



**Machine Learning**  
**ICT-4261**

**By-**  
**Dr. Jesmin Akhter**

Professor  
Institute of Information Technology  
Jahangirnagar University

# Contents

**The course will mainly cover the following topics:**

- ✓ A Gentle Introduction to Machine Learning
- ✓ Linear Regression
- ✓ Logistic Regression
- ✓ Naive Bayes
- ✓ Support Vector Machines
- ✓ Decision Trees and Ensemble Learning
- ✓ Clustering Fundamentals
- ✓ Hierarchical Clustering
- ✓ Neural Networks and Deep Learning
- ✓ Unsupervised Learning.....

# Outline

- ✓ A Gentle Introduction to Machine Learning
  - Unsupervised learning
  - Reinforcement learning
- ✓ Regression

# Types of Unsupervised Learning

- ✓ **Unsupervised** learning can be broken down into three main tasks:
  - Clustering
  - Association rules
  - Dimensionality reduction.

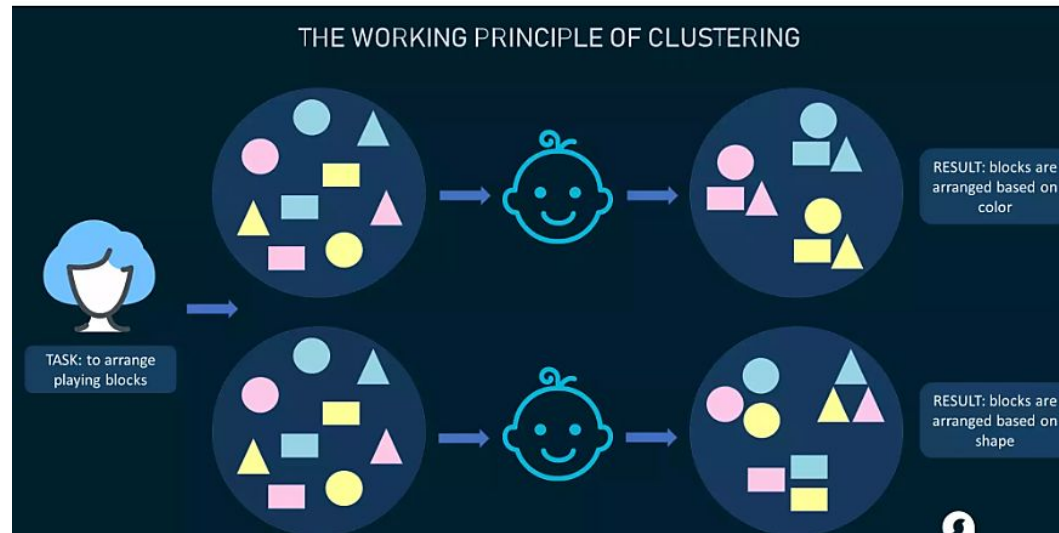
## **Unsupervised Algorithm:**

1. K-means clustering
2. KNN (k-nearest neighbours)
3. Hierarchal clustering
4. Anomaly detection
5. Neural Networks
6. Principle Component Analysis
7. Apriori algorithm

# Types of Unsupervised Learning

## Clustering Algorithms

- ✓ Clustering algorithms only interpret the input data and find natural groups or clusters in feature space



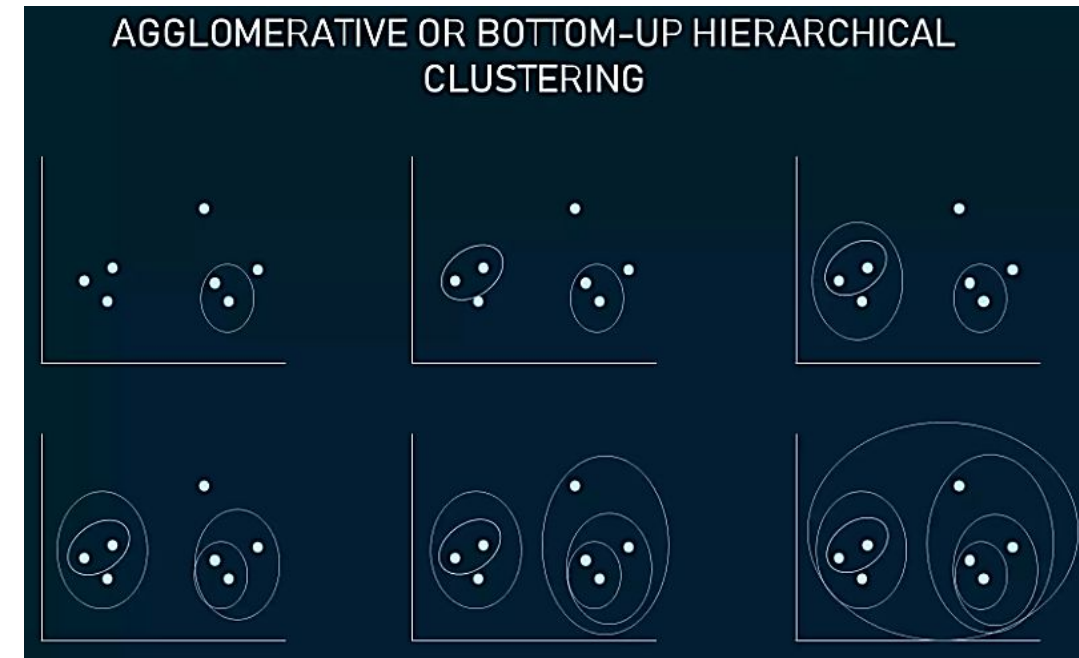
- ✓ Clustering is a popular type of unsupervised learning approach. You can even break it down further into different types of clustering; for example:
  - **Exclusive clustering:** or “hard” clustering
    - It is the kind of grouping in which one piece of data can belong only to one cluster.
  - **Overlapping clustering:**
    - A soft cluster in which a single data point may belong to multiple clusters with varying degrees of membership.

# Types of Unsupervised Learning

## ✓ Clustering Algorithms

### – Hierarchical Clustering:

- Hierarchical clustering develops a hierarchy of clusters by merging or splitting them depending on their similarity. Here, two close cluster are going to be in the same cluster.
- In case you start with all data items attached to the same cluster and then perform splits until each data item is set as a separate cluster, the approach will be called top-down or *divisive* hierarchical clustering.
- Two clusters that are closest to one another are then merged into a single cluster. The merging goes on iteratively till there's only one cluster left at the top. Such an approach is known as bottom-up or *agglomerative*.
- The example shows how seven different clusters (data points) are merged step by step based on distance until they all create one large cluster.

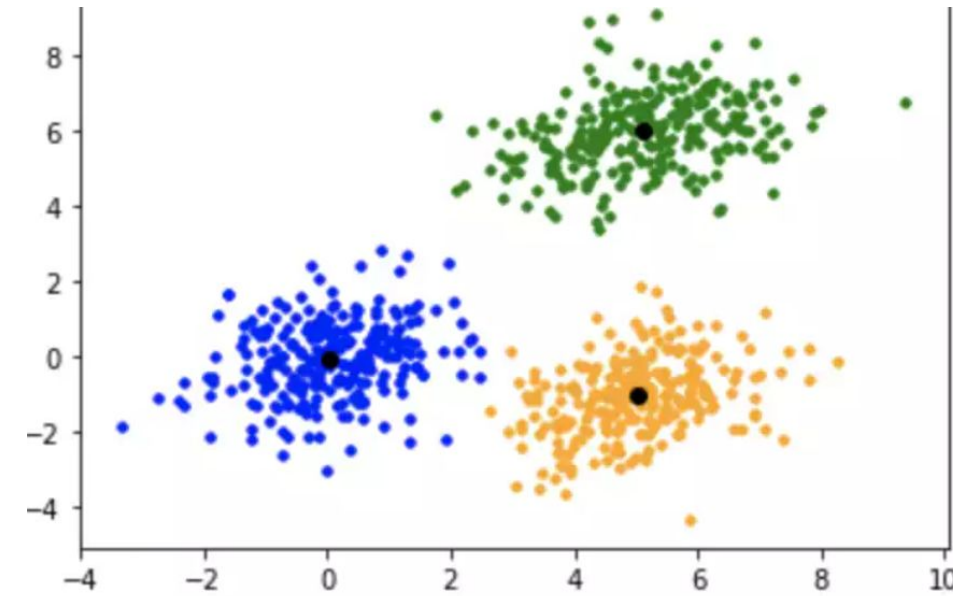


# Types of Unsupervised Learning

## ✓ Clustering Algorithms

### – K-Means Clustering

- In K-means clustering, data is grouped in terms of characteristics and similarities.
- K is a letter that represents the number of clusters. For example, if  $K=5$ , then the number of desired clusters is 5. If  $K=10$ , then the number of desired clusters is 10.
- It puts the data points into the predefined number of clusters  $K$ .
- Each data item then gets assigned to the nearest cluster center, called *centroids* (black dots in the picture). The latter act as data accumulation areas.
- The procedure of clustering may be repeated several times until the clusters are well-defined.



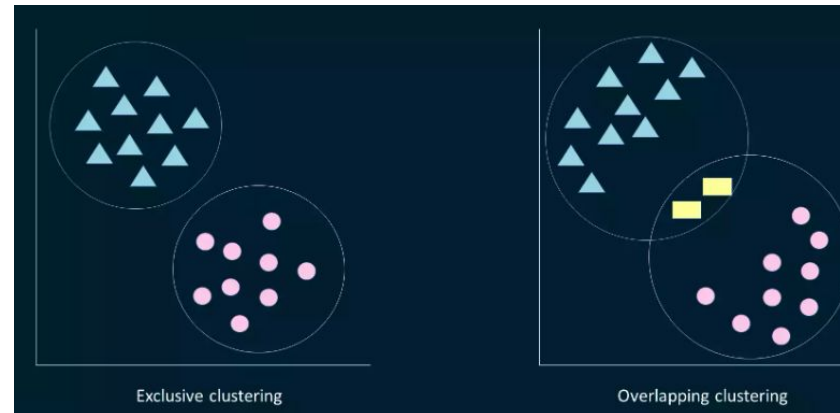
Ideal clustering with a single centroid in each cluster. Source: [GeeksforGeeks](#)

# Types of Unsupervised Learning

## ✓ Clustering Algorithms

### – Fuzzy K-means

- It is an extension of the K-means algorithm used to perform overlapping clustering. Unlike the K-means algorithm, fuzzy K-means implies that data points can belong to more than one cluster with a certain level of closeness towards each.
- The closeness is measured by the distance from a data point to the centroid of the cluster. So, sometimes there may be an overlap between different clusters.

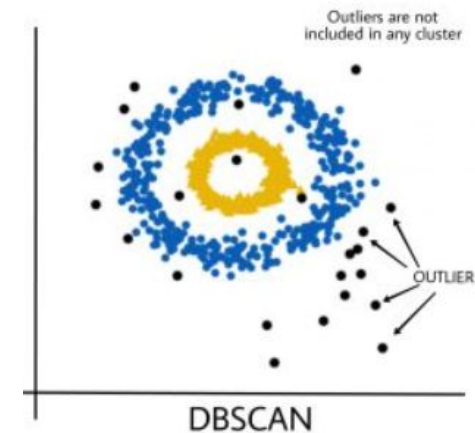
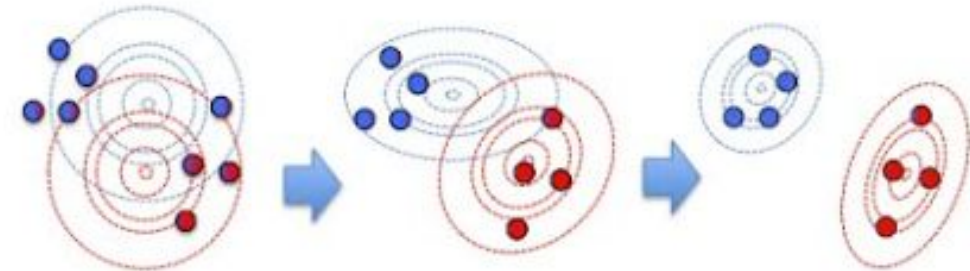




# Types of Unsupervised Learning

## ✓ Clustering Algorithms

- **Gaussian Mixture Models (GMMs)** is an algorithm used in probabilistic clustering.
  - Models assume that there is a certain number of Gaussian distributions, each representing a separate cluster.
  - The algorithm is basically utilized to decide which cluster a particular data point belongs to.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** DBSCAN groups data points based on their density, identifying clusters of high-density regions and classifying outliers as noise.

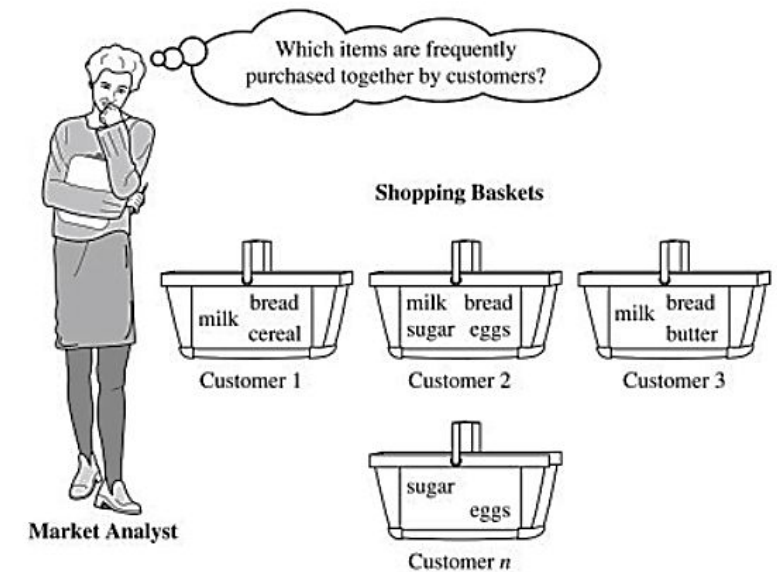


# Types of Unsupervised Learning

## ✓ Association Rule Mining

- Association rule mining focuses on discovering interesting relationships or patterns between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X (suppose a bread) also tend purchase Y (Butter/Jam) item. The widely used algorithm for association rule mining is the **Apriori** algorithm.

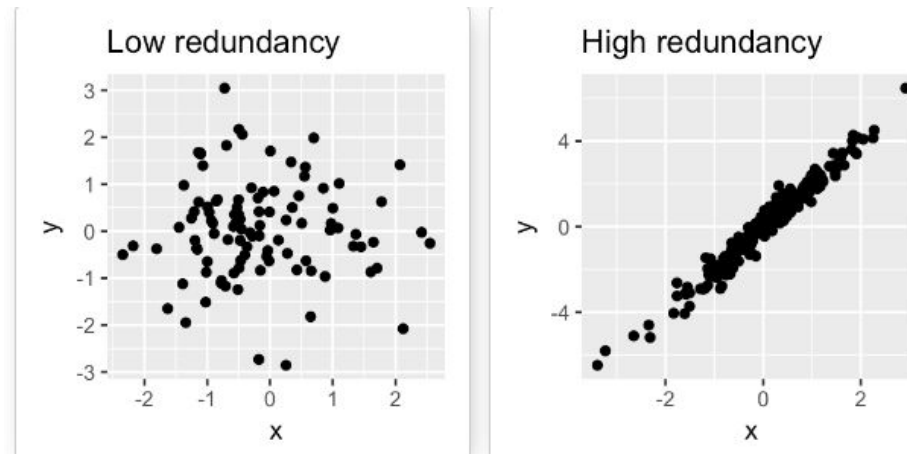
- A real-life example of this is market basket analysis, where retailers analyze customer purchase data to identify relationships between products frequently bought together. For instance, this analysis might reveal that customers who purchase diapers also tend to buy baby wipes



# Types of Unsupervised Learning

✓ **Dimensionality Reduction Algorithms** Dimensionality reduction techniques are used to reduce the number of input variables or features while retaining meaningful information. Some popular dimensionality reduction algorithms include:

- **Principal Component Analysis (PCA):** PCA transforms the original features into a lower-dimensional space while preserving the maximum amount of information.
- The PCA method is particularly useful when the variables within the data set are highly correlated. Correlation indicates that there is redundancy in the data. Due to this redundancy, PCA can be used to reduce the original variables into a smaller number of new variables ( **principal components**) explaining most of the variance in the original variables.

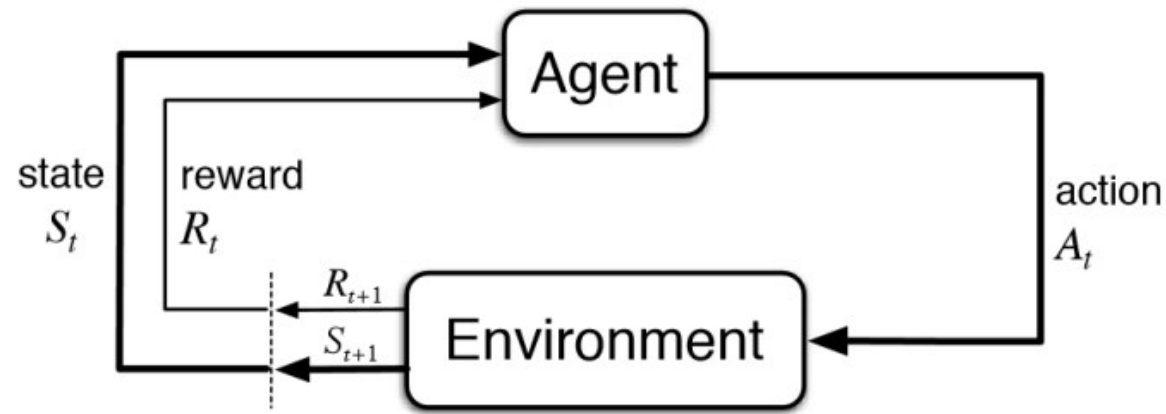


# What is the difference between supervised and unsupervised learning?

- ✓ Supervised learning requires labeled data with input features and corresponding output labels, while unsupervised learning aims to discover patterns or structures in unlabeled data without predefined output labels.

# Reinforcement machine learning

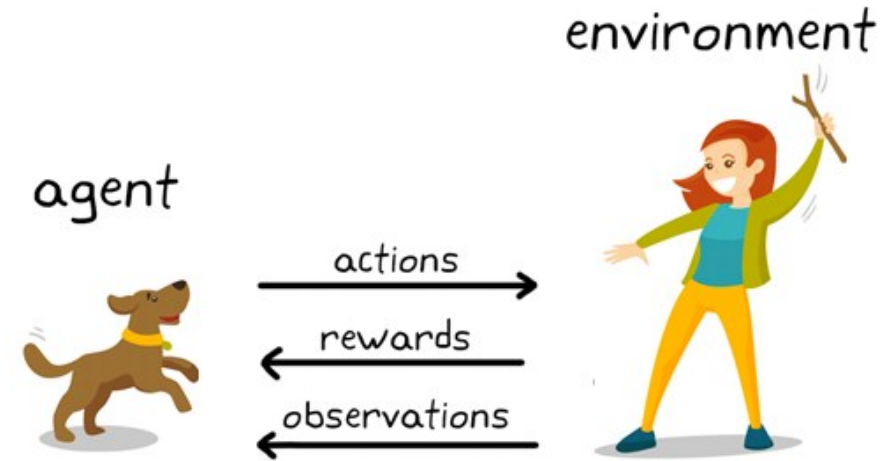
- ✓ Reinforcement Learning(RL) enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.
- ✓ Here, agents are self-trained on reward and punishment mechanisms.
- ✓ It can take actions and interact with it.
- ✓ Reinforcement machine learning algorithm isn't trained using sample data.
- ✓ A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem.



Basic Diagram of Reinforcement Learning

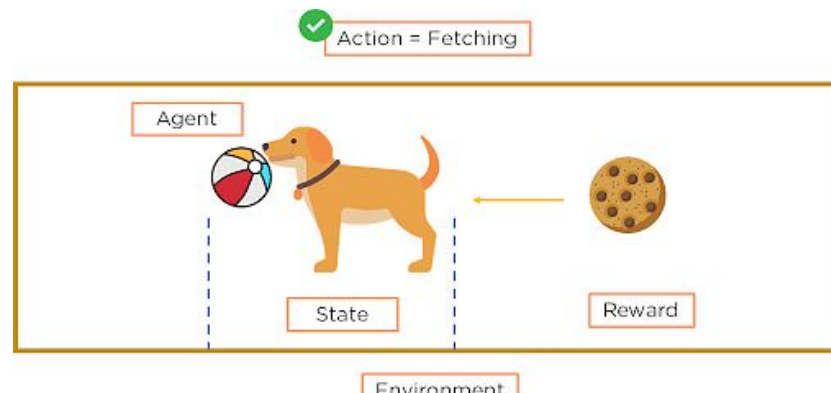
# Reinforcement machine learning

- ✓ Through a series of Trial and Error methods, an agent keeps learning continuously in an interactive environment from its own actions and experiences.
- ✓ The only goal of it is to find a suitable action model which would increase the total cumulative reward of the agent.
- ✓ It learns via interaction and feedback.
- ✓ You can see a dog and a master. Let's imagine you are training your dog to get the stick. Each time the dog gets a stick successfully, you offered him a feast (a bone ).
- ✓ Eventually, the dog understands the pattern, that whenever the master throws a stick, it should get it as early as it can to gain a reward (a bone) from a master in a lesser time.



# Important Terms in Reinforcement Learning

- ✓ **Agent:** Agent is the model that is being trained via reinforcement learning. It is the sole decision-maker and learner
- ✓ **Environment:** a physical world where an agent learns and decides the actions to be performed
- ✓ **Action:** All possible steps that can be taken by the model/agent
- ✓ **State:** The current position/ condition returned by the model/the current situation of the agent in the environment
- ✓ **Reward:** To help the model move in the right direction, it is rewarded/points are given to it to appraise some action. It's usually a scalar value and nothing but feedback from the environment
- ✓ **Policy:** Policy determines how an agent will behave at any time. It acts as a mapping between Action and present State. The agent prepares strategy(decision-making) to map situations to actions.
- ✓ **Value** — Future reward that an agent would receive by taking an action in a particular state



# Outline

- ✓ Linear Regression
  - Linear models
  - A bidimensional example
  - **Hypothesis function for Linear Regression**
  - **Cost function**



# Regression

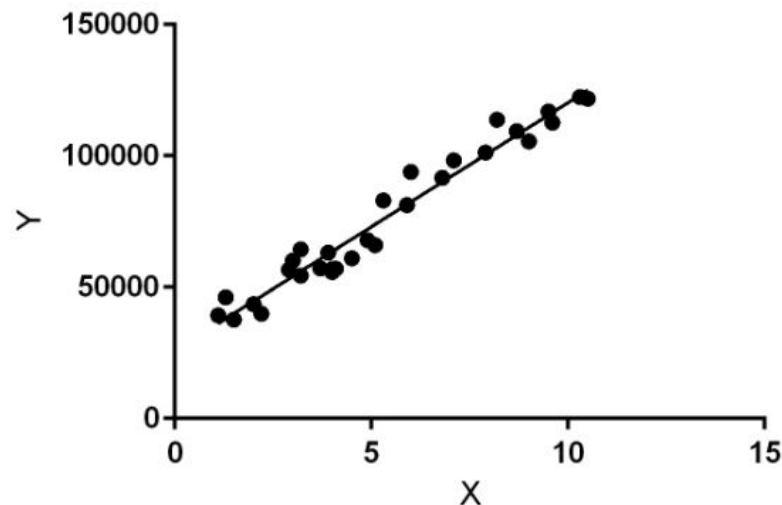
- ✓ Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features.
- ✓ When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression.
- ✓ It predicts the continuous output variables based on the independent input variable like the prediction of house prices based on different parameters like house age, distance from the main road, location, area, size etc.

# Linear Regression

- ✓ The goal of the algorithm is to find the best linear equation that can predict the value of the dependent variable based on the independent variables.
- ✓ The equation provides a straight line that represents the relationship between the dependent and independent variables.
- ✓ The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s).
- ✓ Linear regression is used in many different fields, including finance, economics, and psychology, to understand and predict the behavior of a particular variable.
  - For example, in finance, linear regression might be used to understand the **relationship** between a **company's stock price and its earnings** or to predict the future value of a currency based on its past performance.

# Linear Regression

- ✓ Y is a dependent or target variable and X is an independent variable also known as the predictor of Y.
- ✓ There are many types of functions or modules that can be used for regression.
  - A linear function is the simplest type of function.
- ✓ X may be a single feature or multiple features representing the problem.
- ✓ Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). Hence, the name is Linear Regression.
- ✓ In the figure, X (input) is the work experience and Y (output) is the salary of a person.



# Hypothesis function for Linear Regression

- ✓ salary of a person
  - $m$  = number of training examples
  - $x$  = input variables / features
  - $y$  = output variable "target" variables
    - $(x,y)$  - single training example
    - $(x_i, y_j)$  - specific example ( $i$ th training example)
- ✓ We have assumed that our independent feature is the experience  $x$  and the respective salary  $y$  is the dependent variable.

$x(\text{Work experience in years})$	$y(\text{salary})$
2	3000
3	4000
4	5000
5	6000
10	11000
12	13000

# Hypothesis function for Linear Regression

- ✓ Let's assume there is a linear relationship between X and Y then the salary can be predicted using:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

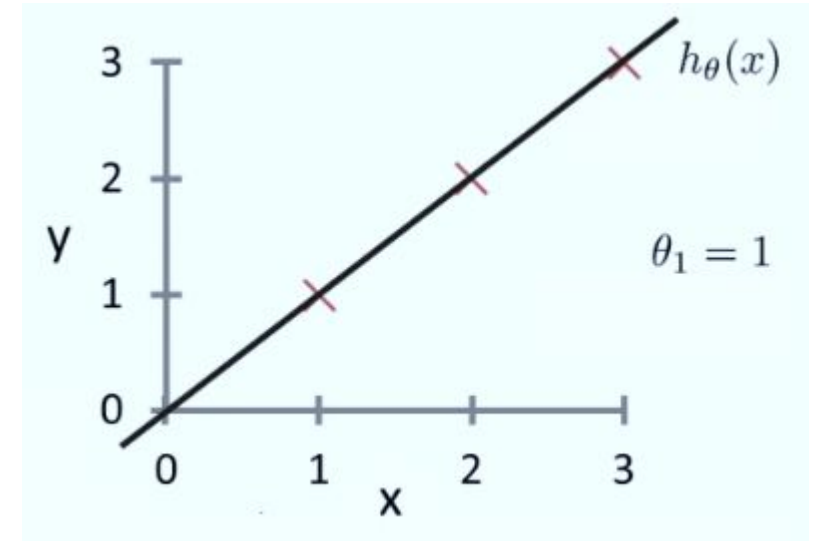
- ✓ The model gets the best regression fit line by finding the best  $\theta_0$  and  $\theta_1$  values.
- ✓  $\theta_0$ : intercept
- ✓  $\theta_1$ : coefficient of x or gradient
- ✓ Chosen these parameters so  $h_{\theta}(x)$  is close to y for our training examples
- ✓ Once we find the best  $\theta_0$  and  $\theta_1$  values, we get the best-fit line. So when we are finally using our model for prediction, it will predict the value of  $h_{\theta}(x)$  for the input value of x.
  - Different values give you different functions
    - If  $\theta_0$  is 1.5 and  $\theta_1$  is 0 then we get straight line parallel with X along 1.5
    - If  $\theta_1$  is  $> 0$  then we get a positive slope
  - Think of  $h_{\theta}(x)$  as a "y imitator" - it tries to convert the x into y, and considering we already have y we can evaluate how well  $h_{\theta}(x)$  does this

# Linear models

- ✓ We can learn with a larger number of features
  - So may have other parameters which contribute towards a price of houses
    - Size
    - Age
    - Number bedrooms
    - Number floors
  - $x_1, x_2, x_3, x_4$
  - With multiple features becomes hard to plot
    - Can't really plot in more than 3 dimensions
    - Notation becomes more complicated too
      - Best way to get around with this is the notation of linear algebra
      - Gives notation and set of things you can do with matrices and vectors
- ✓ Now we have multiple features. A linear model is based on the assumption that it's possible to approximate the output values through a regression process based on the rule. Hypothesis can be written
  - $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 = \theta_0 + \sum_{i=1}^m \theta_i x_i$

# Implementation: cost function(Cont.)

- ✓ **Hypothesis** - is like your prediction machine, throw in an  $x$  value, get a putative  $y$  value
- ✓ The cost function or the loss function is nothing but the error or difference between the predicted value  $h_{\theta}(x)$  and the true value  $Y$ . It is the **Mean Squared Error (MSE)** between the predicted value and the true value.
- ✓ **Cost function** - is a way to, use your training data, determine values for your  $\theta$  values which make the hypothesis as accurate as possible
  - This cost function is reasonable choice for most regression functions
  - Probably most commonly used function



Minimize  $J(\theta_0, \theta_1)$   
 $\theta_0, \theta_1$  Cost function

# Cost function - a deeper look

- ✓ Lets consider some intuition about the cost function and why we want to use it
  - The cost function determines parameters
  - The value associated with the parameters determines how your hypothesis behaves, with different values
  - To achieve the best-fit regression line, the model aims to predict the target value. So the **Cost function** updates the  $\theta_0$  and  $\theta_1$  values, to reach the best value that minimizes the error between the predicted  $h_\theta(x)$  value and the true  $y$  value.
- ✓ Simplified hypothesis
  - Assumes  $\theta_0 = 0$
- ✓ Cost function and goal here are very similar to when we have  $\theta_0$ , but with a simpler parameter
  - Simplified hypothesis makes visualizing cost function  $J(\theta_1)$  a bit easier
- ✓ So hypothesis pass through 0,0

$$h_\theta(x) = \theta_1 x$$
$$\theta_0 = 0$$

$$\theta_1$$

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

$$\underset{\theta_1}{\text{minimize}} J(\theta_1)$$



# Cost function - a deeper look (Cont.)

- Two key functions we want to understand

- $h_{\theta}(x)$

- Hypothesis is a function of  $x$  - function of what the size of the house is

- $J(\theta_1)$

- Is a function of the parameter of  $\theta_1$

- So for example

- $\theta_1 = 1$

- $J(\theta_1) = 0$

- Plot

- $\theta_1$  vs  $J(\theta_1)$

- Data

- 1)

- »  $\theta_1 = 1$

- »  $J(\theta_1) = 0$

- 2)

- »  $\theta_1 = 0.5$

- »  $J(\theta_1) = \sim 0.58$

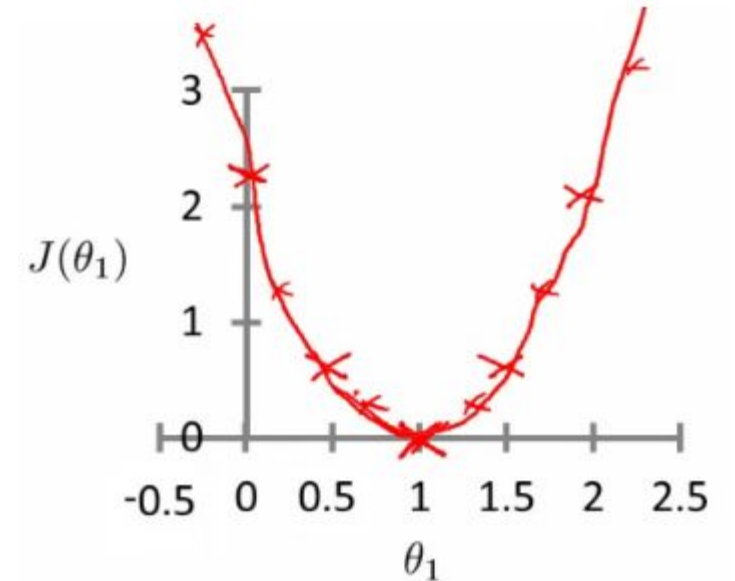
- 3)

- »  $\theta_1 = 0$

- »  $J(\theta_1) = \sim 2.3$

# Cost function - a deeper look (Cont.)

- ✓ If we compute a range of values plot
  - $J(\theta_1)$  vs  $\theta_1$  we get a polynomial (looks like a quadratic)
- ✓ The optimization objective for the learning algorithm is find the value of  $\theta_1$  which minimizes  $J(\theta_1)$ 
  - So, here  $\theta_1 = 1$  is the best value for  $\theta_1$



# Thank You

