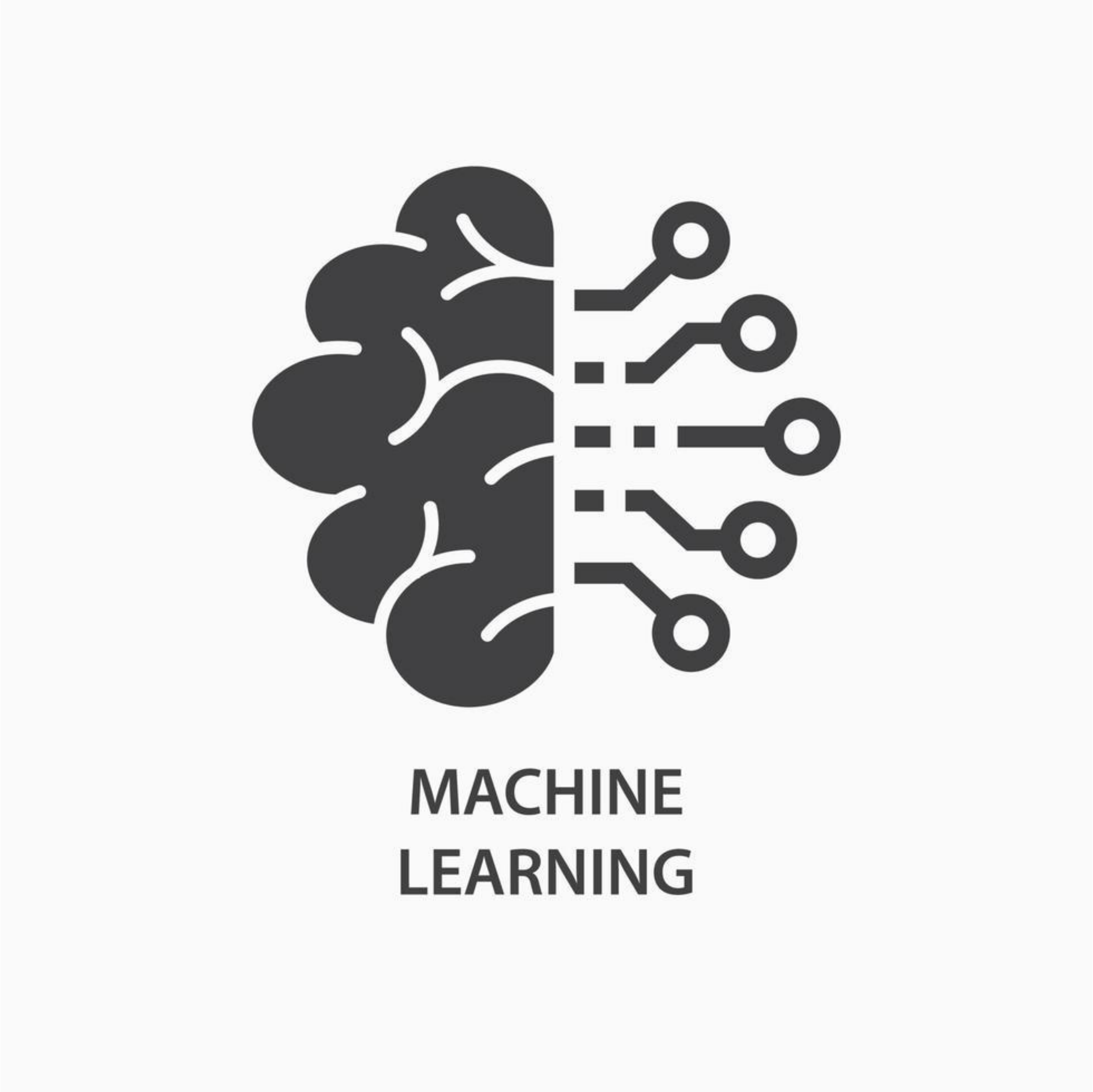
SUPERVISED LEARNING NOTES



Created by: Eng.Ahmed Osama

TABLE OF CONTENT

MAHINE LEARNING DEFINITION

Definition and Apps of ML

- It is sub field of AI, related with learn the machine how to act like humans, applications as
  - Recommending videos like latest watched videos
  - Recognizing Friends names from Instagram Image which uploaded recently
  - The Private Assistant on Mobile Phones like Siri, Bing, etc.
  - Check have Disease or not from X-ray or each another rays
- More Opportunities give for algorithm, will lead to high performance in asking

Machine Learning Process

- ML is a model using for predict new data, by learning from labeled or un labeled data, the input data named as X access for the model to get output Y
- Dataset Division
  - Training Data
    - It is the data which all X in it have it own Y
    - The model uses it for realize the structure of the data
  - Testing Data
    - It is the data which all X in it have not it own Y
    - The model uses it to test it knowledge at the data, by predict Y 's Value for the input X
  - Validation Data
    - It is the data which all X in it have not it own Y

- The model uses it to test it Completely knowledge at the data, by predict Y ‘s Value for the input X

Types of ML

Supervised Learning

- Type of Machine Learning
- More Popular than another types
- SL is type of ML define the prediction data, depends on learning from labeled data
- Have X ‘inputs’ and Y ‘outputs’ in Training Data
- Type of Supervised Learning
  - Regression
  - Predict real numbers
    - Predict Price of House as 12K, 14.5K, 1000K
  - Classification
  - Predict the category
    - Predict of detection Cancer type A or B or C
    - Predict of Image Classification as animal dog or cat or etc.

Unsupervised Learning

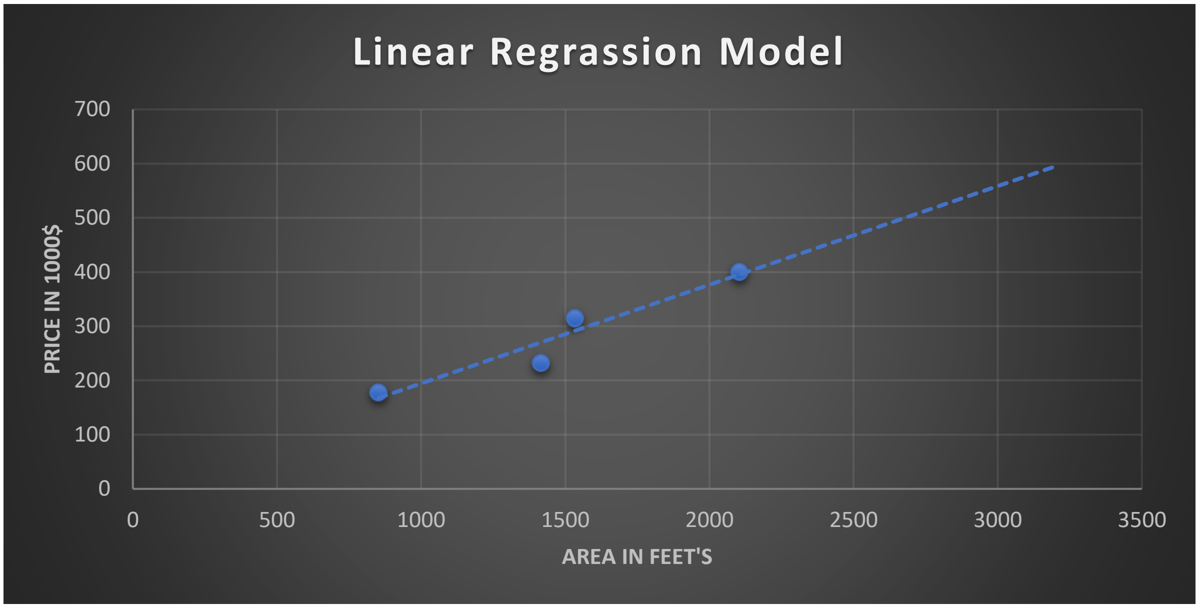
- Type of Machine Learning
- Lite Popular than another types
- USL is type of ML define the prediction data, depends on learning from Un labeled data
- Have only X ‘inputs’ in Training Data
- Type of Unsupervised Learning
  - Clustering
  - Predict by divide the data to clusters ‘groups’

- Anomaly Detection
- Predict the unusually events
- Using in Financial Systems or etc.
- Dimensionality Reduction
- Leads to Decrease the number of labels to few number to increase the prediction rate

LINEAR REGRSSION MODEL

- It is Supervised learning type of model
- It is using for define Infinity number of outputs
- Types
  - Univariate, with 1 feature
  - Multiple Variant, with multiple features
- Example, House Price Prediction with House Area
- Some Definitions about linear regression with 1 feature
  - X, the input ‘feature’ of the model
  - Y, the output ‘target’ of the model
  - I, the counter of row in model
  - M, number of training data of the model
  - (x, y), training example

Number ‘i’	Area in feet’s ‘x’	Price in 1000$ ‘y’
1	2104	400
2	1416	232
3	1534	315
4	852	178

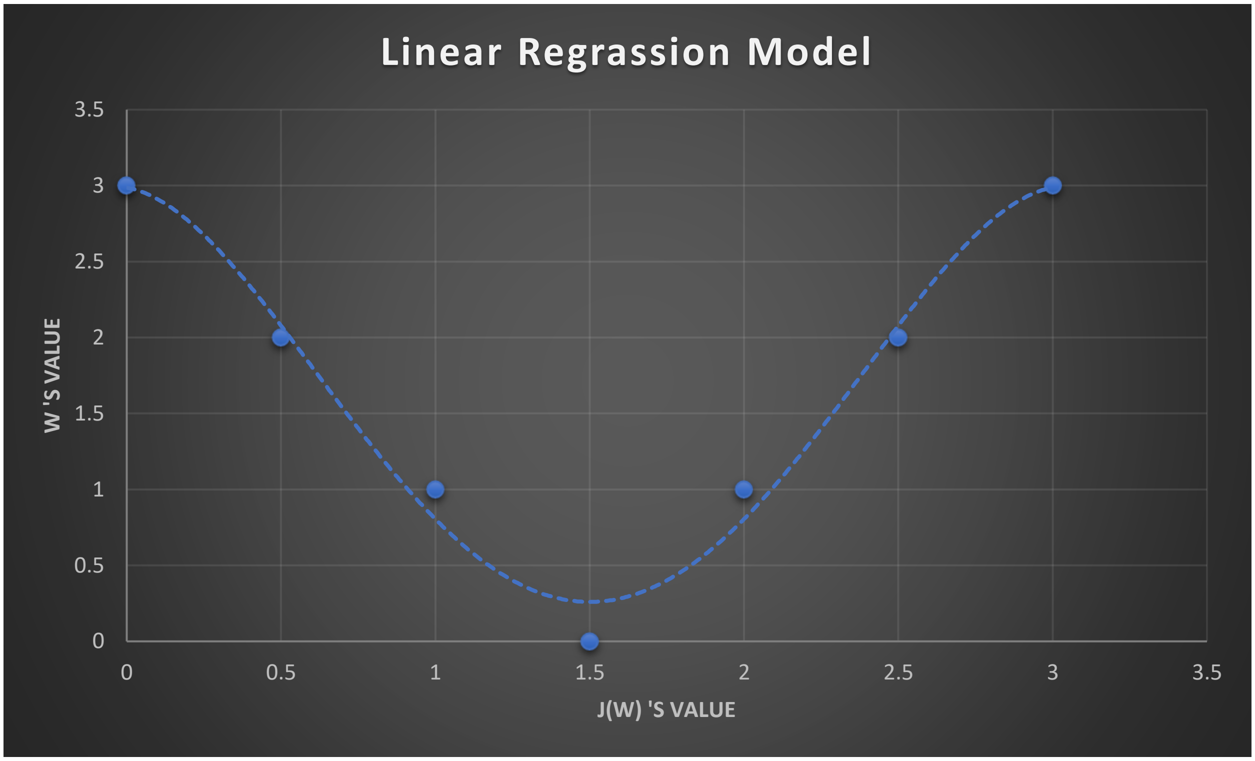


- To Predict y 's value will need to formula, using x 's value in this formula, as
  - ♦ F(x) = w*x + b
  - ➢ W, weight
  - ➢ B, bias
  - ➢ w & b, it is real numbers
- When use F(x) to Predict y, we will find the Predicted value is not exactly similar to main Y 'target or output', so that we will named the predicted value as y-hat ' '
- After Predict the Value, we will get y-hat 's value which equal to F(x) 's value, as
  - ♦ Y-hat = F(x)
- Here, we will get new value called 'Cost Function'

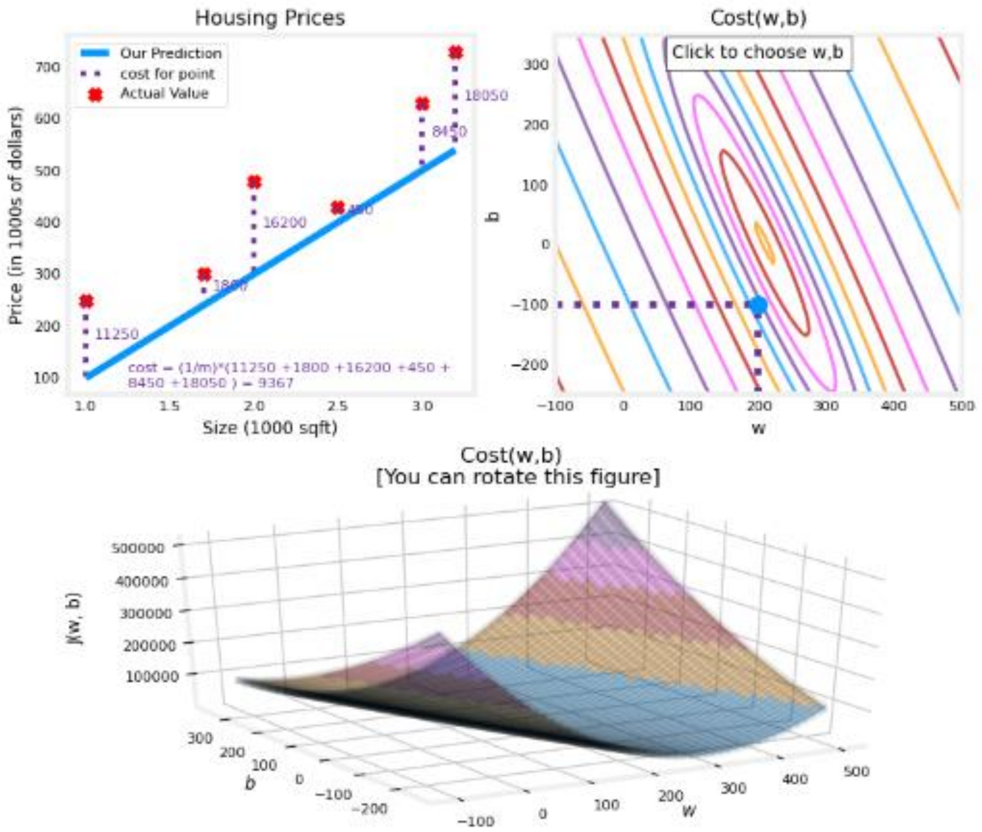
COST FUNCTION FOR LINEAR REGRESSION

- Cost Function, the squared error
  - The different between the main output ‘y’ and the predicted output ‘y-hat’
  - Calculated as
  - $J(w,b) = \frac{\sum_{I}^{M}("y-hat"^{i} - "y"^{i})^{2}}{2M} = \frac{\sum_{I}^{M}(\mathrm{F(x)}^{i} - "y"^{i})^{2}}{2M}$
    - M, number of training data
    - I, counter
    - Y-hat, the predicted value
    - F(x), the predicted value

- The Cost Function Minimizing Algorithm
  - F(x) = wx + b
    - Use, ignore b ‘s value as set it 0
    - Then, F(x) = wx + 0 = wx
    - Then, the Controlled Value is w
    - When apply new F(x) in J(w, b) as
    - J(w, b) will be J(w)
    - $J(w) = \frac{\sum_{I}^{M}\left((w * "x"^{i}) - "y"^{i}\right)^{2}}{2M}$
    - From this Graph we need to minimize J(w) ‘s value, using change w ’s value



- Use, w, b ‘s values
- The Plot will be 3D, so we use x axis as b, y axis as w & z axis as J(w, b)
- At 3D Graph, We search at the minimum value of J(w, b), as this plots in Image

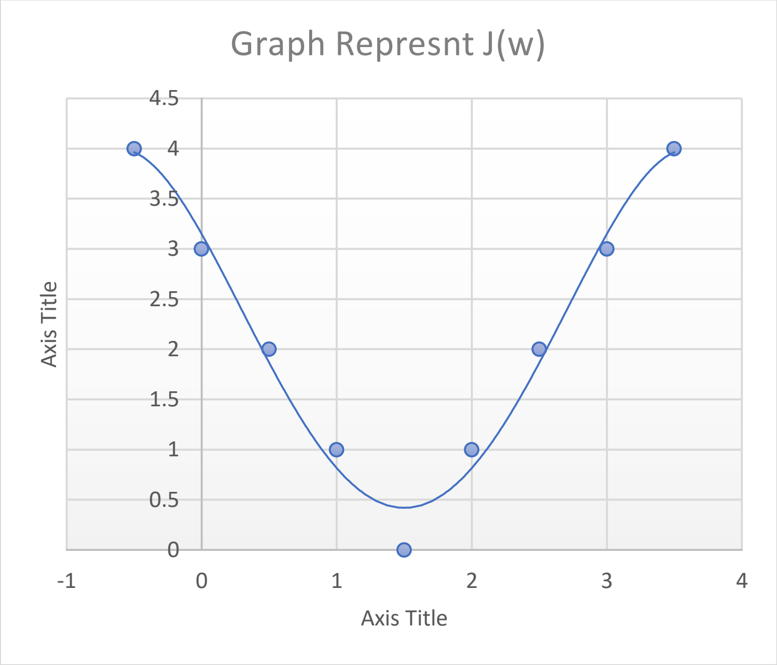

Housing Prices
Our Prediction
cost for point
Actual Value
Price (in 1000s of dollars)
Size (1000 sqft)
100
200
300
400
500
600
700
1.0
1.5
2.0
2.5
3.0
11250
1800
16200
450
8450
18050
cost = (1/m)*(11250 +1800 +16200 +450 + 8450 +18050) = 9367
Cost(w,b)
Click to choose w,b
b
w
300
200
100
0
−100
−200
−100
0
100
200
300
400
500
Cost(w,b)
[You can rotate this figure]
J(w, b)
500000
400000
300000
200000
100000
b
300
200
100
0
−100
−200
w
−100
0
100
200
300
400
500

GRADIENT DESCENT ALGORITHM FOR LINEAR REGRESSION

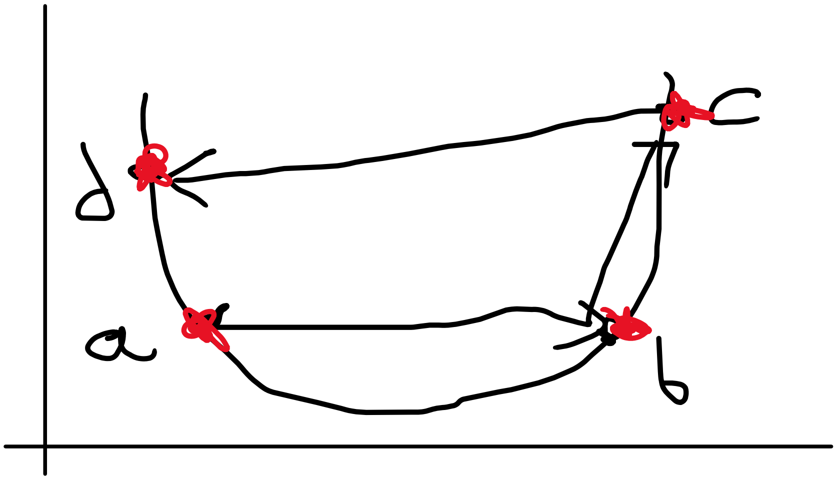
➢ The Gradient Descent Algorithm

- It is expression called at the movements ‘change in w, b values’ which leads to get the minimized j ‘s value, by Change w, b values step by step even J gets smaller and smaller till disappeared ‘be 0 or so close to 0’
- It is not created for only linear regression algorithm; it is using for many another algorithm
- So that we need to make updating in w, b values as, ‘WE MUST MAKE THIS STEPS REGULARLY’:
  - ♦ W-temp = $w - \ \alpha * \left(\frac{d}{dw} J(w, b)\right)$
  - ♦ B-temp = $b - \ \alpha * \left(\frac{d}{db} J(w, b)\right)$
  - ♦ $\frac{d}{dw} J(w, b) = \frac{\sum_{I}^{M} (w*x\text{”}i\text{”} - \text{“}y\text{”}i)^{\square} \ \text{“}x\text{”}i}{M}$
  - ♦ $\frac{d}{db} J(w, b) = \frac{\sum_{I}^{M} (w*x\text{”}i\text{”} - \text{“}y\text{”}i)^{\square}}{M}$
  - ♦ Then,
  - ♦ W = W-temp, W using in J(w, b) to Check Minimizing

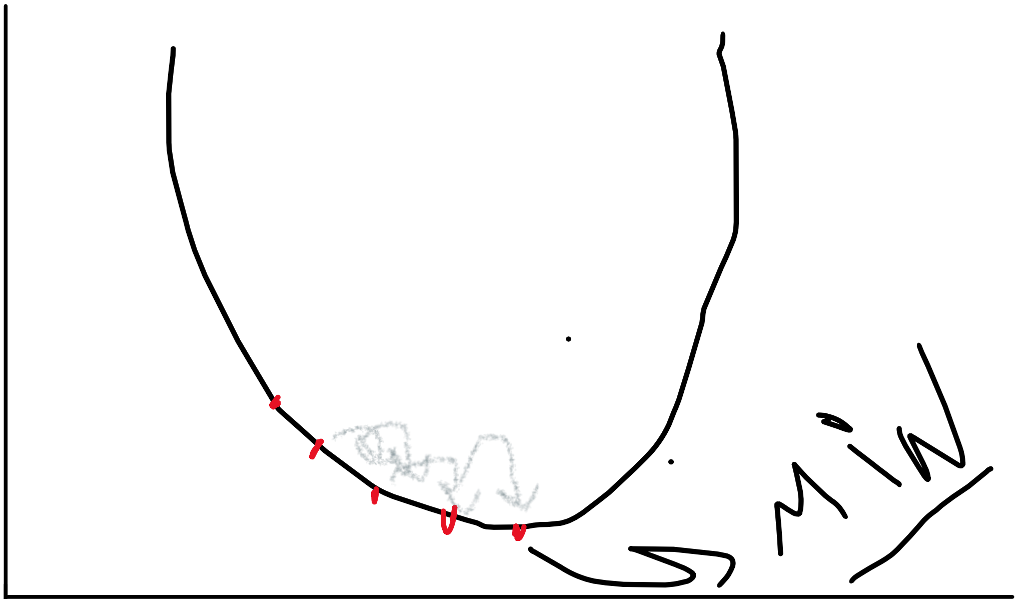
- B = B-temp, B using in J(w, b) to Check Minimizing

- $\left(\frac{d}{dw} J(w,b)\right)$
  - The derivative for the U graph works using Slop
  - If derivative 'slop' positive value, then w get closer to 0
  - If derivative 'slop' negative value, then w get away from
  - 

- α
  - The learning rate 'alpha'
  - If value so big, will causes in over shoot, so it step will be larger than needed, so that cannot reach to min point in graph, as



- If value so tine, will causes many of steps, so it step will be so tine, so that can reach to min point in graph, as



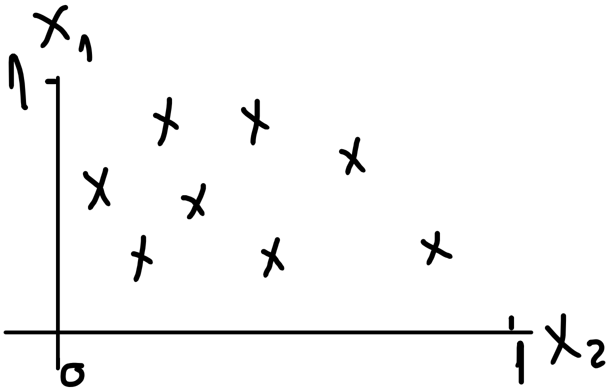
MULTIPLE LINEAR REGRESSION MODEL

- It is type of linear regression model
- It is using for define Infinity number of outputs, Depends on Multiple features
- Some Definitions about linear regression with n feature
  - x^i, the vector of row number i
    - EX, [2, 2014, 20]
  - x_j, the vector of column number j
    - EX, [2, 3, 3, 1, 7]
  - X_j^i, the value of cross pointing at matric x(i,j)
    - EX, X_2^2 is 1416, X_1^3 is 3
  - n, number features

Number 'j'	Beds Num 'x1'	Area in feet's 'x2'	Age of House 'x3'	Price in 1000$ 'y'
1	2	2104	20	400
2	3	1416	12	232
3	3	1534	21	315
4	1	852	321	178

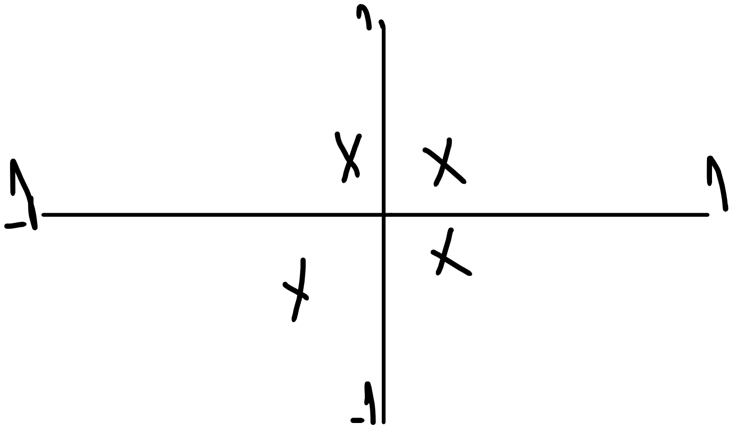
- To Predict y ’s value will need to formula, using x ‘s values in this formula, as
  - F(x) Applied as
    - $w \cdot x \ + \ b$
    - $w_1 * x_1 + w_2 * x_2 + \cdots + \ w_n * x_n + b$
    - $\sum_n^1 w_n * x_n + b$
  - W, weight
  - B, bias
  - w&x, it is a vectors
  - w.x, it is dot product
  - b, it is real numbers
- When use F(x) to Predict y, we will find the Predicted value is not exactly similar to main Y ‘target or output’, so that we will named the predicted value as y-hat ‘ ’
- After Predict the Value, we will get y-hat ‘s value which equal to F(x) ‘s value, as
  - ♦ Y-hat = F(x)

FEATURE SCALING

- If Using Multiple Features in linear regression model will find Problem Between the Huge Different between Feature and another one, so that they create Feature scaling to change feature’s values to be $0 < value \ \leq 1$, they scale as:
  - Assume Feature named x1
  - x1 is in period $a1 < x1 \ \leq a2$
  - a1 is the smallest value of x1
  - a2 is the biggest value of x1
  - will convert using $x_{j,scaled} = \frac{x_j}{a_2}$
  - 

- The Mean Normalization, using for make feature’s values around the zero, negative and positive values as
  - Assume Feature named x1
  - x1 is in period $a1 < x1 \ \leq a2$
  - a1 is the smallest value of x1

- a2 is the biggest value of x1
- will convert using $x_{j,scaled} = \frac{x_j - \mu_j}{a_2 - a_1}$, $\mu_j = \frac{\sum_{j=1}^{n} x_j}{n}$



-

➢ The Z-score Normalization, using for make feature's values around the zero, negative and positive values as

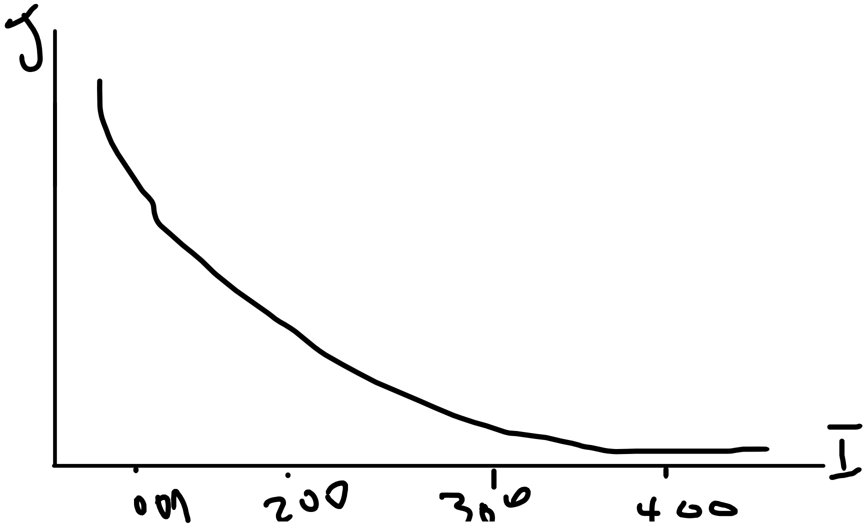
- Assume Feature named x1
- x1 is in period $a1 < x\acute{1} \leq a2$
- a1 is the smallest value of x1
- a2 is the biggest value of x1
- will convert using $x_{j,scaled} = \frac{x_j - \mu_j}{\alpha_j}$, $\mu_j = \frac{\sum_{j=1}^{n} x_j}{n}$

➢ When making Scaling if

- Values too large
- Values too tiny

GRADIENT DESCENT ALGORITHM CONVERGE

- The Number of iterations affect at the learning curve of gradient descent algorithm as .

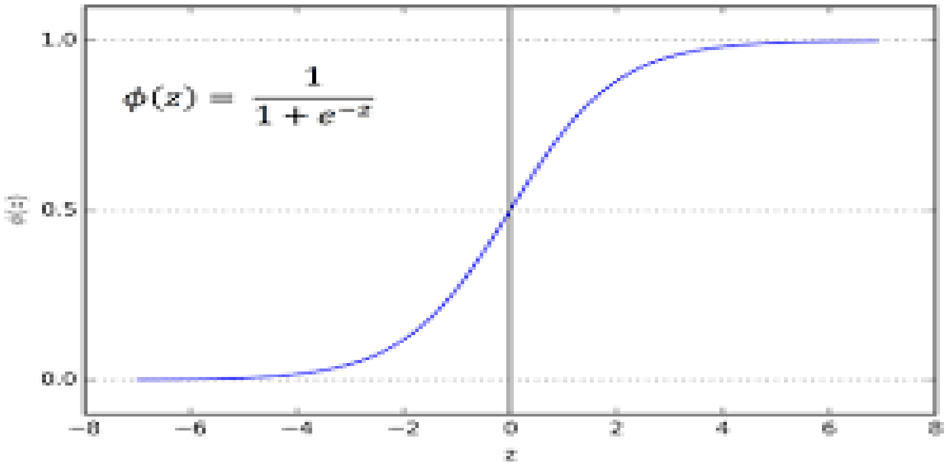


FEATRUES ENGINEERING

- it is a concept using usually to create new features to enhance the prediction of the model as ex
  - Assume we have predicted house price, have
    - x1 as Length of house
    - x2 as Depth of the house
  - then we will make feature engineering as create new feature named as x3 is the area of the house
    - x3 = x1 * x2, this is the new feature

CLASSIFICATION LOGISTIC REGRESSION

- It is Supervised learning type of model
- It is using for define its category or class
- Example, Realize Image Content Cat or Dog
- Linear Regression, it is not suitable algorithm for deals with classification, so it is works well with infinity of data not classes
- Some Definitions about logistic regression
    - To apply this model, we need to additional algorithm named ‘sigmoid function’
        - Sigmoid function, it is formula to reshape the model graph as
        - $g(z) = \frac{1}{1+e^{-z}}$, $0 < g(z) < 1$



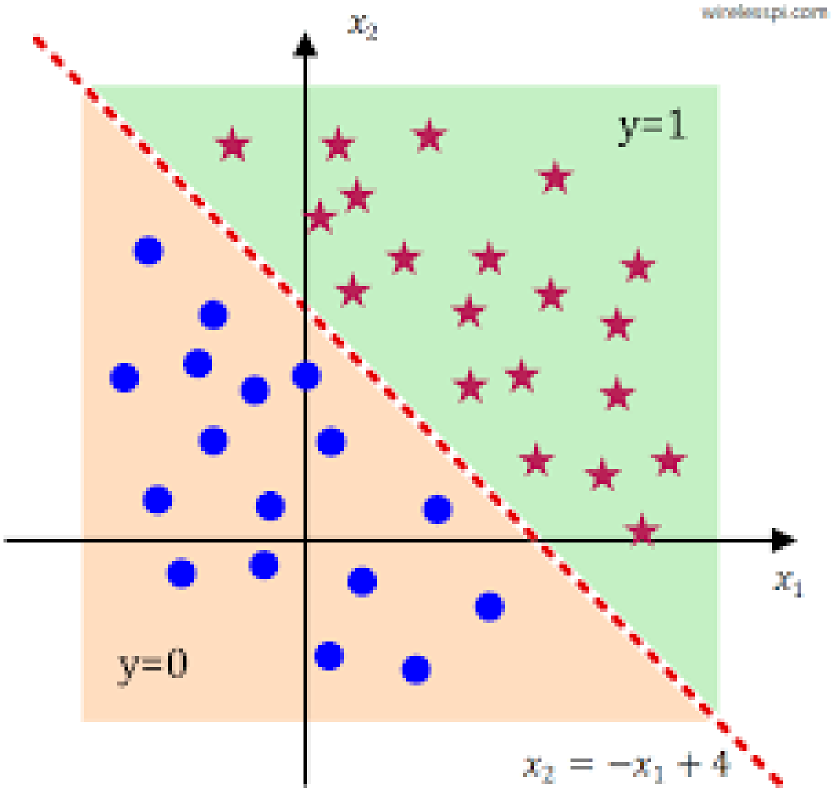
- To apply sigmoid function, we need ‘z’ value
- z, it is will be f(w, b) $z = f(x) = w * x + b$
- the final logistic regression model formula is

- $g(z) = \frac{1}{1+e^{-(w*x+b)}}$
- Large Z, Small Denominator, Close to 1 ‘yes’ ‘True’
- Small Z, Large Denominator, Close to 0 ‘No’ ‘False’

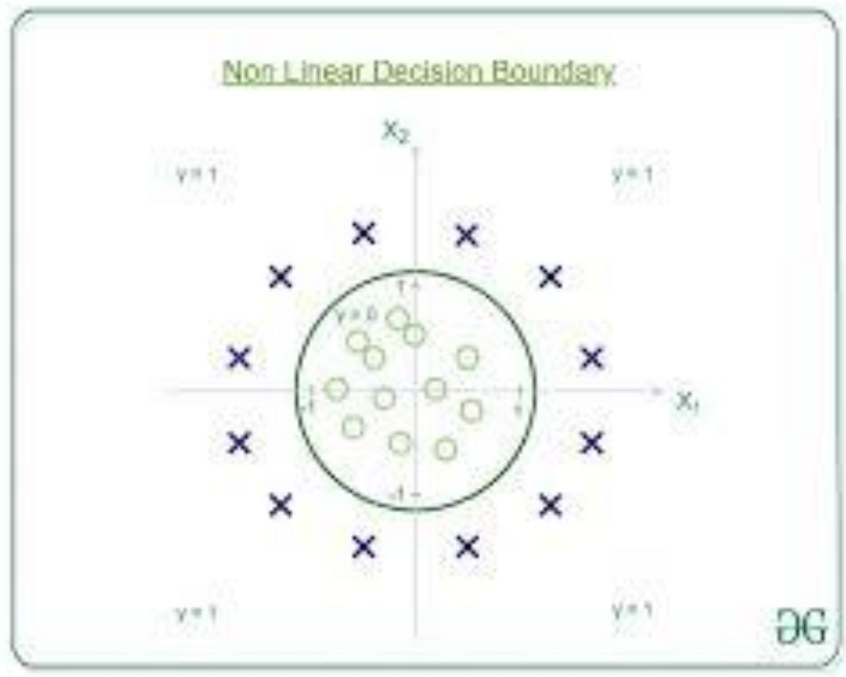
- To Classify the value will using probability if <= 50% will be Classify as High Class
- To Classify the value will using probability if > 50% will be Classify as Low Class

DECISION BOUNDARY

- The Decision Boundary
  - It is the shape which difference between Classes, sometimes it is Simple 'Linear', another time it is Complex 'polynomial or Non-Linear'
    - Linear as

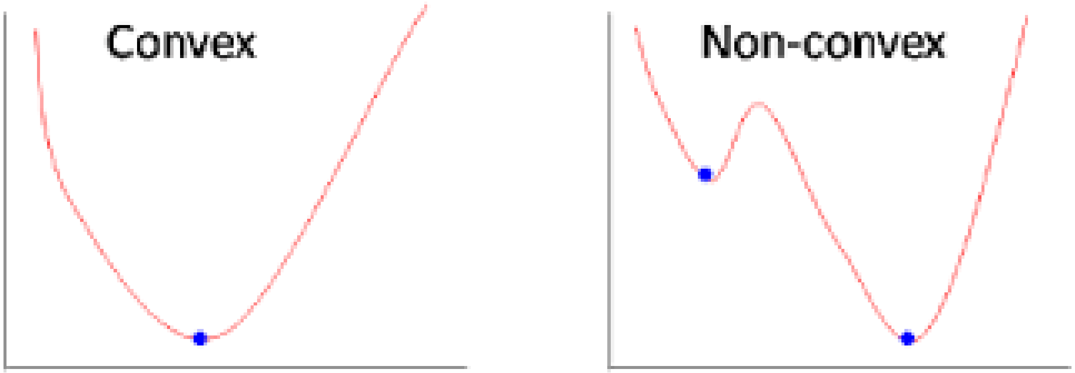


    - Polynomial as

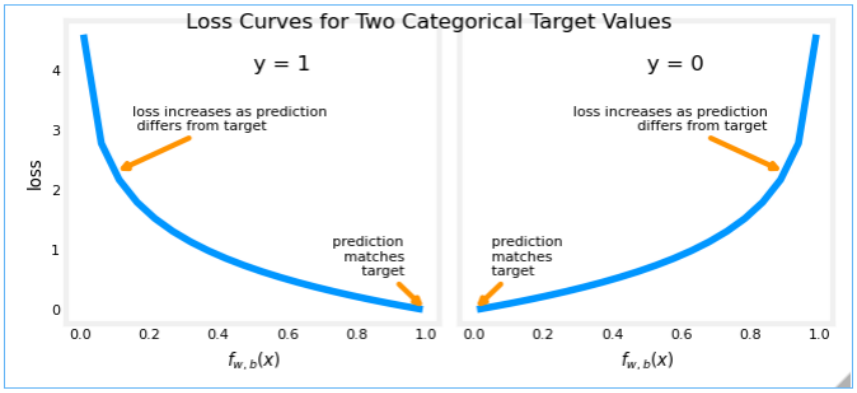

Non Linear Decision Boundary
X_2
y = 1
y = 1
y = 0
1
-1
1
X_1
y = 1
y = 1

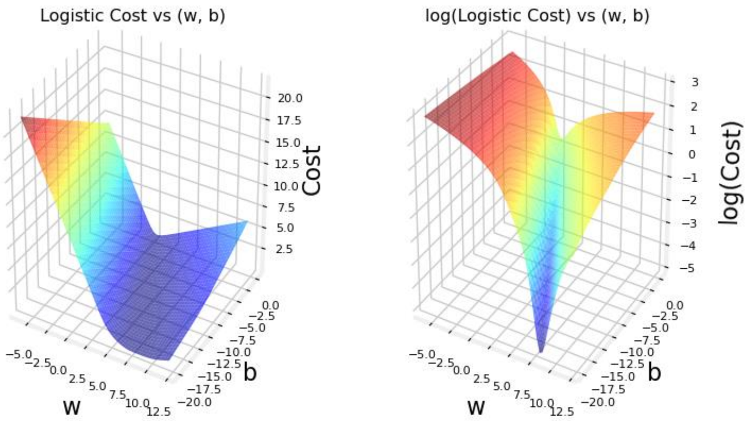
COST FUNCTION FOR LOGISTIC REGRESSION

- To Compute the cost Function for logistic regression, They Try to Compute it with Squared Error Function
  - The Convex Represents the Squared Error Function for Linear Regression
  - The Non-Convex Represents the Squared Error Function for Logistic Regression



  - So that they using another type of function to compute the loss of the algorithm
    - $L\left(f_{w,b}\left(x^i\right), y^i\right) = -\log\left(f\left(x^i\right)\right))$, if y = 1
    - $L\left(f_{w,b}\left(x^i\right), y^i\right) = -(1 - \log\left(f\left(x^i\right)\right))$, if y = 0

- 

- 

- $L\left(f_{w,b}\left(x^i\right), y^i\right) = -y^i * \log\left(f\left(x^i\right)\right) - \left(1 - y^i\right) * \left(1 - \log\left(f\left(x^i\right)\right)\right)$, This Loss Function Using for Compute the Loss for both y=1 or y =0

➢ Now We Can Compute the Cost Function as

- $j(w, b) = \frac{1}{m} \sum_{i=0}^{m} L\left(f_{w,b}\left(x^i\right), y^i\right)$

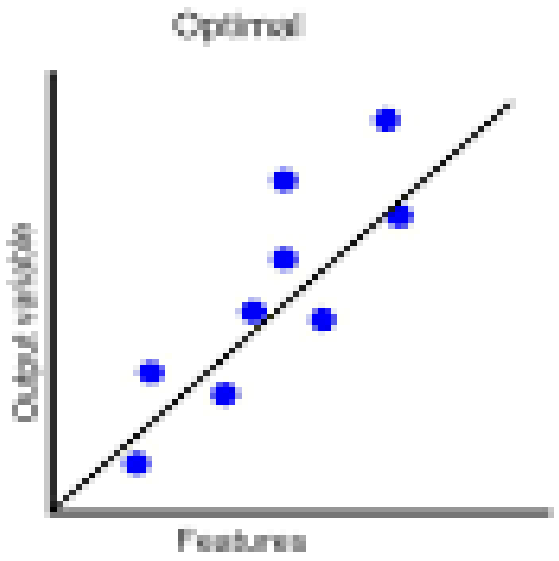
GRADIENT DESCENT ALGORITHM FOR LOGISTIC REGRESSION

- To Compute the gradient descent in logistic regression similar to in linear regression, but only have one different at the y-hat calculation function
  - Linear regression
    - $yhat = f(x) = w * x + b$
  - Logistic regression
    - $yhat = f(x) = \frac{1}{1 - e^{-(w*x+b)}}$
- W-temp = $w - \alpha * \left(\frac{d}{dw} J(w,b)\right)$
- B-temp = $b - \alpha * \left(\frac{d}{db} J(w,b)\right)$
- $\frac{d}{dw} J(w,b) = \frac{\sum_{I}^{M} (f(x)"i" - "y"i)^{\square} \ "x"i}{M}$
- $\frac{d}{db} J(w,b) = \frac{\sum_{I}^{M} (f(x)"i" - "y"i)^{\square}}{M}$

OVERFITTING & UNDERFITTING PROBLEM

What is Underfitting

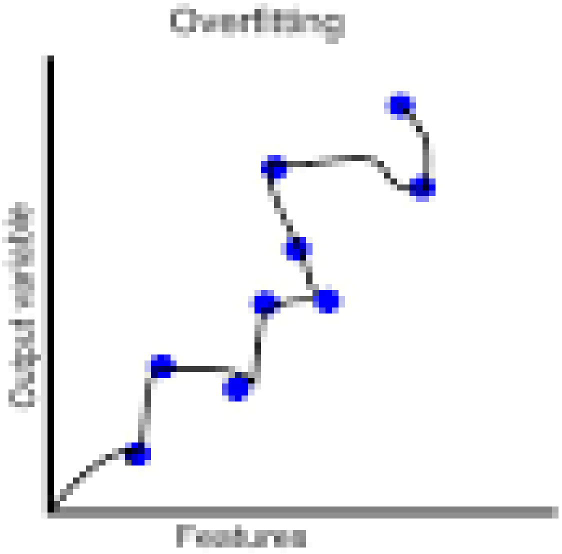
- It is issue show in learn models, the model is very general so that prediction leads to high losing ‘high bias’
- As in image if we will predict new value of the feature will be too away from the real value ‘Underfitting’



-

What is Overfitting

- It is issue show in learn models, the model is very special so that prediction leads to non-losing ‘high variance’, but the model predicts each new feature very away from it real value
- As in image if we will predict new value of the feature will be too away from the real value ‘Overfitting’

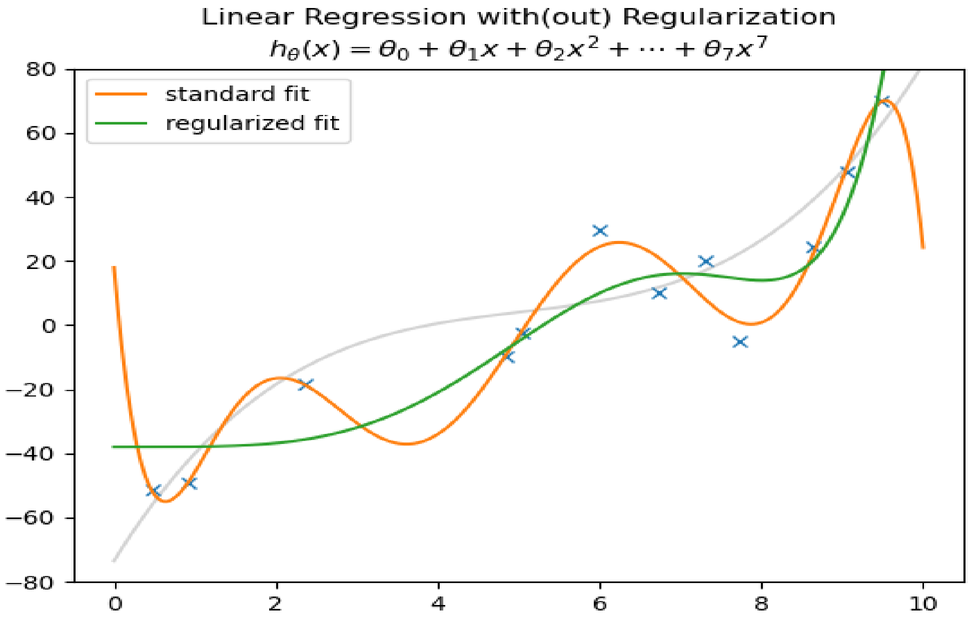


- To Solve Overfitting Problem
  - Increase number of training set

- Decrease Number of used Features ‘Will Apply it in next Course Notes’
- Using Regularization
  - Make all w’s values closed to 0

REGULARIZATION

- It is concept for solve Overfitting Problem
- It is term adding for j(w, b) function aims to avoided the model from overfitting problem
  - The term is $\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$
    - If λ is too large, will make underfitting problem
    - If λ is zero, won't make any changes at the model
    - $0 < \lambda < large\ value$



-

- Regularization with Gradient Descent
  - In Gradient Descent will add new terms in the derivation of j(w, b)
    - $\frac{d}{dw} j(w,b) = \frac{1}{m}\sum_{j=1}^{m}((f(x_j) - y_j) * x_j) + (\frac{\lambda}{m} * w_j)$

- Term $\frac{d}{db}j(w,b)$ won't have any changes so we have not b in the new regularization's term