Python can do $10^7$ calculations in 1 sec

$$\hat{\theta}(\text{best}) = \left[(X^T \cdot X)^{-1} \cdot X^T \cdot Y\right]^T$$

Biggest calculation is = $\underbrace{(X^T \cdot X)^{-1}}$

$(n \times n)$ $\boxed{n^{2.4}}$ secs

$(100)^3$
$= \frac{10^6}{10^7}$
$= \underline{1 sec}$

$$(1000)^3 = \frac{(10)^9}{10^7} \sec = \boxed{100}$$

$n^3$

This means that = for a data set of 1000 rows & 1 feature will take 100 seconds to solve by normal equation method.

(10) (1) $\cancel{(10,000)^3} = \frac{10^{12}}{10^7}$ se

$= 10^5$ secs.

27.7 Hours

Normal equation method fails miserablly

$$y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \ldots + \theta_n x_n$$

$\boxed{MSE}$ = we will fix all parameter except one & see the charge in result

$$\frac{\partial}{\partial \theta_j} \left( MSE(\theta) \right) = \frac{2}{m} \sum_{i=1}^{m} \left( \left( \theta^T \cdot x^{(i)} \right) - y \right) x^{(i)}$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y} - y)^2$$

$$= \frac{1}{m}\sum_{i=1}^{m}(\underline{\theta \cdot \cancel{x}} - y)^2$$

$[\ ]$

$[\ x^T]$

$$x^3 = 3x^2$$

$$\frac{\partial}{\partial\theta}(MSE(\theta)) = \frac{2}{m}\sum_{i=1}^{m}(\theta^T x - y)(x - )$$

$$= \frac{2}{m}\sum_{i=1}^{m}(\theta x - y)x$$

$\theta = .01$

$\theta_2 = .1$

MSE = Somethis

MSE = Som

$$\theta_{next} = \theta - \eta \nabla MSE(\theta)$$

Learning rate