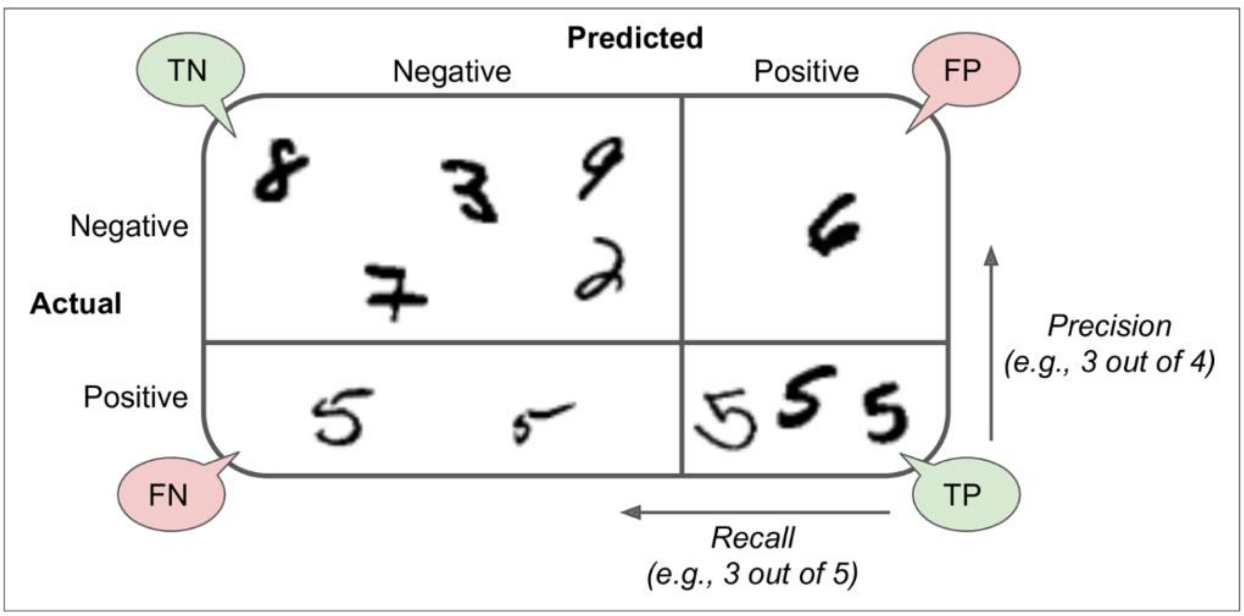
Classification

Performance Measures

Issue with Accuracy in Imbalanced Datasets: Accuracy alone can be misleading in imbalanced datasets. For instance, if a dataset has 99% of 'yes' labels, a naive model that always predicts 'yes' would have 99% accuracy, which doesn't indicate the model's true predictive power.

Better Performance Measure: Confusion Matrix



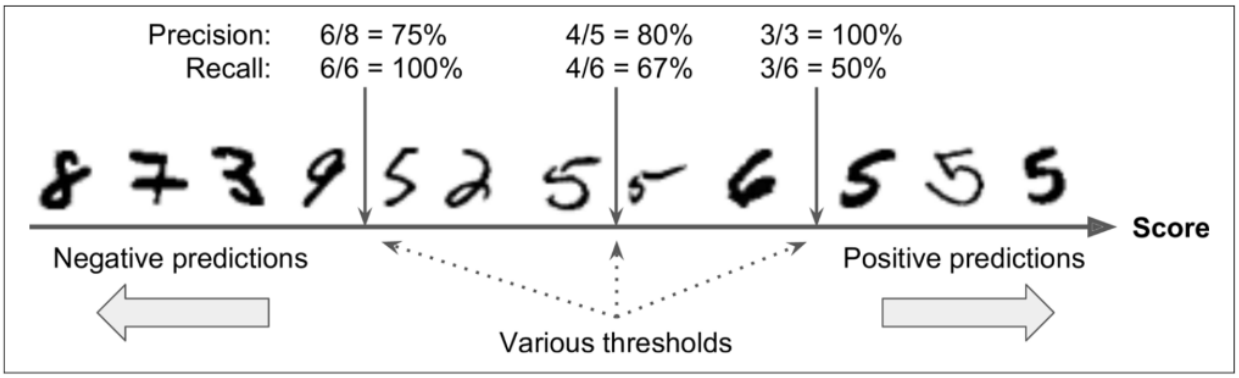
Performance Measures

Issue with Accuracy in Imbalanced Datasets: Accuracy alone can be misleading in imbalanced datasets. For instance, if a dataset has 99% of 'yes' labels, a naive model that always predicts 'yes' would have 99% accuracy, which doesn't indicate the model's true predictive power.

Better Performance Measure: Confusion Matrix

Components of a Confusion Matrix:

- True Negatives (TN)
- False Positives (FP)
- False Negatives (FN)
- True Positives (TP)



Derived Metrics and Formulas:

Recall (Sensitivity):
- Formula: Recall = TP / (TP + FN)
- Example: If there are 5 actual positives and the model correctly identifies 3, Recall = 3 / (3 + 2) = 0.6 or 60%.

Precision:
- Formula: Precision = TP / (TP + FP)
- Example: If the model predicts 4 positives but only 3 of them are correct, Precision = 3 / (3 + 1) = 0.75 or 75%.

F1 Score:
- Formula: F1 = 2 × (Precision × Recall) / (Precision + Recall)
- F1 Score is especially useful when you seek a balance between Precision and Recall.

Precision vs. Recall

Contextual Importance: The relative importance of precision and recall depends on the application.

- Example 1: For child-safe video filtering, high precision (fewer false positives) is more crucial than high recall.
- Example 2:
  In shoplifter detection, high recall (catching almost all shoplifters) is more critical, even at the cost of lower precision (more false alerts).

Precision/Recall Trade-off

Selecting Balance:
The choice of balance between precision and recall depends on specific project needs. For example, one might choose a point just before precision drops sharply, say at 60% recall.

- **Example Illustrating Precision and Recall**:

- **Scenario**: Consider two fishermen, Fisherman A and Fisherman B.

- **Fisherman A (High Precision, Low Recall)**: Fisherman A is very selective and only tries to catch expensive fish.
  He catches fewer fish, but almost all of them are expensive. This represents high precision (most catches are the 'positive' outcome he wants) but low recall (many expensive fish are n ot caught).

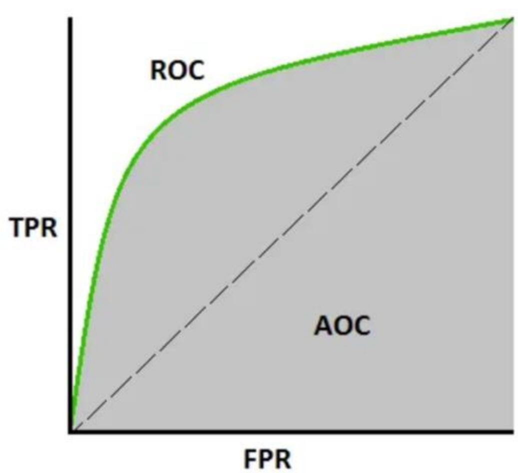
- **Fisherman B (Low Precision, High Recall)**: Fisherman B tries to catch as many fish as possible, hoping some of them are expensive.
  He catches a lot of fish, but only a few are expensive. This represents low precision (fewer c atches are the 'positive' desired outcome) but high recall (he catches most of the expensive fish in the area, along with many others).

In summary, Fisherman A's strategy is akin to a model that prioritizes precision (accuracy of posi tive predictions), while Fisherman B's approach is like a model that focuses on recall (covering a ll actual positives).

ROC Curve and AUC

- **ROC Curve**: Plots True Positive Rate (Recall) versus False Positive Rate (1 - Specificity).
- **Area Under Curve (AUC)**: Measures the entire two-dimensional area underneath the ROC curve.

  A perfect classifier has an AUC of 1, while a random classifier has an AUC of 0.5.



Multiclass Classification

Handling Multiple Classes: Some algorithms like SGD classifiers, Random Forest classifiers, and Naive Bayes can handle multiple classes natively.

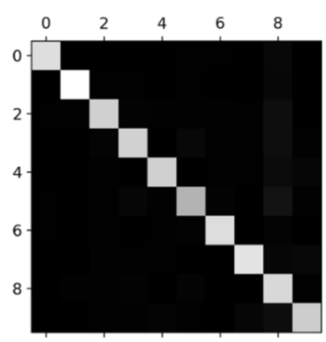
Binary Classifiers for Multiclass Tasks:

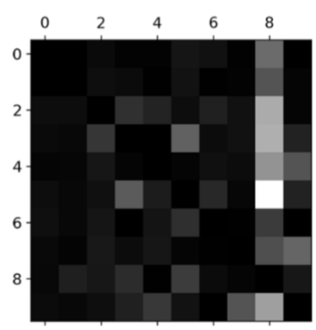
- **One-versus-Rest (OvR)**: Train a separate classifier for each class.
- **One-versus-One (OvO)**: Train a classifier for every pair of classes. For N classes, there are N × (N – 1) / 2 classifiers.

Error Analysis

Using Confusion Matrix for Insights:

- Analyzing the matrix can reveal specific areas where the classifier is underperforming.
- The brightness of cells in the confusion matrix can indicate the frequency of predictions and errors.
- 1st image: The darkest colors is where there are more errors
- 2nd image: The assymetry changes in the color indicate problems in classification, if there is no symetry indicates that the column there are not many images, ex: 8





Multilabel Classification

- **Handling Multiple Labels**: KNeighborsClassifier can be used for tasks where each instance may have multiple labels.
- **Example**: In face recognition, a single image might contain several known faces.
- **Data Balance**: Ensuring balance in the dataset is crucial to prevent bias towards more frequent labels.

Multioutput Classification

- A generalization of multilabel classification where each label can have multiple classes.
- **Example Use-case**: Building a system to remove noise from images, where each pixel can have multiple class labels representing noise levels.