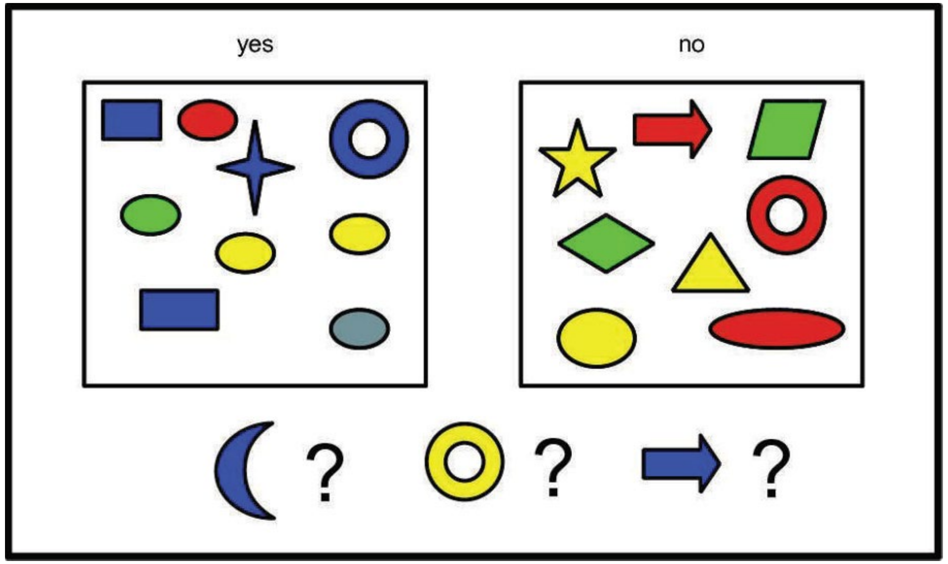
Regression

JinYeong Bak
jy.bak@skku.edu
Human Language Intelligence Lab, SKKU

Supervised Learning

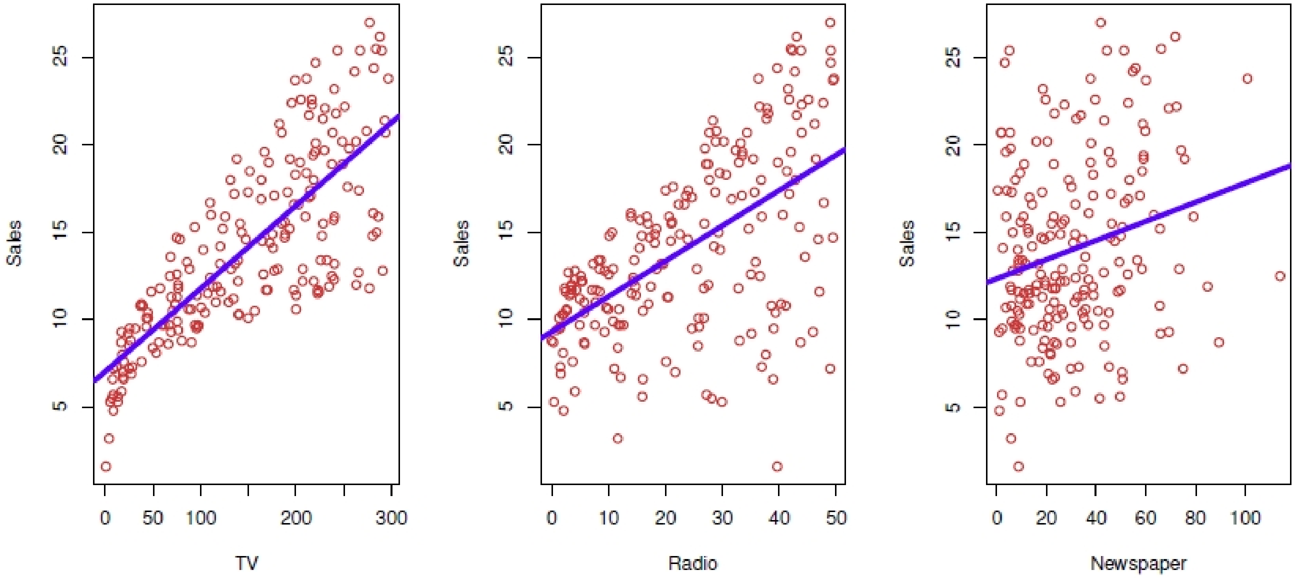
- Given: Training data as labeled instances $\{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$
- Goal: Learn a rule $(f: x \rightarrow y)$ to predict outputs y for new inputs x
- Example)
  - Data: ((Blue, Square, 10), yes), . . . ((Red, Ellipse, 20.7), no)
  - Task: For new inputs (Blue, Crescent, 10), (Yellow, Circle, 12), are they yes/no?



Color	Shape	Size	Label
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	0
Blue	Crescent	10	?
Yellow	Circle	12	?

Supervised Learning

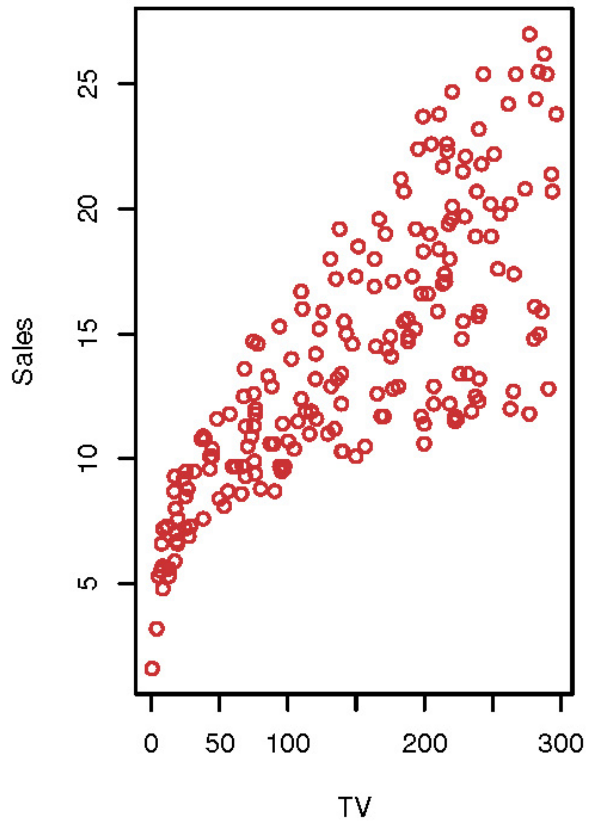
- Regression: Real-valued outputs
- Example)
  - Data: Advertising budgets and sales {(TV, Radio, Newspaper), Sales}
  - Task: Predict sales given new advertising budgets
  - Method: Fitting a line or non-linear curve



SIMPLE LINEAR REGRESSION

Problem

- Data: Advertising budgets and sales {TV, Sales}
- Task: Predict sales given new advertising TV budget



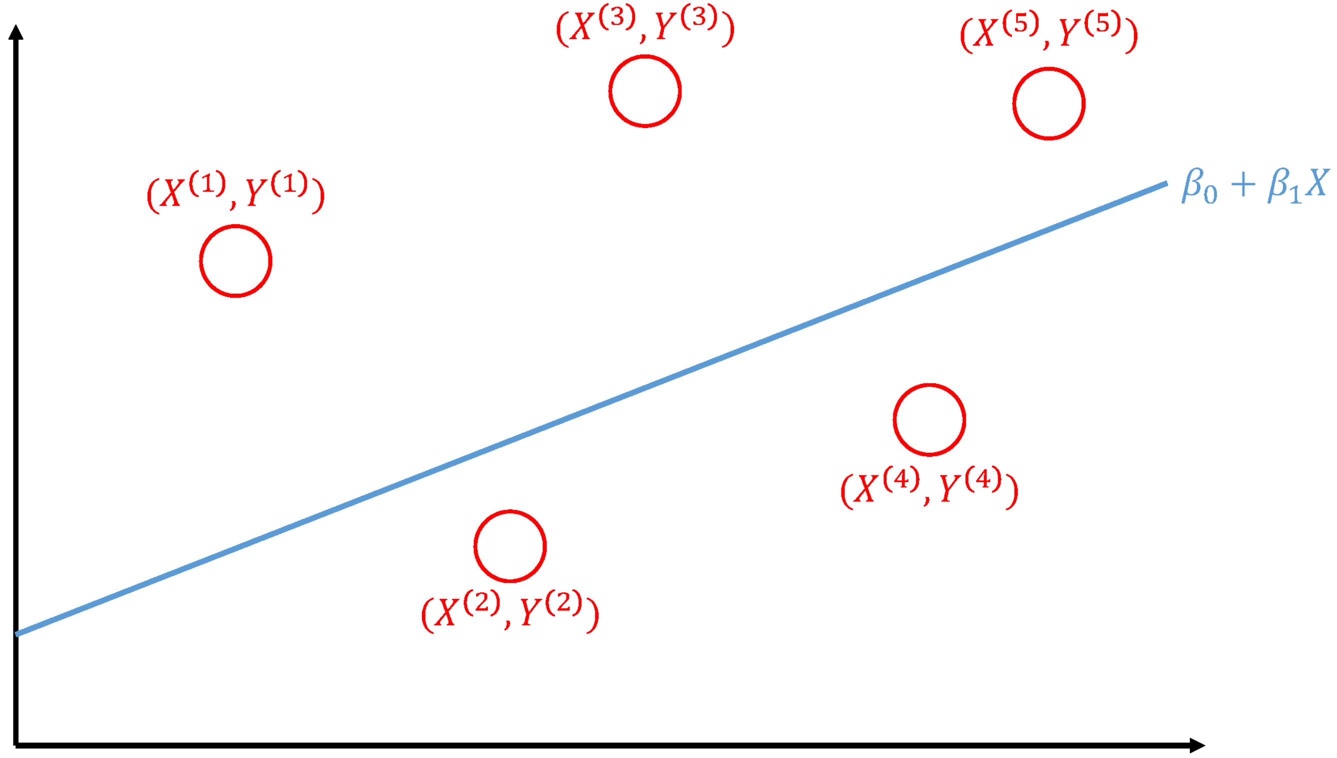
Data

- N: # training data
- X: TV ad budget (input variable, features)
- Y: sales (output variable, response variable)
- (x, y): one training data
- $(x^{(i)}, y^{(i)})$: i-th training data

X	Y
230.1	22.1
44.5	10.4
17.2	9.3
151.5	18.5
180.8	12.9
8.7	7.2
57.5	11.8
⋮	⋮

Linear Regression

- Data: N TV advertising budgets and sales (X, Y)
- Task: Predict sales $y^{(test)}$ given new advertising TV budget $x^{(test)}$
- Model: $Y \approx \beta_0 + \beta_1 X$

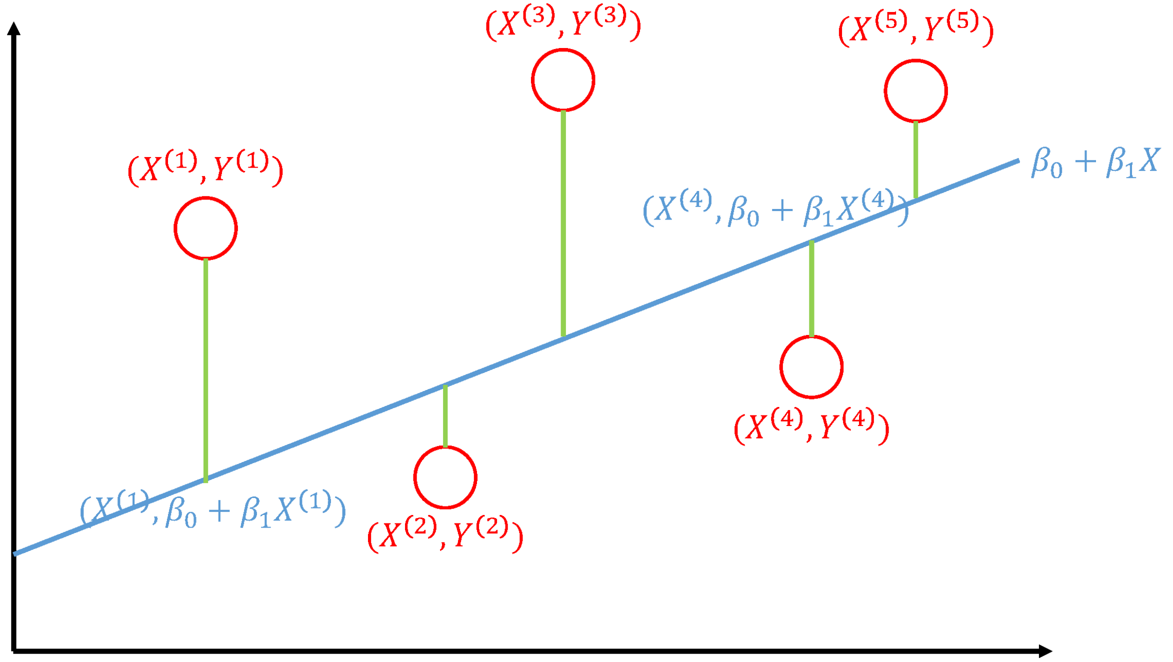


Linear Regression

- Data: N TV advertising budgets and sales (X, Y)
- Task: Predict sales $y^{(test)}$ given new advertising TV budget $x^{(test)}$
- Model: $Y \approx \beta_0 + \beta_1 X$
- New problem: Find the best β_0 and β_1
- A question: What are the best β_0 and β_1?
- Possible answer: Given a data $x^{(i)}$, no difference between
  - $\hat{y}^{(i)}$: output of the model with β_0 and β_1
  - $y^{(i)}$: real data output

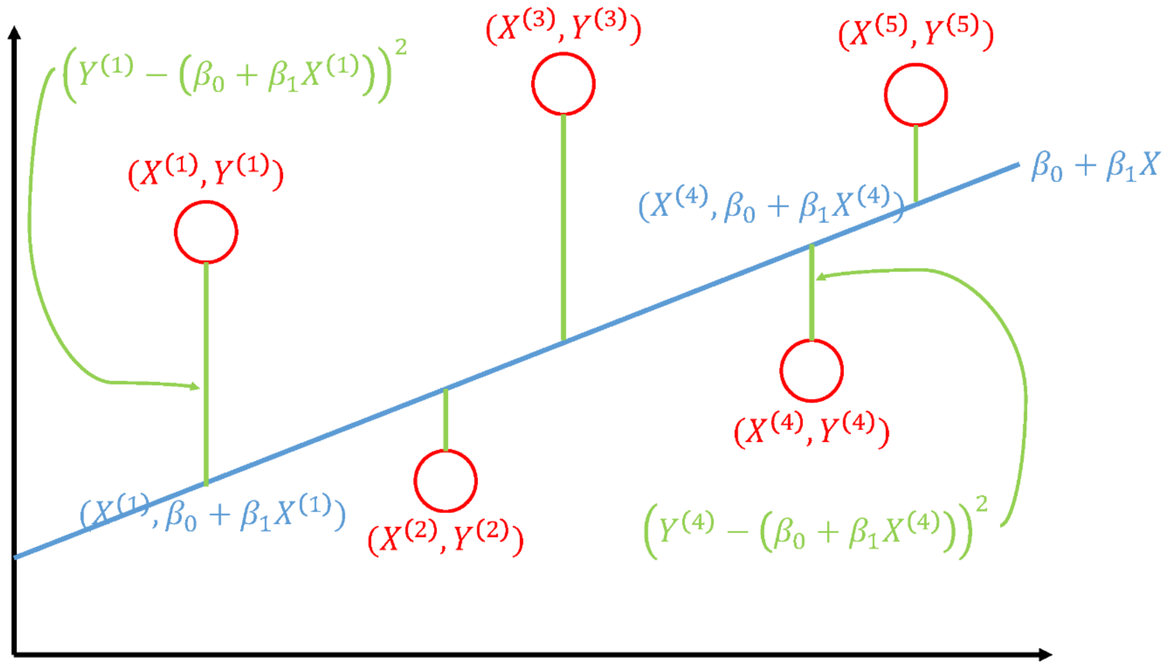
Linear Regression

- Data: N TV advertising budgets and sales (X, Y)
- Task: Predict sales $y^{(test)}$ given new advertising TV budget $x^{(test)}$
- Model: $Y \approx \beta_0 + \beta_1 X$
- Idea: Given a data $x^{(i)}$, minimize the difference between $\hat{y}^{(i)}$ and $y^{(i)}$



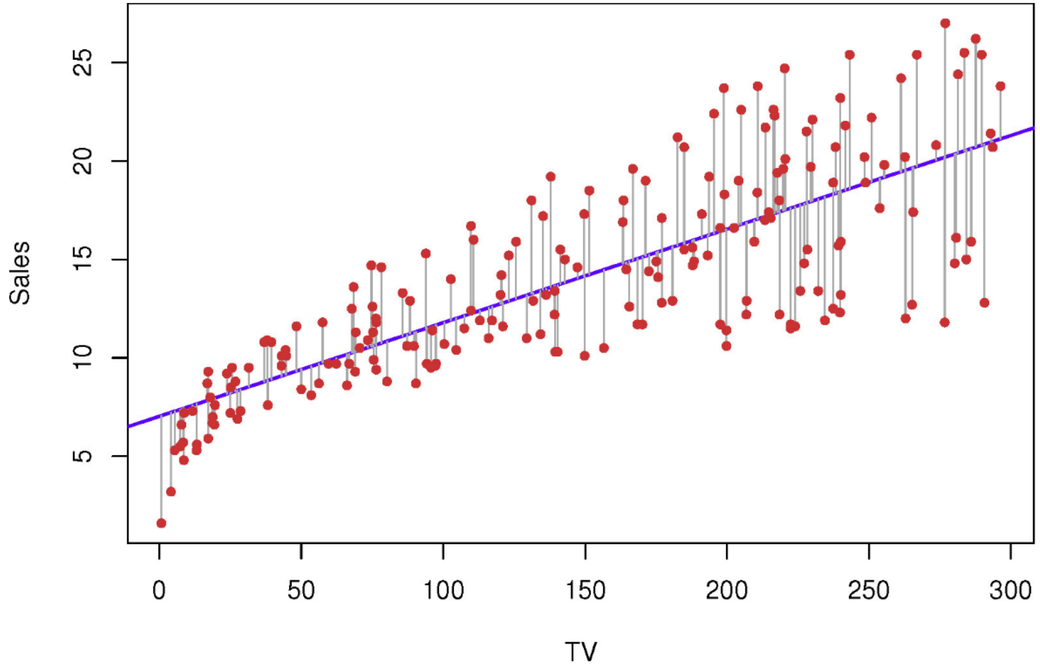
Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X$
- Idea: Given a data $x^{(i)}$, minimize the difference between $\hat{y}^{(i)}$ and $y^{(i)}$
- Difference: $\left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)}\right)\right)^2$



Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X$
- Idea: Given a data $x^{(i)}$, minimize the difference between $\hat{y}^{(i)}$ and $y^{(i)}$
- All data difference: $\sum_i^N \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)}\right)\right)^2$



Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X$
- Idea: Given a data $x^{(i)}$, minimize the difference between $\hat{y}^{(i)}$ and $y^{(i)}$
- All data difference: $\sum_i^N \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)}\right)\right)^2$
- Method: Find the best β_0 and β_1 that minimize the all data difference

$$\underset{\beta_0,\beta_1}{\arg\min} \sum_i^N \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)}\right)\right)^2$$

Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X$
- Parameters: β_0, β_1
- Loss function

$$\mathrm{L}(\beta_0, \beta_1) = \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} \right) \right)^2$$

- Task

$$\underset{\beta_0, \beta_1}{\arg\min} \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} \right) \right)^2$$

Linear Regression

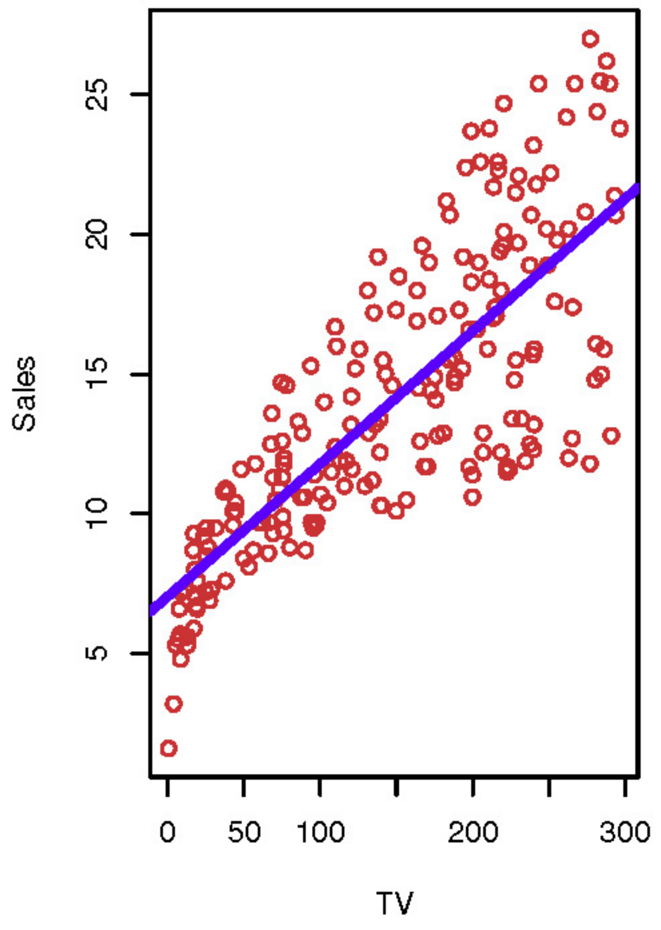
- Model: $Y \approx \beta_0 + \beta_1 X$
- Parameters: β_0, β_1
- Loss function

$$\mathrm{L}(\beta_0, \beta_1) = \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} \right) \right)^2$$

- Task

$$\underset{\beta_0, \beta_1}{\arg\min} \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} \right) \right)^2$$

- Algorithm: Gradient-descent algorithm



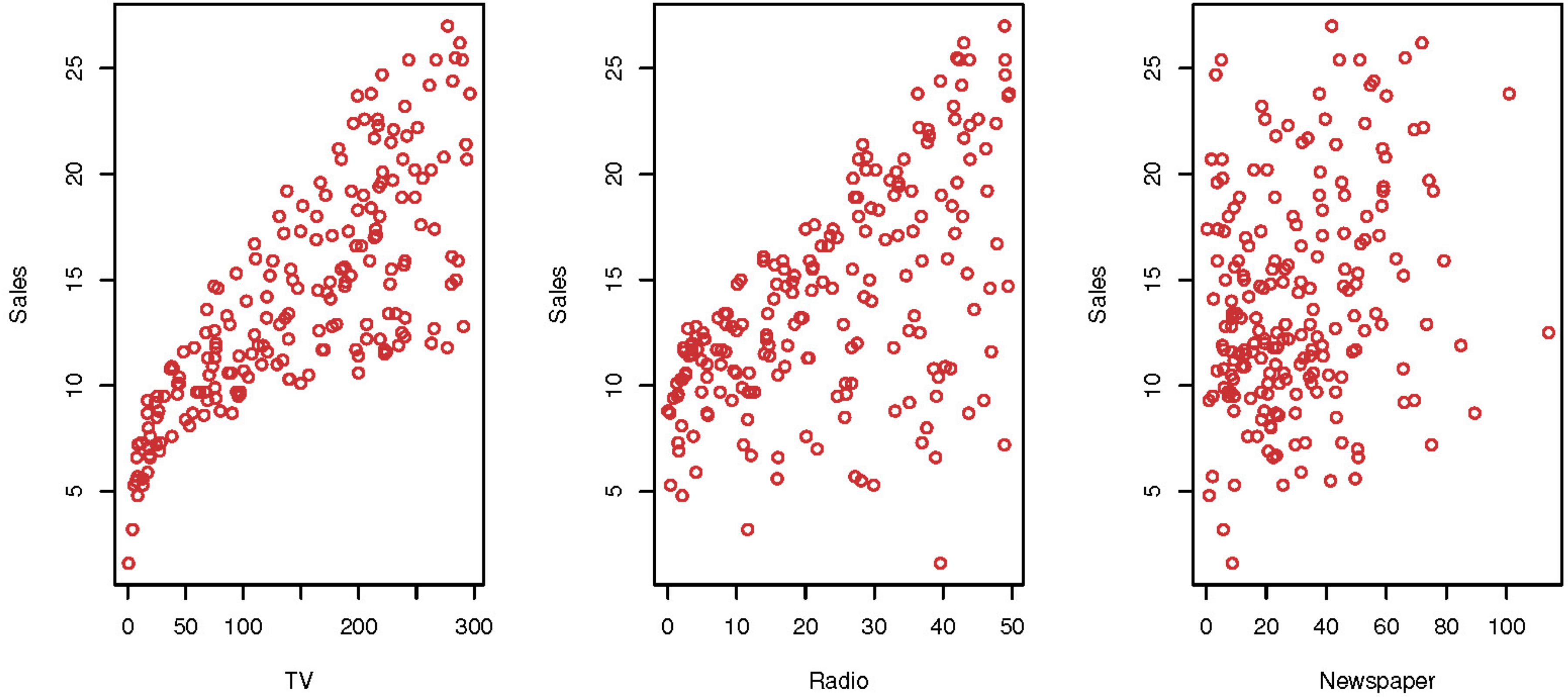
Supervised Learning

- Problem: Predict outputs y for new inputs x based on a rule $(f: x \rightarrow y)$
- Data: labeled instances $\{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$
- Model: Supervised model (e.g. linear regression)
- Parameters: unknown values of the model
- Loss function: Difference between the outputs of the model and the data
- Task: Find the parameters that minimize the loss function
- Algorithm: Various algorithms

MULTIPLE LINEAR REGRESSION

Problem

- Data: Advertising budgets and sales {(TV, Radio, Newspaper), Sales}
- Task: Predict sales given new advertising budgets



Data

- N: # training data
- X_1, X_2, X_3: (TV, Radio, Newspaper) AD budgets
- Y: sales
- (x_1, x_2, x_3, y): one training data
- $(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, y^{(i)})$: i-th training data

X_1	X_2	X_3	Y
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Linear Regression

- Data: N advertising budgets and sales (X_1, X_2, X_3, Y)
- Task: Predict sales $y^{(test)}$ given new ad budgets $x_1^{(test)}, x_2^{(test)}, x_3^{(test)}$
- Model: $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
- New problem: Find the best $\beta_0, \beta_1, \beta_2$ and β_3
- A question: What are the best βs?
- Possible answer: Given a data $x^{(i)}$, no difference between
  - $\hat{y}^{(i)}$: output of the model with $\beta_0, \beta_1, \beta_2$ and β_3
  - $y^{(i)}$: real data output

Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
- Idea: Given a data $x^{(i)}$, minimize the difference between $\hat{y}^{(i)}$ and $y^{(i)}$
- Difference: $\left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \left(y^{(i)} - \left(\beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \beta_3 x_3^{(i)}\right)\right)^2$

Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
- Idea: Given a data $x^{(i)}$, minimize the difference between $\hat{y}^{(i)}$ and $y^{(i)}$
- All data difference

$$\sum_{i}^{N} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \left(y^{(i)} - \left(\beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \beta_3 x_3^{(i)}\right)\right)^2$$

Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
- Idea: Given a data $x^{(i)}$, minimize the difference between $\hat{y}^{(i)}$ and $y^{(i)}$
- All data difference

$$\sum_{i}^{N} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \left(y^{(i)} - \left(\beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \beta_3 x_3^{(i)}\right)\right)^2$$

- Method: Find the best βs that minimize the all data difference

$$\underset{\beta_0, \beta_1, \beta_2, \beta_3}{\arg\min} \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \beta_3 x_3^{(i)}\right)\right)^2$$

Linear Regression

- Model: $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$
- Parameters: $\beta_0, \beta_1, \beta_2, \beta_3$
- Loss function

$$\mathrm{L}(\beta_0, \beta_1, \beta_2, \beta_3) = \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \beta_3 x_3^{(i)} \right) \right)^2$$

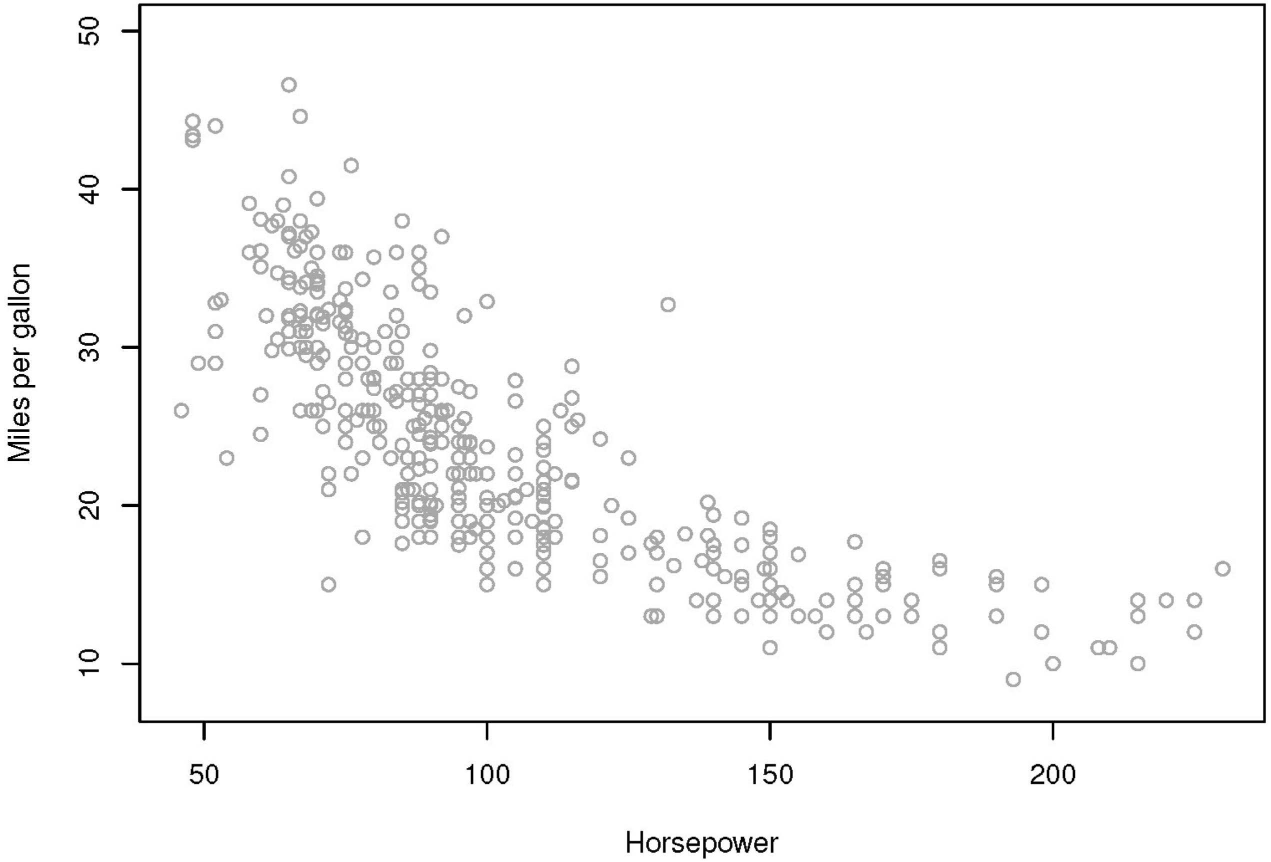
- Task

$$\underset{\beta_0, \beta_1, \beta_2, \beta_3}{\arg\min} \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \beta_3 x_3^{(i)} \right) \right)^2$$

POLYNOMIAL REGRESSION

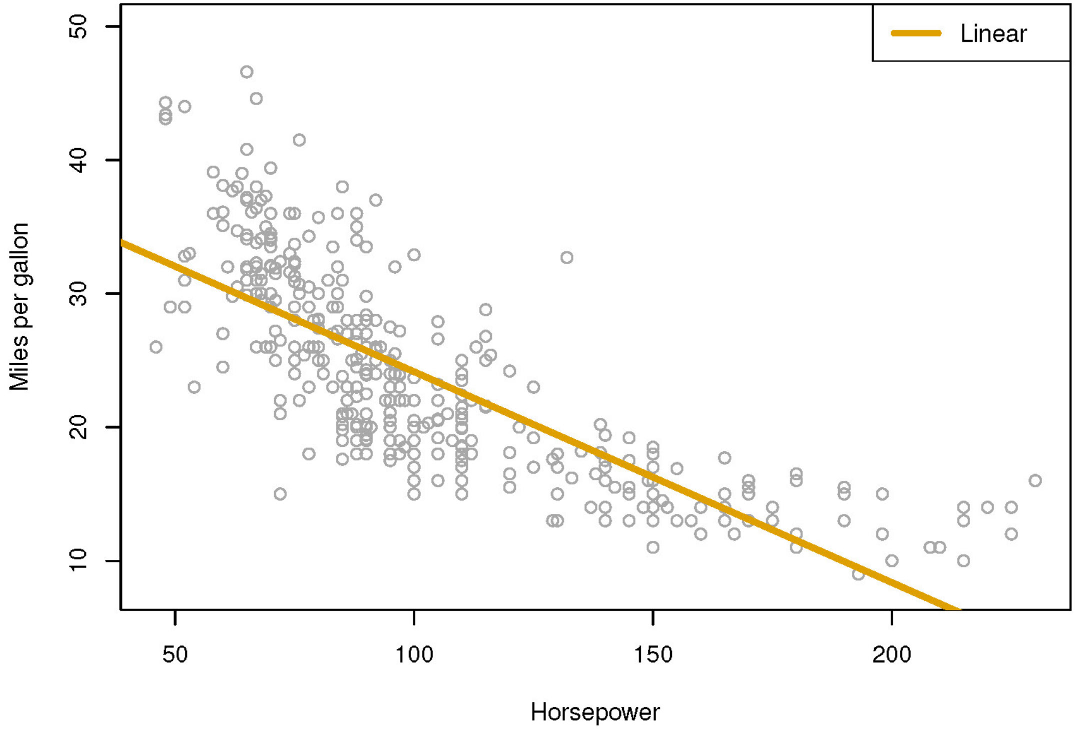
Problem

- Data: Car engine horsepower and miles/gallon {Horsepower, Miles/gallon}
- Task: Predict miles/gallon given a new car engine



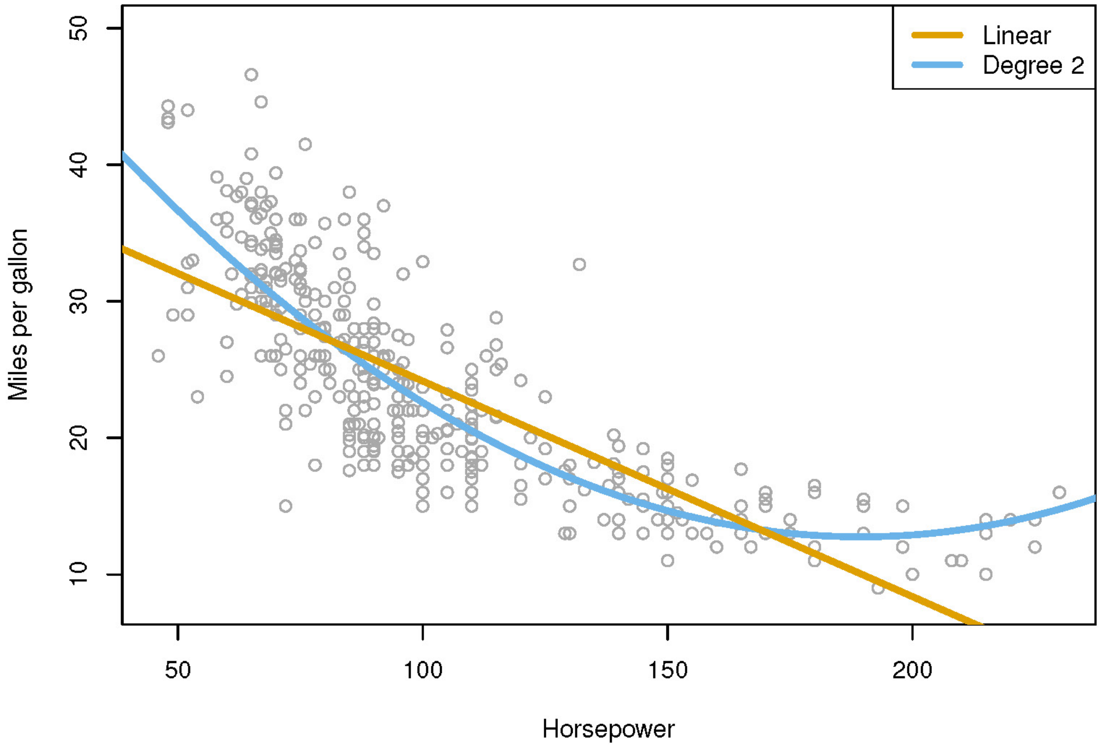
Linear Regression

- Data: Car engine horsepower and miles/gallon {Horsepower, Miles/gallon}
- Task: Predict miles/gallon given a new car engine
- Model: Simple Linear Regression $Y \approx \beta_0 + \beta_1 X$



Polynomial Regression

- Data: Car engine horsepower and miles/gallon {Horsepower, Miles/gallon}
- Task: Predict miles/gallon given a new car engine
- Model: $Y \approx \beta_0 + \beta_1 X + \beta_2 X^2$



Polynomial Regression

- Model: $Y \approx \beta_0 + \beta_1 X + \beta_2 X^2$
- Parameters: $\beta_0, \beta_1, \beta_2$
- Loss function

$$\mathrm{L}(\beta_0, \beta_1, \beta_2) = \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} + \beta_2 (x^{(i)})^2 \right) \right)^2$$

- Task

$$\underset{\beta_0, \beta_1, \beta_2}{\arg\min} \sum_{i}^{N} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} + \beta_2 (x^{(i)})^2 \right) \right)^2$$

- Algorithm: Gradient-descent algorithm