Performance Comparison of Machine Learning Classification Algorithms

K. M. Veena(✉), K. Manjula Shenoy, and K. B. Ajitha Shenoy

Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
{veena.gv, manju.shenoy, ajith.shenoy}@manipal.edu

Abstract. Classification of binary and multi-class datasets to draw meaningful decisions is the key in today's scientific world. Machine learning algorithms are known to effectively classify complex datasets. This paper attempts to study and compare the classification performance if four supervised machine learning classification algorithms, viz., "Classification And Regression Trees, k-Nearest Neighbor, Support Vector Machines and Naive Bayes" to five different types of data sets, viz., mushrooms, page-block, satimage, thyroid and wine. The classification accuracy of each algorithm is evaluated using the 10-fold cross-validation technique. "The Classification And Regression Tree" algorithm is found to give the best classification accuracy.

Keywords: Machine learning · Classification · Datasets · Cross-validation

1 Introduction

The Machine Learning (ML) is the process of preparing systems to perform a specific task automatically. Various ML algorithms have been designed that can learn the characteristics of a specific system, based on experience, and render useful services later on. Machine learning algorithms used to train and test the machines on various data sets. The input data for machine learning algorithms include a set of features and the output is the grouping or ranking of data based on their features. The available data may be classified as training data and testing data. The training data makes the machine learn the task and testing data is used to test the performance of the machine in performing the task. Machine learning tasks include ranking, classification, regression, dimensionality reduction, feature selection, and clustering. The machine learning algorithms may be classified as supervised, unsupervised or semi-supervised. The training data includes target class labels in supervised learning, whereas target class label is not provided in unsupervised learning. Target class label for few input data is available in case of semi-supervised learning.

The classification techniques of ML help in labeling the data sets based on its various distinguishing features or characteristics. Diverse machine learning based classification algorithms exist such as "Classification And Regression Trees (CART), k-Nearest Neighbor (KNN), Support Vector Machines (SVM) Naive Bayes (NB), Neural Networks (NN)", etc. The input features for classification may be binary, continuous or categorical.

M. Singh et al. (Eds.): ICACDS 2018, CCIS 906, pp. 489–497, 2018.
https://doi.org/10.1007/978-981-13-1813-9_49

In this paper, the machine learning classification algorithms namely KNN, CART, NB, and SVM are executed on five different datasets. The performance of each algorithm is evaluated using 10-fold cross-validation procedure.

2 Background

The background focuses on the various machine learning algorithms implemented in this paper. A classification model adjusts its parameters to match its output with the targets. To adjust the model's parameters, a learning algorithm is applied, this occurs in a training phase when the model is being constructed. Many algorithms exist to construct a classification model. Such few algorithms are KNN, CART, NB, and SVM.

The most basic yet effective and efficient non-parametric technique for classification is Nearest Neighbor (NN). NN classifies the unknown data instance based on the known neighboring data points class. In 1967, the k-nearest neighbor technique was proposed [1]. KNN works on the assumption that samples with similar input values are likely to belong to the same class. The class labels of neighboring samples are used to determine label of the new sample. The value of k determines the number of closest neighbors to consider. If k is nine, then nine nearest neighbors of the new sample are considered, in deciding the class label of a new sample using the majority voting method. The new sample will be labeled with that of most of its neighbors. An odd number is chosen as the value of k to break the tie in majority voting. If k value is even number, then an equal number of neighbors may belong to same class. The measure of similarity is found by calculating the distance between samples. Distance measure such as Euclidean distance, Manhattan distance, Hamming distance could be used. KNN can be slow as the distance between the new sample and all samples must be computed to classify new sample [2].

CART is discovered by Breiman et al. [3] in the year 1984. CART technique is used to construct prediction models. The data space is recursively partitioned to obtain the model, the obtained partition can be represented as a decision tree. Whenever the target variable is categorical, binary or nominal, the classification decision tree can be constructed. Whereas when the target variable is continuous the regression decision tree can be constructed. Both the regression and classification binary decision trees can be built using CART algorithm. CART uses Gini index as impurity index, which is a generalization of binomial variance. A sequence of if else bi-division is carried out on the training data [4]. A binary tree is built by making the internal node to hold condition, denoting decision in branches and leaf node to hold class label. The test data is checked against the decision tree branches to decide it's class. Advantages of CART include simple to understand, interpret and visualize. It can handle both numerical and categorical data. Data preparation is easy. Non-linear relationships between parameters do not effect it's performance [5].

Disadvantages of CART are decision tree may create over complex trees that don't generalize the data well. This is known as over-fitting. Decision trees can become unstable because small variations in the data may result in complete different tree generated, known as variance, which needs to be lowered by using methods of bagging

and boosting. Greedy algorithms can't guarantee to return the globally optimal decision tree [6, 7].

NB classification model uses a probabilistic approach for classification. Relationships between input features and class is expressed as probabilities. So, given the input features for a sample the probability for each class is estimated. The class with the highest probability then determines the label for the sample. In addition to using a probabilistic framework for classification the NB classifier also uses the bayes theorem. The application of bayes theorem makes the estimating the probability easier [8]. NB assumes that the input features are statistically independent of one another. For a given class, the value of one feature does not affect the value of any other feature. This independence is over simplified and does not always hold true and so is considered a 'naive' assumption. Naive independence assumption and the use of bayes theorem gives this classification model it's name [9].

NB algorithm works by calculating probabilities and performing some multiplication. So, very simple to implement and probabilities that are needed can be calculated with a single scan of the dataset and stored in a table. Iterative processing of the data is not necessary as with many other machine learning algorithms. So, model building and testing are very fast. Due to independent assumption, the probability of each feature can be independently estimated. This means that feature probabilities can be calculated in parallel. This also means data set size does not have to grow exponentially with the number of features. This avoids many problems associated with higher dimensionality. Nb algorithm does not need lot of data to build the model. The number of parameters scale linearly with the number of features [8].

The independence assumption of NB does not hold true for many cases. However, the NB classifier still tends to perform very well. This is because, even though NB doesn't provide good estimation of correct class, it is sufficient as long as correct class is more probable then any other incorrect class, the correct classification will be reached. The independence assumption also prevents the NB classifier to model the interaction between features which limits it's classification power. The NB classifier has been applied to many real world problems including spam filtering, document classification and sentiment analysis [10].

SVM finds the extreme data points in each class to form a decision boundary, which is also referred as hyper-plane. SVM is a frontier which best segregates two classes using hyper-plane. Unoptimized decision boundary could result in greater misclassifications on new data. Support vectors are vectors which define hyper-planes. The algorithm basically implies that only support vectors are important whereas other training examples are ignorable. The linearly separable classes could be separated using LSVM (linear SVM). For, linearly not separable data non-linear SVM should be used. Non-linear data should be transformed to high-dimensional space to make them linearly separable. The problem with such transformation is that it is computationally expensive. To reduce the transformation's computational cost kernel trick or kernel functions could be used. Kernel function accepts the vectors in original space as input and returns the dot product of the vectors in the feature space. Using a kernel function we can apply a dot product between two vectors so that every point is mapped into high dimensional space through transformation. Some popular kernel functions include polynomial kernel, radial basis function kernel, sigmoid kernel. Choosing a correct

kernel is tricky and it's choice may depend on the task in hand. For any kernel chosen, the kernel parameters need to be decided to achieve good performance [11].

Advantages of SVM include effective in high dimensional space, memory efficient as the subset of training set could be used for testing. Kernel functions could be combined together to achieve complex hyper-plane. Disadvantages of SVM are poor performance when number of features is more than number of samples. SVMs do not directly provide probability estimates. Even though SVM is designed to perform binary classification, it could perform multi-class classification using 1VR [12], 1V1 [13], SimMSVM [14] and other techniques.

3 Methodology

Supervised learning algorithms viz., KNN, CART, NB, SVM are implemented using Python scripting. Their performances are compared using five different data sets available at KEEL dataset repository [15], namely mushrooms, page-block, satimage, thyroid and wine. The characteristics of these data sets are given in Table 1.

Table 1. Properties of input data sets

Data set name	No. of instances	No. of features	Number of classes
Mushrooms	8124	22	2
Page-block	5473	10	5
Satimage	6435	36	7
Thyroid	7200	21	3
Wine	4898	11	11

The mushroom spices dataset is categorized as eatable or deadly. The page-block dataset determines the page block content as graphic, text, picture, horizontal line or vertical line. The satimage dataset use satellite image to identify the soil type.

The thyroid dataset contains people's detail and classifies them as normal, suffering from hyperthyroidism or hypothyroidism. The wine dataset includes the features of white wine and class label quality whose value range from 0 to 10.

The value of k used in KNN is 5. CART uses gini index for impurity calculation and entropy for information gain. The NB uses Gaussian method to find the probability of input features, which uses the following probability density function which is given in Eq. 1.

$$f(x/\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu^2)}{2\sigma^2}\right) \tag{1}$$

SVM implementation uses libsvm. The multiclass classification is achieved using the one-vs-one technique. RBF kernel function is used. Polynomial kernel function degree used is 3.

3.1 Validation

Each of the algorithm's performance is evaluated using 10-fold cross validation procedure. The same random seed is used to ensure that the same splits are used to train and test each of the models. Thus, the models are evaluated in a similar manner.

3.1.1 K-Fold Cross Validation

The problems associated with splitting the dataset into fixed training and testing set is the dilemma in deciding the split ratio giving raise to less accurate results. K-fold cross validation overcomes these problems. It partitions the data set into k bins of equal size. For example, if the data set size is 200 and k is ten then each of the k bins holds 20 data points. Then k number of passes are used to validate the algorithm's performance. In each pass, data points in a single bin are used for testing and data points in remaining $(k - 1)$ bins are used for training. The average performance of k runs is used as the performance validation score. K-cross validation takes more time to compute the performance validation score, but it uses all the data points for training as well as for testing and thus improves the accuracy of validation.

4 Results and Discussion

The performance of the supervised machine learning algorithms is measured using the k-fold cross validation technique, where k is set to 10. The ten different validation measures obtained from 10-fold cross validation technique for each model is depicted using a box and whisker plot. The plot shows the spread of the accuracy scores across each cross-validation fold for each algorithm. The performance accuracy measure of supervised algorithms on five different data sets is shown in the Figs. 1, 2, 3, 4 and 5. The average accuracy of the supervised machine learning algorithms is shown in Table 2.

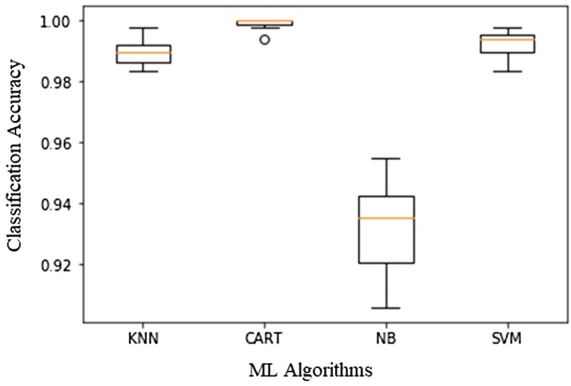

Fig. 1. Performance of machine learning algorithms on mushroom data set

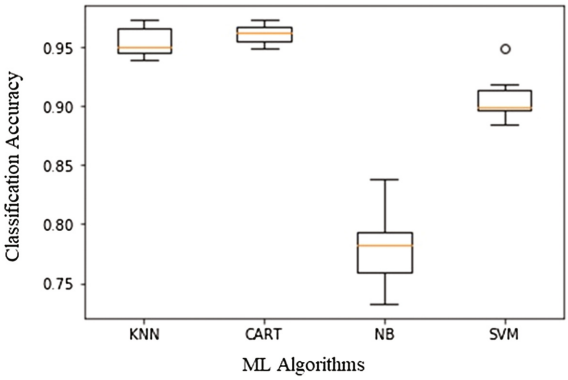

Fig. 2. Performance of machine learning algorithms on page-block data set

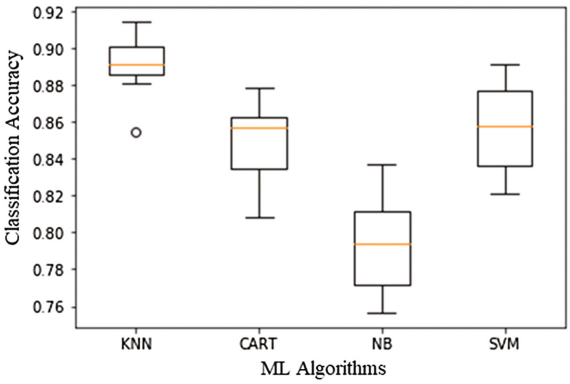

Fig. 3. Performance of machine learning algorithms on a satimage dataset

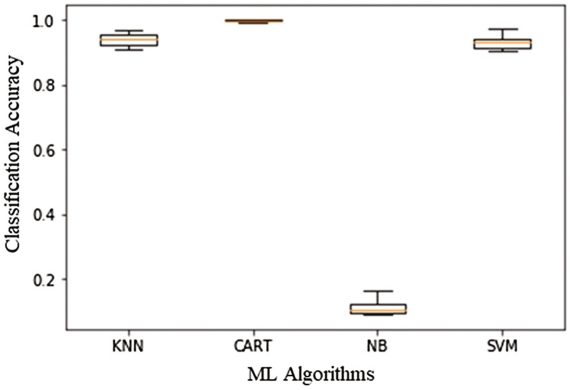

Fig. 4. Performance of machine learning algorithms on thyroid data set

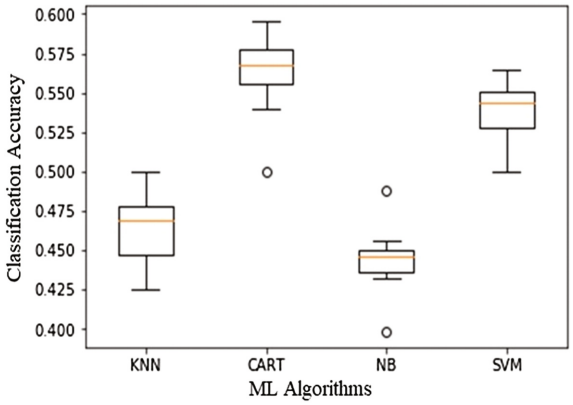

Fig. 5. Performance of machine learning algorithms on wine data set

Table 2. Performance of supervised learning algorithms

Algorithm/data set	Mushroom	Page-block	Satimage	Thyroid	Wine
KNN	98.9%	95.3%	89%	94%	46.5%
CART	99.8%	96.1%	84.7%	99.6%	56.2%
NB	93.2%	77.9%	79.2%	11.1%	44.3%
SVM	99.2%	90.5%	85.5%	93%	53.6%

Figure 6 shows the performance comparison of supervised learning algorithms. CART showed best results on mushroom, page-block, thyroid and wine datasets. KNN has performed best classification on satimage dataset. Mushroom dataset is best

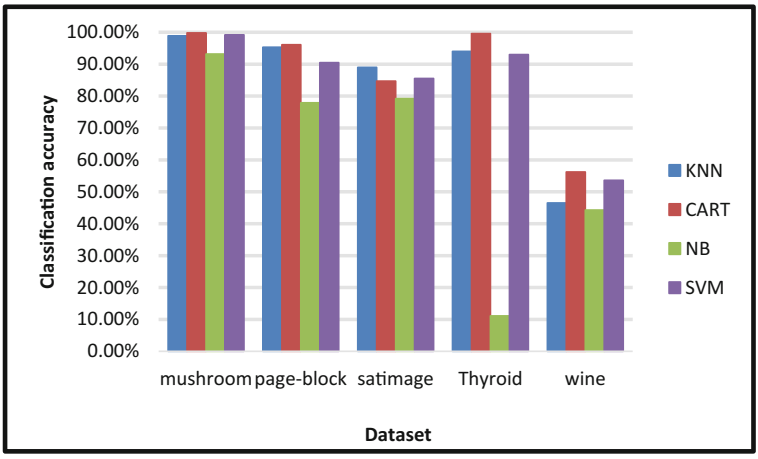

Fig. 6. Performance comparison of supervised learning algorithms

classified by all the classifiers as it has binary class label and class features are distinct across it's classes. Wine dataset is not classified well by any of the used classifiers as most of data values are repeated across its various classes. Thyroid dataset has 15 binary values features which makes NB to misclassify it by achieving only 11.1% classification accuracy. NB after removing 15 binary valued features of thyroid dataset achieved the 94.8% classification accuracy, which shows the ill effect of binary valued features on the NB classifier's performance.

Figure 7 shows the average performance accuracy of classification algorithms. CART has achieved the best results, NB has achieved the worst results. The performance of KNN and SVM are similar. KNN has well performed than SVM on all the datasets except only on wine dataset.

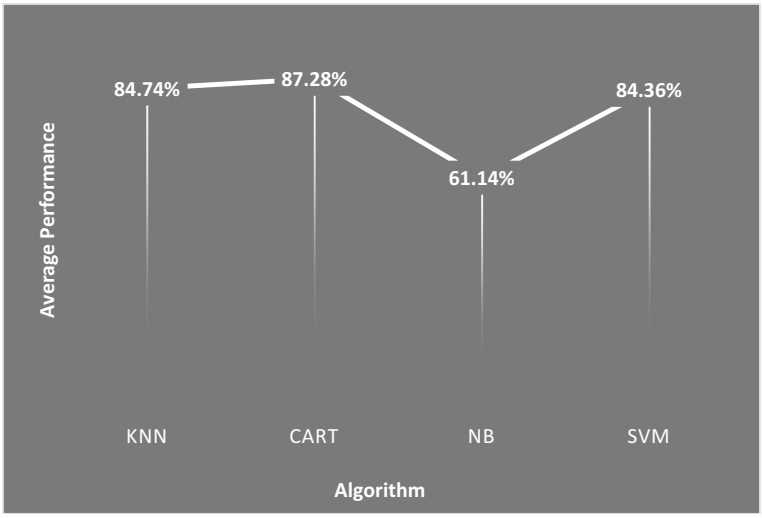

Fig. 7. Average performance of supervised learning algorithms

5 Conclusions

The process of classifying complex datasets can be effectively handled by machine learning algorithms. In this paper four machine learning classification algorithms, viz., KNN, CART, NB and SVM are used to classify five different types of datasets, viz., mushroom, page-block, satimage, thyroid and wine. There performances are compared through their classification accuracies obtained by 10-fold cross validation technique. The CART algorithm is found to perform the classification task the best whereas the NB algorithm performed the worst.

References

1. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
2. Mucherino, A., Papajorgji, P.J., Pardalos, P.M.: k-nearest neighbor classification. In: Mucherino, A., Papajorgji, P.J., Pardalos, P.M. (eds.) Data Mining in Agriculture, pp. 83–106. Springer, New York (2009). https://doi.org/10.1007/978-0-387-88615-2_4
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks, Monterey (1984)
4. Unda-Trillas, E., Rivera-Rovelo, J.: A Method to Build Classification and Regression Trees. In: Bayro-Corrochano, E., Hancock, E. (eds.) CIARP 2014. LNCS, vol. 8827, pp. 448–453. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12568-8_55
5. Trendowicz, A., Jeffery, R.: Classification and Regression Trees, pp. 295–304. Springer, Cham (2014)
6. Berk, R.A.: Classification and regression trees (CART). In: Berk, R.A. (ed.) Statistical Learning from a Regression Perspective, 129–186. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44048-4_3
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, London (2001)
8. Webb, G.I.: Naïve Bayes. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 713–714. Springer, Boston (2011). https://doi.org/10.1007/978-0-387-30164-8_576
9. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: a review of classification techniques (2007)
10. Kirk, M.: Thoughtful Machine Learning: A Test-Driven Approach. O'Reilly Media, Inc., Newton (2014)
11. Ben-Hur, A., Weston, J.: A User's Guide to Support Vector Machines, pp. 223–239. Humana Press, Totowa (2010)
12. Vapnik, V.N., Vapnik, V.: Statistical Learning Theory, vol. 1. Wiley, New York (1998)
13. Kreßel, U.H.-G.: Pairwise classification and support vector machines. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods, pp. 255–268. MIT Press, Cambridge (1999)
14. Wang, Z., Xue, X.: Multi-class support vector machine. In: Ma, Y., Guo, G. (eds.) Support Vector Machines Applications, pp. 23–48. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02300-7_2
15. KEEL dataset repository. http://sci2s.ugr.es/keel/category.php?cat=clas&order=clas#sub2. Accessed 25 Jan 2018