EVALUATING THE PERFORMANCE OF COMMON MACHINE LEARNING CLASSIFIERS USING VARIOUS VALIDATION METHODS

Sharifur Rahman and Pratheepan Yogarajah

School of Computing, Engineering, and Intelligent Systems, Ulster University, Northern Ireland, UK

ABSTRACT

The selection of the proper classifier and the implementation of the proper training strategy have the most impact on the performance of machine learning classifiers. The amount and distribution of data used for training and validation is another crucial aspect of classifier performance. The goal of this study was to identify the optimal combination of classifiers and validation strategies for achieving the highest accuracy rate while testing models with a small dataset. To that end, five primary classifiers were examined with varying proportions of training data and validation procedures. Most of the time, Random Forest and Nave Bayes classifier models outperformed competing classifiers. However, we discovered the best performance when we employed the holdout cross-validation technique using 70% of the available data as a training set and the remaining data as a test set.

KEYWORDS

Machine Learning, Accuracy, Precision, Recall, Cross-validation, Training dataset, F1 score.

1. INTRODUCTION

Data science challenges may be characterized as queries often posed to reach a conclusion or dispel ambiguity while uncovering links between two or more variables. Analyzing the dataset is associated with predicting, classifying, recommending, pattern identification and grouping, irregularity detection and recognition, actionable insights, automated process production via automated decision-making, scoring, rating, and forecasting. Primarily, data science concerns fall into two types. One has a known set of outputs, whereas the other does not have a known set of outcomes. These two challenges need distinct types of machine learning algorithms, supervised and unsupervised, to operate on the dataset. Supervised machine learning may also be used for classification and regression. Classification is used to choose and categorize inputs, while regression is often used to predict results. Unsupervised machine learning clusters inputs into many groups. Some difficulties in data science need domain-specific expertise to comprehend the dataset.

All the experiments and outcomes of the paper came from a wine dataset. With the use of this data set, the authors of this work wanted to compare the performance of models created using various supervised machine learning methods (ML) and various cross-validation techniques with the most suitable finetuning to classify wine into three categories. We will use holdout [7], K-fold cross-validation [8], and leave-one-out cross-validations [9] to check the accuracy of outputs of models prepared with algorithms – K-nearest neighbors, Decision tree, Naïve Bayes, Random Forest, and Support vector machine. Several similar research articles compare various machine

David C. Wyld et al. (Eds): AIAA, DPPR, DSA, ICSS, IOTE, NLPTA, WEST, CACIT, BMLI - 2022
pp. 45-51, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122304

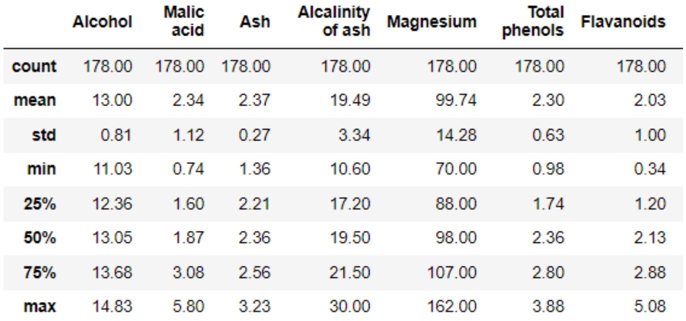
learning (ML) technology implementations. Tougui et al., 2021 [16] investigated the effect of cross-validation method selection on the outcomes of machine learning-based diagnostic applications. Pouriyeh et al., 2017 [17] conducted a thorough examination and comparison of Machine Learning Techniques in the field of cardiovascular illness.

2. Dataset Description

Forina, M. et al., PARVUS - An Extendible Package for Data Exploration, Classification, and Correlation, are the owners of the wine dataset. Brigata Salerno Via, Genoa, 16147, Italy. Institut d'Analyses et de Technologies Pharmaceutiques and Alimentaires Stefan Aeberhard donated it. The wine data came from a chemical analysis of three wines from the same Italian region. The investigation determined the quantity of 13 components (features) observed from each wine varietal. The features are (1) Alcohol, (2) Maliciacid, (3) Ash, (4) Alcalinityiofiash, (5) Magnesium, (6) Totaliphenols, (7) Flavanoids, (8) Nonflavanoidiphenols, (9) Proanthocyanins, (10) Coloriintensity, (11) Hue, (12) OD280/OD315iofidilutediwines, (13) Proline. This is a difficulty with "well-behaved" class hierarchies. The data collection contains 178 records, and three classes do not have the same number of instances. Class A has 59 (33%), class B 71 (40%), and class c (27%) has 48 entries. It is an example of an imbalanced dataset. No null, incomplete, duplicate, or incompatible entry was found in the dataset. Twelve of the thirteen features were represented as float data, while one was represented as integer data.

3. Methodology

A systematic approach to solving a data science problem involves step-by-step activities – 1. Problem statement identification, 2. Data collection, 3. Data cleaning, 4. Data pre-processing, 5. Create a model, 6. Evaluate model performance, 7. Interpret result. After data collection, we used python programming language to investigate data integrity, structure, size, and completeness. In the pre-processing data stage, as the data class was not balanced, WEKA version 3.8.6 was used as a tool to make the data balanced for all three classes. WEKA generated some synthetic data through the synthetic minority over-sampling technique [SMOTE]. WEKA also prepared model preparation, cross-validation, training, and testing model accuracy. Jupiter Notebook 6.3.0 was used as an integrated development environment for Python 3 coding. Python is widely used for Machine learning. Developers do not have to build code from scratch consistently since the Python library supplies essential elements. Continuous data processing is required for machine learning, and Python modules enable retrieval, manipulation, and analyze data. These are some of the most comprehensive AI and ML libraries accessible. For the final model preparation and experiments, WEKA was used.



	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids
count	178.00	178.00	178.00	178.00	178.00	178.00	178.00
mean	13.00	2.34	2.37	19.49	99.74	2.30	2.03
std	0.81	1.12	0.27	3.34	14.28	0.63	1.00
min	11.03	0.74	1.36	10.60	70.00	0.98	0.34
25%	12.36	1.60	2.21	17.20	88.00	1.74	1.20
50%	13.05	1.87	2.36	19.50	98.00	2.36	2.13
75%	13.68	3.08	2.56	21.50	107.00	2.80	2.88
max	14.83	5.80	3.23	30.00	162.00	3.88	5.08

Figure 1. Jupiter Notebook environment

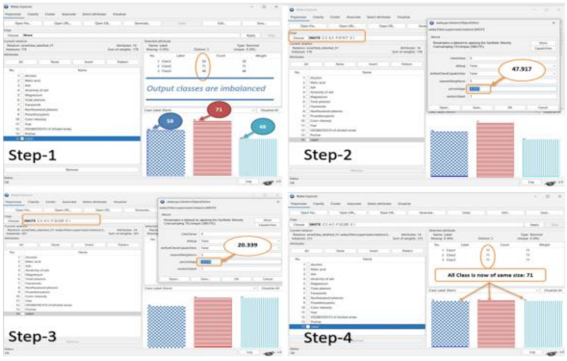

Output classes are imbalanced
59
71
48
Step-1
47.917
Step-2
20.339
Step-3
All Class is now of same size: 71
Step-4

Figure 2. WEKA environment to balance class of data

Figure.1 shows the Jupiter Notebook environment where executing Python code, we got column names of the dataset, and we could figure out the datatypes of the column values. Out of 13 columns, 11 hold float, and two hold integer-type values. The last column is the class definition, and there is no null/missing value. Describe () command confirms again all the columns have values. The Describe () method computes a summary of column statistics for a Data Frame. This function returns the mean, standard deviation, and interquartile range. In addition, the function eliminates character columns and provides a summary of numeric columns. The interquartile range (IQR) is excellent for skewed distributions, as is the median. The mean is average. In a normal distribution, the standard deviation represents the proportion of observations within a range of distances from the mean. Nonetheless, for skewed distributions, the IQR is an excellent option.

Figure. 2 step-1 shows the class imbalance among the three classes. Using the SMOKE function in WEKA, we balanced data classes with 71 instances, with each class accumulating 213 cases of data feeding 35 samples by oversampling in four steps.

4. Modeling

The model preparation stage follows the data pre-processing step. To classify data for our aim of implementing machine learning [ML] with this dataset, we used prominent ML techniques, for example, the k-Nearest Neighbor (KNN) [1], the Decision Tree (DT) [2], the Naive Bayes [3], the Support Vector Machines [SVM] [4], and the Random Forest classifier [RF] [5].To avoid overfitting [6], we employed a cross-validation technique to divide the sample dataset into training and test datasets. We also tested different parameters while partitioning datasets with Holdout [7], K-fold cross-validation [8], and Leave-One-Out cross-validation [9] techniques to identify the most accurate combination for ML classifier and cross-validation techniques.

KNN [1,10] is well-suited to both regression and classification tasks. The distance between training data points determines the test data class. The test data point gets the closest training point, class. The average of K test points is used to calculate regression. DT [2, 11] is best suited for classification. However, it may also be utilized for regression. Decision nodes are used to form multi-branched judgments, while Leaf nodes are the results of such decisions that have no further branches to follow. The evaluations or tests are carried out considering the dataset's characteristics. The Naive Bayes classification [3, 12] applies the Bayes theorem. It is a stochastic classifier, implying it makes estimations based on an item's potential. The Naive Bayes Algorithm is widely used in spam filtering, sentiment classification, and article segmentation. The SVM [4, 13] technique is used for classification and regression. However, this method of solving classification problems is primarily used in Machine Learning. The SVM approach determines the ideal line or decision boundary for classifying each plane to allow future data point additions. A hyperplane is the most acceptable boundary. SVM selects the hyperplane's extreme points/vectors. These extreme situations are referred to as support vectors. RF [5,14] is applied to classification and regression problems. To increase the projected accuracy of a dataset, RF employs many decision trees on multiple subsets of a particular dataset and then averages the results. It collects projections from each tree and forecasts the final output depending on which predictions are most popular among the participants. The greater the number of trees in the forest, the more exact the model and the lower the risk of overfitting the data.

In Holdout [7] cross-validation, the dataset is randomly divided into two separate sets, namely the practice set, and the examination set. The holdout approach is used in many big datasets. The dataset is partitioned into k equal subgroups in k-fold cross-validation [8]. For training, k subsets are employed, whereas just one set is used for testing. The method is performed k times (k folds), with each k subset being tested precisely once. The k estimates (accuracy) from the k folds are averaged to provide the final estimated value. This technique is suitable for moderately sized datasets. Leave-One-Out [9] cross-validation is performed on a set of N experiments with N observations. Each experiment utilizes N -1 samples for training and just 1 sample for testing. Lastly, it calculates the total performance after N experiments. This technique is appropriate for small-size datasets.

In our study, we used an identical dataset and put each of the five classifiers stated to the test individually. As a first step, we validated the classifiers using holdout techniques. Here, we increased the test data ratio from 10 to 90 by dividing the total data into distinct halves (10:90 to 90:10). Following that, we tested the K-fold validation procedure on each classifier. Here, we began with a 2-fold validation and went up to a 100-fold validation. Our most recent experiment tested leave-one-out cross-validation using 213-fold. After each test case, memory was cleaned.

5. Experimental Results

In WEKA, five classifier models were prepared, run, and visualized with default settings on 213 records, including synthetic data. Later all the classifier models were cross-validated with three validation techniques with various parameter tuning. In each case, the result is recorded for comparison.

We utilized Holdout cross-validation for all classifier models, splitting the dataset into 10:90, 20:80, 30:70, 40:60, 50:50, 60:40, 70:30, 80:20, and 90:10 ratios for training data vs. test data, Table 1 contains the outcomes of the study.

Table 1. Classifier Performance for Holdout

Algorithm	**Holdout validation (training data proportion) - Correctly Classified Instances (%)**									
	10%	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**Max**
K-nearest neighbor	93.23	94.12	96.64	96.88	97.17	97.65	**98.44**	97.67	95.24	98.44
Decision tree	65.10	91.76	93.96	92.97	92.45	95.29	96.88	97.67	**100**	100.00
NaïveBayes	92.19	97.06	97.32	98.44	98.11	97.65	**100**	**100**	**100**	100.00
Support vector machine (SVM)	34.38	40.00	42.28	50.00	40.57	47.06	50.00	44.19	**61.90**	61.90
Random forest	84.90	95.29	97.99	98.44	98.11	98.82	**100**	**100**	**100**	100.00

We used 2,5,10,15,20,50,100 folds for all classifiers and 213 folds for Leave-One-Out cross-validation to get the results shown in Table 2.

Table 2. Classifier Performance For K-Fold Cross-Validation and Leave-One-Out Cross-Validation

Algorithm	**K - Fold validation (fold number) - Correctly Classified Instances (%)**								
	2	**5**	**10**	**15**	**20**	**50**	**100**	**213**	**Max**
K-nearest neighbor	**96.71**	94.84	95.31	95.31	95.31	95.31	95.31	95.31	96.71
Decision tree	93.90	93.43	**94.84**	93.90	**94.84**	94.37	**94.84**	**94.84**	94.84
NaïveBayes	97.18	**98.12**	**98.12**	**98.12**	**98.12**	**98.12**	**98.12**	**98.12**	98.12
Support vector machine (SVM)	50.23	60.56	58.22	58.69	**61.03**	54.93	53.05	41.78	61.03
Random forest	98.59	97.65	**99.06**	97.65	97.65	97.65	98.12	98.12	99.06

6. EVALUATION RESULTS

The performance of the classifiers was evaluated using all available features. To keep things simple, we did not use feature prioritization or selection algorithms for this experiment. According to Tables 1 and 2, cross-validation works best for 10-fold in K-fold cross-validation, which is 10% data in each fold and 70% in the training set for Holdout cross-validation. In several circumstances, Leave-One-Out fared poorly compared to 10-fold K-fold data validation. Using the best possible combination of cross-validation performance for all five classifier models, we discovered the data presented in Table 3 below.

Table 3. Classifier Performance Table for Best Possible Cross-Validation

Algorithm	**Performance table for correctly classifying test data**			
	Holdout (30% for the test)	**K-Fold (10-fold)**	**Leave-one-out**	**Average**
K-nearest neighbour	98.44	95.31	95.31	96.35
Decision tree	96.88	94.84	94.84	95.52
NaïveBayes	100.00	98.12	98.12	98.75
Support vector machine (SVM)	50.00	58.22	41.78	50.00
Random forest	100.00	99.06	98.12	**99.06**

The accuracy score, confusion matrix, precision, recall, and F1 score are commonly used to assess the performance of ML classifier models, with the kind of dataset playing an essential role in determining which metric is suited to evaluate model performance. Precision is the likelihood that an observation is positive when a classifier predicts it to be positive; recall represents the likelihood that a positive observation will be recognized, and the mean of precision and recall is F1. In this example, the wine dataset is a type of dataset that allows us to make decisions without bias. The F1 score is better for assessing model performance for this dataset. Table 4 displays the assessment score of the classifier models. The random forest and the Naive Bayes classifier models were determined to have the best fit for this dataset.

Table 4. Assessment Score Of Classifier Models

Algorithm	Cross-validation	Precision	Recall	F1 Score
K-nearest neighbour	Holdout (30% for the test)	99%	98%	98%
Decision tree		97%	97%	97%
NaïveBayes		**100%**	**100%**	**100%**
Support vector machine (SVM)		81%	50%	47%
Random forest		**100%**	**100%**	**100%**
K-nearest neighbour	K-Fold (10-fold)	96%	95%	95%
Decision tree		95%	95%	95%
NaïveBayes		98%	98%	98%
Support vector machine (SVM)		66%	58%	58%
Random forest		99%	99%	99%
K-nearest neighbour	Leave-one-out	96%	95%	95%
Decision tree		95%	95%	95%
NaïveBayes		98%	98%	98%
Support vector machine (SVM)		54%	42%	42%
Random forest		98%	98%	98%

7. Conclusions and Recommendations for Future Work

In this experiment, we utilized the five most popular classification algorithms, each producing output distinctly. K-nearest neighbor retains all available samples and categorizes new data or instances by similarity. The operation of the Decision tree algorithm is determined by the conditions of the characteristics. NaïveBayes is a Bayes Theorem-based probabilistic machine learning algorithm. The objective of the SVM algorithm is to locate a hyperplane in an N-dimensional space that distinguishably classifies the data points. The random forest algorithm collects data randomly, creates a decision tree, and averages the outputs. It does not utilize formulas, unlike Decision trees. As classifiers operate differently, their accuracy rates vary based on the variety of the data. The amount of training data in the data set has a significant impact on the decision tree, random forest, and SVM's performance. When using the holdout validation technique, DT and RF need to utilize at least 30% of the data as the training set to achieve acceptable output accuracy; however, when using the K-fold validation technique, using more than 10 folds is not necessary because performance improvement stops after this point. According to our analysis, the Random Forest and NaïveBayes classifier models outperformed the competition. We discovered that the Holdout strategy performed better in the cross-validation phase, often using 70% of the dataset as training data. Future research on numerous datasets with different quantities of characteristics and observations is still possible to further our understanding.

References

[1] Patrick, E. A., and Fischer, F. P., III (1970) "A generalized k-nearest neighbor rule," Information and control, 16(2), pp. 128–152. DOI: 10.1016/s0019-9958(70)90081-1.

[2] Swain, P. H., and Hauska, H. (1977) "The decision tree classifier: Design and potential," IEEE transactions on geoscience electronics, 15(3), pp. 142–147. DOI: 10.1109/tge.1977.6498972.

[3] Zheng, Z. (1998) "Naive Bayesian classifier committees," in Machine Learning: ECML-98. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 196–207.

[4] Burbidge, R. and Buxton, B., 2001. An introduction to support vector machines for data mining. Keynote papers, young OR12, pp.3-15.

[5] Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.

[6] Ying, X. (2019) "An Overview of Overfitting and its Solutions," Journal of physics. Conference series, 1168, p. 022022. DOI: 10.1088/1742-6596/1168/2/022022.

[7] Hawkins, D. M., Basak, S. C. and Mills, D. (2003) "Assessing model fit by cross-validation," Journal of chemical information and computer sciences, 43(2), pp. 579–586. DOI: 10.1021/ci025626i.

[8] Yadav, S. and Shukla, S. (2016) "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in 2016 IEEE 6th International Conference on Advanced Computing (IACC). IEEE, pp. 78–83.

[9] Cawley, G. C., and Talbot, N. L. C. (2003) "Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers," Pattern Recognition, 36(11), pp. 2585–2592. DOI: 10.1016/s0031-3203(03)00136-5.

[10] Christopher, A. (2021) K-nearest neighbor, The Startup. Available at: https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4 (Accessed: April 9, 2022).

[11] Decision tree classification algorithm (no date) www.javatpoint.com. Available at: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (Accessed: April 9, 2022).

[12] Naive Bayes Classifier in Machine Learning - javatpoint (no date) www.javatpoint.com. Available at: https://www.javatpoint.com/machine-learning-naive-bayes-classifier (Accessed: April 9, 2022).

[13] Support Vector Machine (SVM) Algorithm - javatpoint (no date) www.javatpoint.com. Available at: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (Accessed: April 9, 2022).

[14] Random Forest algorithm (no date) www.javatpoint.com. Available at: https://www.javatpoint.com/machine-learning-random-forest-algorithm (Accessed: April 9, 2022).

[15] Goutte, C. and Gaussier, E. (2005) "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 345–359.

[16] Tougui, I., Jilbab, A., & Mhamdi, J. E. (2021). Impact of the choice of cross-validation techniques on the results of machine learning-based diagnostic applications. Healthcare Informatics Research, 27(3), 189–199. https://doi.org/10.4258/hir.2021.27.3.189

AUTHORS

Dr. Pratheepan Yogarajah, Since January 2016, he has been a Lecturer in Computing Science at Ulster University. Yogarajah earned a Bachelor of Science in Computer Science from the University of Jaffna in Sri Lanka in 2001 and a Master of Philosophy in Computer Vision from Oxford Brookes University in the United Kingdom in 2006. In 2015, he received his Ph.D. from Ulster University in the United Kingdom. He is a member of both the British Computer Society and the IEEE. In 2005, Yogarajah was awarded an Oxford Brookes university HMGCC scholarship. In addition, he was a co-winner of the Proof of Principle (PoP) award from Ulster University in 2012 and the Proof of Concept (PoC) award from Invest Northern Ireland (Invest NI) in 2013. Biometrics, computer vision, image processing, steganography and digital watermarking, and machine learning are among his research interests.

Md. Sharifur Rahman, He is presently undertaking research on Data Science at Ulster University, UK. Before this, he earned master's and bachelor's degrees in computer science and engineering along with the Vice Chancellor's award. Moreover, he earned an MBA degree. He's interested in data mining and modeling for financial and medical research. After sixteen years working in the telecom industry, he was inspired to delve even further into data analysis to connect theoretical understanding with real-world applications to come up with some novel ideas that would be beneficial to both businesses and society.