Classification Systems

A Visual Study Guide & Code Reference for Chapter 3
Hands-On Machine Learning by Aurélien Géron

Contents

1. Binary Classification Fundamentals

The journey into classification begins with the MNIST dataset, a collection of 70,000 handwritten digits. To simplify the problem, we first tackle a **binary classification** task: determining if a digit is a '5' or not. This creates two distinct classes: '5' and 'not-5'.

1.1 The MNIST Dataset & SGD Classifier

We use the `SGDClassifier` from Scikit-Learn. It's an excellent choice for large datasets because it processes instances one at a time (*online learning*), making it highly memory-efficient.

```python
import numpy as np
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import cross_val_score

# Assuming X_train, y_train are pre-loaded
# Create the binary target variables for the '5' class
y_train_5 = (y_train == 5)
y_test_5 = (y_test == 5)

# Initialize and train the Stochastic Gradient Descent Classifier
sgd_clf = SGDClassifier(random_state=42)
sgd_clf.fit(X_train, y_train_5)

# Evaluate using 3-fold cross-validation
scores = cross_val_score(sgd_clf, X_train, y_train_5, cv=3, scoring='accuracy')
print(f"Accuracy scores: {scores}")
```

Listing 1: Setting up the Binary Classification Task

Scikit-Learn Insight

The `SGDClassifier` is a powerful and versatile tool. Its efficiency stems from processing single samples at each step, which is ideal for online learning scenarios where data arrives sequentially.

1.2 Why Accuracy Can Be Misleading

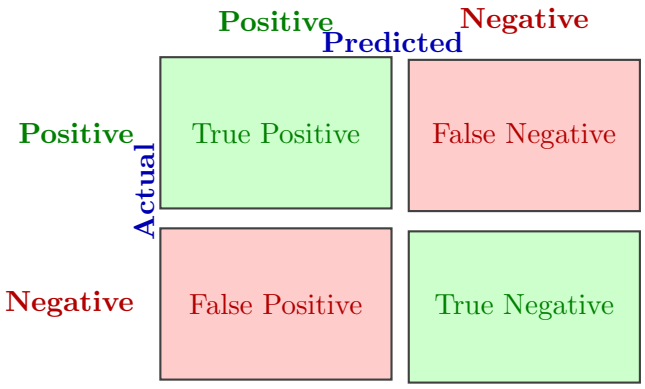
In our '5 vs. not-5' task, only about 10

2. The Confusion Matrix: A Deeper Look

The confusion matrix provides a comprehensive breakdown of classification performance, showing how many times instances of class A were misclassified as class B.

article tikz

article tikz


Positive
Negative
Predicted
Positive
Actual
True Positive
False Negative
Negative
False Positive
True Negative