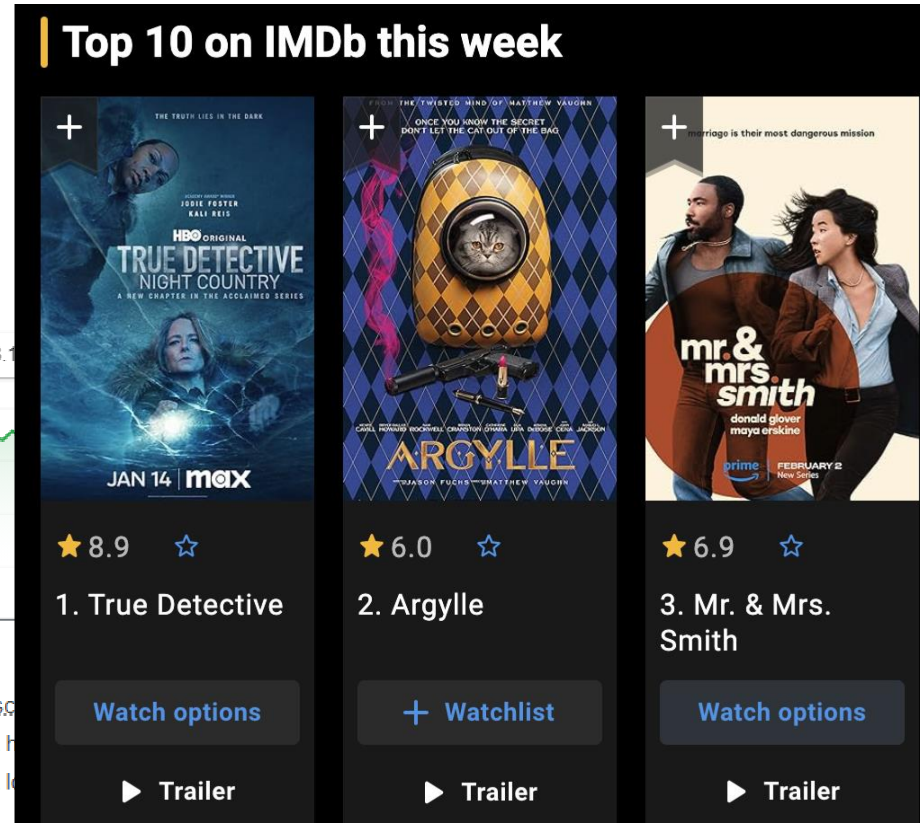
11.09.2024

BITS F464: Machine Learning (1st Sem 2024-25)

REGRESSION MODELS

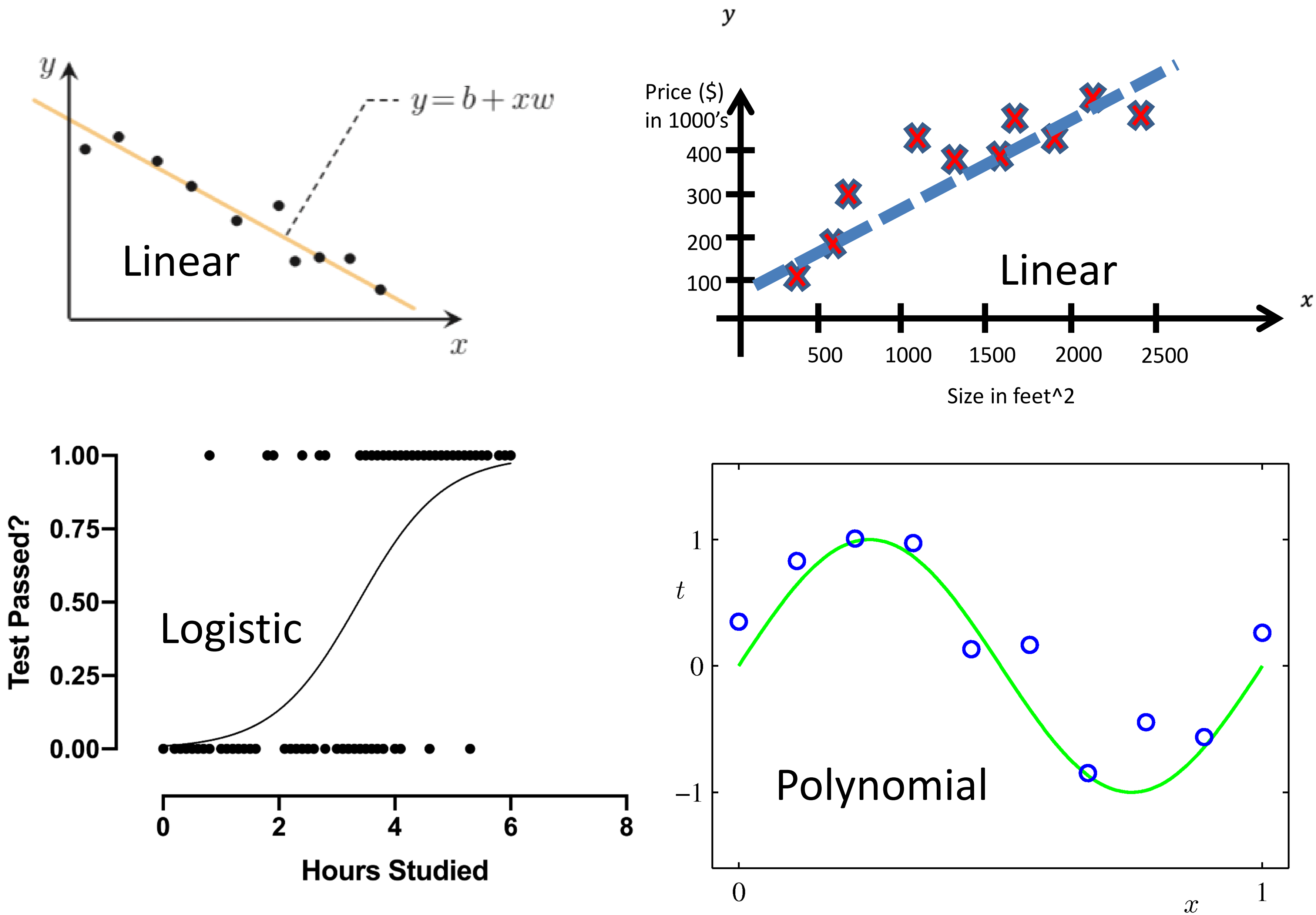
Chittaranjan Hota, Sr. Professor
Dept. of Computer Sc. and Information Systems
hota@hyderabad.bits-pilani.ac.in

What Type of Problems can you solve?



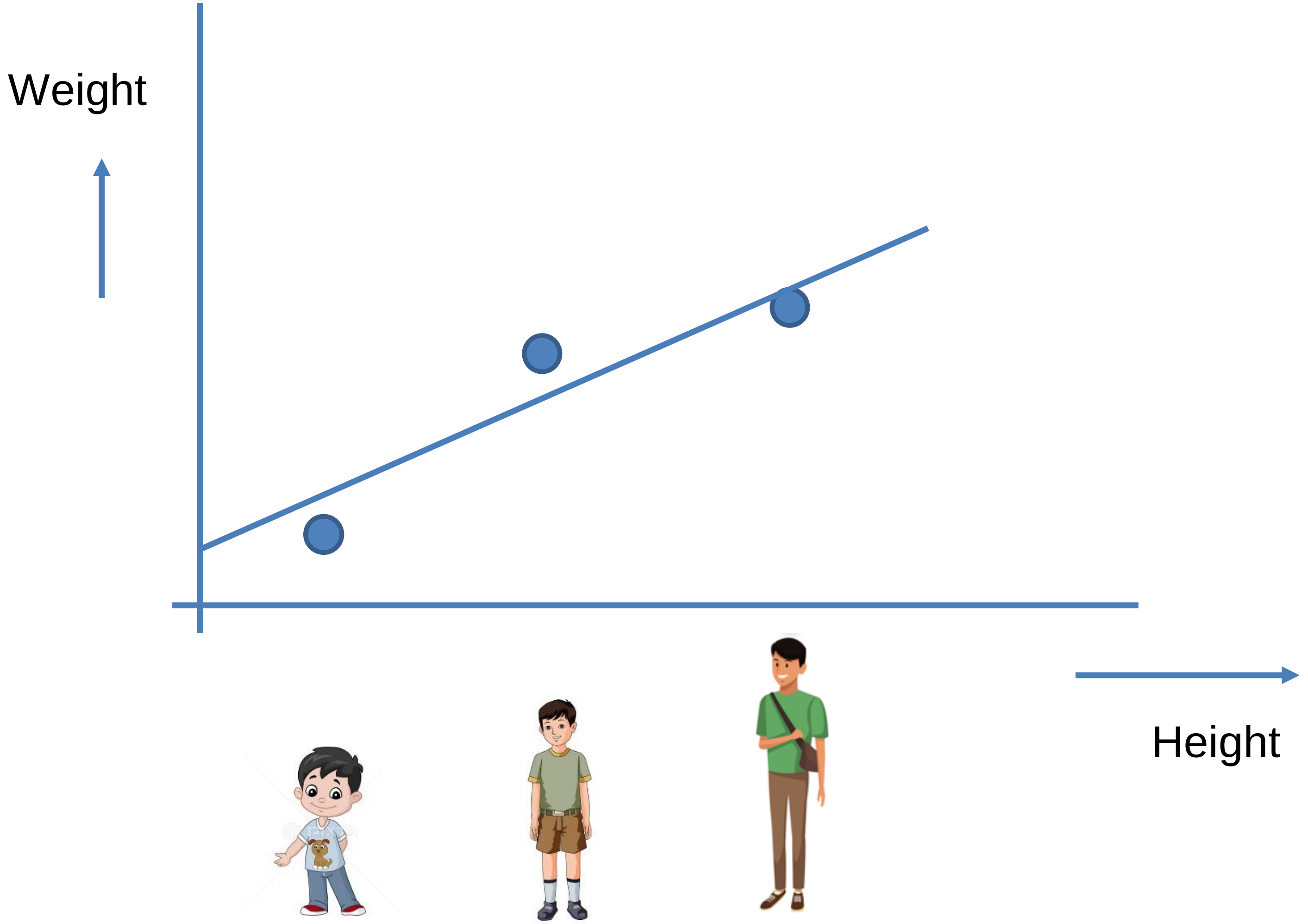
Source: www.macroaxis.com/stocks/

https://www.imdb.com/



Different types of Regression for different purposes. Ridge, Lasso, Bayesian, …

Regression with Scalar Input(Univariate)

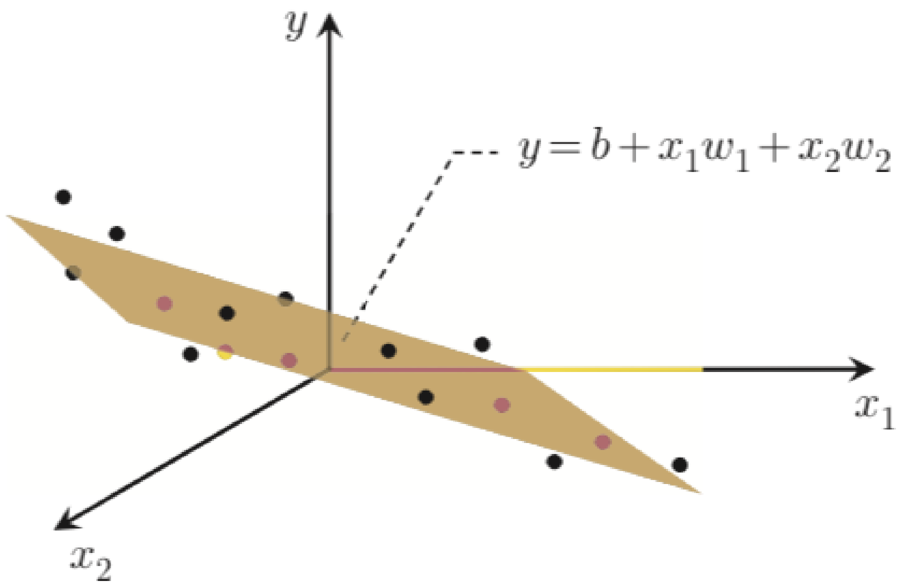


Simple Linear Regression

With Vector inputs (more covariates)

LARGEST ECONOMIES IN THE WOR		
Rank	Country	GDP (in USD Bil
1.	United States of America	26,954
2.	China	17,786
3.	Germany	4,430
4.	Japan	4,231
5.	India	3,730
6.	United Kingdom (UK)	3,332
7.	France	3,052
8.	Italy	2,190
9.	Brazil	2,132
10.	Canada	2,122

https://currentaffairs.adda247.com/



- Unemployment rate, education level, population count, land area, income level, investment rate, life expectancy, … (Multiple Linear Regression: Multi-variate)

Another Example of Multi-variate Regression

Sales = b + w_1 weather + w_2 money + w_3 day

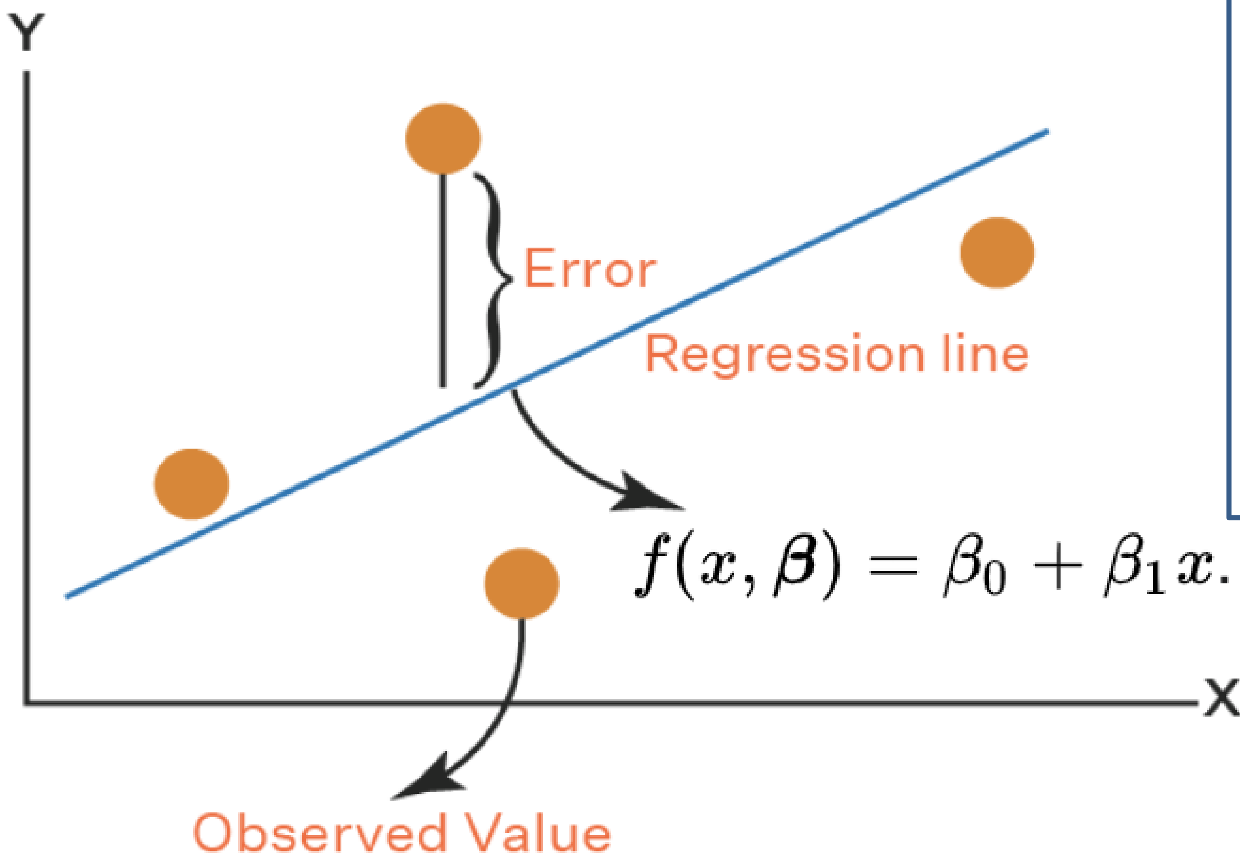
Regression:

Process of finding out relationship between a dependent variable (outcome/ response/ label) and one or more independent variables (predictors/ covariates/ explanatory variables/ features)

Independent variables (X): weather (rainy, sunny, cloudy), amount in hand, day type (working, holiday), Dependent variable: Y (Sales)

How the dependent variable (Y) will react to each variable X taken independently?

Best Fitting a Line: Least Squares Method



$$f(x, \boldsymbol{\beta}) = \beta_0 + \beta_1 x.$$

If $\beta_1 > 0$

How are X and Y related?

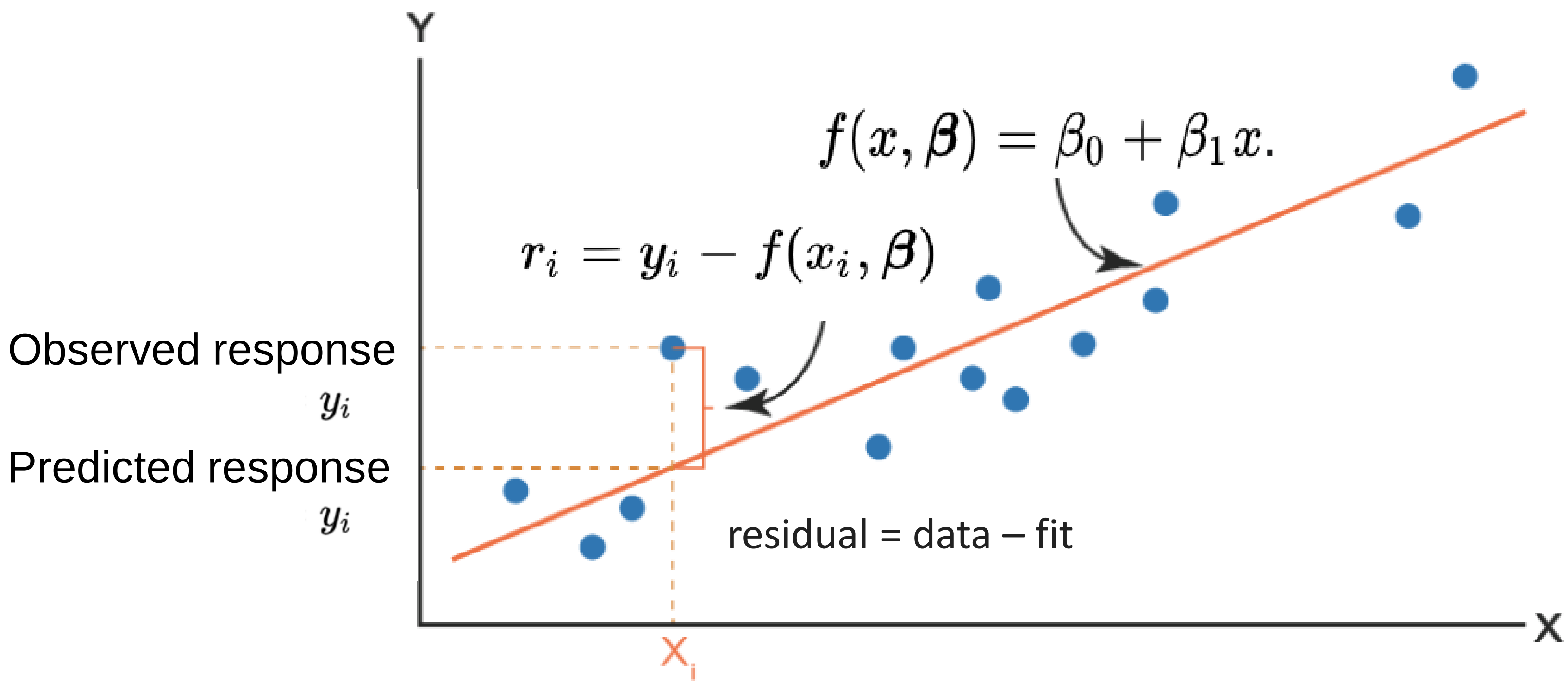
If $\beta_1 < 0$?

If $\beta_1 == 0$?

The target function: $f(x, \boldsymbol{\beta})$, where m adjustable parameters are held in vector $\boldsymbol{\beta}$.

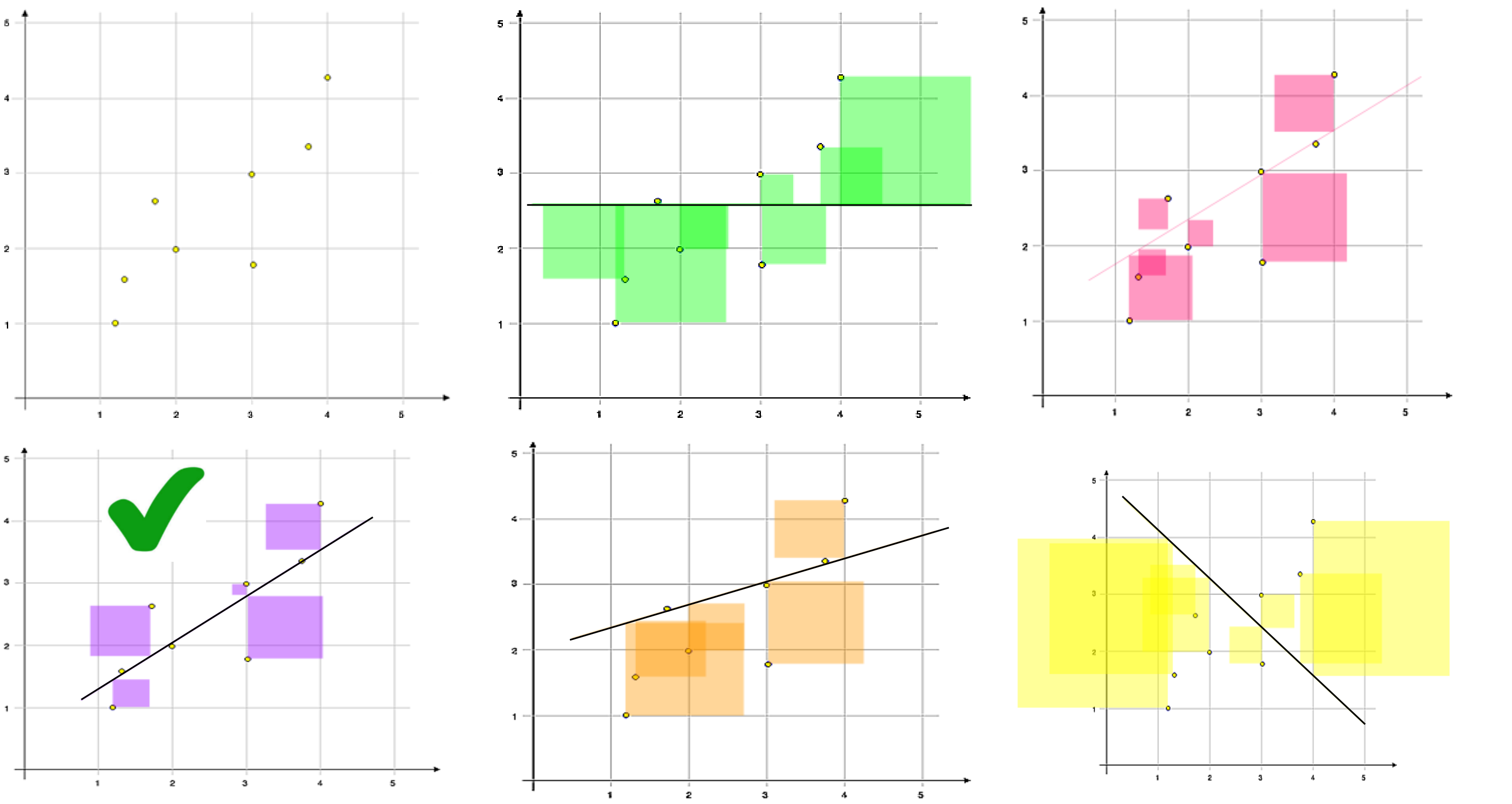
Simple Linear Regression

Best Fitting a Line: Least Squares Method



Find out the optimal parameter values by minimizing the sum of squared residuals $S = \sum_{i=1}^{n} r_i^2$

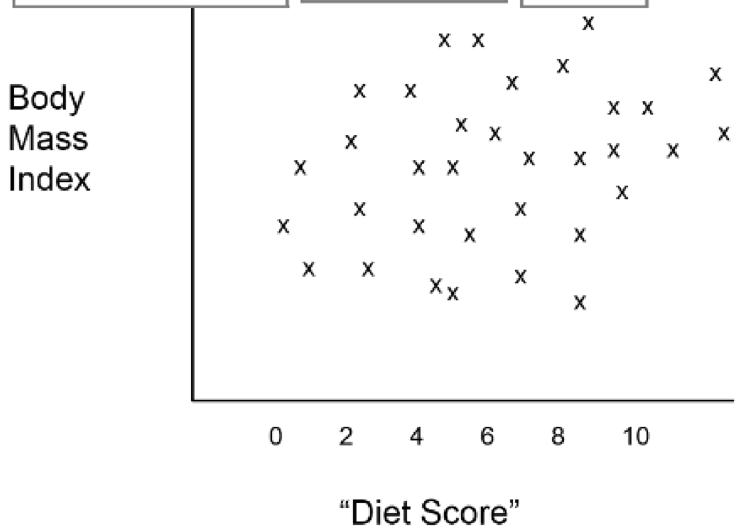
Can you choose the best-fit line?



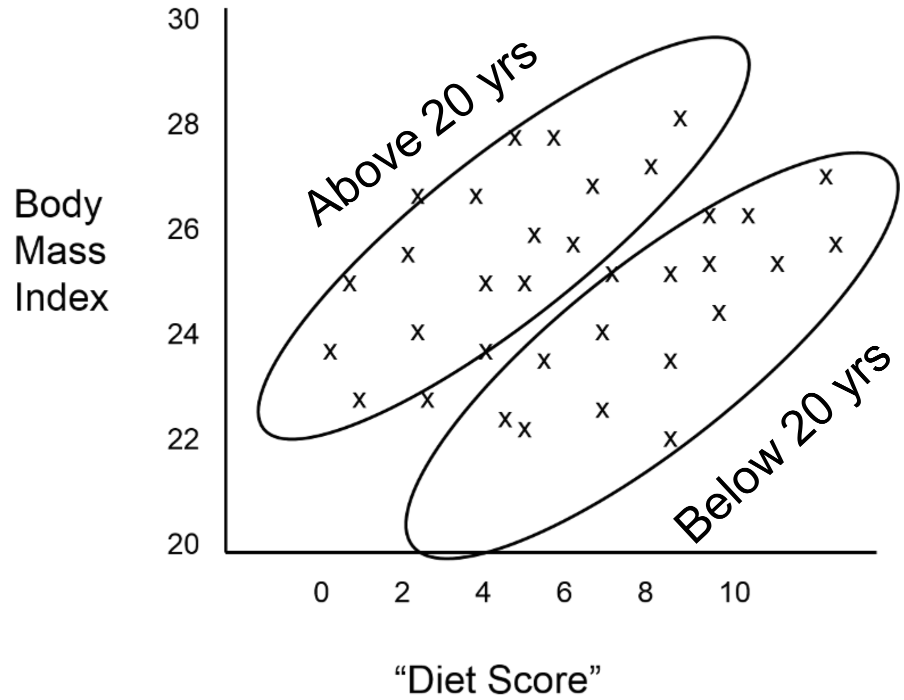
Hypothetically: Say, weight = 2 + 1.5 height

Multiple Linear Regression Analysis

Diet Score	Age>20	BMI
4	1	27
7	1	29
6	0	23
2	0	20
3	1	21
...	...	...



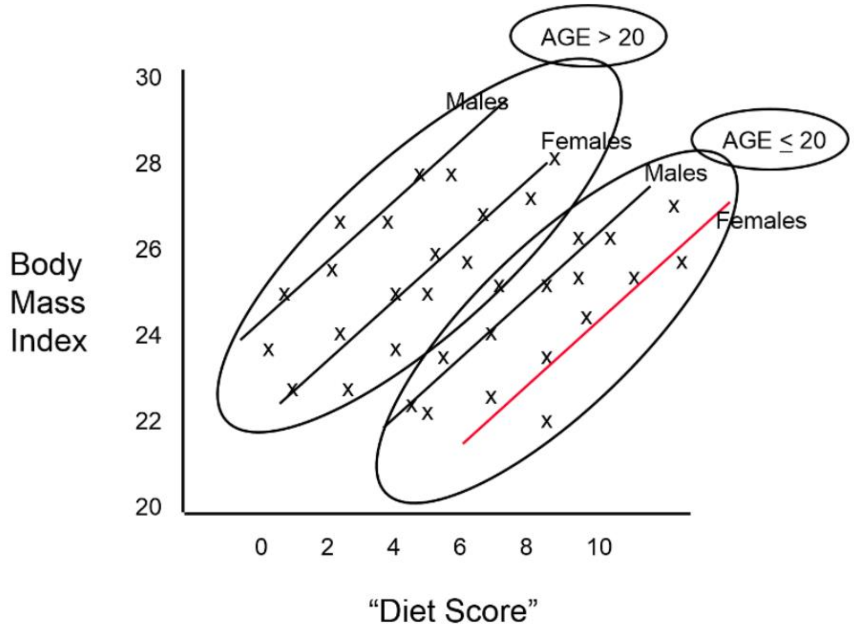
(hardly any association between the two)



(People are clustered based on age)

Img. Source: https://sphweb.bumc.bu.edu/

Continued...



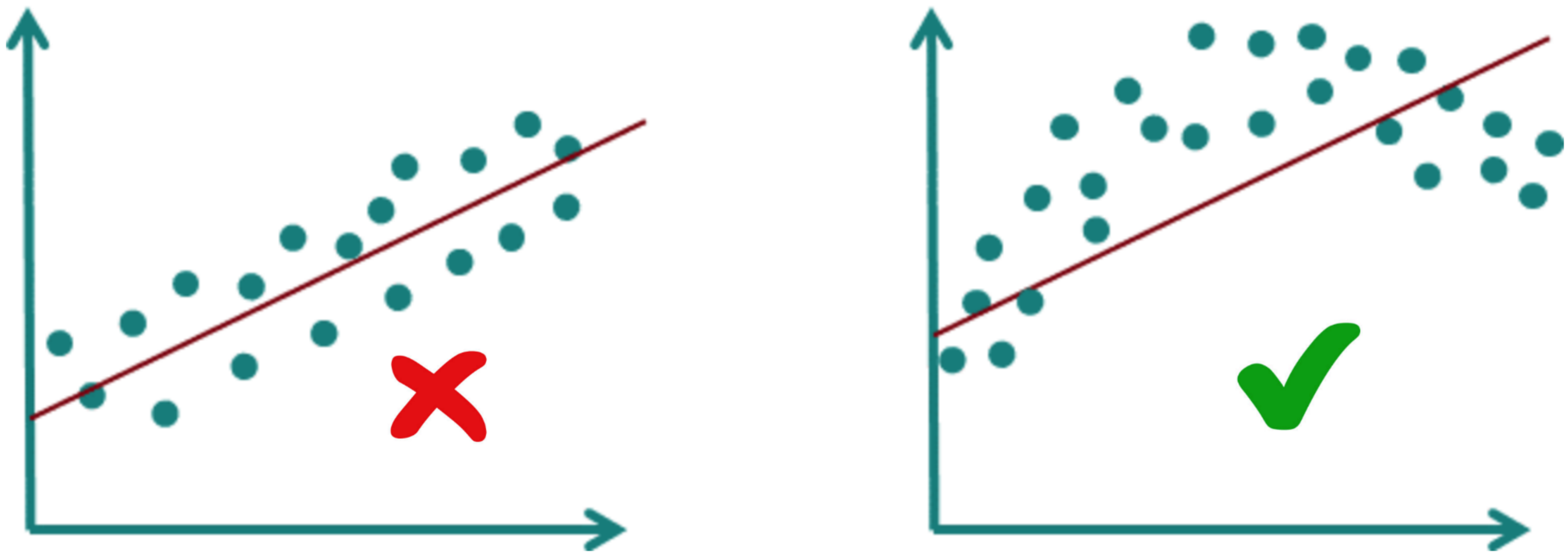
Diet Score	Male	Age>20	BMI
4	0	1	27
7	1	1	29
6	1	0	23
2	0	0	20
3	0	1	21
...	...	...	...

BMI = 18 + 1.5 (diet score) + 1.6 (male) + 4.2 (age > 20)

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

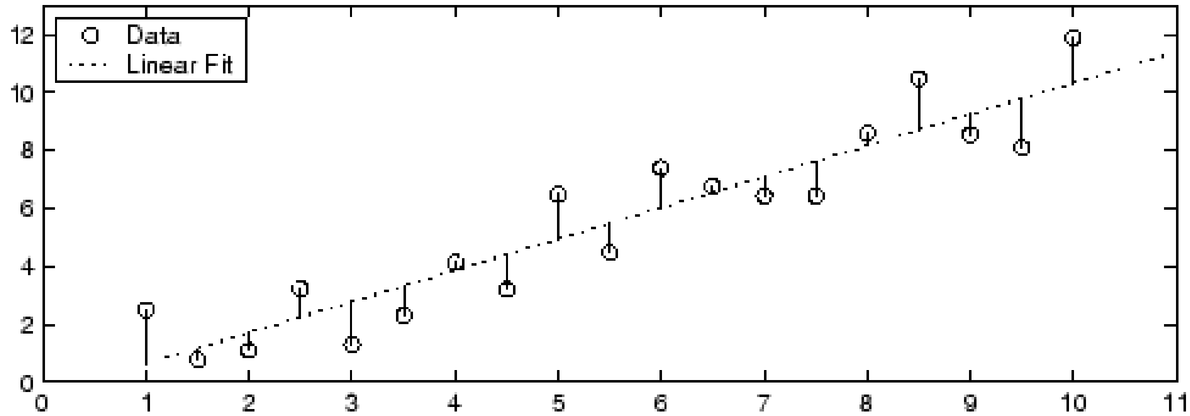
Img. Source: https://sphweb.bumc.bu.edu/

Non-linear relationships

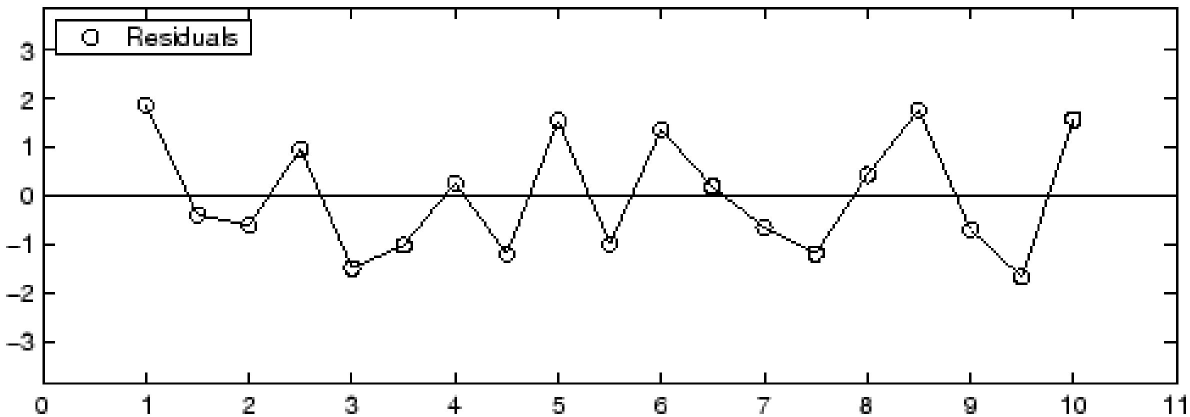


Examples: House price based on Floor area, Electricity consumption based on no. of household members and appliances being used.

Analyzing Residuals

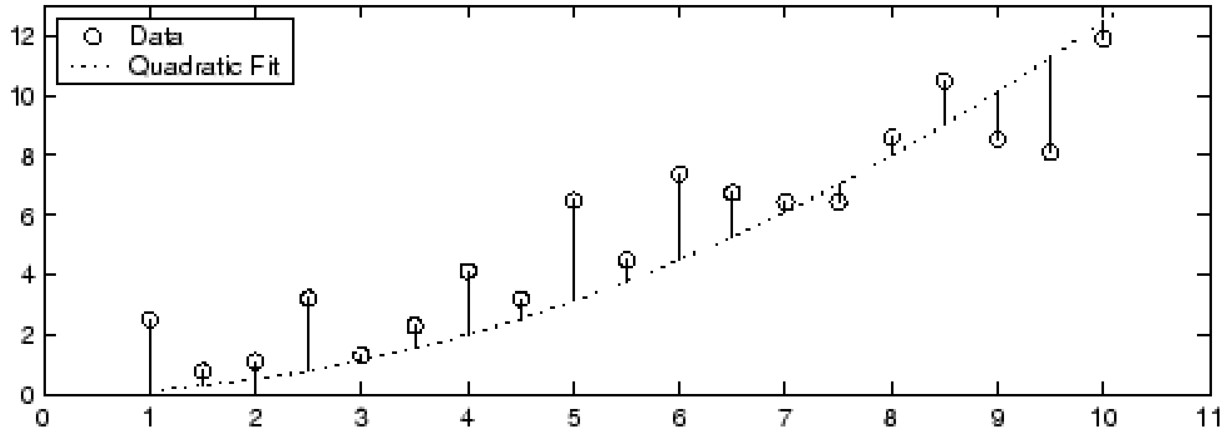


Model describes data well or poor?

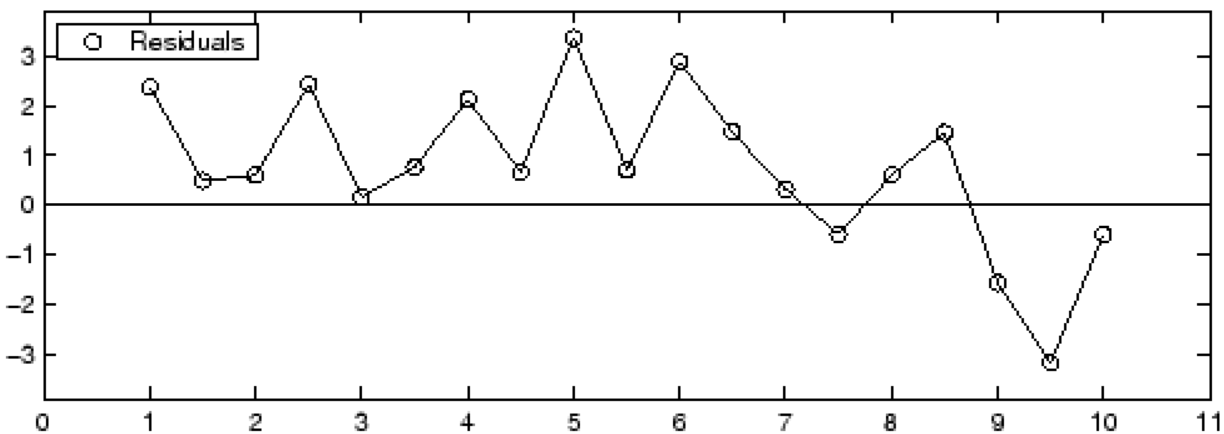


Randomly scattered around zero

Continued…



Model includes a Second-degree polynomial (quadratic term)



Systematically positive for much of the data.

Good or bad fit?

Non-linear relations using Linear models?

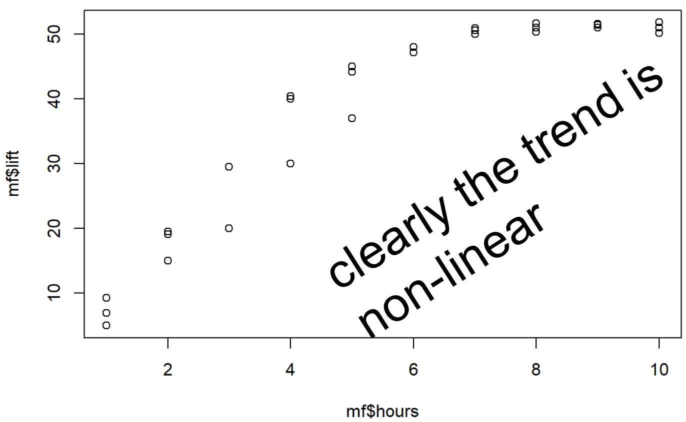
- Feature Engineering: Engineer new features by transforming the existing ones to capture non-linear relationships, e.g, you can include polynomial features (e.g., quadratic, cubic).
- Using Basis Functions: Instead of using the original features, you can use basis functions, which are transformations of the original features, e.g Polynomial basis functions, Gaussian radial basis functions, or Sigmoidal basis functions.
- Regularization: Ridge regression (L2 regularization) or Lasso regression (L1 regularization) to penalize large coefficients.
- Non-linear Regression Models: If the relationship is highly non-linear, use Polynomial, Logistic, exponential, Power-law, Gaussian, Logarithmic regression etc., Decision trees, Random forests, SVMs with non-linear kernels, or Neural networks.

We will see some of these…

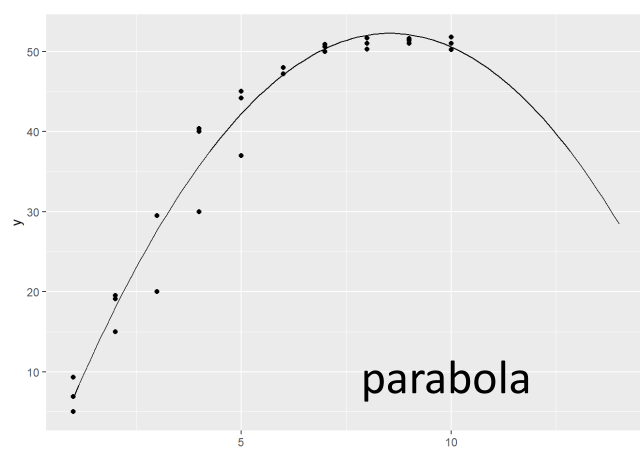
An Example

##	name	lift	hours
## 1	Person 01	5.0	1
## 2	Person 02	15.0	2
## 3	Person 03	20.0	3
## 4	Person 04	30.0	4
## 5	Person 05	37.0	5
## 6	Person 06	48.0	6
## 7	Person 07	50.0	7
## 8	Person 08	51.0	8
## 9	Person 09	51.0	9
## 10	Person 10	51.0	10
## 11	Person 11	6.9	1
## 12	Person 12	19.5	2
## 13	Person 13	29.5	3
## 14	Person 14	40.4	4
## 15	Person 15	45.0	5
## 16	Person 16	48.0	6
## 17	Person 17	50.9	7
## 18	Person 18	50.3	8
## 19	Person 19	51.4	9
## 20	Person 20	51.8	10
## 21	Person 21	9.3	1
## 22	Person 22	19.1	2
## 23	Person 23	29.5	3
## 24	Person 24	40.0	4
## 25	Person 25	44.2	5
## 26	Person 26	47.2	6
## 27	Person 27	50.6	7
## 28	Person 28	51.7	8
## 29	Person 29	51.6	9
## 30	Person 30	50.2	10

- *lift* is the dependent variable, and the independent variable is the '*hours*', i.e the time spent in weight lifting.



- We add a quadratic term as an independent variable in the model. $y = x^2$

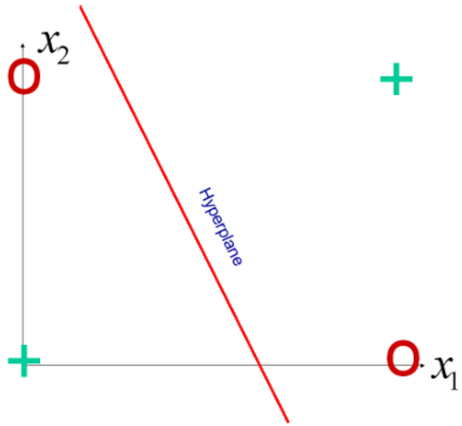


$$\hat{lift} = -6.13 + 13.67 * hours - 0.8 * hours^2$$

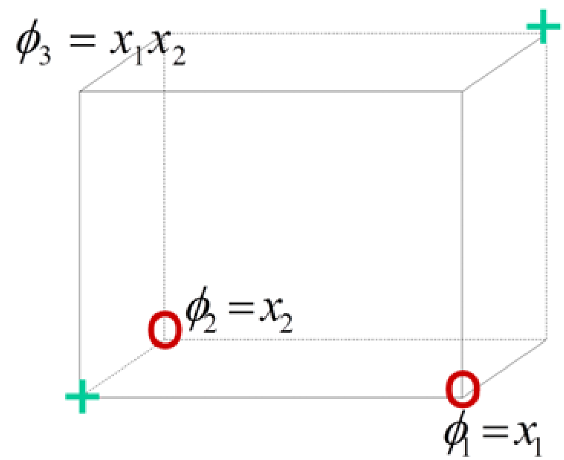
##	name	lift	hours	hoursSq
## 1	Person 01	5.0	1	1
## 2	Person 02	15.0	2	4
## 3	Person 03	20.0	3	9
## 4	Person 04	30.0	4	16
## 5	Person 05	37.0	5	25
## 6	Person 06	48.0	6	36
## 7	Person 07	50.0	7	49
## 8	Person 08	51.0	8	64
## 9	Person 09	51.0	9	81
## 10	Person 10	51.0	10	100
## 11	Person 11	6.9	1	1
## 12	Person 12	19.5	2	4
## 13	Person 13	29.5	3	9
## 14	Person 14	40.4	4	16
## 15	Person 15	45.0	5	25
## 16	Person 16	48.0	6	36
## 17	Person 17	50.9	7	49
## 18	Person 18	50.3	8	64
## 19	Person 19	51.4	9	81
## 20	Person 20	51.8	10	100
## 21	Person 21	9.3	1	1
## 22	Person 22	19.1	2	4
## 23	Person 23	29.5	3	9
## 24	Person 24	40.0	4	16
## 25	Person 25	44.2	5	25
## 26	Person 26	47.2	6	36
## 27	Person 27	50.6	7	49
## 28	Person 28	51.7	8	64
## 29	Person 29	51.6	9	81
## 30	Person 30	50.2	10	100

Source: https://bookdown.org/

Basis Functions: Why are they needed?



Linear or non-linear?



Let us add a basis function x_1x_2 into the input (this term couples two terms non-linearly)

With the third input $z = x_1x_2$ the XOR becomes linearly separable.

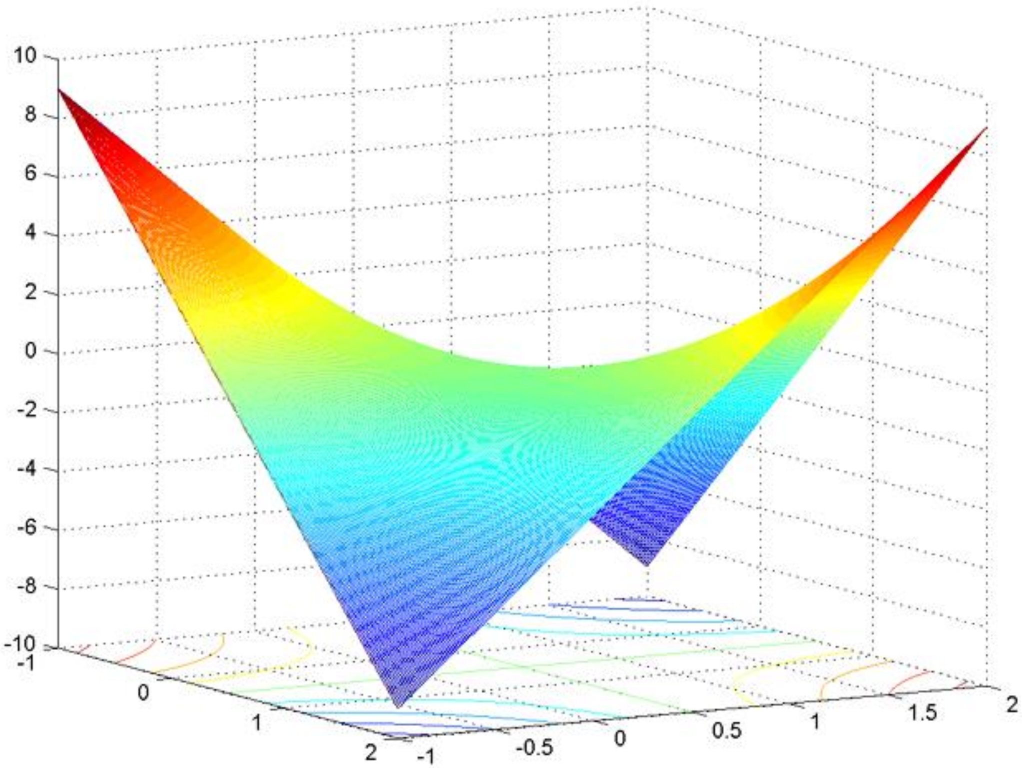
$$f(\mathbf{x}) = 1 - 2x_1 - 2x_2 + 4x_1x_2 = \phi_1(x) - 2\phi_2(x) - 2\phi_3(x) + 4\phi_4(x)$$

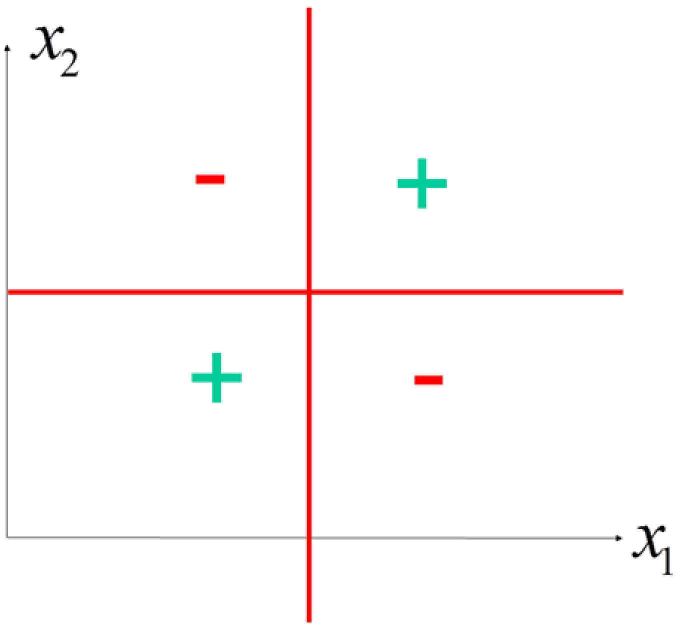
with $\phi_1(x) = 1, \phi_2(x) = x_1, \phi_3(x) = x_2, \phi_4(x) = x_1x_2$

Acknowledgement: Volker Tresp's presentation

Continued…

$$f(\mathbf{x}) = 1 - 2x_1 - 2x_2 + 4x_1x_2$$





Acknowledgement: Volker Tresp's presentation

What are Basis Functions?

Simplest model of Linear Regression: → $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$

Key Property: Linear function of parameters. Also, it is a linear function of its input variables → Imposes serious limitations on the model.

Basis functions come to rescue (called derived features in machine learning) are building blocks for creating more complex functions.

For example, individual powers of x: the basis functions 1, x, x^2, x^3… can be combined together to form a polynomial function.

Basis functions $\phi(x)$ extend this class of models by considering linear combinations of handpicked fixed nonlinear functions of the input variables.

(vector form) $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})$ or $y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$

Non linearity in the data while keeping linearity in parameters.

Where, $\phi(\mathbf{x}) = [\phi_0(x_1), \phi_1(x_2), \ldots, \phi_{M-1}(x_n)]^T$ and $\mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathrm{T}}$

Basis functions for Non-linearity

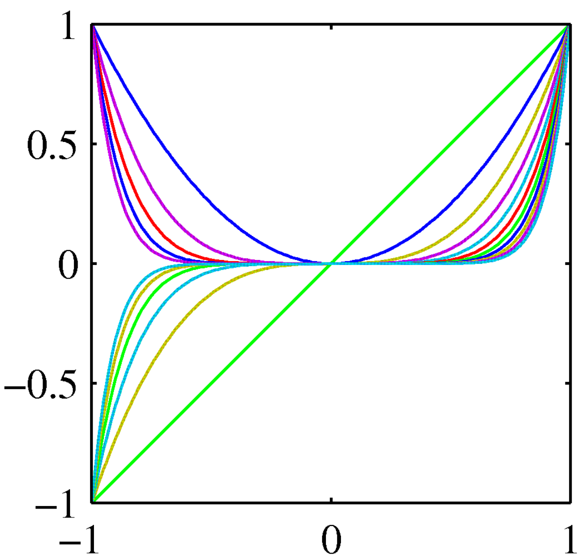
$$\phi_j(x) = x^j$$

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$$

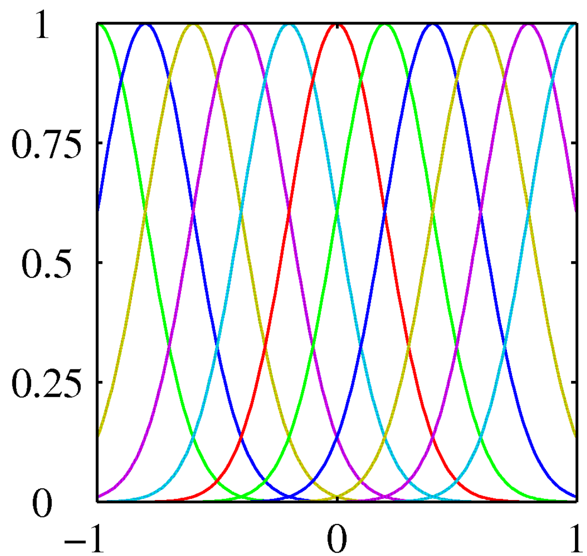
Where,

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$



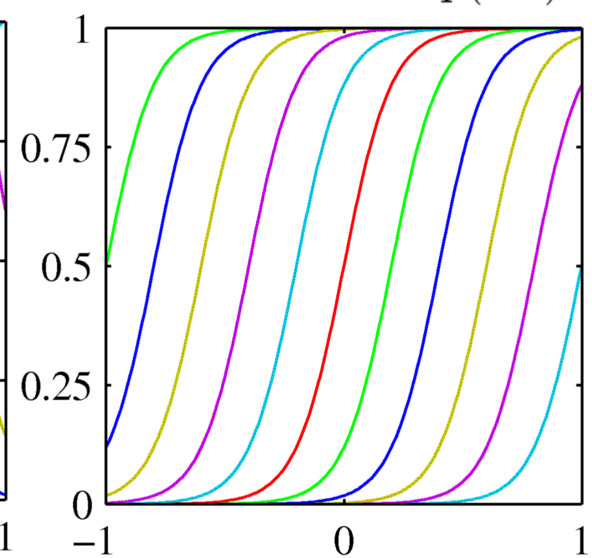
(Polynomial basis function)

Global: a small change in x affects all basis functions



(Gaussian basis function)

Local: a small change in x only affects nearby basis functions.



(Sigmoidal basis function)

Local: a small change in x only affects nearby basis functions.

The Learning Algorithm

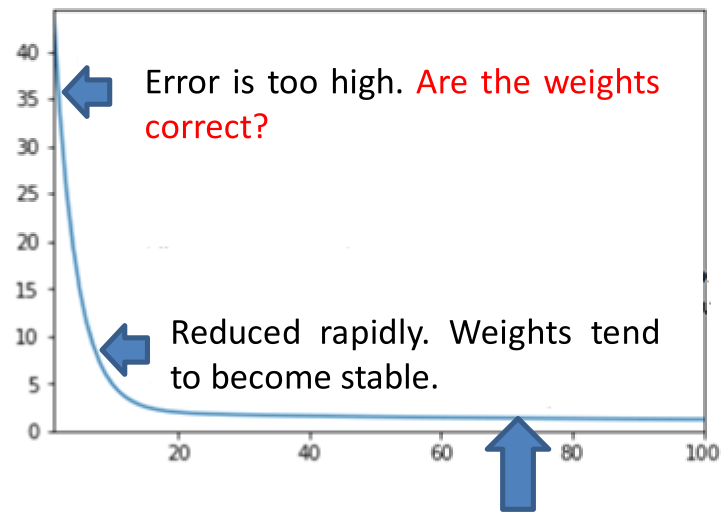
Repeat until the error is minimized

Initial Random Weights

Compute least square error

Compute the gradient to change the weight

Loss is stable, output the model



No more change of the loss/ cost function. Model found best weights.

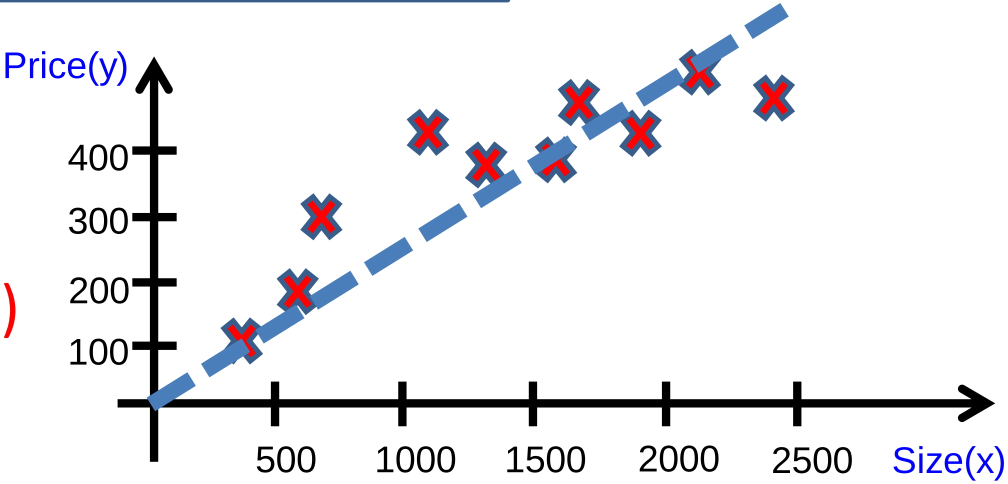
An Example of house price prediction

Size in sq. feet (x)	Price in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Training Set

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

What is the value of θ_0 ?



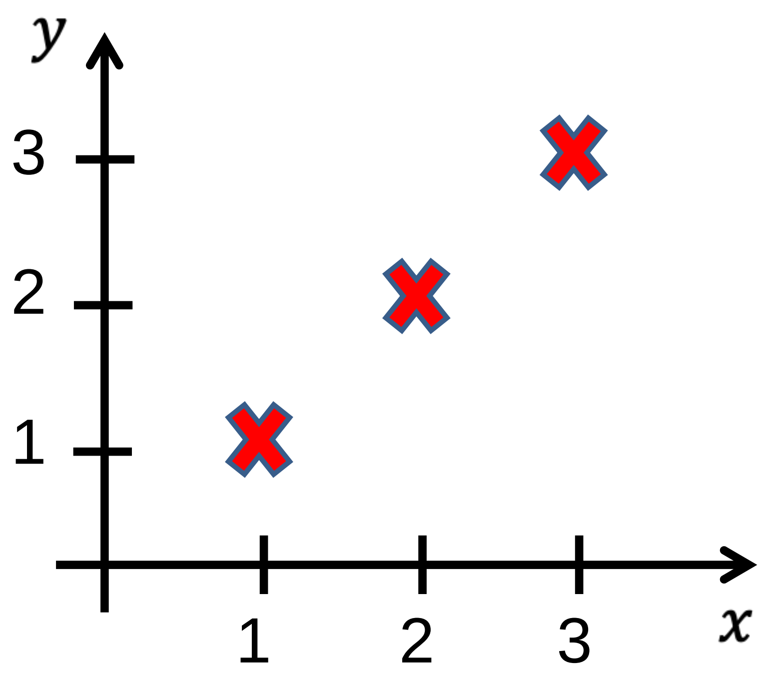
Minimize Cost/ Loss: (MSE)

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

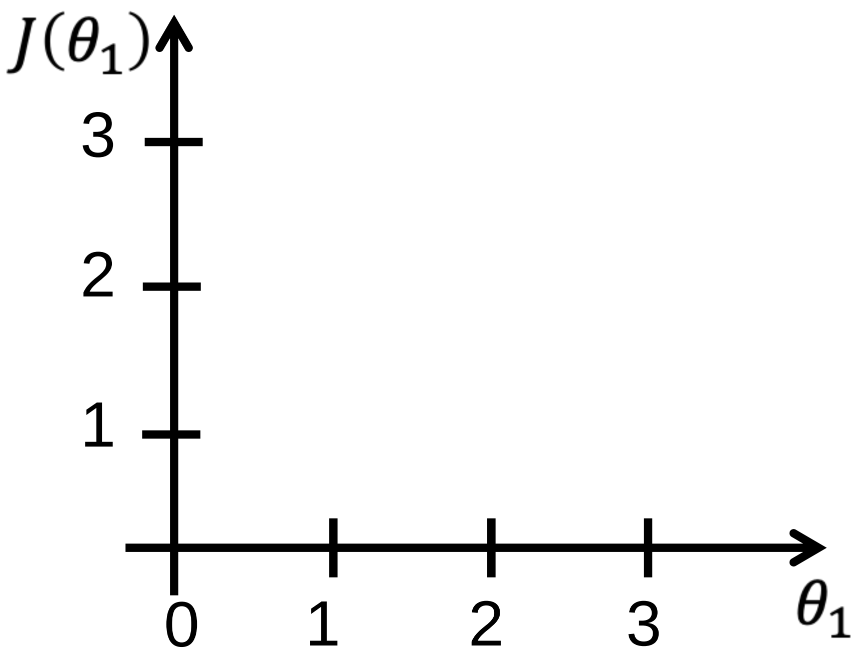
The division by 2 is for convenience and doesn't fundamentally change the result; it simplifies the derivative computation when optimizing models.

Minimizing the Cost Function

$h_\theta(x)$, function of x



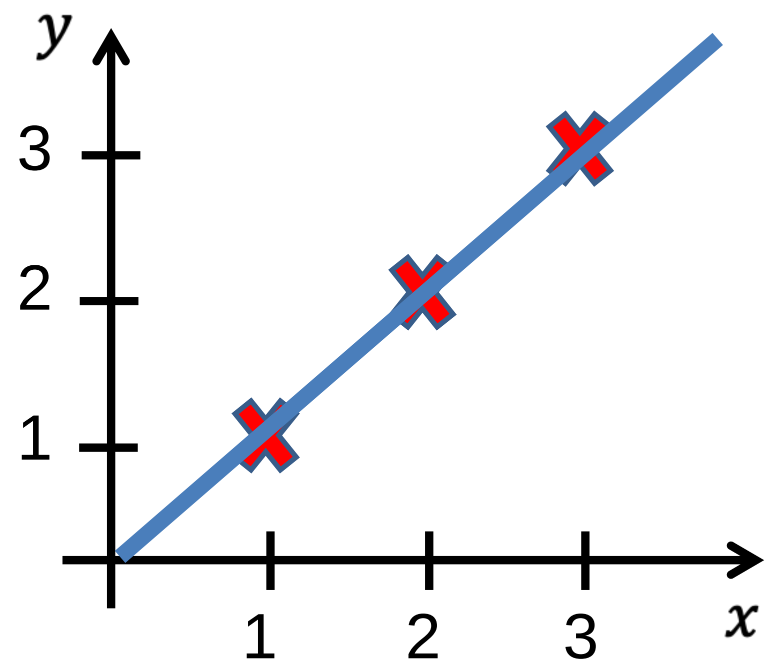
$J(\theta_1)$, function of θ_1



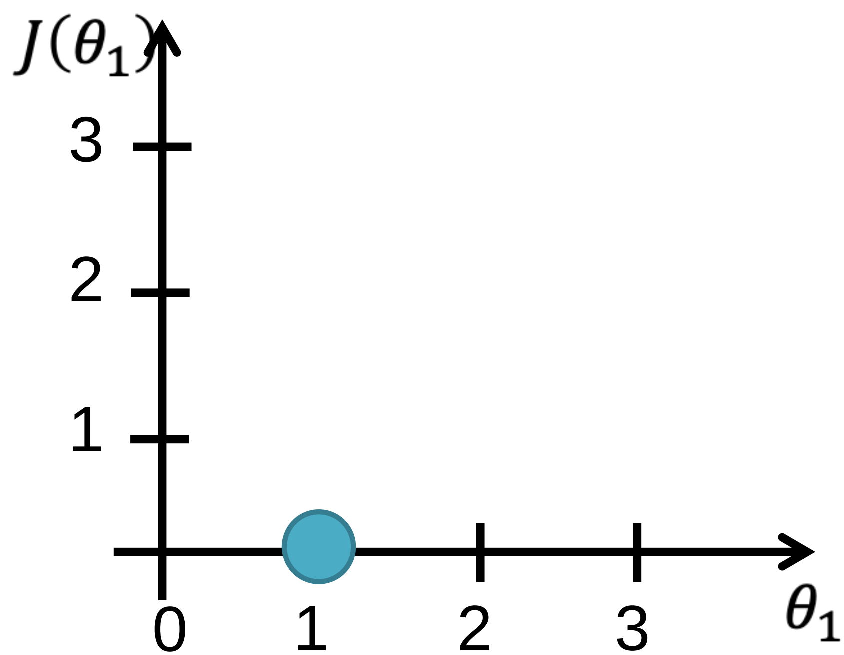
Acknowledgement: Andrew Ng, Stanford

Continued...

$h_\theta(x)$, function of x



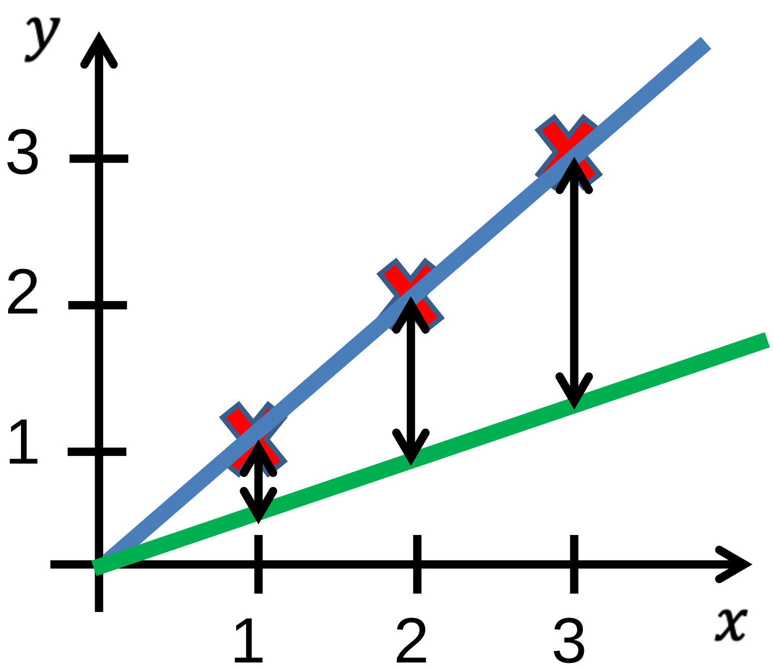
$J(\theta_1)$, function of θ_1



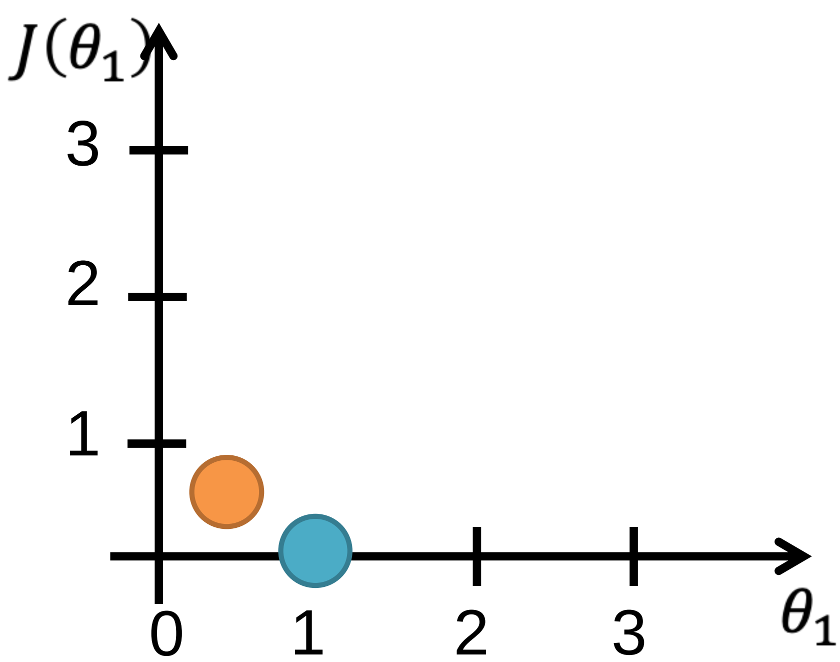
Acknowledgement: Andrew Ng, Stanford

Continued...

$h_\theta(x)$, function of x



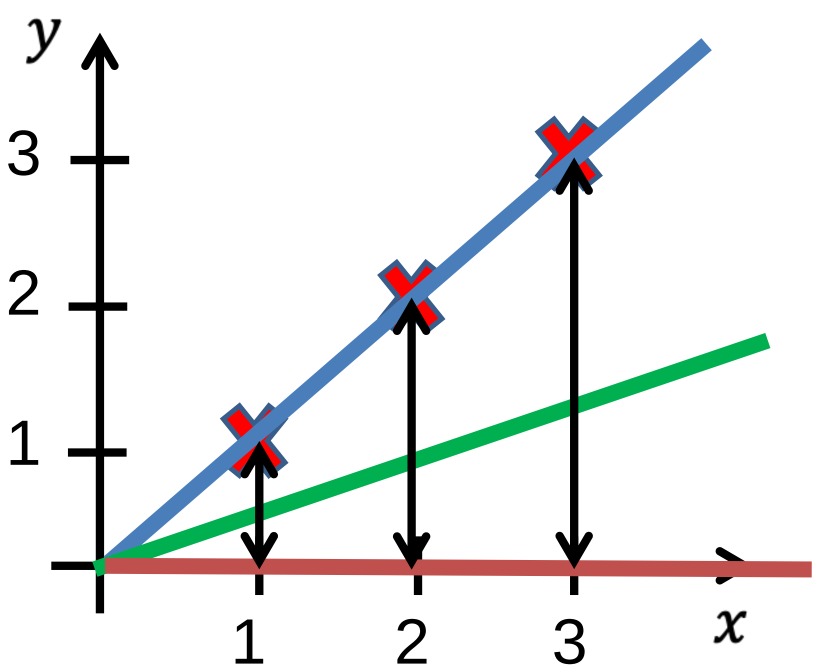
$J(\theta_1)$, function of θ_1



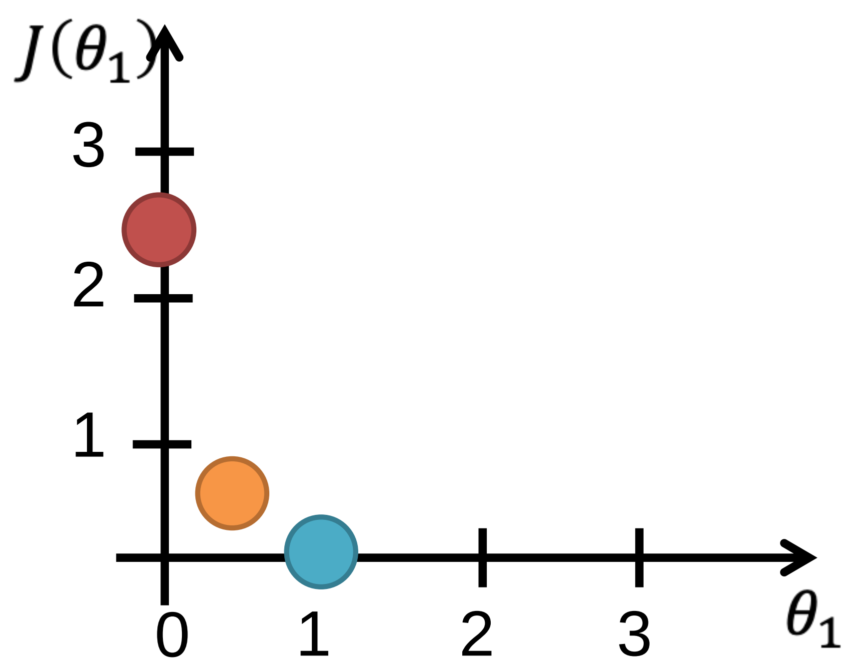
Acknowledgement: Andrew Ng, Stanford

Continued…

$h_\theta(x)$, function of x



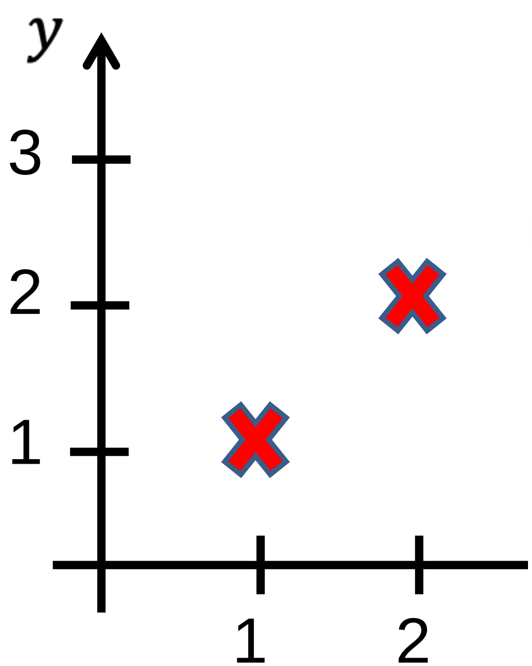
$J(\theta_1)$, function of θ_1

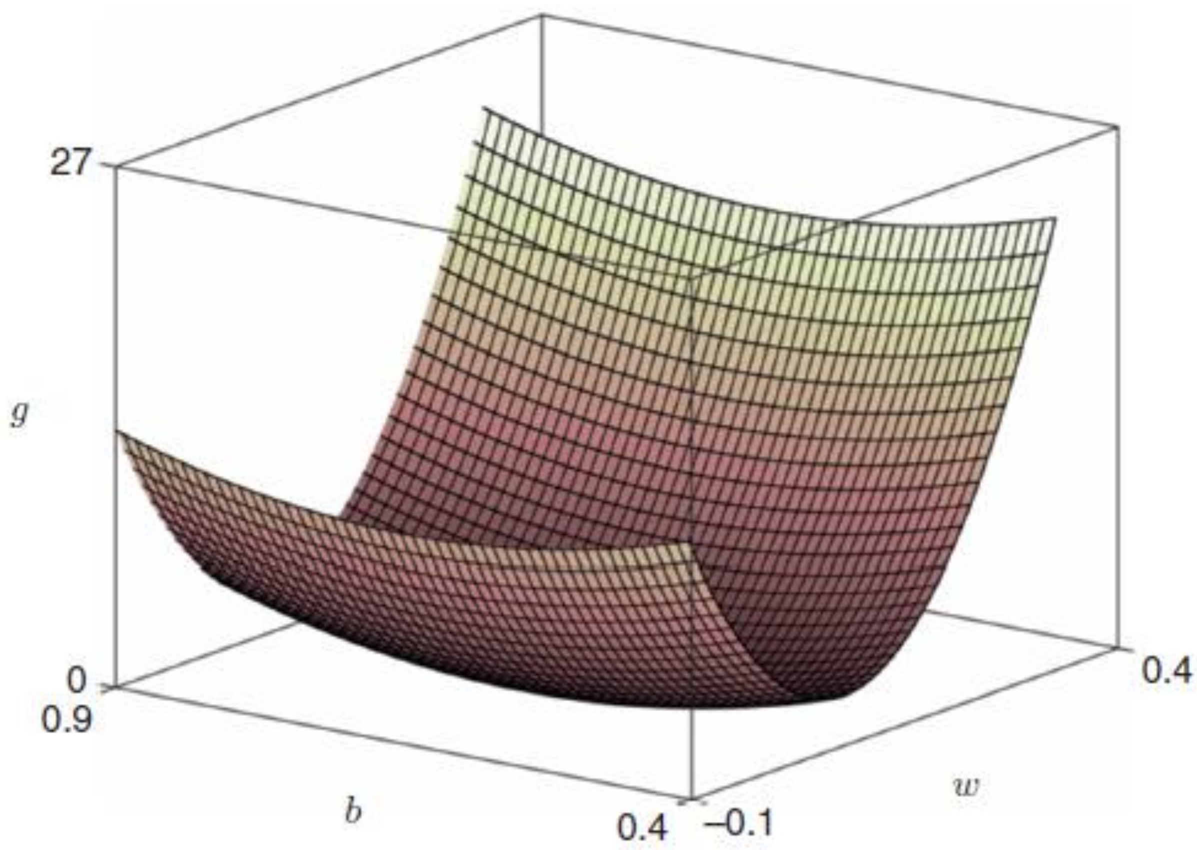


Acknowledgement: Andrew Ng, Stanford

Continued...

$h_\theta(x)$, functi





MSE cost function for linear regression is always Convex.

Gradient Descent: Minimizing the MSE

- Optimization algorithm used to minimize the MSE function by iteratively adjusting parameters in the direction of the negative gradient, aiming to find the optimal set of parameters.

Img. Source: https://www.analyticsvidhya.com/

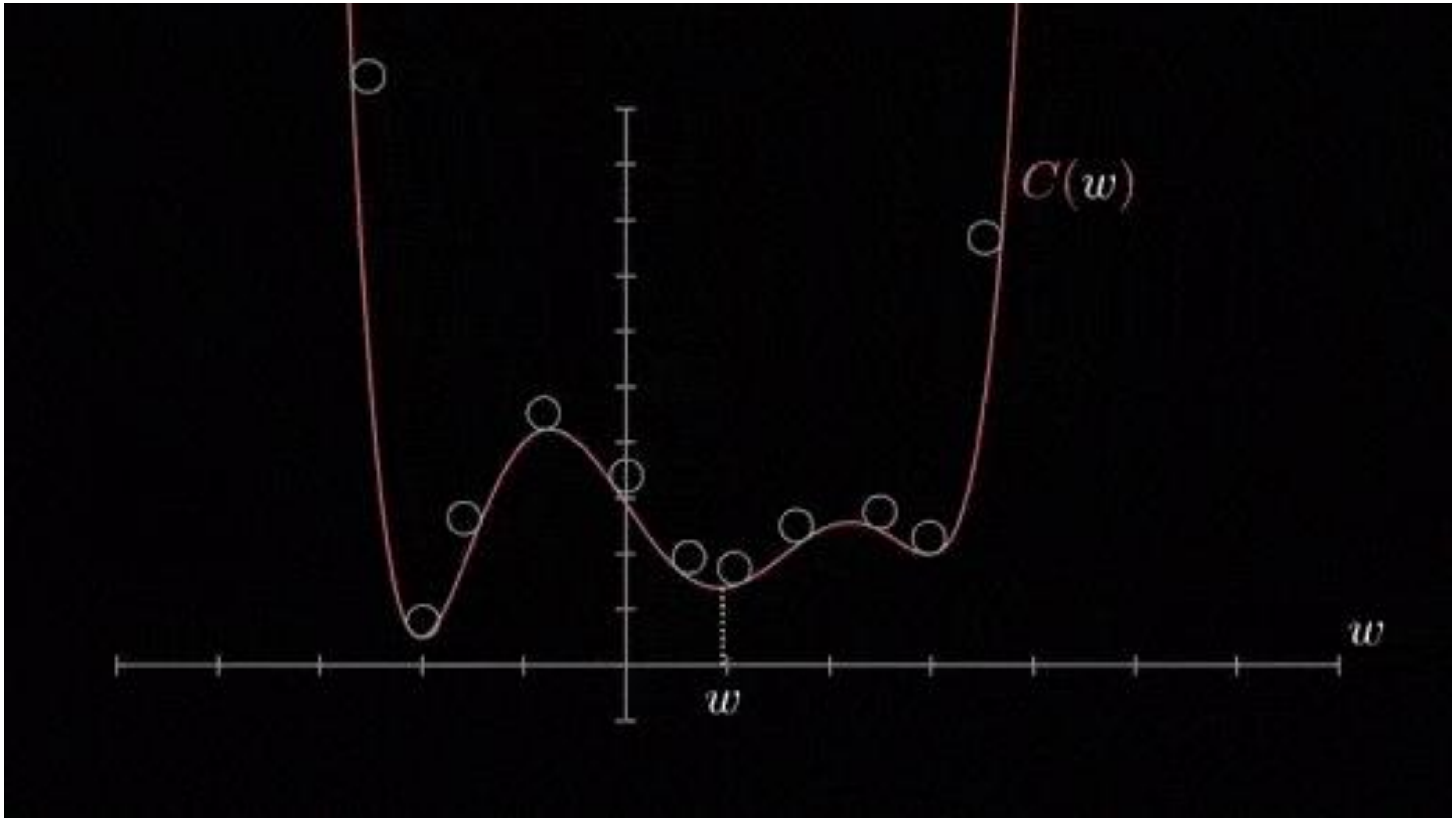
If we represent the gradient of the loss function as ∇L, and the parameters we are optimizing as θ:

Then the update rule for gradient descent is:

$$\theta\_new = \theta\_old - \alpha * \nabla L$$

Move in the opposite direction of the gradient.

Many local minima in gradient descent



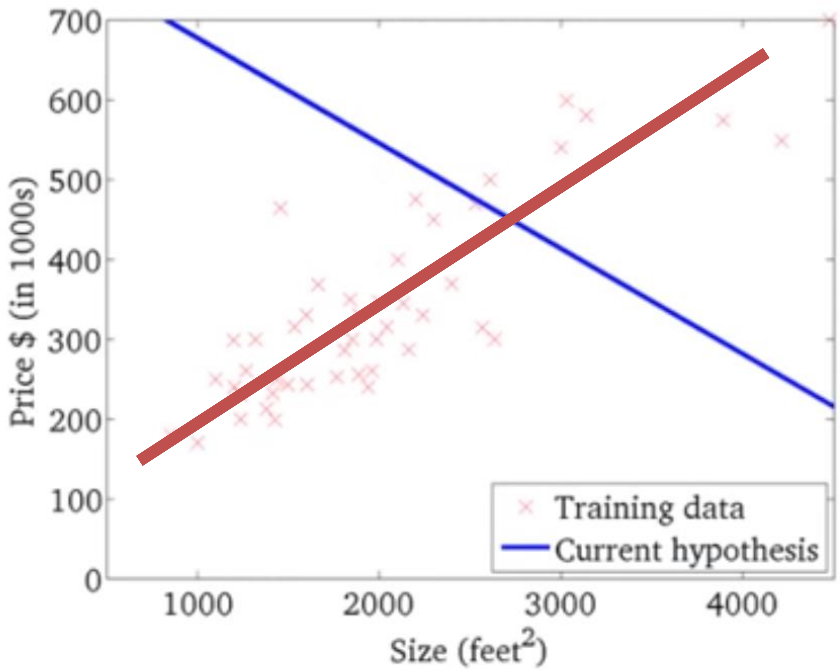
MSE cost function is Convex. Will you get many local minima? No, only one global minima.

Reason: If you pick any two points on the curve, the line joining them will never cross the curve.

Visualizing Gradient Descent

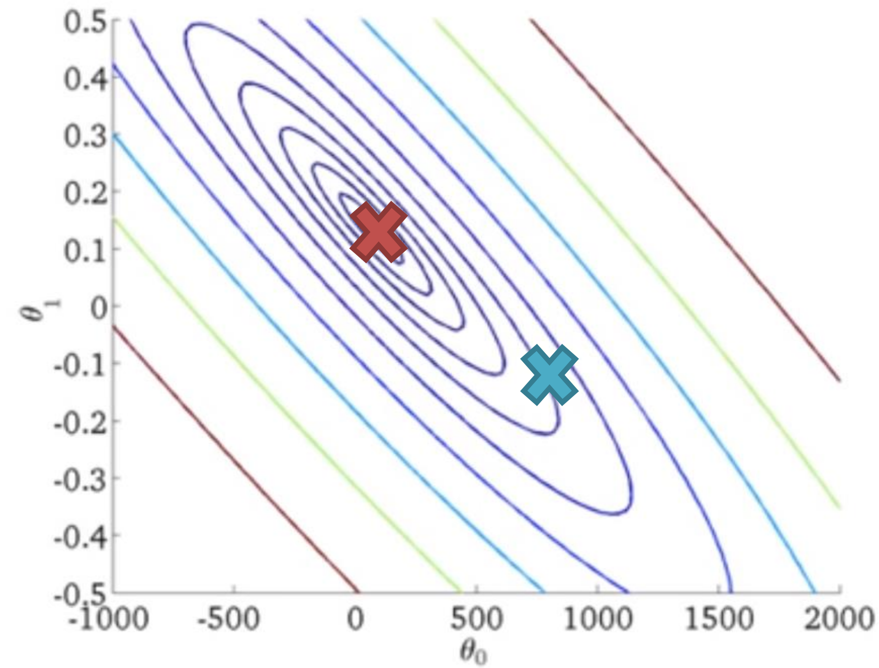
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

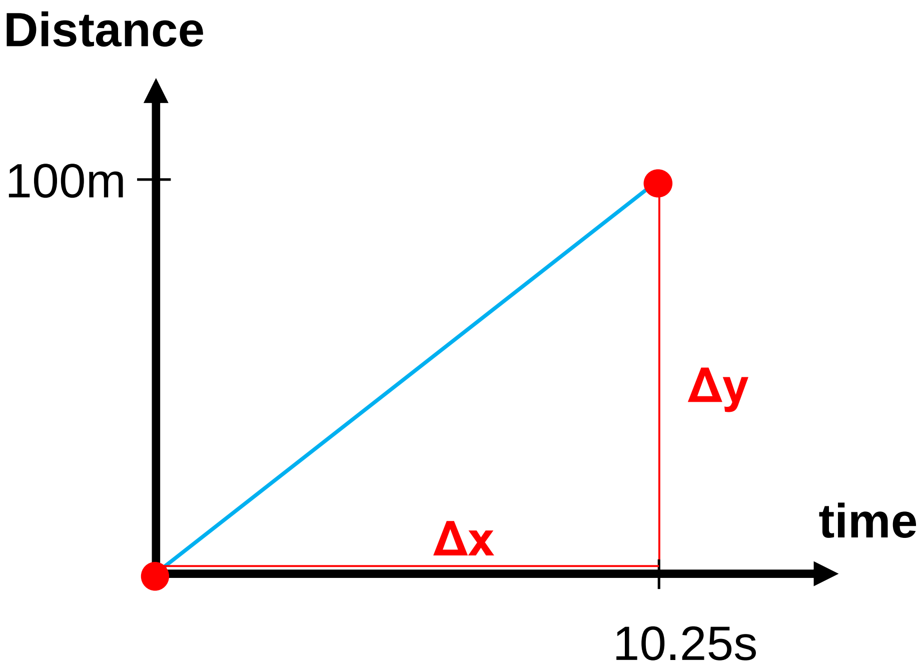
(function of the parameters θ_0, θ_1)



(visualized by using Contours)

Acknowledgement: Andrew Ng, Stanford

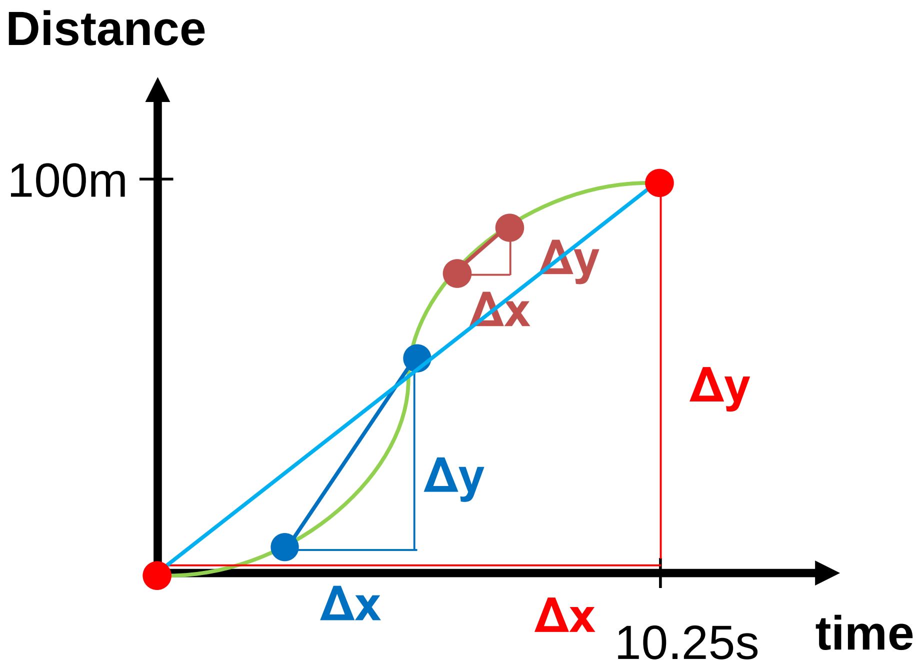
A bit of Math: Derivative of a Function?



What is his Average Speed? **Δy/ Δx**

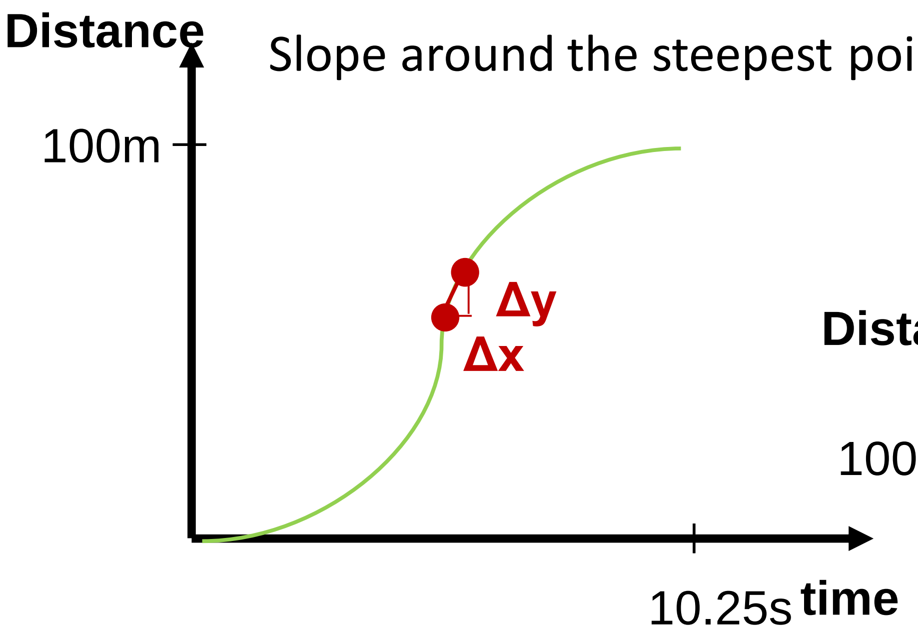
Amlan Borgohain

Instantaneous Speed Vs Average Speed



Will the Δy/Δx or Δy/Δx be different than the average slope, i.e., Δy/Δx? ✔

What would be really the Instantaneous speed?

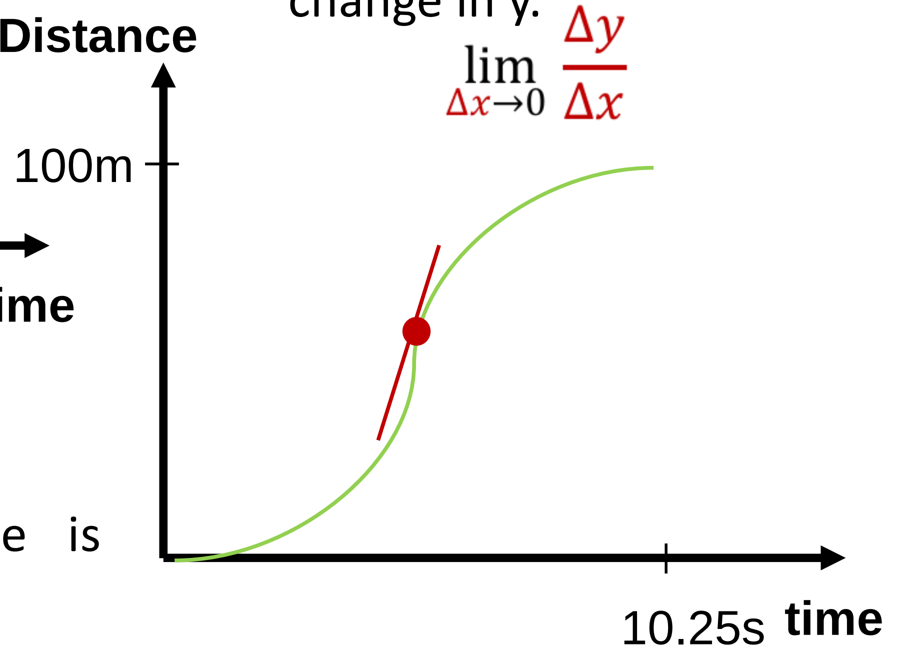


Slope around the steepest point.

Better approximation:

Measure the slope with a smaller and smaller change in x that yields a smaller and smaller change in y.

$$\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}$$



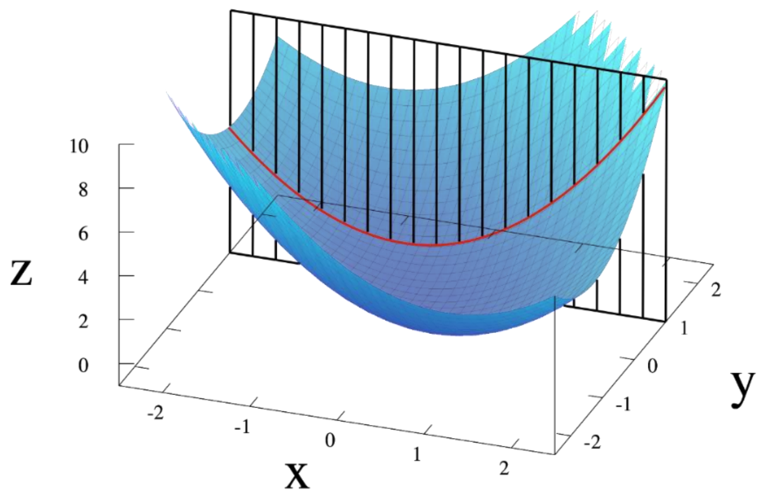
Fastest Instantaneous speed?

An Approximation: As the slope is changing constantly.

Instantaneous Slope is called Derivative: $\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}$ = dy/dx

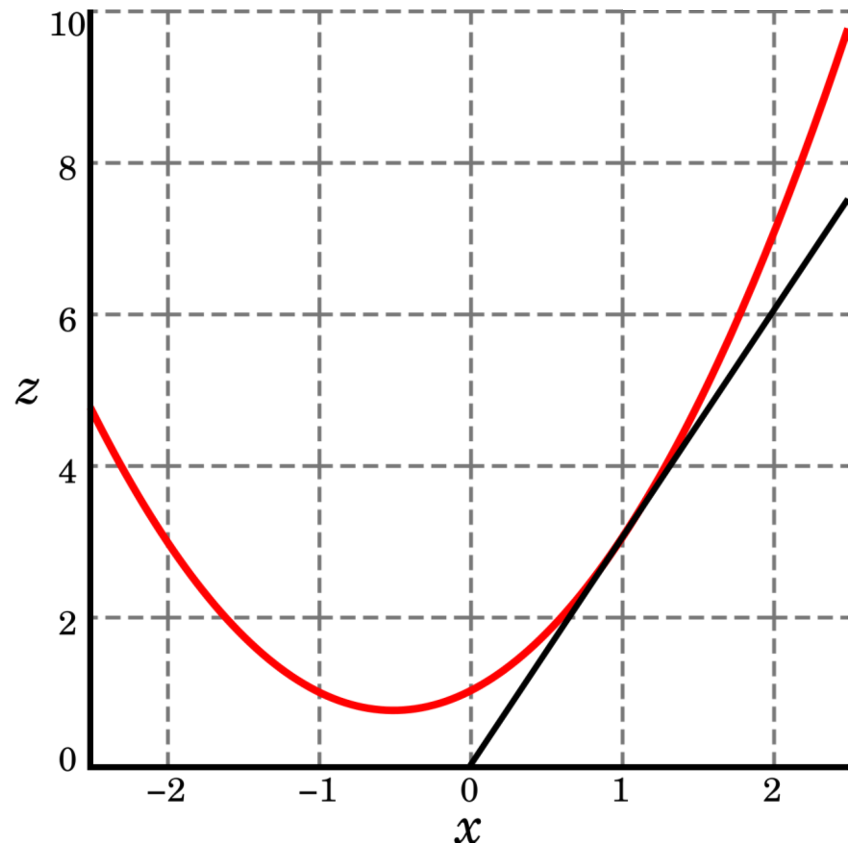
What is Partial Derivative?

What is the partial derivative of this function at P(1,1)? $\frac{\partial z}{\partial x} = 3$



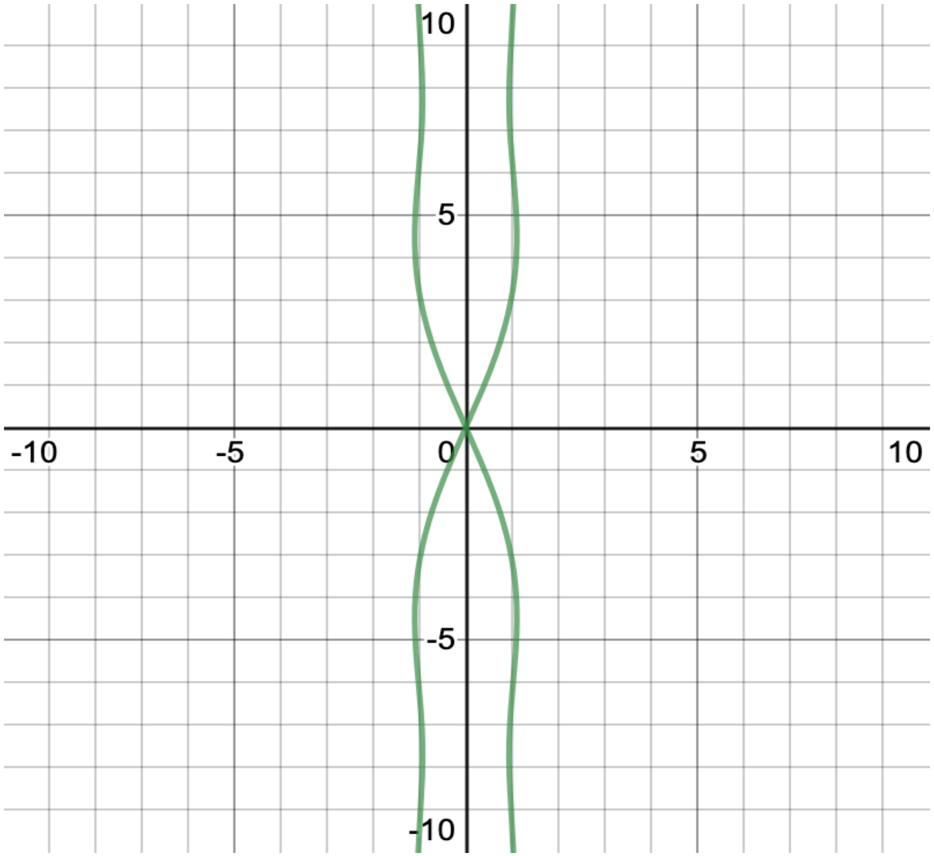
$$z = f(x, y) = x^2 + xy + y^2$$

That is the slope of *f* at the point (*x*, *y*)



(Img. Source: Wiki)

Gradient: All partial derivatives together



Multivariate Function: $f(x, y) = x^2 y + \sin(y)$

$$\frac{\partial f}{\partial x} = 2xy$$

$$\frac{\partial f}{\partial y} = x^2 + cos(y)$$

Gradient

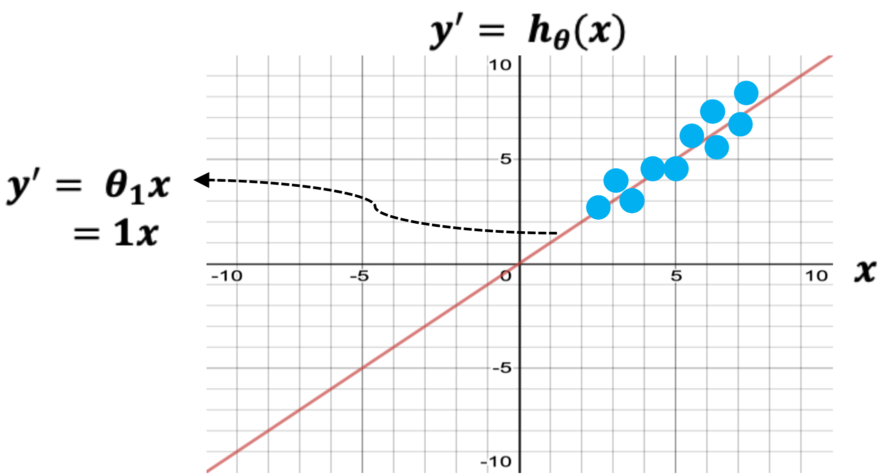
$$\nabla f(x, y) = \nabla\, x^2 y + \sin(y)$$

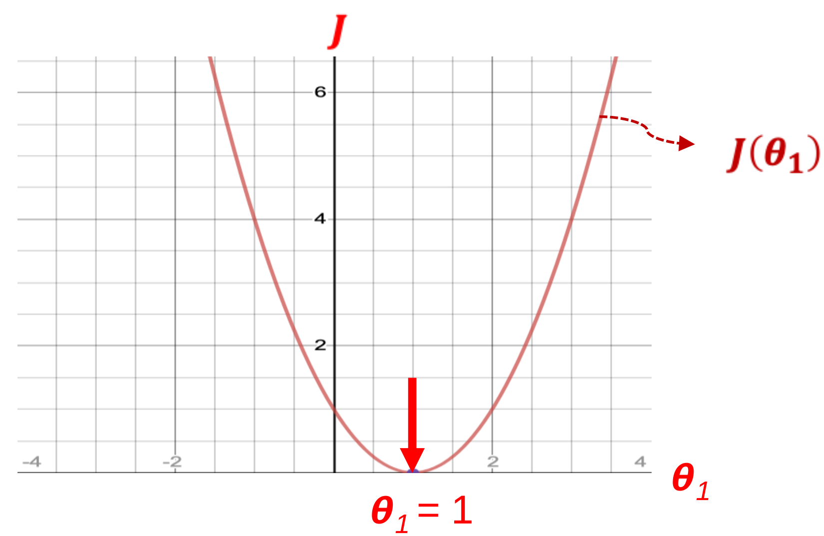
$$= \begin{bmatrix} 2xy \\ x^2 + \cos(y) \end{bmatrix}$$

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

The Impact of Partial Derviative

- For simplicity, let us assume our optimization objective is to $\underset{\theta_0,\theta_1}{\text{minimize}}\, J(\theta_1)$, thus, $\theta_0 = 0$



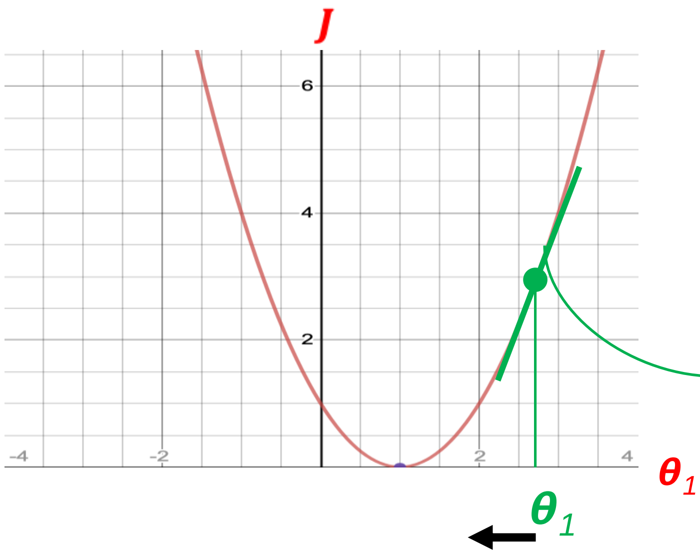


$h_\theta(x)$ is the ***Hypothesis Function***

$J(\theta_1)$ is the ***Cost Function***

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

Continued…



$$\theta_1 = \theta_1 - \alpha \frac{d\, J(\theta_1)}{d\, \theta_j}$$

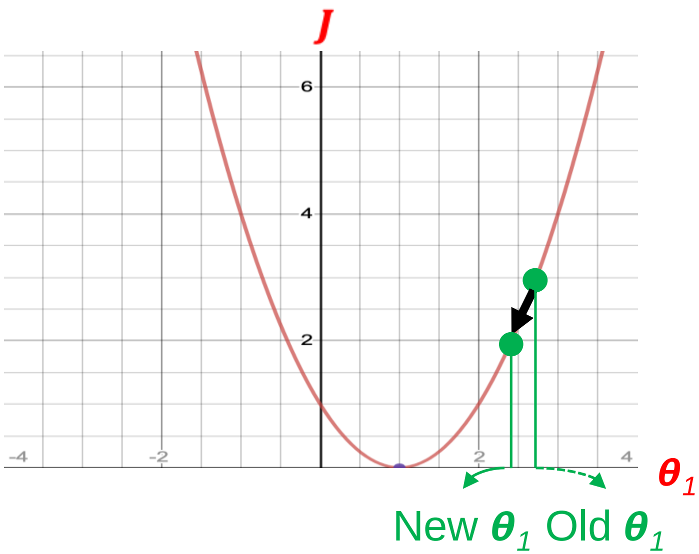
$$= \theta_1 - \alpha\, (Positive\ Number)$$

Decrease θ_1 by a certain value

Positive Derivative

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

Continued...



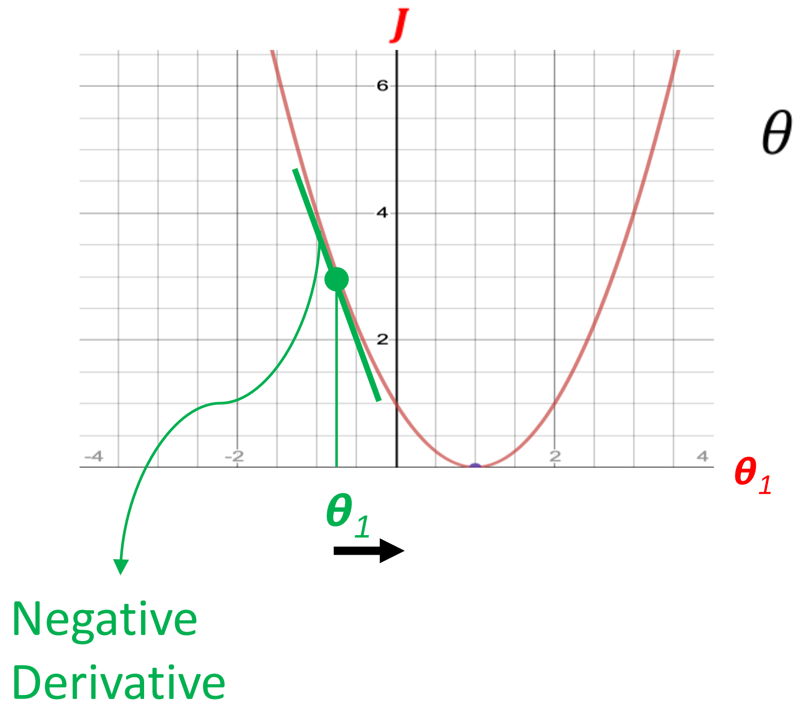
$$\theta_1 = \theta_1 - \alpha \frac{d\, J(\theta_1)}{d\, \theta_j}$$

$$= \theta_1 - \alpha\, (Positive\ Number)$$

Decrease θ_1 by a certain value

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

Continued...



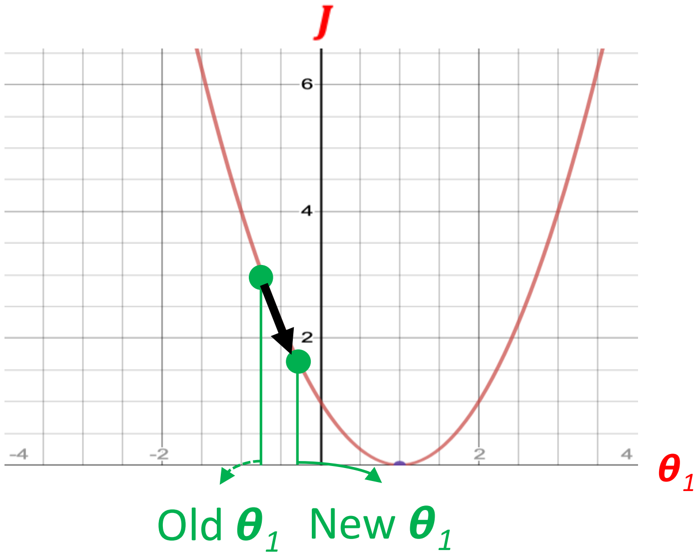
$$\theta_1 = \theta_1 - \alpha \frac{\boldsymbol{d}\, \boldsymbol{J}(\boldsymbol{\theta_1})}{\boldsymbol{d}\, \theta_j}$$

$$= \theta_1 - \alpha\, (Negative\ Number)$$

Increase θ_1 by a certain value

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

Continued...



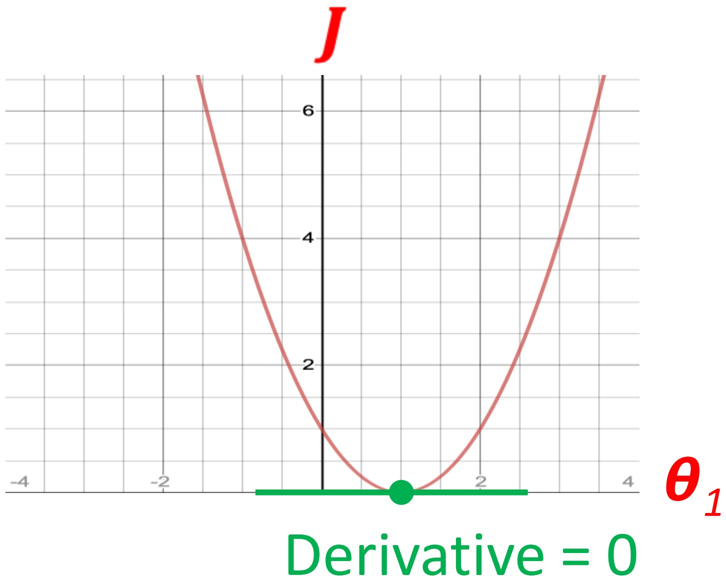
$$\theta_1 = \theta_1 - \alpha \frac{d\, J(\theta_1)}{d\, \theta_j}$$

$$= \theta_1 - \alpha\, (Negative\ Number)$$

Increase θ_1 by a certain value

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

Continued...



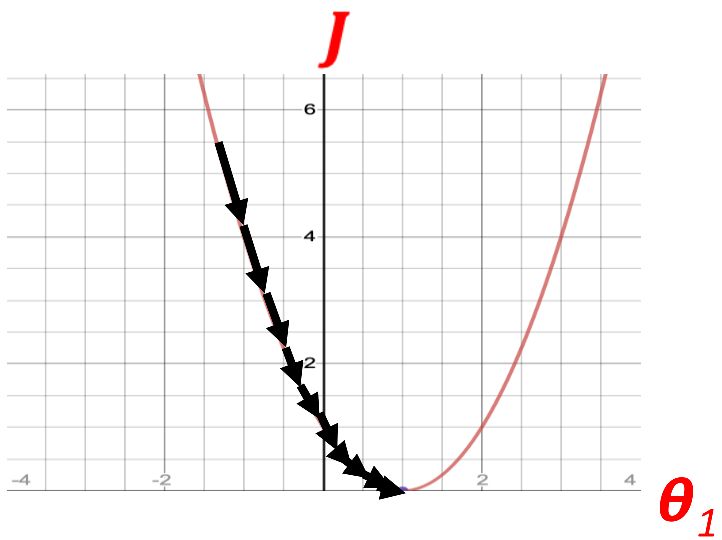
$$\theta_1 = \theta_1 - \alpha \frac{d\, J(\theta_1)}{d\, \theta_j}$$

$$= \theta_1 - \alpha\,(\text{Zero})$$

θ_1 remains the same, and hence, gradient descent has converged.

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

The Impact of Learning Rate



Too Small

Learing Rate

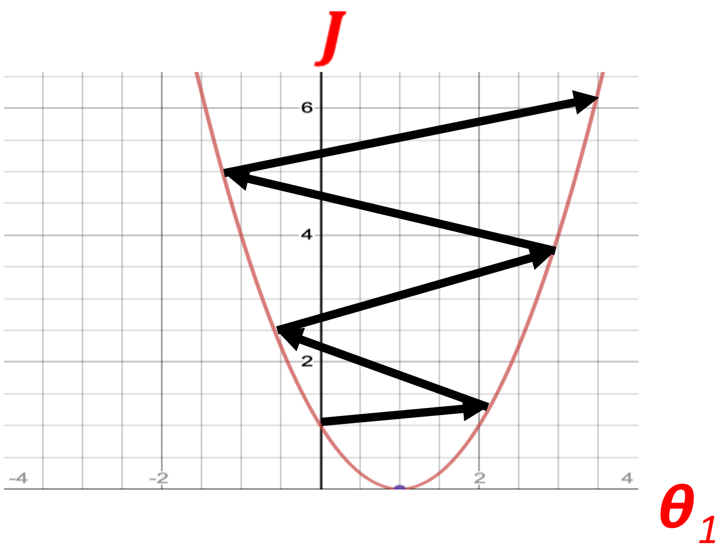
$$\theta_1 = \theta_1 - \alpha \frac{d\,J(\theta_1)}{d\,\theta_j}$$

$$= \theta_1 - (Too\ Small\ Number)\ \frac{d\,J(\theta_1)}{d\,\theta_j}$$

θ_1 changes only a tiny bit on each step, hence, gradient descent will render slow (will take more time to converge)

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

Continued...



$$\theta_1 = \theta_1 - \alpha \frac{d\, J(\theta_1)}{d\, \theta_j}$$

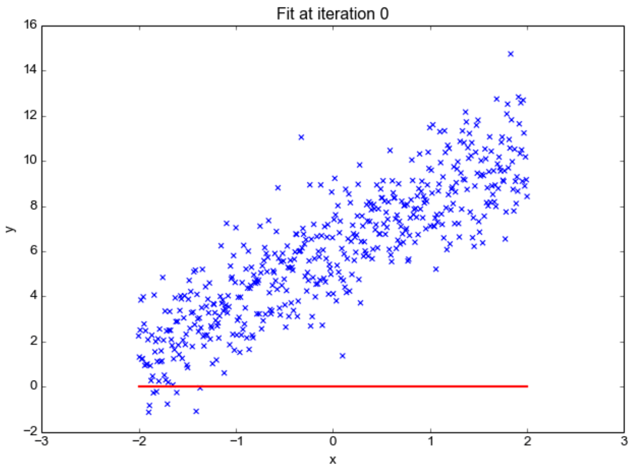
$$= \theta_1 - (Too\ Large\ Number) \frac{d\, J(\theta_1)}{d\, \theta_j}$$

θ_1 changes a lot (and probably faster) on each step, hence, gradient descent will potentially overshoot the minimum and, accordingly, fail to converge (or even diverge)

0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, ..., 0.9, 1

Acknowledgement: Mohammad Hammoud, CMU (Qatar)

Gradient Descent for Linear Regression



Linear regression model:

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

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

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

$$= \frac{\partial}{\partial\theta_j}\frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

Repeat until convergence{

$$j = 0: \quad \frac{\partial}{\partial\theta_0}J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

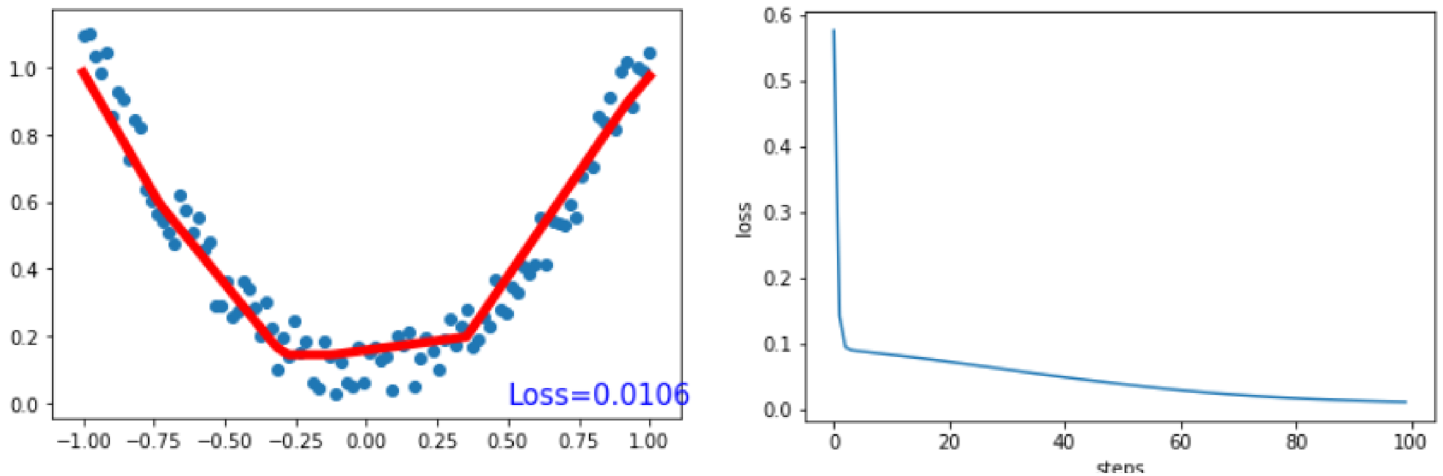
$$j = 1: \quad \frac{\partial}{\partial\theta_1}J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

}

Update θ_0 and θ_1 simultaneously

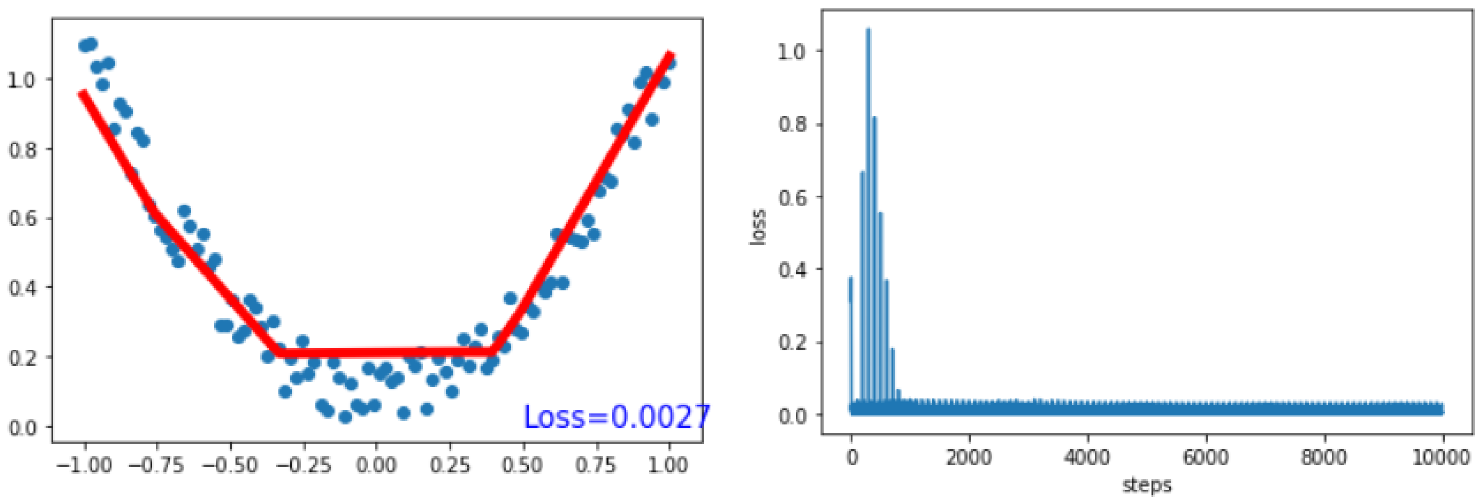
Batch Vs Stochastic Gradient Descent

GD



Very smooth convergence, however using all the data for one update.

SGD



Very noisy convergence, because using only one data point for one update.

Regression vs. Classification

Aspect	Regression	Classification
Objective	Predict continuous values or a range of values (3.4, 8.6, …)	Predict categorical labels (0 or 1; cat, dog, sheep; low,medium,high)
Example	House prices; Stock prices; Body Mass Index; Energy consumption etc.	Spam emails; Image classification; Loan approval (approved/ not approved), Customer churn etc.
Evaluation metrics	MSE, RMSE, MAE, R^2	Accuracy, Precision, Recall, F1, AUC
Algorithms	Linear regression, Ridge, Lasso, Polynomial regression, DT with numerical targets etc.	Logistic regression, DT with categorical targets, Naïve Bayes, SVMs, KNN, …
Types of problems	Continuous outcome (how much?)	Discrete outcomes (which class?)

Logistic Regression

- The linear regression model discussed in the previous class assumes that the dependent variable is quantitative (continuous).

- However, in many situations, the dependent variable is instead qualitative (categorical)

- A patient arrives at the campus medical (BITS) with cough, fever and runny nose.
  - Which disease the patient has? Influenza (Flu) (20-30%), Acute Bronchitis (15-25%), Common cold (10-20%).

Question: Which one is dependent and which one is Independent variable?

Logistic Regression

Subject: Urgent Action Required to Confirm Your Account

Dear Valued Customer,

We have noticed unusual activity on your account and for your protection, we have temporarily suspended access until further verification is completed.

Please follow the instructions below to restore access:

1. Click on the link below to verify your account details: Click Here to Verify Your Account
2. Update your account information by providing the requested details.
3. Failure to verify your account within the next 24 hours will result in permanent suspension of access.

Thank you for your prompt attention to this matter. We apologize for any inconvenience this may cause, and appreciate your cooperation in ensuring the security of your account.

Best regards,
Customer Support Team
[Random Company]

Credential Theft (20-30%)

Malware Distribution (15-20%)

Question: Which one is dependent and which one is Independent variable?

Logistic Regression

- **Logistic regression** is a type of linear regression that predicts the probability of an event occurring based on one or more input features. It's widely used for binary classification problems.
- How does it work?

Step1: Linear combination: Calculate a linear combination of the input features and their weights, which is represented by the equation:

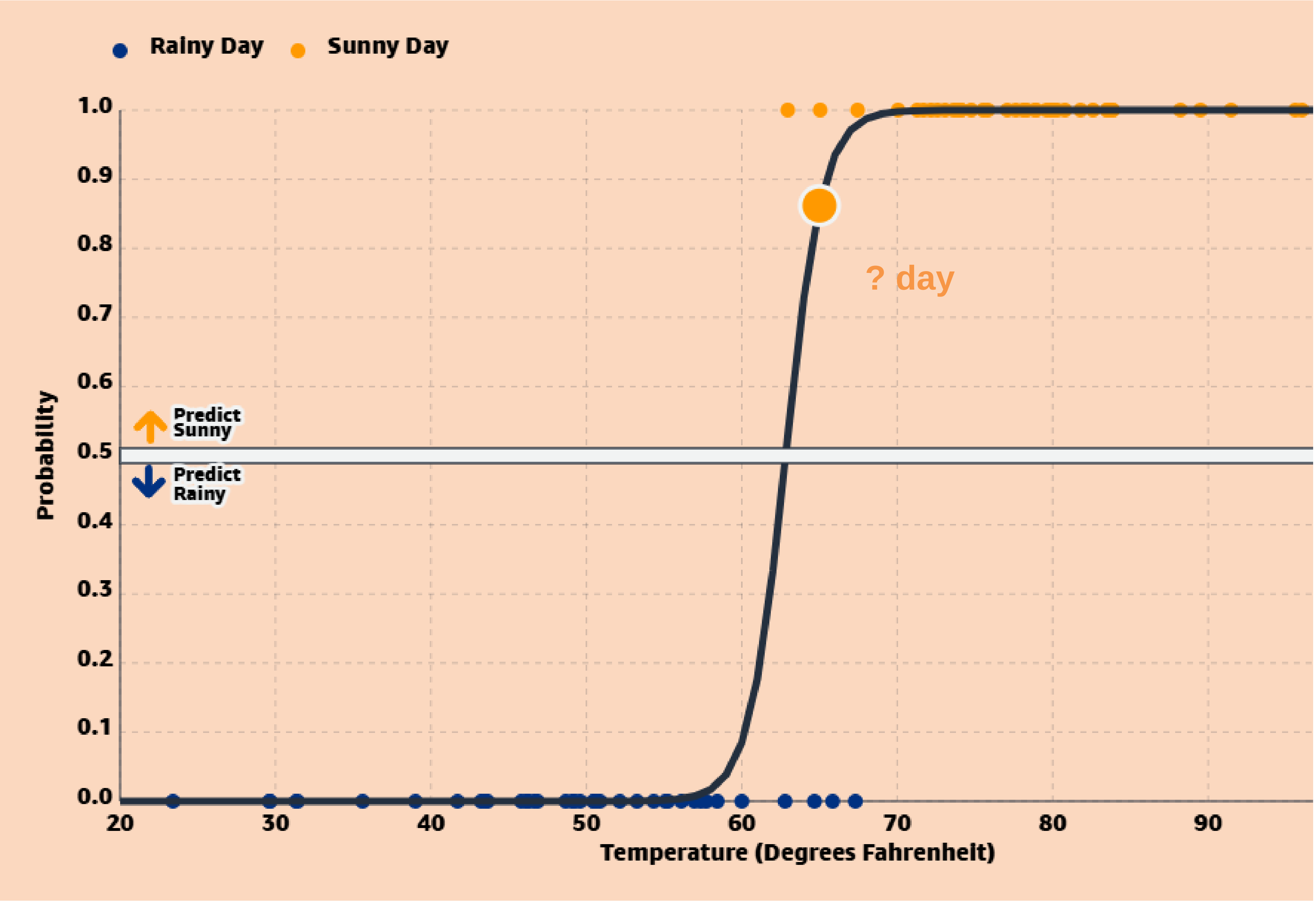
$z = \beta_0 + \beta_1 . x_1 + ... + \beta_n . x_n$, where 'z' is the log odds score.

Step2: Apply the logistic function (also known as the Sigmoid) to the linear combination result (z):

$p = 1 / (1 + \exp(-z))$

Step3: Thresholding: Compare the predicted probability with a threshold value (usually set to 0.5). If $p > 0.5$, predict class 1; otherwise, predict class 0.

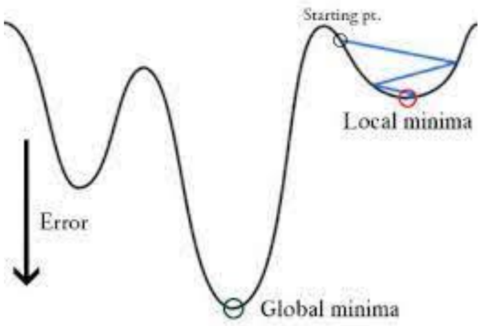
Example: Hiking in Seattle



Should we fit a linear regression model to this data? No

Loss function for logistic regression

- If you use MSE for Logistic regression, what problems it might create?



- A suitable loss function in logistic regression is called the Log-Loss, or binary cross-entropy. This function is:

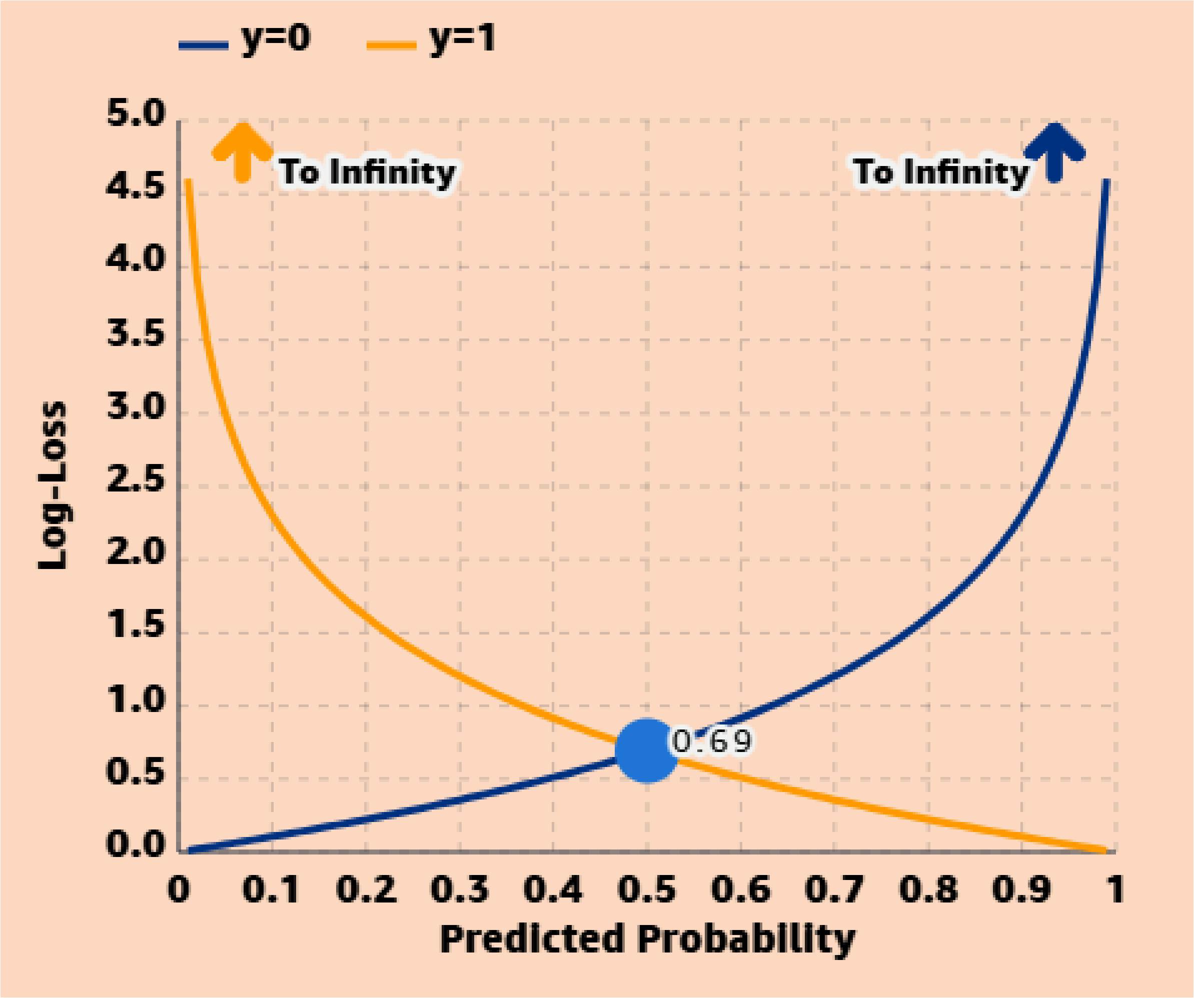
$$\text{Cost} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right]$$

- It penalizes deviations (incorrect probability predictions), offering a continuous metric for optimization during model training.

$$\text{Cost} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] + \boxed{\frac{\lambda}{2}\sum_{j=1}^{n}\beta_j^2}$$

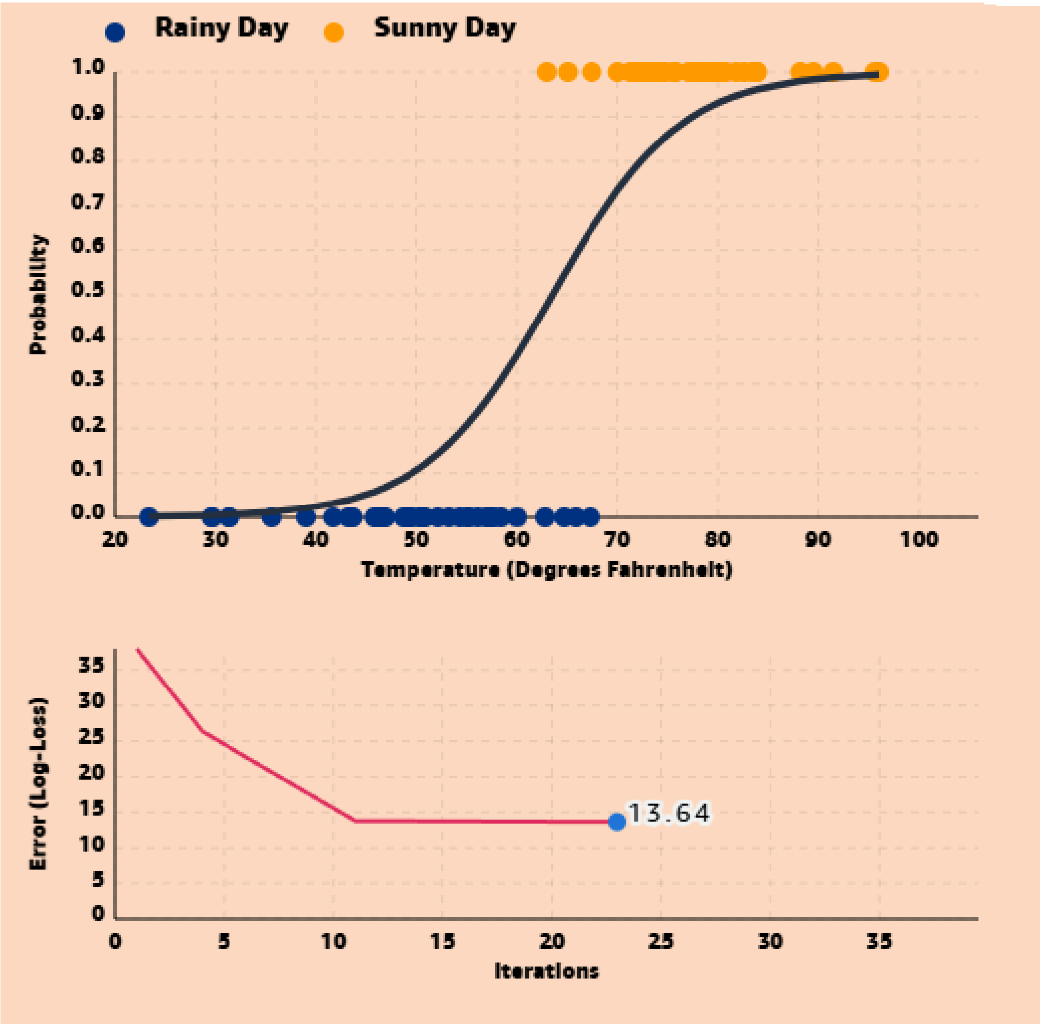
What is it?

Why Log-Loss?



You can see how as the probability gets closer to the true value (p=0 when y=0 and p=1 when *y*=1), the Log-Loss decreases to 0. As the probability gets further from the true value, the Log-Loss approaches infinity.

How Gradient Descent Works for Logistic Regression?



Weight: 0.2
Bias: -10
Our Model: $P(y=1|x) = \dfrac{1}{1+e^{-(-10+0.2x)}}$

Weight: 0.1872
Bias: -10.0002
Our Model: $P(y=1|x) = \dfrac{1}{1+e^{-(-10.0002+0.1872x)}}$

(1 step)

Weight: 0.1602
Bias: -10.0007
Our Model: $P(y=1|x) = \dfrac{1}{1+e^{-(-10.0007+0.1602x)}}$

(5 steps)

Weight: 0.1574
Bias: -10.0007
Our Model: $P(y=1|x) = \dfrac{1}{1+e^{-(-10.0007+0.1574x)}}$

(10 steps)

Source: mlu-explain.github.io/logistic-regression/

Chances of Admission to BITS Pilani: Ex.

Student	BITSAT Math	BITSAT Physics	BITSAT Chemistry	12th Percentage	Admission (0 = No, 1 = Yes)
1	70	80	75	85	1
2	60	65	60	80	0
3	85	90	80	88	1
4	55	50	60	78	0
5	90	85	88	92	1

Define the Logistic Regression Model:

If $p \geq 0.5$, predict admission $= 1$ (admitted).

If $p < 0.5$, predict admission $= 0$ (not admitted).

$$p = 1/(1 + e^{-(\beta 0 + \beta 1. \text{Math} + \beta 2. \text{Physics} + \beta 3. \text{Chemistry} + \beta 4.\text{12th Percentage})})$$

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

# Example data
X = [
    [70, 80, 75, 85],  # Math, Physics, Chemistry, 12th score for student 1
    [60, 65, 60, 80],  # Student 2
    [85, 90, 80, 88],  # Student 3
    [55, 50, 60, 78],  # Student 4
    [90, 85, 88, 92]   # Student 5
]

y = [1, 0, 1, 0, 1]  # Admission outcomes (1 for admitted, 0 for not admitted)

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Train logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)
print(predictions)
```

```
[1]
```

Assignment 3

```
# Predict probability of admission for a new student
new_student = [[75, 82, 80, 87]]
probability = model.predict_proba(new_student)
print(probability)
```

```
[[0.00131369 0.99868631]]
```

Thank You!