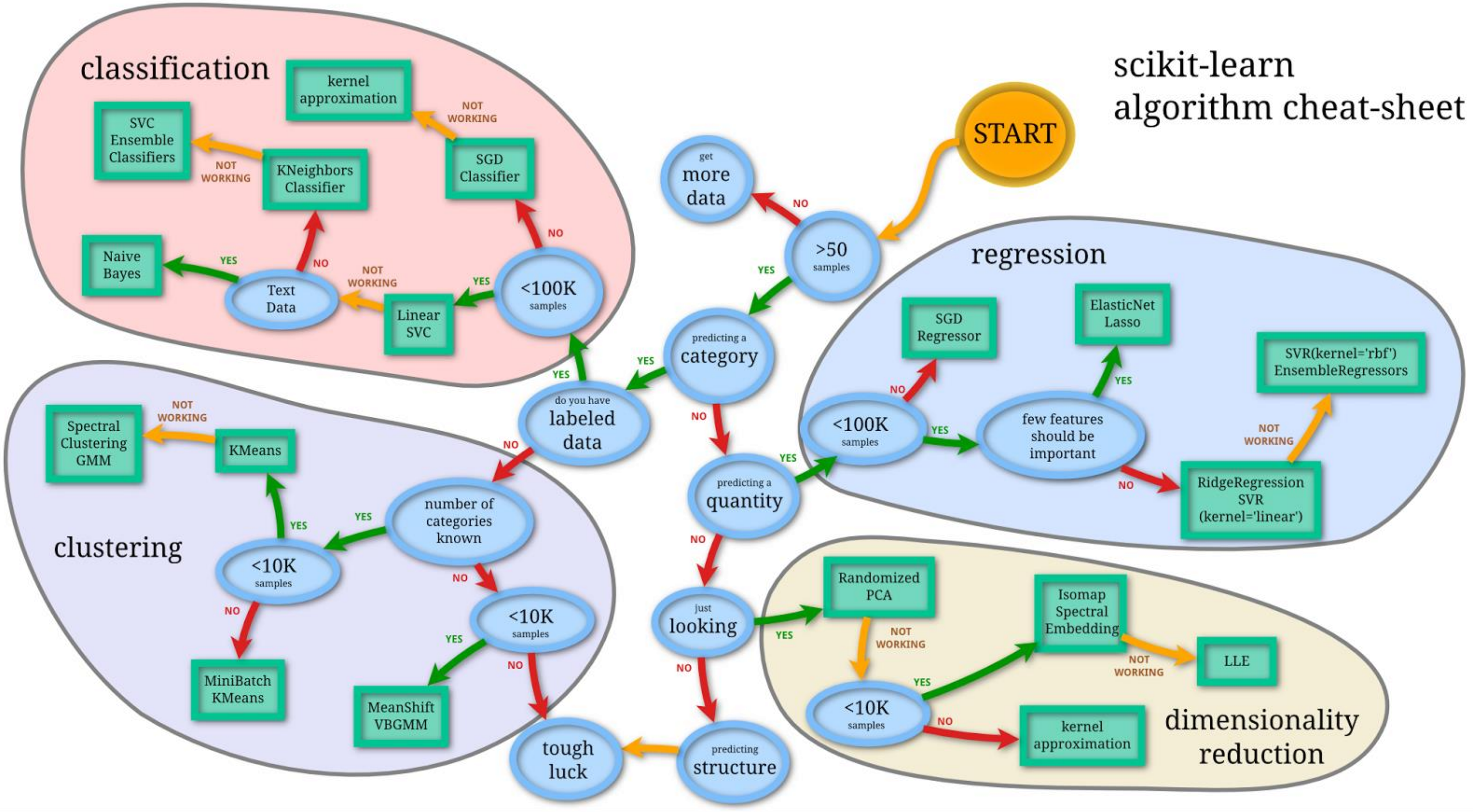
UNIVERSITY OF ARKANSAS

Linear Regression

Jiahui Chen
Department of Mathematical Sciences
University of Arkansas

Scikit-learn algorithm



Data set

Labeled data sets for supervised learning

Regression (R)

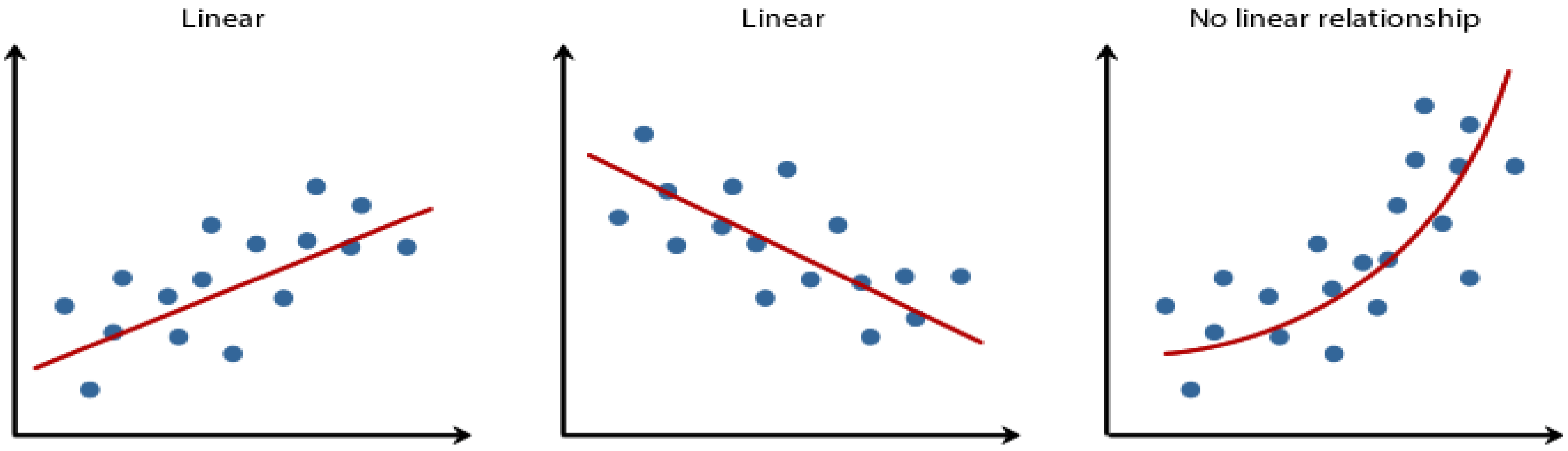
$$\text{Data set (R): } \mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right) \middle| \mathbf{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}\right\}_{i=1}^{M}$$

Classification (C)

$$\text{Data set (C): } \mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right) \middle| \mathbf{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \{0,1\}\right\}_{i=1}^{M}$$

Linear Regression

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).

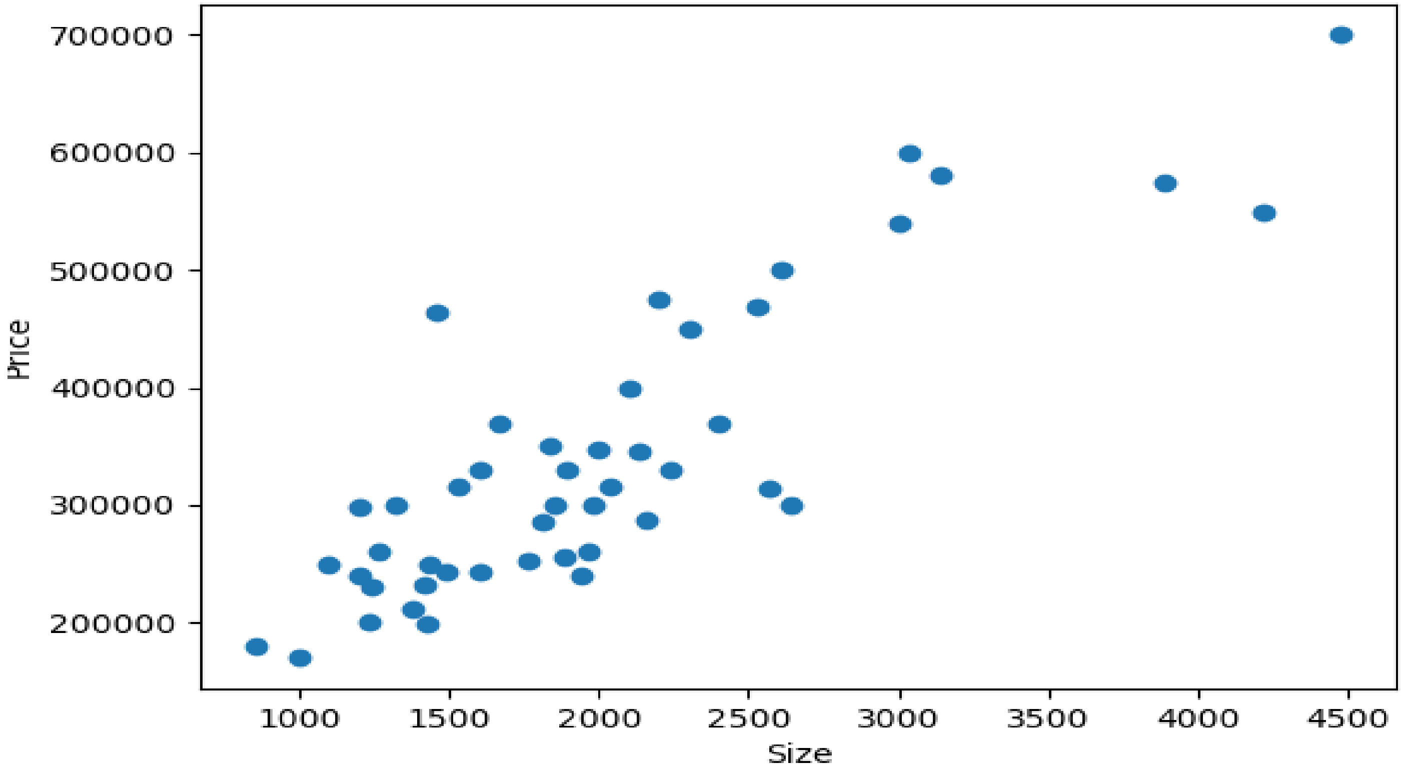


One Variable Linear Regression: Example

Assume we have a dataset giving the living areas and prices of 47 houses from Portland, Oregon:

Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

One Variable Linear Regression: Example



Training/Test Sets

- In each house, we have living area (**feature**) and price (label)
- The previous dataset has given labels; thus we call it **training set**.
- If the dataset does **not** have labels, we call it **test set**

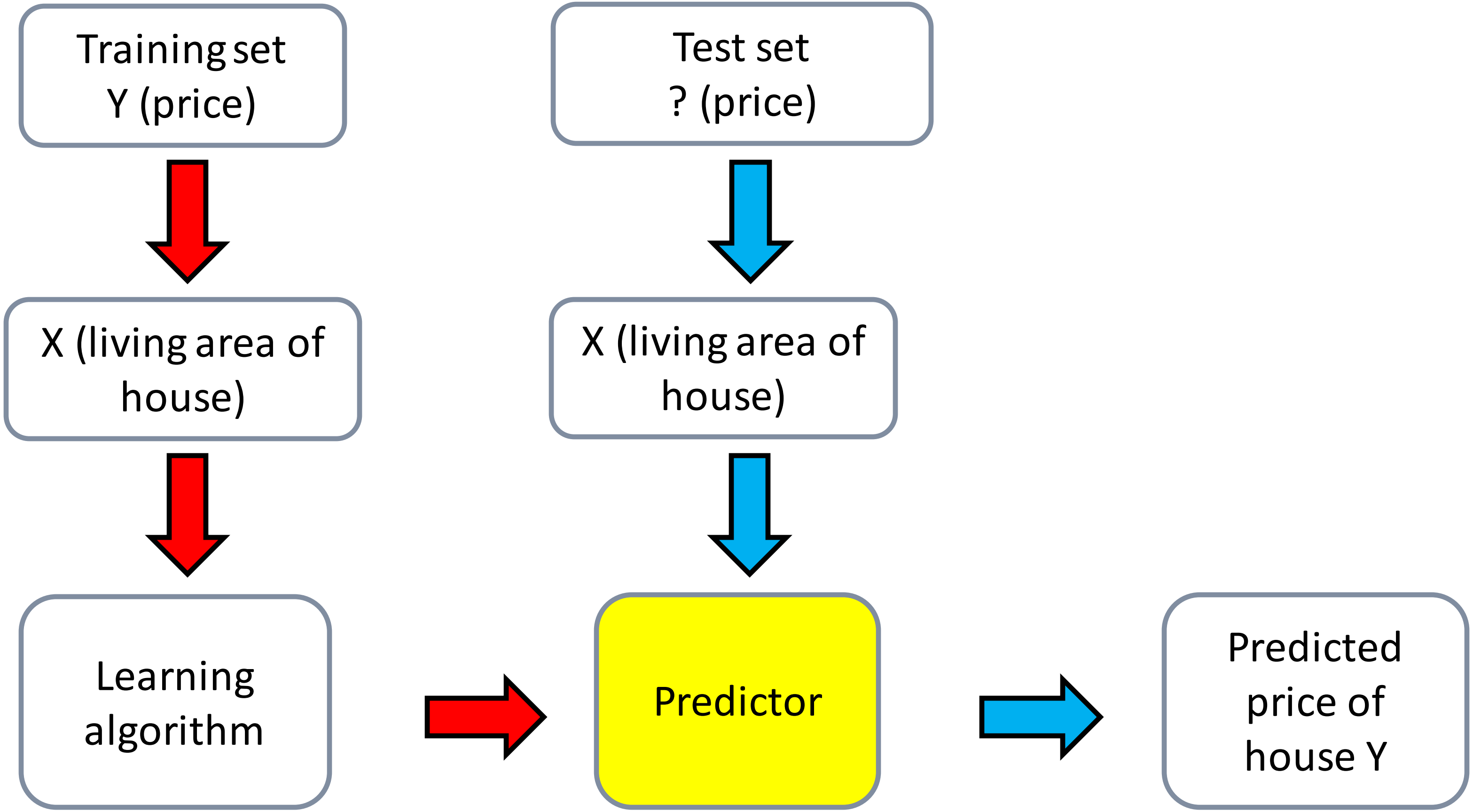
Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

Test set

- If we are given a size of living area in a house , What is the estimated price of that house?

Living area	Estimated Price
1300	?
4000	?
2200	?
2000	?

Model Representation



Predictor and Loss Function

- We assume a predictor that is linear in model parameter (c_0, c_1):

$$p(x) = c_0 + c_1 x$$

- We choose c_0, c_1 such that they minimize the following **loss function**

$$L(c_0, c_1) = \sum_{i=1}^{M} \left(p\left(x^{(i)}\right) - y^{(i)}\right)^2 = \|\mathbf{P} - \mathbf{Y}\|_2^2$$

where: $\mathbf{P} = \left(p\left(x^{(1)}\right), p\left(x^{(2)}\right), \dots, p\left(x^{(M)}\right)\right)^T$

$$\mathbf{Y} = \left(y^{(1)}, y^{(2)}, \dots, y^{(M)}\right)^T$$

Minimizing Loss Function

In the dataset, $x^{(i)}$ and $y^{(i)}$ are, respectively, the living area and price of the i^{th} house. And $M = 45$

$$\min_{c_0, c_1} : L(c_0, c_1) = \sum_{i=1}^{M} \left(p\left(x^{(i)}\right) - y^{(i)}\right)^2$$

is known as the **least-square linear regression problem.**

Minimizing Loss Function

There are two approaches to solve least-square linear regression problems

1. Use Calculus I technique to find minima point
2. Use Projection Matrix in Linear Algebra

Minimizing Loss Function

1. Use gradient to determine minimum value

The optimal values of c_0, c_1 are:

- $\frac{\partial L}{\partial c_j} = 0, j = 0,1 =>$

$$\widehat{c_1} = \frac{\sum_{i=1}^{M} x^{(i)} y^{(i)} - \frac{1}{M} \sum_{i=1}^{M} x^{(i)} \sum_{i=1}^{M} y^{(i)}}{\sum_{i=1}^{M} (x^{(i)})^2 - \frac{1}{M} \left(\sum_{i=1}^{M} x^{(i)}\right)^2}$$

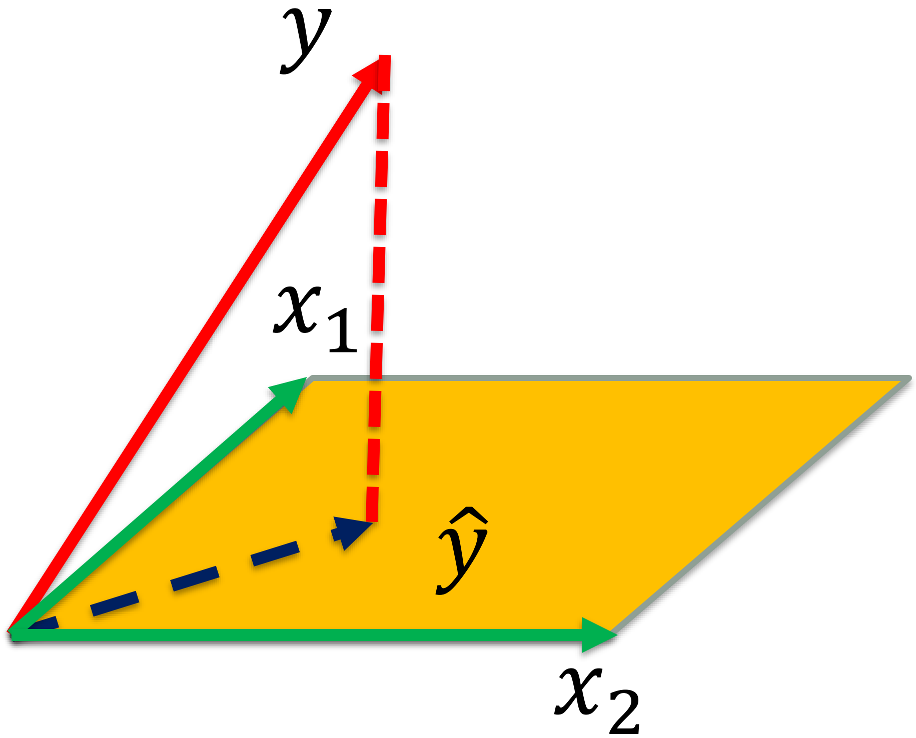
$$\widehat{c_0} = \frac{1}{M} \sum_{i=1}^{M} y^{(i)} - \widehat{c_1} \frac{1}{M} \sum_{i=1}^{M} x^{(i)}$$

2. Projection matrix

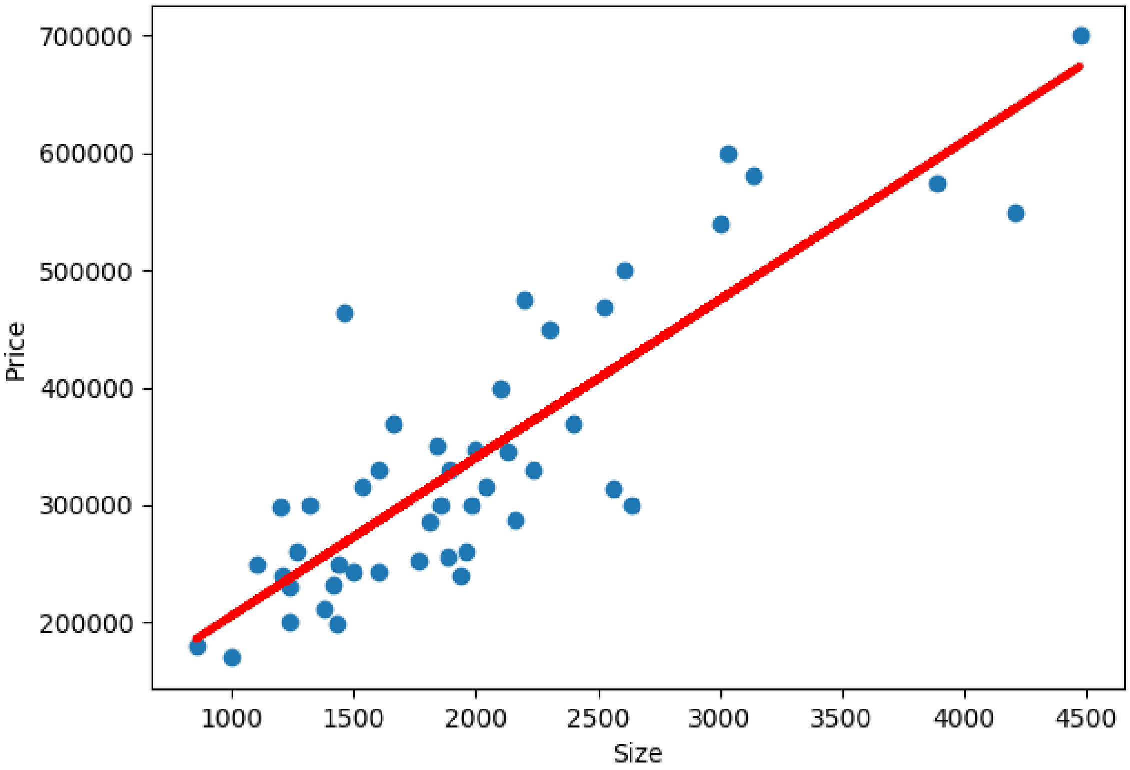
$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

where $\mathbf{X} = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \\ & \cdots \\ 1 & x^{(M)} \end{bmatrix}$,

and $\mathbf{Y} = (y^{(1)}, y^{(2)}, \ldots, y^{(M)})$



Result



700000
600000
500000
400000
300000
200000
Price
1000
1500
2000
2500
3000
3500
4000
4500
Size

Multiple Variables Linear Regression: Example

- Used when having multiple features
- In the housing example, consider a richer dataset with knowing the number of bedrooms in each house

x_1 Living area (feet^2)	x_2 #bedrooms	y Price (1000\$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

Predictor and Loss Function

- We assume our predictor:

$$p(x) = c_0 + c_1 x_1 + c_2 x_2$$

- Find c_0, c_1, c_2 to optimize the loss function:

$$L(c_0, c_1, c_2) = \sum_{i=1}^{M} \left(p\left(x_1^{(i)}, x_2^{(i)}\right) - y^{(i)} \right)^2$$

- Gradient and Projection Matrix can be used for the multivariable case.
- But Projection Matrix approach is preferred?

Minimizing Loss Function

- Solution of the optimization problem is $\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$

where $\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \\ & \dots & \\ 1 & x_1^{(M)} & x_2^{(M)} \end{bmatrix}$, and

$$\mathbf{Y} = (y^{(1)}, y^{(2)}, \dots, y^{(M)})$$

General linear regression model

- In general, we assume our predictor:

$$p(x) = c_0 + c_1 x_1 + \cdots + c_n x_n$$

Find $c_0, c_1, \ldots, c_n$ to optimize the loss function:

$$L(c_0, c_1, \ldots, c_n) = \sum_{i=1}^{M} \left(p\left(x_1^{(i)}, \ldots, x_n^{(i)}\right) - y^{(i)} \right)^2$$

$$\begin{bmatrix} c_0 \\ c_1 \\ \ldots \\ c_n \end{bmatrix} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

General linear regression model

- Solution of the optimization problem is:

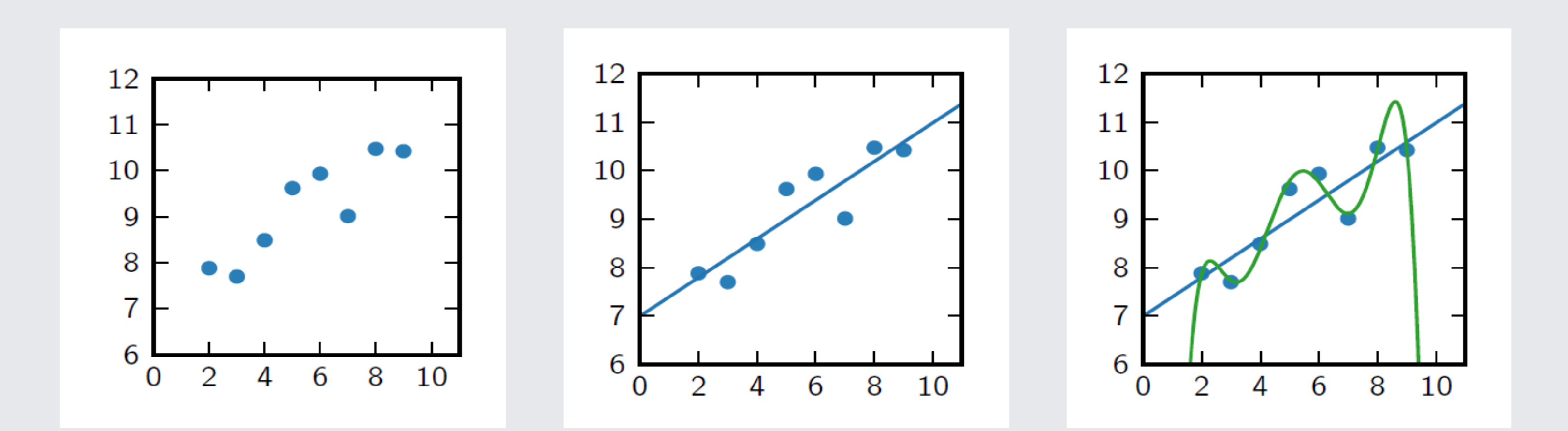
$$\begin{bmatrix} c_0 \\ c_1 \\ \dots \\ c_n \end{bmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

where $\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ & & \dots & \\ 1 & x_1^{(M)} & \dots & x_2^{(M)} \end{bmatrix}$, and

$$\mathbf{Y} = (y^{(1)}, y^{(2)}, \dots, y^{(M)})$$

Discussions: Overfitting & linearity

- A model leads to overfitting when it perfectly fits the training data but poorly fits the test data



- Linear regression is about the linearity with respect to c not $\mathbf{X}$

Discussions: Loss Function minimization with L1 and L2 norms

L1: $$\min_{c_0,c_1} : L(c_0, c_1) = \sum_{i=1}^{M} \left| p\left(x^{(i)}\right) - y^{(i)} \right|$$

Least Squares Regression	Least Absolute Deviations Regression
Not very robust	Robust
Stable solution	Unstable solution
Always one solution	Possibly multiple solutions
No feature selection	Built-in feature selection
Non-sparse outputs	Sparse outputs
Computational efficient due to having analytical solutions	Computational inefficient on non-sparse cases

Sklearn Library

- Linear Regression model is coded in sklearn python library
- Reference: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- Load Linear Regression model

  from sklearn. linear_model import LinearRegression

- Parameters discussion
  - fit_intercept: use the free coefficient, very often, turn it on will improve the performance.
  - normalize: to standardize features, will explain further in upcoming lectures, have to turn it on.

Sklearn Library

- Output attributes
  - coef_: if $y = c_0 + c_1x_1 + c_2x_2$ then coef_ is an array of $[c_1, c_2]$
  - intercept_: use the above formulation as an example, intercept_ will return a bias c_0