Python Fundamentals

Autonise

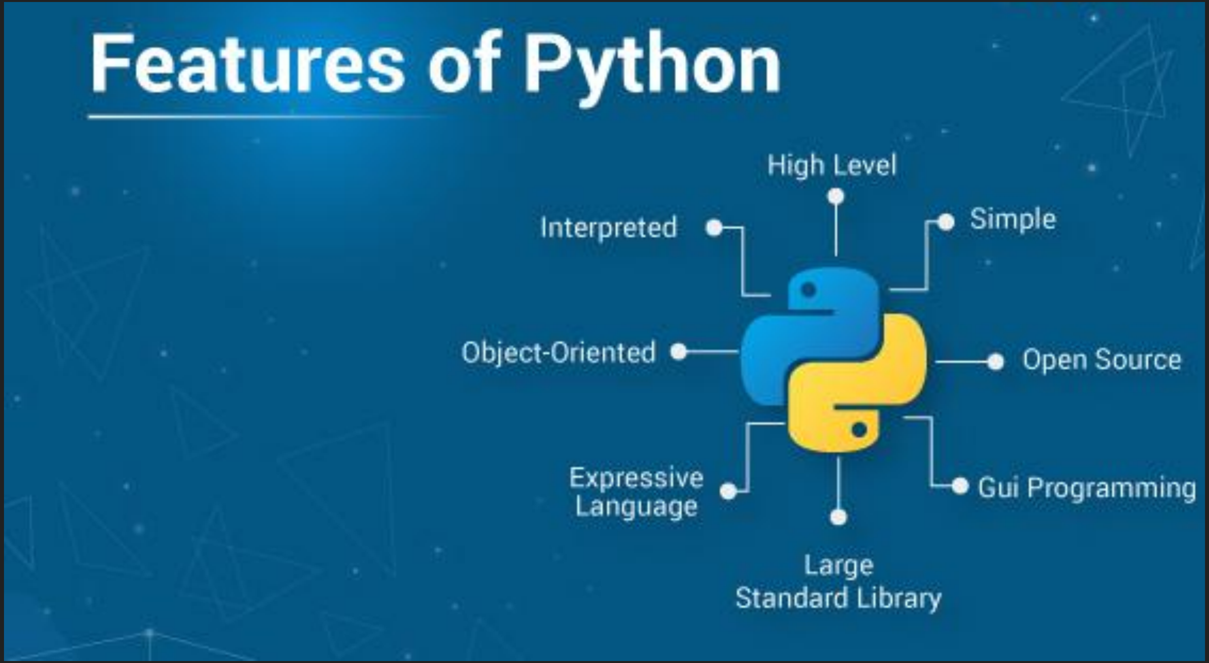
Objectives

1. Introduction and Installation
2. Basic python syntax
3. Functions
4. Data structures(List,Tuple,Dict)

BRIEF History of Python

- Created by Guido van Rossum and first released in 1991
- It is created to emphasize the code readability
- Its language construct and object-oriented approach aim to help programmers write clear, logical code

Let’s Talk about python



Anaconda

- Anaconda is a Package Manager for python
- It pretty much manages everything required for python.

Installation

Anaconda Installers

Windows

Python 3.7

64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (423 MB)

Python 2.7

64-Bit Graphical Installer (413 MB)

32-Bit Graphical Installer (356 MB)

MacOS

Python 3.7

64-Bit Graphical Installer (442 MB)

64-Bit Command Line Installer (430 MB)

Python 2.7

64-Bit Graphical Installer (637 MB)

64-Bit Command Line Installer (409 MB)

Linux

Python 3.7

64-Bit (x86) Installer (522 MB)

64-Bit (Power8 and Power9) Installer (276 MB)

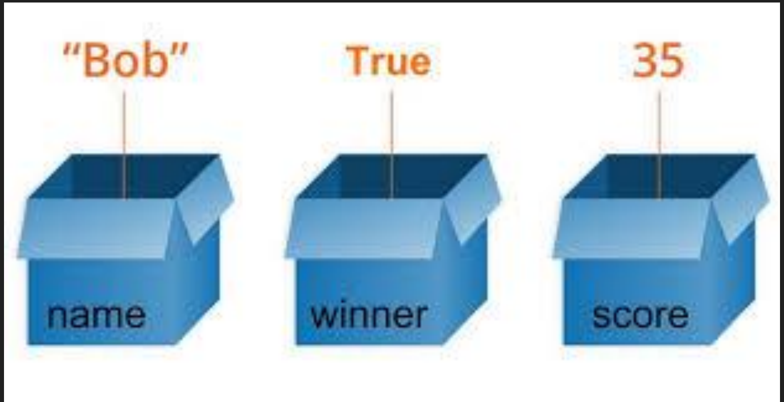
Python 2.7

64-Bit (x86) Installer (477 MB)

64-Bit (Power8 and Power9) Installer (295 MB)

Variable

- Imagine Variable as a box with a label on it.
- Each Variable contains Data of specific type



Data Types

- The Label on the box is called Data type
- These are Fundamental Data types in python
    1. Int(For Integer Numbers)
    2. float(For Real Numbers)
    3. Str(For Names)
    4. bool(For True or False)
- Conversion of one type to other is known as **TypeCasting**

Operators

There are two types of Operators(Arithmetic/Logical):

1. **Arithmetic operators**

   1.1 Addition(+)

   1.2 subtraction(-)

   1.3 Multiplication(*)

   1.3 Division(/)

   1.4 Floor Division(//)

   1.5 Modulo(%)

   1.6 Power(**)

Logical Operators

- Greater than(>), Less Than (<), equals(==), Not Equals(!=),less than equals(<=),greater than equals(>=)
- Key words
  1. not
  2. and
  3. or

String operators

There are 4 operators on strings

“+” -> Used to concatenate strings

“*” -> Repeat the string again and concatenate

[]   -> Used to get the character and particular position

[:] -> Slice characters in certain positions

[::] -> skip certain number of characters

Methods in string

IF Else Block

If expression:

    Perform a task

else:

    Perform other task

For Loop

```
for i in range(5):
    print("Hello World")
```

- Here the variable **i** is known as **iterator**
- **range(5)** is known as **iterable**

While loop

- While loop is used to run program indefinitely until certain **condition** is met.

while condition:

    Perform the task again and again.

break/continue keywords

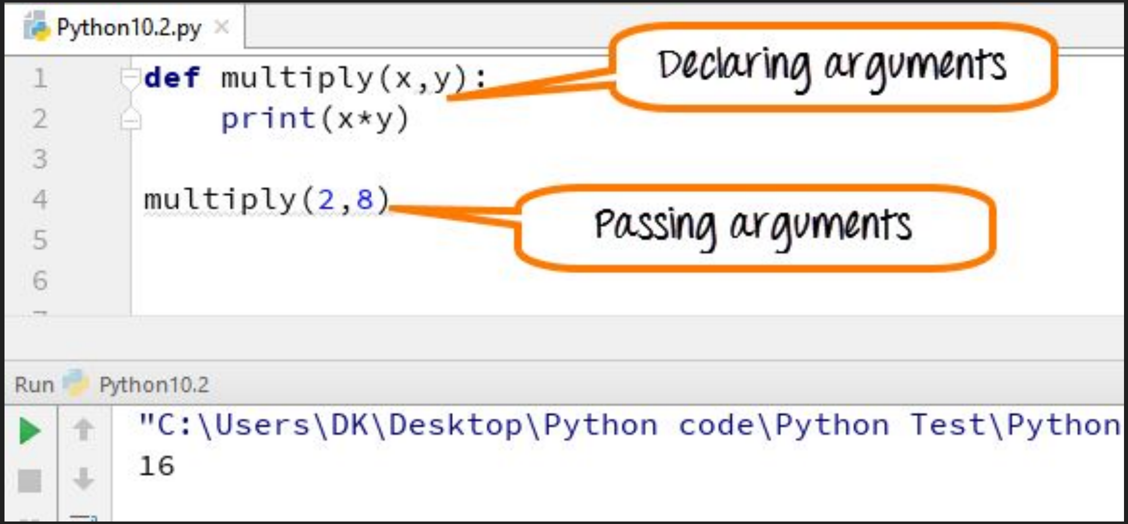
- Break is used to come outside of the loop.
- Continue is used to come to the beginning of the loop

Functions/Methods

- A Function is a pieces of reusable code.
- This make our life easier.
- Method is nothing but a function in **Object Oriented Programming**
- Some inbuilt-functions (max,min,abs,range)

Writing own functions

- A function follows indentation rules same as if conditions and for loops
- The inputs to the function is called **“Arguments”**
- The outputs are given by **return** key word



Range function in python

Syntax:

range(start, stop, step)

Example: range(0,11,2)

Iterable for all the even numbers less than 11.

Other Inbuilt functions: sorted,min, max

List

- Sequence of Data having same data type
- Example: lst = [1,2,3]
- Methods: clear(), append(),reverse(),sort(),copy(),pop()
- Operators (*,[*],+)
- It is mutable object

Tuple

- Has sequence of Data. Data may be of different data types
- student = (“alice”,21,True)
- Methods: count(), index()
- Tuple is Immutable

Set {}

Set is similar to mathematical set. It contains distinct elements. All elements should be immutable.

Methods: add, discard, pop(),clear(),copy()

A set is mutable object.

S1 = {1,3,5,7}

S1.add(5)

s1.remove(3)

Dictionary

Dictionary is the data-structure, which stores keys,values for look-up. This is also called **Hashmap** in other languages. The keys can only be **immutable objects** i.e int,bool,String

M = {}

M[“sunday”] = 0

M[“monday] = 1

Methods: items(),pop(),copy(),clear()

Python Data Science Packages

IDE

Jupyter Notebook

Google Colab

Objectives

1. What is a Python Package?
2. Numpy
3. Pandas
4. Matplotlib

Python Package

A Package is collection python modules. Each python module contains defined set of functions, objects and variables.

Pip is used to install packages in python

```
!pip install package_name
```

```
import package_name as alias
```

A python library is collections of python packages

Numpy

Numpy is a mathematical computing library which supports large multi-dimensional, arrays and matrices.It also contains high level mathematical functions to operate on these arrays.

import numpy as np

Eg:

Lst = [1,2,3]

array = np.array(Lst)

Pandas

Pandas is a package written on top of numpy for data manipulation and analysis. It makes working with relational and labelled data simple and intuitive

```
import pandas as pd

Lst = [1,2,3]

Series = pd.series(Lst)

Dataframe = pd.read_excel(excel_sheet.xlsx)
```

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It makes quality plots with very few lines of code.

Pyplot is a package under matplotlib for 2D- visualizations

import matplotlib.pyplot as plt

plt.plot(x)

Introduction to Machine Learning

Autonise

Objectives

1. Introduction to Machine Learning
2. Supervised Learning Fundamentals
3. Univariable Linear Regression
4. Multi-variable Linear Regression

Introduction

Machine Learning is the set of algorithms which learn by experience of previous data.

Examples:

1. Housing Price prediction
2. Spam Email classification
3. Face Recognition
4. Recommender systems
5. Contextual ads

Supervised Learning:

Learning from labelled data. Algorithms understand the patterns in the data, by learning from previous data which already has ground truth.

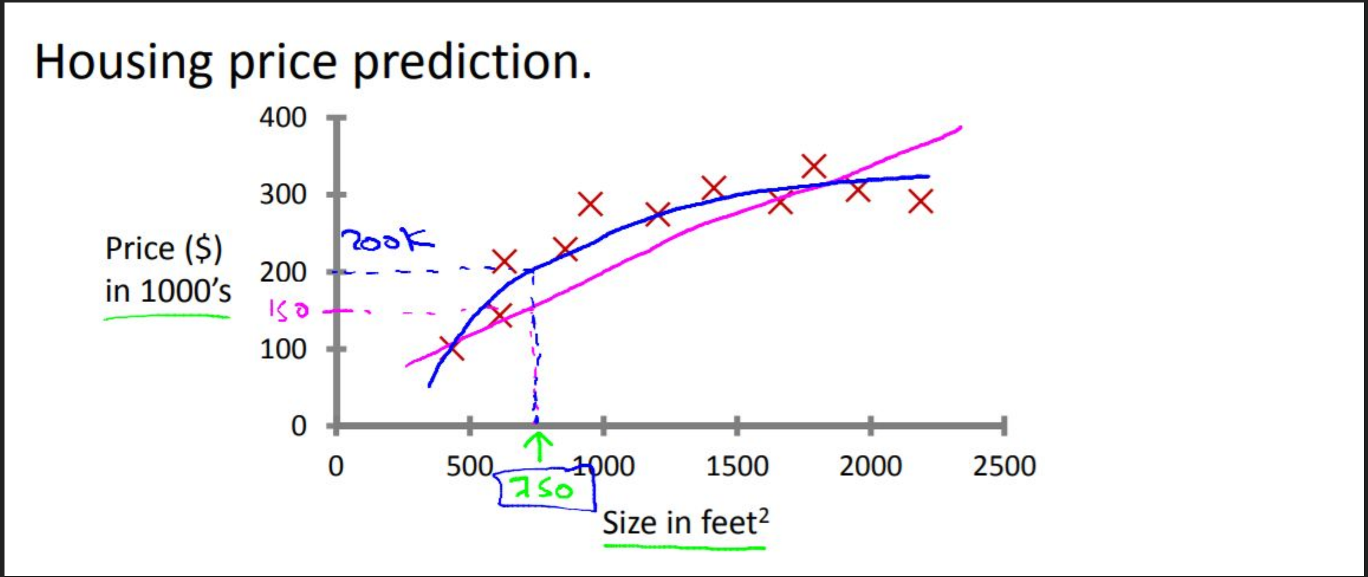
Unsupervised Learning:

Algorithms learn from the data without having given the ground truth.

Regression Problem

Regression Problem: Predicting a continuous variable.

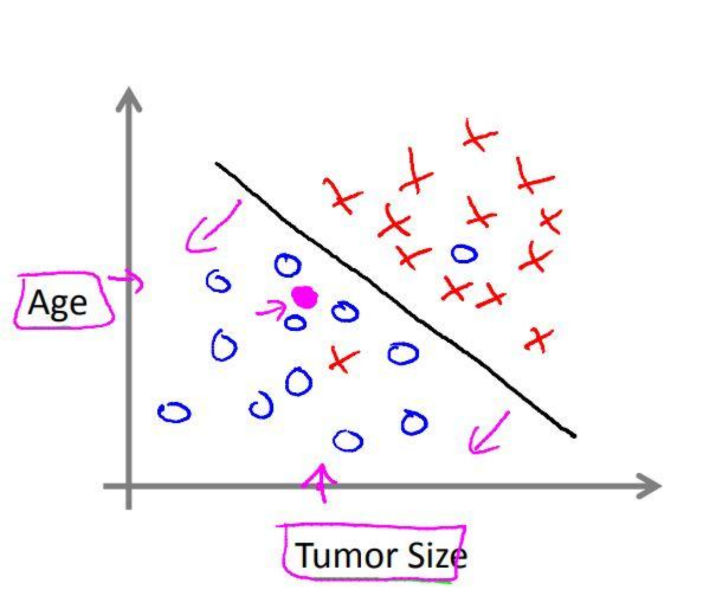
Predicting house price given various factors like area, #rooms, location etc.,



Classification Problem: Predicting a categorical variable.

Example:

Classifying a tumour is malignant or Benign



Features

Features are the measurable quantities which will be observed for a machine learning problem.

Examples:

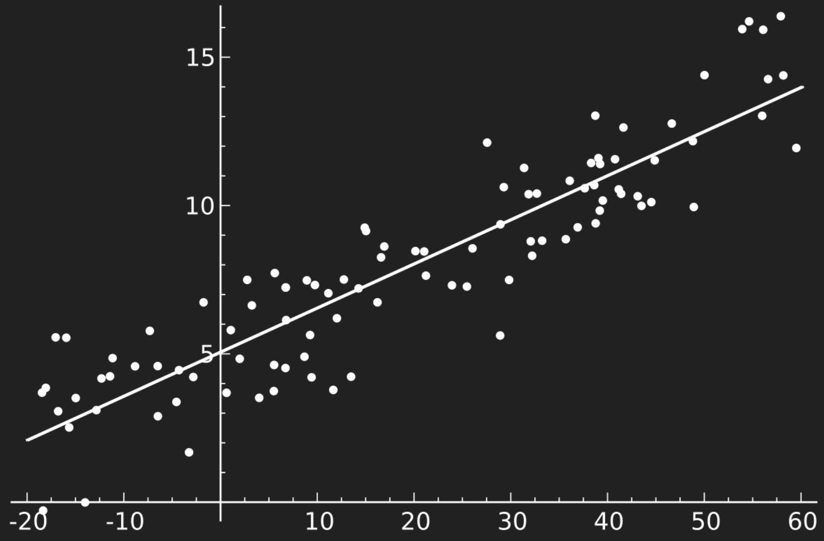
Housing price prediction: number of rooms, area of house, location, age

Cancer classification: clump thickness, uniformity of cell shape, uniformity of cell size

Label is a feature which needs to be predicted by the algorithm.(price of house, type of cancer(0/1)

Linear Regression

Linear regression attempts to model the relationship between the variables by fitting a linear equation to observed data.

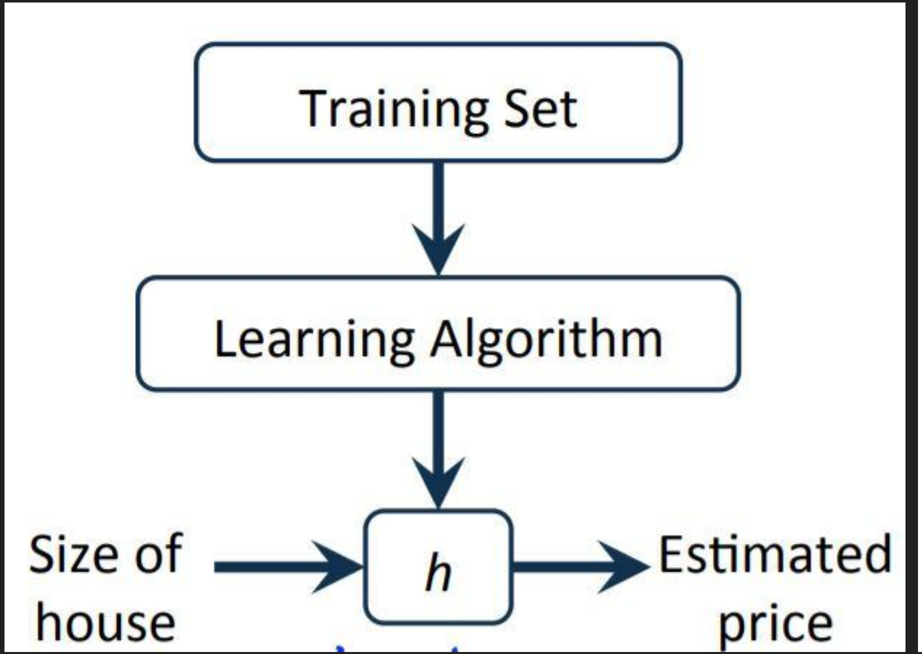


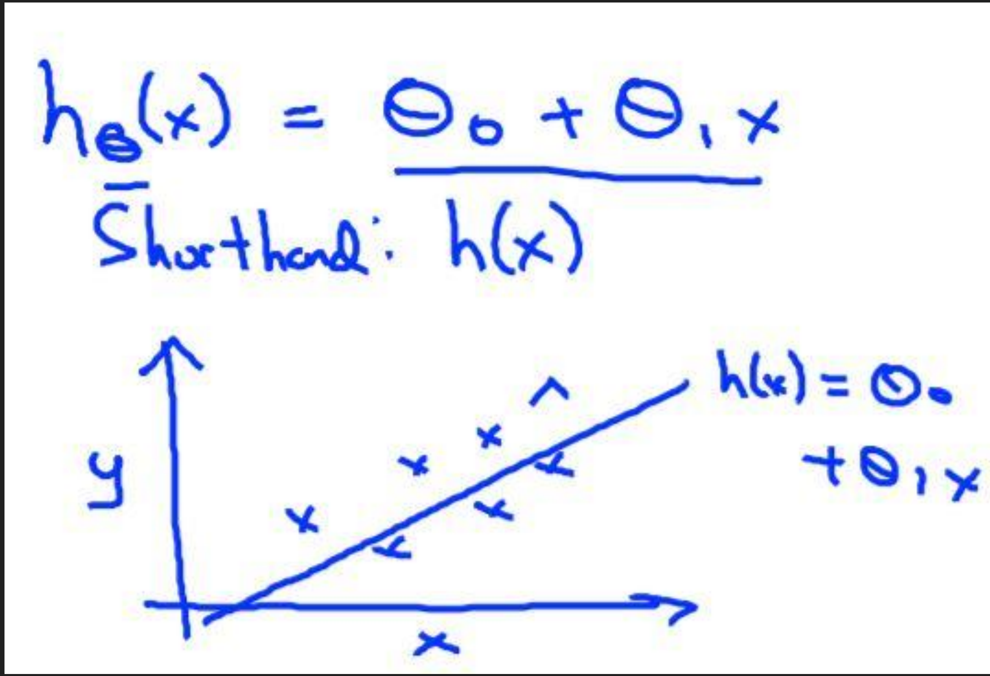
Single variable Linear Regression

Problem: Predict the price of a house, given the area of the house.

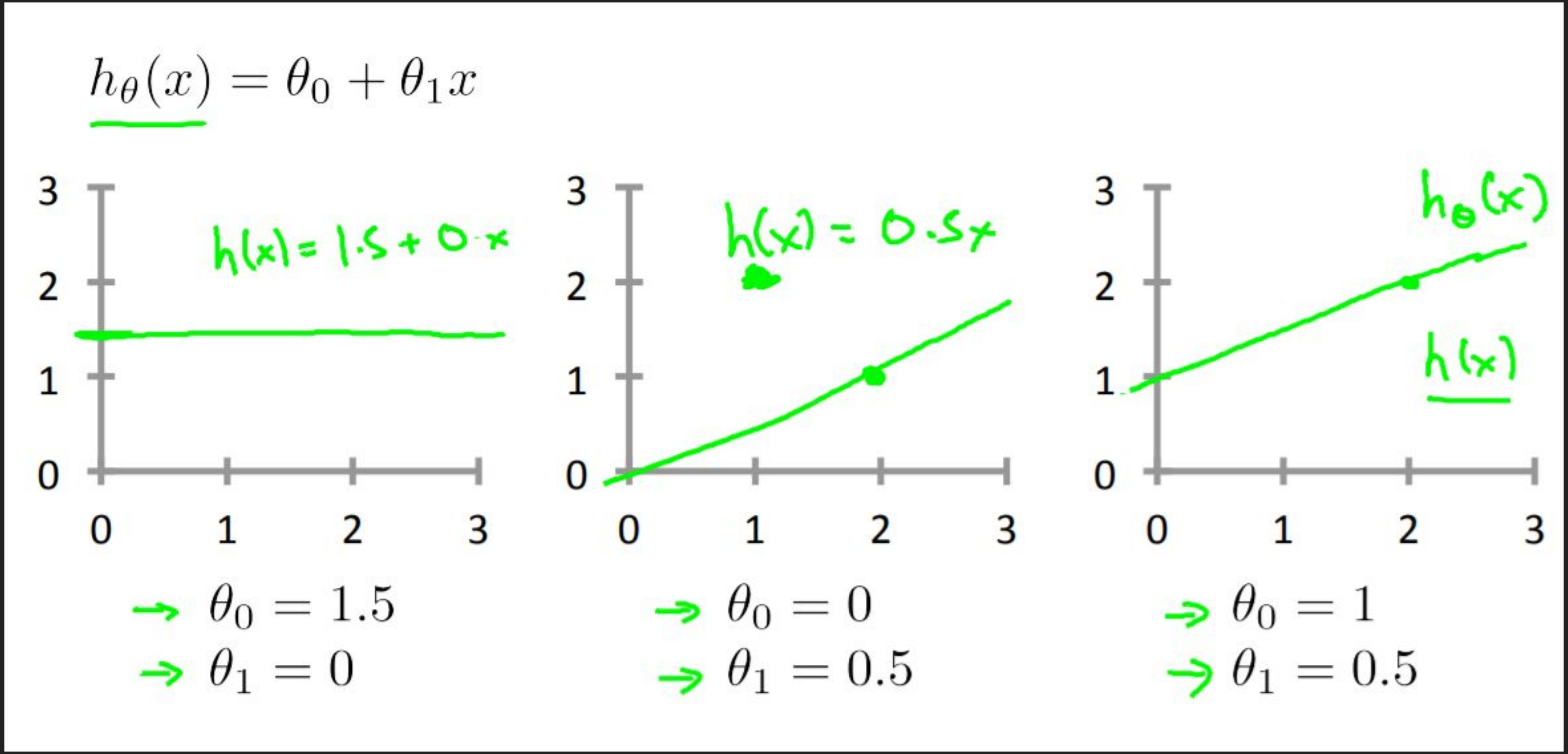
Size in feet^2 (X)	Price is Lakhs(y)
2104	46
1416	25
1543	32
850	18
...	...

Hypothesis(h)





Parameters



$h_\theta(x) = \theta_0 + \theta_1 x$
h(x) = 1.5 + 0·x
h(x) = 0.5x
$h_\theta(x)$
h(x)
$\theta_0 = 1.5$
$\theta_1 = 0$
$\theta_0 = 0$
$\theta_1 = 0.5$
$\theta_0 = 1$
$\theta_1 = 0.5$

Cost Function

Cost/Loss function is the measure of distance between predicted values and ground truth.

Mean squared error:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$

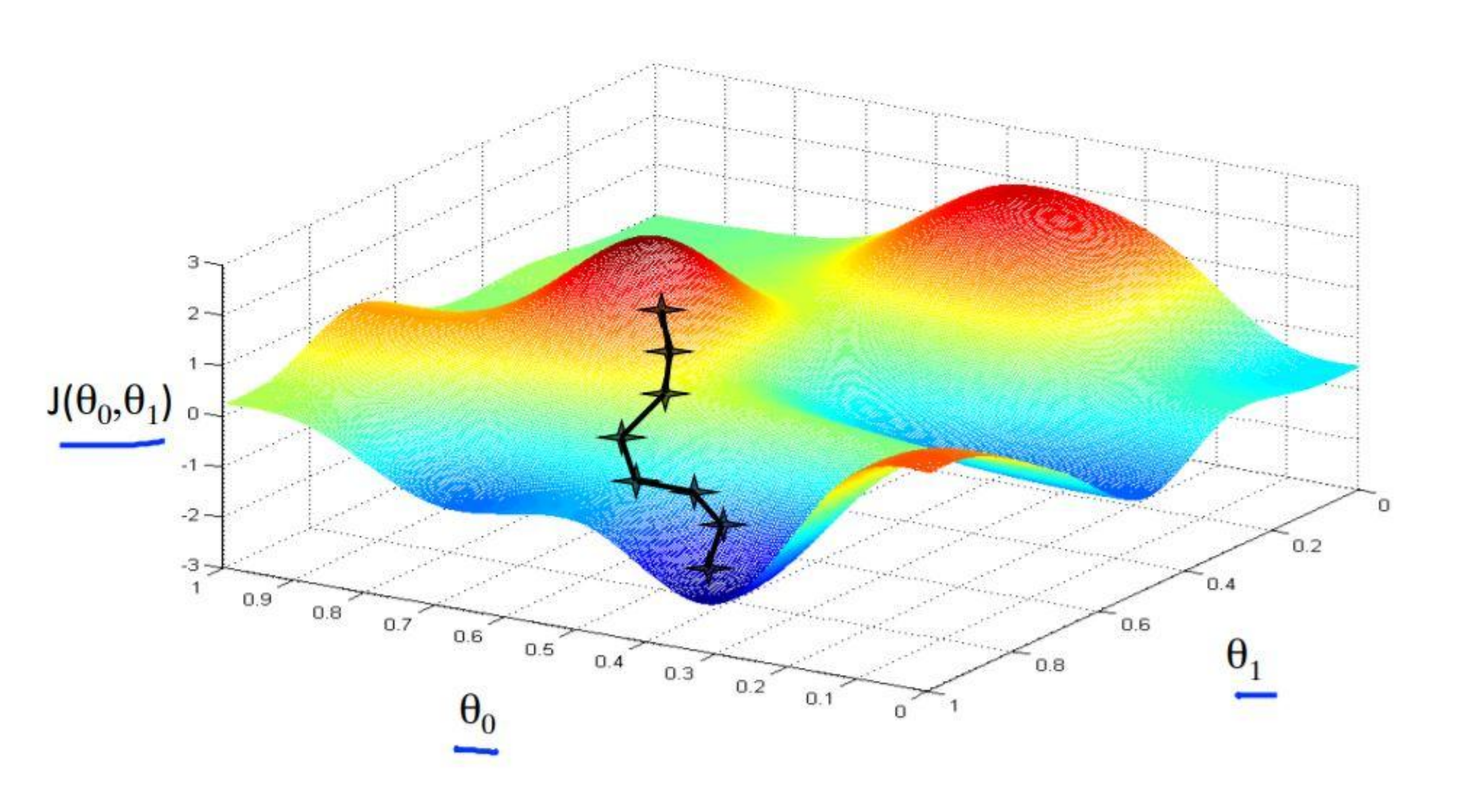
Optimizer

Optimizer is an algorithm which finds the parameters to produce minimum cost.

1. Start with some values of theta0,theta1
2. Keep changing to them to reduce the cost.

   Until we hopefully find the minimum

Parametres vs Cost



Gradient Descent

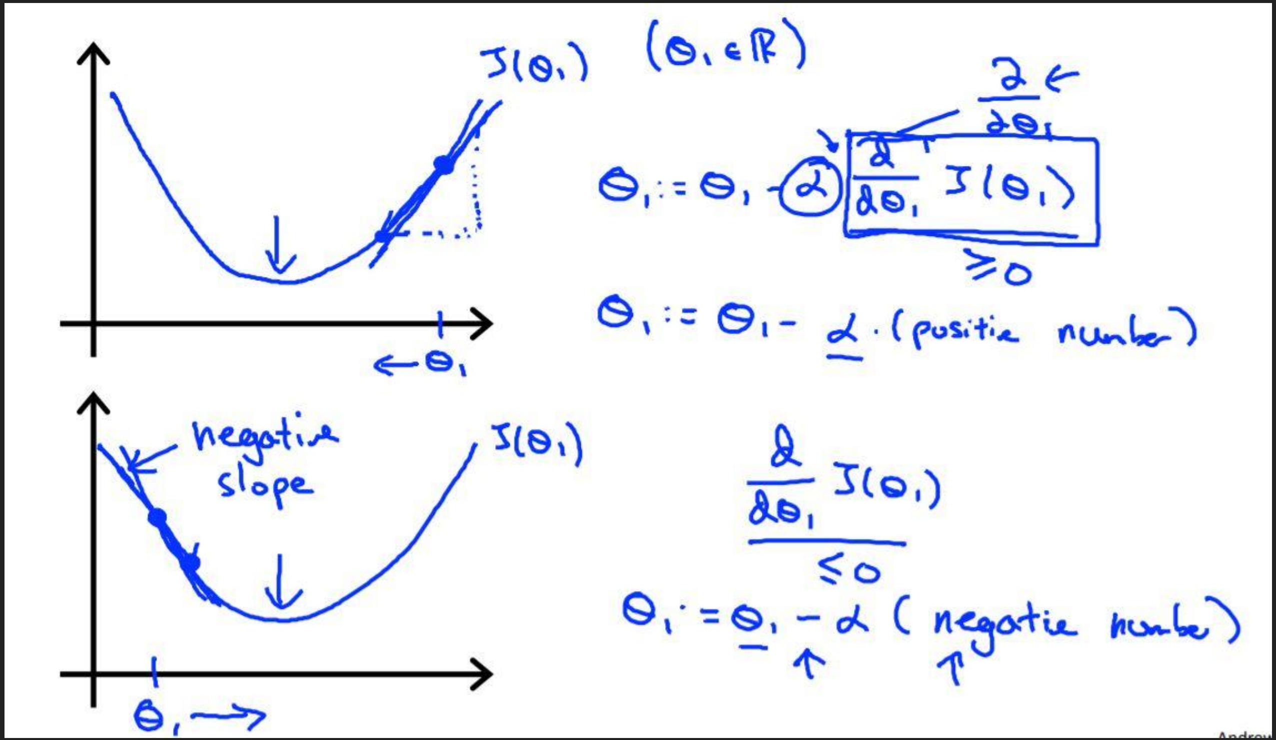
It is the most fundamental optimizer, which created the base for most of the machine learning problems.

Equation:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

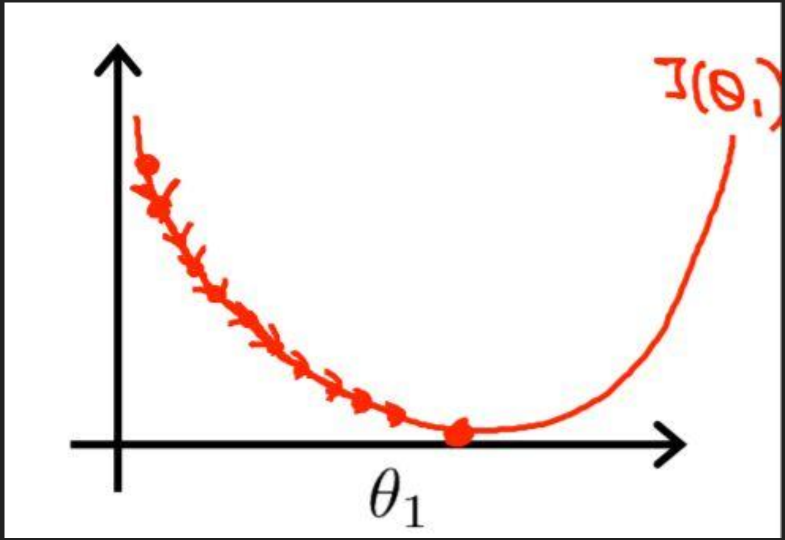
In simple terms, it is simultaneously subtracting,partial derivative from each parameter. **Alpha** is the learning rate(positive quantity, which needs to be tuned)

Gradient Descent intuition

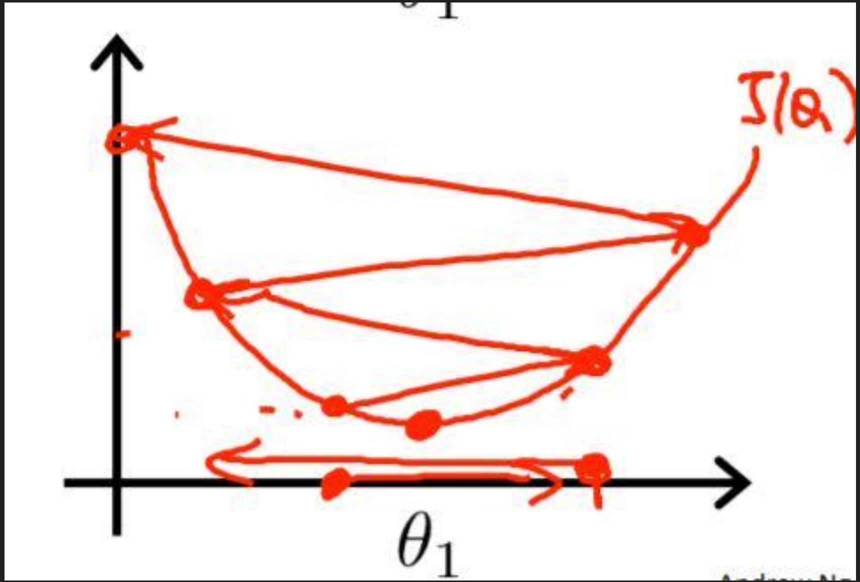


Learning Rate

The variable alpha, in gradient descent is known as Learning Rate. It monitors the rate of convergence.



If alpha is too small



If alpha is too large

Putting all things together

Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\qquad (\text{for } j = 1 \text{ and } j = 0) \\
&\}
\end{aligned}
$$

Linear Regression Model

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

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

Gradient descent algorithm

$$\frac{\partial}{\partial\theta_0} J(\theta_0, \theta_1)$$

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)}$$

$$\theta_1 := \theta_1 - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}}$$

}

$$\frac{\partial}{\partial\theta_1} J(\theta_0, \theta_1)$$

update θ_0 and θ_1 simultaneously

Hyper Parameters

Parameters, which need to be tuned based on the performance of the algorithm are called hyper parameters.

Examples: learning rate, number of iterations

Multivariable Linear Regression

Size in $feet^2$ (X)	Number of Bedrooms	Number of Floors	Age of Home	Price is Lakhs(y)
2104	5	1	45	46
1416	3	2	40	25
1543	3	2	30	32
850	2	1	36	18
...	...	...	...	...

Equations

Hypothesis:

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

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

E.g. $\underline{h_\theta(x)} = \underline{80} + \underline{0.1} x_1 + \underline{0.01} x_2 + 3 x_3 - 2 x_4$

age

$x_0 = 1$

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\cancel{\theta_0, \theta_1, \ldots, \theta_n}$ Θ n+1 - dimensional vector

Cost function:

$$\cancel{J(\theta_0, \theta_1, \ldots, \theta_n)} = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$J(\Theta)$

Gradient descent:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} \cancel{J(\theta_0, \ldots, \theta_n)} \; J(\Theta)}$$

$\}$

(simultaneously update for every $j = 0, \ldots, n$)

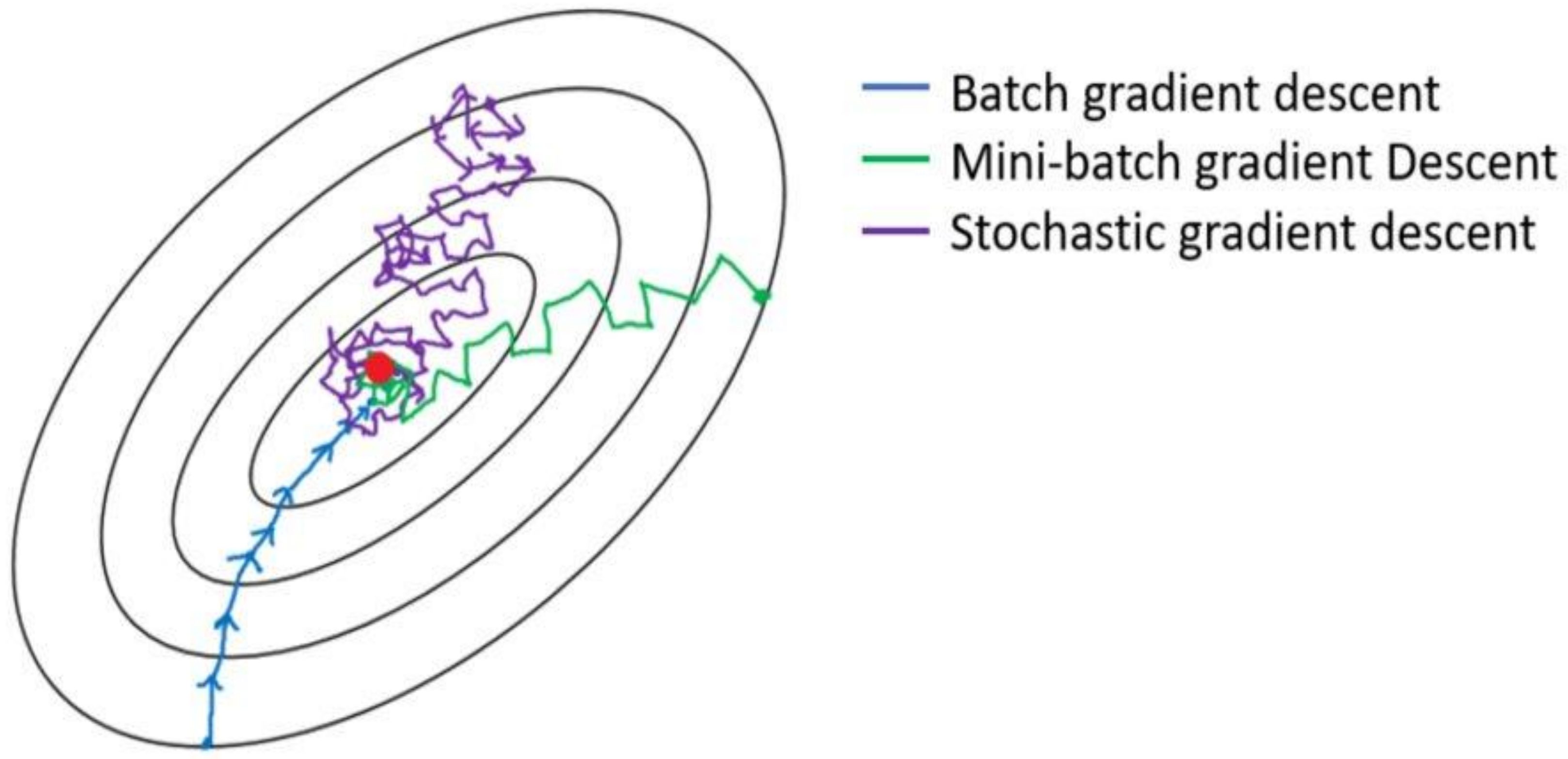
Linear Regression

1. If we have N features, there are N+1 parameters to be tuned
2. For Each parameter gradient/slope is calculated assuming other parameters are constant
3. These parameters are updated using gradient descent optimizer again and again(#iterations)
4. The trained parameters are saved and are used for predicting for future samples

Optimizers

1. Batch Gradient Descent
2. Stochastic Gradient Descent
3. Mini-Batch Gradient Descent

Traditional gradient descent optimizer is also known as “Batch Gradient Descent”. Since, it updates the the gradient by going through

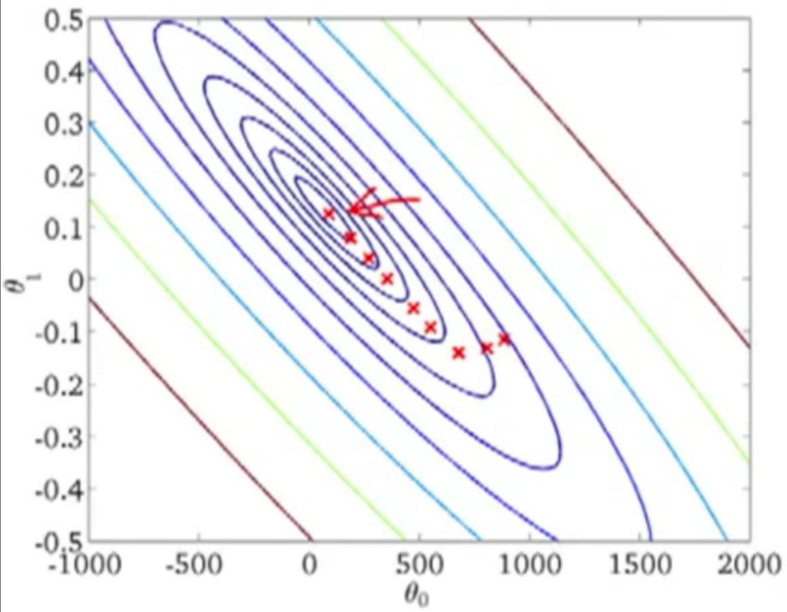

Batch gradient descent
Mini-batch gradient Descent
Stochastic gradient descent

Stochastic Gradient Descent

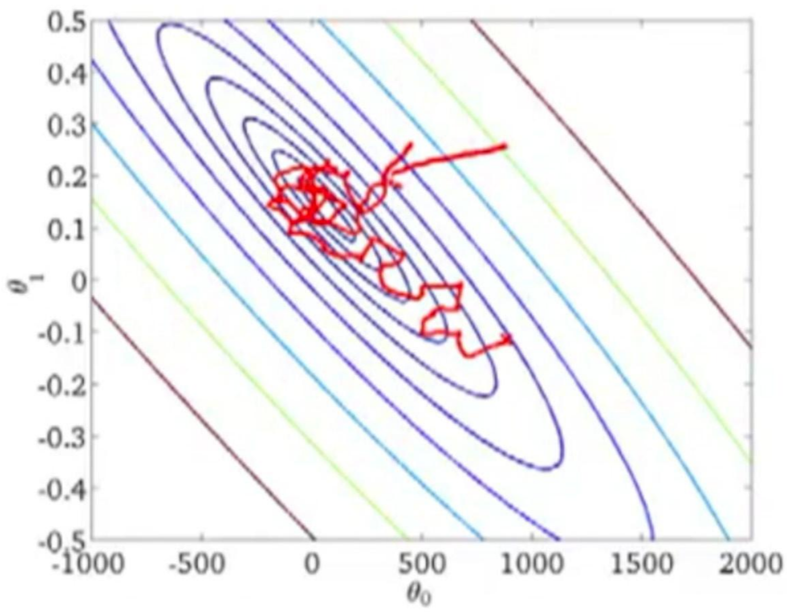
In Stochastic Gradient Descent, we calculate the gradient for each sample, and update the parameters.

If the Data has N samples, in one scan of data, we update the parameters N times.

Each step update in stochastic gradient descent may not point towards global minim. But, cumulatively, it reaches close to global minimum.



Batch Gradient Descent



Stochastic Gradient Descent

Mini-Batch Gradient descent

Mini-Batch gradient descent is a trade-off between batch and stochastic gradient descents. It processes average gradient for batch of samples and updates the parameters.

There is a hyper parameter called, Batch size, which is generally taken in power of 2, to leverage parallel processing in CPU’s and GPU’s

Epoch vs Iteration

Epoch is the no. of times the data is scanned.

Iteration is the no. of times the parameters are updated.

Batch Gradient Descent: 1 epoch = 1 iteration

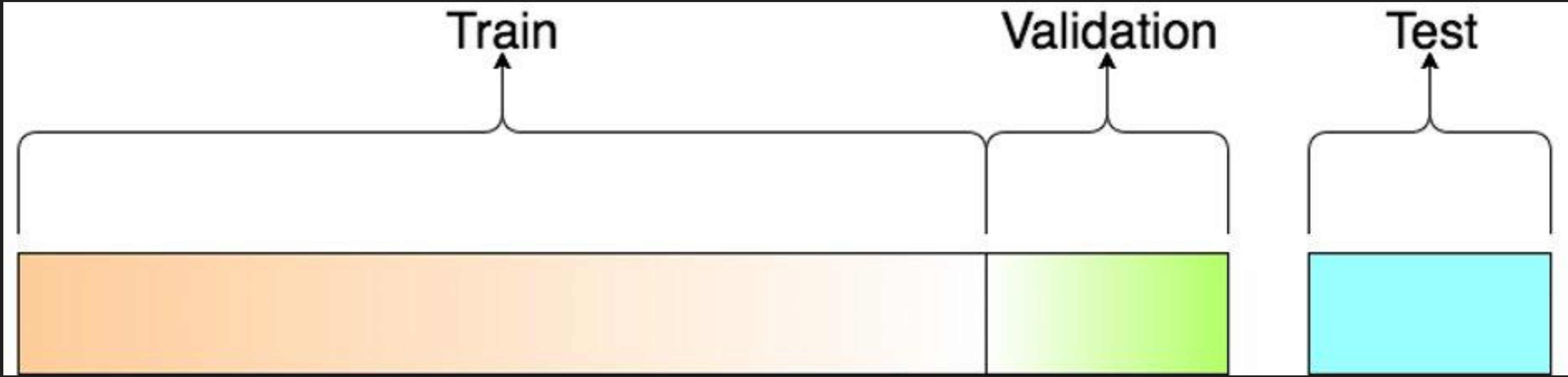
Stochastic Gradient Descent: 1 epoch = N iterations

Stochastic Batch Gradient Descent: 1 epoch = p iterations(p = N/(batch_size))

Objectives

1. Scikit learn, multivariate linear regression
2. Features-scaling,Preprocessing
3. Logistic Regression
4. Multi-class classification
5. Metrics for Evaluating models

Data set split



Feature Scaling

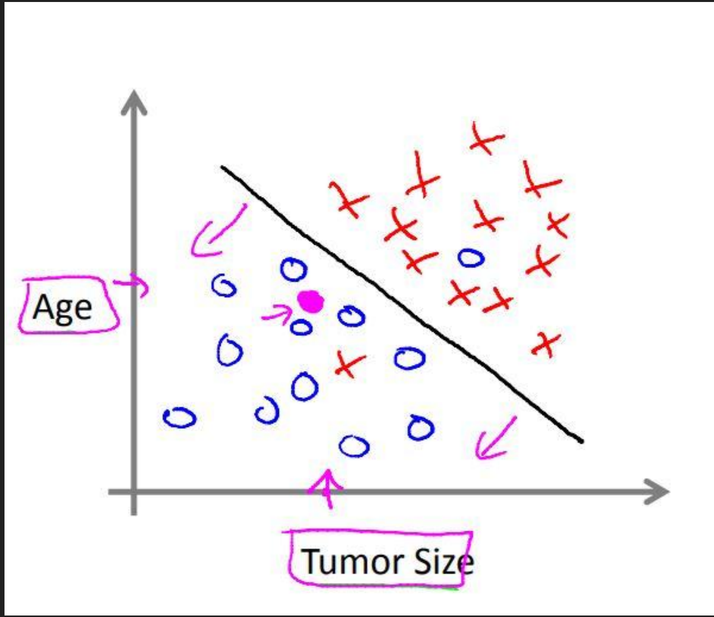
Feature scaling is used to remove the dominance of features with larger values

$$X_{new} = \frac{X_i - X_{mean}}{\text{Standard Deviation}}$$

$$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$

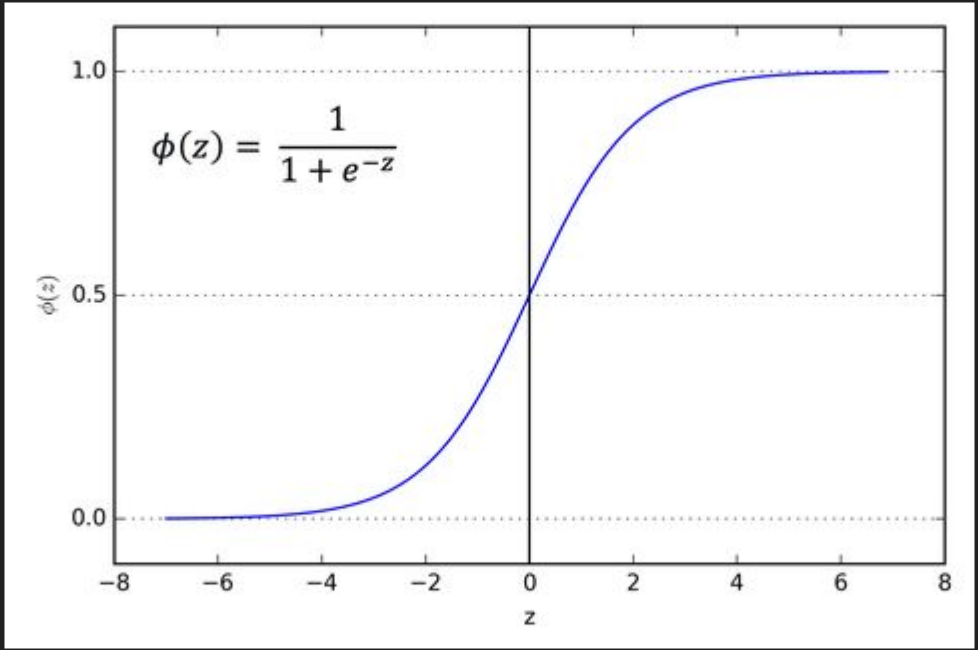
Logistic Regression

Logistic Regression is the fundamental Binary Classification algorithm.



User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1
15617482	Male	45	26000	1
15704583	Male	46	28000	1

Sigmoid activation Function



Hypothesis Function in Logistic Regression

$$h_\theta(x) = \sigma(\theta^T x)$$

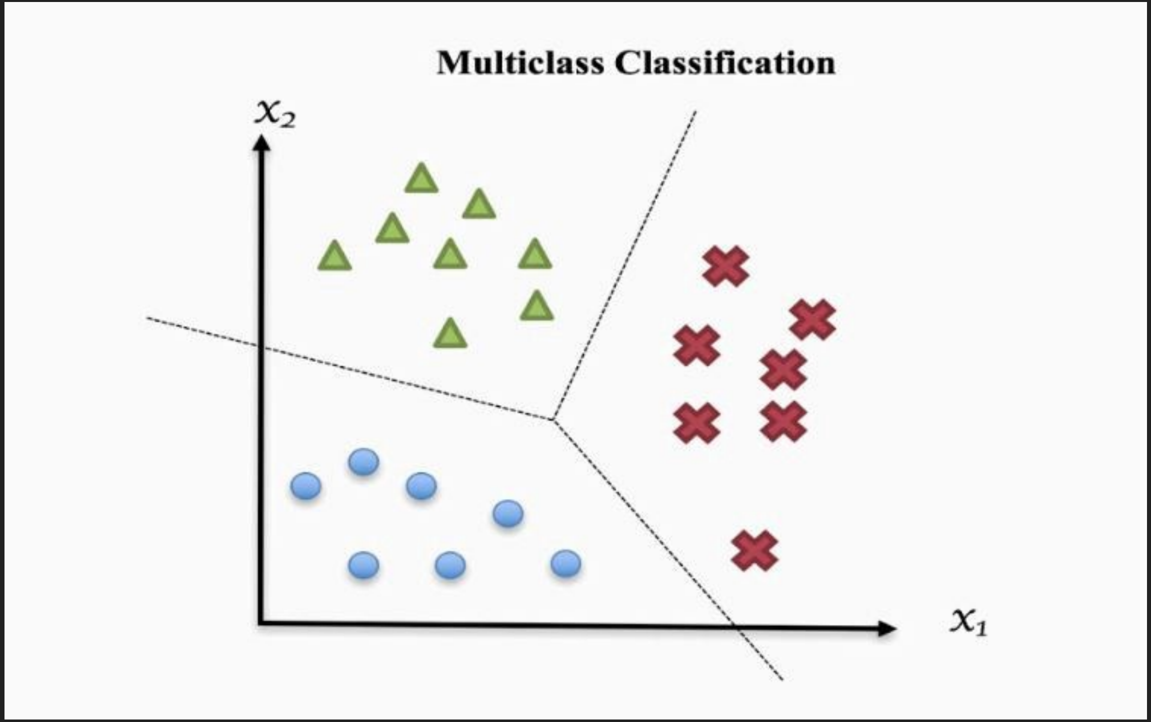
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))]$$

Calculating Gradients is complex!!

Multi-Class classification



One Hot Encoding of Labels

Human-Readable

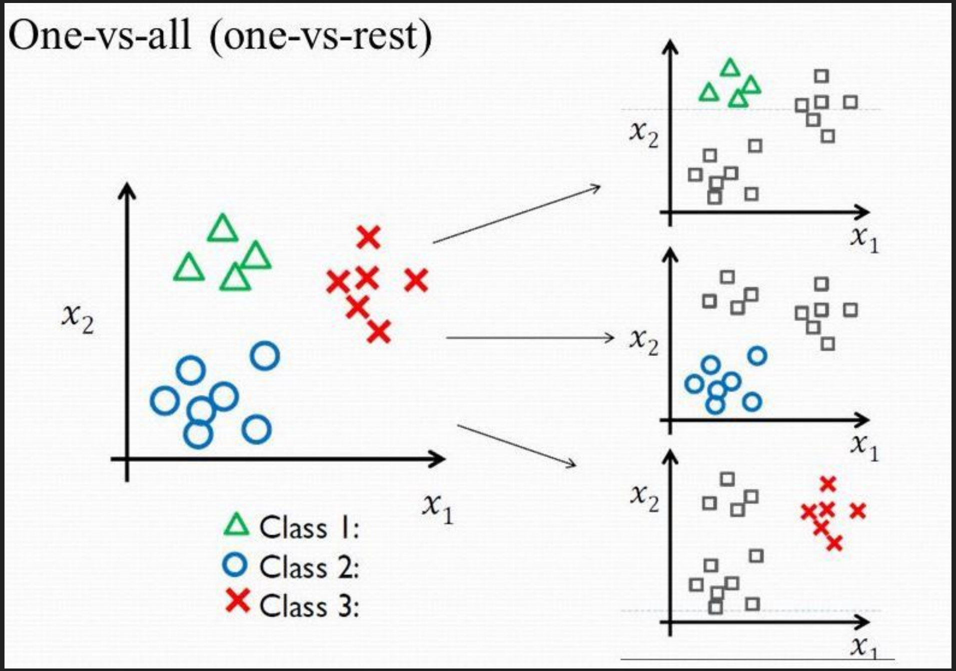
Pet
Cat
Dog
Turtle
Fish
Cat

Machine-Readable

Cat	Dog	Turtle	Fish
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

One Vs All classification

It Means Applying logistic Regression for each class seperatly

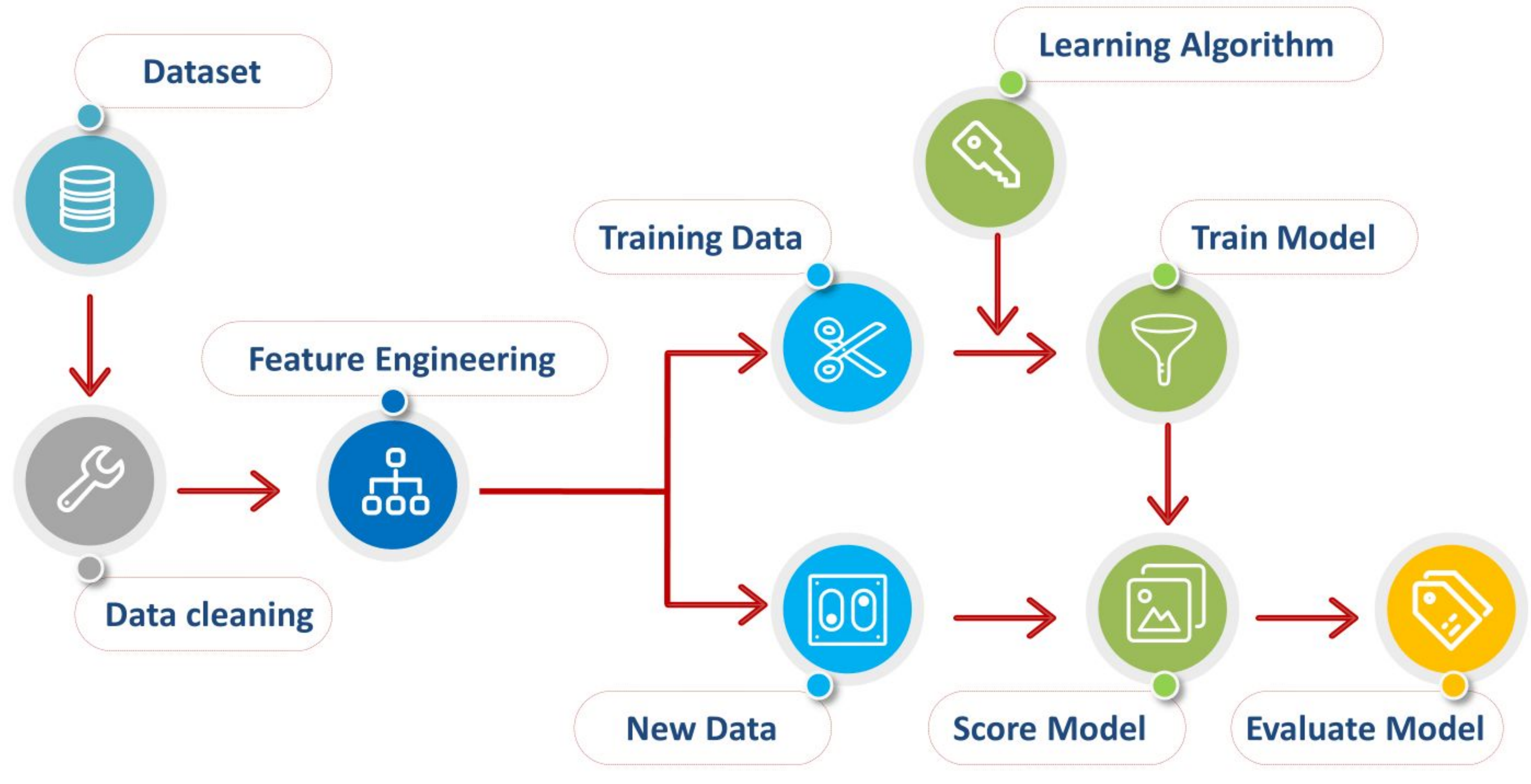


Soft-Max function

Soft-Max returns probabilities such that, the sum of all probabilities adds to 1

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Metrics and Evaluation of Machine Learning Models


Dataset
Data cleaning
Feature Engineering
Training Data
New Data
Learning Algorithm
Train Model
Score Model
Evaluate Model

Continuous Label

Mean Squared error is a standard error for continuous variables. We calculate, test mse for different models and pick the best one.

MAE can also be used in place of MSE

Categorical Label

There are metrics like Precision, Recall, Accuracy, F1 Score are used to evaluate the performance of the models.

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

$$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}$$

$$F_1 \text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Tp -> True Positives
Fp-> False Positives

Confusion Matrix

Confusion matrix also gives give insights on the performance of the classification models.

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110