Lecture Note 11: Classification Performance Metrics

1. Introduction

Classification performance metrics are essential tools used to evaluate and interpret the effectiveness of a classification model in machine learning and statistics. These metrics help in understanding how well a model can distinguish between different classes and predict the correct category for new observations. Given that classification tasks often involve predicting which category or class a particular observation belongs to, it's crucial to have reliable measures that can quantify a model's predictive accuracy, robustness, and error tendencies.

This note will introduce several key metrics that are commonly used, each providing unique insights into the model's performance: accuracy, precision (Positive Predictive Value), recall (Sensitivity, True Positive Rate), specificity (Selectivity, True Negative Rate), F1 Score, and Area Under the ROC Curve (AUC-ROC). All these metrics originate from the confusion matrix, making it the foundational element for understanding model performance in classification. Therefore, our discussion begins with an explanation of the confusion matrix in the context of binary classification, the most straightforward scenario.

2. Performance Metrics for Classification: Binary Case

2.1 Confusion Matrix with Example

A confusion matrix is a tool used to evaluate the performance of classification models by laying out the actual versus predicted classifications in a tabular format. The table below provides a confusion matrix for binary classification. Let's break down each component of the confusion matrix in the context of binary classification, where '1' indicates the positive class and '0' indicates the negative class.

Confusion Matrix: Binary Case

	Predict 1	Predict 0	
1 = Positive	True Positive (TP)	False Negative (FN)	Positive (P)
0 = Negative	False Positive (FP)	True Negative (TN)	Negative (N)
	Predicted Positive (PP)	Predicted Negative (PN)	Total (T)

- **True Positives (TP)**: These are cases where the model correctly predicts the positive class. In other words, the actual class is '1' (positive), and the model also predicts '1'.
- **False Negatives (FN)**: These occur when the model incorrectly predicts the negative class for an observation that is actually in the positive class. Essentially, the actual class is '1', but the model predicts '0'. This error type is sometimes referred to as a "Type II error."

- **False Positives (FP)**: Also known as "Type I errors," these happen when the model incorrectly predicts the positive class for an observation that is actually in the negative class. So, the actual class is '0', but the model predicts '1'.
- **True Negatives (TN)**: These are cases where the model correctly predicts the negative class. The actual class is '0', and the model also predicts '0'.
- **P (Positive)**: Represents the total number of actual positive cases in the dataset. It's the sum of True Positives (TP) and False Negatives (FN), indicating all instances where the true class is '1'.
- **N (Negative)**: Denotes the total number of actual negative cases in the dataset. This is the sum of True Negatives (TN) and False Positives (FP), covering all instances where the true class is '0'.
- **PP (Predicted Positive)**: The total number of cases predicted by the model as positive. It combines True Positives (TP) and False Positives (FP), reflecting every instance where the model's prediction is '1'.
- **PN (Predicted Negative)**: The total number of cases predicted by the model as negative. This includes True Negatives (TN) and False Negatives (FN), representing every instance where the model's prediction is '0'.

In the following example, we aimed to predict whether a customer will remain with the service (existing customer = 0) or churn (attrited customer = 1), using a dataset of 60 customers. Within this dataset, there are 14 churned (P=14) and 46 retained customers (N=46). Our model forecasted 8 customers to churn (PP=8) and 52 to stay (PN=52). Of these predictions, 5 were accurately identified as churned (TP=5) and 43 as retained (TN=43). However, 3 customers were incorrectly flagged as attrition when they were actually retained (FP=3), and 9 customers were mistakenly predicted as retained when they had churned (FN=9).

		Predicted Attrition	Predicted Existing	
		1	0	subtotal
Actual Attrition	1	5	9	14
Actual Existing	0	3	43	46
	subtotal	8	52	60

2.2 Accuracy

Accuracy is one of the most intuitive and commonly used metrics for evaluating the performance of a model in binary classification tasks. It measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. In mathematical terms, accuracy can be expressed as:

$$accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{Total\ Population}$$

For binary classification, where the outcomes are classified as either positive (1) or negative (0), accuracy answers the question: "Of all the predictions made, how many did the model get right?"

Accuracy is straightforward to understand and communicate to non-technical stakeholders, making it an appealing metric for initial model evaluation. It requires no advanced mathematical calculations beyond counting and basic division, facilitating quick assessments.

However, while accuracy is useful, it has several limitations, especially in cases where the dataset is imbalanced or when the cost of false positives and false negatives varies significantly.

1. Imbalanced Classes: In scenarios where the number of instances in each class significantly differs (e.g., 95% negatives and 5% positives), a model can achieve high accuracy by simply predicting the majority class every time. In such cases, accuracy can be misleading, not reflecting the model's ability to identify the minority class effectively.
2. Misleading Interpretations: Accuracy doesn't distinguish between the types of errors made. For tasks where the consequences of false positives and false negatives are drastically different, relying solely on accuracy can be deceptive. For example, in medical diagnosis, falsely predicting a disease when it's not present (false positive) and failing to detect a disease when it is present (false negative) can have very different implications.
3. Does Not Reflect Model Quality in Specific Applications: Accuracy does not account for the costs associated with different types of errors. In many real-world applications, the impact of false positives and false negatives is not equal, and accuracy does not provide insight into these nuances.

Our sample dataset also displays the challenge of imbalanced data, with 46 out of 60 customers being existing ones, accounting for 76.7% of the dataset. A simplistic model that labels every customer as an existing one would attain an accuracy of 76.7%. The accuracy of our model stands at 48 out of 60, or 80%, only marginally surpassing the naive benchmark. This suggests that our model may not possess strong predictive capabilities.

2.3 Precision

Precision is a critical performance metric for binary classification tasks, especially in contexts where the cost of false positives (incorrectly predicting the positive class) is high. It measures the accuracy of the positive predictions made by the model, that is, among all the instances classified as positive, how many were actually positive. Mathematically, precision is defined as:

$$precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

This metric is particularly important in situations where the goal is to minimize false positives. For example, in email spam detection, a high precision model would avoid incorrectly classifying important emails as spam (a false positive).

Precision is valuable when the consequences of false positives are more severe than false negatives. It helps in models where the priority is to ensure a high reliability of the positive predictions. In datasets where positive instances are rare (imbalanced datasets), precision becomes an important metric to assess the model's performance in identifying the positive class accurately.

Here are several drawbacks associated with precision, underscoring the need to complement it with other evaluation metrics.

1. **Does Not Consider False Negatives**: Precision focuses solely on the model's performance in predicting the positive class correctly and does not take into account false negatives (instances

that are positive but predicted as negative). This can be problematic in scenarios where missing out on positive instances carries a high cost.
2. **Not a Comprehensive Measure**: On its own, precision does not provide a complete picture of model performance. It needs to be considered alongside other metrics like recall (which measures the model's ability to identify all actual positives) to fully understand the model's effectiveness.

2.4 Recall

Recall, also known as sensitivity or true positive rate, is a crucial performance metric for binary classification tasks, particularly in scenarios where the cost of missing a positive instance (false negative) is high. Recall measures the proportion of actual positive instances that the model correctly identifies out of all actual positives. Mathematically, recall is defined as:

$$recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

This metric is especially important in fields such as medicine, fraud detection, and law enforcement, where failing to identify positive cases can have serious consequences.

Recall is valuable in contexts where it's crucial to identify as many positive instances as possible, even at the expense of accuracy or increasing false positives. For instance, in medical testing, it's often more acceptable to have false positives (which can be ruled out with further testing) than to miss a true positive case of a disease. In datasets with a small proportion of positive instances, recall helps to measure how effectively the model identifies this minority class, offering insight into its performance beyond what accuracy can provide.

Limitations of Recall

1. **Does Not Account for False Positives**: While recall focuses on minimizing false negatives, it does not penalize the model for false positives. In some applications, this can be a drawback, as excessive false positives could lead to inefficient use of resources or other negative outcomes.
2. **Not a Standalone Metric**: Because it doesn't consider the entire picture of model performance (especially how it handles negative instances), recall should not be used in isolation. It's important to balance it with other metrics, such as precision, to understand the trade-offs between identifying positive instances and avoiding false alarms.

The relationship between recall and precision is often inversely related; improving recall might decrease precision as the model becomes more liberal in predicting positives. Balancing these metrics depends on the specific costs and benefits associated with false positives and false negatives in a given application.

2.5 Specificity

Specificity, also known as the true negative rate, is a critical performance metric for binary classification tasks, particularly when the ability to correctly identify negative instances (true negatives) is as important or more so than identifying positive instances. Specificity measures the proportion of actual negatives that the model correctly identifies, providing insight into the model's accuracy for the negative class. Mathematically, specificity is defined as:

$$specificity = \frac{True\ Negative\ (TN)}{True\ Negative\ (TN) + False\ Positive\ (FP)}$$

This metric is especially vital in scenarios where false positives carry significant consequences or costs, or where the focus is on ensuring the negative instances are accurately identified.

Advantages of Specificity

1. **Focus on Negative Class**: Specificity is particularly useful in contexts where the negative class is of specific interest, and the cost of false positives is high. For example, in screening tests where the goal is to ensure that healthy individuals are not misdiagnosed.
2. **Valuable in Imbalanced Datasets**: In situations where positive instances are rare, specificity can provide additional insights into the model's performance on the more prevalent negative class, complementing metrics focused on the positive class like recall.

Limitations of Specificity

1. **Does Not Address False Negatives**: While specificity measures the model's ability to identify negative instances correctly, it does not take into account false negatives (positive instances incorrectly identified as negative). In applications where missing a positive instance is costly, specificity alone may not provide a complete picture of model performance.
2. **Not a Standalone Metric**: Relying solely on specificity can be misleading, especially if the positive class is of equal or greater importance. It is best used alongside other metrics like sensitivity (recall) to provide a balanced view of the model's performance across both classes.

In summary, specificity is an essential metric for evaluating a binary classification model's performance, especially regarding its precision in identifying negative instances. However, to achieve a comprehensive evaluation of a model's performance, specificity should be considered in conjunction with other metrics like sensitivity (recall), depending on the specific costs associated with false positives and false negatives in the application domain.

2.6 F1 Score

The F1 score is a widely used performance metric for binary classification tasks, especially when navigating the balance between precision and recall is crucial. It is the harmonic mean of precision and recall, providing a single metric that encapsulates both the model's ability to correctly identify positive instances (recall) and its ability to ensure that predictions labeled as positive are truly positive (precision). The F1 score is defined as:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

The F1 score harmoniously combines precision and recall, making it an excellent choice for situations where both false positives and false negatives carry significant costs. It ensures that a high F1 score can only be achieved by excelling in both metrics. In datasets where positive examples are much less common than negative ones, the F1 score provides a more informative measure than accuracy, as it focuses on the model's performance on the minority class. By combining precision and recall into a single metric, the F1 score simplifies the evaluation process, making it easier to compare the performance of different models or configurations.

Limitations of the F1 Score

1. **Equal Weight to Precision and Recall**: The standard F1 score treats precision and recall as equally important, which may not align with specific business objectives or costs associated with false positives and false negatives. In such cases, a weighted F1 score might be more appropriate.
2. **Not Intuitive**: Unlike accuracy, which directly relates to the proportion of correct predictions, the F1 score's meaning can be less intuitive to stakeholders unfamiliar with precision and recall.
3. **May Obscure Model Biases**: A high F1 score might mask biases in model performance across different groups or classes within the data, necessitating further analysis to ensure fair and equitable outcomes.

The F1 score is particularly useful in scenarios with a significant trade-off between precision and recall, such as in information retrieval or legal judgments. It encourages models to maintain a balance, ensuring neither precision nor recall is disproportionately favored at the expense of the other. Depending on the specific requirements of a task, variations of the F1 score, such as the weighted F1 score or F-beta score (which allows for different weighting between precision and recall), can provide flexibility in evaluation. While the F1 score is a powerful tool for model evaluation, it's often used alongside other metrics, like ROC-AUC or the confusion matrix, to gain a comprehensive understanding of model performance.

2.7 AUC-ROC

The Area Under the Receiver Operating Characteristic (AUC-ROC) curve is a performance measurement for binary classification problems at various threshold settings. The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the True Positive Rate (TPR, or recall) against the False Positive Rate (FPR, 1 - specificity) at various threshold settings.

Understanding AUC-ROC

- **ROC Curve**: The Receiver Operating Characteristic (ROC) curve shows the trade-off between sensitivity (or TPR) and specificity (1 - FPR). As the threshold for classifying a positive instance is lowered, the model identifies more true positives but also more false positives, affecting the curve's shape.
- **AUC**: The Area Under the ROC curve (AUC) measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). It provides an aggregate measure of performance across all possible classification thresholds. An AUC of 1 indicates a perfect model; an AUC of 0.5 suggests a model with no discriminative ability (equivalent to random guessing).

Advantages of AUC-ROC

1. **Threshold Invariance**: AUC-ROC provides a measure of model performance across all classification thresholds, making it useful for evaluating models where the optimal threshold is not known a priori.
2. **Imbalanced Classes Handling**: The AUC-ROC is less affected by imbalanced class distributions than other metrics like accuracy, making it particularly useful for evaluating models on imbalanced datasets.
3. **Comparative Measure**: AUC allows for the comparison between different models with a single scalar, regardless of the threshold settings, making it easier to identify the best model among alternatives.

Limitations of AUC-ROC

1. **Not Threshold-specific**: While threshold invariance is an advantage, it can also be a drawback. AUC-ROC does not provide insight into the best threshold for balancing sensitivity and specificity, which can be crucial for operationalizing models.
2. **May Be Misleading in Highly Imbalanced Data**: In extremely imbalanced scenarios, even models with poor performance can achieve a relatively high AUC-ROC, necessitating the use of complementary metrics such as precision-recall curves.
3. **Doesn't Reflect the Magnitude of Errors**: AUC-ROC considers only the order of predictions and actual values, not the magnitude of errors. This can be limiting in problems where the costs of false positives and false negatives are significantly different.

In summary, the AUC-ROC is a powerful tool for evaluating binary classification models, providing insights into their performance across various thresholds and conditions. However, it's most informative when used in conjunction with other metrics and domain-specific considerations to ensure a comprehensive understanding of model effectiveness.

2.7.1 Example

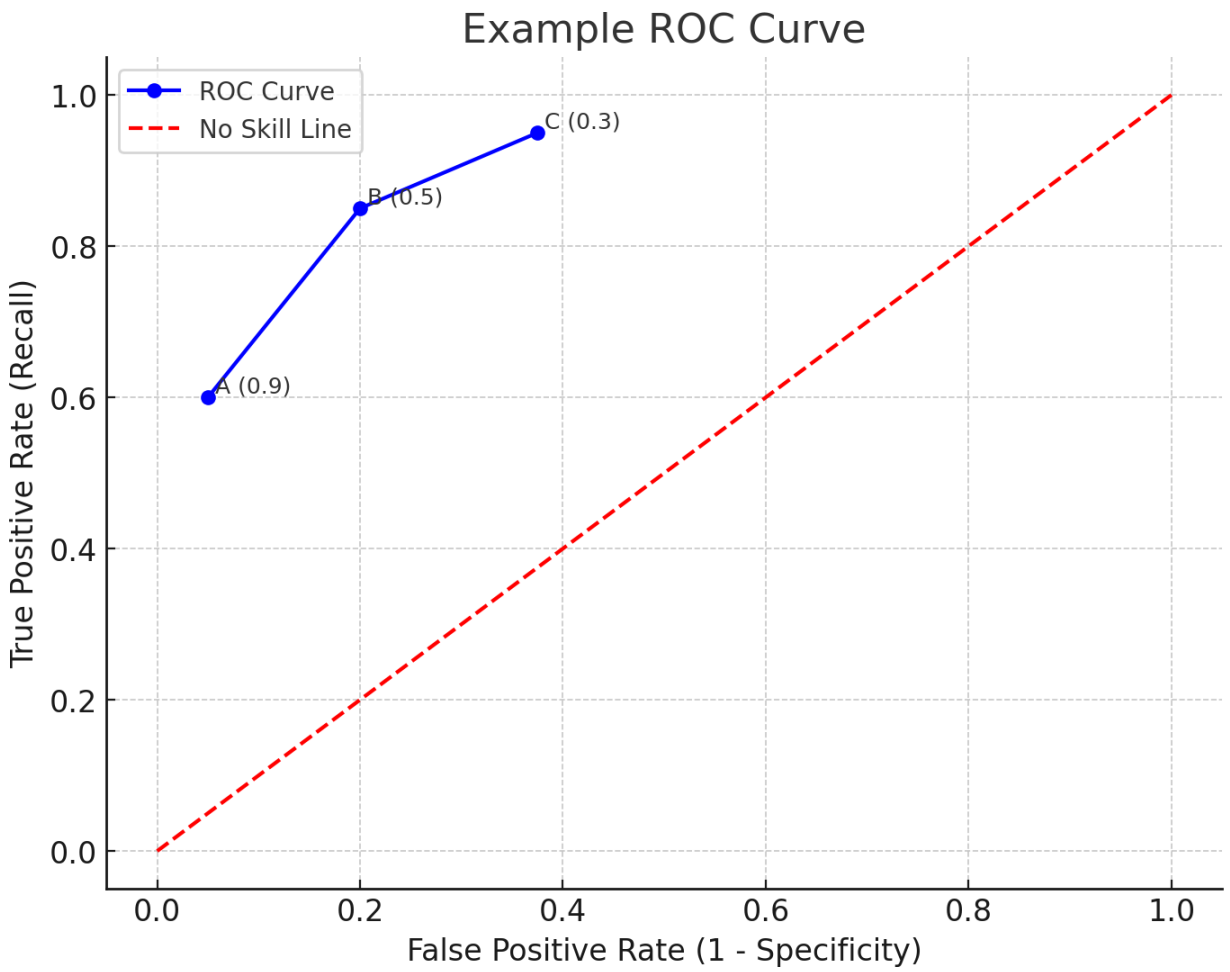
let's consider a binary classification problem where we are building a model to predict whether an email is spam (positive class) or not spam (negative class). The dataset consists of 100 emails, with 20 labeled as spam and 80 as not spam.

After training, our model assigns a probability score to each email, indicating the likelihood that it's spam. Based on these scores, we can plot the ROC curve by calculating the True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold levels.

Here's a simplified example showing how to calculate TPR and FPR for three different thresholds:

- **Threshold A (0.9)**: The model is very conservative, labeling only emails it's very confident about as spam.
    - TPR (Recall): 0.6 (12 out of 20 spam emails are correctly identified)
    - FPR: 0.05 (4 out of 80 non-spam emails are incorrectly labeled as spam)
- **Threshold B (0.5)**: The model adopts a moderate stance, labeling emails as spam more liberally.
    - TPR: 0.85 (17 out of 20 spam emails are correctly identified)
    - FPR: 0.2 (16 out of 80 non-spam emails are incorrectly labeled as spam)
- **Threshold C (0.3)**: The model is very liberal, favoring the identification of as many spam emails as possible.
    - TPR: 0.95 (19 out of 20 spam emails are correctly identified)
    - FPR: 0.375 (30 out of 80 non-spam emails are incorrectly labeled as spam)

Plotting these points on a graph with FPR on the X-axis and TPR on the Y-axis, and connecting them, gives us the ROC curve. The area under this curve (AUC) represents the model's ability to discriminate between spam and not spam across all thresholds. A perfect model would have an AUC of 1.0, directly rising along the Y-axis and then moving horizontally at TPR = 1. This means it correctly identifies all spam emails without misclassifying any non-spam emails, regardless of the threshold. In real-world scenarios, the AUC typically falls between 0.5 (no discrimination) and 1.0 (perfect discrimination).



This curve plots the True Positive Rate (Recall) against the False Positive Rate for different thresholds labeled A (0.9), B (0.5), and C (0.3). The 'No Skill Line' represents a model with no discriminative ability, essentially random guessing. The points on the ROC curve indicate how the balance between sensitivity (recall) and specificity changes as the threshold for classifying an email as spam varies. The area under this curve (AUC) would quantify the model's overall ability to discriminate between spam and not spam across all possible thresholds.

While the above example simplifies the process (usually, many more threshold points are used), it helps illustrate how different thresholds impact the model's true positive and false positive rates, and consequently, its performance. The ROC curve visualizes this performance, and the AUC provides a single measure summarizing the model's effectiveness at distinguishing between classes across all thresholds.