THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

AMA 2233

Data Analytics and Visualization

Yancheng Yuan

Department of Data Science and Artificial Intelligence

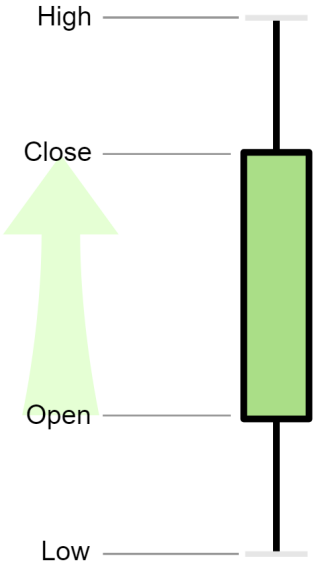
Opening Minds • Shaping the Future
啟迪思維 • 成就未來

Recap of last lecture

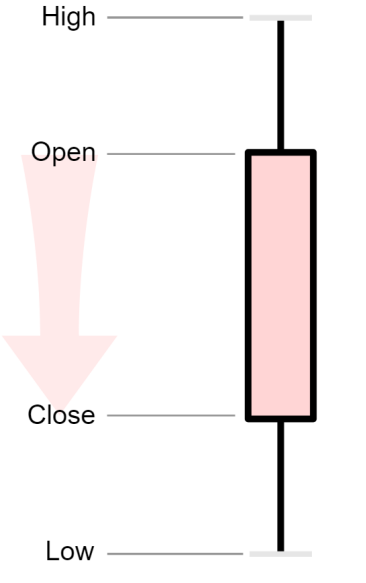
Candlestick Chart

Candlestick Chart: Candlestick represents four important pieces of information for that day: **open** and **close** in the thick body, and **high** and **low** in the "**candle wick**".

Increasing :
Bullish Candle Stick



Decreasing :
Bearish Candle Stick



An example


200
190
180
170
160
150
140
130
Price
5.0
2.5
0.0
−2.5
−5.0
150
100
50
Volume 10^6
2022-Oct-03
2022-Dec-13
2023-Feb-27
2023-May-09
2023-Jul-21
2023-Oct-02

Moving Average

Moving Average (MA): a moving average (rolling average or running average) is a calculation to analyze data points by creating a series of different moving average is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles. lections of the full data set.

Reason for MA: A moving average is commonly used with time series data to **smooth out short-term fluctuations and highlight longer-term trends or cycles**.

Low Pass Filter: Let $\mathrm{x} = \{\dots, x_{t-s}, x_{t-s+1}, \dots, x_{t-1}, x_t, x_{t+1}, \cdots\}$ be a given time series and $w = \{\dots, w_{t-s}, w_{t-s+1}, \dots, w_{t-1}, w_t, w_{t+1}, \cdots\}$ be a filter (some specified weights), then the low frequency signal of the time series x under the filter F is

$$y_t = \sum_{i=-\infty}^{+\infty} w_i x_{t-i}$$

Delay Effect of Simple Moving Average

Moving Average in the time window T: Since we will not have infinite number of data points in practice, we usually consider the average in a given time window T:

$$y_t = \sum_{i=-(T-1)/2}^{(T-1)/2} w_i x_{t-i}$$

No Future Data: Since we do not have future data after time t, the above formula is not practical in practice. Instead, we use the following formula:

$$\hat{y}_t = \sum_{i=0}^{T-1} w_i x_{t-i}$$

Delay Effect of Simple Moving Average

Delay effect of Simple Moving Average: We call it a simple moving average if we compute $\hat{y}_t$ as

$$\hat{y}_t = \frac{\sum_{i=0}^{T-1} x_{t-i}}{T}$$

Time Delay:

$$\hat{y}_t = \frac{\sum_{i=0}^{T-1} x_{t-i}}{T} = \frac{\sum_{j=-(T-1)/2}^{(T-1)/2} x_{t-(T-1)/2-j}}{T} = y_{t-(T-1)/2}$$

Linearly Decay Weights:

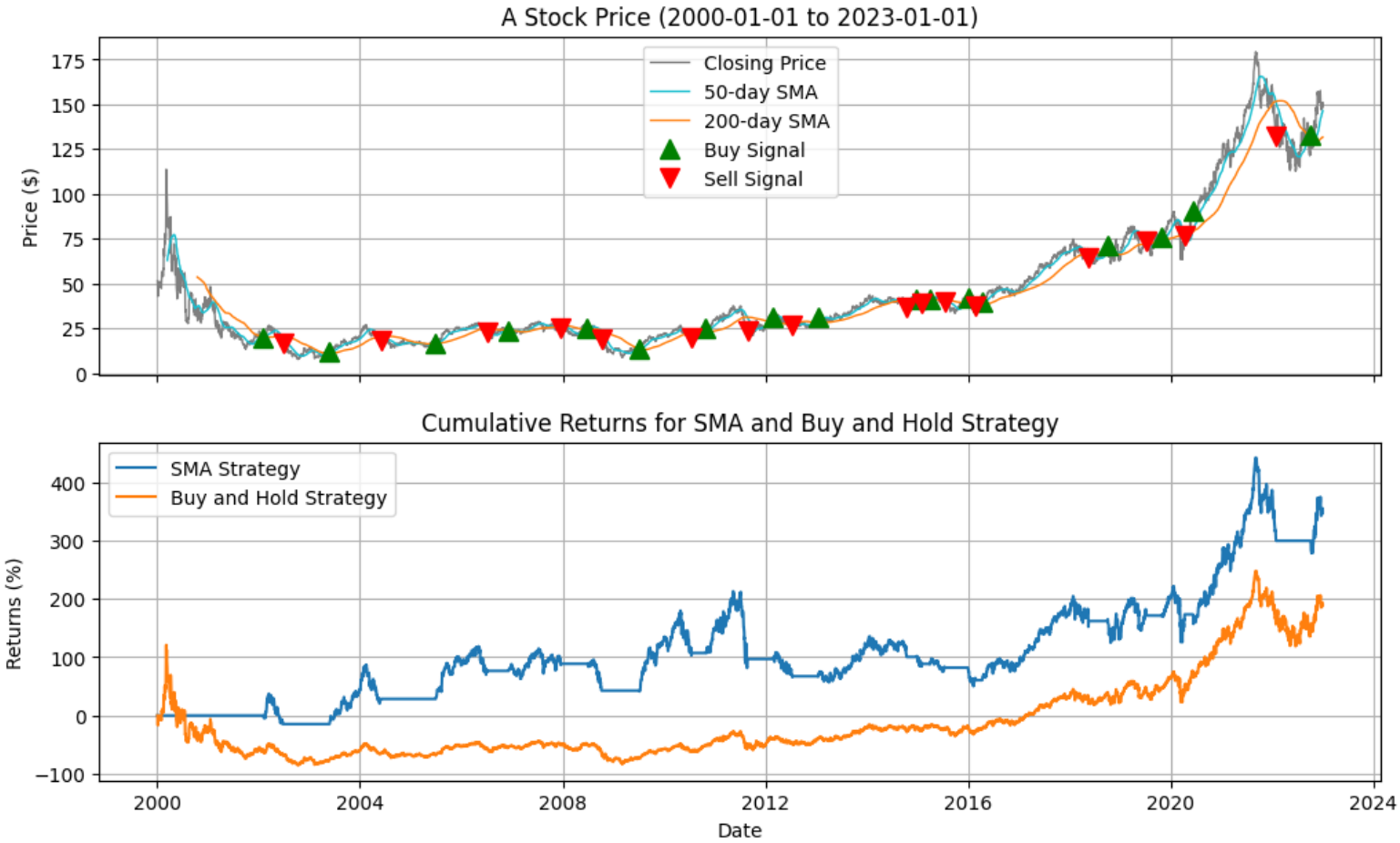
$$\hat{y}_t = (x_{t-T+1} + 2x_{t-T+2} + \cdots + Tx_t)/T(T+1)/2$$

Exponentially Decay Weights:

$$\hat{y}_0 = \alpha x_0, \hat{y}_t = \alpha x_t + (1 - \alpha)\hat{y}_{t-1}.$$

A typical choice for α is $\alpha = \frac{2}{T+1}$ if we use a T time window.

Backtesting SMA Based Trend Trading on AAPL



Moving Average Convergence/Divergence (MACD)

MACD Line: MACD is a trend-following momentum indicator that shows the relationship between two exponential moving averages (EMAs). The MACD line is calculated by subtracting the 26-period EMA from the 12-period EMA.

Signal Line: 9-day EMA of the MACD line.

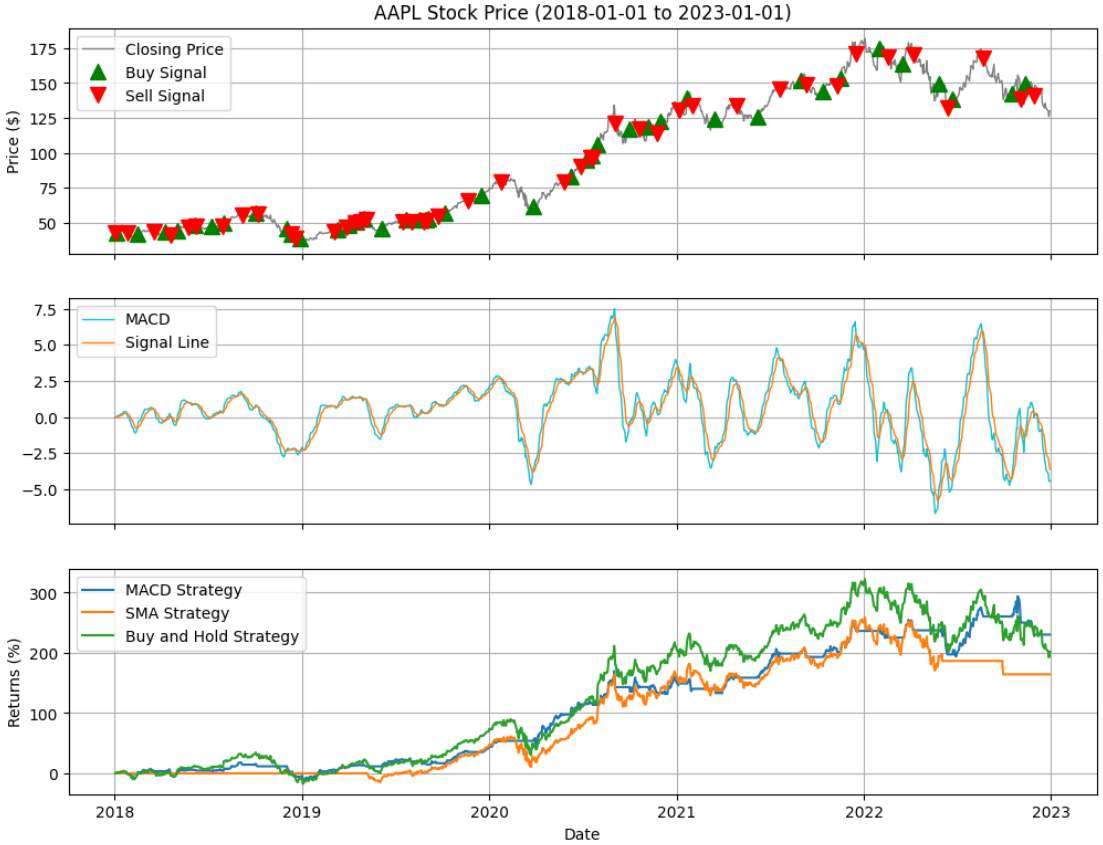
Trading Strategy:

Buy stocks with all cash when a **MACD Line crosses above the Signal Line**.

Sell all stocks when a **MACD Line crosses below the Signal Line**.


200
190
180
170
160
150
140
130
Price
5.0
2.5
0.0
−2.5
−5.0
150
100
50
Volume 10^6
2022-Oct-03
2022-Dec-13
2023-Feb-27
2023-May-09
2023-Jul-21
2023-Oct-02

Backtesting



Agenda



1. Linear Regression
2. Gradient descent algorithm
3. Overfitting and regularization

Linear Regression

Regression Analysis

Regression analysis (Wiki) is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors').

Example (Sales prediction) Build a model which predicts sales based on the money spent on different platforms for marketing.

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

Linear Regression

Linear Regression using a weighted sum of the independent variables as a predictor to approximate the independent variable.

$$y \approx h(x) = w_0 + w_1 x_1 + \cdots + w_n x_n.$$

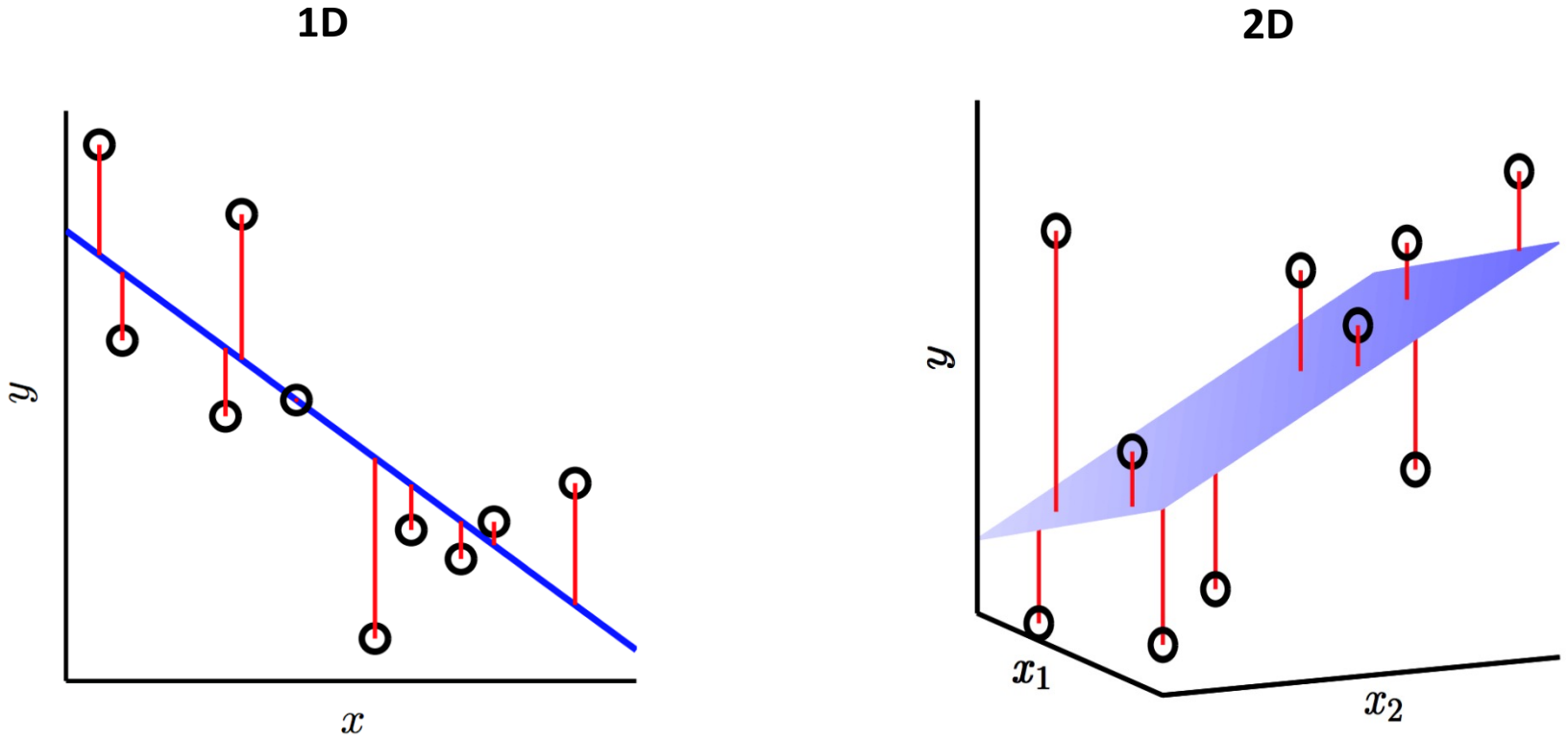
If we denote

$$x = (1, x_1, \ldots, x_n)^T \text{ and } w = (w_0, w_1, \ldots, w_n)^T,$$

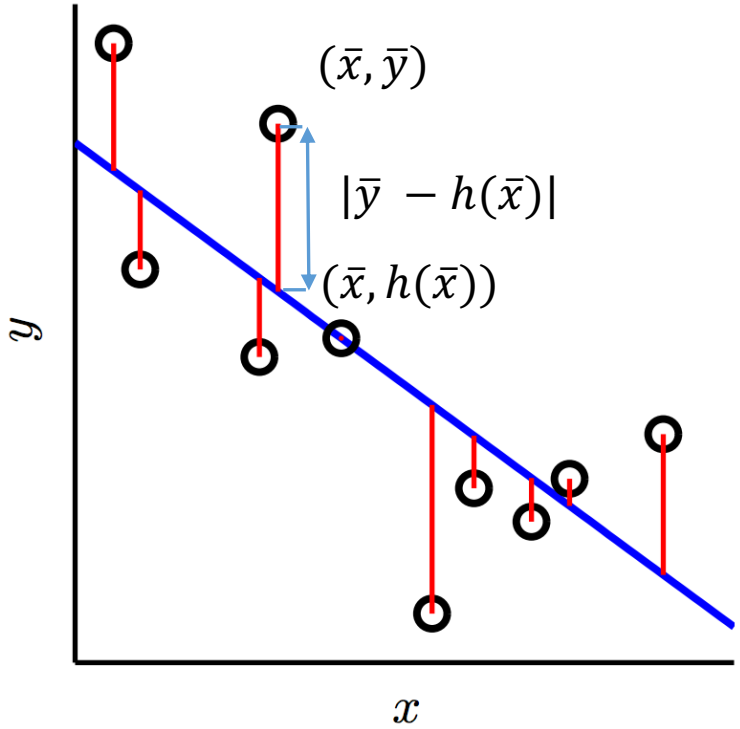
we can write

$$h(x) = w^T x.$$

Linear Regression



How to Get a Better Model? (Error Measure)



How to Get a Better Model? (Error Measure)

Mean Squared Error (MSE): If we have a data set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, the mean squared error of the linear predictor on the data set is defined as

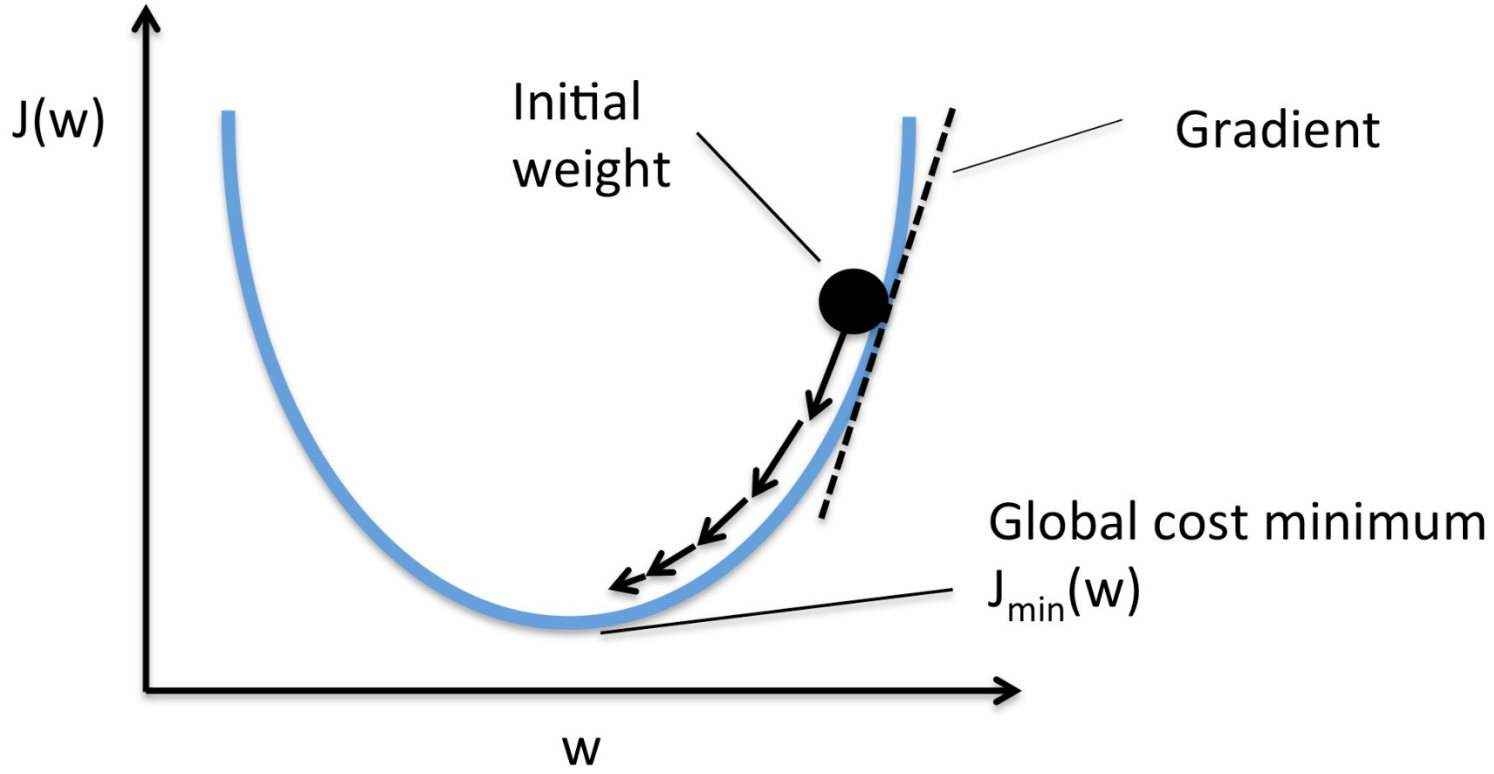
$$J(w) = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - h(x^{(i)})\right)^2$$

Gradient Descent Algorithm

How to Get a Better Model? (Model Training)



How to Get a Better Model? (Model Training)



Gradient Descent Algorithm

Start from some $w^0 \in \mathbb{R}^{n+1}$, gradient descent (GD) algorithm updates as:

$$w^{k+1} = w^k - \alpha_k \nabla J(w^k),$$

until

$$||\nabla J(w^{k+1})|| \leq \varepsilon,$$

for some tolerance $\varepsilon > 0$.

Key points:

1. Compute $\boldsymbol{\nabla J(w^k)}$.
2. Choose step size $\alpha_k > 0$ satisfying

$$\mathbf{J}(\boldsymbol{w^{k+1}}) < \boldsymbol{J(w^k)}.$$

Compute the Gradient of J(w)

Denote

$$X = \left(x^{(1)}, x^{(2)}, \ldots, x^{(m)}\right)^T \text{ and } y = \left(y^{(1)}, y^{(2)}, \ldots, y^{(m)}\right)^T,$$

we can rewrite

$$J(w) = \frac{1}{m}||y - Xw||^2$$

Gradient of J(w):

$$\nabla J(w) = \frac{2}{m} X^T (Xw - y)$$

Analysis of GD Algorithm

Assumption 1

$\mathbf{J}(\boldsymbol{w})$ is continuously differentiable and for any w^1, w^2, we have

$$J(w^1) \leq J(w^2) + \langle \nabla J(w^2), w^1 - w^2 \rangle + \frac{L}{2}||w^1 - w^2||^2.$$

for some L > 0. We call $\boldsymbol{J}$ satisfying this property is a **L-smooth function**.

Analysis of GD Algorithm

Lemma 1 If we choose $\alpha_k = \frac{1}{L}$, then the sequence $\{w^k\}$ generated by the gradient descent algorithm satisfies

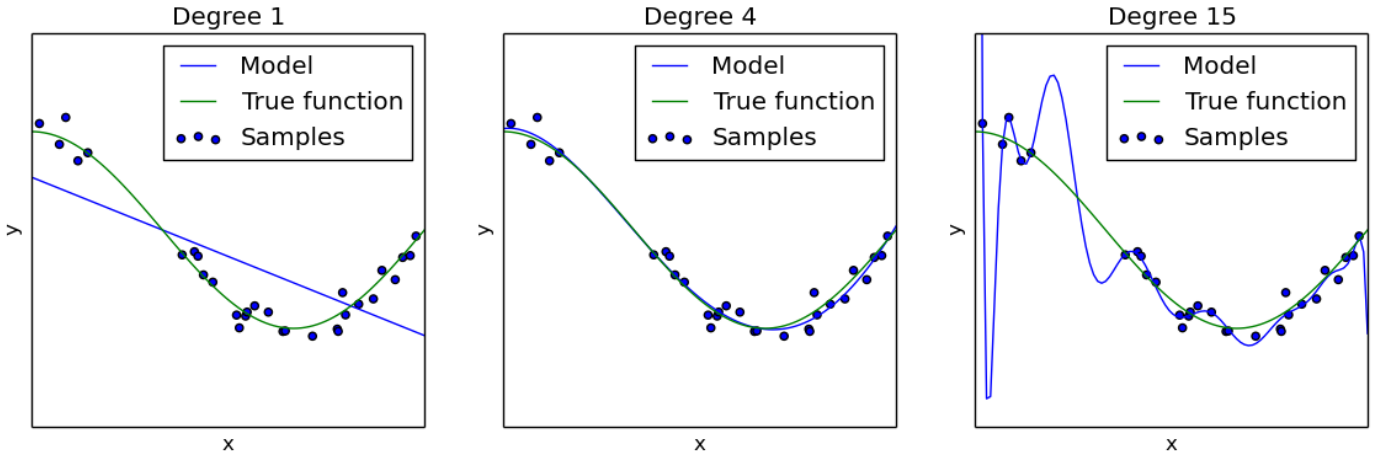
$$J\left(w^{k+1}\right) \leq J\left(w^k\right) - \frac{1}{2L}||\nabla J\left(w^k\right)||^2.$$

Remark: As a consequence of Lemma 1, if the function J is bounded from below, gradient descent algorithm can converge to a point w^* such that

$$\nabla J(w^*) = 0.$$

Overfitting

Overfitting



Regularization

A potential issue for the model overfitting is that the model is **too complicated**. One idea to prevent overfitting is to reduce the model complexity by adding regularization.

L2 Regularization (Ridge Regression)

$$J(\boldsymbol{w}) + \frac{\boldsymbol{\lambda}}{\boldsymbol{2}} ||\boldsymbol{w}||^{\boldsymbol{2}}$$

L1 Regularization (Lasso Regression)

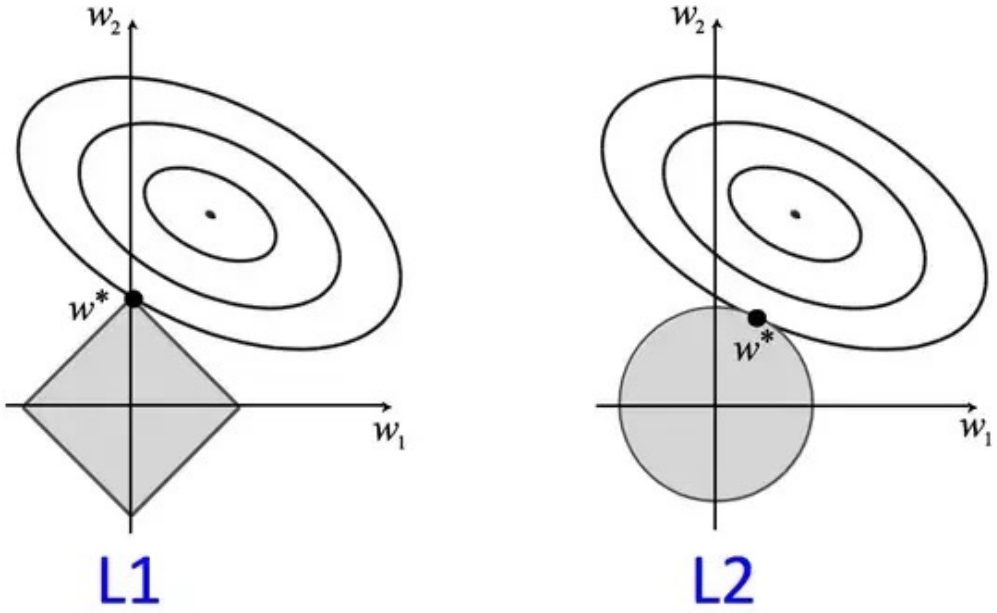
$$J(\boldsymbol{w}) + \boldsymbol{\lambda}||\boldsymbol{w}||_{\boldsymbol{1}}$$

Apply gradient descent algorithm to ridge regression, we have

$$w^{k+1} = w^k - \alpha\left(\nabla \mathbf{J}\left(\mathbf{w^k}\right) + \lambda w^k\right) = (1 - \alpha\lambda)w^k - \alpha\nabla \mathbf{J}(\mathbf{w^k})$$

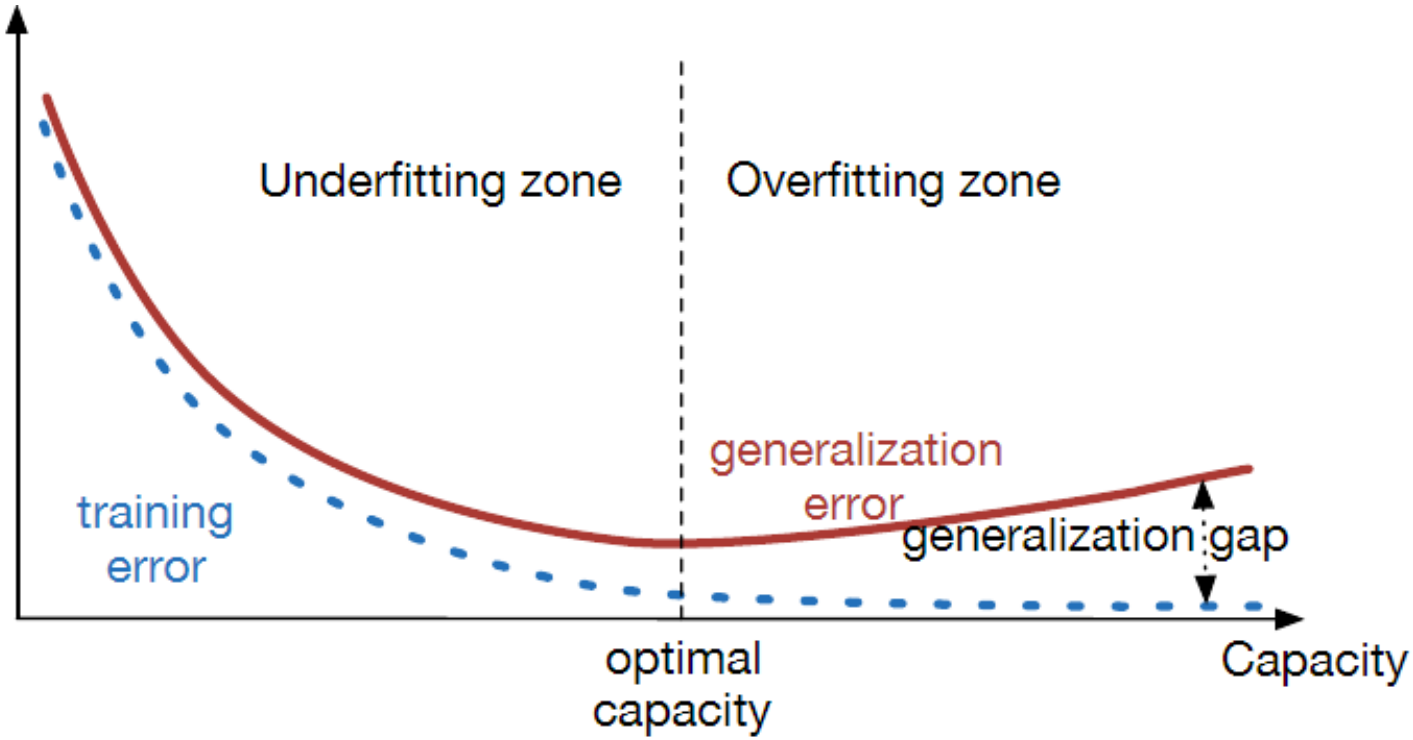
Remark: Adding an L2 regularization will shrink w^k in the gradient descent algorithm.

Regularization

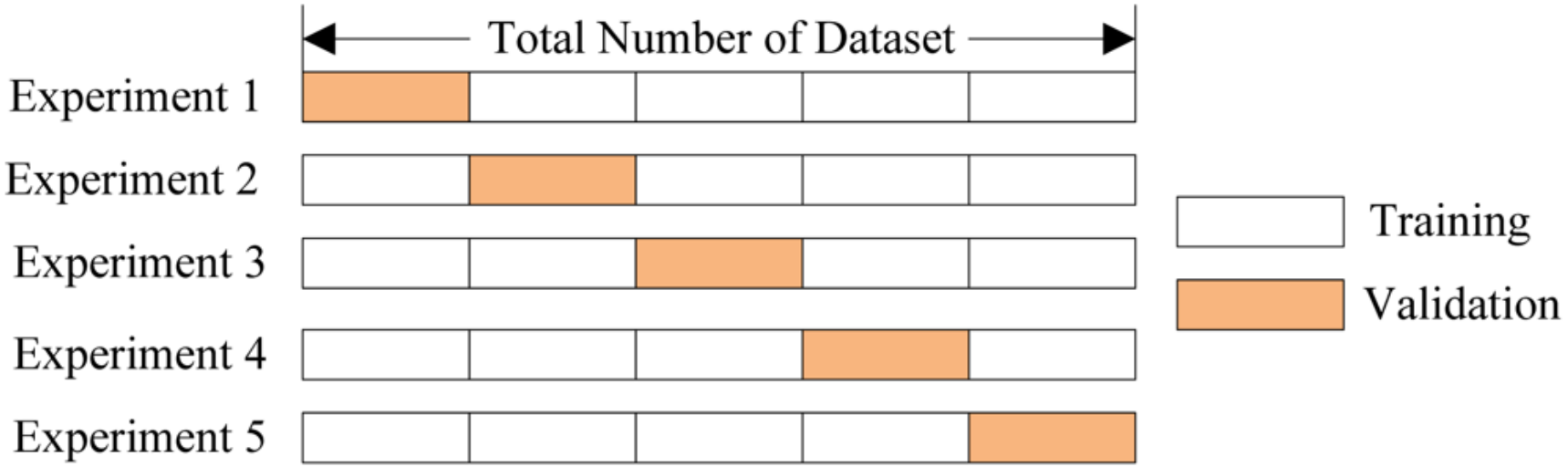


Remark: Adding an L1 regularization will induce sparsity of the parameter w and reduce the model complexity.

Regularization: Early Stopping



Model Selection: Cross Validation



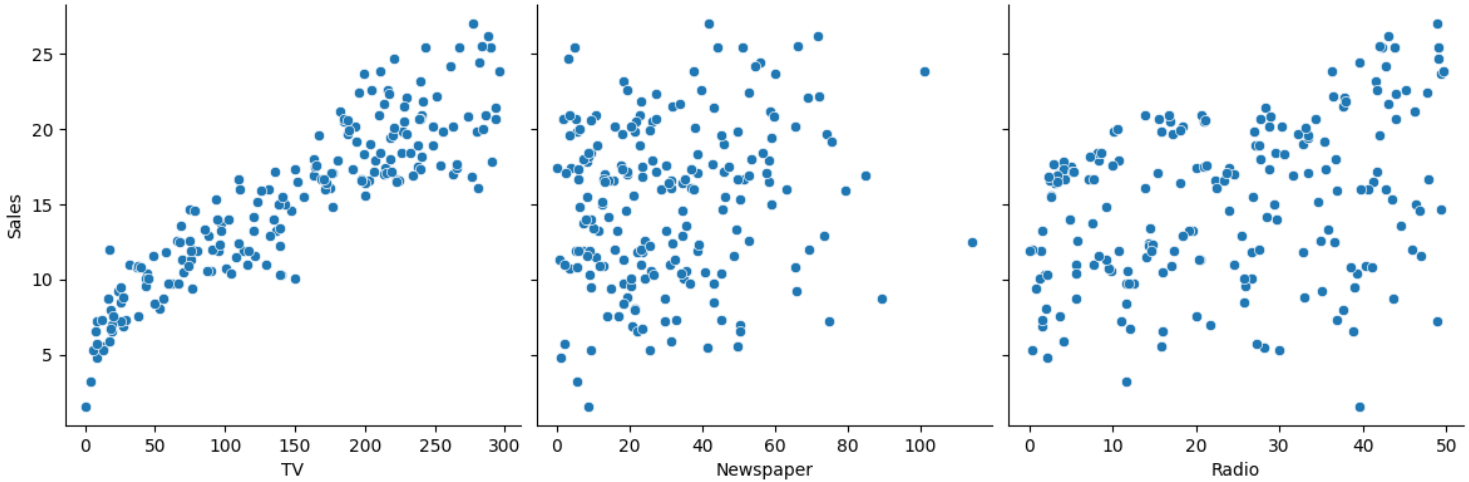
Linear Regression on Sales Prediction

Sales Prediction Problem

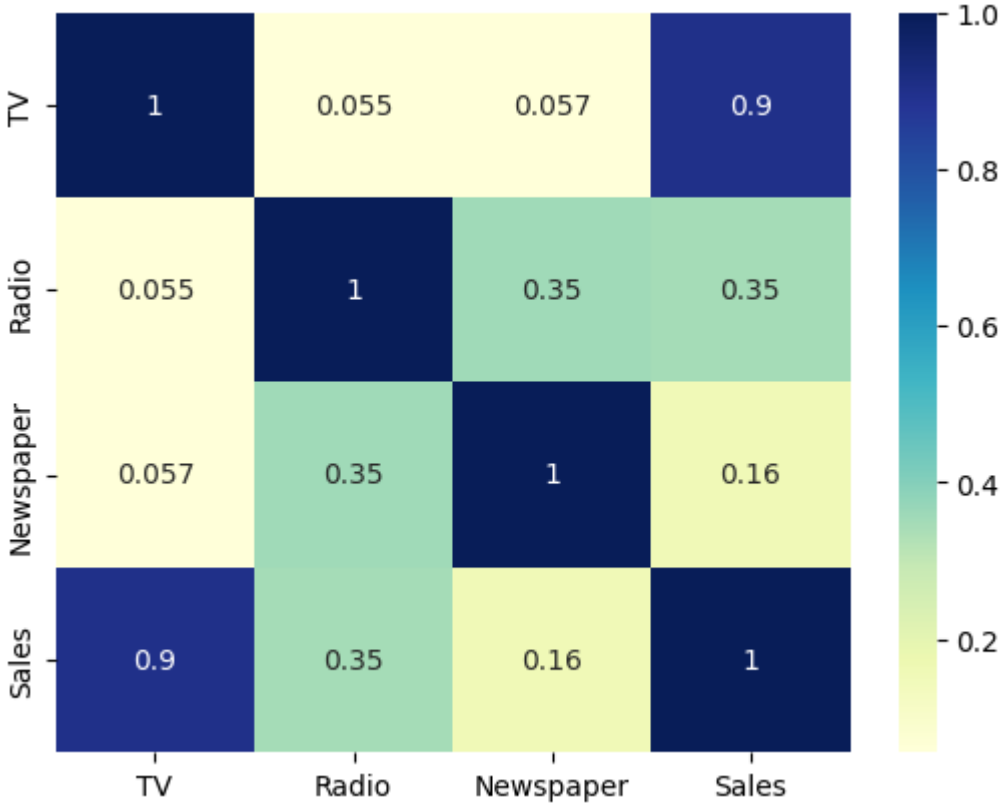
Example (Sales prediction) Build a model which predicts sales based on the money spent on different platforms for marketing.

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

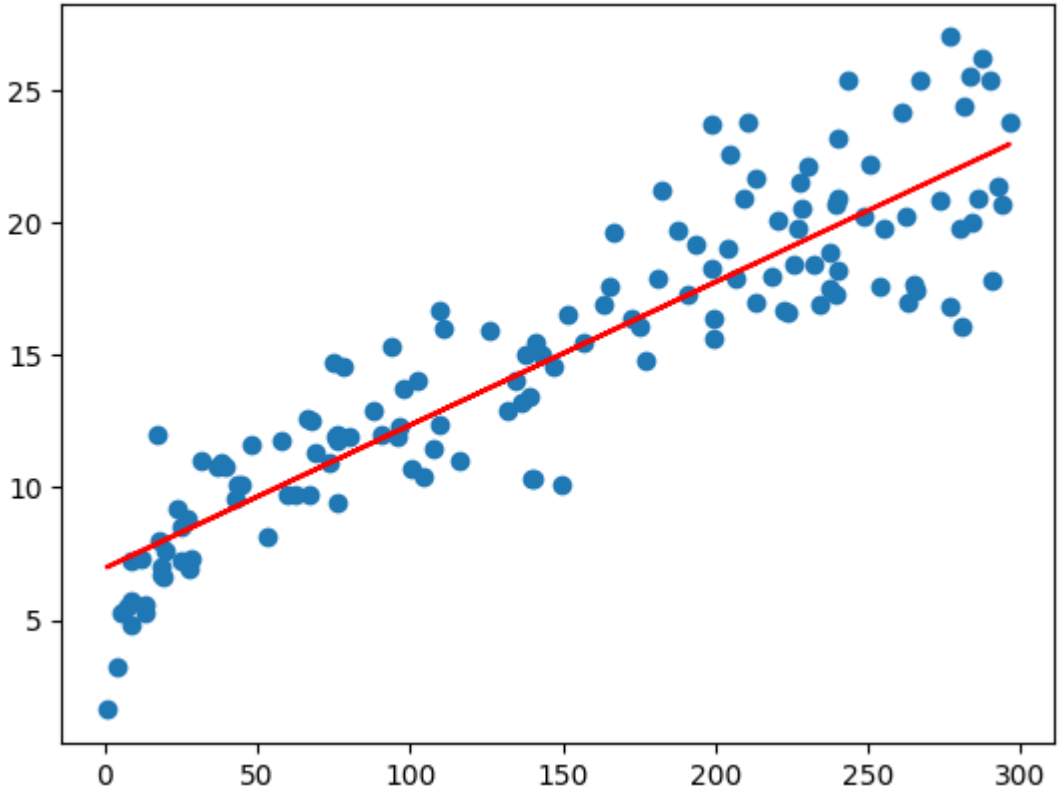
Univariate Feature Visualization



Feature Correlation Test

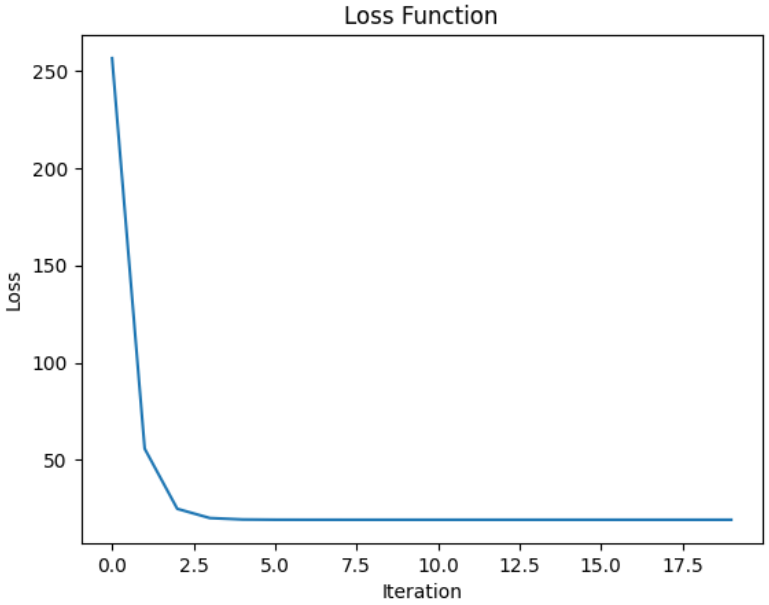


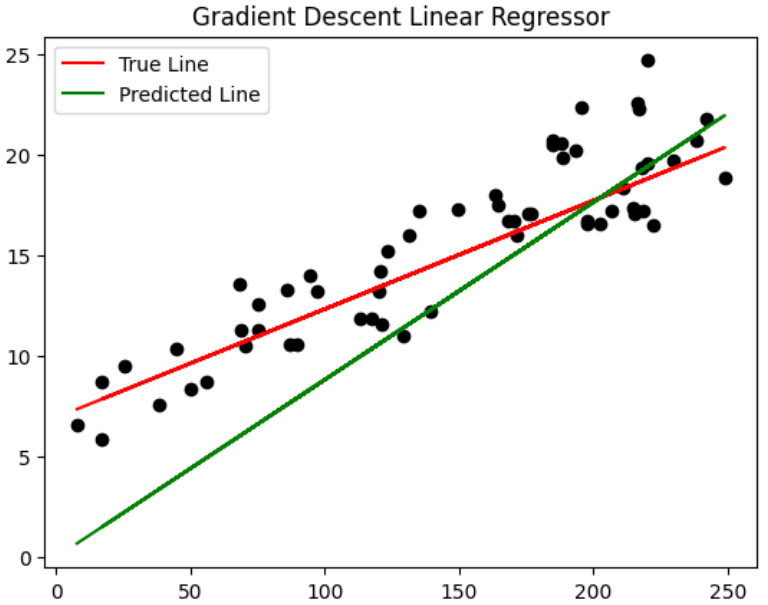
A simple linear regression model with only TV marketing



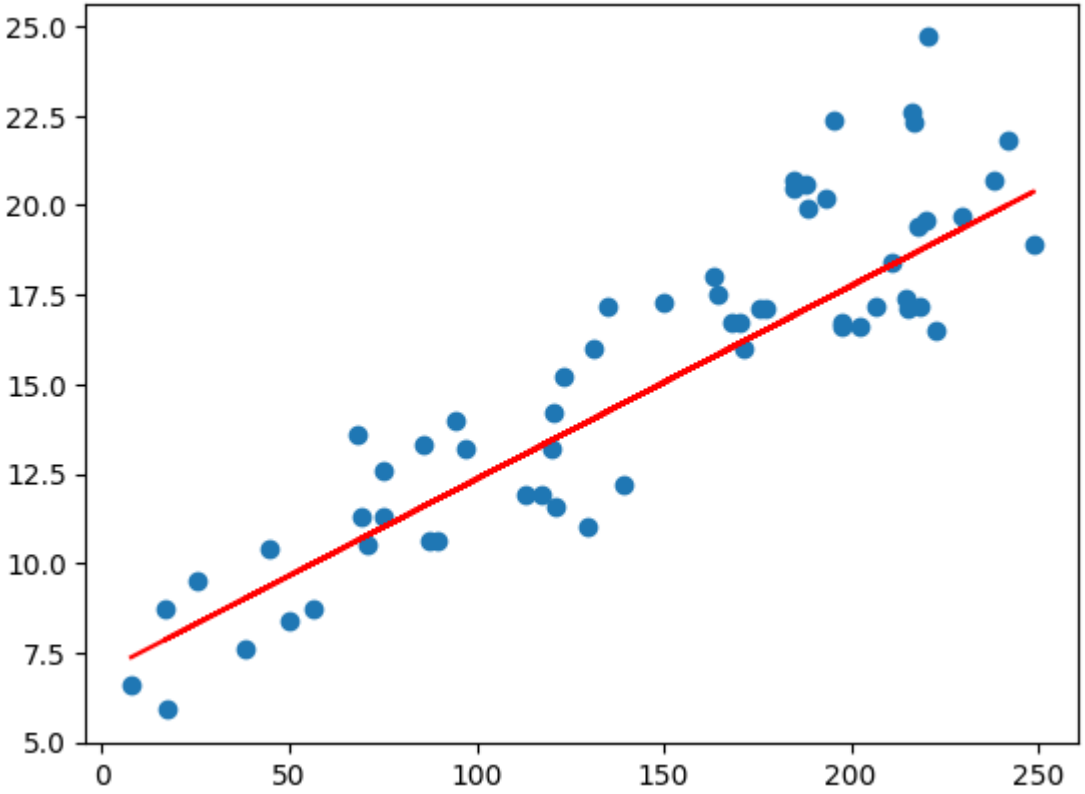
$$\hat{y} = 6.984 + 0.054x_1$$

Training


Loss Function
Loss
Iteration
250
200
150
100
50
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Gradient Descent Linear Regressor
True Line
Predicted Line
25
20
15
10
5
0
0
50
100
150
200
250

Test




Thank You!