Regression & Regularization

Danna Gurari
University of Texas at Austin
Spring 2021

https://www.ischool.utexas.edu/~dannag/Courses/IntroToMachineLearning/CourseContent.html

Review

- Last week:
  - Machine learning today
  - History of machine learning
  - How does a machine learn?

- Assignments (Canvas)
  - Problem Set 1 due tonight
  - Problem Set 2 due next week
  - Lab Assignment 1 due in two weeks

- Questions?

Today’s Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

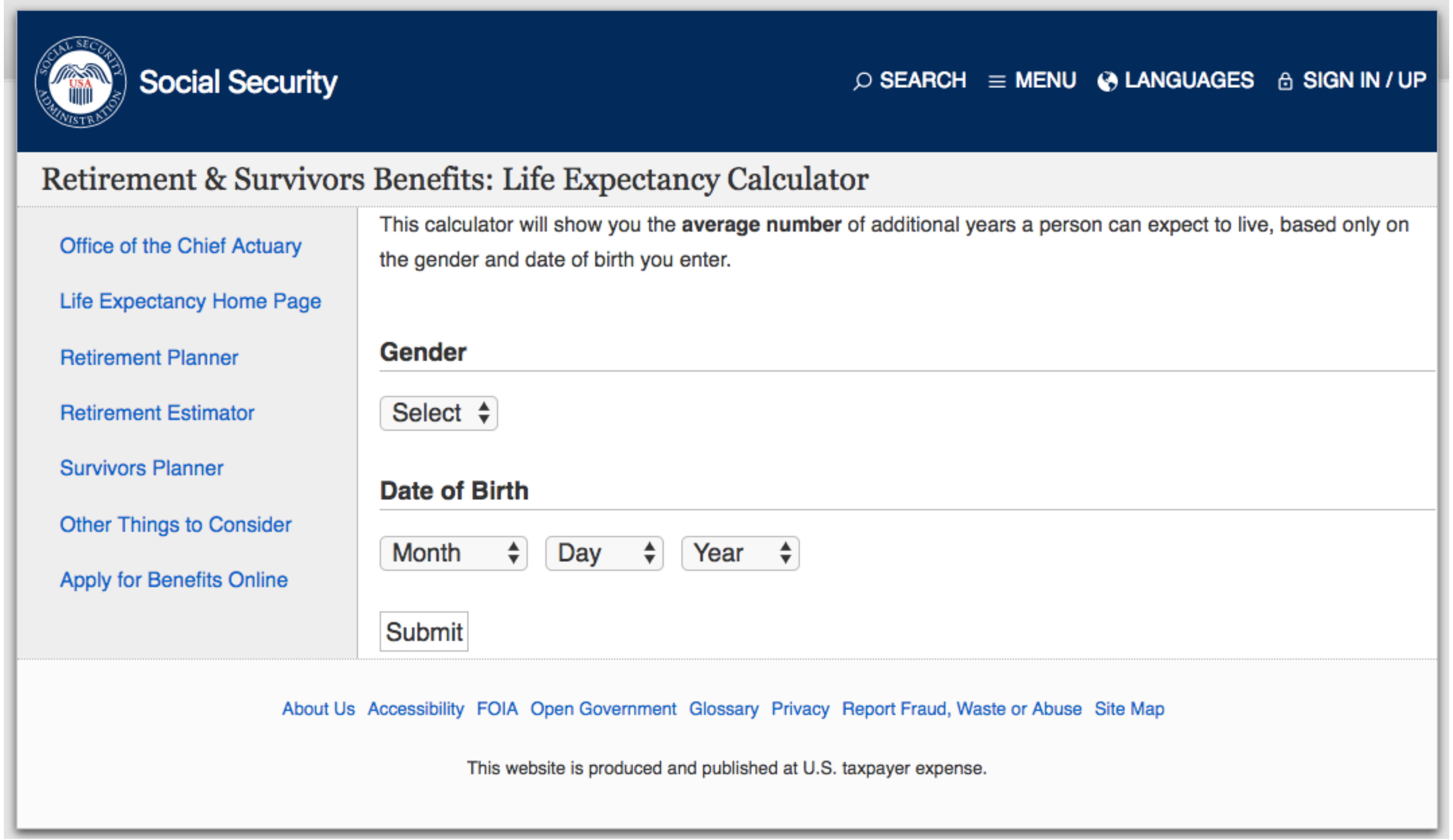
Today’s Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

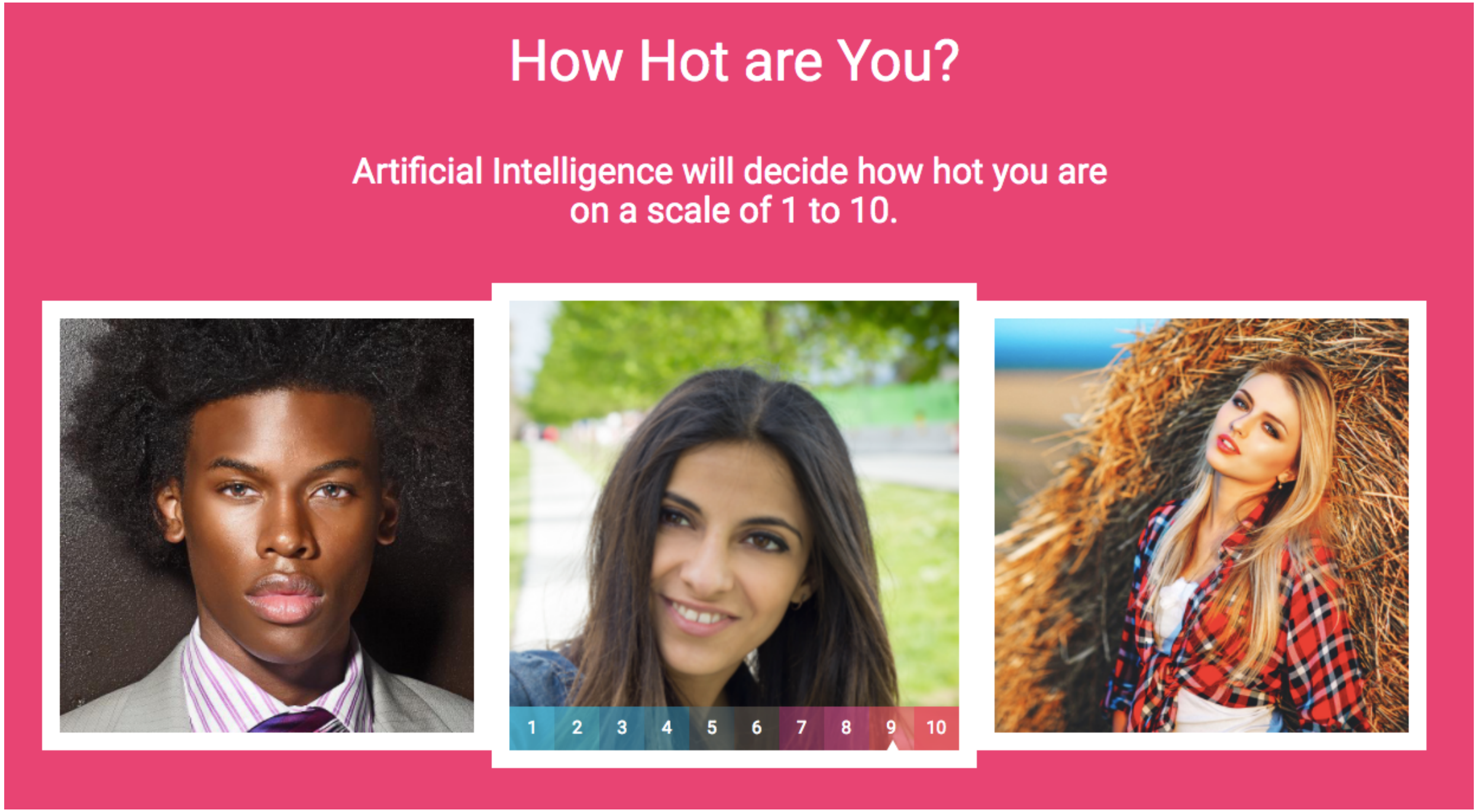
Today's Focus: Regression

Predict **continuous** value

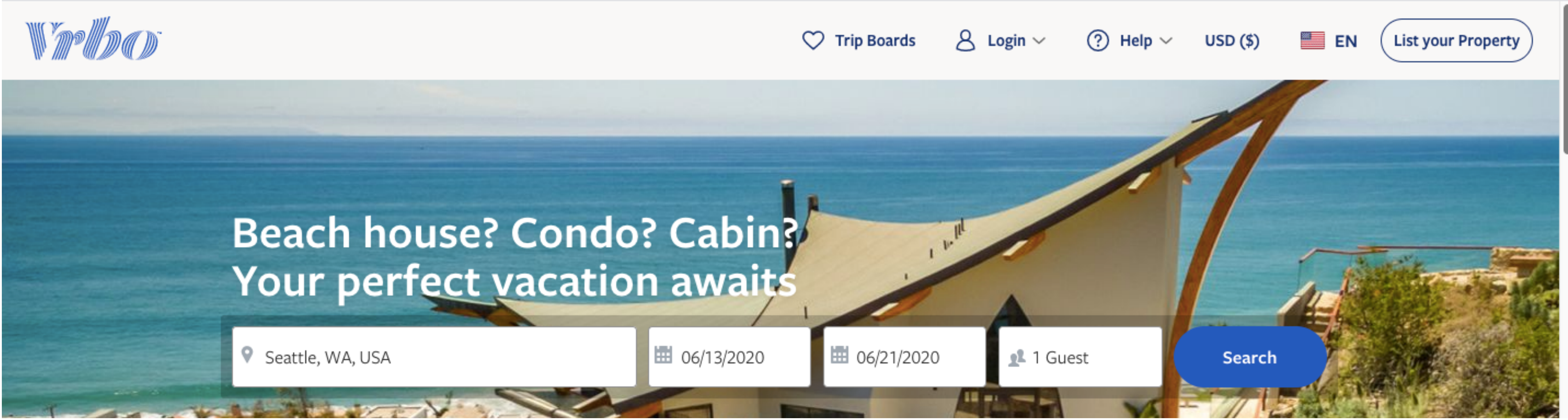
Predict Life Expectancy



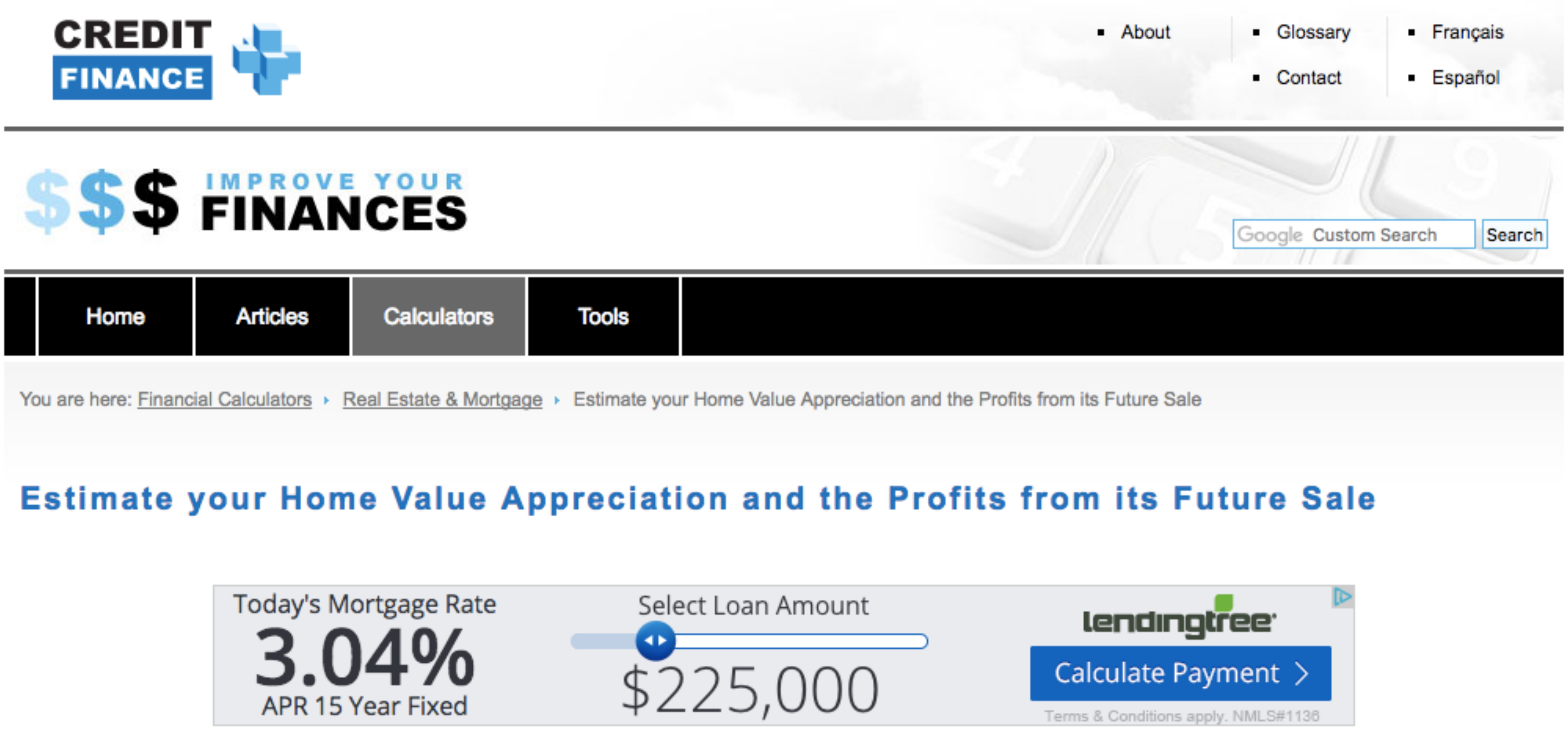
Predict Perceived "Hot"-ness



Predict Price to Charge for Your Home



Predict Future Value of a House You Buy



Predict Future Stock Price

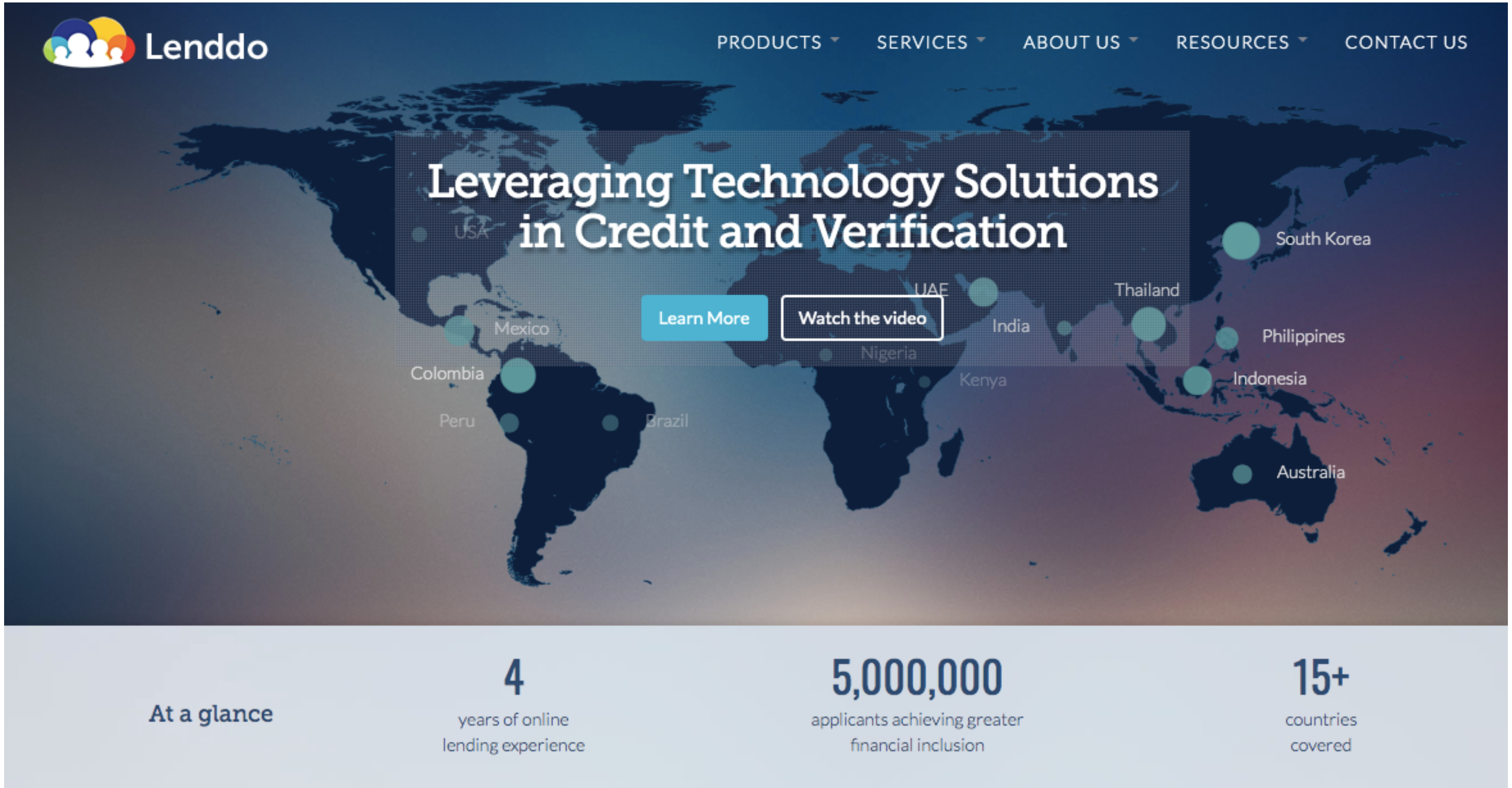
QUANT INSTI GO ALGO Pioneer Institute in Algo Trading

HOME ABOUT US EPAT ™ PLACEMENT RESOURCES WEBINARS BLOG CONTACT US

Home > Blog > Trading Strategies

Machine Learning For Trading – How To Predict Stock Prices Using Regression?

Predict Credit Score for Loan Lenders



Demo: https://www.youtube.com/watch?time_continue=6&v=0bEJO4Twgu4&feature=emb_logo

https://emerj.com/ai-sector-overviews/artificial-intelligence-applications-lending-loan-management/

What Else to Predict?

Insurance Cost

Public Opinion

Popularity of Social Media Posts

Factory Analysis

Call Center Complaints

Class Ratings

Weather

Animal Behavior

Today's Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

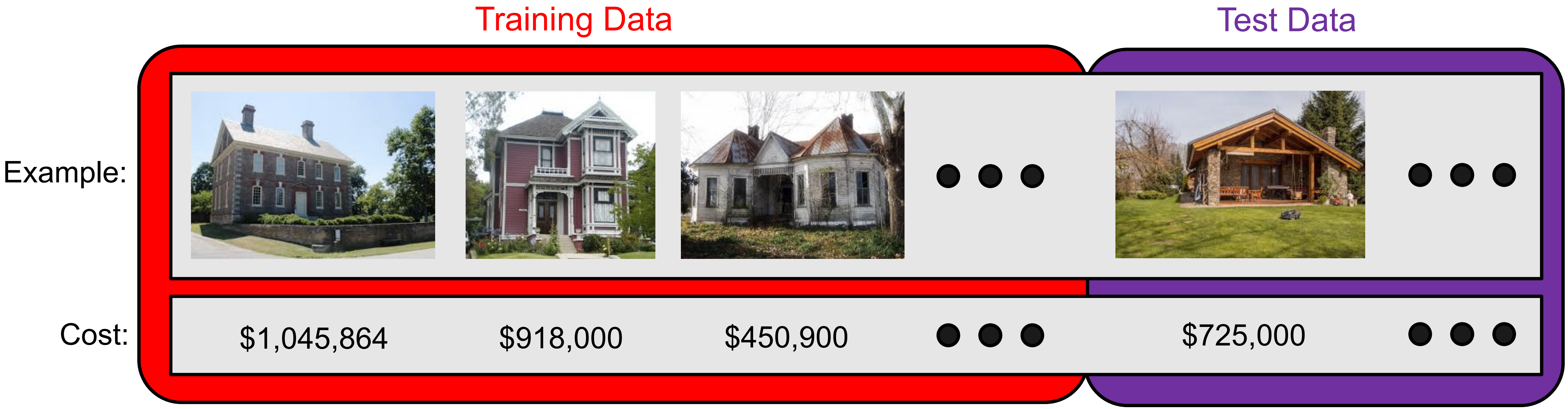
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

Example:    ● ● ●  ● ● ●

Cost: $1,045,864 $918,000 $450,900 ● ● ● $725,000 ● ● ●

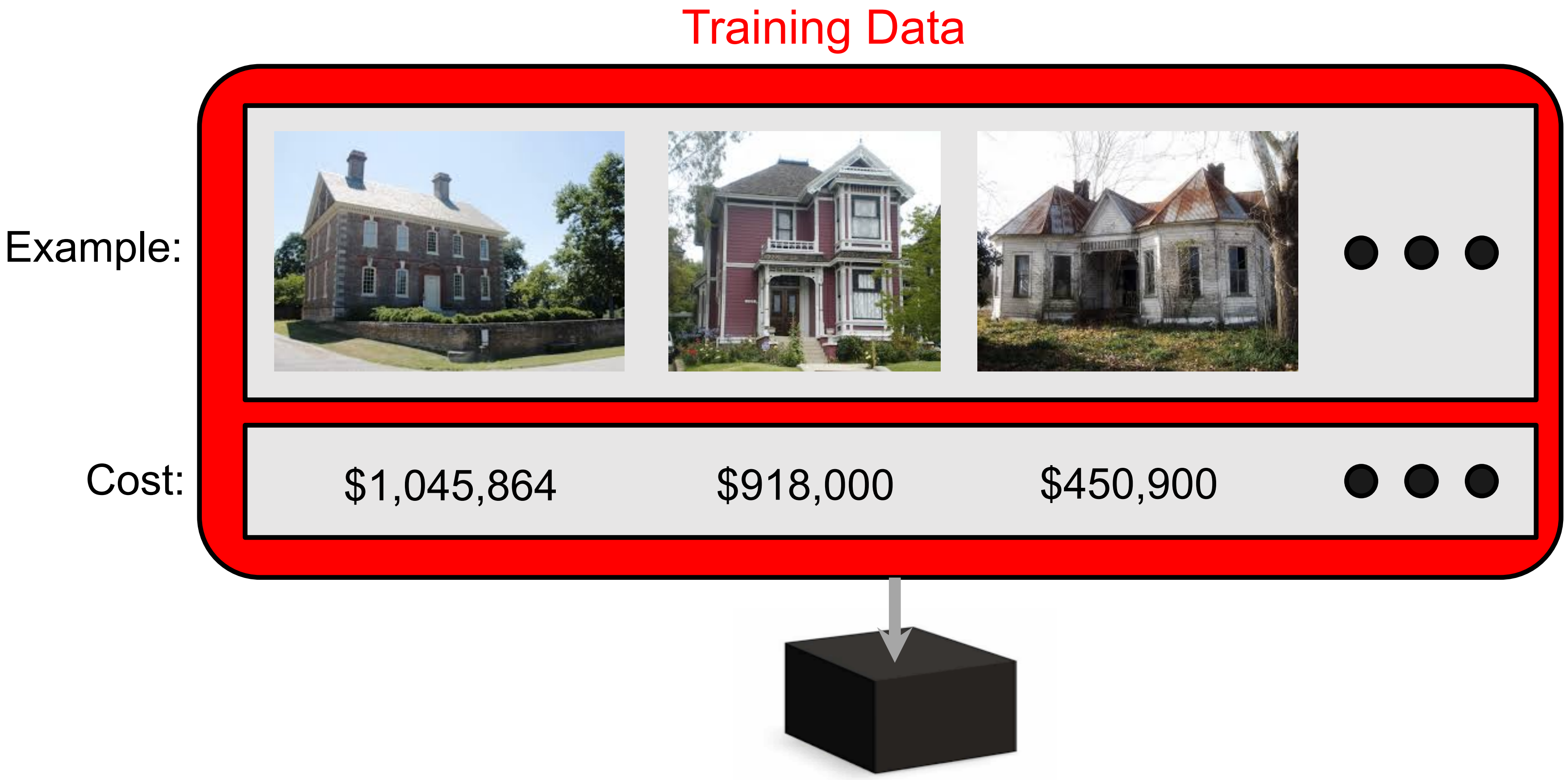
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

1. Split data into a "training set" and "test set"



Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

2. Train model on "training set" to try to minimize prediction error on it



Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error

Prediction Model

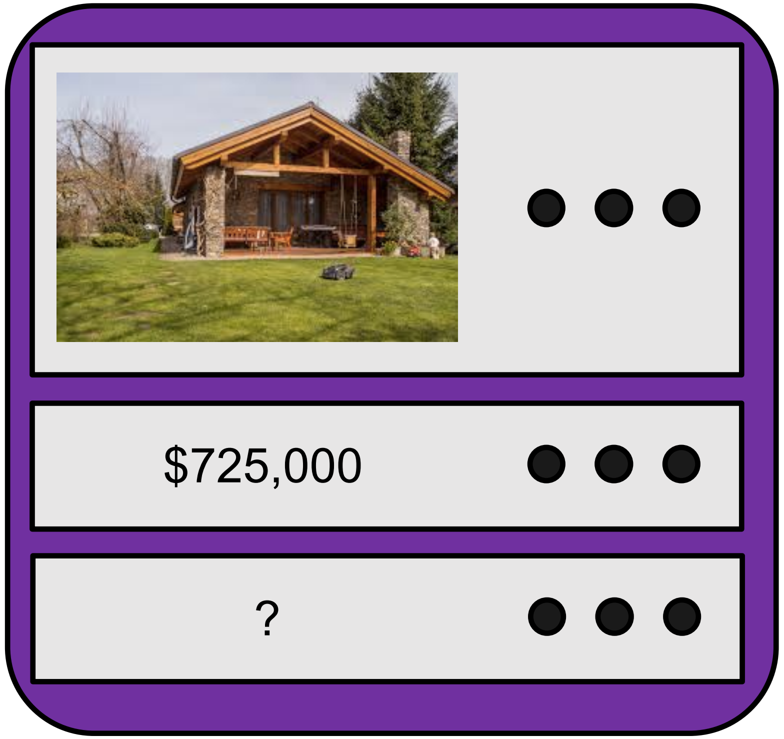
+

Test Data

Example:

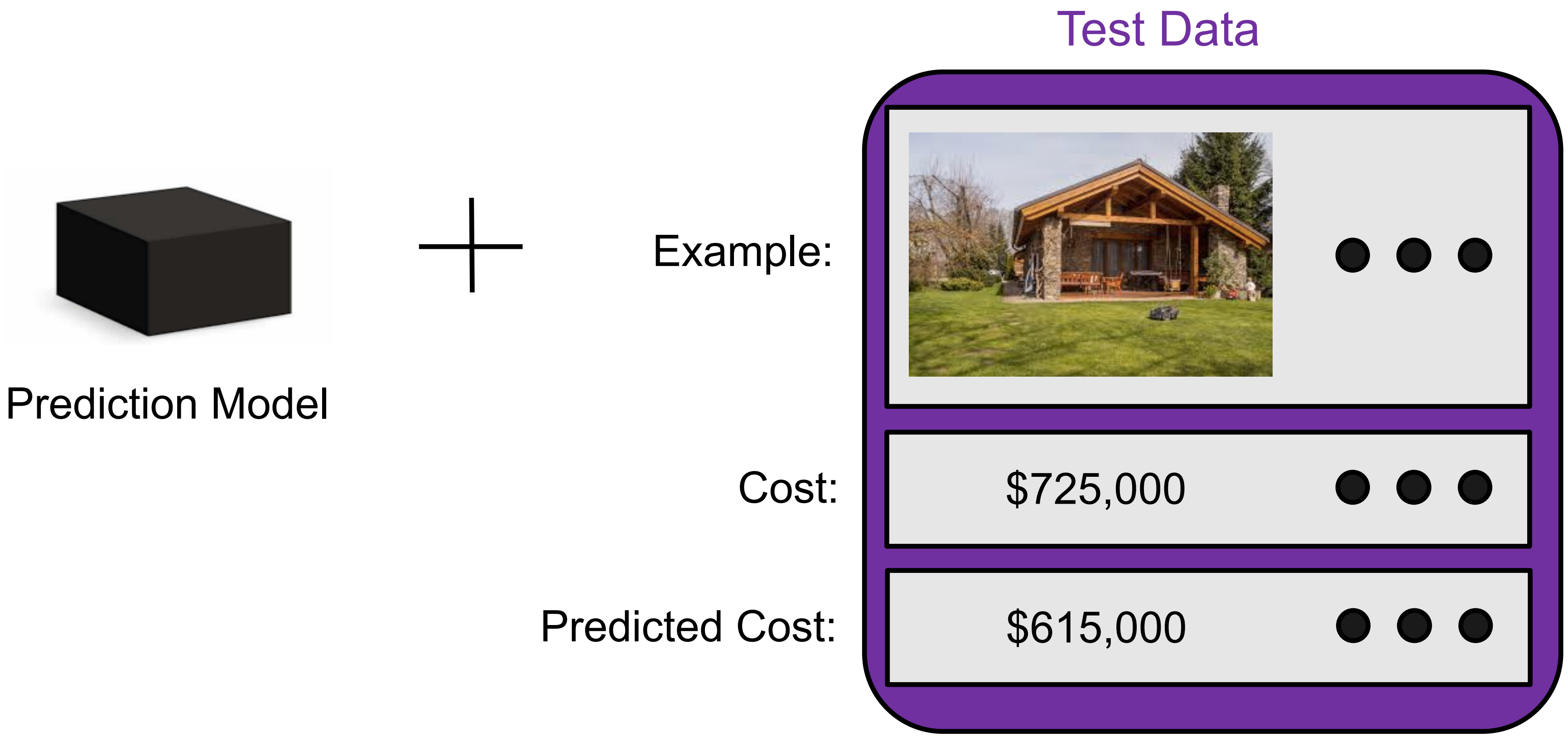
Cost:

Predicted Cost:



Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

3. Apply trained model on "test set" to measure generalization error



Regression Evaluation Metrics

Results: e.g.,

inst#	actual	predicted	error
1	0.18	0.272	0.092
2	0.122	0.434	0.312
3	0.088	0.344	0.256
4	[illegible]	[illegible]	[illegible]
5	0	0.232	0.232
6	[illegible]	[illegible]	[illegible]
7	0.907	0.367	-0.54
8	0.216	0.227	0.011
9	0	0.367	0.367
10	0.048	0.108	0.061
11	0.198	0.145	-0.053
12	[illegible]	[illegible]	[illegible]
13	0.505	0.28	-0.225
14	[illegible]	[illegible]	[illegible]
15	0.12	0.178	0.058
16	0.254	0.235	-0.018

- Mean absolute error
  - What is the range of possible values?
  - Are larger values better or worse?

Regression Evaluation Metrics

Results: e.g.,

inst#	actual	predicted	error
1	0.18	0.272	0.092
2	0.122	0.434	0.312
3	0.088	0.344	0.256
4	0.125	0.238	0.112
5	0	0.232	0.232
6	0	0.092	0.092
7	0.907	0.367	-0.54
8	0.216	0.227	0.011
9	0	0.367	0.367
10	0.048	0.108	0.061
11	0.198	0.145	-0.053
12	0	0.159	0.159
13	0.505	0.28	-0.225
14	0.273	0.097	-0.175
15	0.12	0.178	0.058
16	0.254	0.235	-0.018

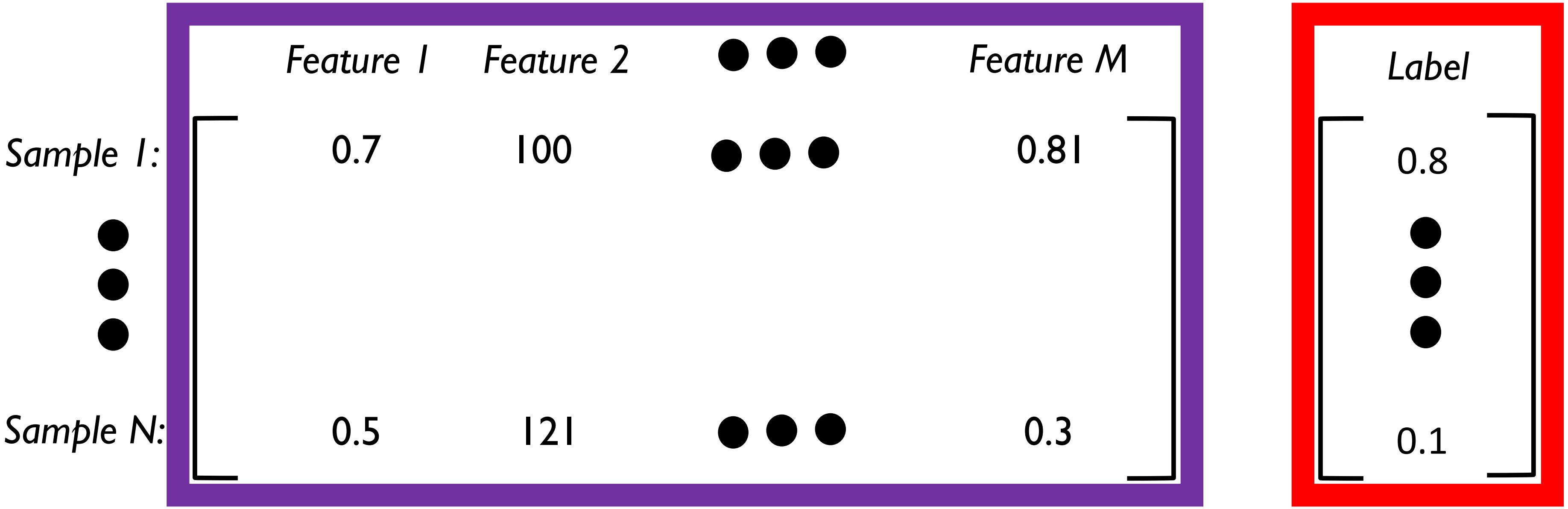
- Mean absolute error
- Mean squared error
  - Why square the errors?

Today's Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

Matrices and Vectors

- **X** : each feature is in its own column and each sample is in its own row
- **y** : each row is the target value for the sample



Matrices and Vectors

- **X** : each feature is in its own column and each sample is in its own row
- **y** : each row is the target value for the sample

$$\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1d} \\ X_{21} & X_{22} & & X_{2j} & & X_{2d} \\ \vdots & & & & & \\ X_{i1} & X_{i2} & & X_{ij} & & X_{id} \\ \vdots & & & & & \\ X_{n1} & X_{n2} & & X_{nj} & & X_{nd} \end{bmatrix} \leftarrow \text{point } X_i^{\top} \qquad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\uparrow \text{ feature column } X_{*j} \qquad \uparrow \; y$$

Vector-Vector Product

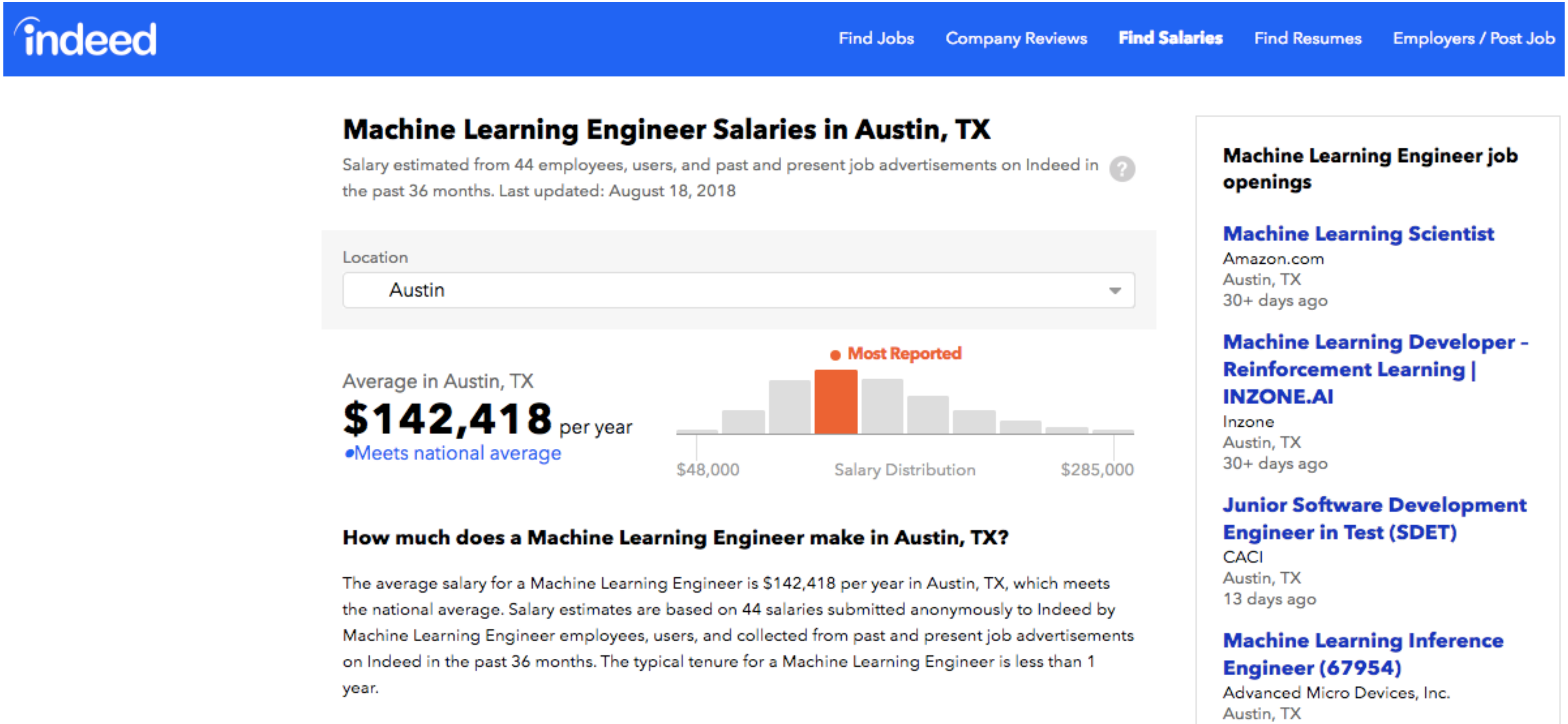
$$\boldsymbol{w}^T\boldsymbol{x} = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = w_1 x_1 + \cdots + w_m x_m$$

e.g.,

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = (1\text{x}4 + 2\text{x}5 + 3\text{x}6) = 32$$

Excellent Review: http://www.cs.cmu.edu/~zkolter/course/15-884/linalg-review.pdf

Class Task: Predict Your Salary If You Become a Machine Learning Engineer



Class Task: Predict Your Salary If You Become a Machine Learning Engineer

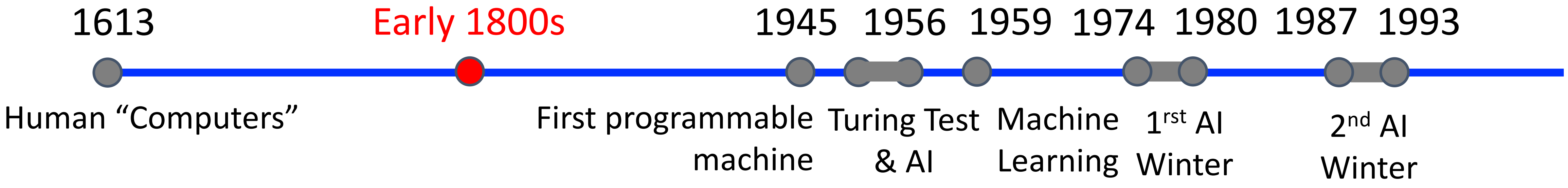
- What features would be predictive of your salary?
- Where can you find data for model training and evaluation (features + true values)?
- What would introduce noise to your data?
- Create a matrix/vector representation of two examples.

Today's Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

Linear Regression: Historical Context

Linear Regression Models
with Least Squares



[Legendre](https://en.wikipedia.org/wiki/Adrien-Marie_Legendre) (1805) and [Gauss](https://en.wikipedia.org/wiki/Carl_Friedrich_Gauss) (1809): https://en.wikipedia.org/wiki/Linear_regression

Linear Regression Model

- General formula:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \ldots + w[p] * x[p] + b$$

Feature vector: **x** = x[0], x[1], …, x[p]

- How many features are there?
  - p+1

Parameter vector to learn: **w** = w[0], w[1], …, w[p]

- How many parameters are there?
  - p+2

Predicted value

“Simple” Linear Regression Model

- Formula:

$$\widehat{y} = w[0] * x[0] + b$$

Feature vector
- How many features are there?
  - 1

Parameter vector to learn
- How many parameters are there?
  - 2

Predicted value

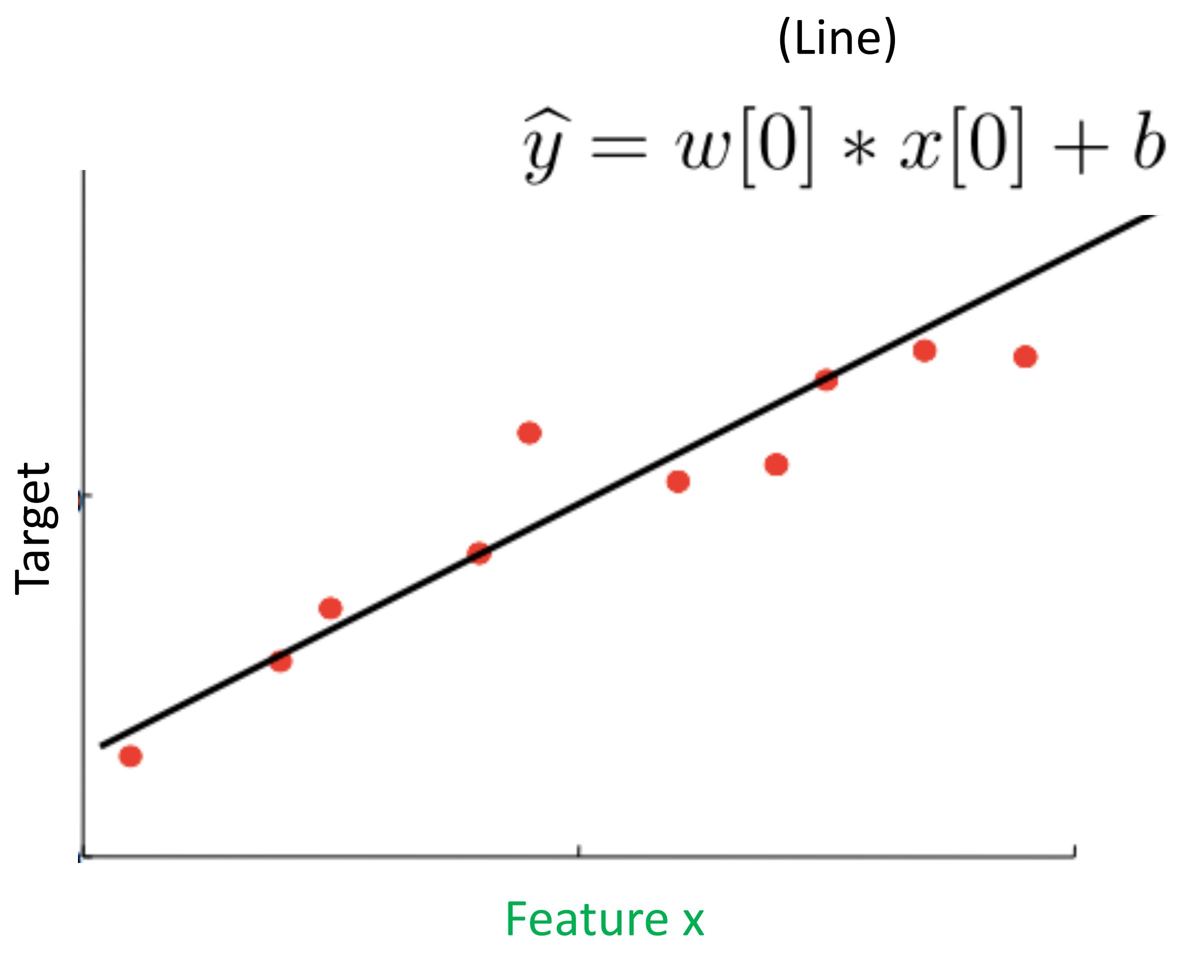


Figure Credit: http://sli.ics.uci.edu/Classes/2015W-273a?action=download&upname=04-linRegress.pdf

“Multiple” Linear Regression Model

- Formula:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + b$$

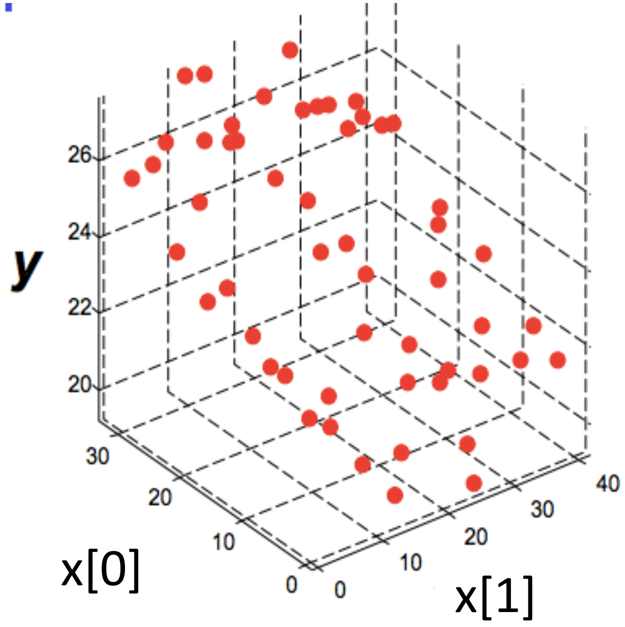
Feature vector

- How many features are there?
  - 2

Parameter vector to learn

- How many parameters are there?
  - 3

Predicted value



(Plane)

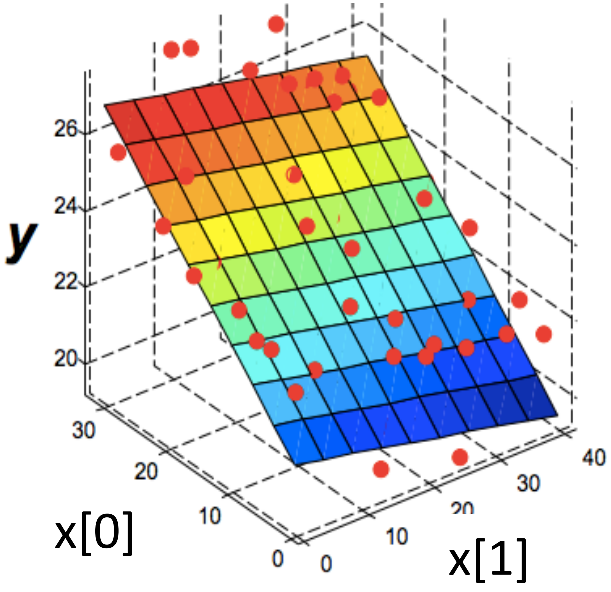


Figure Credit: http://sli.ics.uci.edu/Classes/2015W-273a?action=download&upname=04-linRegress.pdf

Linear Regression Model: What to Learn?

$$\widehat{y} = \boxed{w[0]} * x[0] + \boxed{w[1]} * x[1] + \ldots + \boxed{w[p]} * x[p] + \boxed{b}$$

- Weight coefficients:
  - Indicates how much the predicted value will vary when that feature varies while holding all the other features constant

Linear Regression Model: Learning Parameters

1. Split data into a "training set" and "test set"
2. Train model on "training set" to learn parameters
3. Evaluate model on "test set" to measure generalization error

	Feature 1	Feature 2	● ● ●	Feature M	Label
Sample 1:	0.7	100	● ● ●	0.81	0.9
⋮					⋮
Sample N:	0.5	121	● ● ●	0.3	0.4

Linear Regression Model: Learning Parameters

- Least squares: *minimize* total squared error ("residual") on "training set"
  - Why square the error?

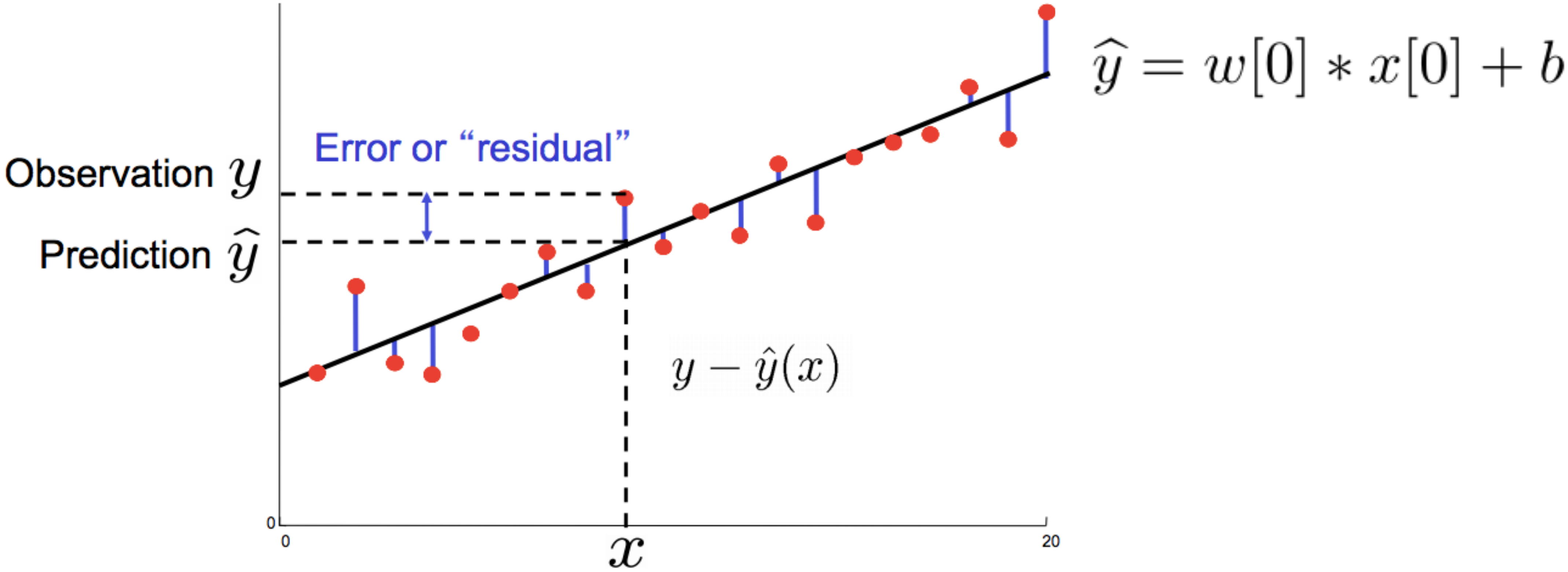


Figure Credit: http://sli.ics.uci.edu/Classes/2015W-273a?action=download&upname=04-linRegress.pdf

Linear Regression Model: Learning Parameters

- Least squares: *minimize* total squared error ("residual") on "training set"

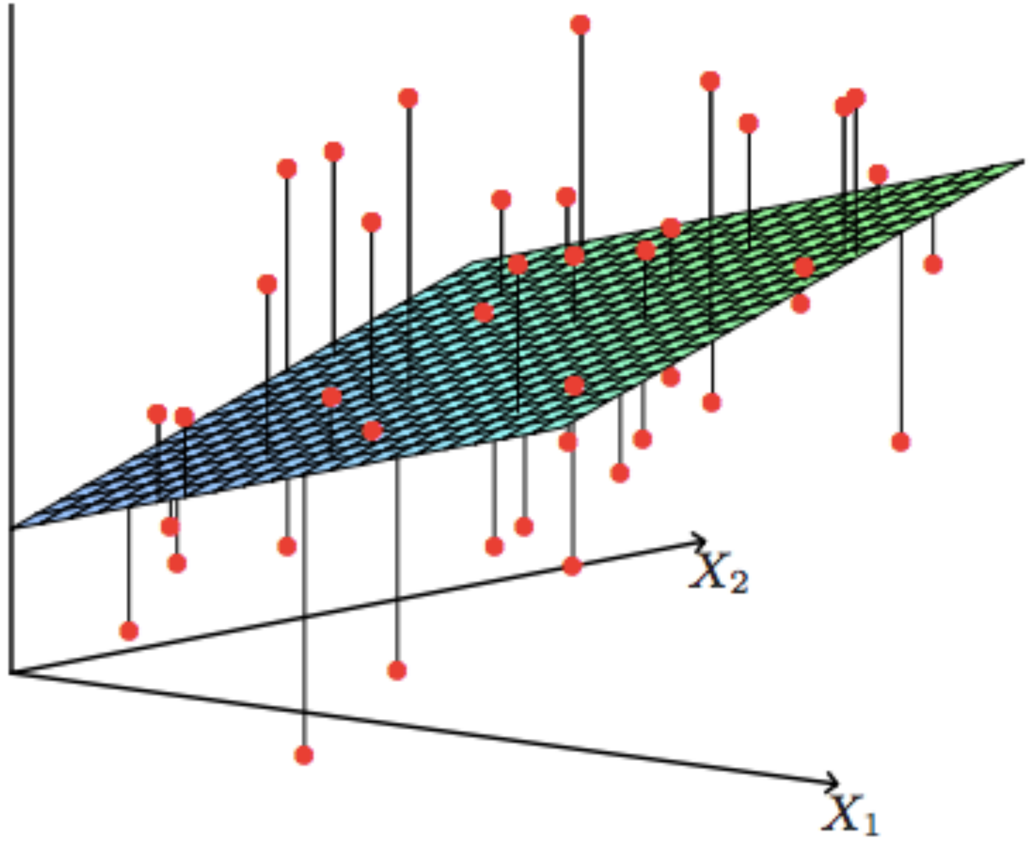


Figure Source: https://web.stanford.edu/~hastie/Papers/ESLII.pdf

Linear Regression Model: Learning Parameters

- Least squares: *minimize* total squared error ("residual") on "training set"
  - Take derivatives, set to zero, and solve for parameters

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

$$2\sum_i x_i(y_i - wx_i) = 0 \Rightarrow$$

$$\sum_i x_i y_i = \sum_i wx_i^2 \Rightarrow$$

$$w = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$$

Great tutorial: http://www.cns.nyu.edu/~eero/NOTES/leastSquares.pdf

Linear Regression Model: Learning Parameters

- Great interactive demo:

https://www.nctm.org/Classroom-Resources/Illuminations/Interactives/Line-of-Best-Fit/

Linear Regression Model: Learning Parameters

- Least squares: *minimize* total squared error ("residual") on "training set"
  - What would be the impact of outliers in the training data?

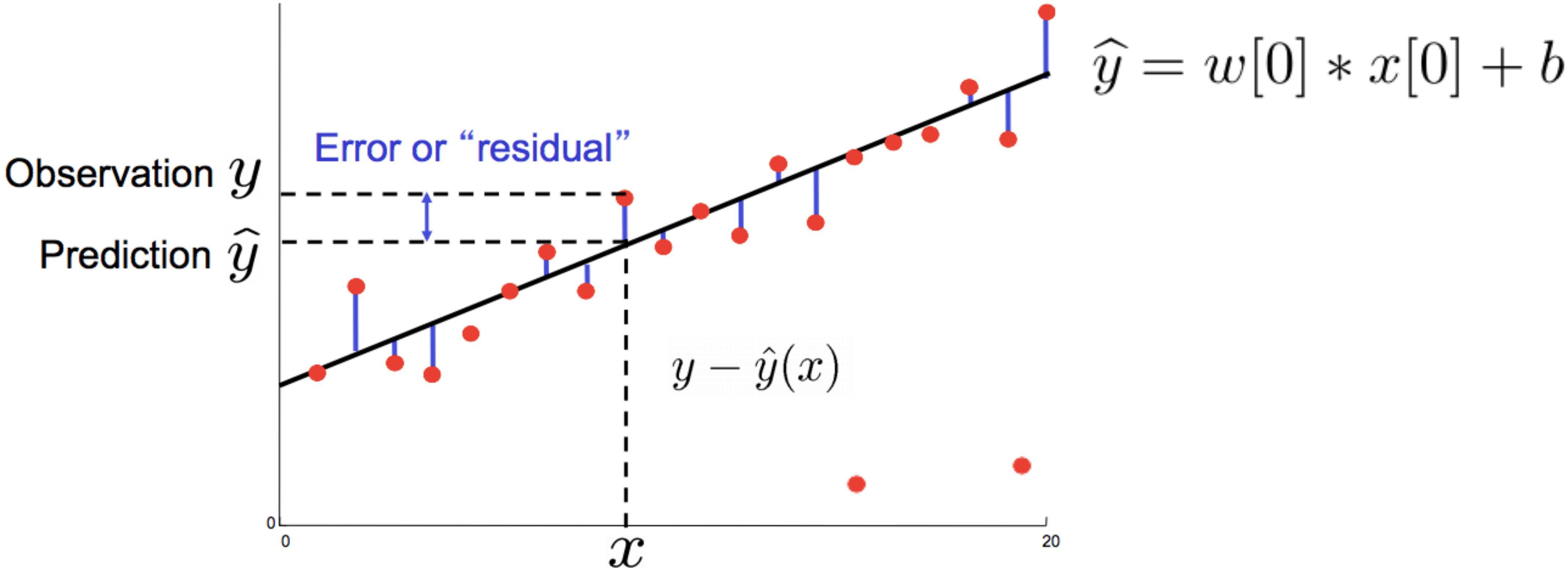


Figure Credit: http://sli.ics.uci.edu/Classes/2015W-273a?action=download&upname=04-linRegress.pdf

Linear Regression: Predict Salary of ML Engineer

(Solution is a hyperplane)

$$\widehat{y} = \boxed{w[0]} * x[0] + \boxed{w[1]} * x[1] + ... + \boxed{w[p]} * x[p] + \boxed{b}$$

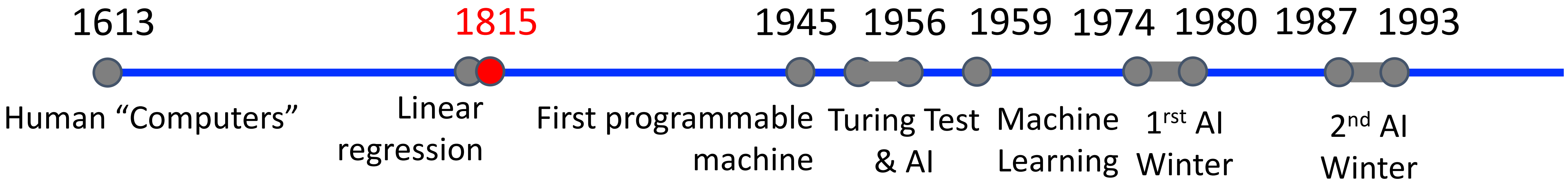
- How would you write the linear model equation?
- How is the weight of different predictive cues learned?

Today’s Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

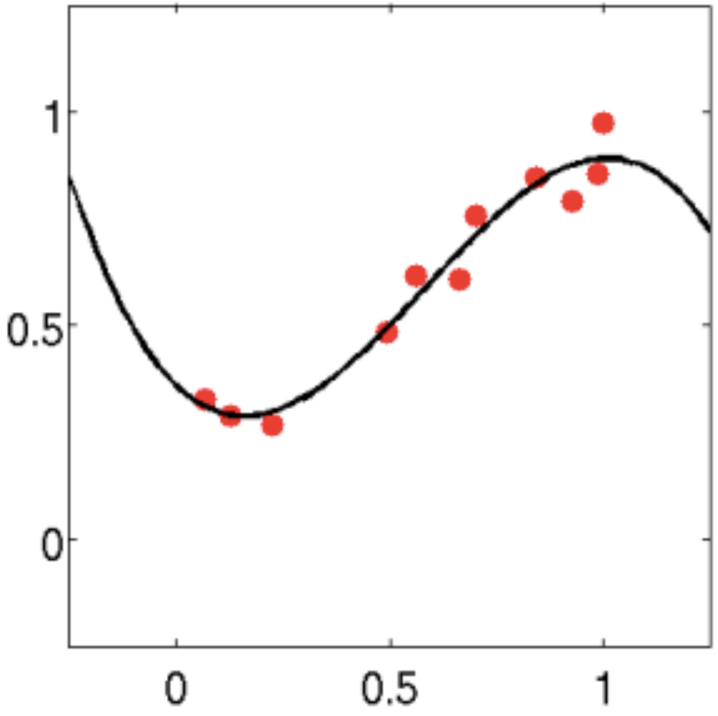
Linear Regression: Historical Context

Polynomial Regression
Models with Least Squares

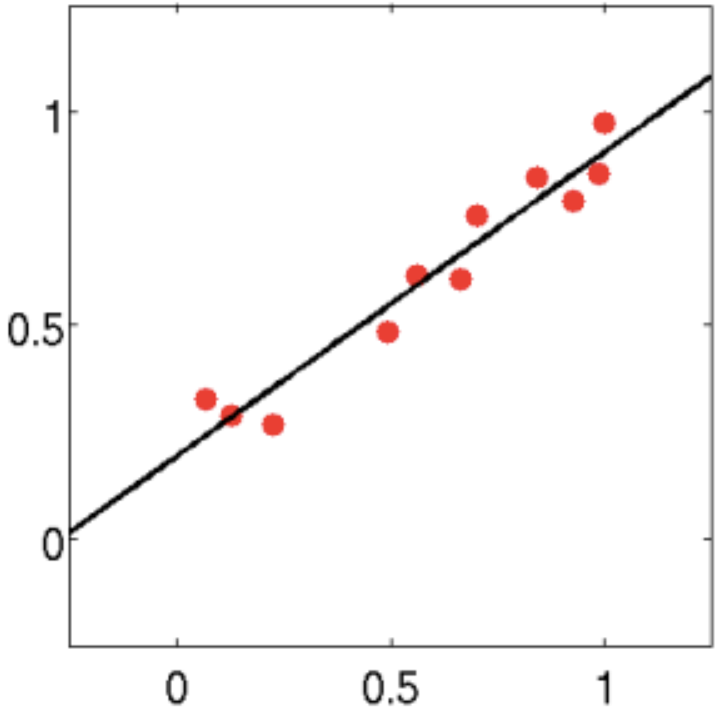


Gergonne, J. D. (November 1974) [1815]. "The application of the method of least squares to the interpolation of sequences". *Historia Mathematica* (Translated by Ralph St. John and S. M. Stigler from the 1815 French ed.). **1** (4): 439–447.

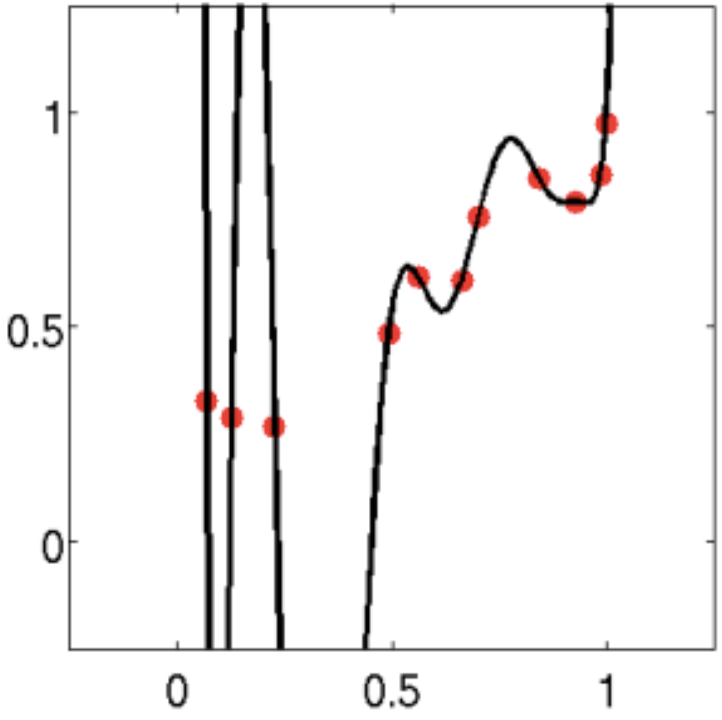
Linear Models: When They Are Not Good Enough, Increase Representational Capacity



polynomial equations
(higher capacity)



linear equations
(lowest capacity)



polynomial equations
(highest capacity)

Polynomial Regression: Transform Features to Model Non-Linear Relationships

- e.g., (Recall) Formula:

$$\widehat{y} = w[0] * x[0] + w[1] * x[1] + b$$

- e.g., New Formula:

$$\widehat{y} = w[0] * x[0] + w[1] * x[0]^2 + b$$

Predicted value

Parameter vector

Feature vector

  - **Still a linear model!**
  - **But can now model more complex relationships!!**

Polynomial Regression: Transform Features to Model Non-Linear Relationships

- e.g., feature conversion for polynomial degree 3

$$D = \{(x^{(j)}, y^{(j)})\} \longrightarrow D = \{([x^{(j)}, (x^{(j)})^2, (x^{(j)})^3], y^{(j)})\}$$

- e.g., What is the new feature vector with polynomial degree up to 3?

Example 1:	2
Example 2:	3
Example 3:	4

→

Example 1:	2	4	8
Example 2:	3	9	27
Example 3:	4	16	64

Polynomial Regression: Transform Features to Model Non-Linear Relationships

- General idea: **project data into a higher dimension** to fit more complicated relationships to a linear fit

- How to **project data into a higher dimension?**

e.g.,

Polynomial: $\phi_j(x) = x^j$ for j=0 … n

Gaussian: $$\phi_j(x) = \frac{(x-\mu_j)}{2\sigma_j^2}$$

Sigmoid: $$\phi_j(x) = \frac{1}{1+\exp(-s_j x)}$$

Polynomial Regression Model: Learning Parameters

- M-th order polynomial function: $y(x, \mathbf{w}) = w_0 + \sum_{j=1}^{M} w_j x^j$
- Still linear model, so can learn with same approach as for linear regression

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

$$2\sum_i x_i(y_i - wx_i) = 0 \Rightarrow$$

$$\sum_i x_i y_i = \sum_i w x_i^2 \Rightarrow$$

$$w = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$$

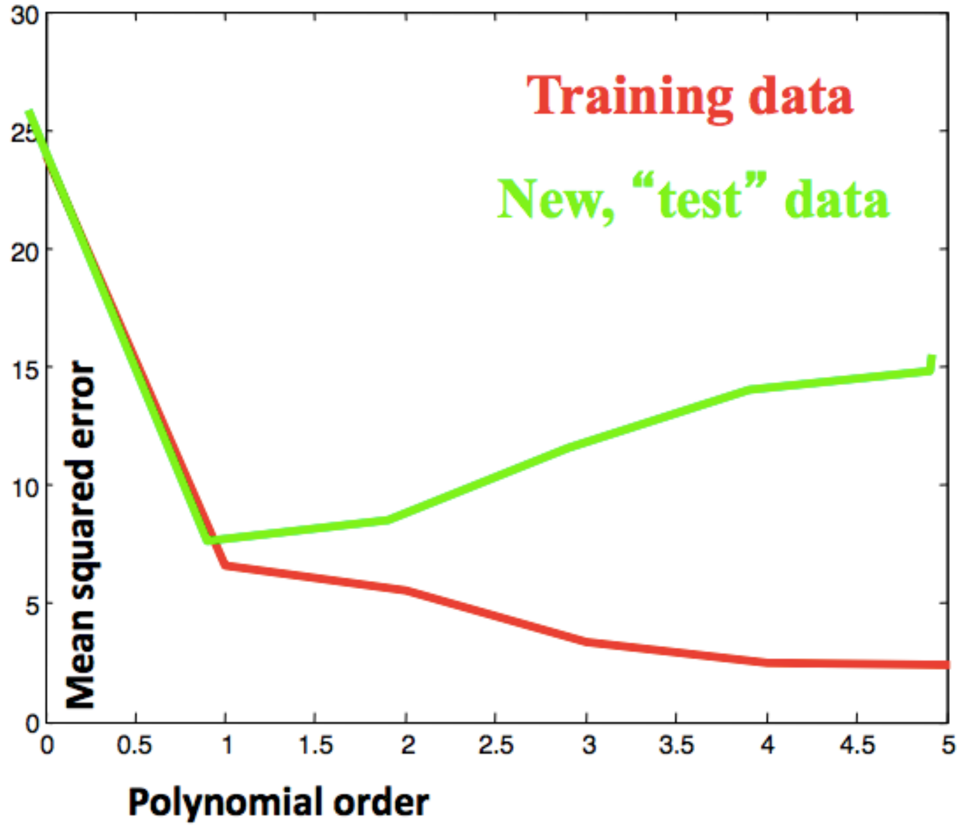
Polynomial Regression Model: Learning Parameters

- Great interactive demo:

  https://arachnoid.com/polysolve/

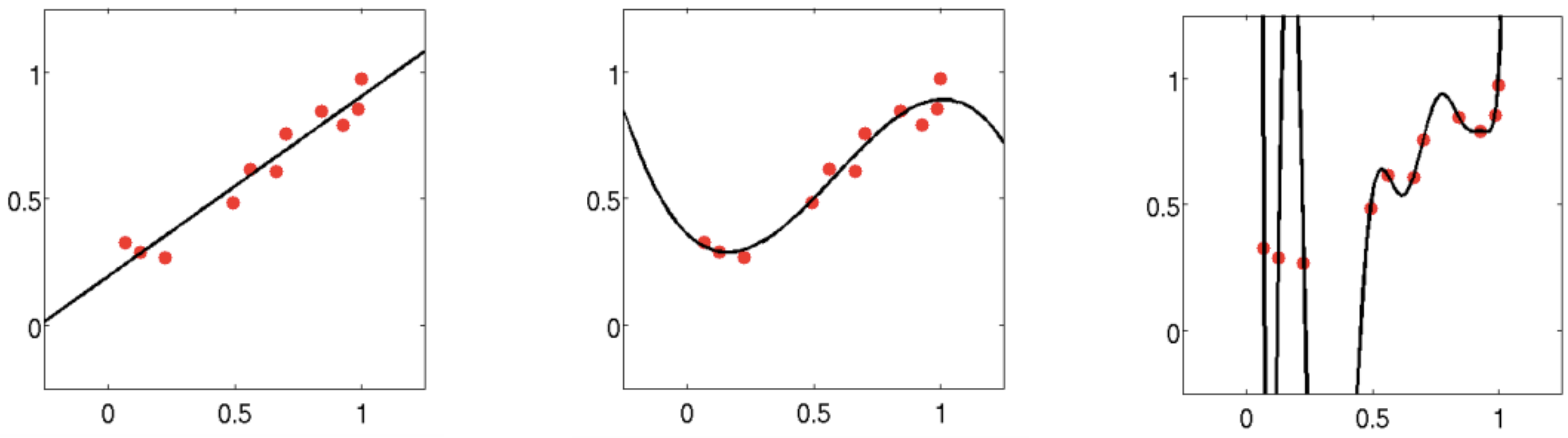
Polynomial Regression Model: What Feature Transformation to Use?

- Plot of error for different polynomial orders:
  - What happens to training data error with larger polynomial order?
    - Error shrinks
  - What happens to test data error with larger polynomial order?
    - Error shrinks and then grows
  - Why does train error ***shrink*** and test error ***grow***?
    - The higher the polynomial order the greater the model "overfits" to the training data ***since it can model noise***! Models capturing noise generalize poorly to new test data
  - What polynomial order should you use?



How to Avoid Overfitting?

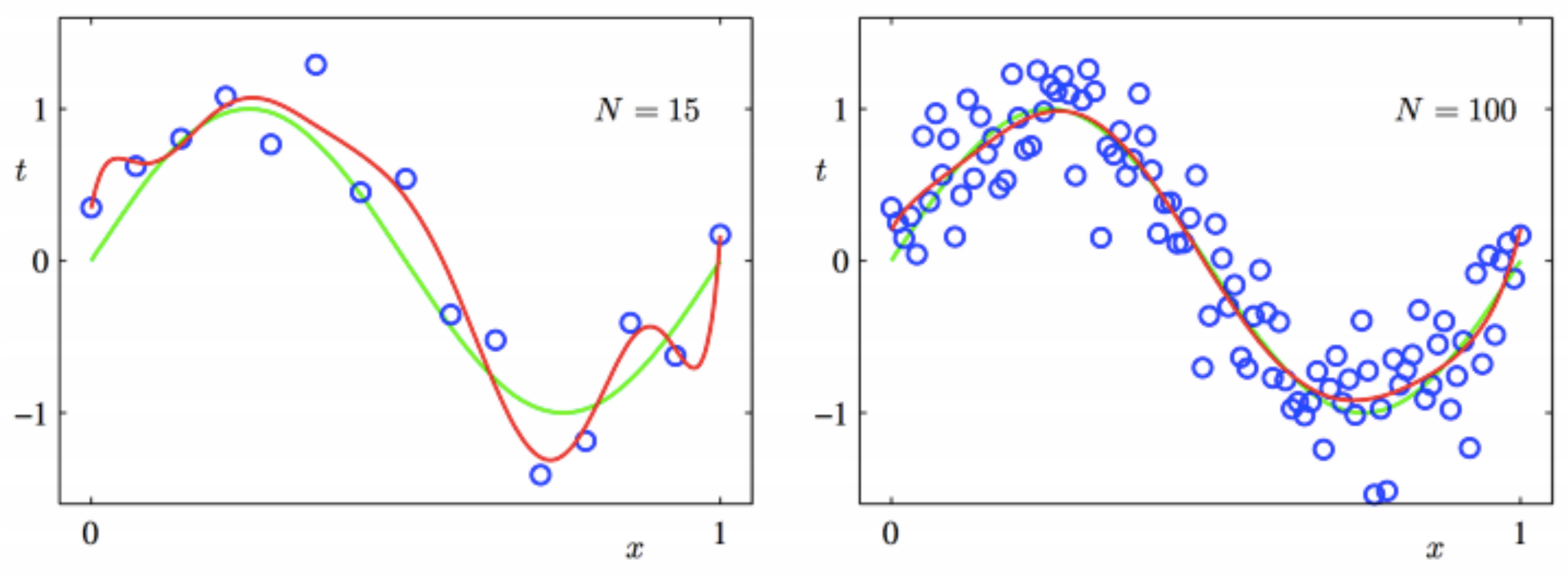
- Use lower degree polynomial:



- Risk: may be underfitting again

How to Avoid Overfitting?

- Add more training data



- What are the challenges/costs with collecting more training data?

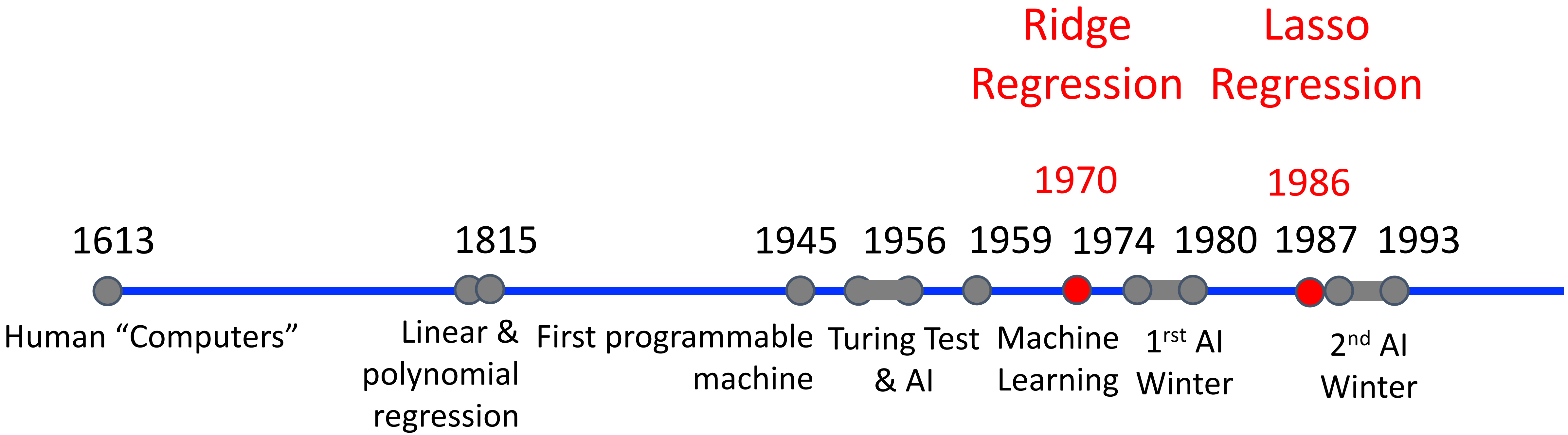
How to Avoid Overfitting?

- Or regularize the model…

Today’s Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

Linear Regression: Historical Context



Santosa, Fadil; Symes, William W. (1986). "Linear inversion of band-limited reflection seismograms". *SIAM Journal on Scientific and Statistical Computing*. SIAM. **7** (4): 1307–1330.

Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the lasso". *Journal of the Royal Statistical Society*. Series B (methodological). Wiley. **58** (1): 267–88.

Arthur E. Hoerl and Robert W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems", Technometrics. 1970.

Problem: Overfitting

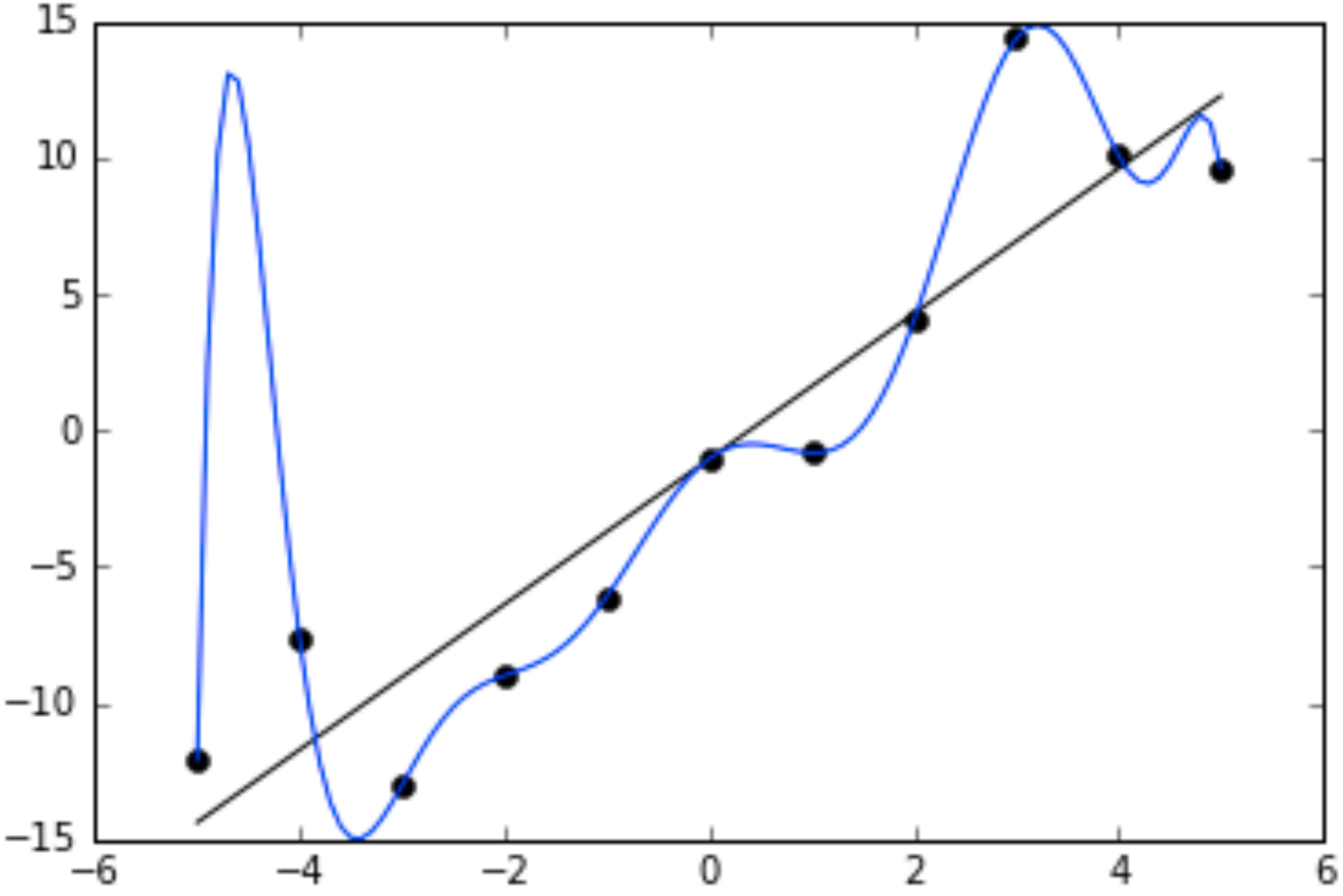


Figure Source: https://en.wikipedia.org/wiki/Overfitting

Problem: Overfitting

- e.g., weights learned for fitting a model to a sine wave function (polynomial degrees 0, 1, …, 9)

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

- Sign of overfitting: weights blow up and cancel each other out to fit the training data

Solution: Regularization

- Regularize model (add constraints)

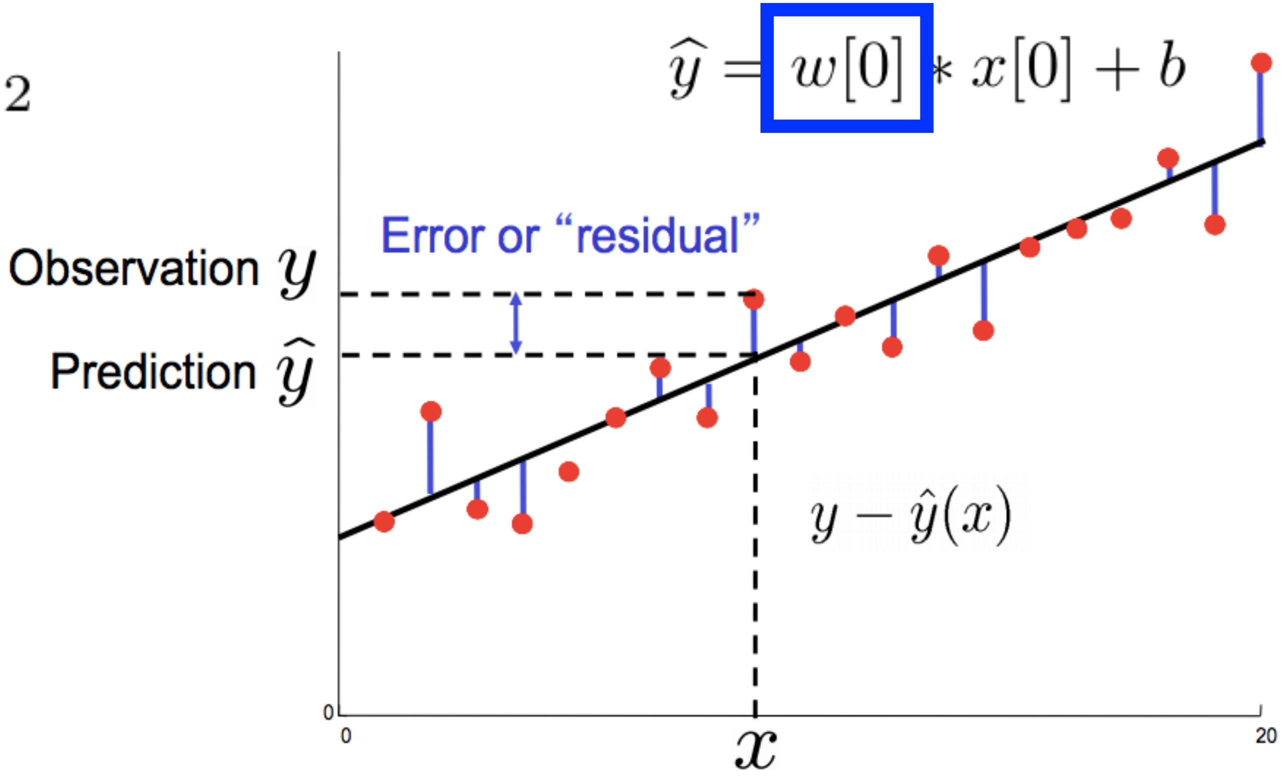
	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

- Idea: add constraint to minimize presence of large weights in models!

Regularization

- Idea: add constraint to minimize presence of large weights in models
- Recall: we previously learned models by *minimizing* **s**um of **s**quared **e**rrors (SSE) for all n training examples:

$$SSE = \sum_{i=1}^{n}(y^{(i)} - \widehat{y}^{(i)})^2$$



Regularization

- Idea: add constraint to minimize presence of large weights in models
- Recall: we previously learned models by *minimizing* **s**um of **s**quared **e**rrors (SSE) for all n training examples:

$$SSE = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

- Ridge Regression (l2): add constraint to penalize squared weight values

$$Error = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2 + \alpha \sum_{j=1}^{m} w_j^2$$

- Lasso Regression (l1): add constraint to penalize absolute weight values

$$Error = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2 + \alpha \sum_{j=1}^{m} |w_j|$$

Regularization: How to Set Alpha?

Recall: $$\hat{y} = \sum_{j=1}^{m} w_j x_j + b$$

What happens when you set alpha to a small value?

What happens when you set alpha to a large value?

- Ridge Regression (l2): add constraint to penalize squared weight values

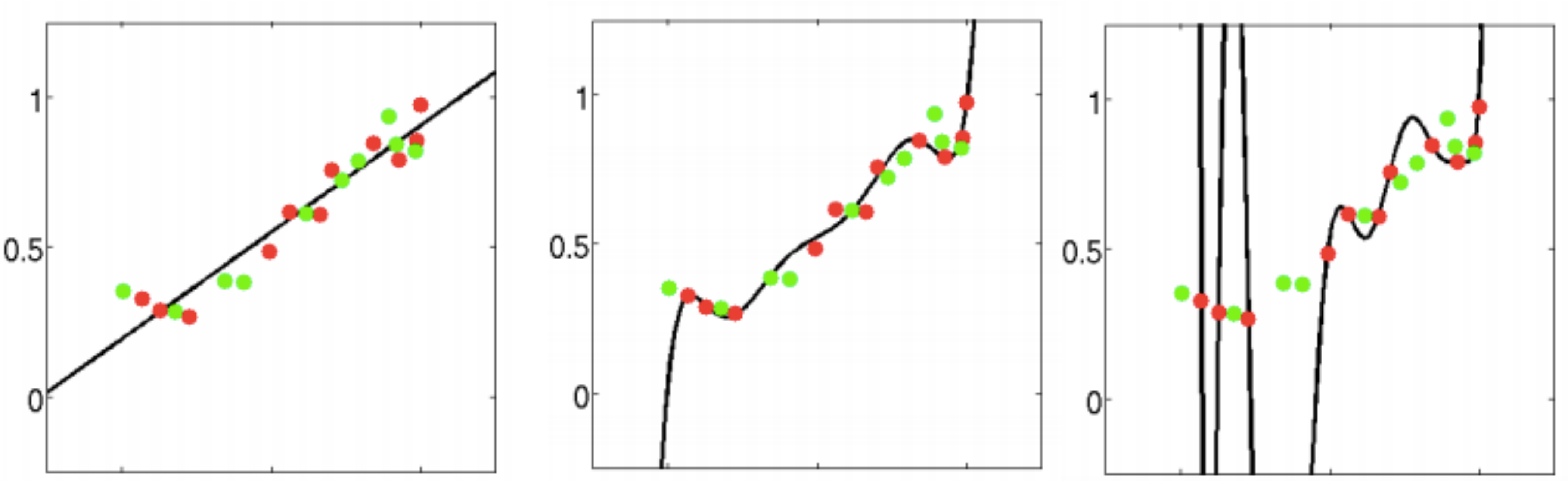
$$Error = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2 + \alpha \sum_{j=1}^{m} w_j^2$$

- Lasso Regression (l1): add constraint to penalize absolute weight values

$$Error = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2 + \alpha \sum_{j=1}^{m} |w_j|$$

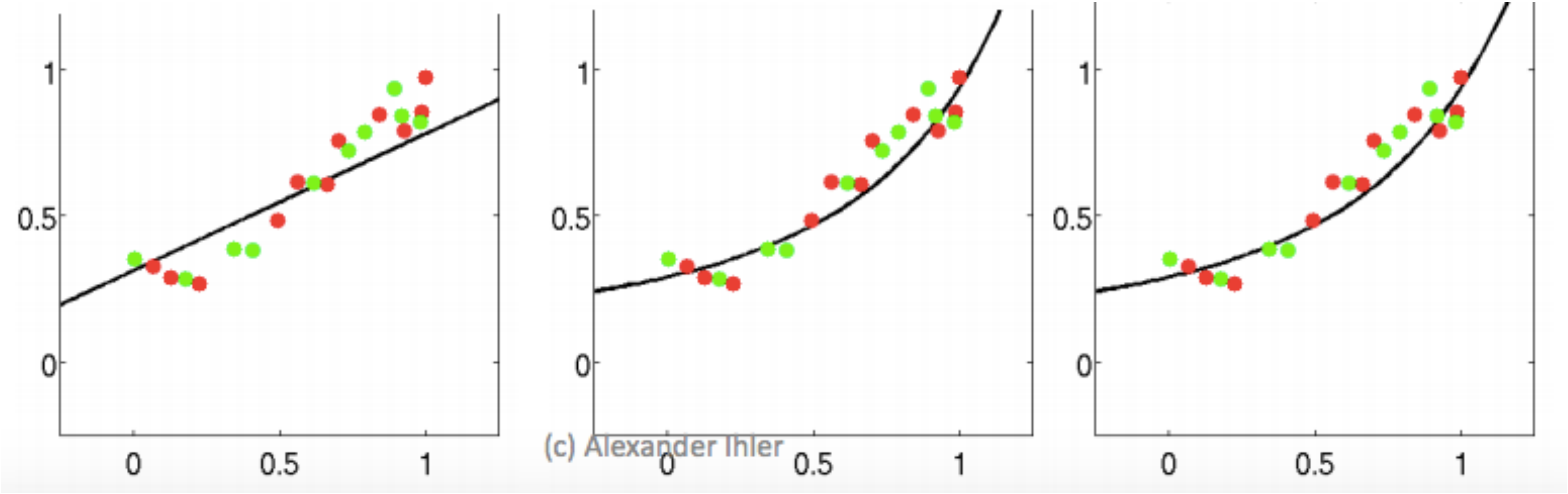
Regularization: How to Set Alpha?

Is alpha set to a small or large value for these three models?



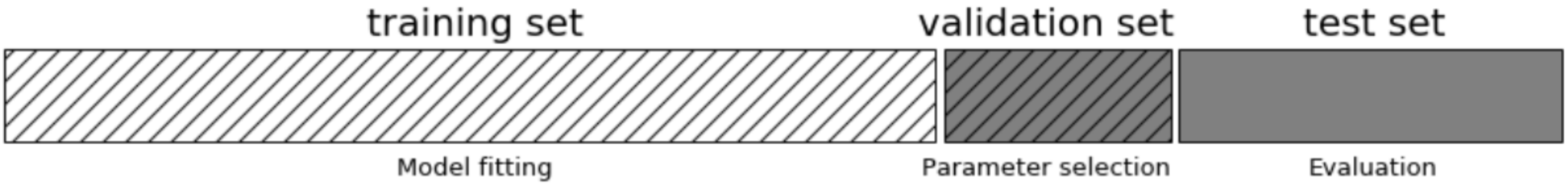
Regularization: How to Set Alpha?

Is alpha set to a small or large value for these three models?



Regularization: How to Set Alpha?

- Split training data into “train” and “validation” datasets



- Algorithm: brute-force, exhaustive approach by evaluating every alpha value to find optimal hyperparameter

Today’s Topics

- Regression applications
- Evaluating regression models
- Background: notation
- Linear regression
- Polynomial regression
- Regularization (Ridge regression and Lasso regression)

Resources Used for Today's Slides

- Deep Learning by Goodfellow et. al
  - pgs. 29-38 for background on linear algebra (e.g., matrices, norms)
- http://www.cs.utoronto.ca/~fidler/teaching/2015/slides/CSC411/
- http://www.cs.cmu.edu/~epxing/Class/10701/lecture.html
- http://web.cs.ucla.edu/~sriram/courses/cs188.winter-2017/html/index.html
- https://people.eecs.berkeley.edu/~jrs/189/
- http://alex.smola.org/teaching/cmu2013-10-701/
- http://sli.ics.uci.edu/Classes/2015W-273a