Performance metrics

- Accuracy
- Confusion matrix
- Precision and recall
- F1-score
- ROC curve - AUC
- Log - loss
- R - squared
- Mean squared prediction error
- Mean squared absolute error
- Median absolute deviation
- Elbow method
- Blue score
- CV error

we will explain each metric in detail in upcoming posts

@learn.machinelearning

Performance Metrics

Accuracy

- We use Accuracy in classification problems and it is the most common evaluation metric.
- Accuracy is defined as the ratio of the number of correct predictions made by the model over all kinds of predictions made.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

- One of the biggest disadvantages of accuracy is it doesn't work well when we have an imbalanced dataset.
- It works well only if there are an equal number of samples belonging to each class.

Performance Metrics

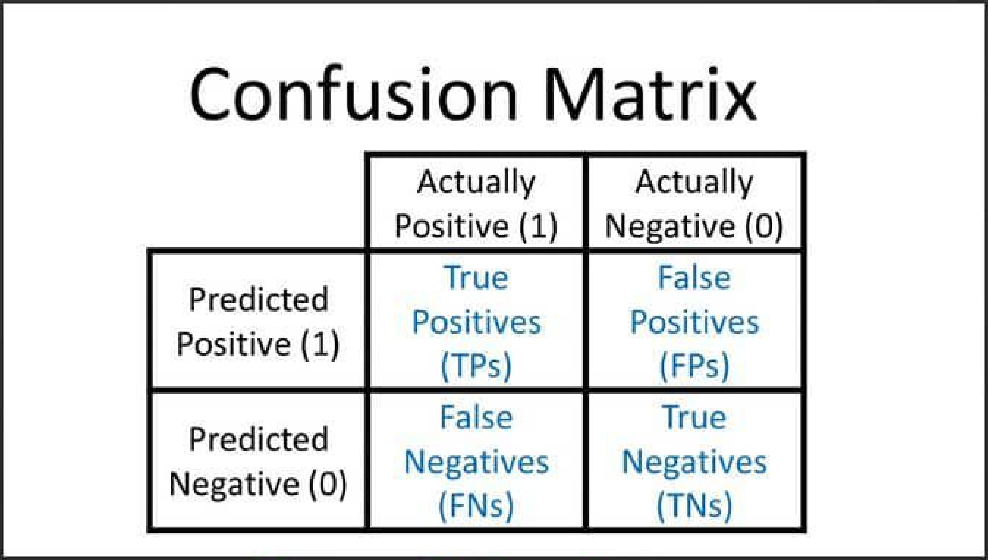
Accuracy

- For example, consider that there are 98% samples of class A and 2% samples of class B in our training set. Then our model can easily get 98% training accuracy by simply predicting every training sample belonging to class A.
- When the same model is tested on a test set with 60% samples of class A and 40% samples of class B, then the test accuracy would drop down to 60%. Classification Accuracy is great but gives us a false sense of achieving high accuracy.
- The real problem arises when the cost of misclassification of the minor class samples are very high. If we deal with a rare but fatal disease, the cost of failing to diagnose the disease of a sick person is much higher than the cost of sending a healthy person to more tests or fraud detection.

Performance Metrics

Confusion matrix

- Confusion Matrix as the name suggests gives us a matrix as output as N X N matrix, where N is the number of classes being predicted.
- This Metric used for finding the correctness and accuracy of the model and even works better for imbalanced dataset.
- 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.

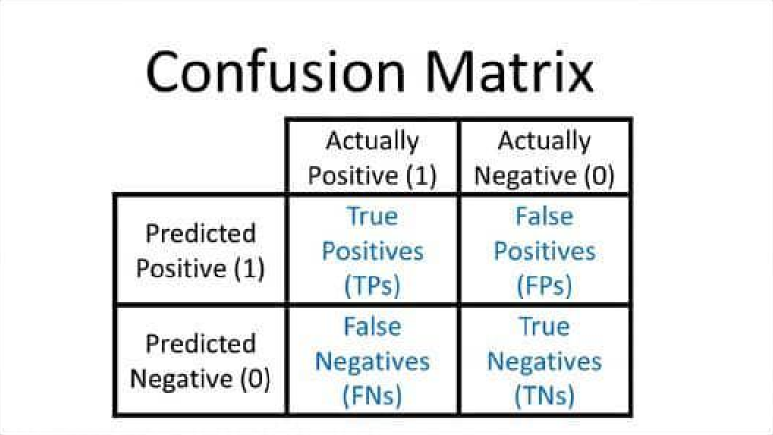


Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Performance Metrics

Confusion matrix



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

- The Confusion matrix in itself is not a performance measure as such, but almost all of the performance metrics are based on Confusion Matrix and the numbers inside it.
- **True Positives:** The cases in which we predicted YES and the actual output was also YES.

  True Negatives: The cases in which we predicted NO and the actual output was NO.

  False Positives: The cases in which we predicted YES and the actual output was NO.

  False Negatives: The cases in which we predicted NO and the actual output was YES.

Performance Metrics

Confusion matrix

When to minimise 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(i.e Model classifying things incorrectly as compared to the actual class).
- 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.

Performance Metrics

Confusion matrix

Minimising False Negatives:

- 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(i.e Model classifying things incorrectly as compared to the actual class).
- 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.

Performance Metrics

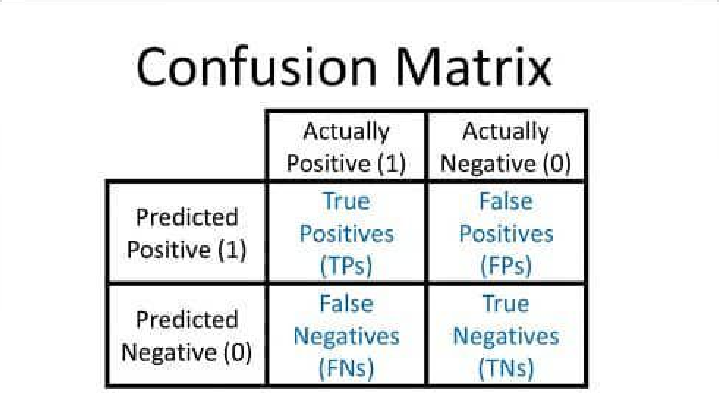
Confusion matrix

Minimising False Positives:

- For a better understanding of False Positives, let’s use a different example where the model classifies whether an email is a 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 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

Precision



Confusion Matrix

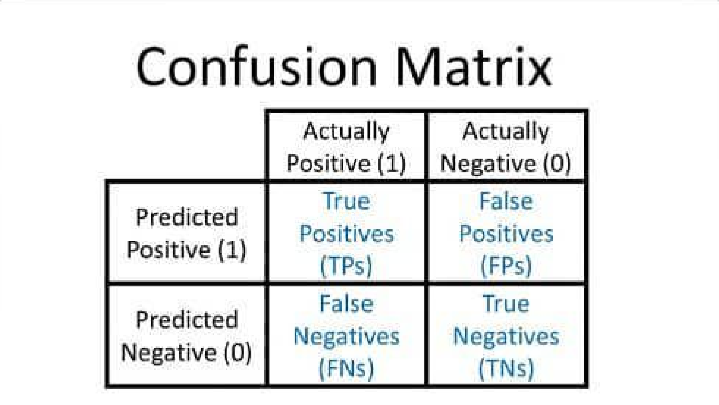
	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

- **Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.**
- **It tells us what proportion of predictions which are predicted as positives that are actual positives to the total predicted positives.**

$$\text{Precision} = \frac{tp}{tp + fp}$$

Performance Metrics

Recall or Sensitivity



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

- **Precision is defined as the number of true positives divided by the number of true positives plus the number of False Negatives.**
- **It tells us what proportion of predictions which are predicted as positives that are actual positives to the total positives in the original dataset.**

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

Performance Metrics

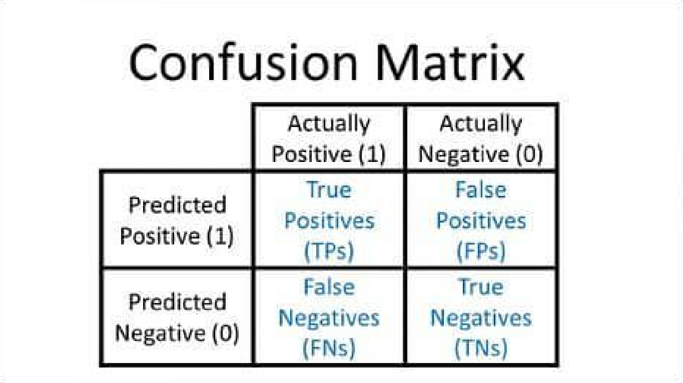
When to use Precision and When to use Recall?

- It is clear that recall gives us information about a classifier’s performance with respect to false negatives (how many did we miss), while precision gives us information about its performance with respect to false positives(how many did we caught).
- Precision is about being precise. So even if we managed to capture only one cancer case, and we captured it correctly, then we are 100% precise.
- Recall is not so much about capturing cases correctly but more about capturing all cases that have “cancer” with the answer as “cancer”. So if we simply always say every case as “cancer”, we have 100% recall.
- So basically if we want to focus more on minimising False Negatives, we would want our Recall to be as close to 100% as possible without precision being too bad and if we want to focus on minimising False positives, then our focus should be to make Precision as close to 100% as possible.

Performance Metrics

F1 Score

- F1 score combines precision and recall relative to a specific positive class.
- F1 score conveys the balance between the precision and the recall AND there is an uneven class distribution.
- F1 score reaches its best value at 1 and worst at 0.



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

$$\text{Precision} = \frac{tp}{tp + fp}$$

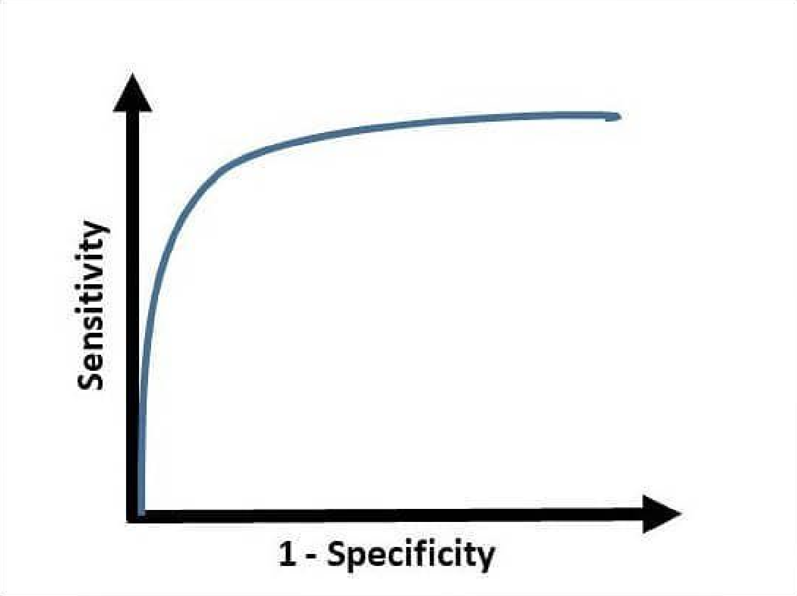
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} = 2 \text{ x } \frac{Precision * Recall}{Precision + Recall}$$

Performance Metrics

AUC - ROC Curve

- When we need to check or visualize the performance of the multi-class classification problem, we use the AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) curve.
- It is one of the most important evaluation metrics for checking any classification model's performance. It is also written as AUROC (Area Under the Receiver Operating Characteristics).



Performance Metrics

AUC - ROC Curve

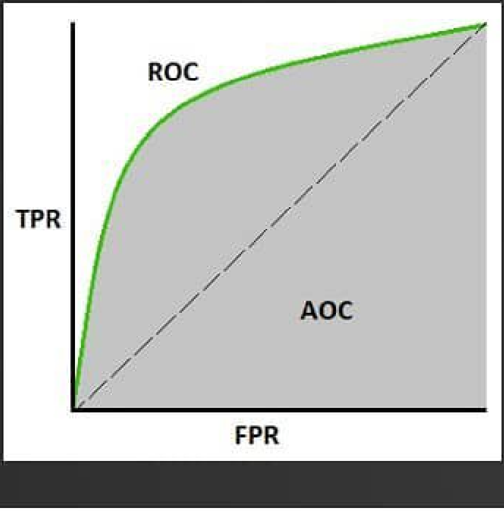
What is AUC - ROC Curve?

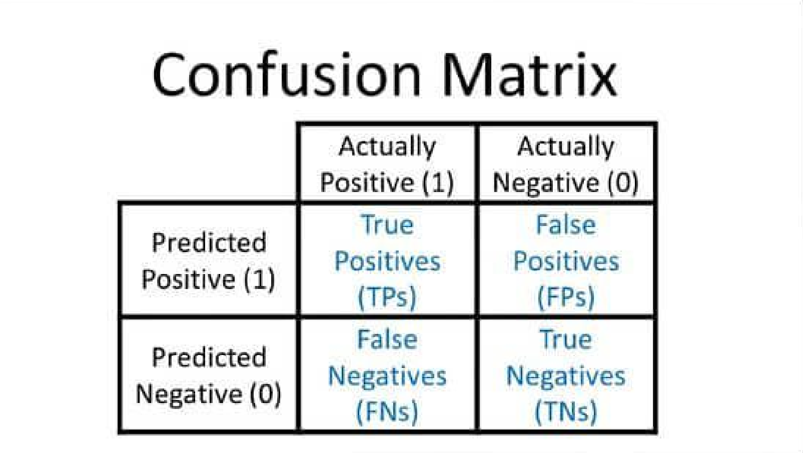
- AUC - ROC curve is a performance measurement for the classification problem at various thresholds settings.
- ROC is a probability curve and AUC represents the degree or measure of separability.
- ROC (Receiver Operating Characteristic) Curve tells us about how good the model can distinguish between two things (e.g If a patient has a disease or no).
- Better models can accurately distinguish between the two. Whereas, a poor model will have difficulties in distinguishing between the two.
- Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with the disease and no disease.

Performance Metrics

AUC - ROC Curve

- The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.





Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

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

$$\text{FPR} = 1 - \text{Specificity}$$

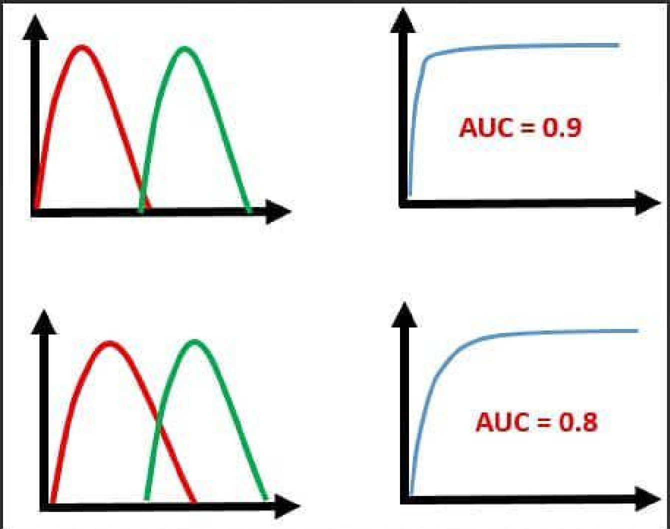
$$= \frac{FP}{TN + FP}$$

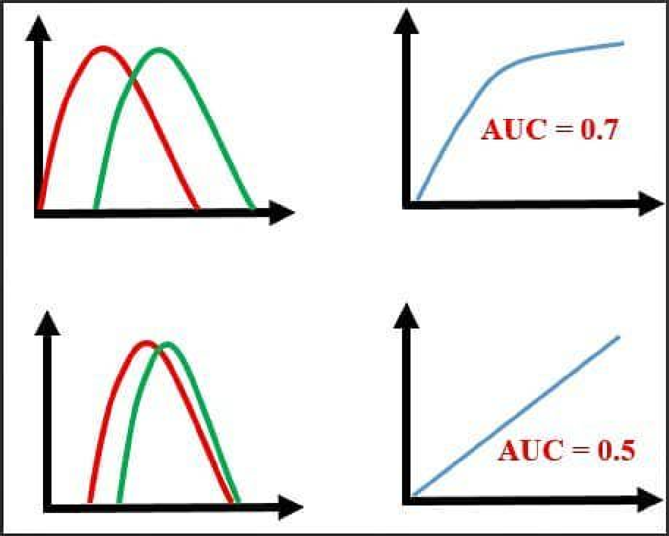
@learn.machinelearning

Performance Metrics

AUC - ROC Curve

- Let’s take a few examples





- As we see, the first model does quite a good job of distinguishing the positive and negative values. Therefore, there the AUC score is 0.9 as the area under the ROC curve is large.
- Whereas, if we see the last model, predictions are completely overlapping each other and we get the AUC score of 0.5. This means that the model is performing poorly and it is predictions are almost random.

Performance Metrics

AUC - ROC Curve

How to use AUC ROC curve for a multi-class model?

- In the multi-class model, we can plot N number of AUC ROC Curves for N number classes using One vs ALL methodology. So for Example, If you have three classes named X, Y and Z, you will have one ROC for X classified against Y and Z, another ROC for Y classified against X and Z, and the third one of Z classified against Y and X.

Performance Metrics

Cohen's Kappa

- **Kappa or Cohen’s Kappa is like classification accuracy, except that it is normalized at the baseline of random chance on your dataset. It is a more useful measure to use on problems that have an imbalance in the classes.**
- **Kappa is similar to Accuracy score, but it takes into account the accuracy that would have happened anyway through random predictions.**

$$\text{Kappa} = \frac{(\text{Observed Accuracy} - \text{Expected Accuracy})}{(1 - \text{Expected Accuracy})}$$