Feature Engineering, Bias-Variance Tradeoff (Reading: Ch 14, Ch 15)

Learning goals:

- Understand how feature engineering extends our repertoire of models.
- Learn about the many factors that affect the bias-variance tradeoff.

UC Berkeley Data 100 Summer 2019
Sam Lau

(Slides adapted from Sandrine Dudoit and Joey Gonzalez)

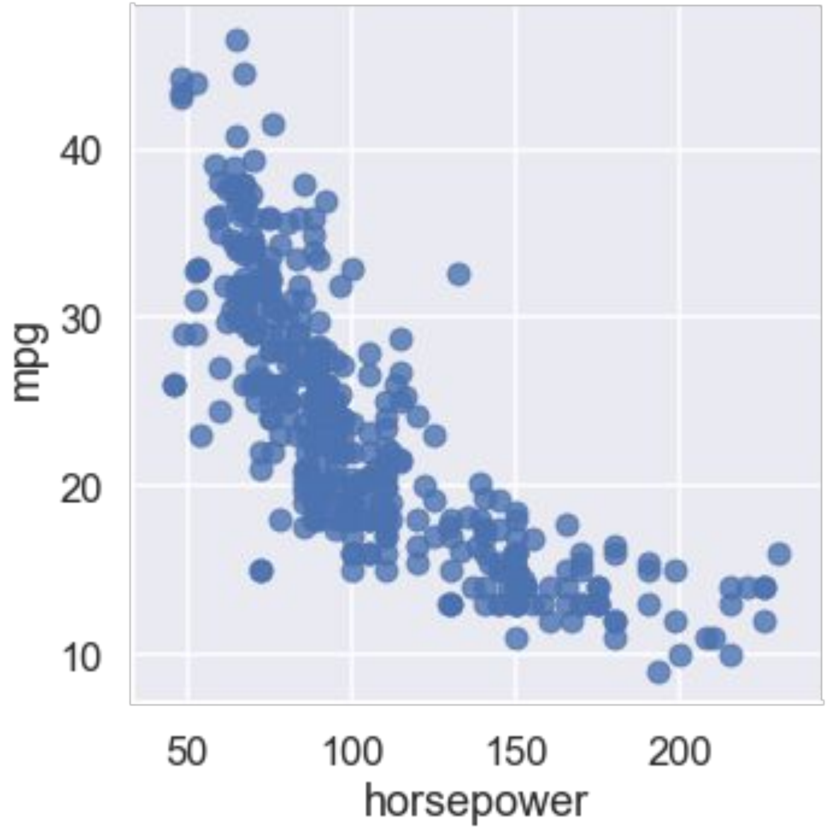
Announcements

- HW5 out, due **Friday**
- HW6 out Friday, due **Tuesday**

Feature Engineering

Linear Models Level Up

- Horsepower and mpg have a nonlinear relationship.
- Can still use linear regression to capture this!
- **Feature engineering**: creating new features from data to give model more complexity.



Adding Features

- For now, predict MPG from horsepower alone.
- Insight: Add a new column to X with horsepower^2.

	bias	hp	hp^2
0	1	130.0	16900.0
1	1	165.0	27225.0
2	1	150.0	22500.0
...	...	...	...
395	1	84.0	7056.0
396	1	79.0	6241.0
397	1	82.0	6724.0

- Now we fit a quadratic function!

$$\begin{aligned} f_\theta(\boldsymbol{x}) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ &= \theta_0 + \theta_1 \mathrm{hp} + \theta_2 \mathrm{hp}^2 \end{aligned}$$

- This is still linear in **model weights** θ, so we call it a linear model.

(Demo)

Polynomial Regression

- For polynomial features of degree n, usually add every possible combination of columns.
  - 4 original columns, degree 2:

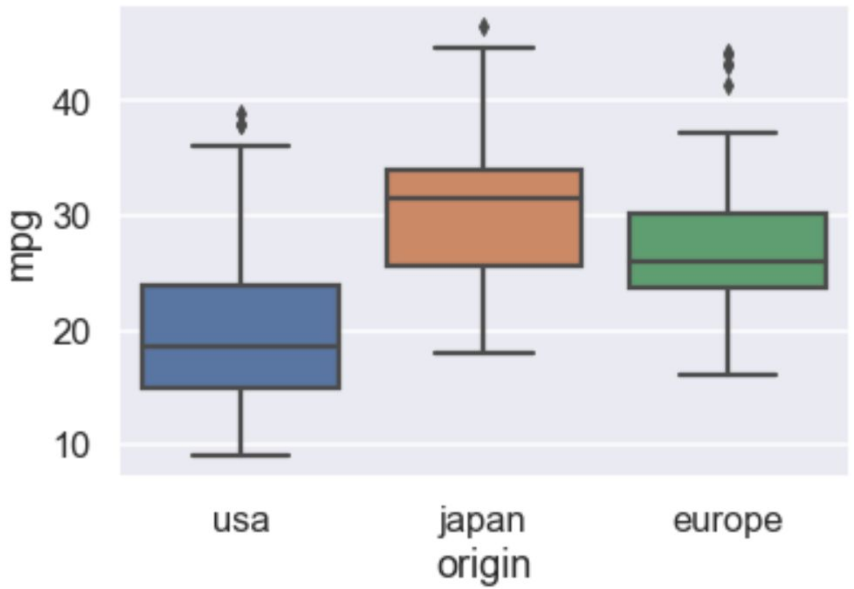
$$x_1, x_2, x_3, x_4,$$
$$x_1^2, x_2^2, x_3^2, x_4^2,$$
$$x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4$$

- Can end up being a lot of columns
- To cope, use kernel trick (covered in advanced courses)

Categorical Features

- Origin column is correlated with MPG. Can we use it?
- Idea: Encode categories as numbers in a smart way.
- Discuss: Why can't we just encode "usa" as 0, "japan" as 1, "europe" as 2?

```
sns.boxplot(x='origin', y='mpg', data=mpg);
```



One-Hot Encoding

- One-hot encoding makes one new column for each unique category:

origin
usa
usa
europe
...
usa
japan
japan

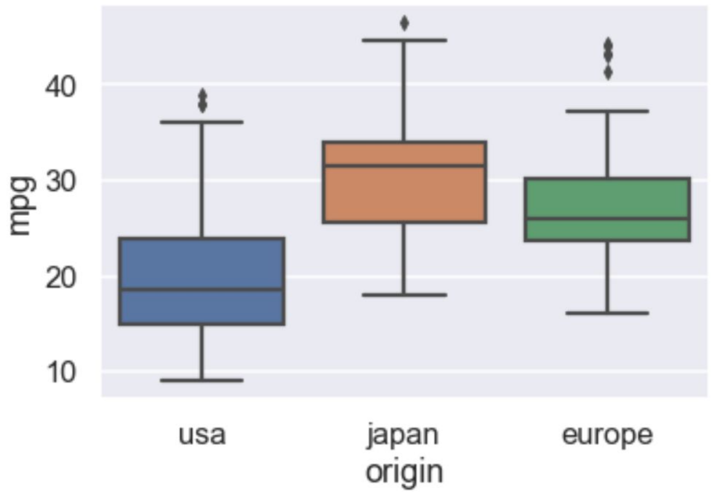
origin=usa	origin=europe	origin=japan
1	0	0
1	0	0
0	1	0
...	...	...
1	0	0
0	0	1
0	0	1

One-Hot Encoding

- What do you expect the largest weight to be?

origin=usa	origin=europe	origin=japan
1	0	0
1	0	0
0	1	0
...	...	...
1	0	0
0	0	1
0	0	1

```
sns.boxplot(x='origin', y='mpg', data=mpg);
```



- Can interpret weight as “contribution” of that category

One Hot Problem

- Problem: Adding a new column for each category makes columns of X **linearly dependent**! Why?
- One-hot columns always sum to 1:

bias		origin=usa		origin=europe		origin=japan
1		1		0		0
1		1		0		0
1		0		1		0
...	=	...	+	...	+	...
1		1		0		0
1		0		0		1
1		0		0		1

- This makes normal equations unsolvable.

$$\hat{\boldsymbol{\theta}} = \boxed{(\boldsymbol{X}^\top \boldsymbol{X})^{-1}} \boldsymbol{X}^\top \boldsymbol{y}$$

Not invertible ^

Weight Interpretation

- Invertibility isn't a problem for gradient descent, but this still affects how we interpret the model weights.
- Linearly dependent columns can "swap" weights:
  - Left: All categories matter. Right: No categories matter!

bias	origin=usa	origin=europe	origin=japan
0	**3**	**3**	**3**
1	1	0	0
1	1	0	0
1	0	1	0
...	...	...	...
1	1	0	0
1	0	0	1

=

bias	origin=usa	origin=europe	origin=japan
3	**0**	**0**	**0**
1	1	0	0
1	1	0	0
1	0	1	0
...	...	...	...
1	1	0	0
1	0	0	1

Drop it Like it's Hot

- Simple fix: Drop the last one-hot column.
- In this case, the weight for USA can be interpreted as "change in MPG between USA and Japan".

origin
usa
usa
europe
...
usa
japan

origin=usa	origin=europe
1	0
1	0
0	1
...	...
1	0
0	0

(Demo)

Features feat. More Features

- Feature engineering is often domain-specific:
  - Standardizing: “How many SDs away from average?”
  - Log transform: Used to fit exponential models.
  - Absolute difference: “How different is the current temperature from 70°?”
  - Binning data, then one-hot encoding: “Are we driving during morning rush hour? Evening rush hour?”
  - Date-related features: year, month, weekday
  - Image-related features: blurring, edge detection, etc.

Break!
Fill out Attendance:
http://bit.ly/at-d100

The Bias-Variance Tradeoff

The Feature Question

- How do we know when to stop adding features?
  - E.g. degree 2 polynomial? Degree 10 polynomial?
- In general, adding a new feature decreases training error. Why?

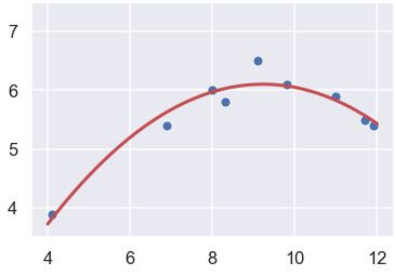
(Demo)

How do we decide to keep a feature?

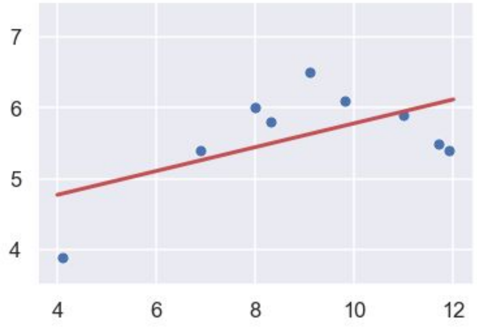
- Adding too many features causes **overfitting**.
- Approach from Data 8: split sample into a training set, validation set, and test set.
  - Fit all models on training set
  - Choose model with lowest validation error
  - Use test error as final error
- Tomorrow: cross-validation technique
- Intuition: we're trying to estimate model error on unseen data, so we need to validate using untrained data.

Bias Redux

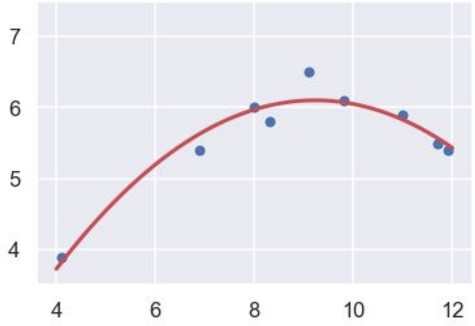
- Remember estimator bias? Idea also applies to models.
- Unbiased models are able to fit the population model.
- If population model is:



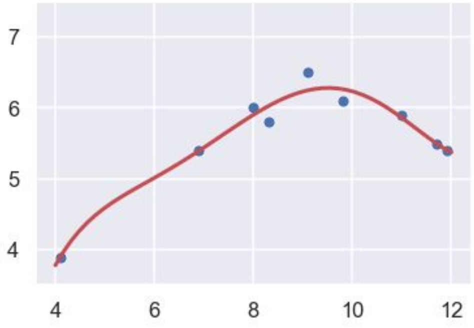
Biased



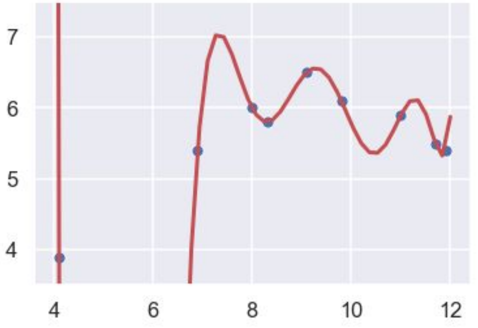
Unbiased



Unbiased



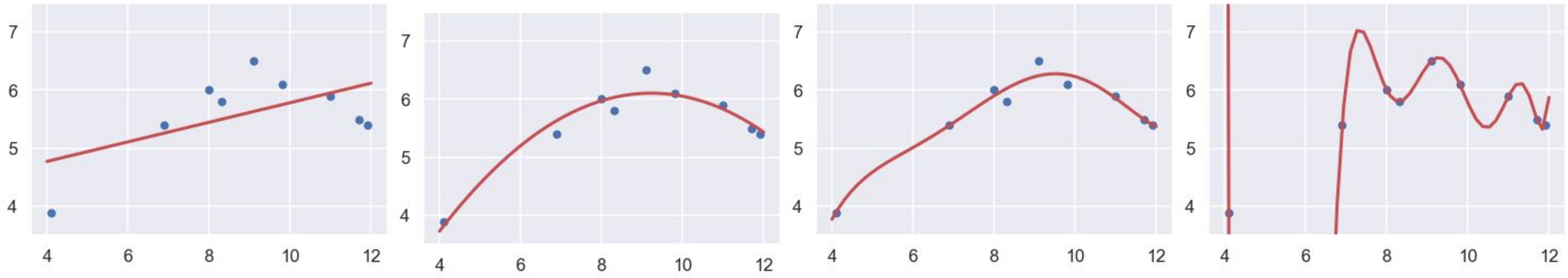
Unbiased



Variance Redux

- Remember estimator variance?
- Models with high variance have very different fits for the same data.

Lowest Variance → **Highest Variance**



Risk Redux

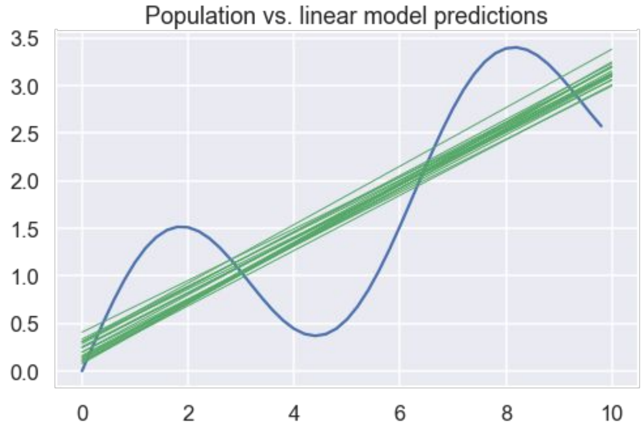
- Model risk is the expected loss for all possible model fits and for all input-output points in the population γ, z.

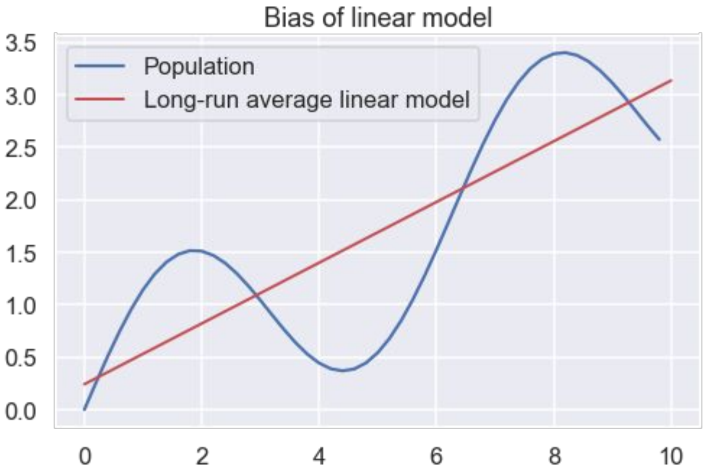
$$R(f_{\hat{\theta}}) = E(\ell(\gamma, f_{\hat{\theta}}(z)))$$

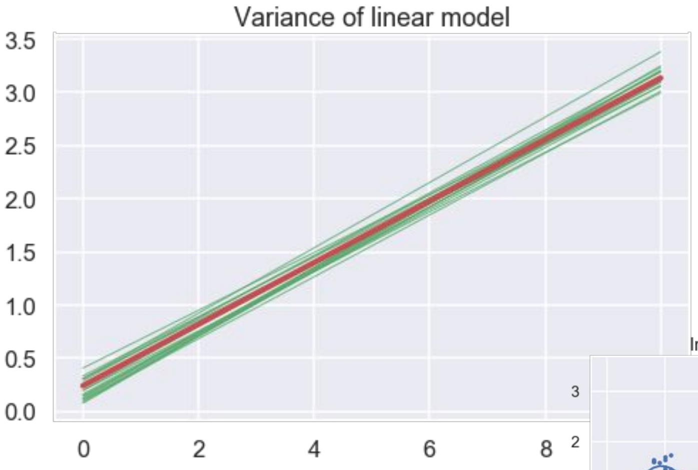
$$R(f_{\hat{\theta}}) = \underbrace{(E[f_{\hat{\theta}}(z)] - f_{\theta}^{*}(z))^2}_{\text{Model bias}} + \underbrace{\mathrm{Var}(f_{\hat{\theta}}(z))}_{\text{Model variance}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible error}}$$

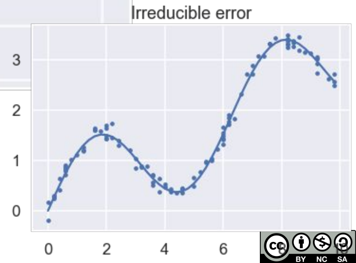
- Simple models have high bias and low variance
- Complex models have low bias and high variance

Bias and Var Visualized

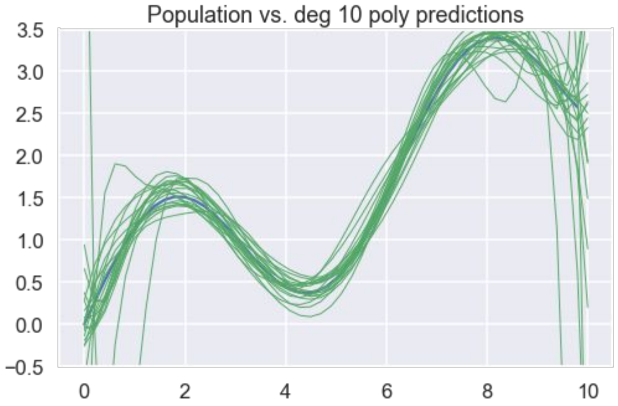


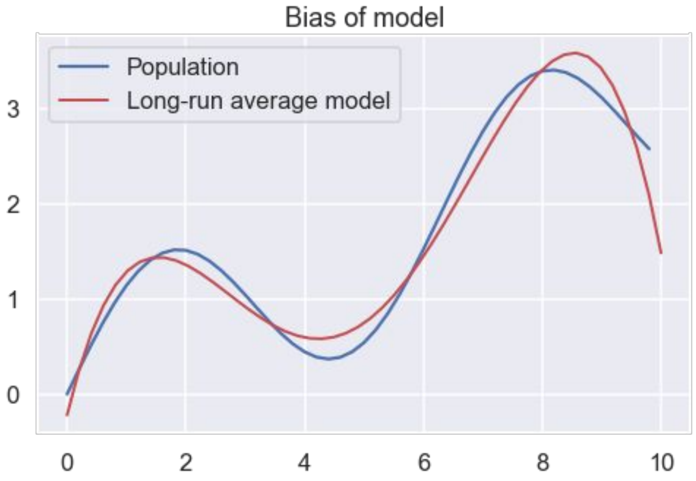


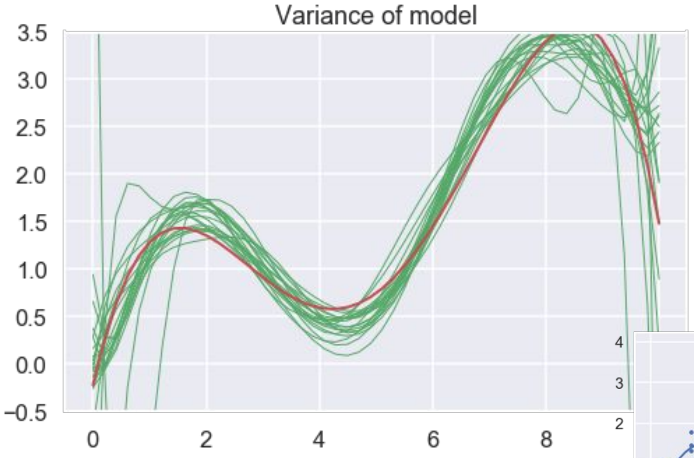


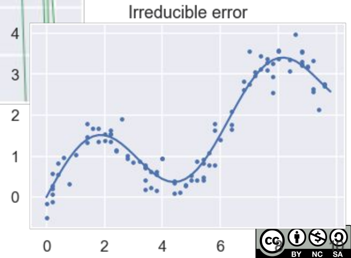


Bias and Var Visualized









Bias-Variance Breakdown

- If population model is linear and we use a linear model, bias is 0 (we get pop model on average), and:

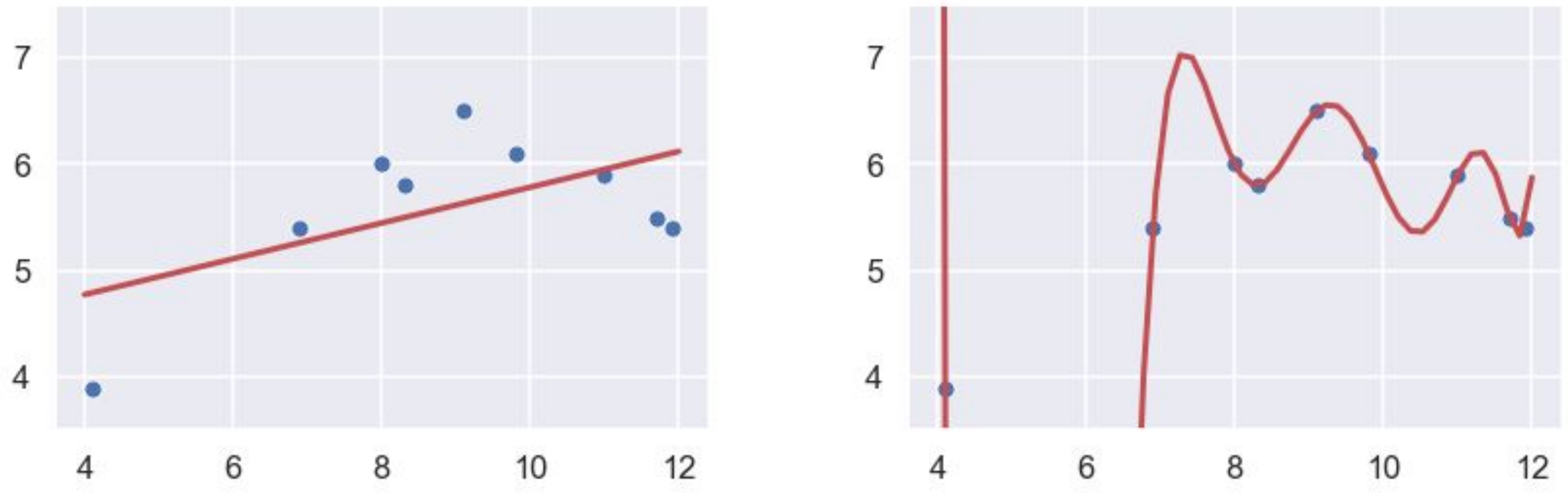
$$\mathrm{Var}(f_{\hat{\theta}}(z)) \approx \sigma^2 \frac{p}{n}$$

σ^2 is the variance of the error
p is the number of features in training data
n is the number of points in training data

- Decrease variance: remove features or collect more data
- Similar breakdown for all models but math is complicated.
  - E.g. For kNN, variance increases as k decreases.

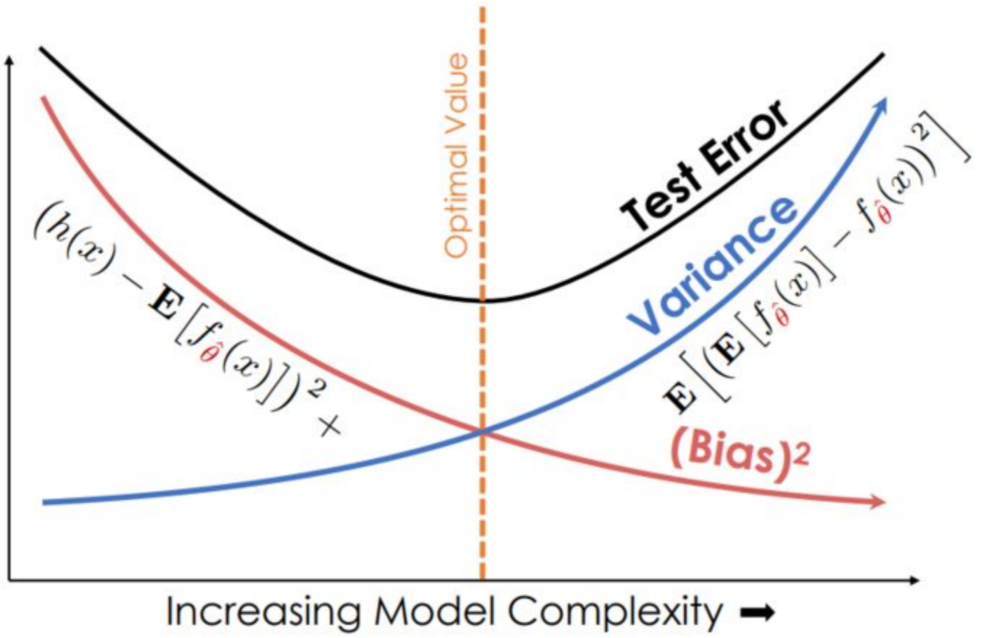
Using the Bias-Variance Tradeoff

- Let's analyze models in terms of bias and variance.
- Model **underfitting** caused by too much bias.
- Model **overfitting** caused by too much variance.



Using the Bias-Variance Tradeoff

- Training error only reflects bias.
- Test error reflects bias and variance.
- This is why we can't see overfitting from training error.
- Test error goes down, then back up as var increases



Using the Bias-Variance Tradeoff

- Adding a good feature decreases bias.
- Adding any feature increases variance even if useless.
- Noise in test set only affects irreducible error Var(ϵ).
- Noise in training set only affects bias and Var(f).
- You need lots of data to model complex relationships.
  - E.g. neural networks have low bias but very high variance, so we often need tons of data to use them.

(Demo)

Summary

- Feature engineering generates flexible models using linear regression.
- The bias-variance tradeoff gives us guidelines for selecting models.
- **Don’t trust training error alone!**