Machine Learning with Python

- Normal Equation
- Features Scaling

Dr. Sherif Eletriby

Lecture_4

Gradient Descent

- In mathematics gradient descent (also often called steepest descent) is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function.
- The idea is to take repeated steps in the opposite direction of the gradient (or approximate gradient) of the function at the current point, because this is the direction of steepest descent. Conversely, stepping in the direction of the gradient will lead to a local maximum of that function; the procedure is then known as gradient ascent.

Normal Equation :

- Normal Equation, an analytical approach used for optimization. It is an alternative for Gradient descent. Normal equation performs minimization without iteration.
  Normal equations are equations obtained by setting equal to zero the partial derivatives of the sum of squared errors or cost function; normal equations allow one to estimate the parameters of multiple linear regression.

Normal equation

- Another algorithm that solve for the optimal value of θ.

$$\frac{\partial}{\partial \theta_j} J(\theta) = 0$$

$$\boxed{\theta = (X^T . X)^{-1} . X^T . y}$$

Where:
X = input features value
y = output value

- One analytical step
- No need to choose learning rate.

Gradient Descent

- An optimization algorithm used to find the optimal values of θ.

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

$$\boxed{\theta_j := \theta_j - \gamma \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)}\right) . x_j^{(i)}}$$

- Iterative process
- We need to choose learning rate.

Example:

$n = 4$

Size	Room#	Age	Floor	Price
90	3	3	1	50
120	5	15	3	70
150	4	10	2	100
200	6	5	4	150

$m = 4$

$$\underset{m \times n}{X} = \begin{array}{ccccc} x_0 & x_1 & x_2 & x_3 & x_4 \\ \end{array} \begin{bmatrix} 1 & 90 & 3 & 3 & 1 \\ 1 & 120 & 5 & 15 & 3 \\ 1 & 150 & 4 & 10 & 2 \\ 1 & 200 & 6 & 5 & 4 \end{bmatrix} \qquad \underset{m}{y} = \begin{bmatrix} 50 \\ 70 \\ 100 \\ 150 \end{bmatrix}$$

Example:
Let's take the feature **Room#** to predict **Price**

$$\boxed{\boldsymbol{\theta = (X^T.X)^{-1}.X^T.y}}$$

$$h(x) = \theta_0 x_0 + \theta_1 x_1$$

Step1

$$X = \begin{array}{cc} x_0 & x_1 \end{array}$$

$$X = \begin{bmatrix} 1 & 3 \\ 1 & 5 \\ 1 & 4 \\ 1 & 6 \end{bmatrix} \qquad y = \begin{bmatrix} 50 \\ 70 \\ 100 \\ 150 \end{bmatrix}$$

Step2

$$X^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 5 & 4 & 6 \end{bmatrix}$$

Step3

$$X^T.X = \begin{bmatrix} 4 & 18 \\ 18 & 86 \end{bmatrix}$$

Step4

$$(X^T.X)^{-1} = \begin{bmatrix} 4.3 & -.9 \\ -0.9 & 0.2 \end{bmatrix}$$

Step5

$$(X^T.X)^{-1}.X^T = \begin{bmatrix} 1.6 & -0.2 & 0.7 & -1.1 \\ -0.3 & 0.1 & -0.1 & 0.3 \end{bmatrix}$$

Step6

$$(X^T.X)^{-1}.X^T.y = \begin{bmatrix} -29 \\ 27 \end{bmatrix} \begin{array}{l} \leftarrow \theta_0 \\ \leftarrow \theta_1 \end{array}$$

$$\boxed{\boldsymbol{h(x) = -29x_0 + 27x_1}}$$

Room#	Gradient Descent	Normal Equation	Actual Price
	$y = -28x_0 + 27x_1$	$y = -29x_0 + 27x_1$	
3	53	52	50
5	107	106	70
4	80	79	100
6	134	133	150

- Works well and faster with Small features set
- Singular/non invertible matrix
- No need for **features scaling**

PART 2

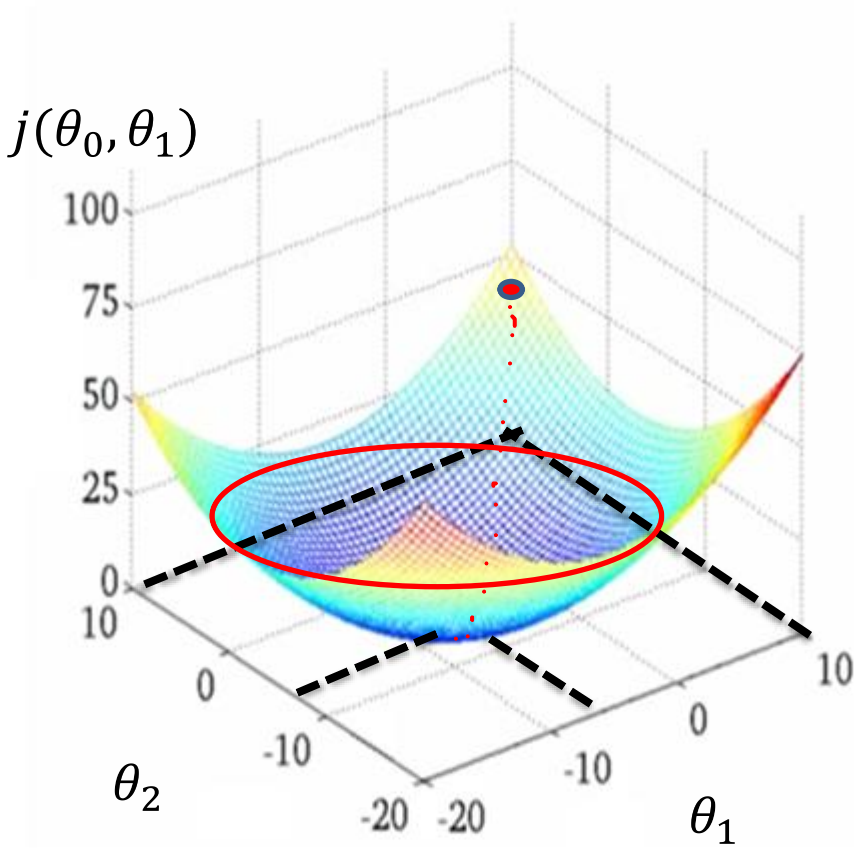
Features Scaling

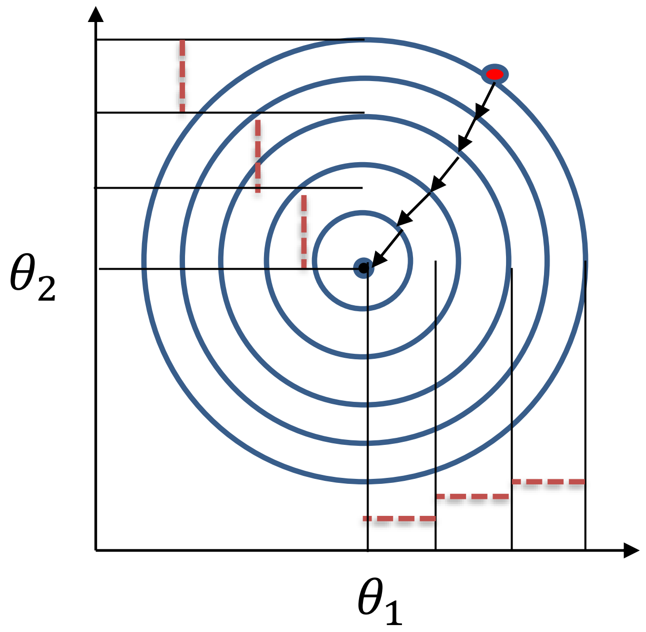
Features Scaling

$$2 \le x_1 \le 7$$ Room #

$$y = \theta_1 x_1 + \theta_2 x_2$$

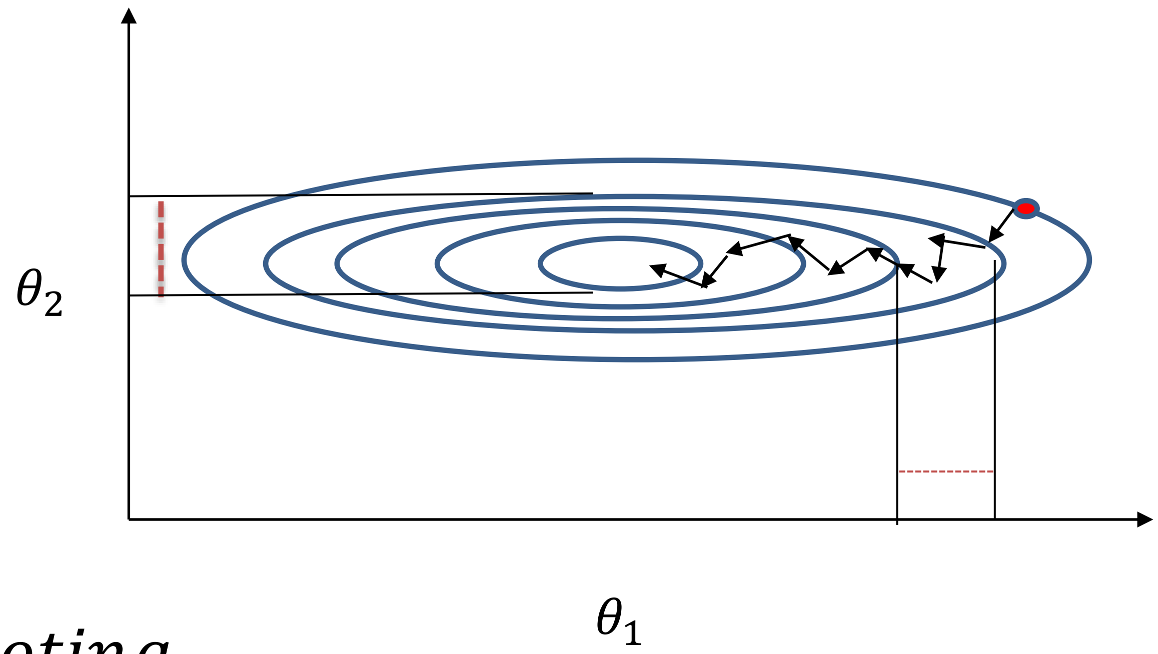
$$65 \le x_2 \le 300$$ Size





Speed (KM) x_1	Height (M) x_2	Price y
200	2.20	100,000
160	1.90	75,000
180	2.10	80,000
240	2.00	150,000

$$price = \theta_1(speed) + \theta_2(height)$$

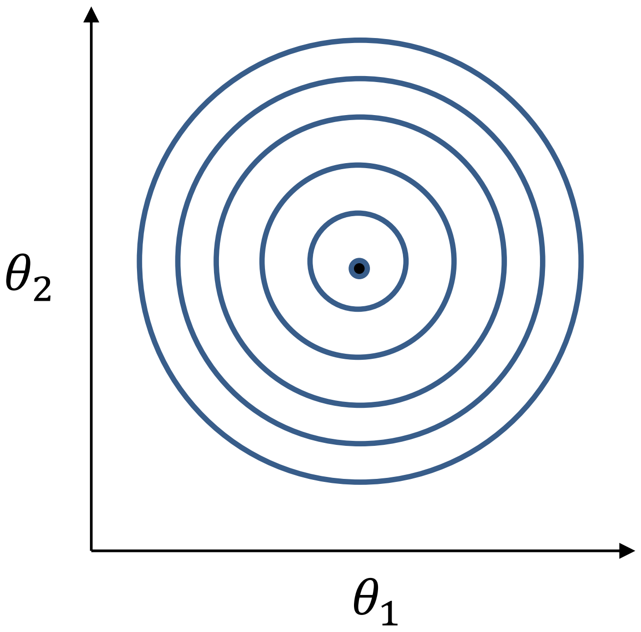


- *overshoting*
- *Long time*

Speed (KM)	Height (M)
200	2.20
160	1.90
180	2.10
240	2.00

Feature scaling

$0 \leq x \leq 1$



$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Speed (KM)	Height (M)
0.5	1
0	0
0.25	0.66
1	0.33

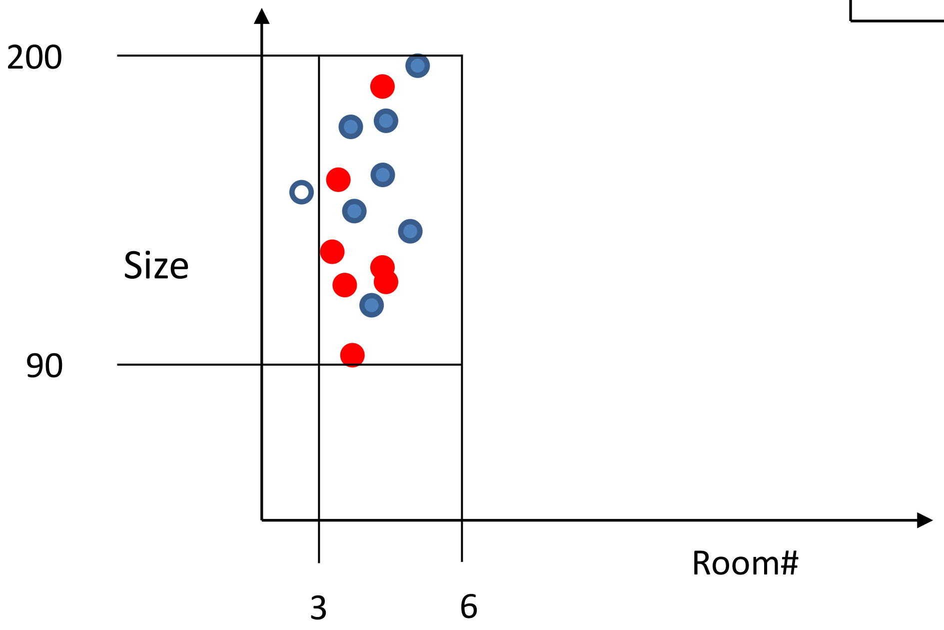
Features Scaling

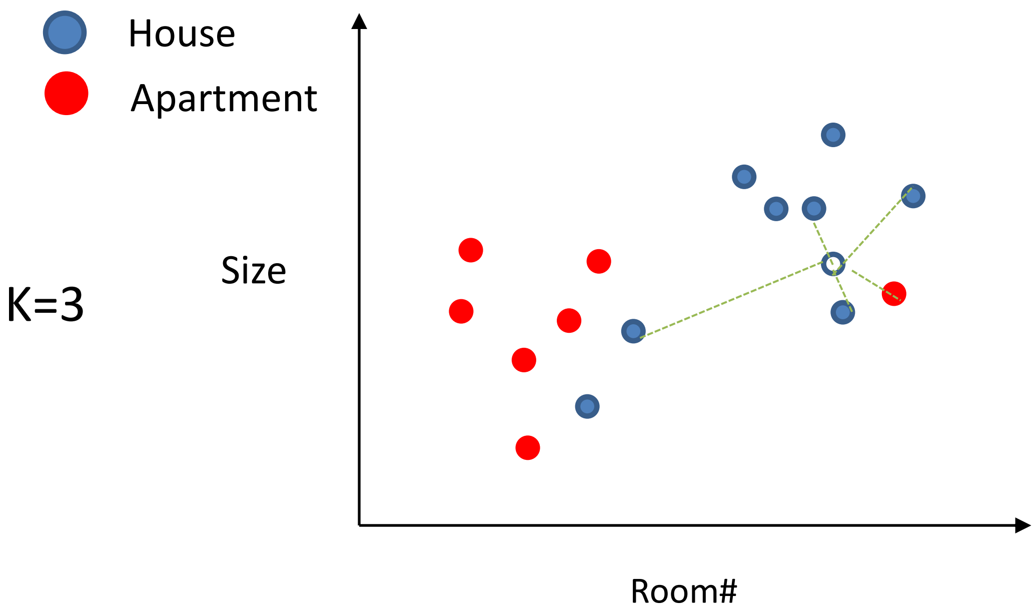
Used to unify the features ranges to be on a similar scale.

Euclidian Distance $\sqrt{(\Delta x_1)^2 + (\Delta x_2)^2}$

SVM, K-Means, K-NN

Size	Room#
90	3
120	5
150	4
200	6





Features Scaling Methods

1. Min-Max Scaling

Rescaling the values into a range [0, 1]

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

2. Standardization (z-score)

Rescaling the values to have a mean of 0 and a standard deviation of 1.

$$x' = \frac{x - \mu}{\sigma}$$

Size	Room#
90	3
120	5
150	4
200	6

Size	Room#
0	0
0.27	0.66
0.54	0.33
1	1

μ =140	$\mu = 4.5$
$\sigma = 40.62$	$\sigma = 1.118$

Size	Room#
-1.23	-1.34
-0.49	0.44
.24	-0.44
1.47	1.34

Report:
All students are required to report on three Common Machine Learning Algorithms

- **List of Common Machine Learning Algorithms**

    1. Linear Regression
    2. Logistic Regression
    3. Decision Tree
    4. SVM (Support Vector Machine)
    5. Naive Bayes
    6. kNN (k- Nearest Neighbors)
    7. K-Means
    8. Random Forest
    9. Dimensionality Reduction Algorithms
    10. Gradient Boosting Algorithms