CS550/DSL501: Machine Learning (2024–25–M)
Assignment-II

Full name: Nitin Mane **ID: M24MT004**

1 Problem Statement

Imagine a machine learning model designed to diagnose a rare disease that affects only 1% of the population. In a test set of 10,000 patients, only 100 actually have the disease. The confusion matrix from the model predictions might look like this:

- **True Positives (TP):** 80 patients correctly diagnosed with the disease.
- **False Negatives (FN):** 20 patients who have the disease but were incorrectly diagnosed as not having it.
- **False Positives (FP):** 100 patients who do not have the disease but were incorrectly diagnosed as having it.
- **True Negatives (TN):** 9,800 patients correctly diagnosed as not having the disease.

so which performance parameter you should prefer for model performance?

1.1 Solution

Assuming that the parameters listed in the problem statement correspond to the model's output, where performance is quantified using a probabilistic approach based on the model's true values and error rates, we can verify that the performance metrics rely on the predictive outcome, assisting in the identification of better responses that are justified in the model's use case in a real-time setting. In order to do this, we must assess the outcome using the following method of computation.

Mathematical Formulas for Metrics

Given the confusion matrix, the following metrics can be calculated:

- **Accuracy:** The proportion of correctly classified instances among the total instances.

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

- **Precision:** The proportion of true positives among all predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall (Sensitivity):** The proportion of true positives among all actual positives.

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

- **F1 Score:** The harmonic mean of precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Mean Squared Error (MSE):** The average squared difference between the actual labels and predictions.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\text{Actual}_i - \text{Predicted}_i)^2$$

- **Root Mean Squared Error (RMSE):** The square root of the MSE.

$$\text{RMSE} = \sqrt{\text{MSE}}$$

Code Implementation

Assume the following confusion matrix values:

TP	80
FN	20
FP	100
TN	9800

Listing 1: Python Code to Calculate Metrics

```
# Assume the following values from the confusion matrix
TP = 80
FN = 20
FP = 100
TN = 9800

# Calculate precision = (TP + TN) / (TP + TN + FP + FN)

# Calculate Precision
precision = TP / (TP + FP) if (TP + FP) > 0 else 0

# Calculate Recall
recall = TP / (TP + FN) if (TP + FN) > 0 else 0

# Calculate the F1 score
f1_score = 2 * (precision * recall) / (precision + recall)
                if (precision + recall) > 0
                else 0
```

```
# Calculate Mean Squared Error (MSE)
actual_labels = [1] * (TP + FN) + [0] * (TN + FP)
predictions = [1] * TP + [0] * FN + [1] * FP + [0] * TN
mse = sum((a - p) ** 2 for a, p in zip(actual_labels, predictions)) /
len(actual_labels)

# Calculate Root Mean Squared Error (RMSE)
rmse = mse ** 0.5

# Display the results
print(f"Accuracy:␣{accuracy:.4f}")
print(f"Precision:␣{precision:.4f}")
print(f"Recall:␣{recall:.4f}")
print(f"F1␣Score:␣{f1_score:.4f}")
print(f"MSE:␣{mse:.4f}")
print(f"RMSE:␣{rmse:.4f}")
```

Results:

- **Accuracy:** 0.9880
- **Precision:** 0.4444
- **Recall:** 0.8000
- **F1 Score:** 0.5714
- **MSE:** 0.0120
- **RMSE:** 0.1095

1.2 Answer

The optimum performance metric for identifying a rare disease should be ***Recall (Sensitivity)*** *because missing a positive case (False Negative) can have serious repercussions. The risk of misdiagnosing a patient with the illness is reduced because it guarantees that the majority of true positive cases are appropriately diagnosed. In medical situations where False Negatives could seriously injure patients, it is imperative to prioritize recall.*