# Binary Classification metrics

## Summary – what metric should I use?

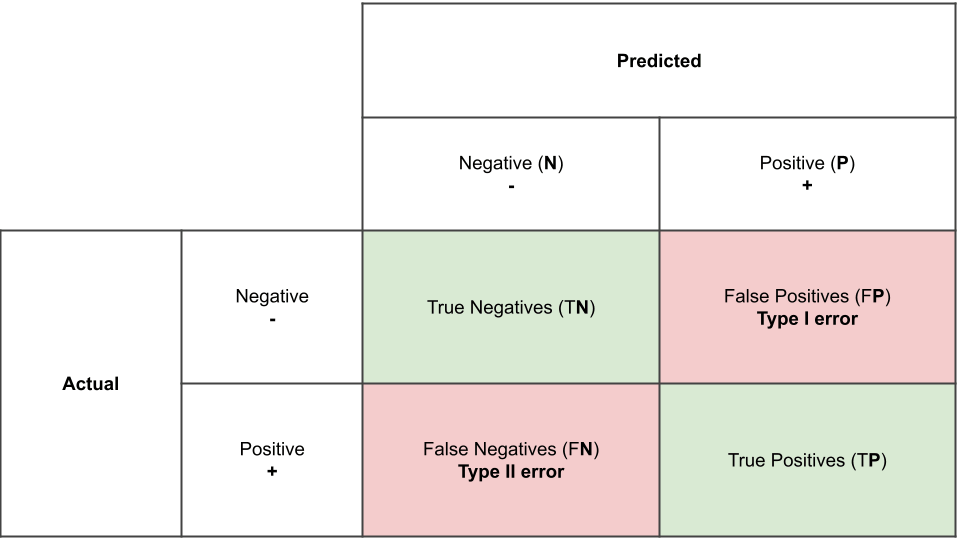
The metrics, penalties & dataset balance shown in the table are explained more in this document.

|  |  |  |
| --- | --- | --- |
|  | Unbalanced Dataset  (≥ 80% in majority class) | ~Balanced dataset  (< 80% in majority class) |
| Unequal Penalty for False Negative (FN) or False Positive (FP) | **f1**  **Precision & Recall** | **Precision & Recall**  **ROC & AUC** |
| Equal Penalty for FN & FP | **f1** | **Accuracy**  **ROC & AUC** |

**Table 1: Recommended binary classification metrics**

In the table above, “unequal penalty” means that your problem has a penalty for a False Positive (FP) that is higher or lower than the penalty for a False Negative (FN). Penalties are discussed more in the precision/recall section on the next page.

## Confusion matrix



**Table 2: Sample confusion matrix**

* Many references use acronyms for the parts of the confusion matrix, such as TP & FN - these acronyms are defined in the graphic above.
* There are many guides to confusion matrices available online, but please note than some of them have the rows/columns flipped – this can add to the “confusion.” The graphic above mirrors what the python confusion\_matrix function outputs.

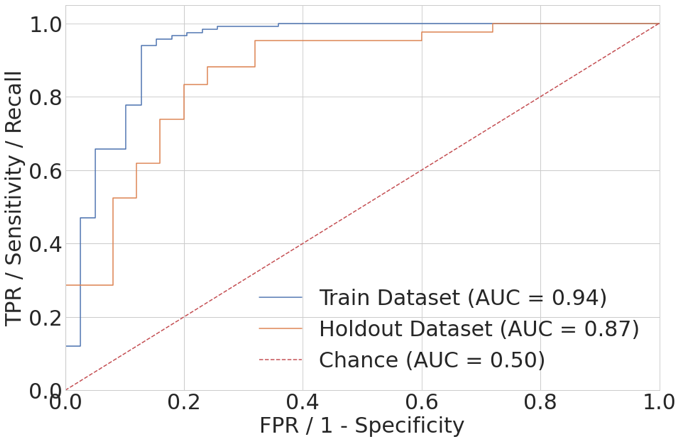
## Metrics based on confusion matrix

Your selection of metrics is based on your problem and your dataset. The **problem** tells you the penalty associated with false negatives & false positives. For example, in a DoD drug testing problem, there is a very high penalty for a false positive, which is a Type I error on the previous page. In this case, a member could face adverse penalties if they didn’t use drugs but the test said they did.

The balanced or unbalanced nature of the **dataset** you are modeling contributes as well, which is the ratio of 0’s and 1’s in the label. It can be balanced (<80% in the majority class) or unbalanced (>80% in the majority class). As shown in Table 1, some metrics are better at providing insight onto the model performance of unbalanced datasets.

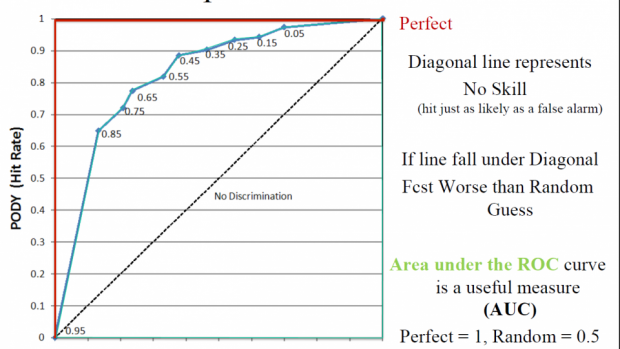
More information on selected metrics is below:

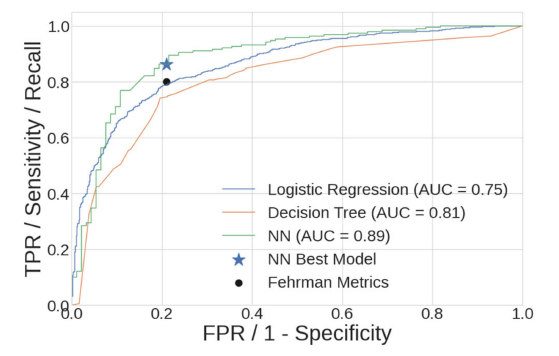
* Use **accuracy** when
  + you have a balanced dataset (majority class <80%), and
  + false negatives & false positives have similar penalties.
  + Accuracy = (all **correct** / all) = TP + TN / TP + TN + FP + FN. Higher is better.
  + Accuracy value of 90% means that of 10 labels:
    - 9 were correct (TP or TN) and
    - 1 was incorrect (FP or FN).
* Use **f1** (harmonic mean of precision and recall) if
  + If you have an unbalanced dataset.
  + Higher f1 is better
  + This metric can be substituted for the g-mean, which is the geometric mean of precision and recall
* Choose **Precision &** **Recall** if
  + There is a high penalty associated with false negatives, and a low penalty for false positives, or vice versa.
  + Diabetes example: positive (1) is diabetes, negative (0) is healthy
    - In diabetes prediction, you’d rather get some extra false positives (healthy people labeled diabetic) than false negatives (a diabetic person labeled healthy). There is a high penalty for missing a diagnosis.
  + Spam example: positive (1) = good email, negative (0) = spam
    - You’d rather have some spam emails in your inbox (false positive) than have legitimate email sent to your spam folder.
    - So, the email company wants to be extra sure that an email is spam before they put it in the spam box and you never get to see it. There is a high penalty for a false negative in this case.
  + **Precision**
    - **true** positives / **predicted** positives. Higher is better.
    - Precision of 80% means that, of 10 emails predicted as good,
      * 8 were good, and
      * 2 were spam 8/(2+8=10).
  + **Recall**
    - **true** positives / all **actual** positives = TP / (TP + FN). Higher is better.
    - Recall value is 70% means that of 10 people:
      * 7 were correctly classified as diabetic (TP), and
      * 3 were diabetic but falsely classified as health (FN)
* **Specificity** & **Sensitivity** are not used in this course, but are commonly used in medical journals
  + **ROC curves** are created based on Specificity & Sensitivity
  + **Sensitivity** = **Recall** (definition above) = True Positive Rate
  + **Specificity =** TN / (TN + FP) = (1 – False Positive Rate)
  + An example ROC curve is shown below, where the axis labels shown the equivalent terms above



**ROC Curve**

* The default classification threshold is 0.5, however sometimes a better model can result from a different classification threshold
* The ROC Curve is created by sweeping the classification threshold from 0 to 1 and plotting the Specificity & Sensitivity (related to Precision and Recall) that result from each classification threshold.
* The first chart below shows a ROC curve where the thresholds (0.95, 0.85 etc) are overlaid on the curve
* The second chart shows comparative ROC curves that have better axis labels. In the labels, TPR = true positive rate and FPR = false positive rate

<https://www.swpc.noaa.gov/content/roc-receiver-operating-characteristic-curves>



**AUC**

* AUC is the integrated area under the ROC Curve.

# Multi-class Classification metrics

## Summary – what metric should I use?

The metrics, penalties & dataset balance shown in the table are explained more in this document.

|  |  |  |
| --- | --- | --- |
|  | Unbalanced Dataset  (visual inspection) | ~Balanced dataset  (visual inspection) |
| Some classes more important than others | **Weighted average of**  **f1, Precision & Recall** | **Macro average of**  **Precision & Recall** |
| All classes equally important | **Weighted average of**  **f1** | **Macro average of**  **Accuracy** |

**Table 3: Recommended multiclass classification metrics**