Relevance As a Metric for Evaluating Machine Learning Algorithms

Aravind Kota Gopalakrishna, Tanir Ozcelebi, Antonio Liotta, and Johan J. Lukkien

[1] System Architecture and Networking (SAN),
Department of Mathematics and Computer Science,
[2] Electro-Optical Communications,
Department of Electrical Engineering,
Eindhoven University of Technology, The Netherlands
`{a.kota.gopalakrishna,t.ozcelebi,a.liotta,j.j.lukkien}@tue.nl`

Abstract. In machine learning, the choice of a learning algorithm that is suitable for the application domain is critical. The performance metric used to compare different algorithms must also reflect the concerns of users in the application domain under consideration. In this work, we propose a novel probability-based performance metric called *Relevance Score* for evaluating supervised learning algorithms. We evaluate the proposed metric through empirical analysis on a dataset gathered from an intelligent lighting pilot installation. In comparison to the commonly used Classification Accuracy metric, the Relevance Score proves to be more appropriate for a certain class of applications.

Keywords: Machine learning algorithms; performance metric; probabilistic approach

1 Introduction

One of the general goals of a machine learning (ML) algorithm is to capture complex relationships and patterns within a dataset and to apply such knowledge on new data that is believed to have the same (or a similar) pattern. The success criteria for the ML algorithm depend on many factors, most importantly, on the user concerns that are specific to the application domain.

Consider a car navigation system that utilizes a specific ML algorithm to select the best route from a source location to a destination. There are many ways to measure the performance of this algorithm objectively. Useful metrics can be, for instance, the actual time to reach the destination, the amount of gas consumption and the amount of tolls paid. However, it is often the case that a combination of these metrics cannot directly be mapped to the real satisfaction of the driver from the navigation experience, whereas this is what matters in the end. Assume that there are only two home-office routes possible and the driver *generally* likes to take the first route, which is shorter, faster and safer. On the other hand, *sometimes*, the driver likes to take the second route that goes along

the sea shore, especially when it is sunny and the driver is in the mood. The ML algorithm can be expected to learn to pick the desired route among the two only if weather is a parameter that can be and is monitored by the navigation device. Monitoring the mood of a person is very difficult nonetheless. Alternatively, the ML algorithm will simply start looking for patterns within other data that is available to it, e.g. the time of the day, which may or may not have any actual impact. In this case, the desired route may also depend on those parameters, i.e. features, which are relevant yet not considered by the navigation device.

It is a difficult task to identify *all* the features that are relevant for the ML algorithm employed by an application. The features that are relevant for the application define a *context*. Even when the context is known, it may be technically very challenging or very intrusive to monitor some features especially when users are involved, e.g. the driver's mood in the previous example. The subset of relevant features that are monitored define an *observed context*. Depending on the application under consideration and the observed context, sometimes evaluating an output, i.e. a prediction made by the ML algorithm, as either *right* or *wrong* might not be sufficient as there are gray areas in between. For example, the intelligent environment described in [1] offers lighting services (light settings) depending on several contextual factors such as user identity, user activity and time of the day. When the dataset collected from this intelligent environment is examined, there are two important remarks to be made. The first remark is that the desired light settings are not fixed for a given instance of observed context. Since perception of lighting also depends on subjective factors that cannot be monitored, even the same user, the same activity and the same time of the day may result in different desired light settings. The second remark is that, for a given observed context, there are acceptable (desired at least once) and unacceptable light settings (never desired).

This means that the overall performance of the ML algorithm may be hindered by the fact that the observed context is not entirely representative of the actual context of an application. When this is the case, it is a challenge to evaluate and compare the performances of different ML algorithms in the application domain. In the literature, different evaluation metrics assess different characteristics of ML algorithms [2]. Classification problems in ML are broadly categorized into multi-class classification and multi-label classification problems. Multi-class classification algorithms [3] are for those categories of problems where, for a given input instance, there is exactly one correct output class, which is selected out of several classes. Examples of multi-class classification include image recognition and diagnosis [4]. The commonly used evaluation metric for multi-class classification problems are accuracy, precision and recall. Multi-label classification algorithms [5] address that category of problems where multiple output classes must be selected for each input instance. Examples of multi-label classification include text [6] and music categorization [7]. The commonly used evaluation metrics for multi-label classification problems are hamming-loss, precision and recall.

As illustrated by the examples of car navigation and intelligent lighting, there also exists another class of applications. In these applications, the ML prediction model has to select a unique output class for a given input instance (as in multi-class classification), and the output for a given input instance may fall into multiple acceptable output classes (as in multi-label classification). An interesting property of such an application is randomness in the output, i.e. there is no single acceptable outcome for a given observed context but there is a statistical regularity associated with different possible and acceptable outcomes. In such cases, even though the prediction models that are used to solve multi-class classification problems suits the need, the evaluation metrics such as accuracy, precision and recall are not suitable. For instance, the use of accuracy as a metric would lead to a score of zero for a misclassified sample, which should not be the case. The bad choice of performance metrics would produce results that are meaningless to achieve the design goal of an application [8] [9].

In this work, we devise a new evaluation scheme to evaluate the performance of supervised learning algorithms for this class of applications and propose a new evaluation metric called *Relevance Score (RS)*. We consider an intelligent lighting scenario to describe and formulate the problem. *Relevance Score* is defined as a percentage measure of how much relevant an output or outcome is, as predicted by the prediction model for a given observed context. The score is based on the probabilistic distances among the different outcomes, i.e. predicted and actual outcome. We use this metric to evaluate several rule based prediction models and then compare it with the commonly used metric *Prediction Accuracy or Classification Accuracy (CA)* to validate the significance of the proposed metric.

The paper is organized as follows. Section 2 discusses the drawbacks of using the commonly used CA metric with an example of intelligent lighting application and identifies improvement directions. Section 3 presents the proposed relevance metric. Section 4 provides an evaluation of the proposed metric. Finally, Section 5 concludes the paper.

2 Problem Definition

In this section, we discuss the drawbacks of *Classification Accuracy (CA)* as a metric for evaluating supervised learning algorithms, which motivates the need for a new metric using an intelligent lighting application scenario.

Let $x_i \epsilon X$ denote the ith sample in a dataset, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ and X represents an n-dimensional input feature space given by $X = X_1 \times X_2 \times X_3 \times \ldots \times X_n$. Let us denote the output class label as y where $y \epsilon Y = \{y_1, y_2, \ldots, y_K\}$ and K denotes the number of possible outcomes. In the intelligent lighting application under consideration [1], there are 8 possible presets for output lighting conditions (K=8) in a breakout area, as shown in the three dimensional space in Fig. 1. Each octant represents a possible light combination (e.g. static-warm-dim).
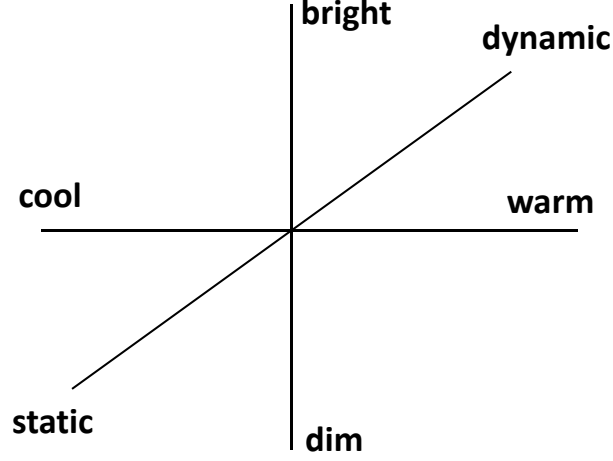
Fig. 1. Possible Output Light Combinations

A user is provided with a lighting condition based on six input features ($n = 6$) namely user-identity, type of activity, area of activity, number of users, time of the day and the external light influence.

In [1], the supervised learning approach is explored where a considered classification model is trained on the breakout dataset to predict a unique light output $y_i \epsilon Y = \{y_1, y_2, \ldots, y_8\}$ for a given input $x_i = [x_{i1}, x_{i2}, \ldots, x_{i6}]$. The success of an intelligent lighting application depends on selecting the best suitable prediction model that selects the desired lighting condition for a given observed context. The performance of the selected models is evaluated using CA as a metric.

Let us assume that a classification model h is trained using the breakout dataset. The breakout dataset contains a sample x 10 times and the output light condition selected by a user is light A *(LA)* four times, light B *(LB)* four times and light C *(LC)* two times. This means that the user for a given input instance x has selected LA, LB and LC with probabilities 0.4, 0.4 and 0.2 respectively, i.e. a sample x has multiple output classes. Assume that the classifier h selects LA based on what it has learnt from the training data, whereas the user actually desires LB as the lighting condition. Here the classifier h is not right with the prediction, but not entirely wrong either. If CA is used as a metric, then the selection made by the classifier h would be assessed as completely wrong. From the application point-of-view the CA metric is not very representative as it is important to measure how relevant the lighting condition is for a given observed context, rather than to measure how accurate it is. It can never be 100% accurate

Table 1. Sample comparison of metrics

Actual Output	Predicted Output	CA Metric	Alternative Metric
LA	LA	100	Good match (100)
LB	LC	0	Decent match (75)
LC	LA	0	Irrelevant match (0)
LA	LB	0	Relevant match (50)
LB	LB	100	Good match (100)
		$200/5 = 40\%$	$325/5 = 65\%$

as the user is not consistent in choosing the desired lighting condition, based on the observed context.

From the example above it is evident that CA is not a relevant metric for similar applications as it fails to capture the degree of relevance when there is a "non-ideal" prediction. Hence, it is necessary to have an evaluation mechanism that is able to compute the distance between actual and predicted outcome when they are not equal. An example of such a metric for a given classifier is shown in Table 1.

As shown in Table 1, by virtue of the problem, *CA* gives a low accuracy of 40%, whereas the user is given relevant outputs 80% of the time and irrelevant outputs only 20% of the time according to the *Alternative Metric*, resulting in an average score of 65%. Here the *Alternative Metric* computes the relevance of the outcome i.e. how good is the output selection made by a prediction model through the data statistics available from the observed environment, thereby making it more suitable than the *CA* metric.

Formally, we propose an evaluation metric named as *Relevance Score*, which is more suitable for the third class of applications mentioned in Section 1 than the commonly used CA metric. In particular, we seek to find a performance function $f : (X, Y) \rightarrow R_+^{100}$ where $R_+^{100} = \{RS \epsilon R | 0 \leq RS \leq 100\}$.

3   Proposed Metric

In this section, we propose a new mechanism to evaluate supervised prediction models used in the presented class of applications, e.g. in intelligent lighting. Here, we formulate a function that quantifies the relevance of the predicted outcome when there is a mismatch between the predicted and actual outcome.
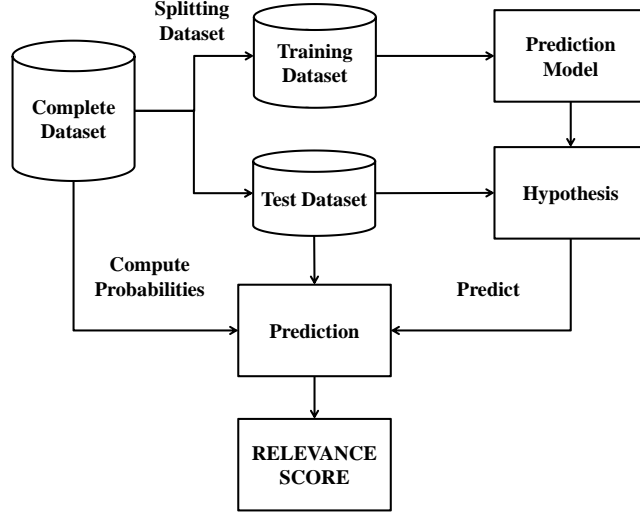
Fig. 2. Relevance Score computation procedure

The well known CA metric is computed as an average of the sum of individual accuracies for k test set samples as in equation 1.

$$CA = \frac{1}{k} \sum_{i=1}^{k} Acc_i \qquad (1)$$

We inherit the same mechanism to compute the *Relevance Score* on a test set for a given prediction model with a minor modification where the accuracy *Acc* for a sample is replaced by *Score* as in equation 2.

$$RS = \frac{1}{k} \sum_{i=1}^{k} Score_i \qquad (2)$$

We divide RS computation into two phases: 1. Probability Computation Phase, and 2. Evaluation Phase. The entire process of computing RS is shown in Fig. 2.

3.1   Probability Computation Phase

An *ErrScore* is a real number that is calculated when there is an error in prediction by the prediction model. It is computed based on five parameters, the predicted outcome (O_P), the actual outcome (O_A), the probability of occurrence of the predicted outcome ($P(O_P)$), the probability of occurrence of the actual

Table 2. Qualitative Relevance

CASE	OUTCOME	PROBABILITIES	QUALITATIVE RELEVANCE
1	$O_P = O_A$	-	Highly Relevant
2	$O_P \neq O_A$	$P(O_H) = P(O_P); P(O_P) > P(O_A)$	Moderately Relevant
3	$O_P \neq O_A$	$P(O_H) > P(O_P) > P(O_A)$	Relevant
4	$O_P \neq O_A$	$P(O_H) > P(O_A) > P(O_P)$	Less Relevant
5	$O_P \neq O_A$	$P(O_H) = P(O_A); P(O_A) > P(O_P)$	Irrelevant

outcome ($P(O_A)$) and the probability of the most frequently selected outcome ($P(O_H)$). These probabilities are the values that are computed from the entire dataset. The probability values are computed for the different output class labels ($y \epsilon Y$) for every input instance (x_i) in the breakout dataset. Formally, compute $P(y|x_i)$ for $i = 1, 2, 3, ..., m$ where m is the number of samples in the entire dataset. In this phase, the values of the most probable feature that is responsible for the randomness in output (e.g. user identity feature in intelligent lighting application) is removed for generalization.

3.2 Evaluation Phase

In this phase, the predictions on the test data are evaluated using the considered classification models. If d denotes the probabilistic distances in general, then the distances between the probabilities are given by,

$$d_{HP} = |P(O_H) - P(O_P)| \tag{3}$$

$$d_{PA} = |P(O_P) - P(O_A)| \tag{4}$$

$$d_{HA} = |P(O_H) - P(O_A)| \tag{5}$$

Several cases are possible for a given input instance based on the predicted and the actual outcomes and their computed probabilities. The cases may be classified qualitatively in terms of relevancy of the outputs provided. The decreasing order of relevance we consider for different cases is summarized in Table 2. The ordering is based on the consequence of individual cases as explained later. Thus, *ErrScore* is a score that is obtained by quantifying these cases.

Case 1: $O_P = O_A$

In this case, for a given input instance in the test set, the predicted outcome and the actual outcome are equal. Thus, there is no error in the predicted outcome.

Case 2: $O_P \neq O_A; P(O_H) = P(O_P) > P(O_A)$

In this case, for a given input instance the predicted outcome and the highest probable outcome are equal but a different output (actual outcome) is selected.

Since the predicted outcome is equal to the highest probable outcome ($d_{PH} = 0$) and the classification model has not been able to capture the switch in output, the error is kept minimum and is equal to $\beta \cdot d_{PA}$ where β is a positive real constant, whose value depends on the application.

Case 3: $O_P \neq O_A; P(O_H) > P(O_P) > P(O_A)$

In this case, for a given input instance the predicted outcome, the actual outcome and the highest probable outcome are not equal. Since the probability of the predicted outcome lies in between that of the highest probable outcome and the actual outcome, the error value is higher than the previous case, and is equal to $(\alpha \cdot d_{HP} + \beta \cdot d_{PA})$ where α and β denote positive real constants.

Case 4: $O_P \neq O_A; P(O_H) > P(O_A) > P(O_P)$

In this case, for a given input instance the predicted outcome, the actual outcome and the highest probable outcome are not equal. Since the probability of the predicted outcome lies farther away than that of the highest probable outcome and the actual outcome, the error value is much higher, and equal to $(\alpha \cdot d_{HP} + \beta \cdot d_{PA})$ as in Case 3.

Case 5: $O_P \neq O_A; P(O_H) = P(O_A) > P(O_P)$

In this case, for a given input instance the actual outcome and the highest probable outcome are equal but the classification model selects a different output (predicted outcome). Since the actual outcome is equal to the highest probable outcome and the performance of the classification model was poor to select a different output, the error rate is much higher than that of the previous cases. The error is equal to $(\alpha + \beta) \cdot d_{HP}$.

Combining above equations and normalizing over $(\alpha + \beta)$, we find the following error score function

$$ErrScore = \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta}. \tag{6}$$

Therefore, the RS for a sample is computed as below

$$Score = (1 - ErrScore) \times 100, \tag{7}$$

$$Score = (1 - \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta}) \times 100. \tag{8}$$

4   Experimentation and Results

In this section, we study the performance of the proposed RS metric and compare it to the CA metric for different classifiers used in the breakout dataset. We further investigate the influence of α and β parameters of the RS metric function.

The performance of RS metric is studied through various experimentations using the Weka [10] simulator on the following rule-based prediction models: DecisionTable [11], JRip [12], Nearest Neighbor with generalization (NNge) [13], PART [14], ConjunctiveRule [15] and Ridor [16] on the breakout dataset [1]. The breakout dataset consists of 236 samples of data from different users gathered

from the breakout area. Here 70% of the dataset is used for training and the remaining 30% is used as test set. We consider 10 randomly shuffled training and test sets for experimentation and then mean the results to avoid biased results. We perform the following investigations:

1. Comparison of RS and CA Metrics
2. Significance of α and β
3. Lower and Upper Bounds of RS with changing α and β
4. Testing of RS on Random Output Data

4.1   Comparison of RS and CA Metrics

Fig. 3 shows the graph of Performance Metric vs. Prediction Model for two different metrics: CA and RS. The RS values are computed with $\alpha = 2$ and $\beta = 1$. As discussed earlier, low accuracy values are achieved with the CA metric. This means that no prediction models provide exact lighting conditions based on the input instances in more than 50% of the test samples. However, with RS as a metric, a maximum of 73% relevant lighting condition is achieved with DecisionTable prediction model.

An interesting observation is that the prediction model Ridor has lower CA than Conjunctive Rule, whereas Ridor has a better RS value than Conjunctive Rule. This is interpreted as follows. For a portion of the data samples, the most desired outcome (the best match) is calculated for each prediction model and ConjunctiveRule performs better than Ridor, i.e. it has higher classification accuracy. However, when it comes to the relevance of the predicted outcomes, the Ridor is more successful than ConjunctiveRule, i.e. it predicts the most relevant outcomes on average. Even though ConjunctiveRule finds the best match slightly more often than Ridor, when it cannot find the best match, it comes up with less relevant predictions. Therefore, Ridor is preferable for applications where relevance of the predicted outcome is more critical than accuracy. As a consequence, for such applications, RS as a performance metric is more relevant.

4.2   Significance of α and β

In this experiment, we investigate the influence of varying values of α and β on RS. Fig. 4 shows the plot of RS vs. Prediction Model for different values of α and β. The graph shows that the prediction model DecisionTable is more consistent with the changing (α, β) whereas the RS values improve with α decreasing and β increasing for other prediction models. The parameters α and β are used as weights for probabilistic distances d_{HP} and d_{PA}, respectively. Thus, when α is higher the product $\alpha \cdot (d_{HP})$ is emphasized more, i.e. the probabilistic distance between the predicted and actual outcome is given less importance. When β is higher, the product $\beta \cdot (d_{PA})$ is emphasized more, i.e. the probabilistic distance between the most probable outcome and the predicted outcome is given less importance.

Fig. 5 shows the plot of RS vs. α, β for different prediction models. From the graph, we find that the predictions made by ConjunctiveRule model are closer to
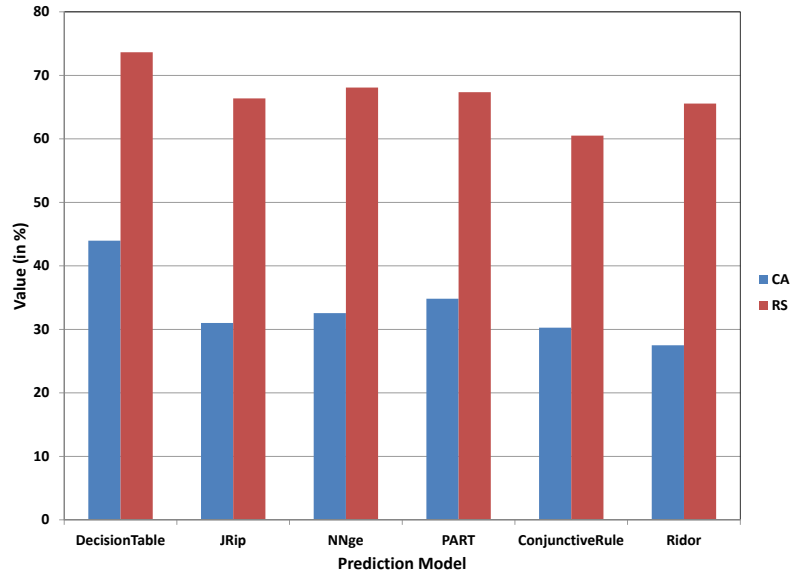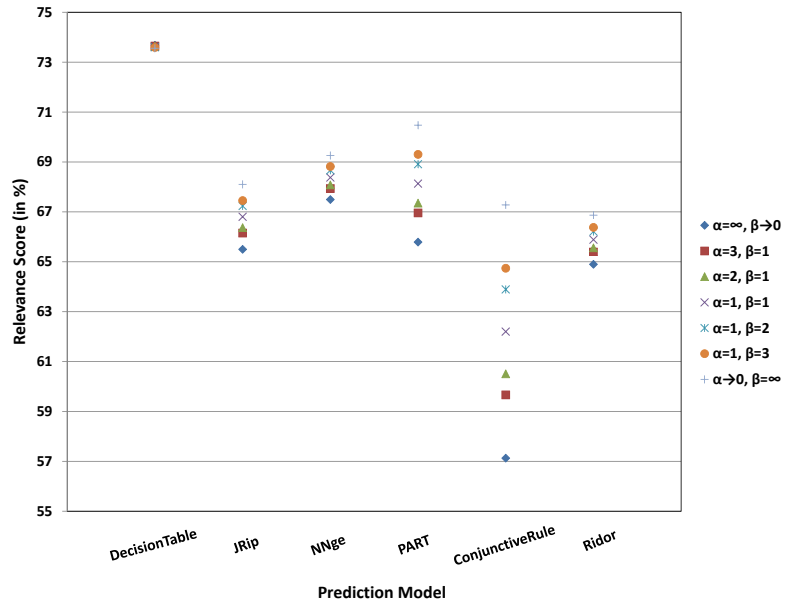
Fig. 3. Performance Metric vs. Prediction Model

Fig. 4. Relevance Score vs. Prediction Model

the actual outcome and hence varying α, β has more influence on the RS values. The low RS values for the ConjunctiveRule are due to the fact that the model is not good in consistently predicting relevant outcomes for the given scenarios.

4.3   Lower and Upper Bounds of RS with changing α and β

Maximum or minimum RS that a prediction model can reach is found by taking either α or β to be very large.

When α is very large compared to β, the error is computed is as follows:

$$ErrScore = \lim_{\alpha \to \infty} \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta} = \lim_{\alpha \to \infty} \frac{\alpha(d_{HP})}{\alpha} = d_{HP}. \tag{9}$$

When β is very large compared to α, the error is computed is as follows:

$$ErrScore = \lim_{\beta \to \infty} \frac{\alpha(d_{HP}) + \beta(d_{PA})}{\alpha + \beta} = \lim_{\beta \to \infty} \frac{\beta(d_{PA})}{\beta} = d_{PA}. \tag{10}$$

The lower and the upper bounds on RS for the considered prediction models are summarized in Table 3.

Table 3. Summary of maximum and minimum RS values for various prediction models on the breakout dataset

	$\alpha \to \infty$	$\beta \to \infty$
DecisionTable	73.68	73.56
JRip	65.50	68.10
NNge	67.50	69.26
PART	65.79	70.48
ConjunctiveRule	57.13	67.28
Ridor	64.90	66.87

4.4   Testing of RS on Random Output Data

In this test, we study the RS results on the same dataset with randomly generated output. One of the main motivations for this research was the fact that the performance of the prediction model should not be punished for inconsistencies in the data pattern that are caused by factors outside the observed context. For example, a prediction model whose goal is to find the desired lighting conditions for the user can be expected to perform at best as good as the user himself. If the user is generating inconsistent data, i.e. if an observed context maps to different outcomes at different times, then the student (the prediction model) will
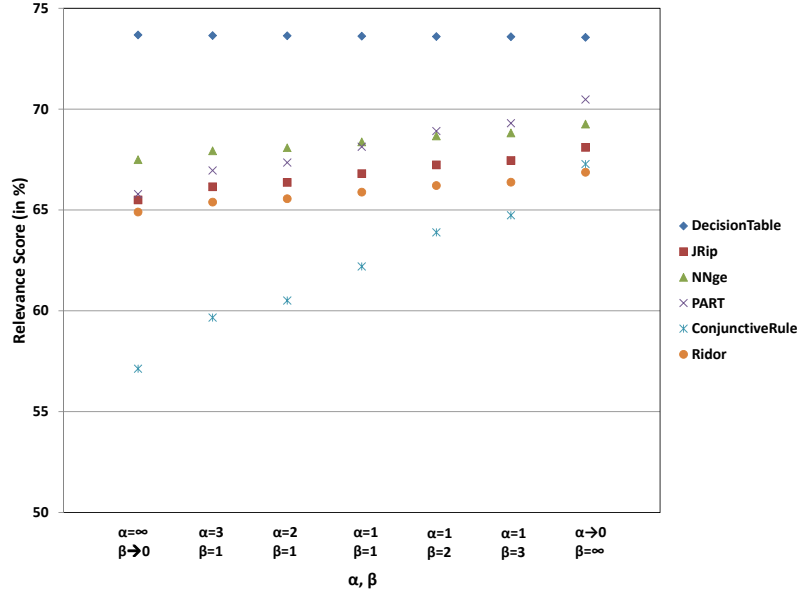
Fig. 5. Relevance Score vs. (α, β)

be learning from an inconsistent teacher (the user), whose reasons for changing mind are completely hidden from the student, i.e. the reasons either don't exist or they are beyond the observed context. Consider the extreme case where there is no relation between the input instances and output lighting conditions, i.e. the user throws an 8-sided dice to select the desired lighting condition and the dice is not part of the observed context. In this case, the CA metric would be equal to 1/8 as there are 8 possible lighting conditions. We also expect to achieve a lower RS in comparison to the scores on the dataset with real output. On the other hand, note that the user makes no distinction between lighting conditions. For a given observed context, even though there is exactly one desired lighting condition, all possible lighting conditions are indeed somewhat relevant. Therefore, we expect that the RS metric will be higher than the CA metric. Fig. 6 shows the plot of RS vs. prediction model for random output data and for the real output data from the breakout dataset.

In the random output dataset, since one of the eight light conditions is chosen at random i.e. with probability 0.125, the output probability distributions are almost uniform for the input instances. This makes it difficult for the prediction models to select the right output choice. Thus all the considered prediction models lie between the range 50 and 60%. But in case of real output dataset, there are some relationships between the input and output causing the output probability distribution to be non-uniform. Thus the prediction model DecisionTable performs the best with a RS of 74%.
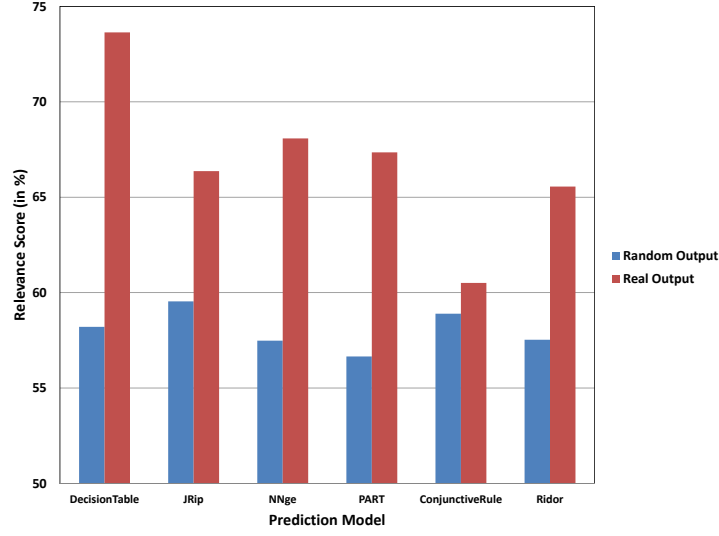
Fig. 6. Relevance Score vs. Prediction Model

From the performed studies, we find that RS is a more suitable metric than the commonly used metrics for the mentioned class of applications. Furthermore, the RS metric provides a provision to select the prediction models that have different characteristics in predicting the outcomes. This depends on the application requirement, and is done by varying parameters α and β. For example, a higher value of α is chosen as compared to β for applications where capturing inconsistencies in the outcome for an observed context is not so critical.

5   Conclusion

The choice of a suitable performance metric is critical to select a machine learning algorithm for intelligent applications. In this direction, we presented the drawback of the commonly used evaluation metric CA with an illustration of intelligent lighting and similar class of applications. We further proposed a new metric named as *Relevance Score* to evaluate machine learning algorithms when there is more than one choice of output class labels suitable for an input instance. The metric is based on the probability values of the outcomes computed from the dataset for a given observed context. We also presented a detailed analysis on the performance and relevance of the proposed metric RS through various experiments on the breakout dataset. From the obtained results, we conclude that RS is a better performance metric for the class of applications where the relevance of predictions is critical rather than the prediction accuracy. Moreover,

variants of RS computation can be used by carefully selecting the parameters of the RS computation function based on the application requirement.

References

1. Gopalakrishna, A. K., Ozcelebi, T., Liotta, A., Lukkien, J. J.: Exploiting Machine Learning for Intelligent Room Lighting Applications. In: The 6th International Conference on Intelligent Systems, pp. 406–411, Sofia (2012)
2. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: AI 2006: Advances in Artificial Intelligence, vol. 4304, pp. 1015–1021, Springer Berlin Heidelberg (2006)
3. Aly, M.: Survey of Multiclass Classification Methods. Technical Report, California Institute of Technology (2005)
4. Witten, I. H., Frank, E., Hall, M. A.: Data Mining: Practical Machine Learning Tools and Techniques. ch. 1, pp. 21–26, Morgan Kaufmann (2005)
5. Tsoumakas, G., Katakis, I.: Multi-Label Classification: An Overview. J. Data Warehousing and Mining 3(3), 1–13 (2007)
6. McCallum, A. K.: Multi-label Text Classification with a Mixture Model trained by EM. In: AAAI '99 Workshop on Text Learning (1999)
7. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-Label Classification of Music into Emotions. In: Proceedings of The International Society for Music Information Retrieval, pp. 325-330 (2008)
8. Japkowicz, N.: Why question machine learning evaluation methods? In: AAAI'06 workshop on evaluation methods for machine learning, pp 6–11. (2006)
9. Jain, R.: The Art of Computer System Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling. John Wiley (1991)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
11. Kohavi, R.: The Power of Decision Tables. In: The Eighth European Conference on Machine Learning, pp. 174-189 (1995)
12. Cohen, W. H.: Fast Effective Rule Induction. In: Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, pp. 115–123 (1995)
13. Martin, B.: Instance-based Learning: Nearest Neighbor with Generalization. Technical Report, University of Waikato (1995)
14. Frank, E., Witten, I. H.: Generating Accurate Rule Sets Without Global Optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 144-151 (1998)
15. Clarke, P., Niblett, T.: The CN2 Rule Induction Algorithm. In: Machine Learning, pp. 261-283 (1989)
16. Gaines, B. R., Compton, P.:Induction of ripple-down rules applied to modeling large databases. Journal of Intelligent Information Systems 5(3), pp 211-228 (1995)