CDS Education

Introduction to Machine Learning for Python

Model Optimization

Announcements


+1!
LEVEL
UP

“Leveling Up” as a Data Scientist

Table of Contents

Model Goals

When training a model we want our models 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.

Hyperparameter Tuning

- Parameters vs Hyperparameters
- Examples:
  - Number of buckets on a decision tree
  - K in KNN
- How to pick the right values
- How do we even measure “doing well”?

Bias and Variance

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

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

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

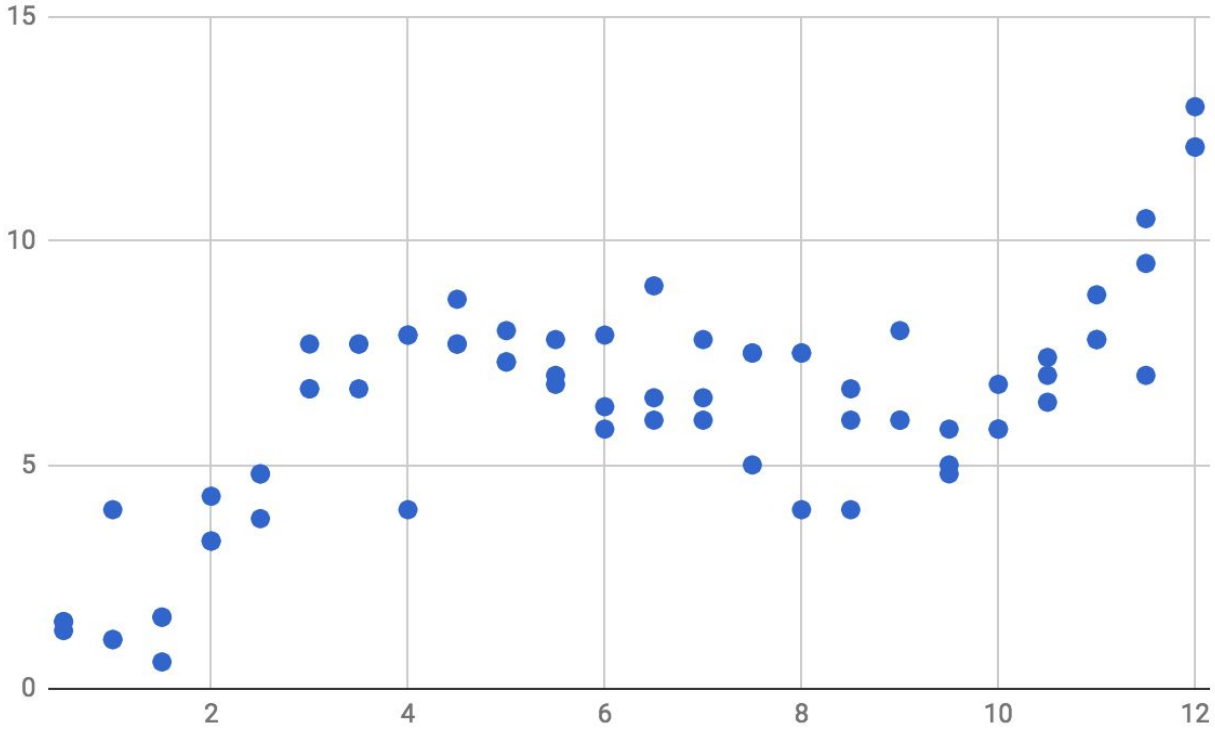
Error = (expected loss of accuracy)^2 + flexibility of model + irreducible error

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

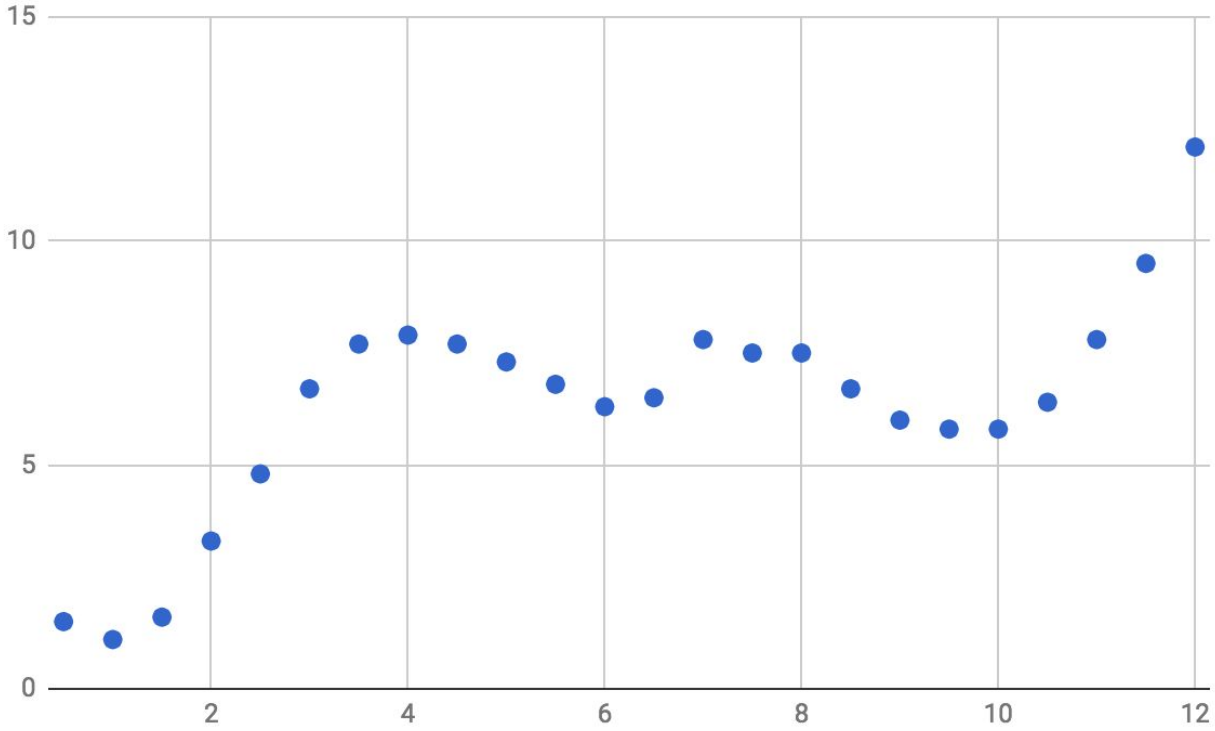
What does this mean intuitively?

- **Bias** results from incorrect assumptions in the learning algorithm
- **Variance** results from sensitivity to fluctuations in the data
- There is a **trade-off** between bias and variance
- Different machine learning algorithms are prone to different kinds of error

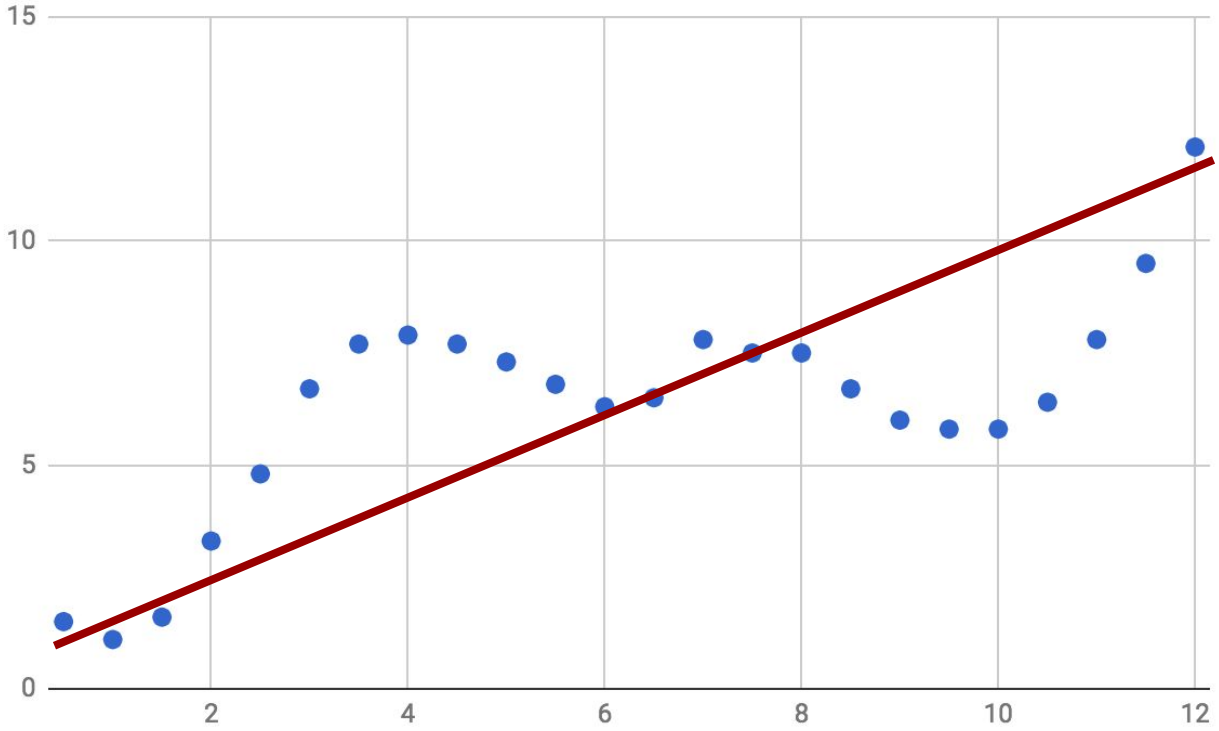
Bias



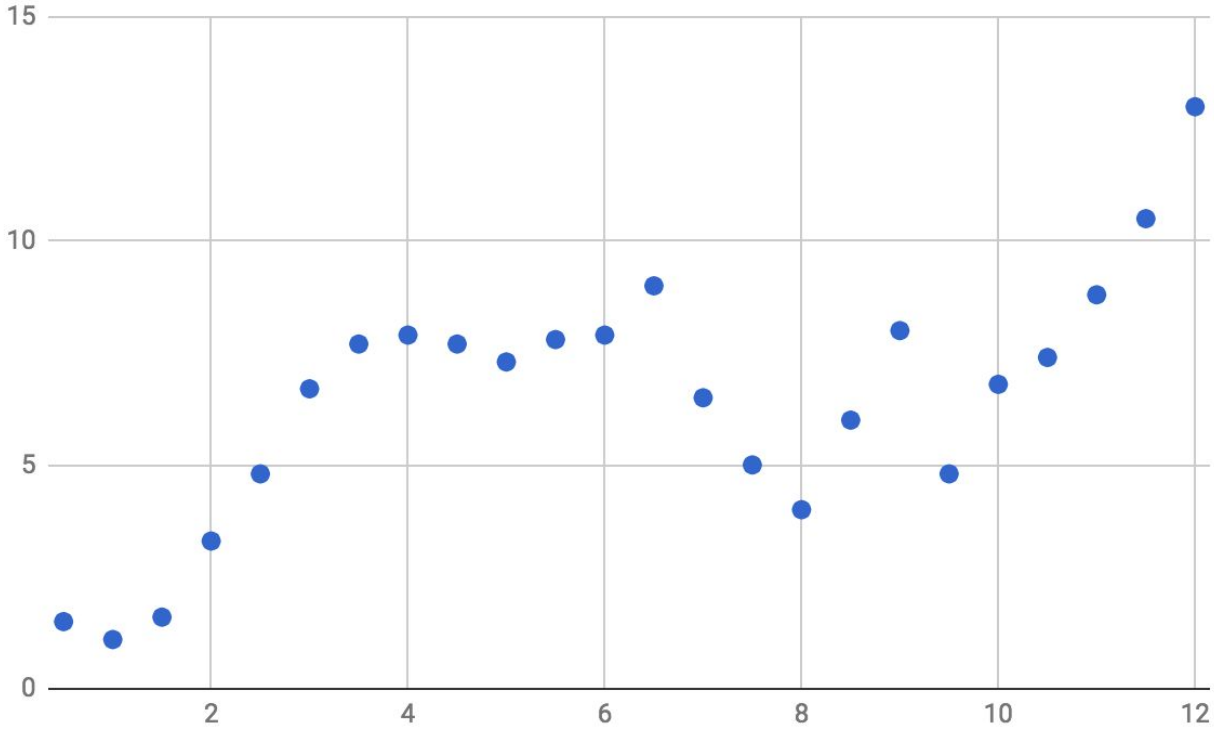
Bias



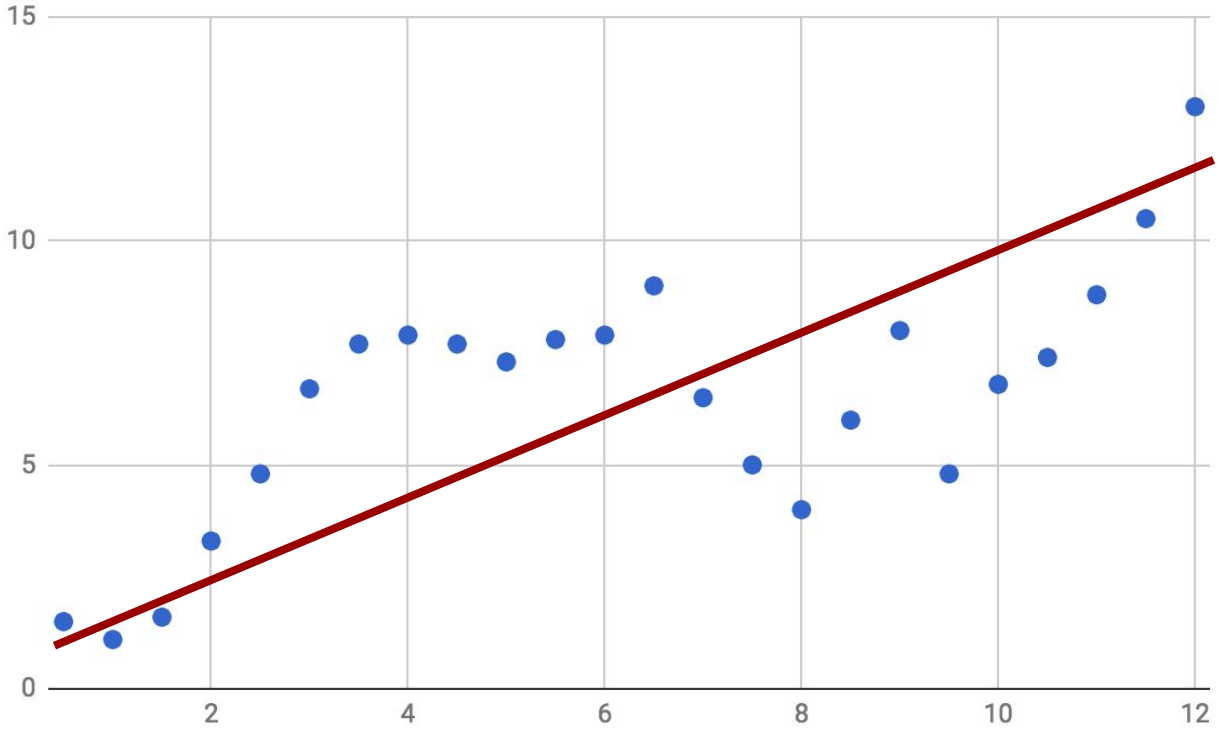
Bias



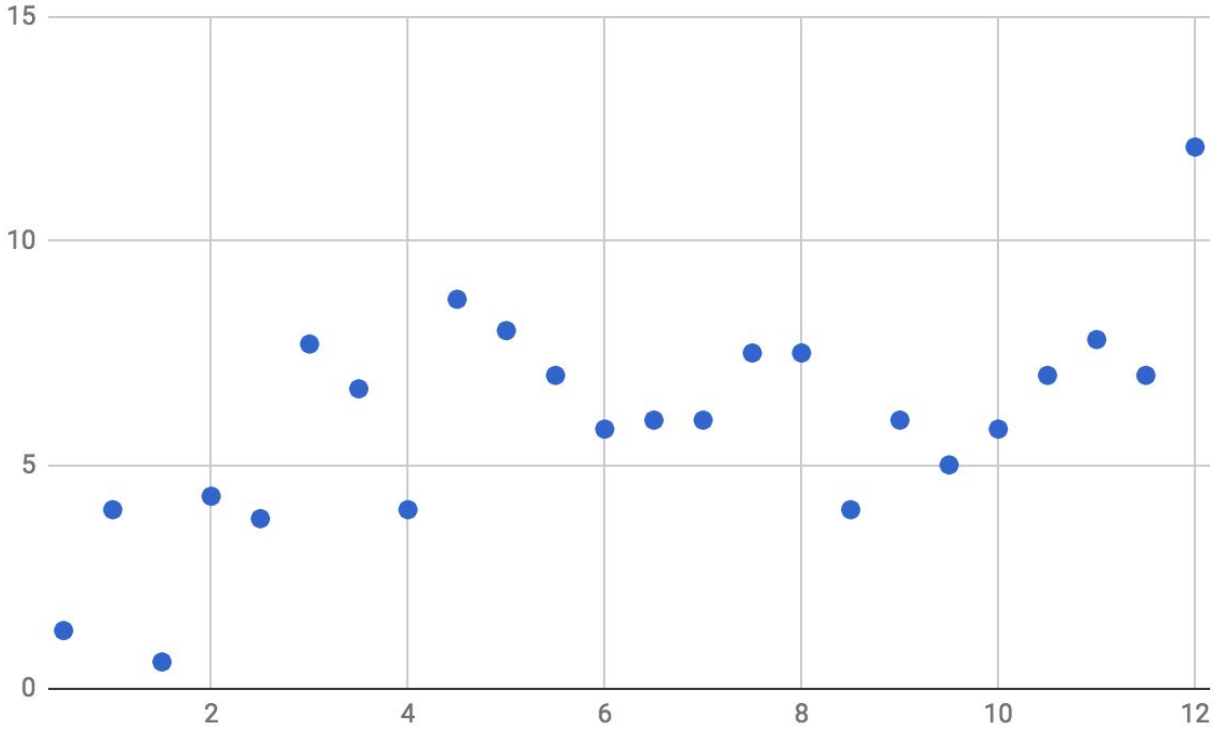
Bias



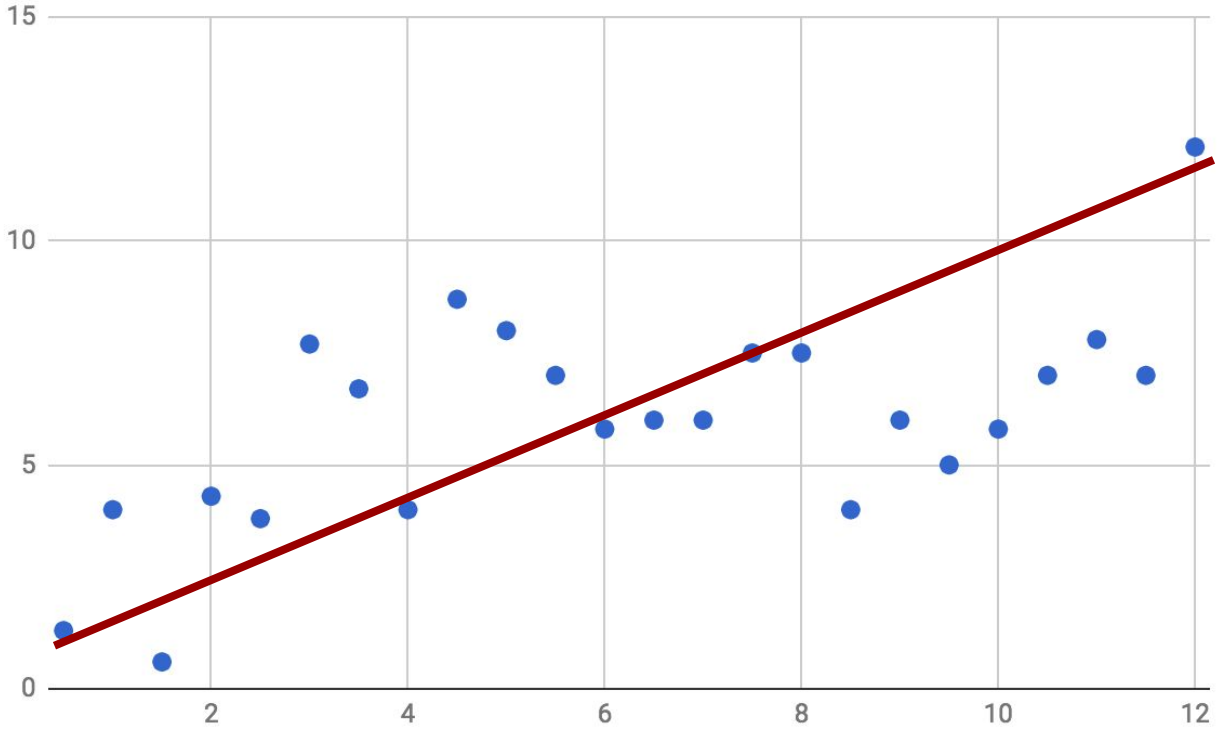
Bias



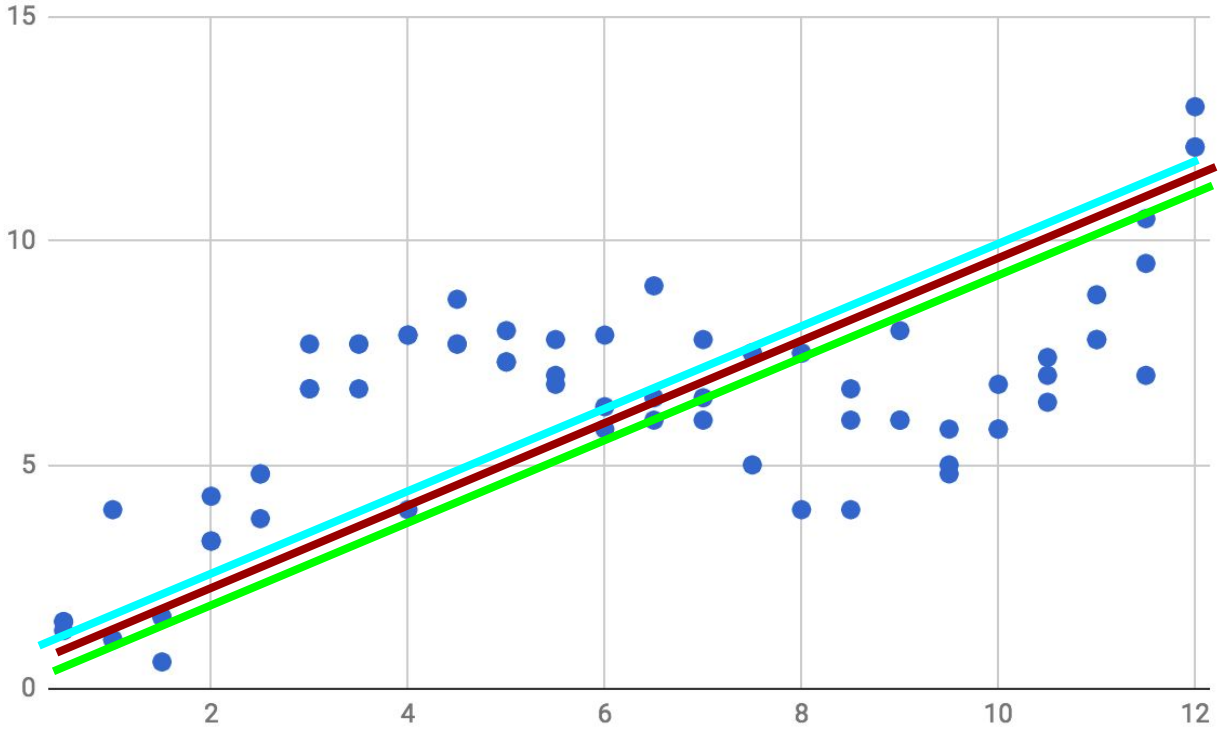
Bias



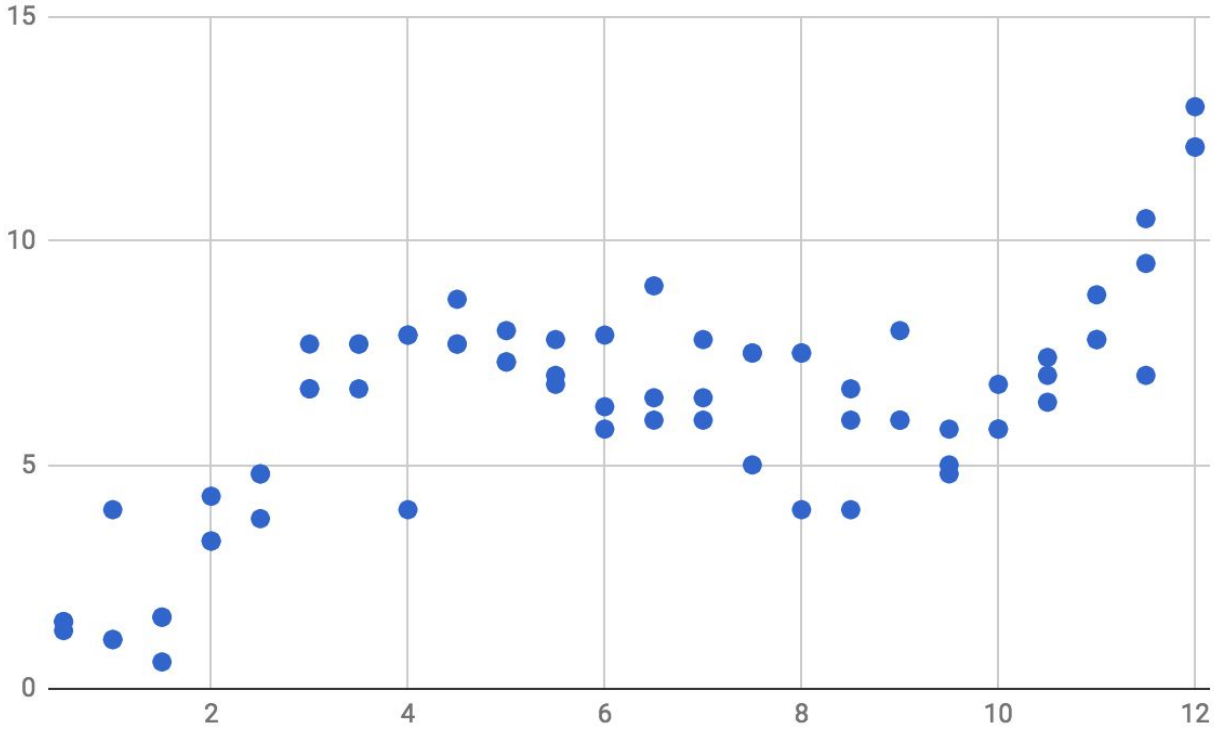
Bias



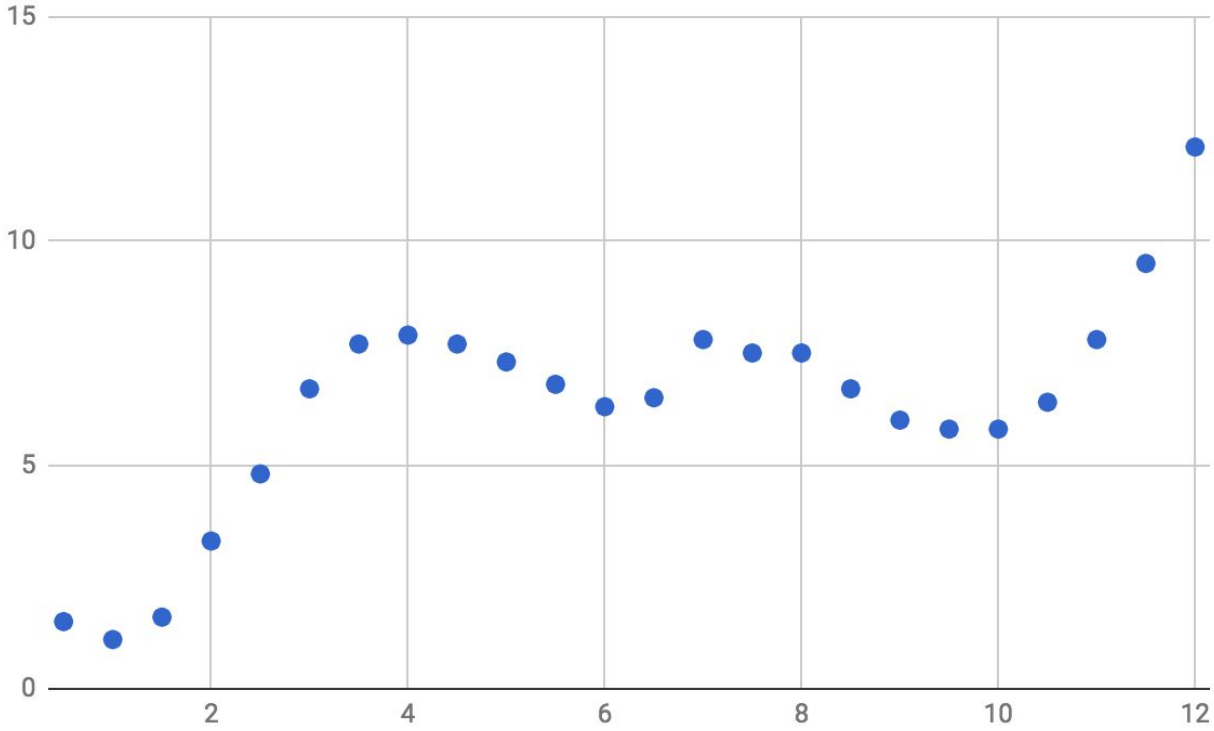
Bias



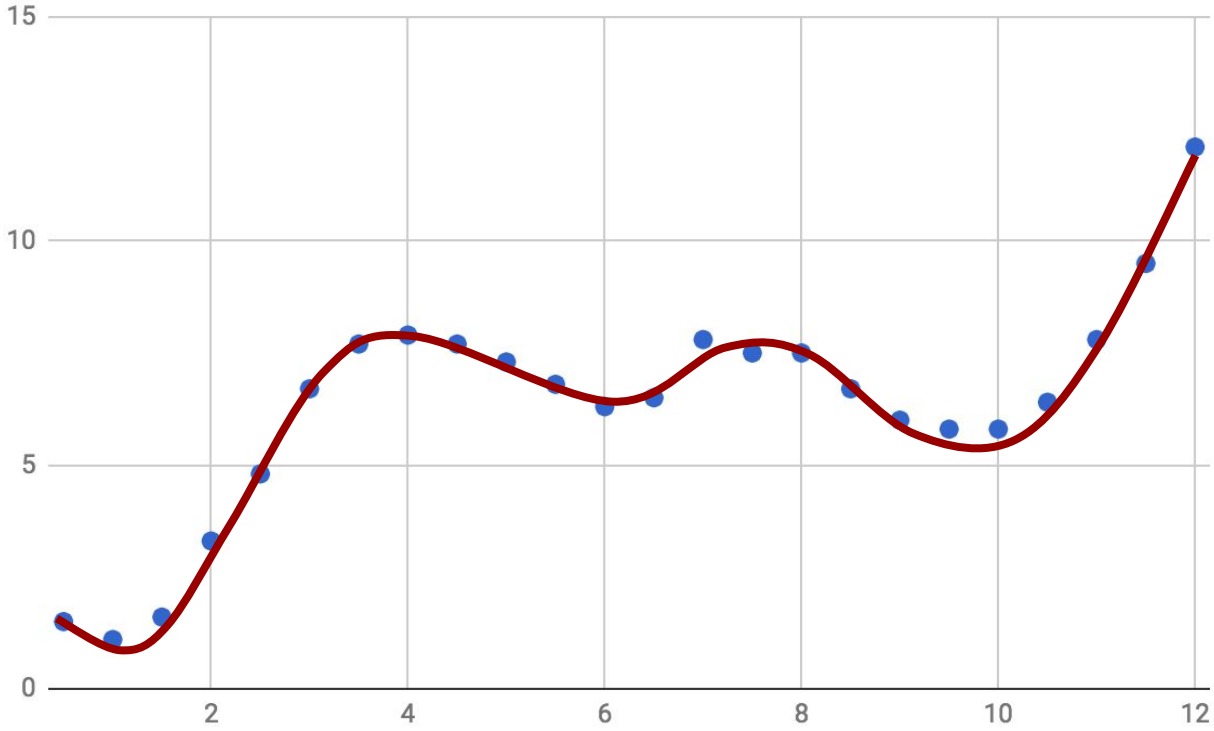
Variance



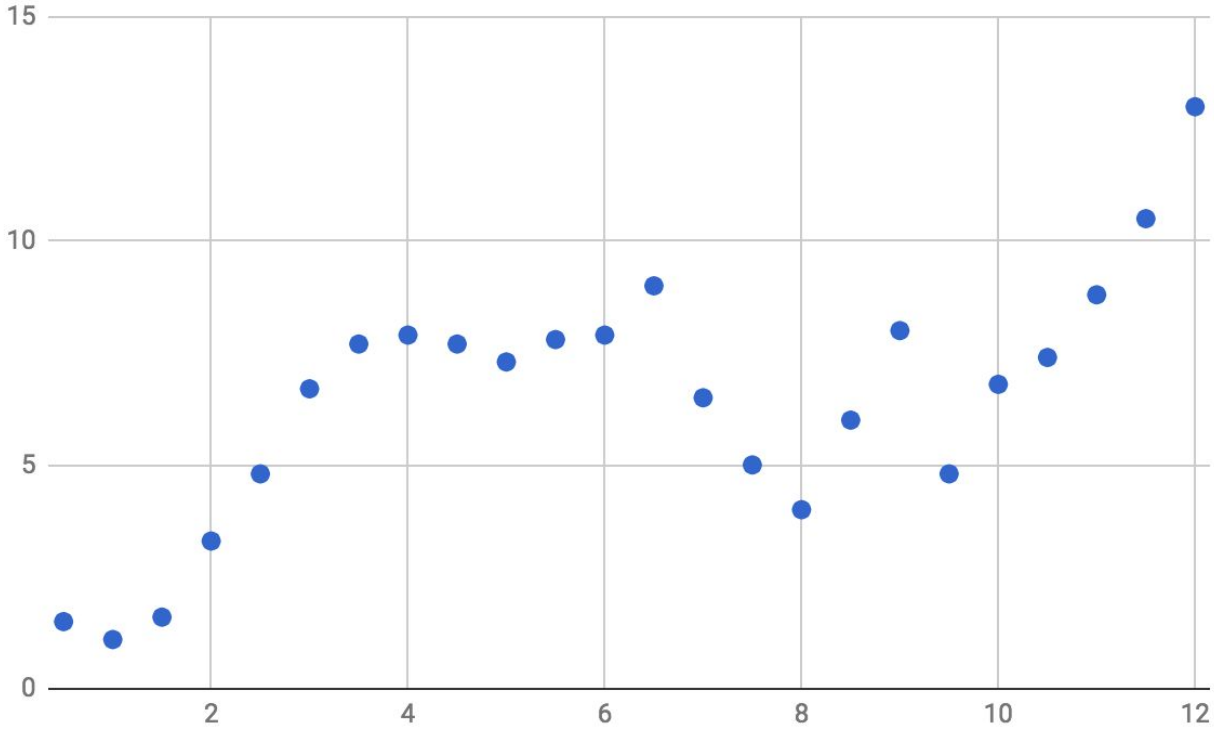
Variance



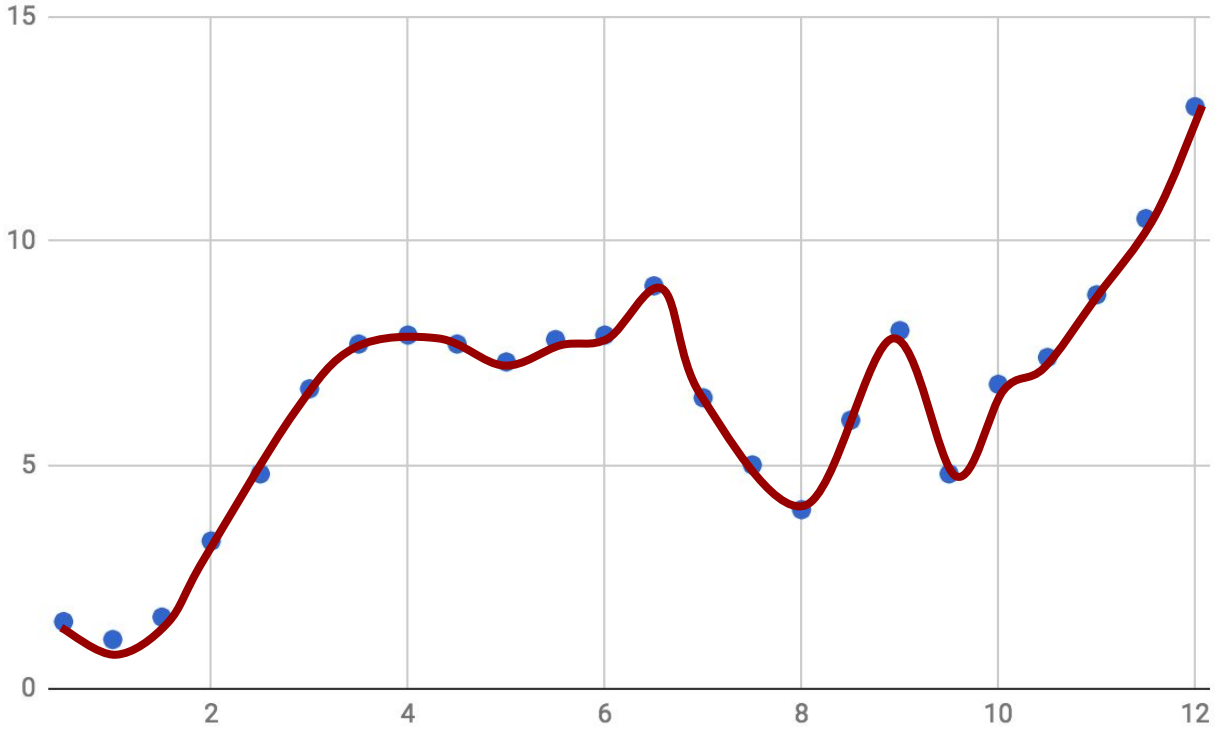
Variance



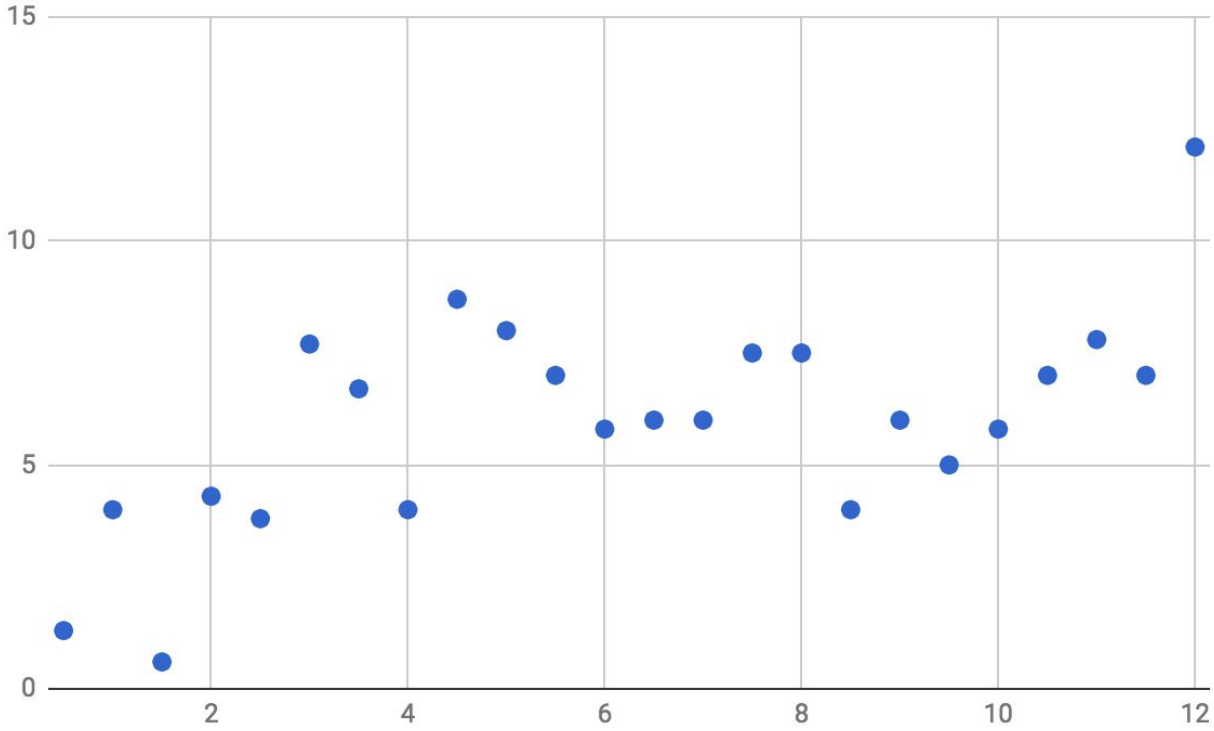
Variance



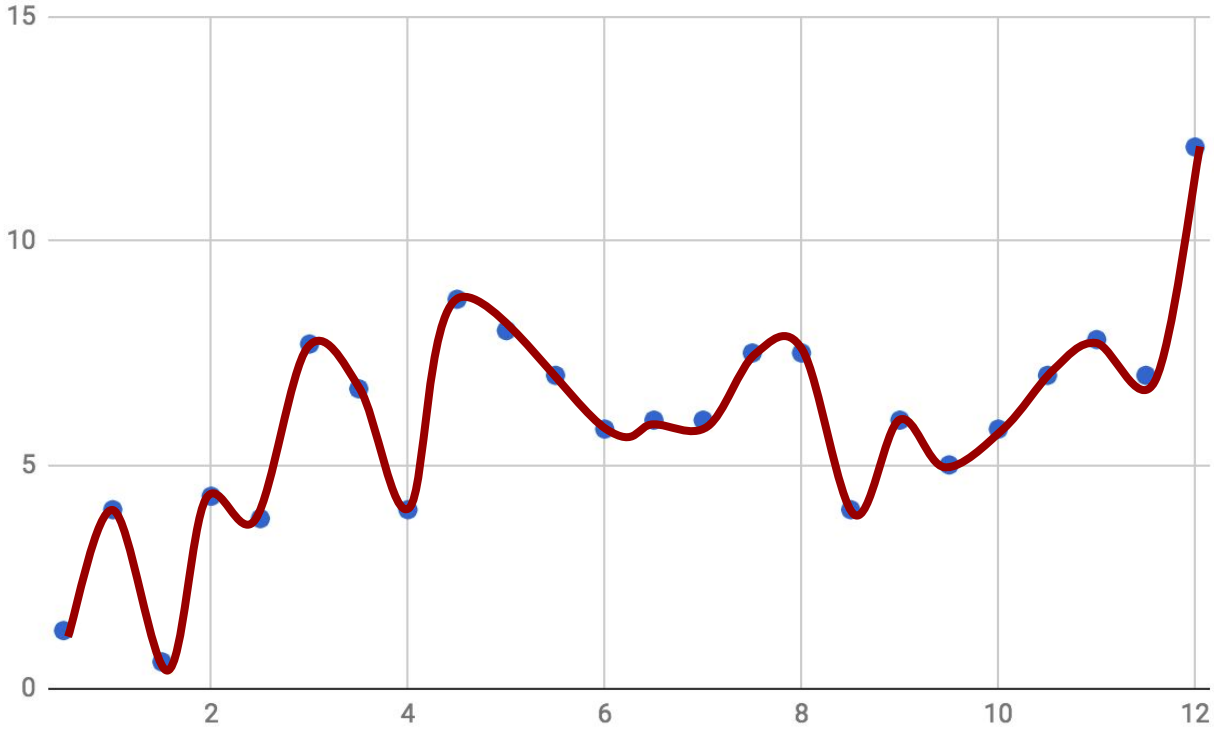
Variance



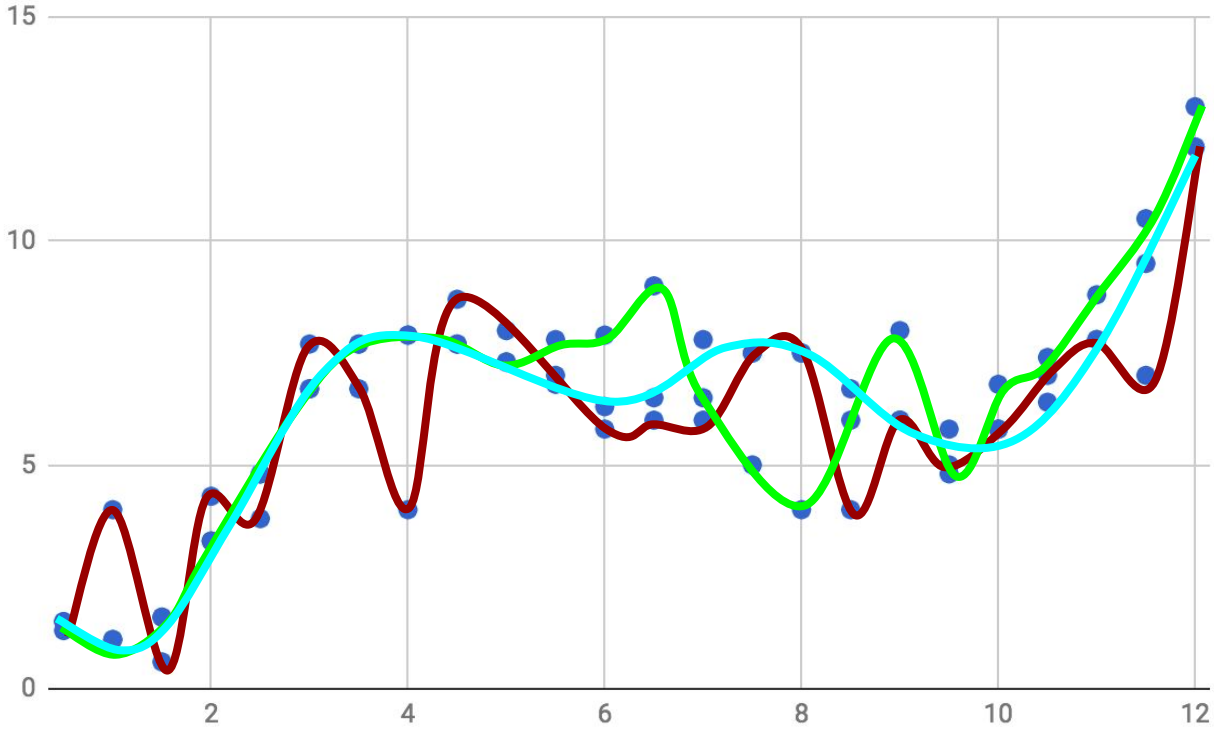
Variance



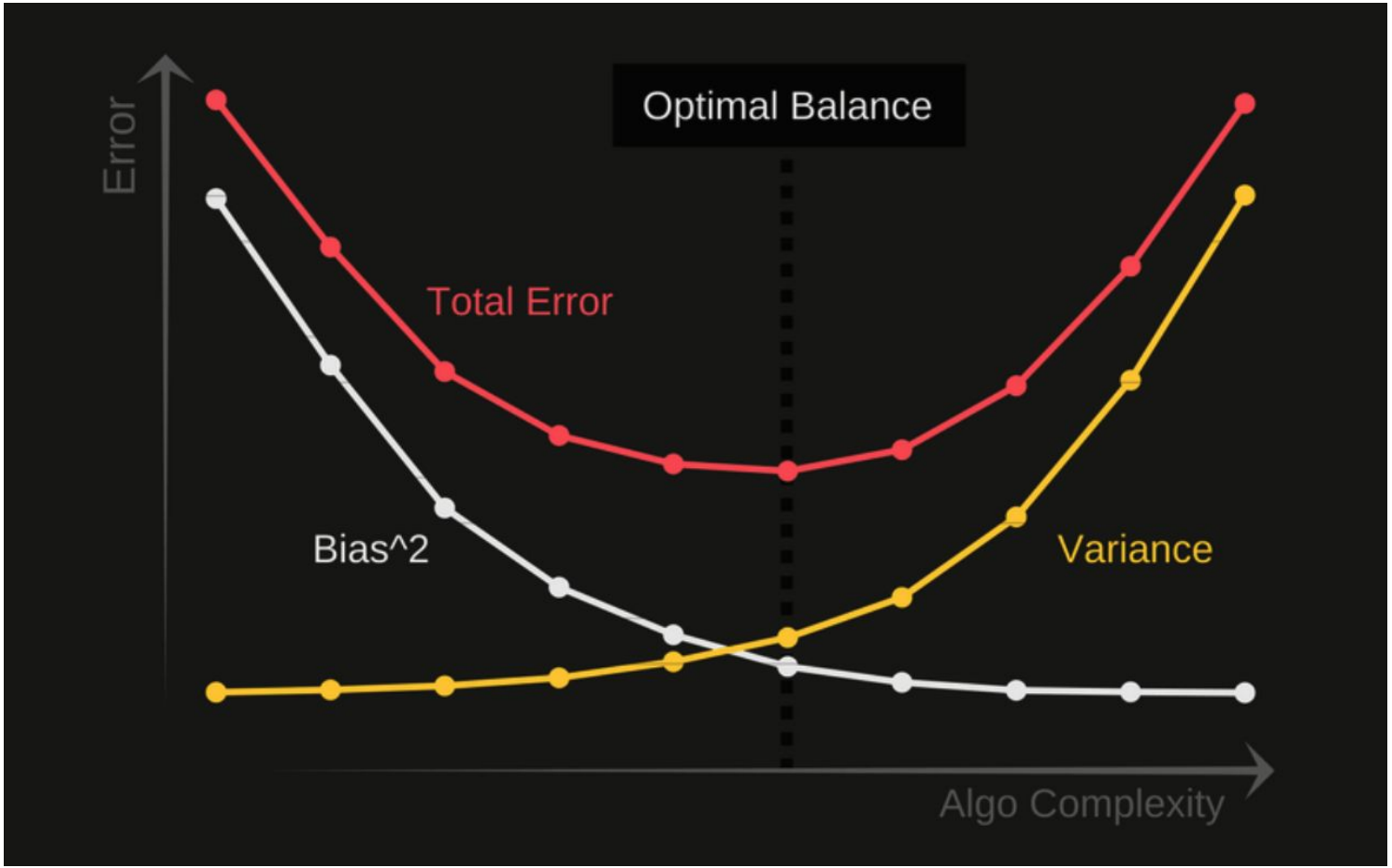
Variance



Variance



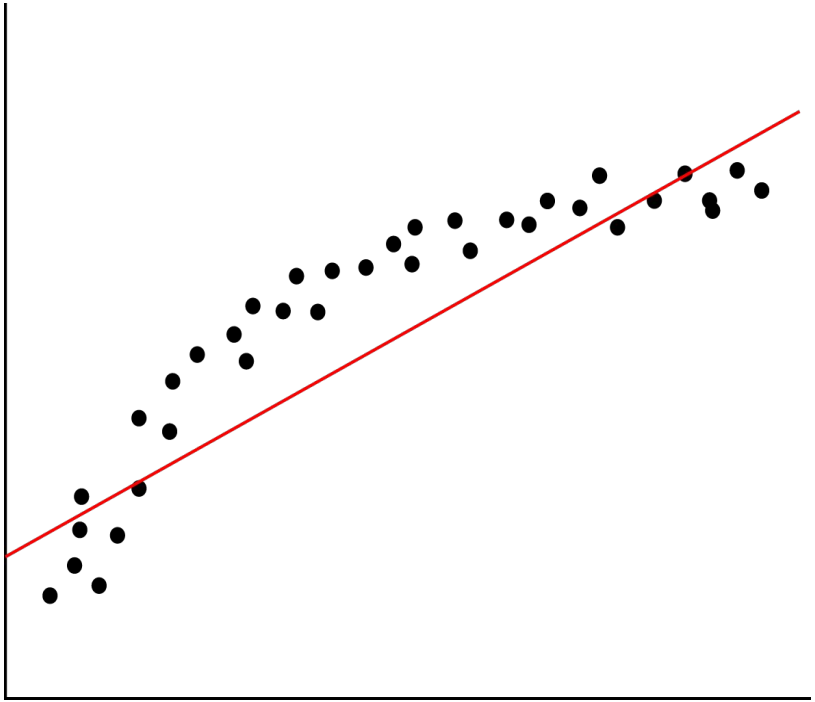
Conceptual Understanding


Optimal Balance
Error
Total Error
Bias^2
Variance
Algo Complexity

Underfitting

Underfitting means we have high bias and low variance.

- Lack of relevant variables/factor
- Imposing limiting assumptions
  - Linearity
  - Assumptions on distribution
  - Wrong values for parameters

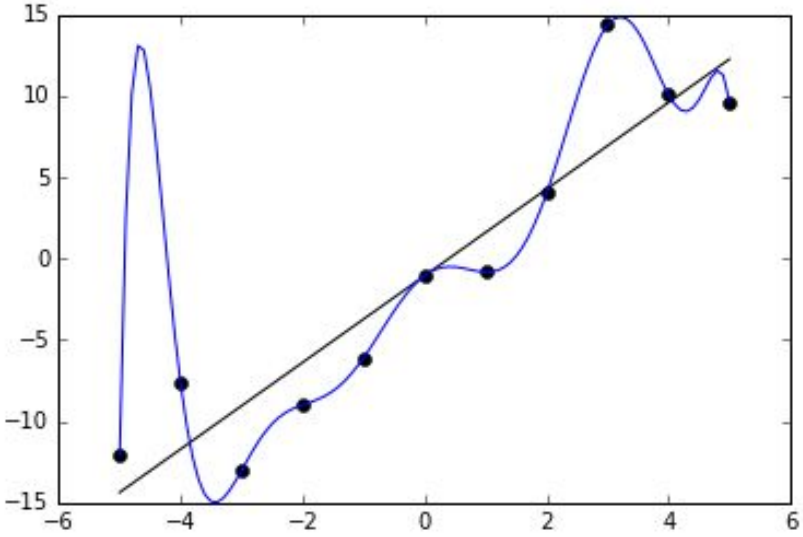


Source

Overfitting

Overfitting means we have low bias and high variance.

- Model fits too well to specific cases
- Model is over-sensitive to sample-specific noise
- Model introduces too many variables/complexities than needed



Source

Question:

Why is overfitting more difficult to control than underfitting?

Variance Reduction

Avoiding overfitting is a **variance reduction** problem

Variance of the model is a function of the variances of each variable

- Reduce the number of variables to use [**Subset Selection**]
- Reduce the complexity of the model [**Pruning**]
- Reduce the coefficients assigned to the variables [**Regularization**]

Cross-validation is used to test the relative predictive power of each set of parameters and subset of features.

Feature Selection

- Lower-dimensional data $\longrightarrow$ faster computation
- Reduces variance in data $\longrightarrow$ less overfitting
- Easier to build intuition with fewer features
- Techniques for picking features

Subset Selection

- **Best subset selection:** Test all 2^p subset selections for best one
- **Forward subset selection**
  - Iterate over k = 0 ... (p-1) predictors
  - At each stage, select the best model with (p-k) predictors
  - Find best model out of the p-1 selected candidates with CV
- **Backward selection** - Reverse of forward subset selection
  - Start from p predictors and work down

In practice, best subset selection method is rarely used, why?

Regularization

We defined our error up until now as:

$$SS_{(residuals)} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Minimizing this equation on training data = minimizing **training loss.**

But we can often do better!

Source

Regularization

To avoid overfitting, we add a penalty term independent of the data, known as **regularization**.

$$\text{Error} = (\text{Training Loss})^2 + \text{Regularization}$$

Ridge Regression

Lasso Regression

http://ww2.tnstate.edu/ganter/BIO-311-Ch12-Eq5a.gif

Ridge Regression

Uses L_2 - regularization penalty:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

λ is the penalty threshold constant and controls sensitivity.

- Useful for <u>non-sparse</u>, <u>correlated</u> predictor variables
- Used when predictor variables have <u>small individual effects</u>
- Limits the magnitudes of the coefficient terms, but not to 0

Lasso Regression

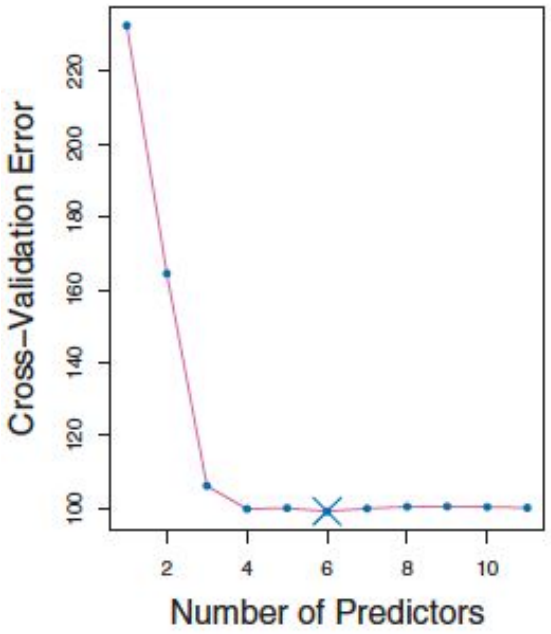
Uses L_1 - regularization penalty:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

This time the penalty term uses absolute value rather than squaring.

- Useful for <u>sparse</u>, <u>uncorrelated</u> variables
- Used when there are few variables with <u>medium to high effects</u>
- Drives coefficients to 0 when λ sufficiently large (performs feature selection)

Training Accuracy vs Test Accuracy



Key idea: Regularization and cross-validation are techniques to limit the model's sensitivity.

If test error is much higher than training error

- If significantly lower:
  - Raise penalty constant
  - Try different subset
  - Try different parameters

Coming Up

Your problem set: Final Project

Next week: Cross Validation and Ensemble

See you then!

Cornell Data Science

CDS Education

Introduction to Machine Learning for Python

Cross Validation

Generally: Cross Validation (CV)

Set of **validation techniques** that use the training dataset itself to validate model

- Allows maximum allocation of training data from original dataset
- Efficient due to advances in processing power

Cross validation is used to test the effectiveness of any model or its modified forms.

Validation Goal

- Estimate Expected Prediction Error
- Best Fit model
- Make sure that the model does not Overfit

Hastie et al. “Elements of Statistical Learning.”

HoldOut Validation

Dataset

HoldOut Validation

Training Sample	Testing Sample

HoldOut Validation

Training Sample	Testing Sample

Advantage: Traditional and Easy
Disadvantage: Varying Error based on how to sample testing

K-fold Validation

Often used in practice with *k*=5 or *k*=10.

Create equally sized *k* partitions, or **folds**, of training data

For each fold:

- Treat the *k-1* other folds as training data.
- Test on the chosen fold.

The average of these errors is the validation error

K-fold Validation

Dataset

Suppose K = 10,

10-Fold CV

K-fold Validation

Training Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Training Sample	Training Sample	Testing Sample

K-fold Validation

Training Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Training Sample	Training Sample	Testing Sample

Calculate RMSE = rmse1

K-fold Validation

Training Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Training Sample	Testing Sample	Training Sample

K-fold Validation

Training Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Training Sample	Testing Sample	Training Sample

Calculate RMSE = rmse2

K-fold Validation

Training Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Testing Sample	Training Sample	Training Sample

K-fold Validation

Training Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Testing Sample	Training Sample	Training Sample

Calculate RMSE = rmse3

K-fold Validation

And so on

K-fold Validation

Testing Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Training Sample	Training Sample	Training Sample

Calculate RMSE = rmse10

K-fold Validation

Testing Sample	Training Sample	Training Sample	Training Sample	Training Sample
Training Sample	Training Sample	Training Sample	Training Sample	Training Sample

RMSE = Avg(rmse1...10)

K-fold Validation

Less matters how we divide up

Selection bias not present

Leave-One-Out Method

Leave-One-Out Method

Dataset

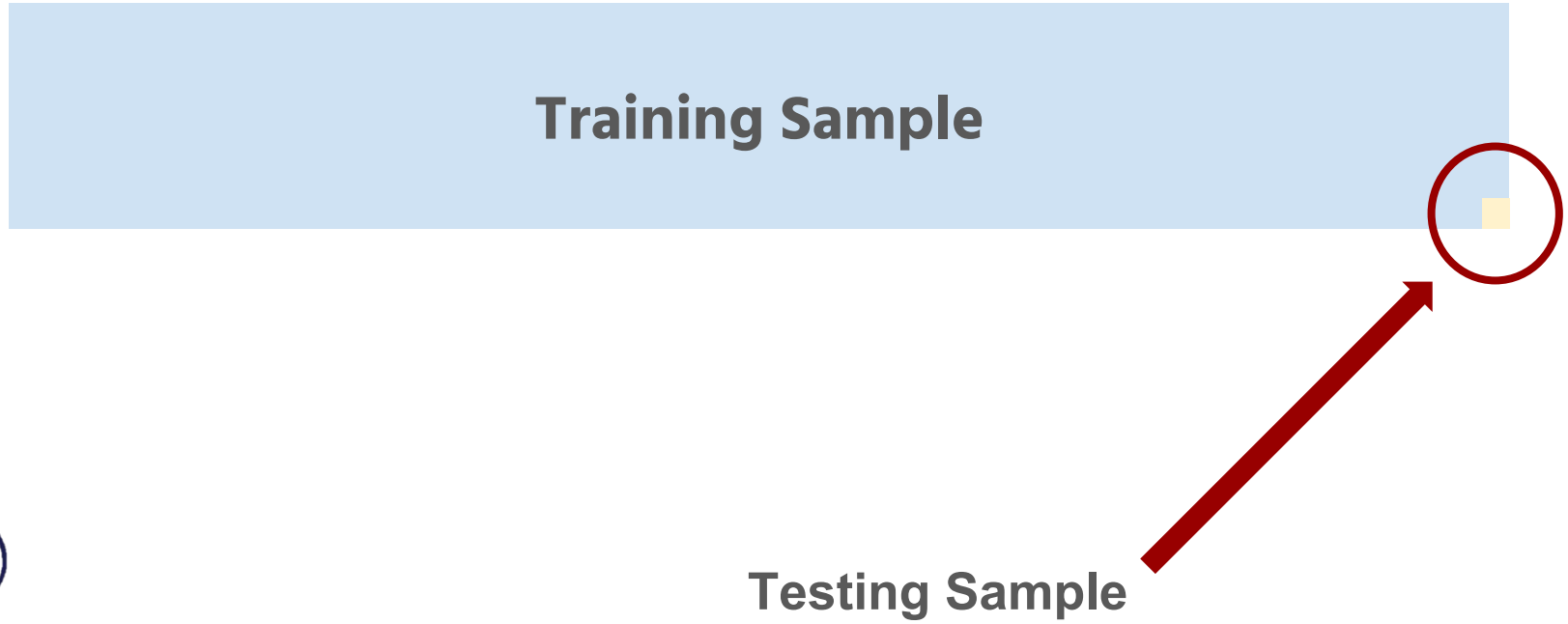
Leave-One-Out Method

Training Sample

Leave-One-Out Method

What just happened?

Leave-One-Out Method



Leave-P-Out Validation

For each data point:

- Leave out p data points and train learner on the rest of the data.
- Compute the test error for the p data points.

Define average of these ${}_nC_p$ error values as validation error

Source

Leave-P-Out Validation

A really exhaustive and thorough way to validate

High Computation Time

Question:

How are *k*-fold and leave-p-out different?

Coming Up

Your problem set: Final Project

Next week: Ensemble

Cornell Data Science