https://doi.org/10.48047/AFJBS.6.15.2024.10984-10997

AFJBS

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Confusion Matrix-Based Performance Evaluation Metrics

Sathyanarayanan Swaminathan[1], B. Roopashri Tantri[2]

1: Sri Sathya Sai University for Human Excellence,
Navanihala Grama, Kamalapur Taluk, Gulbarga district,
Karnataka, India – 585313.
Email: sathyanarayanan.brn@gmail.com
2: Sindhi Institute of Management, Kempapura, Hebbal, Bengaluru,
Karnataka, India – 560024
Email: roopatantri@gmail.com

Volume 6, Issue 15, Sep 2024

Received: 15 July 2024

Accepted: 25 Aug 2024

Published: 25 Sep 2024

doi: 10.48047/AFJBS.6.15.2024.10984-10997

Abstract: Confusion matrices offer an insightful and detailed technique for evaluating classifier performance, which is essential in data science. This paper presents a comprehensive insight into the confusion matrix and its vital role in evaluating machine learning classification models. The fundamental concepts underlying the confusion matrix and its components are examined. Furthermore, the role of the confusion matrix in determining critical performance indicators, such as accuracy, precision, recall, sensitivity, and specificity, as well as false positive rate and F1 score are discussed. The significance of more sophisticated measures for assessing classifier performance, such as the ROC, AUC, and precision-recall curves, is also discussed. The study also highlights other significant metrics, such as G-mean, Cohen's Kappa, prevalence, null error rate, markedness, average precision, and balanced accuracy, outlining their special uses and relevance. These metrics help in making informed choices regarding how to optimise and fine-tune classification models for problems with real-world data.

Keywords: Confusion matrix, Accuracy, Precision, F1 score, ROC curve, precision recall curve.

Introduction

Herein, we intend to provide a background on the confusion matrix, its structure in detail, and its use in evaluating the performance of classifiers.

Confusion matrix: Karl Pearson created the confusion matrix in 1904, when it was first known as a contingency table. It was later referred to as a classification matrix, before being referred to as a confusion matrix in data science. This name should have remained “classification matrix”, which would have been more accurate and eliminated a lot of confusion! The word "confusion" refers to confusion that can occur on a specific metric to be prioritized while trying to improve the model, although several metrics can be obtained from the confusion matrix. The confusion matrix is a square matrix of size N × N, where N denotes the number of output classes. Each row of the matrix represents the number of instances of a predicted class and each column represents the number of instances of the actual class. This provides a class-by-class breakdown of the number of accurate and inaccurate predictions made by a classifier for classification tasks. Categorization can be binary or multiclass. The confusion matrix reveals the classifier's performance, what it is getting right, and the many kinds of mistakes the classifier could make. The metrics derived from the confusion matrix help choose the best course of action to enhance the performance of the model. Because confusion matrices can be constructed for datasets with known target/output values, they are used in supervised learning methods.

The classification algorithm determines whether a particular activity has occurred. For example, whether a person has a specific ailment, whether a person is likely to buy an item, whether an email is spam, or any other factor. The target variable has two potential values: positive and negative. The actual target labels in the dataset used for testing and the predicted labels provided as outputs by the ML model are two key elements.

Consider testing a heart disease detection system. It is positive if the system flags a person with heart disease. If the system flags a person as not having heart disease, the result is negative. Table 1 presents the general structure of the confusion matrix.

Table 1: Confusion Matrix

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	True Positive (TP)	False Positive (Type I error)
	Negative	False Negative (Type II error)	True Negative

There are two types of predictions: correct and incorrect (errors).

True Positive (TP): Both the actual values and the prediction are positive.
False Positive (FP): Although the prediction is positive, the actual value is negative. This is called a "Type I error".

True Negative (TN): The actual value is negative, and the prediction is negative.
False Negative (FN): Although predicted to be negative, the sample is positive. This is also called "Type II error".

The first part of TP, TN, FP, and FN, the terms True or False, concerns whether the prediction is correct. The model's prediction of whether the sample or data point is positive or negative is the second part of the TP, TN, FP, and FN.

Related work: Researchers have used ML approaches in different sectors for performance measurements. Some of these issues are highlighted below: Pushpa et al. [1] conducted an elaborate study on diagnosing diseases such as cancer, diabetes, epilepsy, heart attack, and other diseases. Practical implementations of algorithms such as SVM, KNN, Decision tree, and Naïve Bayes were carried out using Python. This study also highlights the importance of accuracy, precision, recall, and F1-score metrics in the diagnosis of diseases and the impact of machine learning (ML) on the healthcare industry. Kevin et al. [2] proposed confusion matrix-based evaluation measures for binary classification problems and used them in classification applications. Steven et al [3] studied ML approaches used in gastroenterology and performance metrics in the context of binary classification. A web-based tool was used to calculate the metrics.

Amalia et al [4] compared the behaviour of a balanced AC1 score against other performance measures based on a binary confusion matrix. Ahmed Fawzy Gad [5] evaluated deep learning (DL) models based on confusion matrix, accuracy, precision and recall. Alqahtani et al [6] made use of ML and DL models to predict cardiovascular disease for a person. Six classification algorithms were employed. The prediction model yielded an accuracy of 88.7%. Sathyanarayanan et al [7] analysed heart sounds using an efficient ML model that required the least computational resources and computation time. The model exhibits an accuracy of more than 99%. Sathyanarayanan et al [8] made an elaborate study on ML in healthcare. Mahmoud Fahmy Amin [9] presented a step-by-step procedure for obtaining a confusion matrix in binary classification problems. Mohammadreza et al [10] developed a multi-label confusion matrix and applied it to two multi-label datasets, viz, a 12-lead ECG dataset with 9 classes and a movie poster dataset with 18 classes. A comparison was made between the multi-label confusion matrix and other well-known methods to prove its effectiveness. Peter Flach [11] made an elaborate study on performance evaluation in ML and highlighted the need for adopting some good practices in classifier evaluation by developing a proper measurement theory. Arie Ben-David [12] obtained the relationship between ROC curves and Cohen's Kappa by deriving a mathematical formulation that links ROC spaces with the Kappa statistic. The importance of understanding the relationship between ROC and Kappa is also highlighted. Using various examples, Igor Baskin and Alexandre Varnek [13] explained two-class classification models for SAR (Structure-Activity Relationships) analysis. Sathyanarayanan et al [14] used the techniques of artificial intelligence to detect cardiovascular disease. A Customized deep learning architecture (SAINET) was used for automated cardiovascular disease through heart sound analysis. The method provided an accuracy of over 99% for precision, recall, specificity, and the F1 score.

1. Performance measurements

A confusion matrix is used to evaluate several metrics for the suitability and accuracy of the model. Several metrics are described in this work.

Assume that a binary classifier has a yes or no output, as shown in Table 2.

Table 2: Confusion Matrix for Binary Classifier

		Actual Values		**Total**
		Positive	**Negative**	
Predicted Values	**Positive**	TP=25	FP=3	28
	Negative	FN=5	TN=67	72
Total		30	70	100

1.1 Accuracy

Accuracy is the fraction of correct predictions among all predictions or how often the prediction is correct [15].

Accuracy = (Number of correctly classified instances) / (Total number of instances)
Accuracy = (TP + TN) / (TP + FP + TN + FN)

For the data presented in Table 2:

Accuracy = (25 + 67) / (25 + 3 + 67 + 5) = 0.92
Classification accuracy = (correct predictions / total predictions) * 100

The rate of the error is calculated by subtracting the classification accuracy from one. Hence, Error rate = 1 – Classification accuracy

Accuracy is particularly useful when the classes in the dataset are balanced, that is, they have approximately equal numbers of instances for each class. However, the classification accuracy has limitations, especially when dealing with imbalanced datasets, where instances of one class may significantly outnumber the instances of other classes. In such cases, the model may achieve a high accuracy by simply predicting the majority class. However, their performance in minority classes may be poor. Accuracy can be misleading when the class distribution in the dataset is skewed and an accurate prediction of the minority class is critical. For example, considering a cancer prediction system, if 2% of the population has cancer, the system will be 98% accurate, even if it yields a negative for all input cases. The fact that the accuracy of a model is not a good metric for classification when using predictive models is called the accuracy paradox.

1.2 Precision

If there are more than two output classes, the classification accuracy does not provide information on the classes that are predicted accurately. In such cases, a more suitable measure would be precision. Precision is the fraction of correctly predicted positive results.

Precision = TP / (TP + FP)

For the data in Table 2, Precision = 25 / (25 + 3) = 0.89
This metric is useful in situations where FPs outnumber the FNs. A high-precision value suggests that the model has a low rate of false positives, meaning that it makes accurate positive predictions. However, a low precision value indicates a high number of false positives, indicating that the model makes more incorrect positive predictions.

Precision does not correctly evaluate the performance with respect to negative cases. It also does not consider false negatives (Type II errors), which are truly positive instances but are incorrectly predicted as negative. When false negatives need to be minimized or when both false positives and false negatives are equally important, other metrics, such as recall (sensitivity) or the F1-score, which balance precision and recall, can be considered.

1.3 Other metrics

Recall: This measures the proportion of actual positives predicted correctly or how accurately the model predicts positive cases. It is calculated as
Recall = TP / (TP + FN).

Recall is particularly valuable when false negatives (Type II errors) need to be minimized. For example, in a medical diagnostic setting, recall indicates the proportion of actual positive cases (e.g. patients with a specific disease) that the model correctly identified, which is important for avoiding missing potentially critical diagnoses. A high recall value suggests that the model has a low rate of false negatives, meaning that it effectively captures most positive instances. On the other hand, a low recall value indicates a higher number of false negatives, signifying that the model is missing positive instances.

Sensitivity: This is the same as recall and is called the total positive rate (TPR).
Hence, for data in Table 2, Recall = Sensitivity = TPR = 25 / (25 + 5) = 0.83

Specificity: This measures the fraction of negatives correctly predicted and shows how well the model predicts negative results. Specificity is complementary to sensitivity, and is also called the total negative rate (TNR).

Specificity = TN / (FP + TN)

For the data in Table 2, Specificity = 67 / (3 + 67) = 0.95
Specificity focuses on the ability of the model to correctly identify all negative instances, and is particularly valuable when false positives (Type I errors) need to be minimized. For example, in a diagnostic test for a specific non-threatening condition, specificity indicates the proportion of actual negative cases (e.g. healthy individuals) that the model identified correctly, which is crucial for avoiding unnecessary interventions or treatments for healthy individuals. A high specificity value suggests that the model has a low rate of false positives, meaning that it effectively captures most negative instances. However, a low specificity value indicates a higher number of false positives, signifying that the model misclassifies negative instances as positive instances.

False Positive Rate: False Positive rate (FPR) identifies the proportion of negatives classified as positive.

FPR = FP / (FP + TN)

For the data in Table 2, FPR = 3 / (3 + 67) = 0.04
Misclassification rate: This is also called the “error rate”, and shows how wrong the model is.

Misclassification rate = (FP + FN) / (FP + FN + TP + TN)

For the data in Table 2, misclassification rate = (3 + 5) / (3 + 5 + 25 + 67) = 0.08
A steep precision value is important for some models, whereas recall is crucial for others, particularly when labelling a minority class as positive. For example, a cancer-detection model should not label a patient with cancer as negative; hence, high recall is important in this case.

2. Choice between metrics

Precision-recall trade-off: Precision focuses on predicted values, whereas recall focuses on actual values. Increasing recall decreases precision and increasing precision decreases recall. This is called the precision-recall trade-off. The choice between precision and recall depends on the specific application. For example, if a model is built to detect fraud, it prioritizes precision because there is no need to flag too many legitimate transactions as fraudulent. If a model is built to diagnose a disease, recall is preferred because it is important to detect the likelihood of the disease.

For example, a model for predicting whether a patient has cancer has a precision of 90% and recall of 80%, which means that 90% of the patients that the model predicts as having cancer have cancer, and 80% of the patients who have cancer were correctly diagnosed by the model. If the precision of the model is increased to 95%, it is necessary to make it more conservative in its predictions. The model is less likely to predict whether a patient has cancer, even if the patient has cancer. Consequently, the recall of the model decreases to 70%. Hence, if the precision of the model increases, the recall decreases. This tradeoff can be addressed in several ways. Some of them are:

- Understanding specific applications and deciding whether precision or recall is more important.
- Experiment with different thresholds to see the effects of thresholds on precision and recall.
- Use a metric that combines precision and recall, such as the F1-score.

3. F1-score and ROC curve

4.1 F1-score: The F1-score, which combines precision and recall, is important for models in which both are equally important. This is a harmonic mean of precision and recall that considers both false negatives and false positives. The harmonic mean is computed by dividing the number of values in a data series by the sum of the reciprocals of each value in the data series. The harmonic mean is always less than or equal to the arithmetic mean and the geometric mean because the harmonic mean gives more weight to the smaller values in the data series. When the precision and recall are equal, the harmonic mean is the average. However, when they are different, the harmonic mean is closer to the smaller value. Consequently, it is better suited for data with class imbalances than the accuracy metric. It is at its maximum when the precision is equal to recall. It balances Type 1 and Type 2 errors. The understandability of the model is inadequate when using the F1-score because it is not clear whether the classifier attempts to maximize precision or recall. However, it can be combined with other measures to provide a more thorough analysis of outcomes. The F1-score is effective for classification models where both FP and FN impact the model identically and TN is high. The F1-score is computed as

F1-score = (2 * precision * recall) / (precision + recall)
For the data in Table 2, F1-score = (2 * 0.89 * 0.83) / (0.89 + 0.83) = 0.85

If a binary classification model gives a precision and recall of zero and one, respectively, then the harmonic mean is zero, whereas the arithmetic mean is 0.5. In this case, the harmonic mean provides the correct picture. The closer the F1-score is to one, the better the classifier.
Table 3 provides a consolidated list of the various metrics.

Table 3: Various metrics and their values

Metric	Value
Accuracy	0.92
Precision	0.89
Recall	0.83
Sensitivity	0.83
Specificity	0.95
F1-score	0.85

4.2 ROC Curve: The ROC curve, also called the receiver operating characteristic curve [16], is a plot of two parameters, TPR and TNR. It is also known as the relative operating characteristic curve, because it compares the two operating characteristics, TPR and FPR. It was first used for signal detection by radar but is now being used in many fields, including medicine and ML. The performance of the model at all classification thresholds is shown by plotting the FPR and TPR values for each threshold. The different threshold values represent the classification boundaries of the classifier. Decreasing the threshold classifier increases the number of positives and decreases the number of negatives, thereby increasing both FP and TP. Increasing the threshold classifier decreases both FP and TP. The ROC is important for evaluating a model because it evaluates all thresholds for classification compared to the accuracy metric, which is calculated for one threshold.

The ROC curve can help choose a threshold for a classifier to maximize the true positives and minimize the true negatives. An ideal plot of the ROC curve has a TPR of 100% with zero FPs. The ROC curve is better suited for balanced datasets. Figure 1 shows the receiver operating characteristic (ROC) curve of the sample.

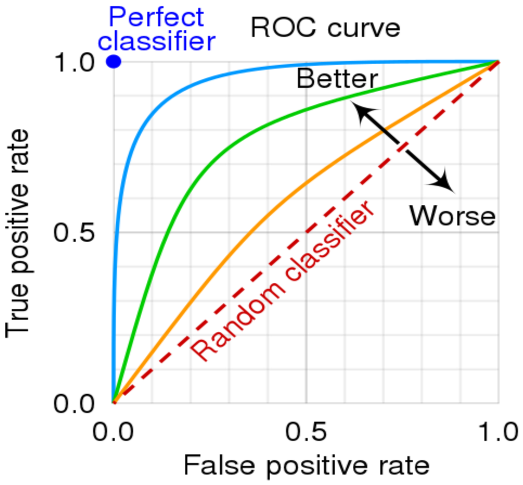

Figure 1: Sample ROC curve [17]

Because computing various points on the curve is inefficient, another metric called the AUC is considered.

4.3 AUC Curve: AUC stands for "Area under the ROC curve" and is a commonly used metric for model evaluation [18]. It is used for binary classification and measures the entire area below the ROC curve from (0,0) to (1,1). It provides an aggregate measure of the model's performance across all classification thresholds.

The ROC AUC score can be calculated in various ways; however, the trapezoidal rule is frequently used. This entails splitting the ROC curve into trapezoids with vertical lines at the FPR values and horizontal lines at the TPR values to roughly estimate the AUC. The area is then calculated by adding the trapezoidal areas.

The AUC value falls between 0 and 1. The AUC is equal to one for a fully correct model, which is then a perfect classifier. The entirely incorrect model has an AUC of 0. A higher AUC value indicates that the model is better at distinguishing between the two classes. The advantage of the AUC is that it is scale-invariant and classification threshold-invariant, as it can predict the quality of the model irrespective of the value of the classification threshold. An AUC value of 0.5 shows that the model is of no use in distinguishing between the two classes, and such models are said to be random classifiers or "no skills" classifiers. Figure 2 shows different ACU classification metrics.

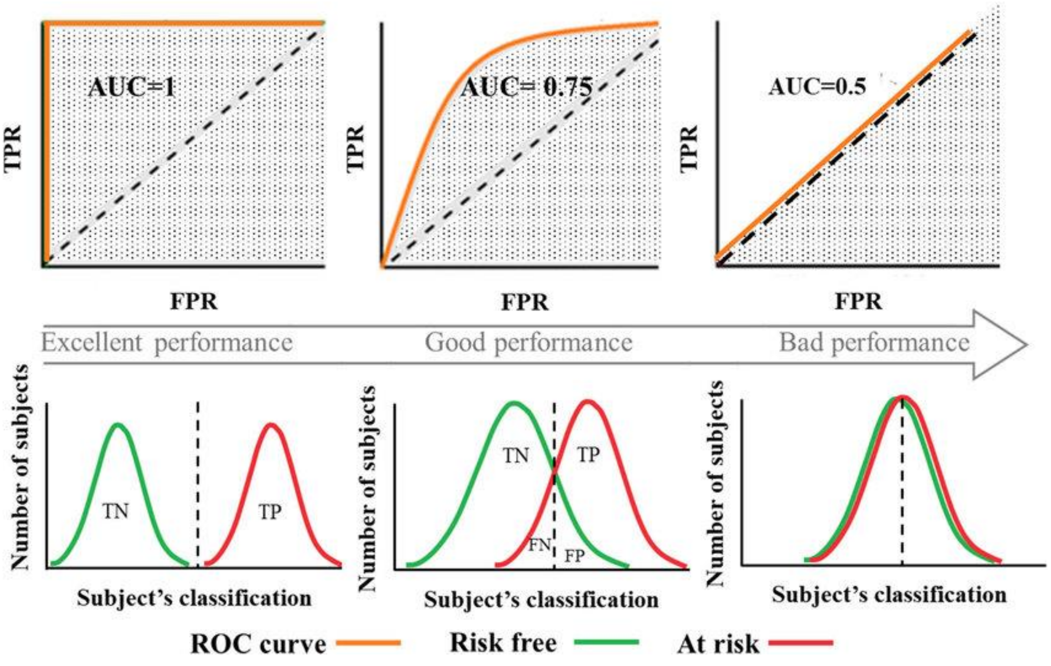

Figure 2: AUC classification metric [19]

However, when minimizing one type of classification error (either a false positive or false negative), the AUC metric is not very useful. The AUC is not intuitive; hence, it is difficult to interpret.

Different types of errors and their effects are not considered in the ROC AUC analysis. False negatives are frequently more expensive than false positives and vice versa. In this situation, attempting to balance recall and precision, and establishing a proper categorization threshold to reduce a particular error is frequently a more effective course of action. ROC AUC is not useful when performing this type of optimisation. When there is a severe class imbalance, AUC can also be deceptive.

Precision-recall curve: The curve of recall versus precision is called the precision-recall curve. This could be an alternative to the ROC curve with unbalanced data. A classifier that produces a curve close to the top-right corner is considered good. Figure 3 shows the sample precision-recall curve.

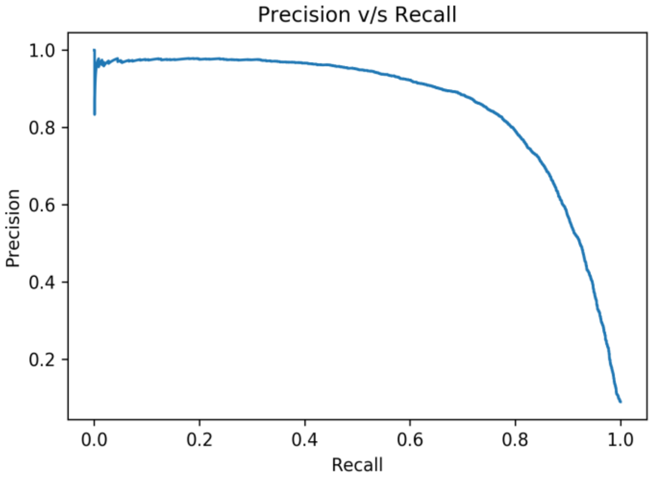

Figure 3: Precision recall curve

4. Multiclass classification

The multiclass classification model has more than two classes and no negative classes. The metrics are calculated for each class separately in a similar manner to the binary class after computing TP, FP, TN, and FN for the particular class.

Example: Dataset considered- iris.
A confusion matrix for the iris dataset is presented in Table 4.

Table 4: Confusion Matrix for iris dataset

		Actual values			
		Setosa	Versicolor	Virginica	**Row total**
Predicted Values	Setosa	10	0	0	10
	Versicolor	0	2	6	8
	Virginica	0	10	2	12
	Column total	10	12	8	30

The values of TP, FP, TN, and FN were calculated for the setosa and non-setosa classes.
TP = 10 (predicted values are the same as the actual values)
TN = 20 (sum of all columns and rows except for the row, and the column involving the class for which we are calculating the TN)
FP = 0 (Total of the values of the column except TP)
FN = 0 (Total of the values of the row except for TP)

TP, FP, TN, and FN were computed for the remaining two species in a similar manner. Precision, recall, and other metrics were individually calculated for each class. The ROC curves are plotted for multiclass classification by converting them into one vs. all to make it a binary classification, and the plot for each of the class is generated independently. A few other metrics are occasionally employed.

Cohen's Kappa: When classes are unbalanced or the model is being assessed on a dataset with fewer samples, Cohen's Kappa is a helpful metric for assessing the effectiveness of a ML classification model. After accounting for agreement that may be expected by chance, Cohen's kappa calculates the degree of agreement between predictions made by a classifier and labels in the real world. A Kappa score of 0 indicates that the classifier performs no better than random guessing, while a score of 1 indicates perfect agreement and -1 indicates complete disagreement. Because it accounts for agreement that may occur by chance, Cohen's kappa is a more reliable indicator of agreement than accuracy for model evaluation in which datasets are imbalanced.

Prevalence: The number of positive samples in the data is called prevalence and is calculated as

Prevalence = (TP + FN) / (TP + TN + FP + FN).

Null Error Rate: Null error rate is the percentage of times the classifier is incorrect if it predicts only the dominant class every time.

Geometric mean: A metric frequently employed for model evaluation is the G-mean (geometric mean of specificity and sensitivity), particularly when the data are unbalanced. The proportion of correctly classified positive examples (sensitivity) and correctly classified negative cases (specificity) in a model are both considered by the G-mean. This provides a more reliable performance indicator than the accuracy, which can be deceptive in unbalanced datasets. The square root of the product of sensitivity and specificity yields the G-mean and is calculated as

G-mean = sqrt (Sensitivity * Specificity)

The model is ideal if the G-mean is 1, with a sensitivity and specificity of 1. The model is useless when the G-mean is 0, with a sensitivity and specificity of 0. It is a useful metric, as it gives equal importance to sensitivity and specificity and forces the model to focus on both positive and negative classes.

G-mean is a helpful indicator for assessing models in several domains, such as medical diagnosis, fraud detection, and predicting customer churn. Although it is a useful metric when the data are imbalanced, it must be used in combination with other metrics, as it is not a perfect metric. Furthermore, they cannot be used in multiclass models.

Other metrics that indicate the probability that a prediction is informed versus chance are:

Informedness: Informedness **(**Youden's J statistic) is the difference between sensitivity and specificity and ranges from -1 to +1. It measures how well the model can steer clear of both false positives and false negatives.

Markedness: Markedness is the difference between the precision and false positive rates. It measures the ability of a model to avoid false positives.

Balanced Accuracy: Balanced accuracy is the average of sensitivity and specificity, and provides a balanced view of the model's performance, particularly in imbalanced datasets.

Average Precision (AP): Average precision is the average of the precision values calculated for different thresholds in a precision-recall curve. It is commonly used in information retrieval and multi-label classification.

5. Results and Discussion

The metrics considered in the previous sections are discussed through a case study with a binary classifier, as explained below.

Dataset considered: SUV [20]
Metadata: The dataset gives information about whether the SUV is purchased or not based on the attributes ‘Age’, ‘Gender’ and ‘Estimated salary’. This dataset has 400 observations, and the variable” purchased’ is a binary categorical variable indicating ‘0’ if SUV is not purchased and ‘1’ if the SUV is purchased. Since the variable ‘Purchased’ is a dependent variable, which is binary, one of the most suitable models in this situation for predicting such a binary variable is logistic regression.

The dataset was divided into training and testing sets in a ratio of 70:30, producing 278 observations in the training set and 122 observations in the testing set. Considering a predicted value of more than 0.6 to be a binary value ‘1’ and using R Studio environment, the logistic regression model output is shown in Table 5.

Table 5: Model output: Logistic Regression

Coefficients:	**Estimate**	**Std. Error**	**z value**	**Pr(>|z|)**
(Intercept)	-1.091e+01	1.356e+00	-8.048	8.41e-16 ***
GenderMale	2.810e-01	3.630e-01	0.774	0.439
Age	2.051e-01	2.703e-02	7.587	3.26e-14 ***
Salary	2.653e-05	5.544e-06	4.786	1.71e-06 ***
Signif. codes:	0 ‘***’ 0.001	‘**’ 0.01	‘*’ 0.05	‘.’ 0.1 ‘ ’ 1

Interpretation of the model output

- The intercept and the variables 'age' and 'salary' are significant.
- Furthermore, the McFadden value of the model is 0.4358, which indicates that the predictor model is a good fit.

The resulting confusion matrix is listed in Table 6.

Table 6: Confusion Matrix for SUV dataset

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	25	1
	Negative	24	72

The various metrics considered in the previous sections were obtained, as listed in Table 7.

Table 7: Metrics based on confusion matrix

Metric	**Value**
Accuracy	0.7951
Precision	0.9615
Recall	0.8621
Sensitivity	0.8621
Specificity	0.9863
False positive rate	0.0137
Misclassification rate	0.2049
Classification accuracy	0.7951
F1-score	0.9091
G-Mean	0.9221
Cohen's kappa	0.5381
Prevalence	0.4016
Markedness	0.9478
Balanced accuracy	0.9242
Informedness	0.4965
Null error rate	0.2049

Observations from metric values

- The logistic regression model that fits the given data had an accuracy of 79.51%. This value is influenced by the selection of the threshold value (0.6 in this case). In this case, model accuracy can be increased by decreasing the threshold value.
- The precision of this model was 0.9615, indicating that the model correctly predicted 96.15% of the positive results.
- The recall value was 0.8621, indicating that 86.21% of the actual values were correctly predicted by the model.
- The specificity was 0.9863, indicating that 98.63% of the negative results were correctly predicted.
- The false positive rate was 0.0137, which means that only 1.37% of the negative values were classified as positive. Hence, the chance of incorrect classification was only approximately 1%.
- The misclassification rate was 0.2049, indicating that the probability of incorrectly predicting the model was approximately 20%. This supports the claim that the accuracy of the model is approximately 80%.
- The classification accuracy was 0.7951, implying that 79.51% of the observations were classified correctly.

- The F1-score is 0.9091, which is very close to '1'. This implies that false positive and false negative values have an equal impact on the model. In addition, the TN value was very high (72).
- G-Mean is 0.9221, which is very close to '1'. This indicates that there is a reason to consider the model a good predictor model.
- Cohen's kappa is 0.5381, which falls in the range 0.41 to 0.6, indicating moderate agreement with classifications.
- The prevalence was 0.4016, indicating that the data had 40% of the positive class.
- The markedness is 0.9478, which means that the model is 94.78% capable of avoiding false positives. Thus, the model has a highly reliable prediction.
- Balanced accuracy, which is based on sensitivity and specificity, yielded a value of 0.9242, This high score indicates that the model not only has a good true positive rate, but also a strong true negative rate and hence the prediction of the model can be trusted across both classes.
- The Informedness was 0.4965, indicating moderate classification by the model. In addition, it was observed from the confusion matrix that out of 122 instances, 25 instances were correctly positively classified, and 72 were correctly negatively classified.
- Error rate is 0.2049, which means that the chances of the predictor model being wrong is 20.49%

Inference:
By observing the values of the above metrics, it can be inferred that the logistic regression model developed for the given dataset produces accurate predictions. However, the choice of metric depends on the problem of interest and objectives of the study. The choice varies in different scenarios for the same problem. Here, since the model is built to predict whether to purchase an SUV depending on age and salary, any of the metrics can be used. However, since the objective is to predict whether to purchase the SUV, recall or specificity may be preferred. A good classification accuracy value indicates that the classification obtained by the logistic regression model is good. Furthermore, the F1 score is very close to one. The closer the F1 score is to 1, the better the classification results. In addition, because the G-mean is close to 1, the logistic regression model used in this study is ideal. Cohen's Kappa value falls in a moderate level of agreement with the classification, which implies that the classification model is effective. In addition, a high value of markedness indicates the strength of the model in avoiding false positives. A very good balance accuracy value supports the claim that the logistic regression model has balanced performance. The low error rate of the model ensures that the predictions of the model are very low. Hence, the model's predictions were almost 80% accurate.

Overall, it can be inferred that the model yields the best results for binary classification when considering any of the above metrics.

6. Conclusion

Every ML practitioner must thoroughly understand the confusion matrix and its accompanying performance metrics. Different parts of the confusion matrix and their usage in measuring various aspects of classifier performance are discussed in detail. Each metric, from the often-used accuracy, precision, recall, sensitivity, specificity, and false positive rate to the F1-score, which balances precision and recall, offers distinct insights into the advantages and disadvantages of the model. Furthermore, the complexities of the ROC curve and AUC were explored, providing a comprehensive understanding of the model's ability to differentiate between classes. In addition, dealing with imbalanced datasets has been made much easier owing to the precision-recall curve and average precision. Less well-known but essential measures that offer a new viewpoint on the evaluation process, including the G-mean, Cohen's kappa, prevalence, null error rate, markedness, and balanced correctness, have also been discussed. Having information about these various metrics, one can make well-informed choices when optimizing and fine-tuning the categorization models, so that real-world difficulties can be easily dealt with. Using the strengths of the confusion matrix and its measurements will enhance ML efforts and lead to predictions that are more accurate and trustworthy across a variety of domains.

Acknowledgements: The authors thank Sri Sathya Sai University for Human Excellence and Sindhi Institute of Management for enabling this work.

References:

1. Pushpa Singh, Narendra Singh, Krishna Kant Singh and Akansha Singh. "Diagnosing of disease using machine learning", Machine Learning and the Internet of Medical Things in Healthcare, 2021, pp., 89-111 (https://doi.org/10.1016/B978-0-12-821229-5.00003-3).
2. Kevin Riehl, Michael Neunteufel, Martin Hemberg, Hierarchical confusion matrix for classification performance evaluation, *Journal of the Royal Statistical Society Series C: Applied Statistics*, Volume 72, Issue 5, November 2023, Pages 1394–1412 (https://doi.org/10.1093/jrsssc/qlad057)
3. Hicks SA, Strümke I, Thambawita V, Hammou M, Riegler MA, Halvorsen P, Parasa S. On evaluation metrics for medical applications of artificial intelligence. Sci Rep. 2022 Apr 8;12(1):5979. doi: 10.1038/s41598-022-09954-8. PMID: 35395867; PMCID: PMC8993826.
4. Vanacore, A., Pellegrino, M.S. & Ciardiello, A. Fair evaluation of classifier predictive performance based on binary confusion matrix. Comput Stat 39, 363–383 (2024). (https://doi.org/10.1007/s00180-022-01301-9)
5. Ahmed Fawzy Gad, "Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision and Recall", Digital ocean, August 2024. (https://www.digitalocean.com/community/tutorials/deep-learning-metrics-precision-recall-accuracy).
6. Alqahtani A, Alsubai S, Sha M, Vilcekova L, Javed T. Cardiovascular Disease Detection using Ensemble Learning. Comput Intell Neurosci. 2022 Aug 16;2022:5267498. doi: 10.1155/2022/5267498. PMID: 36017452; PMCID: PMC9398727.
7. Swaminathan Sathyanarayanan, Krishnamurthy Srikanta Murthy, Chandrashekar & Gudada, Satish Kumar Mallappa and Neeraj Ali, "Heart Sound Analysis with Machine Learning Using Audio Features for Detecting Heart Diseases", International Journal of Computer Information Systems and Industrial Management Applications, Vol. 16 (2024), 131-147.
8. Swaminathan Sathyanarayanan & Chitnis Sanjay. (2022). A Survey of Machine Learning in Healthcare. 10.1201/9781003241409-1.
9. Amin, Mahmoud. (2022). Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial. Journal of Engineering Research. 6. 10.21608/erjeng.2022.274526.
10. M. Heydarian, T. E. Doyle and R. Samavi, "MLCM: Multi-label Confusion Matrix," in *IEEE Access*, vol. 10, pp. 19083-19095, 2022, doi: 10.1109/ACCESS.2022.3151048.
11. Flach, P. (2019). Performance Evaluation in Machine Learning: The Good, the Bad, the Ugly, and the Way Forward. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9808-9814. (https://doi.org/10.1609/aaai.v33i01.33019808).
12. Arie Ben-David, About the relationship between ROC curves and Cohen's kappa, Engineering Applications of Artificial Intelligence, Volume 21, Issue 6, 2008, Pages 874-882, ISSN 0952-1976. (https://doi.org/10.1016/j.engappai.2007.09.009).
13. https://www.coursehero.com/file/10213742/classification-tutorial/
14. S. Sathyanarayanan and K. Srikanta Murthy. (2024). Heart Sound Analysis Using SAINet Incorporating CNN and Transfer Learning for Detecting Heart Diseases, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA),* volume: 15, number: 2 (June), pp. 152-169. DOI: 10.58346/JOWUA.2024.I2.011.
15. https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-

performance-measures-you-can-use/

16. https://en.wikipedia.org/wiki/Receiver_operating_characteristic
17. https://commons.wikimedia.org/wiki/File:Roc_curve.svg
18. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
19. https://www.researchgate.net/figure/llustration-of-AUC-classification-metric_fig5_343326638
20. https://www.kaggle.com/datasets/amizhanand/suv-dataset