Introduction to Research

#F1-score
#Precision
#Accuracy
#Recall

What is 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.

$$\#Accuracy = \frac{\text{correct predictions}}{\text{All predictions}}$$

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

- TP (True Positives): Correctly predicted positive instances.
- TN (True Negatives): Correctly predicted negative instances.
- FP (False Positives): Incorrectly predicted positive instances (should have been negative).
- FN (False Negatives): Incorrectly predicted negative instances (should have been positive).

Example:

Let’s say you’re building a spam detection system, and out of 100 emails:

- 90 are spam, and the system correctly identifies 85 of them as spam
- 5 are non-spam, and the system correctly identifies them as non-spam

Accuracy would be:

$$\text{Accuracy} = \frac{85+5}{100} = 90\%$$

Caveat:

Accuracy can be misleading in imbalanced datasets. For instance, if 95% of the data is spam and the model predicts every email as spam, it would still get 95% accuracy, even though it never correctly identifies the non-spam emails.

What is 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).Precision is high when the model doesn’t make many False Positive errors.

#Formula:Precision= $\frac{TP}{TP+FP}$

- TP (True Positives): Correctly predicted positive instances.
- FP (False Positives): Incorrectly predicted positive instances (should have been negative).

Example:

In the spam detection system, the model predicts 100 emails as spam, but only 85 of those are actually spam (True Positives). The rest (15) are non-spam (False Positives). Precision would be:

Precision: $\frac{85}{85+15} = 85\%$

When to use it:

Precision is especially important in cases where False Positives are costly. For example, in medical diagnosis or fraud detection, predicting something incorrectly as positive could have severe consequences

What is 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).A high recall means the model is good at finding all the positive cases, even if it means generating some False Positives.

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

Example:

Using the spam detection system example: out of 100 actual spam emails, the model correctly identifies 85 as spam (True Positives), but it misses 5 actual spam emails (False Negatives). Recall would be:

$$\text{Recall} = \frac{85}{85+5} = 94.4\%$$

When to use it:

Recall is crucial when False Negatives are more concerning. For example, in disease screening, you don't want to miss any actual cases, even if it means occasionally flagging healthy individuals for further testing.

What is F1-Score?

The F1-Score is the harmonic mean of Precision and Recall. It’s a balance between the two and is especially useful when you need to find a trade-off between precision and recall. A perfect F1-Score (1.0) occurs when both precision and recall are equally high.

The formula is:

$$\#F1=2\times\frac{(\text{Precision}\times\text{Recall})}{(\text{Precision}+\text{Recall})}$$

Example:

Using the spam detection example, let's say the model has:

- Precision = 85%
- Recall = 94.4% The F1-Score would be:

$$F1 = 2 \times \frac{(0.85 \times 0.944)}{(0.85 + 0.944)} \approx 89.4\%$$

When to use it:

F1-Score is helpful when you need a balance between False Positives and False Negatives. It's especially valuable in cases where the data is imbalanced or when neither precision nor recall can be prioritized over the other.

The code:

You will need to prepare your dataset that includes predicted values for each class and true labels and pass it to the tool. You will instantly get an interactive report that includes a confusion matrix, accuracy, precision, recall metrics, ROC curve and other visualizations. You can also integrate these model quality checks into your production pipelines.

```
classification_performance_report = Report(metrics=[
    ClassificationPreset(probas_threshold=0.7),
])

classification_performance_report.run(current_data=bcancer_cur, reference_data=bcancer_ref)

classification_performance_report
```

Conclusion

- Accuracy: Overall correctness.
- Precision: How often positive predictions are right.
- Recall: How many real positives are captured.
- F1-Score: Balance between precision and recall.

Thank You