***Question:*** What are the calculated values for accuracy, precision, recall, and F1-score?

Its code in jupyter file

Accuracy: 96.35

Precision: 98.80

Recall: 73.66

F1-Score: 84.40

What do these metrics tell you about your model's performance?

0 = No Spam, 1 = Spam

Actual Values

Predicted Values

0

1

0

1

|  |  |
| --- | --- |
| TP  1446 | FP  2 |
| FN  59 | TN  165 |

Accuracy =

Precision =

Recall = = =

F1 Score =

When FP and FN both are important

then  **= 1** it is known as **harmonic mean**

When FP more important than FN then  **= 0.5**

When FN more important than FP then **= 2**

**Accuracy:**

* **Definition**: Accuracy measures the proportion of correctly classified instances (both true positives and true negatives) out of the total number of instances.
* An accuracy score of, for example, 96.35 means that the model correctly predicted 96.35% of the emails' categories in the test set.

**Precision:**

* **Definition**: Precision measures the proportion of true positive predictions (correctly predicted positives) out of all positive predictions made by the model.
* A precision score of 98.80 indicates that when the model predicted an email belonging to a specific category, it was correct 98.80% of the time. Precision is important when minimizing false positives is crucial.

**Recall:**

* **Definition**: Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset.
* A recall score of 73.66 suggests that the model correctly identified 73.66% of all emails that actually belonged to a specific category. Recall is important when it's critical to capture all positive instances and minimize false negatives.

**F1-Score:**

* **Definition**: F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.
* An F1-score of 84.80 indicates a good balance between precision and recall. It's particularly useful when there's an uneven class distribution or when both precision and recall are important.

**1. Confusion Matrix**

**Task**: Create a confusion matrix for your classification model on the test set.

Its code in jupyter file

Confusion Matrix:

 [

    [1446   2 ]

    [ 59   165]

 ]

The confusion matrix would look like this:

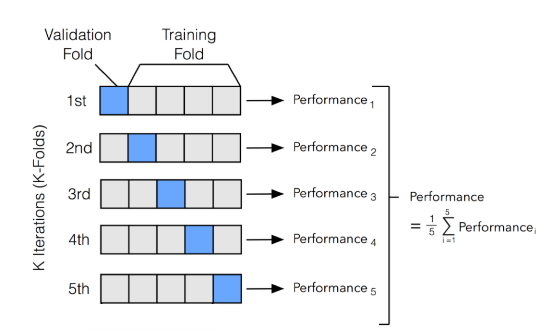
Actual Values

Predicted Values

|  |  |
| --- | --- |
| TP  ( 1446 ) | FP  ( 2 ) |
| FN  ( 59 ) | TN  ( 165 ) |

* True Positive (TP): 1446 emails
* True Negative (TN): 165 emails
* False Positive (FP): 2 patients
* False Negative (FN): 59 patients

These values help in calculating various performance metrics for the classification model, such as accuracy, precision, recall, and F1-score.



**Help in understanding the Model Performance:**

* The confusion matrix assesses model performance by revealing correct and incorrect predictions.
* It offers insights into which classes are confused with each other, aiding error analysis.
* Precision and recall, derived from the matrix, highlight trade-offs in model performance.
* This matrix guides improvements by identifying where the model makes mistakes, focusing on reducing false positives or negatives.
* By analyzing the confusion matrix, we can gain a deeper understanding of your model's strengths and weaknesses, guiding further iterations or adjustments to enhance its performance.

**3. ROC/AUC Calculation**

**ROC Curve**

The ROC curve is a graphical representation of a model’s ability to distinguish between classes. It plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 — Specificity) for different classification thresholds. Here’s what these terms mean:

* True Positive (TP): The model correctly predicts the positive class.
* False Positive (FP): The model incorrectly predicts the positive class when it’s actually negative.
* True Negative (TN): The model correctly predicts the negative class.
* False Negative (FN): The model incorrectly predicts the negative class when it’s actually positive.
* Sensitivity (True Positive Rate): The proportion of actual positive samples correctly predicted.
* Specificity (True Negative Rate): The proportion of actual negative samples correctly predicted.

**Area Under the Curve (AUC):**

AUC quantifies the overall performance of a classification model. It represents the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. In simpler terms, it measures the model’s ability to distinguish between positive and negative classes.\

AUC ranges from 0 to 1, where 0.5 indicates a random classification, and 1 signifies a perfect classifier.

**Importance of ROC Curve and AUC:**

**Robustness to Class Imbalance**: ROC curves are less affected by imbalanced datasets. They provide an unbiased evaluation even when the classes are not evenly distributed.

**Threshold Agnostic**: ROC curves consider all possible classification thresholds, providing a comprehensive view of the model’s performance.

**Comparative Analysis**: AUC allows for easy comparison between different models. The model with the higher AUC is generally preferred.

**4. Importance of Cross-Validation:**

**Improving Model Accuracy**

Cross-validation allows us to estimate the true performance of our models by testing them on independent data subsets. This evaluation helps identify and address issues such as overfitting or underfitting, resulting in improved model accuracy and generalization capabilities.

**Preventing Overfitting and Underfitting**

Overfitting occurs when a model performs exceptionally well on the training data but fails to generalize on unseen data. Similarly, underfitting happens when a model fails to capture the underlying patterns from the training data. Cross-validation helps combat these issues by providing insights into whether our models are suffering from overfitting or underfitting, allowing us to make informed adjustments accordingly.