Machine Learning

Dr. Muhammad Adeel Nisar

Assistant Professor – Department of IT,
Faculty of Computing and Information Technology,
University of the Punjab, Lahore

Recap of the Last Lecture

- Why Do you Need to Learn Machine Learning?
- What is Machine Learning?
- Applications of Machine Learning, Based on
  - Images and Videos Data
  - Sensor Data
  - Audio Data
  - Tabular Data
  - Text Data
- Machine Learning Life-Cycle

Today’s Contents

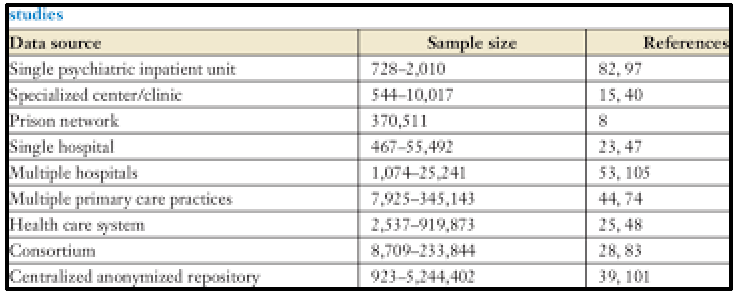
- Machine Learning Life-Cycle
- Types of Machine Learning
- Supervised Machine Learning
- Regression
    - Hypothesis Function
    - Cost Computation
    - Optimization Mechanism

Machine Learning Life-Cycle

- Data Acquisition
- Data Preparation
- Feature Extraction
- Train Model
- Test Model
- Evaluate and Improve

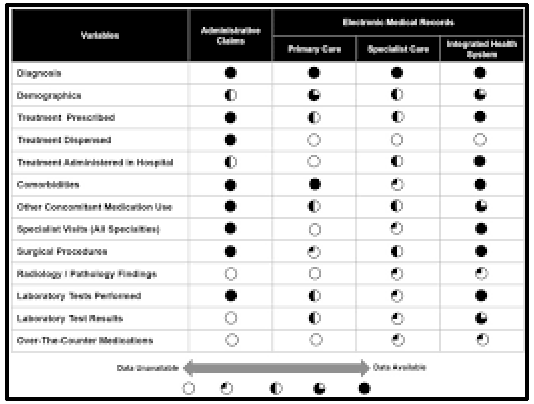
Data Acquisition

- Types of data
  - Electronic Records/ Tabular data



studies

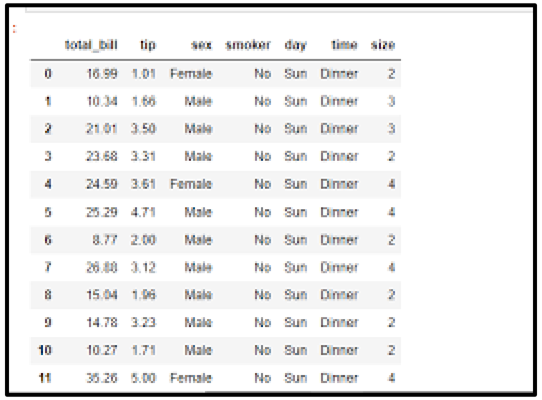
Data source	Sample size	References
Single psychiatric inpatient unit	728–2,010	82, 97
Specialized center/clinic	544–10,017	15, 40
Prison network	370,511	8
Single hospital	467–55,492	23, 47
Multiple hospitals	1,074–25,241	53, 105
Multiple primary care practices	7,925–345,143	44, 74
Health care system	2,537–919,873	25, 48
Consortium	8,709–233,844	28, 83
Centralized anonymized repository	923–5,244,402	39, 101



Variables	Administrative Claims	Electronic Medical Records		
		Primary Care	Specialist Care	Integrated Health System
Diagnosis	●	●	●	●
Demographics	◐	◕	◐	◕
Treatment Prescribed	●	◐	◐	●
Treatment Dispensed	●	○	○	○
Treatment Administered in Hospital	◐	○	◐	●
Comorbidities	●	●	◔	●
Other Concomitant Medication Use	●	◐	◐	◕
Specialist Visits (All Specialities)	●	○	◔	●
Surgical Procedures	●	◔	◐	●
Radiology / Pathology Findings	○	○	◔	◔
Laboratory Tests Performed	●	◐	◔	●
Laboratory Test Results	○	◐	◔	◕
Over-The-Counter Medications	○	○	◔	◔

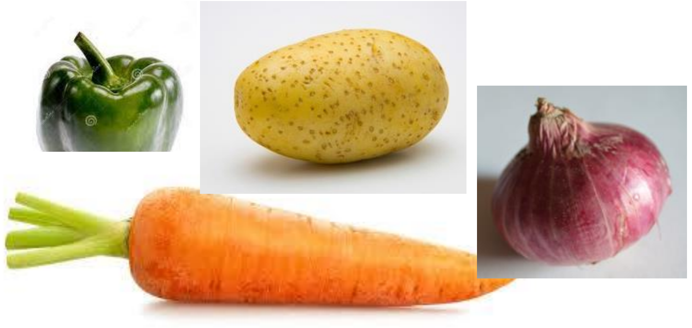
Data Unavailable ↔ Data Available

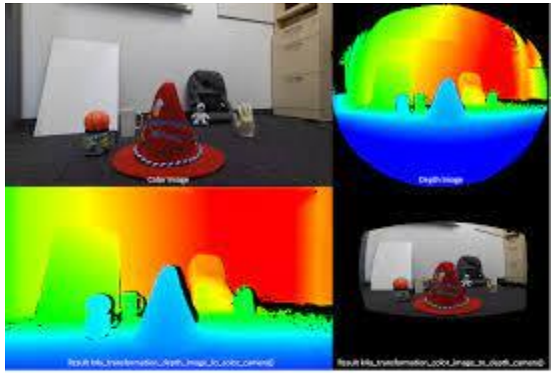
○ ◔ ◐ ◕ ●



	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
5	25.29	4.71	Male	No	Sun	Dinner	4
6	8.77	2.00	Male	No	Sun	Dinner	2
7	26.88	3.12	Male	No	Sun	Dinner	4
8	15.04	1.96	Male	No	Sun	Dinner	2
9	14.78	3.23	Male	No	Sun	Dinner	2
10	10.27	1.71	Male	No	Sun	Dinner	2
11	35.26	5.00	Female	No	Sun	Dinner	4

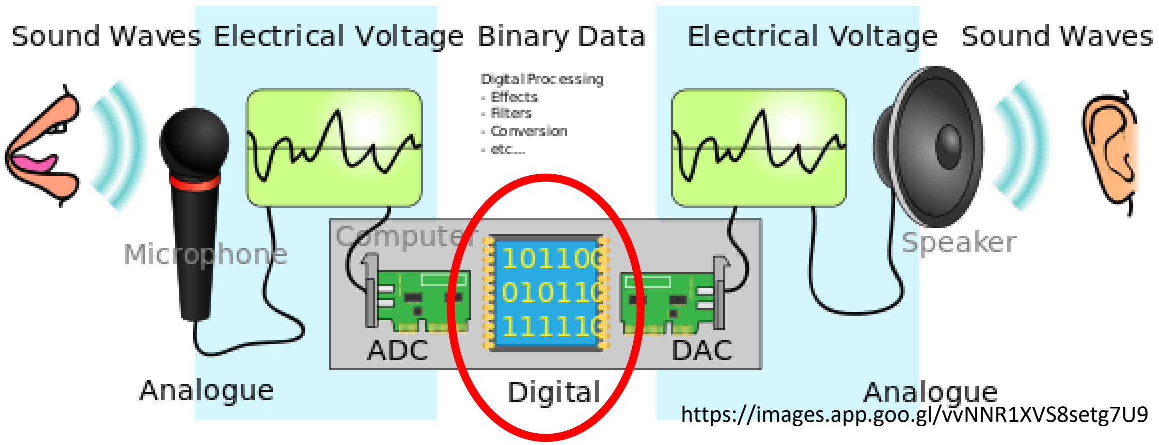
  - Images





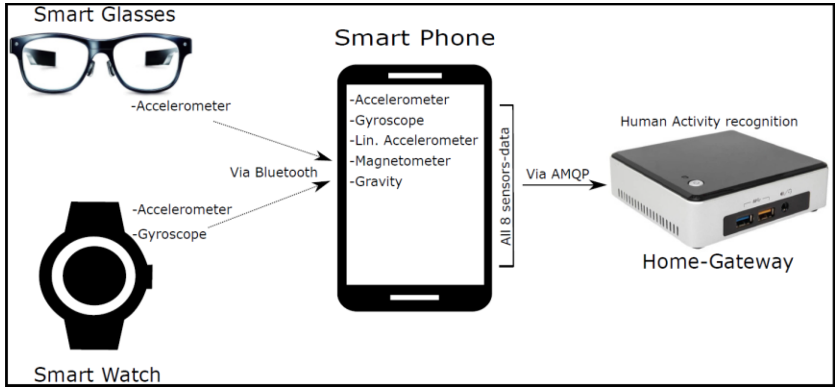
Data Acquisition

- Types of data
  - Audio



https://images.app.goo.gl/vvNNR1XVS8setg7U9

  - Sensors Data

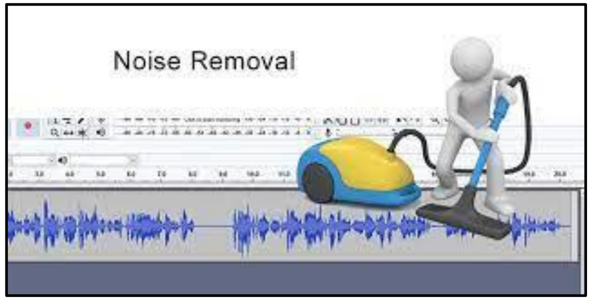


https://instock.pk/emotiv-epoc-14-channel-mobile-eeg.html

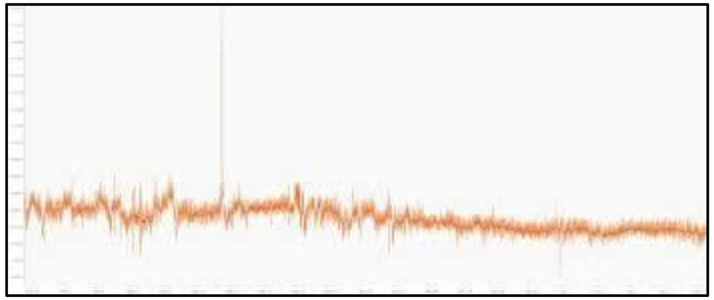
https://images.app.goo.gl/m5uUHz9keixcwV3u8

Data Preparation

- Noise Removal



https://images.app.goo.gl/WGpHsUFYAqr2DcmE8



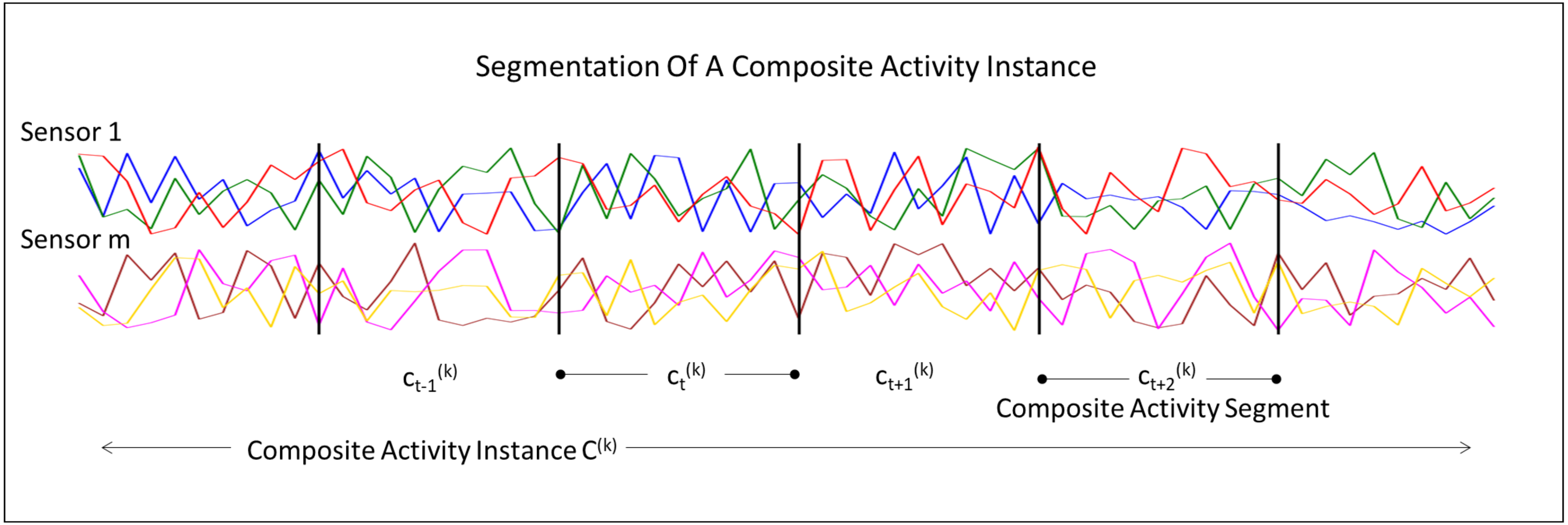
https://images.app.goo.gl/MtPN7BpxiHyYBj9T9

https://images.app.goo.gl/Ny4gD1d5cRFpGm1n7

- Python Libraries for noise removal
  - https://pypi.org/project/noisereduce/
  - https://scikit-image.org/docs/stable/auto_examples/filters/plot_denoise.html

Data Preparation

- Segmentation



Sensor-Based Human Activity Recognition for Assistive Health Technologies by Dr. Muhammad Adeel Nisar, Logos Verlag Berlin GmbH, 2023

Types of Machine Learning

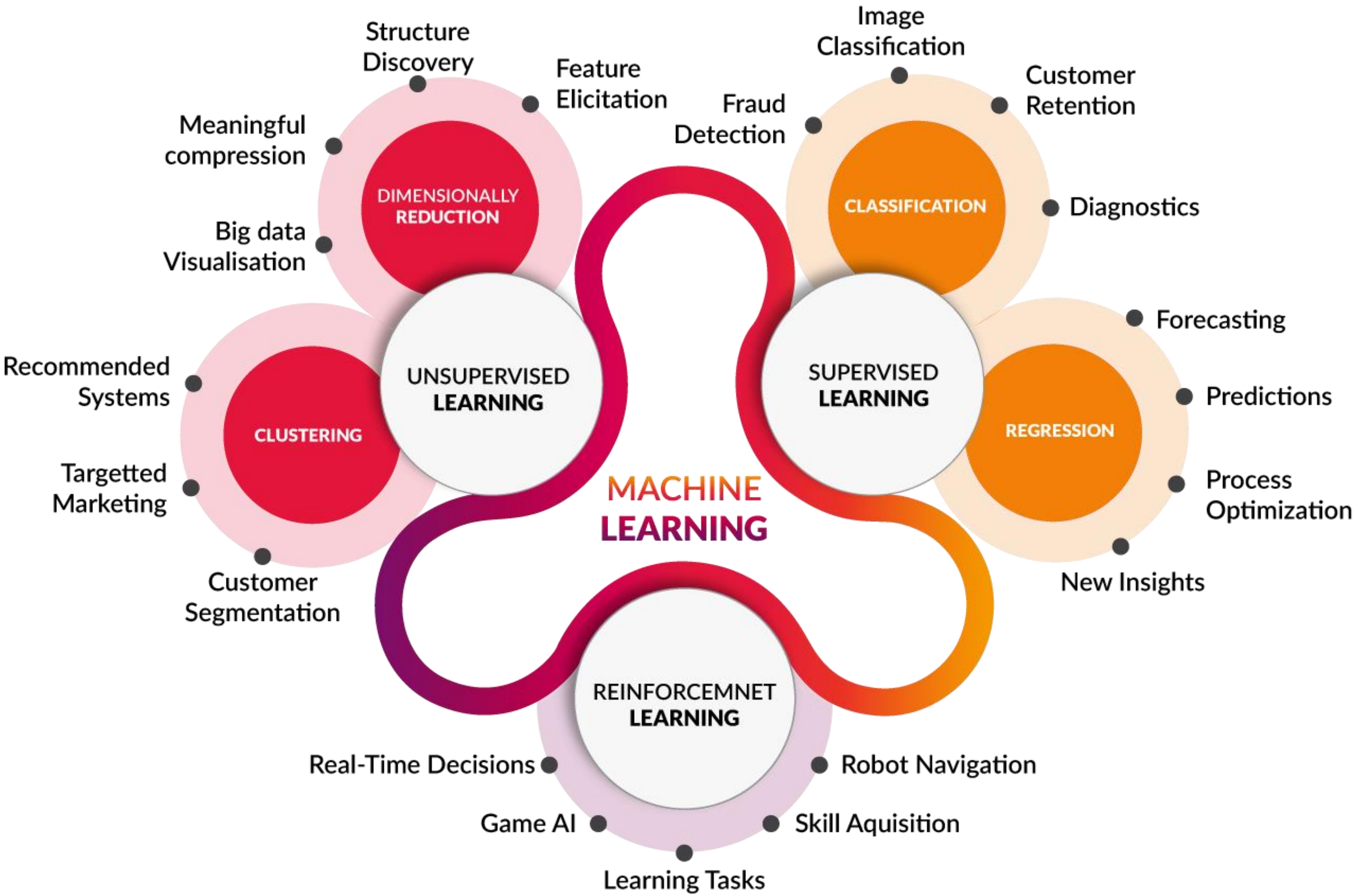
- **Supervised Machine Learning** assumes that a set of labelled training data $\{(x^{(i)},y^{(i)})\}$ where i is 1 to m is available and the classifier is designed by exploiting this a-priori known information.
- Two further types
  - Regression
    - Linear Regression
    - Nonlinear Regression
  - Classification
    - Logistic Regression
    - Naïve Bayes
    - Support Vector Machines etc

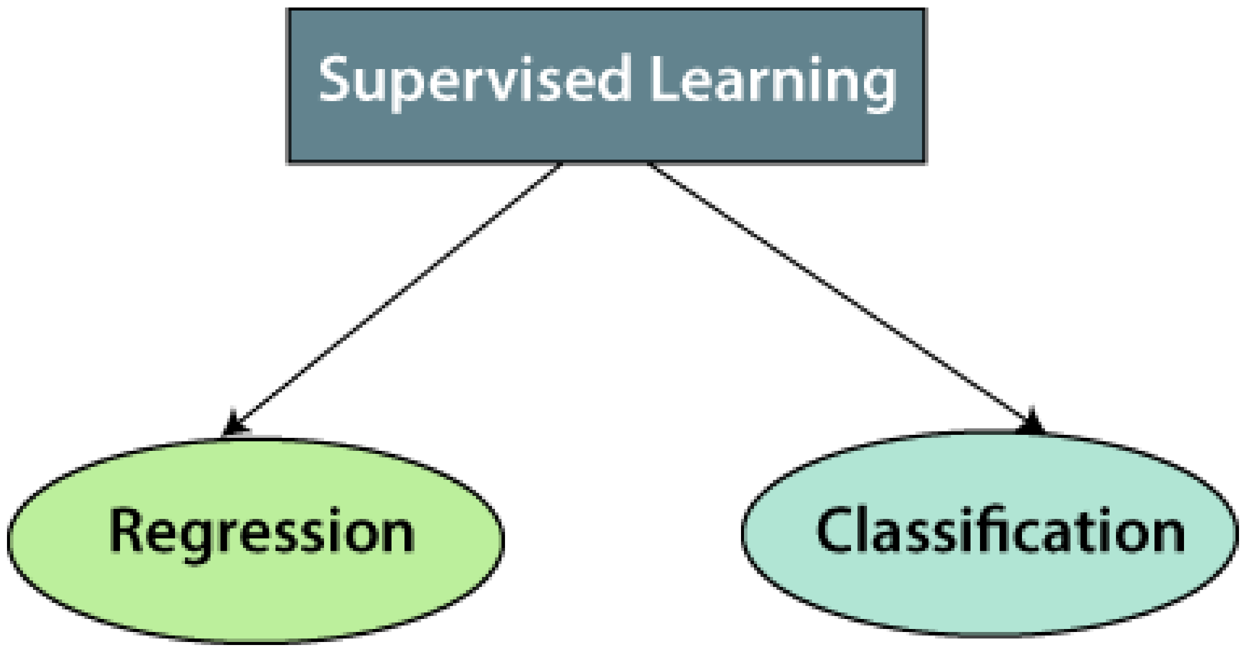
Types of Machine Learning

- **Unsupervised Machine Learning** clusters unlabeled training data, $\{x^{(i)}\}$ where i is 1 to m, described by feature vectors, $x^{(i)}$, into similar groups
  - Clustering
    - K-Means Clustering
  - Dimensionality Reduction
    - Principal Component Analysis
    - Autoencoders

Types of Machine Learning

- In **Semi-supervised Machine Learning** the dataset contains both labeled and unlabeled examples. Usually, the quantity of unlabeled examples is much higher than the number of labeled examples. The goal of a semi-supervised learning algorithm is the same as the goal of the supervised learning algorithm.
- **Reinforcement Learning** solves a particular kind of problems where decision making is sequential, and the goal is long-term, such as game playing, robotics, resource management, or logistics.


MACHINE LEARNING
UNSUPERVISED LEARNING
DIMENSIONALLY REDUCTION
Structure Discovery
Feature Elicitation
Meaningful compression
Big data Visualisation
CLUSTERING
Recommended Systems
Targetted Marketing
Customer Segmentation
SUPERVISED LEARNING
CLASSIFICATION
Image Classification
Customer Retention
Fraud Detection
Diagnostics
REGRESSION
Forecasting
Predictions
Process Optimization
New Insights
REINFORCEMNET LEARNING
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks


Supervised Learning
Regression
Classification

Regression

- A Supervised learning algorithm
- Taking input variables and trying to fit the output onto a continuous values.
- Linear regression with one variable is also known as “Univariate linear regression”.
- Univariate linear regression is used when you want to predict a single output value y from a single input value x.
- The Hypothesis Function $y' = h\theta(x) = \theta_0 + \theta_1 x$
- Given the training data with right answers, predict the real-valued output for the test data.

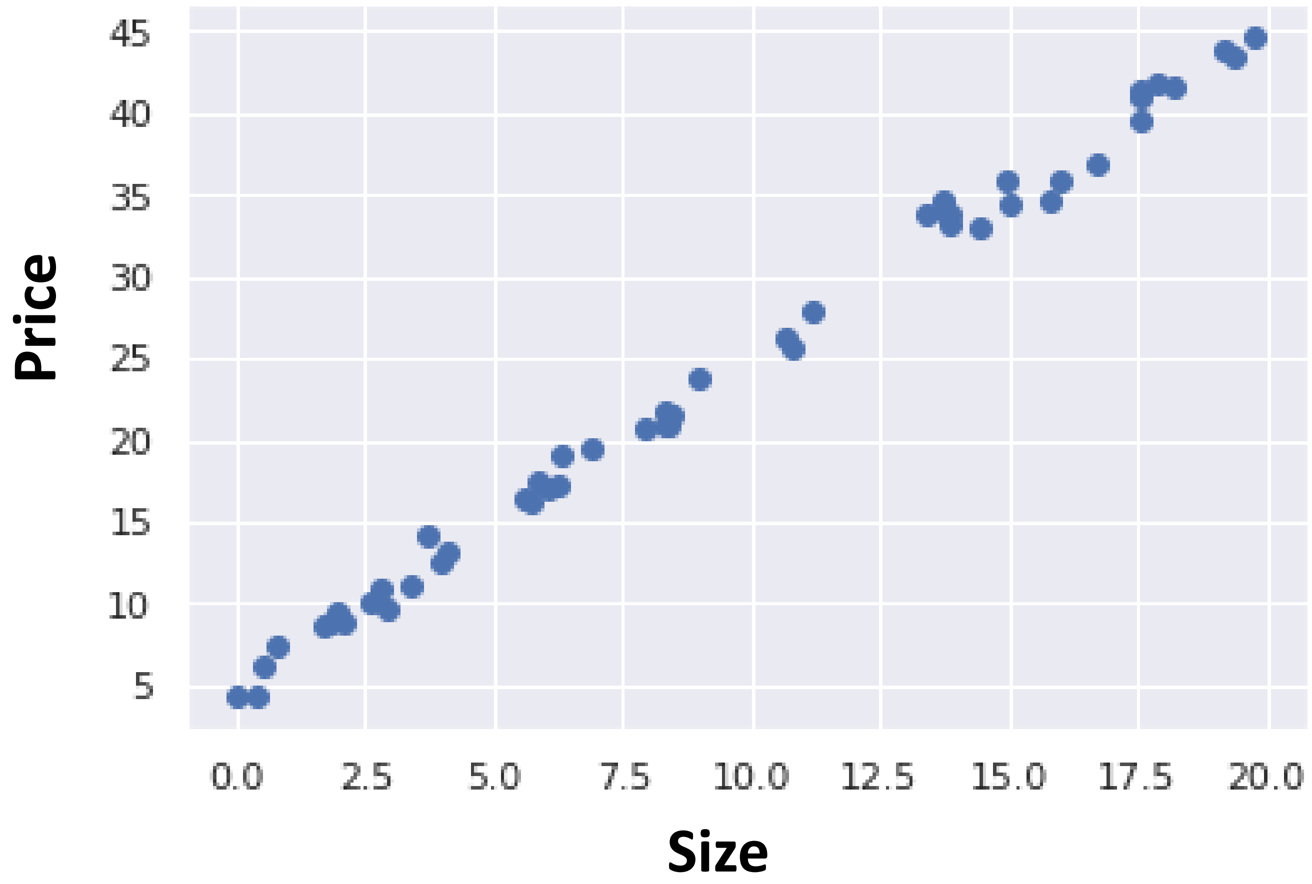
Dataset and Notations

- Notations
- m = Number of Training Examples
- **x** = Input Variable/ Features
- **y** = Output Variable / Target Value
- **(x,y)** is one training example
- **$(x^{(i)},y^{(i)})$** is i^{th} training example
- **$(x^{(1)},y^{(1)})$ = (8.3, 20.99)**

	Input Data (x)	Correct Answer (y)
1	8.3	20.99
2	14.4	32.89
3	6.05	17.1
	..	..

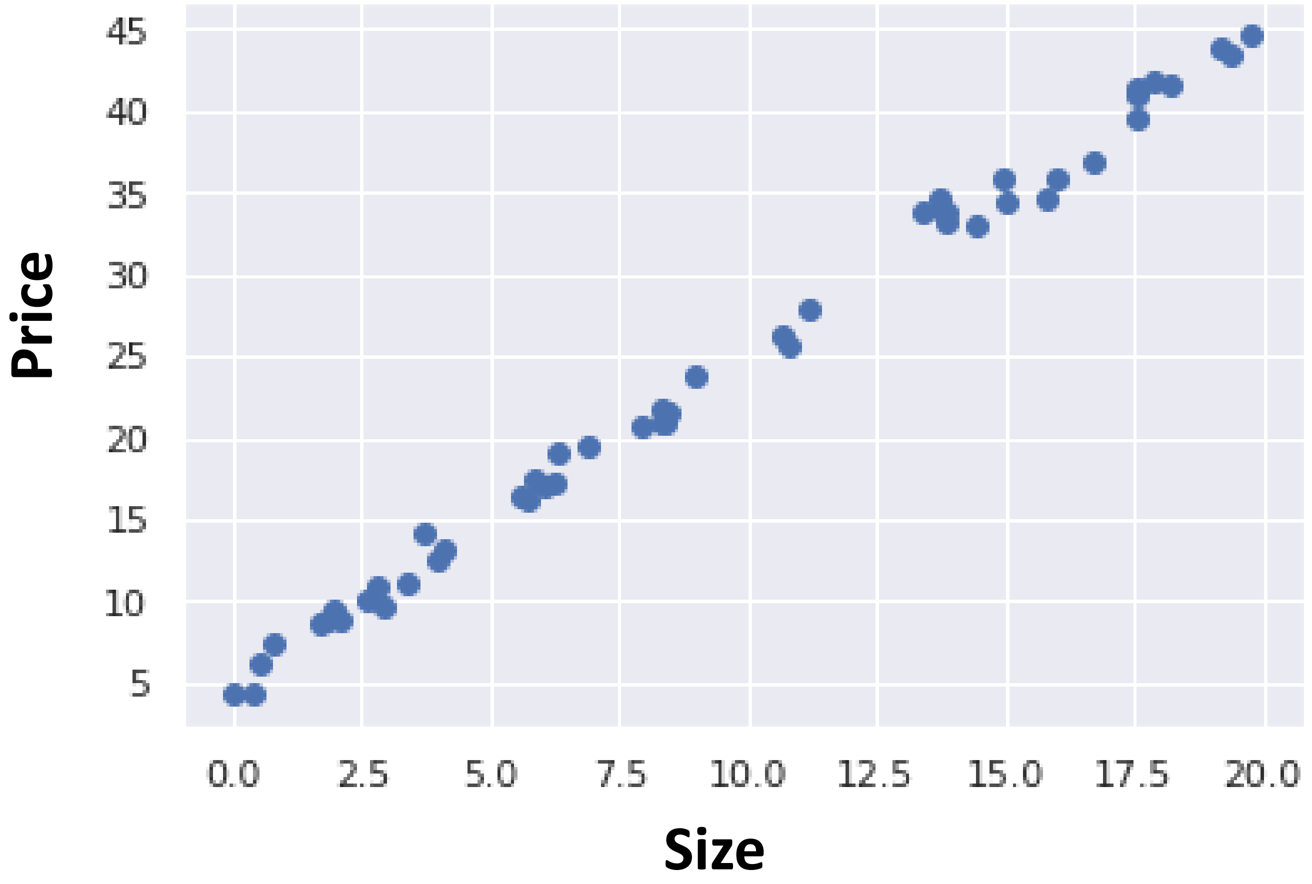
Dataset and Plotted Graph

Input Data (x)	Correct Answer (y)
8.3	20.99
14.4	32.89
6.05	17.08
..	..



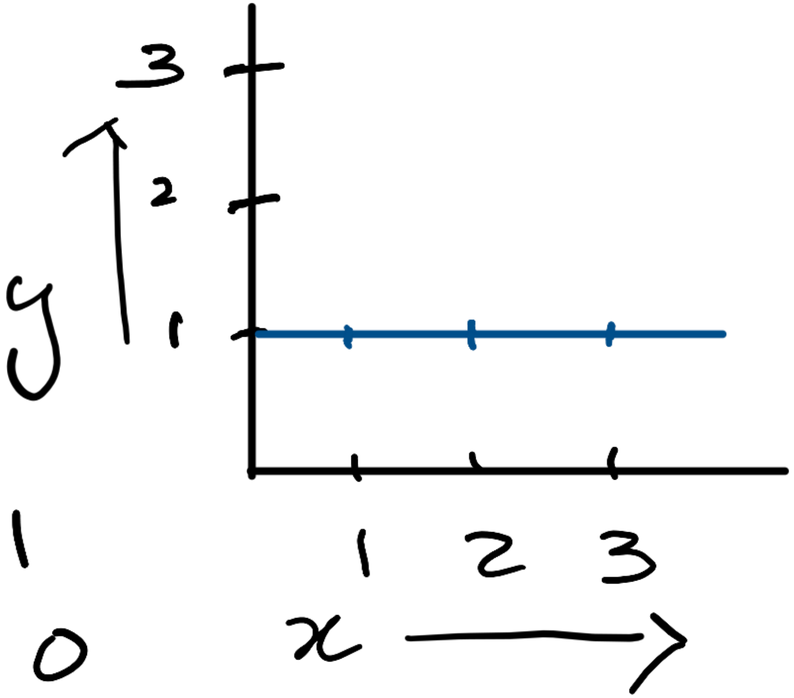
Linear Regression with One Variable

- This is like the equation of a straight line.
- We give $h\theta(x)$ values for θ_0 and θ_1 to get our estimated output y'.
- We are trying to create a function that will map out input data to our output data.

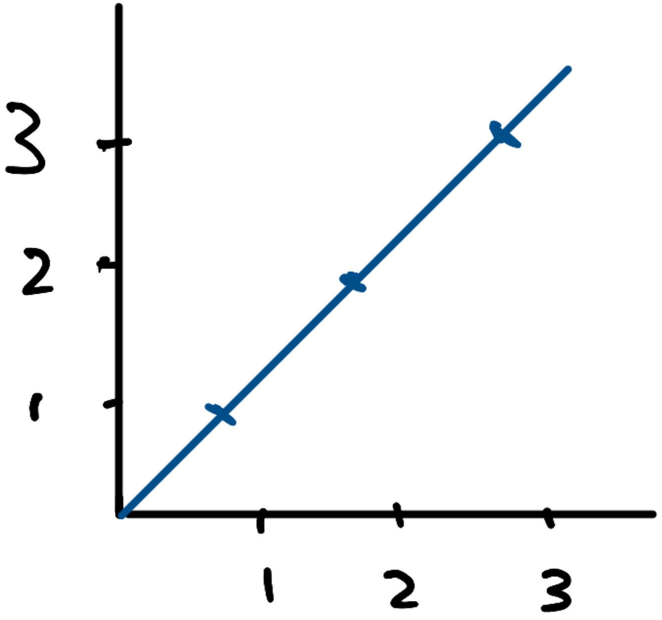


Equation of Line

Linear Functions With Varying Values of Θ



$\theta_0 = 1$
$\theta_1 = 0$

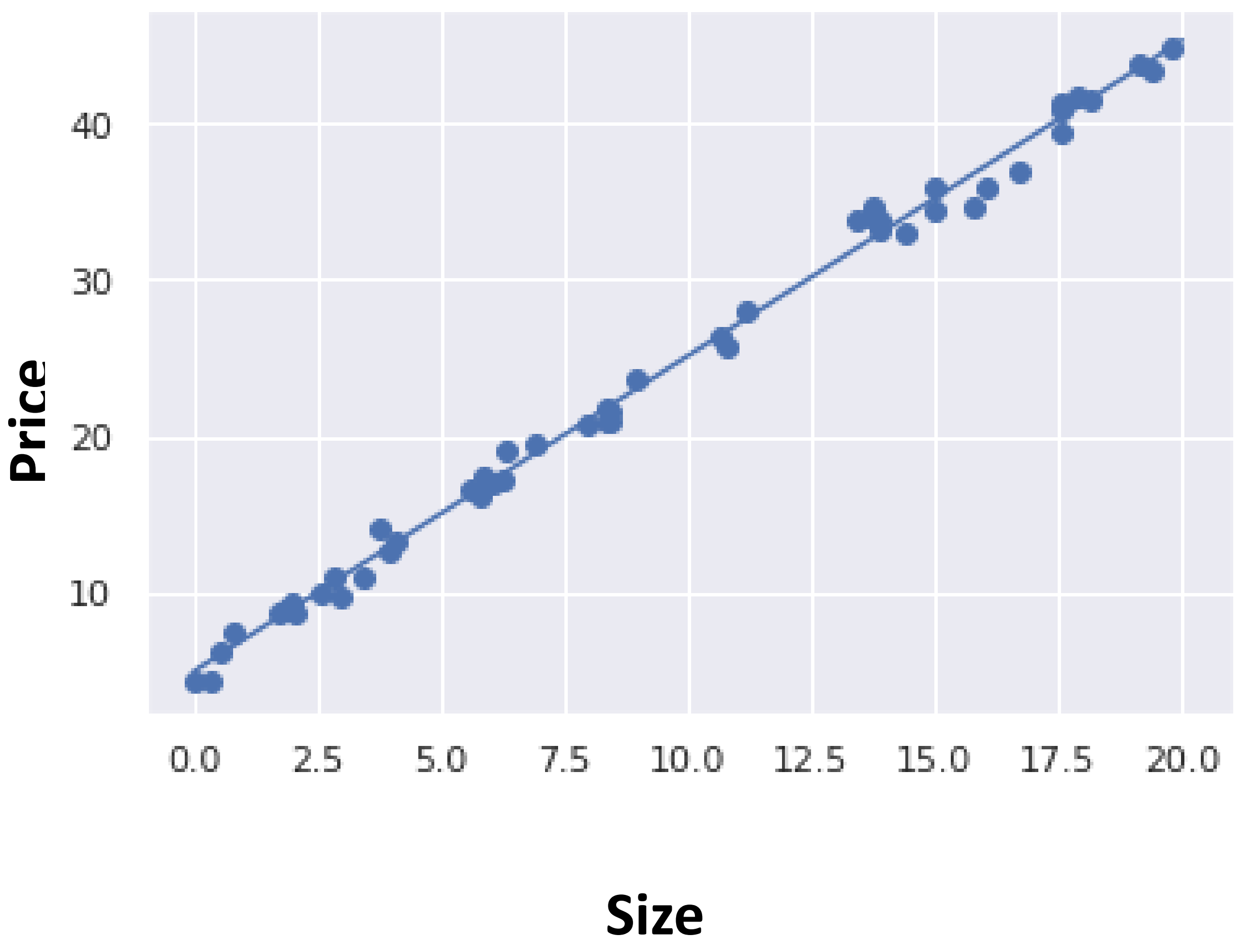


$\theta_0 = 0$
$\theta_1 = 1$

Linear Regression with One Variable

- $y' = h\theta(x) = \theta_0 + \theta_1 x$
- Intercept: $\theta_0 = 5$
- Slope: $\theta_1 = 2$

Input Data (x)	Correct Answer (y)
5	15
10	25
15	35
..	..



Cost Function

- **Hypothesis Function**

$$y' = h\vartheta(x) = \vartheta_0 + \vartheta_1 x$$

- **Cost function** (to measure the performance of hypothesis function)

$$J(\vartheta_0, \vartheta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\ y'^{(i)} - \ y^{(i)}\ )^2 = \frac{1}{2m}\sum_{i=1}^{m}(\ h\vartheta(x^{(i)}) - \ y^{(i)}\ )^2$$

Cost Function

Input Data (x)	Correct Answer (y)	Estimated Answer	Error
8.3	20.99	21.6	-0.61
14.4	32.89	33.8	-0.81
6.05	17.1	17.08	0.02
..	..	..	..

$$Mean\ Square\ Error\ (MSE) = J(\vartheta_0, \vartheta_1) = \frac{1}{2m}\sum_{i=1}^{n}(\ y^{(i)} - \ y'^{(i)}\ )^2$$

Cost Function

- **Hypothesis Function**

$$y' = h\vartheta(x) = \vartheta_0 + \vartheta_1 x$$

- **Cost function** (to measure the performance of hypothesis function)

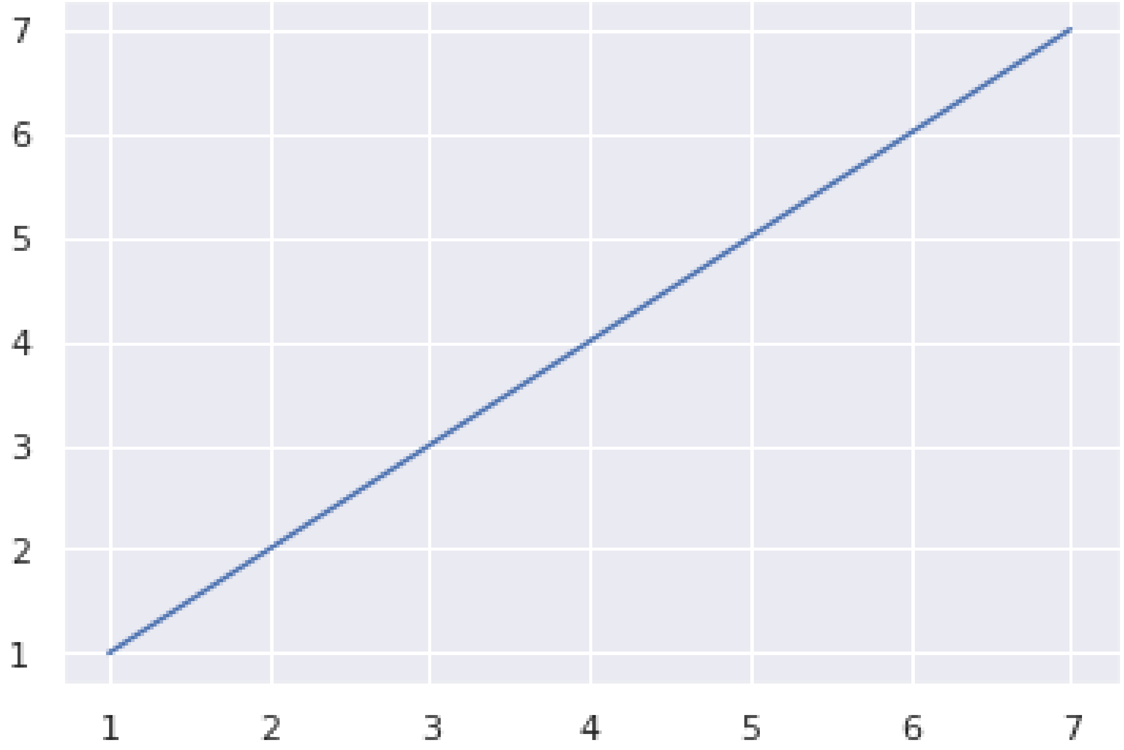
$$J(\vartheta_{0,}\vartheta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\ y'^{\,(i)}\ -\ y^{\,(i)}\ )^2 = \frac{1}{2m}\sum_{i=1}^{m}(\ h\vartheta(x^{\,(i)})\ -\ y^{\,(i)}\ )^2$$

- **Objective:**

$$\min_{\vartheta_{0,}\vartheta_1} \mathrm{J}(\vartheta_{0,}\vartheta_1)$$

Cost Function - Example

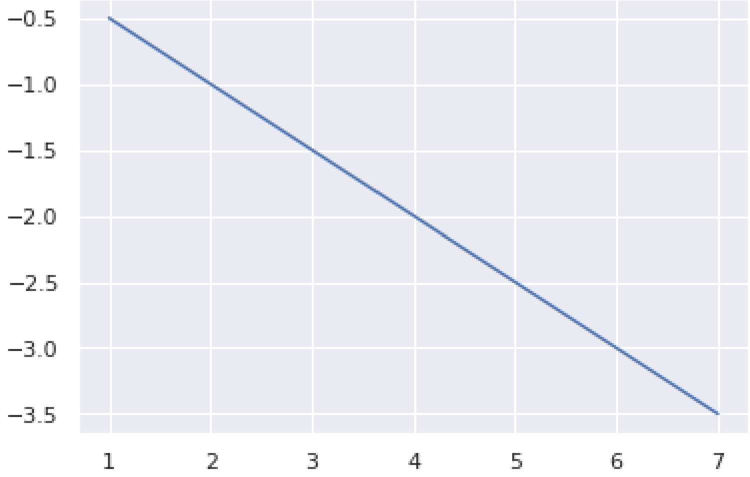
- **x** = [1, 2, 3, 4, 5, 6, 7]
- **y** = [1, 2, 3, 4, 5, 6, 7]
- $y' = h\theta(x) = \theta_0 + \theta_1 x$
- Assume $\theta_0 = 0$
- So, $y' = h\theta(x) = \theta_1 x$

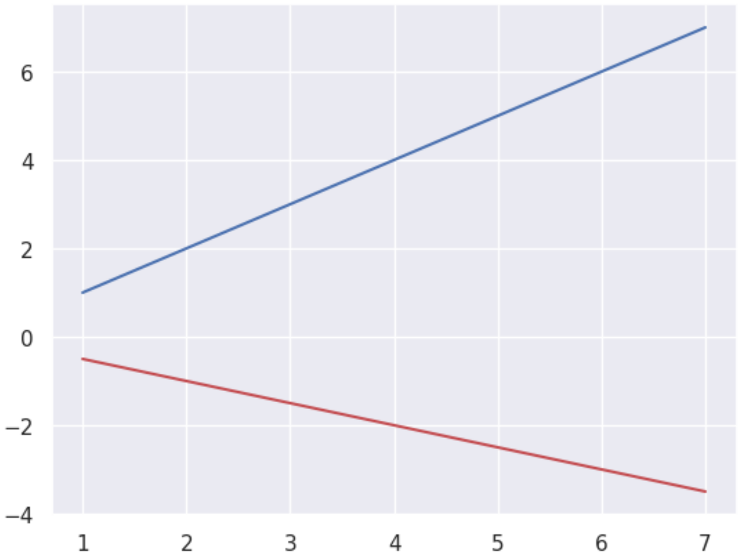


θ_1 = [-0.5, -0.25, 0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25]

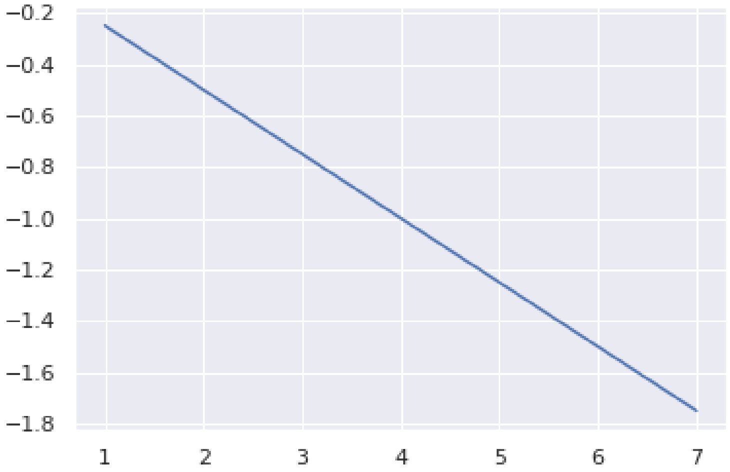
Cost Function - Example

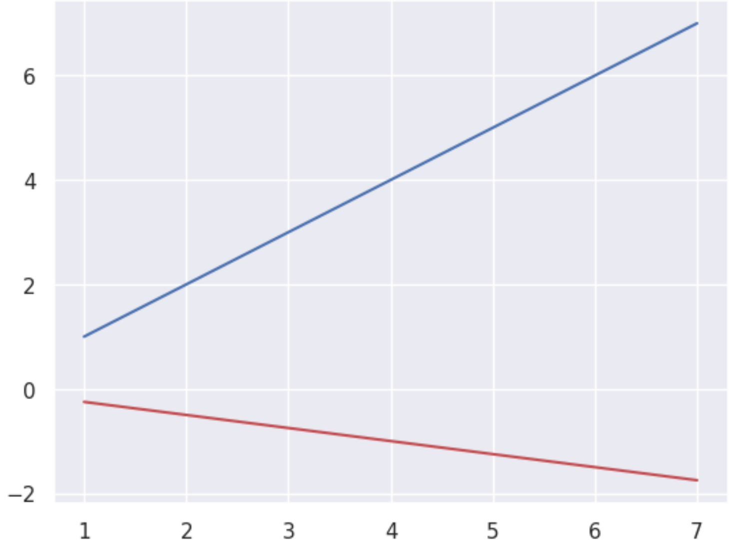
- ϑ_1 = -0.5, J(ϑ_1)= 45.0





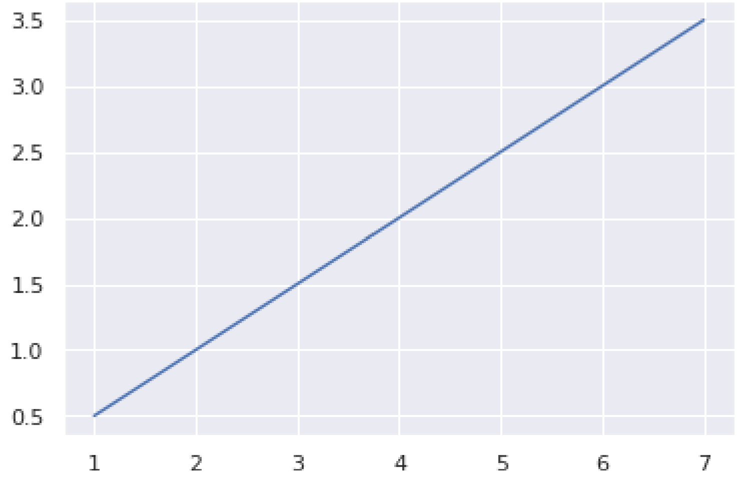
- ϑ_1 = -0.25,J(ϑ_1)= 31.25

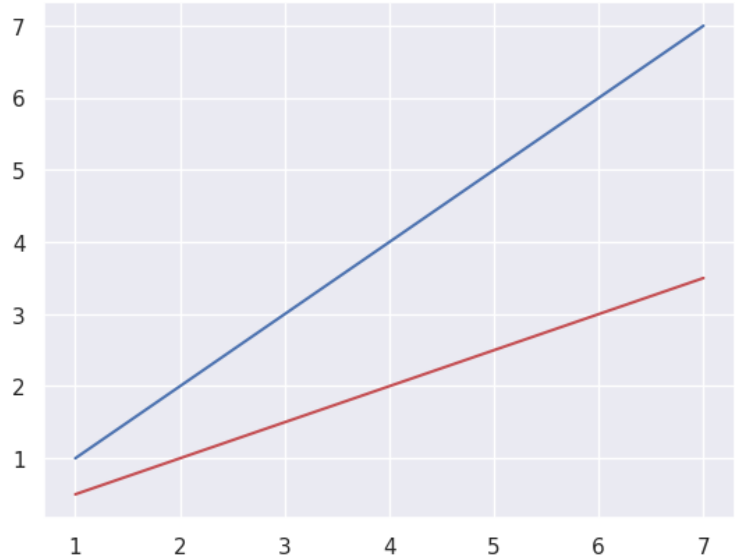




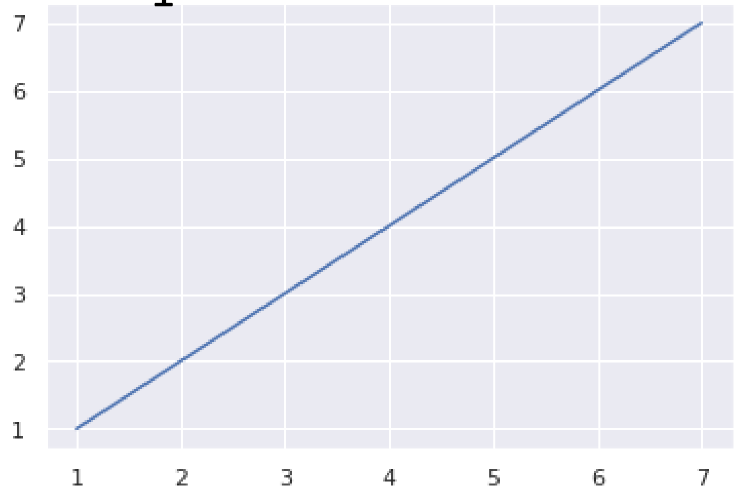
Cost Function - Example

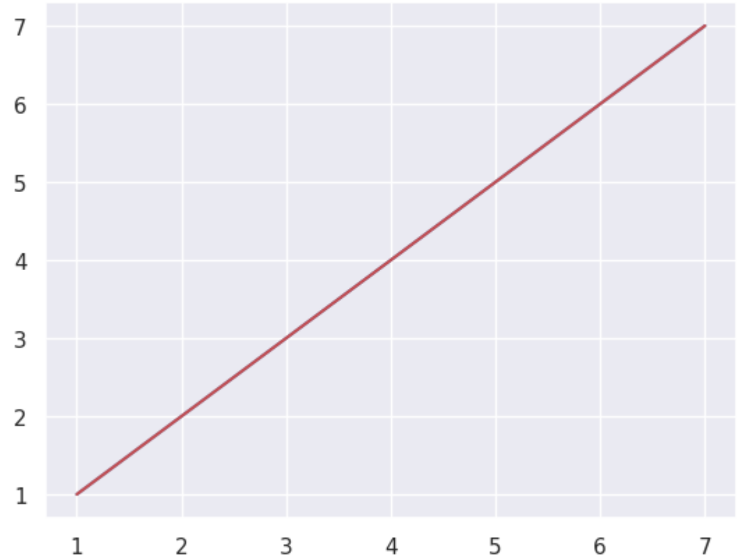
- $\vartheta_1 = 0.5,\ J(\vartheta_1) = 5.0$





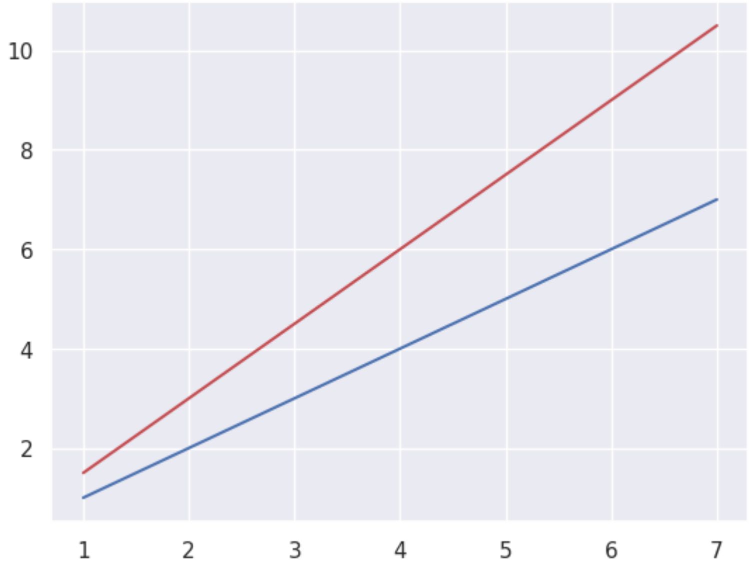
- $\vartheta_1 = 1.0,\ J(\vartheta_1) = 0.0$



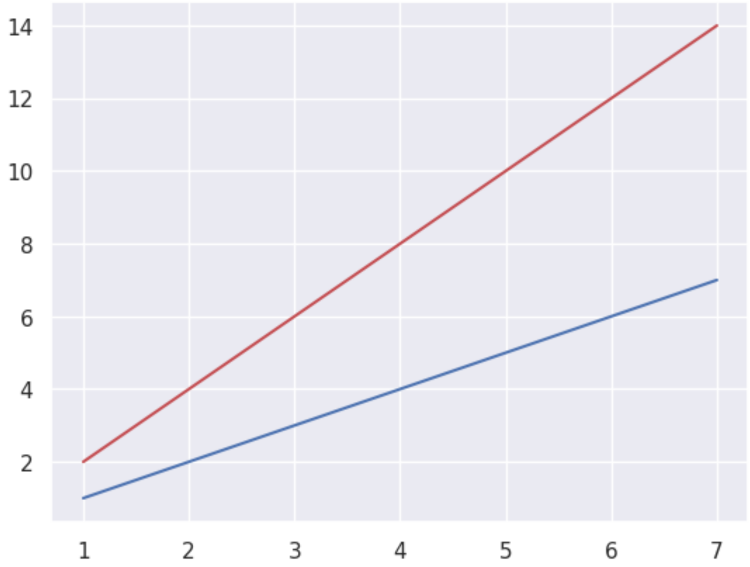


Cost Function - Example

- $\vartheta_1 = 1.5$, $J(\vartheta_1) = 5.0$

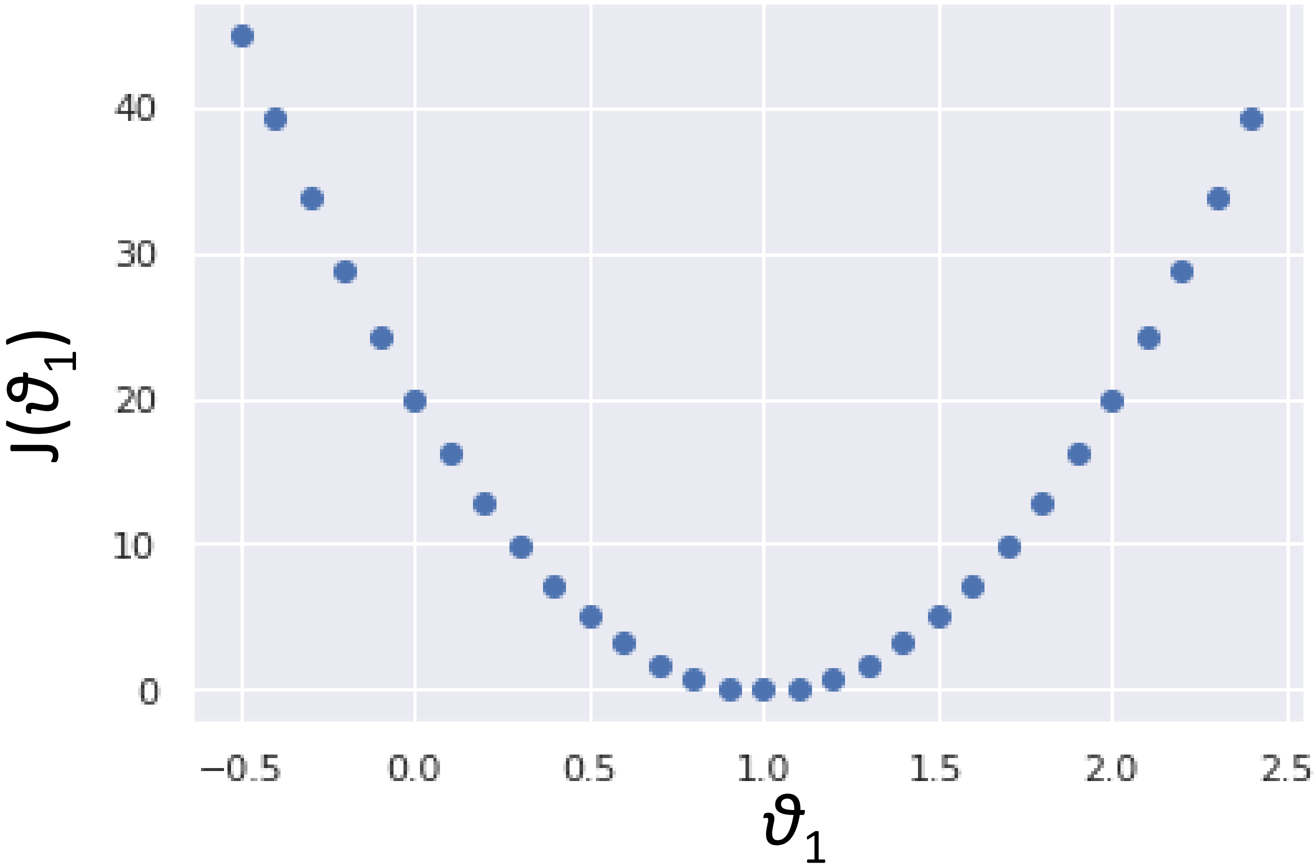


- $\vartheta_1 = 2.0$, $J(\vartheta_1) = 20.0$

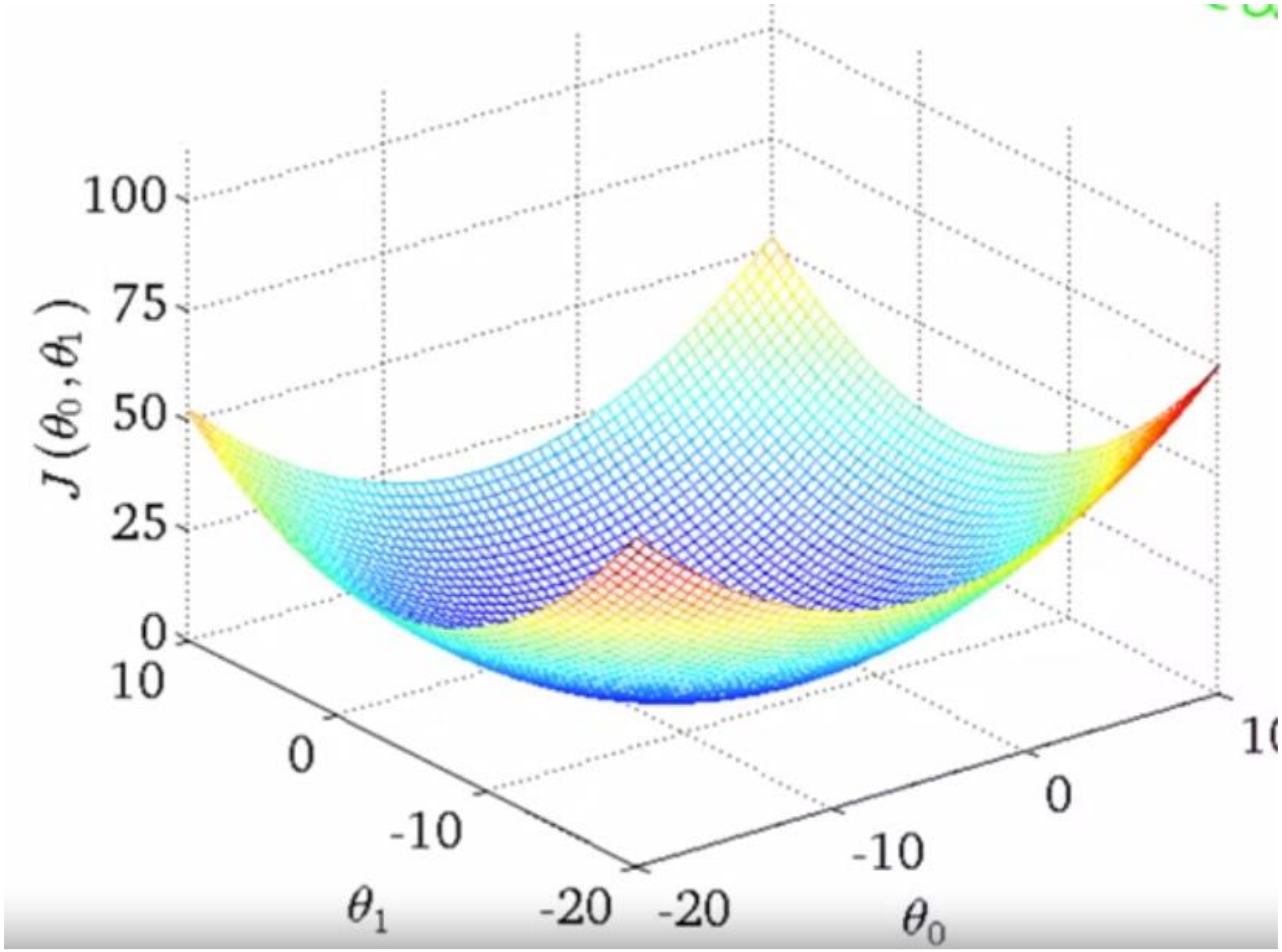


Graph of J(ϑ_1)

- θ_1 = [-0.5, -0.25, 0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25]
- $J(\theta_1)$ =[45.0, 31.25, 20.0, 11.25, 5.0, 1.25, 0.0, 1.25, 5.0, 11.25, 20.0, 31.25]



Graph of J(ϑ_0, ϑ_1)



Some Concepts

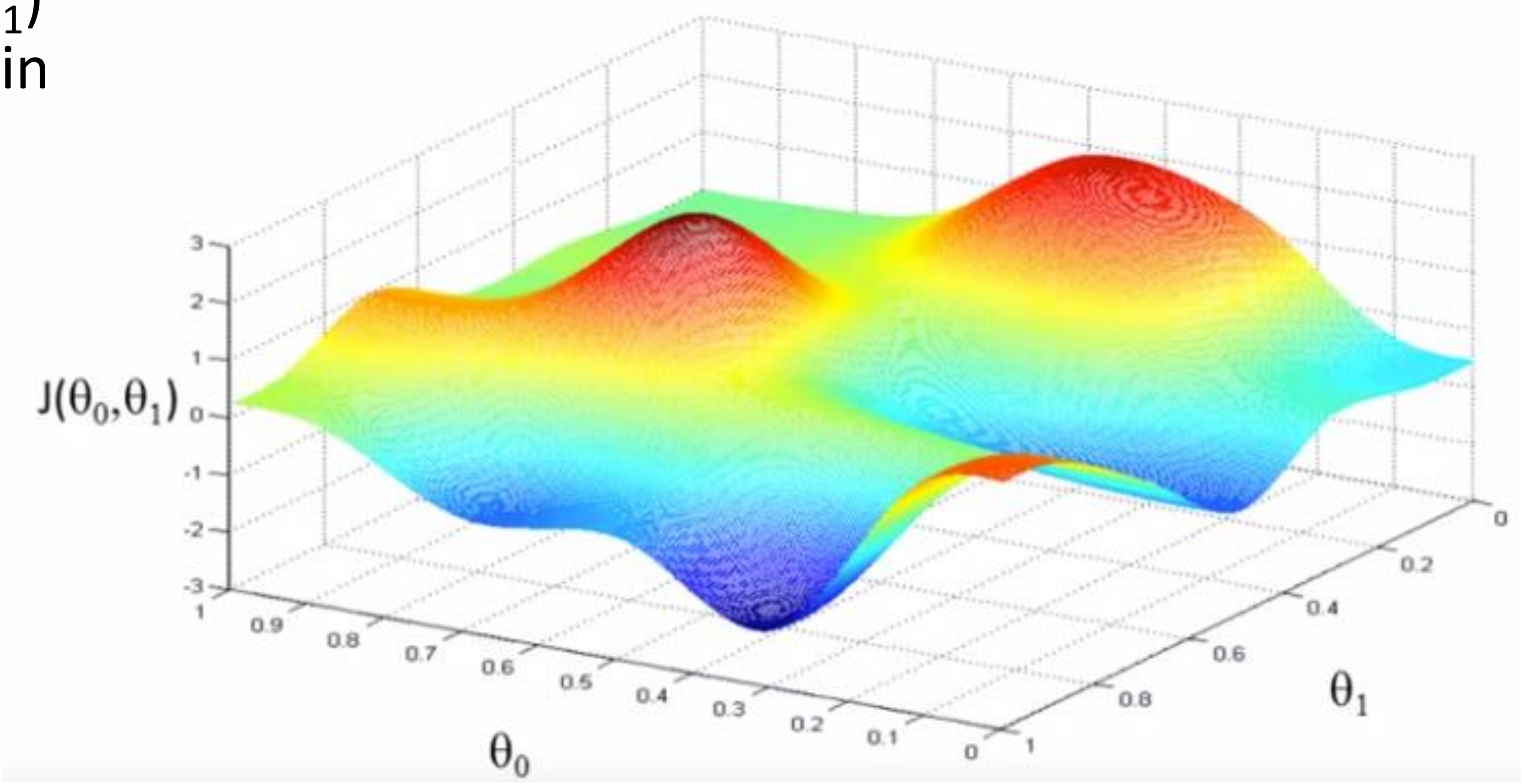
- Global Minimum

- Local Minimum

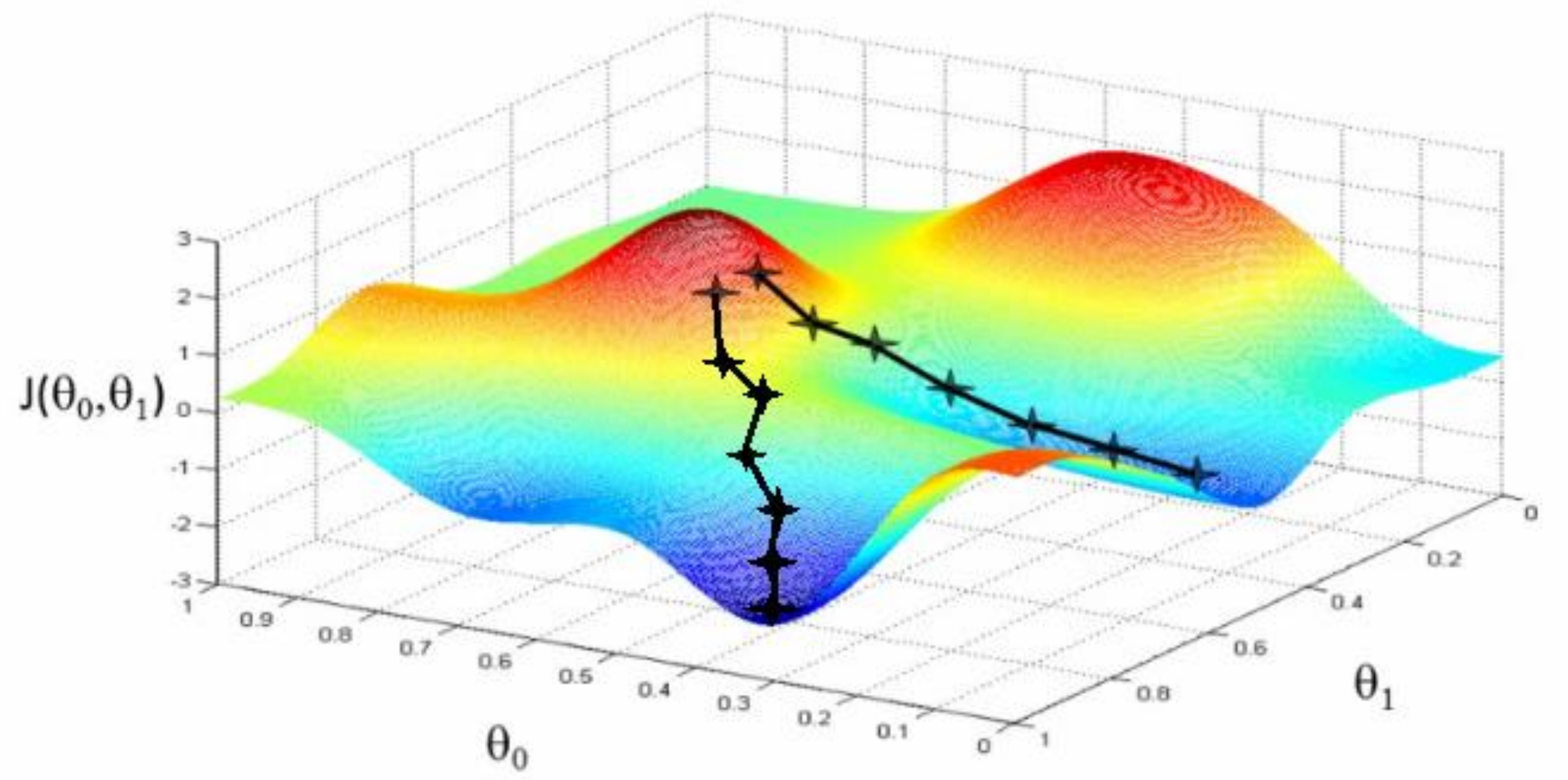
- Convex Functions

Gradient Descent Algorithm

- We have J(ϑ_0, ϑ_1) and we want min J(ϑ_0, ϑ_1)



Solving Minimization Problem



Recap of the Latest Contents

- Univariate Linear Regression

$$y' = h_\theta(x) = \theta_0 + \theta_1 x$$

- Cost/Loss function, Mean Squared Error

$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(y'^{(i)} - y^{(i)})^2 = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Derivatives

$f(x) = 4x$

$f(x) = x^3$

$f(x) = (x + 2)^4$

$f(x,y) = (3x + 2y + 2)^2$

Gradient Descent Algorithm

- $\vartheta_j := \vartheta_j - \alpha \frac{\partial}{\partial \vartheta_j} J(\vartheta_0, \vartheta_1)$ (for j = 0 and j = 1)
- $\frac{\partial}{\partial \vartheta_j} J(\vartheta_0, \vartheta_1)$ is a partial derivative term
- α : (Alpha) is learning rate
- Simultaneous Update
- temp0 = $\vartheta_0 - \alpha \frac{\partial}{\partial \vartheta_0} J(\vartheta_0, \vartheta_1)$
- temp1 = $\vartheta_1 - \alpha \frac{\partial}{\partial \vartheta_1} J(\vartheta_0, \vartheta_1)$
- $\vartheta_0 :=$ temp0
- $\vartheta_1 :=$ temp1

Linear Regression with Gradient Descent

$$\vartheta_j := \vartheta_j - \alpha \frac{\partial}{\partial \vartheta_j} J(\vartheta_0, \vartheta_1)$$

$$\frac{\partial}{\partial \vartheta_j} J(\vartheta_0, \vartheta_1) = \frac{\partial}{\partial \vartheta_j} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(h\vartheta(x^{(i)}) - y^{(i)} \right)^2\right)$$

$$\frac{\partial}{\partial \vartheta_j} J(\vartheta_0, \vartheta_1) = \frac{\partial}{\partial \vartheta_j} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right)^2\right)$$

Linear Regression with Gradient Descent

$$\frac{\partial}{\partial \vartheta_j} J(\vartheta_0, \vartheta_1) = \frac{\partial}{\partial \vartheta_j} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right)^2\right)$$

$$\frac{\partial}{\partial \vartheta_0} J(\vartheta_0, \vartheta_1) = \frac{\partial}{\partial \vartheta_0} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right)^2\right)$$

$$\frac{\partial}{\partial \vartheta_0} J(\vartheta_0, \vartheta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right)$$

$$\frac{\partial}{\partial \vartheta_1} J(\vartheta_0, \vartheta_1) = \frac{\partial}{\partial \vartheta_1} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right)^2\right)$$

$$\frac{\partial}{\partial \vartheta_1} J(\vartheta_0, \vartheta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(\left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right) x^{(i)}\right)$$

Linear Regression with Gradient Descent

- Repeat until **converge**

$$\vartheta_0 := \vartheta_0 - \alpha \left(\frac{1}{m}\sum_{i=1}^{m}\left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right)\right)$$

$$\vartheta_1 := \vartheta_1 - \alpha \left(\frac{1}{m}\sum_{i=1}^{m}\left(\vartheta_0 + \vartheta_1 x^{(i)} - y^{(i)} \right) x^{(i)}\right)$$

- Simultaneous update

Programming - Homework

- Write a Python program that implements a function **computeCost** to compute the Mean Squared Error (MSE) for univariate linear regression. The function should take the following parameters:

  - **X**: A list of m data points (a list of m real numbers).
  - **Y**: A list of m target values (a list of m real numbers corresponding to X).
  - **bias**: The intercept parameter (a real number).
  - **weight**: The slope parameter (a real number).

- The function should return the cost, computed using the MSE formula.

Programming - Homework

- Write a Python program that calculates the cost for different combinations of parameters and prints the minimum cost along with the optimal parameters. The program should include a function findMinCost that takes the following parameters:
  - **X**: A list of m data points (a list of m real numbers).
  - **Y**: A list of m target values (a list of m real numbers corresponding to X).
  - **bias_lst**: A list of k possible values for the intercept parameter (a list of k real numbers).
  - **weight_lst**: A list of j possible values for the slope parameter (a list of j real numbers).
- The function should compute the cost for each combination of intercept (from bias_lst) and slope (from weight_lst), and return the minimum cost and the corresponding parameters.

Reading - Homework

- Machine Learning
  - Resource R1
  - Book B1: 1.3, 2.1, 2.6, 3.1
  - Book B3: 2.6