INFO 1998: Introduction to Machine Learning

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

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

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.

What We'll Cover

Last Time's Goal: be able to write code to do some kind of ML (to some extent)

This Time's Goal: create *useful* ML models

Agenda

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

Review
(placeholder text for aesthetic)

Review: Defining ML

<u>We want to predict the future</u>

- 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

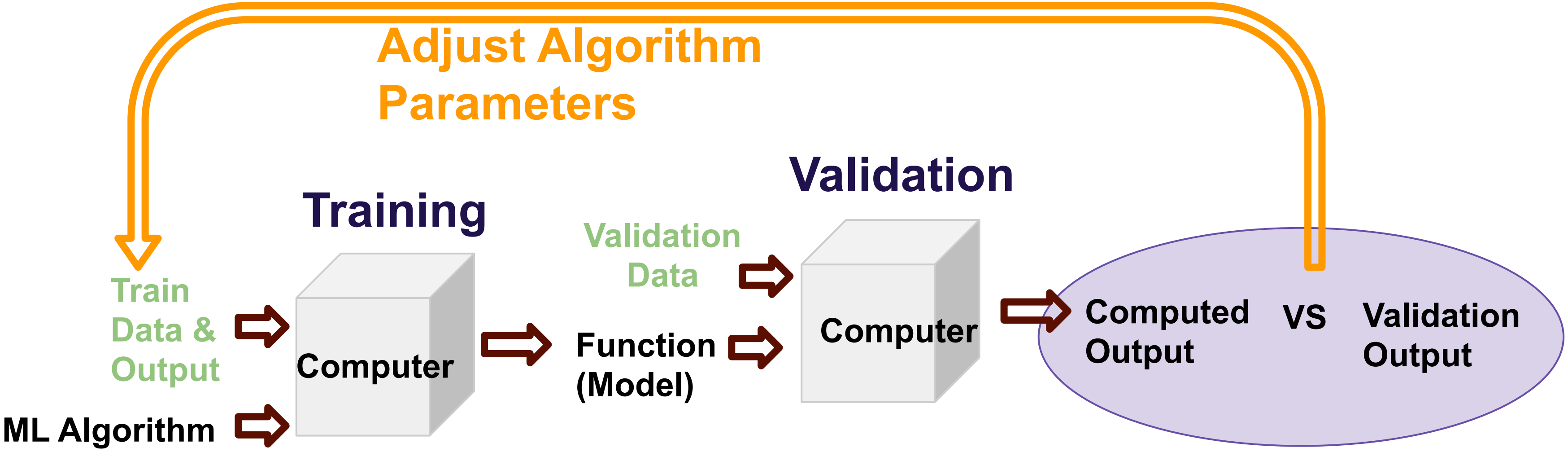
Now: *how* do we learn the data's pattern?

Review: Model

- Something you use to predict outputs
- The Linear Regression Algorithm produces Linear Regression Models
- “Model training” = learn a relationship/program
- “Model validation” = see if the learned relationship is accurate on other data

Measuring Bias / Loss

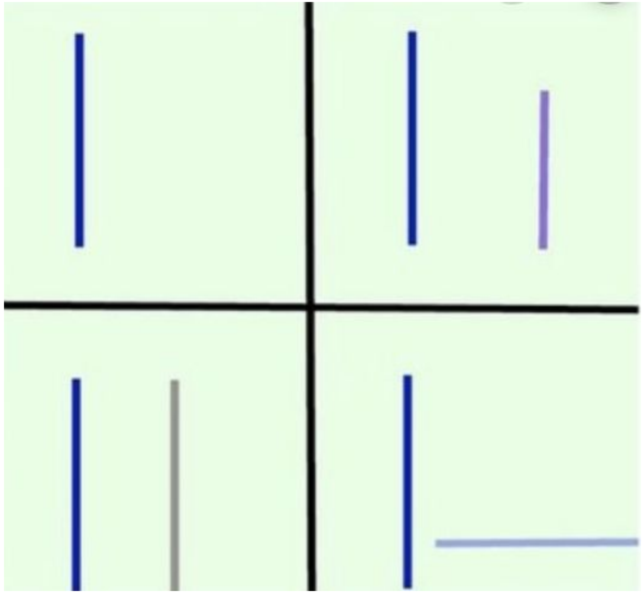
(training accuracy)



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 missing 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



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

Demo

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

Overfitting and Underfitting

(what’s a good model?)

Model Goals

When training a model, we want our model to:

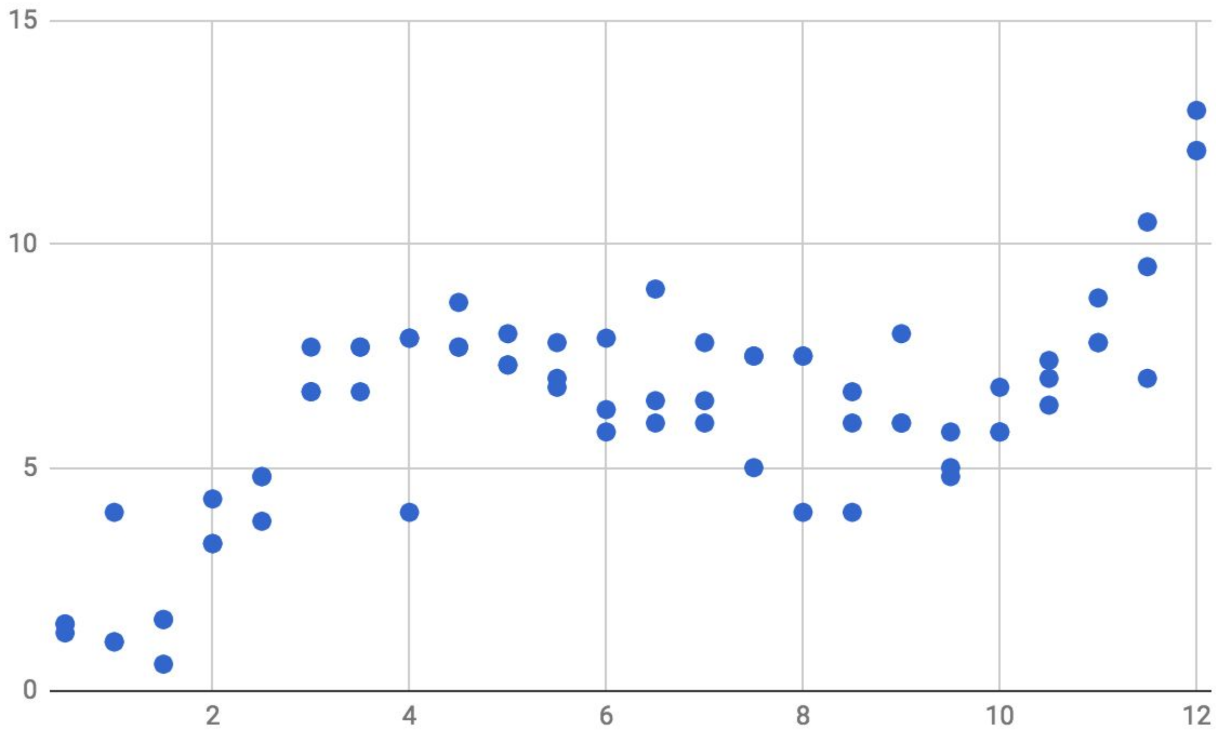
- Capture the trends of the training data
- Generalize well to other samples of the population
- Be moderately interpretable

The first two are especially difficult to do simultaneously!

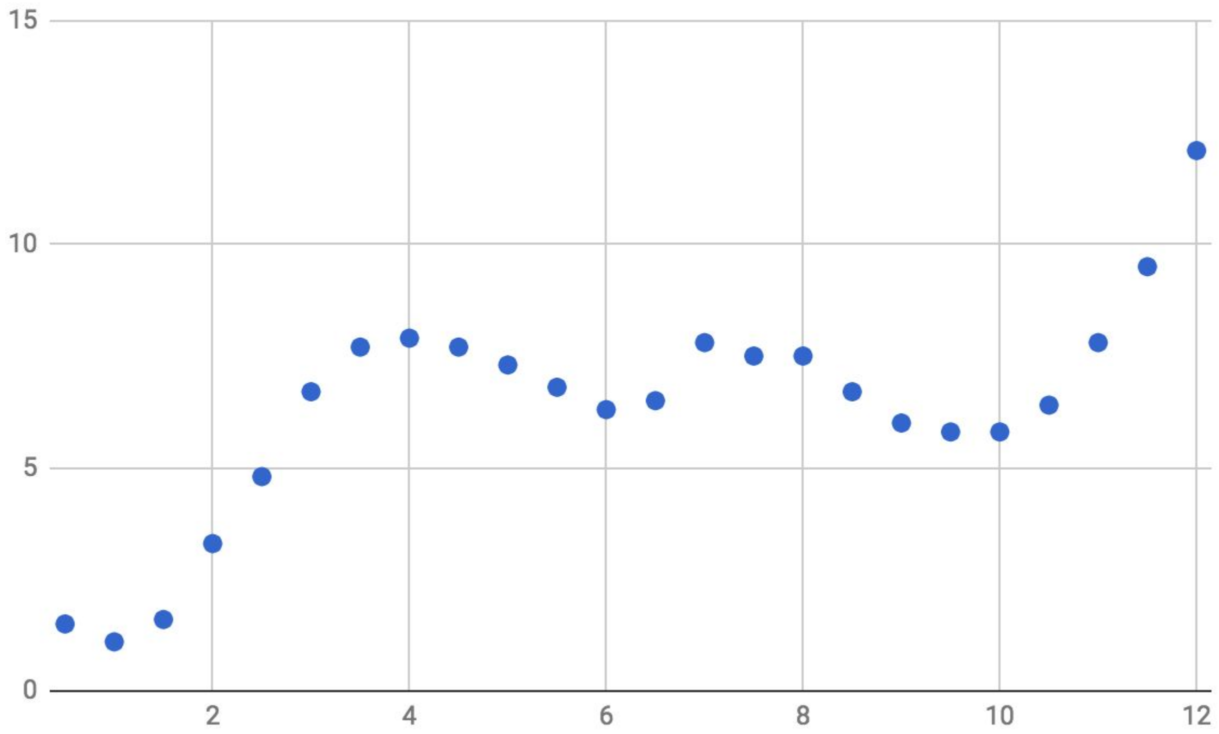
The more sensitive the model, the less generalizable and vice versa.

Demo

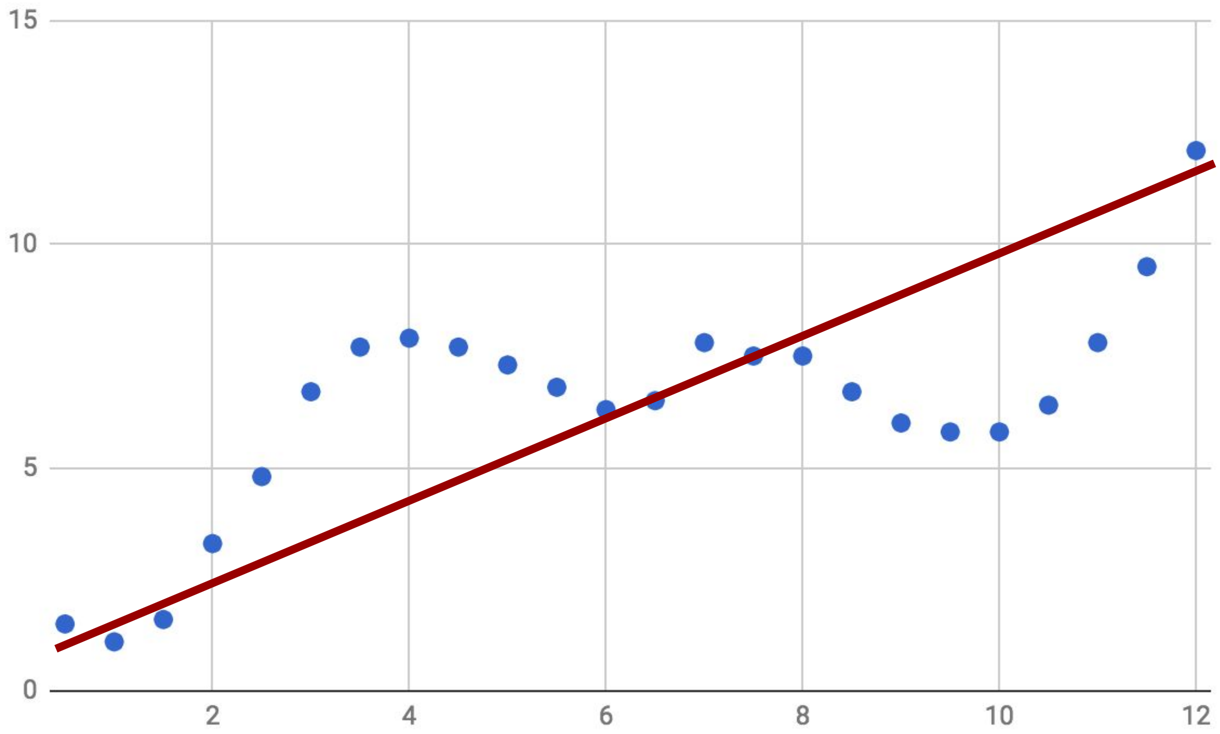
Underfitting



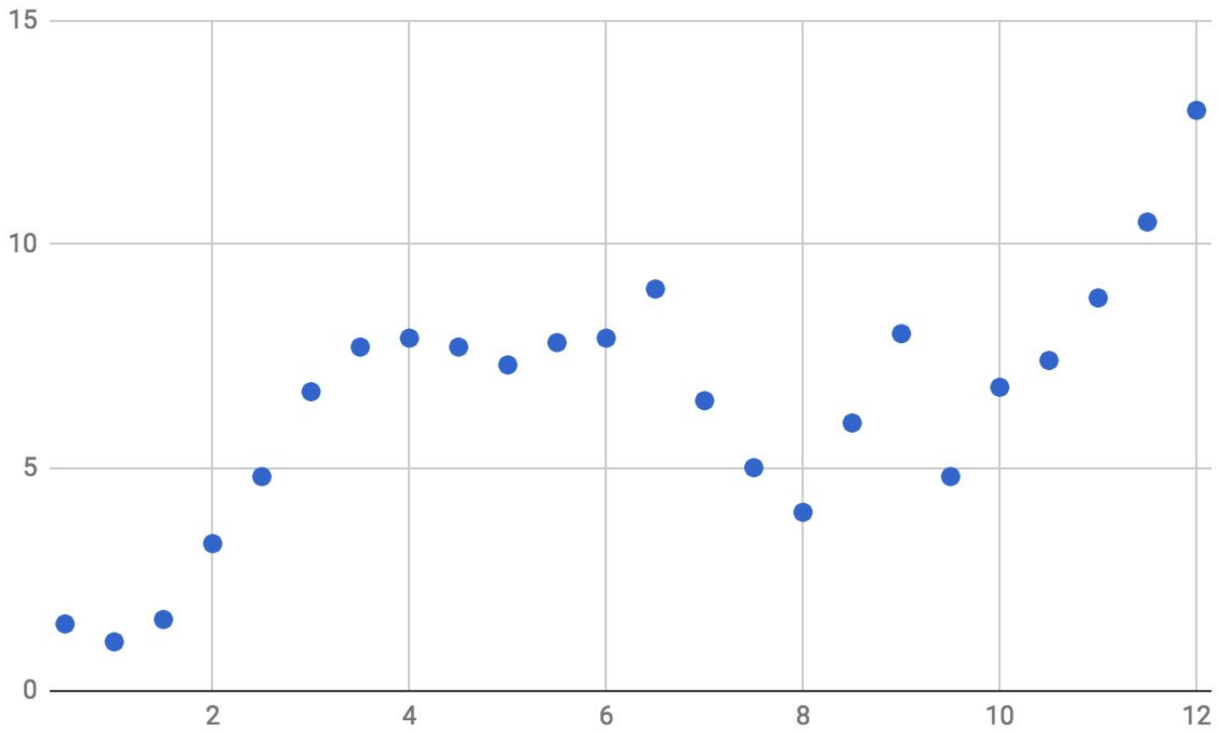
Underfitting



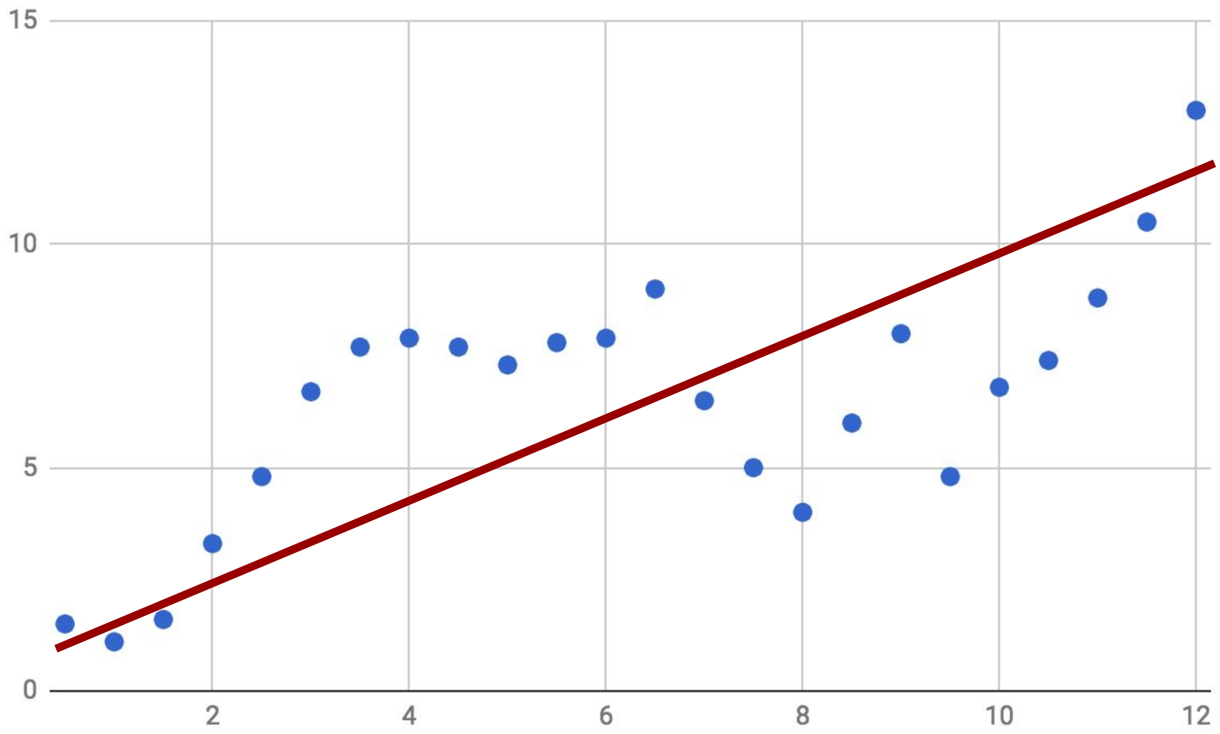
Underfitting



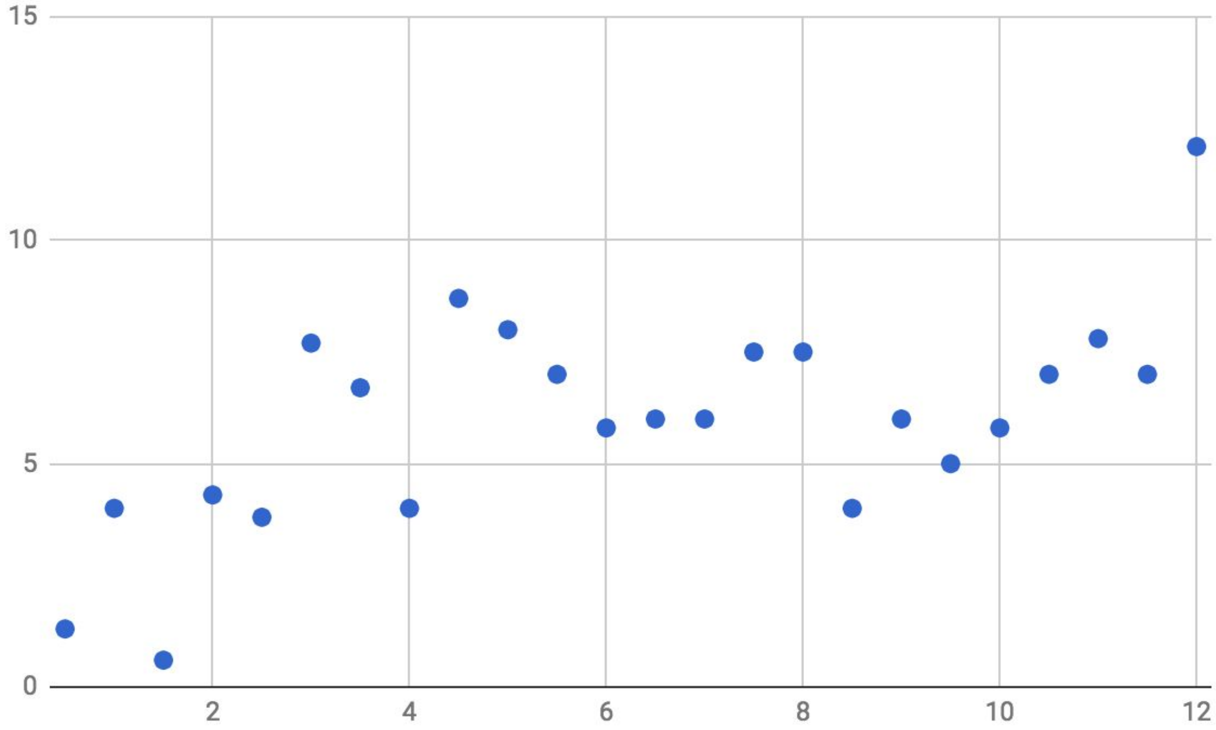
Underfitting



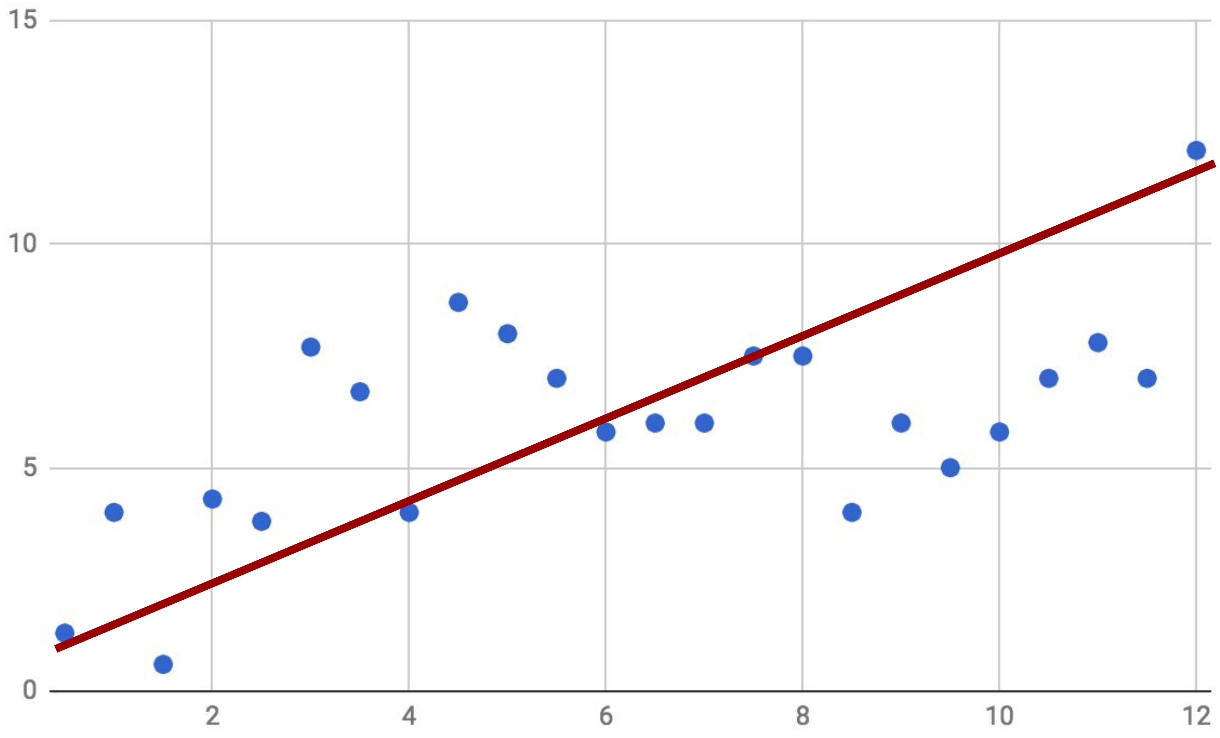
Underfitting



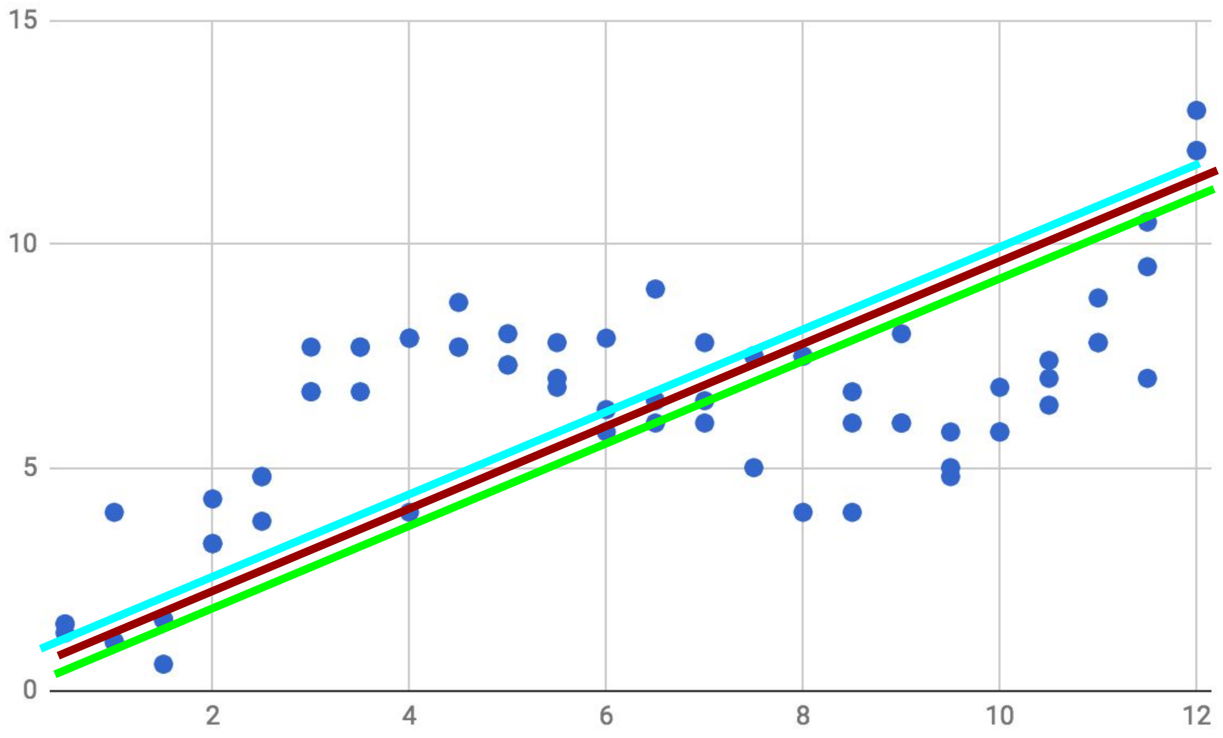
Underfitting



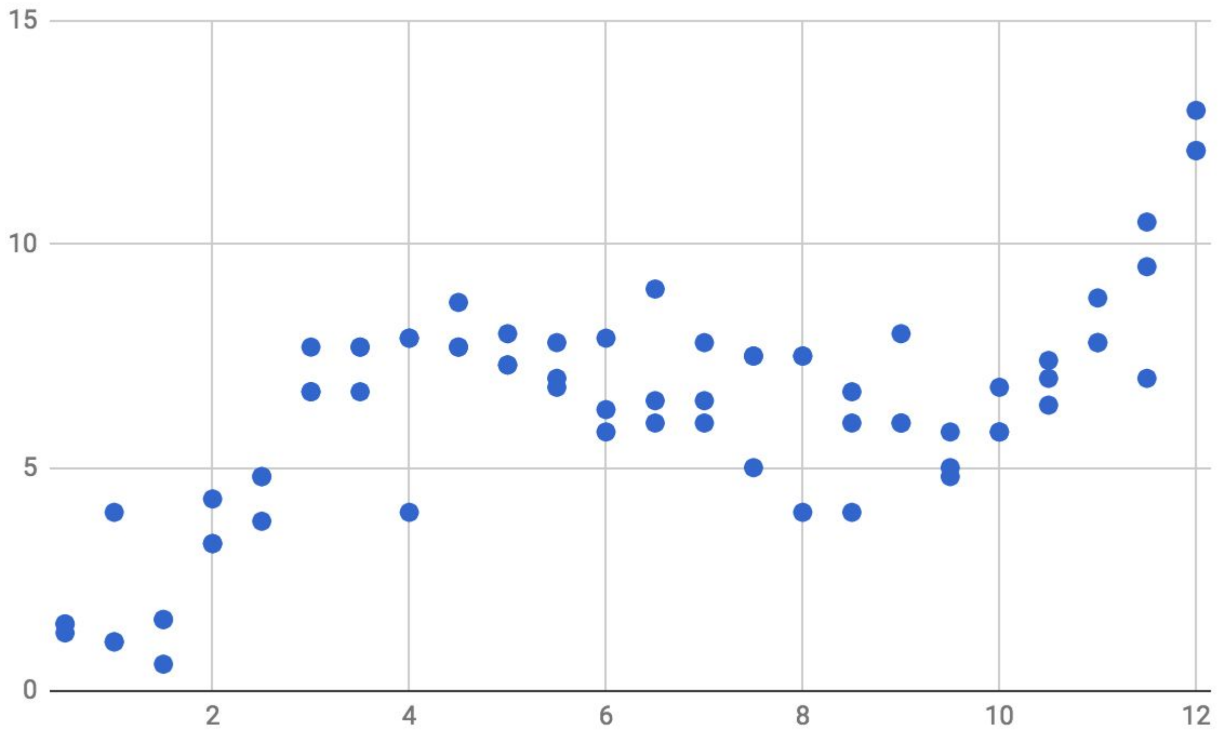
Underfitting



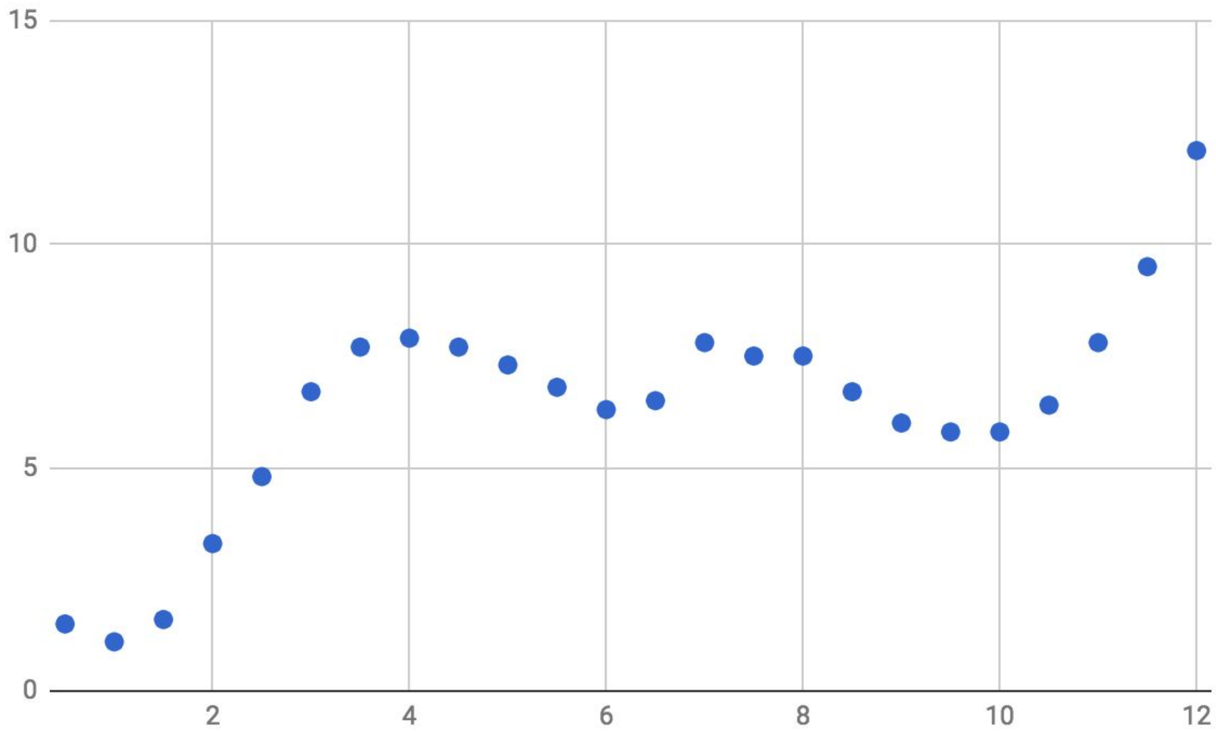
Underfitting: at least the models are consistent...



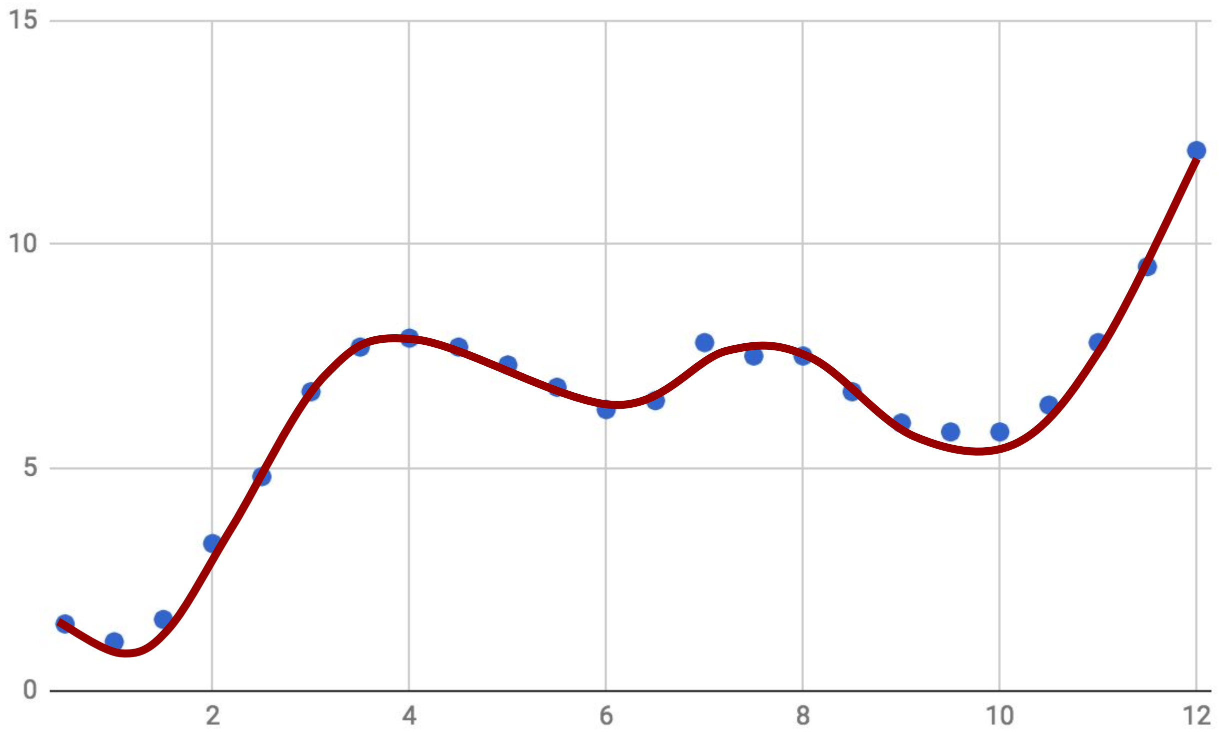
Overfitting



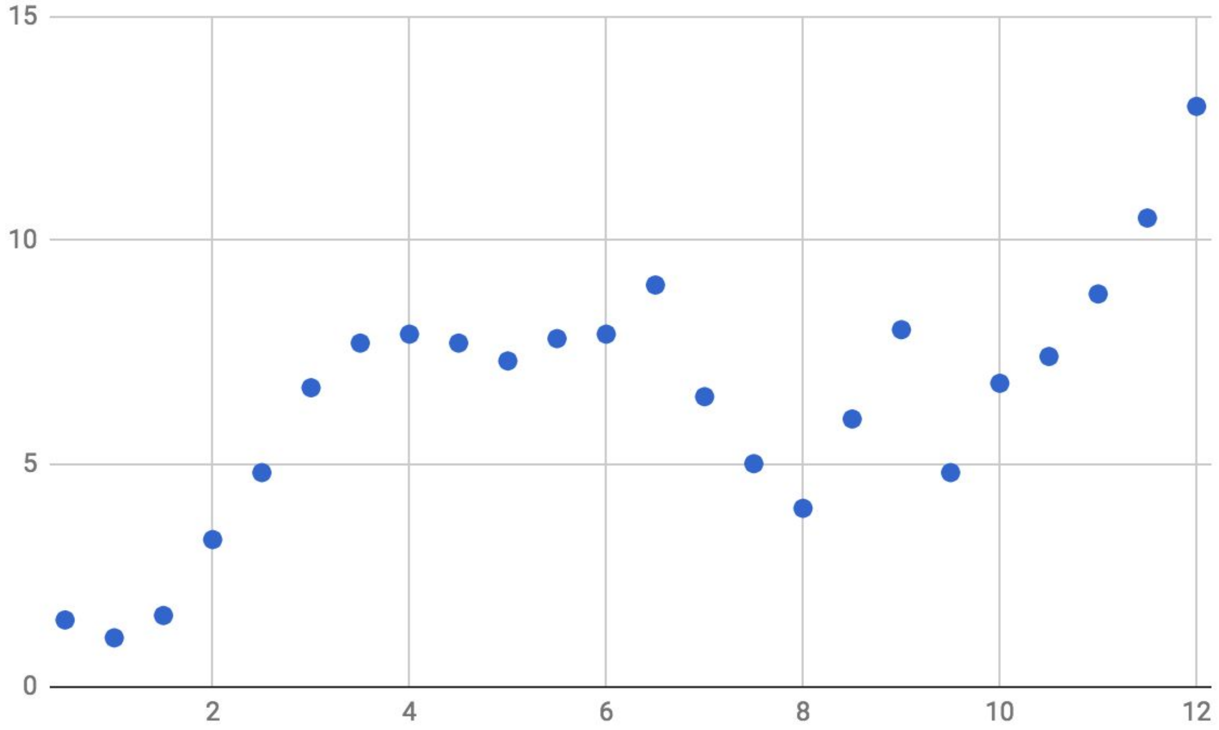
Overfitting



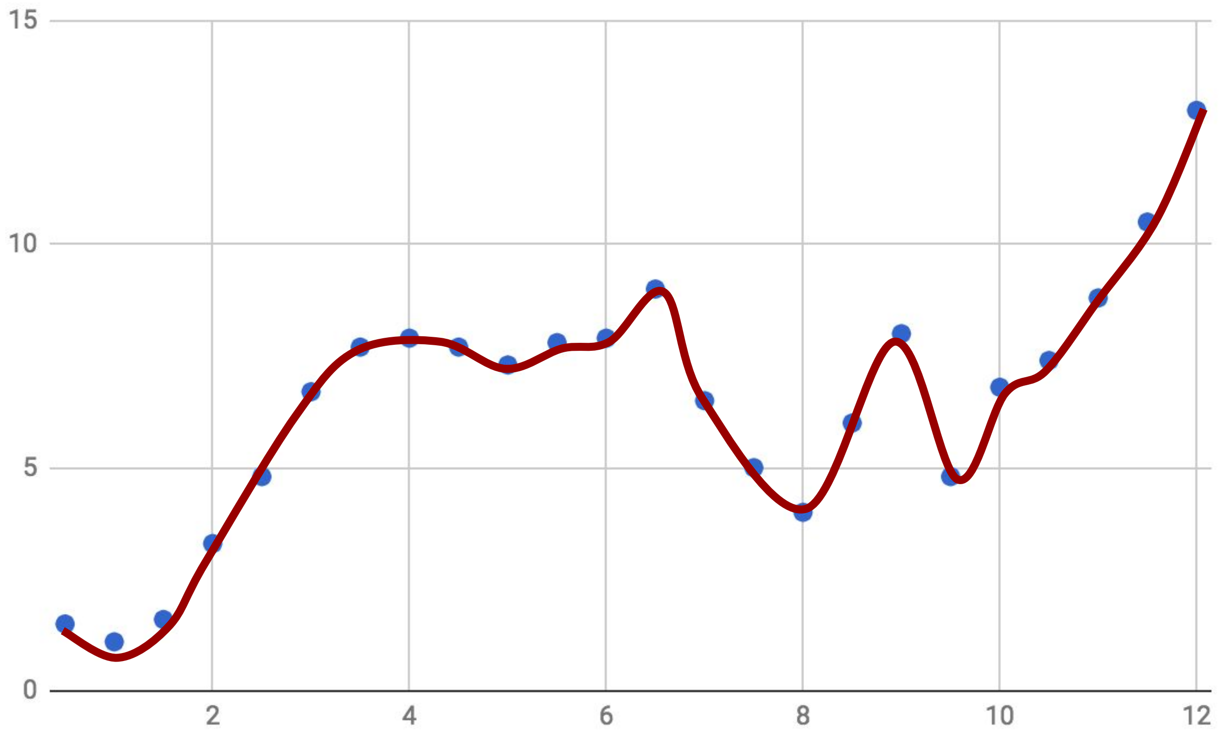
Overfitting



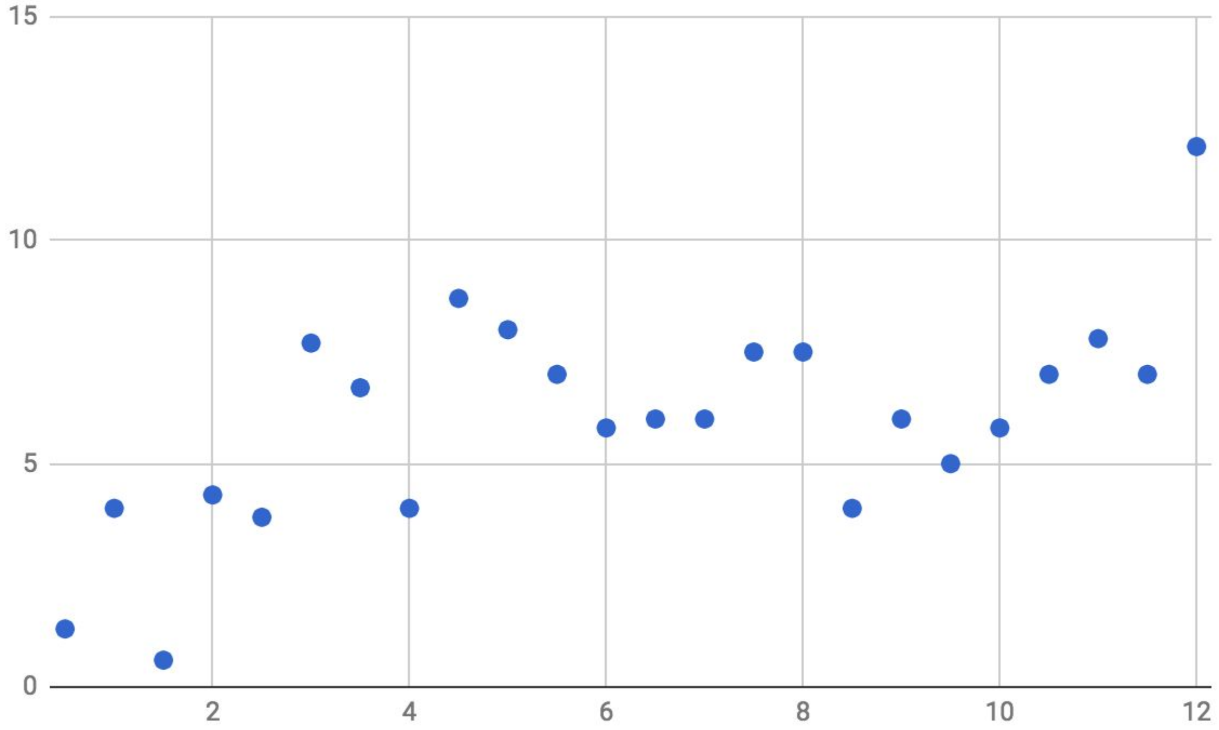
Overfitting



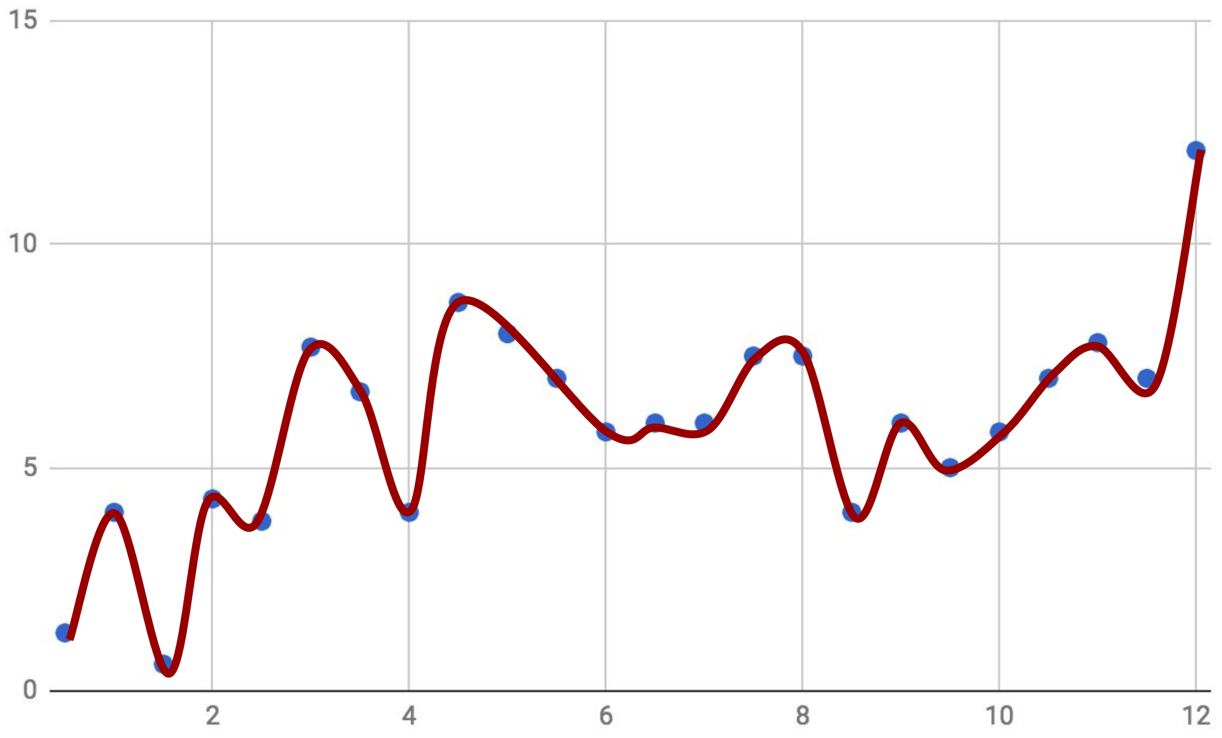
Overfitting



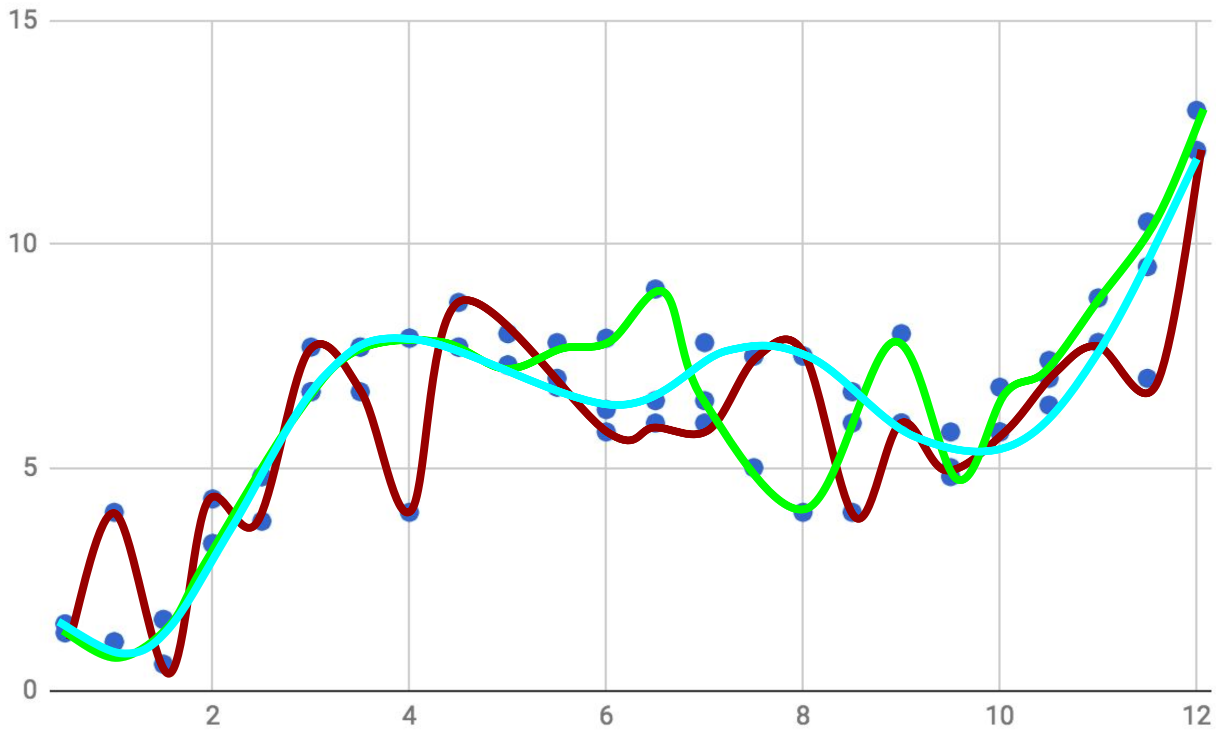
Overfitting



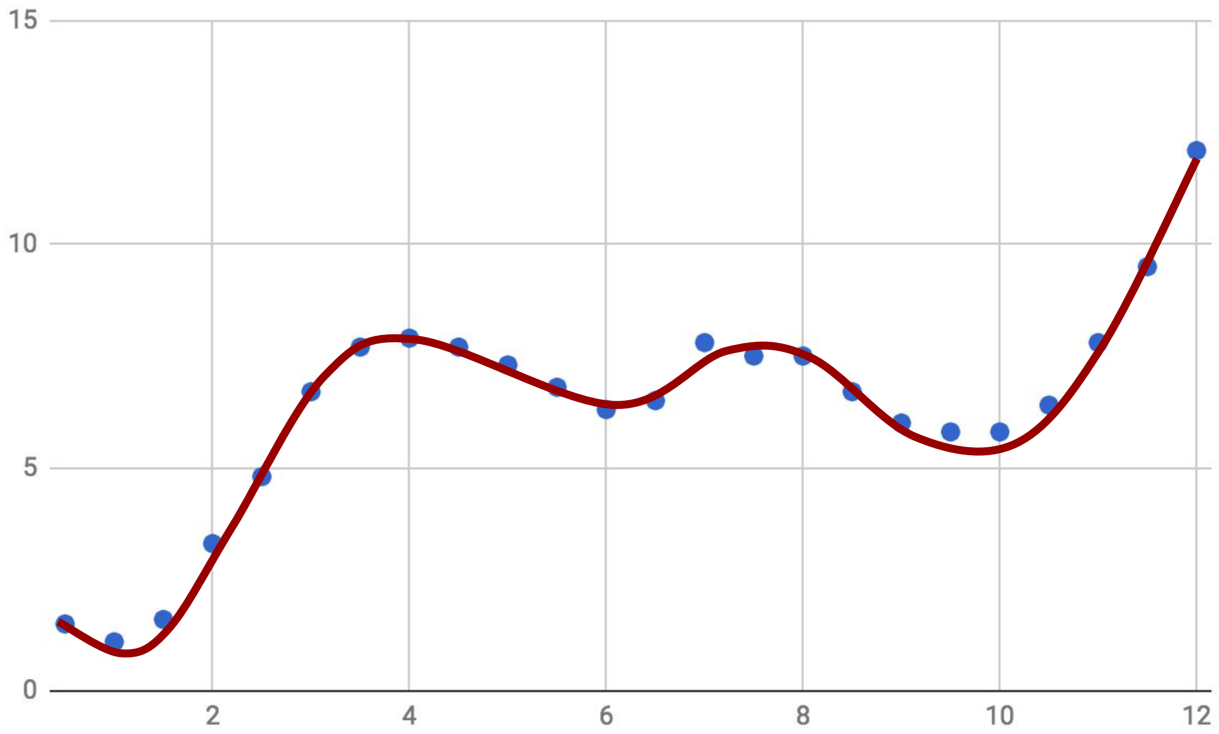
Overfitting



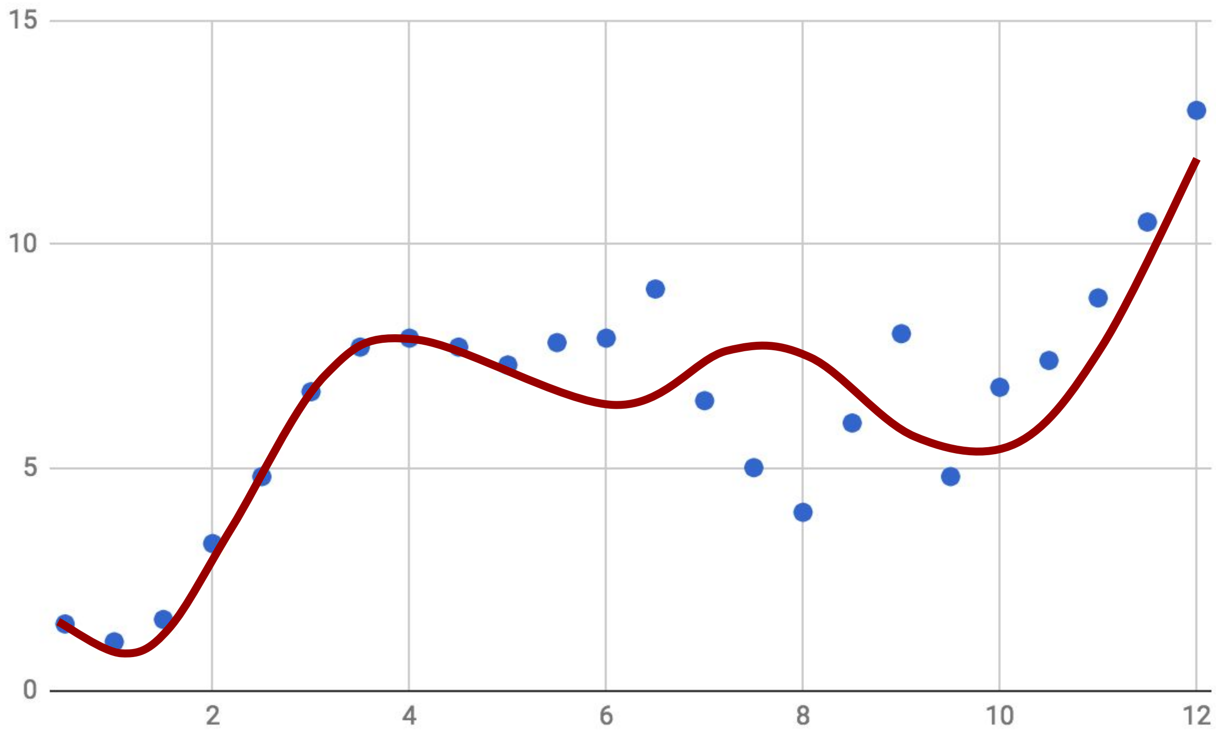
Overfitting: Inconsistent Models!



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

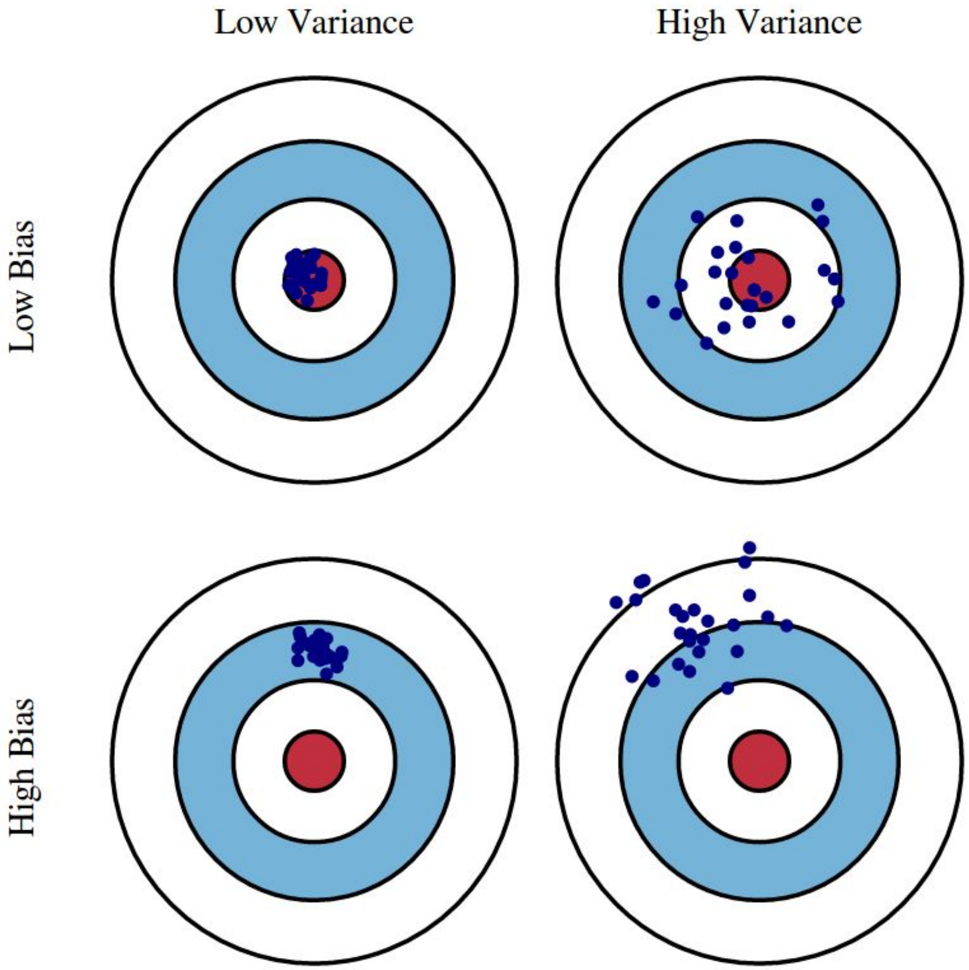
Balancing Bias and Variance

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

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

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

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


Low Variance
High Variance
Low Bias
High Bias

What does this mean intuitively?

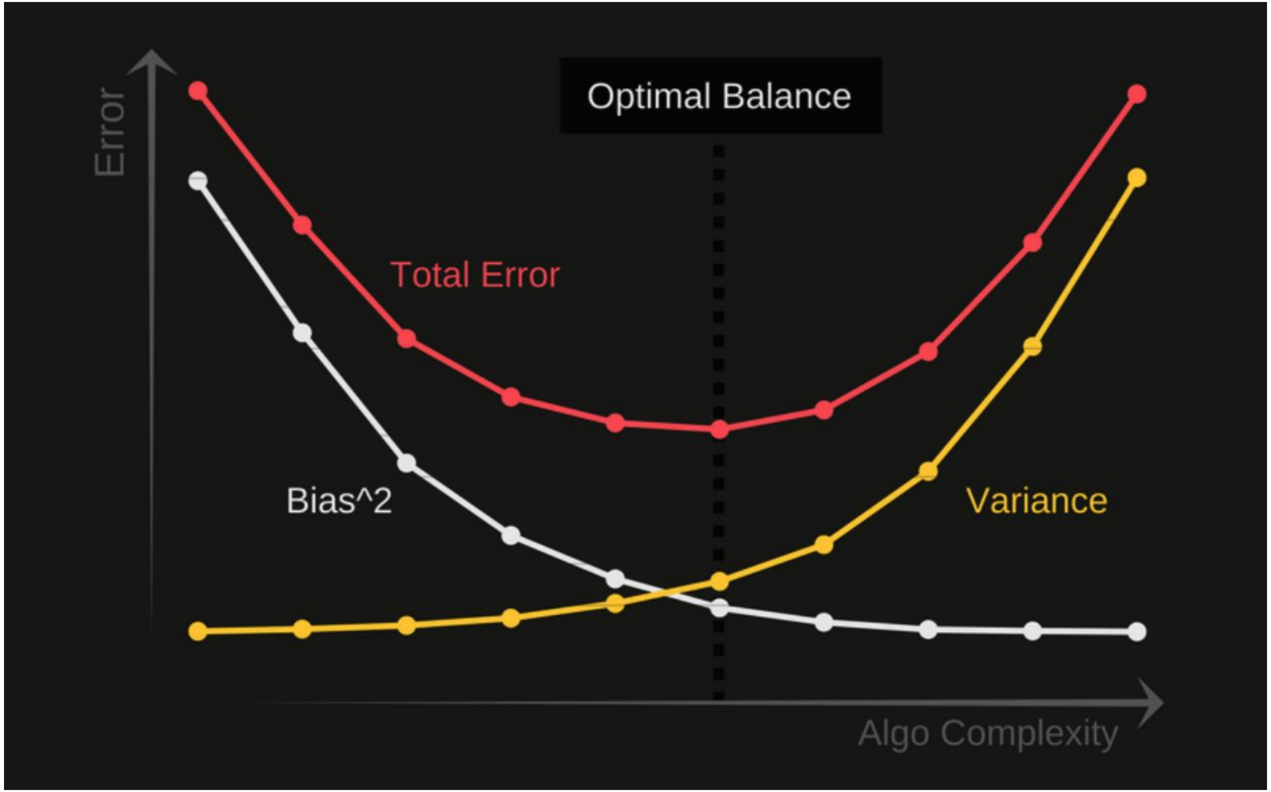
Bias

- Bad
- Results from incorrect assumptions in the learning algorithm

Variance

- Bad
- Results from sensitivity to fluctuations in the data

Balancing Bias and Variance



Feature Selection

(adjusting models)

Methods

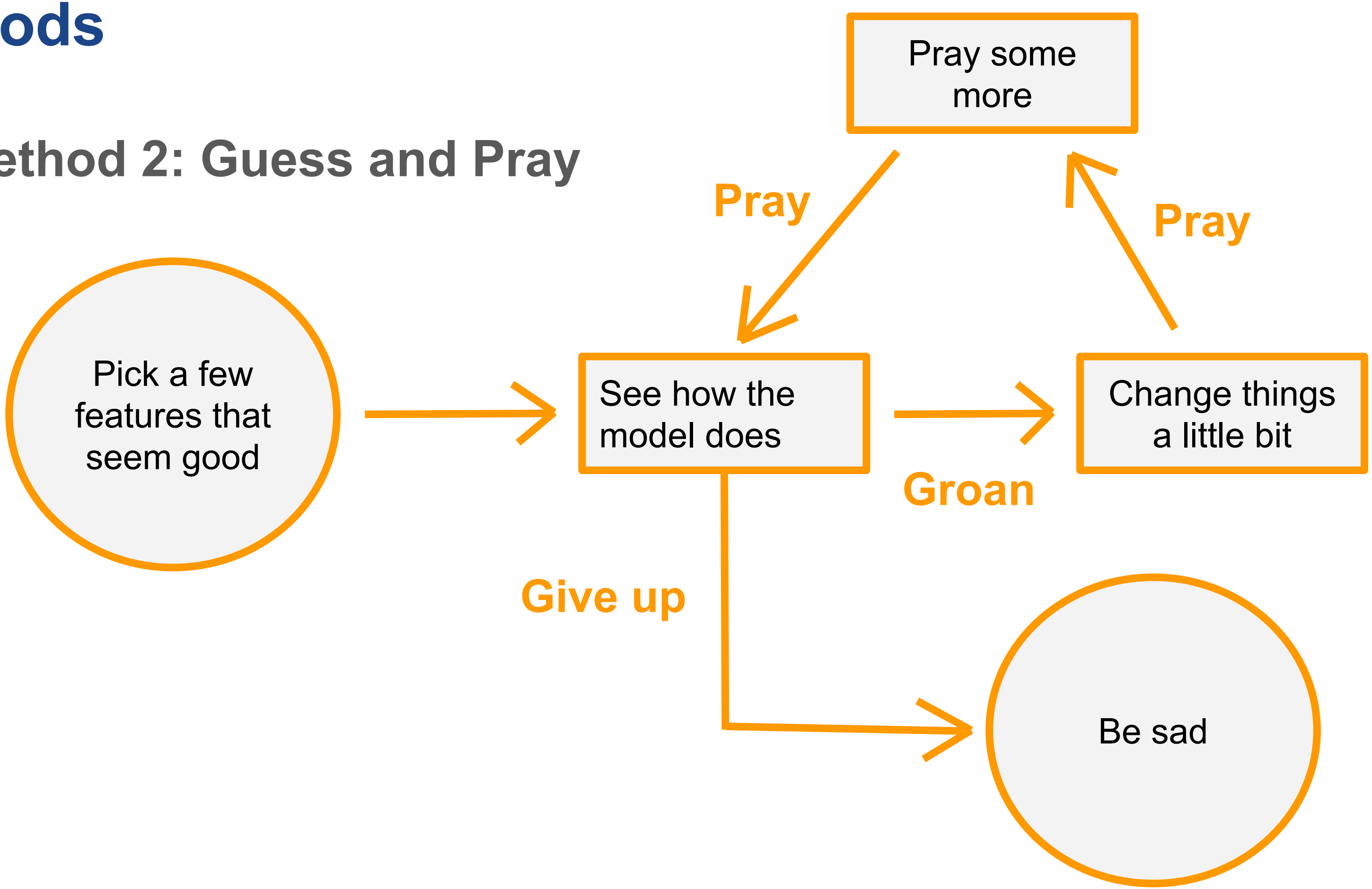
- **Goal:** Find subset of features that gives a good enough model, in a reasonable amount of time.

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

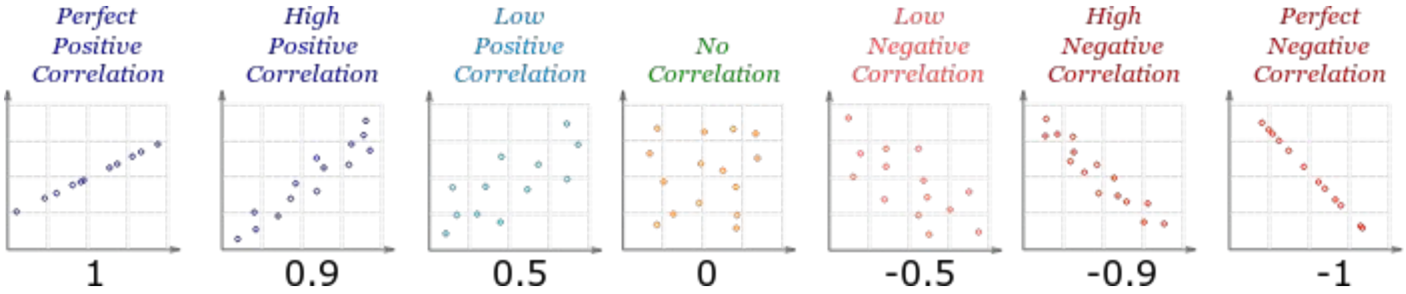
Methods

- **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* that no features used are collinear

Side Note: Scaling and Normalizing

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

Other Ways to Adjust your Model

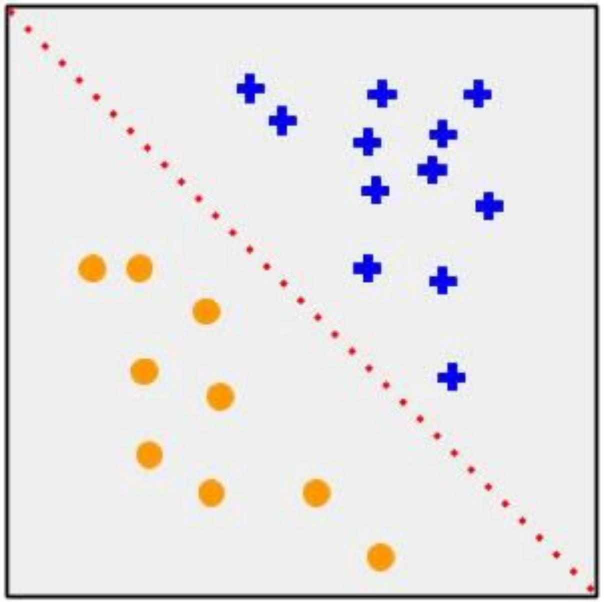
- HyperParameters

- Feature engineering

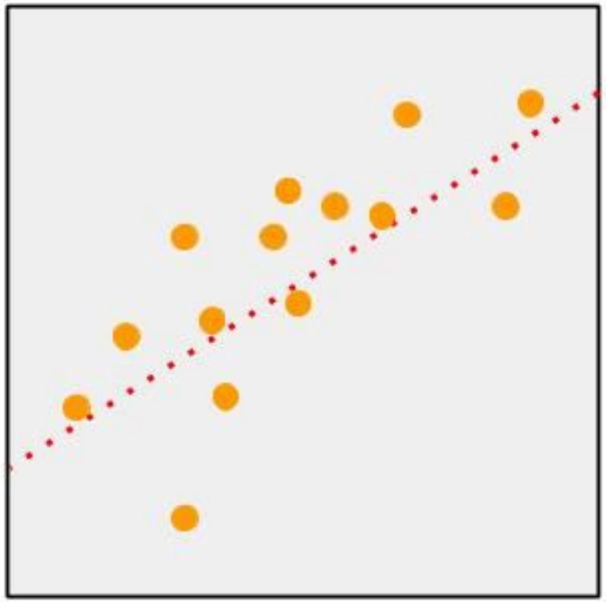
- Just changing to a different algorithm

Different Types of ML
(classification & unsupervised)

Classification vs. Regression



Classification



Regression

Supervised vs. Unsupervised

Supervised learning…

- Known target variable info
- Validation examples

Unsupervised learning…

- Unknown target variables
- Difficult to validate

Other Classes of ML Algorithms (which we won't cover)

- What if you can't / don't want to see all your data at once?

- Maybe you only want to use a few pieces of your data (but don't have the time to manually select each piece of data…)

- A different approach, Trial & Error: The algorithm tries one thing, sees how that works, makes adjustments, tries again, etc.

Final Notes

Always remember both bias and variance!

Coming Up

- **Assignment 5:** Due at 5:30pm on March 18, 2020
- **Midsemester Project**
- **Next Lecture**: Intro to Classification

CDS Education

We explore, learn, and educate big minds.