Supervised Learning Algorithms and Evaluation Metrics in Machine Learning

Bharath Ganji, Dept. of. E.C.E, RGUKT, Nuzvid, A.P, India. Email: bharath@rguktn.ac.in
Riyaz Hussain Shaik, Dept. of. E.C.E, RGUKT, Nuzvid, A.P, India. Email: riyazhussain@rguktn.ac.in
Satya Prakash G Dept. of. E.C.E, RGUKT, Nuzvid, A.P, India.

Article Info
Volume 82
Page Number: 2710 - 2717
Publication Issue:
January-February 2020

Article History
Article Received: *14 March 2019*
Revised: *27 May 2019*
Accepted: *16 October 2019*
Publication: *18 January 2020*

Abstract:
In the era of emerging technologies, advancements in machine learning techniques have created a great impact in our lives. These advancements made us to develop new way of approaches to solve problems in different areas such as cancer diagnosis, predictive forecasting, speech recognition etc. Various machine learning techniques like Supervised learning, Unsupervised learning, Reinforcement learning have been extensively used to transform a computer into an intelligent machine to solve complex and challenging problems in real world. The advancements in our technologies in terms of computational power and acquiring large data made machine learning models to grow complex day-by-day. So building a good model is not sufficient since evaluating a model is equally important. So a good metric is needed to estimate the prediction error. Although plethora of metrics are available in the community of machine learning, confusion arises very often in choosing the right metric. However there is no common theory in choosing a metric to evaluate a model. This paper describes popular supervised learning algorithms along with the mathematics behind them. We analyze and evaluate the important evaluation metrics in classification and regression algorithms. After assessing the discussed evaluation metrics, a conclusion is made on choosing the right evaluation metric for the given problem.

Keywords:*Supervised Learning, Classification, Regression, Evaluation metrics.*

I. INTRODUCTION

Machine learning the area in which computers are made to mimic human intelligence. This can be achieved by devising models using computer software and advanced algorithms. These provide the ability for a computer to learn and acquire knowledge to improve its performance and accuracy over a period of time without being programmed explicitly. These created models help machine to extract hidden relationships between the features of given data and give a reliable and corresponding output. Simple tasks such as finding factors of a given number are very easy and can be easily programmed with some set of inputs to get respective outputs. But some tasks such as spam email filtering [1-2] cannot be understood easily and this is where machine learning comes into picture. Such set of circumstances which cannot be programmed in a straightforward way to get results, we make the machine to learn and analyze from the data to make an intelligent prediction. Machine learning has gained importance in different areas such as Medical diagnosis [3], Robotics [4], autonomous driving cars [5] etc. This way of learning data to obtain outputs made machine learning very diverse and helped researchers and engineers to implement this concept to get valid results and intelligent decisions.

Evaluating the performance is equally important as building a good model. So the benchmarks we consider to evaluate our model play a significant role in tuning the performance of the model to yield good results. Therefore, knowing most common factors which tamper the performance of the model should be known. In this paper we mainly focus on supervised learning algorithms [6] and their evaluation metrics [7].

Machine learning strategies are broadly classified into supervised learning and unsupervised learning. Unsupervised learning algorithms deal with unlabeled data and are commonly used in areas like clustering [8], image segmentation [9], anomaly detection [10] etc. This paper mainly focuses on supervised learning algorithms and their corresponding metrics.

In supervised learning, algorithms learn from a set of predefined inputs and outputs. These algorithms maps the input data to its corresponding output based on the collected input output data pairs. In this way the algorithm learns the patterns in given data and estimates the output when a new input is given. This mode of learning depends on the truth that original class of the input training data is known. For example, while building as dog classifier we train the algorithm with a large set of dog and non-dog pictures. This makes the algorithm to study the features in dog pictures and acquire knowledge. When a new input is given to the algorithm it classifies the given picture as a dog or not-dog based on the knowledge which is acquired during the training process. Supervised learning is broadly divided into classification and regression techniques.

In the remainder of the paper, we define y as the true output of the input dataset and y' as the predicted output of the model. X represents the input feature vector, x is the single input and $x_j$represents the j^{th} training example from the input space. $N$is the number of examples in the input data.

The remainder of the paper is organized as follows:
Section II describes commonly used classification algorithms and the respective evaluation metrics are discussed in section III. Later in section IV and section V regression algorithms and their classification metrics are discussed.

II. CLASSIFICATION

Classification deals with the task of predicting the class of data to which it belongs to, based on one or more independent features. It approximates the input variables X to discrete output values Y. The output values Y are often called as categories or labels. Classification task with two possible outcomes is called as a binary classification problem. One example for a binary classification problem is classifying an email as spam or non-spam. Multi-class classification gives multiples discrete valued outputs. Hand written digits classification which gives outputs from 0-9 is one of the example for a multi class classification problem. The major classification algorithms in supervised learning are discussed below.

A. Logistic regression

Very widely used algorithm for binary classification problems. Logistic regression gives the probability values of the classes to which the input belongs to. The logistic regression maps the output value between 0 and 1 using a squishing function called sigmoid function. The mathematical representation of sigmoid function is given below.

$$p = \frac{1}{1+e^{-Z}} \qquad (1)$$

Where Z is the function of input variables.

B. Naïve Bayes Classifier

This classifier uses conditional probabilities to determine the class of the given data. Naïve Bayesian classifier assumes that all the input features are independent and calculates the probability for every outcome and pick the class with highest probability. This classifier takes motivation from the Bayes theorem of probability and is defined as

$$P(y/X) = \frac{P(X/y)P(y)}{P(X)} \qquad (2)$$

Where,

$P(y/X)$	-Conditional probability of class when feature is given.
$P(X/y)$	-Conditional probability of predictor when the class is given.
$P(y)$	-Probability of the label to be

predicted

$P(X)$ -Probability of the feature

This classifier is based on Bayes theorem of probability and very useful for large datasets. This classifier is often used in classification tasks like Text classification, spam filtering and in recommender systems. The main problem with this classifier is it cannot produce a valid output when the conditional probability is zero for a particular feature.

C. K-Nearest Neighbor

The basic assumption of KNN algorithm is that similar things stay together. KNN uses the distance measure between a given input data point and k-nearest points. Based on this distance measure, the given data point is allocated to the class which has greatest comparability among the k-nearest neighbors. The distance measure is chosen based on the problem we are solving. The Minkowski distance is a generalized form of distance measures used in KNN algorithm and is defined as

$$D = \left(\sum_{j=1}^{m} |x_j - y_j|^q \right)^{1/q} \quad (3)$$

The value of q can be manipulated to obtain different measures such as

Euclidian distance : q= 1
Manhattan distance : q=2
Chebyshev distance : q= ∞

This algorithm is less prone to noise and it is very easy to implement a multiclass problem. But the main drawback of this algorithm is to choose optimal value of "k". KNN algorithm is very easy to implement but the effectiveness or speed of the algorithm decreases with increase in the size of dataset.

D. Decision Trees

Decision trees [11] can be used in concept of both classification and regression. The decision tree classifier works on "if-then" rule. The tree can be interpreted by two types of nodes namely, decision nodes and leaf nodes. The leaf nodes represent the outcomes or decisions of the task, based on the results of tests applied at decision nodes. Generally Iterative Dichotomiser 3 (ID3) algorithm is used to build a decision tree. ID3 algorithm users the concept of concept of information gain and entropy to make up a decision tree. The Shannon entropy (H) measure is used to measure the randomness in the given data and is given by the following formula:

$$H = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)} \quad (4)$$

III. PERFORMANCE EVALUATION METRIC FOR CLASSIFICATION PROBLEM

Performance measures for classification tasks [12] are discussed below.

Confusion Matrix:

It is one of the most instinctive and simple approach to evaluate the performance of a model. Though the formation of the matrix does not give any direct measures as such, but relatively all the metrics to evaluate the performance of a model can be derived from it. Confusion matrix can also perform feature selection [13]. This generally is a square matrix of size $n * n$ where n defines the number of labels or the classes of the given data set. The rows of the matrix the true labels or classes of the data and the columns of the matrix represent the estimated classes obtained from the model.

Below are the terms related to the confusion matrix shown in Table.1

Table.1: Confusion matrix

	Predicted Values	
	Positive	Negative
Actual True	TP	FP
Actual false	FN	TN

True positives (TP): True positives are the instances where the predicted class and the actual class are evaluated true.

True negatives (TN): True negatives are the instances where the predicted class and the actual class are evaluated false.

False positive (FP): False positives are the instances where the actual class is false but it is evaluated true.

False negative (FN): False negatives are the instances where the actual class is true but it is evaluated false.

It can observed that the error in the model is due to incorrectly classified data i.e. false positives and false negatives. So based on these parameters we now devise the evaluation metrics as follows:

A. Precision

It can defined as the ratio of true positive classes to the predicted positive classes. Precision determines the confidence of the model. Precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \qquad (5)$$

Precision [14] tells how precise our model is i.e. out of all classes present, how many classes does our classifier predicted correctly. It should be as high as possible

B. Recall

It is the ratio of true positives to the actual positive classes. It can also be defined as the percentage of positive cases which are classified correctly. Recall is also called as sensitivity of the model and is calculated as:

$$Recall = \frac{TP}{TP + FN} \qquad (6)$$

C. Accuracy

It is the ratio of correct predictions to the overall predictions made by the classifier. Accuracy is a good evaluation metric when the target labels of the data set are almost symmetrical i.e. no. of positive classes = no. of negative classes. But when the given data is skewed then accuracy is not a good measure. Accuracy is given as:

$$Accuracy = \frac{TP + FN}{TP + FP + TN + FN} \qquad (7)$$

D. F1 Score

Sometimes precision and recall alone cannot give a good measure of the performance of the model. So it is good to have a tradeoff between these two metrics. One way to do this is to compute the harmonic mean of precision and recall. This gives us the F1 score [14] of our model and can be described mathematically as:

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision} \qquad (8)$$

E. AUC-ROC curves

AUC (Area under Curve) and ROC (Receiver Operating Characteristics) curves [15] give a good visualization on the classification strength of the model. These curves give us a good estimation of our model at different threshold settings. AUC tells us how good our classifier at differentiating the classes is. Higher the AUC, better the model in differentiating the classes of the dataset. ROC curve gives the probability and is plotted between false positive values on X- axis and True positives on Y-axis. A model which is good at separability have AUC equal to 1 and a worst model will have AUC equal to zero. Let us understand this statement from the figures. Let Blue circle represent the positive class and orange circle represent the negative class.

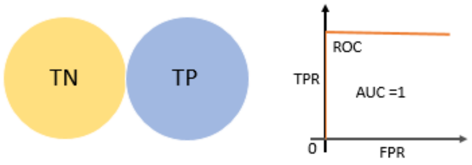

Fig.1: AUC= 1

Fig.1 is an ideal model which has high degree of separability i.e. it can classify positive and negative examples very accurately.

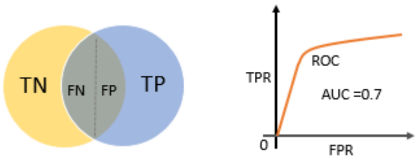

Fig.2: AUC= 0.7

When two classes overlap, error can be observed. Here in Fig.2 for AUC is 0.7 there is a 70% chance that the model will predict accurately.

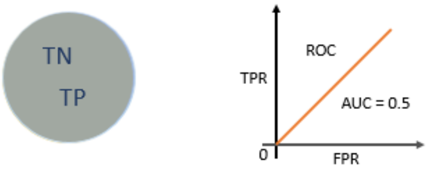

Fig.3: AUC= 0.5

Fig.3 represents a worst model. When AUC is 0.5 the classifier is not capable in distinguishing the classes. It treats both classes as either positive or negative

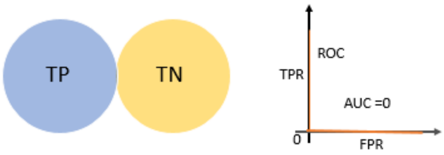

Fig.4: AUC = 0

This model is misclassifying all the examples i.e. predicting positive class as negative and vice-versa. The categorisation of ROC curves is given in Table 2.

Table 2. Categorisation Of ROC curves

AUC	Category
0.9 - 1.0	Very good
0.8 - 0.9	Good
0.7 - 0.8	Fair
0.6 - 0.7	Poor
0.5 - 0.6	Fail

F. Log-Loss

Log lossis the best evaluation metric when the model is a binary classifier. It determines the performance based on the confidence of the prediction and penalizes the model for a wrong classification. Log loss measures the cross entropy between the original and predicted class. Hence it is also called as Entropy loss. For binary classifications (y belongs to 0, 1) log loss is defined as:

$$P = -\frac{1}{N}\sum_{j=1}^{N} y_j \, log(y') + (1 - y_j) \, log(1 - y') \qquad (9)$$

Where y_j is the label of original class and $y'$is the prediction probability of the classifier and is given by equation (1)

Out of all the metrics discussed, a question often arises in choosing a suitable metric. So to summarize briefly, inferences can be drawn as below.

In case of balanced dataset i.e. number of positive classes are nearly equal to negative classes, Log-loss is preferred when we need the probabilistic difference between original and predicted classes. If only the final predictions matter then the AUC score is the metric to go with. F1 score is very helpful when comparing two models for the same problem. The threshold must be fine-tuned before comparing the models. If we want to minimize the error due to false positives precision should be made as high as possible. On the other hand recall should be made close to 100% if we want to minimize the false negatives.

In case of imbalanced data set i.e. if the dataset is skewed, F1 score is preferred when the there is a small positive class. If a smaller class is to be taken care of whether it is a positive or negative class AUC score will be a good option.

IV. REGRESSION

Unlike classification regression predicts continuous valued outputs. Regression demonstrates the relation between input and output variables. Regression algorithms extract the hidden relationship between inputs and outputs of the dataset to predict a new output when a new input is given. Regression is widely used in real world applications such as stock price prediction, weather data analysis, astronomical data analysis etc. Linear regression is the most intrinsic and simple algorithm of regression.

A. Linear Regression

Linear regression establishes a linear relationship between input variables X and output variables Y. It predicts the output value Y by multiplying the input variables X with constants called weights and then summing them. A simple linear regression model can be mathematically defined as follows:

$$y' = \eta_1 x + \eta_0 \qquad (10)$$

Where y' is the predicted regression line and η_1, η_0 are the weights of the model. The main disadvantage of linear regression is that it is more sensitive to the noise in the data i.e. outliers in data greatly affect the regression line.

From the fig 5 it seems that that linear model is not the best fit for our data. So a complex function is needed which will give a good estimate .This can be done by adding more complex features and is called as polynomial regression.

B. Polynomial regression:

A regression is polynomial when the order of the input independent variable is greater than 1. This is preferred when there is no linear relationship between the inputs and outputs in the given data. In polynomial regression, the model is a curve rather than a straight line as seen in linear regression model. The predictive function for a 2nd order polynomial regression function can be represented as:

$$y' = \eta_0 + \eta_1 x + \eta_2 x^2 \qquad (11)$$

But sometimes adding more complexity raise a problem of over fitting of the data. Over fitting is the problem of fitting the data too well as shown in fig (5-c). Over fitting prevents the model from generalizing the data and thus fail to predict the output when a new input is given. So methods like lasso regression and ridge regression are used to overcome the problem of over fitting.

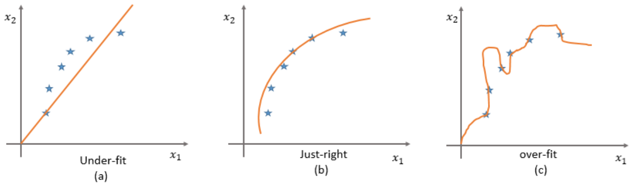

Fig.5: demonstration of different models

C. Ridge regression

Ridge regression solves the problem of multicollinearity in the input data. This occurs when there is high correlation between the independent input variables. Ridge regression reduces the error by adding a degree of bias to original regression estimate. This uses L2 regularization technique (adding squares of weights to the error function to penalize the error) to shrink the value of weights, thus reduce over fitting of the data. The error function of ridge regression is given by:

$$Cost = \sum_{j=1}^{N} \underbrace{(y-y')^2}_{\text{Loss}} + \underbrace{\lambda |\eta|^2}_{\text{Penalty}} \qquad (12)$$

Where λ is called as the regularization parameter [16]. It controls the tradeoff between the original loss and the penalty term. It shrinks the values of weights but doesn't make them to reach zero.

D. Lasso regression

Lasso stands for least absolute shrinkage and selection operator. It uses L1 regularization technique (adding absolute values of weights) to penalize the error and reduce the variance in the data. It helps in feature selection by setting some weights exactly equal to zero thus making the model more sparse and generalized.

$$Cost = \sum_{j=1}^{N} \underbrace{(y-y')^2}_{\text{Loss}} + \underbrace{\lambda |\eta|}_{\text{Penalty}} \qquad (13)$$

Lasso regression is very helpful when there are dependent values in the dataset. This algorithm just pick the right one and shrinks weight coefficients of remaining variables to zero.

V. EVALUATION METRICS FOR REGRESSION

The most commonly used regression metrics are discussed below:

A. Mean squared error (MSE)

Mean squared error basically calculates the average squared error between the predicted values and the original values. MSE is the most commonly used metric since it is very simple and easy to implement. The equation for the MSE is given as:

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - y_j^{'})^2 \qquad (14)$$

The higher the error the worst the model is. MSE is never negative because of squaring the error. The error for a good model will be near to zero. But the main disadvantage of MSE is making a single bad prediction results in huge error because of the squaring term. So MSE is more prone to the noise present in the data set. RMSE (Root Mean Squared Error) is another measure which is similar to MSE is and is calculated as

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - y_j^{'})^2} \qquad (15)$$

Although RMSE and MSE are similar in in terms of calculating the error, they differ in their respective gradient based models. MSE is widely used of the two because it is easy to implement.

B. Mean Absolute Error (MAE)

Unlike MSE and RMSE, MAE calculates the absolute error between the predicted and original values. The gradient of MAE is a step function and it is equal to +1 when y' is greater than the target and -1 when it is smaller. The main advantage of MAE over MSE is that it is not that sensitive to the noise present in the data. MAE is calculated by using the below formula:

$$MAE = \frac{1}{N}\sum_{j=1}^{N}| y_j - y_j^{'} | \qquad (16)$$

C. R-Squared error (R^2- error):

It is defined as the measure of closeness of data points to the fitted regression line. This can also be interpreted as how much variance in the data is explained by the model fit. It is also called as coefficient of determination and the values of R-Squared error lies between $-\infty$to 1. R-squared error is calculated as:

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)} \qquad (17)$$

Where MSE (model) is mean squared error of the model and MSE (baseline) is defined as the total variation in the data and is given by

$$MSE(baseline) = \frac{1}{N}\sum_{j=1}^{N}(y_j - y_{mean})^2 \qquad (18)$$

y_{mean} is the mean seen in y_j. The closer the value to 1, the better the model fits our data, Although R-squared error alone cannot predict how good the model is. It cannot estimate whether the model is biased or not. So to say simply R-squared measure is the ration of how good the predicted model is and how good is the mean model. To summarize, R-Squared error alone cannot predict how good a model is, it gives a comparison between the original model and the mean model. MAE should be preferred when there are outliers in the dataset since it is less prone to noise as compared to MSE and RMSE. MSE or RMSE can be used if there are no outliers since it is easier to implement**.**

VI. CONCLUSION

In this paper we have focused on supervised learning algorithms mainly classification and regression. We delved deeply into the types of classification and regression algorithms and presented their mathematical representations. Plethora of evaluation metrics are discussed and a conclusion is drawn on choosing the right metric based on whether the dataset is balanced or imbalanced.

VII. REFERENCES

1. Yan Zhang, PengFei Liu and JIngTaoYao, “Three-way Email Spam Filtering with Game-theoretic Rough Sets”, International Conference on Computing, Networking and Communications ICNC): Communications and Information Security Symposium Pp.552-556, 2019.
2. Chirag Visani, Navjyotsinh Jadeja and Manali Modil, “A Study on Different Machine Learning Techniques for Spam Review Detection”, International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017) Pp.676-679.
3. Qian Wang, Yinghuan Shi, and Dinggang Shen, “Machine Learning in Medical Imaging”, IEEE Journal Of Biomedical And Health Informatics, Vol. 23, No. 4,pp.1361-1362 July 2019.

4. DongWook Kim , Jae In Kim and Yong-Lae Park , "A Simple Tripod Mobile Robot Using Soft Membrane Vibration Actuators", IEEE Robotics and Automation Letters , Volume: 4 , Issue: 3 ,pp. 2289 – 2295,July 2019.
5. Juraj Kabzan, Lukas Hewing, Alaxander Liniger and Melanie N. Zeilinger, "Learning-Based Model Predictive Control for Autonomous Racing", IEEE Robotics and Automation Letters, Vol. 4, No. 4, pp.3363-3370, October 2019.
6. Hemant Kumar Gianey and Rishabh Choudhary, "Comprehensive Review on Supervised Machine Learning Algorithms", International Conference on Machine learning and Data Science.Pp.37-43, 2017.
7. Johannes Furnkranz and Peter A. Flach, "An Analysis of Rule Evaluation Metrics." Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.
8. Wenbin Song, Long Wen, Lian Gao and Xinye Li, "Unsupervised fault diagnosis method based on iterative multi-manifold spectral clustering", IET Collab. Intell. Manuf., 2019, Vol. 1 Iss. 2, pp. 48-55, June 2019.
9. Brian H. Wang , Wei-Lun Chao , Yan Wang , Bharath Hariharan , Kilian Q. Weinberger and Mark Campbell, "LDLS: 3-D Object Segmentation Through Label Diffusion From 2-D Images", IEEE Robotics And Automation Letters (Volume: 4 , Issue: 3 , pp. 2902-2902 July 2019.
10. K. Nandhini, M. Pavithra, K. Revathi and A. Rajiv, "Anamoly detection for safety monitoring", Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)", October 2017.
11. S.R.Safavian and D. Landgrebe, "A survey of decision tree classifier methodology", IEEE Transactions on Systems, Man, and Cybernetics (Volume: 21, Issue: 3) Pp: 660 – 674, May/Jun 1991.
12. Marina Sokolova , Guy Lapalme, "A systematic analysis of performance measures for classification tasks" Information Processing and Management 45 Pp.427–437, 2009.
13. Sofia Visa, Brian Ramsay, Anca Ralescu and Esther van der knap, "confusion matrix based feature selection", The 22nd Midwest Artificial Intelligence and Cognitive Science Conference, Cincinnati, Ohio, USA, April 16-17, 2011.
14. Yutaka Sasaki, "The truth of F-measure", 2017.