# Evaluation metrics for classification models

## Why do we need evaluation metrics for classification models?

- In machine learning on broad aspect, we have a problem statement to address, then we fetch data related to it, perform analysis and feature engineering. Then use the data to train the model which we finally use to do the prediction.
- Thus, the output of trained machine learning model is consumed by end users for making their decisions.
- In our cybersecurity use case, the impact of the models becomes extremely critical because of the nature of outcome helps to make important decisions about benign and malicious events or type of malicious events.
- In order to use machine learning models in real world scenario, we need to address the fundamental questions such as: -
    1. Why should the end user trust the trained model?
    2. How does our model perform relative to the other models trained by others?
- To address the above fundamental questions, we need to define the governance and framework of evaluation of models which help us understand the given model's performance and also compare them on reliable and useful metrics with other models, which finally allows the end users to make decisions on determining the quality of output produced by the given model and describe the same in detail.
- In terms of building structure for evaluation of classification models, we need to perform seven major tasks: -

    1. List the metrics that can be used for the use case.

    2. Define each metric in detail and explain its benefits and limitations (if any).

    3. Document the evaluation results of all previous models observed from literature survey.

    4. Compute the performance of our model based on each metric defined in task 2.

    5. Quantitatively document the comparison of performance of our model with previously trained models observed in literature survey.

    6. Describe the performance of our model with respect to previously trained model using the data documented in task 6. We need to compute the gap between performance of our model with respect to other models for all available metrics.

    7. Derive the inferences based on task 5 and task 6, explain reason for the same. If our model performs better than previously trained models, we need to explain the reasons for achieving better results. Similarly, if our model performs worse than previously trained models, we need to identify the gaps that we need to work on to reach that performance.

- Robust documentation of the above tasks will enable us define the performance of our models which will provide clarity about its application and also convey the same to end users.
- Additionally, in real world scenarios, the evaluate and decisions to adopt machine learning solutions are taken by different stakeholders. Thus, the specific details in evaluation metrics along with relevant context and research will build the ability of our project to articulate well for different audiences.

## Task 1: List of metrics for evaluation of classification models (both binary and multi-class)

1. Confusion Matrix
2. Accuracy
3. Precision
4. Recall
5. F1-Score
6. ROC curve
7. AUC score
8. Balanced accuracy
9. Matthews Correlation Coefficient (MCC)
10. Negative predictive value
11. False discovery rate
12. Cohen kappa metric
13. Precision – Recall curve

## Task 2: Definition and details about each metric: -

1. Confusion matrix: -
    - It is used to consolidate data to measure and evaluate performance of classification model.
    - In binary classification, we have 2X2 matrix.
    - In multi-class classification, we have matrix of size same as number of classes in the target feature.

    - Representation of Confusion Matrix for Binary classifier: -

| | | **Actual values** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Predicted values** | **Positive** | True Positive | False Positive |
| | **Negative** | False Negative | True Negative |

- Representation of Confusion Matrix for Multi-class classifier: -
  Assuming 3 classes: Class 1, Class 2, Class 3.

| | | **Actual values** | | |
|---|---|---|---|---|
| | | **Class 1** | **Class 2** | **Class 3** |
| | **Class 1** | Cell 1 | Cell 2 | Cell 3 |

| | | | | |
|---|---|---|---|---|
| **Predicted values** | **Class 2** | Cell 4 | Cell 5 | Cell 6 |
| | **Class 3** | Cell 7 | Cell 8 | Cell 9 |

  - True Positive: {Cell 1}, {Cell 5}, {Cell 9}
  - False Positive: {Cell 2 + Cell 3}, {Cell 4 + Cell 6}, {Cell 7 + Cell 8}
  - True Negative: {Cell 5 + Cell 6 + Cell 8 + Cell 9}, {Cell 1 + Cell 3 + Cell 7 + Cell 9}, {Cell 1 + Cell 2 + Cell 4 + Cell 5}
  - False Negative: {Cell 4 + Cell 7}, {Cell 2 + Cell 8}, {Cell 3 + Cell 6}

- True Positive: The number of records model correctly predicts the positive outcome.
- True Negative: The number of records model correctly predicts the negative outcome.
- False Positive: The number of records model incorrectly predicts the positive outcome.
- False Negative: The number of records model incorrectly predicts the negative outcome.
- Examples of above four components in the domain of cyber security: -
  - True Positive: The model correctly predicts a malicious event as an attack.
  - True Negative: The model correctly predicts a normal event as normal.
  - False Positive: The model incorrectly predicts a normal event as an attack.
  - False Negative: The model incorrectly predicts an attack event as normal.

- Confusion matrix is useful in Binary classification due to compact nature of the structure and complex in multi-class classification due to a greater number of classes to be incorporated in the matrix its dimensions will have higher order and interpreting the results will become difficult.
- Confusion matrix can also give incorrect or misleading representation of a model’s performance in the datasets having an imbalanced nature of target classes. This is because if the target class is heavily skewed in one direction, and thus the model may showcase high accuracy by predicting everything in the favour of dominant class while failing to detect the rare class.
- For example: In a network dataset, we have 1000 events, 995 are benign and 5 are malicious. Thus, the dataset is highly imbalanced and skewed against malicious events. Now if the classification model predicts everything as benign then the confusion matrix may portray the model has high accuracy but in reality, it failed to correctly detect malicious events which was more critical for evaluating performance of the classification model.

2. Accuracy: -
   - It gives overall correctness of the model.
   - Accuracy = (True Positive + True Negative)/(True Positive + True Negative + False Positive + False Negative)
   - It helps us understand how often the model predicts correctly.

- It is useful in scenarios when the dataset is balanced that is the target feature has balanced representation of all classes.
- It fails to justify false negative in imbalanced dataset.

3. Precision: -
    - It gives accuracy of positive predictions.
    - Precision = True Positive/(True Positive + False Positive)
    - It emphasizes on positive predictions made by the model.
    - It is better than accuracy while working on imbalanced datasets since it demands minimization of False Positives to have higher score.

4. Recall: -
    - It is also called Sensitivity.
    - It gives model's ability to find all positive cases.
    - Recall = (True Positive)/(True Positive + False Negative)
    - In our cybersecurity use-case, if we have 10 malicious events, how many events were successfully classified as malicious by the model will give the value of recall.
    - Thus, if a model has high recall, it means it mostly identifies malicious events correctly.
        - classified as benign. Then recall=2/10 = 0.2 (Scenario 4.3)
    - It works well on imbalanced datasets.

5. F1-Score: -
    - It takes both precision and recall as inputs to give the output.
    - F1-Score=2*(Precision * Recall)/(Precision + Recall)
    - In binary classification, if F1-Score is close to 1, then the model has high accuracy and recall, which indicates model has good performance.
    - It is useful when we work on imbalanced dataset.
    - It assumes that both precision and recall have equal importance, however, it does not align with our cybersecurity use case. This is because, misclassification of malicious event as benign is a bigger problem than misclassification of benign event as malicious.

Examples: -

Let us consider there are 20 events, 16 are benign and 4 are malicious. Thus, the unknown dataset for the classifier is imbalanced.

Case 1: - The classifier predicts all 20 events as Benign.

| | | **Actual values** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Predicted values** | **Positive** | True Positive = 0 | False Positive = 0 |
| | **Negative** | False Negative = 4 | True Negative = 16 |

Accuracy = 0.8

Precision = Not defined ~ 0

Recall = 0

F1-Score = 0

Case 2: - The classifier predicts all 20 events as Malicious.

| | | **Actual values** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Predicted values** | **Positive** | True Positive = 4 | False Positive = 16 |
| | **Negative** | False Negative = 0 | True Negative = 0 |

Accuracy = 0.2

Precision = 0.2

Recall = 1

F1-Score = 0.33

Case 3: - The classifier predicts all 4 Malicious events as Malicious. And it incorrectly predicts 3 Benign events as Malicious.

| | | **Actual values** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Predicted values** | **Positive** | True Positive = 4 | False Positive = 3 |
| | **Negative** | False Negative = 0 | True Negative = 13 |

Accuracy = 0.85

Precision = 0.57

Recall = 1

F1-Score = 0.72

Case 4: - The classifier incorrectly predicts 2 malicious events as Benign.

| | | **Actual values** |
|---|---|---|

| | | **Positive** | **Negative** |
|---|---|---|---|
| **Predicted values** | **Positive** | True Positive = 2 | False Positive = 0 |
| | **Negative** | False Negative = 2 | True Negative = 16 |

Accuracy = 0.9

Precision = 1

Recall = 0.5

F1-Score = 0.67

6. ROC curve: -
   - ROC: Reverse Operating Characteristics
   - Here, we plot true positive rate (on y axis) and false positive rate (on x axis).
   - The area under the curve is used to measure the model's performance.
   - True Positive Rate = True Positive / (True Positive + False Negative)
   - False Positive Rate = False Positive / (True Negative + False Positive)

7. AUC score: -
   - AUC: Area Under the Curve
   - It is used to for binary classifier and used to differentiate among the classes.
   - It is computed using ROC curve.

8. Balanced accuracy: -
   - Balanced accuracy = (True Positive Rate + True Negative Rate)/2
   - True Positive Rate = True Positive / (True Positive + False Negative)
   - True Negative Rate = True Negative / (True Negative + False Positive)
   - It is useful when classes of the dataset are imbalanced.
   - Example: -
     - Case 1: -
       - True Positive Rate = 0
       - True Negative Rate = 1
       - Balanced accuracy = 0.5
     - Case 2: -
       - True Positive Rate = 1
       - True Negative Rate = 0
       - Balanced accuracy = 0.5
     - Case 3: -
       - True Positive Rate = 1
       - True Negative Rate = 0.8125
       - Balanced accuracy = 0.90625
     - Case 4: -
       - True Positive Rate = 0.5

    - True Negative Rate = 1
    - Balanced accuracy = 0.75
- If the score is closer to 1, then the model has higher performance.
- It ranges between 0 to 1.

9. Matthews Correlation Coefficient (MCC): -
    - It ranges between -1 to +1.
    - MCC = (True Negative * True Positive) – (False Negative * False Positive) / sqrt((True Positive + False Positive)* (True Positive + False Negative) * (True Negative + False Positive) * (True Negative + False Negative))
    - If MCC= 0, the classifier performs random classification.
    - It is used for binary class and multi-class classification.

10. Negative predictive value: -
    - It tells how likely the event is not malicious if it is classified as benign.
    - Negative predictive value = True Negative / (True Negative + False Negative)
    - Example: -
        - Case 1: - NPV = 16 / (16 + 4) = 0.8
        - Case 2: - NPV = 0 / (0 + 0) = Not defined ~ 0
        - Case 3: - NPV = 13 / (13 + 0) = 1
        - Case 4: - NPV = 16 / (16 + 2) = 0.89

11. False discovery rate: -
    - False Discovery Rate = False Positive / (True Positive + False Positive)
    - Here we determine out of all the events that the model classified as malicious; how many were incorrect.
    - Thus, this helps us determine the noise generated by the model.
    - Example: -
        - Case 1: - FDR = 0 / (0 + 0) = Not defined ~ 0
        - Case 2: - FDR = 16 / (16 + 4) = 0.8
        - Case 3: - FDR = 3 / (3 + 4) = 0.428
        - Case 4: - FDR = 0 / (0 + 2) = 0
    - We can also use it to for feature selection, to determine features that are associated with malicious events.

12. Cohen kappa metric: -
    - It takes into account that model may correctly classify the some of the events purely by chance.
    - It is also called Kappa Score.
    - k=(p0-pe)/(1-pe)
      p0 : Relative measured among models
      pe : Hypothetical probability of chance agreement
    - k=1: There is complete agreement between the models.
    - k<=0: There is no agreement between the models.
    - Example: -

| **Case** | **p0** | **pe** | **k** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | 0.8 | 0.8 | 0 |
| 2 | 0.2 | 0.2 | 0 |
| 3 | 0.85 | 0.59 | 0.63 |
| 4 | 0.9 | 0.74 | 0.615 |

- Following are the interpretations of Cohen's kappa: -

| **Cohen's kappa** | **Interpretation** |
|---|---|
| 0 | No agreement |
| 0.10 - 0.20 | Slight agreement |
| 0.21 - 0.40 | Fair agreement |
| 0.41 - 0.60 | Moderate agreement |
| 0.61 - 0.80 | Substantial agreement |
| 0.81 - 0.99 | Near perfect agreement |
| 1 | Perfect agreement |

- As per our 4 cases: -
  Case 1 and case 2 have no agreement. Thus, the two models are far away from expected model.
  Case 3 and case 4 have substantial agreement. Thus, the two models are closer to the expected model.

13. Precision – Recall curve: -

- It is useful for imbalanced dataset.
- In our cyber security use case, instead of predicting the binary classifier classes: malicious and benign directly, we predict the probability of an event being malicious.
- Then we plot the graph with Recall on x-axis and Precision on y-axis.
- The area under Precision Recall curve is the indicator of performance. Larger the area, better the model performs.