CISC /CMPE 251 Assignment 3 Part 1: Theoretical

Q1: Precision vs. Recall vs. F1-Score

Explain the differences between **Precision**, **Recall**, and **F1-Score** in classification problems. Under what circumstances would you prefer to prioritize one metric over the others? Provide examples of real-world applications for each case.

In classification problems, precision, recall, and F1-Score are performance metrics. Each of them gives insight to how well the model performs when handling the imbalance of classes. All of the metrics have a different focus in terms of classification accuracy. The decision of which metric to prioritize will be different based on the goals and risks of the classification problem being solved. A summary of each metric is provided below:

1. Precision

- The ratio of correctly predicted positive observations to the total predicted positives. Precision has the following formula:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- When should precision be the prioritized metric?
    - Precision is important when the cost of false positives is high. You want to ensure that when the model predicts a positive, it is almost certainly correct.
- What is an example of a case where precision should be prioritized?
    - Let's say I am trying to filter the spam from my email inbox. In this circumstance it's crucial that my legitimate emails aren't filtered and labelled as spam because this could lead to them going unseen.

2. Recall

- The ratio of correctly predicted positive observations to all real positives. Recall has the following formula:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- When should recall be the prioritized metric?
    - Recall is important when the cost of a false negative is high, and you want to identify as many true positives as you can.
- What is an example of a case where recall should be prioritized?
    - Recall should always be prioritized in medical diagnoses. Take cancer screening for example, it's vital to detect as many positives as possible, even if a few

negatives are incorrectly labelled. A missing positive case could lead severe consequences.

3. F1-Score

- The harmonic mean of precision and recall, balancing both metrics the F1-score is calculated using the following formula:

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

- When should F1-score be the prioritized metric?
    - F1-score is best prioritized when you need a balance between both precision and recall. When both false positives and false negatives carry heavy weight then F1-score is important.
- What is an example of a case where F1-score should be prioritized?
    - F1-score is used heavily in search engine applications. A search engine will rank relevant content and must balance producing lots of relevant results, high recall, with ensuring the irrelevant content is filtered out, high precision.

Q2: Regression Evaluation Metrics

Given the following predicted and actual values for a regression problem, calculate the **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, and **R^2 Score**.

***Predicted values**: [3, 4.5, 6, 7.5, 9]*

***Actual values**: [3.2, 4, 6.1, 7.4, 8.9]*

1. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$MAE = \frac{|3.2 - 3| + |4 - 4.5| + |6.1 - 6| + |7.4 - 7.5| + |8.9 - 9|}{5} = 0.20$$

2. Mean Square Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$MSE = \frac{(3.2 - 3)^2 + (4 - 4.5)^2 + (6.1 - 6)^2 + (7.4 - 7.5)^2 + (8.9 - 9)^2}{5} = 0.064$$

3. R^2 Score

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - y)^2}$$

Where *y* is the mean of the actual values

a. Calculate *y*

$$y = \frac{3.2 + 4 + 6.1 + 7.4 + 8.9}{5} = 5.92$$

b. Calculate total sum of squares

$$total\ sum\ of\ squares\ = \sum_{i=1}^{n}(y_i - y)^2$$

$$= (3.2 - 5.92)^2 + (4 - 5.92)^2 + (6.1 - 5.92)^2 + (7.4 - 5.92)^2 + (8.9 - 5.92)^2 = 22.188$$

c. Calculate the sum of squares of residuals

$$total\ sum\ of\ squares\ of\ residuals\ = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$= (3.2 - 3)^2 + (4 - 4.5)^2 + (6.1 - 6)^2 + (7.4 - 7.5)^2 + (8.9 - 9)^2 = 0.064$$

d. Calculate R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - y)^2} = 1 - \frac{0.064}{22.188} = 0.997$$

Q3: Weighted Accuracy

Weighted accuracy in classification adjusts standard accuracy by assigning different weights to each class, accounting for class importance or imbalance. It gives more significance to underrepresented or important classes, offering a more balanced performance measure, especially in imbalanced datasets. The general formula is:

$$weighted\ accuracy = \sum_{i=1}^{n} w_i \times accuracy_i$$

Where …

- n is the number of classes.
- w_i is the weight assigned to class i (often proportional to the number of samples in that class or based on class importance).
- *accuracy* is the accuracy for class *i*, which is the fraction of correctly predicted samples for that class.

For a classification problem, given the following confusion matrix, calculate the **weighted accuracy** of the model assuming class weights are based on the class distribution in the training set.

	Pred Class 1	Pred Class 2
Actual Class 1	50	10
Actual Class 2	5	35

Assume Class 1 had 200 samples in the training set, and Class 2 had 100 samples. Calculate the weighted accuracy.

We will start by interpreting the given confusion matrix:

1. Class #1:
    a. 50 samples were **correctly** classified as class #1
    b. 10 samples were **misclassified** as class #2
2. Class #2:
    a. 35 samples were **correctly** classified as class #2
    b. 5 samples were **misclassified** as class #1

Next, we calculate the accuracy for each class:

1. Accuracy for class #1: the faction of correctly classified samples in class #1

$$accuracy_1 = \frac{true\ positives}{total\ samples} = \frac{50}{60} = 0.833$$

2. Accuracy for class #2: the faction of correctly classified samples in class #2

$$accuracy_2 = \frac{true\ positives}{total\ samples} = \frac{35}{40} = 0.875$$

Now we calculate the class weights based on training distribution:

1. Weight for class #1:

$$w_1 = \frac{total\ samples\ class\ 1}{total\ samples\ in\ training} = \frac{200}{300} \approx 0.667$$

2. Weight for class #2:

$$w_2 = \frac{total\ samples\ class\ 2}{total\ samples\ in\ training} = \frac{100}{300} \approx 0.333$$

Finally, we can calculate the weighted accuracy:

$$weighted\ accuracy = \sum_{i=1}^{n} w_i \times accuracy_i$$

$$= (0.667 \times 0.833) + (0.333 \times 0.875) = 0.847$$

Therefore, the weighted accuracy of the model is approximately 0.847. Weighted accuracy reflects the accuracy adjusted for the class distribution in the training set. More weight is given to class #1 as it had majority of samples.

Q4: Difference Between Cross-Entropy Loss and Hinge Loss in Classification

Explain the difference between **cross-entropy loss** (used for logistic regression and SoftMax classifiers) and **hinge loss** (used in support vector machines). What are the strengths and weaknesses of each in handling noisy data and outliers?

Both cross entropy loss and hinge loss are common loss functions leveraged is machine learning applications. Similar to the success metrics examined in question 1, different loss functions will be utilized dependent on the specific problem or task being performed. Both loss functions are explained in detail below:

1. Cross Entropy Loss (CEL)

Also referred to as log loss, CEL refers to the distance between predicted probability distribution and the ground truth distribution. CEL is calculated with the following formula:

$$L_{CE} = \sum_i y_i \log(\hat{y}_i)$$

Where y_i is the true label and $\hat{y}_i$ is the predicted probability of the true class.

CEL is best suited for models that produce probability distributions. It encourages the model to output higher confidence in correct predictions and it is a smooth differentiable function. Cross entropy can be easily adapted to multiclass classification with the SoftMax function. However,

CEL is relatively sensitive to outliers as it will tend to increase rapidly when the model is very confident but incorrect. Furthermore, because the function emphasizes highly confident predictions, cross entropy may be skewed by noisy labels which can cause overfitting or noise.

2. Hinge Loss (HL)

Hinge loss is typically used within support vector machines (SVMs), and it penalizes points that lie on the incorrect side of the margin line. For binary classification where labels $y \in \{-1, +1\}$ we have the following equation:

$$L_{Hinge} = \max(0, 1 - y * \hat{y})$$

Where y is the true label and $\hat{y}$ is the predicted score before applying the sign function.

Hinge loss penalizes predictions that don't meet a certain margin which encourages decision boundaries with a strong margin. Contrary to CEL, HL is not smooth as points that do not meet the margin are penalized but there is no penalty if margins are met. Since the focus of this loss function is the margin errors, it is less sensitive to the misclassified outliers. Hinge loss prioritizes margins between classes this results in successful generalizations for low noise datasets. For higher noise datasets however, this function is less successful. When classes overlap significantly hinge loss does not account for uncertain regions. Hinge loss also does not produce probability scores, so it doesn't provide highly interpretable data in terms of prediction confidence.

Q5: RMSE vs. MAE on Skewed Data

You are working on a regression task where the target variable is highly skewed, with many values clustered near zero and a few extreme outliers. You trained two models and computed the RMSE and MAE on the test set:

Model A: RMSE = 10.5, MAE = 5.0

Model B: RMSE = 12.0, MAE = 4.5

Which model would you prefer based on these results, and why? What does the difference between RMSE and MAE suggest about the distribution of errors?

As previously discussed in question 2, MAE represents the Mean Absolute Error and can be calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

As for RMSE, this represents the root mean square error which is calculated with the following formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Therefore, RMSE represents the square root of average squared differences between predicted values and ground truth values. RMSE gives higher weight to large errors, so it has higher sensitivity to outliers.

In observing the model results we can see that model B has lower MAE than model A which tells us that it generally produces smaller errors on average along with being better at capturing a broader range of data points. Alternately, we observe model A has a lower RMSE than B suggesting it is likely handling extreme outliers better. We can conclude that model A has fewer large errors than B.

Another factor to consider is the difference between RMSE and MAE for each model A and B. We see that there is a higher difference between each type of error on model B than A. This indicates that model B probably has larger error for certain outliers but model A having a relatively small difference between RMSE and MAE indicates there may be smaller more consistent errors.

The question informs us that the target variable is highly skewed, with many values clustered near zero and a few extreme outliers. Given these characteristics we can conclude that it is likely model B will have higher success on this application.