Logistic regression theory and metrics

20 January 2023 02:13 PM

Classification

In regression the output that we predict is a continuous variable but suppose I want to get my output as some categories

eg:

	Runs	home town	career wickets (bowled)	W (runout)	W (cb)	Feilding point	Batsman, Bowler, WK
→	5624	chenn.	8	16	12	120	Bat
→	4891	KOLK	7	14	38	200	WK
→	1024	Delhi	38	16	8	250	Bowler

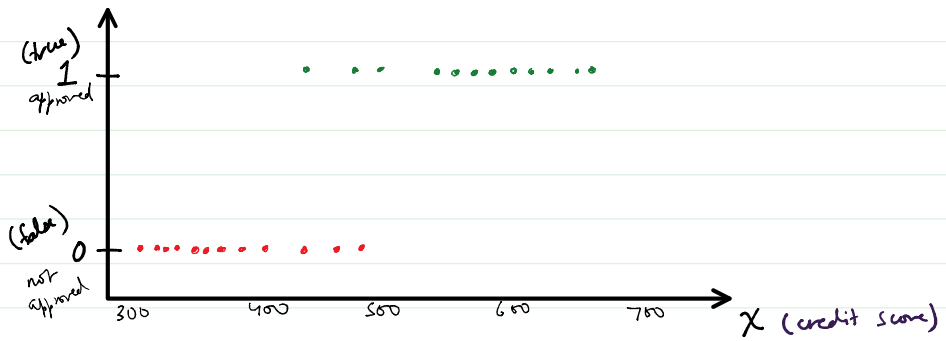
When numerical/categorical features are together used to [classify] an outcome & not give a continuous answer, this type of algorithm is called a classification algorithm.

Regression vs classification

→ Both are supervised learning algorithms
→ output of both is a different kind of variable (cont. vs. categorical)
→ regression usually fits a line to your data
→ classification uses other methods, discussed below.

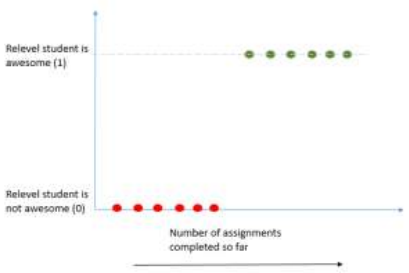
Problem statement

Given a person's credit score will his loan get approved or not?



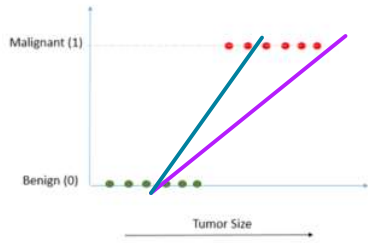
- Used for Binary Classification - Logistic Regression is used for Binary Classification (most of the times). For example, its job is usually to predict whether something is True or False, Big or Small, Dead or Alive etc. To make things easier we usually assign 0 and 1 labels to these 2 Target labels. Thus, in more simpler terms, Logistic regression predicts something to belong to either 0 or 1.

This is contrary to Linear Regression which can predict over a range of values.



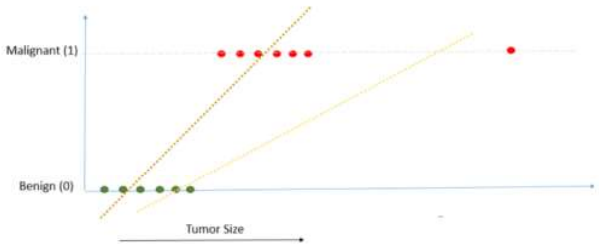
Why linear regression fails here: different example

Let's plot this on a chart as below –



(including ★)
Linear regression line is not a good fit on this data
Linear regression w/o ★

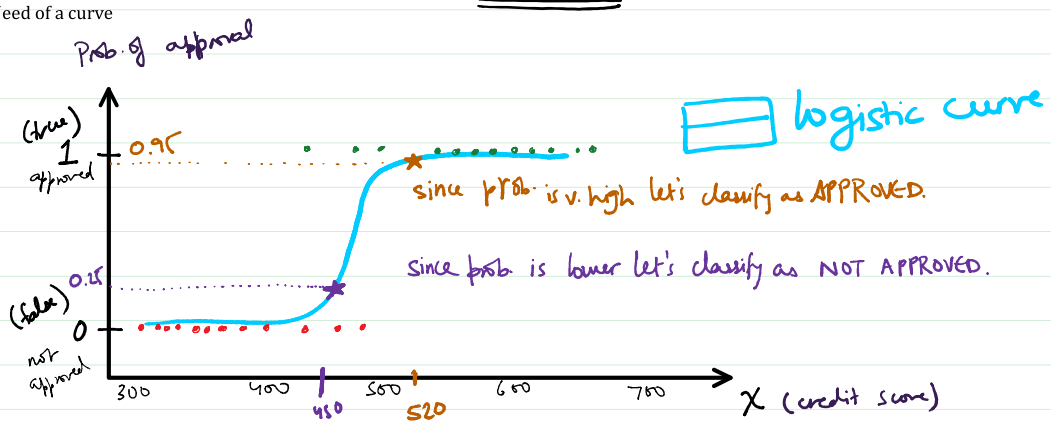
Adding the regression and seeing effect of a single outlier and instability among the rest



Linear regression curve fails in this case :
1) It's not a good fit to start with
2) Outliers have a massive effect in this type of data.

LOGISTIC REGRESSION

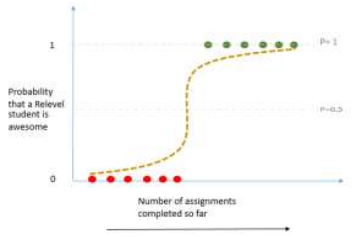
Need of a curve



Logistic Regression is called a regression because under the hood it predicts the probability of an event (a no. b/w 0 & 1) and using that classifies the output to give an estimate.

- Fits and S Shaped Curve - Instead of fitting a straight line to the data, Logistic Regression fits and S shaped Curve. The range of this Curve is 0 to 1 on y Axis, and Y axis represents Probability. Probability 0 belongs to –ve class and 1 belongs to +ve class.

We'll get into high level details of how Logistic Regression fits this S curve in a few minutes.



Relating X (features) to a p and intro of bernoulli

we need to take a continuous value (values) and convert it into a probability measure using the data we already have.

linear combination of features: $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \rightarrow$ single number

this single number holds information of all the features and all the coefficients so we will try to derive a probability value using this linear combination.

Differences between Linear Regression and Logistic Regression –

- Linear Regression uses Residuals and Least Squared method. It also calculates R2 for comparison. Logistic regression does not and cannot use Residuals or Least squared method. It uses a concept called *Maximum Likelihood*, which we'll discuss in this session. An R2 value cannot be derived for Logistic regression. (Although theoretically in advance mathematics , statisticians do calculate R2 for Logistic regression sometimes, but there are multiple methods and there is little consensus among statisticians on which method should be accepted as standard.)
- The range for Logistic Regression predictor is 0 to 1 , while for Linear regression is can be any real number.
- Logistic Regression is used for Classification and Linear regression is used for regression models.

Probability, odds, logit, log(odds)

1) Probability : 'p' → $\frac{\#\text{ successes}}{\#\text{ total trials}}$ → prob. of success

2) Odds:

$1-p \rightarrow \frac{\#\text{ failures}}{\#\text{ total trials}}$ → prob. of failure

$\text{Odds} = \frac{p}{1-p}$ → how likely am I to get a success as compared to failure.

eg: Ind vs. SA history

played	36
Ind won	20
Ind lost	16

$p = \frac{20}{36} = \frac{10}{18} = \frac{5}{9}$

$1-p = \frac{16}{36} = \frac{4}{9}$

Odds of India winning = $\frac{5/9}{4/9} = \frac{5}{4} = 1.25$ {odds can range from 0 to ∞}

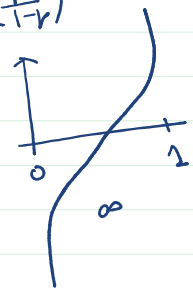
India is 1.25 x more likely to win than to lose.

3) logit → log of odds = $\ln\left(\frac{p}{1-p}\right)$ → $\frac{p}{1-p} \in [0, \infty)$

$\ln(1.25) = 0.2231$

$\log_e = \ln$ (natural log)

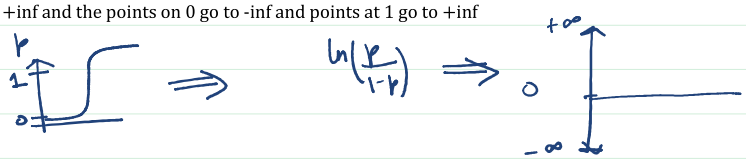
$\Rightarrow \ln\left(\frac{p}{1-p}\right) \in (-\infty, +\infty)$



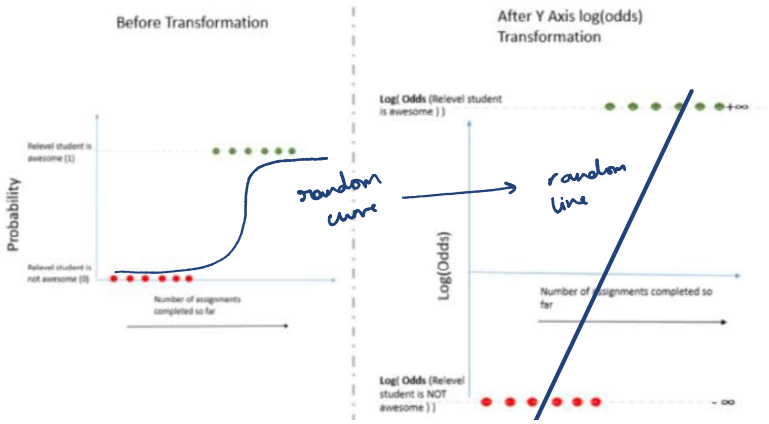
Why to do log of odds:
1. We need a curve so : a--> log gives an S shape and b--> log needs 0 to inf
2. Assymetry of odds! (losing and winning eg of matches) takong log odds will give ans bw -inf to +inf

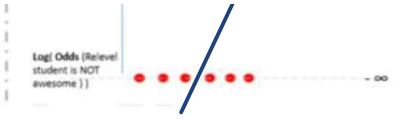
Fitting the logistic curve {we are more comfortable fitting a straight line, so we take a random curve and convert that to a line. Optimize the line so essentially a reverse transform of that line will give us an optimized curve}

Step1: transform the y axis from p to logit(p) so that the range is now -inf to +inf and the points on 0 go to -inf and points at 1 go to +inf



This transformation is shown in visuals below for better grasp :

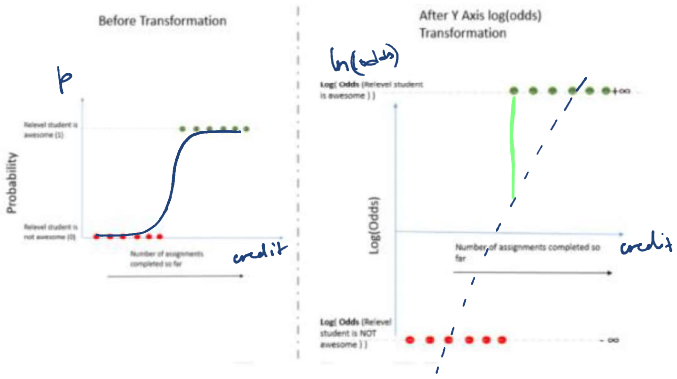




TRANSFORMATION

Initially Y-axis had p, now we're transforming it to $\ln\left(\frac{p}{1-p}\right)$

This transformation is shown in visuals below for better grasp :



squiggle converted to a straight line

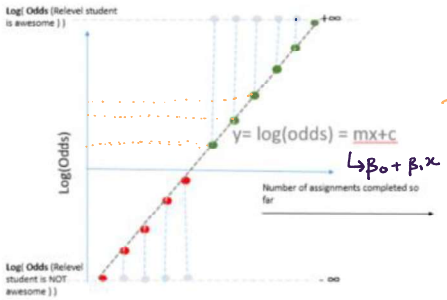
like in previous class we used sum of squared errors to minimize & find β_0, β_1
here errors are essentially as (——) too to some y-value is ∞.

Thus we do something else

Step 2: finding log odds of these points using this straight line and x axis coordinate of these points

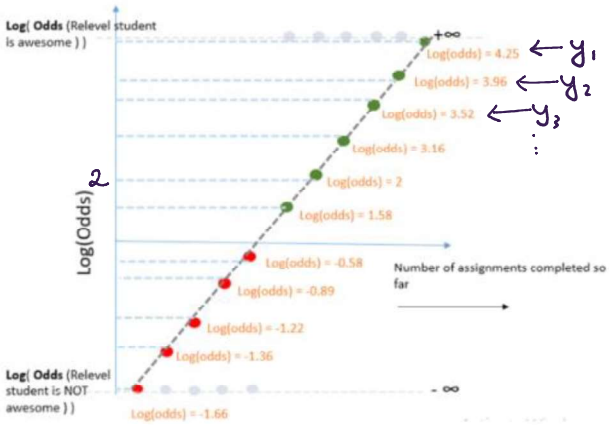
Since transformation is made only on the Y-axis, the X-axis values (credit scores) remain as it is and we know them.

Since the datapoints are at extreme ends on this axis (+∞ an - ∞) , they are projected on to the line as shown in the visual below.



finding y-axis value (log odds) of these projected points.

After projection, each data point will have a corresponding log(odds) value.



Step 3: getting back probabilities from this line to go back to a curve

now essentially this line is used to see the log odds, and we know log odds can be converted back to p, so we inverse transform step 1 and get back the curve

$$\ln\left(\frac{p}{1-p}\right) = y_1$$

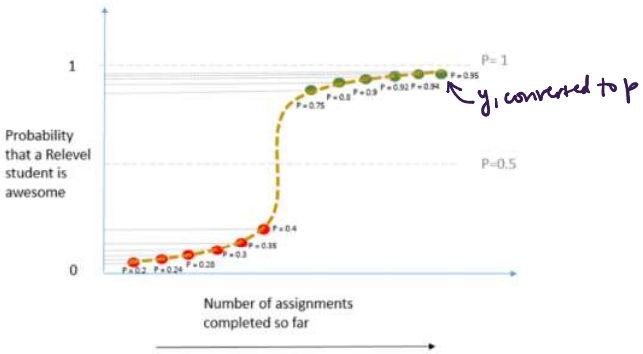
$$\Rightarrow \frac{p}{1-p} = e^{y_1} \Rightarrow p = e^{y_1} - pe^{y_1} \Rightarrow p + pe^{y_1} = e^{y_1}$$

$$\Rightarrow p(1+e^{y_1}) = e^{y_1}$$

$$\Rightarrow \left(p = \frac{e^{y_1}}{1+e^{y_1}}\right)$$

INVERSE TRANSFORMATION

Geometrically, this function is transforming back out log(odds) axis to Probabilities. Our Straight line is convert into an S-Curve with the help of this function.



y_1 converted to p

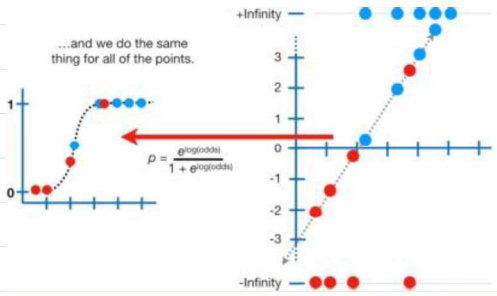
eg. y_1 was 4.25 $\Rightarrow$ $p = \frac{e^{y_1}}{1+e^{y_1}} = \frac{e^{4.25}}{1+e^{4.25}} = 0.9859$

y_2 was 3.96 $\Rightarrow$ $p = \frac{e^{y_2}}{1+e^{y_2}} = \frac{e^{3.96}}{1+e^{3.96}} = 0.9812$

Step 4: calculating the likelihood function as a measure of fit

REMEMBER: Initially we just knew that we need an S-shaped curved but now that we've got a random curve we need to optimize it.

At this stage we use likelihood. Likelihood function here is just the product of the probability values which were inverse transformed w/ the help of original values that we know should be classified to 0 or 1.



now red points we know should be classified as 0 ∵ that's their true value
and blue points we know should be classified as 1 ∵ that's their true value.

So drawing analogy from the previous class as we calculated $(y_i - \hat{y}_i)$ as errors we can do the following:

For blue points (y_i) [actual values are 1] so points with calculated probability close to 1 are good.

∴ we multiply all of them. $p_{b_1} \times p_{b_2} \times p_{b_3} \times \ldots$ — (1)

For red points (y_i) [actual values are 0] so any points with calculated probability close to 1 are bad.

∴ we multiply all of them after subtracting them by 1 $(1-p_{r_1}) \times (1-p_{r_2}) \times (1-p_{r_3}) \times \ldots$ — (2)

since for red points we need prob. of NOT happening we use 1-p.

Likelihood is given as ① × ② i.e.

$$L = p_{b_1} \times p_{b_2} \times p_{b_3} \times \cdots \times (1-p_{r_1}) \times (1-p_{r_2}) \times (1-p_{r_3}) \times \cdots$$

and the curve which maximizes this L is the optimal curve and this is called Maximizing the likelihood function.

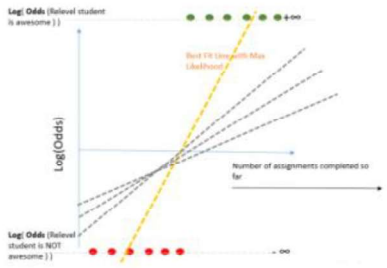
Random curve led to a random line of random β_0, β_1 in $\beta_0 + \beta_1 x$ now optimal β_0, β_1 will be found.

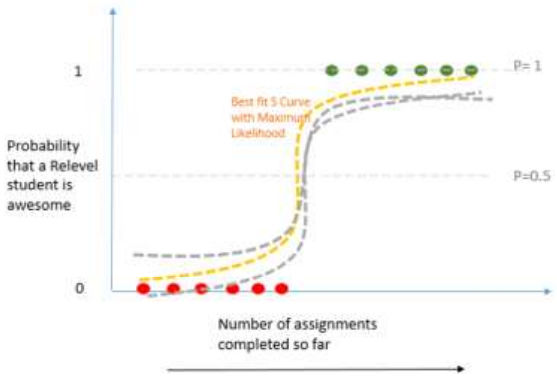
Step 5 : iterations to find the best curve based on MLE

Now the initial straight line with y axis as log(odds)) will be rotated as per Gradient Descent process discussed earlier. A new straight line will be obtained, again projections would be taken and maximum likelihood would be calculated for the new S Curve obtained.

This iterative process will happen until the algorithm will reach minima on Gradient Descent, and that minima will be the configuration at which we'll get maximum likelihood.

At that configuration, the model will have its best-fit line on log(odds) axis ; and corresponding S-Curve on original axis with y axis as Probabilities.





How a computer finds it: Gradient descent introduction

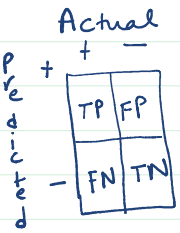
It changes the model and sees in which direction it's getting a better fit and then by fine tuning it finds the optimal curve in just a few steps.

Measures of goodness in a classification problem

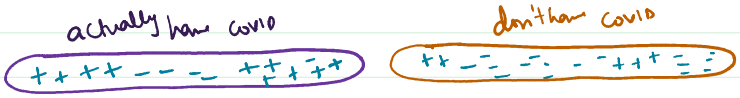
Confusion matrix

TP (a) (b) FP (type I error)

FN (c) (type II error) (d) TN



Accuracy = $\frac{a+d}{a+b+c+d}$



1. Recall score / True Positive Rate (TPR) / Sensitivity (how many + samples were correctly classified)

higher is better

For a binary classification, it is a score which tells us the ratio of **correct +ve class** predictions with respect to **all +ve class predictions**. Recall is also known as True Positive Rate. We'll look at the mathematical formula for better understanding -

$$Recall\ (TPR) = \frac{TP}{(TP+FN)} = \frac{a}{a+c}$$

Actual

Predicted	Apple	Orange	Banana
Apple	3 (TP)	4	2
Orange	1 (FN)	8	6
Banana	4 (FN)	5	2

$\frac{3}{3+1+4} = \frac{3}{8}$ recall of apple

$\frac{\text{TP for Apple}}{\text{Total actual apple}}$

2. Specificity / True Negative Rate (TNR)

higher is better

$$Specificity = \frac{TN}{TN+FP} = \frac{d}{d+b}$$

3. Precision

For a binary classification, It is the ratio of correct +ve class predictions with respect to Total +ve class predictions. Mathematically,

$$Precision = \frac{TP}{(TP+FP)} \quad \frac{a}{a+b}$$

Actual

Predicted	Apple	Orange	Banana
Apple	3	4	2
Orange	1	8	6
Banana	4	5	2

$Precision = \frac{3}{\quad} = \frac{3}{9} = \frac{1}{3}$

3+4+2

4. False Positive Rate (FPR) / 1-specificity (how many — samples were incorrectly classified)

lower is better

$$FPR = \frac{FP}{FP + TN}$$

5. F beta score

Now that we have understood Precision and Recall, it is easy to get the idea behind F-Beta.

Generally, there might be cases where both Type-1 and Type-2 errors are of huge impact and it is important to control both. Or there may be cases where although both are important, but comparatively controlling Type-1 is of more importance than Type-2 and vice versa.

Precision and Recall only help us in controlling either Type-1 or Type-2 error, but not both at the same time.

This is achieved by the use of F-Beta Score.

The mathematical formula for F-Beta is given as :

$$F_{Beta} = (1 + \beta^2) \frac{Precision * Recall}{(\beta^2 * Precision) + Recall}$$

There are 2 important things to note here –
1. F Beta uses both Precision and Recall in its formula, and thus it is able to take into account both
2. There is a variable β which has following significance :

At $\beta = 1$,
the formula becomes :

$$F_{Beta} = \frac{2 * Precision * Recall}{Precision + Recall}$$

This is nothing but the harmonic mean of 2 metrics. Which basically means that both are given equal significance.
This formula at $\beta = 1$ is also known as **F1 Score**.

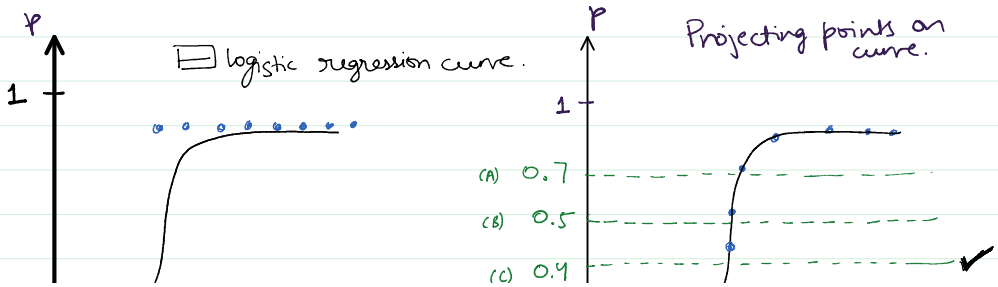
At $\beta = 0.5$,
The coefficient with Precision in denominator is squared, and is less than 1, thus Precision gets more significance overall.

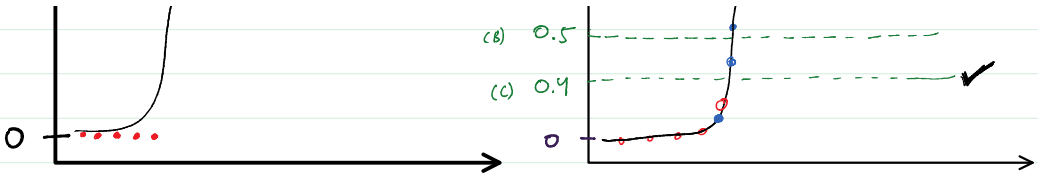
Thus, $\beta < 1$, Type-I errors are controlled better than Type-2 errors. Lower the value for β, more prominent the role o
Precision is.

On the other hand, at $\beta > 1$, Type-2 errors are controlled better than Type-1 errors. Higher the value for β, more prominent the role of Recall becomes.

ROC curve and AUC — Receiver Operating Characteristic Curve

Threshold.





Confusion Matrix

Threshold-B $\begin{bmatrix} 6 & 2 \\ 0 & 5 \end{bmatrix}$ acc: $\frac{11}{13}$ {0.5}

Threshold-A $\begin{bmatrix} 5 & 3 \\ 0 & 5 \end{bmatrix}$ acc: $\frac{10}{13}$ {0.7}

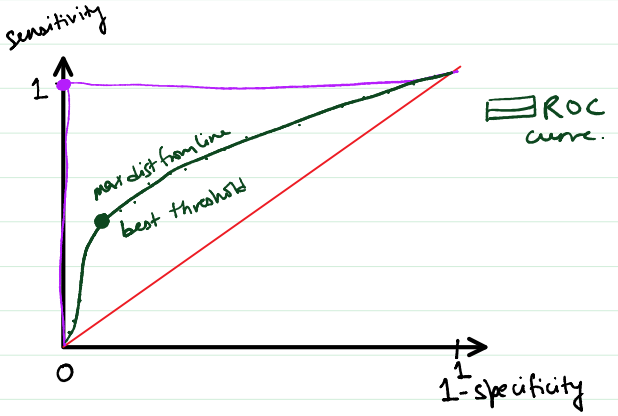
Threshold-C $\begin{bmatrix} 7 & 1 \\ 0 & 5 \end{bmatrix}$ acc: $\frac{12}{13}$ {0.4}

How changing a threshold can help

changing a threshold can make different confusion matrices and each of them have different accuracies. The threshold with the best accuracy can be selected as the final model. But since making confusion matrix is not feasible manually for each threshold, we make an ROC curve.

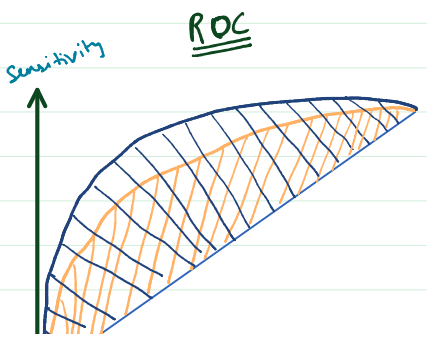
Selecting best threshold using ROC

ROC curve:



ROC curve.

- Higher sensitivity ⇒ better + prediction
  Lower (1-specificity) => Higher specificity ⇒ better − prediction
- Plot for various thresholds
- One ROC for one model.
- AUC is used to compare multiple models
- Best classifier → max AUC

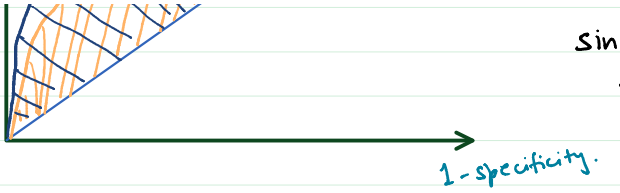


Algorithm 1
Algorithm 2

- Theoretically best ROC

AUC 1
AUC 2

Since AUC 2 > AUC 1
Algorithm 2 is better



1 - specificity.

since AUC 2 > AUC 1

Algorithm 2 is better for classification.

AUC: area under curve.