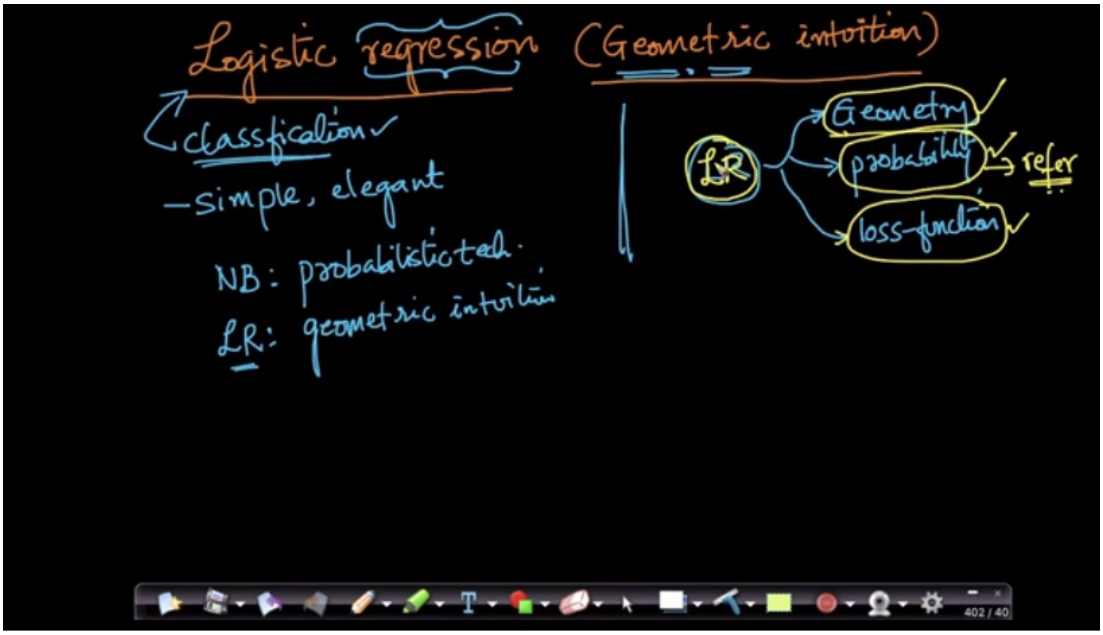
Notes

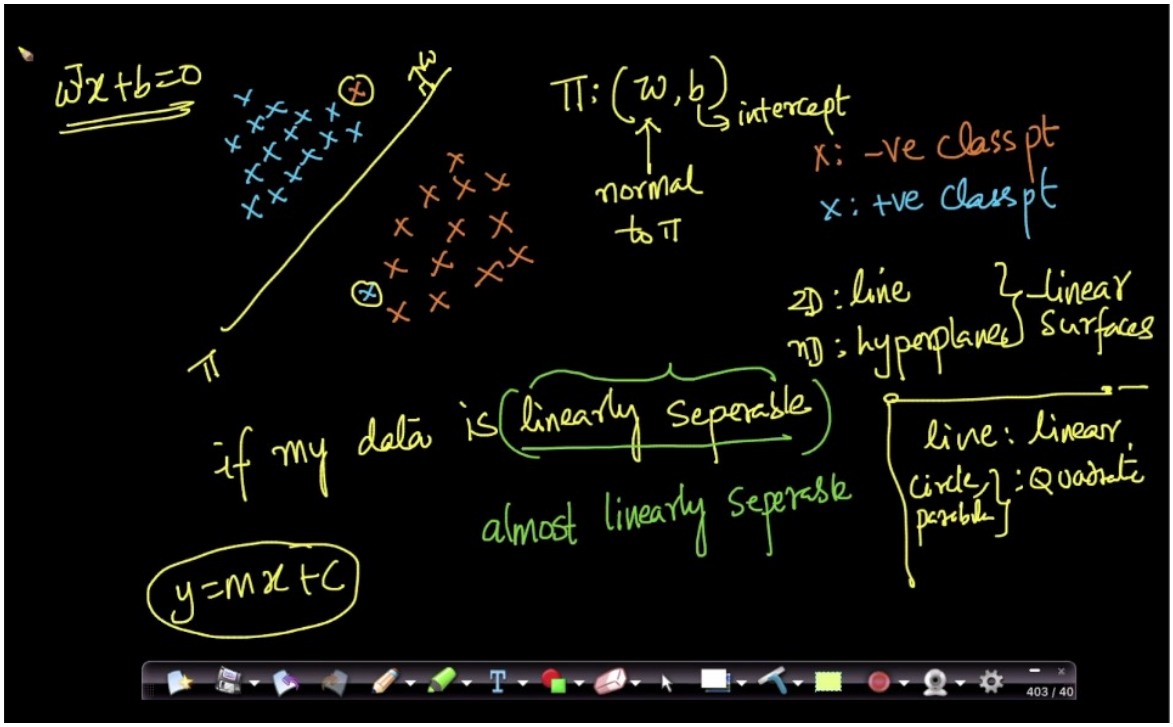
Logistic Regression
Geometric intuition of Logistic Regression:

LR can be derived using Geometry, probability and loss-function interpretation.



The two classes of points, we can draw a hyper plane in 2D to separate the points and hyper-plane in d – dimensional space.

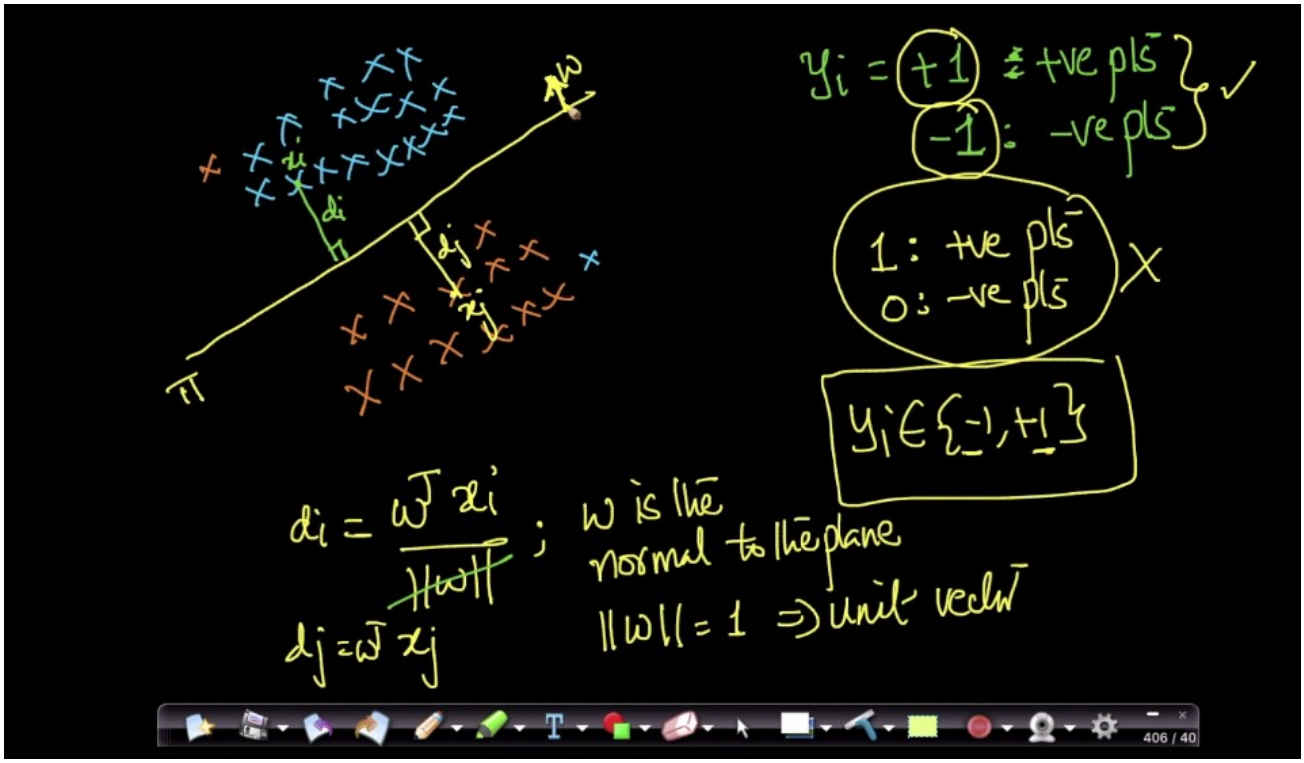
The data can be separated linearly and quadratic planes.



The equation of the plane is the product of the normal to the plane and the parameters of the plane plus the intercept term.

The equation of the line is y=mx +b.

Logistic regression assumes that the data is almost/perfectly linearly separable.

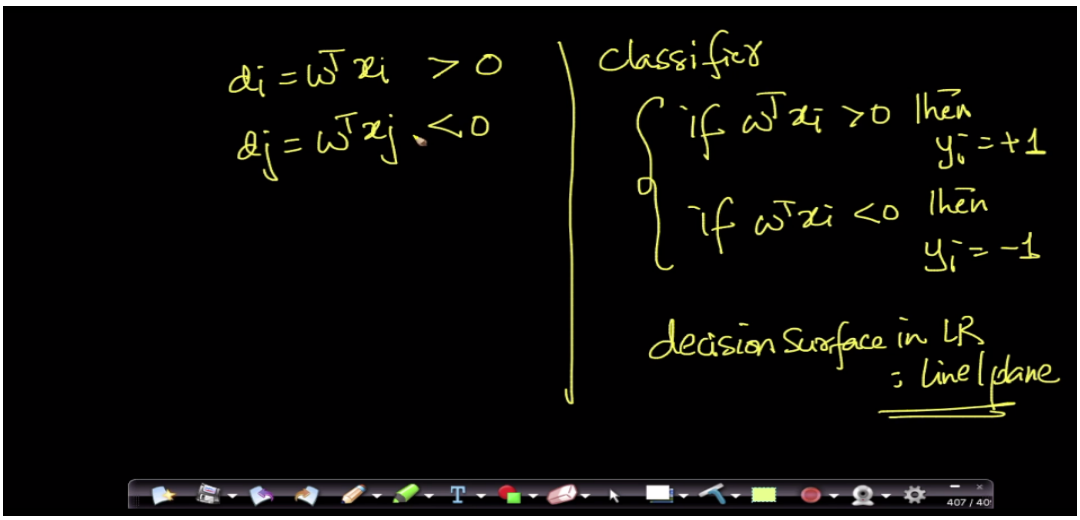


Here, **w** is the unit vector.

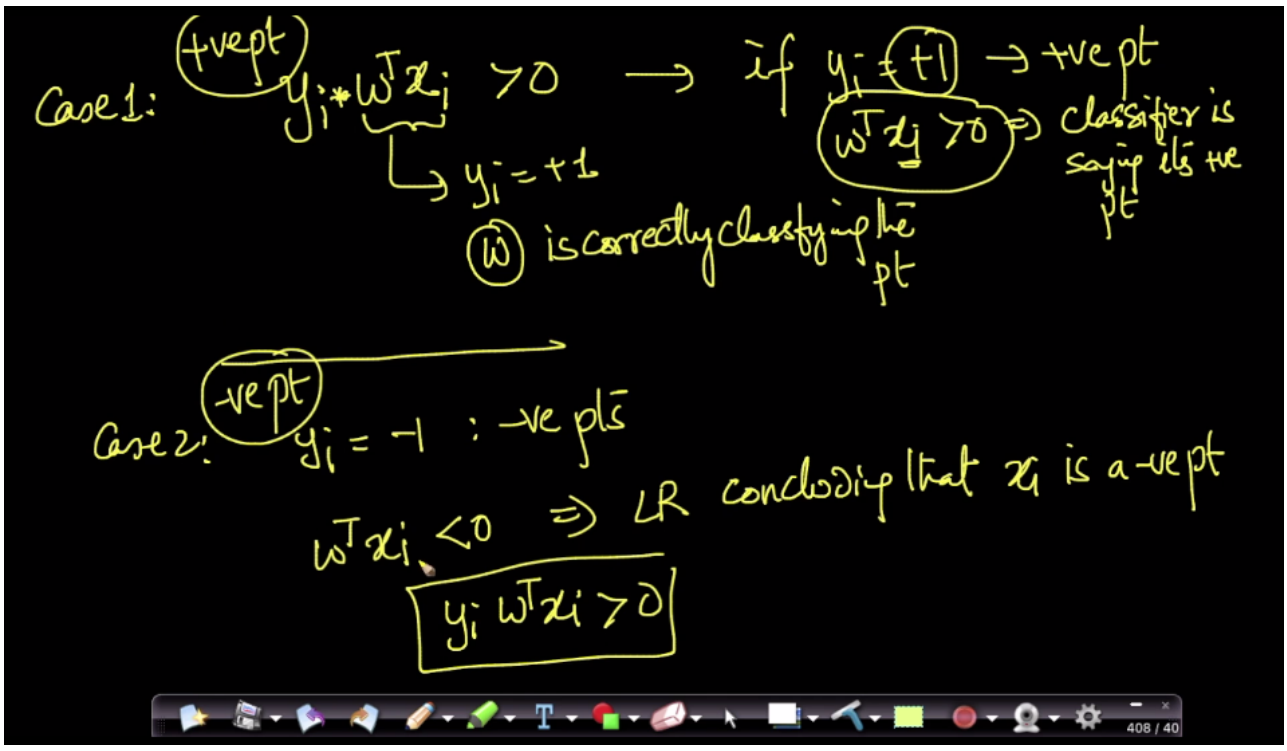
Note: The positive and negative points are denoted as +1 and -1 respectively.

The distance to the points xi and xj are computed where the distances are positive for xi and negative for xj.

Lets assume that the line passes through the origin.

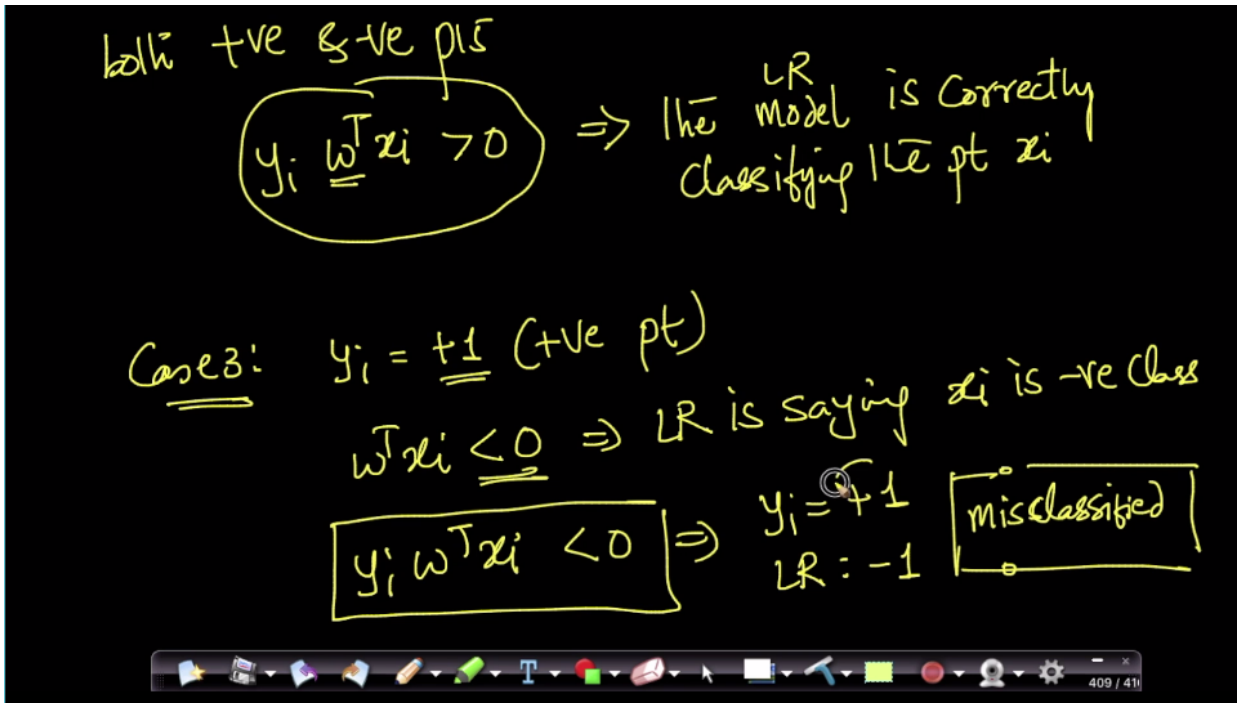


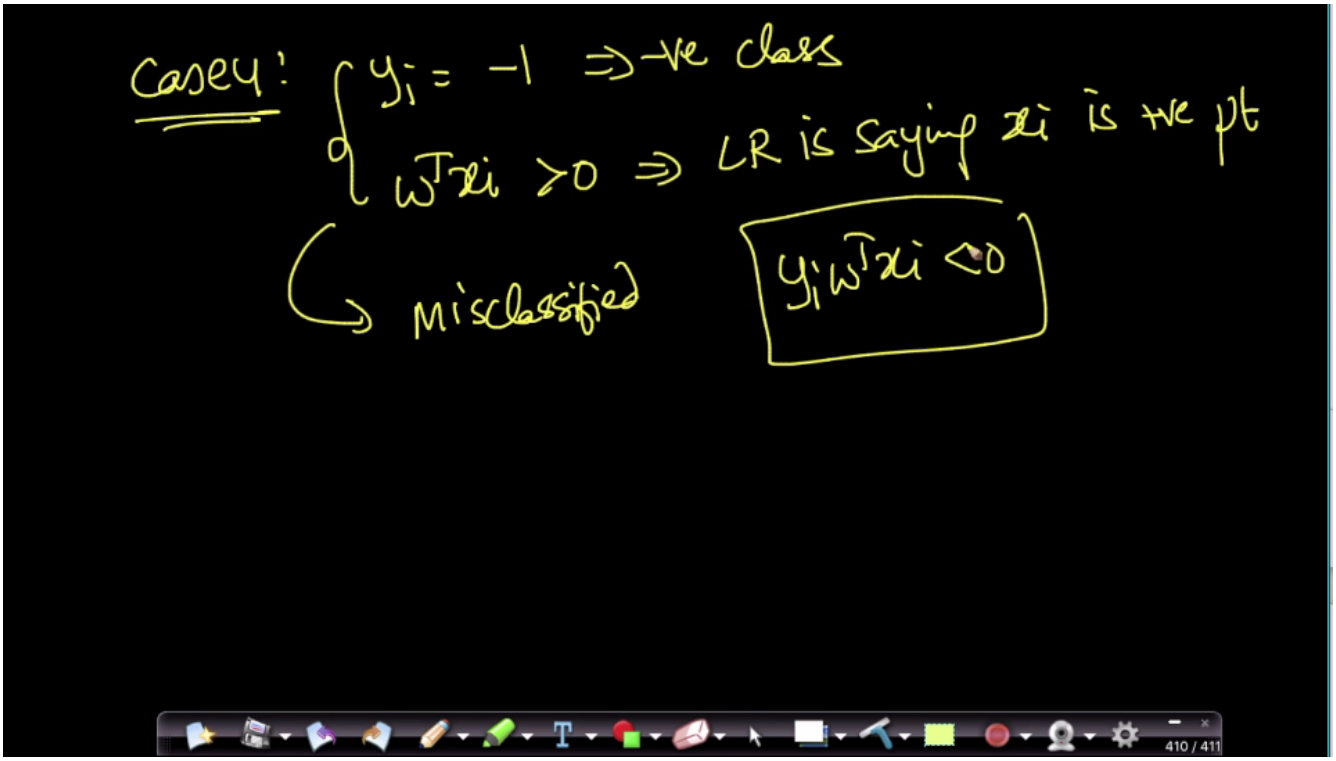
Different cases of optimization techniques.



For both positive and negative points the (yi * w.T * xi) > 0, the model is correctly classifying the point xi.

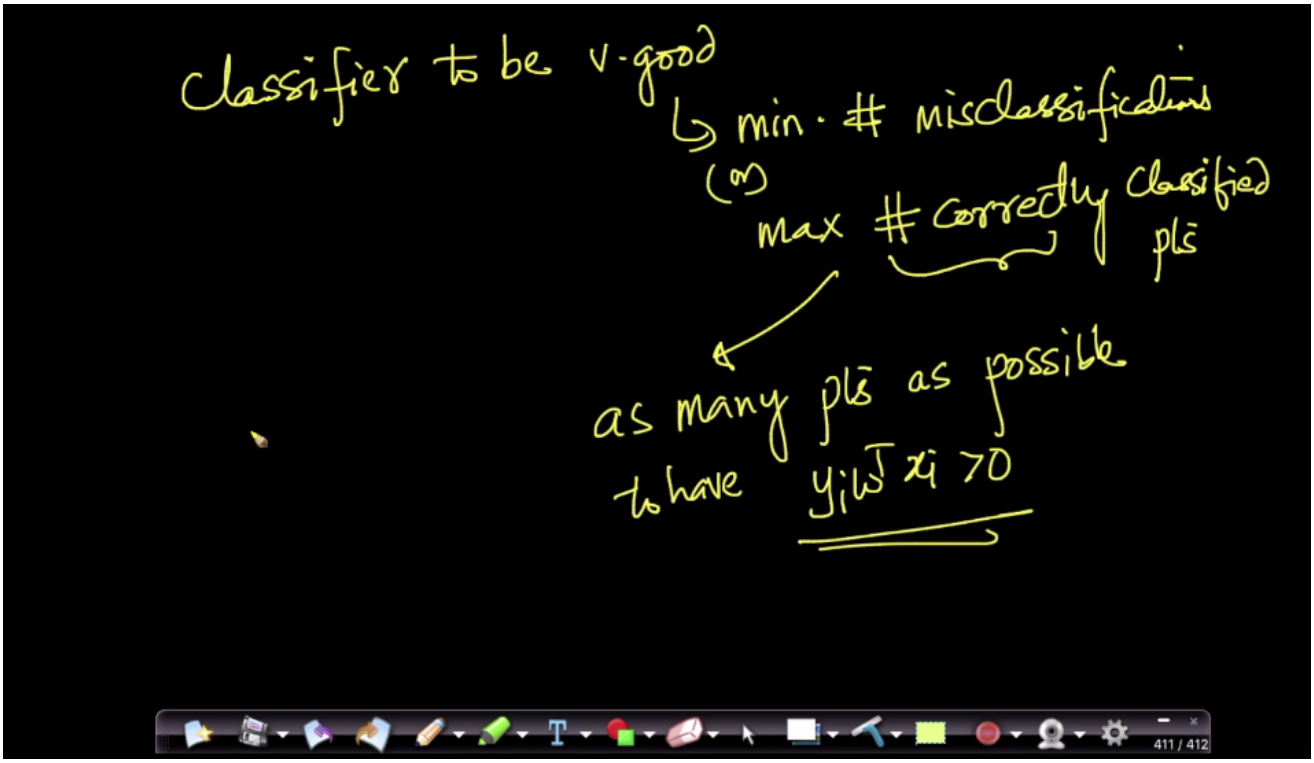
We want the point wi that correctly classifies the point.





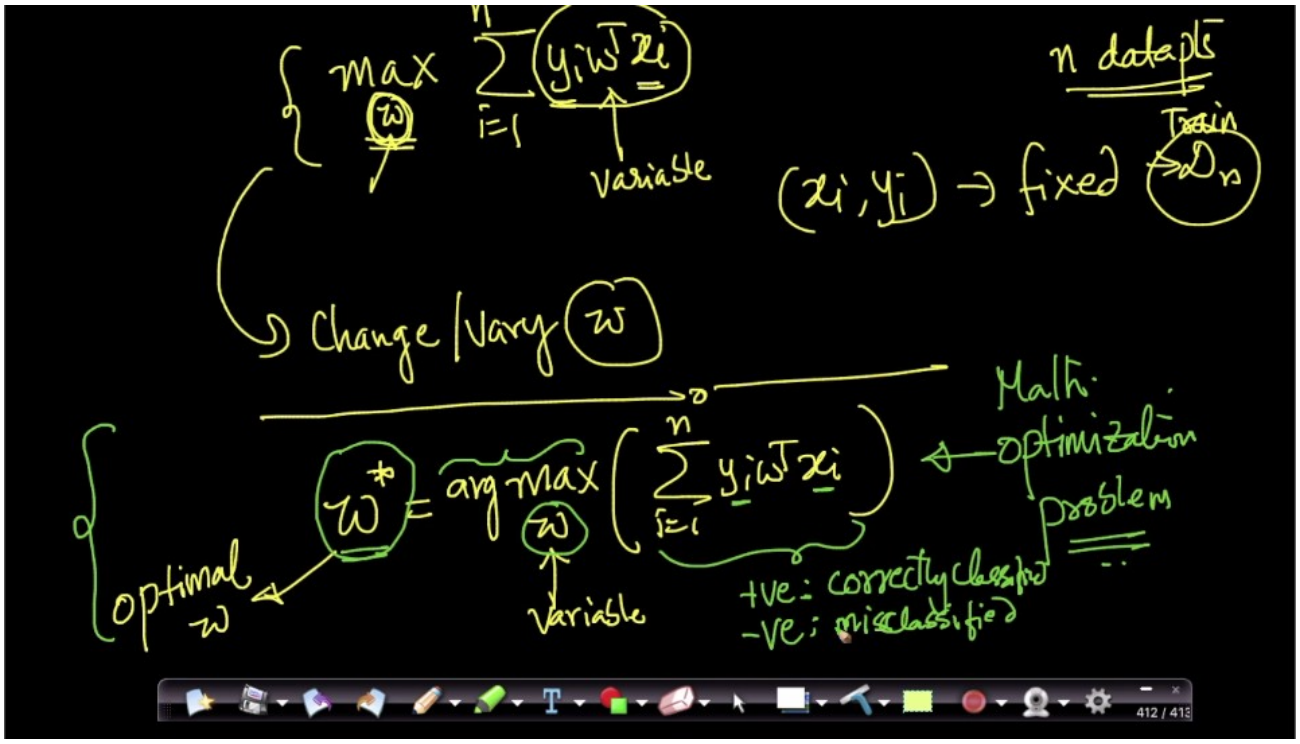
Finally, we want out classifier to make minimum number of misclassifications.

Characteristics of LR:



Mathematical formulation:

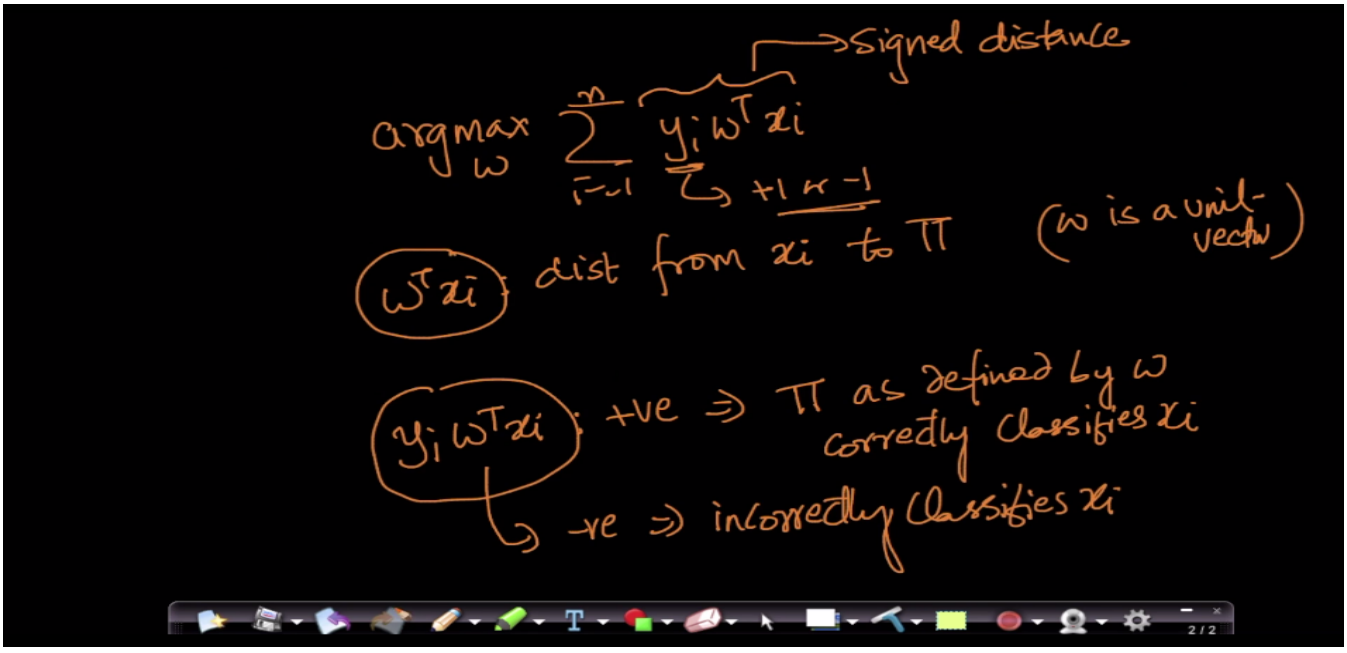
The objective function formulation.



The ultimate goal is to find the optimal **W** that maximizes the accuracy of the model.

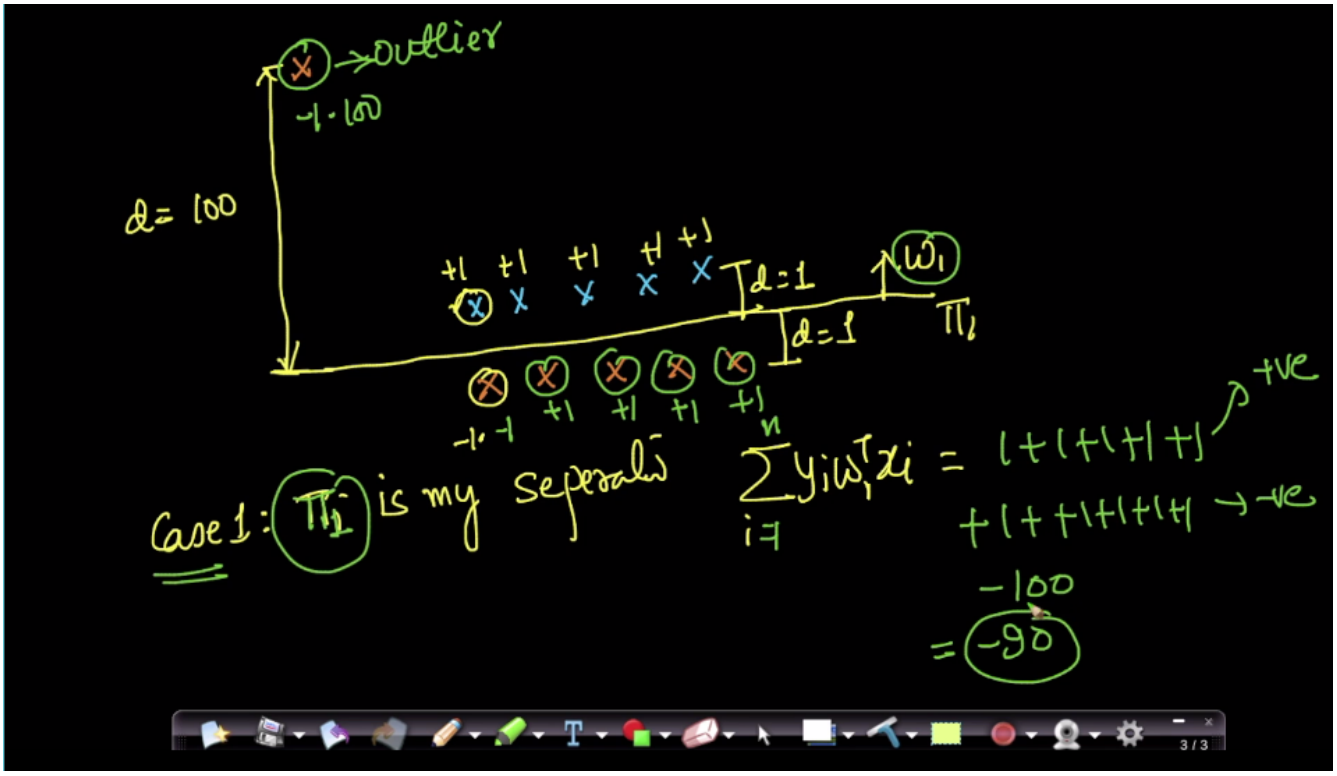
Sigmoid function: Squashing

Our objective is



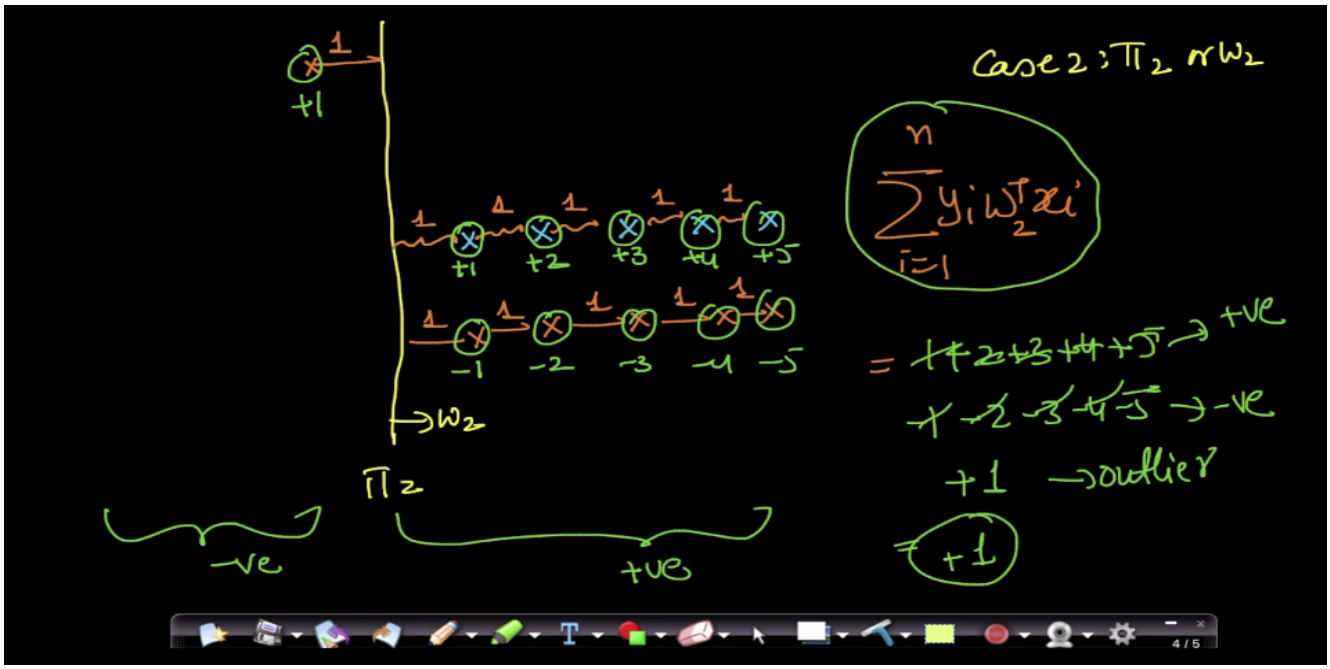
Edge cases:

All the points are assumed to be at the unit distance from the hyper plane, except one -ve point considered as outlier at a distance of 100 units. This point will mess up the objective function.



Our objective is to maximize the sum of signed distances.

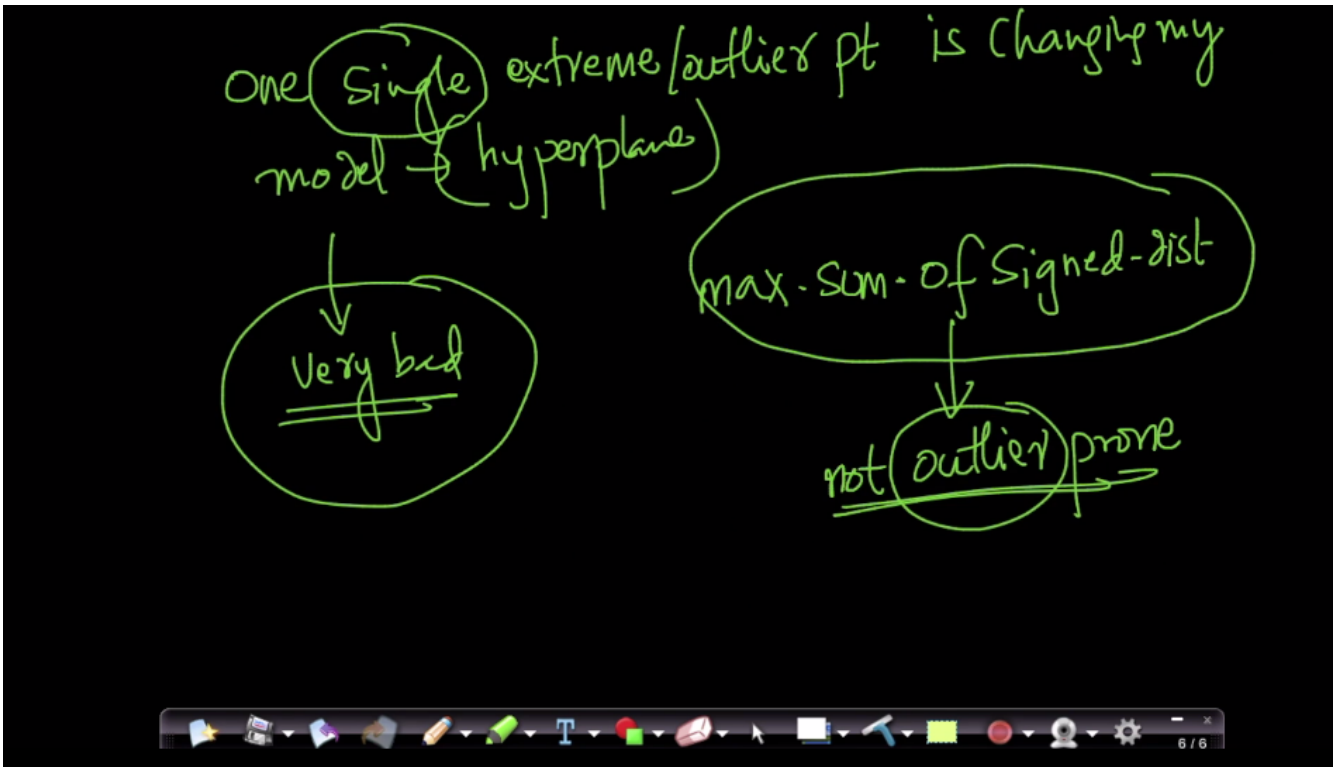
Edge case hyper plane:



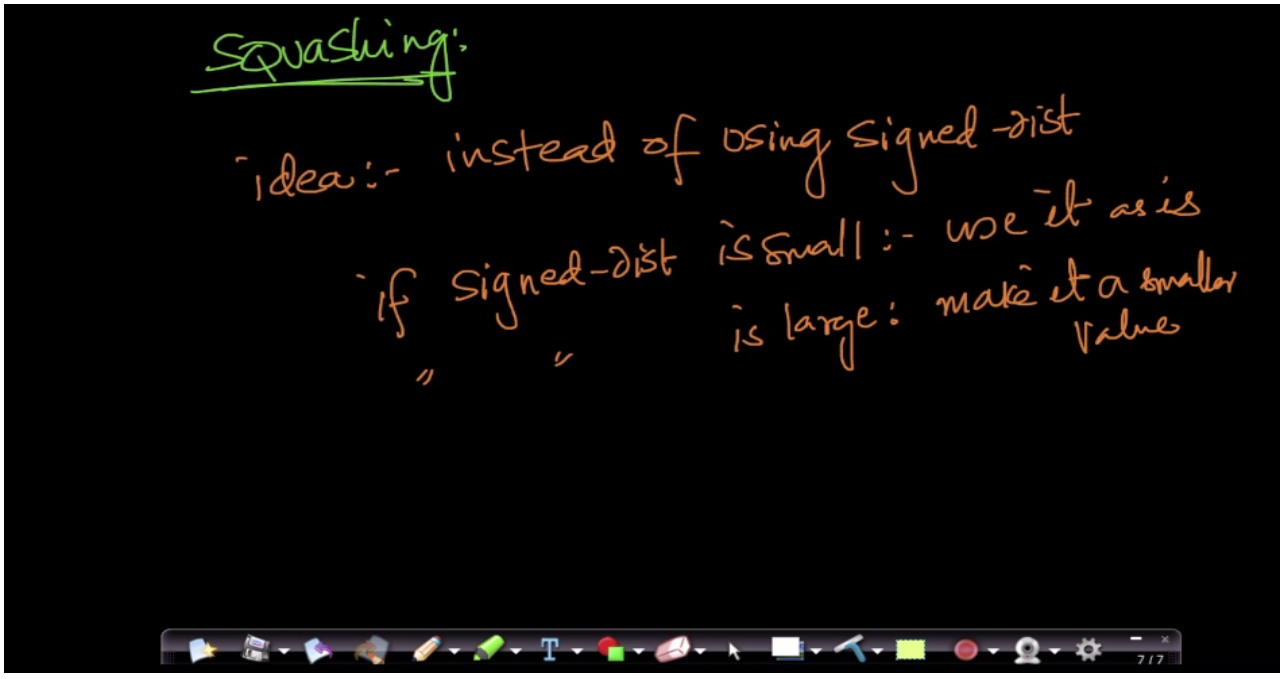
The sum of signed distance for **pi1** is less than the signed distance **pi2** then we will choose the **pi2**.

But geometrically the plane pi2 is a terrible choice, because accuracy suffers.

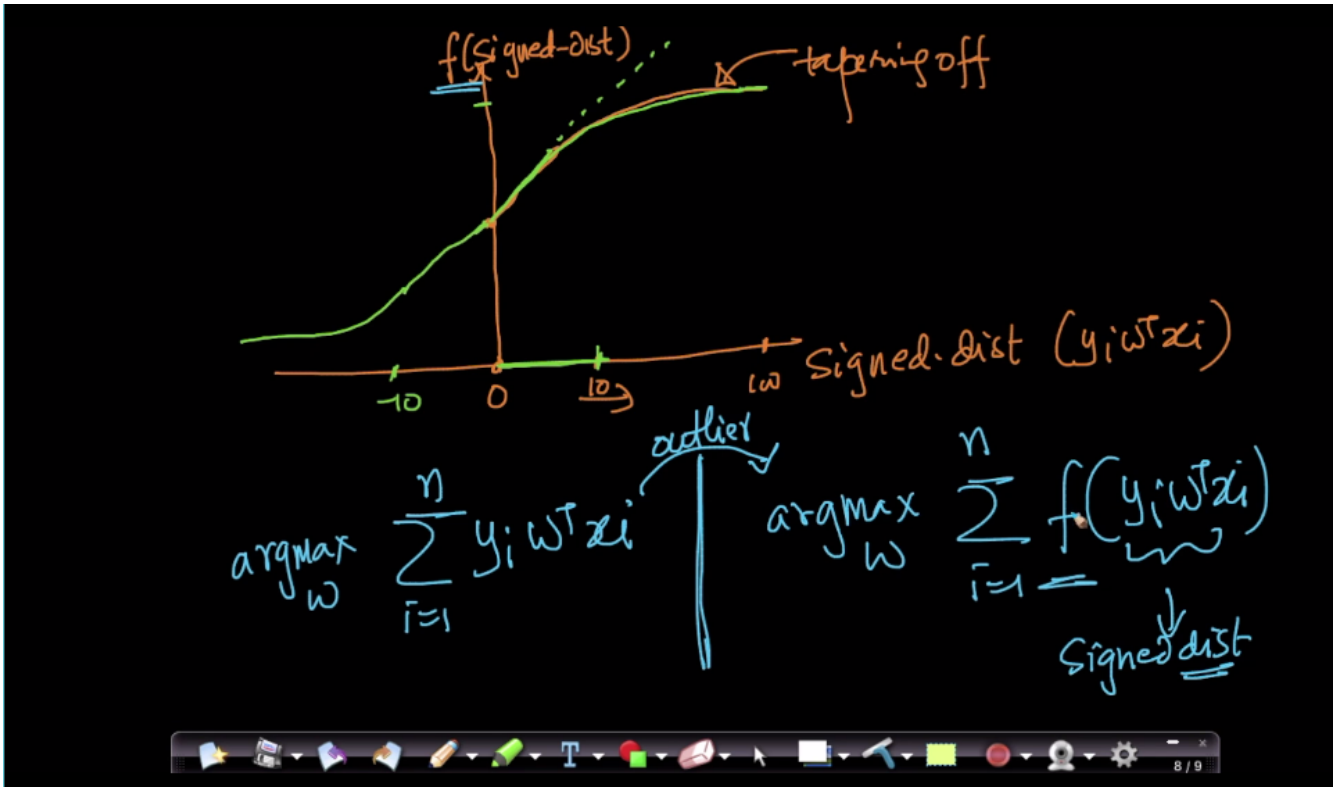
Intuitively, **pi1** is better than **pi2.**



In order to overcome this situation we will use the technique called squashing.

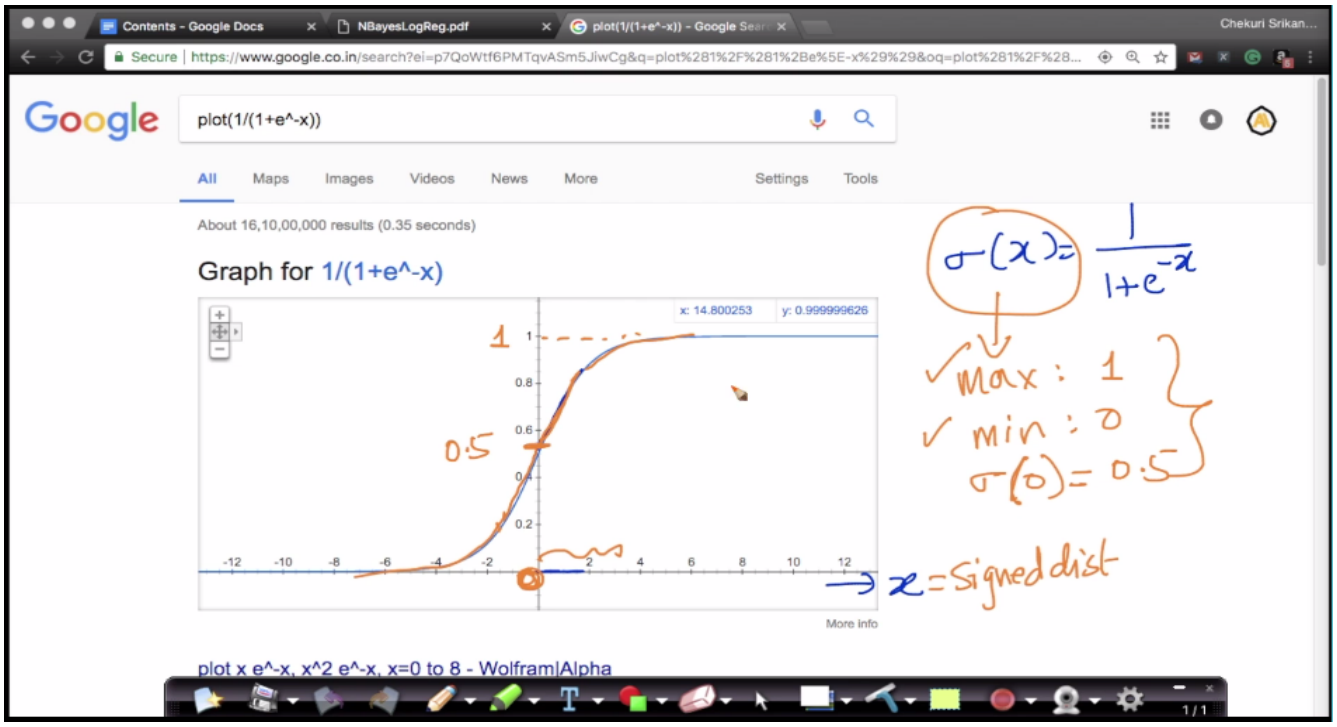


In this technique we will apply a function on top of signed distances to make it prone from the outlier effect.



The sigmoid function makes this possible.

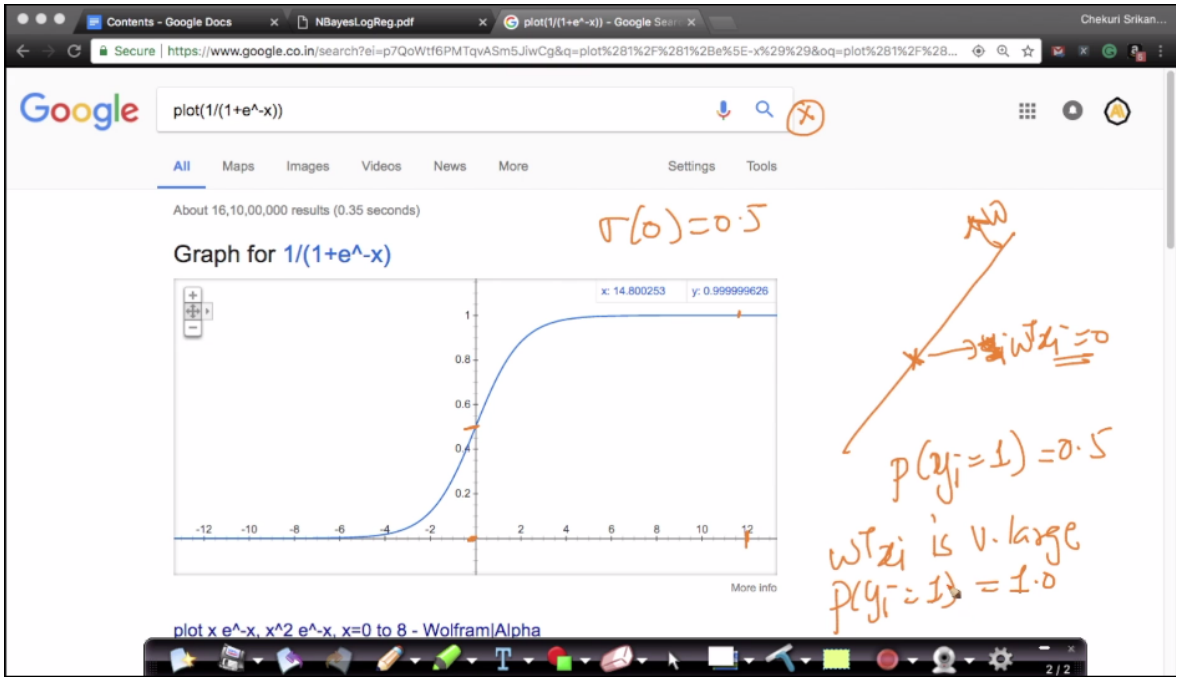
Properties of Sigmoid funtion:



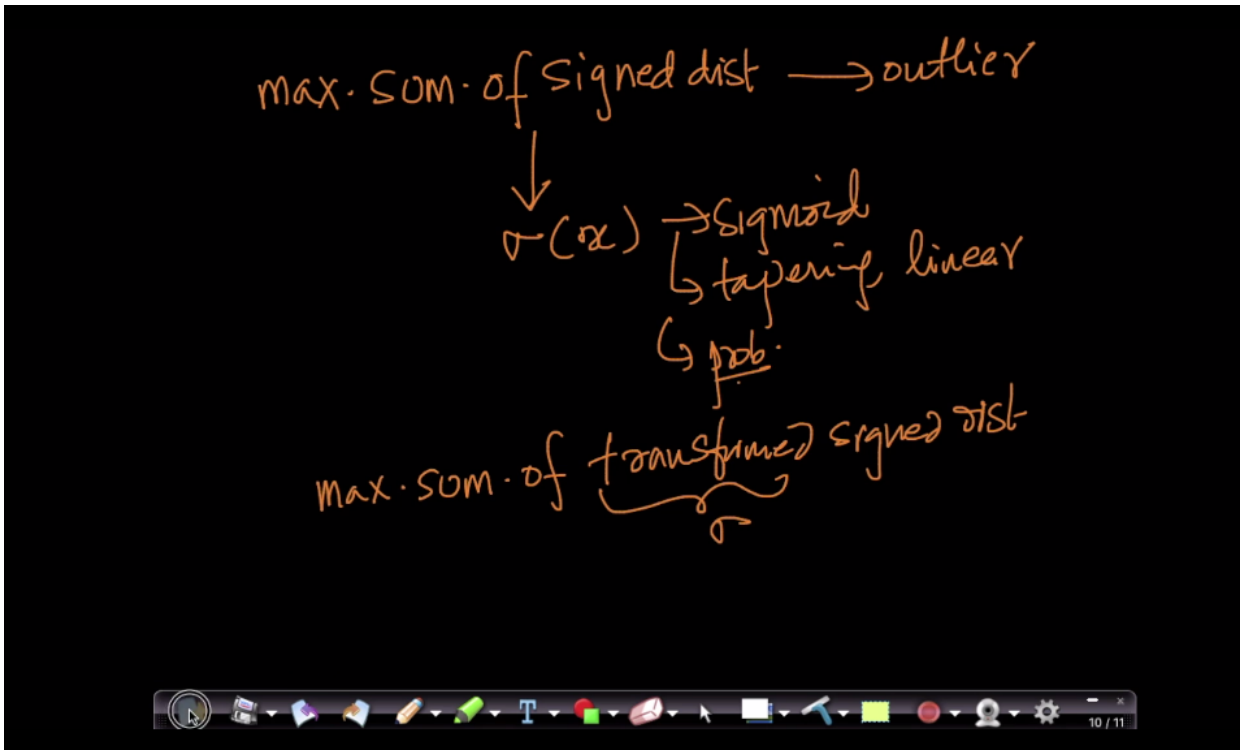
We choose sigmoid function because it has probabilistic interpretation.

If the point is on the curve we cannot decide the probability of the class, sigmoid function gives the probability as 0.5.

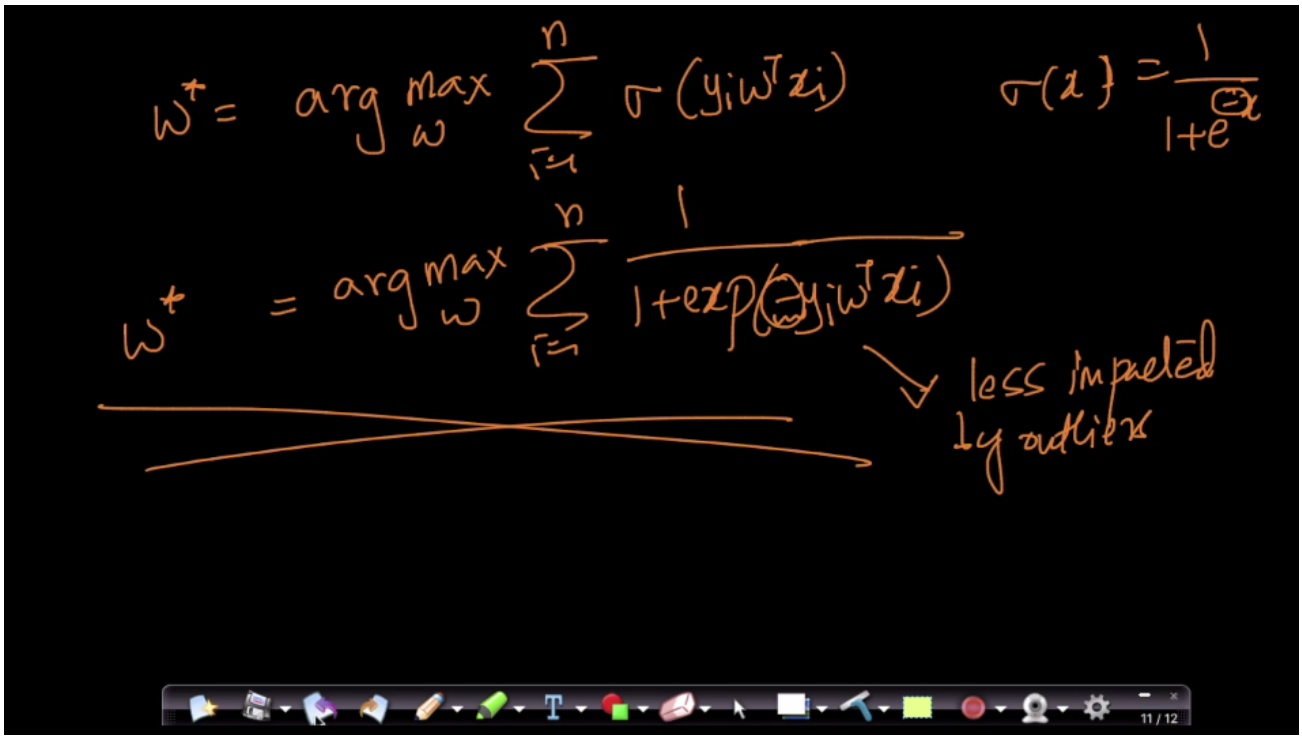
If the signed distance is very large then the class will be near to the maximum of the sigmoid that is 1.



Our transformed objective function:

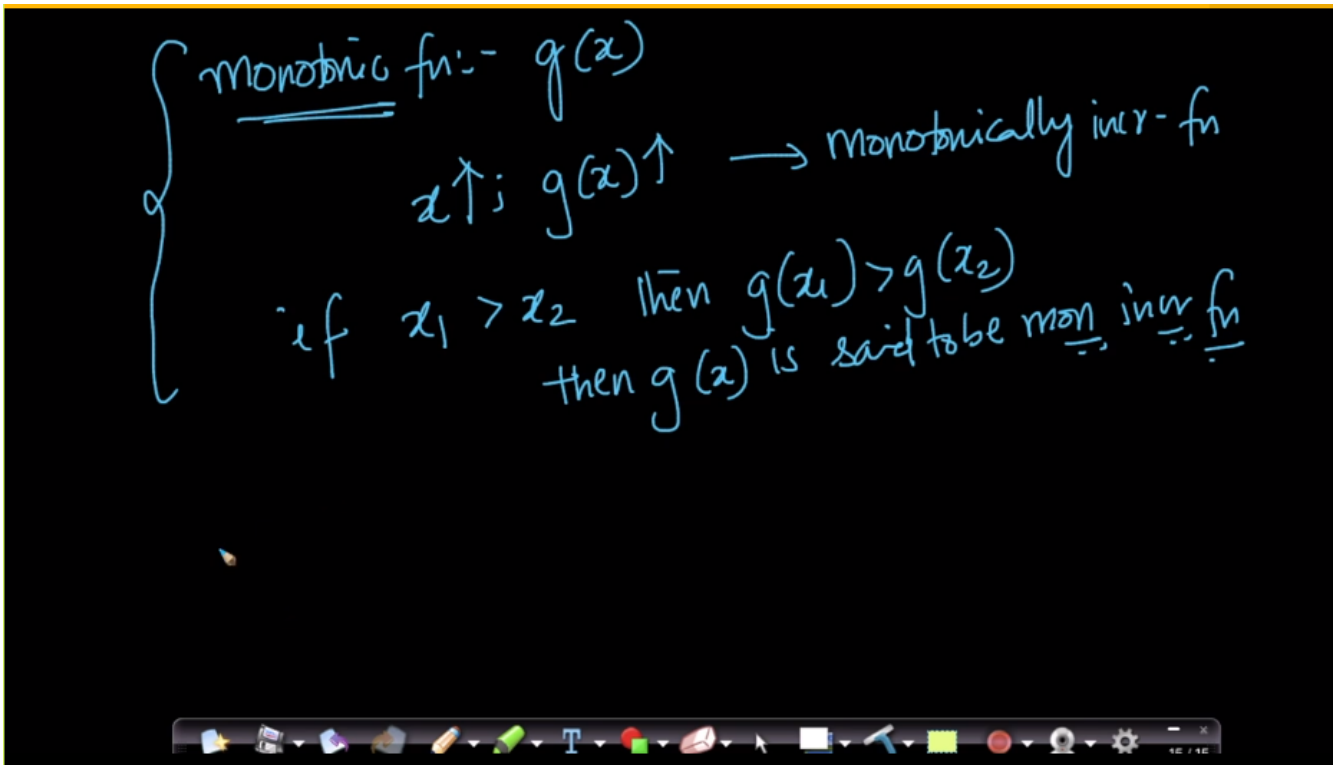


Transformed mathematical objective function.



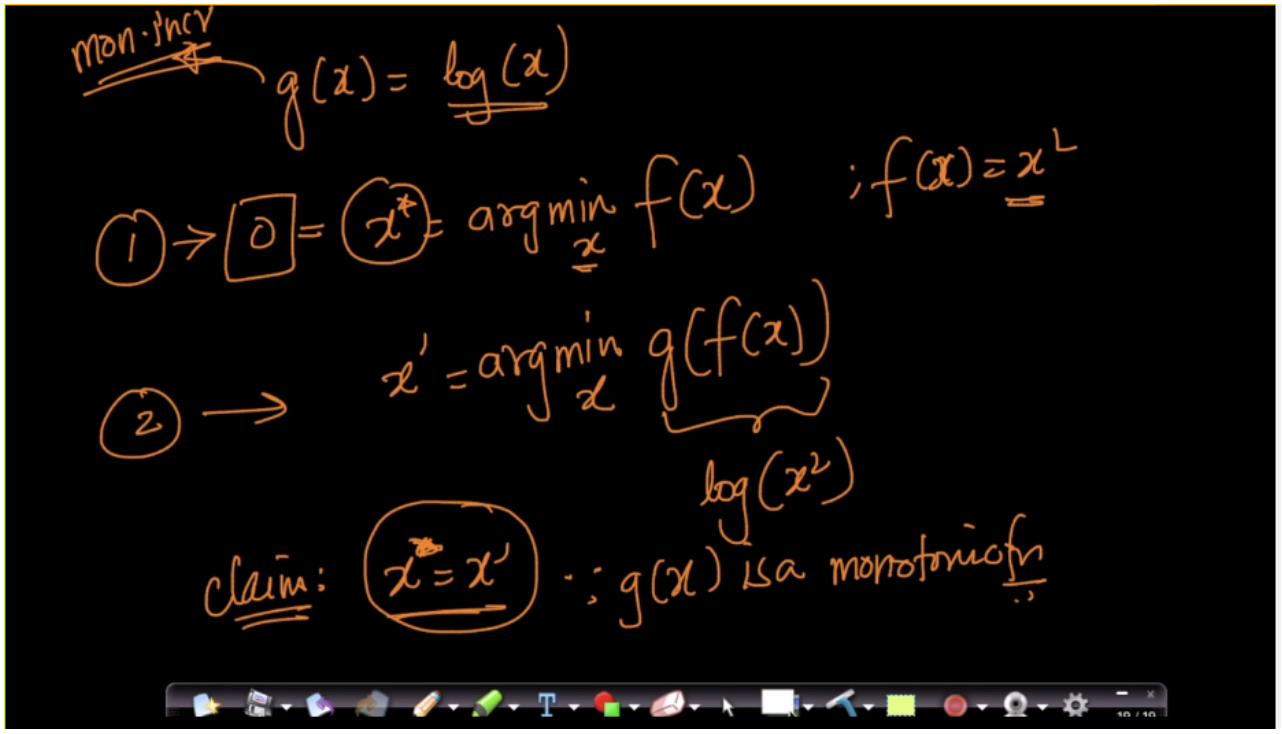
Mathematical formulation of Objective function:

A function g(x) is said to be monotonic function as x increases g(x) increases.

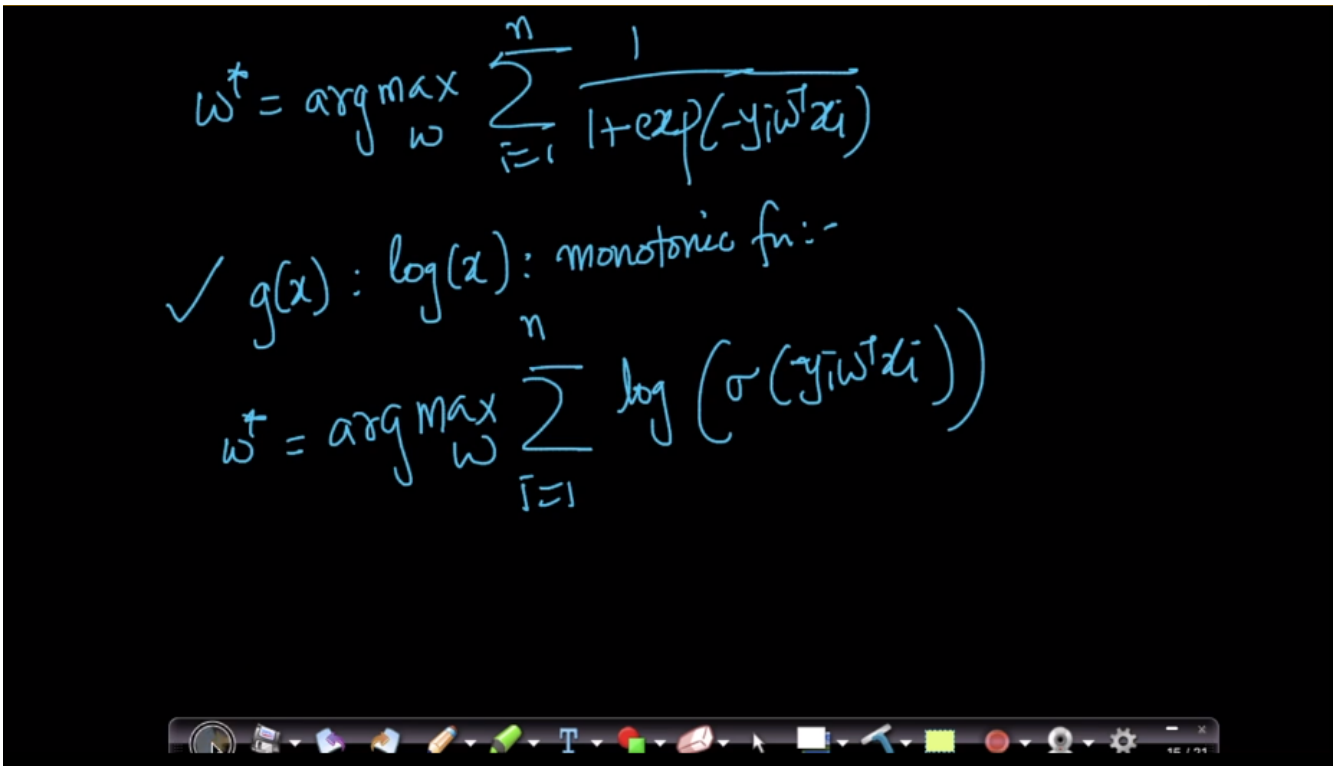


Optimizing the fun(fun(x)):

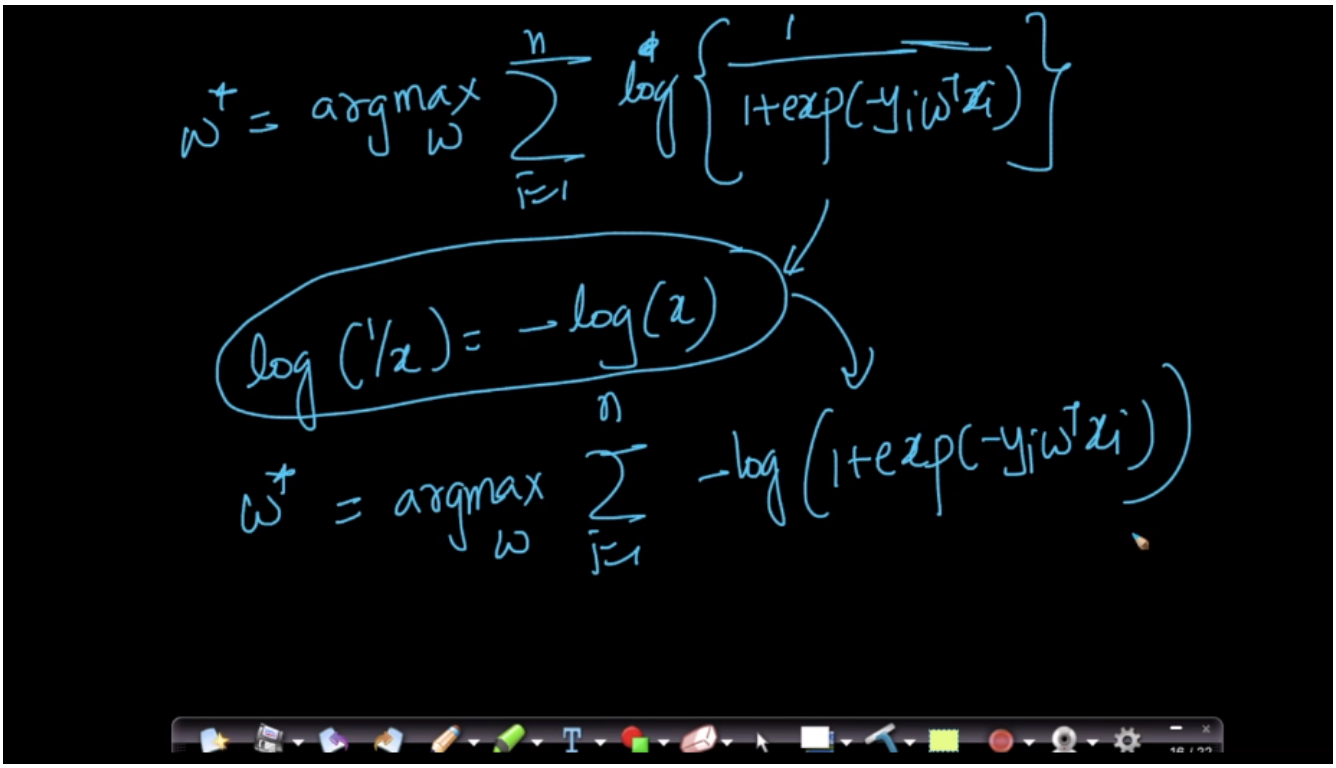
We can claim that the value of the whole function that minimizes is equal to the value that minimizes the inner function. Since, both the functions are monotonic.



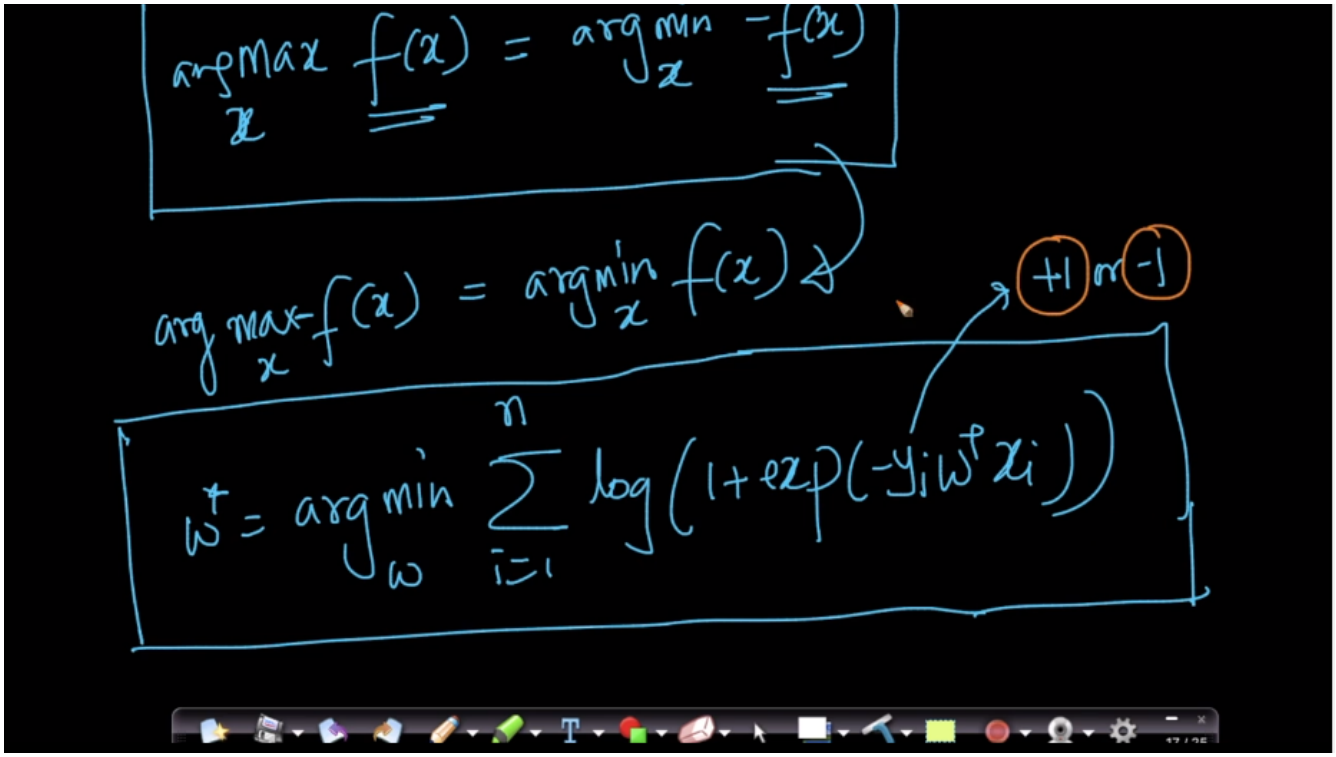
This is the function to be optimized.



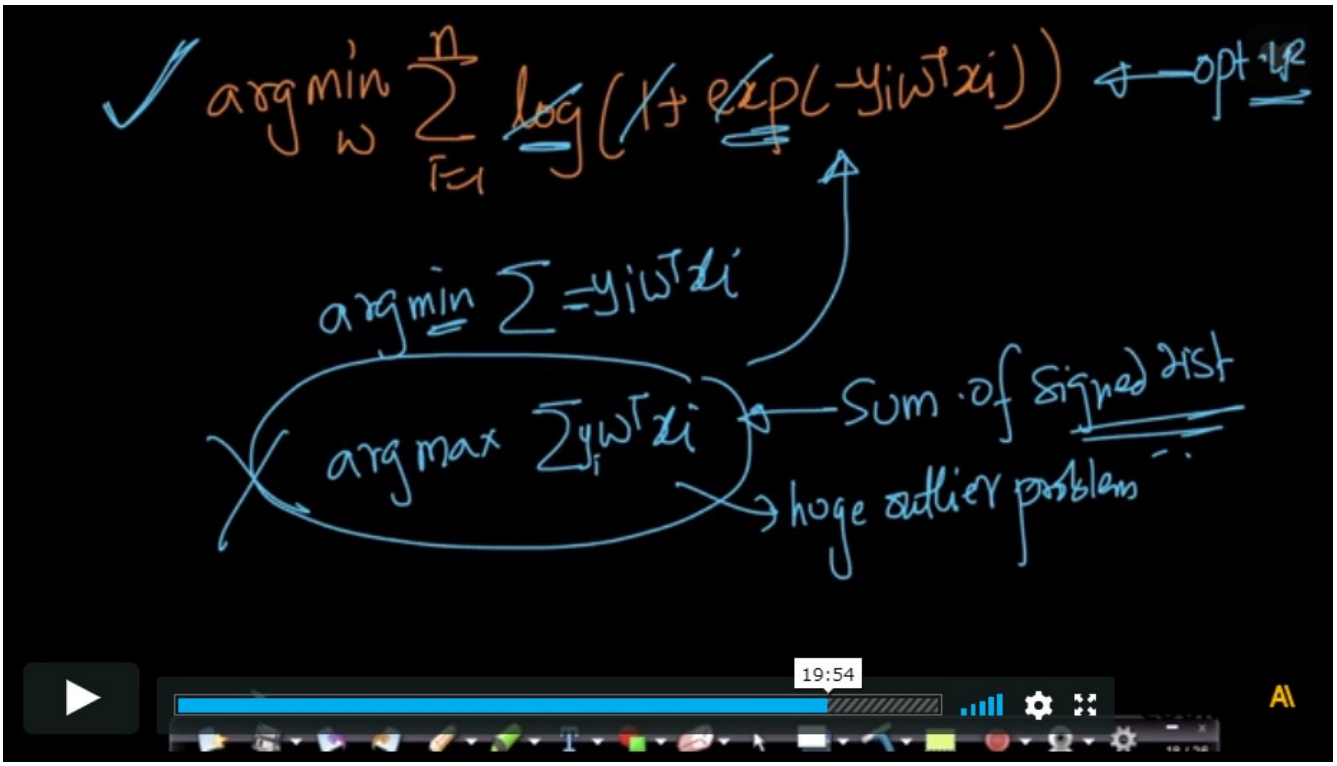
Apply log to the objective function.



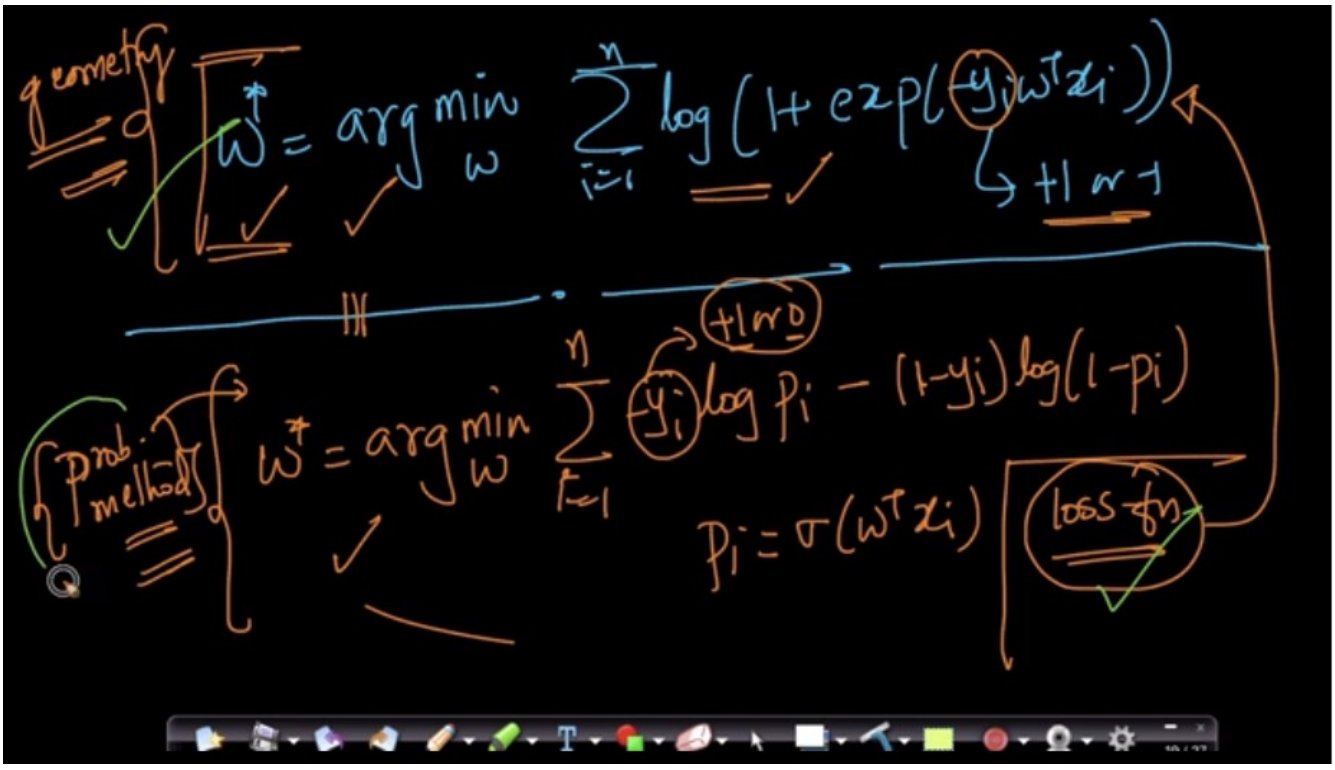
In optimization the value which maximizes the f(x) is also minimizes the -f(x).



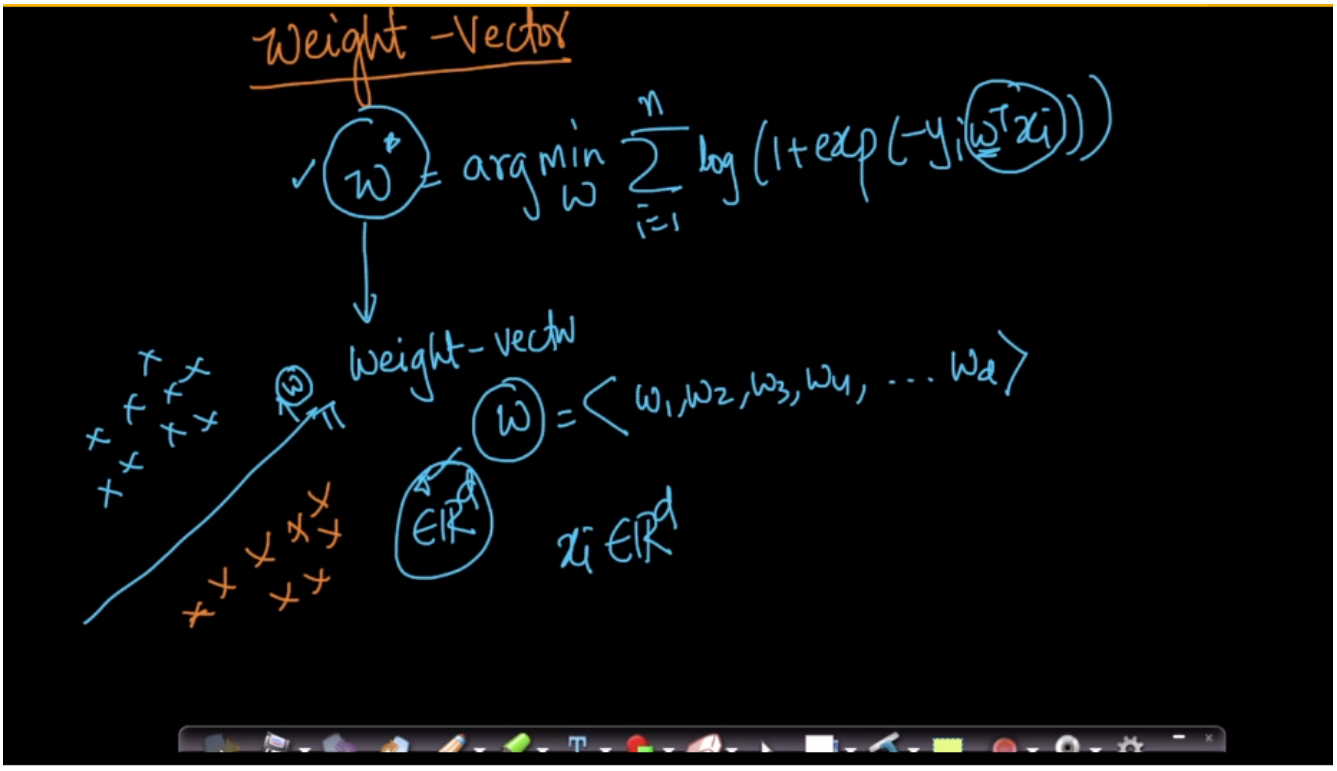
Removing the 1 is will be more favorable, because the log and exp will cancel out each other.



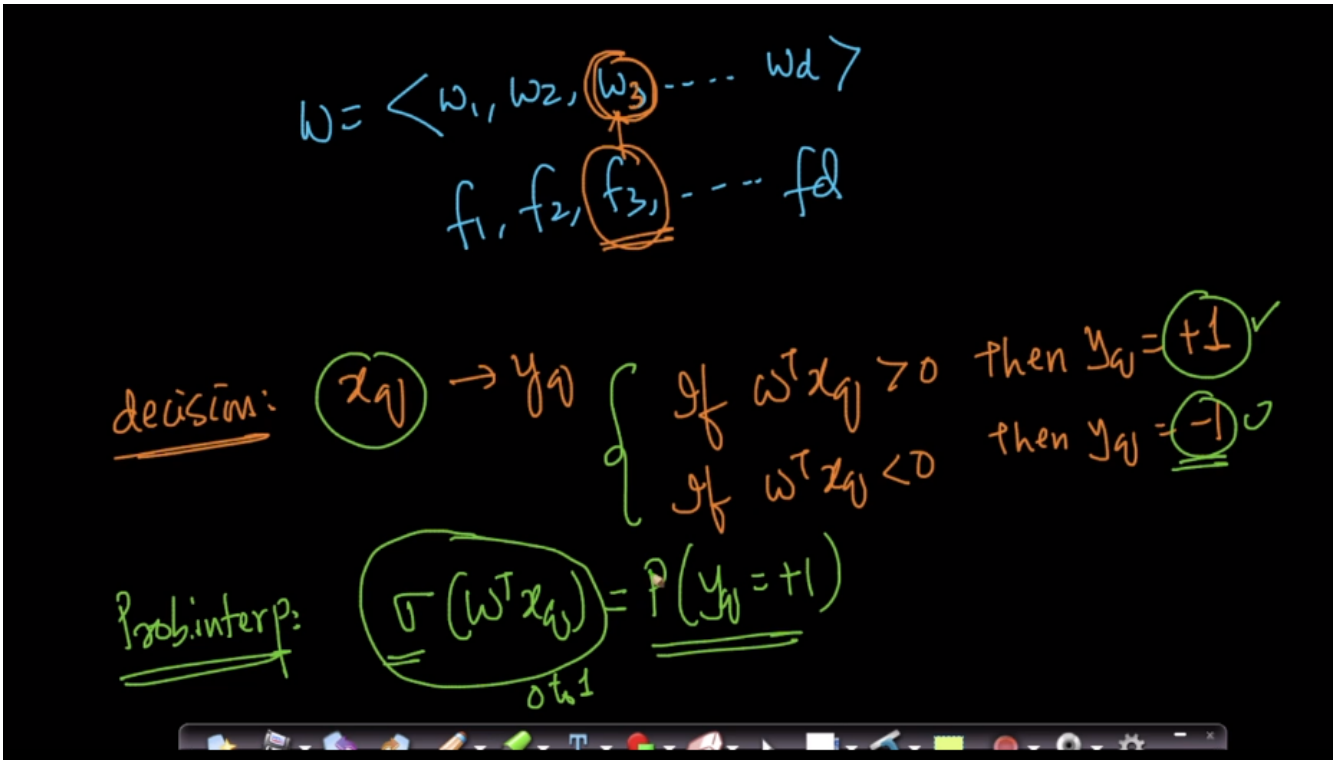
Other formulation of the Objective function(probabilistic method). The both formulations are one and the same.



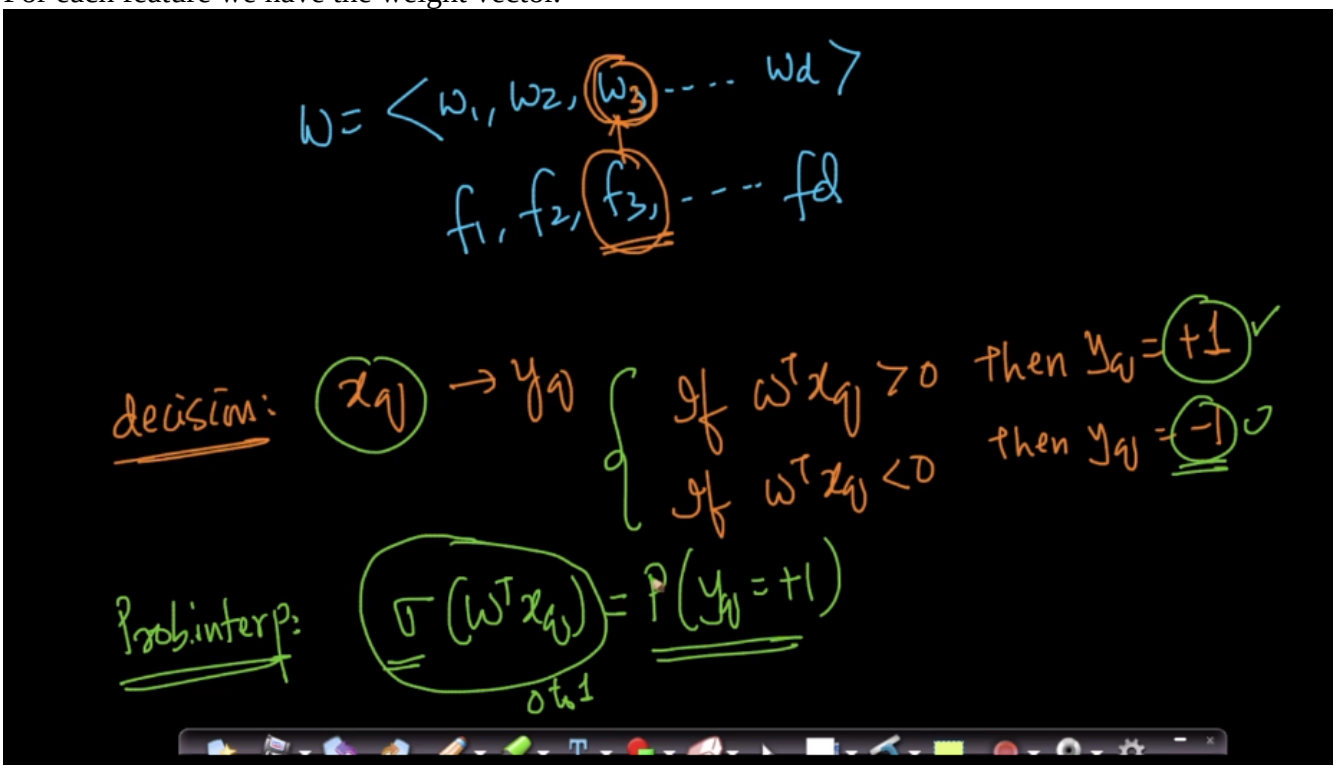
Weight vector:
W is also a d dimensional vector.



Probabilistic interpretation:

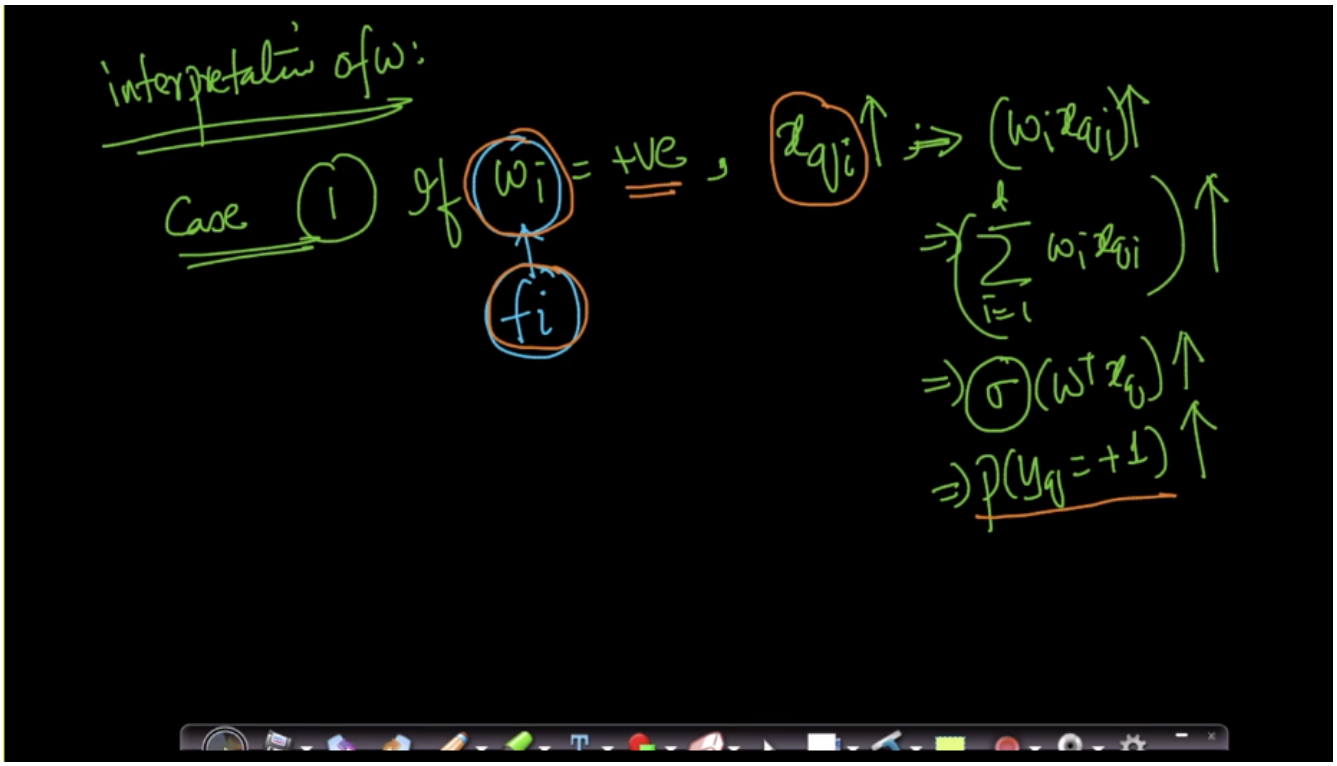


For each feature we have the weight vector.

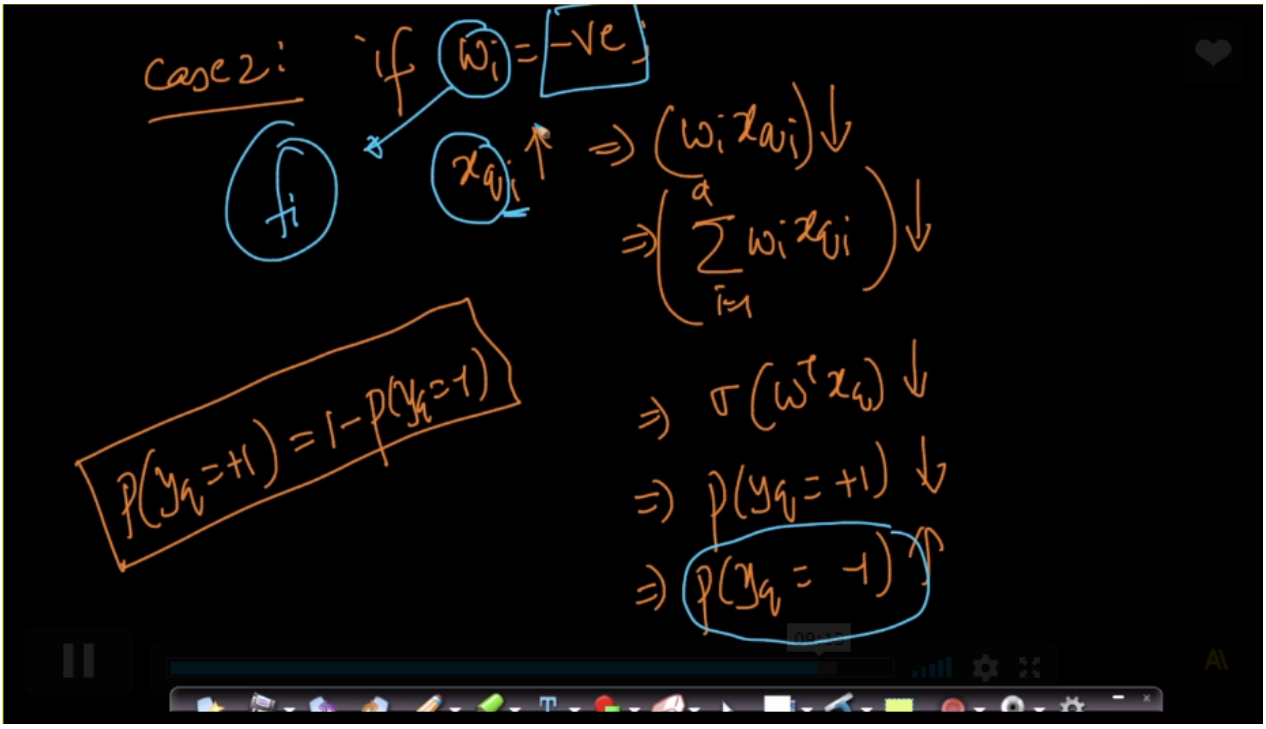


Interpretation of W:

Case – 1:
As **wi** increases the point **xq*wi** also increases and the probability of the point **yq** will also increase

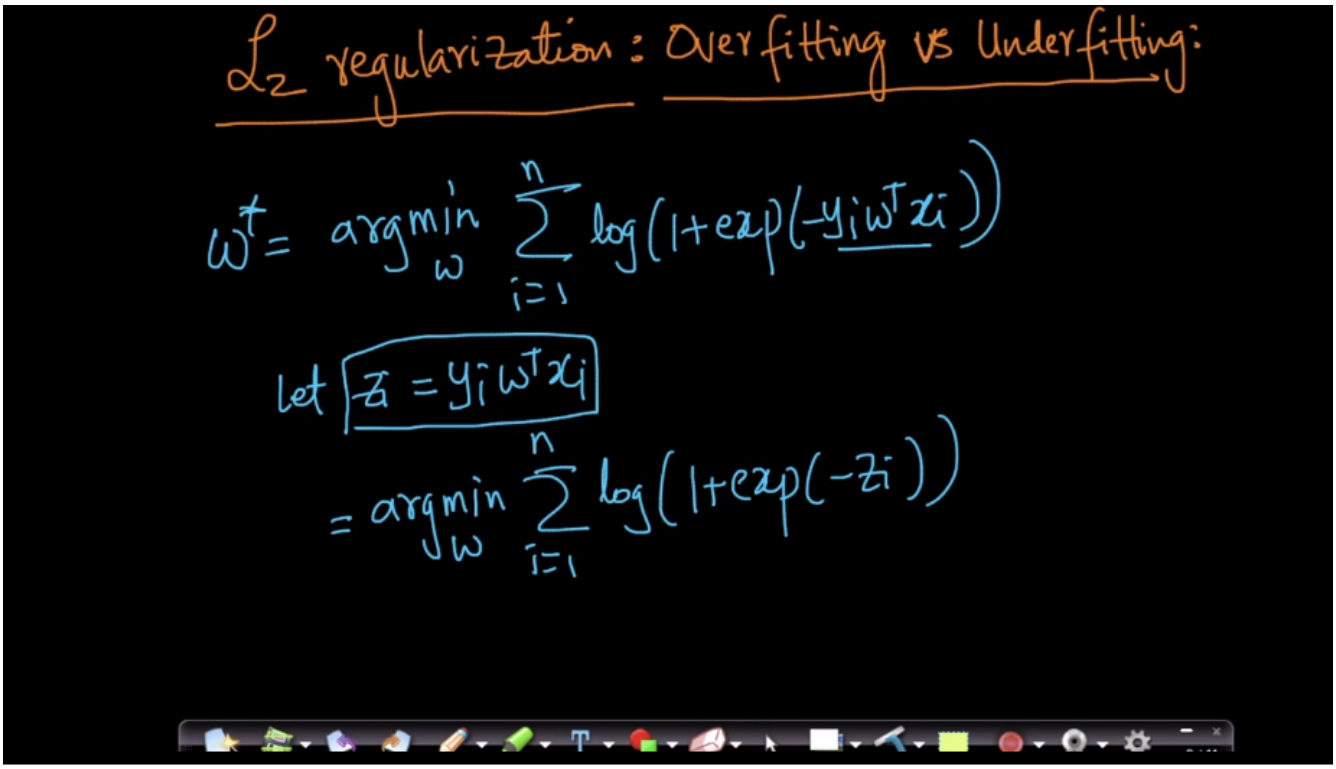


Case – 2:
if **wi** is negative the then **wi*xq** also decrease and the probability of **yq** being 1 also decreases, which implies **yq** being -1 also increases.

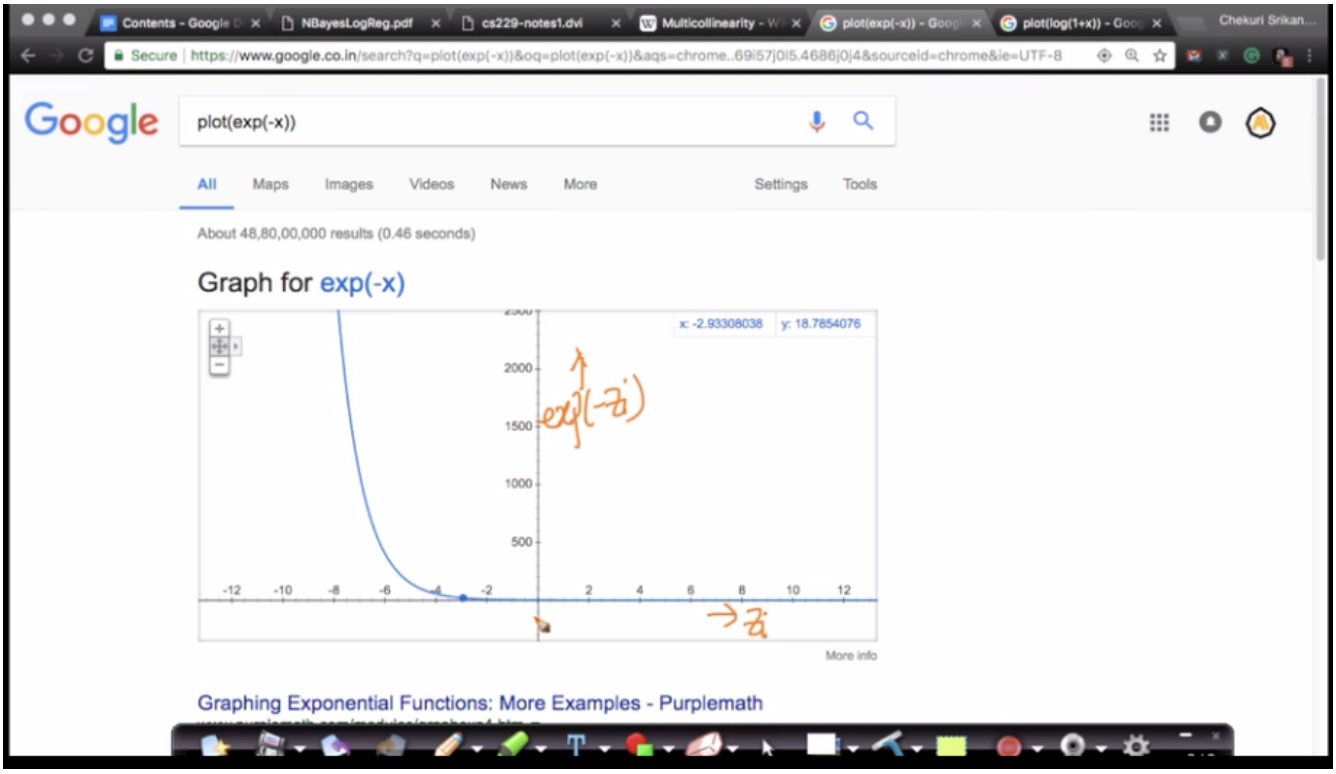


L2 Regularization: Over fitting and Under fitting:

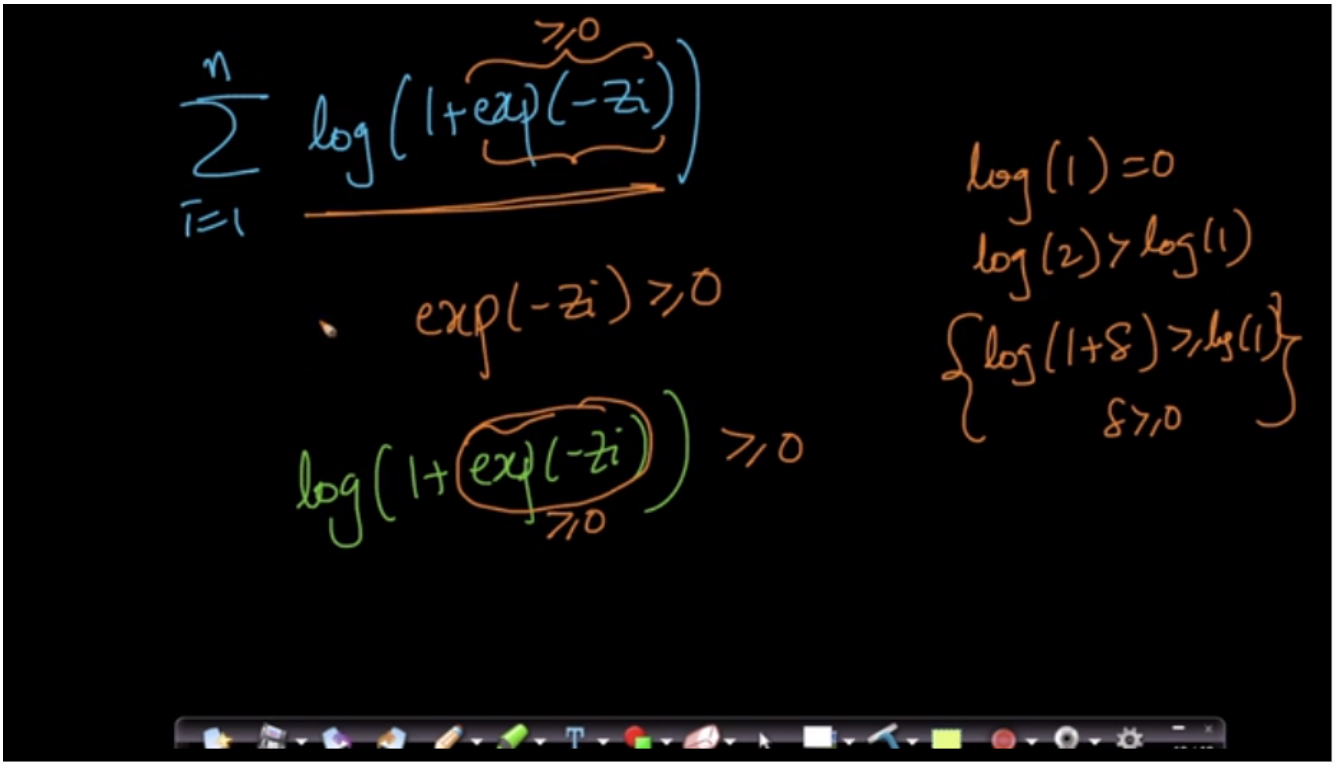
Z is the signed distance.



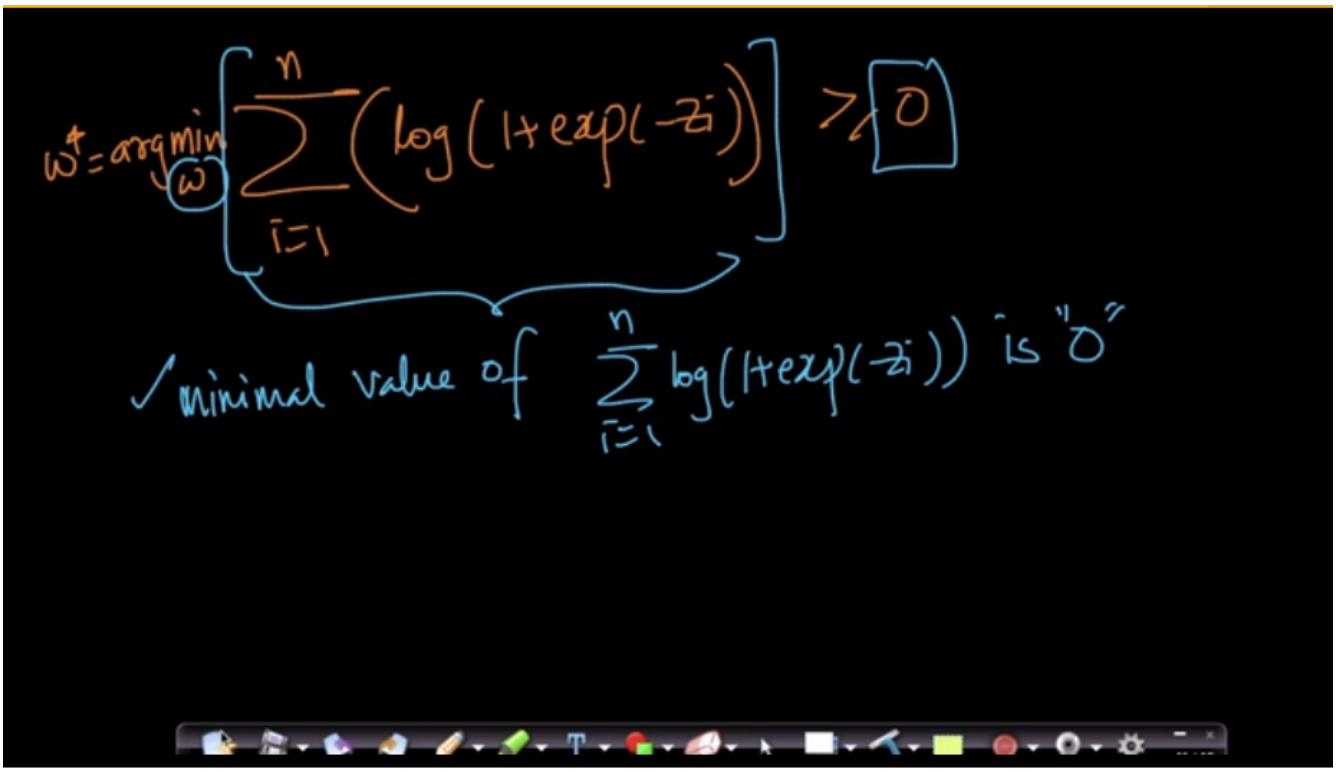
In the Objective function, exp(-zi) is always positive.



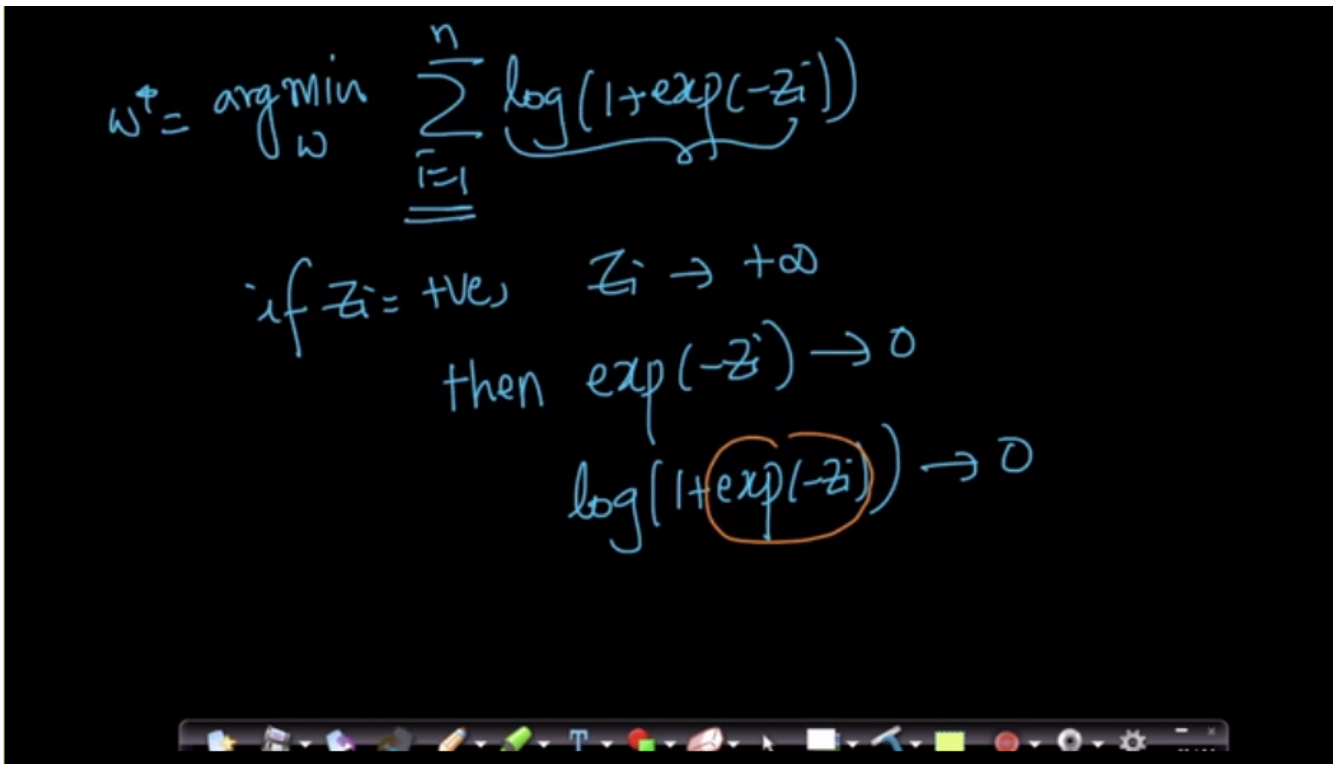
Interpretation of different functions.



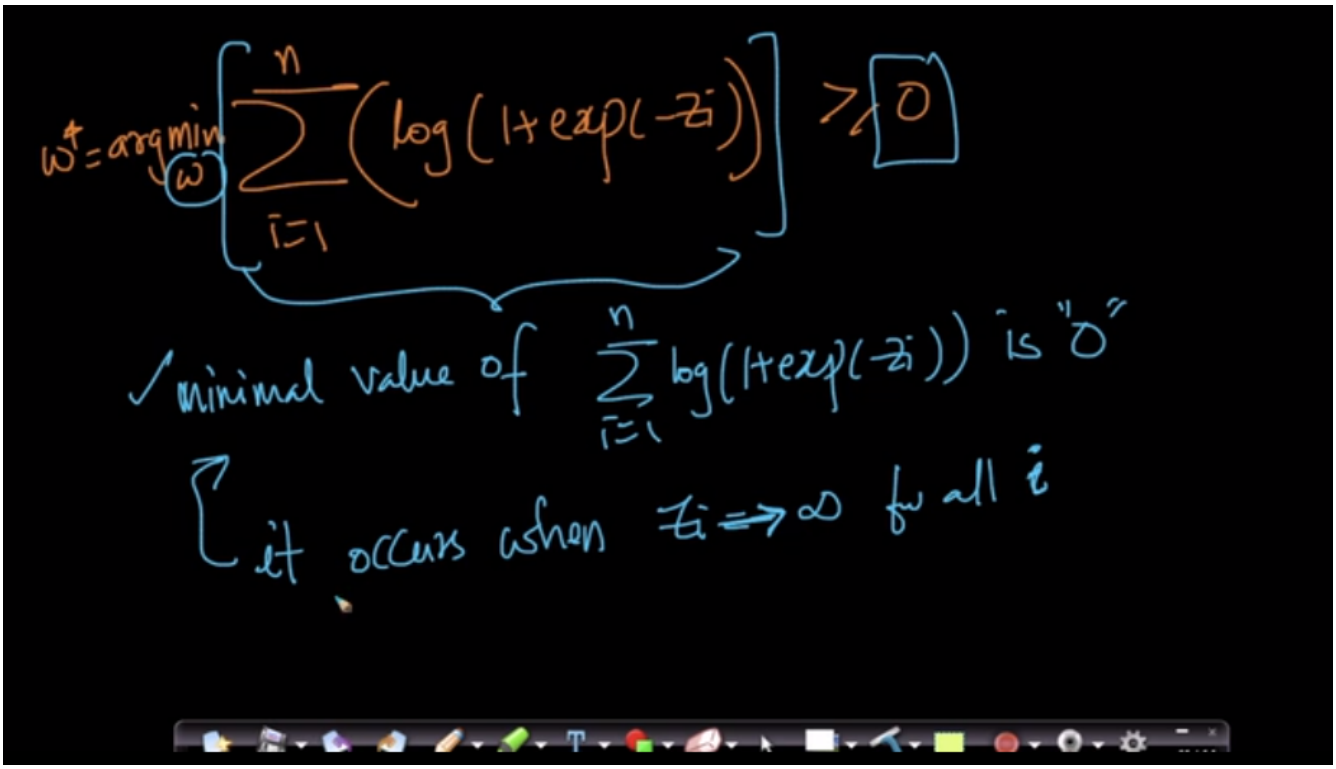
The minimal value of the function occurs at Zero.



If zi tends towards infinite then exp(-zi) tends towards 0

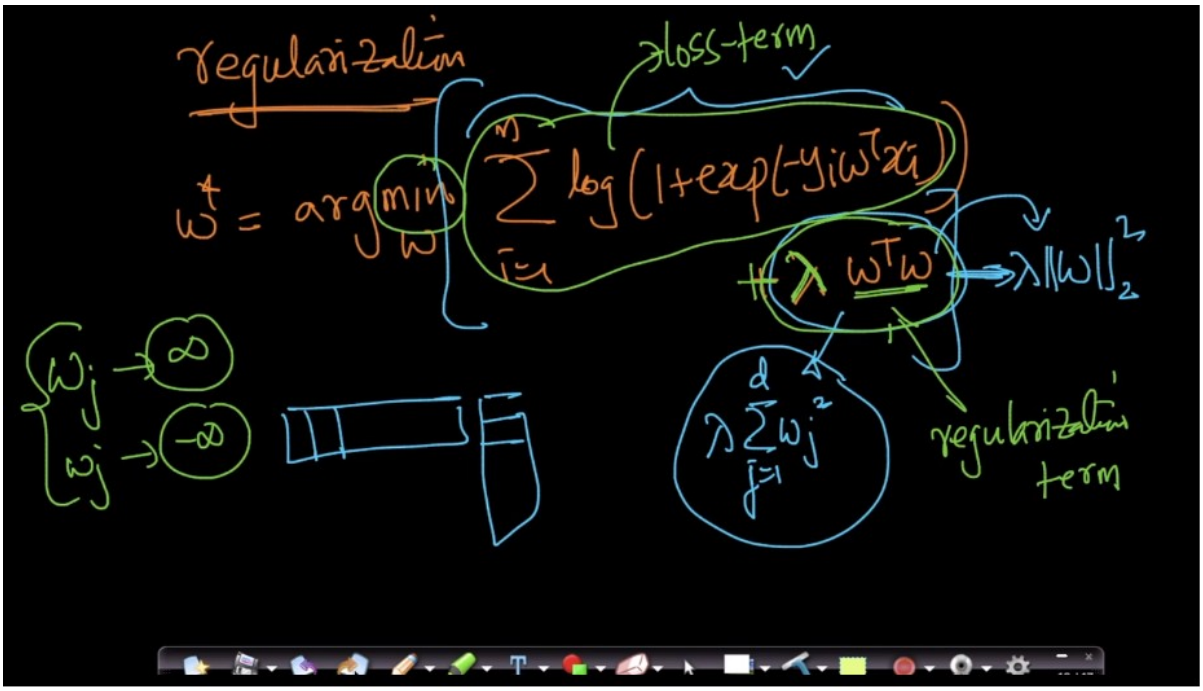


The minimal value of the function is zero when zi tends to infinite.

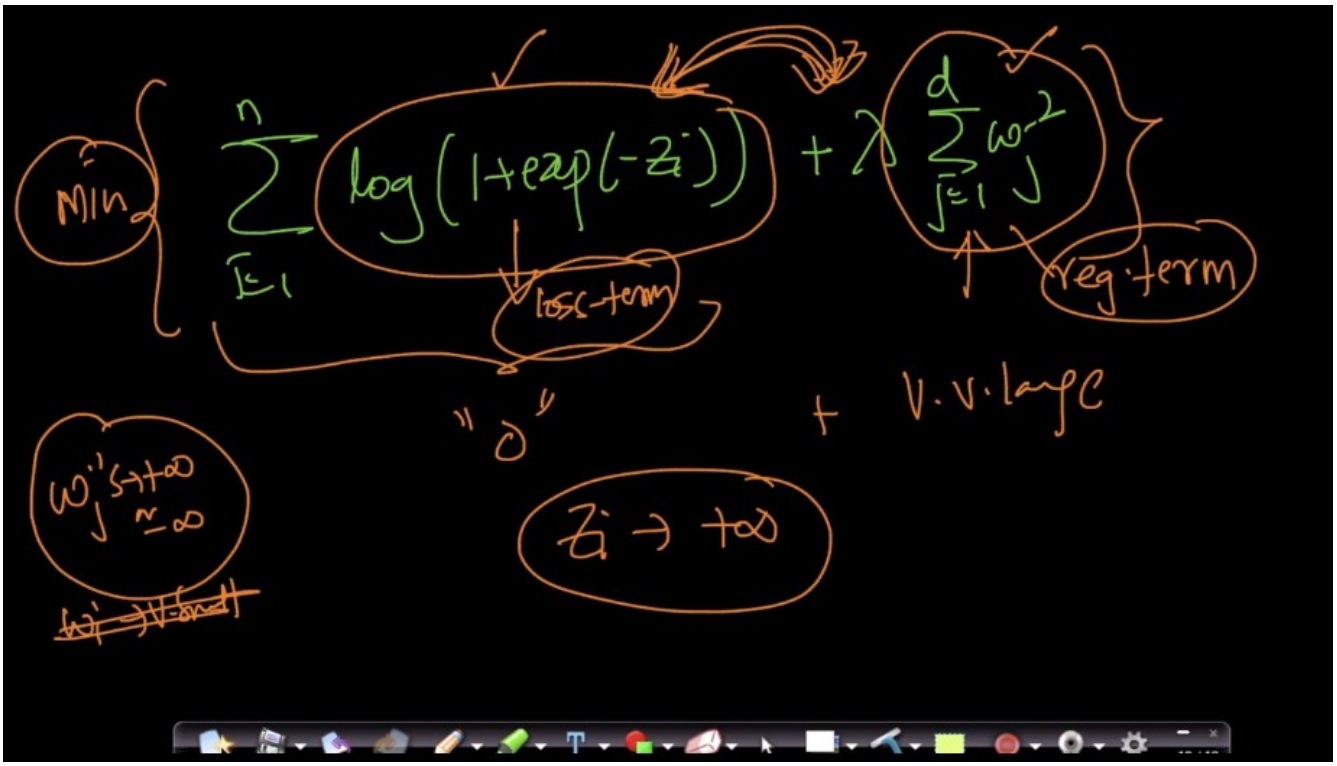


As the function above is monotonic function, the solution for the optimization problem will be wi is either +infinite or -infinite and over fit to the data.

So, in order to overcome the problem we use the regularization term for making the objective function not equal to zero.



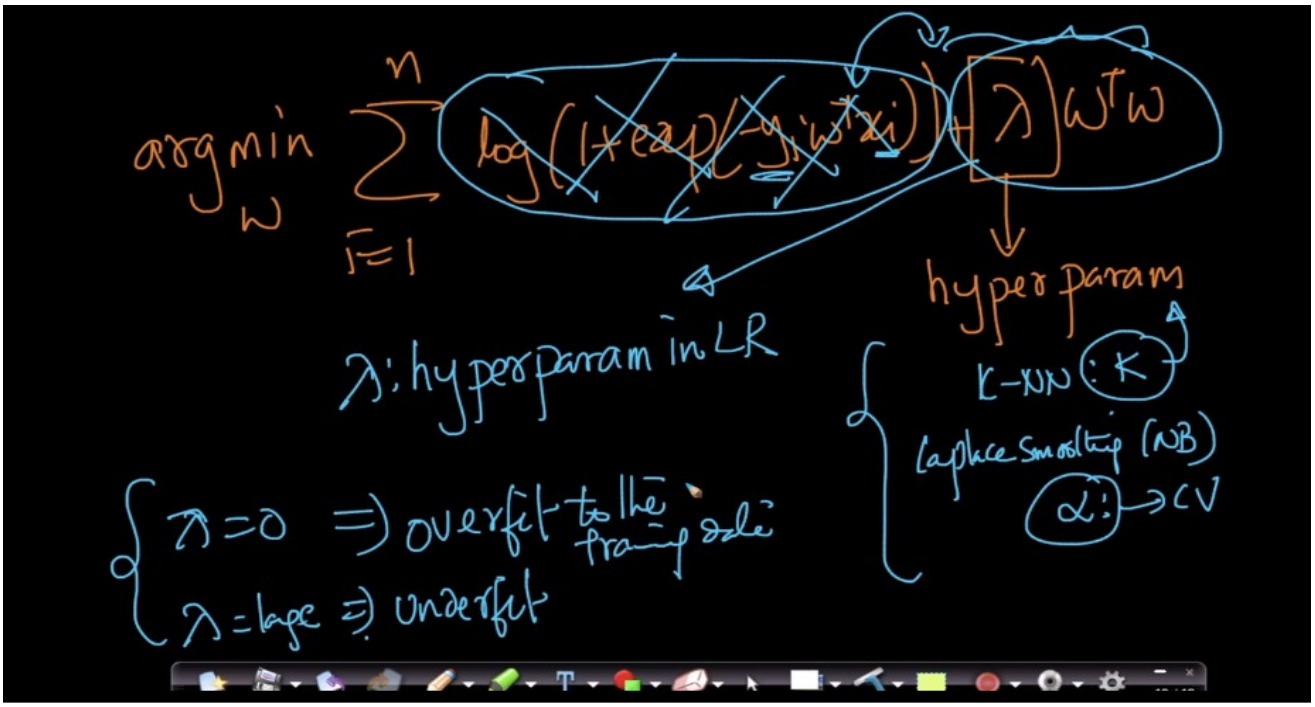
The two terms in the function tries to attain the right value of the objective function.



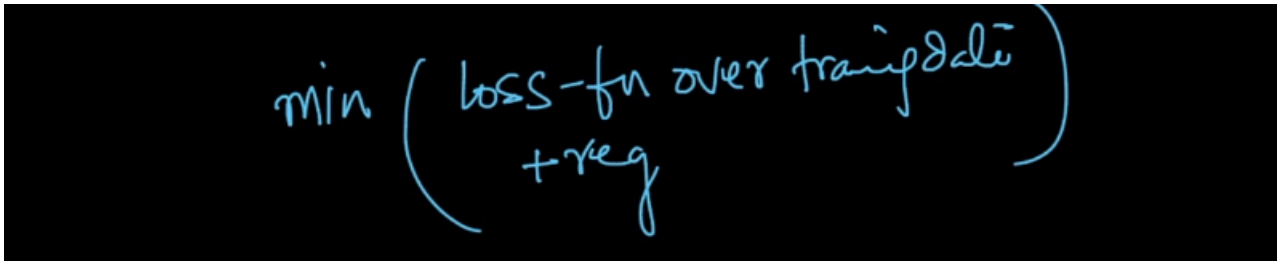
Lambda is the hyper parameter in Logistic regression.

If lambda = 0, then the model over fit the training data.

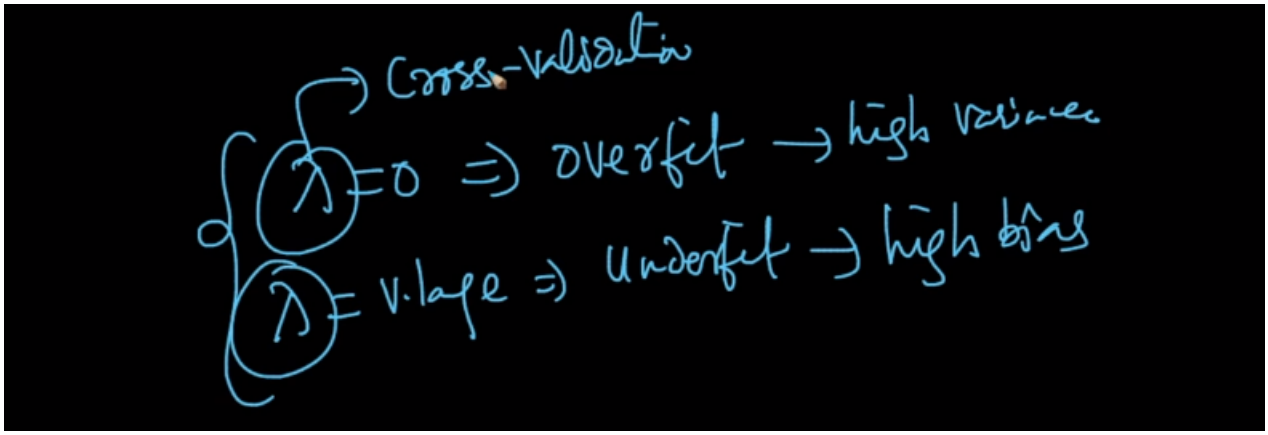
If lambda = large then the model will under fit the data.



The general equation of optimization function in Machine Learning will be.



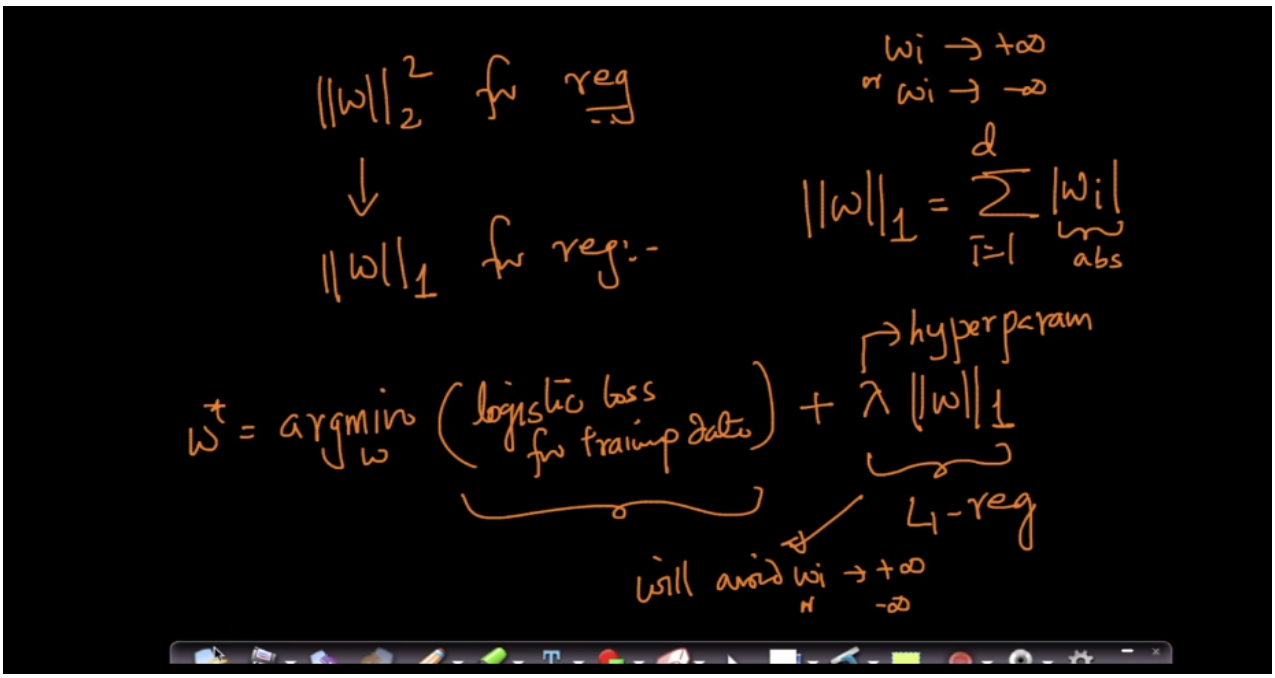
Right lambda is determined using cross – validation.



L1 regularization and sparsity:

L1 regularization is the alternate to L2 regularization.

L2 norm is nothing but the Manhattan distance.

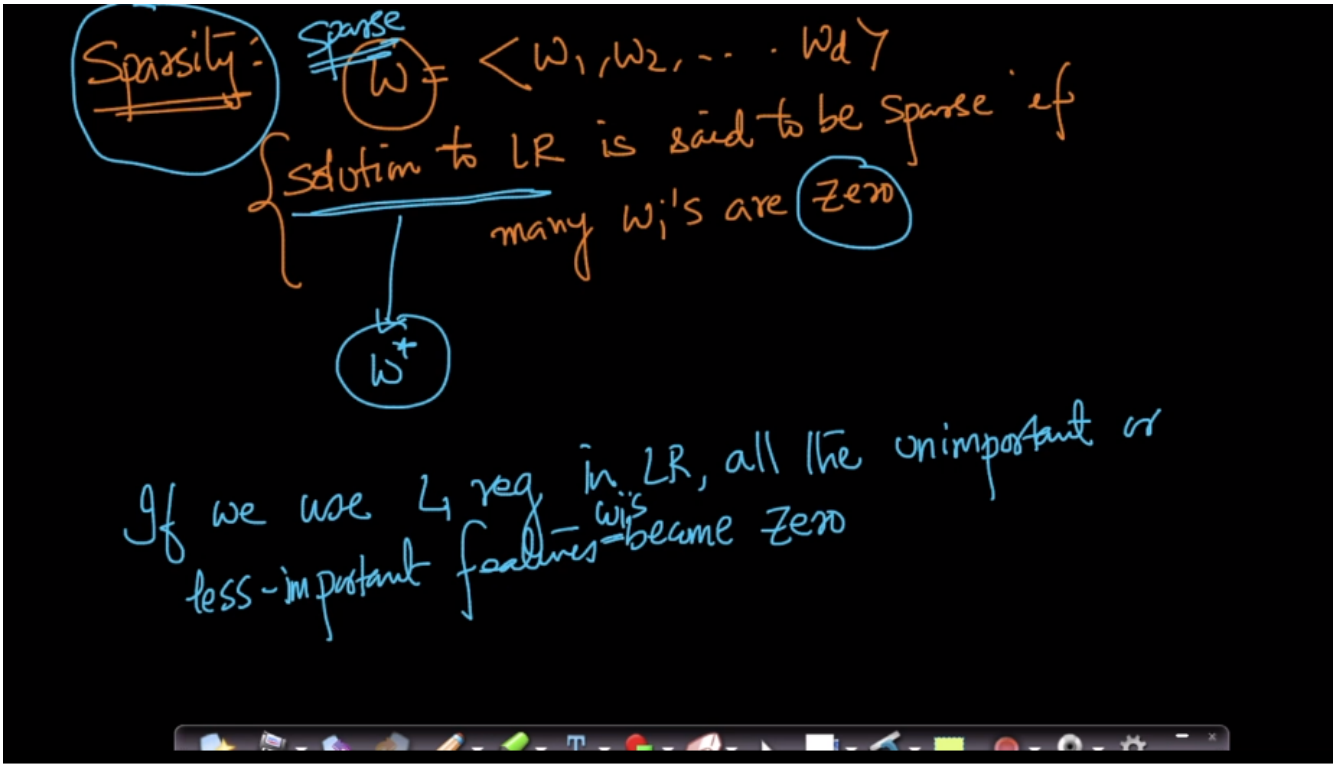


L1 regularization serves the same purpose of the L2 regularization. Since it uses the absolute value of W.

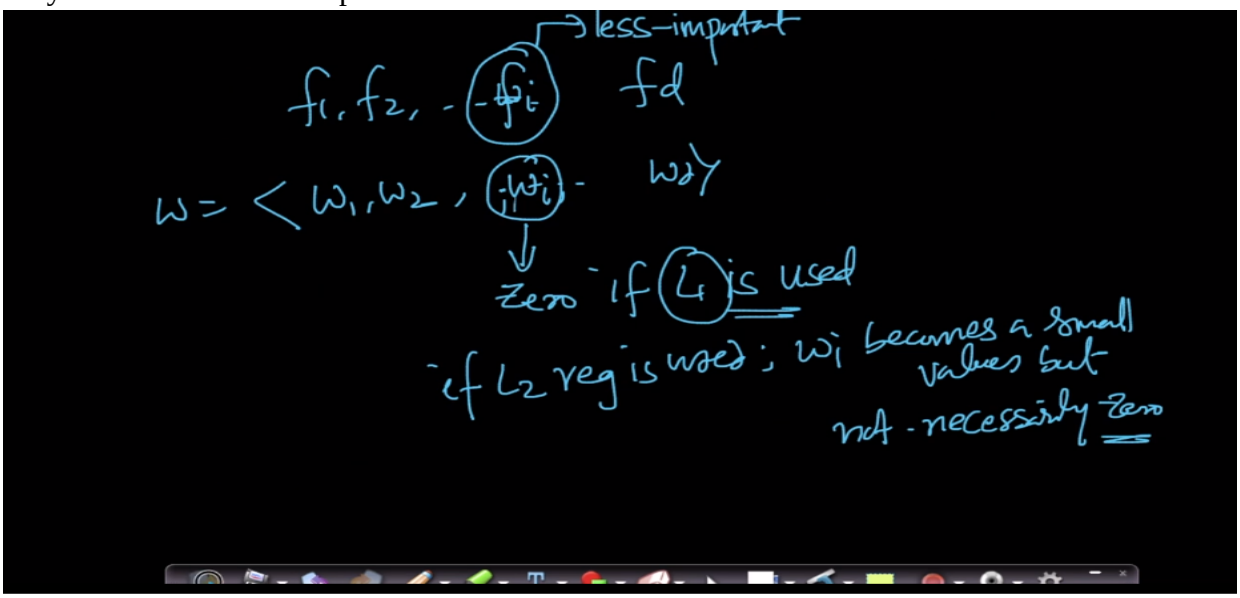
It has one disadvantage.

Sparsity- The vector is said to be sparse if many of the values of the vector are zeros.

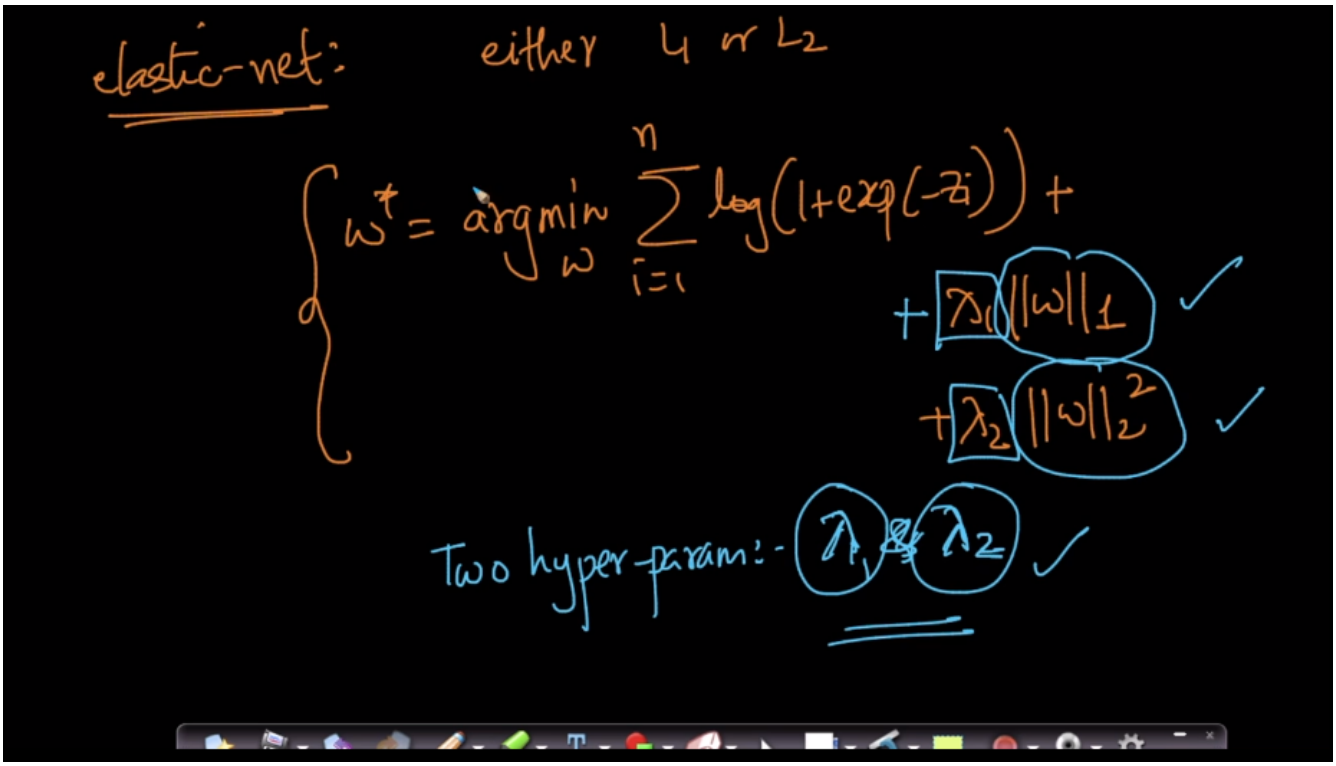
If we use L1 regularization all the less important features become Zeros.



If any of the feature less important the wi value for the feature becomes Zero.

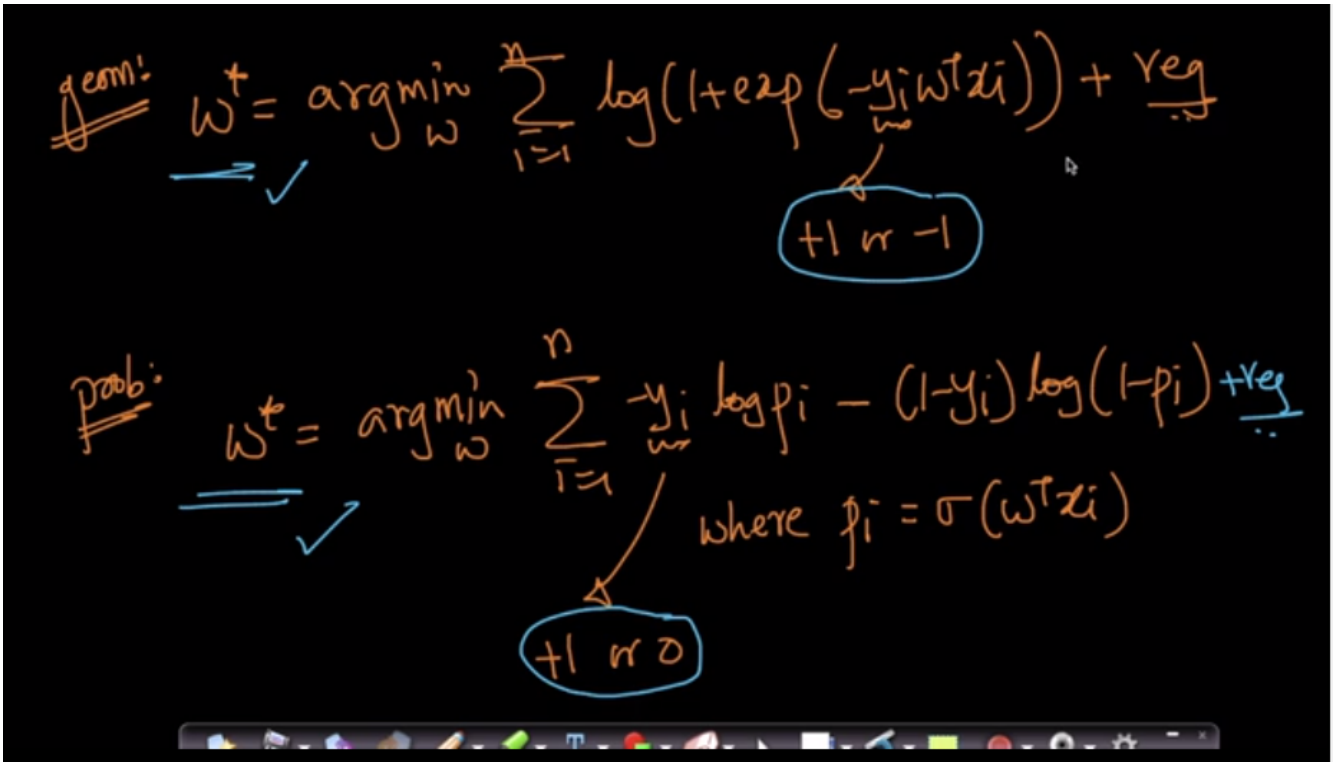


There is elastic – net implementation of regularization in which they take both L1 and L2 norm.
In this case we have two hyper parameters.



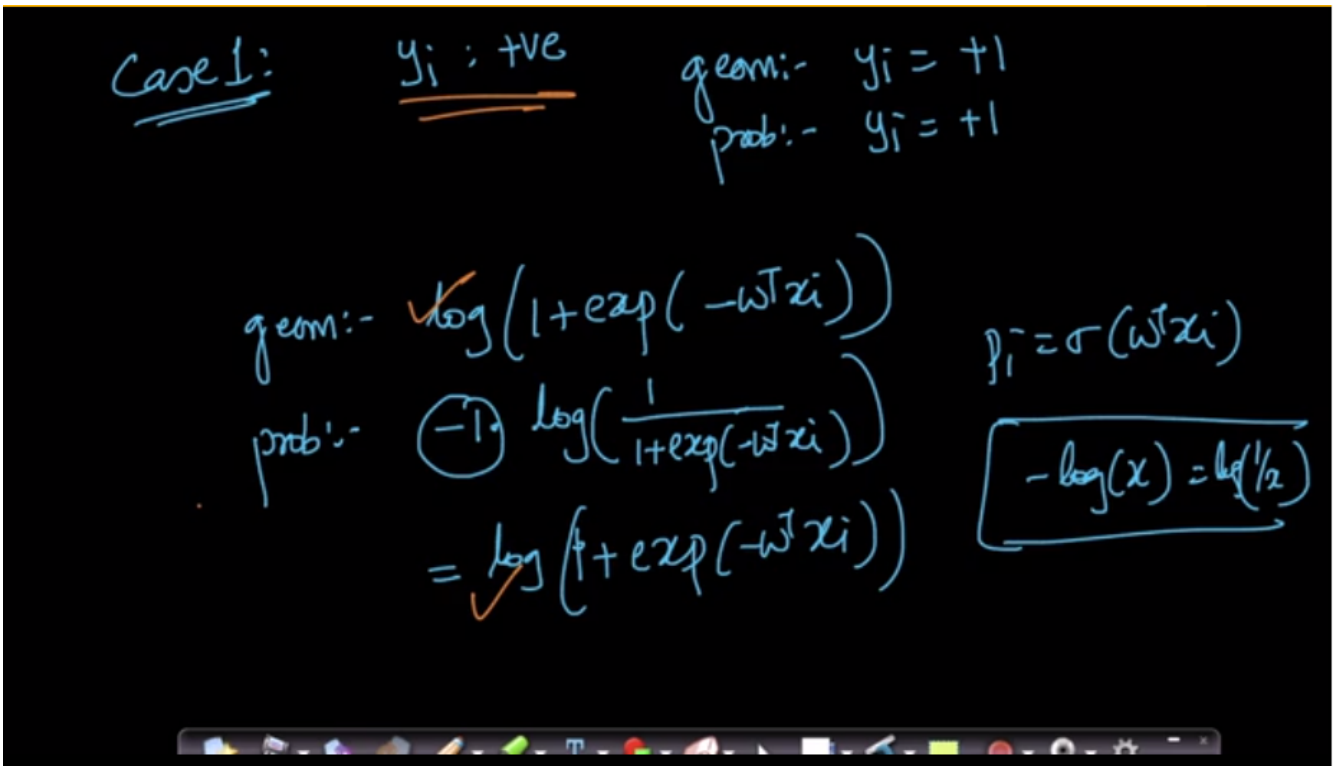
Probabilistic Interpretation: Gaussian Naive Bayes for logistic regression:

The random variable for coin tosses can be used as Bernoulli random variable.

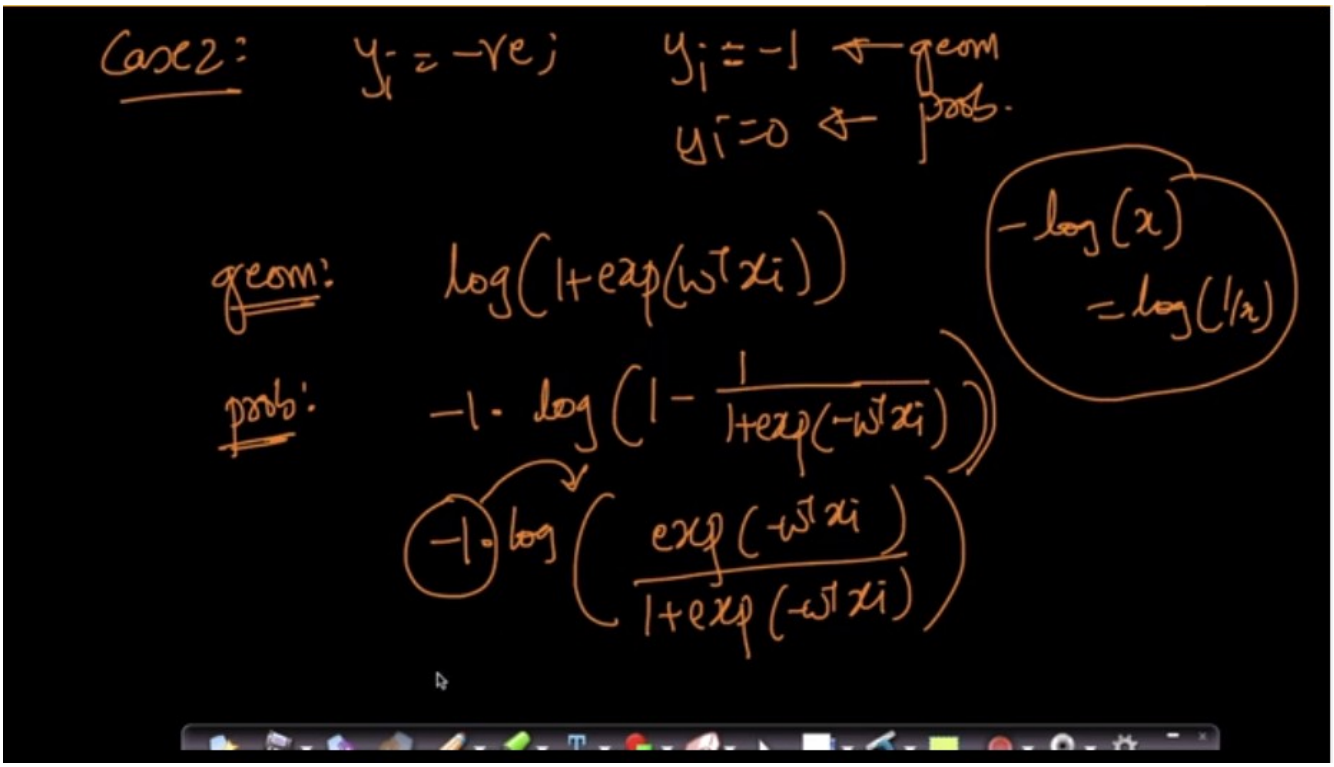


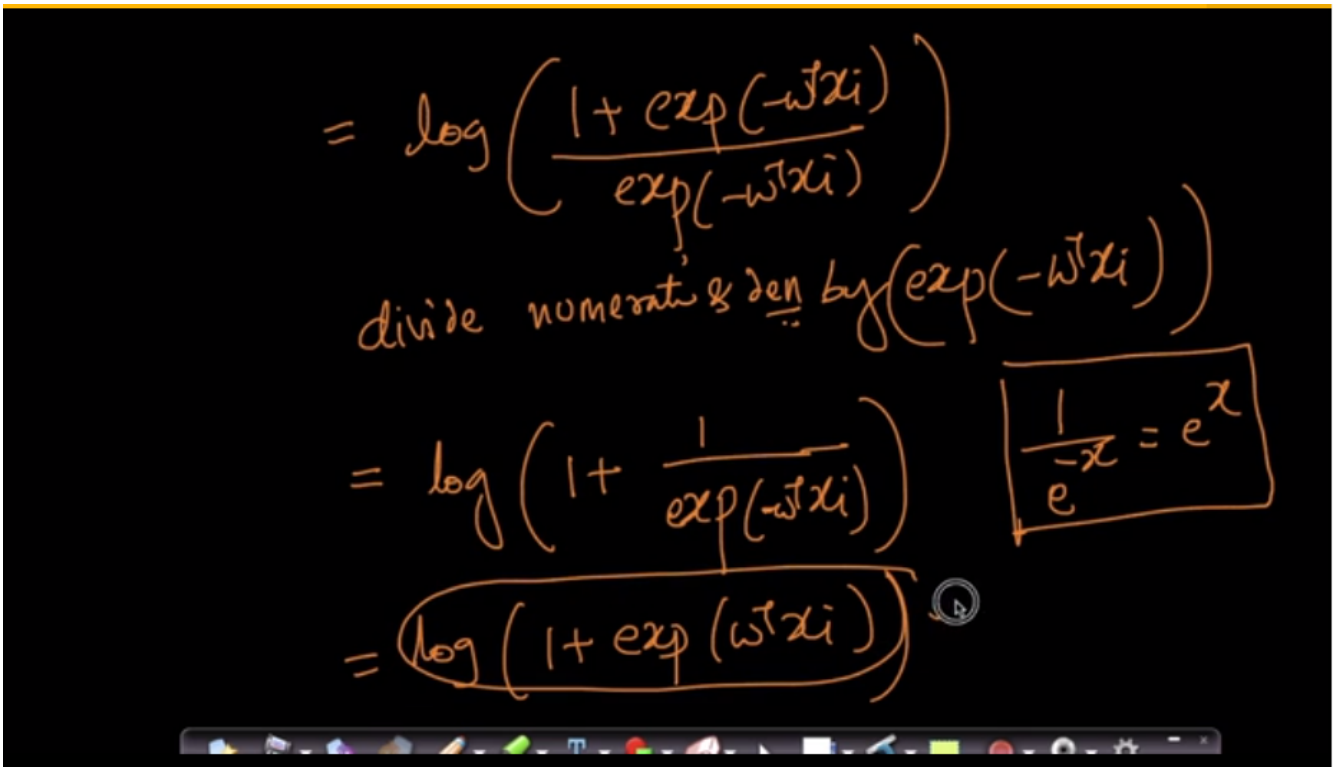
The both formulations are one and the same.

When y1 is positive.



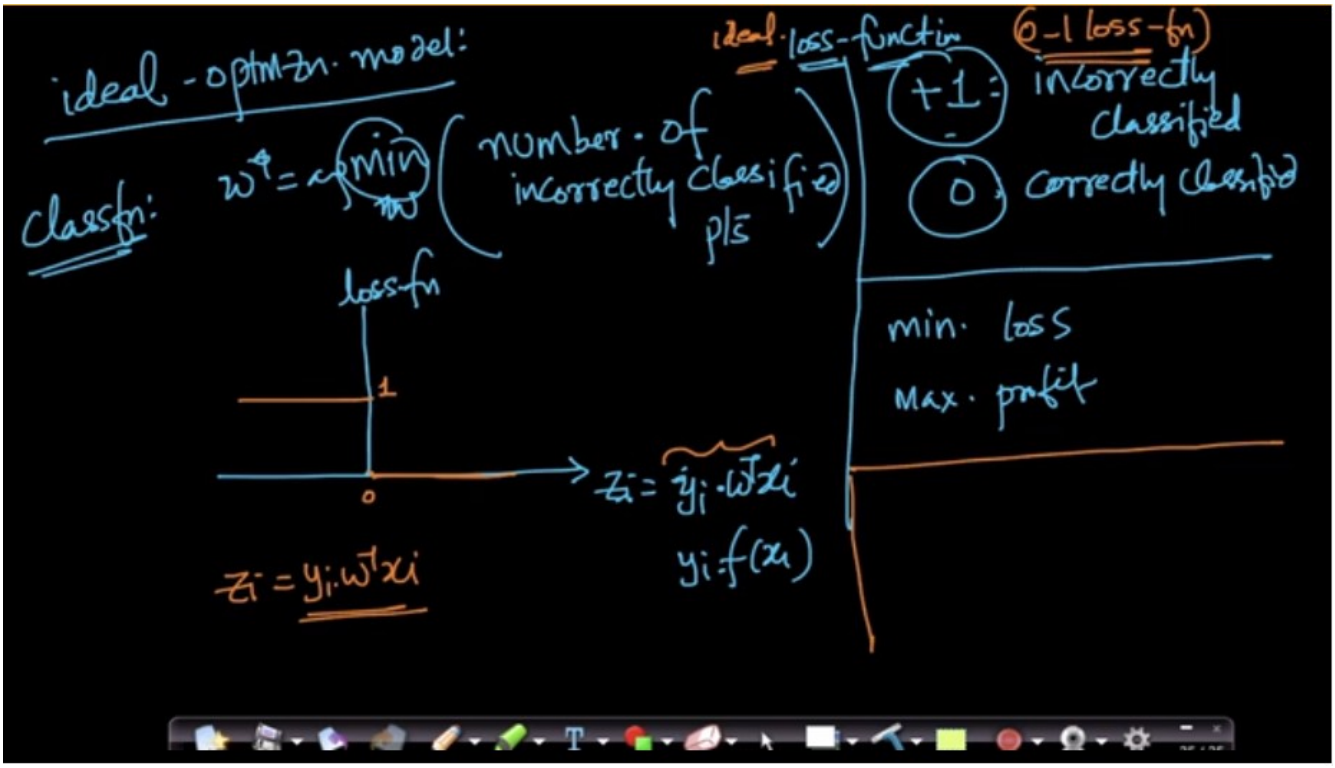
When yi is negative.



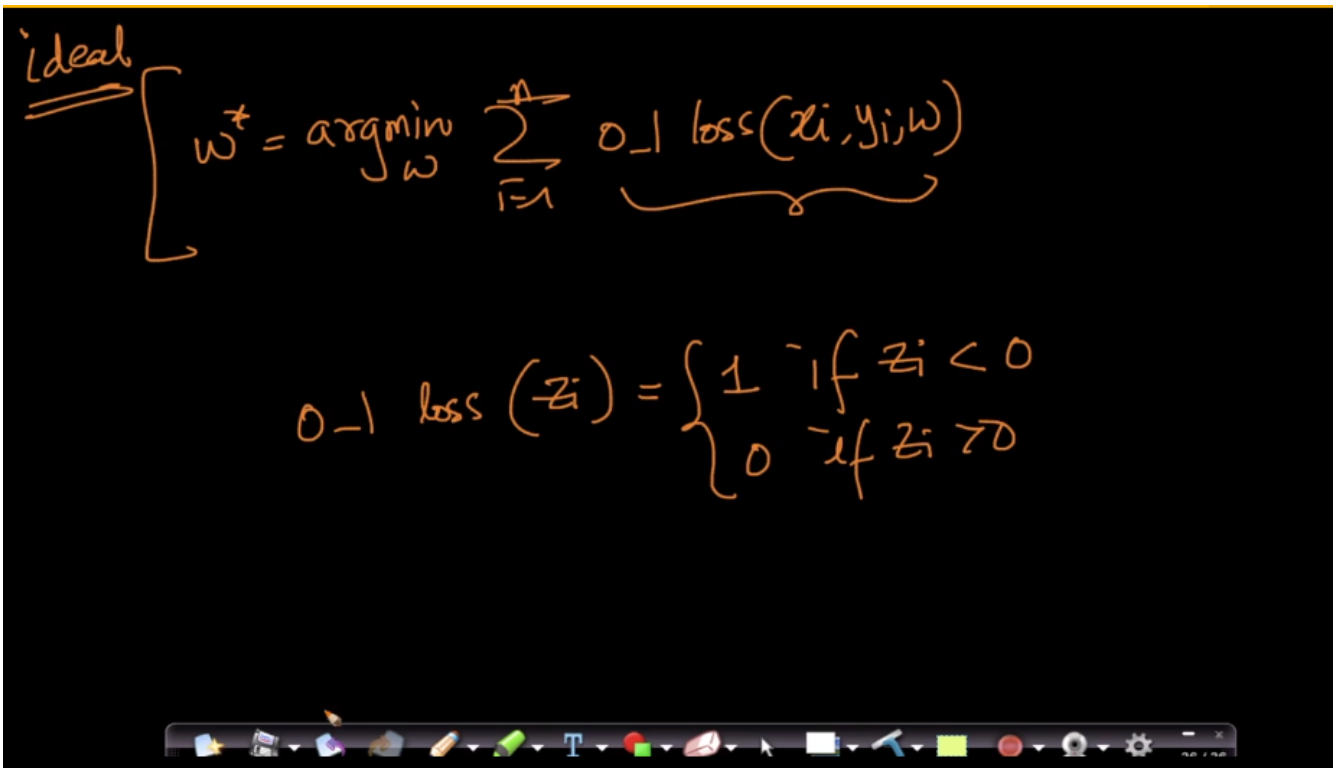


Loss minimization interpretation:

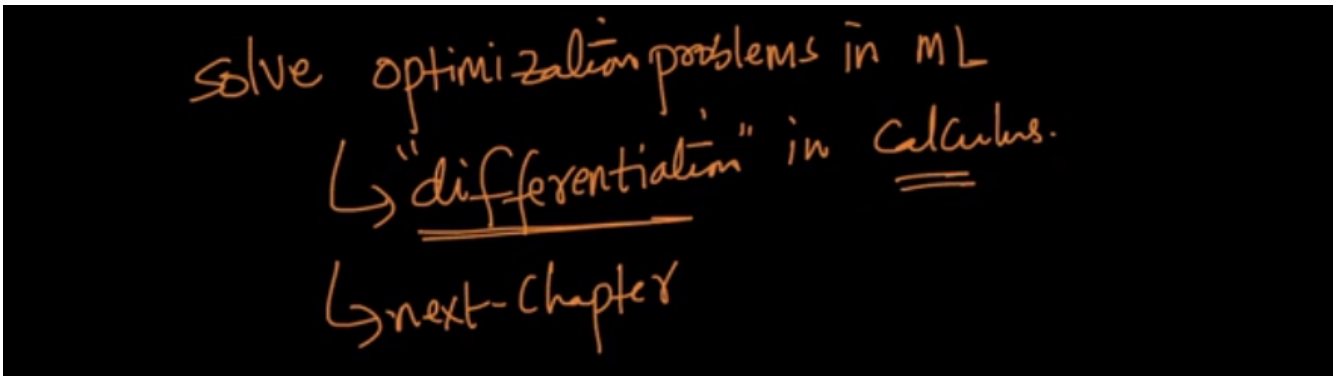
An ideal optimization model can be found by minimizing the miss-classification.



Mathematical formulation



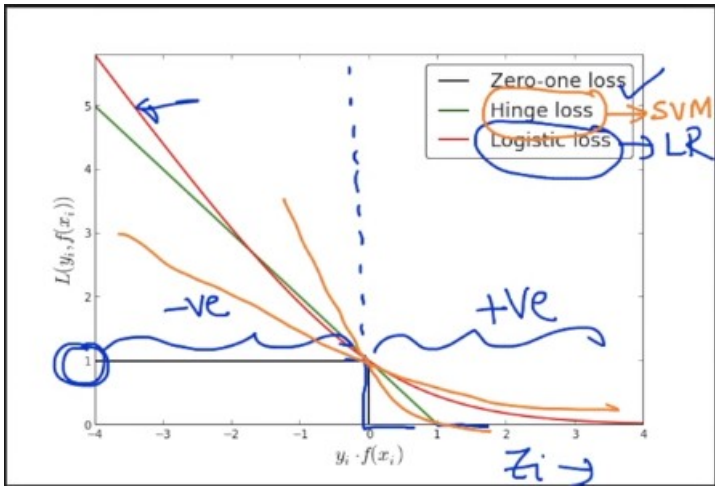
To solve optimization problems in machine learning we use calculus.



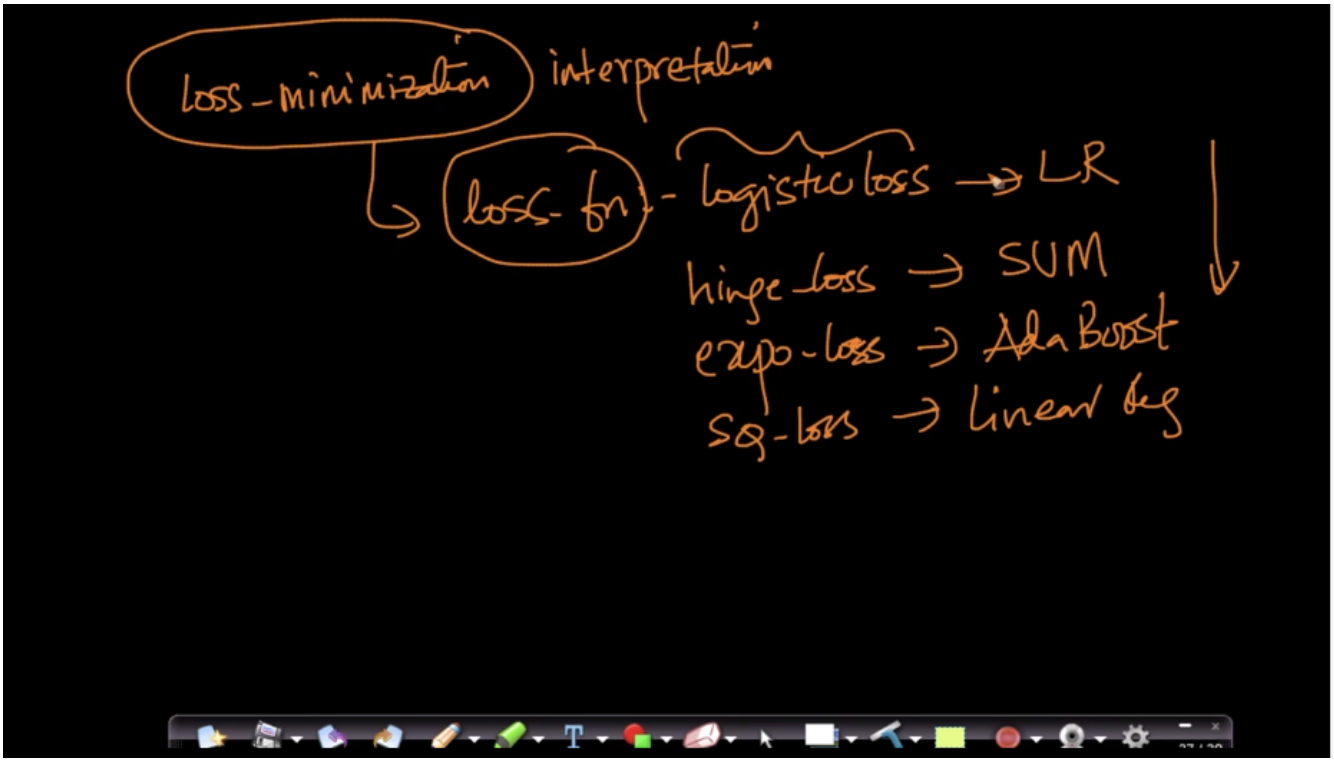
loss function is also called as 0 – 1 loss.

We need continuous functions to differentiate the loss function. Since, we cannot differentiate the 0-1 loss we will use approximate to obtain **logistic loss**.

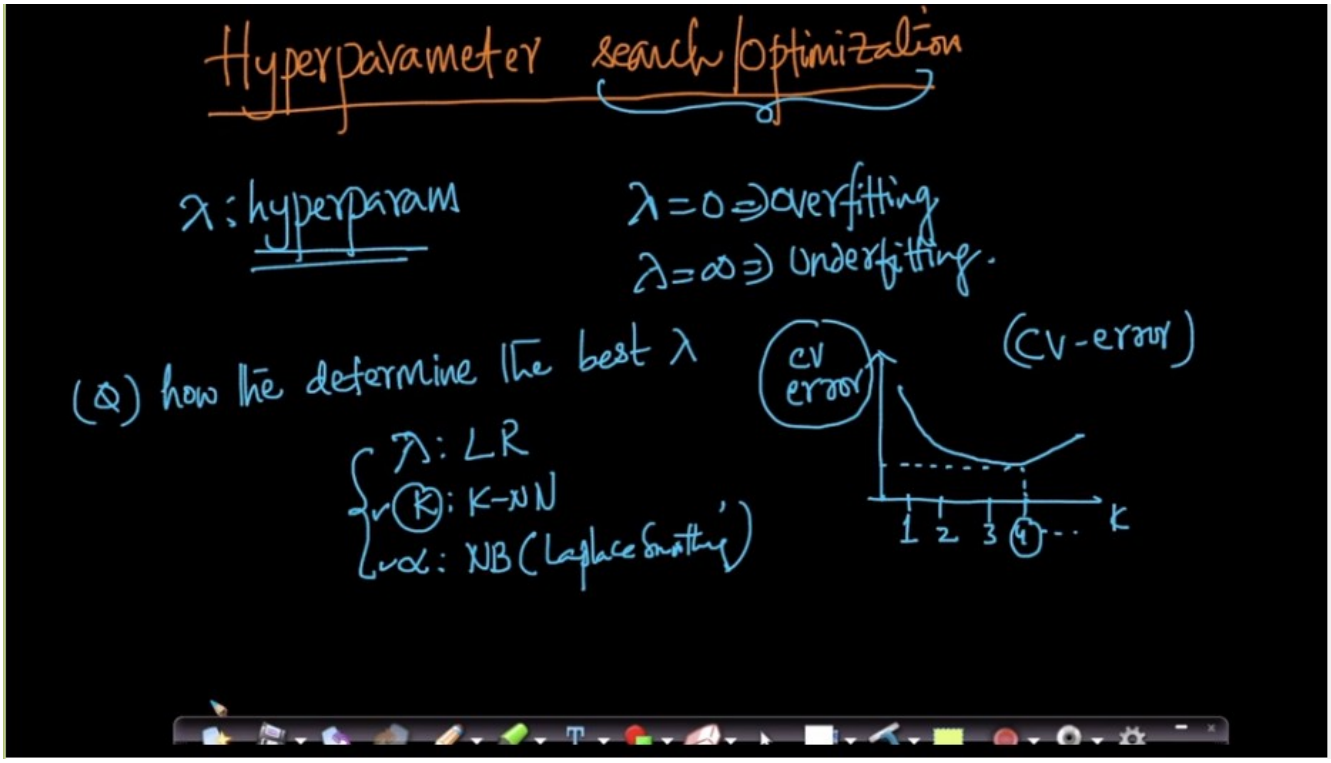
The second form of approximate the loss is **hinge loss**.



Various loss function minimization leads to multiple machine learning algorithms.

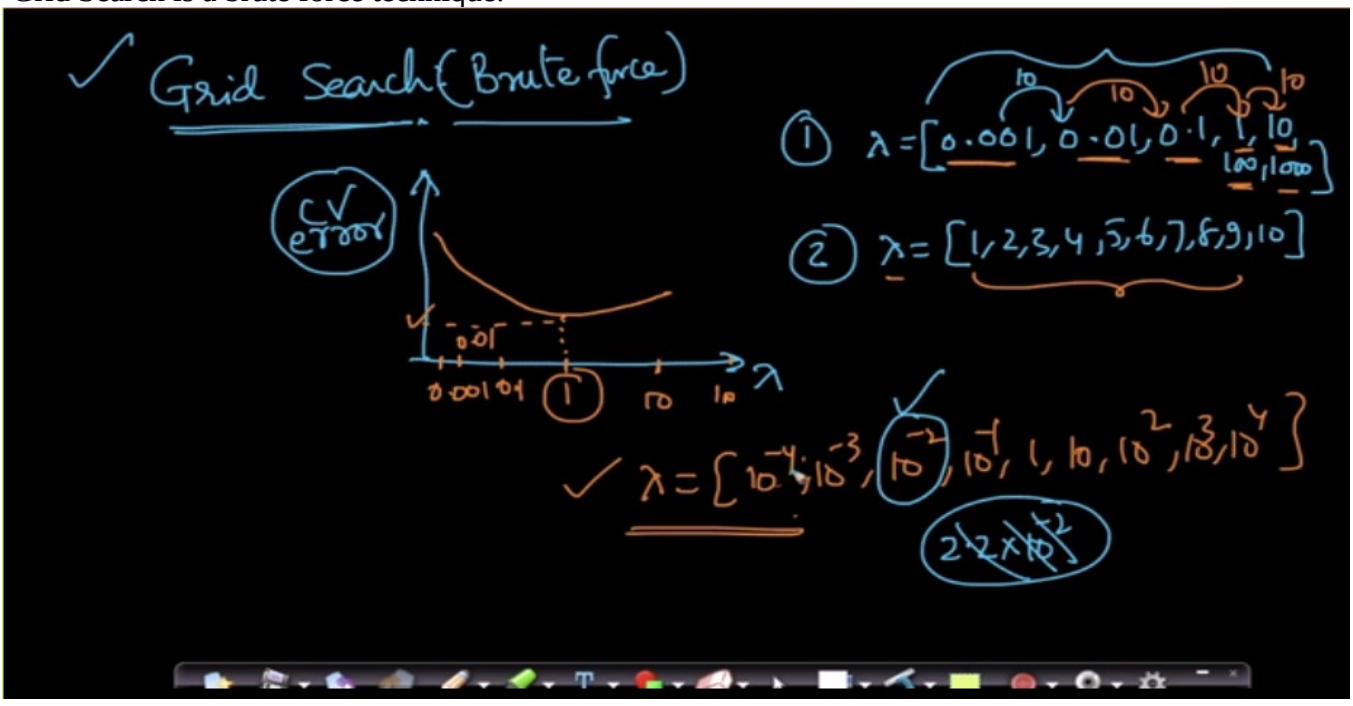


Hyper parameter search: Grid Search and Random Search

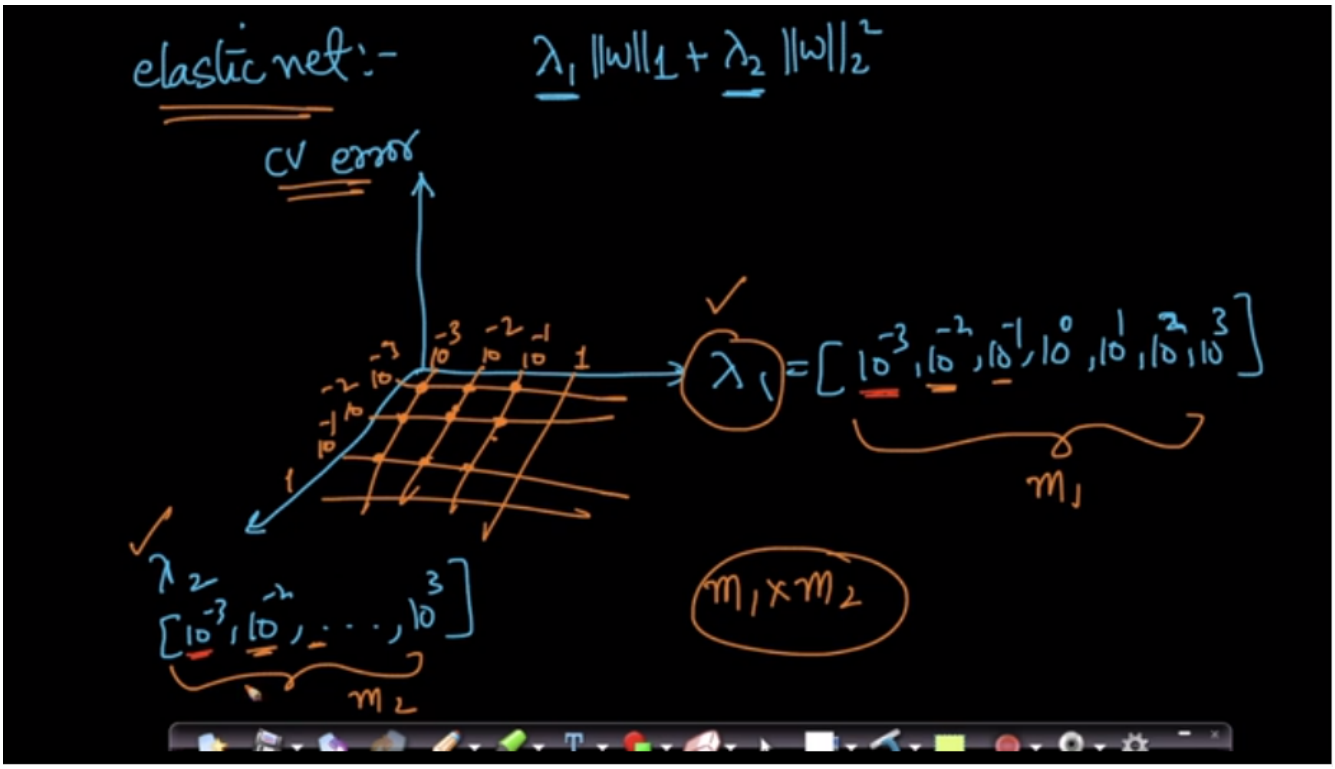


K in KNN is an integer {1,2,3,4,5, .. ,n}
Lambda in the LR is a real number

Grid Search is the technique to find the Hyper parameters.
Grid Search is a brute force technique.

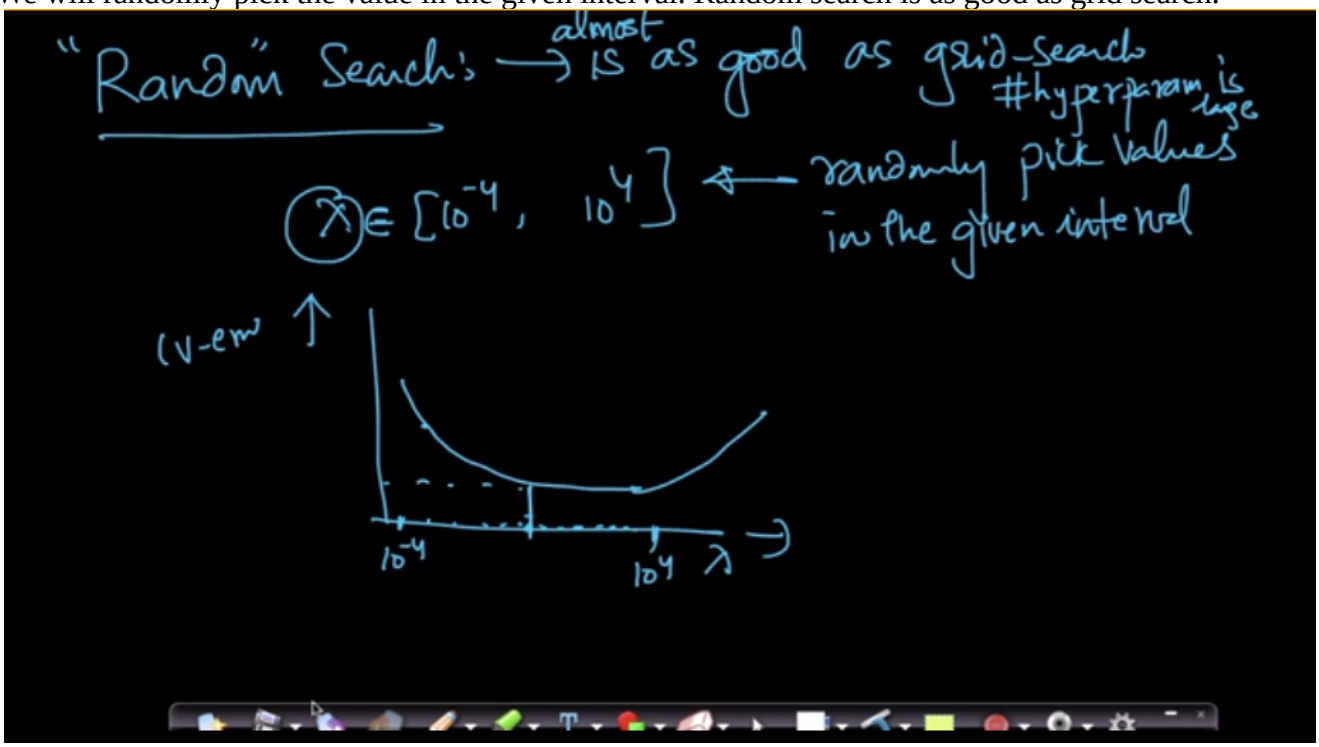


For elastic net hyper parameter choosing:

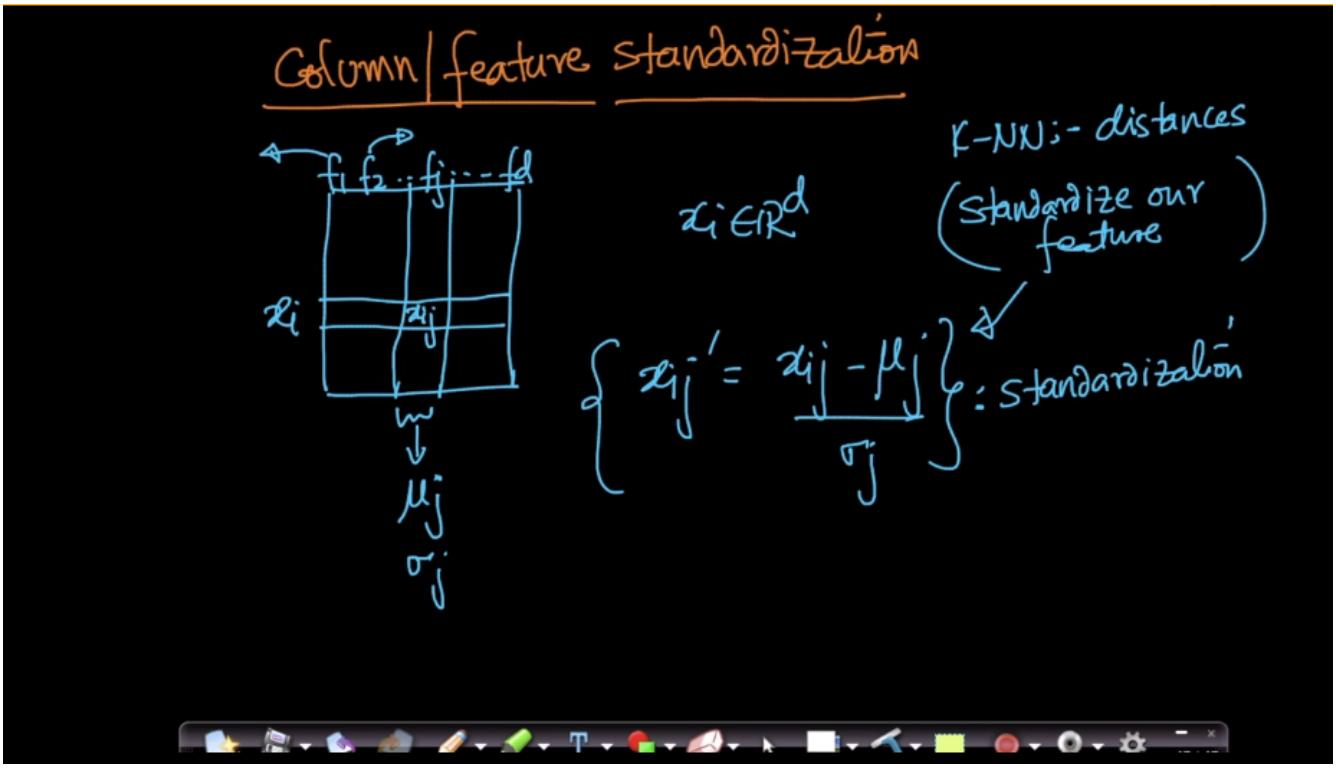


Grid search is not good in case of many hyper parameters.

There is an alternative to the grid search called random search.
We will randomly pick the value in the given interval. Random search is as good as grid search.



Column Standardization:

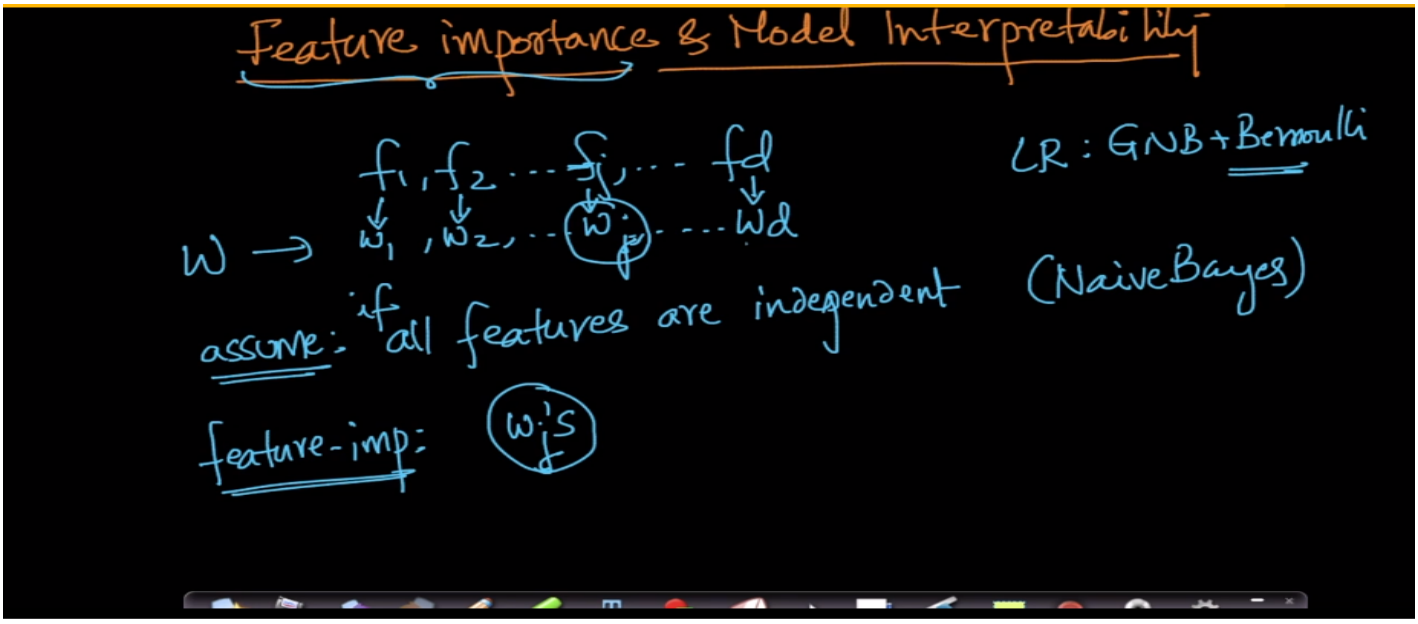


Even in Logistic regression it is **mandatory to perform feature (or) column standardization.**

Distances are impacted by the feature scales.

Feature importance and model interpret-ability:

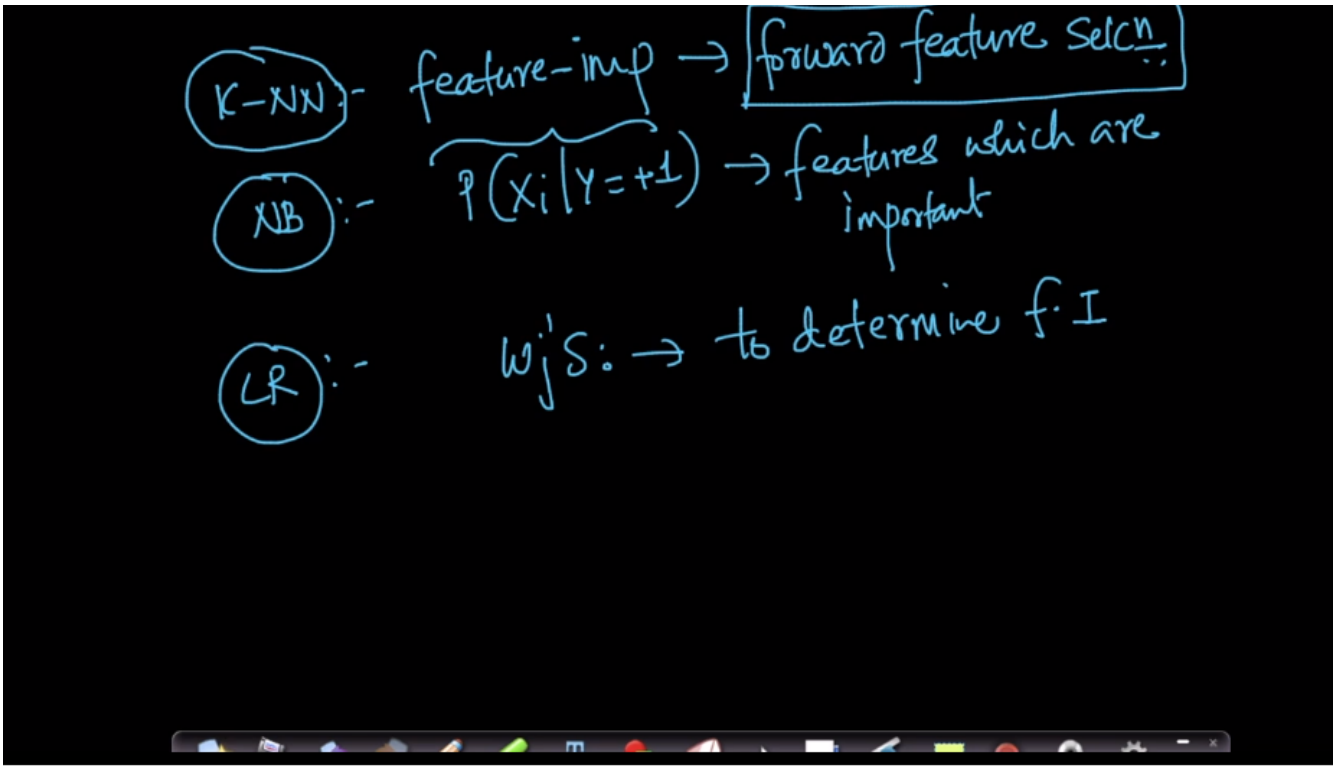
If all features are independent



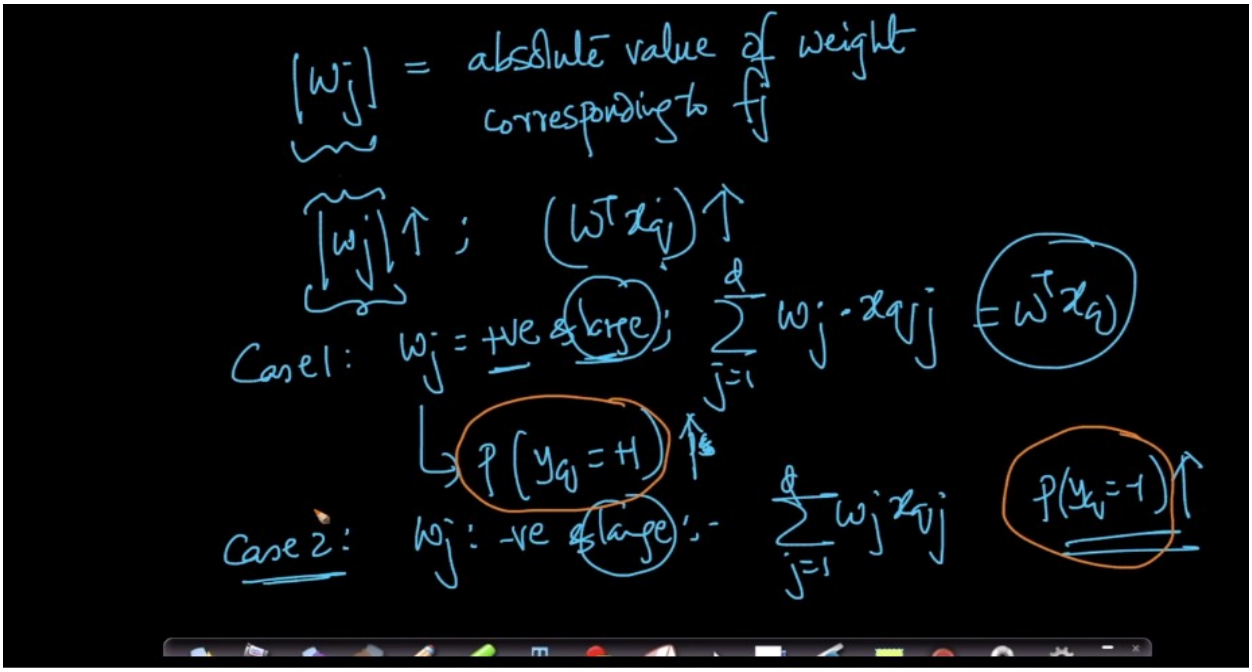
In K nearest neighbors feature importance can be found using forward feature selection.

In NB we can get the features importance by using the probability scores.

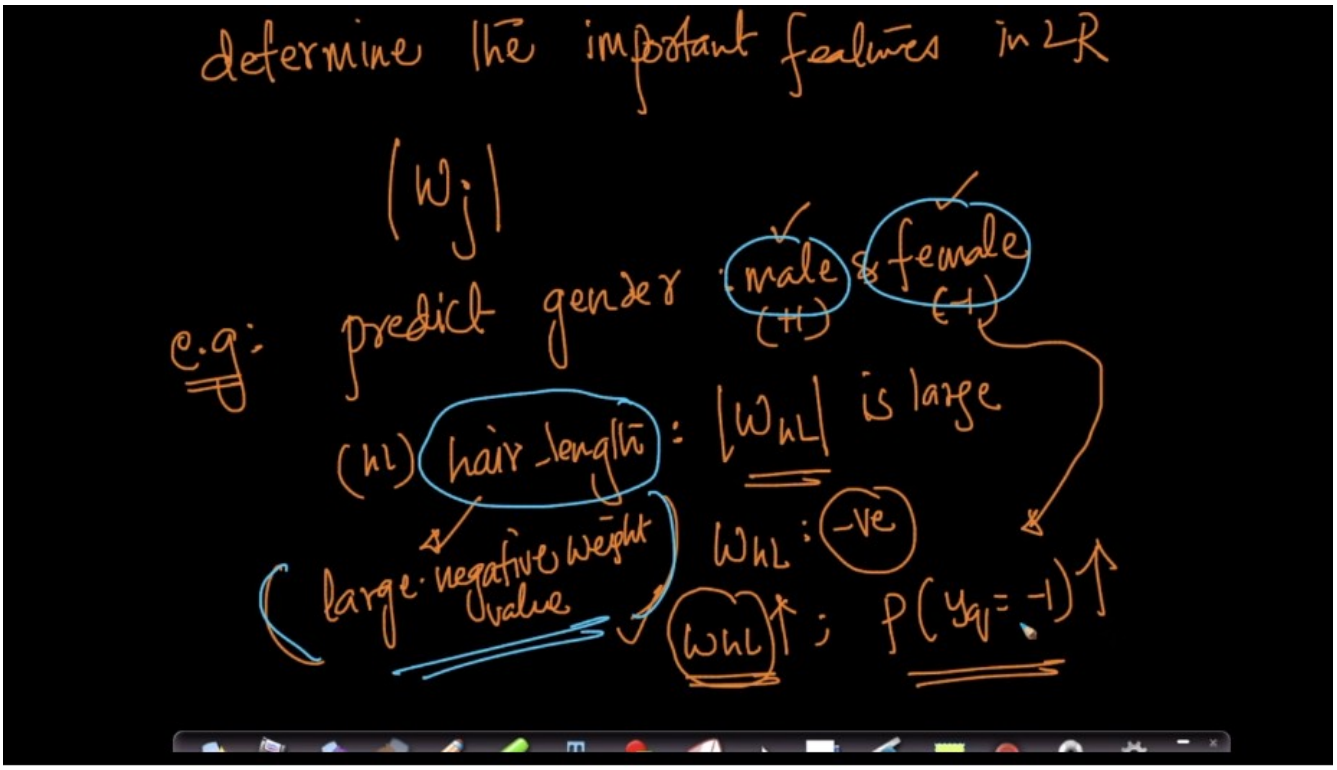
FFS is general technique.



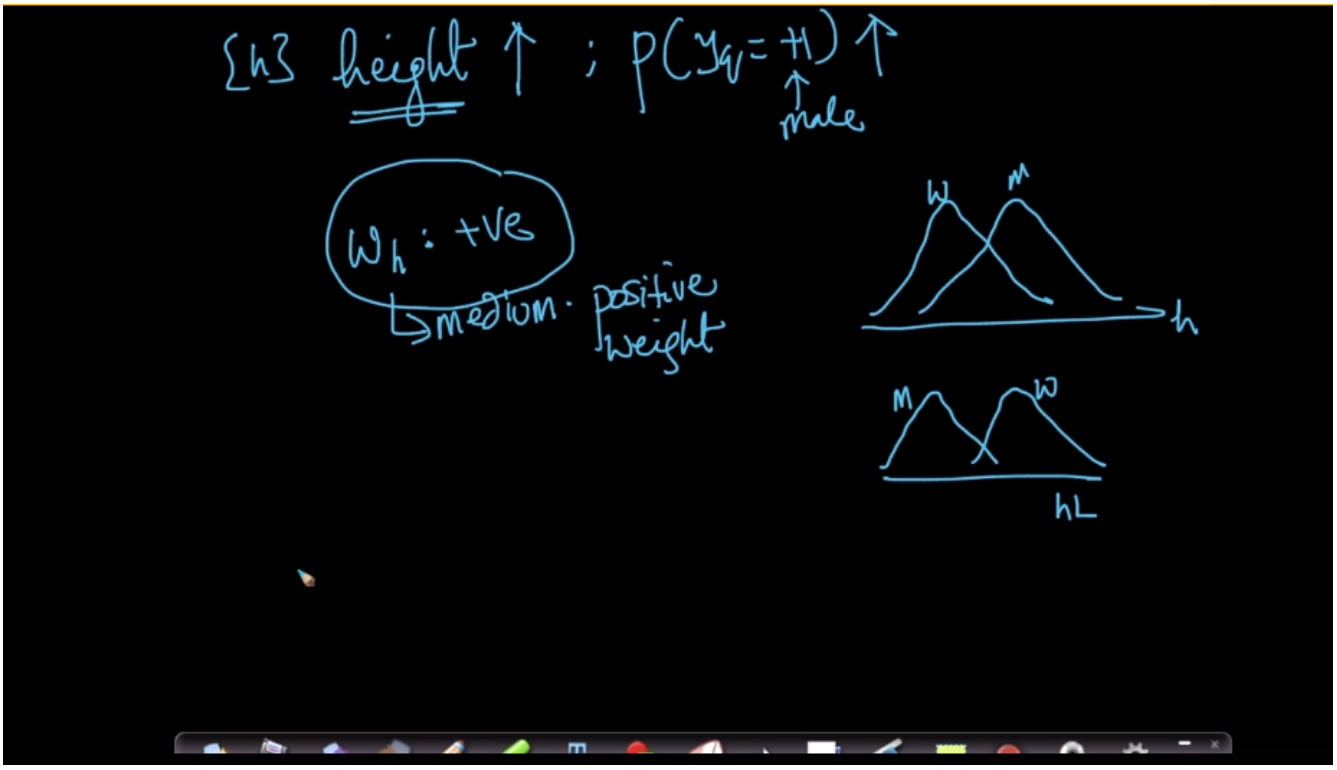
A feature can be found using the absolute value of the weight, the positive and negative values does not matter.



We can determine the important features by determining each of the feature weights.

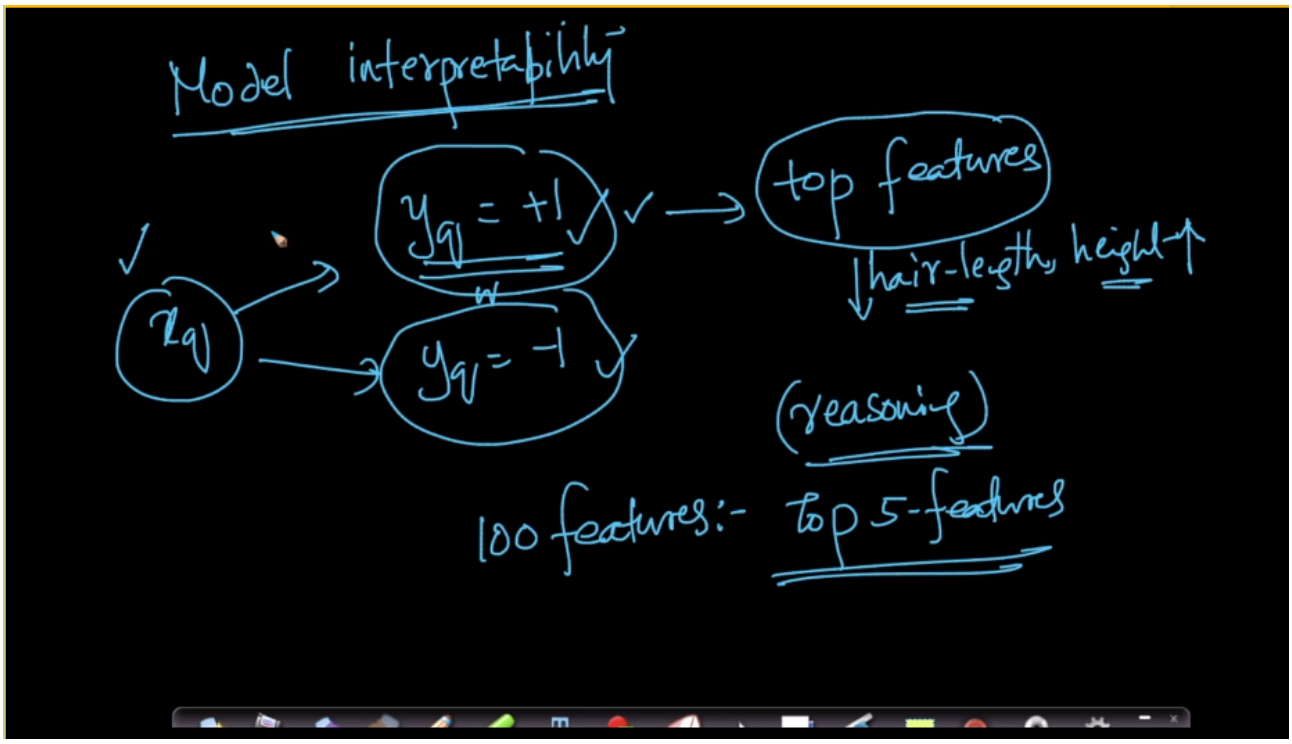


Females tend to have more hair length, negative weight will increase because the hair length of female is more.
For height we will get minimum weight to height.



Model interpreteability:

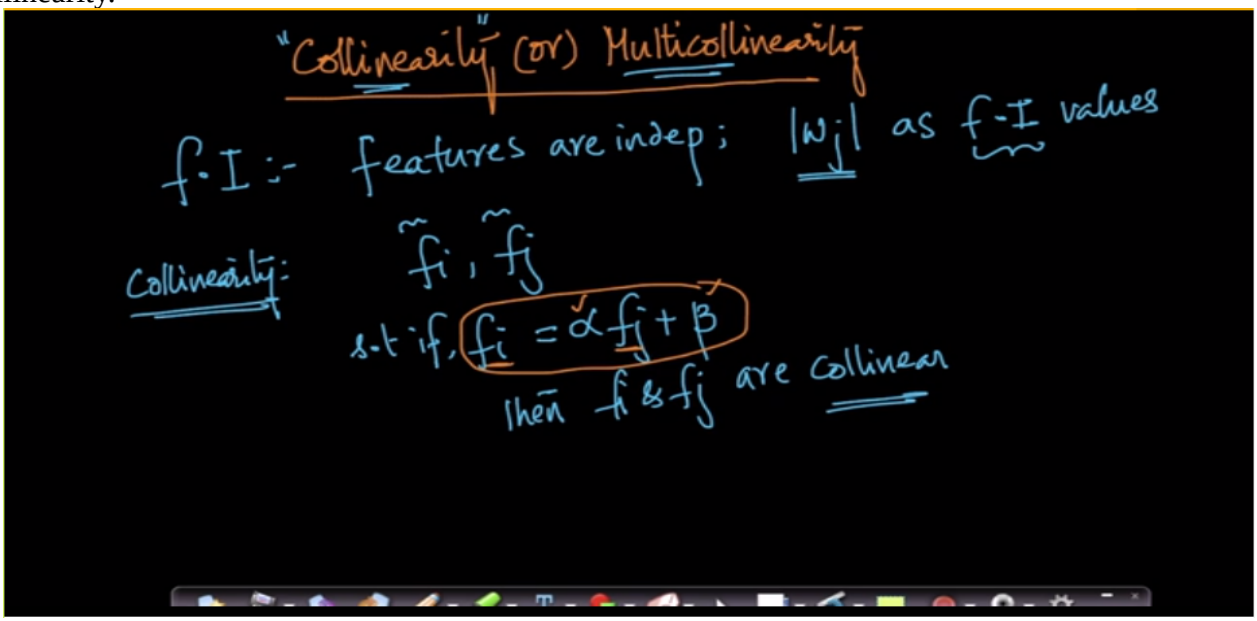
The top features are concluded using more absolute value of the weight to the feature.



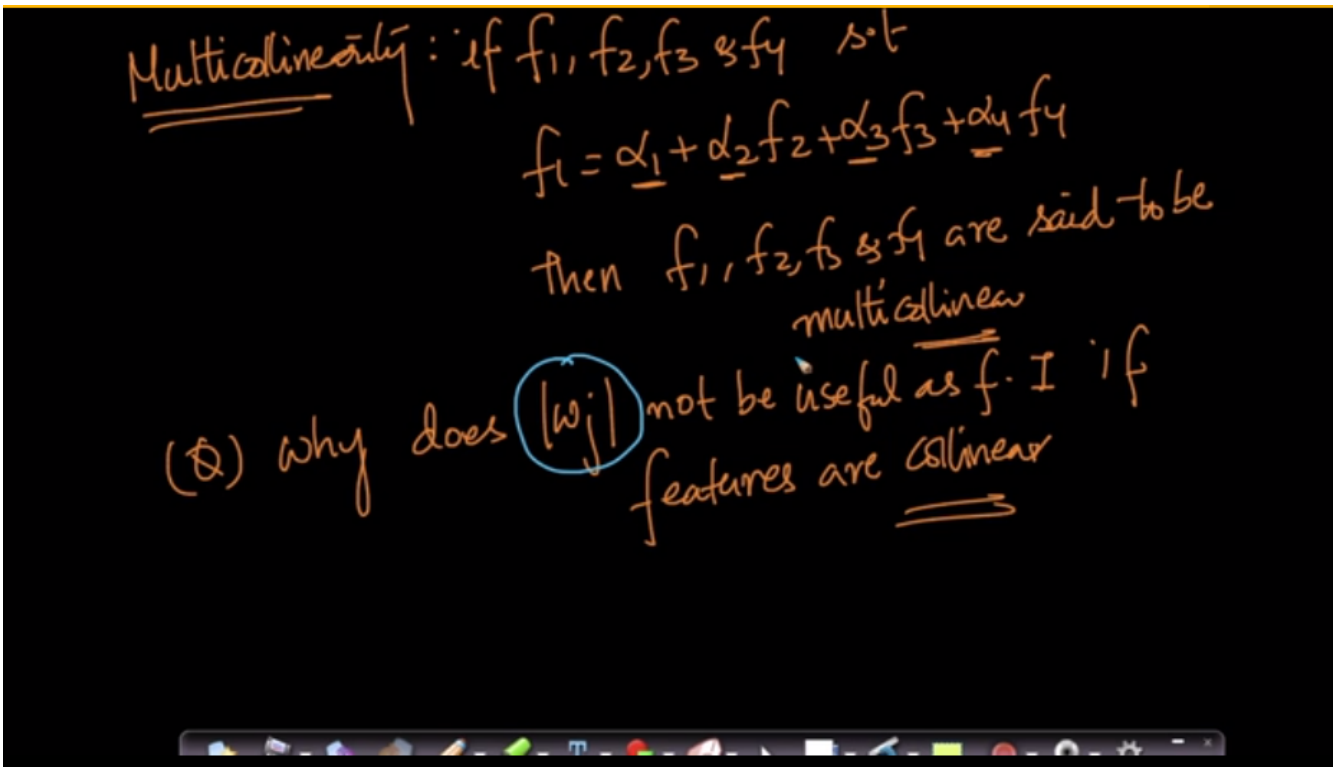
Col linearity (or) multicollinearity:

If features are independent we can use the absolute weight values for feature importance.

Features are not independent in collinearity and multi-collinearity.
Collinearity: If the features are represented in a linear relationship then this relationship is called collinearity.

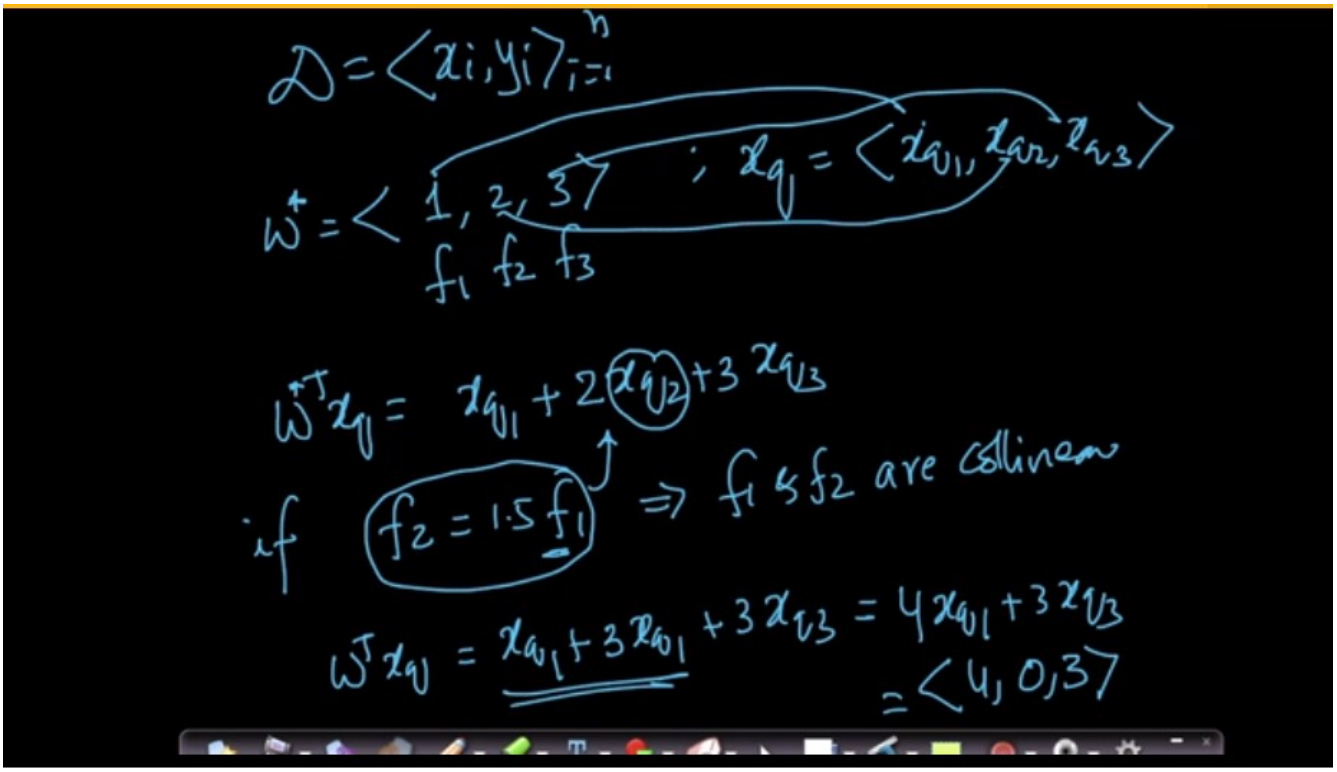


Multicollinearity: In this several features are connected with a linear relationship.

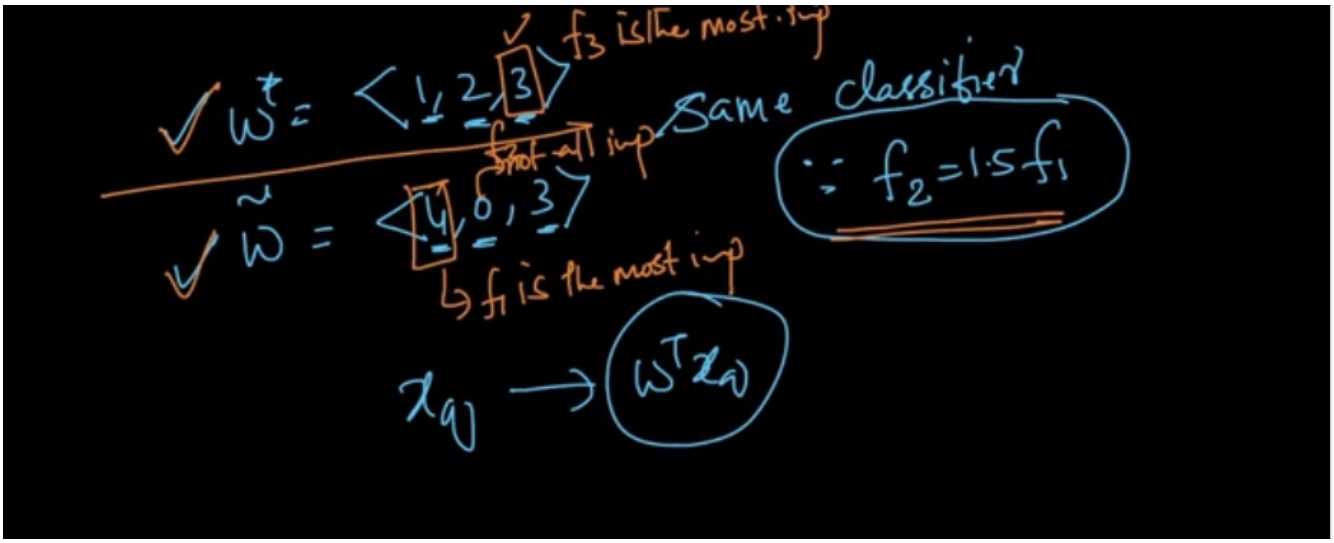


|wi| is not useful for col-linear features, because there is a relationship between the features.

If there is a relationship between f1 and f2 then the function changes, with the relation of the features.

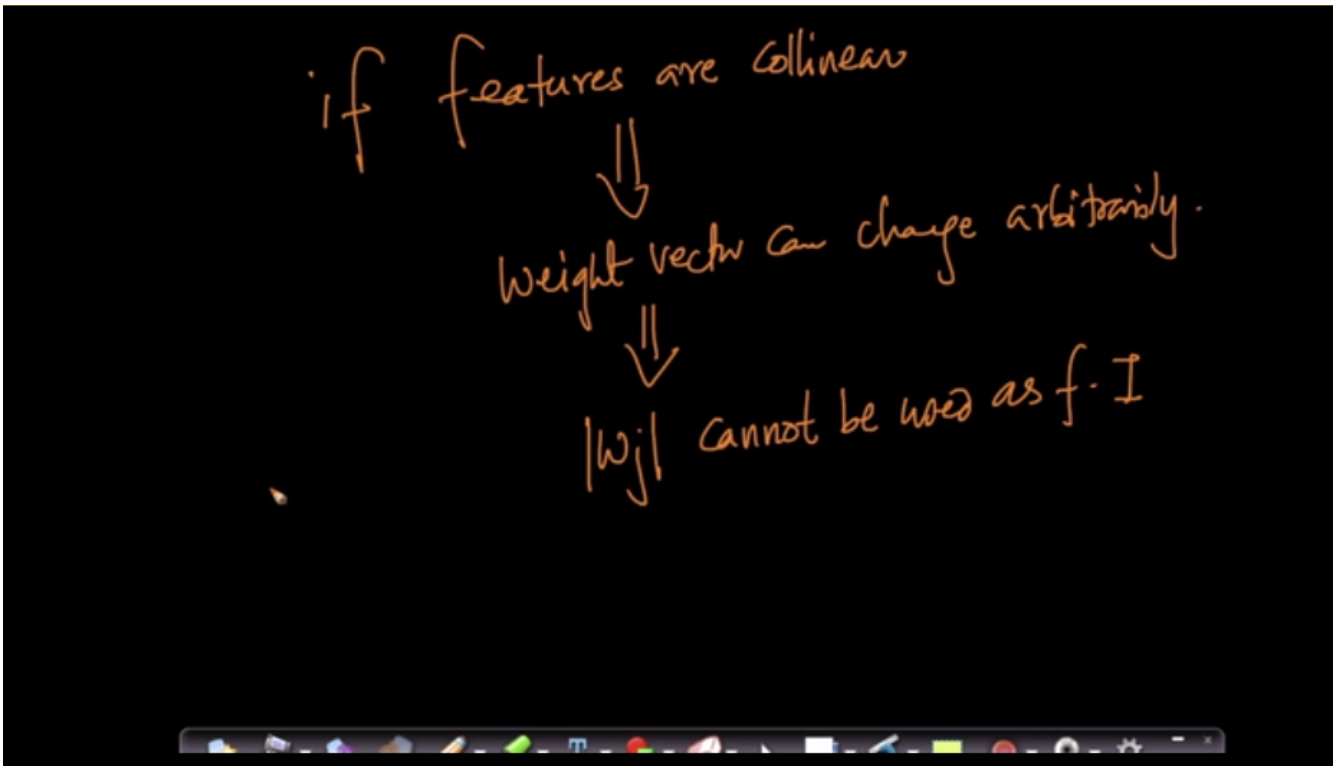


Since, the features are co linear the feature importance changes.



If features are collinear (or) multicollinear then the weight vector changes arbitrarily.

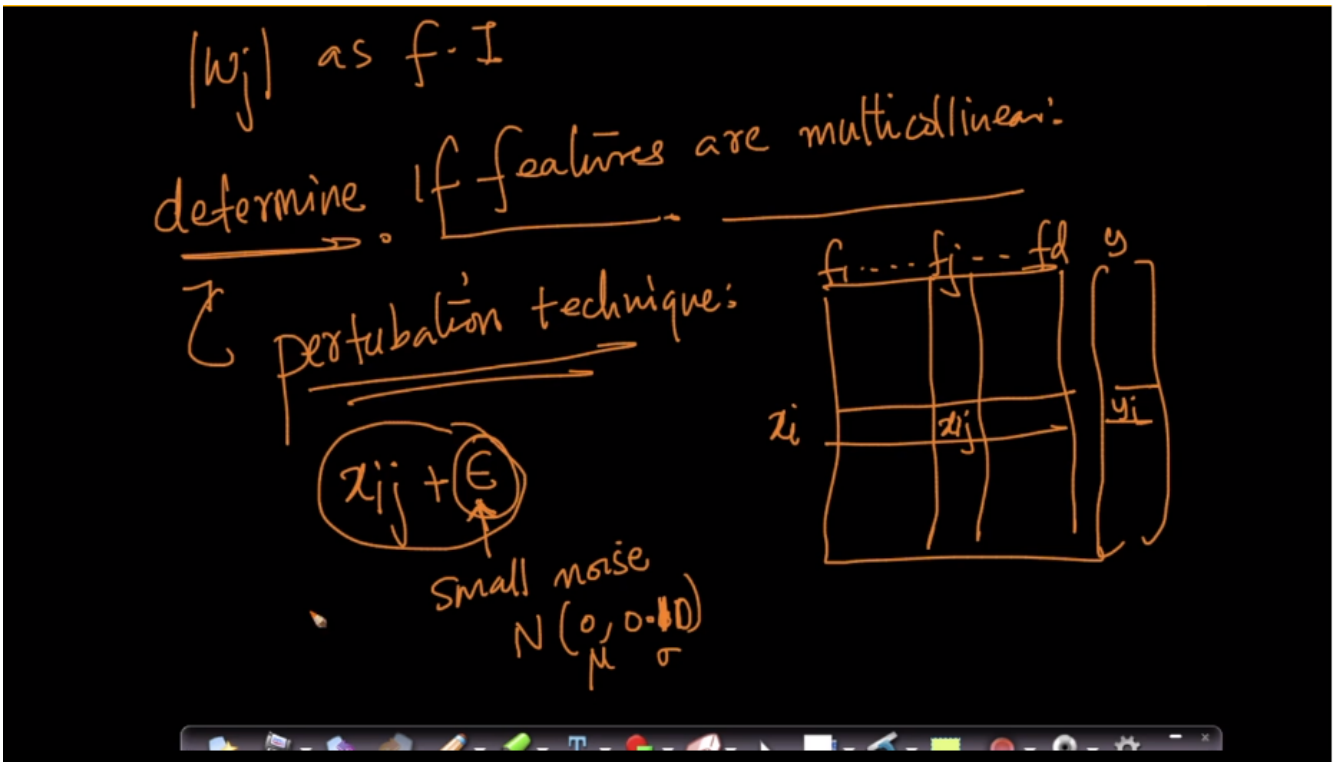
Conclusion:



Determining if features are multi collinear or not?

This can be determined by using perturbation technique.

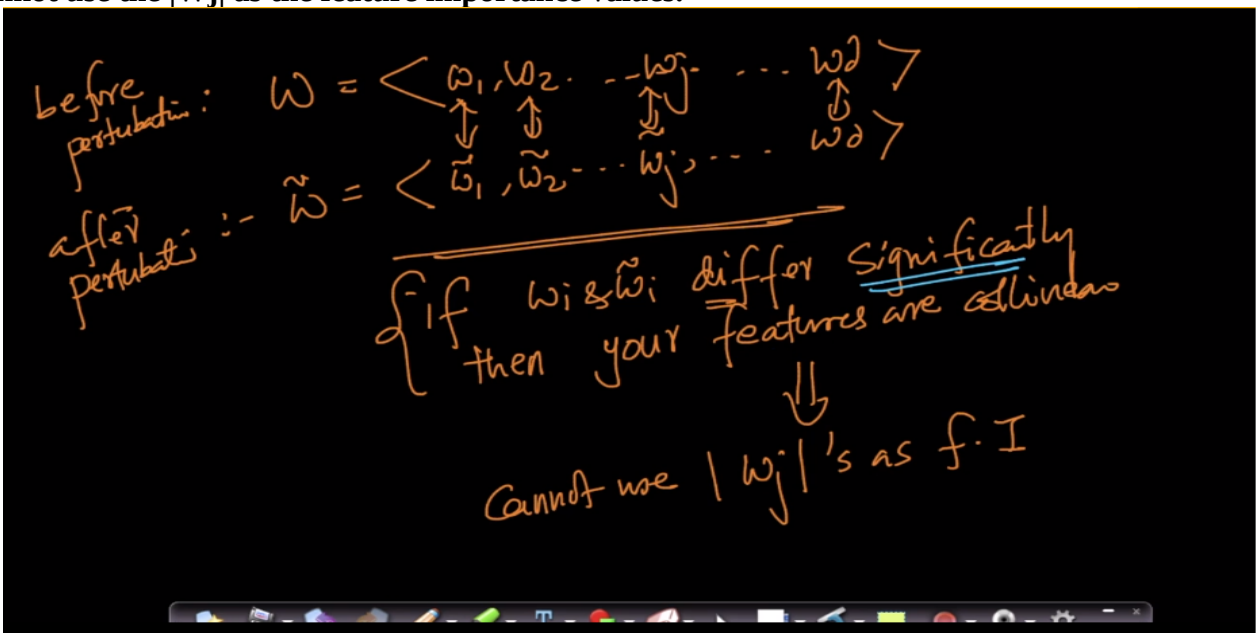
In this technique we will add small noise to each data point in the feature space.



Technique:

Before perturbation we will compute the weights of the feature W' vector and after the perturbation W''.

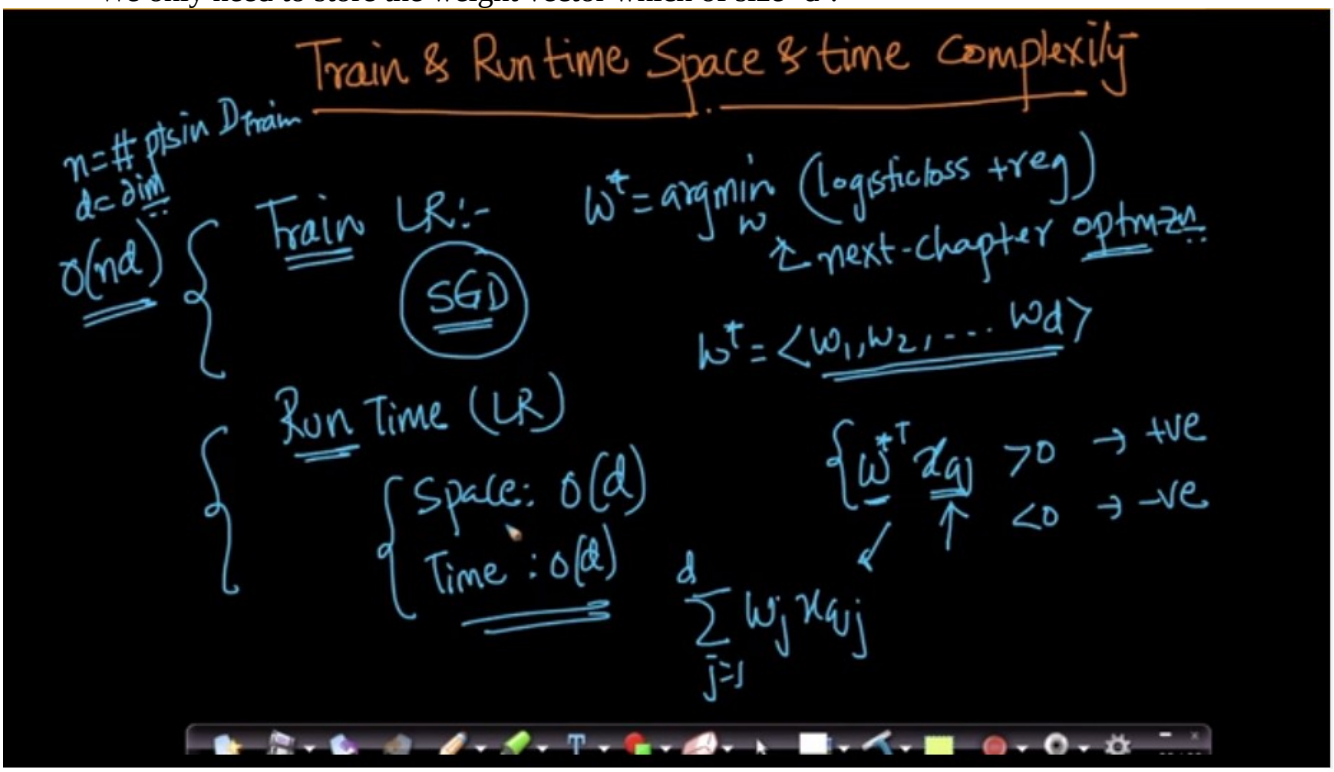
If w' and w'' differ significantly then we can conclude that features are collinear, then we cannot use the |Wj| as the feature importance values.



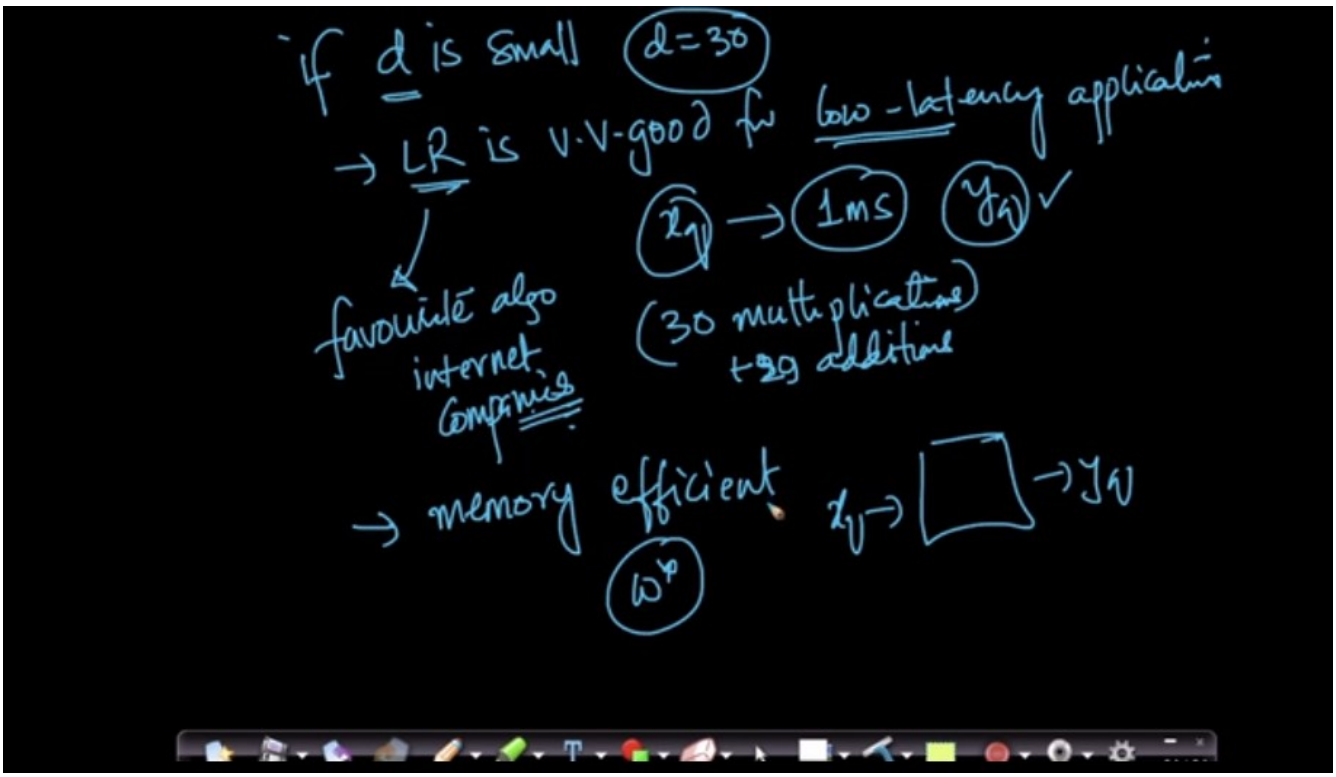
If weights cannot be used in case of col-linear features, THERE IS ALWAYS FORWARD FEATURE SELECTION technique that can be implemented.

Time and space complexity of Logistic regression:

We only need to store the weight vector which of size ‘d’.

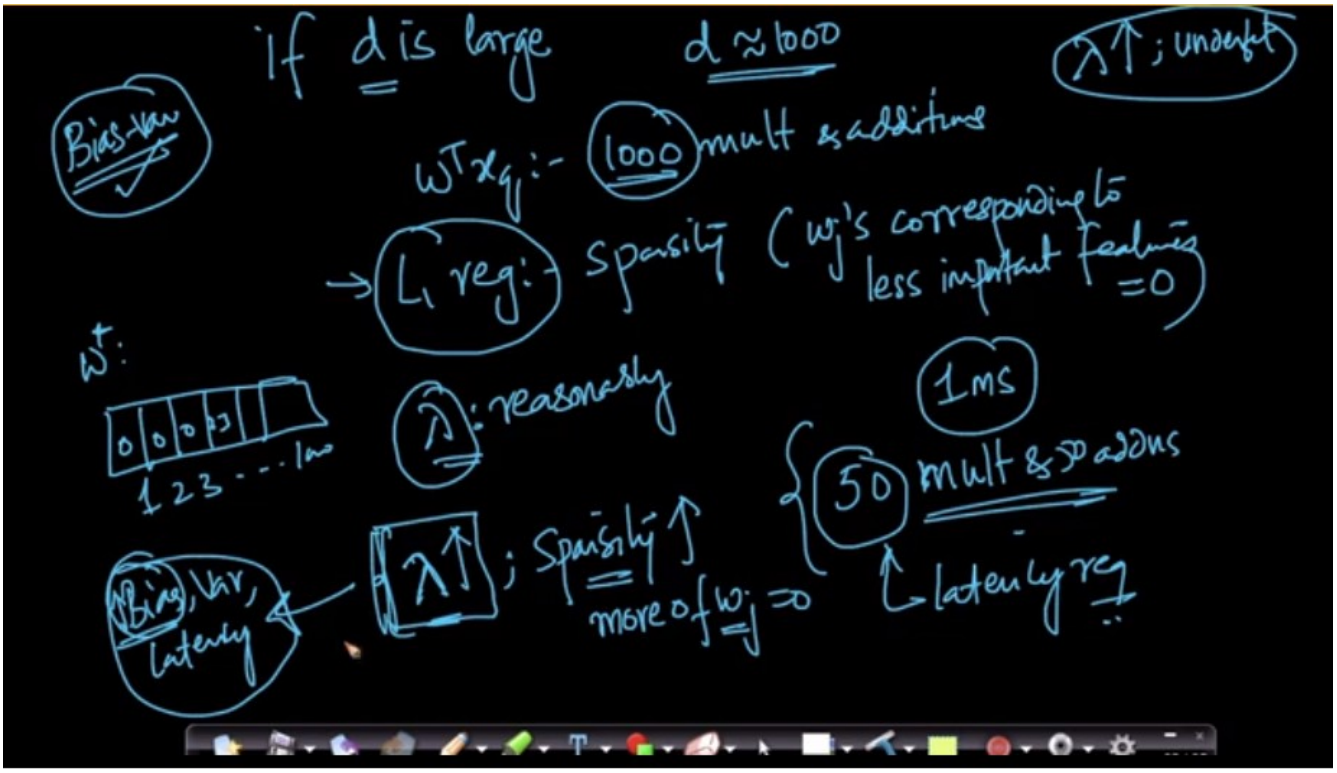


If the ‘d’ is very small then the latency of the result is low for the query point.

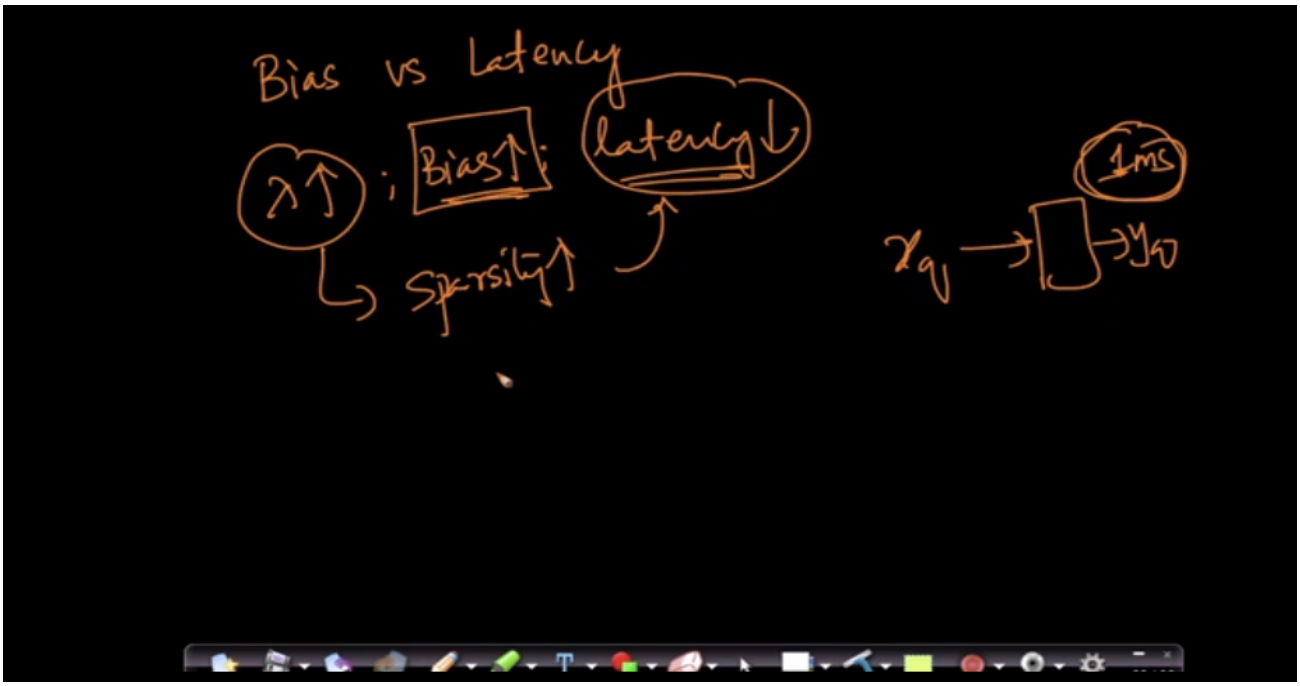


If ‘d’ is very large, we need 1000’s of features, We use L1 regularization to make feature importance.

For using L1 regularization we need to take care of the sparsity, lambda, bias and variance for making the model.

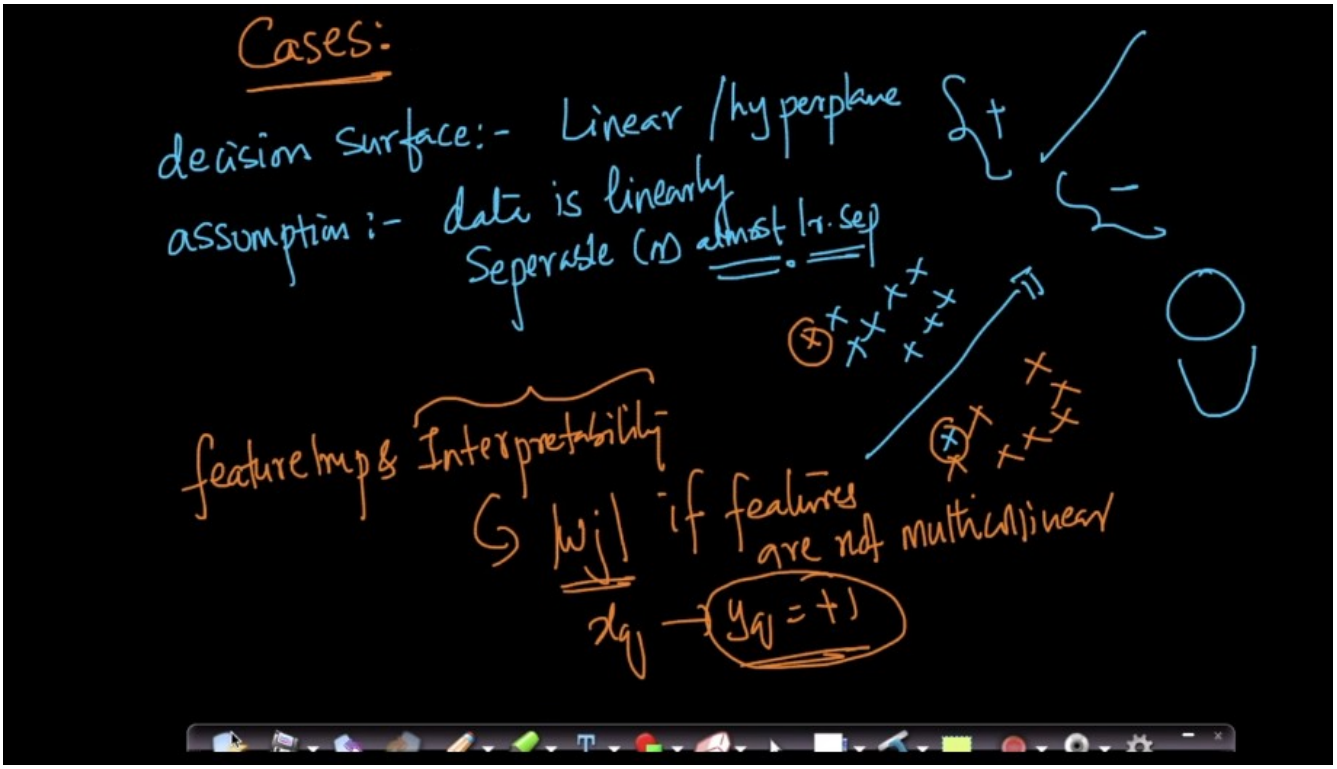


In some cases, having the working model is more important than the not working model.

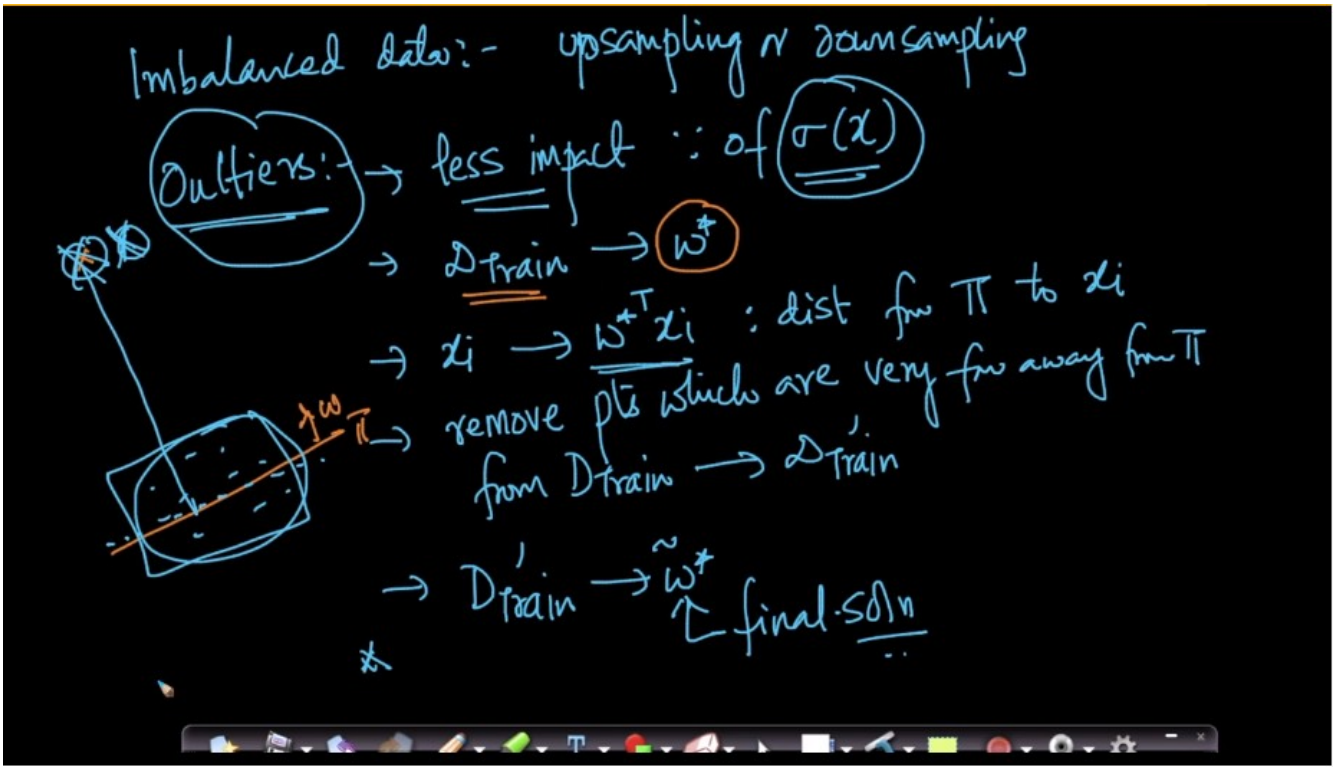


Real world Cases:

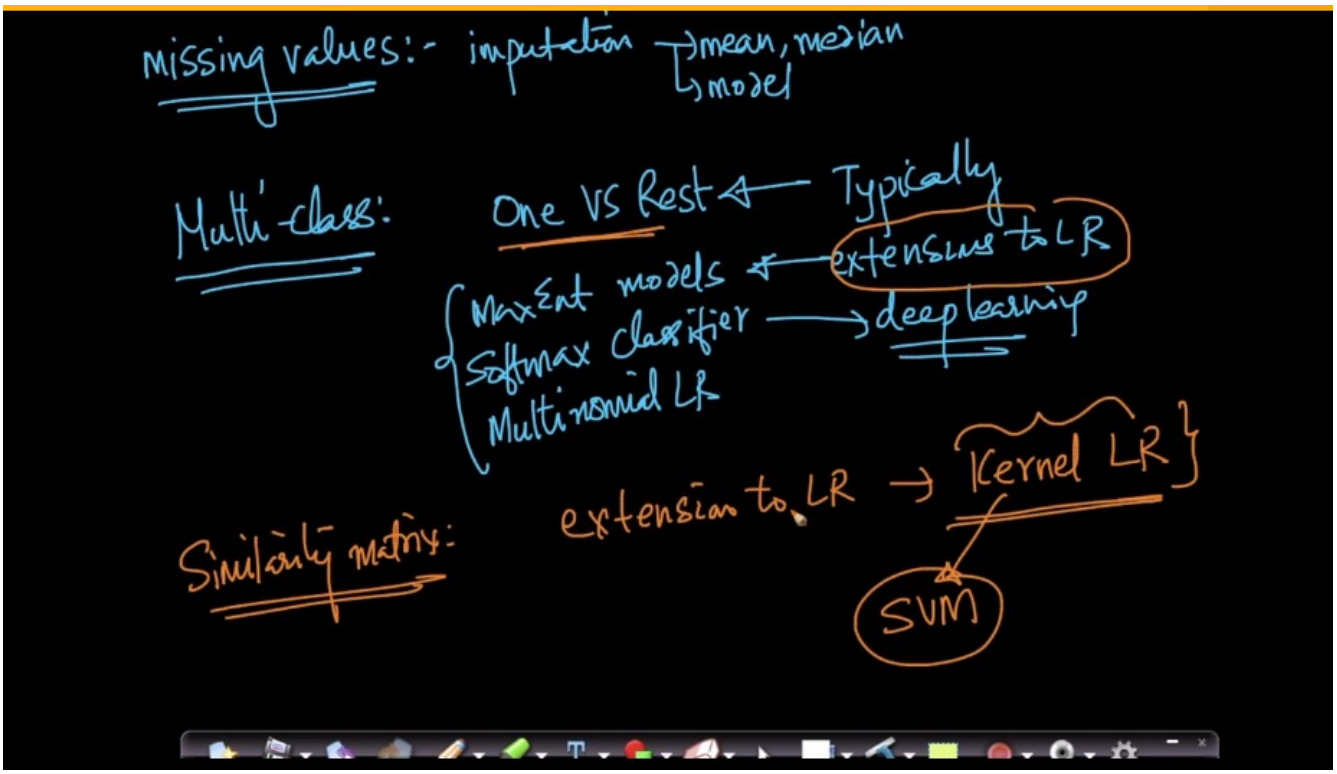
Decision surface: Linear / hyper-plane
Assumption: Data is linearly separable (or) almost linearly separable.



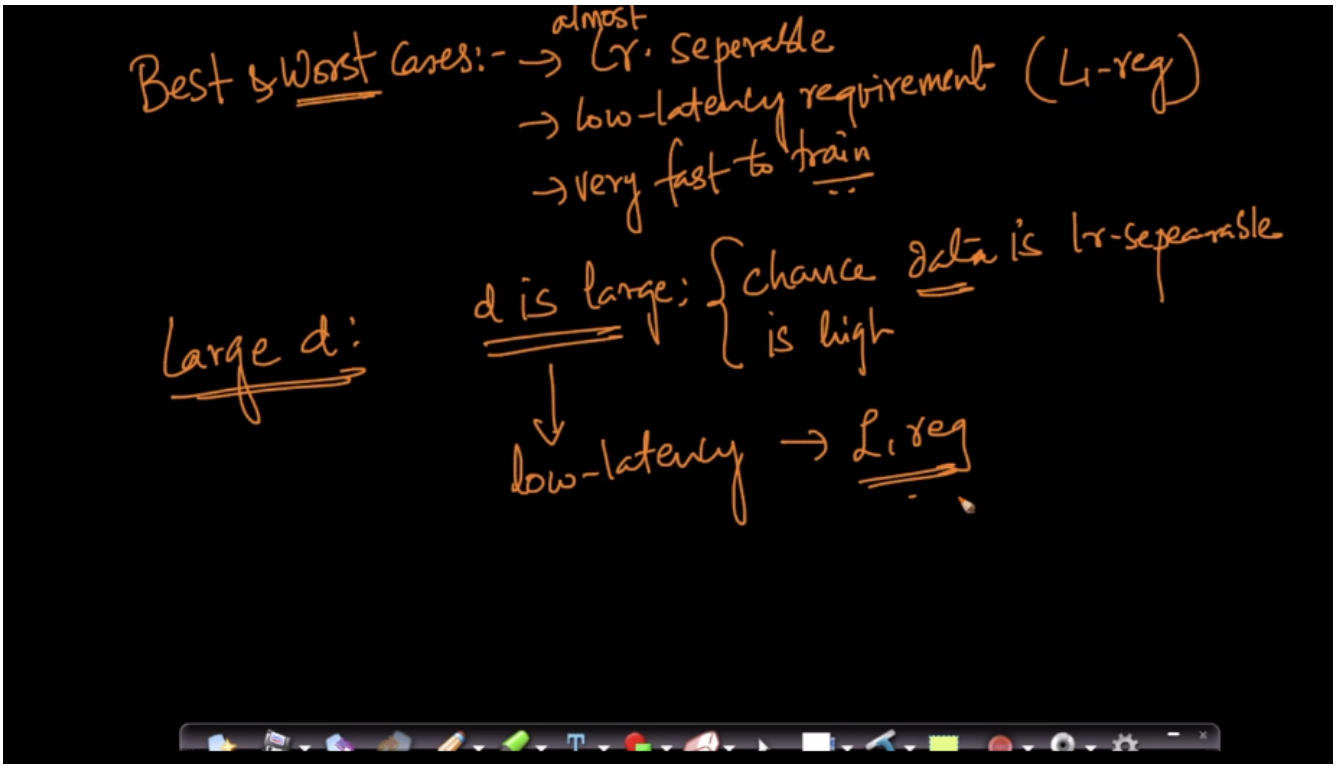
Imbalanced data: We use up sampling or down sampling.



Regular logistic regression does not support similarity matrix. We use Kernel LR.



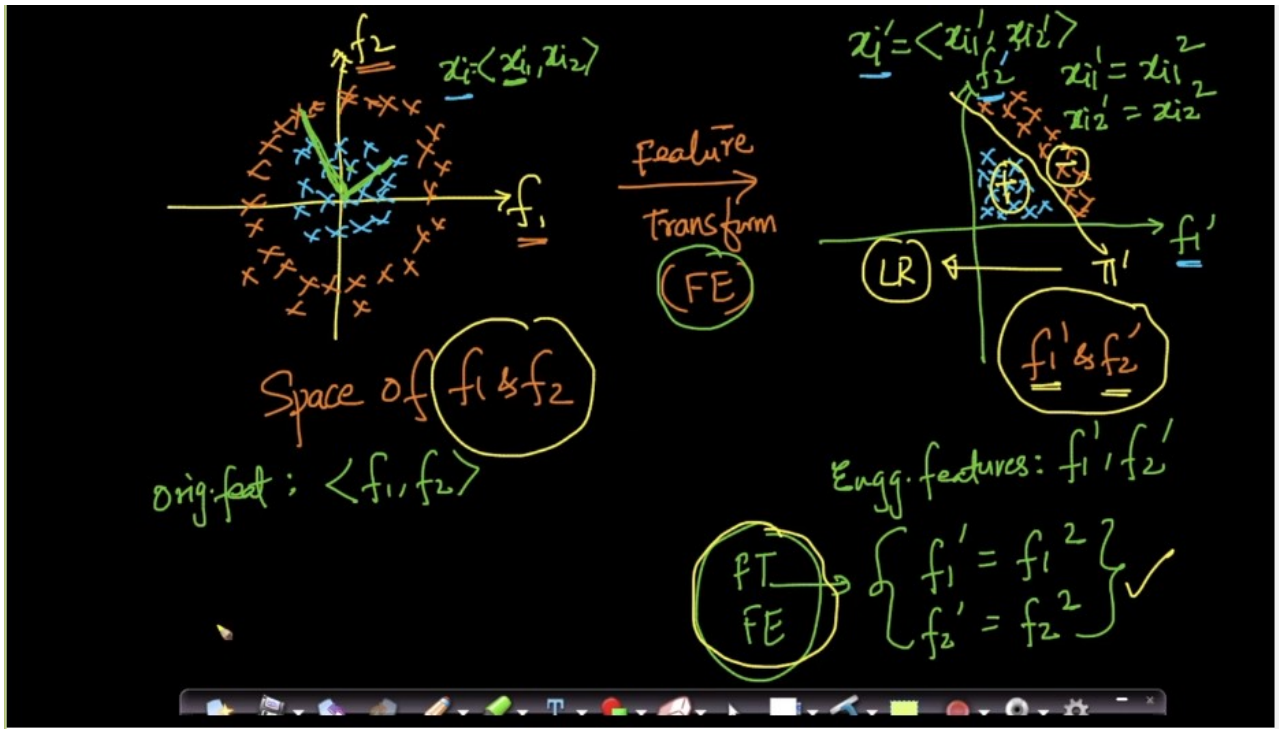
Best and worst cases:



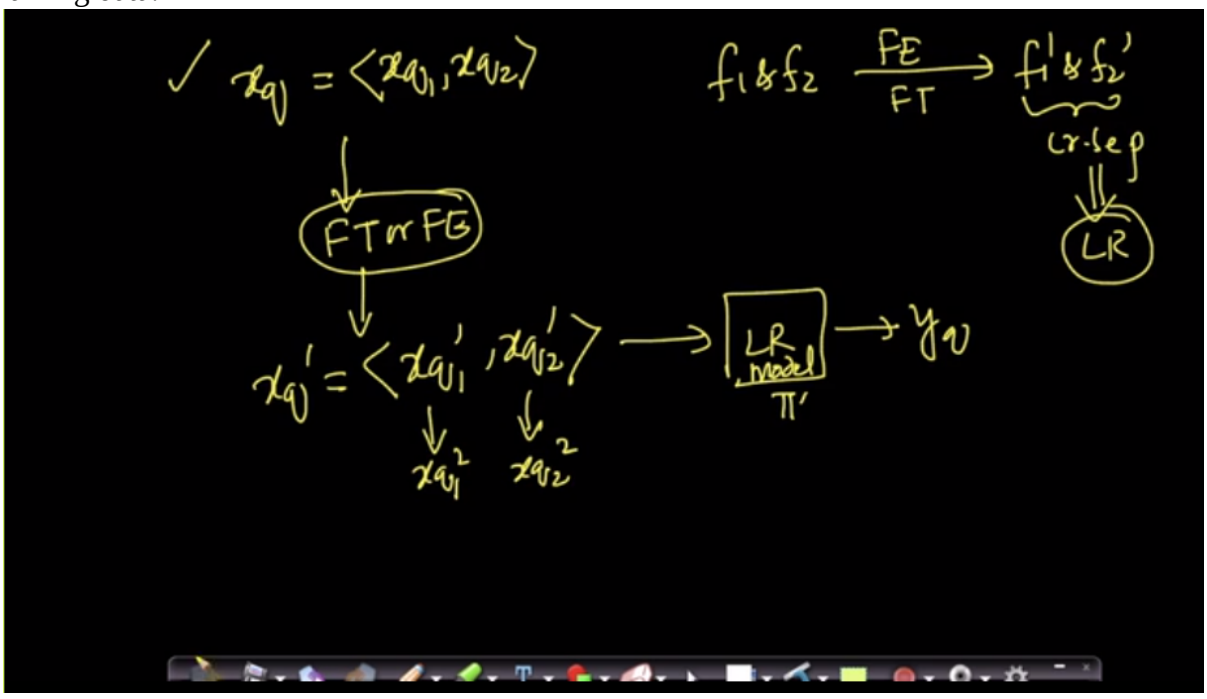
Non – linearly separable data and feature engineering:

Tweaking the data to some extent we can get the features that can be separable by logistic regression.

IMPORTANT: CASE - 1

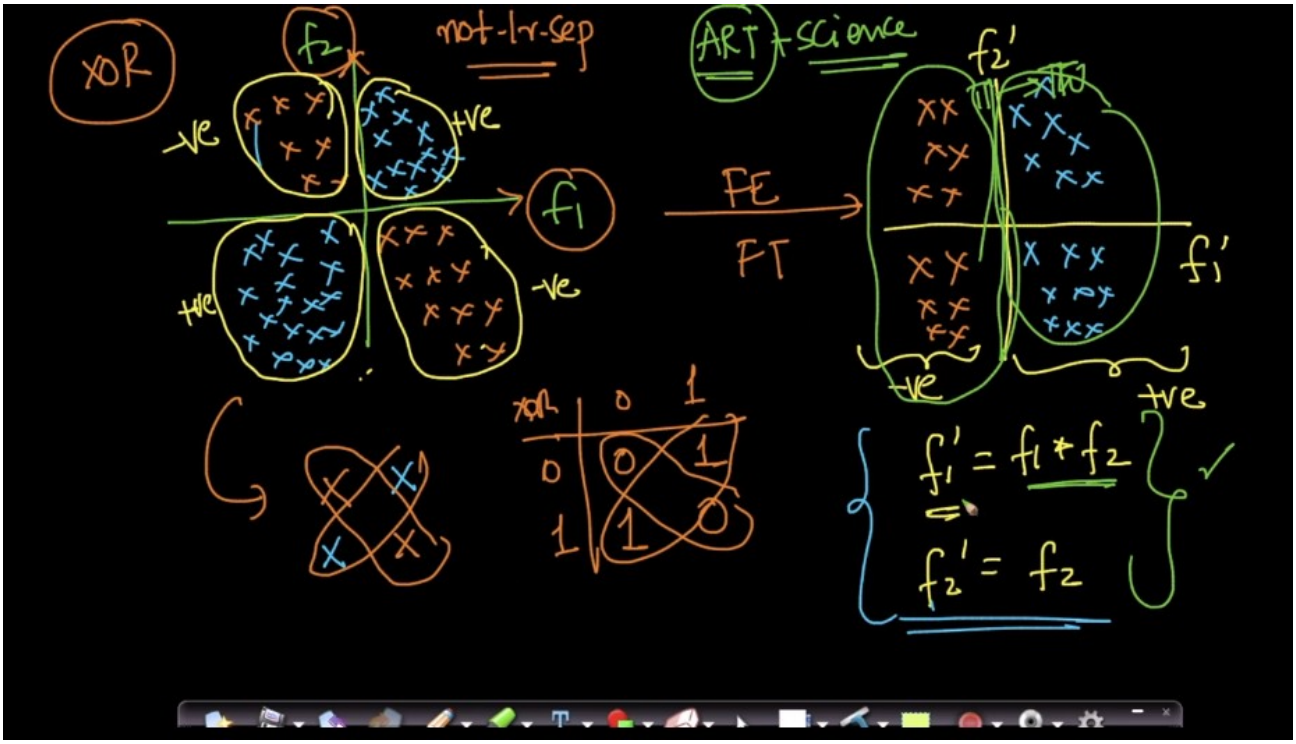


Even for the query point we will transform the features in the same way of transformation done to training data.

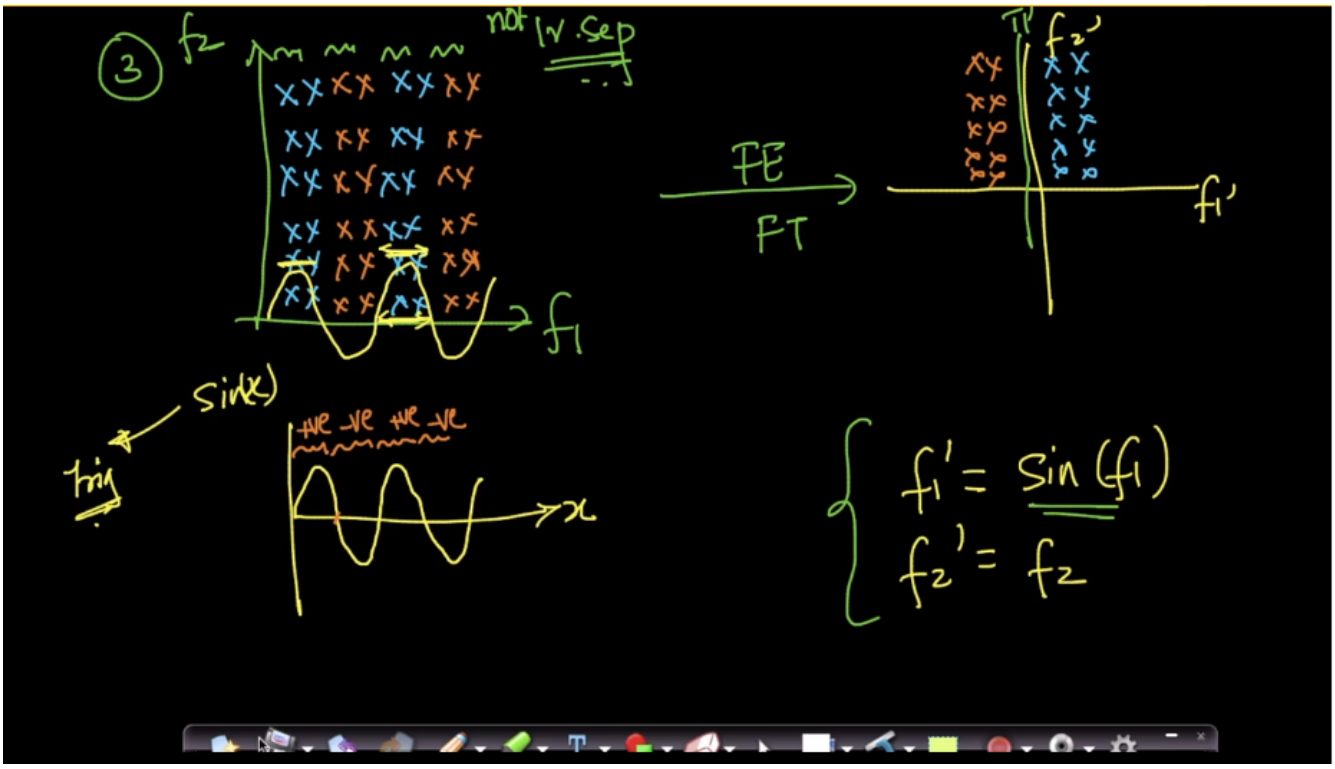


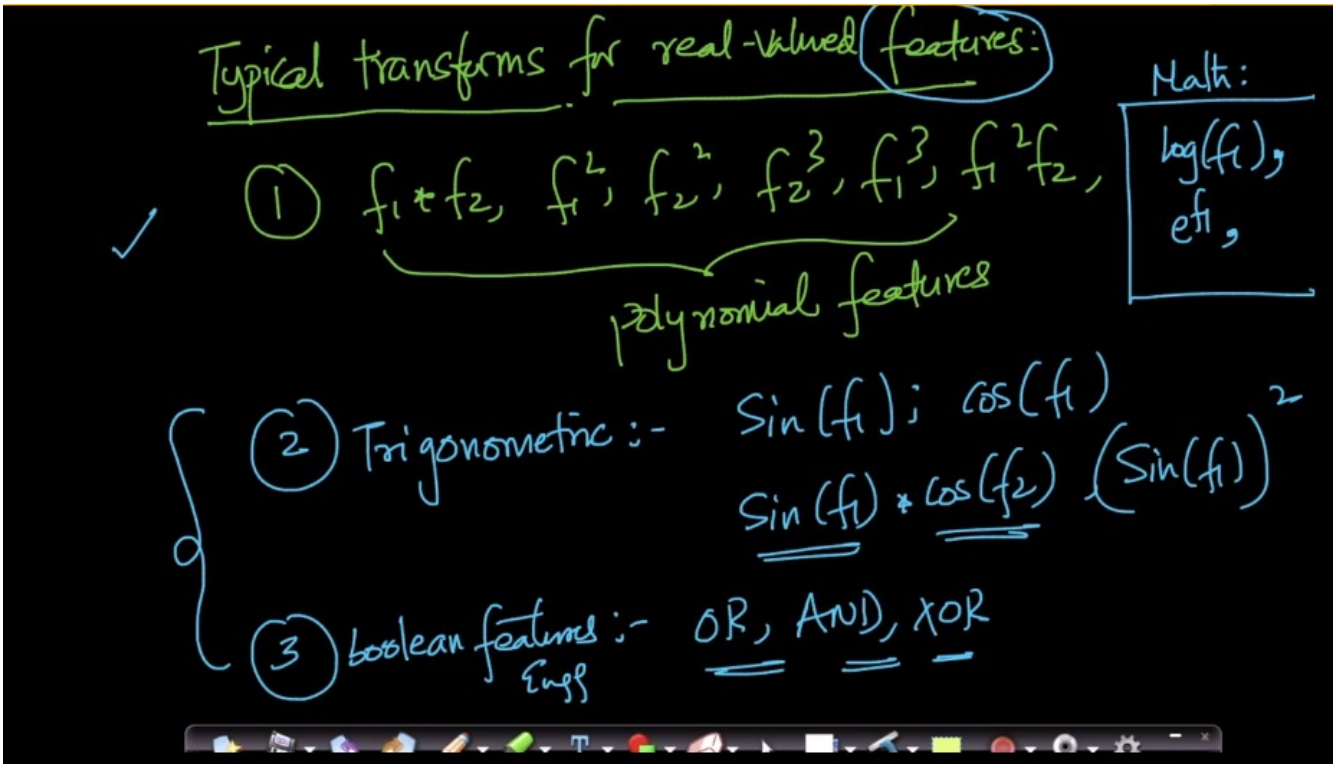
How to find which feature transformation to apply?
This feature transformation can only come from experience.

Another case: **CASE - 2**

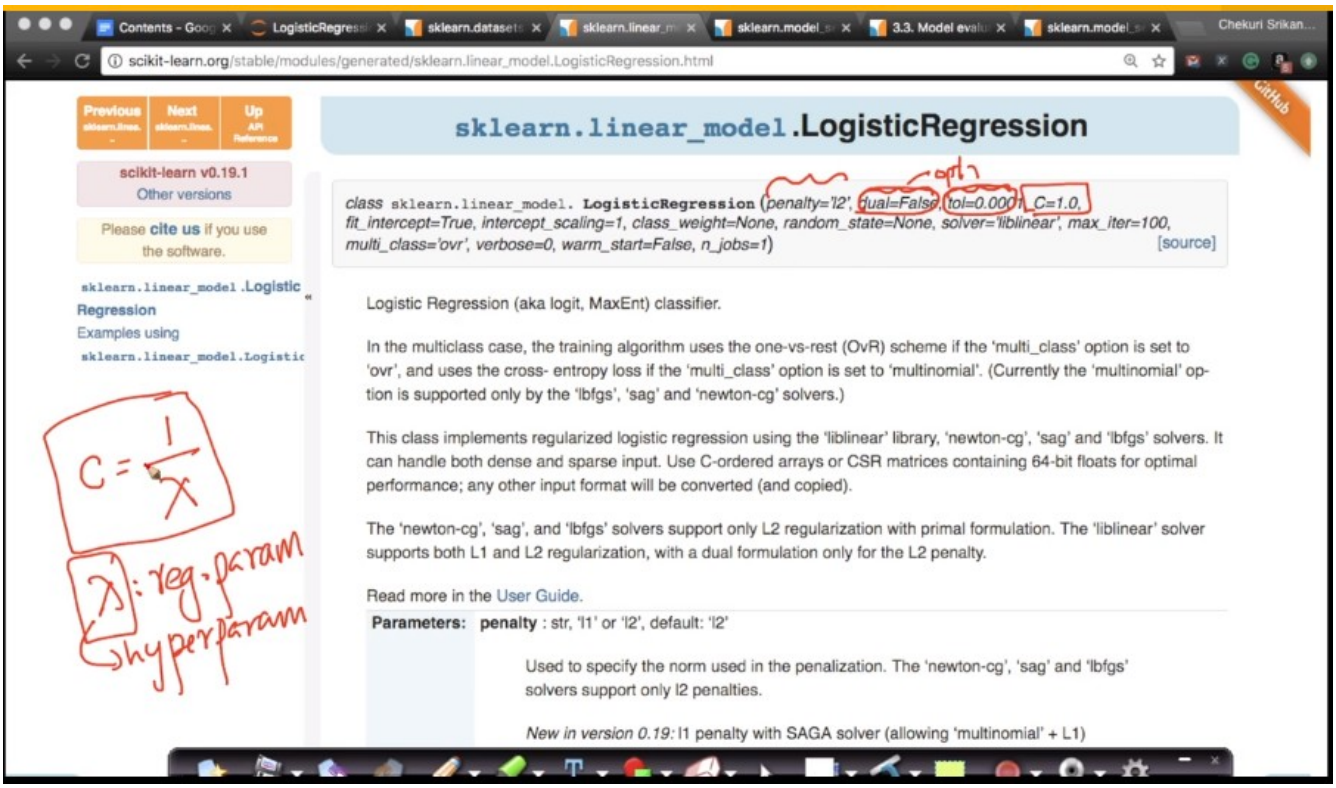


CASE – 3

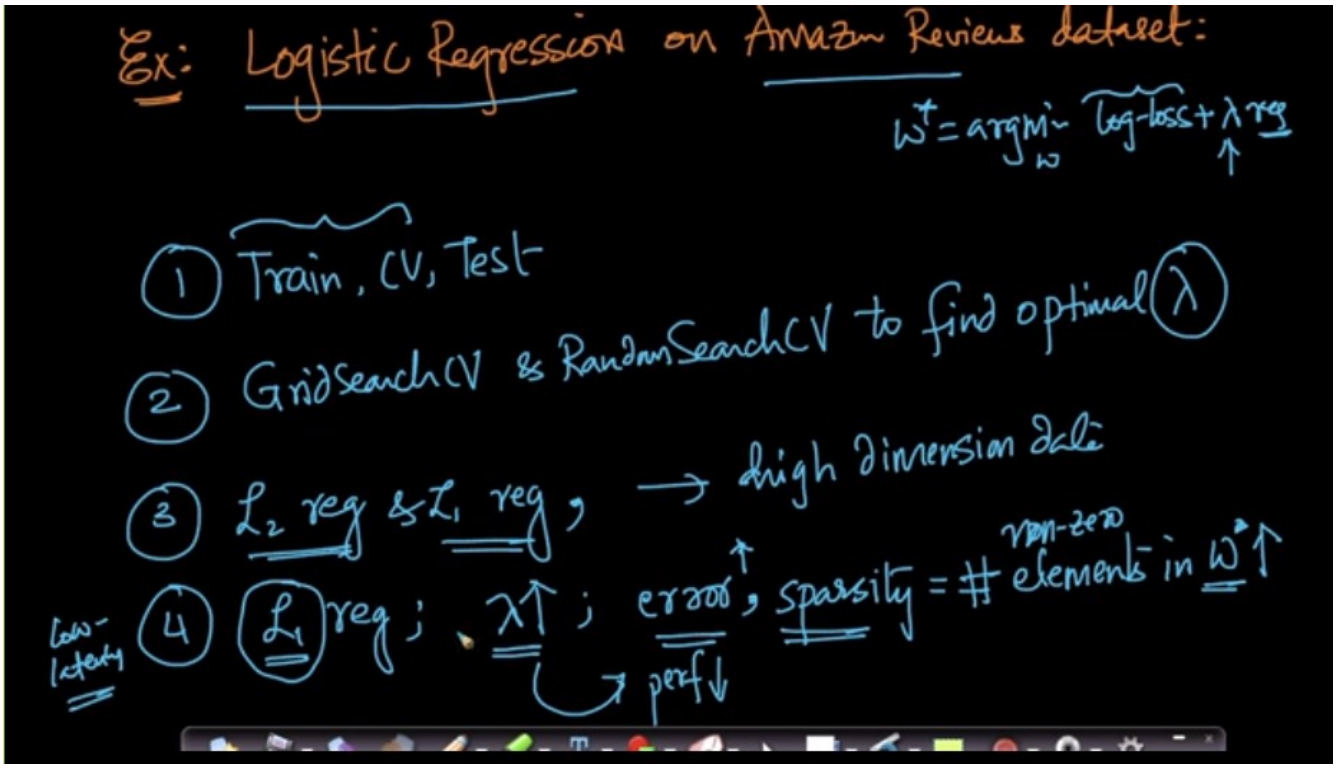




Sklearn implementation of Logistic regression:



Exercise:



Generalized linear models:

Multi nominal distribution: It is for the multi class classification.

Linear regression: probabilistic assumption.

Poisson regression: Poisson distribution- How many visitors will visit the restaurant.