Today

- (Ch 13) Regression
  - The regression problem
  - Training a linear regression model using least squares
  - Evaluating a model using the R-squared metric

Next lecture

- (Ch 13) Regression
  - Outliers, overfitting and regularization
  - Nearest neighbors regression

A charming house minutes from Apple HQ

Source: zillow.com

Wait … is that a reasonable price?

10341 N Portal Ave
Cupertino, CA 95014

4 beds · 3 baths · 2,621 sqft

Extensive Luxury Remodel, Fantastic Price Per Square Foot of $1,101.87!

Facts and Features

Type Single Family	**Year Built** 1910	**Heating** Forced air
Cooling None	**Parking** 5 spaces	**Lot** 0.25 acres
Days on Zillow 133 Days	**Price/sqft** $1,064	**Saves** 29

FOR SALE

$2,788,000

Price cut: -$100,000 (11/15)

Zestimate®: $2,984,865

EST. MORTGAGE

$11,325/mo

Get pre-qualified

DATE	EVENT	PRICE		$/SQFT
11/15/2018	Price change	$2,788,000	-3.5%	$1,064
11/12/2018	Back on market	$2,888,000	--	$1,102
10/22/2018	Pending sale	$2,888,000	--	$1,102
10/18/2018	Back on market	$2,888,000	--	$1,102
10/15/2018	Pending sale	$2,888,000	--	$1,102
10/10/2018	Price change	$2,888,000	-3.3%	$1,102
8/7/2018	Price change	$2,988,000	-9.1%	$1,140
7/18/2018	Listed for sale	$3,288,000	+28.9%	$1,254

Source: zillow.com

Can we use data to predict the sale price?

Cupertino Real Estate Just Sold – COE by Nov 17, 2018

Cupertino Single Family Home Sales

ADDRESS	ORGLD	ORIG LSPRC	LIST PRICE	SALE PRICE	SQFT	LOTSZ	COE	DOM	ZIP
6060 Willowgrove LN	Oct-03	1,858,000	1,858,000	1,800,000	1574	6935	Nov-14	23	95014
10156 Byrne AVE	Aug-14	1,950,000	1,825,000	1,825,000	1015	6623	Nov-09	36	95014
10630 Gascoigne DR	Oct-17	1,938,000	1,938,000	1,900,000	1905	5508	Nov-08	8	95014
10408 Normandy CT	Oct-03	1,798,000	1,798,000	2,025,000	1937	9775	Nov-15	8	95014
1322 Flower CT	Oct-16	2,088,000	2,088,000	2,050,000	1853	9900	Nov-01	9	95014
21980 Mcclellan RD	Sep-27	1,988,988	1,988,988	2,100,000	1838	7500	Nov-13	23	95014
21524 Conradia CT	Oct-17	2,190,000	2,088,000	2,150,000	1548	7850	Nov-14	12	95014
20646 Craig CT	Oct-02	1,988,888	1,988,888	2,360,101	1416	7490	Nov-08	8	95014
8077 HYANNISPORT DR	Sep-25	2,488,000	2,488,000	2,410,000	2397	6222	Nov-13	21	95014
21559 Edward WAY	Oct-12	2,198,000	2,198,000	2,666,000	2135	7500	Nov-05	5	95014
22044 San Fernando CT	Sep-04	2,849,000	2,698,000	2,750,000	2817	7282	Nov-16	44	95014
22416 Cupertino RD	Oct-06	3,289,000	3,289,000	3,225,000	3559	10454	Nov-16	12	95014

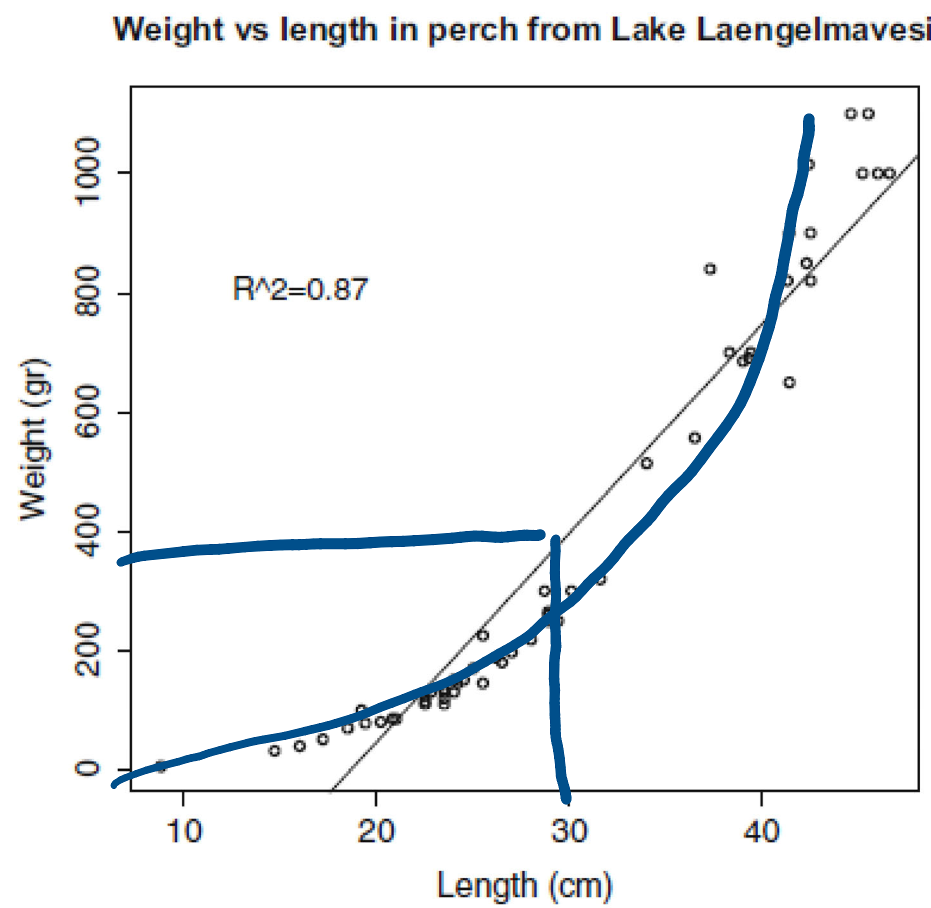
Source: julianalee.com/cupertino/cupertino-home-sales.htm

The regression problem

- Given a set of **feature vectors** $\mathbf{x}_i$ where each has a **numerical label** y_i, we want to train a model that can map unlabeled vectors to numerical values

- We can think of regression as fitting a line (or curve or hyperplane, etc.) to data

- Regression is like classification except that the prediction target is a number, not a class label (and that changes everything)



Some terminology

- Suppose the dataset $\{(\mathbf{x}, y)\}$ consists of N labeled items $(\mathbf{x}_i, y_i)$
- If we represent the dataset as a table
  - The d columns representing $\{\mathbf{x}\}$ are called **explanatory variables $\mathbf{x}^{(j)}$**
  - The numerical column y is called the **dependent variable**

N=3

$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1	3	0
2	3	2
3	6	5

d=2

Linear model

- We begin by modeling y as a linear function of $\mathbf{x}^{(j)}$ plus randomness

$$y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \cdots + \mathbf{x}^{(d)}\beta_d + \xi$$

where ξ is a zero-mean random variable that represents model error

- In vector notation

$$y = \overbrace{\mathbf{x}^T}^{[x^{(1)}\ x^{(2)}\ \cdots\ x^{(d)}]}\boldsymbol{\beta} + \xi \qquad \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{bmatrix}$$

where $\boldsymbol{\beta}$ is the d-dimensional vector of coefficients that we train

Each data item gives an equation ...

$(d=2)$

Model: $y = \mathbf{x}^T \boldsymbol{\beta} + \xi = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$

Training data

$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1	3	0
2	3	2
3	6	5

$$0 = \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \xi_1$$

$$2 = \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \xi_2$$

$$5 = \begin{bmatrix} 3 & 6 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \xi_3$$

… which together form a matrix equation

$$\begin{bmatrix} 0 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 3 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}$$

$$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$$

$$y = X \cdot \beta + e$$

Training the model means choosing $\boldsymbol{\beta}$

- Given a training dataset $\{(\mathbf{x}, y)\}$, we want to fit a model $y = \mathbf{x}^T\boldsymbol{\beta} + \xi$

- Define $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$ and $X = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$ and $\mathbf{e} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_N \end{bmatrix}$

- To train the model, we must choose $\boldsymbol{\beta}$ that makes $\mathbf{e}$ small in the matrix equation

$$\mathbf{y} = X\boldsymbol{\beta} + \mathbf{e}$$

Training using least squares

- In the least squares method, we aim to minimize $||\mathbf{e}||^2$

$$||\mathbf{e}||^2 = ||\mathbf{y} - X\boldsymbol{\beta}||^2 = (\mathbf{y} - X\boldsymbol{\beta})^T(\mathbf{y} - X\boldsymbol{\beta})$$

- Differentiating and setting to zero (and skipping some matrix calculus) gives

$$X^T X\boldsymbol{\beta} - X^T\mathbf{y} = \mathbf{0}$$

- If $X^T X$ is invertible, the least squares estimate of the coefficients is

$$\boxed{\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T\mathbf{y}}$$

Training using least squares example

Model: $y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi = \mathbf{x}^T\boldsymbol{\beta} + \xi$

Training data

$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1	3	0
2	3	2
3	6	5

X $\quad$ y

$$\hat{\beta} = (X^T X)^{-1} X^T y = \begin{bmatrix} 2 \\ -1/3 \end{bmatrix}$$

$$\hat{\beta}_1 = 2$$

$$\hat{\beta}_2 = -1/3$$

Prediction

- If we train the model with coefficients $\widehat{\boldsymbol{\beta}}$, we can predict y_0^p from $\mathbf{x}_0$

$$\mathrm{y}_0^p = \mathbf{x}_0^T \widehat{\boldsymbol{\beta}}$$

- In the model $y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$ with $\widehat{\boldsymbol{\beta}} = \begin{bmatrix} 2 \\ -{}^{1}/_{3} \end{bmatrix}$
  - the prediction for $\mathbf{x}_0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is $\mathrm{y}_0^p =$ $\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -1/3 \end{bmatrix} = 4 - 1/3 = 3\,2/3$
  - the prediction for $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is $\mathrm{y}_0^p =$ $\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ -1/3 \end{bmatrix} = 0$

A linear model with constant offset

- The problem with the model $y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$ is that it always predicts $y_0^p = 0$ if the input feature vector $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

- Let's add a constant offset β_0 to the model

$$y = \beta_0 + \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$$

Training and prediction with constant offset

Model: $y = \underline{\beta_0 + \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2} + \xi = \mathbf{x}^T\boldsymbol{\beta} + \xi$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

$$\mathbf{x}^T = \begin{bmatrix} 1 & x^{(1)} & x^{(2)} \end{bmatrix}$$

Training data

1	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y
1	1	3	0
1	2	3	2
1	3	6	5

X $\quad Y$

$$\hat{\beta} = (X^T X)^{-1} X^T Y = \begin{bmatrix} -3 \\ 2 \\ 1/3 \end{bmatrix}$$

If $x^{(1)} = 0$ & $x^{(2)} = 0$, then $y_0 ? = [1\ 0\ 0]\begin{bmatrix} -3 \\ 2 \\ 1/3 \end{bmatrix} = -3$

Evaluating models using R-squared

- The least squares estimate satisfies this property (proven in book)

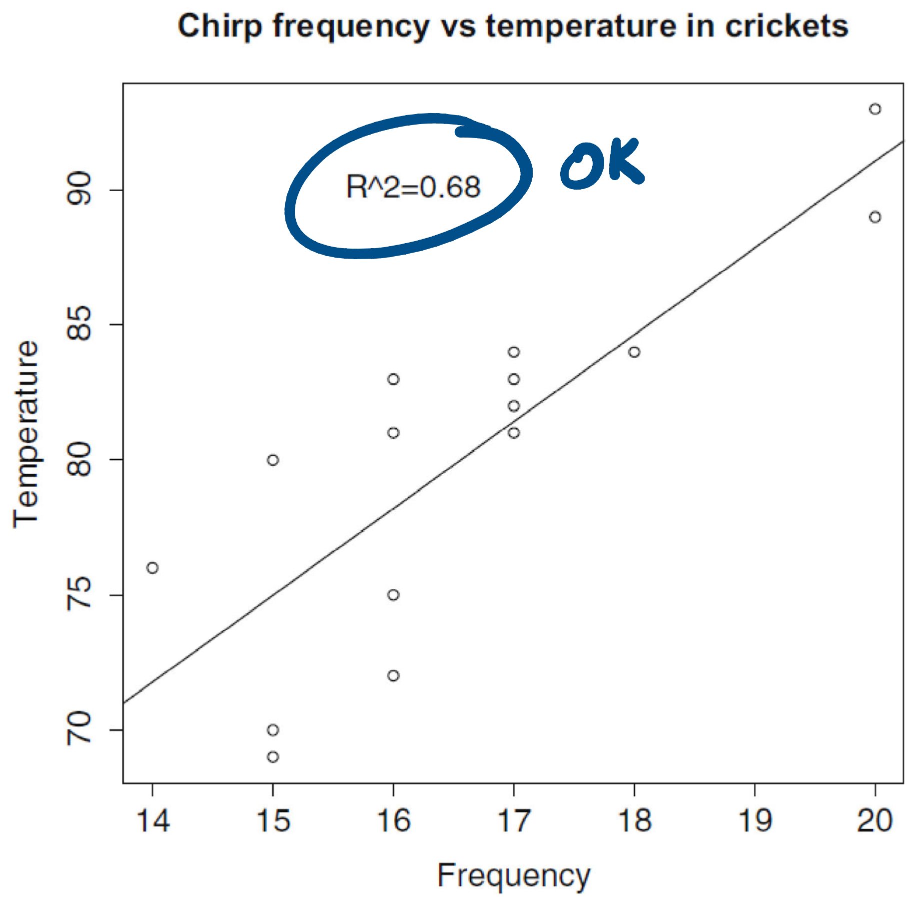
$$\mathrm{var}(\{y_i\}) = \mathrm{var}\left(\left\{\mathbf{x}_i^T\widehat{\boldsymbol{\beta}}\right\}\right) + \mathrm{var}(\{\xi_i\})$$

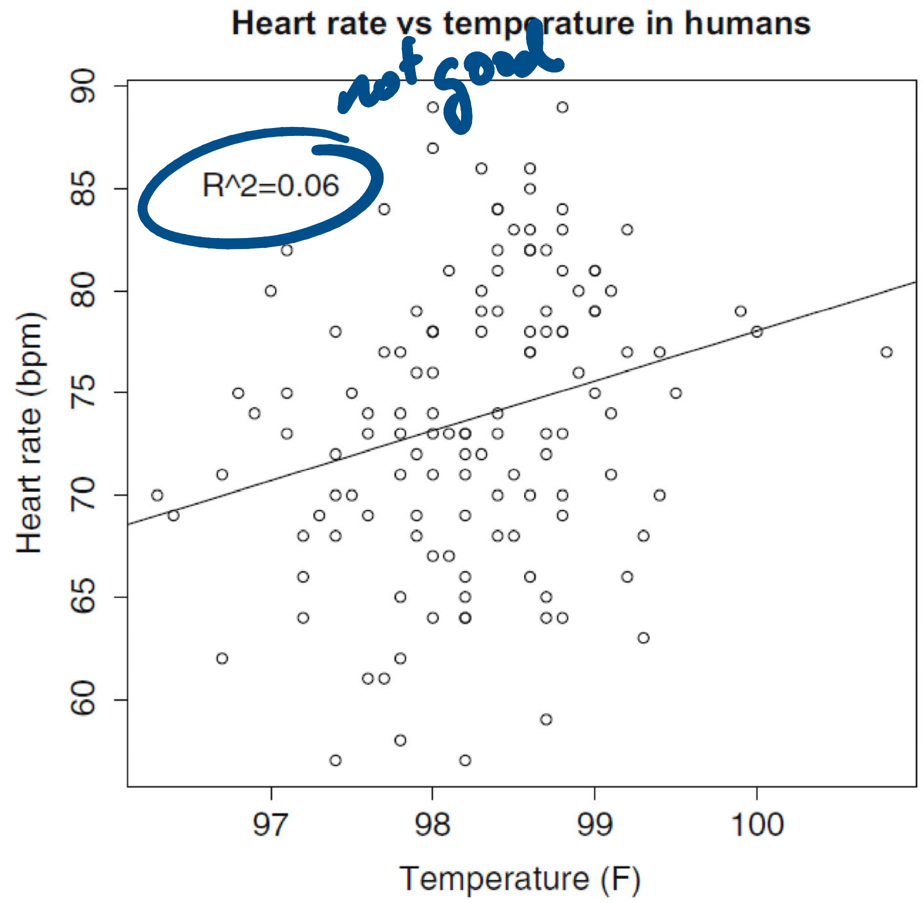
- This property gives us an evaluation metric called R squared

$$R^2 = \frac{\mathrm{var}\left(\left\{\mathbf{x}_i^T\widehat{\boldsymbol{\beta}}\right\}\right)}{\mathrm{var}(\{y_i\})}$$

- We have $0 \leq R^2 \leq 1$ with a larger value meaning a better fit

R-squared examples





Comparing our example models

$$y = \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$$

$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y	$\mathbf{x}^T\widehat{\boldsymbol{\beta}}$
1	3	0	1
2	3	2	3
3	6	5	4

$$\widehat{\boldsymbol{\beta}} = \begin{bmatrix} 2 \\ -{}^{1}/_{3} \end{bmatrix}$$

$$R^2 = \frac{var(\{1,3,4\})}{var(\{0,2,5\})} = 0.37$$

$$y = \beta_0 + \mathbf{x}^{(1)}\beta_1 + \mathbf{x}^{(2)}\beta_2 + \xi$$

1	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	y	$\mathbf{x}^T\widehat{\boldsymbol{\beta}}$
1	1	3	0	0
1	2	3	2	2
1	3	6	5	5

$$\widehat{\boldsymbol{\beta}} = \begin{bmatrix} -3 \\ 2 \\ {}^{1}/_{3} \end{bmatrix}$$

$$R^2 = \frac{var(\{0,2,5\})}{var(\{0,2,5\})} = 1$$

perfect fit!