Lesson_8: Classification metrics

Ali Aburas, PhD

Slide credit:Andrew Ng

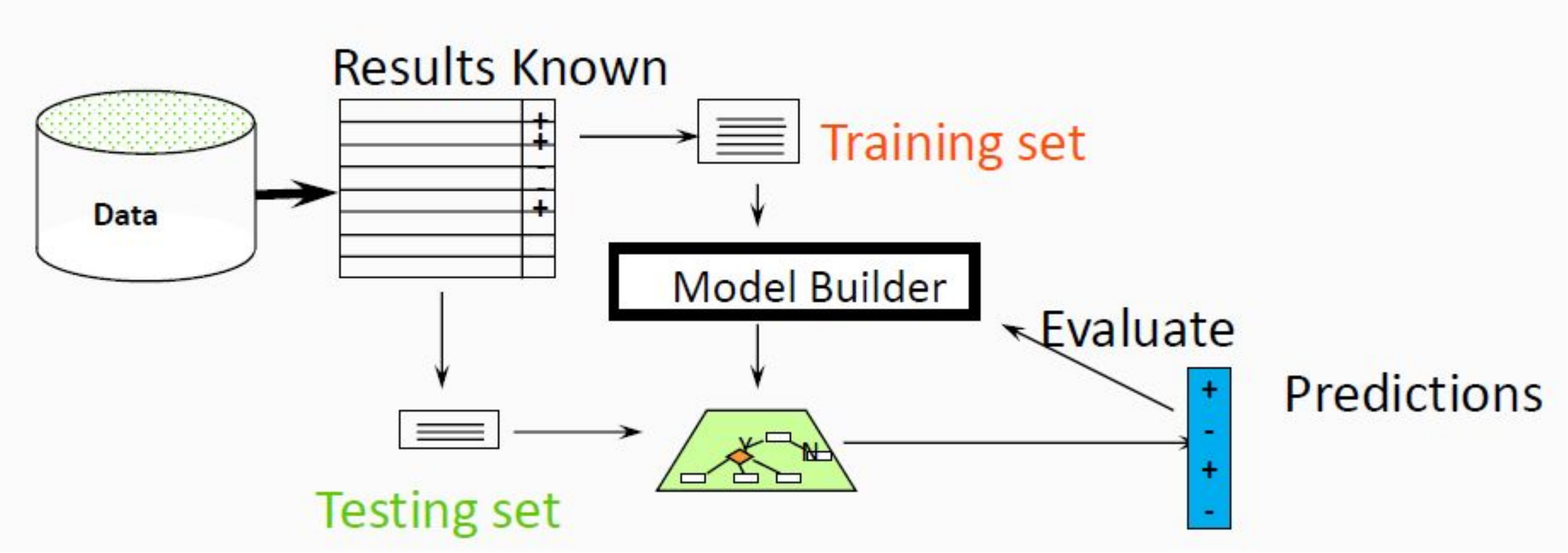
Outline

1. Metrics
2. Precision and recall and F1-Score
3. Model Evaluation and Improvement

Applied Machine Learning Process

- Each machine learning project is different because the specific data at the core of the project is different. .
- Even though your project is unique, the steps on the path to a good or even the bestresult are generally the same from project to project. This is sometimes referred to as the ***applied machine learning process***: the process using the four high-level steps:
  - **Step 1**: ***Define Problem***. This step is concerned with learning enough about the project to select the framing or framings of the prediction task.
  - **Step 2**: ***Prepare Data***. This step is concerned with transforming the raw data that was collected into a form that can be used in modeling.
  - **Step 3**: ***Evaluate Models***. This step is concerned with evaluating machine learning models on your dataset.
  - **Step 4**: ***Finalize Model***. This step is concerned with selecting and using a final model. Once a suite of models has been evaluated, you must choose a model that represents the solution to the project.

Applied Machine Learning Process



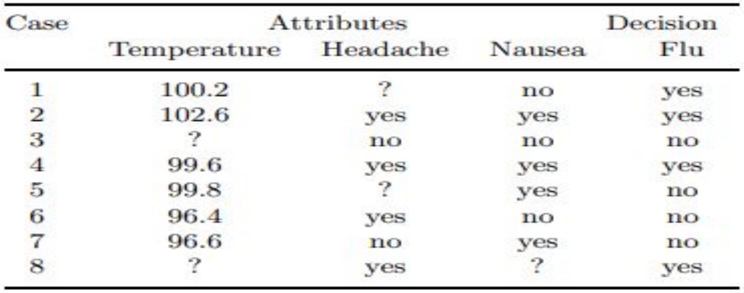
What Is Data Preparation?

There are common or standard tasks that you may use or explore during the data preparation step in a machine learning project. These tasks include:

- Data Cleaning: Identifying and correcting mistakes or errors in the data.
- Feature Selection: Identifying those input variables that are most relevant to the task.
- Data Transforms: Changing the scale or distribution of variables.
- Feature Engineering: Deriving new variables from available data.
- Dimensionality Reduction: Creating compact projections of the data.

What Is Data Preparation? (Example-1)

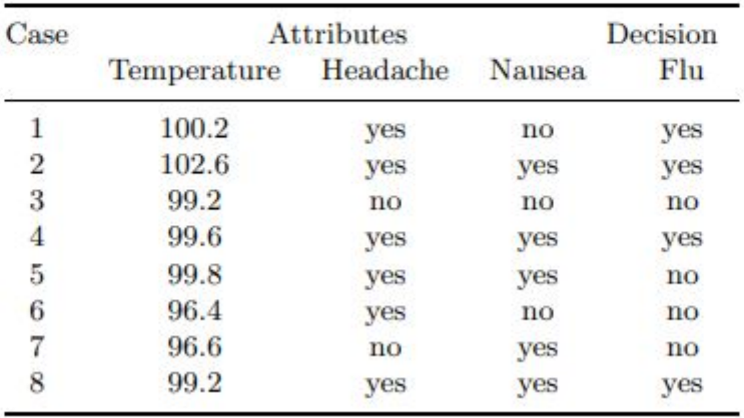
Handling Missing Attribute Values



Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	?	no	yes
2	102.6	yes	yes	yes
3	?	no	no	no
4	99.6	yes	yes	yes
5	99.8	?	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	?	yes	?	yes

REPLACING MISSING ATTRIBUTE VALUES BY THE ATTRIBUTE MEAN

a. every missing attribute value for a numerical attribute is replaced by the arithmetic mean of known attribute values.



Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	yes	no	yes
2	102.6	yes	yes	yes
3	99.2	no	no	no
4	99.6	yes	yes	yes
5	99.8	yes	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	99.2	yes	yes	yes

What Is Data Preparation? (Example-2)

Dealing with Imbalanced datasets

- when one of two classes is much more frequent than the other one.
- There are quite a few ways to handle imbalanced data in machine classification problems.
    - Random under-sampling:
        - The random under-sampling technique works by randomly eliminating the samples from the majority class until the classes are balanced in the remaining dataset.
    - Random over-sampling
        - In this technique, we try to increase the instances of the minority class by random replication of the already present samples.
    - Synthetic minority oversampling technique: SMOTE
        - In SMOTE, a subset of minority class is taken and new synthetic data points are generated based on it. These synthetic data points are then added to the original training dataset as additional examples of the minority class.

Performance metrics

- Performance metrics in machine learning are essential for assessing the effectiveness and reliability of models.
- The metrics can be broadly categorized into two main types: **regression** and **classification metrics**.
- **Regression metrics** are used to evaluate the performance of algorithms that predict continuous numerical values.
  - **Mean Absolute Error (MAE)** measures the average magnitude of errors in the predictions made by the model (does not penalize larger errors heavily).
  - **Mean Squared Error (MSE)** measures the average squared difference between the actual and predicted values, penalizing larger errors more heavily than smaller ones.
  - **Root Mean Squared Error (RMSE**) measures the average squared difference between the predicted and actual values. Use RMSE to penalize larger errors and obtain a metric with the same unit as the target variable.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{y}(\mathrm{i}) - \hat{\mathrm{y}}(\mathrm{i})^2}{\mathrm{n}}}$$

Classification metrics

- Classification metrics assess the performance of machine learning models for classification tasks.
- Accuracy is a fundamental evaluation metric for assessing the overall performance of a classification model.
- Accuracy measures the proportion of correct predictions made by the model out of all predictions.

- We train on our training data **Train = {xi, yi}i=1,m**
- We test on **Test data**.
- When we deal with binary classification we often measure performance simply using **Accuracy**:

$$\text{accuracy} = \frac{\#\ \text{correct predictions}}{\#\ \text{test instances}}$$

  - If a classifier make 10 predictions and 9 of them are correct, the accuracy is 90%.

Any possible problems with it?

Evaluate Classification Models: metrics for imbalanced dataset

Heart Disease classification example

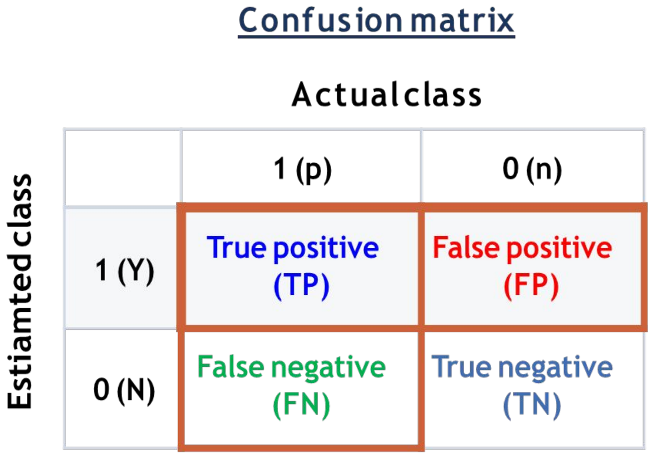
- Assume that we have dataset contains info of 100 patients
- The y = 1 class has very few samples with respect to the y = 0 class
- Assume that only 0.5% of patients **actually have** disease
- If we use DummyClassifier that **always classifies** the observations to the **$\mathbf{0}$ class,** we get 99.5% of accuracy!!

$$\text{accuracy} = \frac{\#\ \text{correct predictions}}{\#\ \text{test instances}}$$

For **imbalanced dataset,** the accuracy metric can be deceptive

Evaluate Classification Models: Confusion Matrix

- A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the total number of target classes.
- The matrix compares the actual target values with those predicted by the machine learning model.
- For a binary classification problem, we would have a 2 x 2 matrix, as shown below, with 4 values:
- Important Terms in a Confusion Matrix
  - **True Positive (TP)**: The actual value was positive, and the model predicted a positive value.
  - **True Negative (TN)**: The actual value was negative, and the model predicted a negative value.
  - **False Positive (FP)**:The actual value was negative, but the model predicted a positive value.
  - **False Negative (FN)**: The actual value was positive, but the model predicted a negative value.



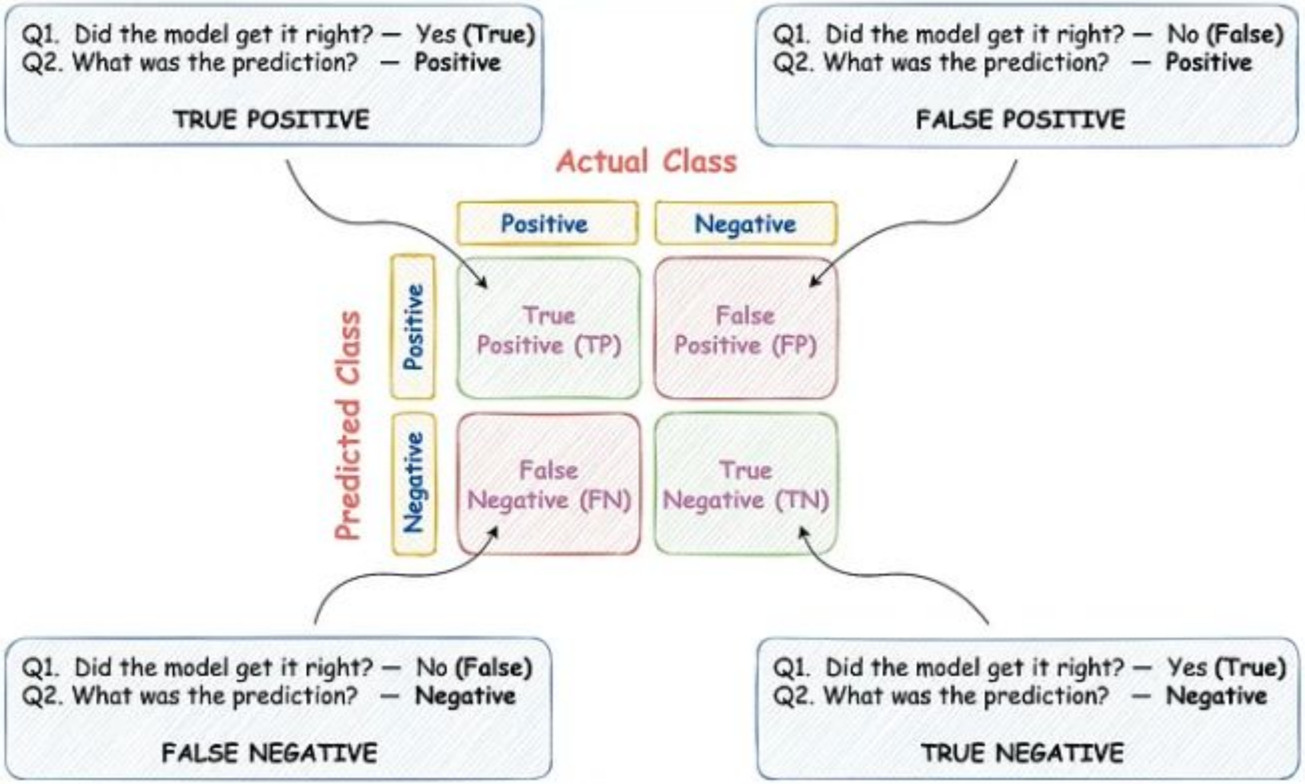
Confusion matrix

Actualclass

Estiamted class

	1 (p)	0 (n)
1 (Y)	True positive (TP)	False positive (FP)
0 (N)	False negative (FN)	True negative (TN)

Evaluate Classification Models: Confusion Matrix



Q1. Did the model get it right? — Yes (True)
Q2. What was the prediction? — Positive

TRUE POSITIVE

Q1. Did the model get it right? — No (False)
Q2. What was the prediction? — Positive

FALSE POSITIVE

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

Q1. Did the model get it right? — No (False)
Q2. What was the prediction? — Negative

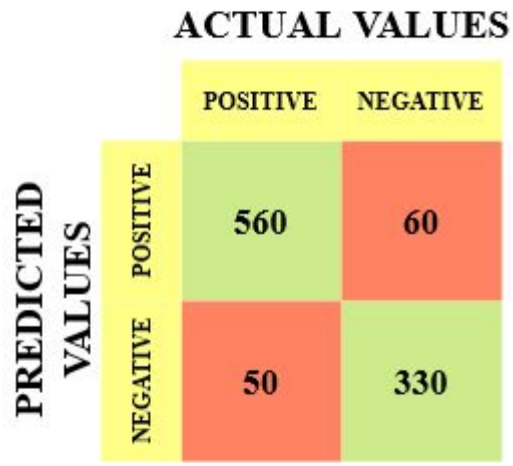
FALSE NEGATIVE

Q1. Did the model get it right? — Yes (True)
Q2. What was the prediction? — Negative

TRUE NEGATIVE

Evaluate Classification Models: Confusion Matrix

- Suppose we had a classification dataset with 1000 data points. We fit a classifier (say logistic regression or decision tree) on it and get the below confusion matrix:



		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	560	60
	NEGATIVE	50	330

- True Positive (TP) = 560, meaning the model correctly classified 560 positive class data points.
- True Negative (TN) = 330, meaning the model correctly classified 330 negative class data points.
- False Positive (FP) = 60, meaning the model incorrectly classified 60 negative class data points as belonging to the positive class.
- False Negative (FN) = 50, meaning the model incorrectly classified 50 positive class data points as belonging to the negative class.

Evaluate Classification Models: Precision vs. Recall

- **Precision** tells us how many of the correctly predicted cases actually turned out to be positive.

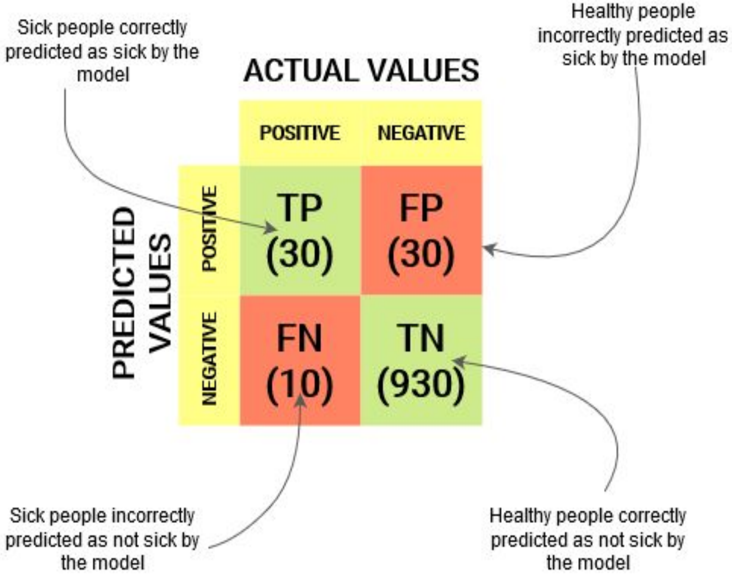
Here’s how to calculate Precision: $\text{precision} = \dfrac{TP}{TP + FP}$

In **precision**, our focus is to increase TP and TN and to decrease FP.

- **Recall (Sensitivity)** tells us how many of the actual positive cases we were able to predict correctly with our model.

here’s how we can calculate Recall: $\text{recall} = \dfrac{TP}{TP + FN}$

In **recall**, our focus is to increase TP and to decrease FN



Sick people correctly predicted as sick by the model

Healthy people incorrectly predicted as sick by the model

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP (30)	FP (30)
	NEGATIVE	FN (10)	TN (930)

Sick people incorrectly predicted as not sick by the model

Healthy people correctly predicted as not sick by the model

$$Precision = \frac{30}{30+30} = 0.5$$

$$Recall = \frac{30}{30+10} = 0.75$$

Evaluate Models: Trading off precision and recall

- **Precision** is a useful metric in cases where False Positive is a higher concern than False Negatives.
  - **Precision** is important in music or video recommendation systems, e-commerce websites, etc. Wrong results could lead to customer leave and be harmful to the business.
- **Recall** is a useful metric in cases where False Negative surpass False Positive.
  - **Recall** is important in medical cases where it doesn't matter whether we raise a false alarm, but the actual positive cases should not go undetected!

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

In **precision**, our focus is to increase TP and TN and to decrease FP.

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

In **recall**, our focus is to increase TP and to decrease FN

Evaluate Models: Trading off precision and recall

There are some cases you mostly care about the precision and in other context you mostly care about the recall.

1. **Example of High Precision** As we know we have multiple platform for video streaming like well known YouTube, you have restricted mode to restrict the violent and adult videos for the kids.
   a. So model focus on high precision {TP/(TP+FP)} by reducing the false positive.
      i. If model has classified the video is good for kids it must be safe to watch by kids. So, this can be done by reducing the false positive.

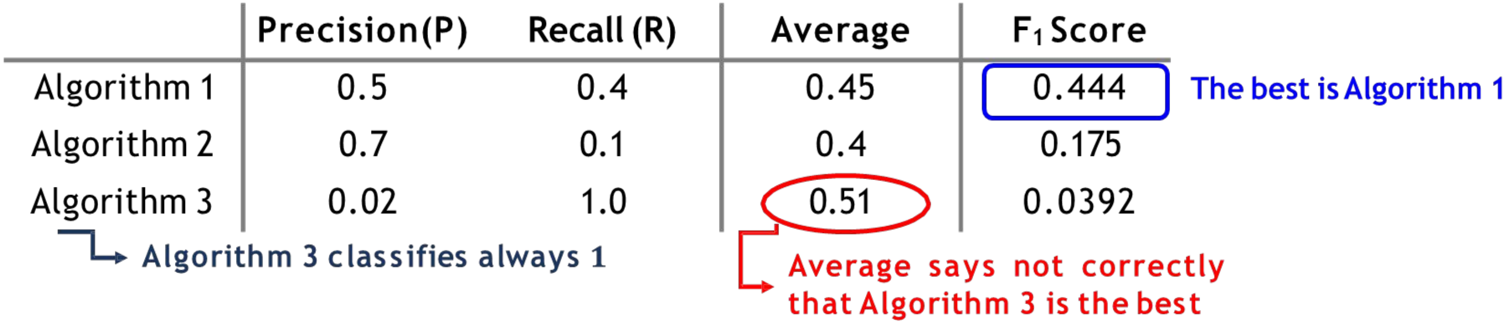
2. **Example of High Recall** Let's take an example, you are creating a model to detect a patient is having disease or not.
   a. In this case the aim of the model is to have high recall {TP/(TP+FN)} means a smaller number of false negative.
      i. If model predict a patient is not having a disease so, he must not have the disease.

Evaluate Models: What Is F1-Score?

- In practice, when we try to increase the precision of our model, the recall goes down, and vice-versa.
- The F1-score captures both the trends in a single value:

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \Longleftrightarrow F_1 \text{score} = 2\frac{P \cdot R}{P + R}$$

- **F1-score** is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall.



	Precision(P)	Recall (R)	Average	F_1 Score
Algorithm 1	0.5	0.4	0.45	0.444
Algorithm 2	0.7	0.1	0.4	0.175
Algorithm 3	0.02	1.0	0.51	0.0392

The best is Algorithm 1

Algorithm 3 classifies always 1

Average says not correctly that Algorithm 3 is the best

- we use it in combination with other evaluation metrics, giving us a complete picture of the result.

Confusion Matrix Practical Example

Let’s go through a practical example to demonstrate this process.

Assume that we have a classifier that identifies an email as spam or not spam, let’s create a hypothetical dataset where spam is Positive and not spam is Negative. We have the following data:

- Amongst the 200 emails, 80 emails are actually spam in which the model correctly identifies 60 of them as spam .
- Amongst the 200 emails, 120 emails are not spam in which the model correctly identifies 100 of them as not spam .
- Amongst the 200 emails, the model incorrectly identifies 20 non-spam emails as spam .
- Amongst the 200 emails, the model misses 20 spam emails and identifies them as non-spam .

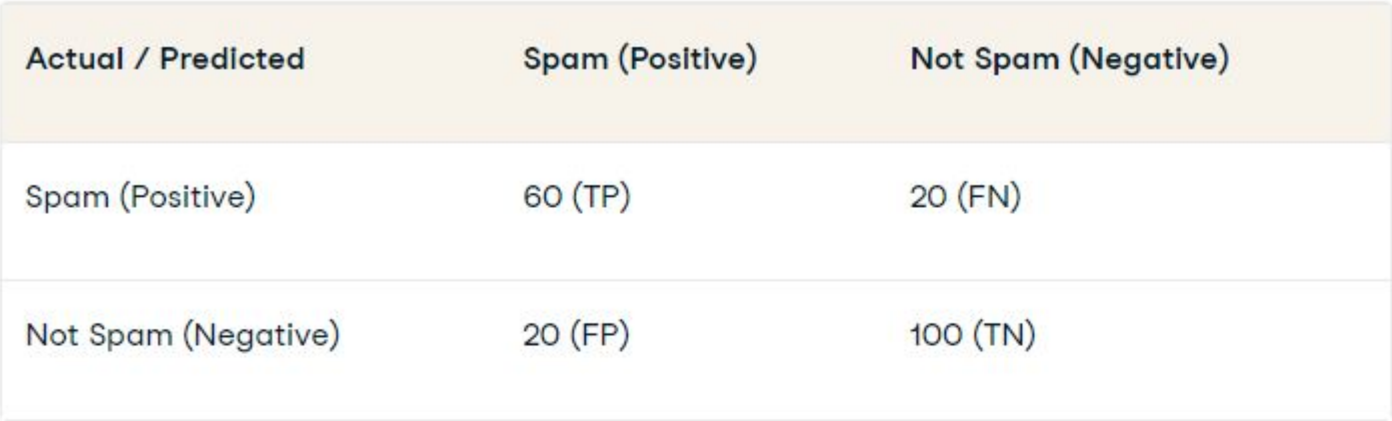
Confusion Matrix Practical Example

Let’s go through a practical example to demonstrate this process.

Assume that we have a classifier that identifies an email as spam or not spam, let’s create a hypothetical dataset where spam is Positive and not spam is Negative. We have the following data:

- Amongst the 200 emails, 80 emails are actually spam in which the model correctly identifies 60 of them as spam .
- Amongst the 200 emails, 120 emails are not spam in which the model correctly identifies 100 of them as not spam .
- Amongst the 200 emails, the model incorrectly identifies 20 non-spam emails as spam .
- Amongst the 200 emails, the model misses 20 spam emails and identifies them as non-spam .

Turn the outcome into a Confusion Matrix and calculate Acurracy, Recall, Precision and F1-Score.



Actual / Predicted	Spam (Positive)	Not Spam (Negative)
Spam (Positive)	60 (TP)	20 (FN)
Not Spam (Negative)	20 (FP)	100 (TN)

$$\boxed{Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{60+100}{60+100+20+20} = \frac{160}{200} = 0.8}$$

$$\boxed{Recall = \frac{TP}{TP+FN} = \frac{60}{60+20} = \frac{60}{80} = 0.75}$$

$$\boxed{Precision = \frac{TP}{TP+FP} = \frac{60}{60+20} = \frac{60}{80} = 0.75}$$

$$\boxed{Specificity = \frac{TN}{TN+FP} = \frac{100}{100+20} = \frac{100}{120} \approx 0.833}$$

$$\boxed{F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.75 \times 0.75}{0.75+0.75} = 0.75}$$

Area Under the Receiver Operating Characteristic Curve (AU-ROC)

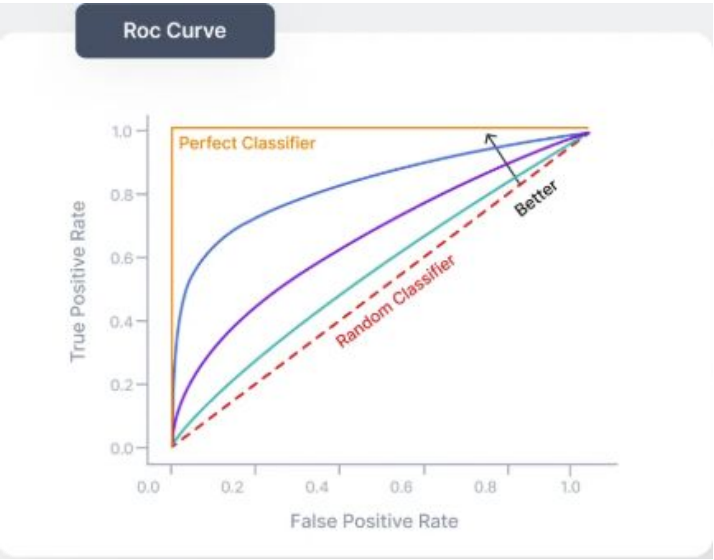
- The ROC curve plots the **true positive rate** (recall) against the **false positive rate** (1 - specificity) at various classification thresholds.

$$\text{TPR, Recall, Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{FPR, } (1 - \text{Specificity}) = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

- The AU-ROC represents the area under the ROC curve, and a higher value indicates better model performance.

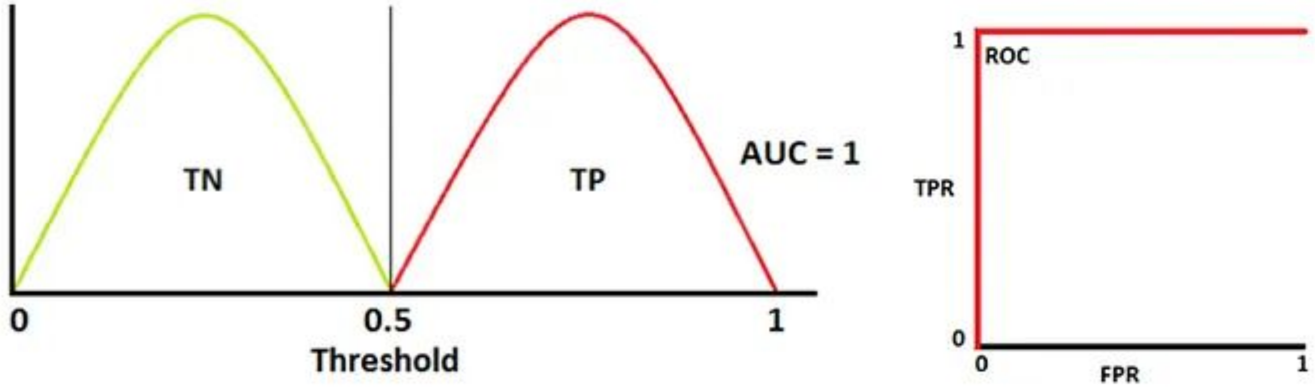
AU-ROC represents the model's ability to discriminate between positive and negative classes.

A **higher AU-ROC** value indicates better classification performance, with a perfect classifier having an AU-ROC of 1 and a random classifier having an AU-ROC of 0.5.



Understanding the AUC

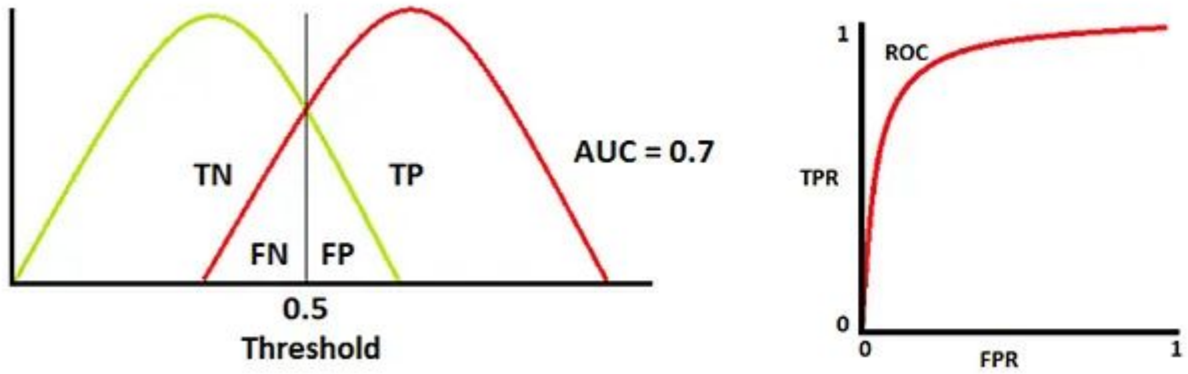
1. *No overlap, i.e. perfect classification result!*



- Red distribution curve is of the positive class (patients with disease) and the green distribution curve is of the negative class (patients with no disease).
- When the two curves don’t overlap at all the model has an ideal measure of separability. It is perfectly able to distinguish between positive class and negative class and the AUC is equal to one and the ROC aligns with the y axis

Understanding the AUC

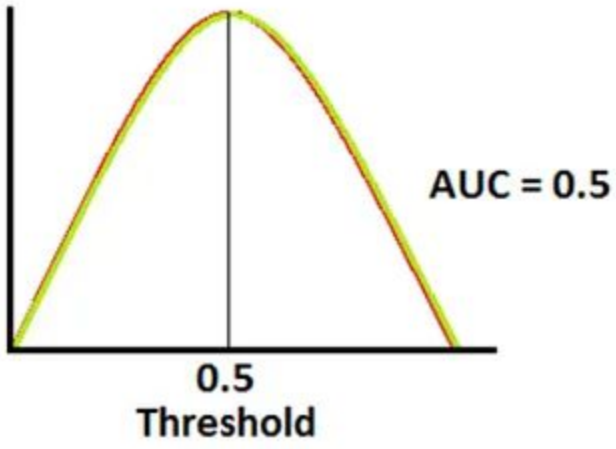
2. *Some overlap — normal scenario*

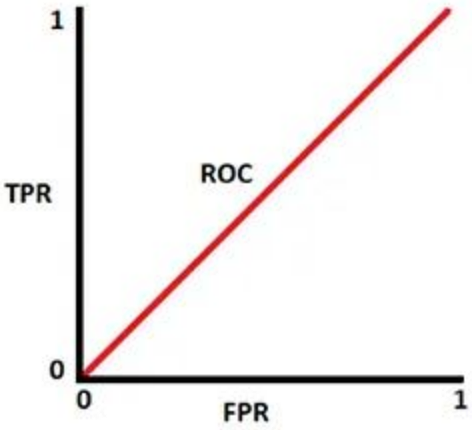


- This is the normal scenario. Here we have some overlap. There is some misclassification going on depending on the threshold value.
- We get the usual ROC and an AUC value which is greater than .5 and less than 1.

Understanding the AUC

3. *Complete overlap — impossible to classify!*





- Here there is perfect overlap and there is no way the model will be able to distinguish a positive from a negative at any threshold. In this case the ROC is a straight diagonal and the AUC is .5.