INFO 1998: Introduction to Machine Learning

Download Lecture5Homework.ipynb, lecture5dataA.csv, and lecture5dataB.csv

(also pull up Lecture4Homework.ipynb — you'll find it helpful)

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

Complete in OH or after lecture anytime between now and **next Friday (03/22)**.

Cornell Drop Deadline: **March 18th**

What We'll Cover

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

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

Agenda

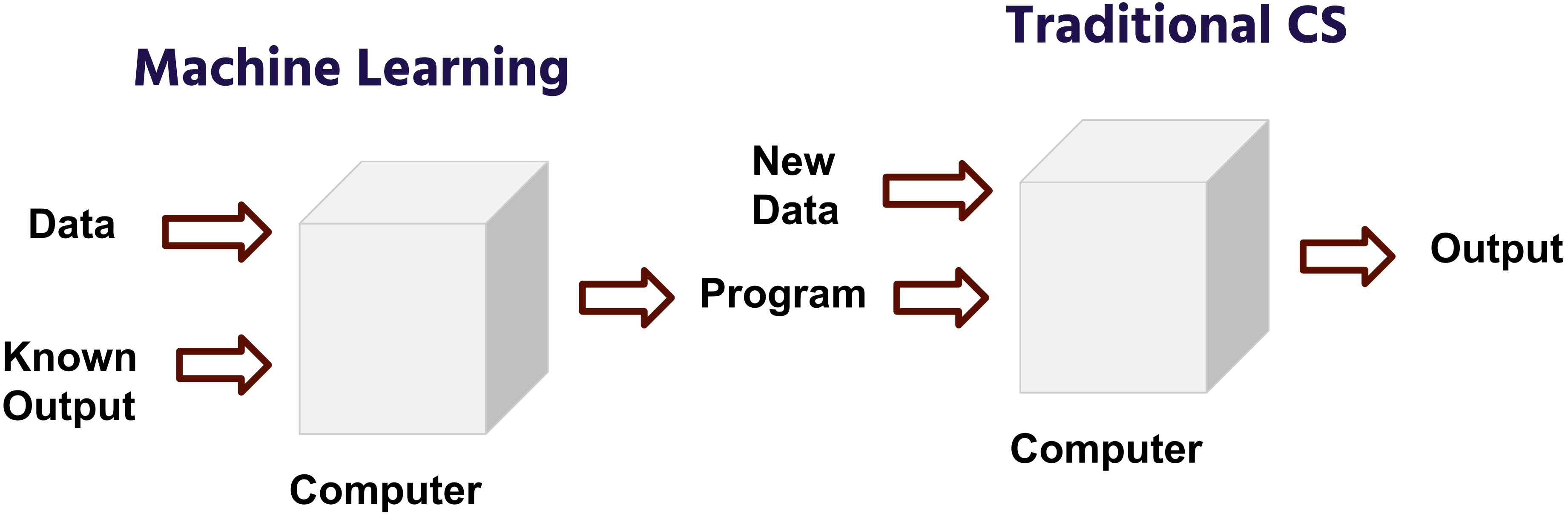
1. Review
2. Measuring Accuracy
3. Bias-Variance trade-off
4. Feature Selection
5. Other Types of machine learning

Review: Defining ML

We want to predict the future
- Take some known input and output
- Learn the data’s pattern and come up with a way to, given a future input, predict the corresponding output

Now: *how* do we learn the data’s pattern?

Review: ML Pipeline

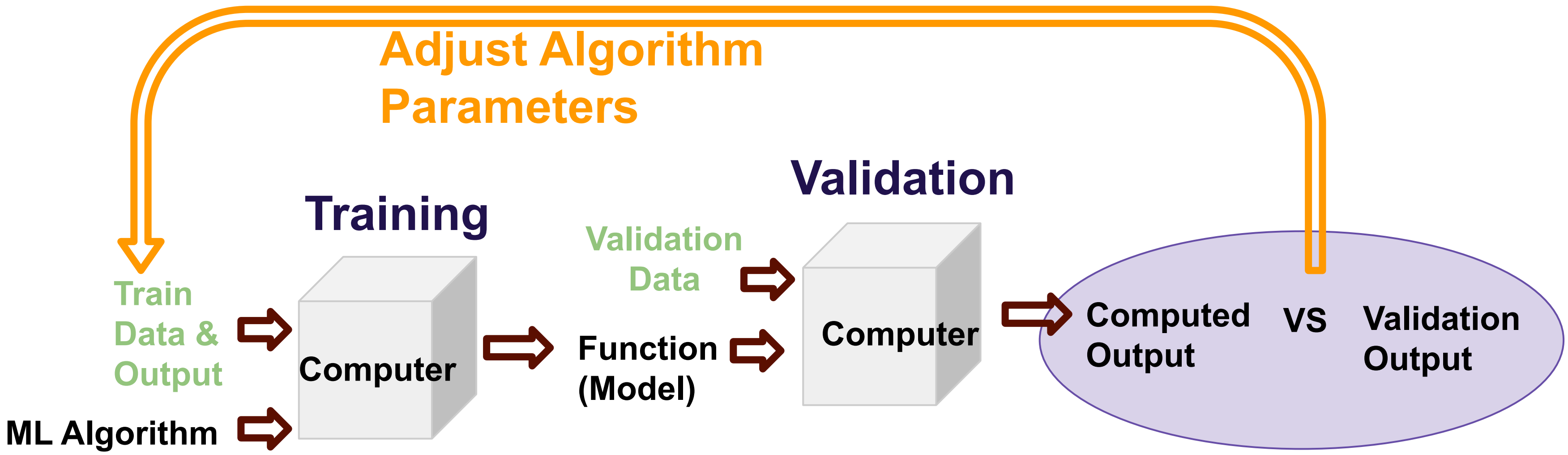


Review: Model

- “Model training” = learn a relationship/program
- “Model validation” = see if the learned relationship is accurate on data not part of your training set
- “Model testing” = final model performance

Measuring Bias / Loss

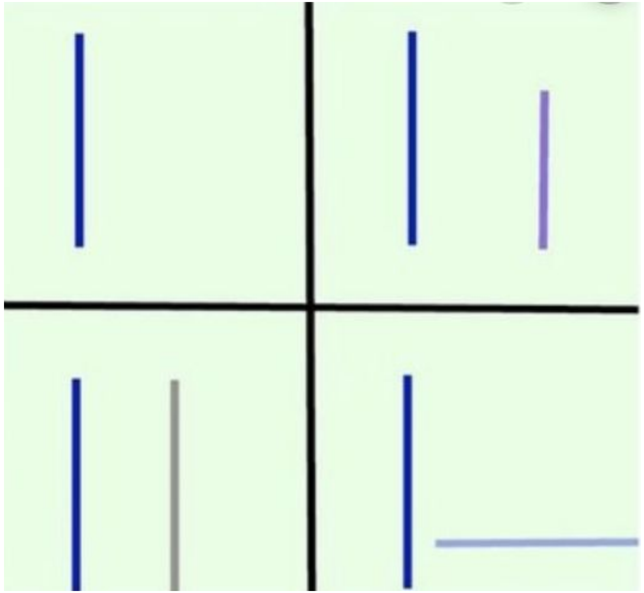
(training accuracy)



1. Split data (lecture 7)
2. Assess model accuracy (today)
3. Adjust Model (a bit today)

Loss, Cost, and Score Functions

- **Loss Function**
  - Penalty for mislabelling a single data point
- **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)
- **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



Linear Regression Loss Formula: Euclidean Distance

$$\text{loss}(x_i, y_i) = (h(x_i) - y_i)^2$$

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

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

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*

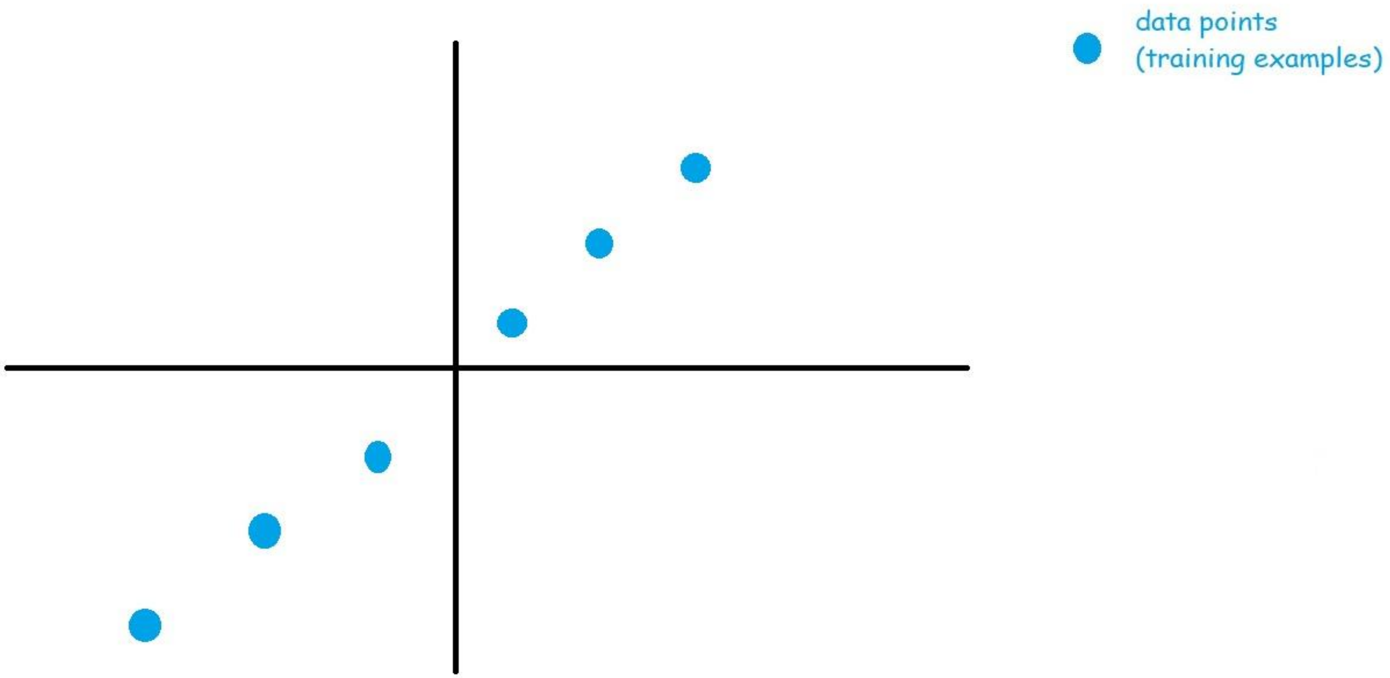
Cost -> Accuracy Score

- sklearn's score function is:

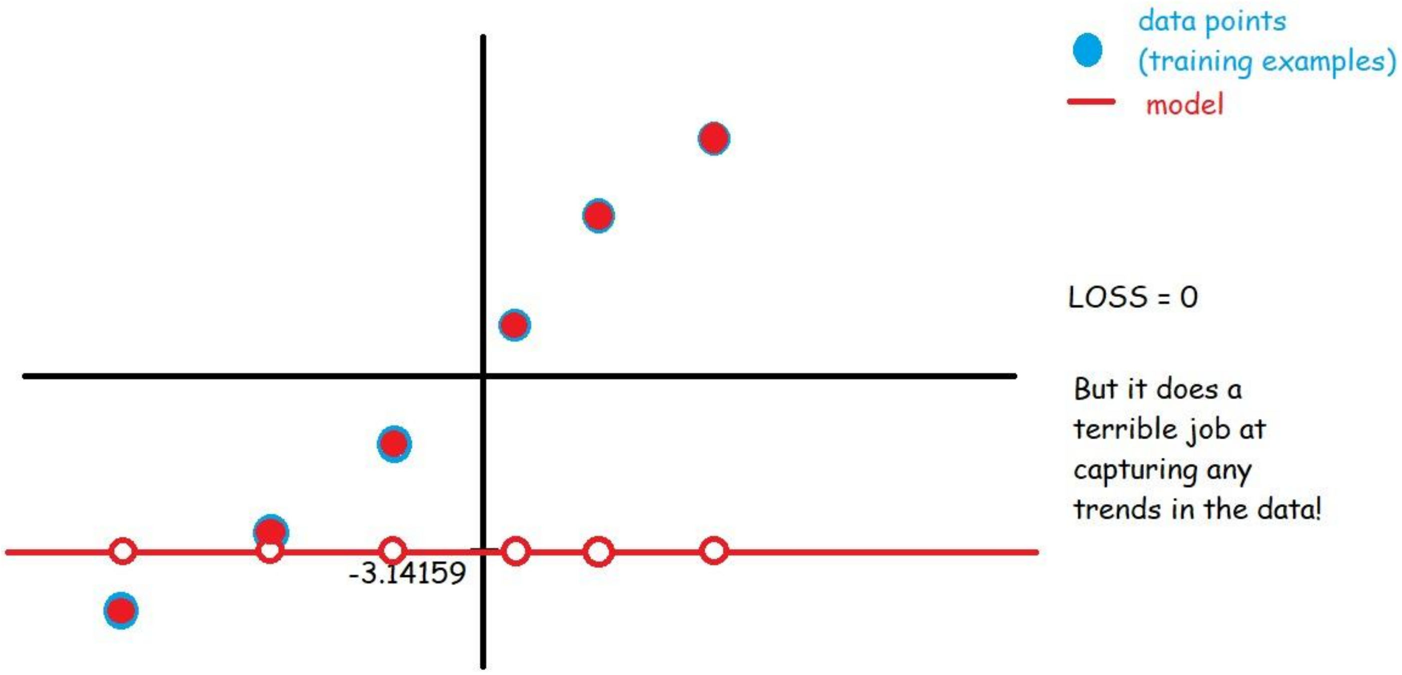
  1 - ([Cost of model] / [Cost of baseline])

- **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

Training Data



Cost = 0, but model is horrible…



MORAL: Assumptions are important!

Overfitting and Underfitting

(what makes a model good?)

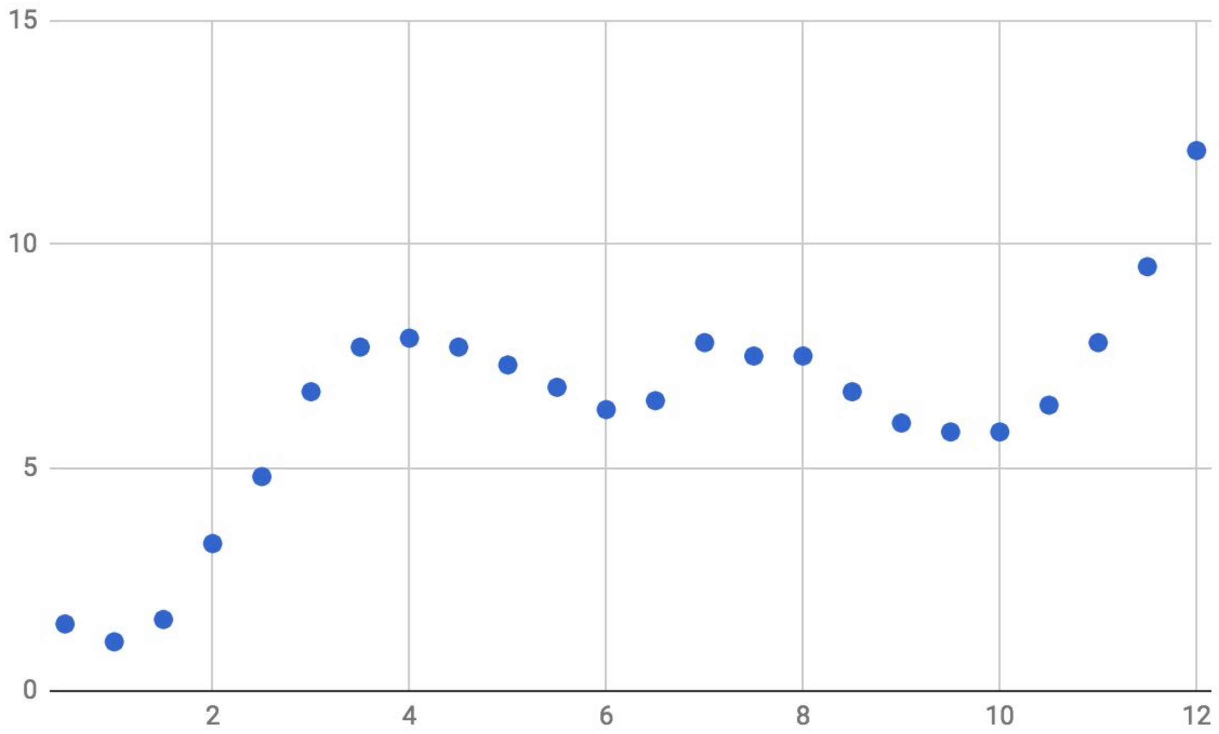
Model Goals

When training a model, we want our model to:

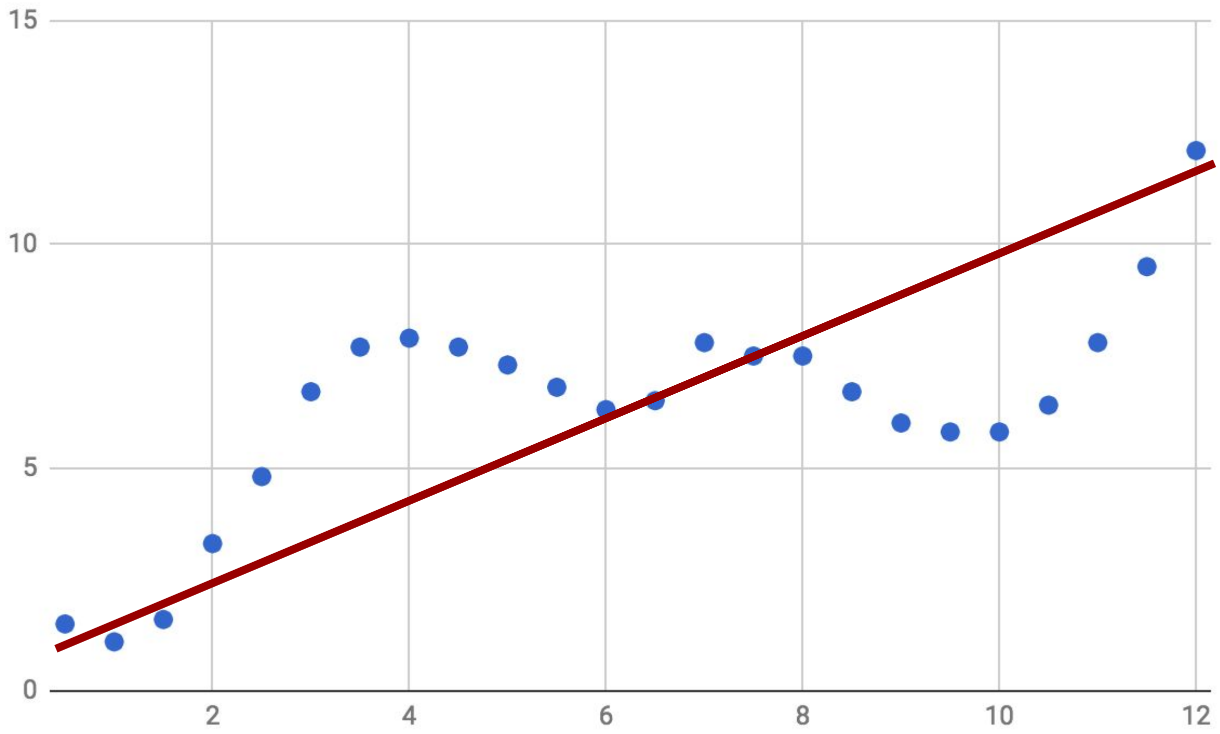
- Capture the trends of the training data
- Generalize well to other samples of the population
- Be moderately interpretable

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

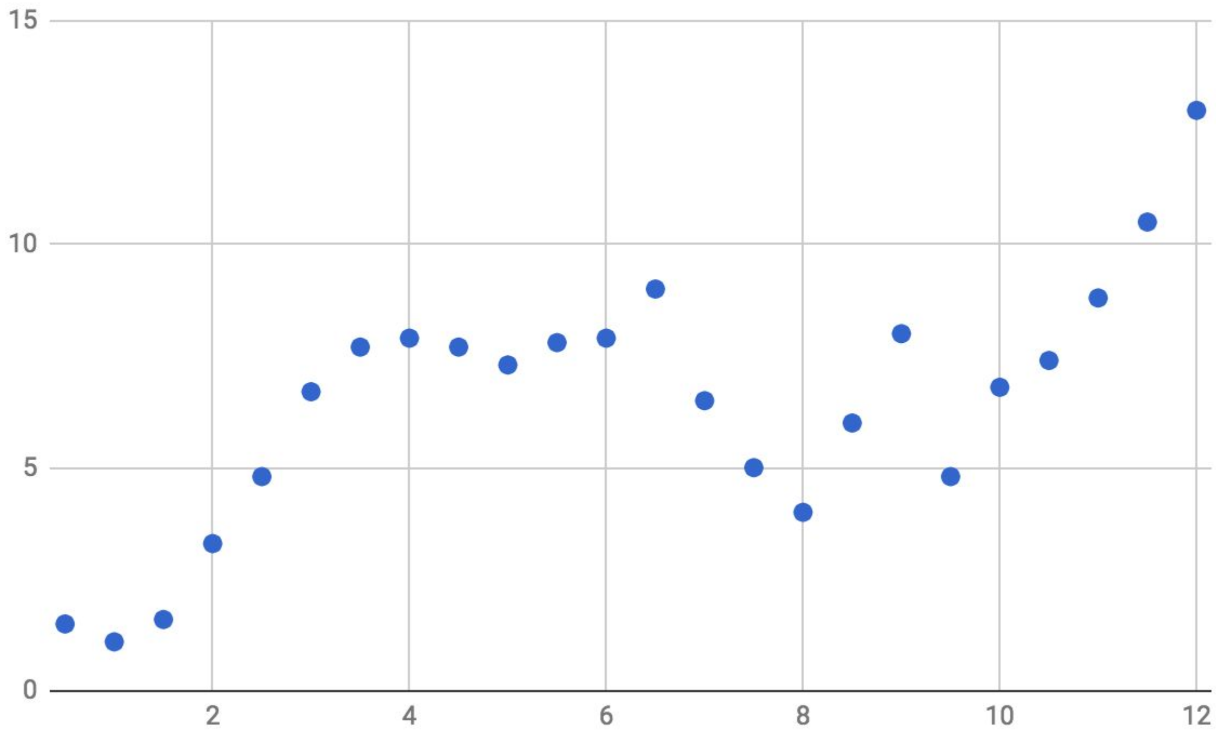
Underfitting



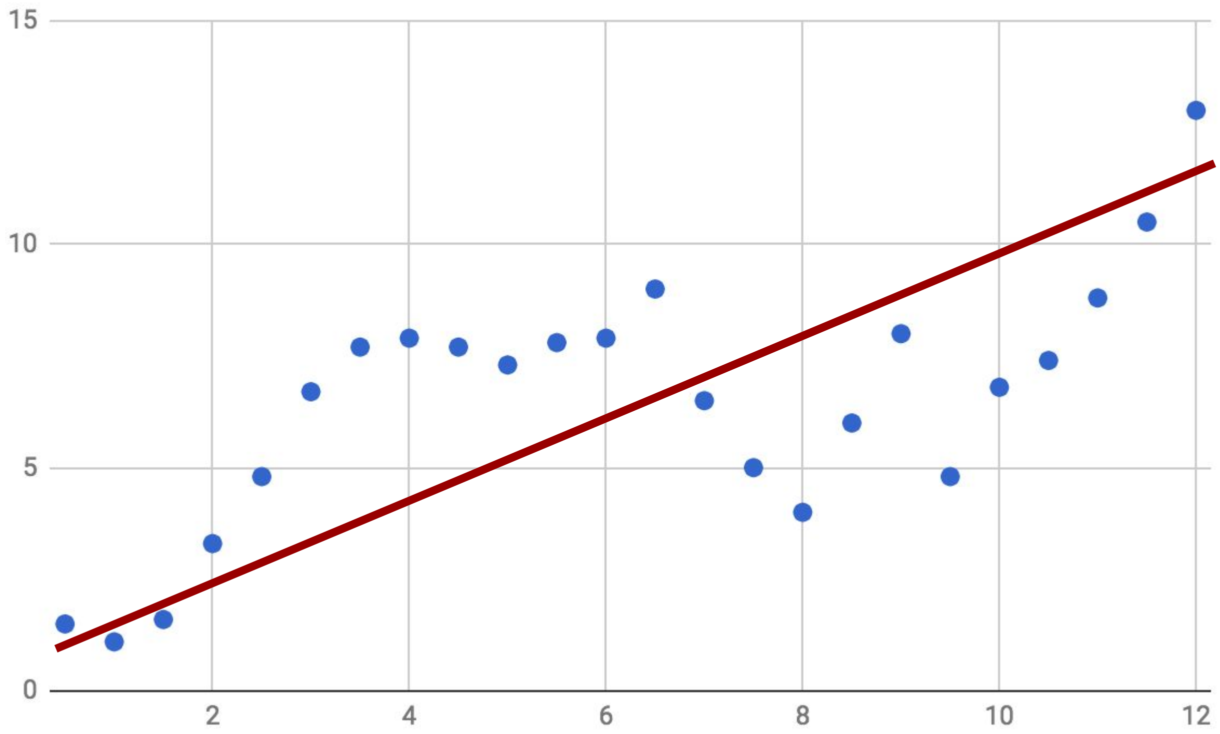
Underfitting



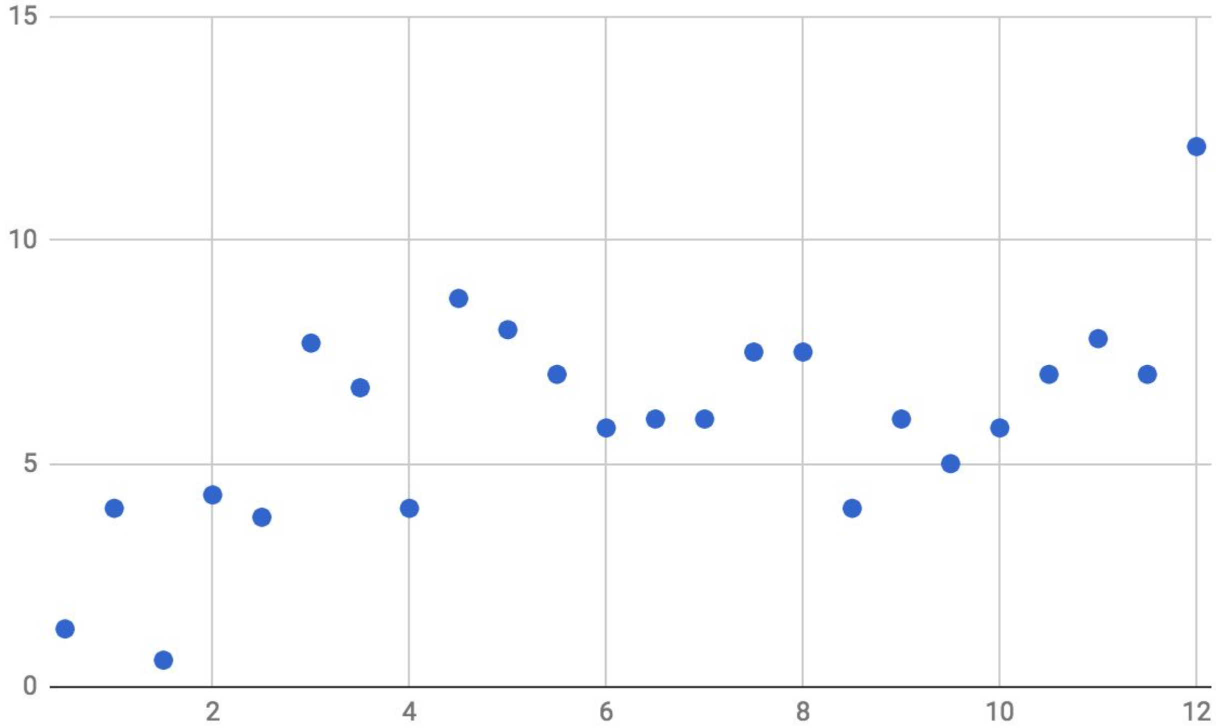
Underfitting



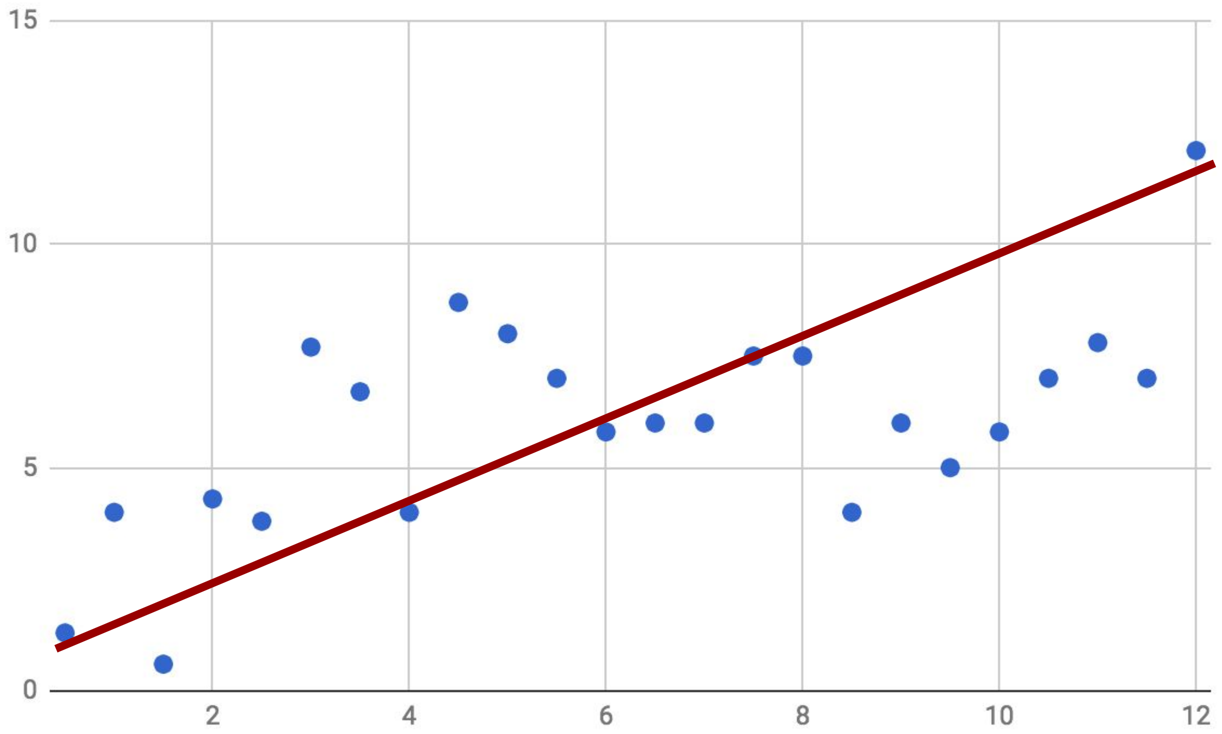
Underfitting



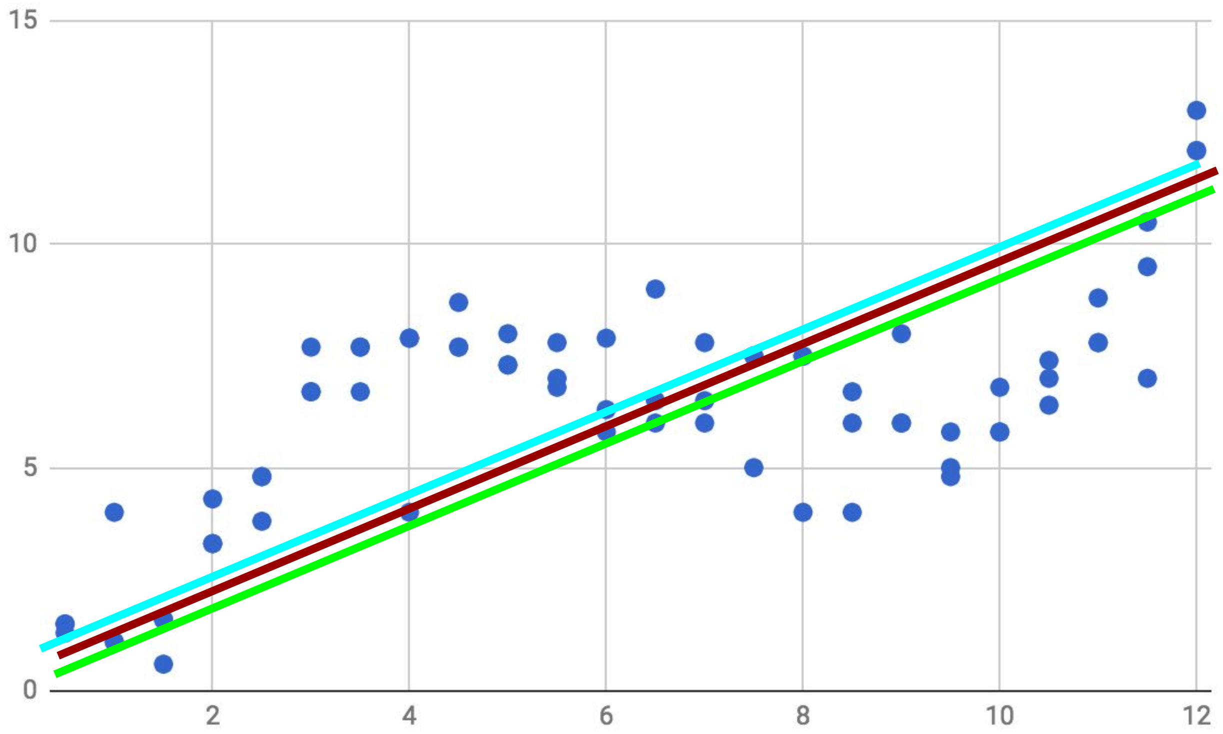
Underfitting



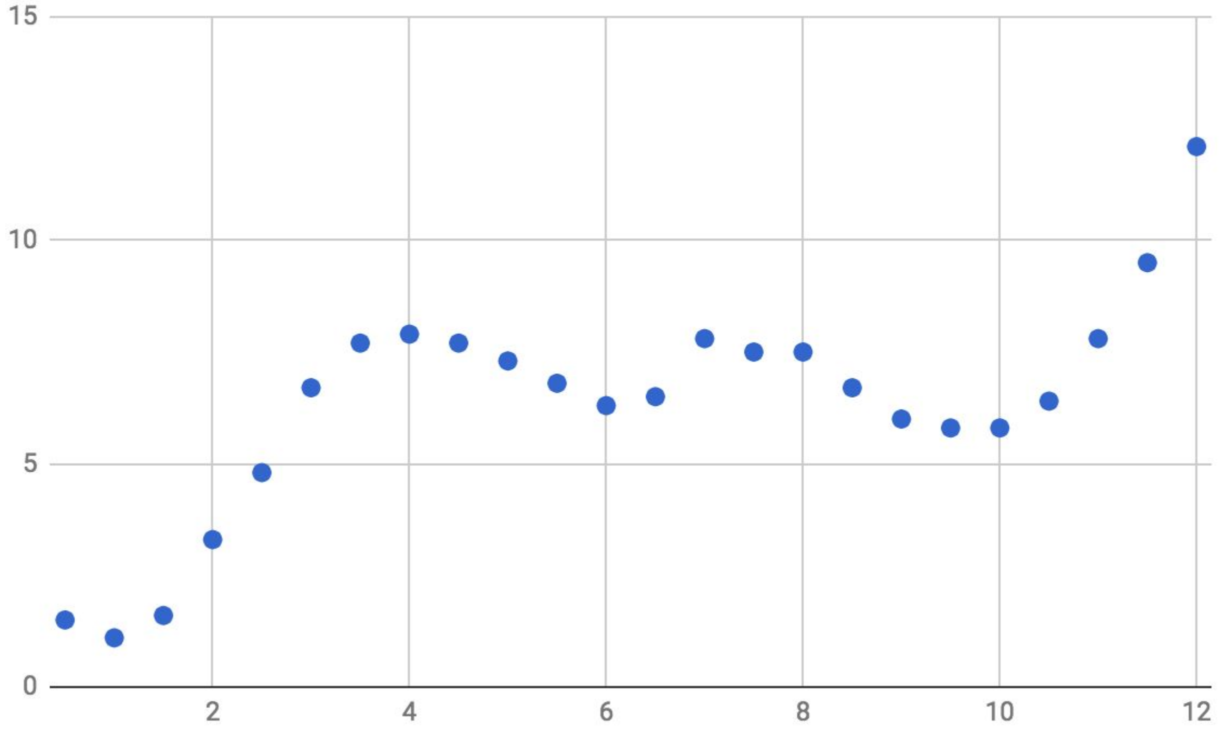
Underfitting



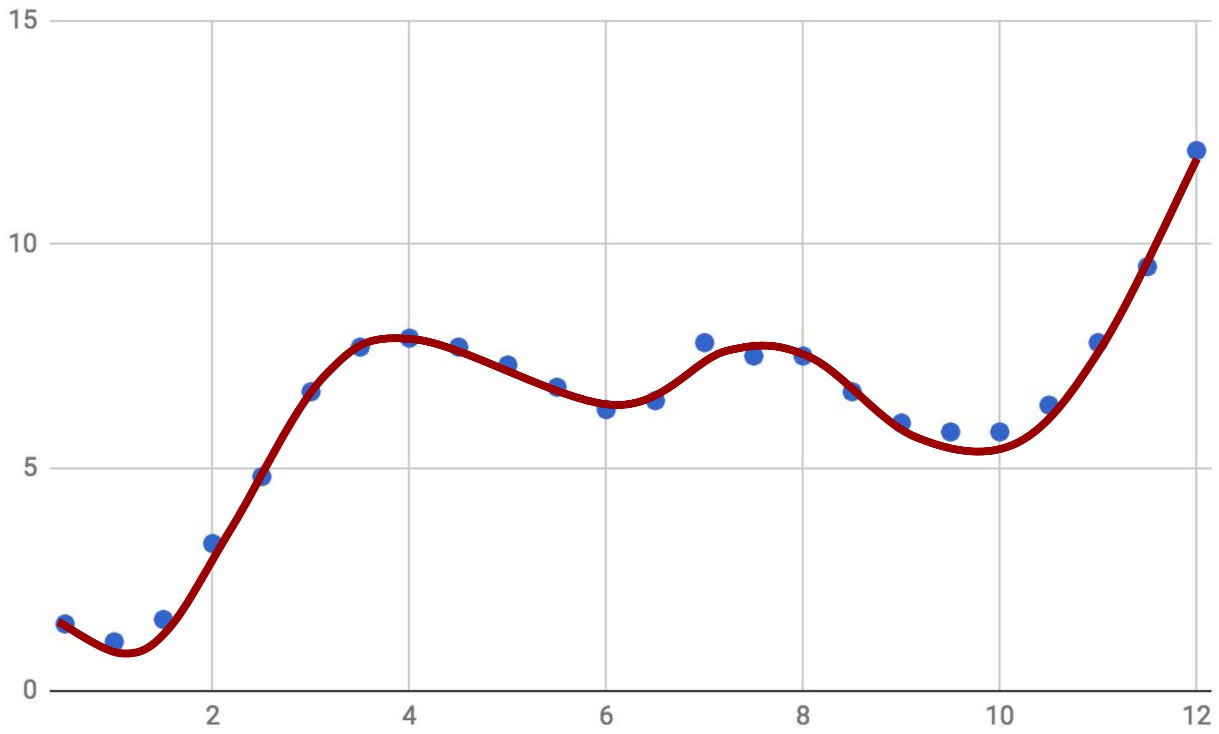
Underfitting: at least the models are consistent...



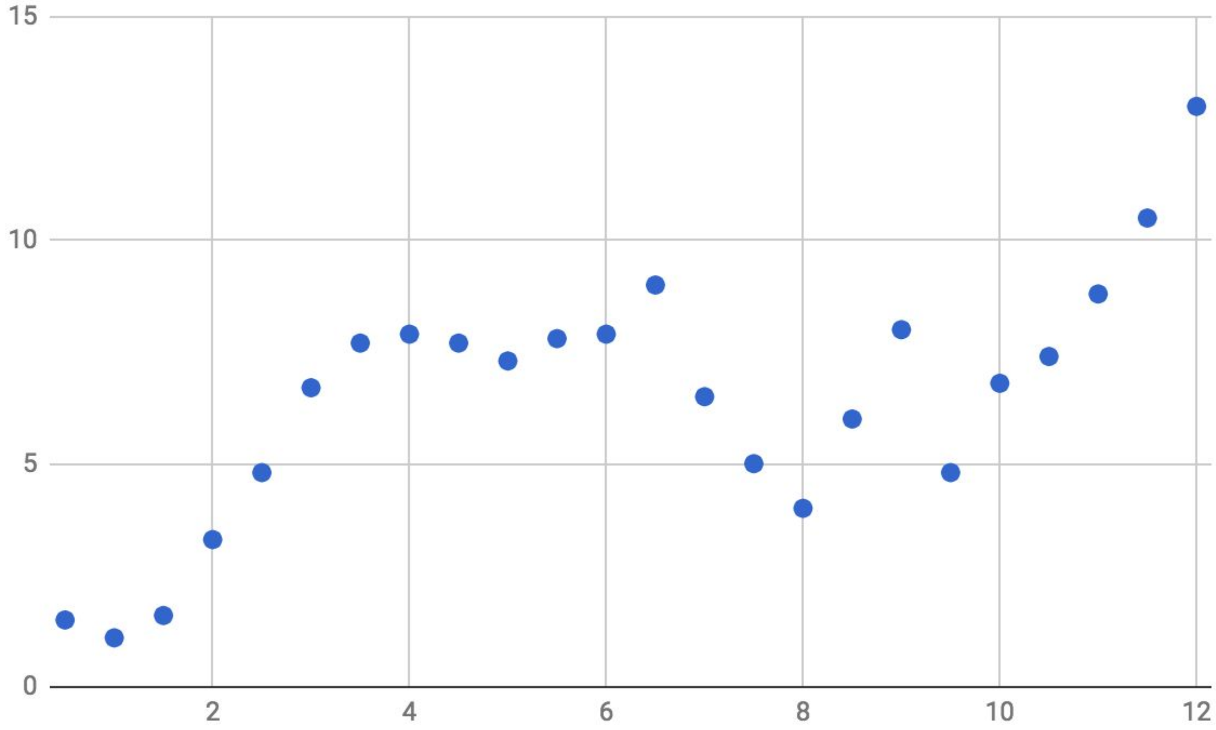
Overfitting



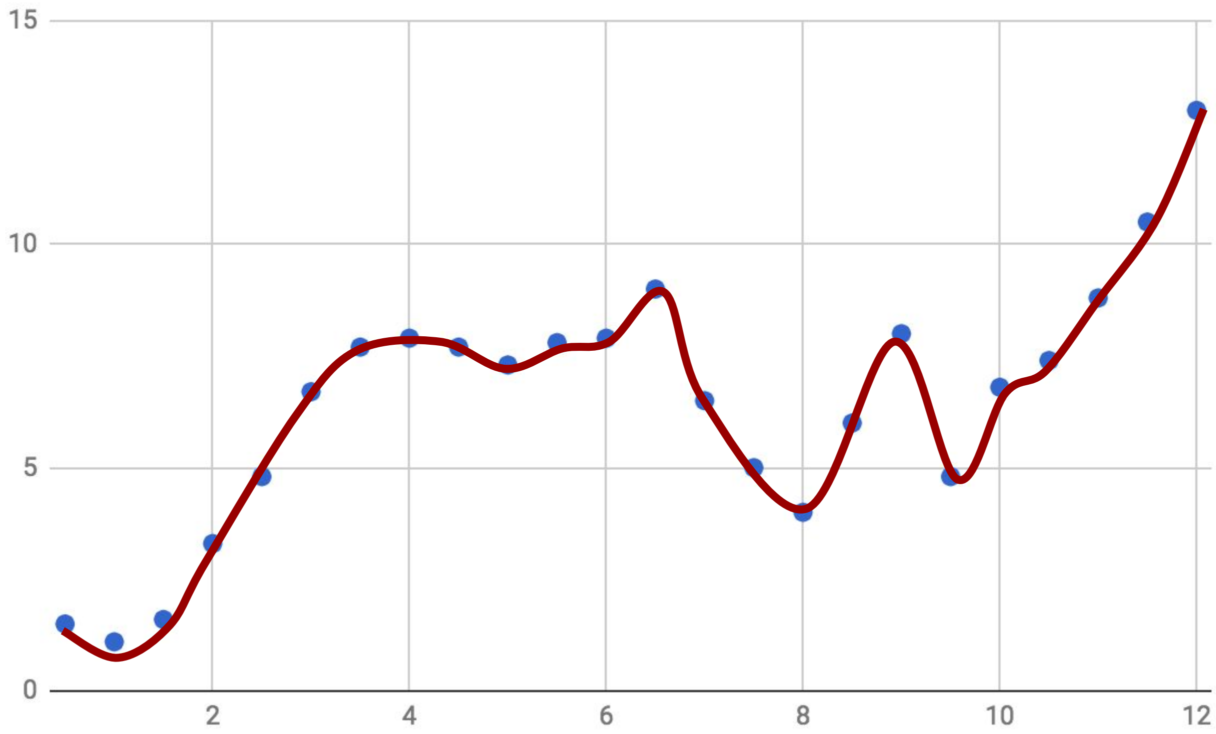
Overfitting



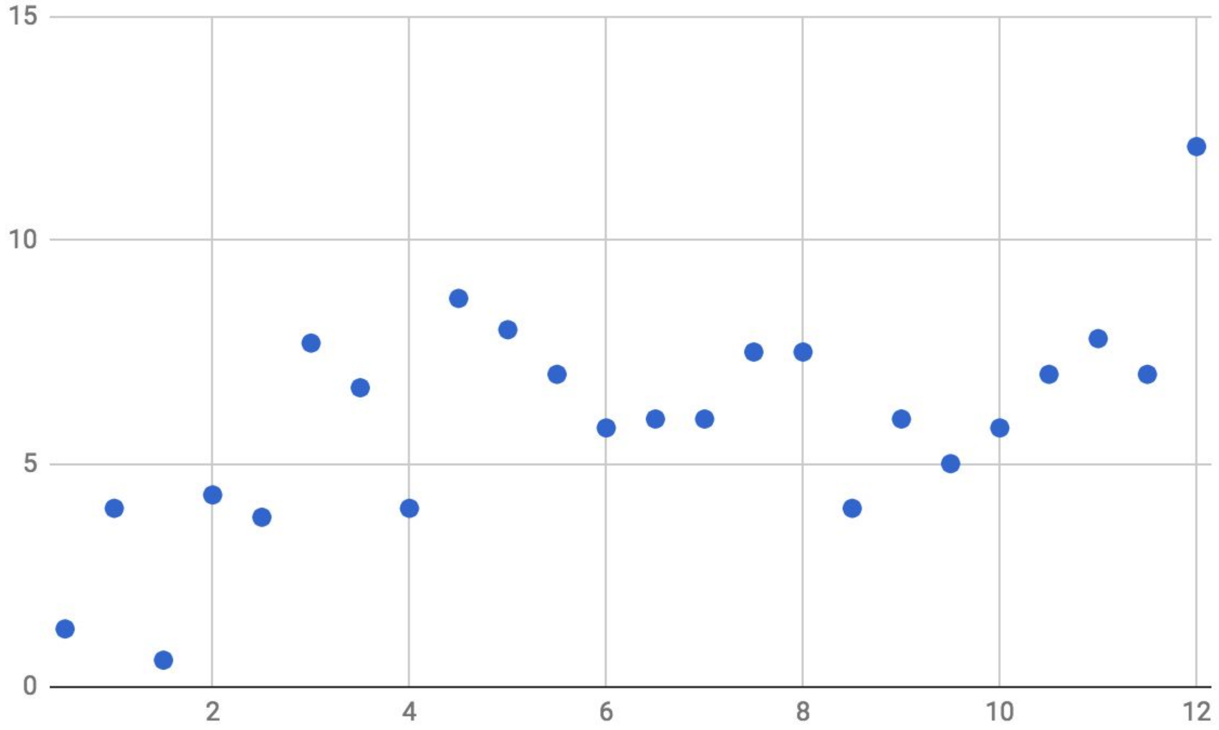
Overfitting



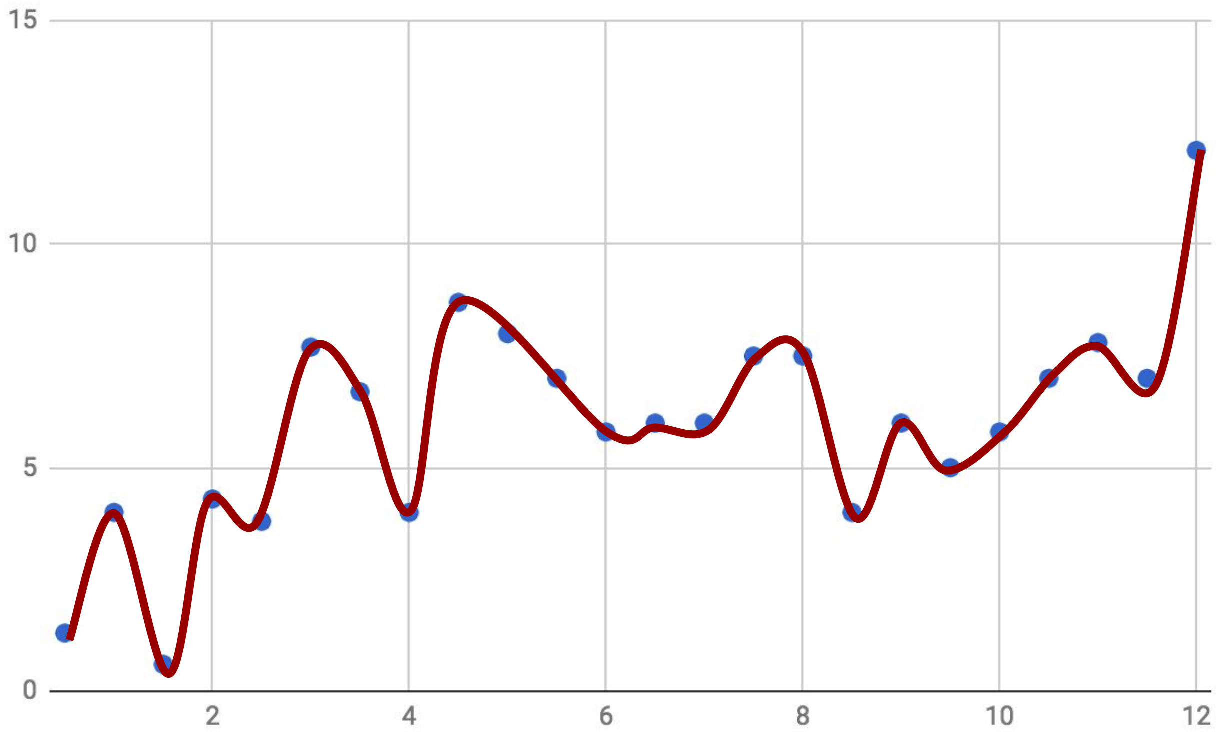
Overfitting



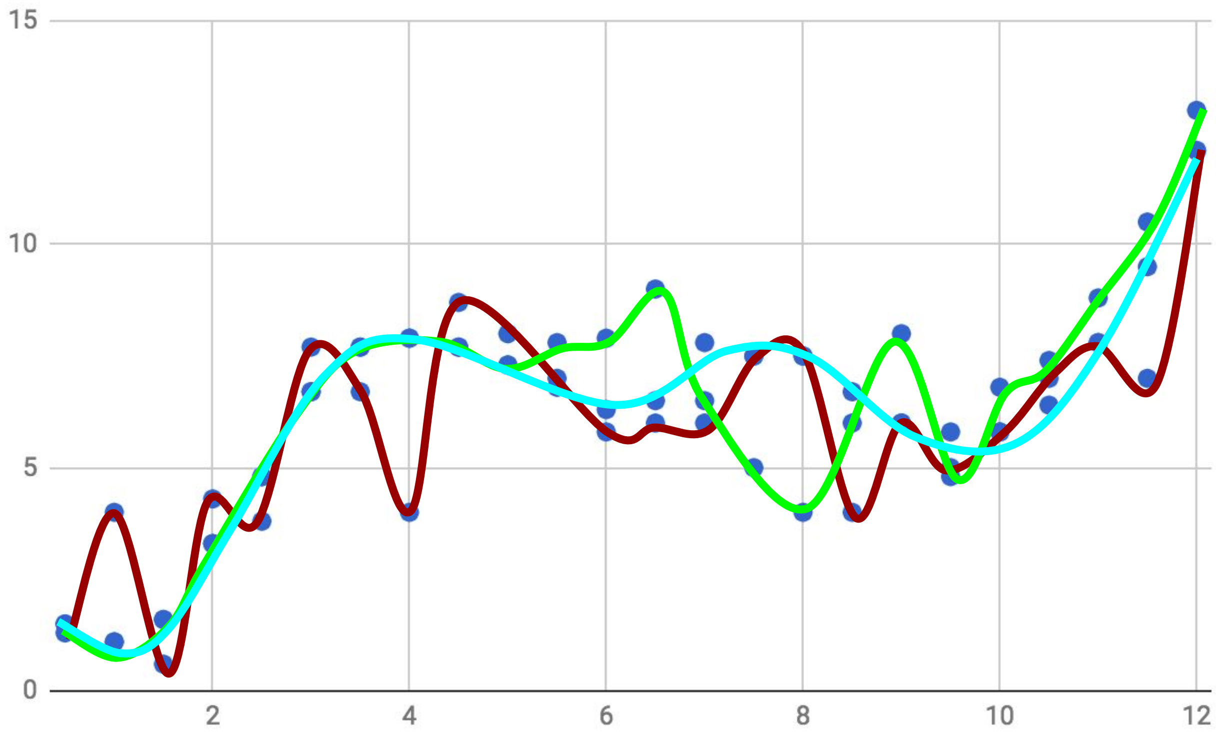
Overfitting



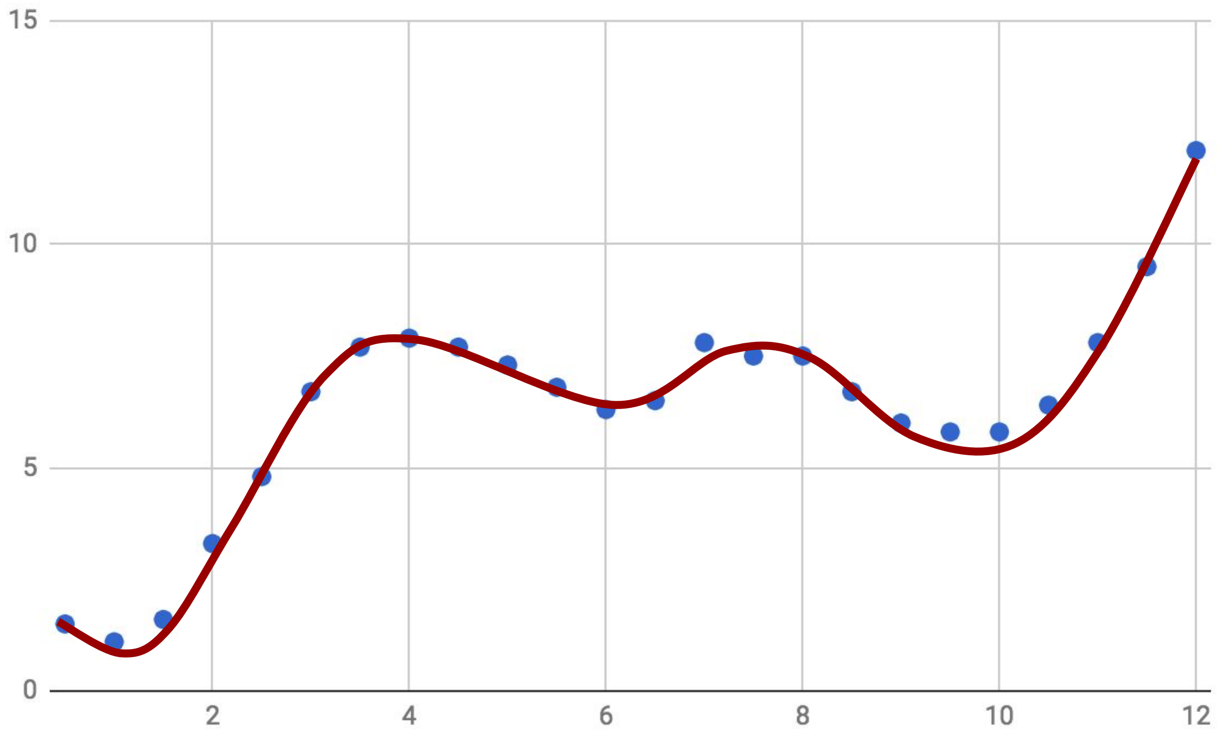
Overfitting



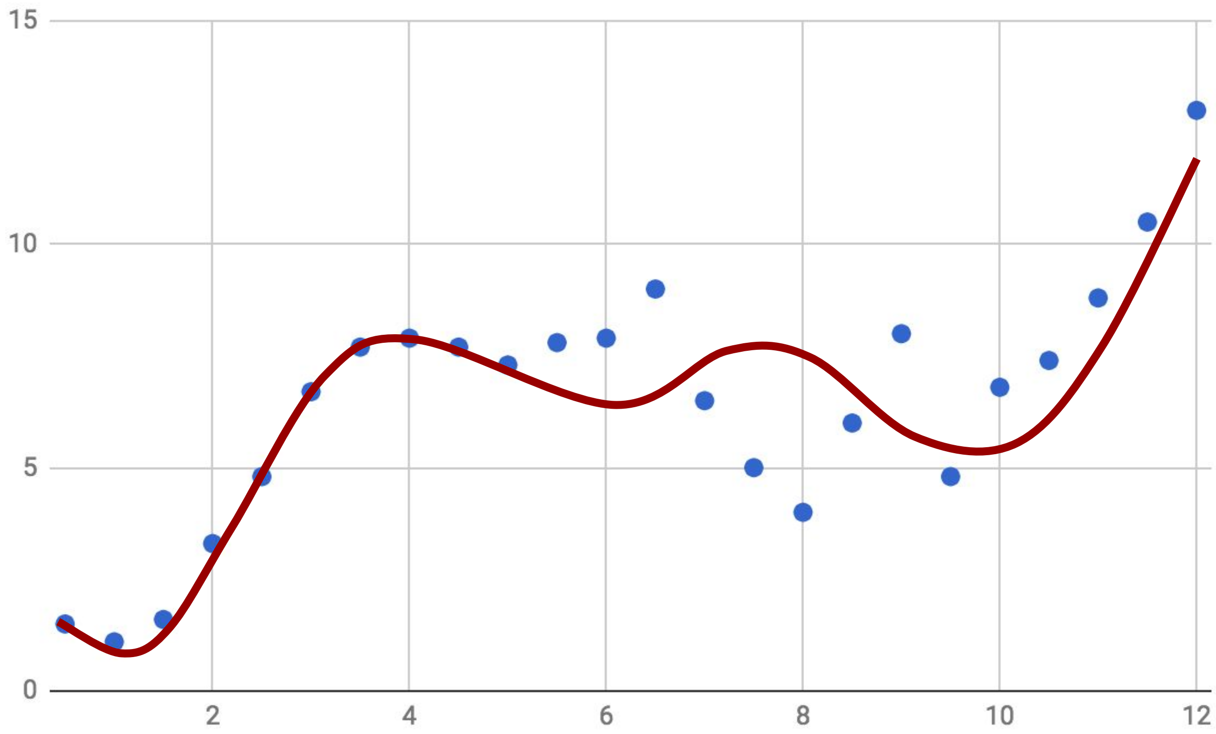
Overfitting: Inconsistent Models!



Overfitting: Results from training with high sensitivity



Overfitting: doesn't generalize well!



Definitions

Bias

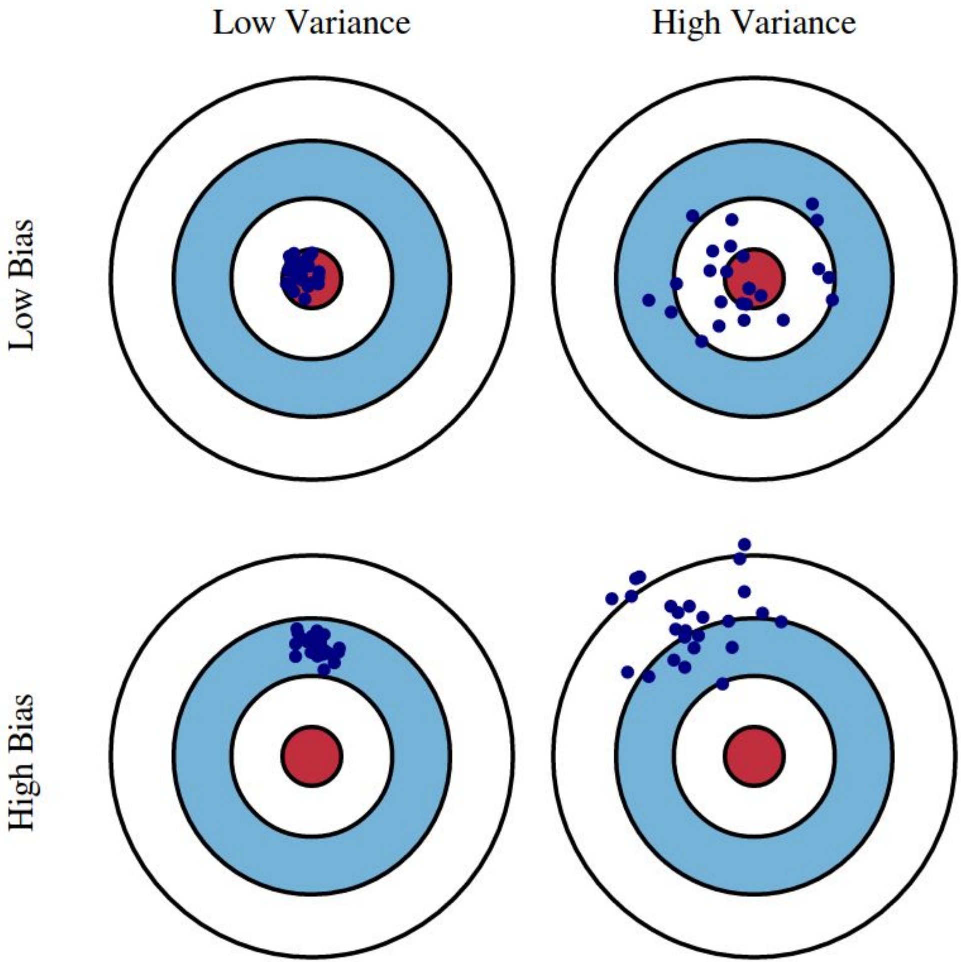
- A measure of underfitting

Variance

- A measure of overfitting

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

http://www.r2d3.us/visual-intro-to-machine-learning-part-2/



Source

What does this mean intuitively?

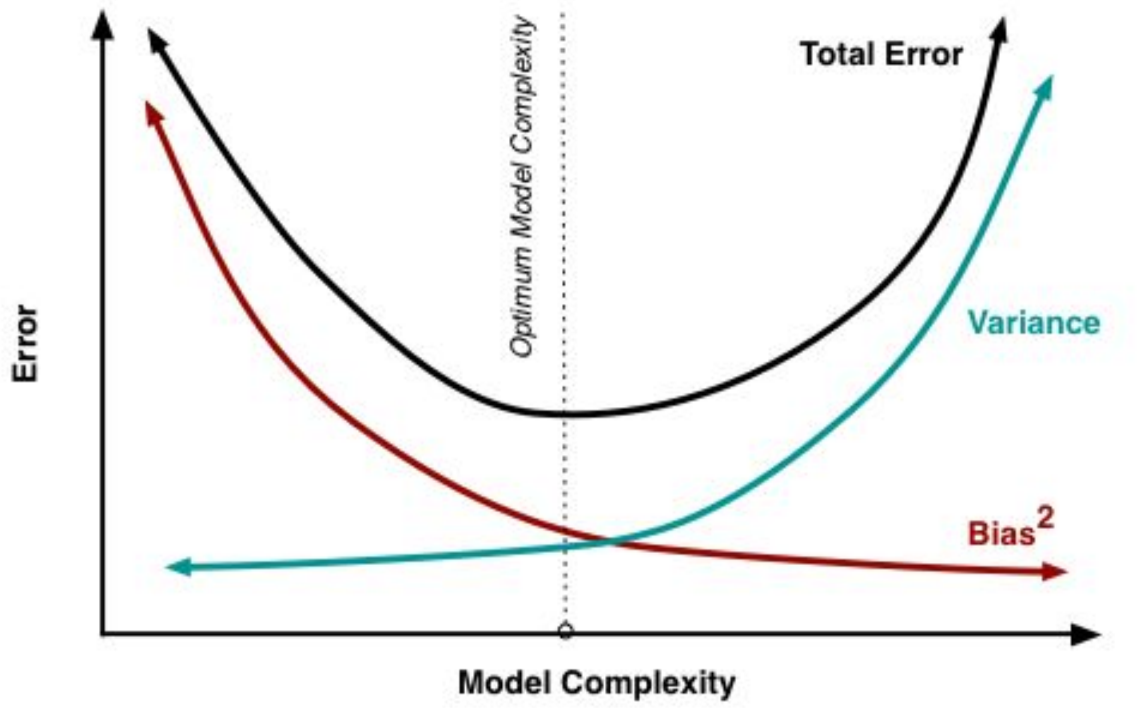
Bias

- Bad
- Results from incorrect assumptions in the learning algorithm

Variance

- Bad
- Results from sensitivity to fluctuations in the data

Balancing Bias and Variance



Source

Feature Selection

(adjusting models)

Methods

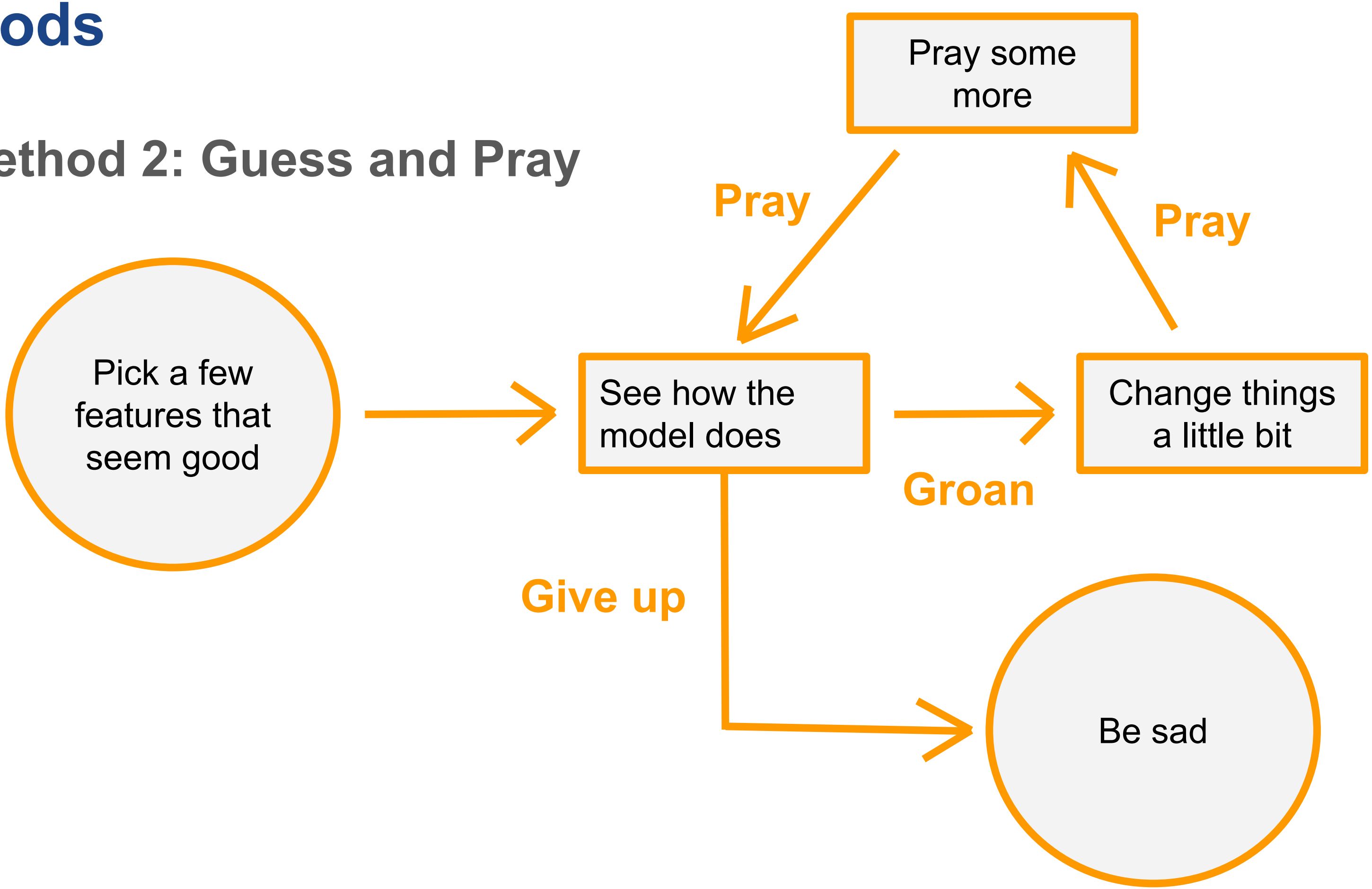
- **Goal:** Find subset of features that gives a good enough model, in a reasonable amount of time.

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

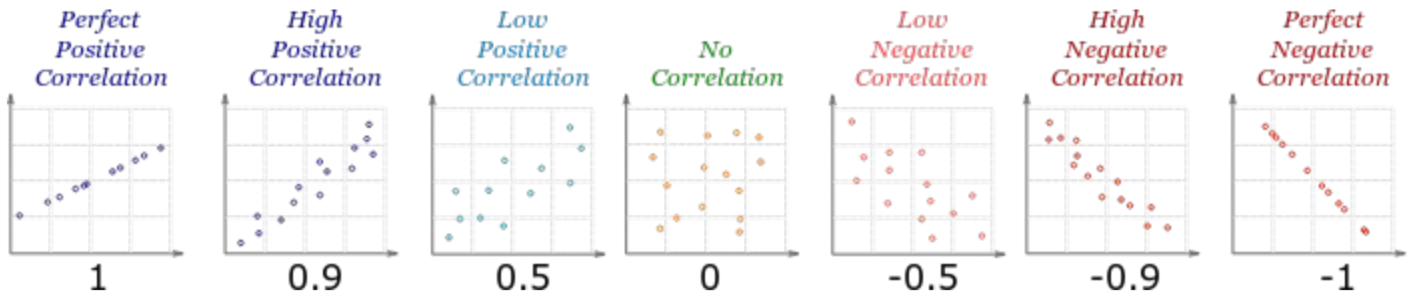
Methods

- **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
  - Can divide by max, minus min divide by max minus min, minus mean divide by standard deviation.

Other Ways to Adjust your Model

- HyperParameters

- Feature engineering

- Just changing to a different algorithm

Demo

Different Types of ML

(supervised & unsupervised)
(classification & regression)

Supervised vs. Unsupervised

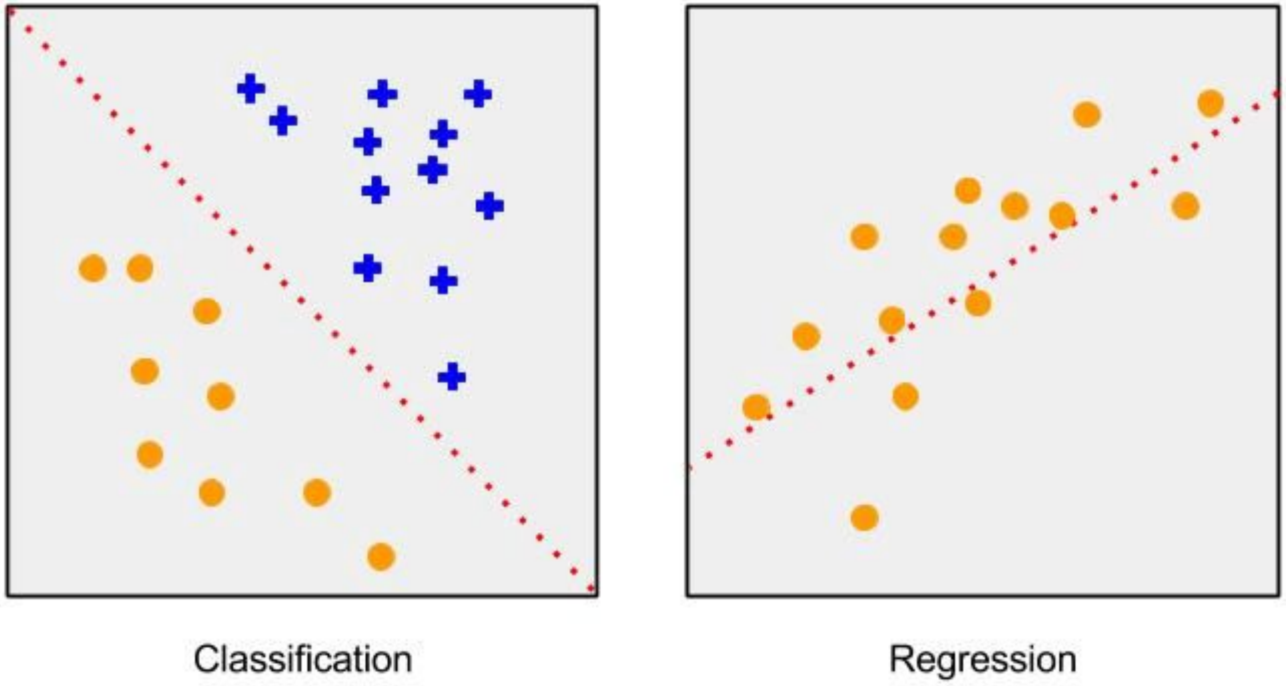
Supervised learning…

- Known target variable info
- Validation examples

Unsupervised learning…

- Unknown target variables
- Difficult to validate
- Discover underlying trends in the data

Classification vs. Regression



Final Notes

Always remember both bias and variance!

Coming Up

- **Assignment 4:** Due tonight at midnight!
- **Assignment 5:** Due at midnight next Wednesday, March 20th
- **Mid-Semester Check-In:** Now till Friday, March 22nd.
- **Next Lecture**: Intro to Classification