Classification

(Introduction, Logistic Regression)

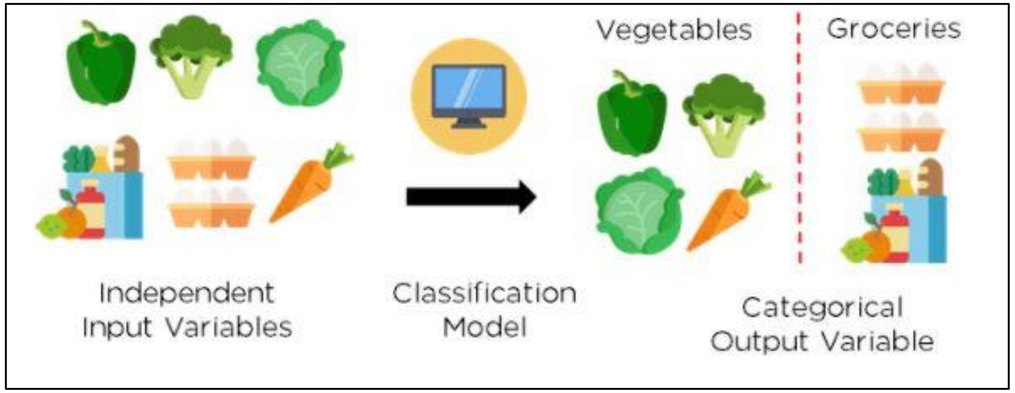
Dr. JASMEET SINGH

ASSISTANT PROFESSOR, CSED

TIET, PATIALA

Classification- Introduction

- Classification is a supervised learning technique that is used to identify the category of new observations on the basis of training data.
- In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups.
- Mathematically, classification analysis uses an algorithm to learn the mapping function from the input variables to the output variable (Y) i.e. **Y = f(x) where Y has discrete values**



Classification of vegetables and groceries

Types of Classification

1) **Binary Classification**
   - It is a type of classification problem in which the output variable has only binary values (True/False, 0/1, Yes/No)
   - Examples of Binary classification are classifying Email Spam Detection (spam/ham), Medical Testing (patient having disease or not), customer risk analysis (fraudulent/non-fraudulent)

2) **Multi-Class Classification**
   - It is a type of classification problem in which the output variable has more than two discrete values.
   - For example, risk evaluation of customers (low risk, medium risk, high risk), text classification into different categories (sports, politics, entertainment), etc.

Types of Classification (Contd…)

3) Multi-Label Classification

- It is a type of multi-class classification in which the examples can be labelled with multiple categories.
- For instance, in text classification a text may belong to Sports as well as politics category (Virat Kohli joined politics).

Why Regression models are not used for Classification ?

- Classification models are not useful for regression because of following reasons:

1) Regression models give continuous values of output variable and does not give probabilistic values.

2) Linear Regression models are insensitive to imbalance data.

Example for Point 2:

- Let’s say we create a perfectly balanced dataset , where it contains a list of customers and a label to determine if the customer had purchased or not.
- In the dataset, there are 20 customers.
- 10 customers age between 10 to 19 who purchased, and 10 customers age between 20 to 29 who did not purchase.

Why Regression models are not used for Classification ? (Contd….)

- According to Linear regression model, the line of best fit is shown in Fig. 1(a).
- To use this model for prediction is pretty straight forward.
- Given any age, we are able to predict the value along the Y-axis. If Y is greater than 0.5 (above the green line), predict that this customer will make purchases otherwise will not make purchases.

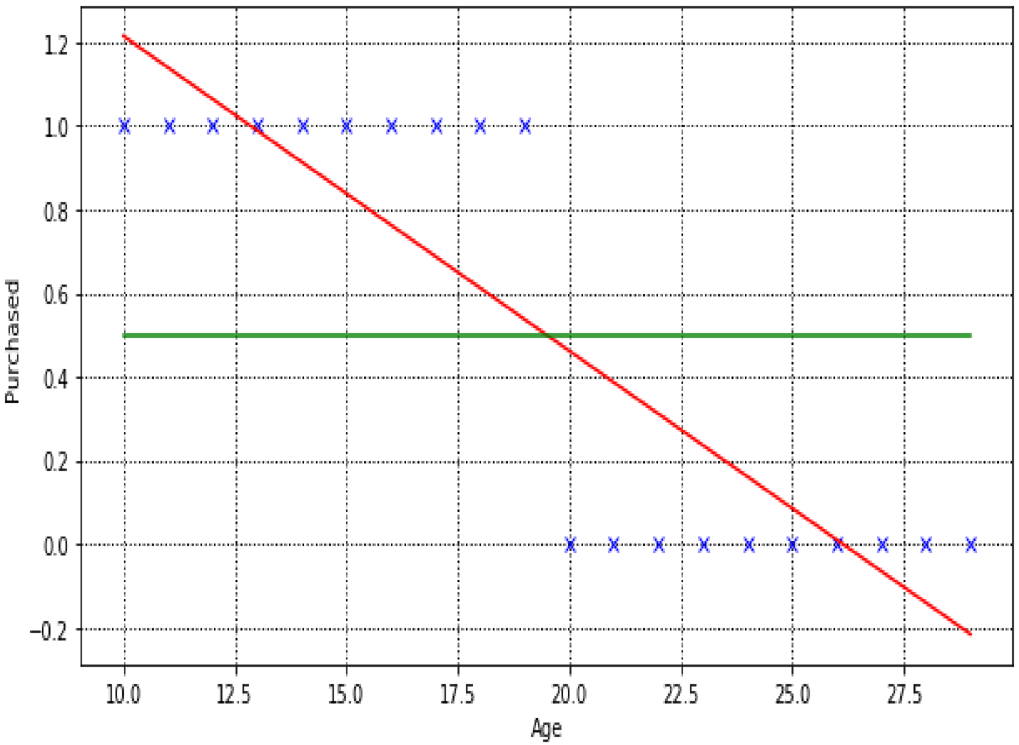


Figure 1(a)

Why Regression models are not used for Classification ? (Contd….)

- Let’s add 10 more customers age between 60 to 70, and train our linear regression model, finding the best fit line.

- Our linear regression model manages to fit a new line (Figure 1(b)), but if you look closer, some customers (age 20 to 22) outcome are predicted wrongly.

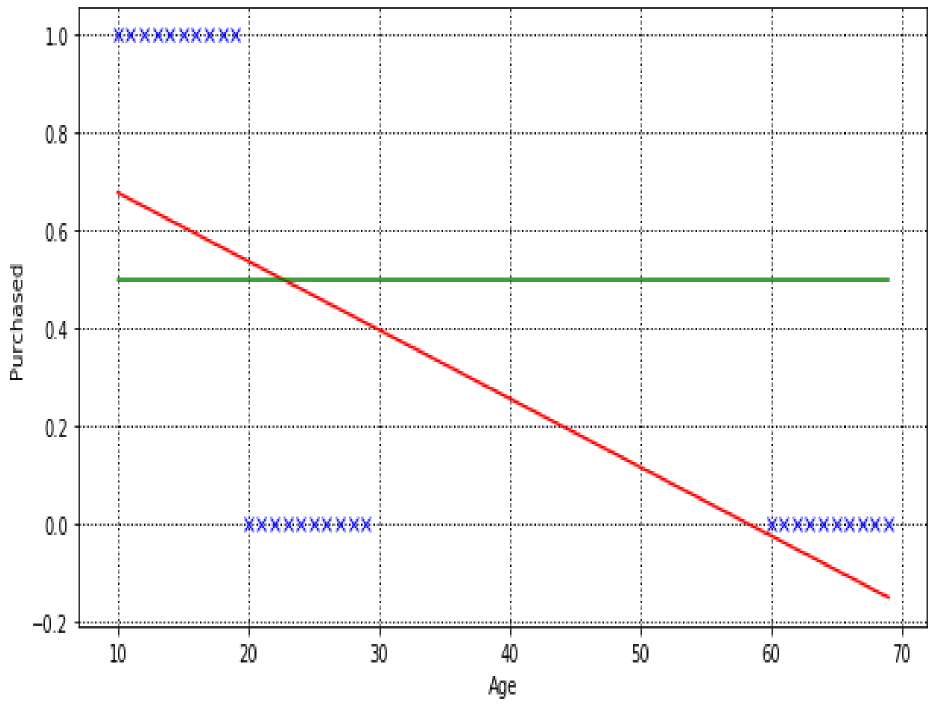


Figure 1(b)

Logistic Regression

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique.
- It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**
- Logistic regression uses the concept of predictive modeling as regression; therefore, it is called logistic regression

Logistic Regression-Hypothesis Function

- The hypothesis function that maps the given values of the input variable to the output variable is a sigmoid (logistic) function given by:

$$y^{\wedge} = f(x) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots\ldots\ldots\ldots\ldots+\beta_k x_k)}}$$

where, x_1, x_2, x_3…..x_k are k independent features on which the output variable depends and β_1, β_2, β_3…..β_k are coefficients of independent features

In other words, hypothesis function, is given by:

$$y^{\wedge} = f(x) = \frac{1}{1 + e^{-z}}$$

and z $= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \ldots \ldots \ldots \ldots .. + \beta_k x_k$

Hypothesis function- Characteristics

Hypothesis function is:

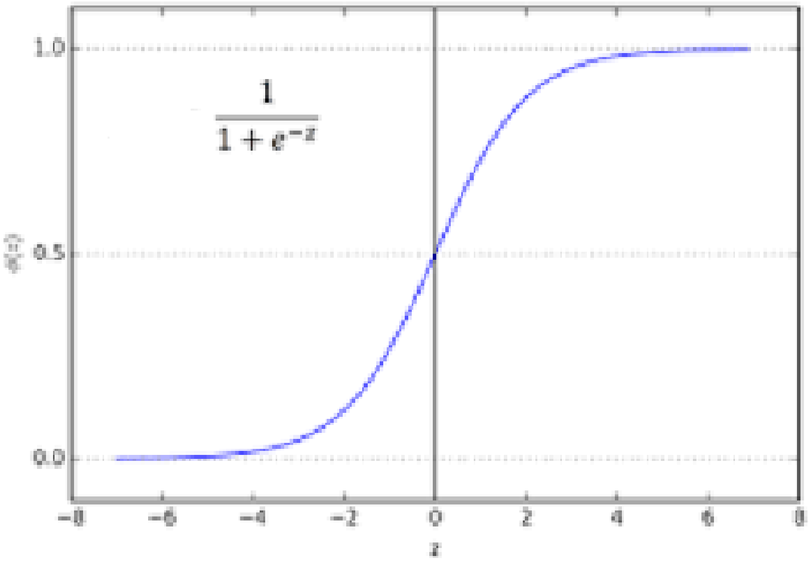
$$y^{\wedge} = f(x) = \frac{1}{1 + e^{-z}}$$

and z $= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \ldots \ldots \ldots \ldots . . +\beta_k x_k$

- If z=0; $y^{\wedge} = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-0}} = \frac{1}{1+1} = 0.5$
- If z=∞; $y^{\wedge} = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-\infty}} = \frac{1}{1+0} = 1$
- If z=-∞; $y^{\wedge} = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{\infty}} = \frac{1}{1+\infty} = 0$

Therefore, The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit.

So, it forms a curve like the "S" form.



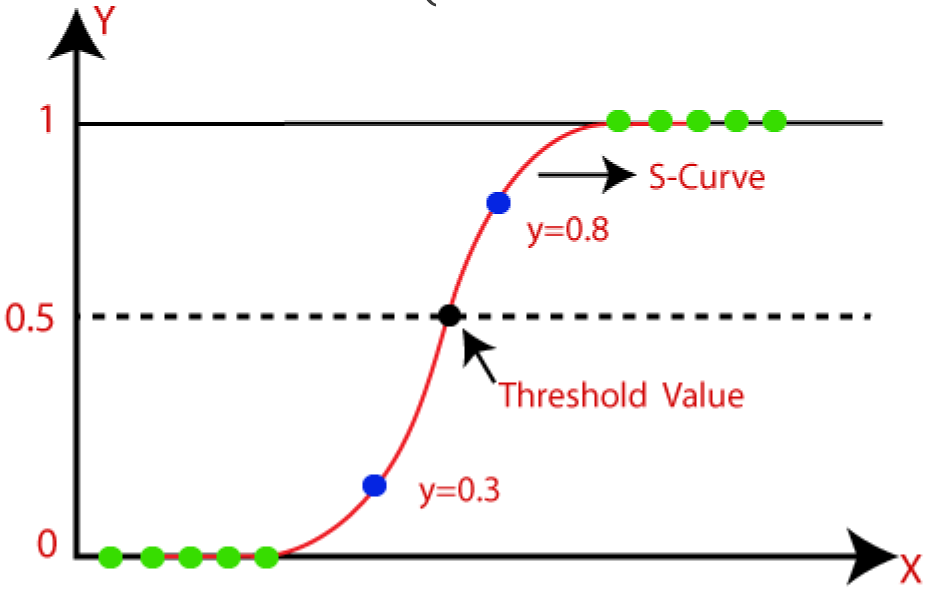
Interpretation of Hypothesis Function

- The sigmoid (logistic) hypothesis function gives value between 0 and 1.
- So, the output of sigmoid function is considered as the probability of label to be 1 given some value of input variables, i.e.,

$$y^{\wedge} = P(y = 1|x_1x_2\ x_3\ldots..\ x_k)$$

- Therefore, if the probability is greater than or equal to 0.5, we assign label 1, else we assign label 0.

$$Label = \begin{cases} 1\ if\ y^{\wedge} \geq 0.5 \\ 0\ if\ y^{\wedge} < 0.5 \end{cases}$$



Decision Boundary

- If $y^{\wedge} \geq 0.5 \;\Rightarrow label = 1$

This is possible iff ; z $\geq$ 0 (because if z $\geq$0 then $\frac{1}{1+e^{-z}} \geq 0.5$)

i.e., $\boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \boldsymbol{\beta_2 x_2} + \cdots \ldots \ldots \ldots \ldots .. + \boldsymbol{\beta_k x_k} \geq \mathbf{0}$

- If $y^{\wedge} < 0.5 \;\Rightarrow label = 0$

This is possible iff ; z < 0 (because if z < 0 then $\frac{1}{1+e^{-z}} < 0.5$)

i.e., $\boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \boldsymbol{\beta_2 x_2} + \cdots \ldots \ldots \ldots \ldots .. + \boldsymbol{\beta_k x_k} < \mathbf{0}$

Decision Boundary Contd…

- For example, in the figures shown below, depending upon the age (x_1) and length of hair(x_2) the person is classified as male (1) or female (0).
- So, the hypothesis function will be:

$$y^{\wedge} = f(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

- Now, examples are labelled as male (1)

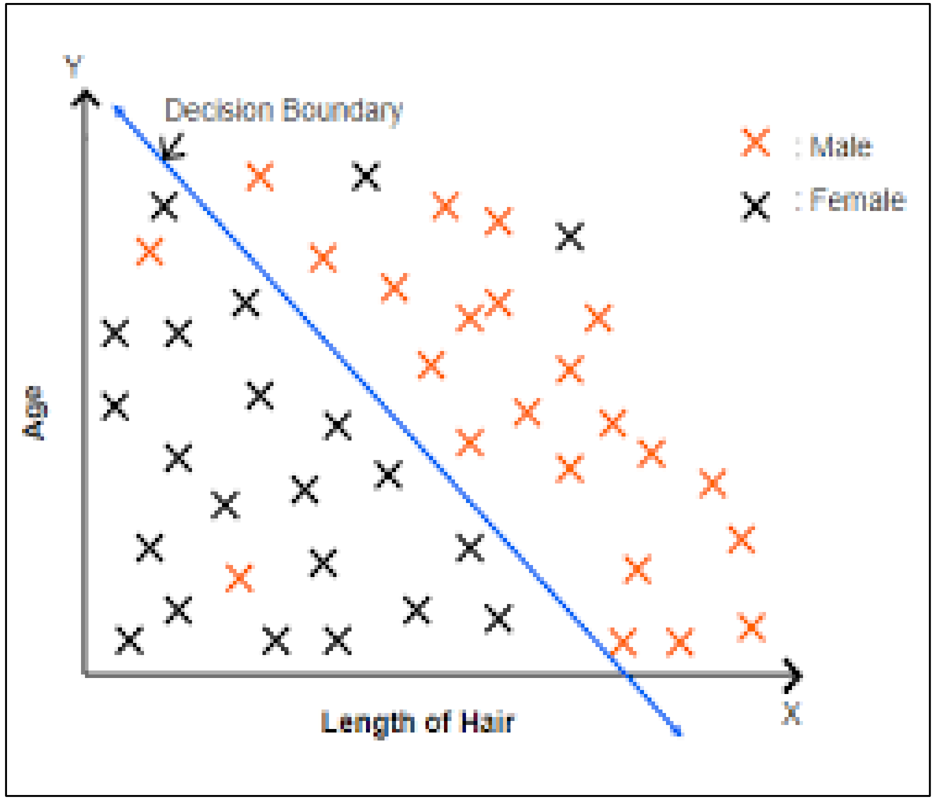
if $\beta_0 + \beta_1 x_1 + \beta_2 x_2 \geq 0$

i.e., all points above or on the line $\beta_1 x_1 + \beta_2 x_2$= - β_0

- Now, examples are labelled as ~~male~~ Female(0)

if $\beta_0 + \beta_1 x_1 + \beta_2 x_2 < 0$

i.e., all points below the line $\beta_1 x_1 + \beta_2 x_2$= - β_0

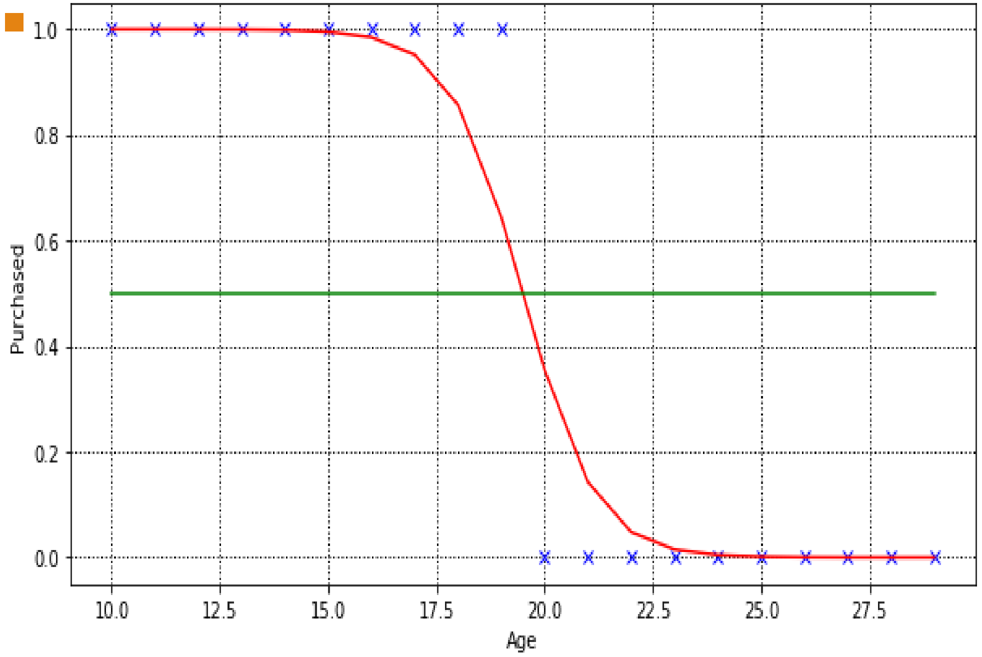


Decision Boundary Contd…

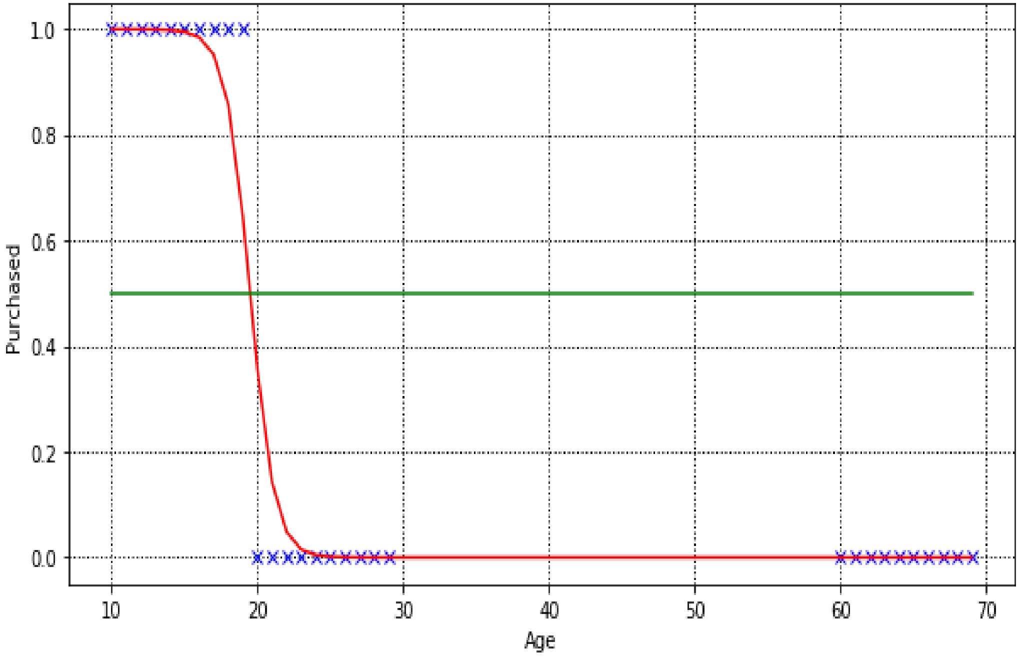
- The decision boundary may be of any shape linear or non linear (depending upon the z function we choose for the sigmoid function).
- The decision boundary is **insensitive to balanced or imbalanced data** and is characteristic of hypothesis function.
- For example, for the purchase labeling problem discussed in slides 6 and 7, the logistic regression will classify correctly in both the cases (as shown in figures in the next slide).

Decision Boundary Contd…

LOGISTIC REGRESSION MODEL FOR 20 CUSTOMERS (BALANCED DATA)



LOGISTIC REGRESSION MODEL FOR 30 CUSTOMERS (IMBALANCED DATA)



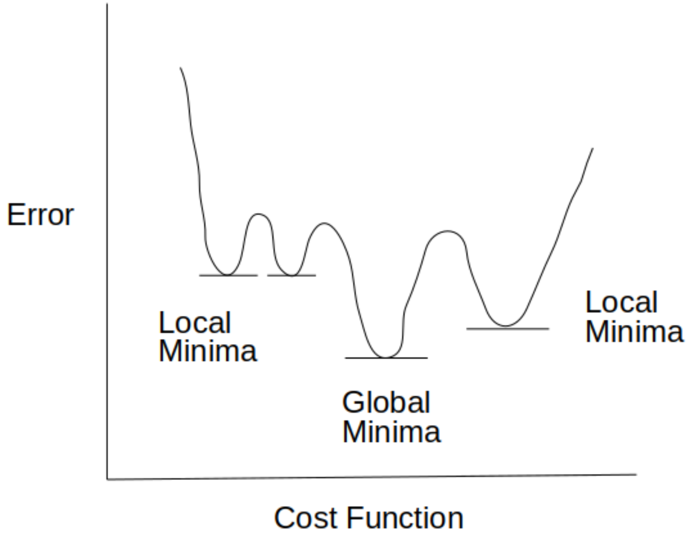
Logistic Regression- Cost Function

- Logistic regression uses the concept of predictive modeling as regression i.e., it find the optimal value of coefficients (β's) by minimizing the error/cost in labeling each training example.

- But in case of logistic regression, we **do not use mean square error (MSE)** cost function given by the equation below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \frac{1}{1+e^{-(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots\ldots\ldots\ldots+\beta_k x_{ik})}}\right)^2$$

- This is due to the reason, if we use mean square error cost function with logistic function, it provides non-convex outcome which results in many local minima. (as shown below)



Cost Function Contd…….

- Thus, for logistic regression, we use maximum likelihood cost function (cross entropy function) which is computed as follows for every labeled example:

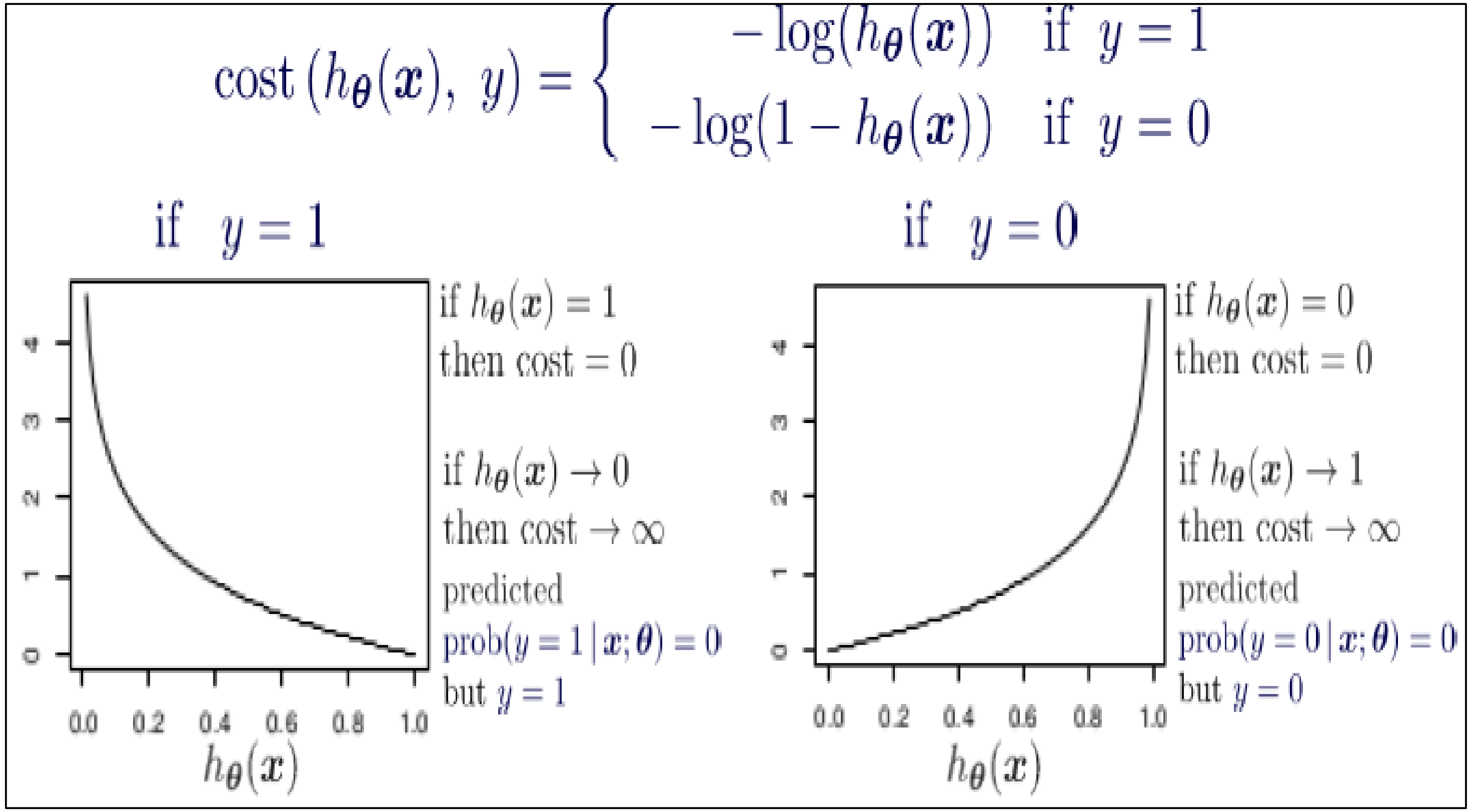
$$Cost\ or\ Error = \begin{cases} -\log\big(f(x)\big) & if\ y = 1 \\ -\log\big(1 - f(x)\big) & if\ y = 0 \end{cases}$$

where, y is the actual value of the training example and f(x) gives the corresponding predicted value given by the sigmoid function.

- The cross entropy cost function with logistic function gives convex curve with one local/global minima.
- It adds zero cost if the actual and the predicted values are same (i.e., both zero or both one) else, it adds some positive cost proportional to the difference between actual and predicted value.

(shown in figure in next slide)

Cost Function Contd…..



$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}),\ y) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } \ y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } \ y = 0 \end{cases}$$

if $y = 1$

if $h_\theta(x) = 1$
then cost $= 0$

if $h_\theta(x) \to 0$
then cost $\to \infty$

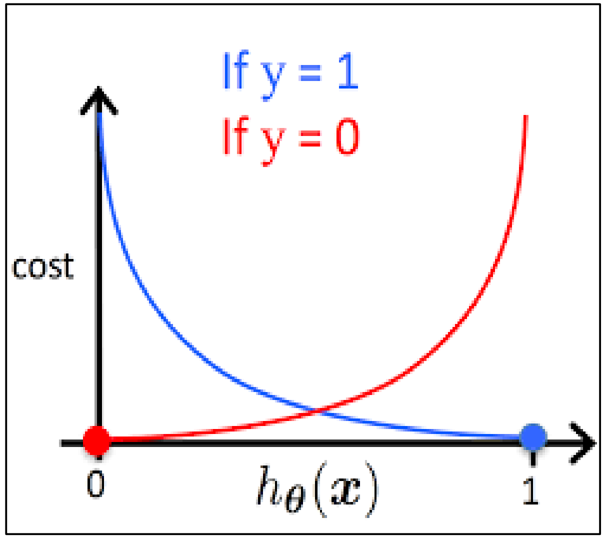
predicted
$\text{prob}(y = 1\,|\,x; \boldsymbol{\theta}) = 0$
but $y = 1$

if $y = 0$

if $h_\theta(x) = 0$
then cost $= 0$

if $h_\theta(x) \to 1$
then cost $\to \infty$

predicted
$\text{prob}(y = 0\,|\,x; \boldsymbol{\theta}) = 0$
but $y = 0$



Cost Function Contd.....

- The two separate equations for y=1 and y=0 can be combined in the single equation as follows:

$$Cost = -ylog(f(x)) - (1-y)\log(1-f(x))$$

When y=1 ; Cost $= -log(f(x)$ and when y=0 ; Cost $= -\log(1-f(x))$

- The total error for all the n training examples is thus computed as

$$J = -\frac{1}{n}\sum_{i=1}^{n} y_i log(f(x_i)) + (1-y_i)\log(1-f(x_i))$$

- This cost function is function of coefficients of input variables (β's) whose optimal values are computed using optimization techniques like gradient descent optimization.

Gradient Descent Optimization for Logistic Regression

- In logistic regression also, we use gradient descent optimization, for finding optimal values of β's by minimizing the total cost over the training examples.
- The gradient descent optimization considers gradient (slope/derivative) of the cost function.
- First lets find out the partial derivative of the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ w.r.t z

$$\frac{\partial f(x)}{\partial z} = -1 \times (1+e^{-x})^{-1-1} \frac{\partial(1+e^{-x})}{\partial z} = -(1+e^{-x})^{-2}\left(0 + e^{-x}\frac{\partial(-x)}{\partial z}\right)$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}\frac{\partial x}{\partial z} = \frac{1}{(1+e^{-x})} \times \frac{1+e^{-x}-1}{(1+e^{-x})}\frac{\partial x}{\partial z} = \frac{1}{(1+e^{-x})} \times \left(1 - \frac{1}{(1+e^{-x})}\right)\frac{\partial x}{\partial z}$$

$$= f(x)(1-f(x))\frac{\partial x}{\partial z}$$

Thus, partial derivative of sigmoid function f(x) w.r.t some variable z, is the product of f(x) and (1-f(x)) and derivative of power w.r.t z.

Gradient Descent Optimization for Logistic Regression (Contd….)

For logistic regression, cost function is given by:

$$J = -\frac{1}{n}\sum_{i=1}^{n} y_i log(f(x_i)) + (1-y_i)\log(1-f(x_i))$$

Where $\mathrm{f}(x_i) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots\ldots\ldots\ldots+\beta_k x_{ik})}}$

Gradient of cost function w.r.t any jth coefficient is given by:

$$\frac{\partial J}{\partial \beta_j} = -\frac{1}{n}\sum_{i=1}^{n} \frac{\partial y_i \log(f(x_i))}{\partial \beta_j} + \frac{\partial(1-y_i)\log(1-f(x_i))}{\partial \beta_j}$$

$$= -\frac{1}{n}\sum_{i=1}^{n} y_i \frac{\partial \log(f(x_i))}{\partial \beta_j} + (1-y_i)\frac{\partial \log(1-f(x_i))}{\partial \beta_j}$$

$$= -\frac{1}{n}\sum_{i=1}^{n} y_i \times \frac{1}{f(x_i)} \frac{\partial f(x_i)}{\partial \beta_j} + (1-y_i) \times \frac{1}{1-f(x_i)} \frac{\partial(1-f(x_i)}{\partial \beta_j}$$

$= -\frac{1}{n}\sum_{i=1}^{n} y_i \times \frac{1}{f(x_i)} \times f(x_i) \times (1-f(x_i) \times x_{ij} + (1-y_i) \times \frac{1}{1-f(x_i)} \times (0-f(x_i)(1-f(x_i)) \times x_{ij})$

(Using the derivative of sigmoid function computed in previous slide and derivative of power $\boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_{i1}} + \boldsymbol{\beta_2 x_{i2}} + \cdots \ldots + \boldsymbol{\beta_k x_{ik}}$ w.r.t β_j is the input variable values).

$= -\frac{1}{n}\sum_{i=1}^{n} y_i \times (1-f(x_i) \times x_{ij} - (1-y_i) \times f(x_i) \times x_{ij}$

$= -\frac{1}{n}\sum_{i=1}^{n} x_{ij}y_i - f(x_i)y_i x_{ij} - f(x_i)x_{ij} \ + y_i f(x_i)\, x_{ij}$

$$= \frac{1}{n}\sum_{i=1}^{n} (f(x_i) - y_i) \times x_{ij}$$

Gradient Descent Optimization for Logistic Regression (Contd….)

- Thus derivative of cost function w.r.t to β_j is same as in case of linear regression.
- The only difference is that in case of linear regression the hypothesis function is linear function of input variables whereas in logistic regression the hypothesis function is a sigmoid function of input variables.
- The gradient descent optimization for Logistic Regression is summarized as below:

1. Initialize β_0 =0 , $\beta_1 = 0$, $\beta_2 = 0$,……………………………. $\beta_k = 0$
2. Update parameters until convergence or for fixed number of iterations using following equation:

$$\beta_j \; = \beta_j - \frac{\alpha}{n}\sum_{i=1}^{n}\left(\frac{1}{1+e^{-(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots\ldots\ldots\ldots+\beta_k x_{ik})}} - y_i\right) \times x_{ij}$$

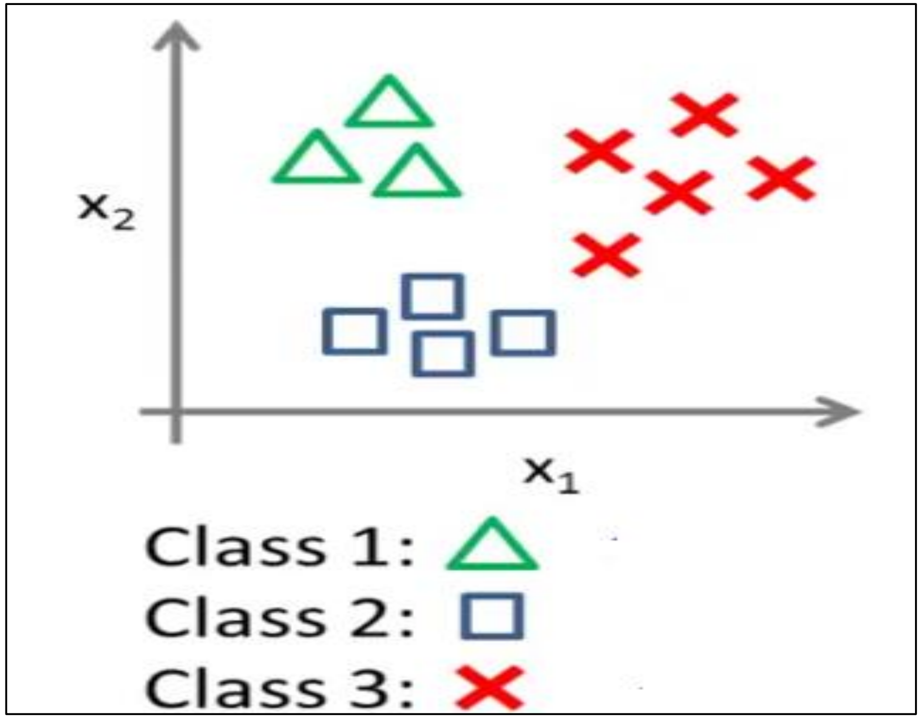
For j=0,1,2,3……………..k

Where x_{i0}=1 and k are the total number of iterations

Logistic Regression for Multi-Class Classification

- We will use a strategy called **one-vs.-all (one-vs.-rest) classification**, where **we train a binary classifier for each distinct class and choose the class that has the largest value returned by the sigmoid function.**

- For instance, consider a classification problem, in which there are two input variables on the basis of which the examples are classified into three classes (marked as triangles, crosses, and squares in the figure)



Logistic Regression for Multi-Class Classification (Contd.....)

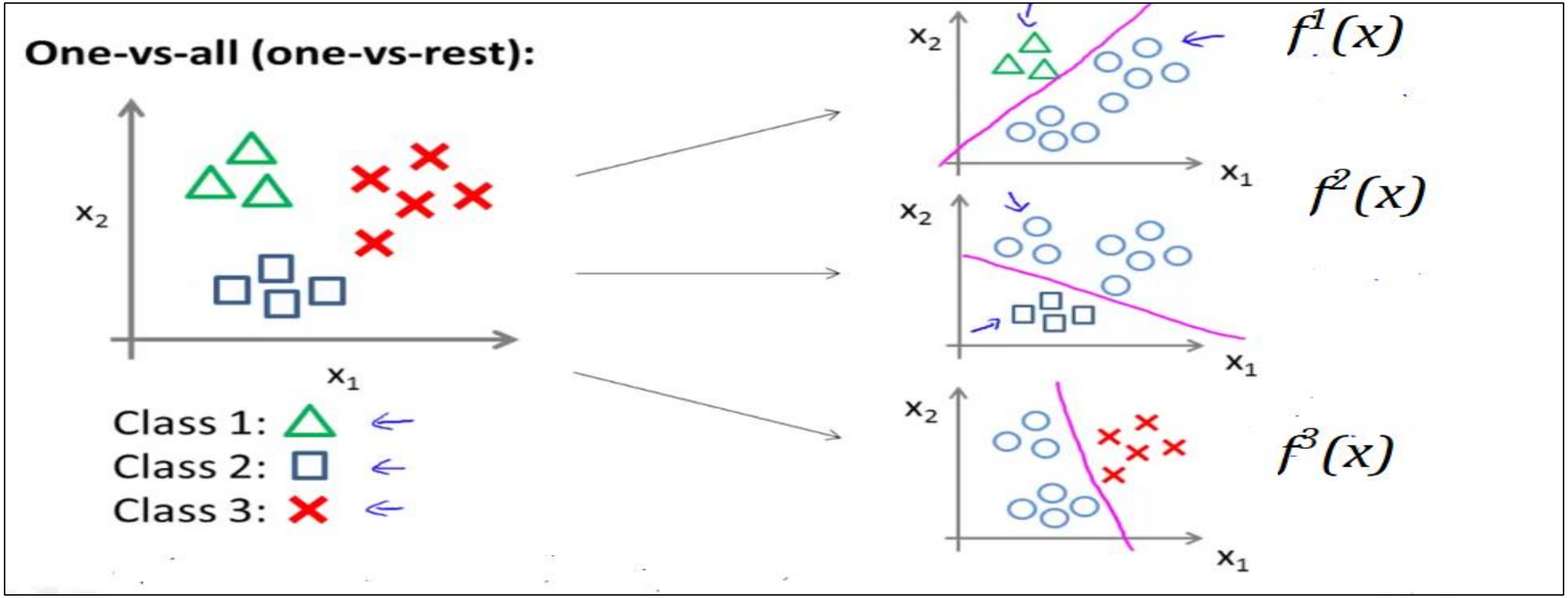
- For each binary classifier that we train, we will need to relabel the data such that the outputs for our class of interest is set to 1 and all other labels are set to 0.
- As an example, we have 3 groups A (0), B (1), and C (2) — we must make three binary classifiers:
  (1) A set to 1, B and C set to 0
  (2) B set to 1, A and C set to 0
  (3) C set to 1, A and B set to 0
- After training, choose the class that has the largest value returned by the sigmoid function for each test case (as shown in figure)

- *Each binary classifier will give probability of i^{th} label given the input feature values and choose that label for which probability is maximum.*

$$f^i(x) = P(y = i|x_1x_2)$$

$$i = \underset{i}{\operatorname{argmax}} f^i(x)$$

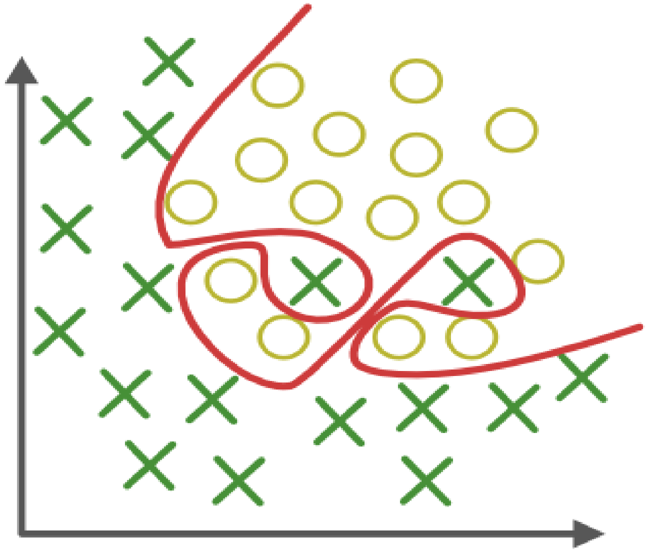
Logistic Regression for Multi-Class Classification (Contd.....)



Regularization for Logistic Regression

- Overfitting is also a problem of classification models, as we may fit a very complex decision boundary (lot of curves and angles) that considers each training examples but does not generalize well.

- The problem of overfitting can be handled using regularization that shrinks the coefficients of input variables thereby smoothen the decision boundary to generalize well.



Over-fitting

(forcefitting--too good to be true)

Ridge Regularization - Logistic Regression

- In ridge regression, we penalize the cost function by adding a factor which is proportional to sum of square of input coefficients.

$$J = -\frac{1}{n}\sum_{i=1}^{n} y_i \log\big(f(x_i)\big) + (1 - y_i)\log\big(1 - f(x_i)\big) + \frac{1}{2n}\lambda\sum_{j=0}^{k}\beta_j^2$$

*2 has been divided in the penalty factor just for simplification in gradient descent optimization (it will not effect minimization process.

$$\frac{\partial J}{\partial \beta_j} = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i) \times x_{ij} + \frac{\lambda}{n} \times \beta_j$$

(The derivative of cross entropy function has been derived in slide 21; and derivative of penalty factor is 2 β_j)

Ridge Regularization - Logistic Regression

Therefore β_j is updated as :

$$\beta_j = \beta_j - \alpha \times \left(\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i) \times x_{ij} + \frac{\lambda}{n} \times \beta_j\right)$$

$$\beta_j = \beta_j\left(1 - \frac{\alpha\lambda}{n}\right) - \frac{\alpha}{n} \times \left(\sum_{i=1}^{n}(f(x_i) - y_i) \times x_{ij}\right)$$

This is same as Ridge Regression for Linear Regression. But the hypothesis function is a sigmoid function given as:

$$\mathrm{f}(x_i) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots\ldots\ldots\ldots+\beta_k x_{ik})}}$$

The factor $\left(1 - \frac{\alpha\lambda}{n}\right)$ is less than 1. So the coefficients will be shrinked according to the value of this factor.