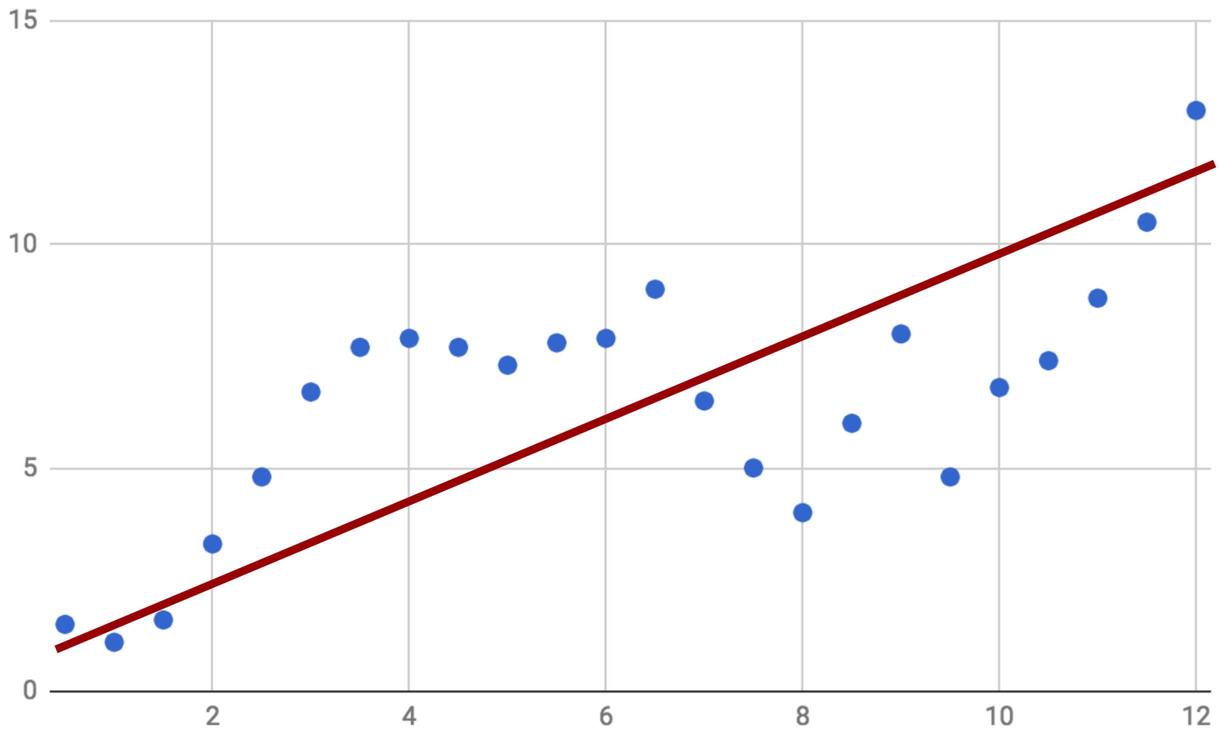
Underfitting



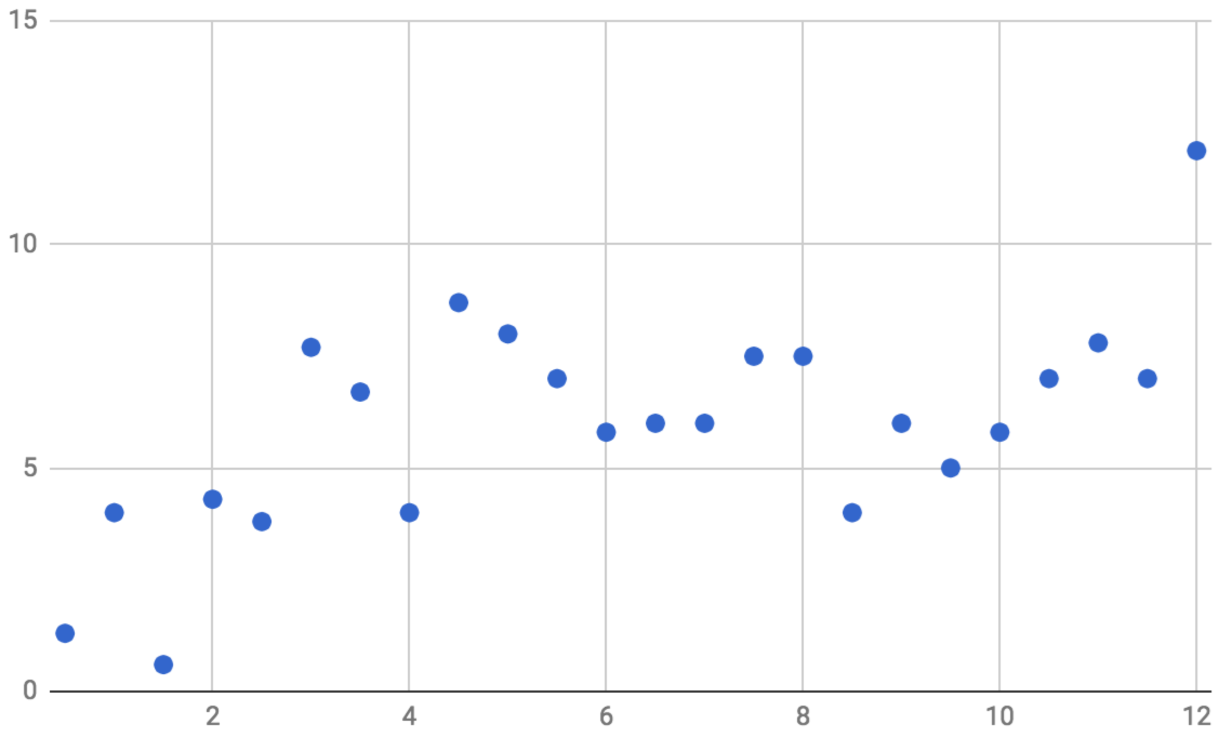
Underfitting



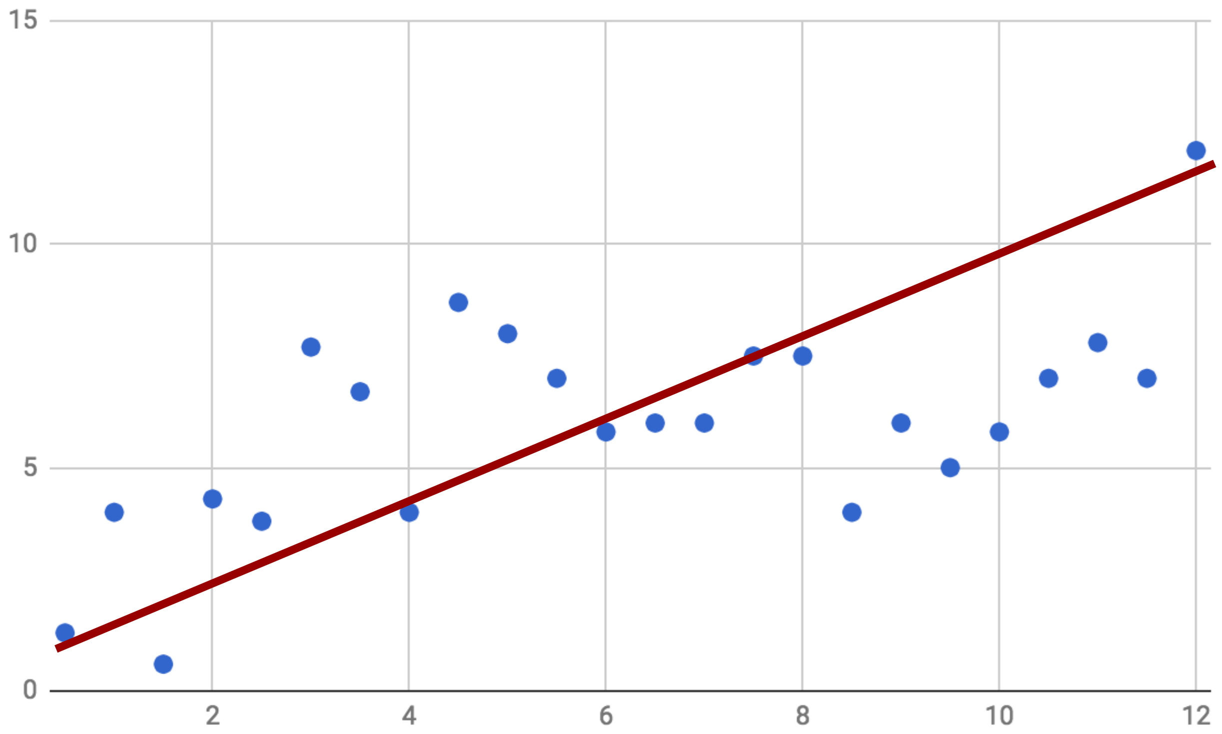
Underfitting



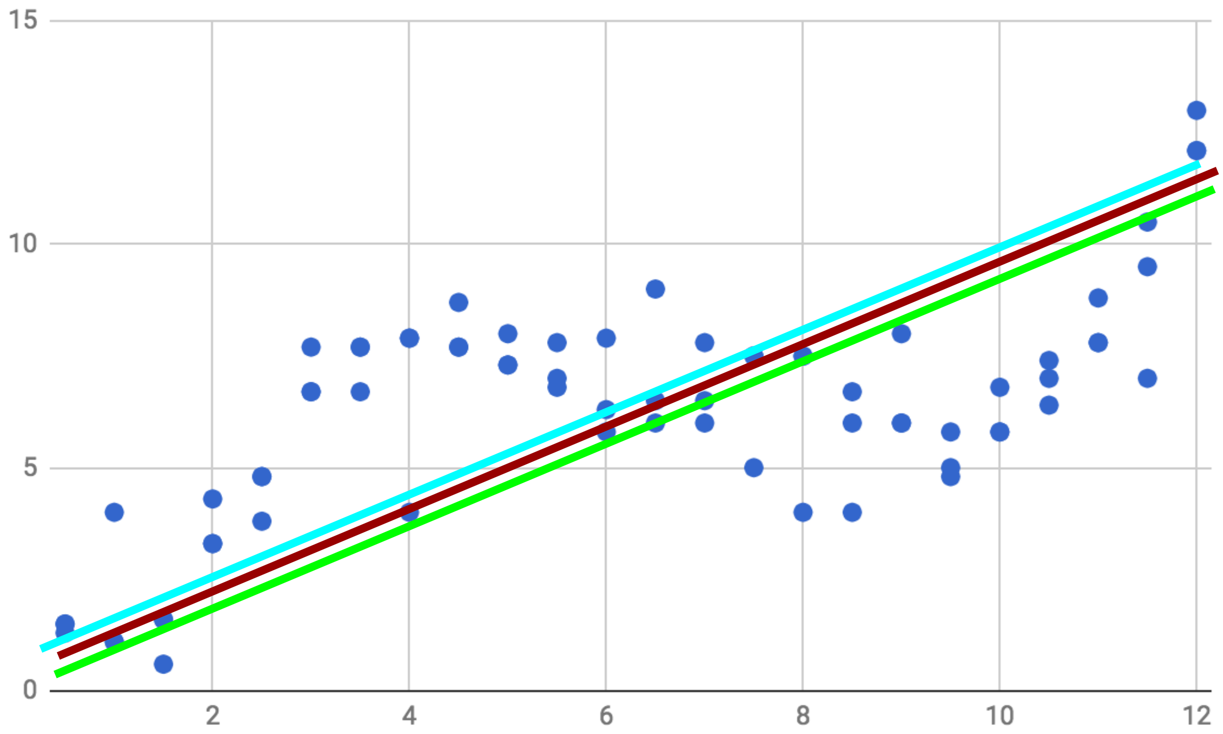
Underfitting



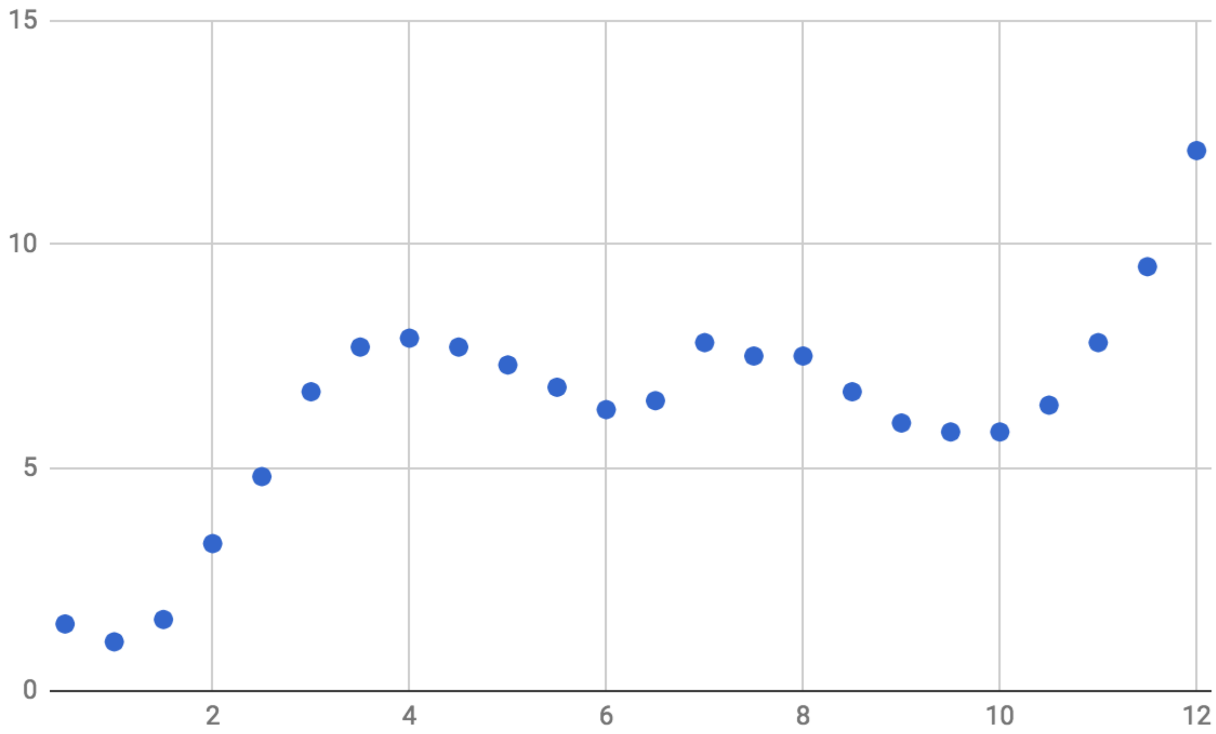
Underfitting



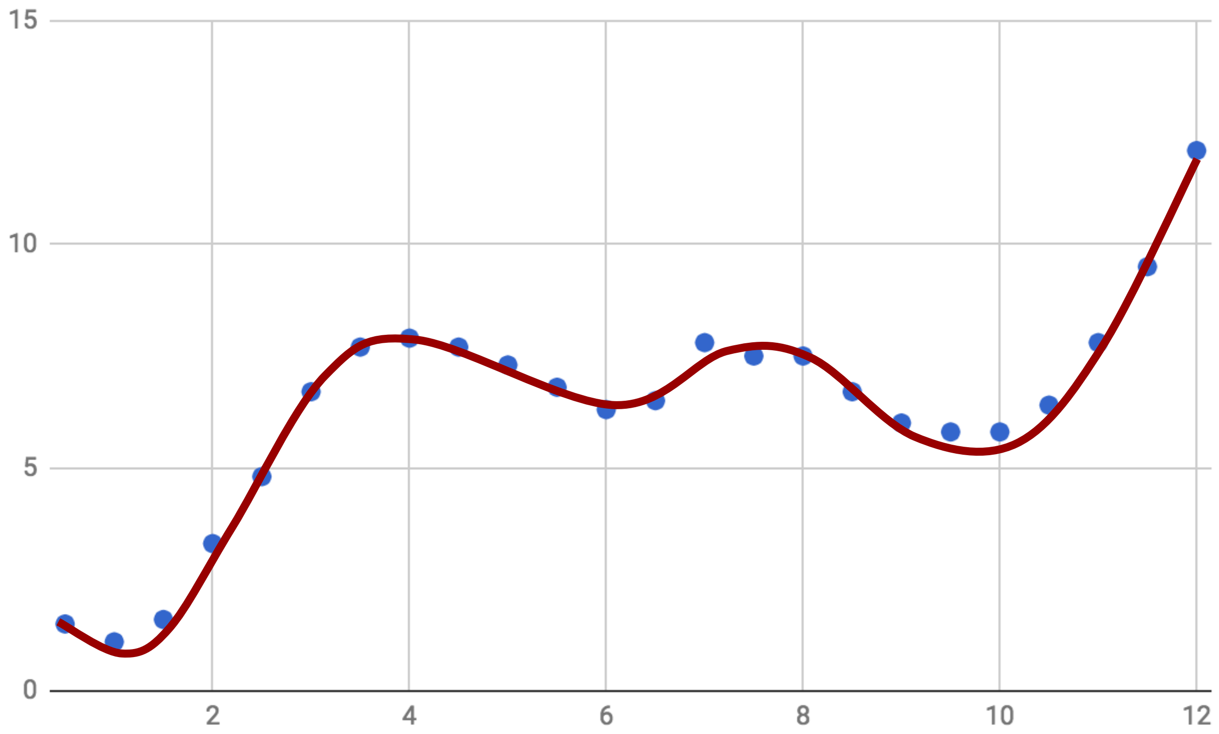
Underfitting: at least the models are consistent...



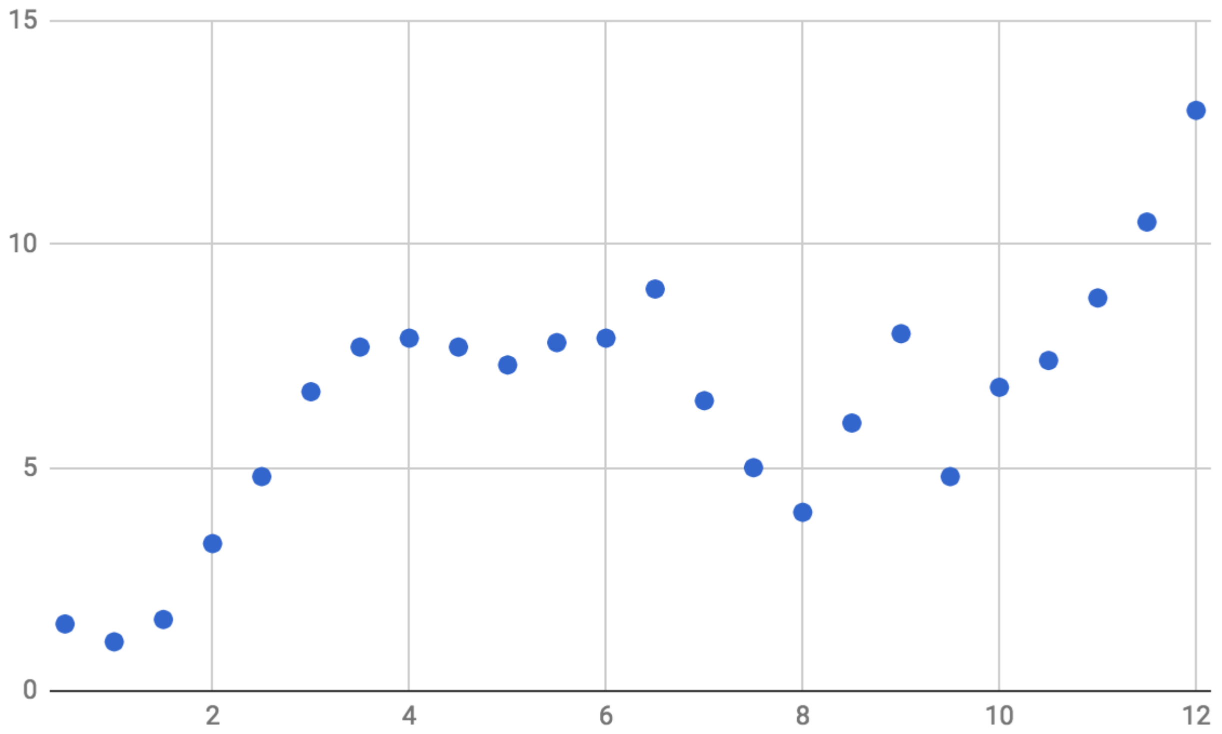
Overfitting: A situation when your model is **too complex** for your data.



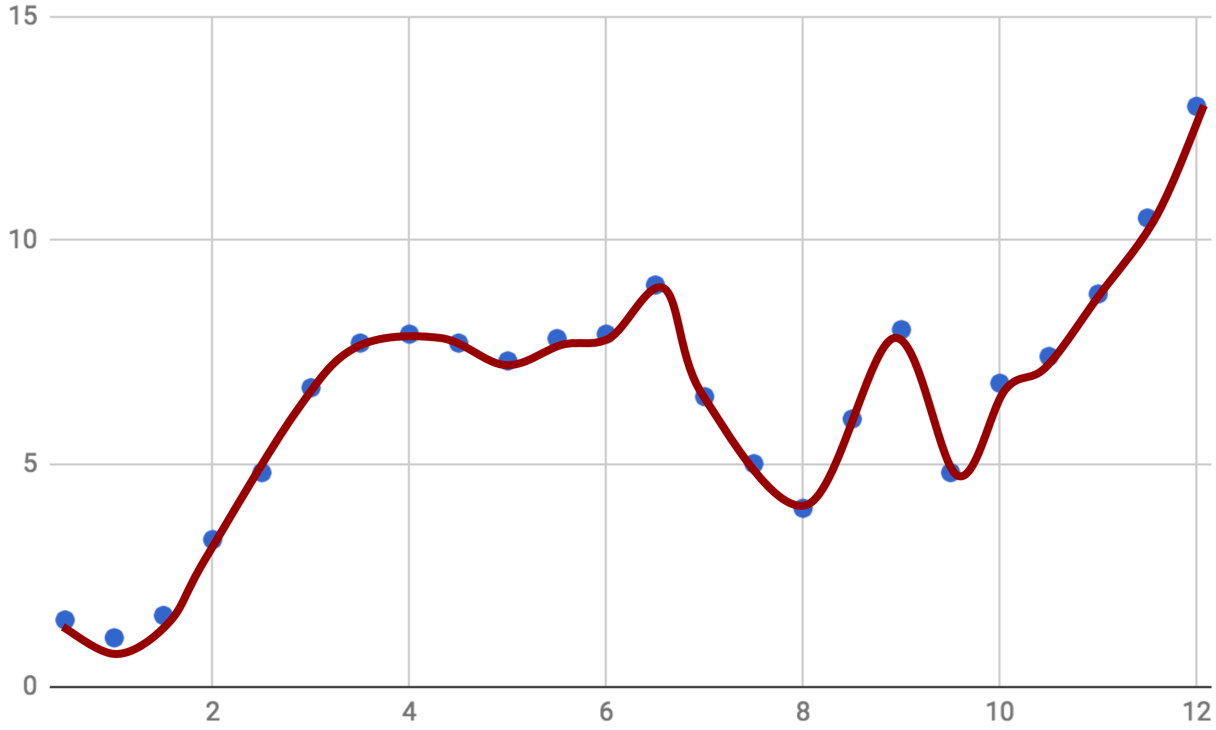
Overfitting



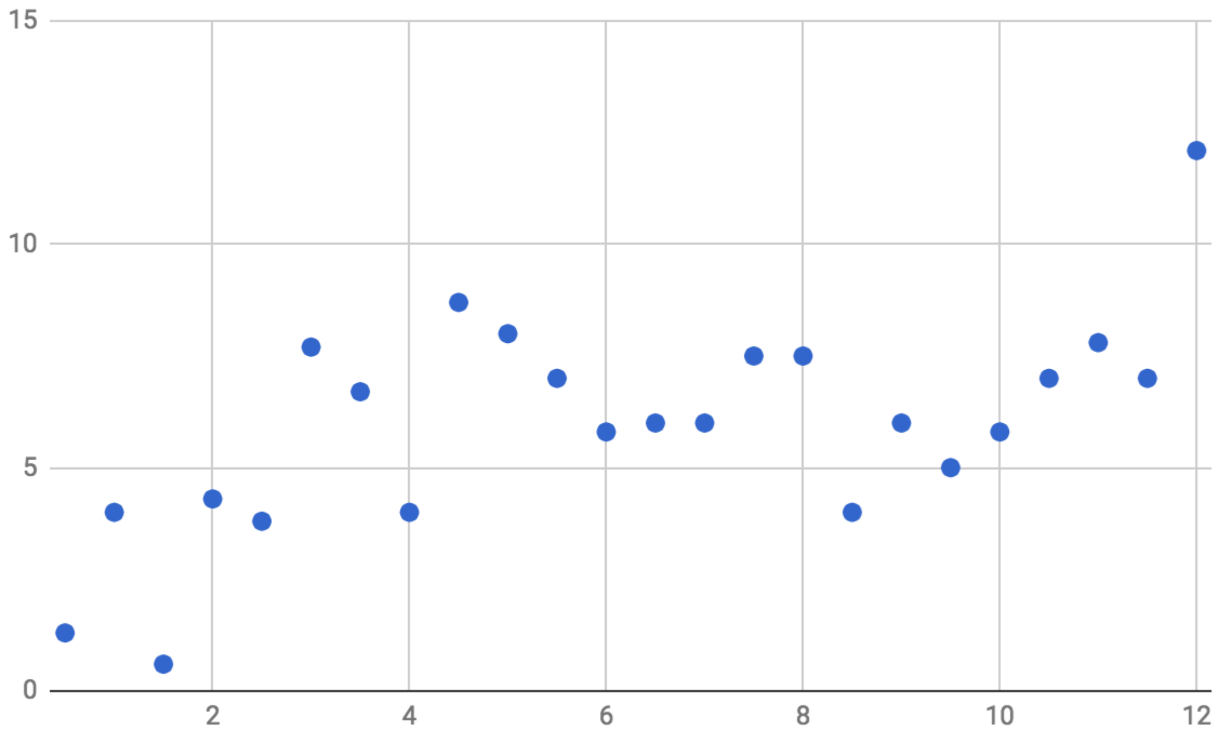
Overfitting



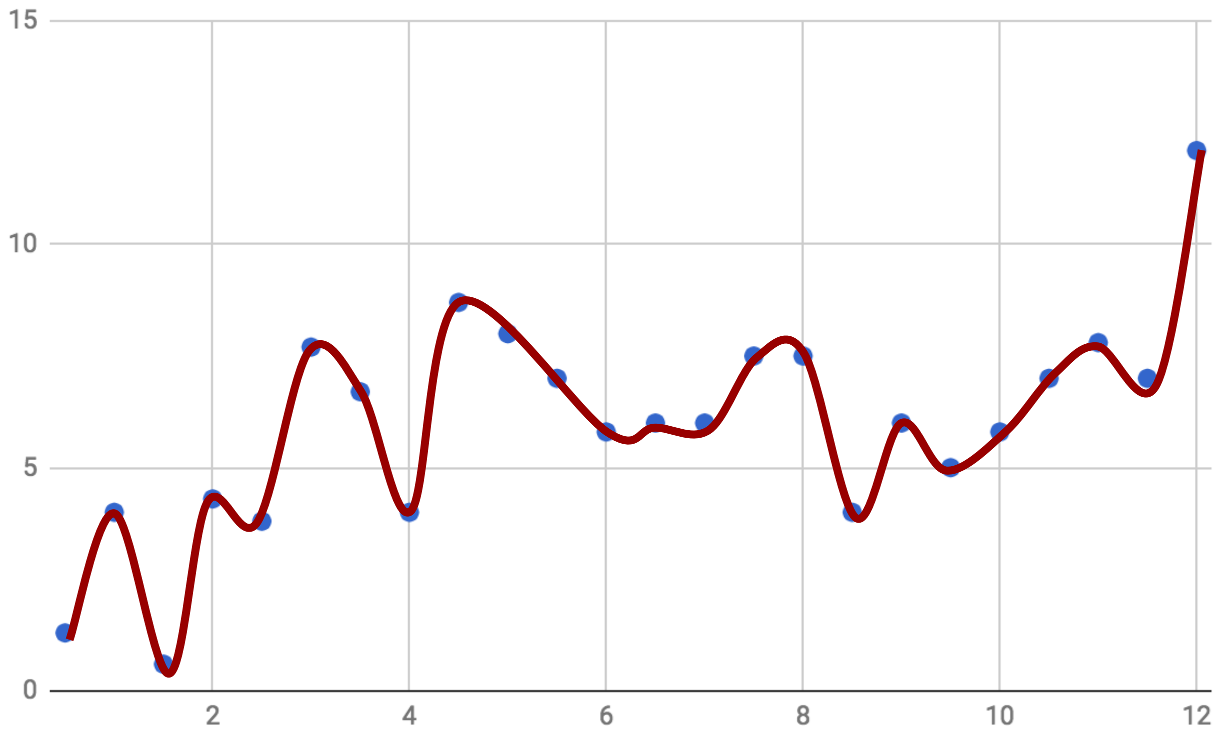
Overfitting



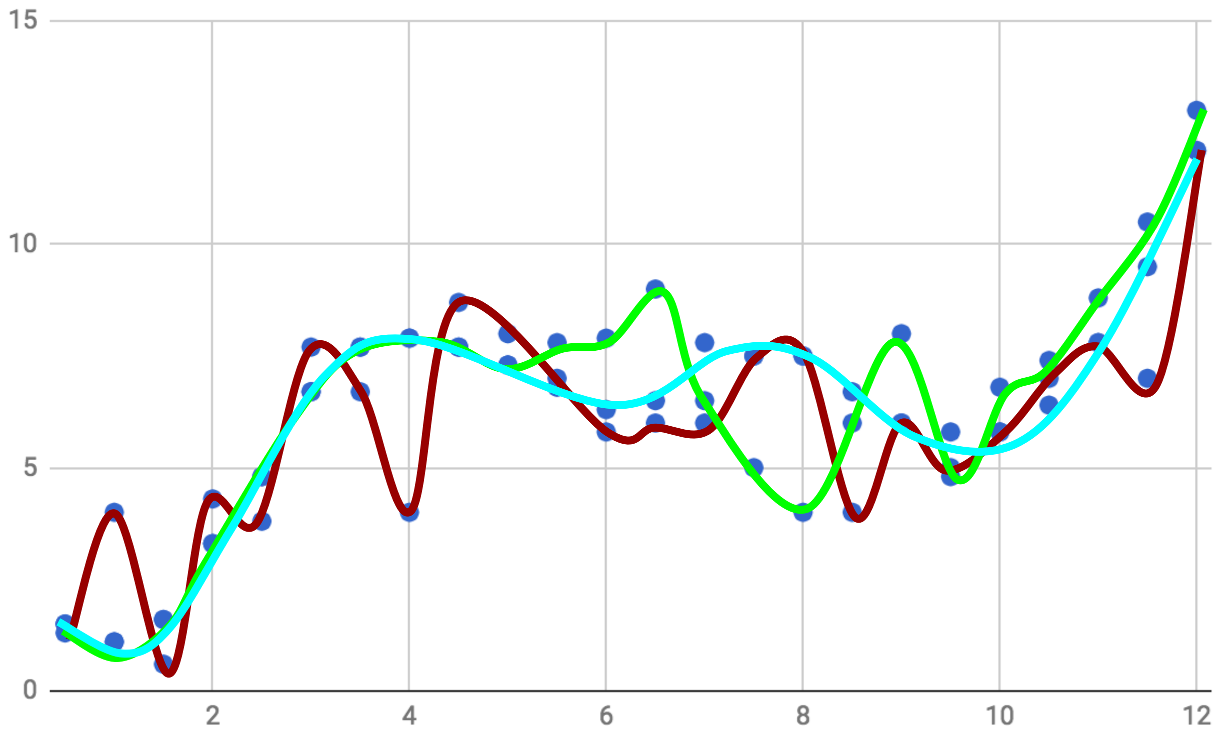
Overfitting



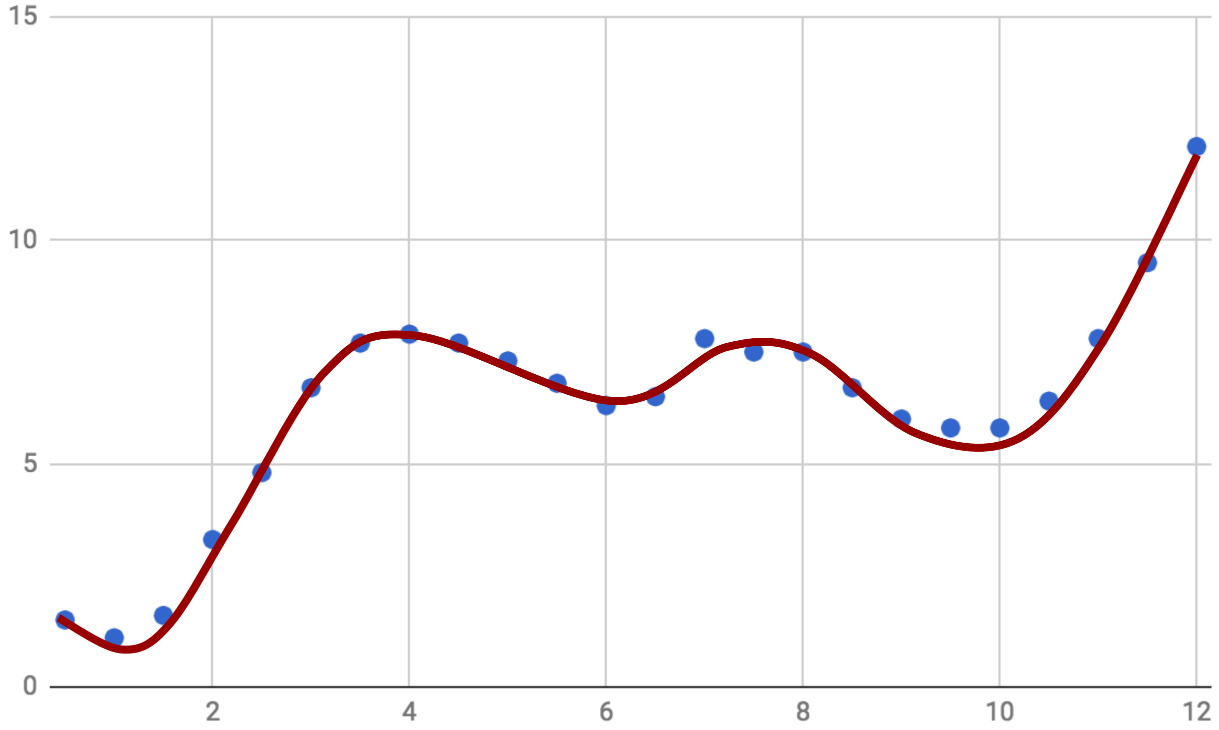
Overfitting



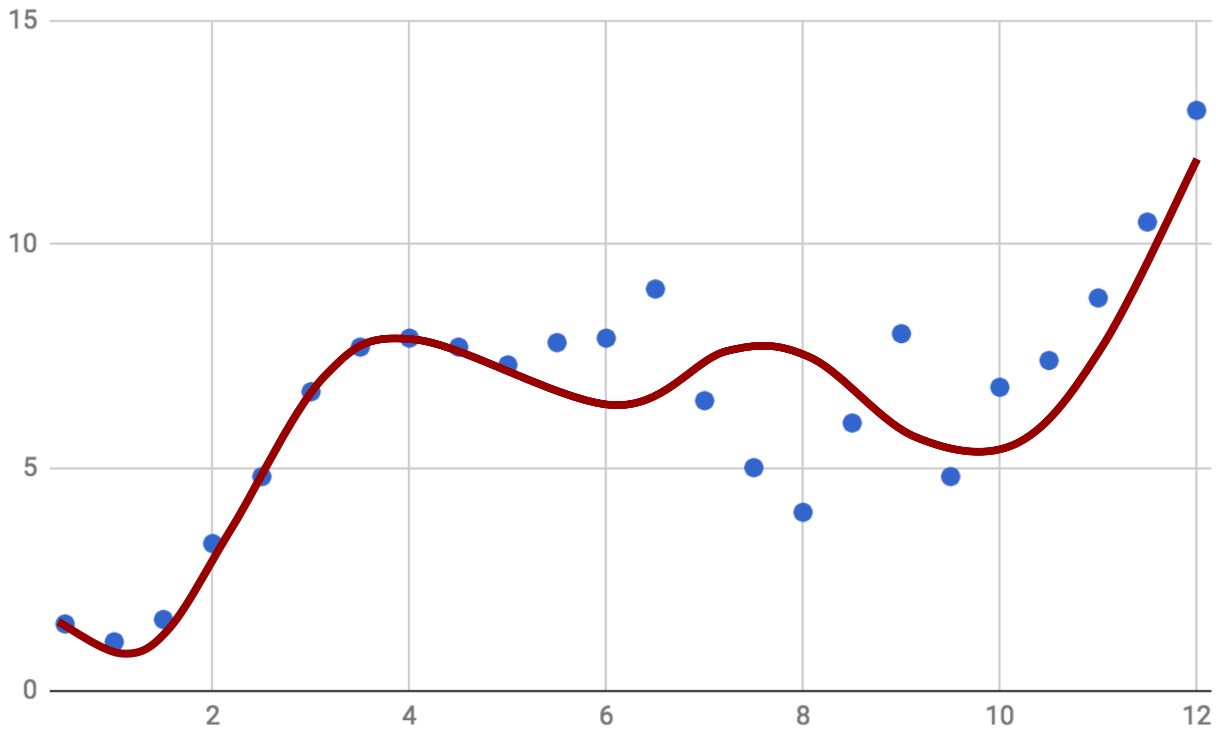
Overfitting: Inconsistent Models!



Overfitting: Results from training with high sensitivity



Overfitting: doesn't generalize well!



Bias and Variance

Definitions

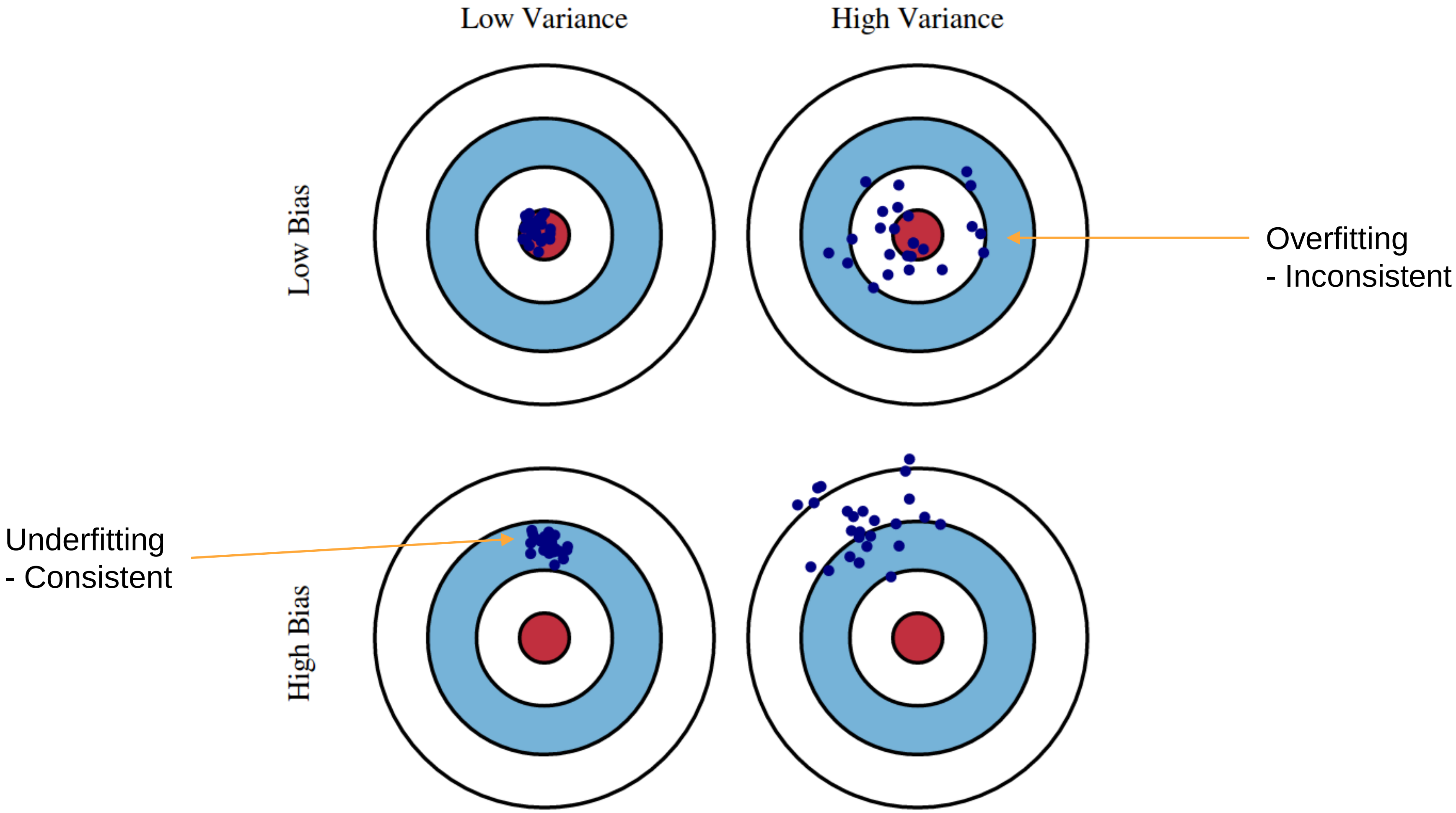
Bias

- A measure of underfitting

Variance

- A measure of overfitting

Either alone is hard to interpret, but together they are helpful!


Low Variance
High Variance
Low Bias
High Bias
Overfitting
- Inconsistent
Underfitting
- Consistent

Balancing Bias and Variance

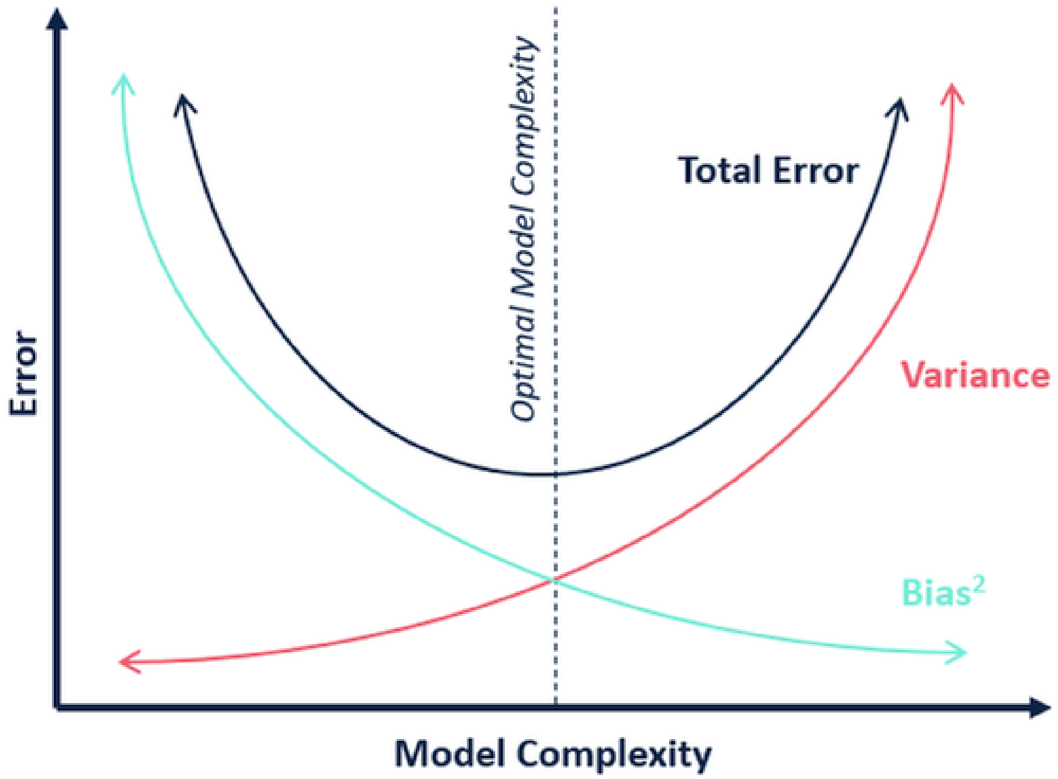
$$E\left[\left(y - \hat{f}(x)\right)^2\right] = \text{Bias}\left[\hat{f}(x)\right]^2 + \text{Var}\left[\hat{f}(x)\right] + \sigma^2$$

$$\text{Bias}\left[\hat{f}(x)\right] = E\left[\hat{f}(x) - f(x)\right]$$

$$\text{Var}\left[\hat{f}(x)\right] = E[\hat{f}(x)^2] - E[\hat{f}(x)]^2$$

Error = (expected loss of accuracy)^2 + inconsistency of model + irreducible error

Balancing Bias and Variance



What does this mean intuitively?

Bias

- Bad
- Results from incorrect assumptions in the learning algorithm

Variance

- Bad
- Results from sensitivity to fluctuations in the data

What can you do to reduce Bias and Variance?

Bias

- Increase model size
- Change type of model
- Add new features

Variance

- Add more data
- Decrease model size
- Reduce features

Feature Selection

(adjusting models)

Methods

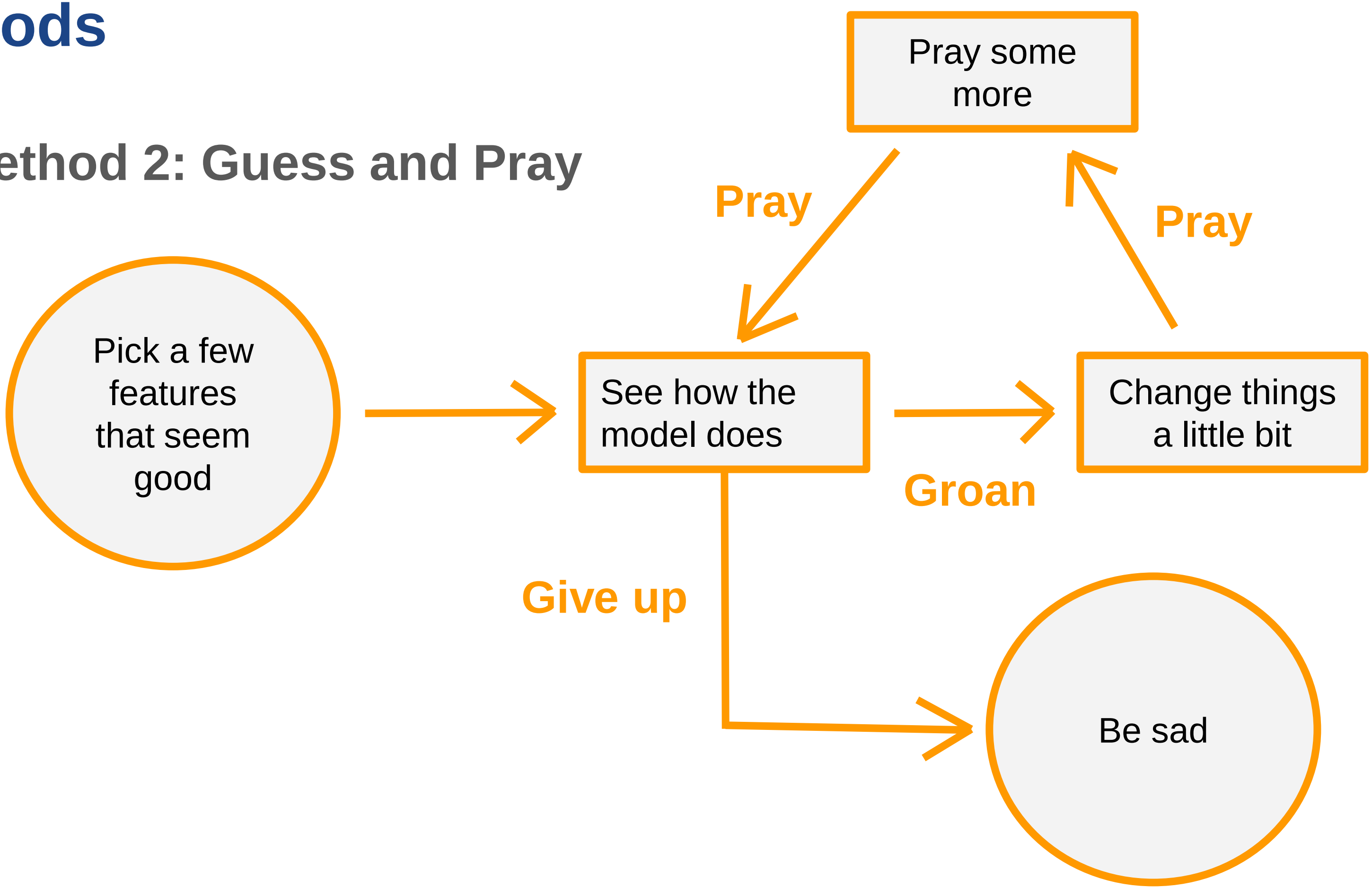
- **Goal:** Find subset of features that gives a good enough model, in a reasonable amount of time.

Methods

- **Goal:** Find subset of features that gives a good enough model, in a reasonable amount of time.
- **Method 1: Best Subset**
  - Test **all** subsets for best one
  - Benefits:
    - **Best** subset out of current features
  - Drawbacks:
    - Slow
    - Even slower with feature engineering

Methods

- **Method 2: Guess and Pray**



Methods

- **Goal:** Find subset of features that gives a good enough model, in a reasonable amount of time.
- **Method 2: Guess and Pray**
  - Guess
  - Benefits:
    - ??
  - Drawbacks:
    - Time consuming for data scientist
    - Unreliable

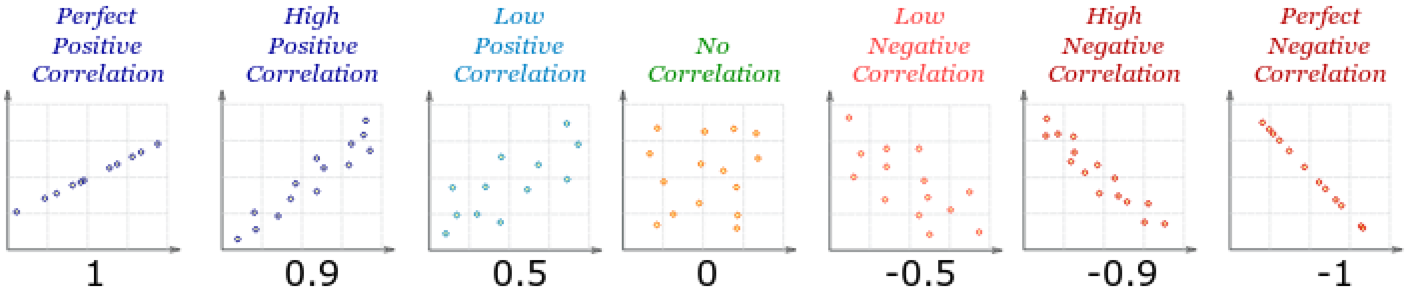
Methods

- **Goal:** Find subset of features that gives a good enough model, in a reasonable amount of time.
- **Method 3: Stepwise**
  - Pick a few features, then programmatically add/remove features using statistics
  - Benefits:
    - Complexity and runtime are adjustable
  - Drawbacks:
    - Can do very badly if you’re not careful
    - Requires more thinking

Correlation, r

The correlation between two variables describes to what extent changing one would change the other.

- Real-valued in [-1,1]
- A variable is always perfectly correlated with itself (correlation=1)



Important Case: Collinearity

Collinear: when two features have a correlation near -1 or 1

- If a feature is collinear with the target, then it's a good choice for linear regression
- If two features are collinear, they're *redundant*
  - Might as well not use one of them
  - Some models *require/assume* no collinear features
  - Takes more time, and doesn't add much information at the cost of *increased variance/sensitivity*

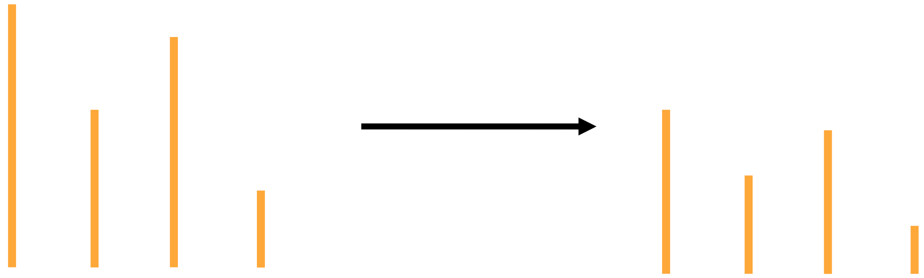
Side Note: Scaling and Normalizing

- Some models require data to be centered
- Some models need features to be on the same scale
    - Divide by max, minus min divide by max minus min, minus mean divide by standard deviation.

Side Note: Scaling and Normalizing

- Some models require data to be centered
- Some models need features to be on the same scale

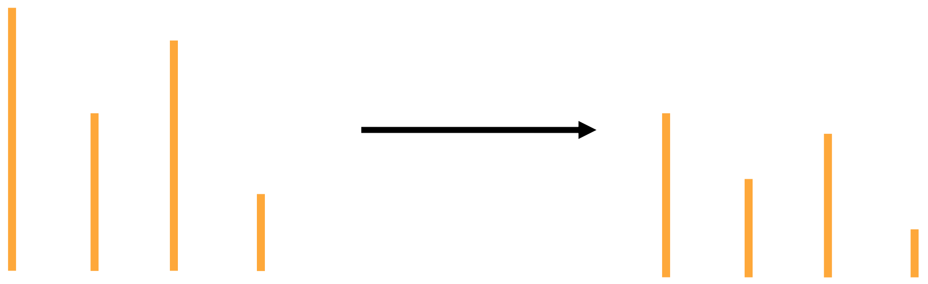
Divide by max: Bounds data <= 1



Side Note: Scaling and Normalizing

- Some models require data to be centered
- Some models need features to be on the same scale

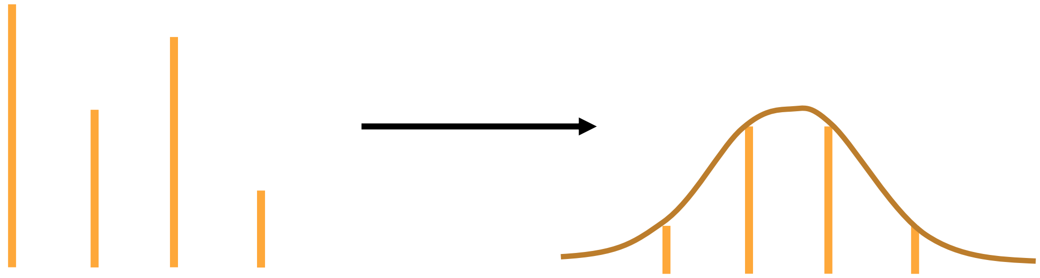
min / (max – min) : Bounds data between [0,1]



Side Note: Scaling and Normalizing

- Some models require data to be centered
- Some models need features to be on the same scale

mean / standard deviation: Z scores – Distance from mean



Other Ways to Adjust your Model

- Hyper Parameters – Ex: Learning rate, etc.

- Feature engineering – Ex: Manipulating dataset

- Just changing to a different algorithm

Demo

Final Notes

Always remember both bias and variance!

Coming Up

- Assignment 5: **Next Wednesday**, October 18th
- Next Lecture: Intro to Classification
- Last day to drop: **Next Monday**, October 16th

CDS Education

We explore, learn, and educate big minds.