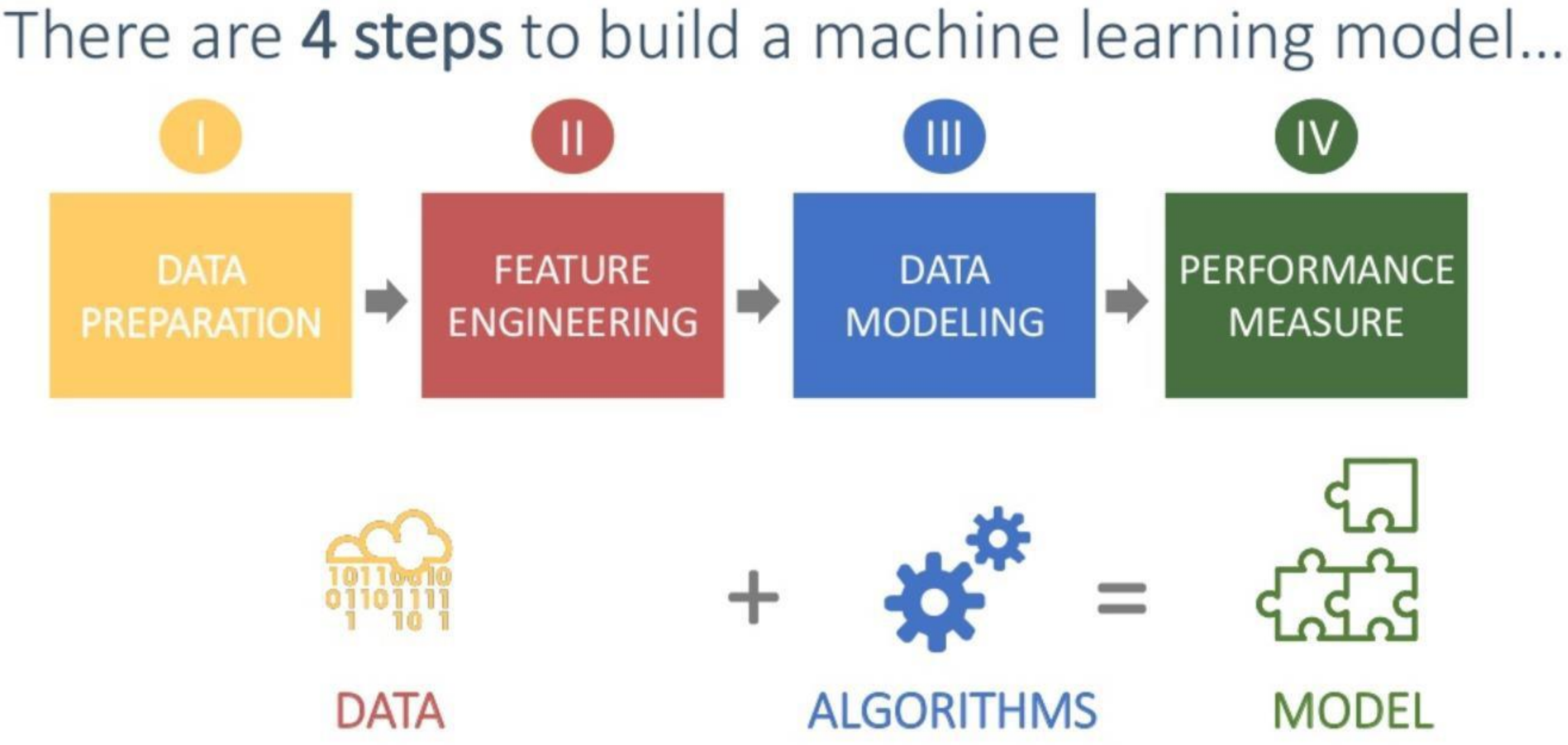
Performance Metrics

COMP 606, Machine Learning

Hongyu Guo
Guo7407@saskpolytech.ca
August 2023

Performance Metrics



Performance Metrics

After doing the usual Feature Engineering, Selection, and of course, implementing a model and getting some output in forms of a probability or a class, the next step is to find out how effective is the model based on some metric using test datasets.

The metrics that you choose to evaluate your machine learning model is very important. Choice of metrics influences how the performance of machine learning algorithms is measured and compared.

Different metrics used:

- **Confusion Matrix**
- Accuracy
- Precision
- Recall or Sensitivity
- Specificity
- F1 Score
- Log Loss
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

Confusion Matrix

Just opposite to what the name suggests, confusion matrix is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes.

Let’s say we are solving a classification problem where we are predicting whether a person is having cancer or not.

Let’s give a label of to our target variable:
1*: When a person is having cancer*
0: *When a person is NOT having cancer.*

Alright! Now that we have identified the problem, the confusion matrix, is a table with two dimensions (“Actual” and “Predicted”), and sets of “classes” in both dimensions. Our Actual classifications are columns and Predicted ones are Rows.

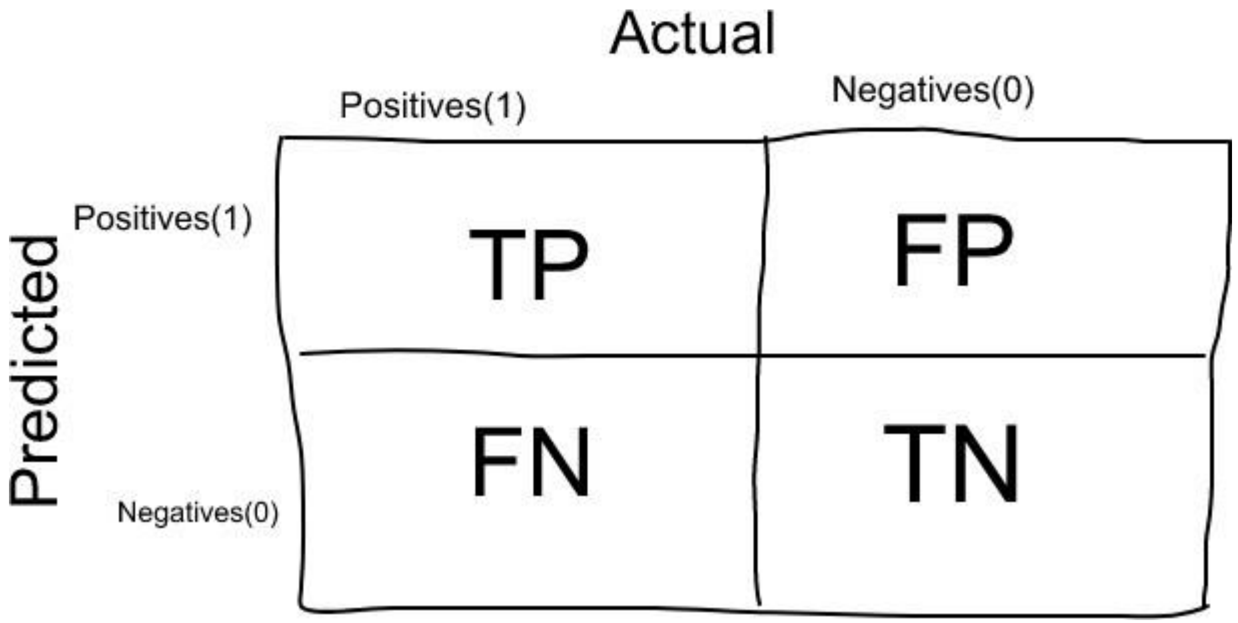
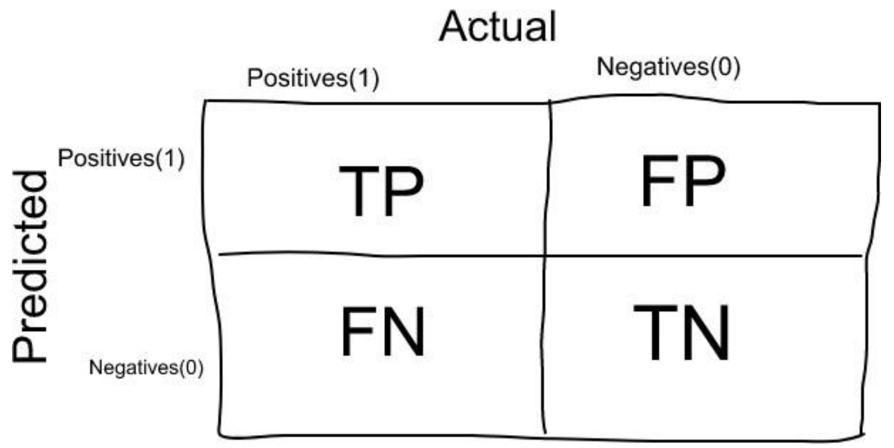


Fig. 1: Confusion Matrix

Confusion Matrix

The Confusion matrix is not a performance measure as such, a lot of the performance metrics are based on Confusion Matrix and the numbers inside it.



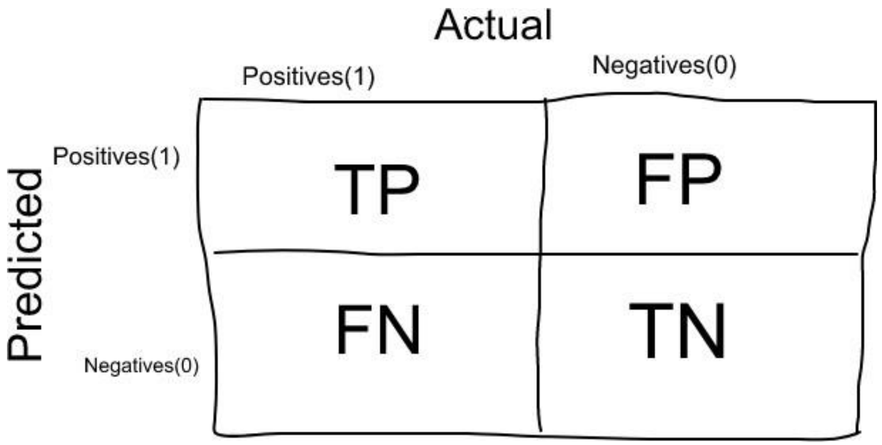
True Positives (TP) - True positives are the cases when the actual class of the data point was 1(True) and the predicted is also 1(True).

Ex: The case where a person is actually having cancer(1) and the model classifying his case as cancer(1) comes under True positive

True Negatives (TN) - True negatives are the cases when the actual class of the data point was 0(False) and the predicted is also 0(False)

Ex: The case where a person NOT having cancer and the model classifying his case as Not cancer comes under True Negatives.

Confusion Matrix



False Positives (FP) - False positives are the cases when the actual class of the data point was 0(False) and the predicted is 1(True). False is because the model has predicted incorrectly and positive because the class predicted was a positive one. (1)

Ex: A person NOT having cancer and the model classifying his case as cancer comes under False Positives.

False Negatives (FN) - False negatives are the cases when the actual class of the data point was 1(True) and the predicted is 0(False). False is because the model has predicted incorrectly and negative because the class predicted was a negative one. (0)

Ex: A person having cancer and the model classifying his case as No-cancer comes under False Negatives.

The ideal scenario that we all want is that the model should give 0 False Positives and 0 False Negatives.
But that's not the case in real life as **any model will NOT** be 100% accurate most of the times.

Confusion Matrix

When to minimize what?

We know that there will be some error associated with every model that we use for predicting the true class of the target variable. This will result in False Positives and False Negatives

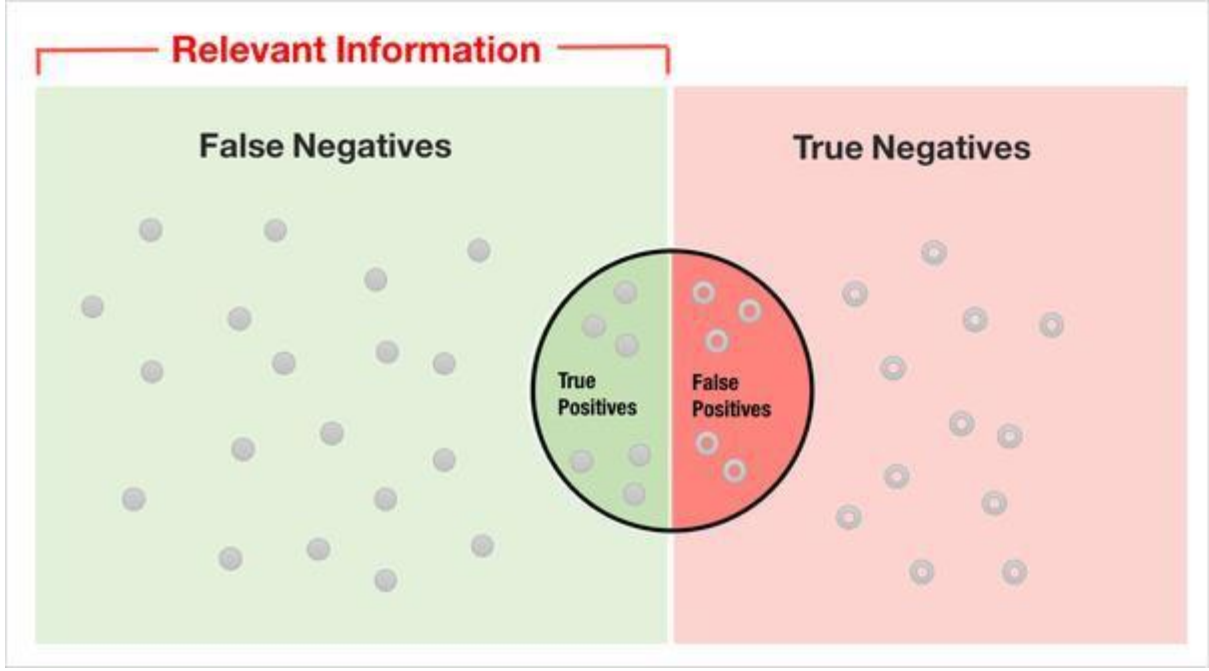
There's no hard rule that says what should be minimised in all the situations. It purely depends on the business needs and the context of the problem you are trying to solve. Based on that, we might want to minimise either False Positives or False negatives.

Confusion Matrix

When to minimize what?

Minimizing False Negatives

Let’s say in our cancer detection problem example, out of 100 people, only 5 people have cancer. In this case, we want to correctly classify all the cancerous patients as even a very BAD model(Predicting everyone as NON-Cancerous) will give us a 95% accuracy. But, to capture all cancer cases, we might end up making a classification when the person actually NOT having cancer is classified as Cancerous. This might be okay as it is less dangerous than NOT identifying/capturing a cancerous patient since we will anyway send the cancer cases for further examination and reports. But missing a cancer patient will be a huge mistake as no further examination will be done on them.



Confusion Matrix

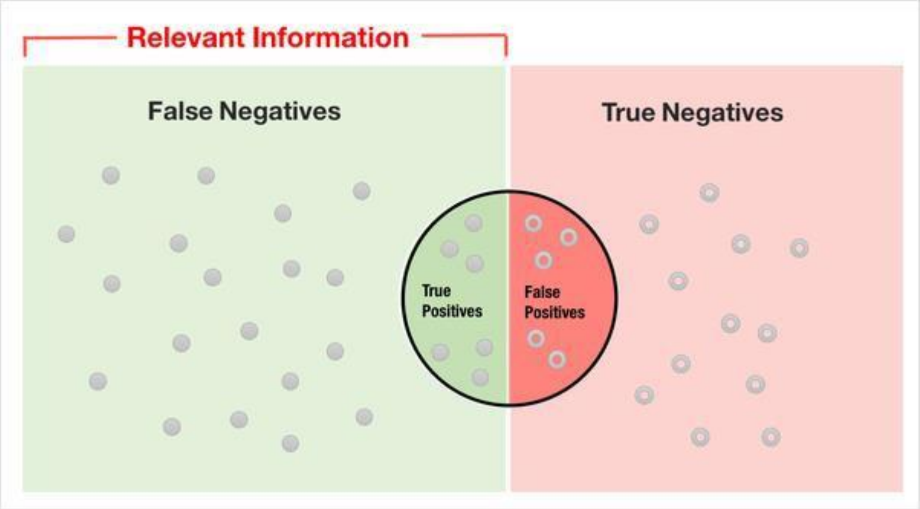
When to minimize what?

Minimizing False Positives:

For better understanding of False Positives, let’s use a different example where the model classifies whether an email is spam or not.

Let’s say that you are expecting an important email like hearing back from a recruiter or awaiting an admit letter from a university. Let’s assign a label to the target variable and say,**1:** “Email is a spam” and **0:**”Email is not a spam”.

Suppose the Model classifies that important email that you are desperately waiting for, as Spam(case of False positive). Now, in this situation, this is pretty bad than classifying a spam email as important or not spam since in that case, we can still go ahead and manually delete it and it’s not a pain if it happens once a while. So in case of Spam email classification, minimising False positives is more important than False Negatives.



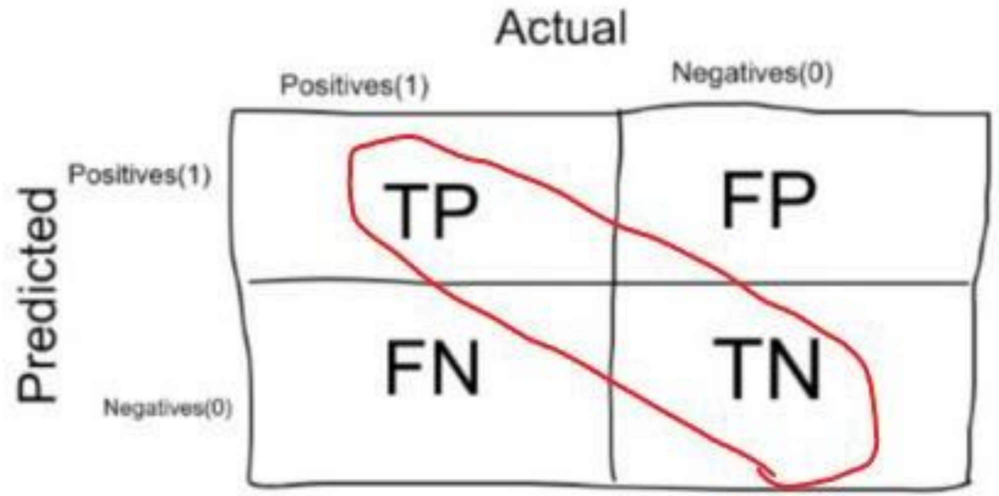
Performance Metrics

Different metrics used:

- Confusion Matrix
- **Accuracy**
- Precision
- Recall or Sensitivity
- Specificity
- F1 Score
- Log Loss
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

Accuracy

Accuracy in classification problems is the number of correct predictions made by the model over all kinds predictions made.



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

In the Numerator, are our correct predictions (True positives and True Negatives) (marked as red in the fig above) and in the denominator, are the kind of all predictions made by the algorithm (right as well as wrong ones).

Accuracy

When to use Accuracy:

Accuracy is a good measure when the target variable classes in the data are nearly balanced.
Eg: 60% classes in our fruits images data are apple and 40% are oranges.

A model which predicts whether a new image is Apple or an Orange, 97% of times correctly is a very good measure in this example.

When not to use Accuracy:

Accuracy should never be used as a measure when the target variable classes in the data are a majority of one class.
Eg: In our cancer detection example with 100 people, only 5 people has cancer.

Let’s say our model is very bad and predicts every case as No Cancer. In doing so, it has classified those 95 non-cancer patients correctly and 5 cancerous patients as Non-cancerous. Now even though the model is terrible at predicting cancer, The accuracy of such a bad model is also 95%.

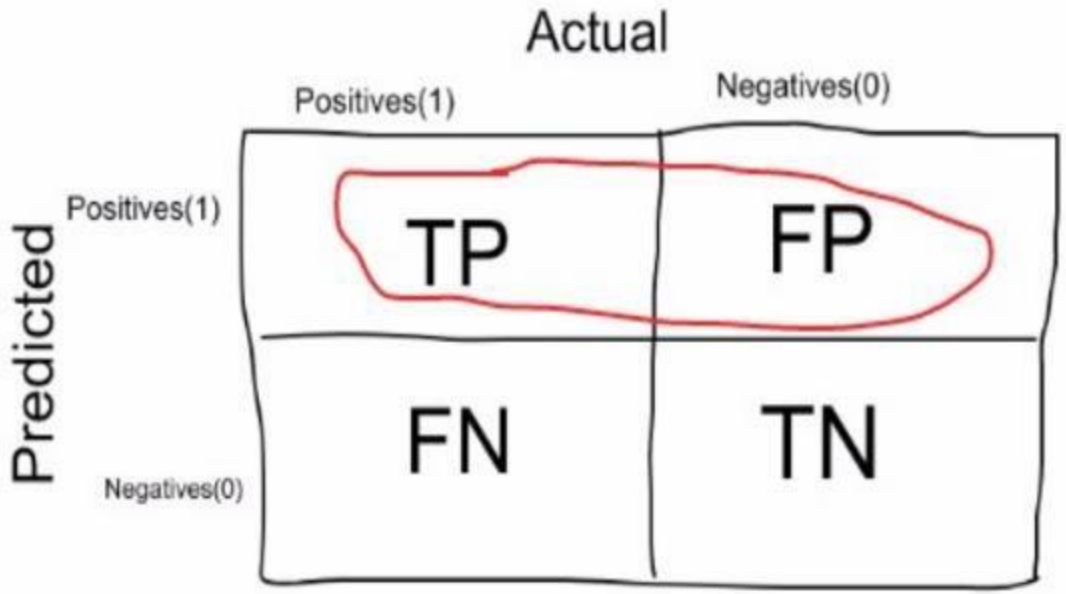
Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- **Precision**
- Recall or Sensitivity
- Specificity
- F1 Score
- Log Loss
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

Precision

Precision is a measure that tells us what proportion of patients that we diagnosed as having cancer, actually had cancer. The predicted positives (People predicted as cancerous are TP and FP) and the people actually having a cancer are TP.



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

In our cancer example with 100 people, only 5 people have cancer. Let's say our model is very bad and predicts every case as **Cancer**. Since we are predicting everyone as having cancer, our denominator (True positives and False Positives) is 100 and the numerator, person having cancer and the model predicting his case as cancer is 5. So in this example, we can say that **Precision** of such model is 5%.

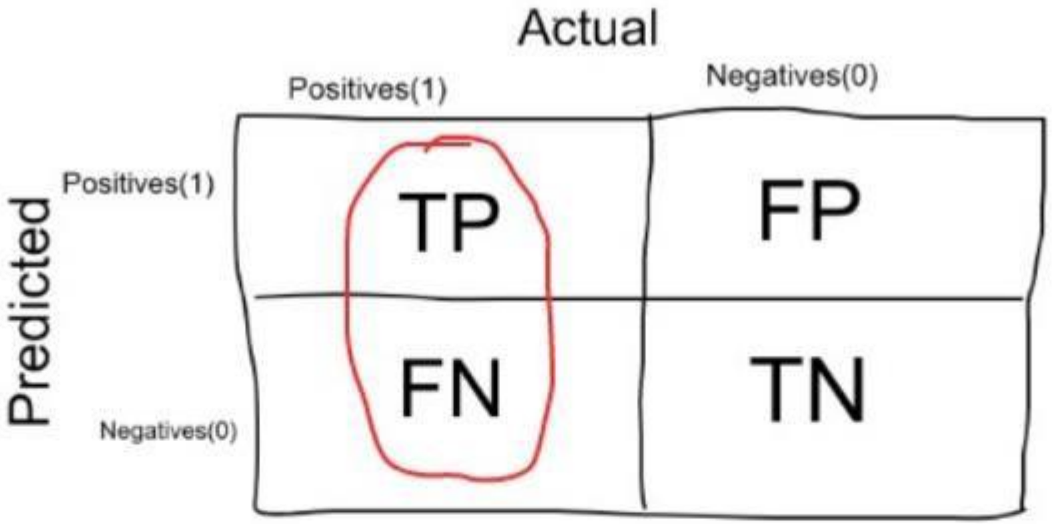
Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- Precision
- **Recall or Sensitivity**
- Specificity
- F1 Score
- Log Loss
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

Recall or Sensitivity

Recall is a measure that tells us what proportion of patients that actually had cancer was diagnosed by the algorithm as having cancer. The actual positives (People having cancer are TP and FN) and the people diagnosed by the model having a cancer are TP. (Note: FN is included because the Person actually had a cancer even though the model predicted otherwise).



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

Ex: In our cancer example with 100 people, 5 people actually have cancer. Let’s say that the model predicts every case as cancer. So our denominator(True positives and False Negatives) is 5 and the numerator, person having cancer and the model predicting his case as cancer is also 5(Since we predicted 5 cancer cases correctly). So in this example, we can say that the **Recall** of such model is 100%. And Precision of such a model(As we saw above) is 5%

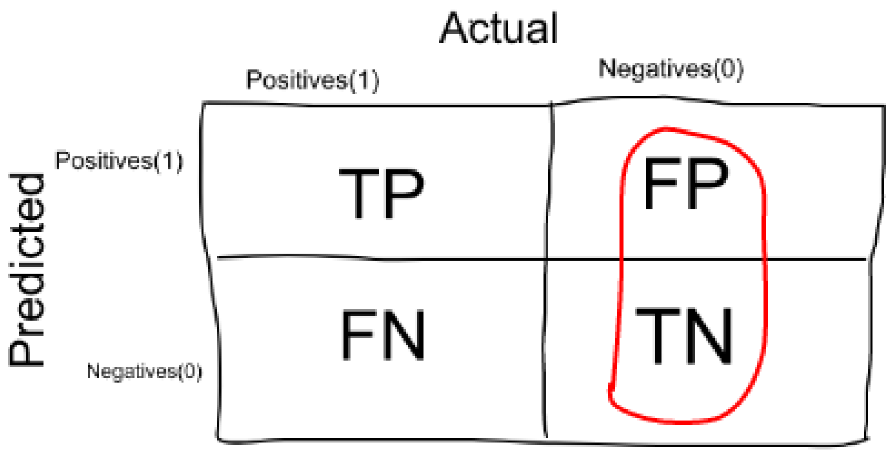
Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- Precision
- Recall or Sensitivity
- **Specificity**
- F1 Score
- Log Loss
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

Specificity

Specificity is a measure that tells us what proportion of patients that did NOT have cancer, were predicted by the model as non-cancerous. The actual negatives (People actually NOT having cancer are FP and TN) and the people diagnosed by us not having cancer are TN.



$$\text{Specificity} = \frac{TN}{TN + FP}$$

Specificity is the exact opposite of Recall.
Ex: In our cancer example with 100 people, 5 people actually have cancer. Let's say that the model predicts every case as cancer.
So our denominator(False positives and True Negatives) is 95 and the numerator, person not having cancer and the model predicting his case as no cancer is 0 (Since we predicted every case as cancer). So in this example, we can that that **Specificity** of such model is 0%.

Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- Precision
- Recall or Sensitivity
- Specificity
- **F1 Score**
- Log Loss
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

F1 Score

The F1 Score is a metric used to evaluate the performance of a classification model, especially when dealing with imbalanced datasets (where one class has far more samples than the other).
It combines two important metrics:

- Precision – out of all the positive predictions, how many were actually correct?
- Recall – out of all the actual positives, how many were correctly identified?

The F1 score is the harmonic mean of precision and recall:

F1 Score = Harmonic Mean(Precision, Recall)

F1 Score = 2 * Precision * Recall / (Precision + Recall)

The higher the F1 score, the better the balance between precision and recall.

The F1 score is a single number that balances precision and recall, making it a useful metric when evaluating classification models on **imbalanced datasets.**

Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- Precision
- Recall or Sensitivity
- Specificity
- F1 Score
- **Log Loss**
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

Log Loss

Logarithmic Loss or Log Loss, works by penalising the false classifications.

It works well for multi-class classification. When working with Log Loss, the classifier must assign probability to each class for all the samples.

Suppose, there are N samples belonging to M classes, then the Log Loss is calculated as below :

$$LogarithmicLoss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

where,

y_ij, indicates whether sample i belongs to class j or not
p_ij, indicates the probability of sample i belonging to class j

Log Loss has no upper bound and it exists on the range [0, ∞). Log Loss nearer to 0 indicates higher accuracy, whereas if the Log Loss is away from 0 then it indicates lower accuracy.

In general, minimising Log Loss gives greater accuracy for the classifier.

Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- Precision
- Recall or Sensitivity
- Specificity
- F1 Score
- Log Loss
- **Area under the curve (AUC)**
- MAE – Mean Absolute Error
- MSE – Mean Squared Error

AUC – Area Under the ROC Curve

ROC Curve

An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

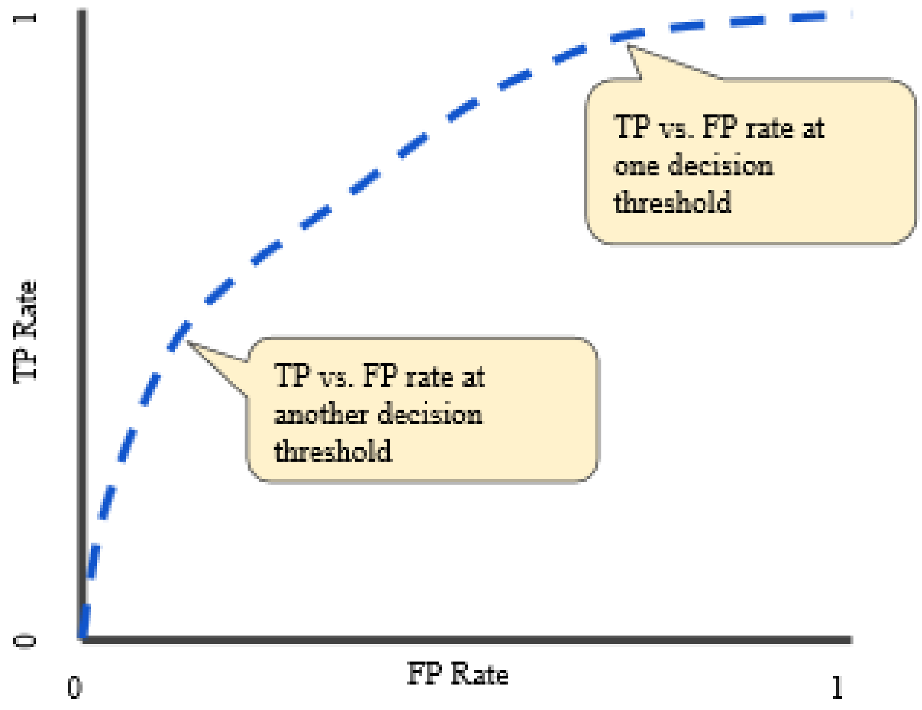
True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

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

False Positive Rate (**FPR**) is defined as follows:

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

An ROC curve plots TPR vs. FPR at different classification thresholds. (as shown in the right fig.)



AUC – Area Under the ROC Curve

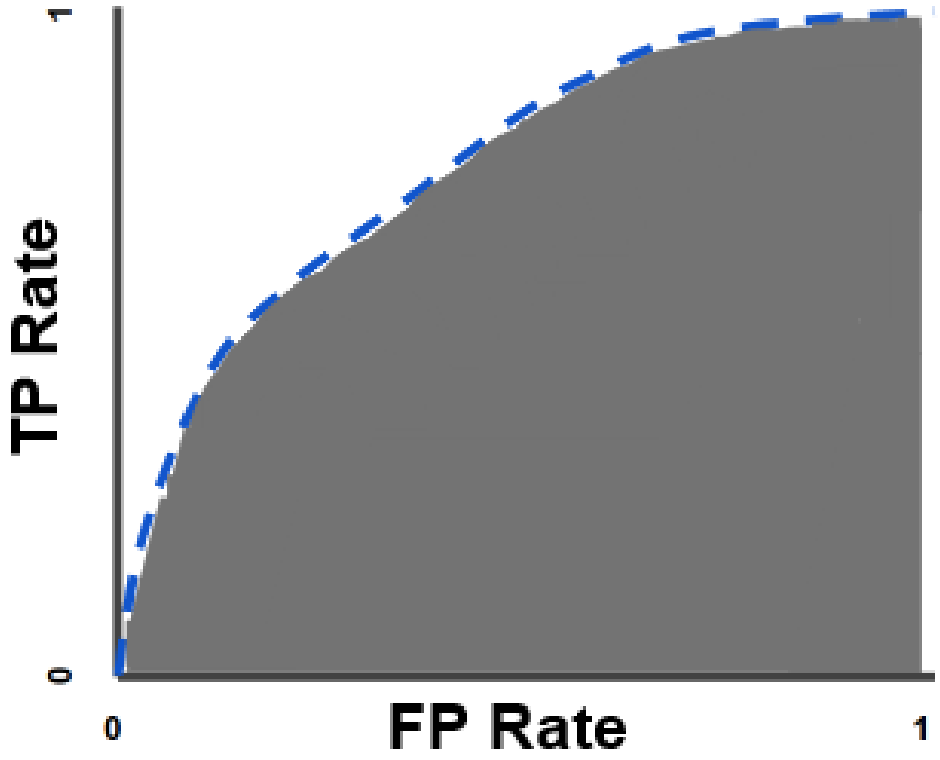
AUC Value

Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

To compute the points in an ROC curve, we could evaluate a logistic regression model many times with different classification thresholds, but this would be inefficient. Fortunately, there's an efficient, sorting-based algorithm that can provide this information for us, called **AUC**.

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1)

AUC provides an aggregate measure of performance across all possible classification thresholds.



Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- Precision
- Recall or Sensitivity
- Specificity
- F1 Score
- Log Loss
- Area under the curve (AUC)
- **MAE – Mean Absolute Error**
- MSE – Mean Squared Error

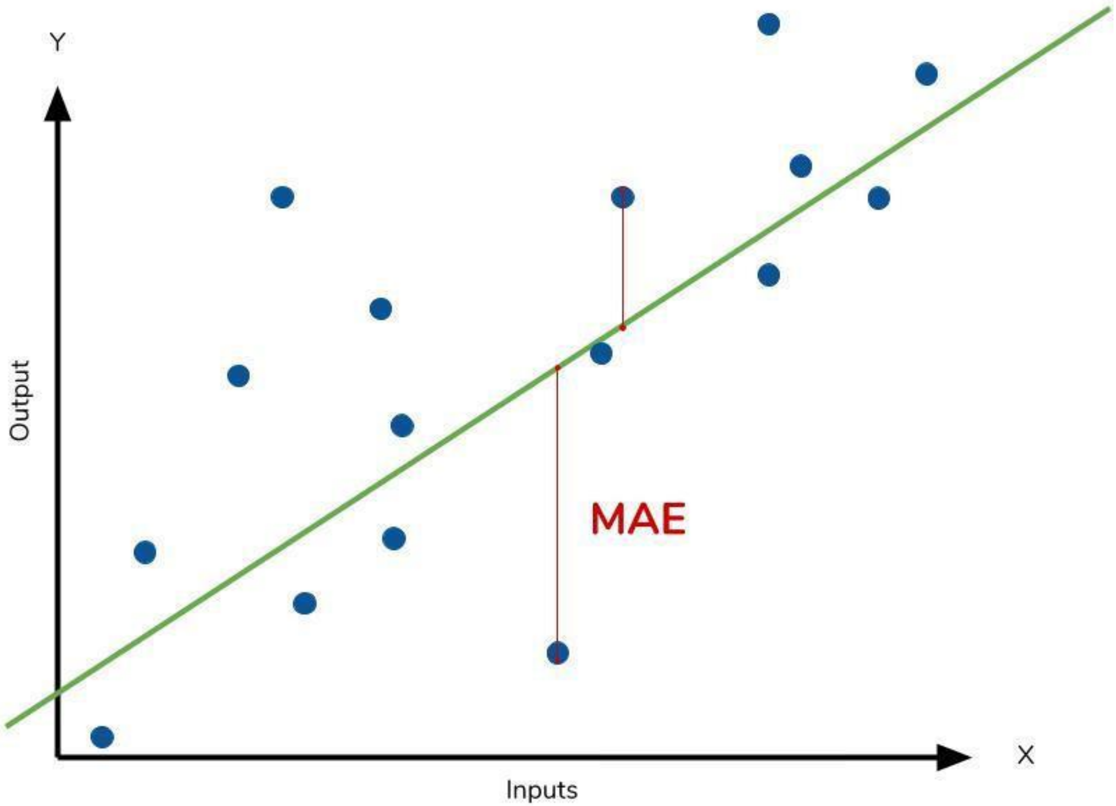
Mean Absolute Error

Mean Absolute Error is the average of the difference between the original values and the predicted values.

It gives us the measure of how far the predictions were from the actual output. However, they don't gives us any idea of the direction of the error i.e. whether we are under predicting the data or over predicting the data.

Mathematically, it is represented as :

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$



Performance Metrics

Different metrics used:

- Confusion Matrix
- Accuracy
- Precision
- Recall or Sensitivity
- Specificity
- F1 Score
- Log Loss
- Area under the curve (AUC)
- MAE – Mean Absolute Error
- **MSE – Mean Squared Error**

Mean Squared Error

Mean Squared Error(MSE) is quite similar to Mean Absolute Error, the only difference being that MSE takes the average of the **square** of the difference between the original values and the predicted values.

As, we take square of the error, the effect of larger errors become more pronounced then smaller error, hence the model can now focus more on the larger errors.

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

Instead of MSE, we generally use RMSE, which is equal to the square root of MSE.

Taking the square root of the average squared errors has some interesting implications for RMSE. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors.

This means the RMSE should be more useful when large errors are particularly undesirable.

Mean Squared Error

The three tables below show examples where MAE is steady and RMSE increases as the variance associated with the frequency distribution of error magnitudes also increases.

CASE 1: Evenly distributed errors

ID	Error	|Error|	Error^2
1	2	2	4
2	2	2	4
3	2	2	4
4	2	2	4
5	2	2	4
6	2	2	4
7	2	2	4
8	2	2	4
9	2	2	4
10	2	2	4

MAE	RMSE
2.000	2.000

CASE 2: Small variance in errors

ID	Error	|Error|	Error^2
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	3	3	9
7	3	3	9
8	3	3	9
9	3	3	9
10	3	3	9

MAE	RMSE
2.000	2.236

CASE 3: Large error outlier

ID	Error	|Error|	Error^2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	20	20	400

MAE	RMSE
2.000	6.325

Mean Squared Error

Conclusion:

1. RMSE has the benefit of penalizing large errors more so can be more appropriate in some cases, for example, if being off by 10 is more than twice as bad as being off by 5. But if being off by 10 is just twice as bad as being off by 5, then MAE is more appropriate.
2. From an interpretation standpoint, MAE is clearly the winner. RMSE does not describe average error alone and has other implications that are more difficult to tease out and understand.
3. One distinct advantage of RMSE over MAE is that RMSE avoids the use of taking the absolute value, which is undesirable in many mathematical calculations