Classification metrics

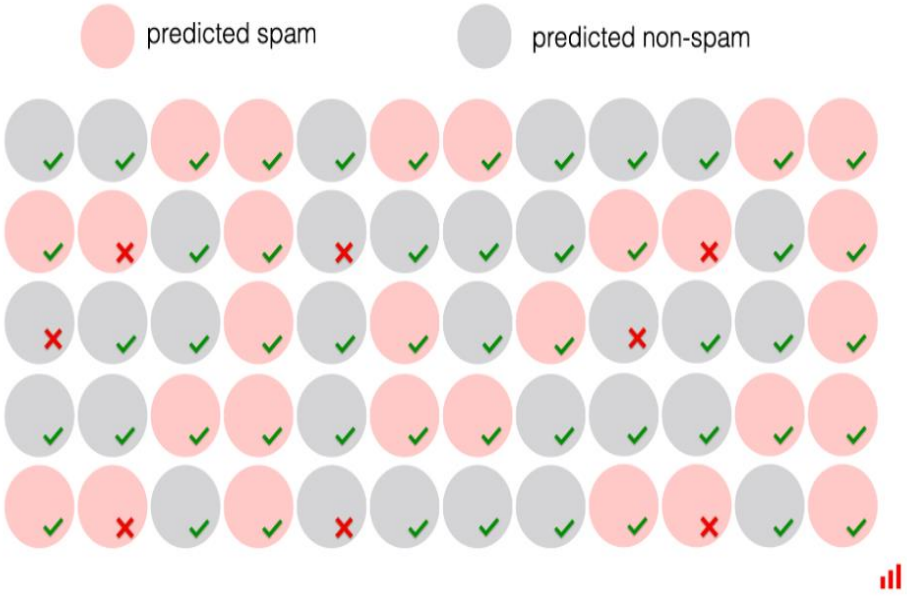
Marziyeh Mousavi

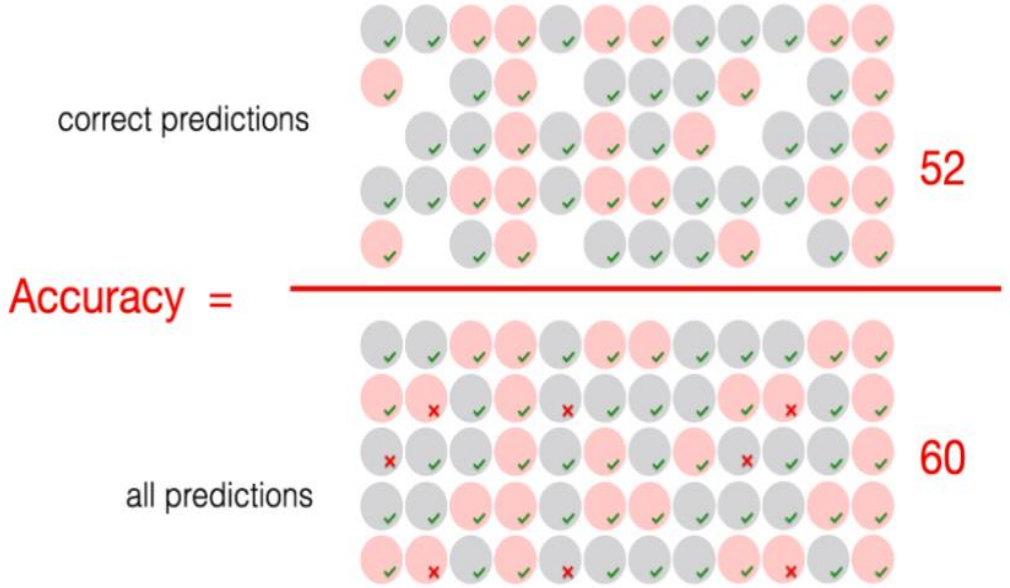
Accuracy

- **Accuracy** is a metric that measures how often a machine learning model correctly predicts the outcome. You can calculate accuracy by dividing the number of correct predictions by the total number of predictions.
- You can measure the accuracy on a scale of 0 to 1 or as a percentage. The higher the accuracy, the better. You can achieve a perfect accuracy of 1.0 when every prediction the model makes is correct.
- This metric is simple to calculate and understand. Almost everyone has an intuitive perception of accuracy: a reflection of the model's ability to correctly classify data points.

Example

We have some spam and non spam emails and we have a model to predict the label.





Pros and Cons Of accuracy

› **Pros:**

- Accuracy is a helpful metric when you deal with **balanced classes** and care about the overall model "correctness" and not the ability to predict a specific class.
- Accuracy is **easy to explain** and communicate.

› **Cons:**

- If you have **imbalanced classes**, accuracy is less useful since it gives equal weight to the model's ability to predict all categories.
- Communicating accuracy in such cases **can be misleading** and disguise low performance on the target class

Confusion matrix

- In binary classification, there are two possible target classes, which are typically labeled as "positive" and "negative" or "1" and "0". In our spam example a, the target (positive class) is "spam," and the negative class is "not spam."
- When evaluating the accuracy, we looked at correct and wrong predictions disregarding the class label. However, in binary classification, we can be "correct" and "wrong" in two different ways.

Confusion Matrix

› **Correct predictions** include so-called true positives and true negatives. This is how it unpacks for our spam use case example:

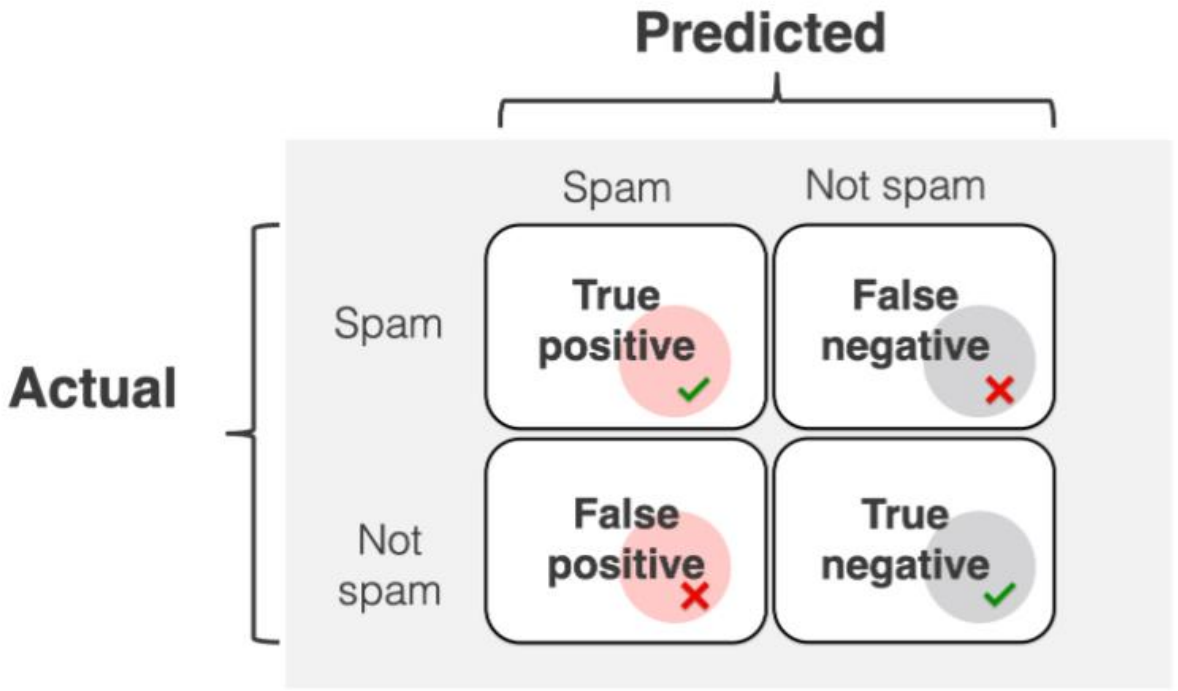
- **True positive (TP):** An email that is actually spam and is correctly classified by the model as spam.
- **True negative (TN):** An email that is actually not spam and is correctly classified by the model as not spam.

› **Model errors** include so-called false positives and false negatives. In our example:

- **False Positive (FP):** An email that is actually not spam but is incorrectly classified by the model as spam (a "false alarm").
- **False Negative (FN):** An email that is actually spam but is incorrectly classified by the model as not spam (a "missed spam").

Confusion Matrix

› This is a visualization of matrix

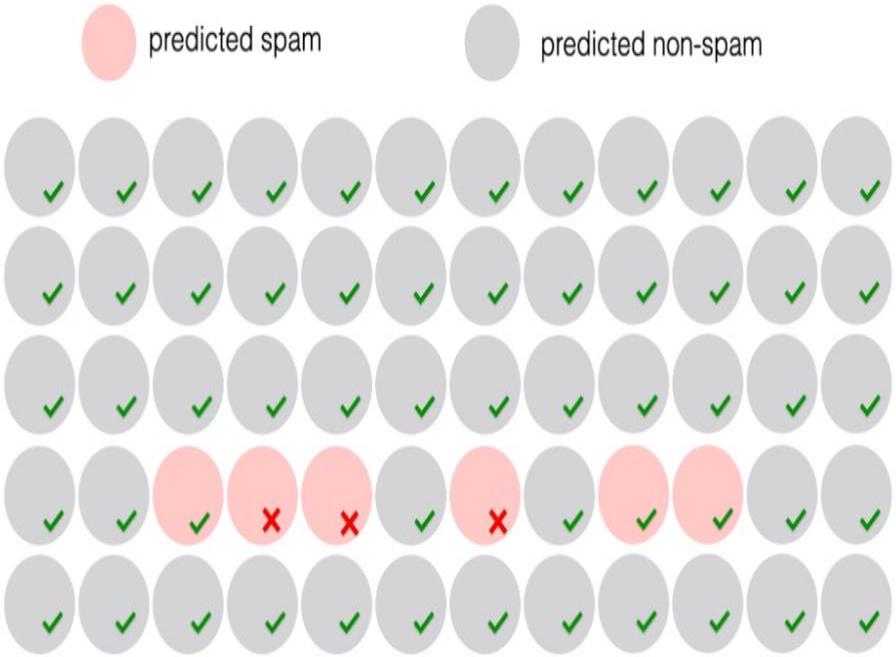


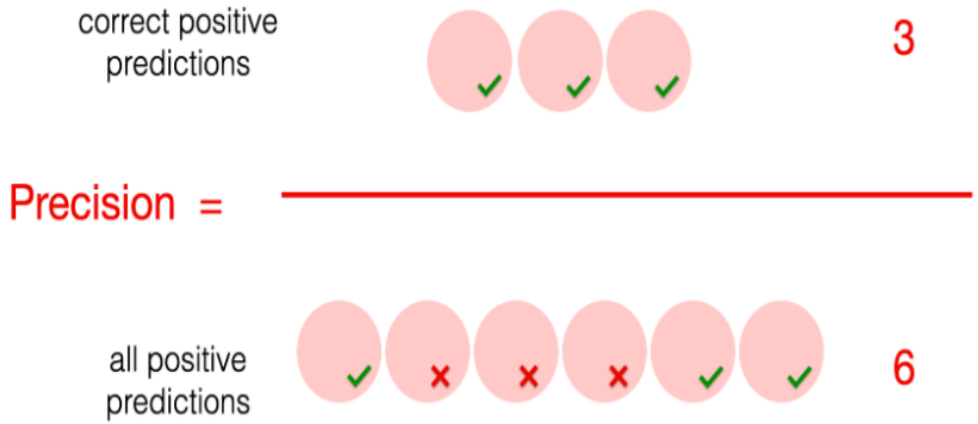
Precision

- Precision is a metric that measures how often a machine learning model correctly predicts the positive class. You can calculate precision by dividing the number of correct positive predictions (true positives) by the total number of instances the model predicted as positive (both true and false positives).

- In other words, precision answers the question: how often the positive predictions are correct?

Example


predicted spam
predicted non-spam

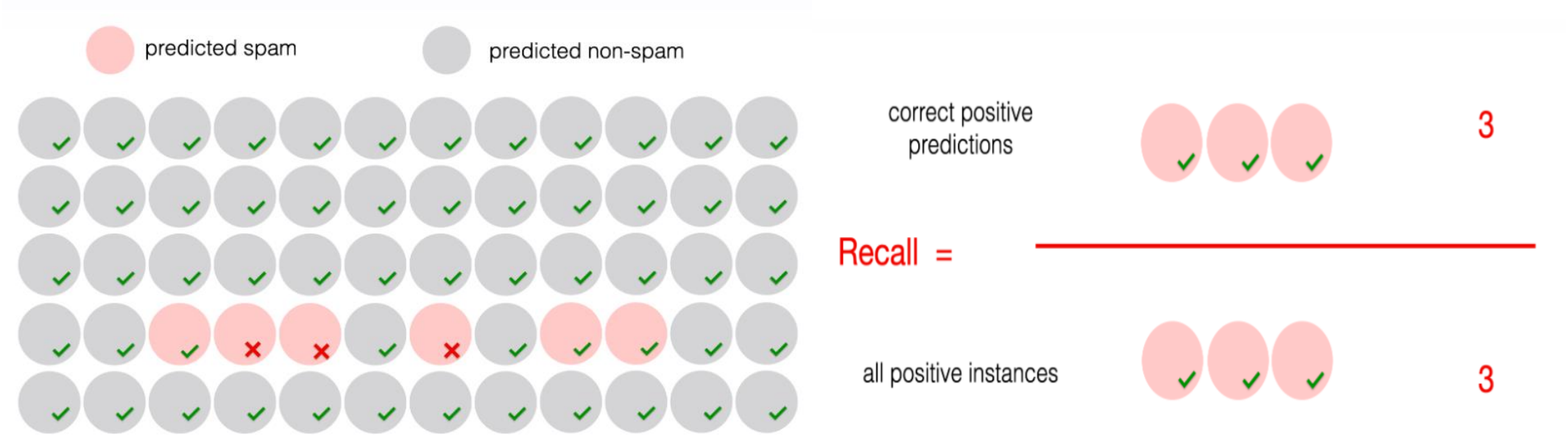

correct positive predictions
3
Precision =
all positive predictions
6

Recall

- Recall is a metric that measures how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset. You can calculate recall by dividing the number of true positives by the number of positive instances. The latter includes true positives (successfully identified cases) and false negative results (missed cases).

- In other words, recall answers the question: can an ML model find all instances of the positive class?

Example


predicted spam
predicted non-spam
correct positive predictions
3
Recall =
all positive instances
3

F1-score

- The F1 score combines precision and recall using their harmonic mean, and maximizing the F1 score implies simultaneously maximizing both precision and recall. Thus, the F1 score has become the choice of researchers for evaluating their models in conjunction with accuracy.

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

TP = number of true positives

FP = number of false positives

FN = number of false negatives

Implementaion in Python

```
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score

#the_y_test is our labels and the predictions is the output
of the model

accuracy = accuracy_score(the_y_test, predictions)

precision = precision_score(the_y_test, predictions)

recall = recall_score(the_y_test, predictions)

f1 = f1_score(the_y_test, predictions)
```