Diagnosing ML System:
Bias, Variance and Regularization

Debugging a learning algorithm

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- You have built an awesome linear regression model
- Work perfectly on the training data

Source: Andrew Ng

Debugging a learning algorithm

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- You have built you awesome linear regression model predicting price
- Work perfectly on you testing data
- Then it fails miserably when you test it on the new data you collected

Source: Andrew Ng

Debugging a learning algorithm

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- You have built you awesome linear regression model predicting price
- Work perfectly on you testing data
- Then it fails miserably when you test it on the new data you collected
- What to do now?

Source: Andrew Ng

Things You Can Try

- Get more data
- Try different features
- Try tuning your hyperparameter

Things You Can Try

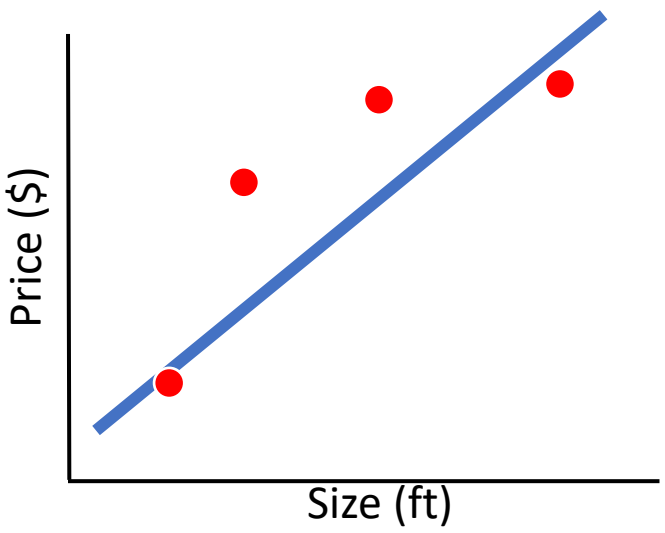
- Get more data
- Try different features
- Try tuning your hyperparameter

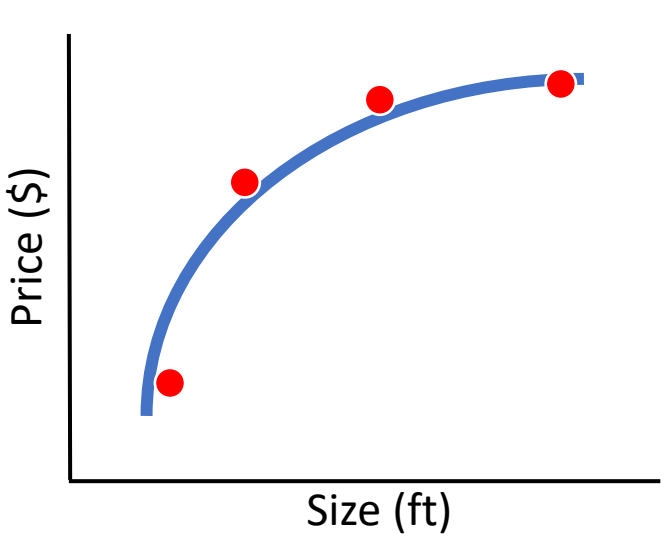
- But which should I try first?

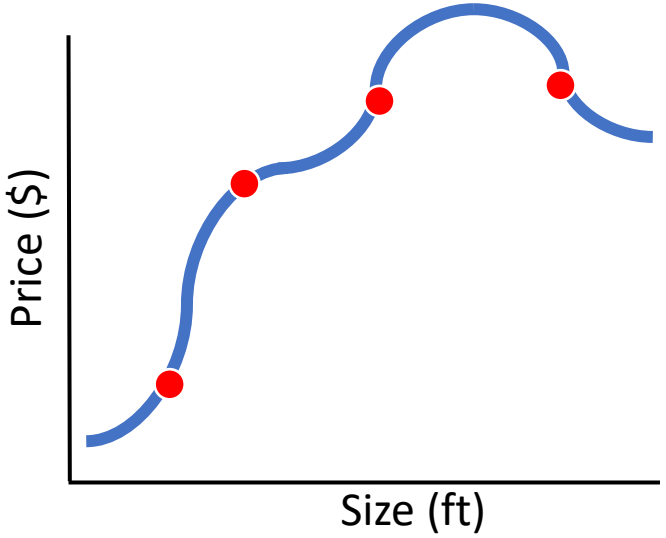
Diagnosing Machine Learning System

- Figure out what is wrong first
- Diagnosing your system takes time, but it can save your time as well
- Ultimate goal: low generalization error → Regularization

Evaluate Your Hypothesis: Regression

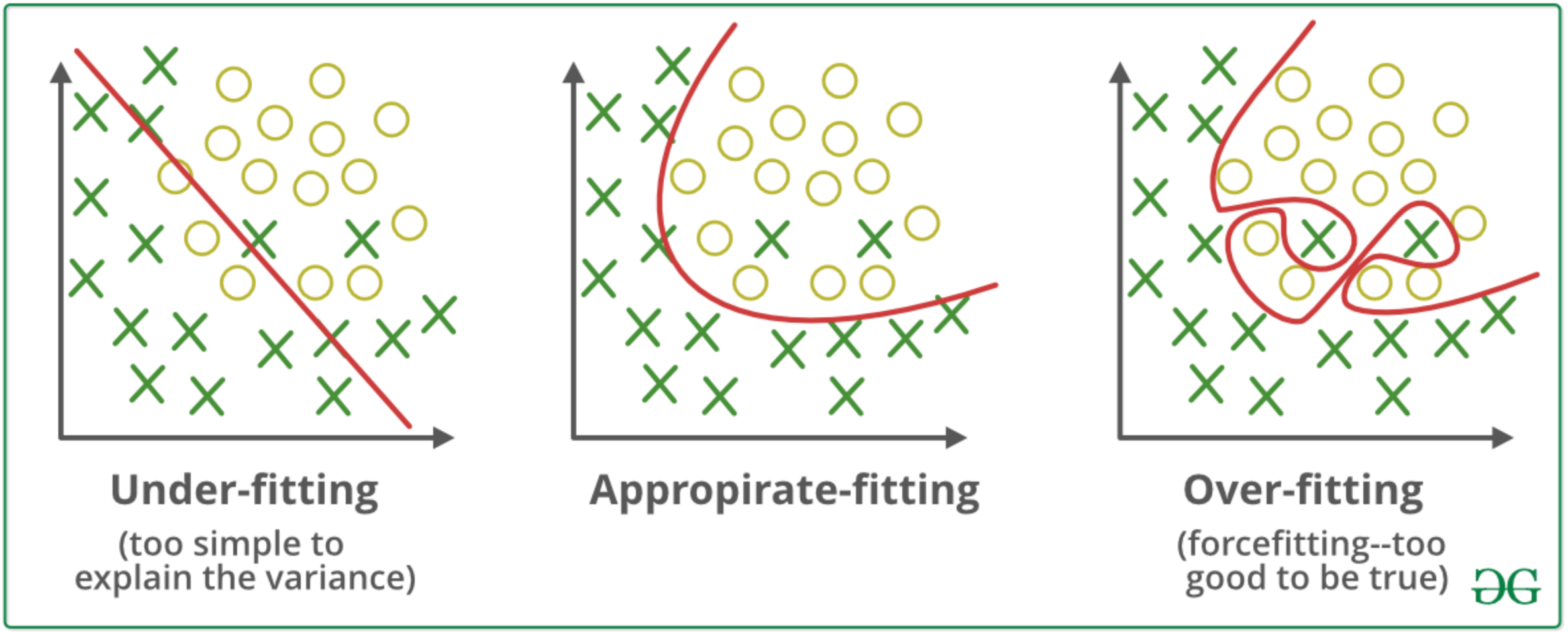






Source: Andrew Ng

Evaluate Your Hypothesis: Classification



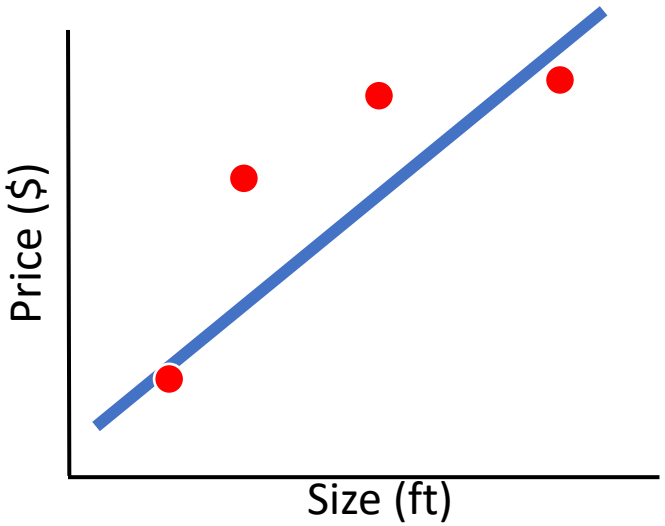
Regularization: A solution to Overfitting

- **Overfitting** is a phenomenon that occurs when a Machine Learning model is constraint to training set and not able to perform well on unseen data.

- **Regularization** is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoiding overfitting.
  The commonly used regularization techniques are :

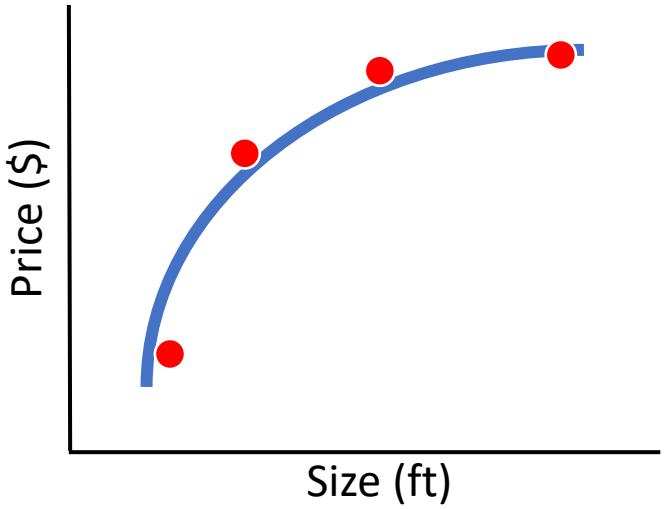
  - L1 regularization → LASSO Regression
  - L2 regularization → RIDGE Regression

Evaluate Your Hypothesis



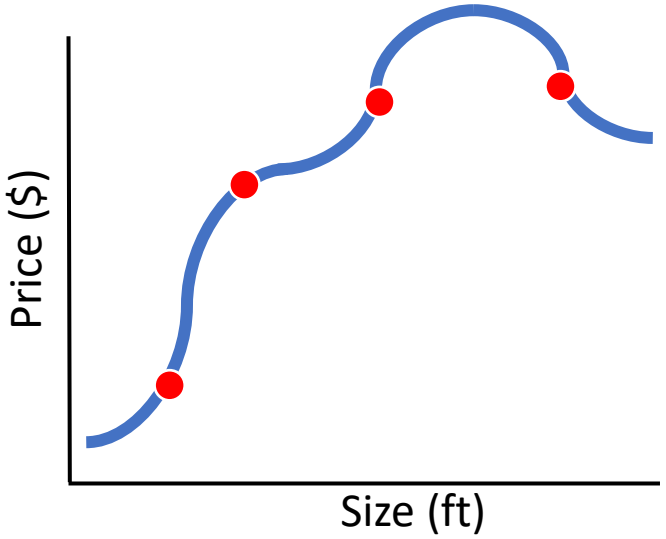
$\theta_0 + \theta_1 x$

Underfit



$\theta_0 + \theta_1 x + \theta_2 x^2$

Just right



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfit

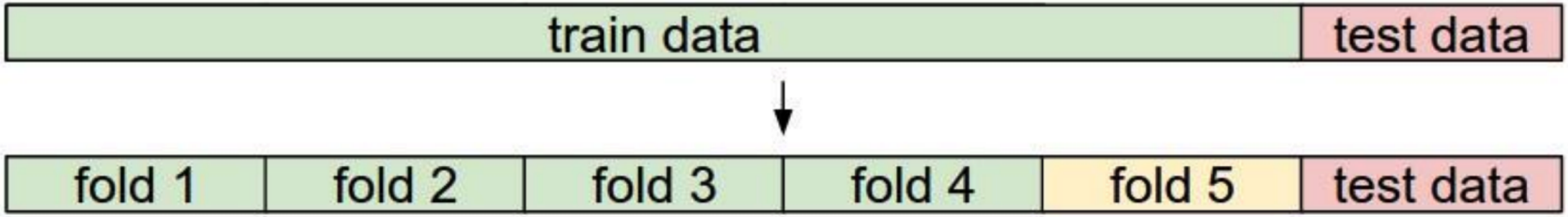
Source: Andrew Ng

Model Selection

- Model does not generalize to unseen data
  - Fail to predict things that are not in training sample
  - Pick a model that has **lower generalization error**

Cross Validation

- Model does not generalize to unseen data
  - Fail to predict things that are not in training sample
  - Pick a model that has **lower generalization error**



Model Selection

- Model does not generalize to unseen data
  - Fail to predict things that are not in training sample
  - Pick a model that has **lower generalization error**

- How to evaluate generalization error?
  - Split your data into *train*, *validation*, and *test set*.
  - Use test set error as an *estimator* of generalization error

Model Selection

- Training error

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

- Validation error

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

- Test error

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

Model Selection

- Training error

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

- Validation error

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

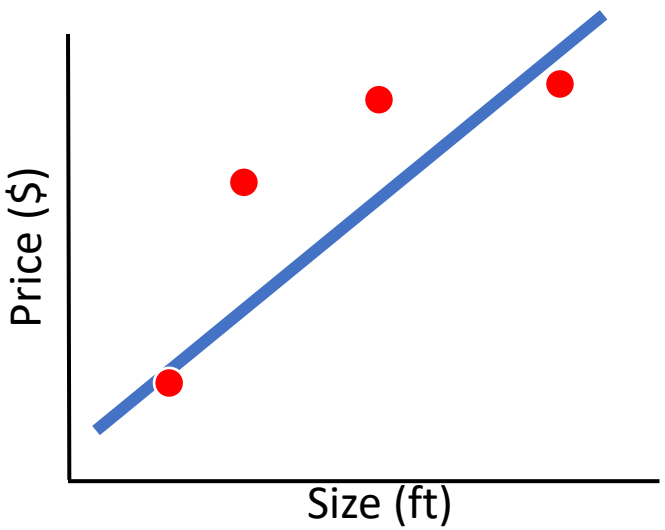
- Test error

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

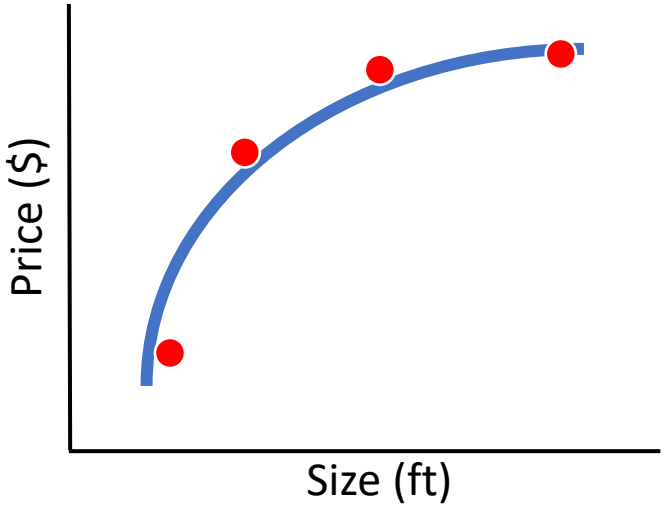
Procedure:

Step 1. Train on training set
Step 2. Evaluate validation error
Step 3. Pick the best model based on Step 2.
Step 4. Evaluate the test error

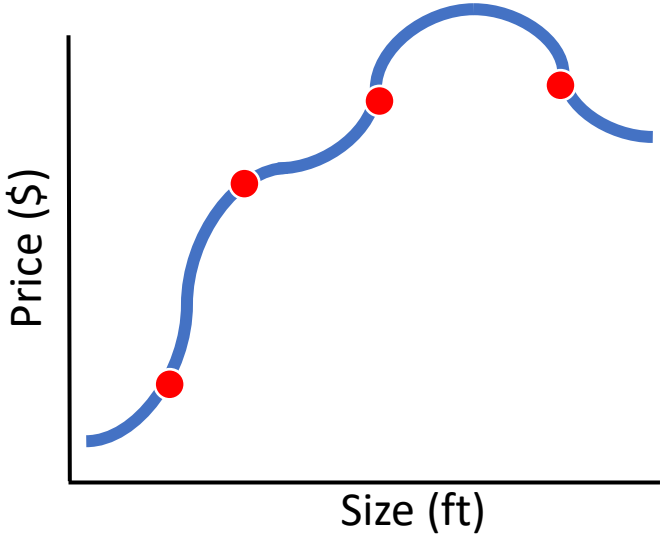
Bias/Variance Trade-off



Underfit



Just right



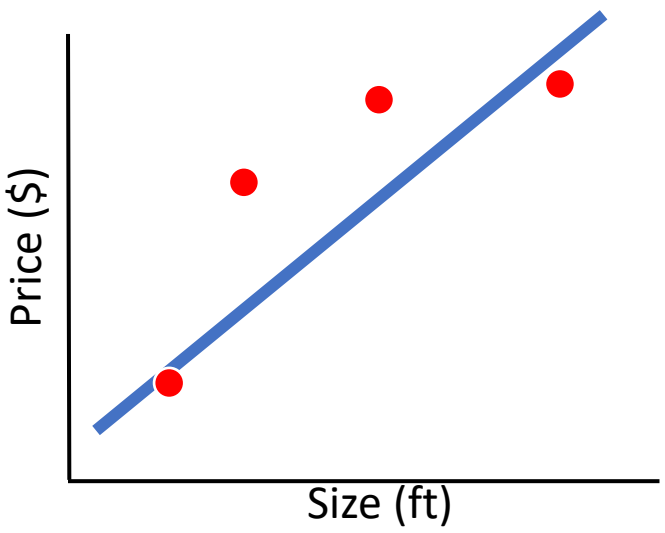
Overfit

Source: Andrew Ng

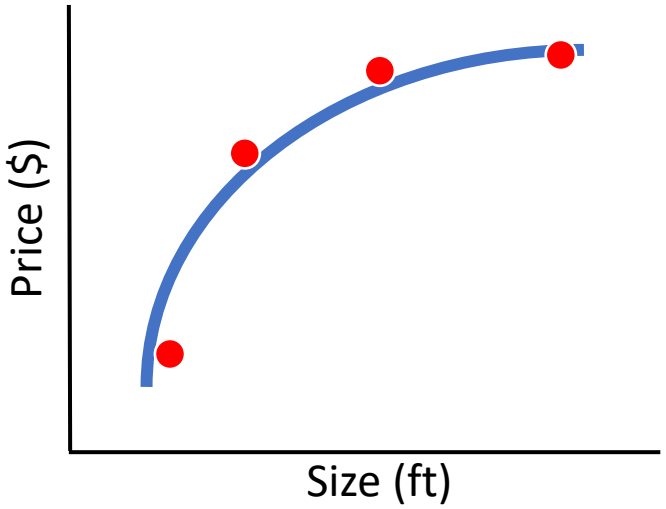
Bias vs. Variance

- We can define **bias** as the error between average model prediction and the ground truth. Moreover, it describes how well the model matches the training data set:
  - A model with a higher bias would not match the data set closely.
  - A low bias model will closely match the training data set.
- **Variance** refers to the changes in the model when using different portions of the training data set/unseen data. It is the variability in the model prediction—how much the ML function can adjust depending on the given data set.

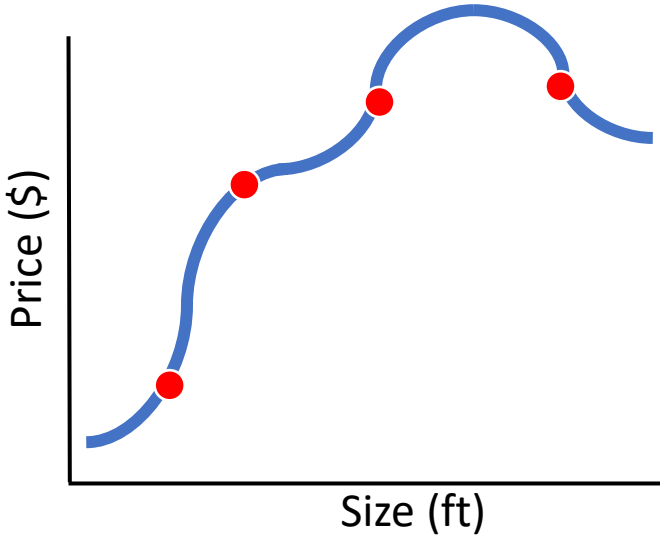
Bias/Variance Trade-off



Underfit
High bias



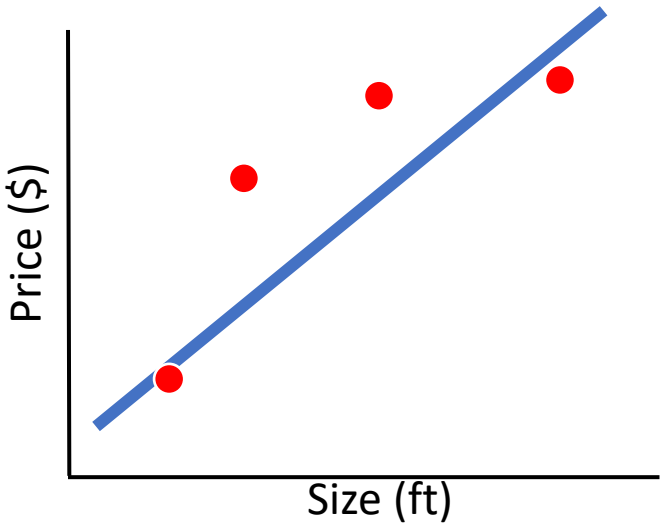
Just right



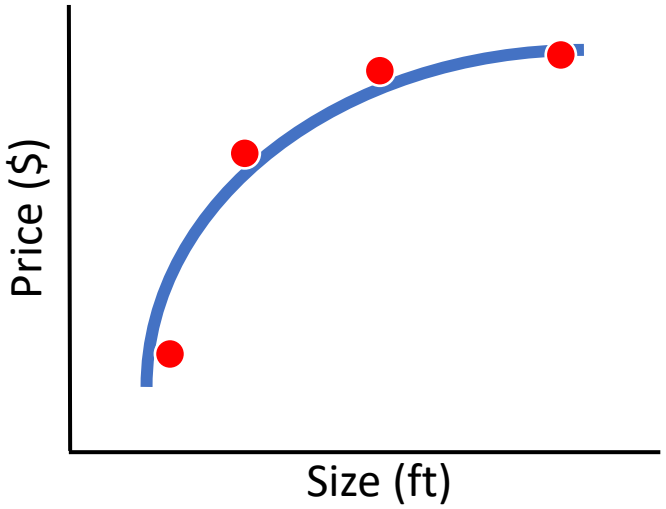
Overfit
High Variance

Source: Andrew Ng

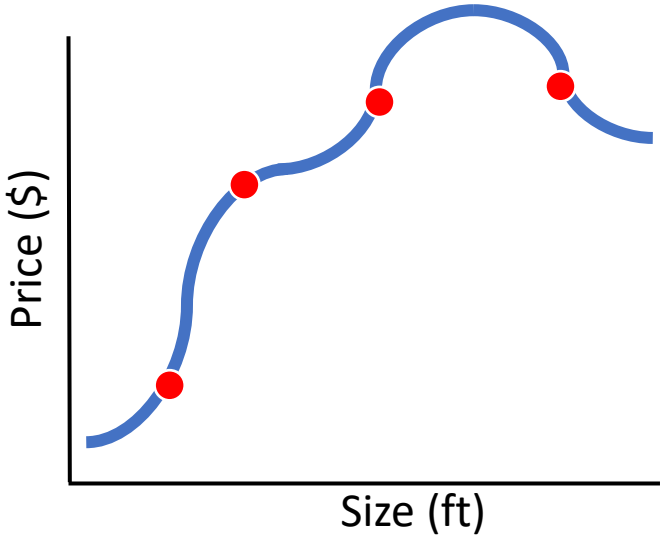
Bias/Variance Trade-off



Underfit
High bias
Too simple



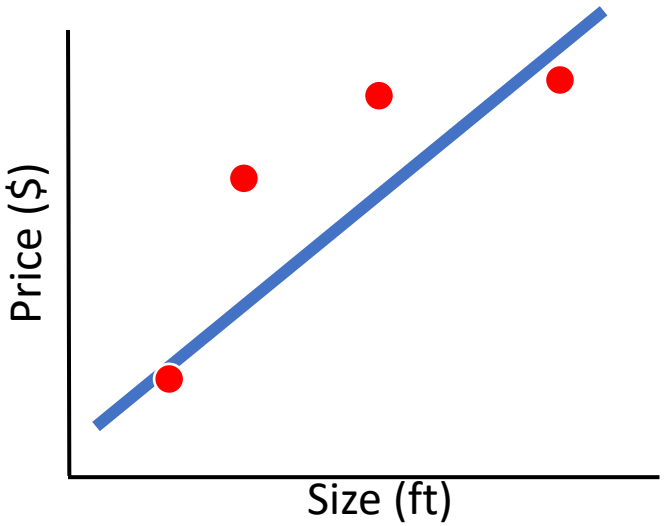
Just right



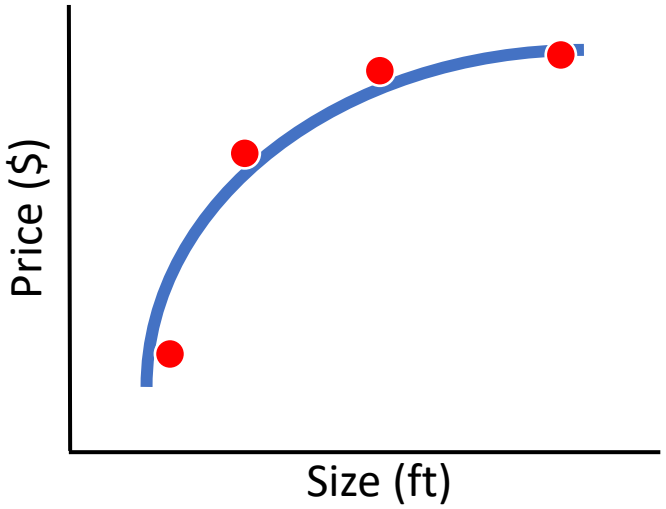
Overfit
High Variance
Too Complex

Source: Andrew Ng

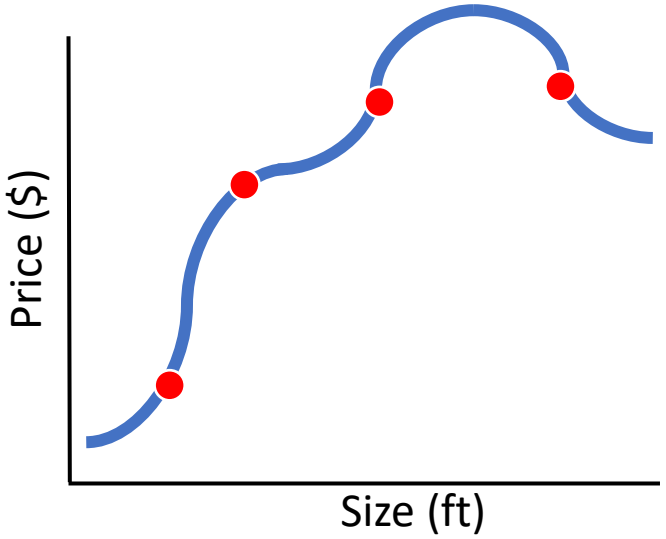
Linear Regression with Regularization



Underfit
High bias
Too simple
Too much regularization



Just right

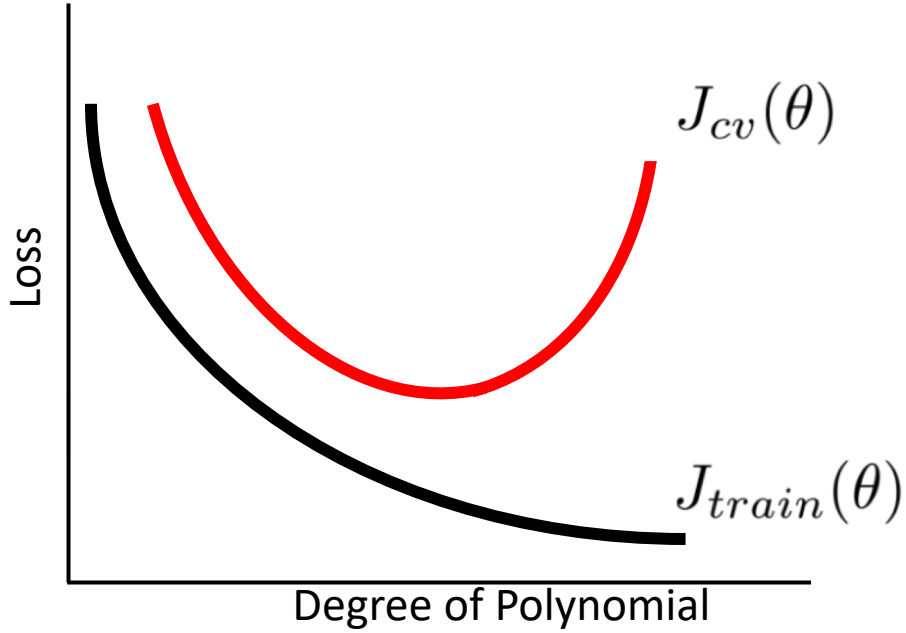


Overfit
High Variance
Too Complex
Too little regularization

Source: Andrew Ng

Bias / Variance Trade-off

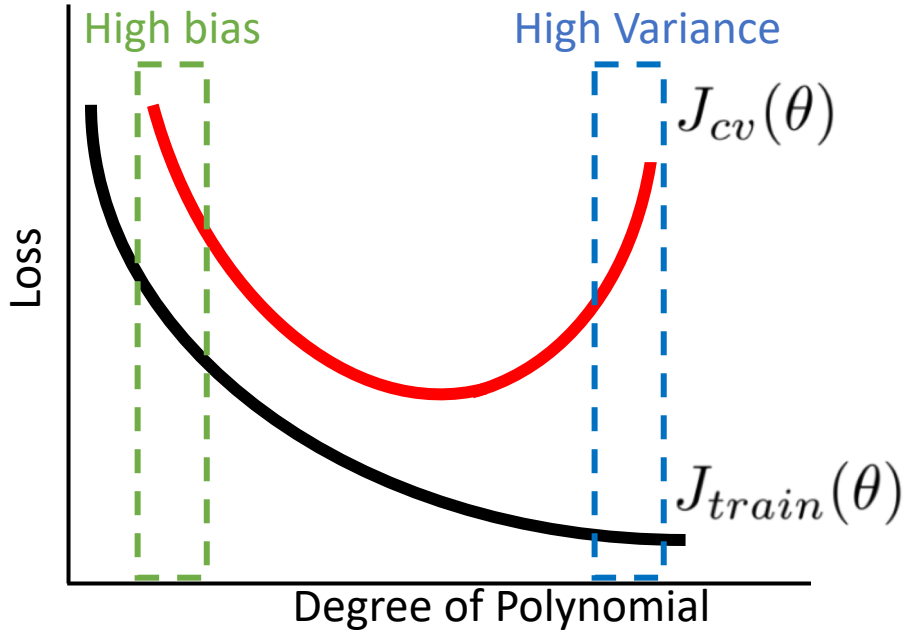
- Training error $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$
- Cross-validation error $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



Source: Andrew Ng

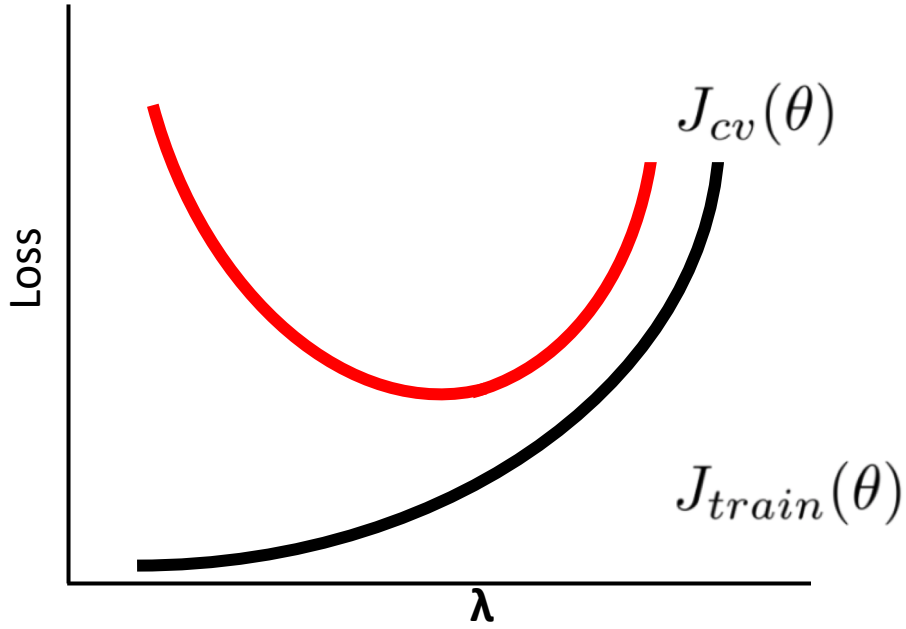
Bias / Variance Trade-off

- Training error $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$
- Cross-validation error $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



Bias / Variance Trade-off with Regularization

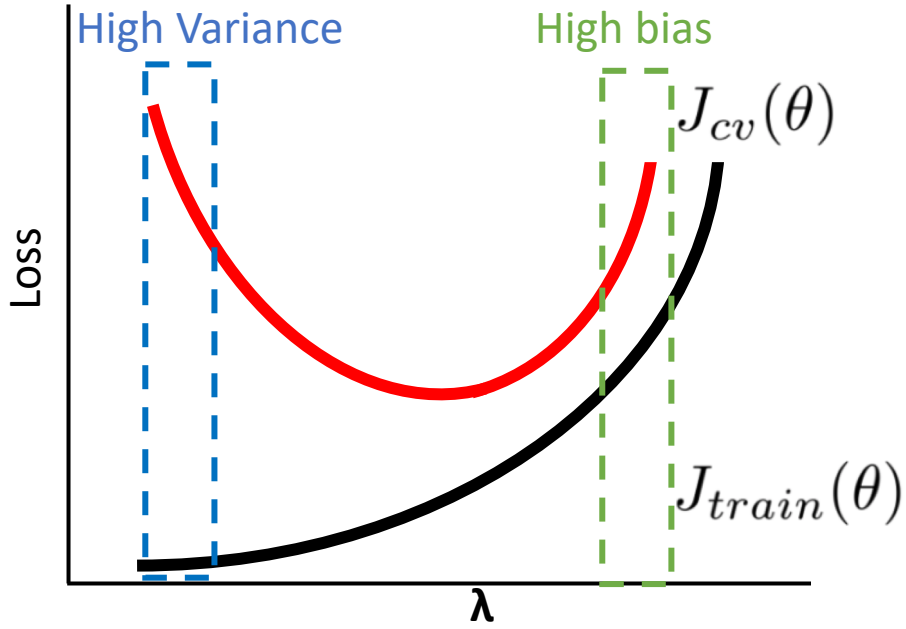
- Training error $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$
- Cross-validation error $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2 + \frac{\lambda}{2m_{cv}}\sum_{j=1}^{m_{cv}}\theta_j^2$



Source: Andrew Ng

Bias / Variance Trade-off with Regularization

- Training error $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$
- Cross-validation error $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2 + \frac{\lambda}{2m_{cv}}\sum_{j=1}^{m_{cv}}\theta_j^2$



Source: Andrew Ng

Problem: Fail to Generalize

- Should we get more data?

Problem: Fail to Generalize

- Should we get more data?

- Getting more data does not always help

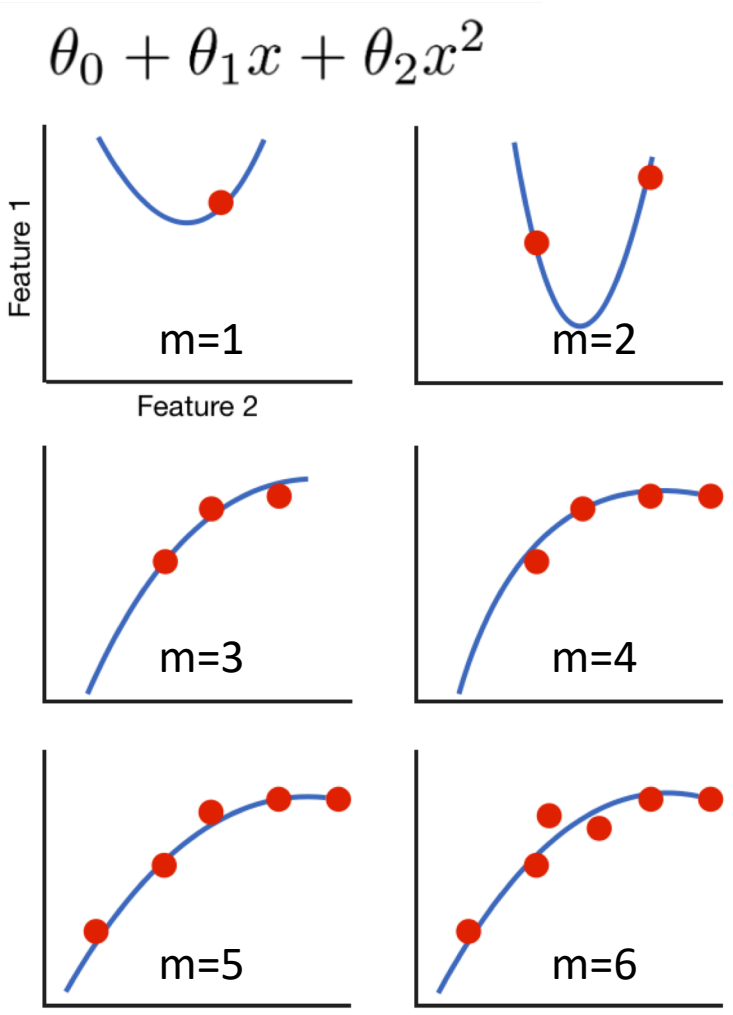
Problem: Fail to Generalize

- Should we get more data?
- Getting more data does not always help
- How do we know if we should collect more data?

Learning Curve

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

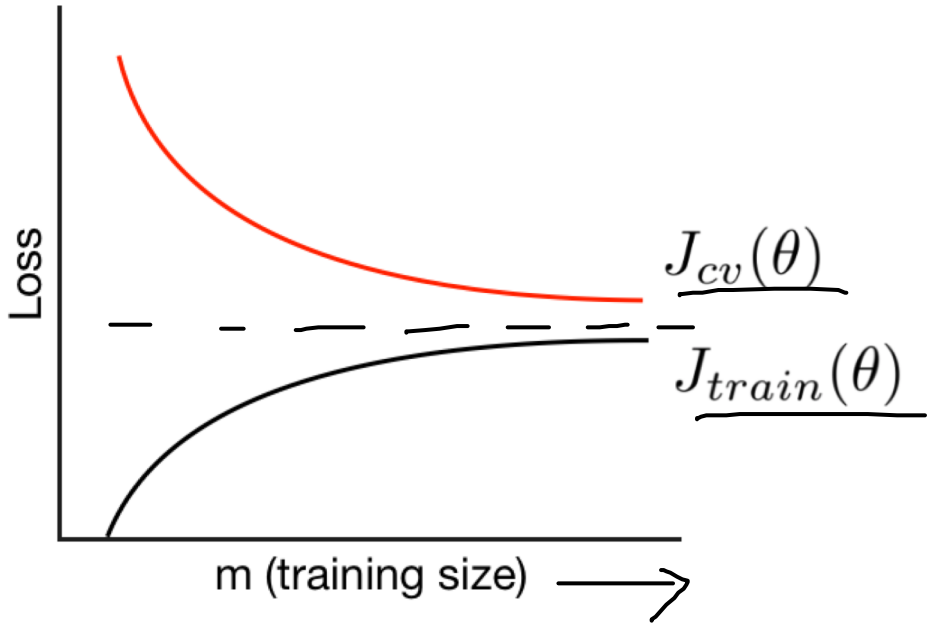
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$



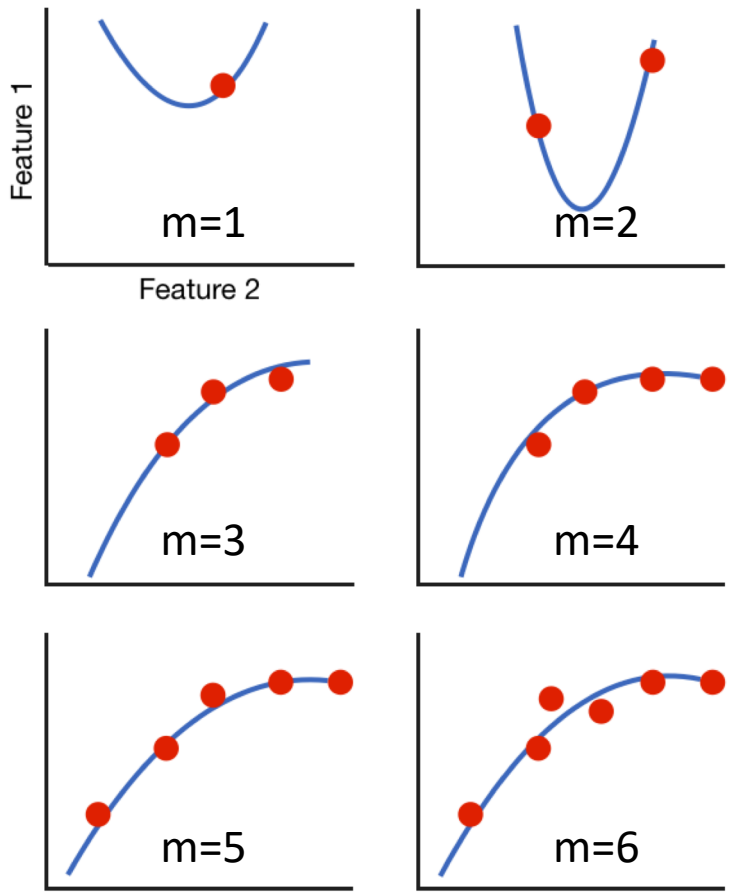
Learning Curve

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

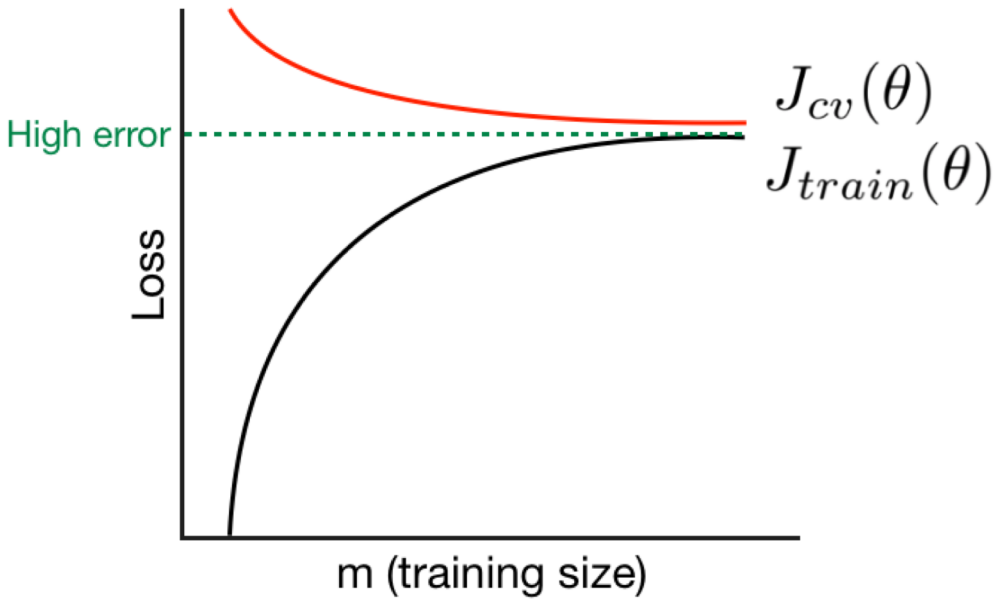
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

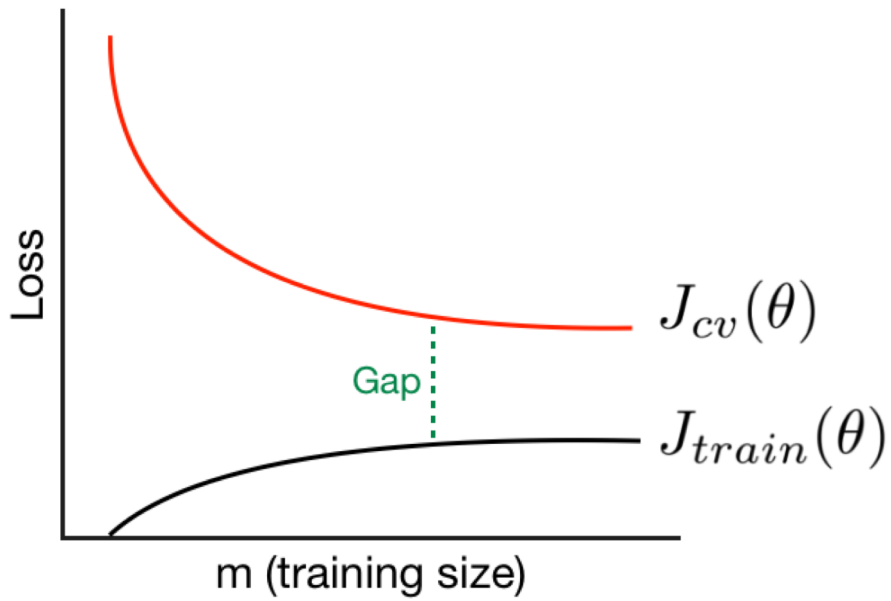


$\theta_0 + \theta_1 x + \theta_2 x^2$

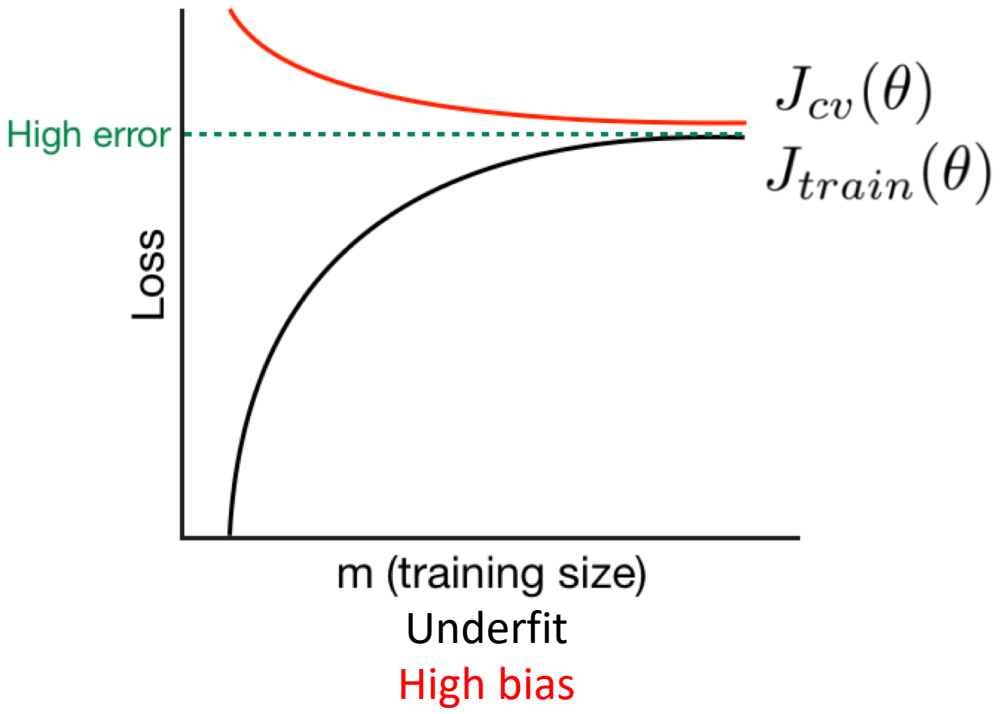


Learning Curve



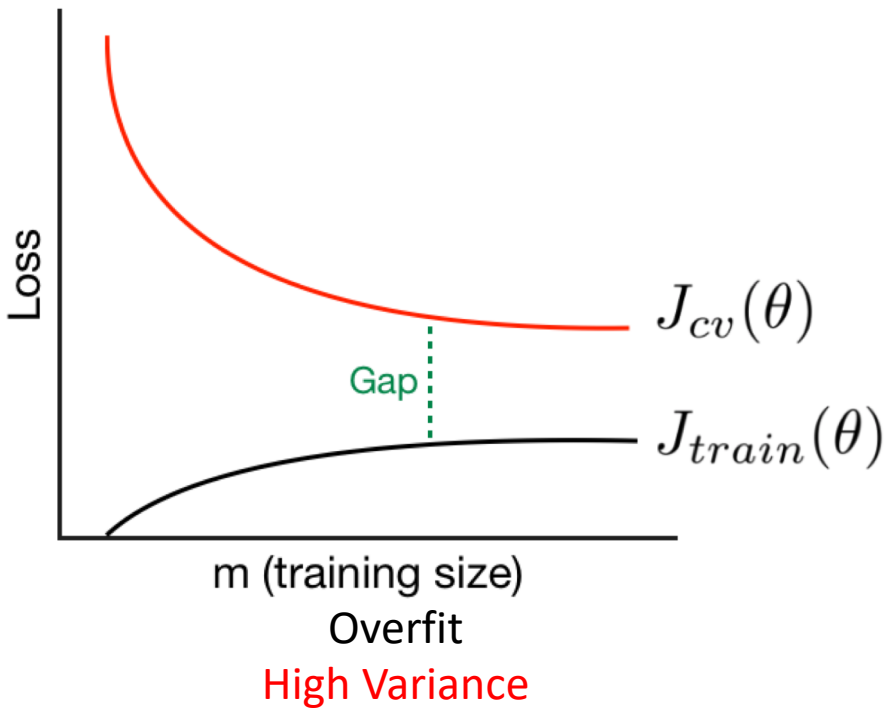


Learning Curve



Underfit

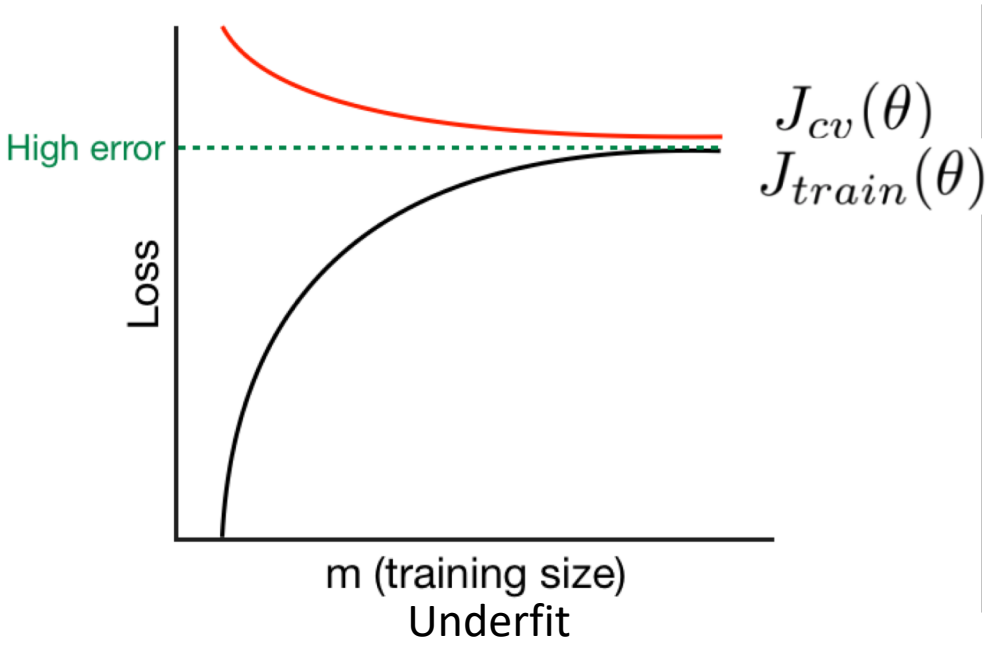
High bias



Overfit

High Variance

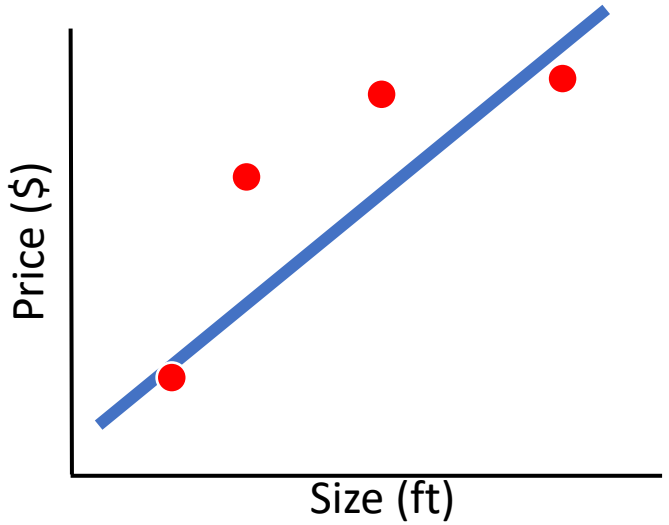
Learning Curve



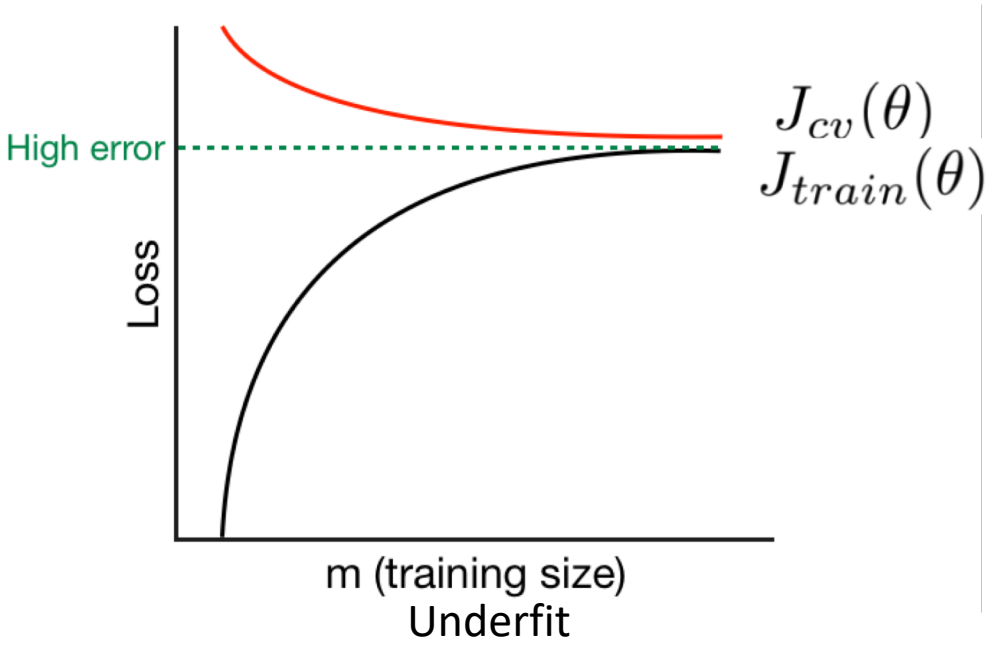
Underfit

High bias

Does adding more data help?



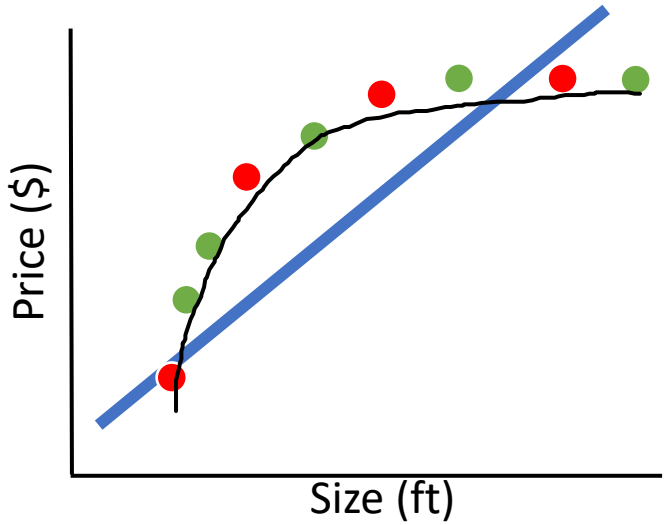
Learning Curve



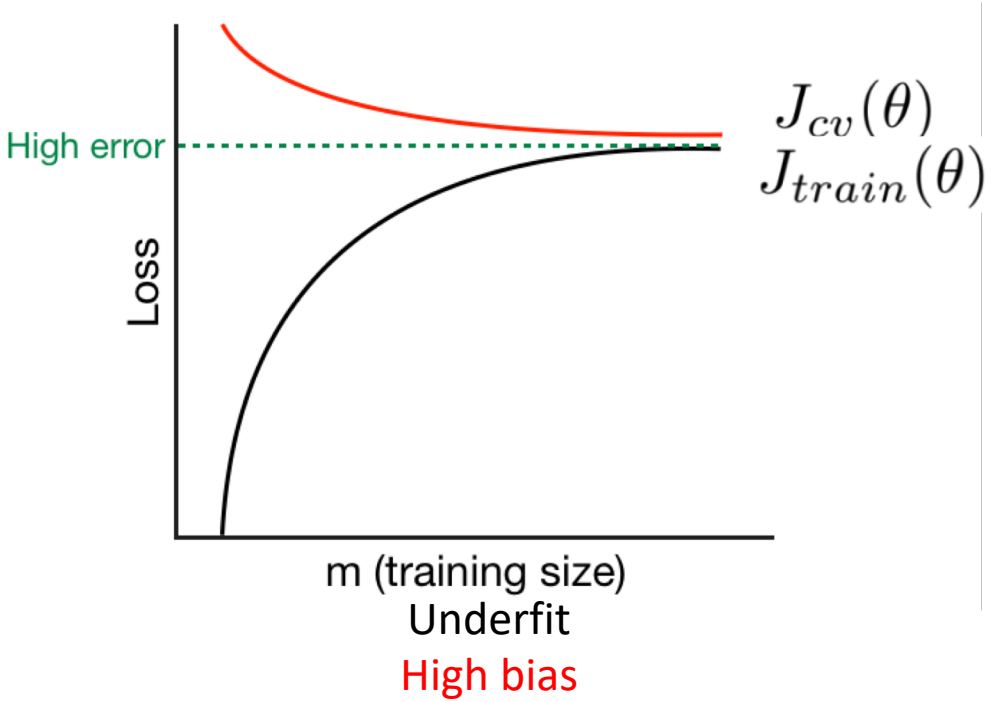
Underfit

High bias

Does adding more data help?



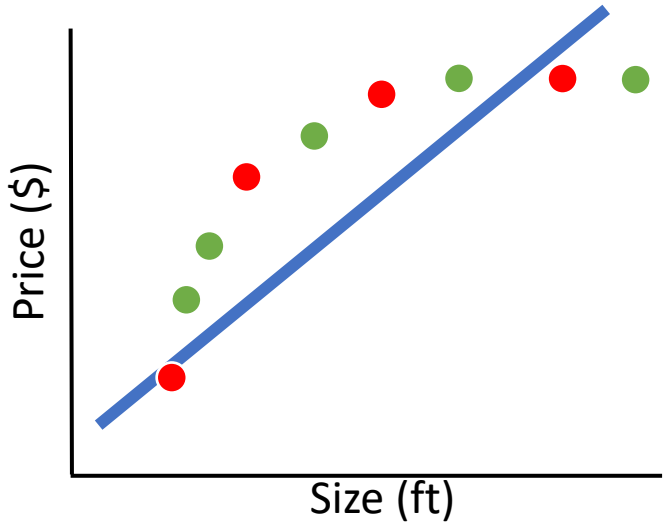
Learning Curve



Underfit

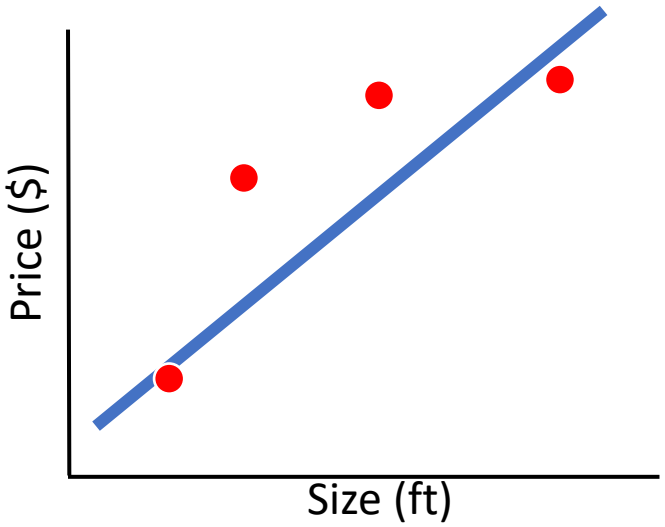
High bias

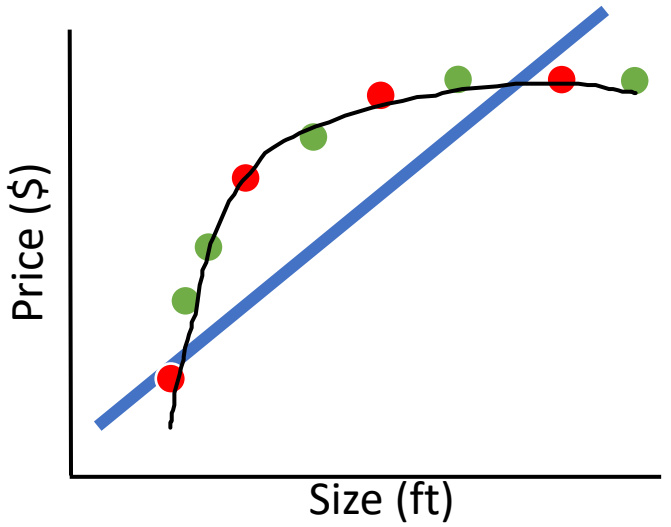
Does adding more data help?



Learning Curve

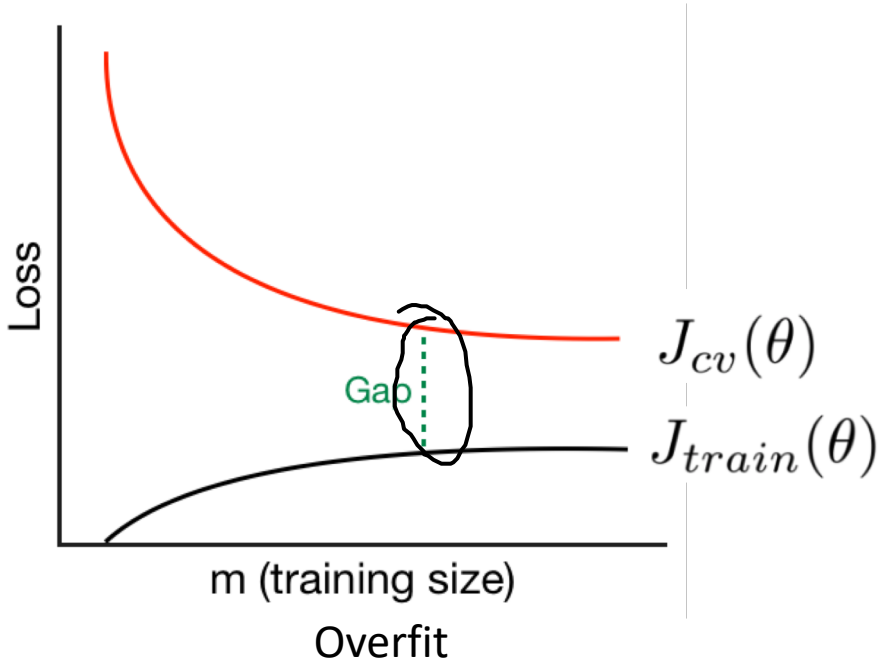
Does adding more data help?





More data doesn't help when your model has high bias

Learning Curve

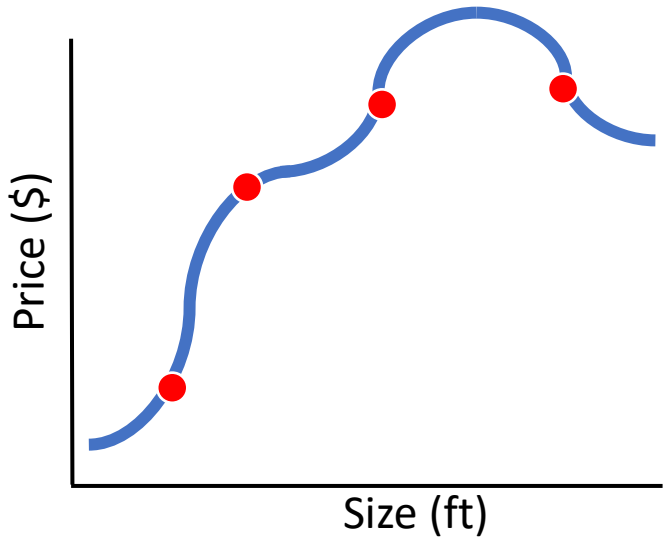


Overfit

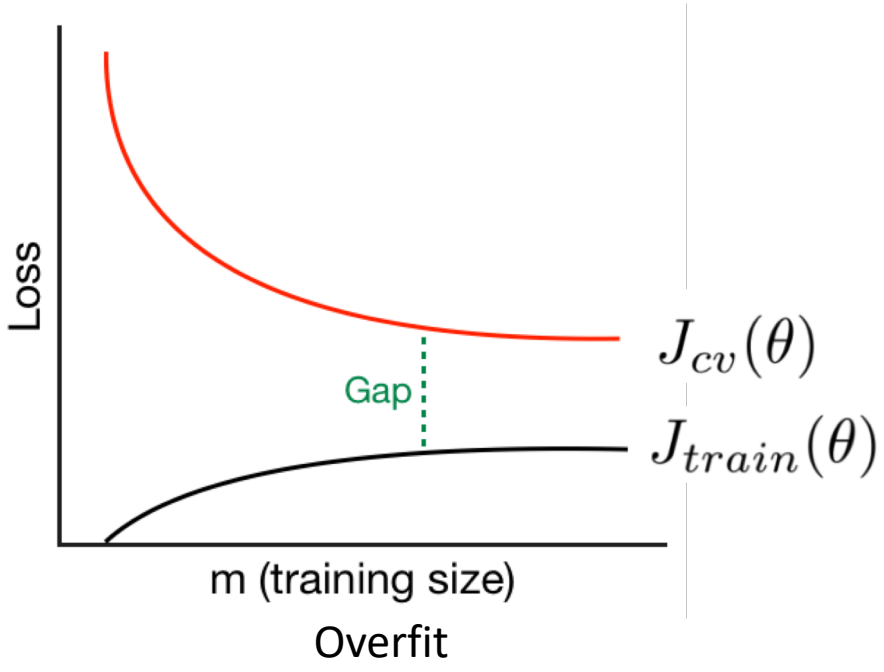
High Variance

Low bias

Does adding more data help?



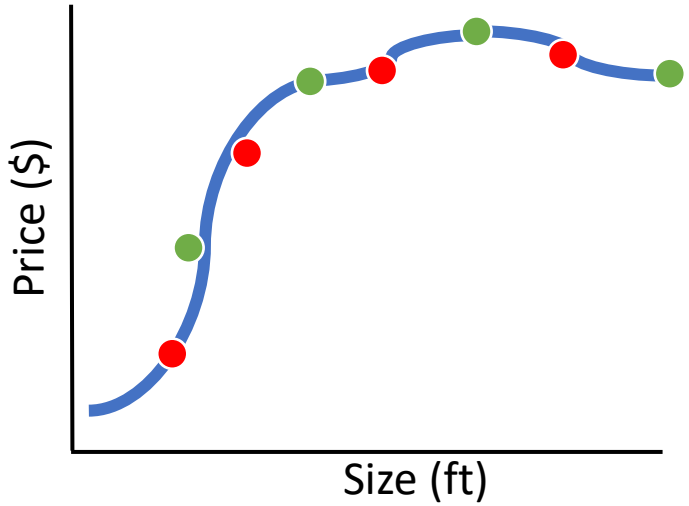
Learning Curve



Overfit

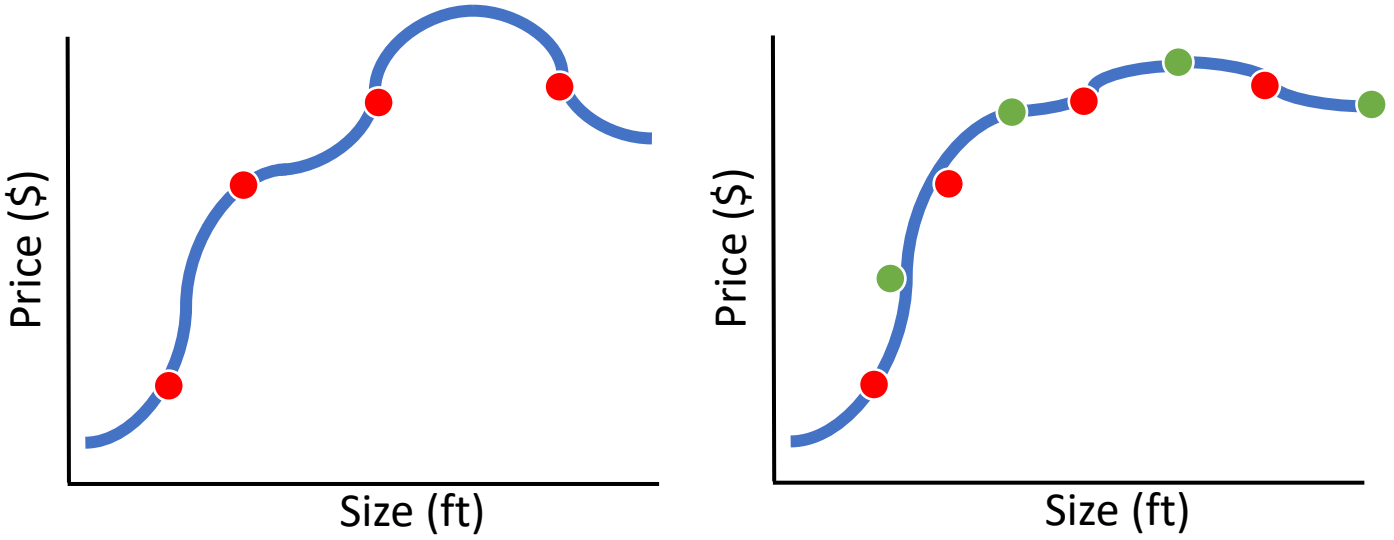
High Variance

Does adding more data help?



Learning Curve

Does adding more data help?



More data is likely to help when your model has high variance

Algorithm	Bias	Variance
Linear Regression	High Bias	Less Variance
Decision Tree	Low Bias	High Variance
Bagging	Low Bias	High Variance (Less than Decision Tree)
Random Forest	Low Bias	High Variance (Less than Decision Tree and Bagging)

Things You Can Try

- Get more data
  - When you have high variance
- Try different features
  - Adding feature helps fix high bias
  - Using smaller sets of feature fix high variance
- Try tuning your hyperparameter
  - Decrease regularization when bias is high
  - Increase regularization when variance is high

Things You Can Try

- Get more data
  - When you have high variance
- Try different features
  - Adding feature helps fix high bias
  - Using smaller sets of feature fix high variance
- Try tuning your hyperparameter
  - Decrease regularization when bias is high
  - Increase regularization when variance is high

Analyze your model before you act

Types of Linear Regression (with Regularization Parameter)

- Polynomial → by changing the degree of the hypothesis
  - $Y = b0 + b1x + b2x^2 + \ldots.$
- Lasso Regression
  - Uses L1 regularization → $\frac{\lambda}{2m}\sum \theta_j$
- Ridge Regression
  - Uses L2 regularization → $\frac{\lambda}{2m}\sum \theta_j^2$
- Elasticnet regression
  - Combines both L1 and L2 regularization