Complete Guide to Machine Learning Evaluation Metrics

Shashwat Tiwari · Follow

Published in Analytics Vidhya · 12 min read · Oct 20, 2019

193 3

Photo by Tim Swaan on Unsplash

Hello All,

Building Machine learning Model is based on the principle of continuous feedback. The Machine learning Models are built and model performance is evaluated further Models are improved continuously and continue until you achieve a desirable accuracy. Model Evaluation metrics are used to explain the performance of metrics. Model Performance metrics aim to discriminate among the model results.

Making a Machine learning model and carrying out prediction is a simple task. Your end goal is to create a model that gives high accuracy on out of sample data. Hence, It is important to check performance metrics before carrying out predictions.

In AI Industry we have different kinds of metrics in order to evaluate machine learning models. Beside all these Evaluation metrics cross-validation popular and plays an important role in evaluating machine learning models.

> ***Basic Machine learning Warmups***

When we are talking about the classification probelm there are always two types of an algorithm we deal -

- Some Algorithm like SVM & KNN generates a class or label output. However, in a binary classification problem, the output generated is either 0 or 1. But due to advancement in machine learning, we have some algorithm which can convert the class output to probabilities.
- Logistic Regression, Random Forest and Gradient Boosting etc are some algorithms that generate probability outputs. Converting probability outputs to class output is just a matter of creating a threshold probability.

Moreover, when we are dealing with a regression problem, the output value is continuous so it does not require any further operation.

> So Let’s Talk about Evaluation metrics

> *Machine learning Evaluation metrics*
>
> **Evaluation metrics for classification,Regression & Clustering**

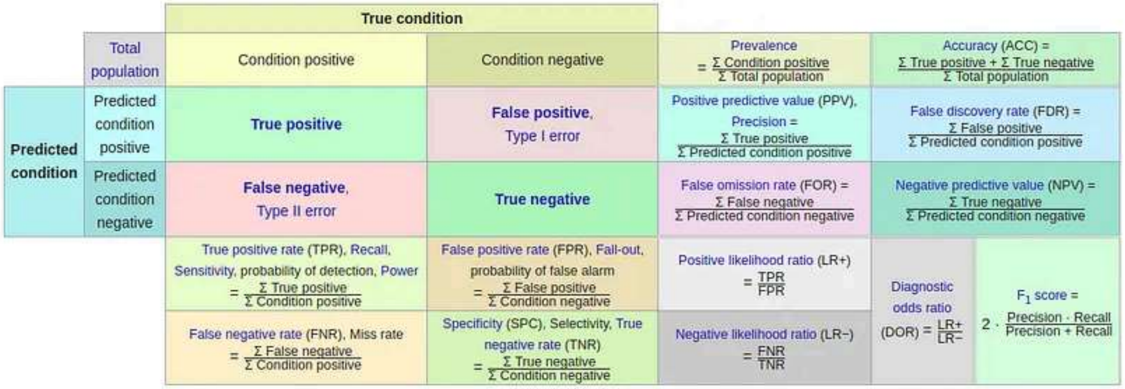
1 — For Classification

1. *Confusion Matrix*

Beginning with the laymen definition of the confusion matrix

A confusion matrix is a table that outlines different predictions and test results and contrasts them with real-world values. Confusion matrices are used in statistics, data mining, machine learning models and other artificial intelligence (AI) applications. A confusion matrix can also be called an error matrix.

Mostly *confusion matrix* is used for in-depth analysis of statistical data efficiently and faster analysis by using data visualization.



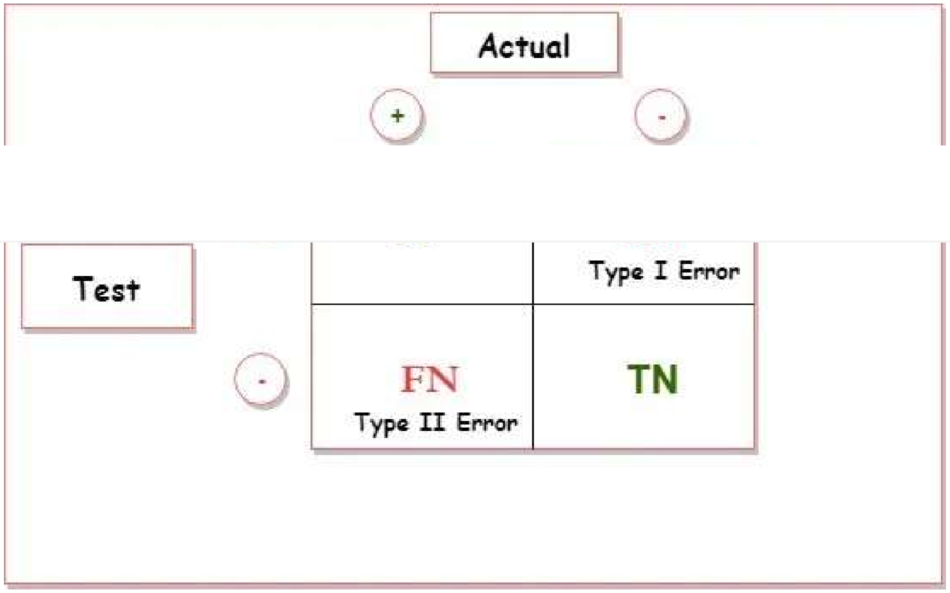
		True condition				
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$		

source:https://en.wikipedia.org/wiki/Confusion_matrix

Above confusion matrix seems a bit confusing, There are some terms which you need to remember for confusion matrix:

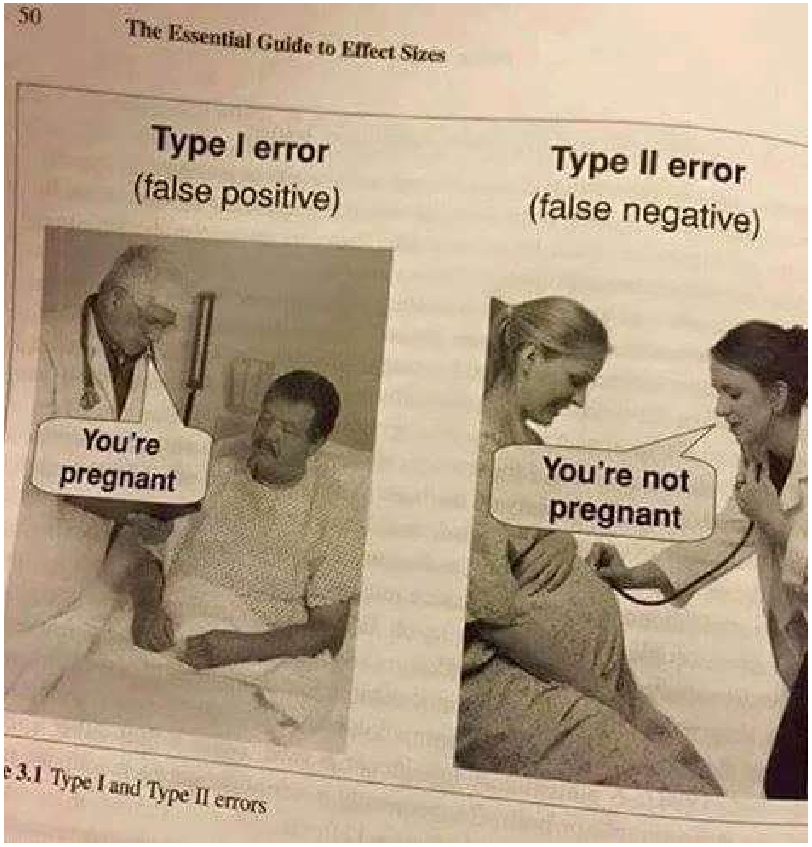
- **Accuracy:** the proportion of the total number of predictions that were correct.
- **Positive Predictive Value or Precision:** the proportion of positive cases that were correctly identified.
- **Negative Predictive Value:** the proportion of negative cases that were correctly identified.
- **Sensitivity or Recall:** the proportion of actual positive cases that are correctly identified.
- **Specificity:** the proportion of actual negative cases that are correctly identified.

Let's understand some concepts with the help of an example. We will take example of kidney diseases.



- **True Positive:** The person has Kidney diseases and they actually have the disease.
- **True Negative:** The person not suffering from Kidney diseases and **actually** doesn't have the disease.
- **False Positives (FP):** Person has kidney disease & they actually don't have the disease. (Also known as a "Type I error.")
- **False Negatives (FN):** Person does not have the Kidney disease & they actually have the Kidney disease. (Also known as a "Type II error.").

One of great illustration of Type I and Type II error which I explored is -



e 3.1 Type I and Type II errors

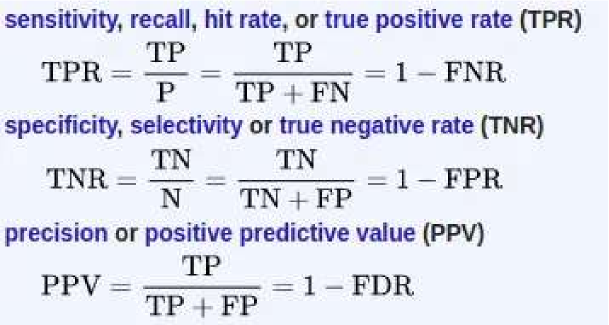
Few other points related to confusion matrix are -

- High recall & low precision represents most of the positive prediction are correctly recognized means we have very less false negative however there is a significant increase in false positive.
- Low recall & high precision represents that we miss a lot of positive examples which inturns have high false-negative but those we predict as positive are for sure positive predictions.

- Every column of the confusion matrix represents predicted class instances.
- Every row of the matrix represents actual class instances.
- A high precision score gives more confidence to the model's capability to classify 1's. Combining this with Recall gives an idea of how many of the total 1's it was able to cover.
- Confusion matrix does not only provide us errors made by our classification model but also the types of errors we made.

2. Recall,Sensitivity & Specificity

Starting with Sensitivity, it calculates the ratio of positive class correctly detected. This metric gives how good the model is to recognize a positive class.



sensitivity, recall, hit rate, or true positive rate (TPR)

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = 1 - \mathrm{FNR}$$

specificity, selectivity or true negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} = 1 - \mathrm{FPR}$$

precision or positive predictive value (PPV)

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} = 1 - \mathrm{FDR}$$

Precision, however, shows the accuracy of positive class it computes how likely positive class prediction is correct.

On the other hand, Specificity is characterized as the ratio of actual negatives, which model predicted as a negative class or true negative. Hence we can conclude that there will be another proportion of actual negative, which got predicted as positive and could be termed as false positives.

Generally, when we are dealing with the above-defined metrics. In the case of health care organizations, they will be more concerned with a minimal wrong positive diagnosis. They will be more focused on high specificity. On the other hand, another predictive model will be more concerned with Sensitivity.

3. F1 Score

In some cases, data scientists and machine learning engineers try to obtain the best precision and recall simultaneously.*F1 Score* is the harmonic mean for precision and recall values. The formula for F1 score goes this way-

$$F_1 = \left(\frac{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}{2}\right)^{-1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

So why there is a need of taking Harmonic mean for recall & precision values instead of Geometric mean or Arithmetic mean? The answer is simple and straight Harmonic mean punished the most extreme values. There are situations however for which a data scientist would like to give a percentage more importance/weight to either precision or recall.

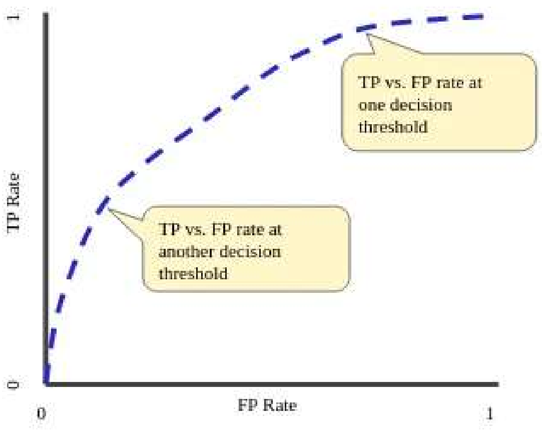
The higher the F1 score more is the predictive power of the classification model. A score close to 1 means a perfect model, however, score close to 0 shows decrement in the model's predictive capability.

4. AUC-ROC(Area under ROC curve)

A ***ROC curve (receiver operating characteristic curve)*** is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate- *Number of True positive divided by the sum of the number of True positive and the number of false negatives. It describes how good the model is at predicting the positive class when the actual outcome is positive.*
- False Positive Rate- *The number of false positives divided by the sum of the number of false positives and the number of true negatives.*

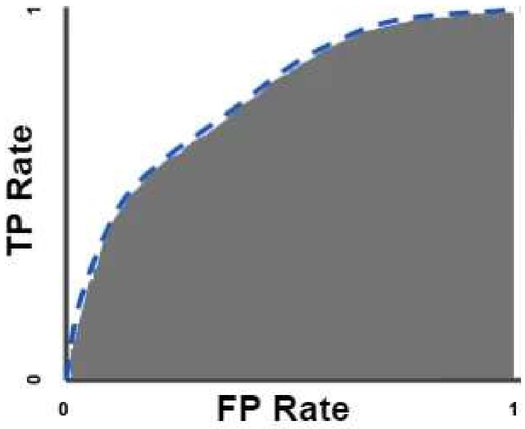
The ROC curves plot the graph between True positive rate and false-positive rate. These plots are generated at different classification thresholds.So if we have a low classification threshold then we can able to classify more items as positive thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.



Source: https://developers.google.com/machine-learning/crash-course/

The points in the ROC curve can be calculated by evaluating a supervised machine learning model like logistic regression with, but this would be inefficient. The solution to this problem is the sorting based algorithm known as AUC.

AUC is an acronym for Area under the curve. It computes the entire 2D area under the ROC curve.



Source: https://developers.google.com/machine-learning/crash-course/

In a more intuitive way, it is a plot of FPR(False positive rate) on the x-axis and TPR(True positive rate) on the y-axis for different threshold ranging from 0.0 to 1.

The AUC ROC Plot is one of the most popular metrics used for determining machine learning model predictive capabilities. Below are some reasons for using AUC ROC plot-

- The Different Machine learning models curves can be checked with different thresholds.
- Model predictive capability is summarized by the area under the curve(AUC).
- AUC is considered to be scaled variant, it measures the rank of predictions rather than its absolute values

- AUC always focuses on the quality of the Model’s skills on prediction irrespective of what threshold has been chosen.

5. Logarithmic Loss

AUC ROC curve determines the model’s performance by taking the predicted probabilities at various thresholds. There are some concerns over the AUC ROC curve as it accounts for the order of probabilities, not the model’s capability to predict positive data points with higher probability.

In this case, *Log loss* came into the picture, Logarithmic Loss or Log loss works by penalizing the false classification. However Mathematically it is nothing but a negative average of the log of corrected predicted probabilities for each instance.Log loss mostly suits the multi-class classification problem.Log loss takes the probabilities for all classes present in the sample. If we minimize the log loss for a particular classifier we get better performance metrics.

The math formula is given below

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

Here,

- y_ij, indicates whether sample i belongs to class j or not
- p_ij, indicates the probability of sample i belonging to class j

Log loss has a range [0, ∞). Moreover, it has no upper bound limit. We can interpret log loss as it is nearer to 0 have higher accuracy whereas if log loss moves away from 0 indicates lower accuracy.

2 — For Regression

1. *Root Mean Squared Error*

Root mean squared error is the most popular metrics used in Regression problems.RMSE is defined by the standard deviation of prediction errors. These prediction errors are some times called Residuals. Residuals are basically the measurement of the distance of data points from the Regression line.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Where:

- Σ = summation (“add up”)
- (yi— yj)Sup>2 = differences, squared
- N = sample size

Putting it in a simple way RMSE tells us how well the concentration of data points around the regression line. With RMSE it is assumed residuals are unbiased and follow a normal distribution. Below are some interesting points related to Root mean squared error.

- RMSE works efficiently when we are dealing with a large volume of data points. Hence error reconstruction becomes more reliable.
- As per RMSE mathematical formula the “square root” shows a large number deviation.
- Before using RMSE be sure that there are no outliers in the dataset because RMSE is heavily influenced by outliers.
- Root mean squared error has higher weightage and it also penalizes errors as compared to other evaluation metrics.

2. Mean Absolute Error

The Average taken between the original values and predicted values is called *Mean Absolute Error*. It also measures the average magnitude of error i.e.how far the predictions from the actual output. Moreover, MAE does not provide us any direction of error i.e. whether we are overfitting the data or underfitting the data.

$$MeanAbsoluteError = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j|$$

3. Mean Squared Error

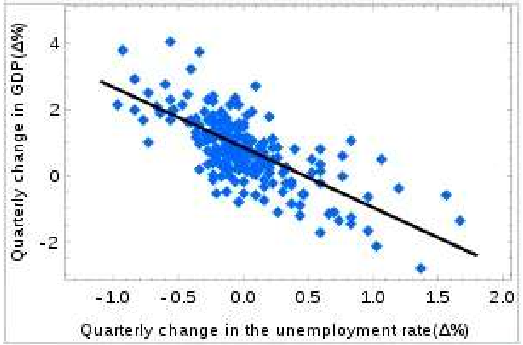
There is a minor difference between MSE and MAE. Deviation comes in MSE takes the average of the **square** of the difference between the original values and the predicted values. In MSE computation of gradient becomes easier than MAE which requires computational tools in order to compute gradients.

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

Mean Squared Error

Mean Squared Error is good to use when the target column is normally distributed around the mean value. Mean squared error comes into the picture when outliers are present in our dataset and it becomes necessary to penalize them.

4. R Squared/Adjusted R Squared



source:https://blog.minitab.com/blog/adventures-in-statistics-2/multiple-regession-analysis-use-adjusted-r-squared-and-predicted-r-squared-to-include-the-correct-number-of-variables

R squared is a statistical measure of how close the data point is fitted to the regression line. It is also known as the coefficient of determination.R-Squared is defined by the explained variation divided by total variation that is explained by the linear model.

R squared value always lies between 0% to 100 % hence 0% indicates none of the variability of the response data around its mean and 100 % shows model explains all the variability of the response data around its mean. This clearly means a higher R square value model perfect your model is.

R-squared = Explained variation / Total variation

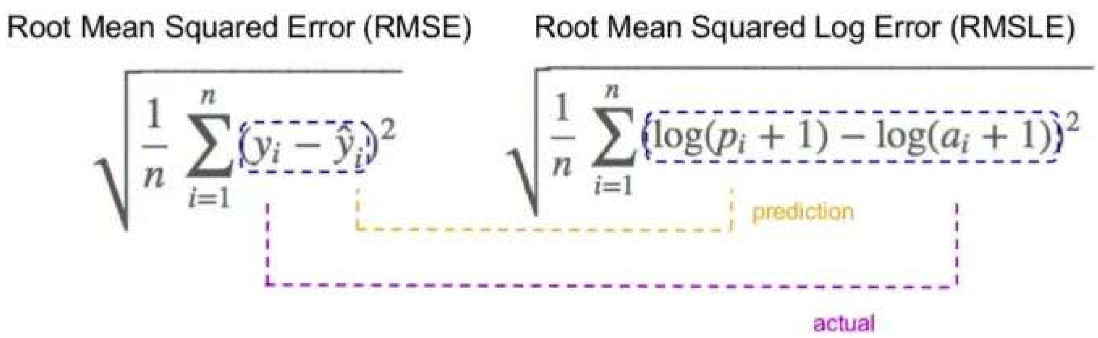
On the other hand, R squared *cannot* determine whether the coefficient estimates and predictions are biased. So Adjusted R squared come into the picture, It has explanatory power for regression models that has a different number of predictors Putting it in a simple way Adjusted R squared basically explains regression models having multiple independent variables or predictors.

$$\bar{R}^2 = 1-\left(1-R^2\right)\left[\frac{n-1}{n-\left(k+1\right)}\right]$$

The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. It increases only if the new term improves the model more than would be expected by chance. Adjusted R-squared is not a typical model for comparing non-linear models, but multiple linear regressions.

5. Root mean squared logarithmic error

As the name suggest *Root mean squared logarithmic error* takes the log of actual values and predicted value. This type of evaluation metric is usually used when we don’t want to penalize huge differences in the predicted and the actual values and these predicted and actual values are considered to be huge numbers.


Root Mean Squared Error (RMSE)
Root Mean Squared Log Error (RMSLE)
$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$
prediction
actual

source:https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/

3 — For Clustering

As compared to classification, it is difficult to figure out the quality of results from clustering. Evaluation metric cannot depend on the labels but only on the goodness of split. Moreover, we do not usually have true labels of the observations when we use clustering.

1. *Adjusted Rand Score*

Adjusted Rand score does not depend on the label values but on the cluster split. In other words, the Adjusted rand score calculates a share of observations for which these splits i.r.initial and clustering result is consistent.

One point is noted here that this metric is symmetric and does not depend upon the label permutations. The formula for Adjusted Rand score is given by

$$\mathrm{RI} = \frac{2(a+b)}{n(n-1)}.$$

Where N be the number of observations in a sample, a to be the number of observation pairs with the same labels and located in the same cluster, and b to be the number of observations with different labels and located in different clusters.

2.Adjusted mutual information

Much more similar to Adjusted Rand score as it also does not depend on the permutation of labels and is basically symmetric metrics. *Adjusted mutual information* is defined by entropy function and interpreted by sample split which is the likelihood of assigning a cluster. Mutual information is basically higher for two clusters two with a larger number of clusters, regardless of whether there is actually more information shared.

Basically, the MI measures whether the share of information common for both clusterings splits i.e. how information about one of them decreases the uncertainty of the other one.

AMI lies between range 0 and 1 values close to 0 means splits are independent and value close to 1 means they are similar.

$$h = 1 - \frac{H(C \mid K)}{H(C)}, c = 1 - \frac{H(K \mid C)}{H(K)},$$

Here K is a clustering result and C is the initial split. *h* evaluates whether each cluster is composed of same class objects and *c* measures how well the same class fits the clusters.

3.Silhouette

The coefficient of *silhouette* score is calculated by mean intra-cluster distance and the mean nearest-cluster distance for each sample.

$$s = \frac{b - a}{\max(a, b)}.$$

The Silhouette distance shows up to which extent the distance between the objects of the same class differs from the mean distance between the objects from different clusters. Values of silhouette score lie between -1 to +1 .if the value is close to 1 then it corresponds to good clustering results having dense and well-defined clusters however if the value is close to -1 then it represents bad clustering. Therefore, the higher the silhouette value is, the better the results from clustering.

Also with the silhouette score, we can also define the optimum number of clusters by taking the number of clusters that maximize the silhouette coefficient.

> HUSSH! We came to this wonderful journey of machine learning evaluation metrics. There are a lot of others performance metrics too you can check out the references section for more info.

If you are in a dilemma that which metrics to choose for your Machine learning Algorithm check out this Awesome Blog.

References

- https://www.kaggle.com/kashnitsky/topic-7-unsupervised-learning-pca-and-clustering
- https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/
- https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234
- https://www.coursera.org/lecture/big-data-machine-learning/metrics-to-evaluate-model-performance-pFTGm
- https://link.springer.com/chapter/10.1007/978-3-642-39712-7_1

- https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

If you like this post, please follow me. If you have noticed any mistakes in the way of thinking, formulas, animations or code, please let me know.

Cheers!

Machine Learning · Artificial Intelligence · Data Science · Evaluation Metric · Performance

Published in Analytics Vidhya

Follow

70K Followers · Last published Oct 15, 2024

Analytics Vidhya is a community of Generative AI and Data Science professionals. We are building the next-gen data science ecosystem https://www.analyticsvidhya.com

Written by Shashwat Tiwari

Follow

167 Followers · 33 Following

Senior Applied Data Scientist at EY || Machine Learning and Deep Learning Ardent ||

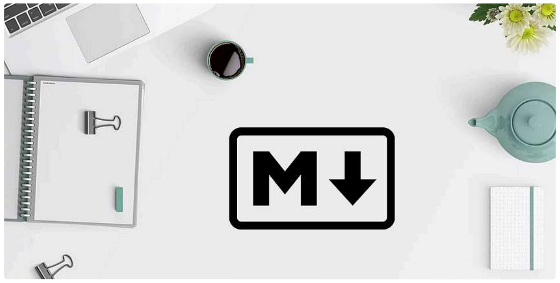
More from Shashwat Tiwari and Analytics Vidhya

In Geek Culture by Shashwat Tiwari

7 Important Distance Metrics every Data Scientist should know.

7 Important Distance Metrics every Data Scientist should know.

Jun 30, 2021 · 98 · 1

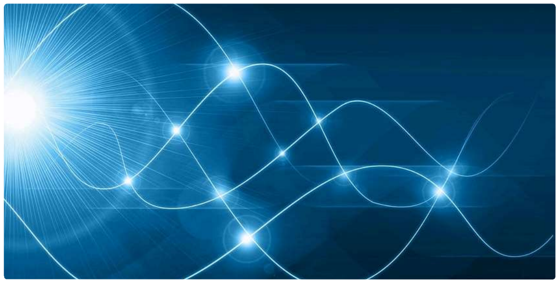


In Analytics Vidhya by Hannan Satopay

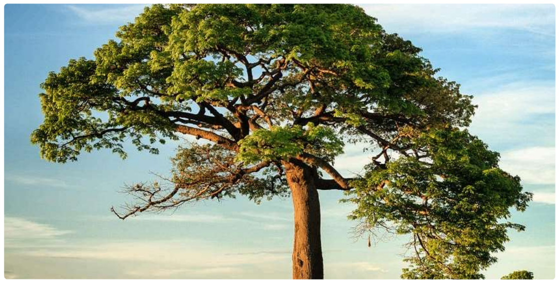
The Ultimate Markdown Guide (for Jupyter Notebook)

An in-depth guide for Markdown syntax usage for Jupyter Notebook

Nov 18, 2019 · 2.4K · 13



In Analytics Vidhya by Leland Roberts



In Analytics Vidhya by Shashwat Tiwari

Understanding the Mel Spectrogram

(and Other Topics in Signal Processing)

Mar 6, 2020 2.2K 27

Introduction to Computer Vision & OpenCV in Python

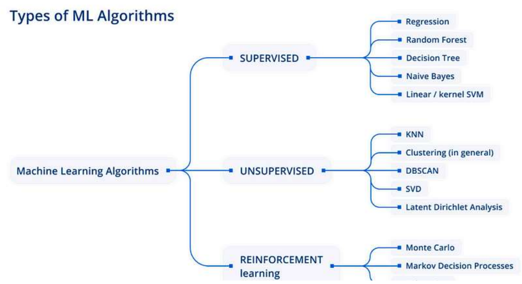
Experimenting with OpenCV in Python

Dec 16, 2019 271

See all from Shashwat Tiwari

See all from Analytics Vidhya

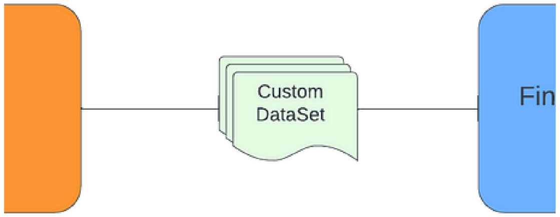
Recommended from Medium



John Vastola

10 Must-Know Machine Learning Algorithms for Data Scientists

Machine learning is the science of getting computers to act without being explicitly…



Suman Das

Fine Tune Large Language Model (LLM) on a Custom Dataset with…

The field of natural language processing has been revolutionized by large language…

Dec 6, 2022 926 12

Jan 25 1.96K 21

Lists

Predictive Modeling w/ Python

20 stories · 1700 saves

Practical Guides to Machine Learning

10 stories · 2068 saves

Natural Language Processing

1842 stories · 1466 saves

ChatGPT prompts

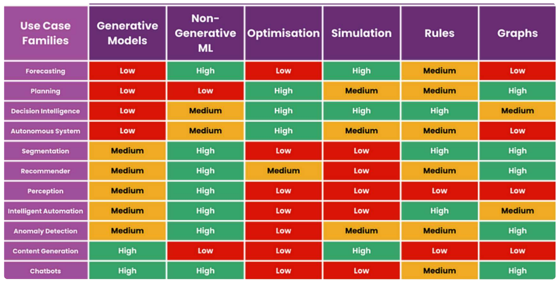
50 stories · 2317 saves

In Stackademic by Abdur Rahman

Python is No More The King of Data Science

5 Reasons Why Python is Losing Its Crown

Oct 23 9K 34



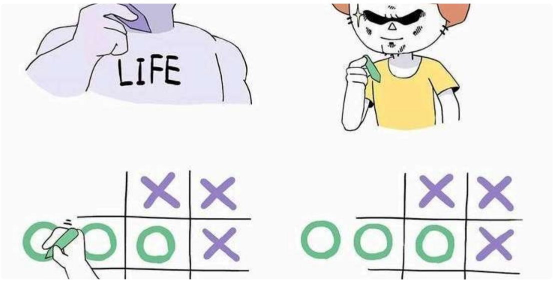
Use Case Families	Generative Models	Non-Generative ML	Optimisation	Simulation	Rules	Graphs
Forecasting	Low	High	Low	High	Medium	Low
Planning	Low	Low	High	Medium	Medium	High
Decision Intelligence	Low	Medium	High	High	High	Medium
Autonomous System	Low	Medium	High	Medium	Medium	Low
Segmentation	Medium	High	Low	Low	High	High
Recommender	Medium	High	Medium	Low	Medium	High
Perception	Medium	High	Low	Low	Low	Low
Intelligent Automation	Medium	High	Low	Low	High	Medium
Anomaly Detection	Medium	High	Low	Medium	Medium	High
Content Generation	High	Low	Low	High	Low	Low
Chatbots	High	High	Low	Low	Medium	High

In Towards AI by Christopher Tao

Do Not Use LLM or Generative AI For These Use Cases

Choose correct AI techniques for the right use case families

Aug 10 3.8K 40

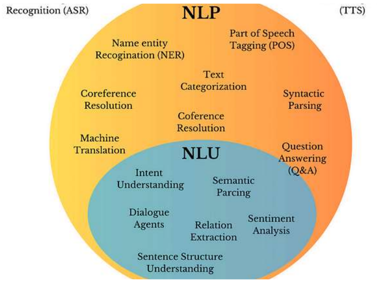


In Biased-Algorithms by Amit Yadav

SHAP Values vs Feature Importance

If you think you need to spend $2,000 on a 120-day program to become a data scientist…

Sep 19 49



Vipra Singh

LLM Architectures Explained: NLP Fundamentals (Part 1)

Deep Dive into the architecture & building of real-world applications leveraging NLP…

Aug 15 2.1K 13

See more recommendations