Classification Problems: Assigning Categories

Distinct from regression (which predicts continuous values), **Classification** is a type of supervised learning task where the goal is to predict a **discrete output variable (y)** based on one or more input variables (features, X). The model learns an approximate mapping function f such that y = f(X).

* **Discrete Output:** The target variable y belongs to one of a finite number of categories or classes. These output variables are often called **labels** or **categories**.
* **Goal:** The mapping function predicts the class label or category for a given observation (set of input features).
* **Example:** Classifying an email based on its text content as belonging to one of two classes: "spam" or "ham" (not spam).

Key Characteristics of Classification:

* **Predicting a Category:** The core task is to assign an instance to a predefined class.
* **Labeled Data:** Requires training data where examples are already labeled with the correct class.
* **Input Types:** Like regression, input features (X) can be real-valued or discrete.
* **Types of Classification Problems:**
  + **Binary Classification:** Problems with exactly two possible output classes (e.g., spam/ham, yes/no, true/false, positive/negative).
  + **Multi-class Classification:** Problems with more than two possible output classes (e.g., classifying handwritten digits 0-9, identifying different types of fruit).
  + **Multi-label Classification:** Problems where an example can be assigned *multiple* classes simultaneously (e.g., tagging a news article with topics like 'politics', 'economy', 'europe').

Predicting Probabilities

It's common for classification models not to output the class label directly. Instead, they often predict a **continuous value representing the probability** (or a score related to probability) of a given example belonging to each possible output class.

* These probabilities can be interpreted as the model's likelihood or confidence.
* To get the final class label, a decision rule is applied, typically selecting the class label with the highest predicted probability.

*Example:* For a specific email, a model might output: P(spam) = 0.1 P(ham) = 0.9 Since the probability for "ham" (0.9) is higher than for "spam" (0.1), we convert these probabilities to the class label "ham".

Performance Metrics for Classification Problems

Evaluating classification models requires different metrics than regression because we are dealing with correct/incorrect category assignments rather than the magnitude of numerical error. The foundation for most classification metrics is the **Confusion Matrix**.

The Confusion Matrix

A confusion matrix is a table that summarizes the performance of a classification model by comparing the predicted class labels against the actual class labels for a set of test data. It shows how many predictions were correct and what types of incorrect predictions were made.

Let's consider a **Binary Classification** example with two labels: **Positive (P)** and **Negative (N)**.

The confusion matrix captures the counts of the following outcomes:

|  | **Predicted class = P** | **Predicted class = N** | **Total Actual** |
| --- | --- | --- | --- |
| **Actual class = P** | **TP** (True Positive) | **FN** (False Negative) | **P** |
| **Actual class = N** | **FP** (False Positive) | **TN** (True Negative) | **N** |
| **Total Predicted** | **TP + FP** | **FN + TN** | **P + N** |

Where:

* **True Positives (TP):** Count of Positive labels that the model correctly predicted as Positive. (Actual=P, Predicted=P)
* **True Negatives (TN):** Count of Negative labels that the model correctly predicted as Negative. (Actual=N, Predicted=N)
* **False Positives (FP):** Count of Negative labels that the model incorrectly predicted as Positive. (Actual=N, Predicted=P). Also known as a **Type I Error**.
* **False Negatives (FN):** Count of Positive labels that the model incorrectly predicted as Negative. (Actual=P, Predicted=N). Also known as a **Type II Error**.

The diagonals (TP, TN) represent correct predictions, while the off-diagonals (FP, FN) represent errors.

Common Classification Metrics Derived from the Confusion Matrix

1. Accuracy (ACC)

* **Definition:** The most intuitive metric; it measures the overall proportion of predictions that the model got correct (both positive and negative).
* **Formula:**
* Accuracy = (TP + TN) / (Total Population)
* = (TP + TN) / (P + N)
* = (TP + TN) / (TP + TN + FP + FN)
* **Caution:** Accuracy can be misleading, especially on **imbalanced datasets**. If 99% of emails are "ham," a model predicting "ham" for every email achieves 99% accuracy but fails entirely at the task of identifying "spam."

2. Sensitivity (Recall / True Positive Rate - TPR)

* **Definition:** Measures the proportion of actual positive instances that were correctly identified by the model. Answers the question: "Of all the actual positive cases, how many did we catch?"
* **Formula:**
* Sensitivity / Recall / TPR = TP / (Actual Positives)
* = TP / P
* = TP / (TP + FN)
* **Relationship:** TPR = 1 - FNR (where FNR is the False Negative Rate = FN / P)
* **Importance:** High recall is crucial when missing a positive case is costly (e.g., disease detection - minimizing False Negatives).

3. Specificity (Selectivity / True Negative Rate - TNR)

* **Definition:** Measures the proportion of actual negative instances that were correctly identified by the model. Answers the question: "Of all the actual negative cases, how many did we correctly rule out?"
* **Formula:**
* Specificity / TNR = TN / (Actual Negatives)
* = TN / N
* = TN / (TN + FP)
* **Relationship:** TNR = 1 - FPR (where FPR is the False Positive Rate = FP / N)
* **Importance:** High specificity is important when incorrectly classifying a negative case as positive is costly (e.g., spam filtering - minimizing False Positives that block legitimate emails; or avoiding unnecessary expensive medical procedures).

4. Precision (Positive Predictive Value - PPV)

* **Definition:** Measures the proportion of instances predicted as positive that were actually positive. Answers the question: "Of all the cases we predicted as positive, how many were actually positive?"
* **Formula:**
* Precision / PPV = TP / (Predicted Positives)
* = TP / (TP + FP)
* **Relationship:** PPV = 1 - FDR (where FDR is the False Discovery Rate = FP / (TP + FP))
* **Importance:** High precision is crucial when the cost of a False Positive is high (e.g., recommending a product - you want the recommendations to be relevant; medical diagnosis where a positive result leads to serious treatment).

5. F1-Score

* **Definition:** The harmonic mean of Precision and Recall. It tries to provide a single score that balances both concerns. The harmonic mean gives more weight to lower values, so the F1-score will be high only if both precision and recall are high.
* **Formula:**
* F1 = 2 \* ( (Precision \* Recall) / (Precision + Recall) )
* = 2 \* TP / (2\*TP + FP + FN)
* **Importance:** Useful when you need a balance between Precision and Recall, especially if the class distribution is uneven. A high F1 score indicates that the model has low false positives AND low false negatives.

Choosing the right metric (or combination of metrics) depends heavily on the specific goals of the classification task and the relative costs associated with different types of errors (False Positives vs. False Negatives).