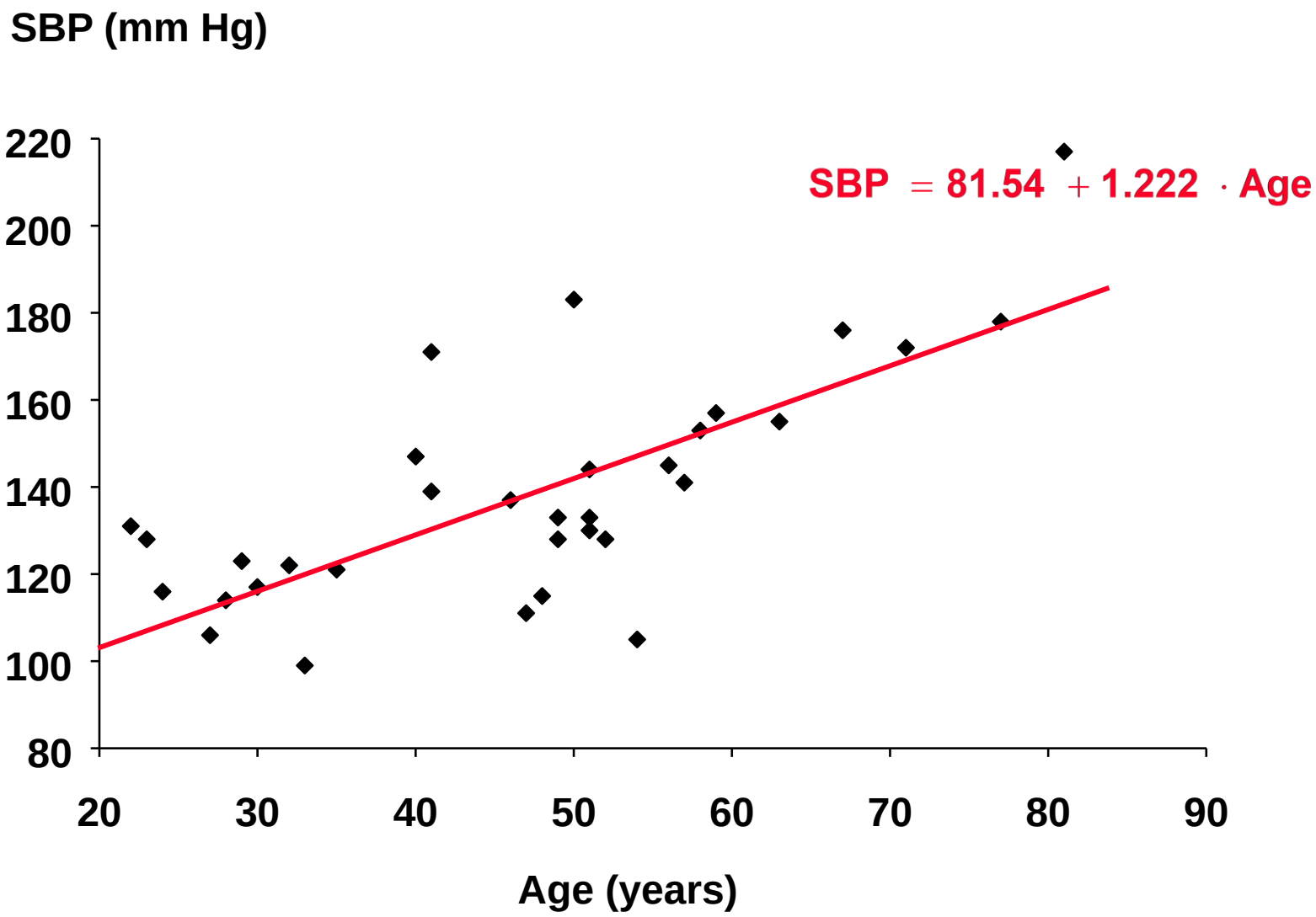
Logistic Regression

Dr. Dipak Kumar Mohanty

Simple linear regression

Table 1 Age and systolic blood pressure (SBP) among 33 adult women

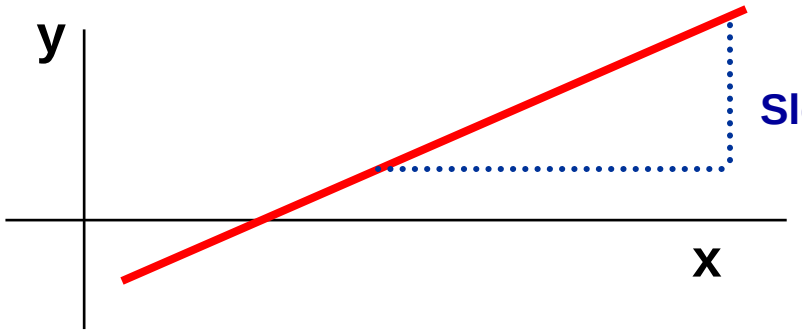
Age	SBP	Age	SBP	Age	SBP
22	131	41	139	52	128
23	128	41	171	54	105
24	116	46	137	56	145
27	106	47	111	57	141
28	114	48	115	58	153
29	123	49	133	59	157
30	117	49	128	63	155
32	122	50	183	67	176
33	99	51	130	71	172
35	121	51	133	77	178
40	147	51	144	81	217



adapted from Colton T. Statistics in Medicine. Boston: Little Brown, 1974

Simple linear regression

- **Relation between 2 continuous variables (SBP and age)**



$$y = \alpha + \beta_1 x_1$$

- **Regression coefficient β_1**
  - **Measures association between y and x**
  - **Amount by which y changes on average when x changes by one unit**
  - **Least squares method**

Multiple linear regression

- **Relation between a continuous variable and a set of i continuous variables**

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i$$

- **Partial regression coefficients β_i**
  - **Amount by which y changes on average when x_i changes by one unit and all the other x_is remain constant**
  - **Measures association between x_i and y adjusted for all other x_i**

- **Example**
  - **SBP *versus* age, weight, height, etc**

Multiple linear regression

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i$$

Predicted
Response variable
Outcome variable
Dependent

Predictor variables
Explanatory variables
Covariables
Independent variables

Categorical Response Variables

- **Whether or not a person smokes**
- **Success of a medical treatment**

- **Binary Response**

$$Y = \begin{cases} \text{Non} - \text{smoker} \\ \text{Smoker} \end{cases}$$

$$Y = \begin{cases} \text{Survives} \\ \text{Dies} \end{cases}$$

- **Opinion poll responses**

- **Ordinal Response**

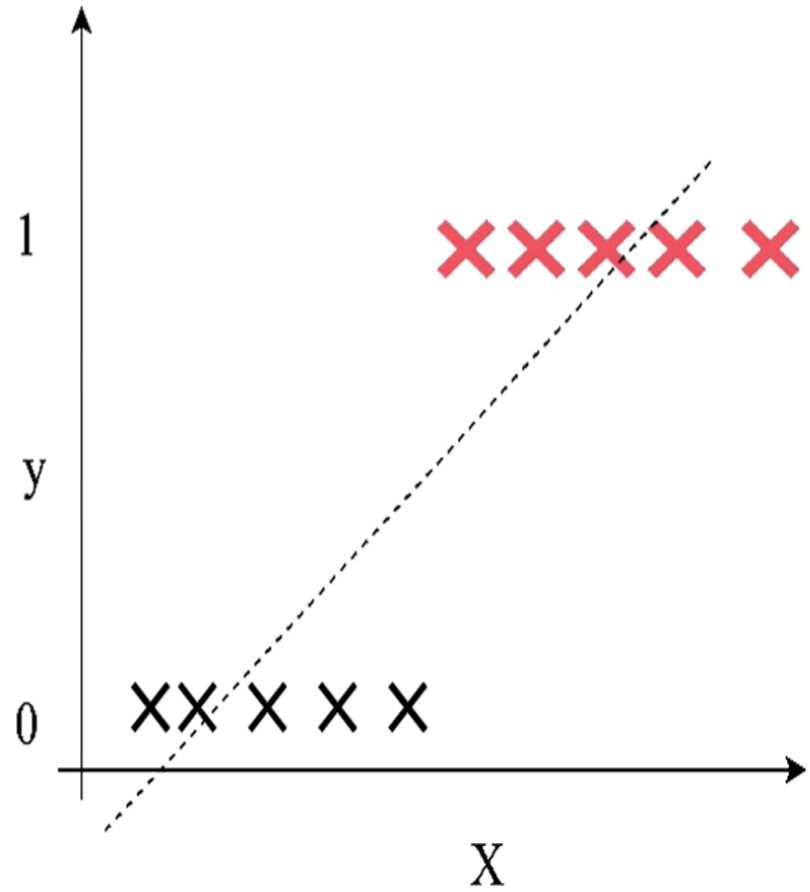
$$Y = \begin{cases} \text{Agree} \\ \text{Neutral} \\ \text{Disagree} \end{cases}$$

Why not Linear Regression for Classification?

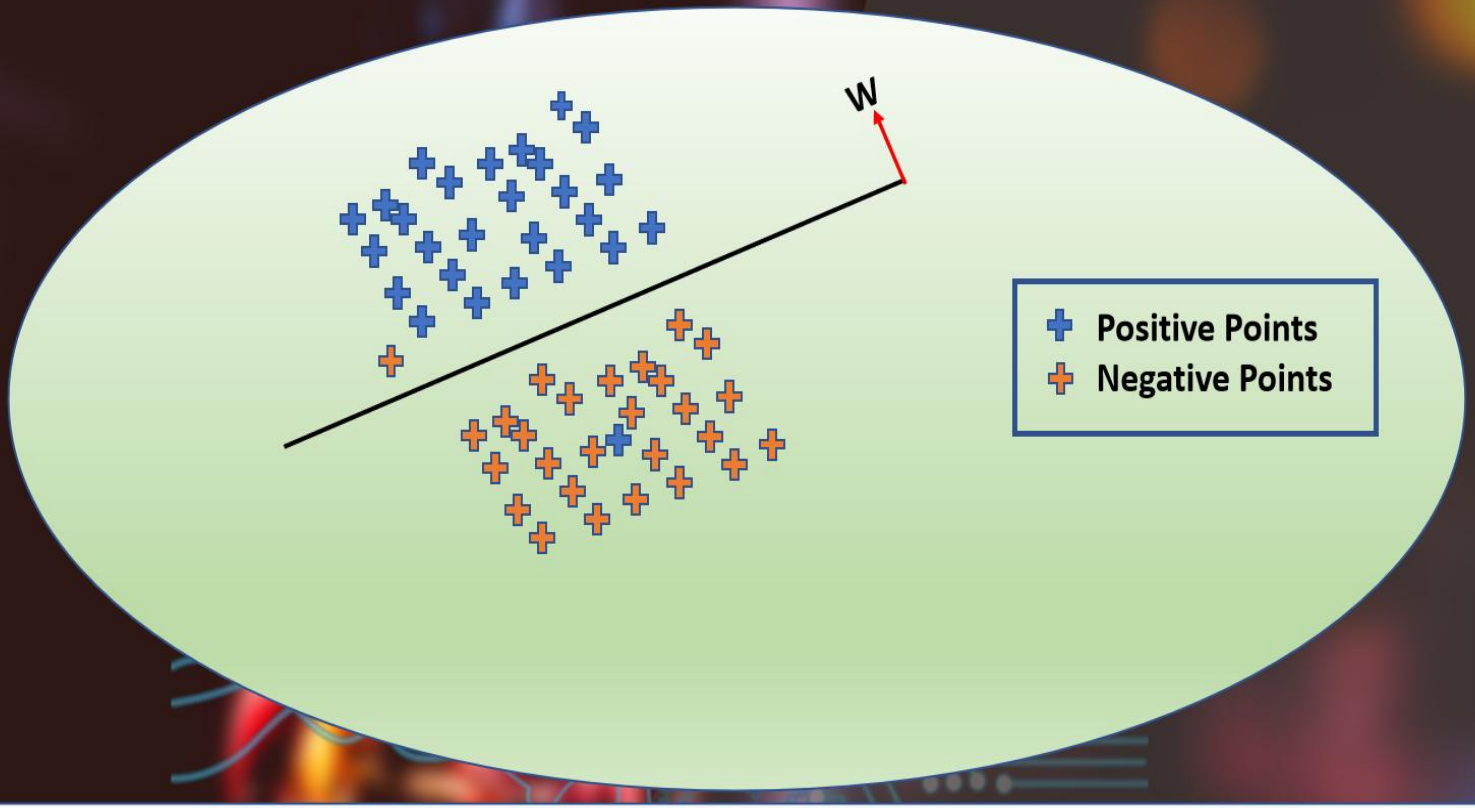
- Multi class classification: Numerically encoding classes looses meaning for comparison.

- Binary classification: Range of Dependent variable may lie outside [0,1].

$$y \epsilon [0, 1]$$



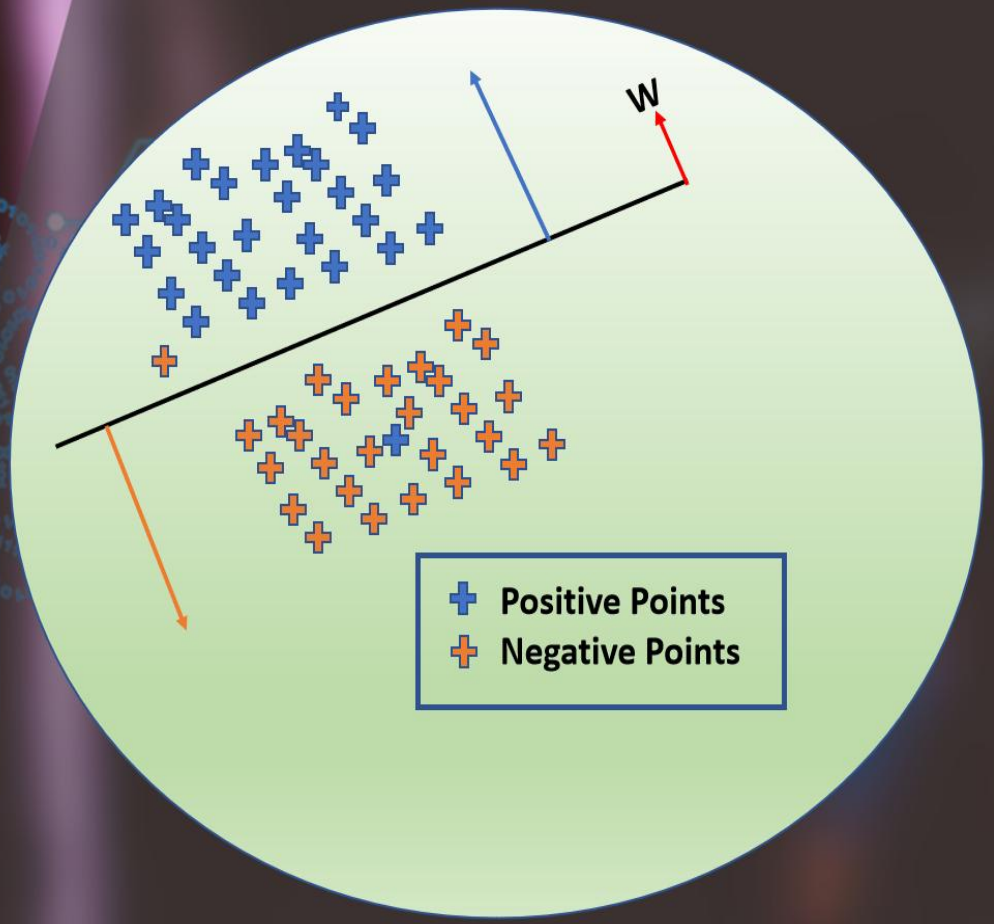

Visual Intuition of Logistic Regression



- Assumptions: Classes are linearly/almost linearly separated
- Type: Classification (Specifically Binary classification, but can be used for Multi-Class also)
- Decision Surface: Separates classes by creating Hyper-Plane ($\boldsymbol{W^T.X + b = 0}$)
- Training Task: Find the best Hyper plane by finding $\boldsymbol{W}$ & $\boldsymbol{b}$

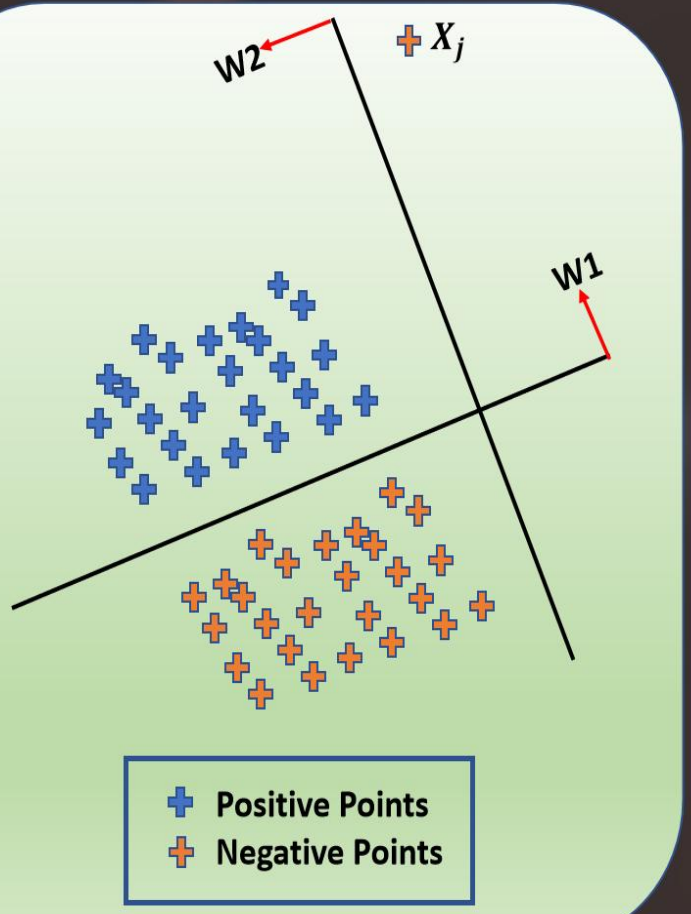
Find best hyper-plane

- $D_{all} = \{X_i, y_i\} \forall\ i \epsilon\ 1, 2, .., n\}$
- y_i = +1 ∀ positive points & -1 ∀ negative points
- For all positive points, $W^T.X_i \geq 0$ and for all negative points $W^T.X_i < 0$
- Correct Prediction: $y_i.W^T.X_i \geq 0$
- Incorrect Prediction: $y_i.W^T.X_i < 0$
- $$W^* = argmax_w \left(\sum_{i=1}^{n} y_i.W^T.X_i \right)$$
- **All the above points are ignoring bias term (b)**



Sigmoid Squashing

- $$W^* = argmax_w \left(\sum_{i=1}^{n} y_i . W^T . X_i \right)$$
- The above optimization equation is affected by outliers.
- X_j is a outlier so the distance from plane1 (W1) will be much than other distances and eventually the best hyper-plane will be shifted to somewhere near W2
- Scenario(plane W1):
  - Total Positive points: 100 (average distance 1 unit from plane)
  - Total Negative points: 100 (average distance 1 unit from plane)
  - A single outlier point (X_j) → Negative point (distance from plane is 1000 units)
  - $\sum_{i=1}^{100} y_i . W_1 . X_i$ (for +ve points) = 100
  - $\sum_{i=1}^{100} y_i . W_1 . X_i$ (for -ve points) = 100
  - $y_i . W_1 . X_j$ = -1000 (as X_j is on the wrong side)
  - $$\sum_{i=1}^{n} y_i . W_1 . X_i = 100 + 100 - 1000 = -800$$
- Scenario(plane W2):
  - $$\sum_{i=1}^{n} y_i . W_2 . X_i = 200 - 200 + 1 = 1$$



Logistic Regression

- Supervised learning method for classification.
- "logit" = "log odds"

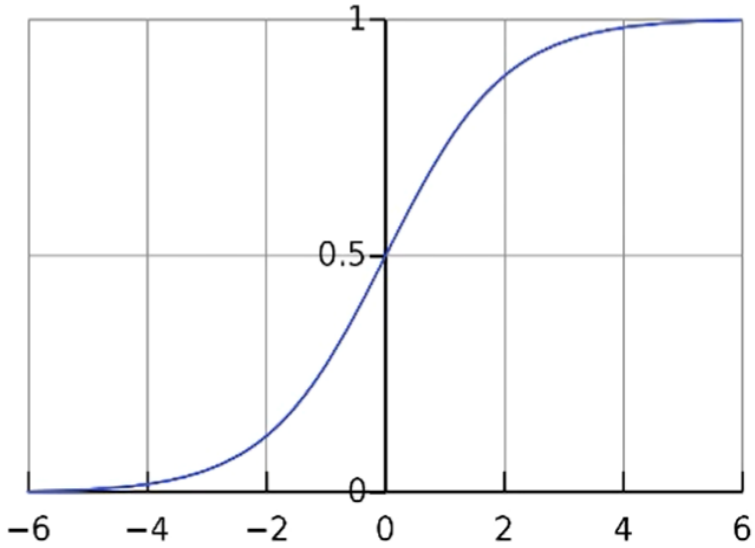
$$odds = \frac{P(event)}{1 - P(event)}$$

$$X \in \boldsymbol{R}$$

$$p(X) \in [0, 1]$$

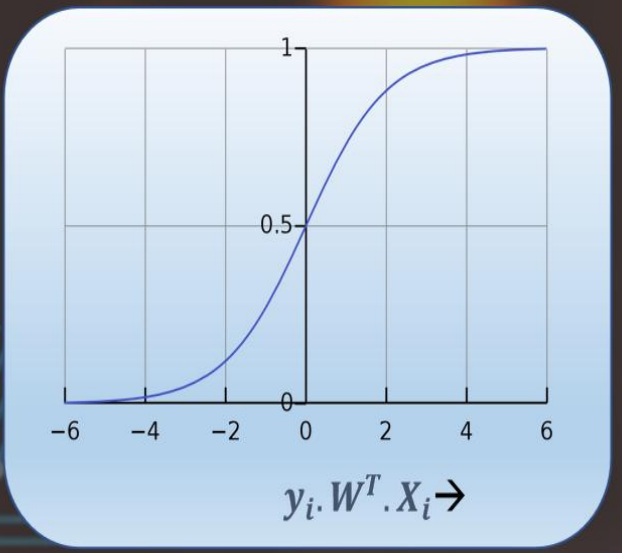
- Let $\Pr(y = 1 | X) = p(X)$
- Sigmoid Function: $p(X) = \frac{1}{1 + e^{-\beta X}}$

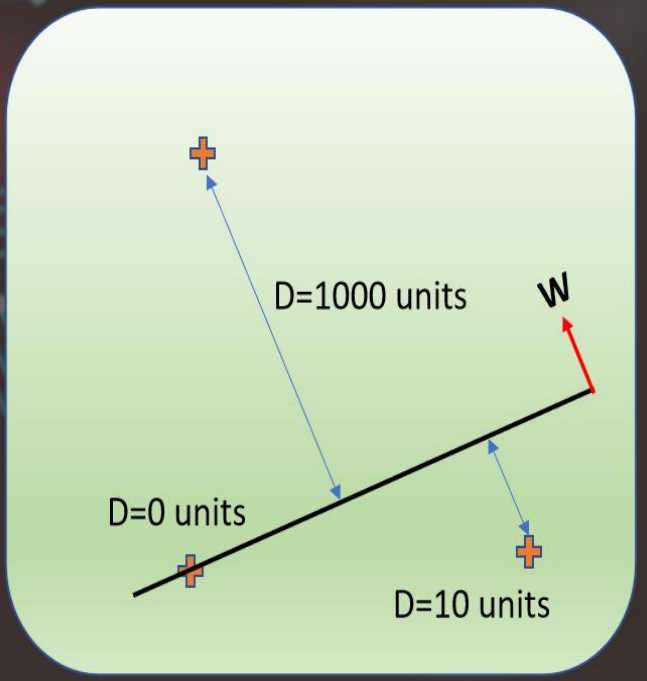
$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta X$$



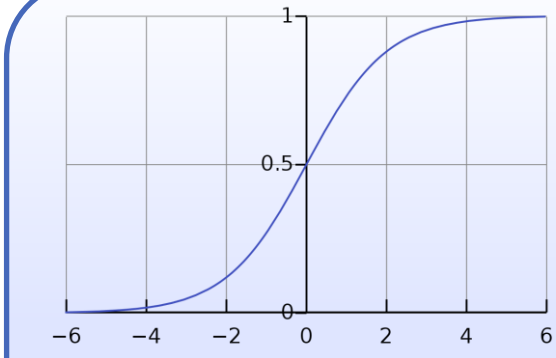
Sigmoid Function

- $\sigma(x) = \frac{1}{1+e^{-x}}$ & $\lim_{x \to \pm\infty} \sigma(x) = 1, 0$
- $\sigma(0) = 0.5$
- *Modified optimization equation:*
- $W^* = argmax_w \left(\sum_{i=1}^{n} \sigma(y_i . W^T . X_i) \right)$
- *Sigmoid gives probabilistic interpretations by concentrating the range between 0 to 1*
- *If any point is exactly on the plane, then the $W^T . X_i$=0 so $\sigma(y_i . W^T . X_i)$ = 0.5 i.e. 50% probability that the point could be positive or negative*

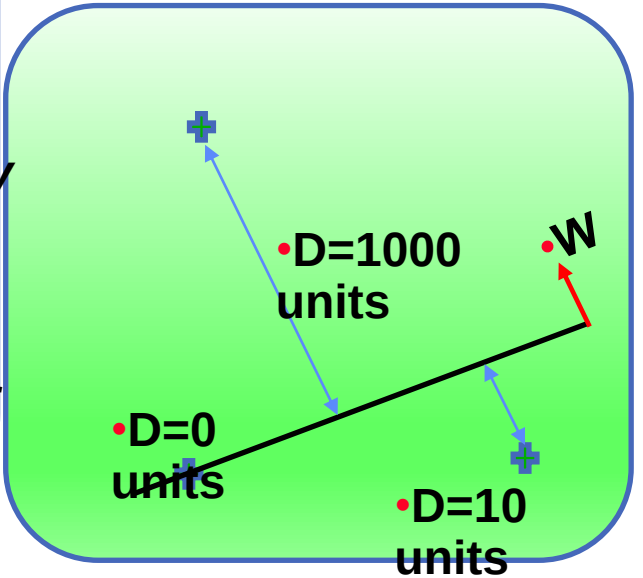




- $\sigma(x) = \frac{1}{1+e^{-x}}$ & $\lim_{x \to \pm\infty} \sigma(x) = 1, 0$
- $\sigma(0) = 0.5$
- ***Modified optimization equation:***
- $W^* = argmax_w\left(\sum_{i=1}^{n} \sigma(y_i . W^T . X_i)\right)$
- ***Sigmoid gives probabilistic interpretations by concentrating the range between 0 to 1***
- ***If any point is exactly on the plane, then the $W^T . X_i$=0 so $\sigma(y_i . W^T . X_i)$ = 0.5 i.e. 50% probability that the point could be positive or negative***



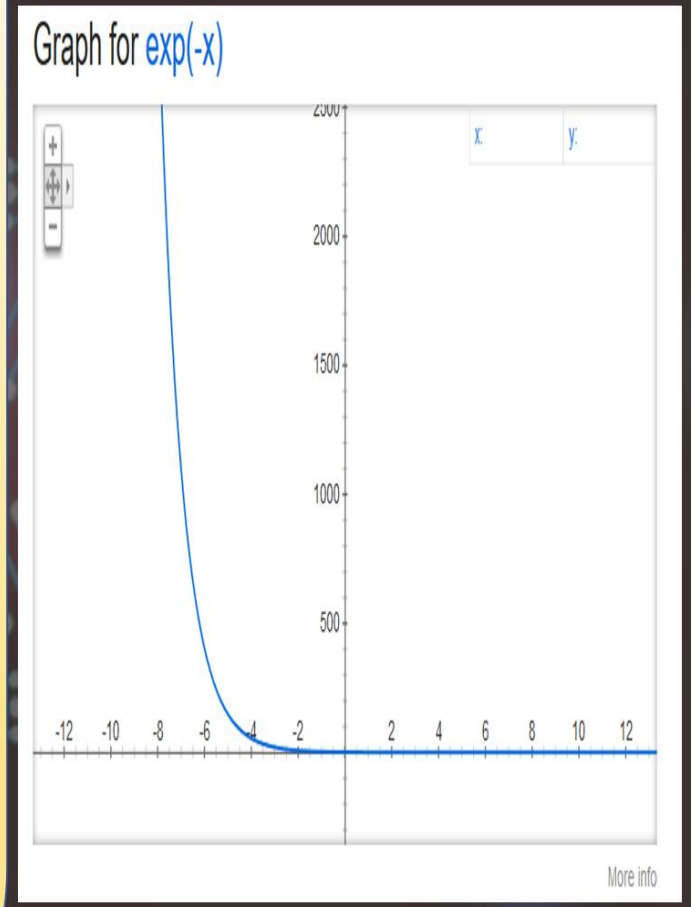
- $y_i . W^T . X_i$



- $W^* = argmax_w\left(\sum_{i=1}^{n} \sigma(y_i . W^T . X_i)\right)$
- $W^* = argmax_w\left(\sum_{i=1}^{n} \log \sigma(y_i . W^T . X_i)\right)$
- $W^* = argmax_w\left(\sum_{i=1}^{n} \log \frac{1}{1+e^{-y_i . W^T . X_i}}\right)$
- $W^* = argmax_w - \left(\sum_{i=1}^{n} \log(1 + e^{-y_i . W^T . X_i})\right)$
- $W^* = argmin_w \left(\sum_{i=1}^{n} \log(1 + e^{-y_i . W^T . X_i})\right)$

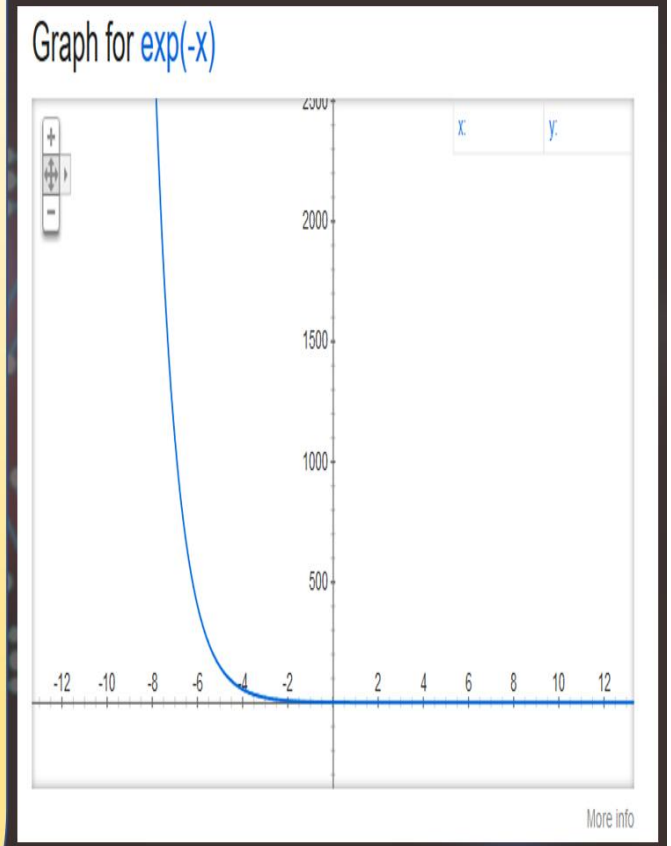
L2 - Regularization

- $$W^* = argmin_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \right)$$
- $$e^{-y_i . W^T . X_i} \geq 0$$
- $$\log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- $$\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- So the above summation will have a minimum value of 'Zero'
- The above equation will be 'Zero' when $e^{-y_i . W^T . X_i} = 0$
- $y_i . W^T . X_i \to \infty \therefore W \to \infty$
- W will always try to be infinite based on the Training data, eventually make the model overfit.
- $$W^* = argmin_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) + \lambda \sum_{j=1}^{D} W_j^2 \right)$$
- The first term will try to increase W while the second term try to decrease W maintaining the balance.
- $\lambda \rightarrow$ Hyper-Parameter
- $\lambda = 0$ then no Regularization term, hence overfitting.
- $\lambda \to \infty$ then there will be no effect of the first term, so no effect of the training features, hence underfitting.
- We have to tune λ to get the perfect balance



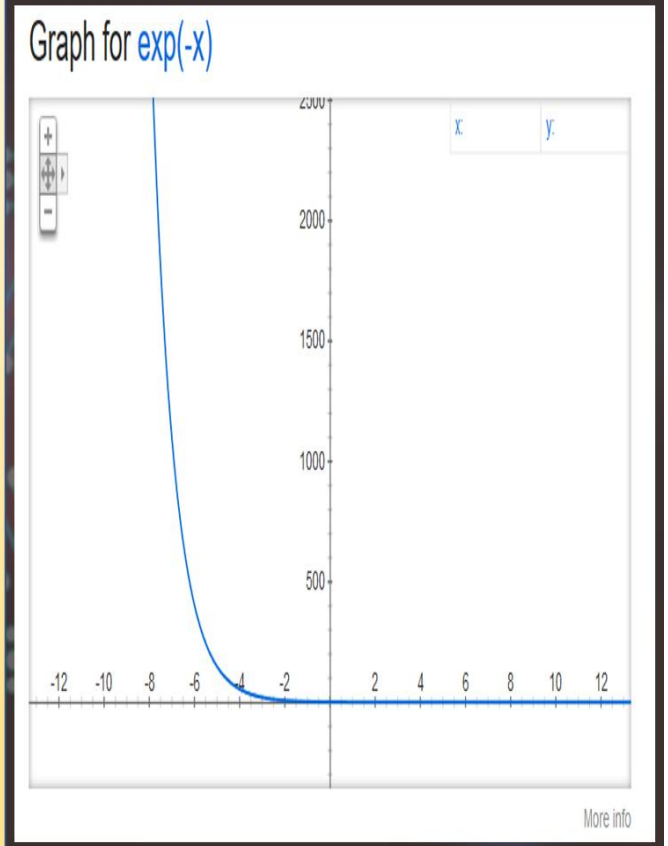
L2 - Regularization

- $$W^* = \boldsymbol{argmin}_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \right)$$
- $$e^{-y_i . W^T . X_i} \geq 0$$
- $$\log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- $$\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- So the above summation will have a minimum value of 'Zero'
- The above equation will be 'Zero' when $e^{-y_i . W^T . X_i} = 0$
- $y_i . W^T . X_i \to \infty \therefore W \to \infty$
- W will always try to be infinite based on the Training data, eventually make the model overfit.
- $$W^* = \boldsymbol{argmin}_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) + \lambda \sum_{j=1}^{D} W_j^2 \right)$$
- The first term will try to increase W while the second term try to decrease W maintaining the balance.
- $\lambda \rightarrow$ Hyper-Parameter
- $\lambda = 0$ then no Regularization term, hence overfitting.
- $\lambda \to \infty$ then there will be no effect of the first term, so no effect of the training features, hence underfitting.
- We have to tune λ to get the perfect balance



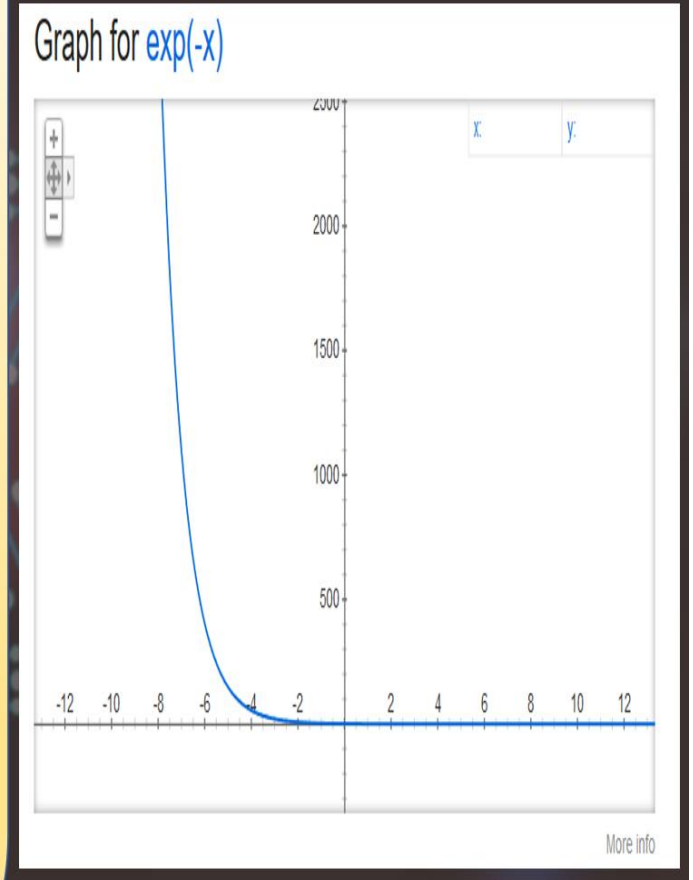
L1 - Regularization

- $$W^* = \boldsymbol{argmin}_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \right)$$
- $$e^{-y_i . W^T . X_i} \geq 0$$
- $$\log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- $$\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- So the above summation will have a minimum value of 'Zero'
- The above equation will be 'Zero' when $e^{-y_i . W^T . X_i} = 0$
- $y_i . W^T . X_i \to \infty \therefore W \to \infty$
- W will always try to be infinite based on the Training data, eventually make the model overfit.
- $$W^* = \boldsymbol{argmin}_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) + \lambda \sum_{j=1}^{D} |W_j| \right)$$
- The first term will try to increase W while the second term try to decrease W maintaining the balance.
- $\lambda \rightarrow$ Hyper-Parameter
- $\lambda = 0$ then no Regularization term, hence overfitting.
- $\lambda \to \infty$ then there will be no effect of the first term, so no effect of the training features, hence underfitting.
- We have to tune λ to get the perfect balance
- **L1-Reg creases sparsity by setting 'Zero' value to some weight values corresponding to less-important features.**



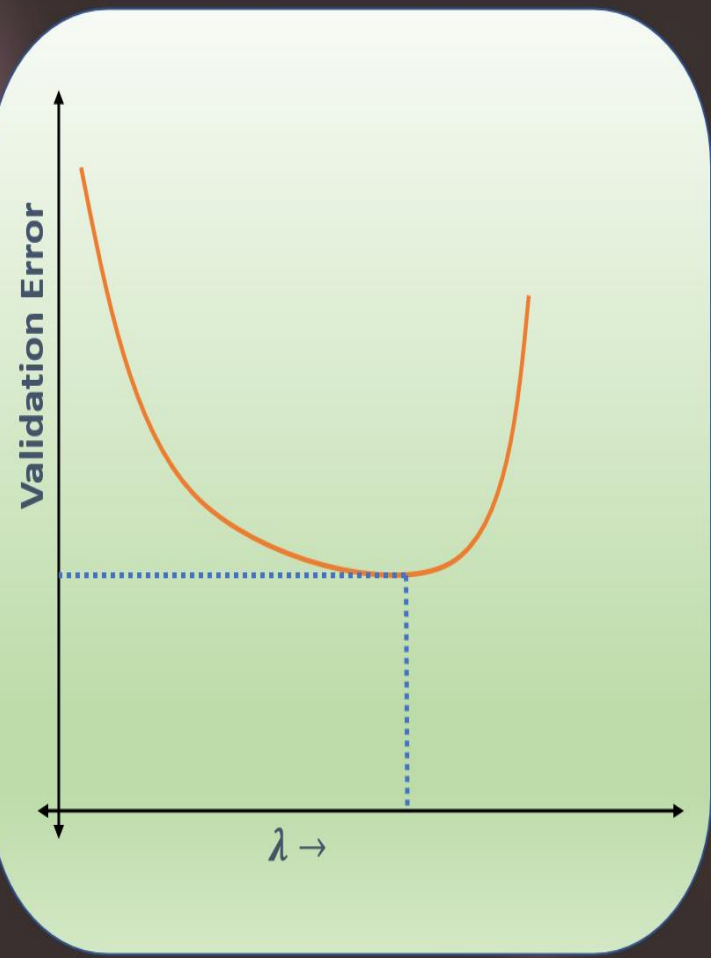
Elastic Net - Regularization

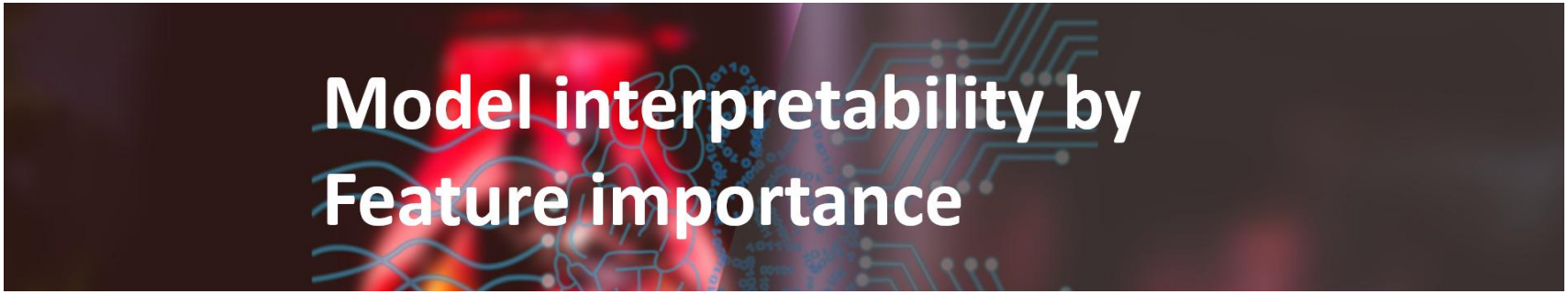
- $$W^* = argmin_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \right)$$
- $$e^{-y_i . W^T . X_i} \geq 0$$
- $$\log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- $$\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) \geq 0$$
- So the above summation will have a minimum value of 'Zero'
- The above equation will be 'Zero' when $e^{-y_i . W^T . X_i} = 0$
- $y_i . W^T . X_i \to \infty \therefore W \to \infty$
- W will always try to be infinite based on the Training data, eventually make the model overfit.
- $$W^* = argmin_w \left(\sum_{i=1}^{n} \log\left(1 + e^{-y_i . W^T . X_i}\right) + \lambda_1 \sum_{j=1}^{D} W_j^2 + \lambda_2 \sum_{j=1}^{D} |W_j| \right)$$
- The first term will try to increase W while the second term try to decrease W maintaining the balance.
- $\lambda_1, \lambda_2 \rightarrow$ Hyper-Parameters




Hyper Parameter Tuning in Logistic
Regression

- $$W^* = argmin_w \left(\sum_{i=1}^{n} \log(1 + e^{-y_i . W^T . X_i}) + \lambda \sum_{j=1}^{D} W_j^2 \right)$$
- $\lambda \rightarrow \text{Hyper} - \text{Parameter}$
- $\lambda = [10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2} \ldots]$
- Train the LR model will different values of λ and log the validation error.
- Plot the curve and select the value of best λ for which the validation error is maximum
- **Drawback of Grid-search**
  - For multiple hyper-parameters, the model will be trained for the total number of combinations
  - $\lambda_1 = [10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1} \ldots]$
  - $\lambda_2 = [10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1} \ldots]$
  - Model will be trained for: length(λ_1) x length(λ_2) times
- **Random-search**
  - It will pick hyper-parameters value randomly and that value will be used for plotting the curve.
  - Random-search is almost as powerful as grid-search with less time complexity sometimes.




Model interpretability by
Feature importance

f_a	1	2	3	4	5	6	7
f_b	2	4	6	8	10	12	14
f_c	2	5	6	1	8	13	67

Collinearity

- If two features have a linear relationship like: $f_i = \alpha_0 + \alpha_1 f_j$ then f_i & f_j are collinear.
- In the above example: $f_b = 2f_a$ so f_a &f_b are collinear.
- Collinear features in a dataset can mislead to find the feature importance.
- Example:
  - For the above dataset we got W^* as [3,5,7] So as per this we can say that f_c is the most important feature and f_a is the least important.
  - Now $W^{*T}.X = 3f_a + 5f_b + 7f_c$ and $f_b = 2f_a$ so,
  - $$W^{*T}.X = 3f_a + 10f_a + 7f_c = 13f_a + 0f_b + 7f_c$$
  - New $W^* = [13, 0, 7]$ now according to this value f_a is the most important feature and f_b is the least important.
- Optimal weight vector W^* might change arbitrarily causing wrong feature importance.

f_a	1	2	3	4	5	6	7
f_b	2	4	6	8	10	12	14
f_c	2	5	6	1	8	13	67

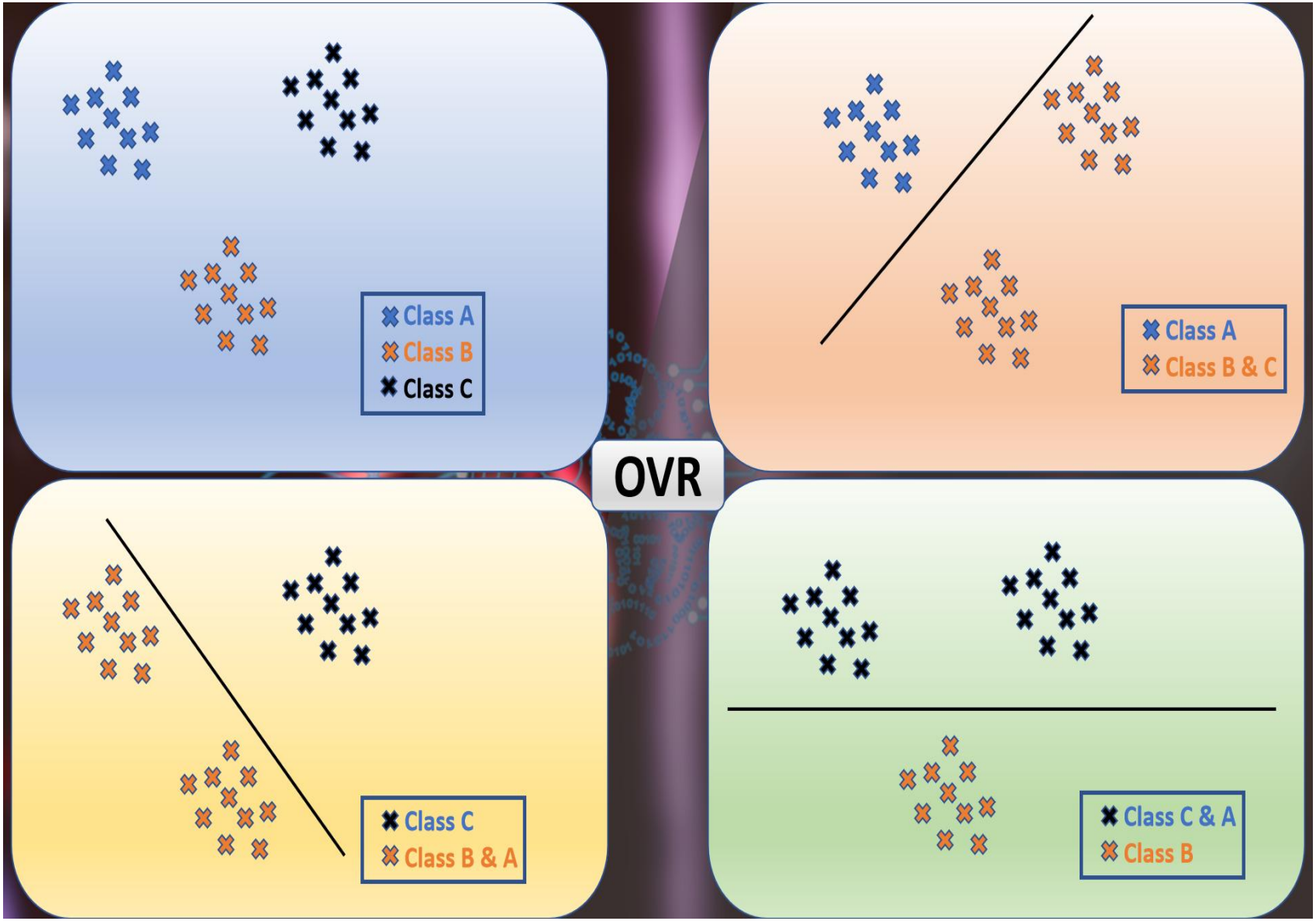
Collinearity

- If two features have a linear relationship like: $f_i = \alpha_0 + \alpha_1 f_j$ then f_i & f_j are collinear.
- In the above example: $f_b = 2f_a$ so f_a &f_b are collinear.
- Collinear features in a dataset can mislead to find the feature importance.
- Example:
  - For the above dataset we got W^* as [3,5,7] So as per this we can say that f_c is the most important feature and f_a is the least important.
  - Now $W^{*T}.X = 3f_a + 5f_b + 7f_c$ and $f_b = 2f_a$ so,
  - $$W^{*T}.X = 3f_a + 10f_a + 7f_c = 13f_a + 0f_b + 7f_c$$
  - New $W^* = [13, 0, 7]$ now according to this value f_a is the most important feature and f_b is the least important.
- Optimal weight vector W^* might change arbitrarily causing wrong feature importance.

Time and Space Complexity in Logistic Regression

- Space Complexity: O(D) where D is the number of dimensions of feature vectors.
  - We have to only store the optimal weight vector in memory which is D-dimensional
- Time Complexity: O(D) where D is the number of dimensions of feature vectors.
  - We are only computing $W^T . X_q$, So we are only computing D multiplications and D-1 additions (typically)
- If D is small:
  - We can use Logistic Regression in any production system which requires low latency.
- If D is large:
  - We can use L_1 regularization to create spare weight vector, which will eventually reduce the time complexity


Multi class Classification
using Logistic Regression


Class A
Class B
Class C
Class A
Class B & C
OVR
Class C
Class B & A
Class C & A
Class B

Logistic Regression

- Supervised learning method for classification.
- "logit" = "log odds"

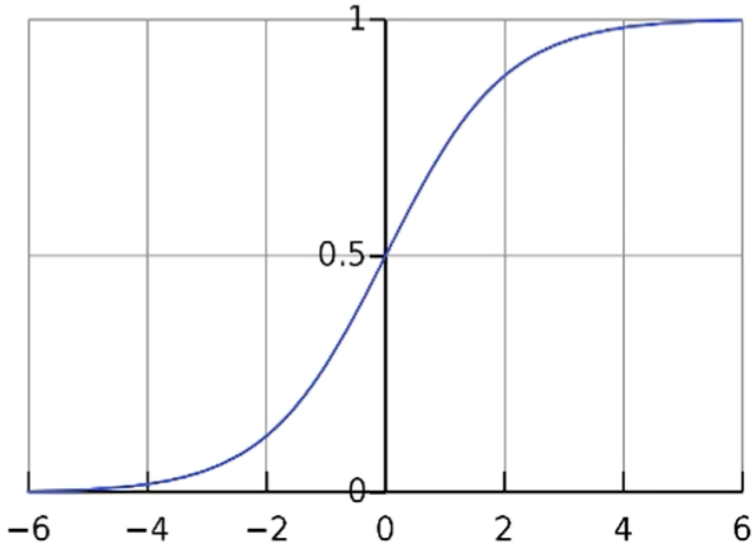
$$odds = \frac{P(event)}{1 - P(event)}$$

$$X \in \boldsymbol{R}$$

$$p(X) \in [0, 1]$$

- Let $\Pr(y = 1 | X) = p(X)$
- Sigmoid Function: $p(X) = \frac{1}{1 + e^{-\beta X}}$

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta X$$



Parameter Estimation

- Goal of learning is to estimate parameter vector $\widehat{\beta}$

- Logistic Regression uses Maximum Likelihood for parameter estimation. How does this work?

  - Consider N samples with labels either 0 or 1

  - For samples labelled "1": Estimate $\widehat{\beta}$ such that $\widehat{p(X)}$ is as close to 1 as possible

  - For samples labelled "0": Estimate $\widehat{\beta}$ such that $1 - \widehat{p(X)}$ is as close to 1 as possible

$$\prod_{s\ in\ yi=1} p(x_i) \qquad \prod_{s\ in\ yi=0} (1 - p(x_i))$$

Parameter Estimation

$$L(\beta) = \prod_{s\ in\ yi=1} p(x_i) \times \prod_{s\ in\ yi=0} (1 - p(x_i))$$

$$L(\beta) = \prod_{s}^{n} p(x_i)^{y_i} \times \left(1 - p(x_i)\right)^{1-y_i}$$

$$l(\beta) = \sum_{i=1}^{n} y_i \log\left(p(x_i)\right) + (1 - y_i) \log\left(1 - p(x_i)\right)$$

$$l(\beta) = \sum_{i=1}^{n} y_i \log\left(\frac{1}{1 + e^{-\beta x_i}}\right) + (1 - y_i) \log\left(\frac{e^{-\beta x_i}}{1 + e^{-\beta x_i}}\right)$$

$$l(\beta) = \sum_{i=1}^{n} y_i \left[\log\left(\frac{1}{1 + e^{-\beta x_i}}\right) - \log\left(\frac{e^{-\beta x_i}}{1 + e^{-\beta x_i}}\right)\right] + \log\left(\frac{e^{-\beta x_i}}{1 + e^{-\beta x_i}}\right)$$

$$l(\beta) = \sum_{i=1}^{n} y_i \left[\log\left(e^{\beta x_i}\right)\right] + \log\left(\frac{e^{-\beta x_i}}{1 + e^{-\beta x_i}} \times \frac{e^{\beta x_i}}{e^{\beta x_i}}\right)$$

$$l(\beta) = \sum_{i=1}^{n} y_i \beta x_i + \log\left(\frac{1}{1 + e^{\beta x_i}}\right)$$

$$l(\beta) = \sum_{i=1}^{n} y_i \beta x_i - \log\left(1 + e^{\beta x_i}\right)$$

$$\beta = \arg\max_{\beta}\ l(\beta)$$

Parameter Estimation

$$l(\beta) = \sum_{i=1}^{n} y_i \beta x_i - \log\left(1 + e^{\beta x_i}\right)$$ - transcendental Equation

- Numerical Methods can be used for approximation

- Consider the Newton Raphson Method

$$\nabla_\beta l(\beta) = \nabla_\beta l(\beta^*) + (\beta - \beta^*)\nabla_{\beta\beta} l(\beta^*)$$

$$\nabla_\beta l(\beta^*) + (\beta - \beta^*)\nabla_{\beta\beta} l(\beta^*) = 0$$

$$\beta = \beta^* - \frac{\nabla_\beta l(\beta^*)}{\nabla_{\beta\beta} l(\beta^*)}$$

$$\beta^{t+1} = \beta^t - \frac{\nabla_\beta l(\beta^t)}{\nabla_{\beta\beta} l(\beta^t)}$$

Parameter Estimation

$$\nabla_\beta l = \nabla_\beta \sum_{i=1}^{n} y_i \beta x_i - \log\left(1 + e^{\beta x_i}\right)$$

$$\nabla_\beta l = \sum_{i=1}^{n} \nabla_\beta \left[y_i \beta x_i - \log\left(1 + e^{\beta x_i}\right) \right]$$

$$\nabla_\beta l = \sum_{i=1}^{n} \nabla_\beta \left[y_i \beta x_i \right] - \nabla_\beta \left[\log\left(1 + e^{\beta x_i}\right) \right]$$

$$\nabla_\beta l = \sum_{i=1}^{n} y_i x_i - \left[\frac{1}{1 + e^{\beta x_i}} \, e^{\beta x_i} \, x_i\right]$$

$$\nabla_\beta l = \sum_{i=1}^{n} y_i x_i - \left[\frac{1}{1 + e^{-\beta x_i}} \, x_i\right]$$

$$\nabla_\beta l = \sum_{i=1}^{n} y_i x_i - \left[p(x_i) \, x_i\right]$$

$$\nabla_\beta l = \sum_{i=1}^{n} \left[y_i - p(x_i)\right] x_i$$

Parameter Estimation

$$\nabla_{\beta\beta} l = \nabla_{\beta} \sum_{i=1}^{n} [y_i - p(x_i)] x_i$$

$$\nabla_{\beta\beta}\, l = \sum_{i=1}^{n} \nabla_{\beta}\, [y_i - p(x_i)] x_i$$

$$\nabla_{\beta\beta}\, l = \sum_{i=1}^{n} \nabla_{\beta}\ - p(x_i) x_i$$

$$\nabla_{\beta\beta}\, l = \sum_{i=1}^{n} \nabla_{\beta} - \left[\frac{1}{1 + e^{-\beta x_i}}\right] x_i$$

$$\nabla_{\beta\beta}\, l = \sum_{i=1}^{n} \left[\frac{1}{1 + e^{-\beta x_i}}\right]^2 e^{-\beta x_i}\, (-x_i)\, x_i$$

$$\nabla_{\beta\beta}\, l = -\sum_{i=1}^{n} \left[\frac{e^{-\beta x_i}}{1 + e^{-\beta x_i}}\right] \left[\frac{1}{1 + e^{-\beta x_i}}\right] x_i^T\, x_i$$

$$\nabla_{\beta\beta}\, l = -\sum_{i=1}^{n} p(x_i)(1 - p(x_i)\, x_i^T\, x_i$$

Parameter Estimation

$$\nabla_\beta l = \sum_{i=1}^{n} [y_i - p(x_i)]x_i$$

$$\nabla_\beta l = X^T(Y - \hat{Y})$$

$$\nabla_{\beta\beta}\, l = -\sum_{i=1}^{n} p(x_i)(1 - p(x_i)\, x_i^T\, x_i$$

$$\nabla_{\beta\beta}\, l = -X^T P(1-P)X$$

$$\nabla_{\beta\beta}\, l = -X^T W X$$

$$\beta^{t+1} = \beta^t - \frac{\nabla_\beta l(\beta^t)}{\nabla_{\beta\beta} l(\beta^t)}$$

update rule =

$$\beta^{(t+1)} = \beta^{(t)} + \left(X^T\, W^{(t)} X\right)^{-1} X(Y - \hat{Y}^{(t)})$$

$$p(X) = \frac{1}{1 + e^{-\beta X}}$$

Fitting equation to the data

- **Linear regression: Least squares**
- **Logistic regression: Maximum likelihood**
- **Likelihood function**
  - **Estimates parameters α and β**
  - **Practically easier to work with log-likelihood**

$$L(\mathrm{B}) = \ln\left[l(\mathrm{B})\right] = \sum_{i=1}^{n} \left\{ y_i \ln\left[\pi(x_i)\right] + (1 - y_i) \ln\left[1 - \pi(x_i)\right] \right\}$$

- **where π(X)= Sigma (B1x1+ B2X2+...)**

Maximum likelihood

- **Iterative computing**
  - **Choice of an arbitrary value for the coefficients (usually 0)**
  - **Computing of log-likelihood**
  - **Variation of coefficients' values**
  - **Reiteration until maximisation (plateau)**

- **Results**
  - **Maximum Likelihood Estimates (MLE) for α and β**
  - **Estimates of P(y) for a given value of x**

MLEs of Means

- Suppose we draw a random sample of size $n = 3$ from a normal population with unknown mean μ and known variance $\sigma^2 = 5$. The observed values are $x_1 = 4$, $x_2 = 5$, and $x_3 = 6$. What is the best guess of μ?
- Basic stats answer: $\bar{x} = (4 + 5 + 6)/3 = 5$. Rationale: common sense
- Maximum likelihood answer: pick μ so that the probability of observing the three values given μ is maximized

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x - \mu)^2\right]$$

$$L(\mu) = \prod_{i=1}^{3} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x_i - \mu)^2\right]$$

$$\begin{aligned} l(\mu) &= \sum_{i=1}^{3} \left[-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}(x_i - \mu)^2\right] \\ &= -k_1 - k_2 \sum_{i=1}^{3} (x_i - \mu)^2 \end{aligned}$$

$$
\begin{aligned}
l(\mu) &= \sum_{i=1}^{3}\left[-\frac{1}{2}\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}(x_i-\mu)^2\right] \\
&= -k_1-k_2\sum_{i=1}^{3}(x_i-\mu)^2
\end{aligned}
$$

$$
\frac{dl(\mu)}{d\mu}=2k_2\sum_{i=1}^{3}(x_i-\mu)=0
$$

$$
\Longrightarrow \mu=\frac{1}{3}\sum_{i=1}^{3}x_i
$$

$$
\text{Note } \frac{d^2l(\mu)}{d\mu^2}=-3<0
$$

Likelihood Function for Logistic Regression

- Assume we have a random sample (observations independent, each have same probability of selection) and that we make two measurements on each observation (x_i, y_i), where y_i is a 0-1 variable and $i = 1, \ldots, n$
- Let $\pi_i = P(y_i = 1)$, i.e., prob(person i says yes), and

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \alpha + \beta x_i$$

- Note that the probability distribution for person i is

$$f_i(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

- Since observations are independent, the probability distribution (likelihood) and log-likelihood for our sample is

$$f(y_1, \ldots, y_n) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

$$\log(f) = \sum_{i=1}^{n} [y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)]$$

- Since observations are independent, the probability distribution (likelihood) and log-likelihood for our sample is

$$f(y_1, \ldots, y_n) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

$$\begin{aligned}
\log(f) &= \sum_{i=1}^{n} [y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)] \\
&= \sum_{i=1}^{n} \left[y_i \log(\frac{\pi_i}{1-\pi_i}) + \log(1-\pi_i) \right] \\
&= \sum_{i=1}^{n} y_i(\alpha + \beta x_i) - \sum_{i=1}^{n} \log[1 + \exp(\alpha + \beta x_i)]
\end{aligned}$$

We maximize this with respect to α and β

Multiple logistic regression

- More than one independent variable
  - Dichotomous, ordinal, nominal, continuous …

$$\ln\left(\frac{P}{1-P}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_i x_i$$

- Interpretation of β_i
  - Increase in log-odds for a one unit increase in x_i with all the other x_is constant
  - Measures association between x_i and log-odds adjusted for all other x_i