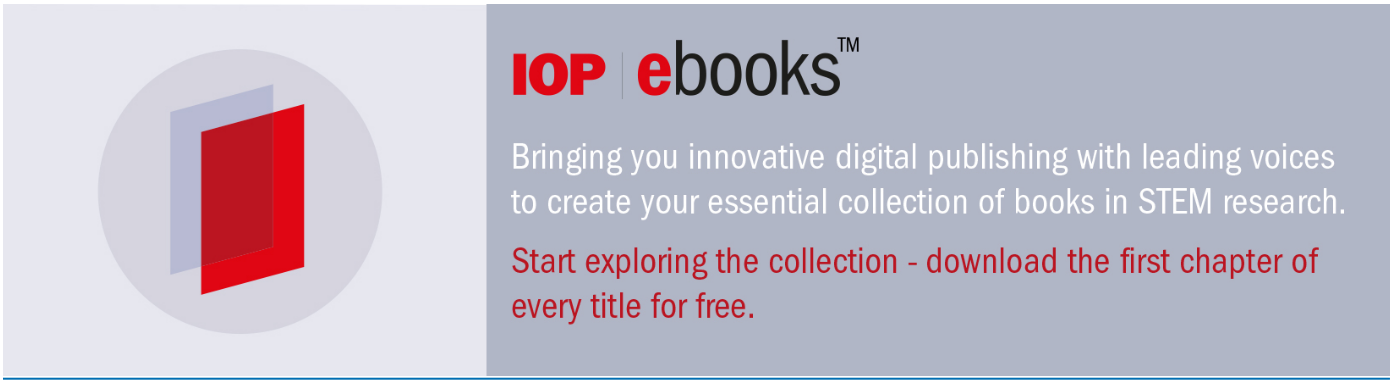
IOP Conference Series: Materials Science and Engineering

Assessment of various supervised learning algorithms using different performance metrics

To cite this article: S M Susheel Kumar *et al* 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 042087



Assessment of various supervised learning algorithms using different performance metrics

S M Susheel Kumar, Deepak Laxkar, Sourav Adhikari and V Vijayarajan
School of Computer Science and Engineering, VIT University, Vellore-632014, India

Email: vijayarajan.v@vit.ac.in

Abstract. Our work brings out comparison based on the performance of supervised machine learning algorithms on a binary classification task. The supervised machine learning algorithms which are taken into consideration in the following work are namely Support Vector Machine(SVM), Decision Tree(DT), K Nearest Neighbour (KNN), Naïve Bayes(NB) and Random Forest(RF). This paper mostly focuses on comparing the performance of above mentioned algorithms on one binary classification task by analysing the Metrics such as Accuracy, F-Measure, G-Measure, Precision, Misclassification Rate, False Positive Rate, True Positive Rate, Specificity, Prevalence.

1. Introduction

Supervised Learning is the machine learning methodology in which we aim to approximate a mapping function to map input values to the target values or output using training data which is already labelled. By learning the association between input and the given correct output, supervised learning will build a model that can predict the output value given input value. Supervised learning methodology can be divided into Regression and Classification problems.

Regression problems are those in which the output is a real valued number such as 'gross revenue' and Classification problems are those in which the output is a category such as 'spam' and 'non-spam'. Following are the major supervised machine learning algorithms: Linear Regression, Logistic Regression, Support Vector Machines, Naïve Bayes, Decision Trees, Linear Discriminant Analysis, K-Nearest Neighbor algorithm, and Neural Networks.

In this paper, we have considered only binary classification problem and for this purpose we evaluated the performance of Support Vector Machines, Decision Tree, K Nearest Neighbor, Naïve Bayes and Random Forest on one binary classification problem with nine performance metrics: Accuracy, F-Measure, G-Measure, Precision, Misclassification Rate, False Positive Rate, True Positive Rate, Specificity and Prevalence.

SVM algorithm is a non-probabilistic machine-learning algorithm which learns to build its model by classifying points in the feature space [1]. In this work, we have used Radial Basis Function.
KNN algorithm which is non-parametric in nature will aim to locate the majority vote in group of the k-closest neighbors but it does not depend on overall data structure and hence does not require training it explicitly [2].

Naive Bayes classifier is based on Bayes theorem and it simplifies learning by assuming that features are independent given each class [3].

Decision Tree is non-parametric based which is aimed to predict label for binary classification by building a tree-structure model. But the problem with Decision Tree is that the tree can grow complicatedly very deep with serious over fitting problems [4].

Random Forest splits each node using best predictor among a subset of predictors randomly chosen at that node and is robust against over fitting [5].

In section 2 we have Literature Survey of the different methodologies, then in section 3 we discuss about experimental setup, dataset used along with required pre-processing of datasets before experiment and the performance metrics to be considered for evaluation. Finally, in section 4 we have experimental analysis followed by conclusion in section 5.

2. Literature Survey

The supervised machine learning algorithms are those algorithms which needs external support in terms of input variables. The dataset is separated into training part and testing part. The output variable or the target variable needs to be predicted or classified depending upon the problem. Every algorithm learns some patterns from the training part of the dataset and applies them to the testing part of dataset to appropriately predict or classify. The various types of supervised learning algorithms are discussed in this section.

The Support Vector Machine is mainly used for classification purposes. It works on the idea of calculation of margin between the classes. These margins are calculated and drawn in such a way that the distance between the margin and the classes is maximum which in turn helps in minimizing the error in classification [6].

In K Nearest Neighbor well labeled training data is fed into the learning module. When the testing data is fed to this module, it compares both the data and the k most correlated data taken from training set. The majority of k value is taken which serves as the new class for the test data [2].

Naïve Bayes is mainly used for clustering and classification purpose. It depends on the conditional probability and Bayes theorem. Using this concept, it creates trees based on their probability of happening. These trees are known as Bayesian Network [7].

Decision trees are those types of trees which groups attribute by sorting them based on their values and are mainly for classification purpose. Each tree consists of nodes and branches. Each node represents attributes in a group that is to be classified and each branch represents a value that the node can take [8].

Random Forests are a combination of tree predictors in which each tree relies on the values of an independent random vector with uniform distribution for all trees in the forest [9].

3. Methodology

This section deals with details regarding supervised learning algorithms and information about the dataset used in this work.

3.1 Supervised Learning Algorithms

3.1.1 Support Vector Machine (SVMs)

We have trained SVM model with Radial Basis Kernel function, cost value is set to 1 and gamma set to 0.125.

3.1.2 K-Nearest Neighbors (KNN)

For KNN algorithm, we have used the "class package provided by R to train the model.
We have set the value of k to the square root of the number of instances in the dataset.
We had 768 instances of the data so we have used k as 27.

3.1.3 Decision Tree (DT)

For Decision Tree Algorithm, we have used the party package provided by R to train the model.

3.1.4 Naïve Bayes (NB)

For Naïve Bayes Algorithm we have used the e1071 and miner packages provided by R to train the model.

3.1.5 Random Forest (RF)

For Random Forest, we have used random Forest package provided by R to train the model. We have used default number of trees as 500 and variables to be tried at each split as 2.

3.2 Performance Metrics

To evaluate the performance of each classifier, we have used these nine metrics: Accuracy, F-Measure, G-Measure, Precision, Misclassification Rate, False Positive Rate, True Positive Rate, Specificity and Prevalence.

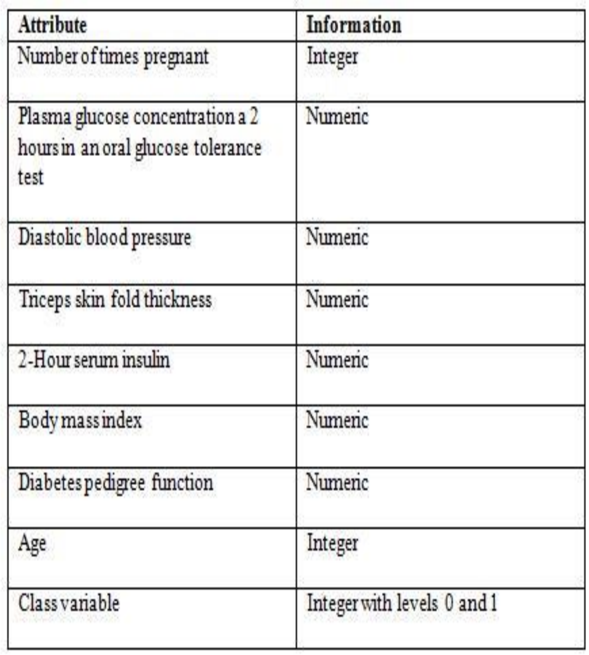
3.3 Data Sets

For this paper, we have used the Pima Indians Diabetes Data Set [10] which is publicly available from UCI repository [11] and compared supervised learning algorithms mentioned above on a binary classification problem.

3.3.1 Pima Indians Diabetes Data Set

The problem is to predict whether a given female is likely to have diabetes or not based on given data. It has 768 instances and 9 attributes including the target variable as shown in Table 1.

Table 1. Pima Indians Diabetes Dataset Attribute Information



Attribute	Information
Number of times pregnant	Integer
Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Numeric
Diastolic blood pressure	Numeric
Triceps skin fold thickness	Numeric
2-Hour serum insulin	Numeric
Body mass index	Numeric
Diabetes pedigree function	Numeric
Age	Integer
Class variable	Integer with levels 0 and 1

We have normalized all the features to rescale them between ranges {0-1}

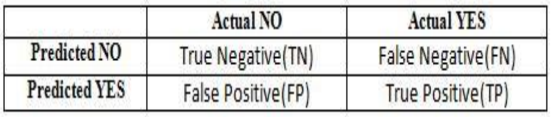
4. EXPERIMENTAL ANALYSIS

We have split our dataset into two parts Training and Testing. Out of total 768 instances, the Training partition contains 70% of the total instances which comes around 538 instances in training partition. And remaining 30%, around 230 instances in testing partition.

4.1 Calculation of Performance Based on Metrics

A Confusion Matrix is a table that is widely used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known as shown in Table 2 [12].

Table 2. Confusion Matrix terminology



	Actual NO	Actual YES
Predicted NO	True Negative(TN)	False Negative(FN)
Predicted YES	False Positive(FP)	True Positive(TP)

4.1.1 Accuracy

It is simply the ratio of correctly predicted observations. It shows overall how often the classifier is correct.

The formula for Accuracy is: (TN+TP)/number of instances (1)

4.1.2 Misclassification Rate

It shows how often the classifier is wrong. It is also called the Error Rate
.
The formula for Misclassification Rate is: (1-Accuracy) (2)

4.1.3 True Positive Rate

It shows if it is actually yes, then how often classifier predicted yes. It is also called as Sensitivity or Recall.

The formula for True Positive Rate is:(TP/FN+TP) (3)

4.1.4 False Positive Rate

It shows if it is actually no, then how often classifier predicted yes.

The formula for False Positive Rate is: FP/TN+FP) (4)

4.1.5 Specificity

It shows if it is actually no, then how often classifier predicted no.

The formula for Specificity is: (1-False Positive Rate) (5)

4.1.6 Precision

It shows when it predicts yes, then how often it is correct.

The formula for Precision is: (TP/FP+TP) (6)

4.1.7 Prevalence

It shows how often the yes actually appears in the sample.

The formula for Prevalence is: (FN+TP/number of instances) (7)

4.1.8 F-Measur

This is a harmonic mean of precision and recall. It is also called as F1-Score or F-Score.

The formula for F-Measure is:
{2*(Recall * Precision)} / (Recall + Precision) (8)

4.1.8 G-Measure

This is a geometric mean of precision and recall. It is also called as G1-Score or G-Score.
The formula for G-Measure is:

$$\sqrt{\text{Recall} * \text{Precision}} \tag{9}$$

Table 3 shows the performances of all the classifiers on a binary classification task. Overall, SVM with radial basis kernel function Random Forest have the best accuracy while Decision Tree has the lowest accuracy. The best classifiers are boldfaced.

Table 3. Performance of classifiers based on Metrics

Performance Metric	SVM-RBF	DT	KNN	Naïve Bayes	Random Forest
Accuracy	0.801	0.756	0.787	0.769	0.808
Misclassification Rate	0.195	0.243	0.212	0.230	0.191
True Positive Rate	0.569	0.392	0.502	0.620	0.594
False Positive Rate	0.072	0.052	0.068	0.152	0.079
Specificity	0.927	0.947	0.931	0.847	0.920
Precision	0.803	0.794	0.784	0.680	0.796
Prevalence	0.343	0.343	0.333	0.343	0.343
F-Measure	0.666	0.525	0.610	0.649	0.681
G-Measure	0.676	0.558	0.626	0.649	0.688

A columnar graph of performance of supervised learning algorithms with metrics is shown in Figure 1

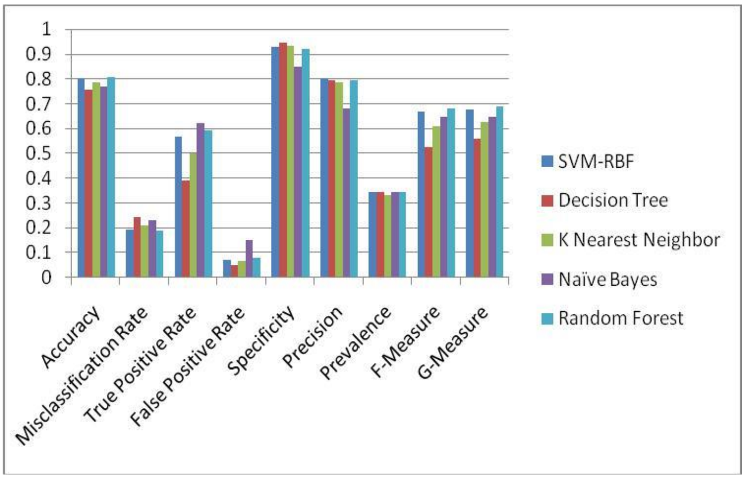


Figure 1. Column Graph of classifiers with performance Metrics

5. Conclusion

Finally, we conclude that though SVM with radial basis function and Random Forest worked efficiently on our dataset but some tuning is required on other classifiers so as to boost their performance. The performance of a classifier is majorly affected by the choice of kernel functions and parameter settings. We should evaluate the performance of a classifier on more than one metric since one performance metric comparison will eventually give biased conclusion.

References

[1] Xuegong, Zhang. "Introduction to statistical learning theory and support vector machines." Acta Automatica Sinica 26.1 (2000): 32-42.

[2] Weinberger, Kilian Q., John Blitzer, and Lawrence Saul. "Distance metric learning for large margin nearest neighbor classification." Advances in neural information processing systems 18 (2006): 1473.

[3] Williams, Nigel, Sebastian Zander, and Grenville Armitage. "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification." ACM SIGCOMM Computer Communication Review 36.5 (2006): 5-16.

[4] Safavian, S. Rasoul, and David Landgrebe. "A survey of decision tree classifier methodology." IEEE transactions on systems, man, and cybernetics 21.3 (1991): 660-674.

[5] Liaw, Andy, and Matthew Wiener. "Classification and regression by random Forest." R news 2.3 (2002): 18-22.

[6] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1-16.

[7] John, George H., and Pat Langley. "Estimating continuous distributions in Bayesian classifiers." Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1995.

[8] Kohavi, Ronny, and J. Ross Quinlan. "Data mining tasks and methods: Classification: decision-tree discovery." Handbook of data mining and knowledge discovery. Oxford University Press, Inc., 2002.

[9] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[10] https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/

[11] Blake, Catherine, and Christopher J. Merz. "{UCI} Repository of machine learning databases." (1998).

[12] http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/