Model Performance Metrics

Team Id	NM2023TMID04410
Project Name	Project-Drug Traceability

Introduction:

- Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms.
- When evaluating a machine learning model, it is crucial to assess its predictive ability, generalization capability, and overall quality. Evaluation metrics provide objective criteria to measure these aspects. The choice of evaluation metrics depends on the specific problem domain, the type of data, and the desired outcome.
- The choice of evaluation metric completely depends on the type of model and the implementation plan of the model. After you are finished building your model, these 12 metrics will help you in evaluating your model's accuracy. Considering the rising popularity and importance of cross-validation, I've also mentioned its principles in this article.

Types of Predictive Models:

- When we talk about predictive models, we are talking either about a regression model (continuous output) or a classification model (nominal or binary output). The evaluation metrics used in each of these models are different.In classification problems, we use two types of algorithms

- When we talk about predictive models, we are talking either about a regression model (continuous output) or a classification model (nominal or binary output). The evaluation metrics used in each of these models are different.
- In classification problems, we use two types of algorithms (dependent on the kind of output it creates):

1. **Class output:** Algorithms like SVM and KNN create a class output. For instance, in a binary classification problem, the outputs will be either 0 or 1. However, today we have algorithms that can convert these class outputs to probability. But these algorithms are not well accepted by the statistics community.
2. **Probability output:** Algorithms like Logistic Regression, Random Forest, Gradient Boosting, Adaboost, etc., give probability outputs. Converting probability outputs to class output is just a matter of creating a threshold probability.

In regression problems, we do not have such inconsistencies in output. The output is always continuous in nature and requires no further treatment.

Illustrative Example

For a classification model evaluation metric discussion, I have used my predictions for the problem BCI challenge on Kaggle. The solution to the problem is out of the scope of our discussion here. However, the final predictions on the training set have been used for this article. The predictions made for this problem were probability outputs which have been converted to class outputs assuming a threshold of 0.5.

F1 Score:

- In the last section, we discussed precision and recall for classification problems and also highlighted the importance of choosing a precision/recall basis for our use case.
- What if, for a use case, we are trying to get the best precision and recall at the same time? F1-Score is the harmonic mean of precision and recall values for a classification problem. The formula for F1-Score is as follows:

$$F_1 = \left(\frac{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Now, an obvious question that comes to mind is why you are taking a harmonic mean and not an arithmetic mean. This is because HM punishes extreme values more. Let us understand this with an example. We have a binary classification model with the following results:

Precision: 0, Recall: 1

Here, if we take the arithmetic mean, we get 0.5. It is clear that the above result comes from a dumb classifier that ignores the input and predicts one of the classes as output. Now, if we were to take HM, we would get 0 which is accurate as this model is useless for all purposes.

This seems simple. There are situations, however, for which a data scientist would like to give a percentage more importance/weight to either precision or recall. Altering the above expression a bit such that we can include an adjustable parameter beta for this purpose, we get:

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}.$$

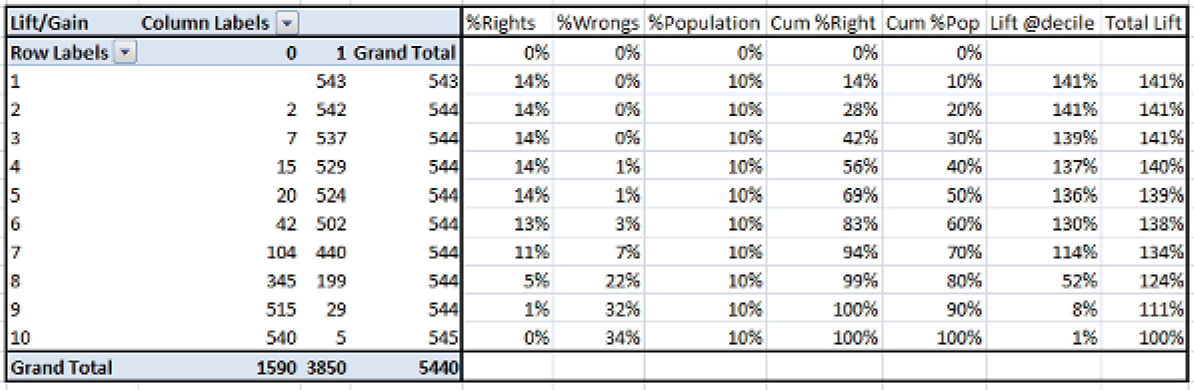
Fbeta measures the effectiveness of a model with respect to a user who attaches β times as much importance to recall as precision.

Gain and Lift Charts:

Gain and Lift charts are mainly concerned with checking the rank ordering of the probabilities. Here are the steps to build a Lift/Gain chart:

- Step 1: Calculate the probability for each observation
- Step 2: Rank these probabilities in decreasing order.
- Step 3: Build deciles with each group having almost 10% of the observations.
- Step 4: Calculate the response rate at each decile for Good (Responders), Bad (Non-responders), and total.

You will get the following table from which you need to plot Gain/Lift charts:



Lift/Gain	Column Labels			%Rights	%Wrongs	%Population	Cum %Right	Cum %Pop	Lift @decile	Total Lift
Row Labels	0	1	Grand Total	0%	0%	0%	0%	0%		
1		543	543	14%	0%	10%	14%	10%	141%	141%
2	2	542	544	14%	0%	10%	28%	20%	141%	141%
3	7	537	544	14%	0%	10%	42%	30%	139%	141%
4	15	529	544	14%	1%	10%	56%	40%	137%	140%
5	20	524	544	14%	1%	10%	69%	50%	136%	139%
6	42	502	544	13%	3%	10%	83%	60%	130%	138%
7	104	440	544	11%	7%	10%	94%	70%	114%	134%
8	345	199	544	5%	22%	10%	99%	80%	52%	124%
9	515	29	544	1%	32%	10%	100%	90%	8%	111%
10	540	5	545	0%	34%	10%	100%	100%	1%	100%
Grand Total	1590	3850	5440							

This is a very informative table. The cumulative Gain chart is the graph between Cumulative %Right and Cumulative %Population. For the case in hand, here is the graph:

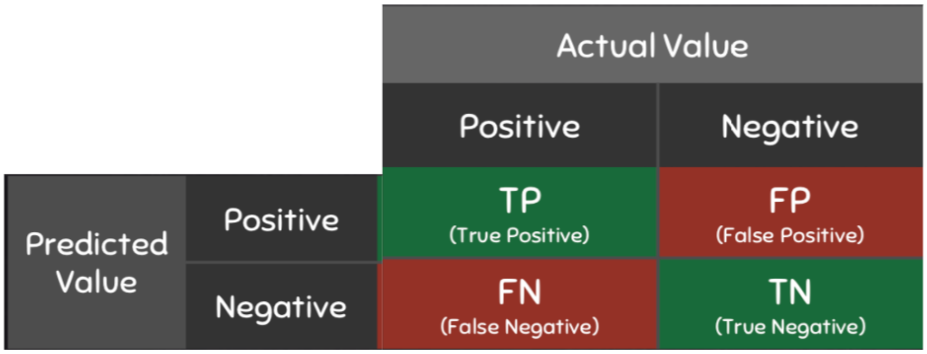
Types of Predictive Models:

- When we talk about predictive models, we are talking either about a regression model (continuous output) or a classification model (nominal or binary output). The evaluation metrics used in each of these models are different.In classification problems, we use two types of algorithms (dependent

Confusion Matrix:

- Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model.
- The confusion matrix provides a more insightful picture which is not only the performance of a predictive model, but also which classes are being predicted correctly and incorrectly, and what type of errors are being made.

- To illustrate, we can see how the 4 classification metrics are calculated (TP, FP, FN, TN), and our predicted value compared to the actual value in a confusion matrix is clearly presented in the below confusion matrix table.



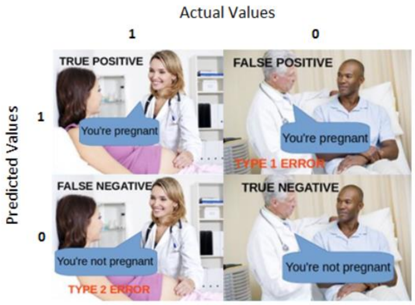
		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

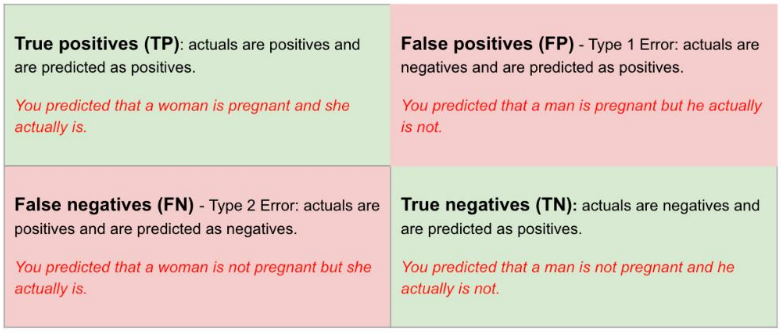
- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Possible Classification Outcomes: TP, FP, FN, TN.

The confusion matrix is useful for measuring Recall (also known as Sensitivity), Precision, Specificity, Accuracy, and, most importantly, the AUC-ROC Curve.

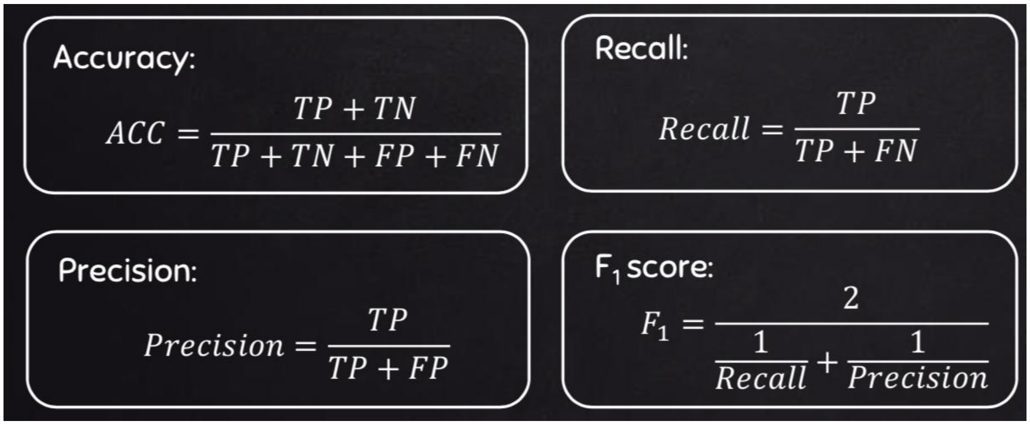
Do you feel confused when you were reading the table? That’s expected. I was also before. Let me put it in an interesting scenario in terms of pregnancy analogy to explain the terms of TP, FP, FN, TN. We can then understand Recall, Precision, Specificity, Accuracy, and, most importantly, the AUC-ROC Curve.





True positives (TP): actuals are positives and are predicted as positives. *You predicted that a woman is pregnant and she actually is.*	**False positives (FP)** - Type 1 Error: actuals are negatives and are predicted as positives. *You predicted that a man is pregnant but he actually is not.*
False negatives (FN) - Type 2 Error: actuals are positives and are predicted as negatives. *You predicted that a woman is not pregnant but she actually is.*	**True negatives (TN):** actuals are negatives and are predicted as positives. *You predicted that a man is not pregnant and he actually is not.*

The Equations of 4 Key Classification Metrics



Accuracy:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall:

$$Recall = \frac{TP}{TP + FN}$$

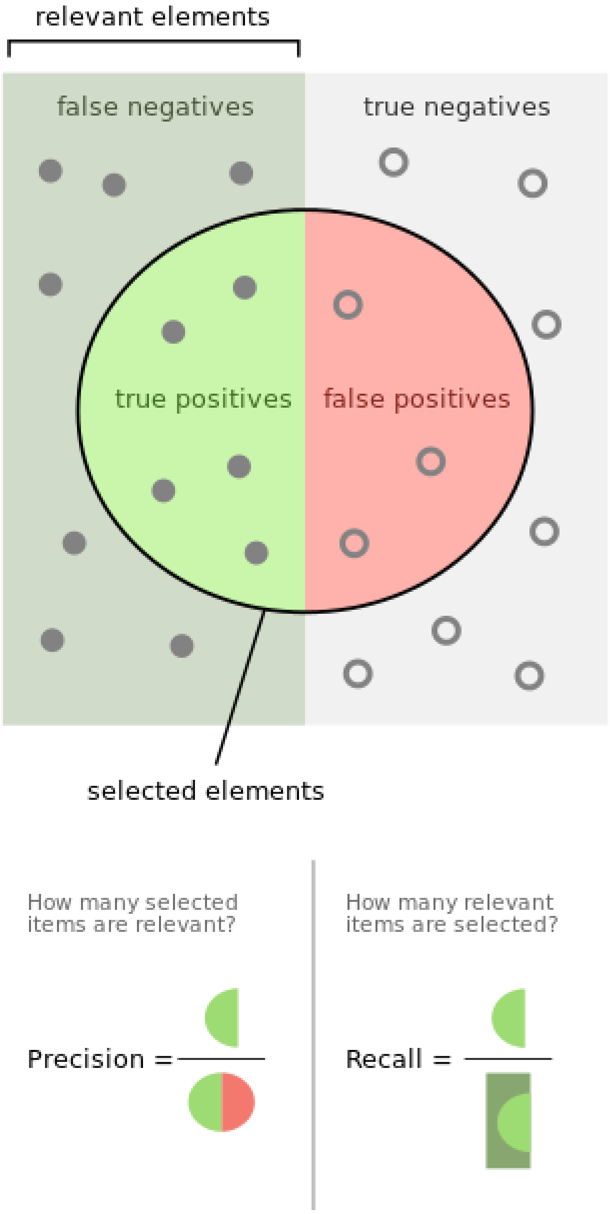
Precision:

$$Precision = \frac{TP}{TP + FP}$$

F_1 score:

$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Recall versus Precision



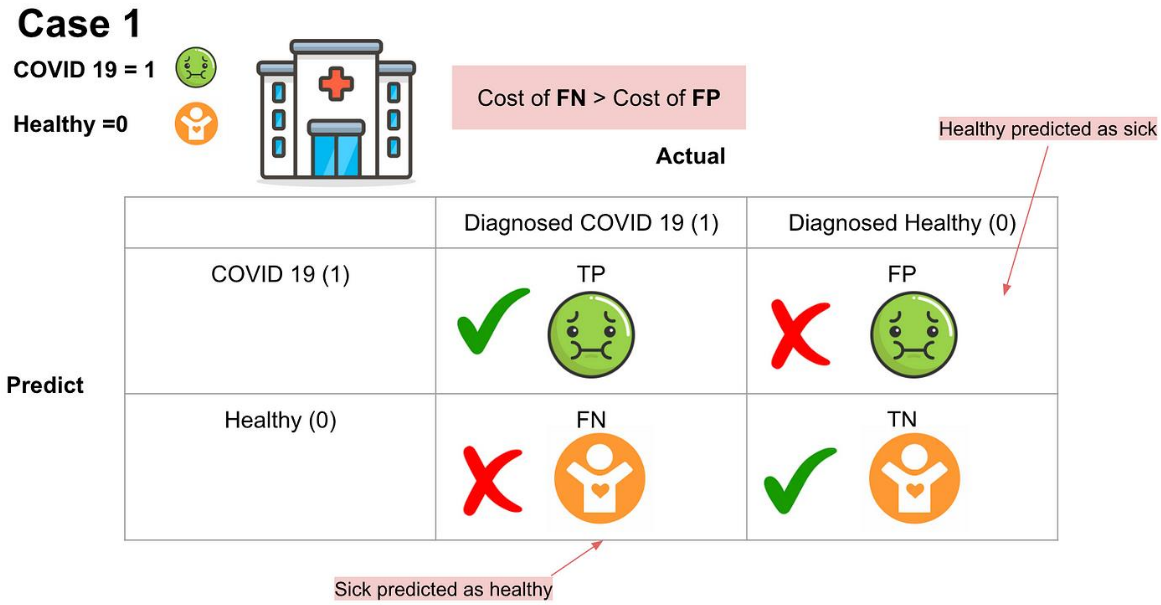
Precision is the ratio of *True Positives* to all the positives predicted by the model.

Low precision: the more False positives the model predicts, the lower the precision.

Recall (Sensitivity)is the ratio of *True Positives* to all the positives in your Dataset.

Low recall: the more False Negatives the model predicts, the lower the recall.

The idea of recall and precision seems to be abstract. Let me illustrate the difference in three real cases.



Case 1

COVID 19 = 1

Healthy =0

Cost of **FN** > Cost of **FP**

Actual

Predict

	Diagnosed COVID 19 (1)	Diagnosed Healthy (0)
COVID 19 (1)	TP ✓	FP ✗
Healthy (0)	FN ✗	TN ✓

Healthy predicted as sick

Sick predicted as healthy

- the result of TP will be that the COVID 19 residents diagnosed with COVID-19.
- the result of TN will be that healthy residents are with good health.
- the result of FP will be that those actually healthy residents are predicted as COVID 19 residents.
- the result of FN will be that those actual COVID 19 residents are predicted as the healthy residents

In case 1, which scenario do you think will have the highest cost?

Imagine that if we predict COVID-19 residents as healthy patients and they do not need to quarantine, there would be a massive number of COVID-19 infections. The cost of f*alse negatives* is much higher than the cost of f*alse positives.*

Case 2

Spam = 1

Not Spam =0

Cost of **FP** > Cost of **FN**

Not spam predicted as spam

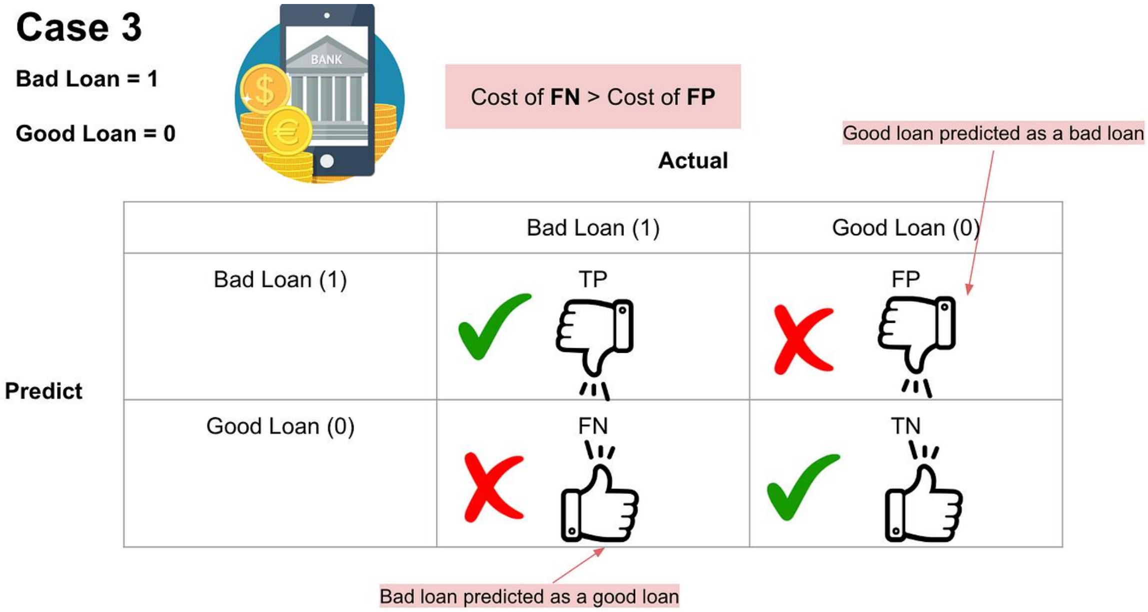
		Actual	
		Spam (1)	Not Spam (0)
Predict	Spam (1)	TP ✓	FP ✗
	Not Spam (0)	FN ✗	TN ✓

Spam predicted as not spam

- the result of TP will be that spam emails are placed in the spam folder.
- the result of TN will be that important emails are received.
- the result of FP will be that important emails are placed in the spam folder.
- the result of FN will be that spam emails are received.

In case 2, which scenario do you think will have the highest cost?

Well, since missing important emails will clearly be more of a problem than receiving spam, we can say that in this case, FP will have a higher cost than FN.



Case 3

Bad Loan = 1

Good Loan = 0

Cost of **FN** > Cost of **FP**

Good loan predicted as a bad loan

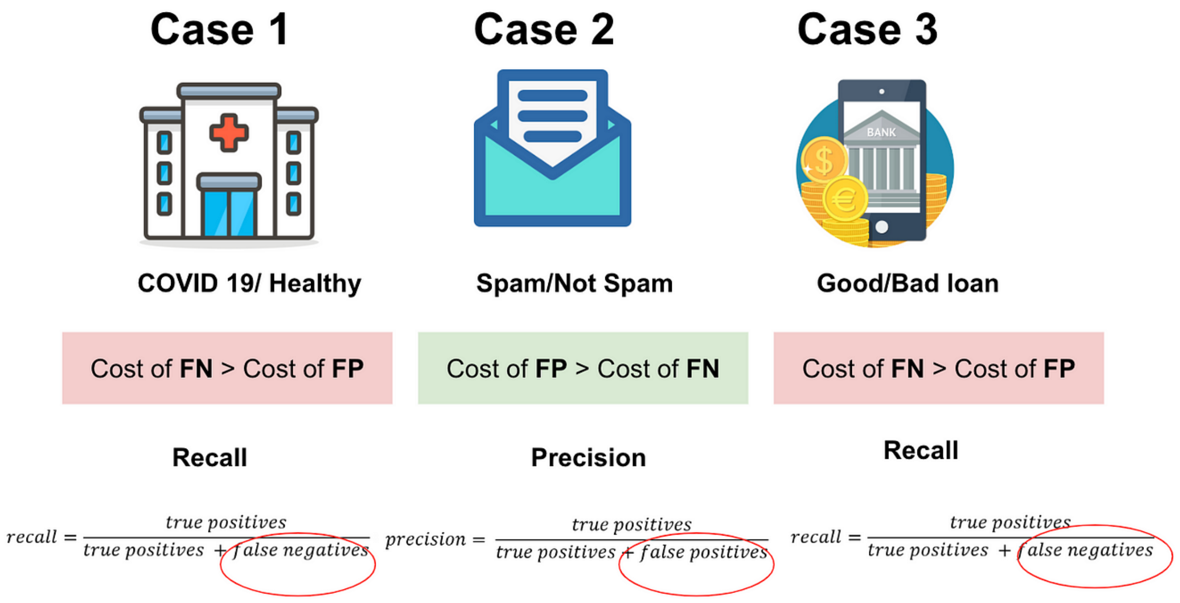
		Actual	
		Bad Loan (1)	Good Loan (0)
Predict	Bad Loan (1)	TP ✓ 👎	FP ✗ 👎
	Good Loan (0)	FN ✗ 👍	TN ✓ 👍

Bad loan predicted as a good loan

- the result of TP will be that bad loans are correctly predicted as bad loans.
- the result of TN will be that good loans are correctly predicted as good loans.
- the result of FP will be that (actual) good loans are incorrectly predicted as bad loans.
- the result of FN will be that (actual) bad loans are incorrectly predicted as good loans.

In case 3, which scenario do you think will have the highest cost?

The banks would lose a bunch amount of money if the actual bad loans are predicted as good loans due to loans not being repaid. On the other hand, banks won't be able to make more revenue if the actual good loans are predicted as bad loans. Therefore, the cost of *False Negatives* is much higher than the cost of *False Positives.* Imagine that.



Case 1

COVID 19/ Healthy

Cost of **FN** > Cost of **FP**

Recall

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Case 2

Spam/Not Spam

Cost of **FP** > Cost of **FN**

Precision

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Case 3

Good/Bad loan

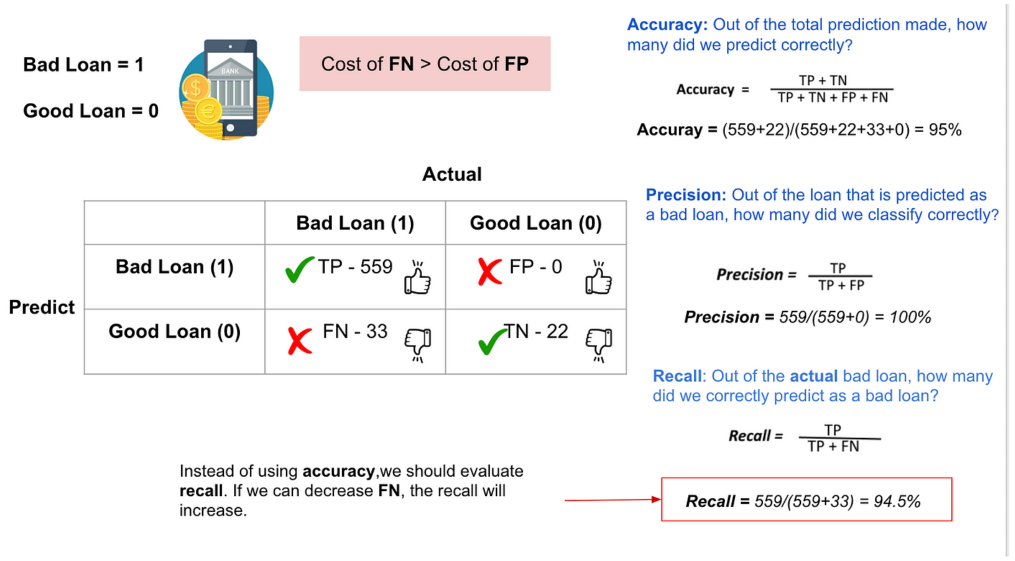
Cost of **FN** > Cost of **FP**

Recall

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Combining Precision and Recall:

In the above three cases, we want to maximize either recall or precision at the expense of the other metric. For example, in the case of a good or bad loan classification, we would like to decrease FN to increase recall. However, in cases where we want to findan optimal blend of precision and recall, we can combine the two metrics usinthe F1



Bad Loan = 1

Good Loan = 0

Cost of **FN** > Cost of **FP**

Actual

Predict

	Bad Loan (1)	**Good Loan (0)**
Bad Loan (1)	✓ TP - 559 👍	✗ FP - 0 👍
Good Loan (0)	✗ FN - 33 👎	✓ TN - 22 👎

Instead of using **accuracy**,we should evaluate **recall**. If we can decrease **FN**, the recall will increase.

Accuracy: Out of the total prediction made, how many did we predict correctly?

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuray = (559+22)/(559+22+33+0) = 95%

Precision: Out of the loan that is predicted as a bad loan, how many did we classify correctly?

$$Precision = \frac{TP}{TP + FP}$$

***Precision =** 559/(559+0) = 100%*

Recall: Out of the **actual** bad loan, how many did we correctly predict as a bad loan?

$$Recall = \frac{TP}{TP + FN}$$

***Recall =** 559/(559+33) = 94.5%*

F-Measure provides a single score that balances both the concerns of precision and recall in one number. A good F1 score means that you have low false positives and low false negatives, so you're correctly identifying real threats, and you are not disturbed by false alarms. An F1 score is considered perfect when it's 1, while the model is a total failure when it's 0.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Decision Threshold:

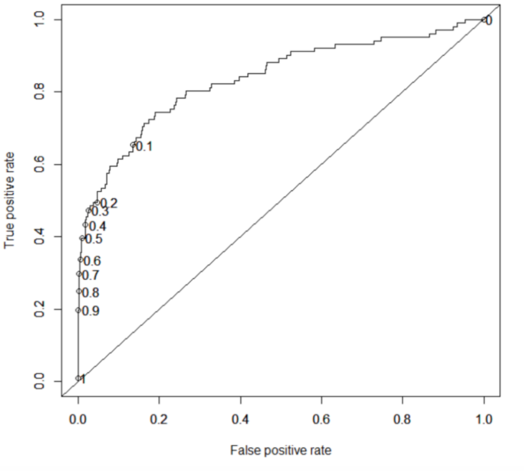
ROC is a major visualization technique for presenting the performance of a classification model. It summarizes the trade-off between the true positive rate (tpr) and false positive rate (fpr) for a predictive model using different probability thresholds.

$$true\ positive\ rate = \frac{true\ positives}{true\ positives + false\ negatives} \qquad false\ positive\ rate = \frac{false\ positives}{false\ positives + true\ negatives}$$

The equation of tpr and fpr.

The true positive rate (tpr) is the recall and the false positive rate (FPR) is the probability of a false alarm.

A ROC curve plots the true positive rate (tpr) versus the false positive rate (fpr) as a function of the model’s threshold for classifying a positive. Given that **c** is a constant known as decision threshold, the below ROC curve suggests that by default c=0.5, when c=0.2, both tpr and fpr increase. When c=0.8, both tpr and fpr decrease. In general, tpr and fpr increase as c decrease. In the extreme case when c=1, all cases are predicted as negative; tpr=fpr=0. On the other hand, when c=0, all cases are predicted as positive; tpr=fpr=1.



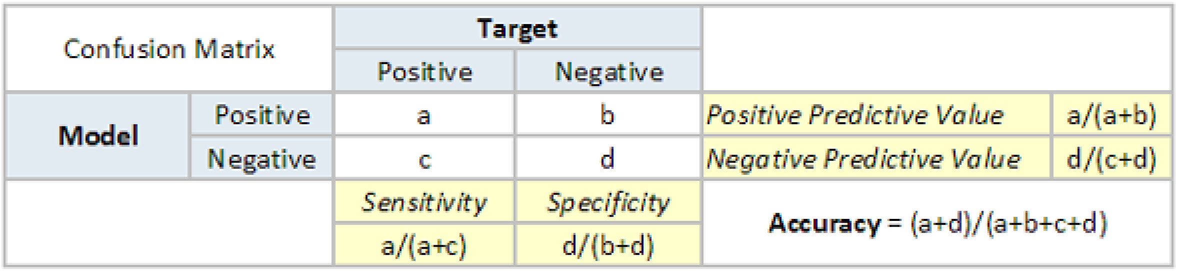
ROC chart.

Finally, we can assess the performance of the model by the area under the ROC curve (**AUC**). As a rule of thumb, 0.9–1=excellent; 0.8-.09=good; 0.7–0.8=fair; 0.6–0.7=poor; 0.50–0.6=fail.

Area Under the ROC Curve (AUC – ROC):

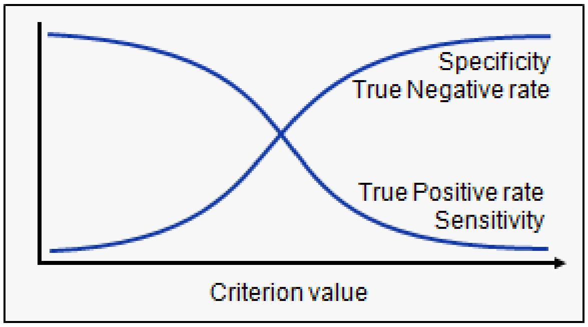
- This is again one of the popular evaluation metrics used in the industry. The biggest advantage of using the ROC curve is that it is independent of the change in the proportion of responders.
- This statement will get clearer in the following sections.

Let’s first try to understand what the ROC (Receiver operating characteristic) curve is. If we look at the confusion matrix below, we observe that for a probabilistic model, we get different values for each metric.

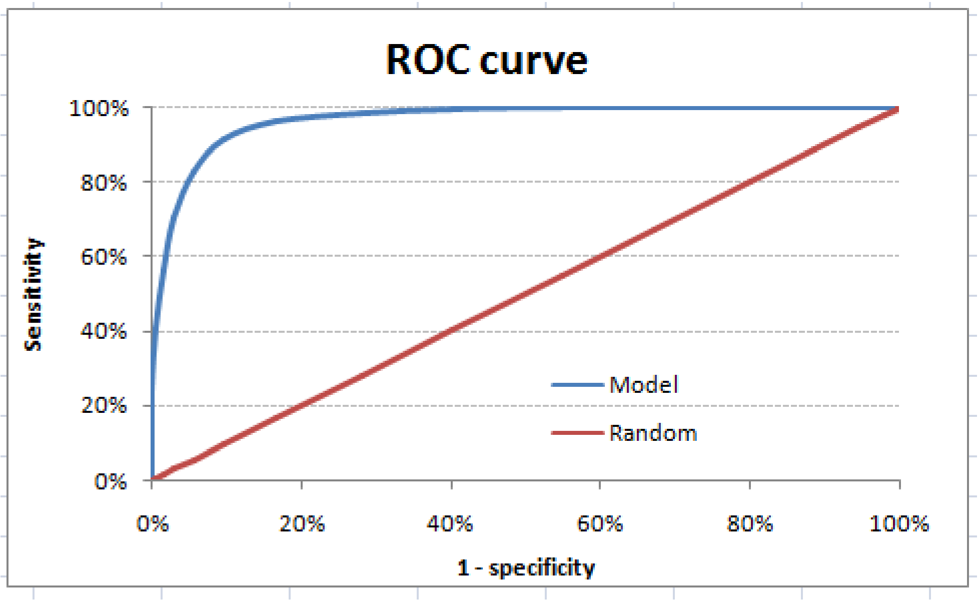


Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

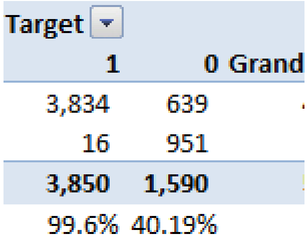
Hence, for each sensitivity, we get a different specificity. The two vary as follows:



The ROC curve is the plot between sensitivity and (1- specificity). (1- specificity) is also known as the false positive rate, and sensitivity is also known as the True Positive rate. Following is the ROC curve for the case in hand.



Let’s take an example of threshold = 0.5 (refer to confusion matrix). Here is the confusion matrix:



Target ▾		
1	0	Grand
3,834	639	
16	951	
3,850	**1,590**	
99.6%	40.19%	

As you can see, the sensitivity at this threshold is 99.6%, and the (1-specificity) is ~60%. This coordinate becomes on point in our ROC curve. To bring this curve down to a single number, we find the area under this curve (AUC).

Note that the area of the entire square is 1*1 = 1. Hence AUC itself is the ratio under the curve and the total area. For the case in hand, we get AUC ROC as 96.4%. Following are a few thumb rules:

- .90-1 = excellent (A)

- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

We see that we fall under the excellent band for the current model. But this might simply be over-fitting. In such cases, it becomes very important to do in-time and out-of-time validations.

Conclusions:

- Metrics like accuracy, precision, recall are good ways to evaluate classification models for balanced datasets, but if the data is imbalanced then other methods like ROC/AUC perform better in evaluating the model performance.
- ROC curve isn’t just a single number but it’s a whole curve that provides nuanced details about the behavior of the classifier. It is also hard to quickly compare many ROC curves to each other.