**MISCELLANEOUS TOPICS**

[CALIBRATION OF MODEL](#_jk26v4dmgtyb)

[CALIBRATION PLOTS](#_1jcym8v78klv)

[PLATT SCALING/ SIGMOIDAL CALIBRATION](#_nwrkirx7cvwc)

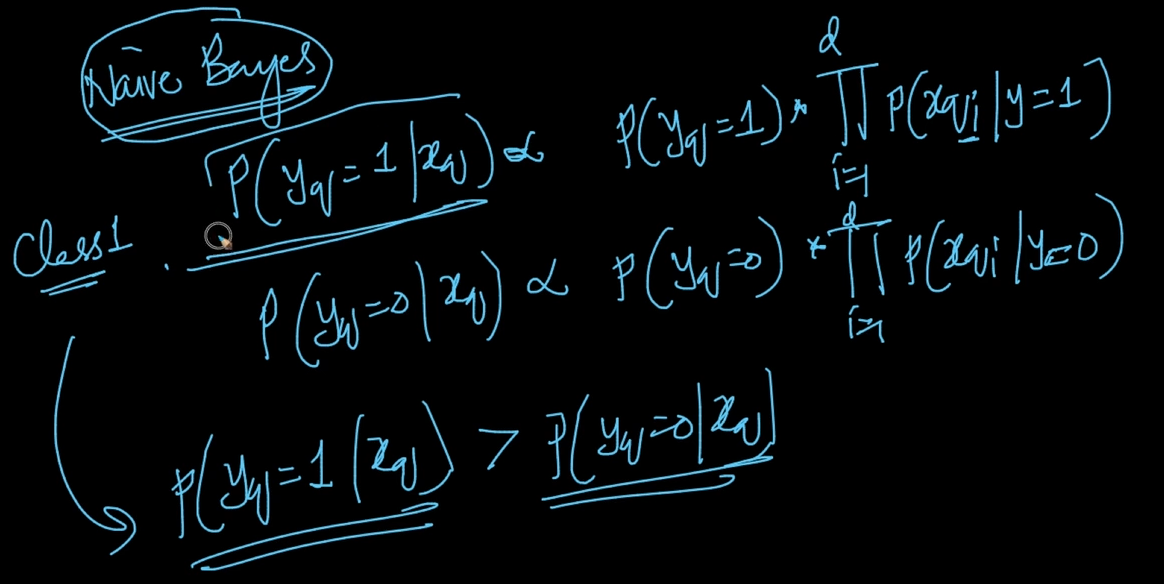
[ISOTONIC REGRESSION / CALIBRATION](#_lge6i8f8fnks)

[RANDOM SAMPLING CONSENSUS (RANSAC)](#_ng45ouxg2kmb)

# **CALIBRATION OF MODEL**

We’ve a 2 class classification. With we get f() = but suppose if we want to study

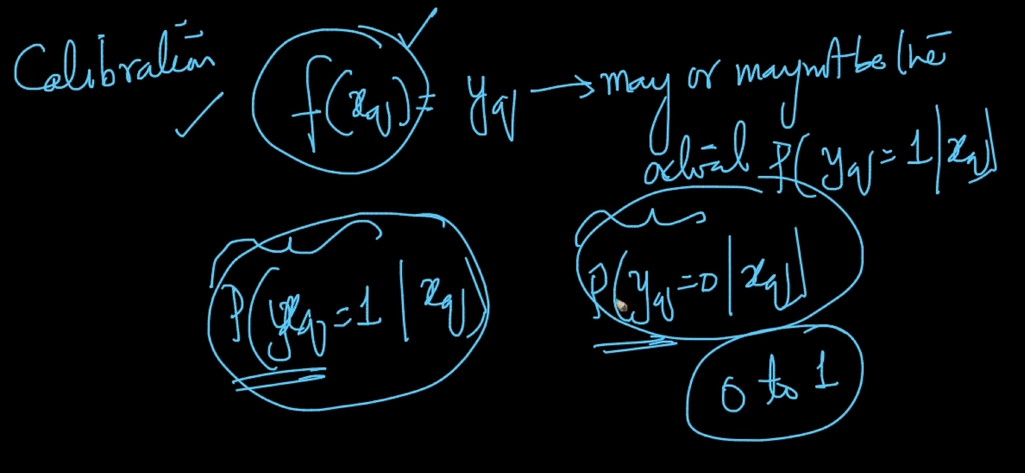
P(not just whether = 0 or 1



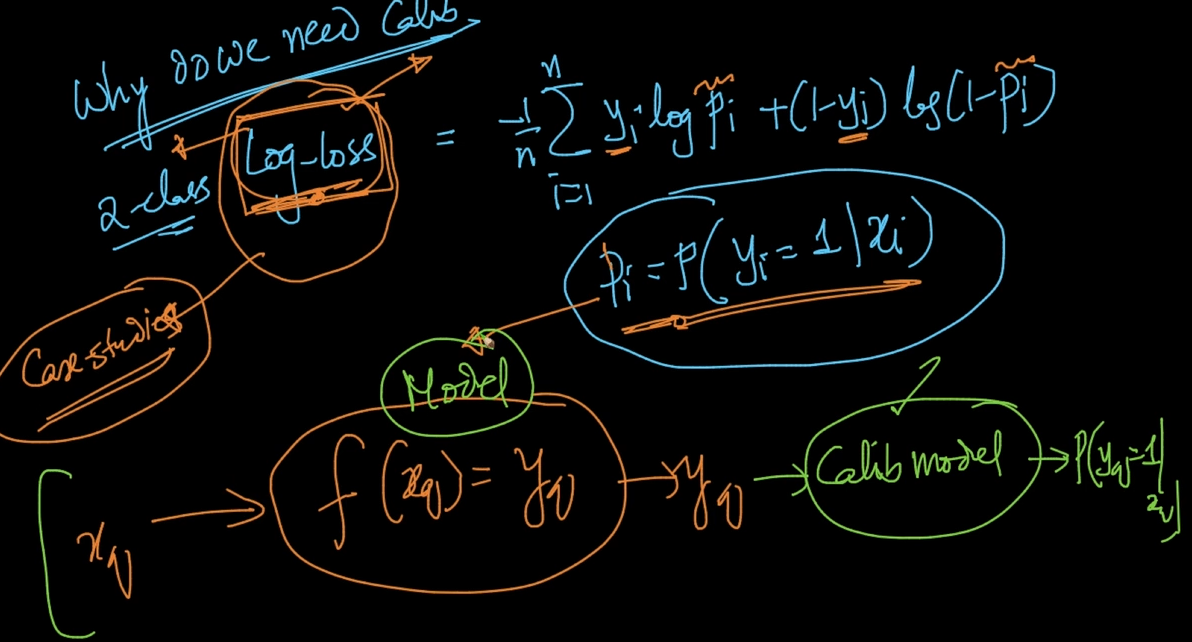
In Naive Bayes, P( or P( are proportional to the above equation

We say that our model belongs to class 1 or y = 1 if P(> P(but these

is also not giving exact value because P(is just proportional not exact



Our f() = may or may not give actual P(but in Calibration we want exact value of P(or P(given our f()



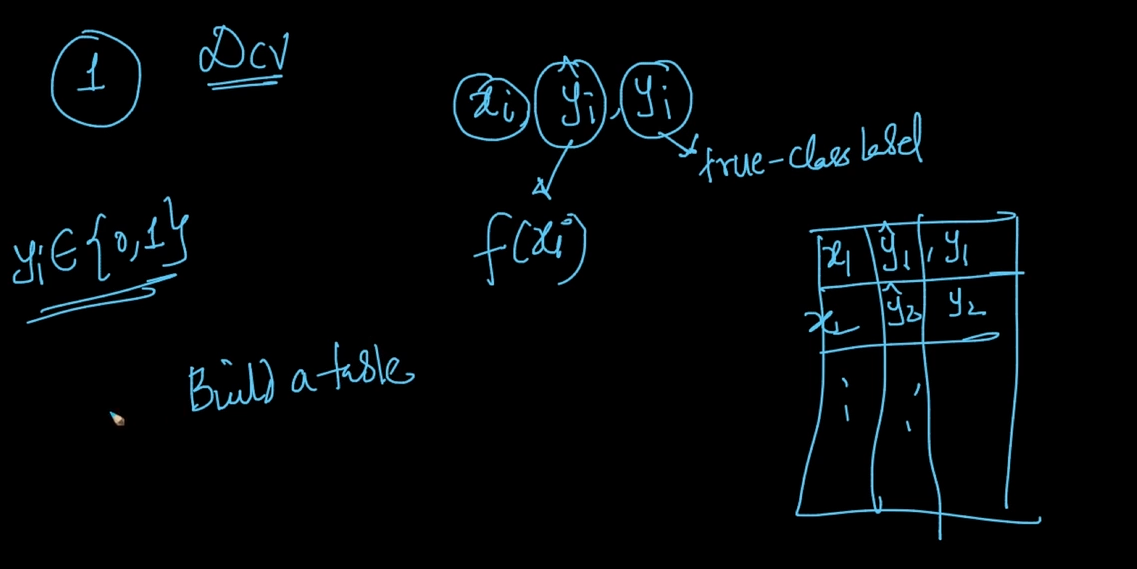
Why do we need Calibration? As we can see log-loss formulae above where

P(and log-loss is very important in ML so we need the exact Probability value .So from we get = so on top of we apply calibration model to get

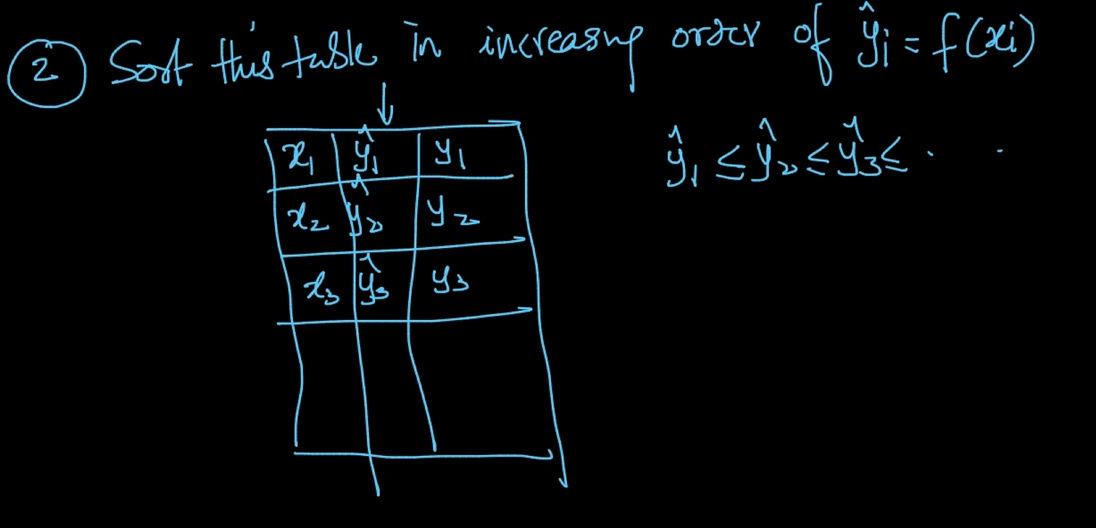
P(

# **CALIBRATION PLOTS**

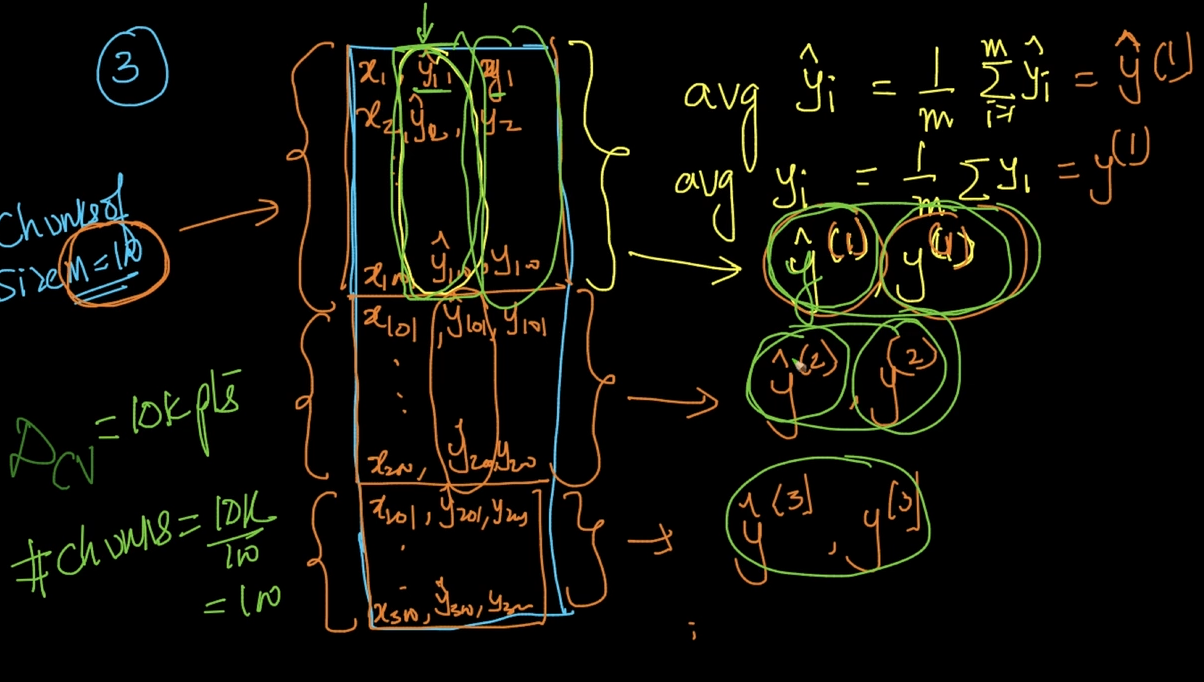
We give our Train data and get a model ‘f’ . We’ve and for each in we apply model f and get = f()

****

**Step 1 :** Build a table with and . Our problem is a binary class . So y = { 0,1 }

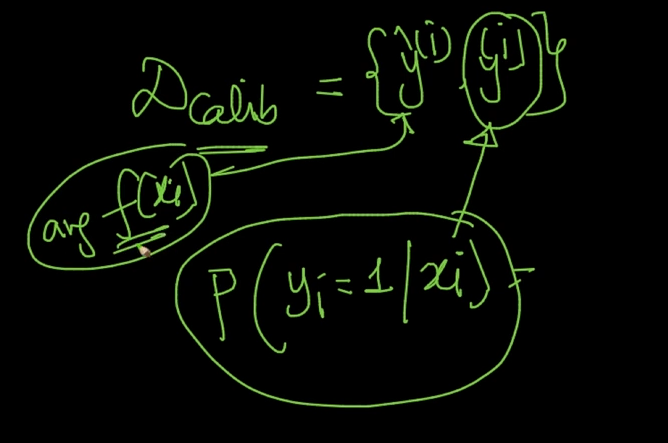


**Step 2:** Sort this table in increasing order of = f() i.e



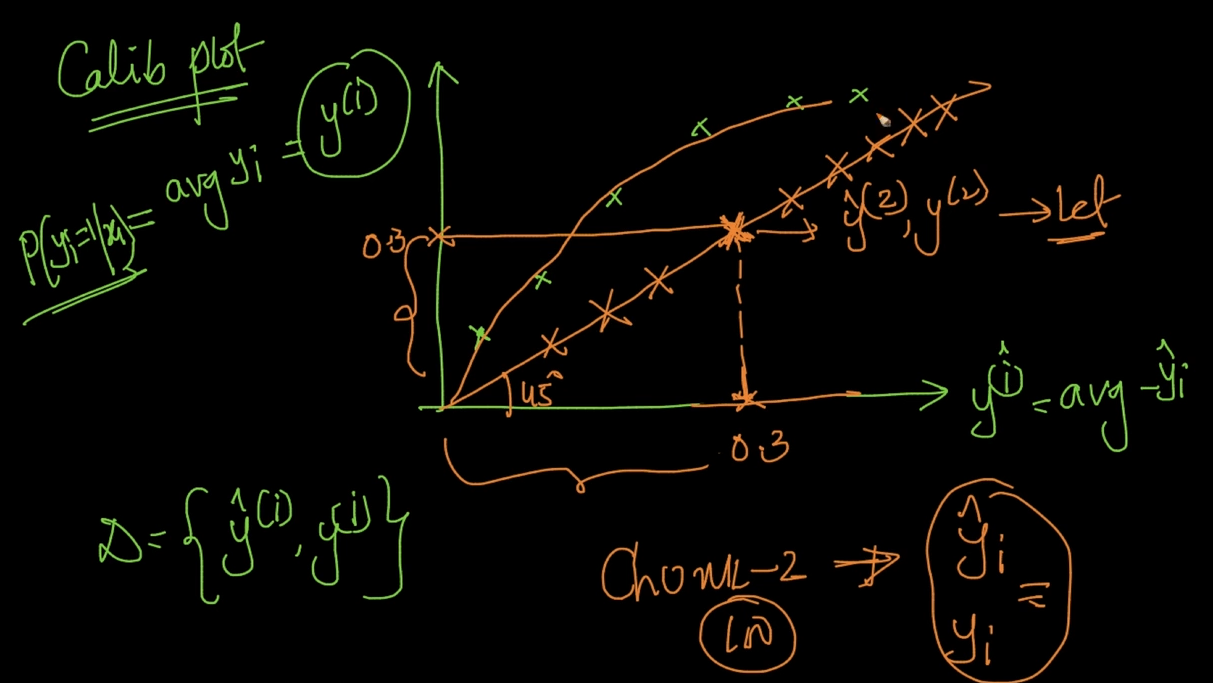
**Step 3 :**We divide the data into chunks let’s say = 10k points we make the size of

chunk = 100. So we’ve 100 chunks. We’ll denote it by and for 1st chunk and it’ll be avg of and y respectively in each chunk



So our calibration data = { } where = average of f()/ and

P() = average of y

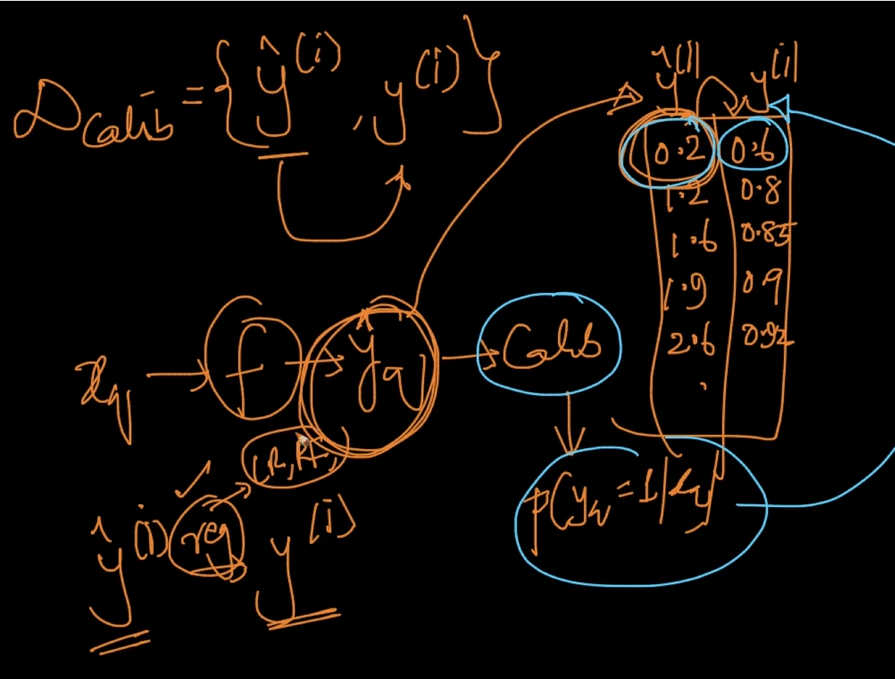


Calibration plot is plotted using as x-axis and as y-axis. In an ideal case let’s say

both have same values let’s say 0.3 it’ll mean that one chunk has all the predicted values

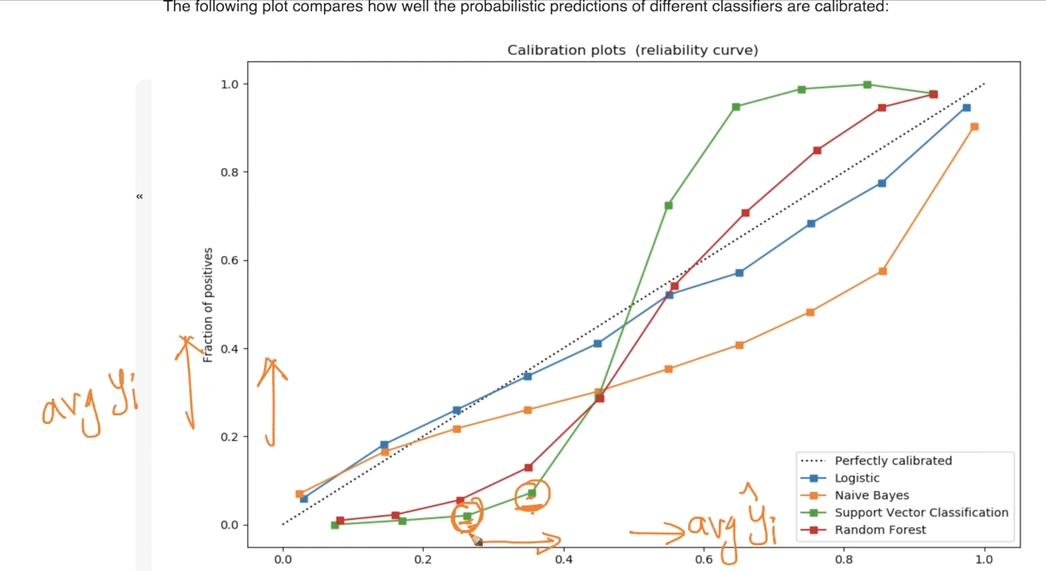
and actual values same meaning 100% accuracy. If ideal then a 45 degree line is plotted

But actually it’s slightly curved like above



We want to map to i.e On our we apply a function ‘f’and get and on top of that we apply our calibration model to get P()as seen in the table above

As seen it’s like a regression problem. We can apply Linear Regression , Random Forest Regressor

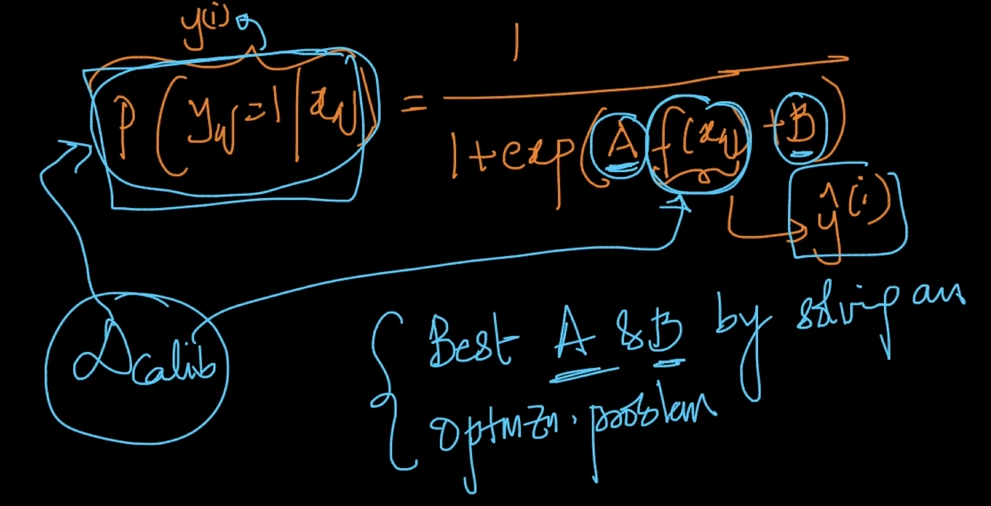


Calibration Plots for various algorithms. All these are monotonic functions

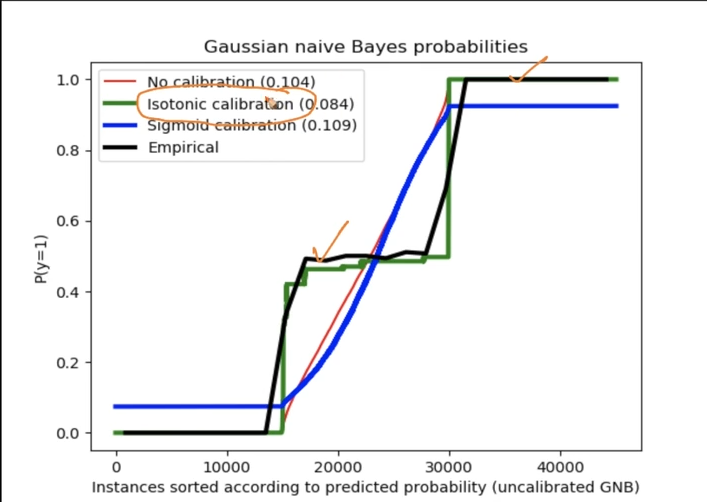
# **PLATT SCALING/ SIGMOIDAL CALIBRATION**

From we predict and as seen in the plots above it’s looking like a sigmoid function

So why can’t we apply a modified sigmoid on and get



The above is the formula for P() and the best A & B is found by solving the optimization problem. We use the dataset to solve optimization problem

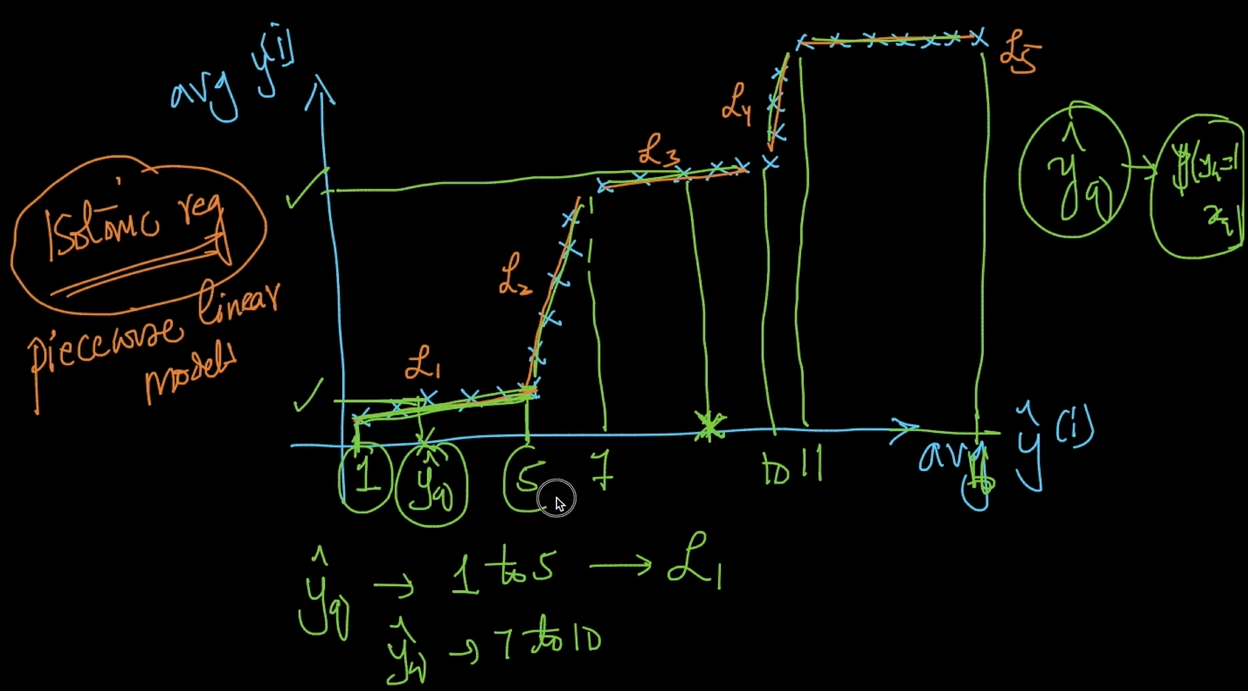


But there’s a problem as seen above our empirical data is our and our sigmoid function is not doing a good job in solving that . So , it can be concluded that our Sigmoidal Calibration is only good when our Calibration plot looks like sigmoid function

It can also be observed that Isotonic Calibration is working excellently for our data as it’s almost overlapping with the real Calibrated data. Isotonic Calibration is the most used calibration model used for Calibration

# **ISOTONIC REGRESSION / CALIBRATION**

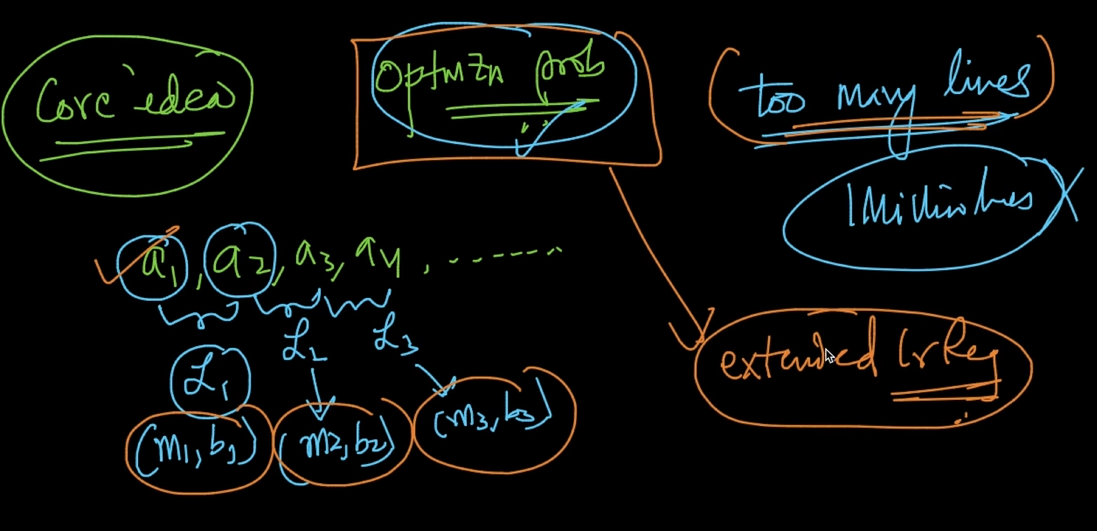
Isotonic works even when Calibration Plot not Sigmoidal. Basically we want to apply a function g() =



As seen, Sigmoidal function won’t work in the above data. In Isotonic Regression what it tries to do is that it tries to approximate this relationship using a bunch of Linear Models

It’s called piecewise Linear models i.e we get different lines for different curves as seen above. We can use the equation of line to predict value of . If the value of lies between 1-5 then we’ll use to predict the value of and so on for others .

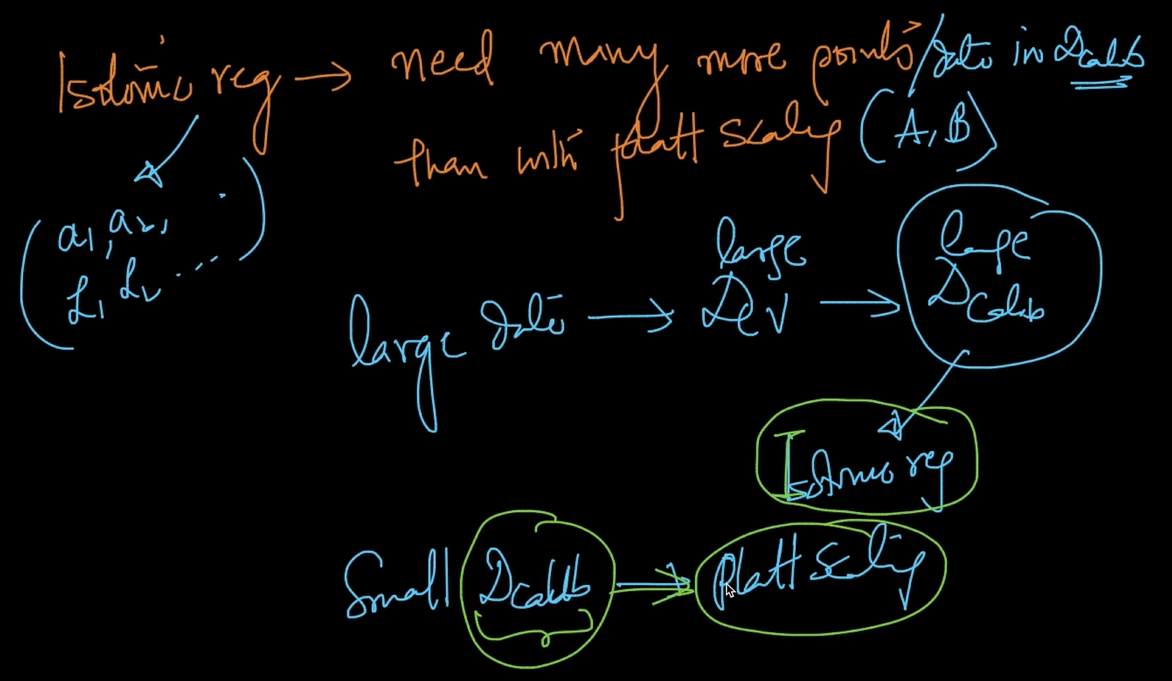
How do we find the right thresholds and lines here?



So the core idea is that it’s an Optimization problem. …. are thresholds

and lies between and so on. For lines we also need to find () and so on

For every line. We don’t want too many lines just 10-15 is enough so that’s also should be kept in mind. It’s like an extended Linear Regression



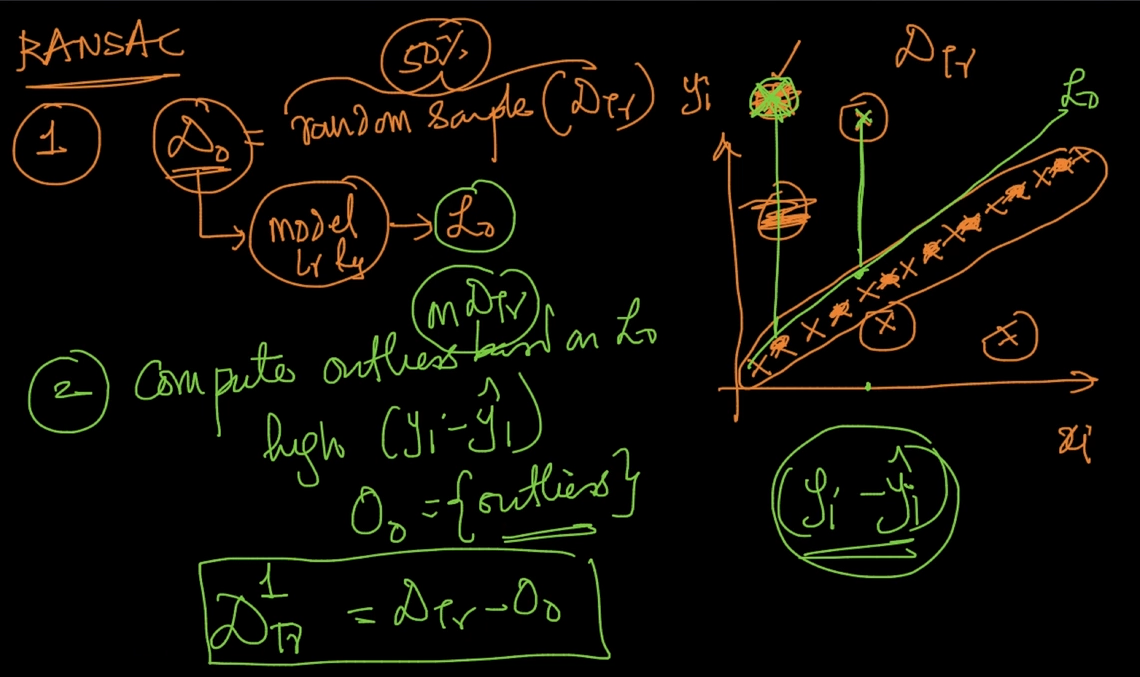
Isotonic Regression needs more points / data in Platt scaling. So if large data then Isotonic but if small data then Platt Scaling does fine

# **RANDOM SAMPLING CONSENSUS (RANSAC)**

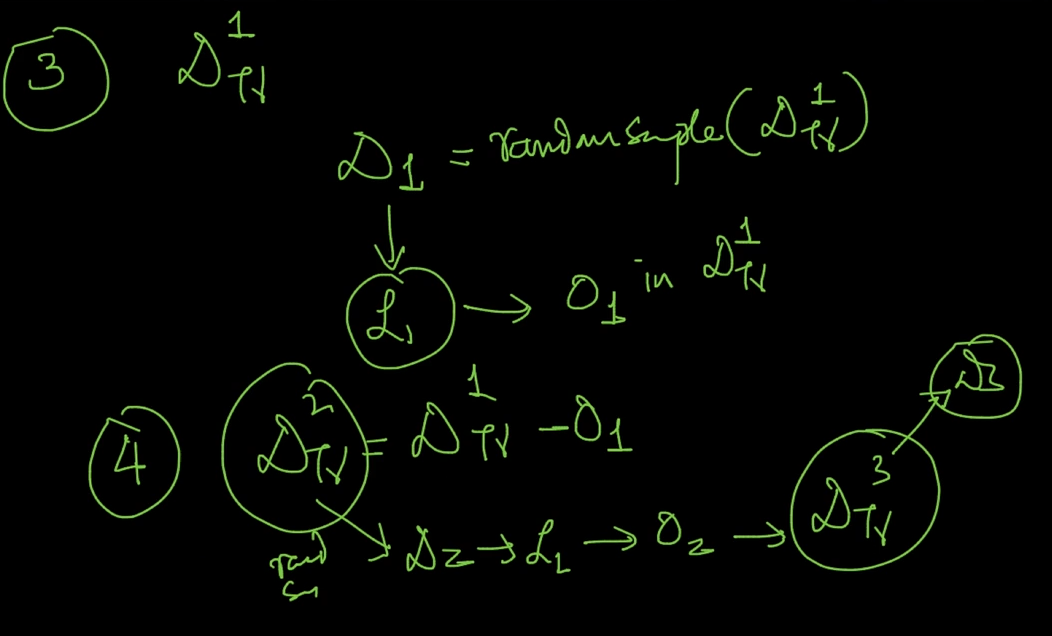
Can we build a robust model in presence of outliers. Robust model means some model which is not impacted by Outliers

As seen is the ideal fit but if we train a model we’ll get which is non-robust since it’s impacted by Outliers. Since upper side outliers are more it’s leaning towards that side as seen above.

We want to minimize and in LInear it’s squared loss

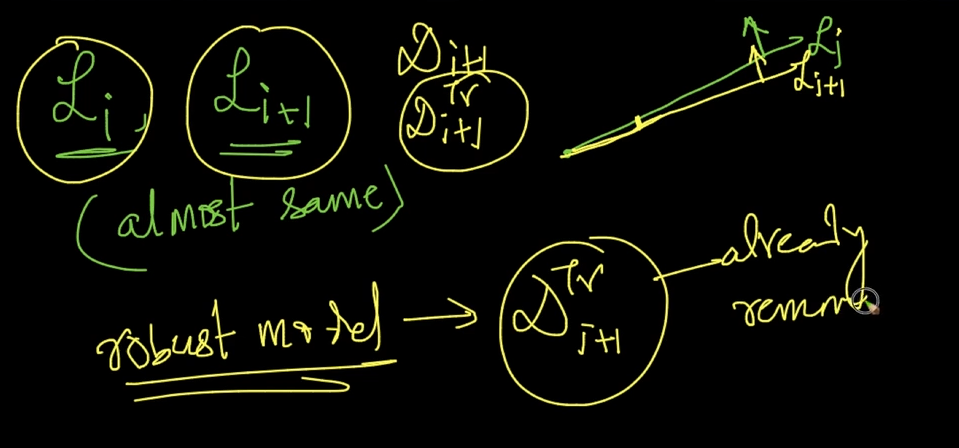


1 ) We select = random sample i.e 50 % of . It means 50% of Outliers won’t come in . We train a model on

2 ) We compute outliers on using model i.e from . Outliers are points which have high ). We get a set {Outliers}. Now , = 

We get = random sample () and train and compute outliers . =

We get using and train model and keep repeating the models on and on

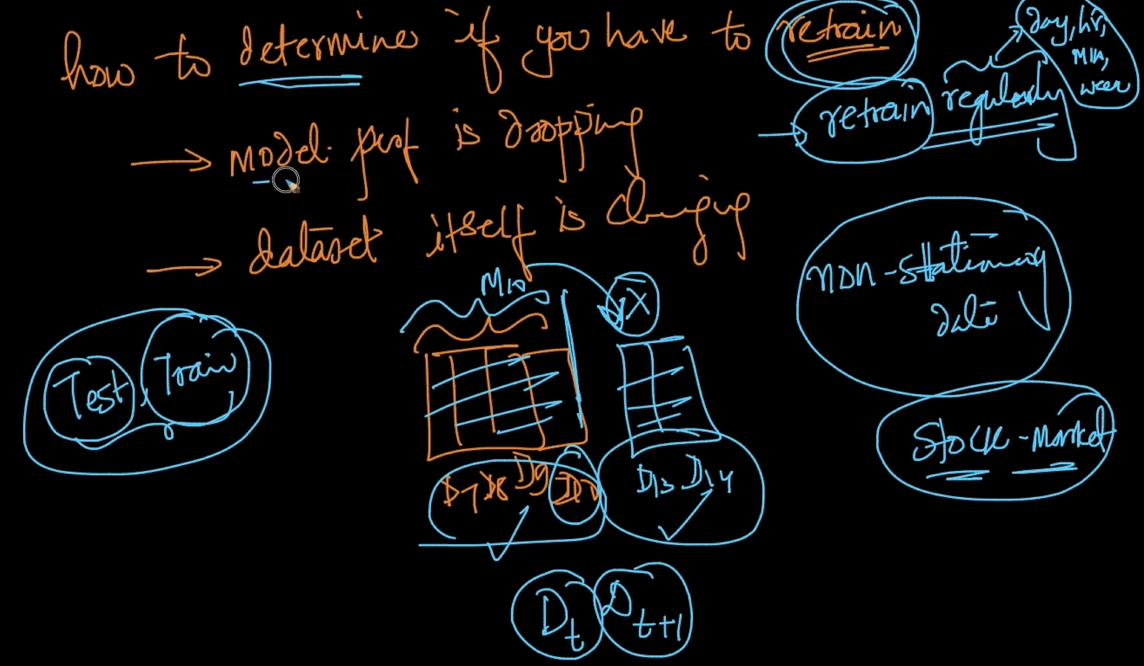


If is almost same then we select as it’s using which is using a random sample of which is a robust model since all the outliers are removed here. Now we build a robust model using

# **RETRAINING MODELS PERIODICALLY**

We’ve trained model using 10 days of data and checking AUC score for days as seen

Day 13’s AUC got low so that we can’t use . So we try to use more recent data and for that we retrain and get model



These are the points on determining when to retrain your model. If dataset is changing then we’ve to retrain your model