Research Article

Machine Learning Algorithms Evaluation Methods by Utilizing R

Hozan K. Hamarashid [1,a], Shko M. Qader[1,2,b,*], Soran A. Saeed [3], Bryar A. Hassan [4,c,*], Nzar A. Ali [5]

[1] Information Technology Department, Computer Science Institute, Sulaimani Polytechnic University, Sulaymaniyah, Kurdistan Region, Iraq
[2] Information Technology Department, University College of Goizha, Sulaimani, 46001, Iraq
[3] Higher Education and Scientific Affairs Department, Sulaimani Polytechnic University, Sulaymaniyah, Kurdistan Region, Iraq
[4] Department of Information Technology, Kurdistan Institution for Strategic Studies and Scientific Research, Sulaimani, Kurdistan Region, Iraq
[5] Department of Computer Science, College of Science, Cihan Univerisity of Sulaimaniya, Sulaymaniyah, Kurdistan Region, Iraq

[a] hozan.khalid@spu.edu.iq, [b] shko.qader@spu.edu.iq, [c] bryar.hassan@kissr.edu.krd

Access this article online		
Received on: 02 May 2021	**Accepted on:** 17 December 2021	**Published on:** 30 June 2022
DOI: 10.25079/ukhjse.v6n1y2022.pp1-11	**E-ISSN:** 2520-7792	

Abstract
Machine Learning (ML) is a part of Artificial intelligence (AI) that designs and produces systems, which is capable of developing and learning from experiences automatically without making them programmable. ML concentrates on the computer program improvement, which has the ability to access and utilize data for learning from itself. There are different algorithms in ML field, but the most important questions that arise are: Which technique should be utilized on a dataset? and How to investigate ML algorithm? This paper presents the answer for the mentioned questions. Besides, investigation and checking algorithms for a data set will be addressed. In addition, it illustrates choosing the provided test options and metrics assessment. Finally, researchers will be able to conduct this research work on their datasets to select an appropriate model for their datasets.

Keywords: Artificial Intelligence, Machine Learning algorithms, Machine Learning Metric Evaluation, Machine Learning Test Options, R programming.

1. Introduction
To date, various technologies have been developed, especially in artificial intelligence (AI). One of the most common field in AI is Machine Learning (ML). ML is one of the most important components in AI (Smola and Vishwanathan, 2010). The idea of ML is lies in the fact that which a computer program is able to learn and develop that leads to construction of new data without interference by human beings (Mujtaba, 2020; Hamarashid, 2021). Various measurements are included to collect, pre-process, and contribute valuable information from datasets. ML, which is an important feature of AI, is utilized for processing massive data. Various ML applications are designed via complicated algorithms that are built in with computer programs. This program constructs a model that describes the dataset. In

addition, the model utilizes parameters inside the algorithms, in which a decision-making procedure can be designed and constructed. Thus, ML is utilized in various fields, for instance, e-commerce, marketing, etc. which has the capability to provide significant suggestions and recommend users demands accurately relying on their history searches and earlier transactions (Kumar, 2018). Depending on this concept, ML could be involved to predict stock price, loan, etc. Besides, it could be used in fraud detection for bank systems. Therefore, ML could be incorporated in a broader area including business, government sectors, etc.

There are various types of ML algorithms, which are supervised, unsupervised, semi-supervised and reinforcement categories.

1.1. Supervised ML Algorithms

In this type, the algorithm could be applied to what has learned from experience to a new data by utilizing a labelled class, for instance, predict future events. This could start by analysing trained dataset. Then, the ML algorithm derives a method to construct predictions on the outcome values. Besides, after adequate training, it has the ability to supply a target to a new input. In addition, the algorithm has the capability to compare obtained results with the correct, expected results and discover mistakes or errors to alter the model appropriately (Expert System Team, 2020; Hamarashid et al, 2021).

1.2. Unsupervised ML Algorithms

This type of ML algorithms is utilized in a situation where the utilized data in training set is not classified or labelled. Unsupervised ML deals with the systems that are able to conclude a method to explain a hidden part from unlabelled data. In this type of ML, the system does not appraise the correct result, but it discovers the data which enables the deduction or assumption to be picked from the data to illustrate the hidden part or feature from unlabelled data (Hamarashid et al, 2021).

1.3. Semi-Supervised ML Algorithms

This category could be in between Supervised and Unsupervised ML types, because Semi-Supervised could utilize both Supervised and Unsupervised. In other words, this means labelled and unlabelled learning during training data, especially utilizing a small or limited amount of data in labelled class, in addition to a huge amount of data in unlabelled data. In this type of ML, learning accuracy could be significantly improved during designing the systems. Put it another way, these types of systems have the ability to increase the accuracy of learning significantly. In Semi-Supervised learning, to achieve or obtain unlabelled data of relevant and associated resources are not basically required. In contrast, to obtain labelled data, associated resources and additional resources are required. This is conducted in order to train/test data and learn from the data (Reddy et al, 2018; Van Engelen and Hoos, 2020).

1.4. Reinforcement ML Algorithms

This type of ML has a connection or related with its environment by construction functions and discovering rewards or errors. Characteristics of reinforcement learning contains delayed rewards and searching for trial and errors. In this technique, the machines are permitted to discover the optimal behaviour with determined content; the purpose is to maximize performance. To learn which function or method is the optimal/best value, the reward feedback is required for the machine (François-Lavet et al, 2018; Mahesh, 2020).

Figure 1 shows types of machine learning algorithms.

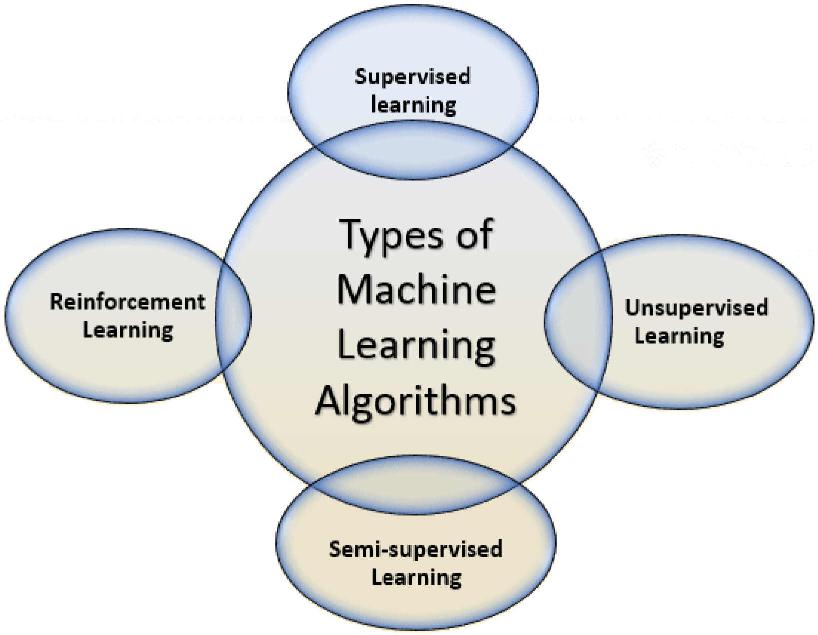

Figure 1. Types of ML Algorithms (EDUCBA, 2020)

One measure for assessing the techniques is accuracy. Accuracy refers to the percentage of correct predictions made by the technique or model. The following is the formal definition of accuracy:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{total\ number\ of\ predictions} \qquad (1)$$

On the other hand, accuracy is calculated based on the following formula to calculate accuracy in terms of positives and negatives for binary data and classification:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (2)$$

Where TP stands for True Positives, TN stands for True Negatives, FP stands for False Positives, and FN stands for False Negatives.

This research paper consists of various features that are addressed in the following sections. The rest of the paper is concerned with determining or selecting appropriate algorithm for a problem. After that, utilizing trial and error to select an algorithm is discussed. Then, Spot-check algorithm is illustrated. Later, test options, building models and selecting models are represented respectively.

2. Methodology

Selecting an appropriate algorithm for a problem is an important issue by researchers. All the researchers seek the best and the most appropriate model for their dataset. This is the main aim of all researchers during their studies, especially for the predictive models. On the other hand, there is no researcher can determine which model is suitable or the best for a provided or determined dataset for the purpose of getting the best outcome. However, sometimes because of having a deep knowledge for a difficulty by the researchers, they know which algorithm could give the best outcome only for a specific dataset. At this time, ML is not the first place or choice. Otherwise, the researchers do not know an appropriate learning algorithm to be utilized for solving a problem. In addition, they do not know which parameters are the best to be utilized with an algorithm to solve a problem. There are different ways to support handling this difficulty. Figure 2 represents the research paper methodology to select the most appropriate model.

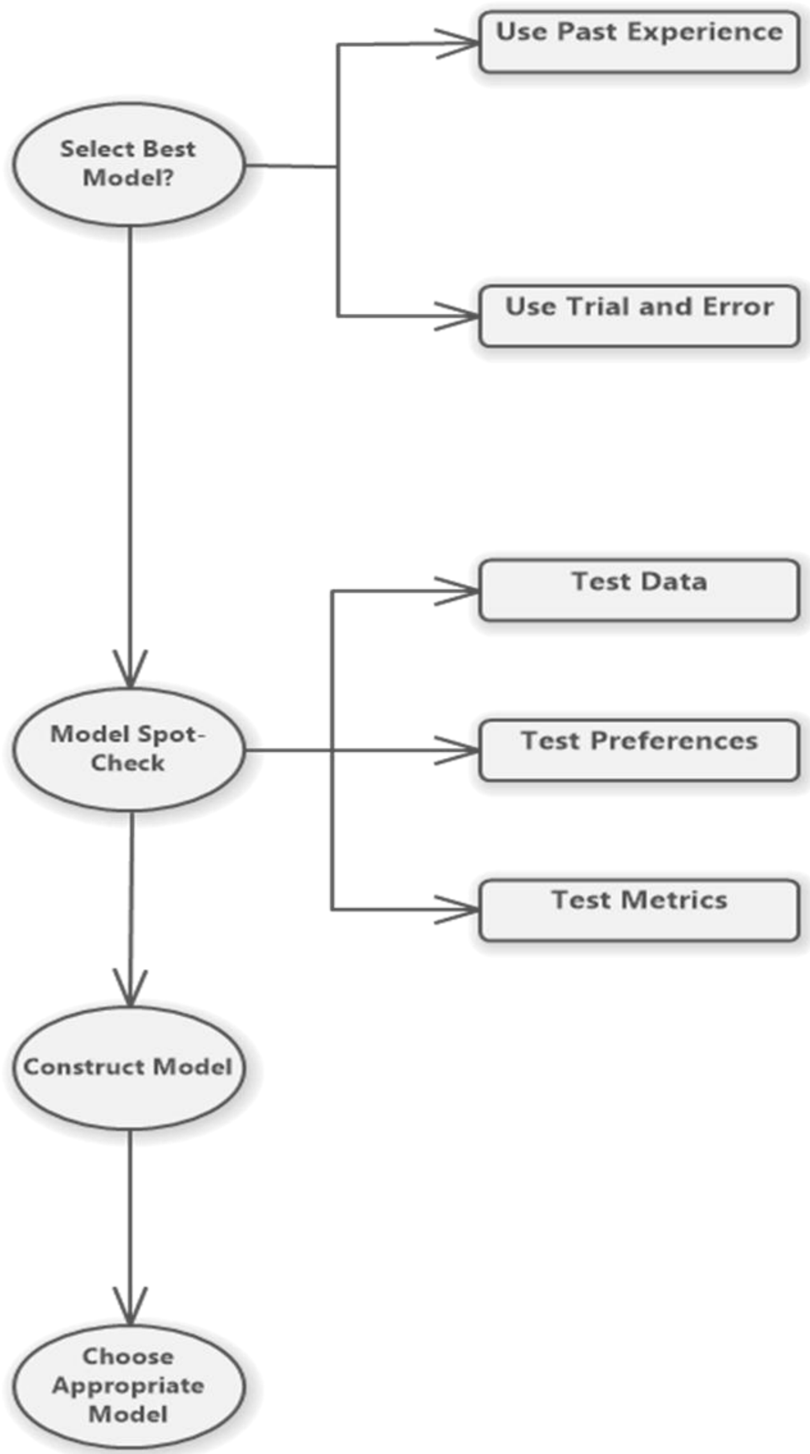

Figure 2. The Research Work Methodology.

2.1. Utilize Past Experience

Experience is an option that researchers can depend on to choose an algorithm for handling a problem. This can include the research works that have previously examined similar problems. It might include several experiences collected in a determined field by conducting literature review on the related research papers, books, and other resources to find out the idea of which algorithms fits or obtain the best results considering the difficulty arisen. This is an important point to start but not the point to stop on. Thus, trial and error finding could be utilized.

2.2. Utilize Trial and Error

Utilizing trial and error is another method that could be used to discover a better algorithm or appropriate algorithm for a provided dataset. Accordingly, various set of algorithms on the provided dataset is assessed to discover which algorithm is appropriate and what algorithm is not appropriate for the determined dataset. This can be named as spot-check for the algorithms. To conduct this task, a brief list of algorithms is available as a suitable algorithm for the determined problem, and the researcher should be good at selecting the appropriate algorithm to figure out the problem. Therefore, the brief list of algorithms should be concentrated on to choose the best and most appropriate algorithm. Moreover, these algorithms could be improved by altering the algorithm parameters or tuning them. On the other hand, the algorithms could be developed by hybridizing multiple algorithms such as mixing predictive algorithms, and ensemble techniques could be utilized.

3. Algorithms spot-check

This part represents a case study to assess appropriate algorithms for a testing purpose by utilizing R Programming. The utilized test problem in this case study contains a class variable to classify the dataset. The dataset is about diabetes on female patients. The dataset consists of bio-information records for the patients with the class variable. The class variable records are binary or Boolean value. The class variable results reveals that the patient has started diabetes or not, based on earlier medical assessment. Three phases can be conducted to do the task; test on the dataset, then, creating multi-prediction model depending on the data; After that, techniques are compared to choose a brief list of algorithms.

3.1. Testing Data

The first step is about testing the data that includes three points: first, the dataset that is utilized for training the model; second, the utilized test options to assess the technique like re-sampling technique; last, the fascinating metrics to measure and compare.

In the utilized data, the problem should be represented to spot-check algorithm. Besides, the entire data should not be included. The task of spot-checking algorithms must be quick. If there is a huge amount of data, then to train it, more time is needed to compute with the algorithm that desired to check. Checking or training is different from a dataset to another, but it is recommended to check, for example, 15000 instances. This may take few minutes to process. In contrast, for a massive amount of data, a random instance from the huge dataset should be taken. Thus, it can be found out later that the taken sample instances affect a brief list of selected algorithms. Therefore, to load the data to the program the following steps was done:

```
# import data
data(biomedicaldata)
biodata <- biomedicaldata
```

3.2. Test Preferences

After the data is loaded, test preferences should be considered. Test choices are associated with the algorithms that are utilized to conduct the unseen data accuracy in a model. Re-sampling technique in statistics is an example. Test choices are as follows:

1. Splitting train and test data: if the data is huge, then a massive amount of data is needed to make the model accurate.
2. Cross validation: a frequently utilized agreement of pace, computation time, and generalize error assessment by using 5 to 10 folds.
3. Cross validation repeat: if a small amount of data is provided, by utilizing 5 to 10 folds and repeating it, three times to provide more strong estimation. To conduct this task, the following code snippet was used:

```
seed <- 5
ctl <- trainCtl(method="repeatedcv", number=10, repeats=3)
```

A random number of seed variable was set in order to be re-set before each algorithm was trained. It is crucial to be assure that every algorithm was assessed on the same split or separated data. Later, this process permits true to true comparison rigorously.

3.3. Test Metrics

The test metric was used to evaluate the model. There are various available metrics to be selected to evaluate the model. Several significant test metrics that can be utilized in various difficulty types are as follows:

- Kappa: it is easily understandable which takes the class distribution as a base.

$$Kappa = \frac{P_0 - P_e}{1 - P_e} \tag{3}$$

$$P_0 = \frac{number\ in\ agreement}{total} \tag{4}$$

$$P_e = P_{correct} + P_{incorrect} \tag{5}$$

- Accuracy: correct prediction is divided by total prediction. This is the most common metric that is utilized to assess a model.

$$Accuracy = \frac{x\ correct}{y\ total\ instance} \tag{6}$$

- Goodness of fit: It is determination coefficient.
- RMSE: Root mean square error is another common metric, which depends on discovering mean square error then the root of MSE will provide the RMSE value.

$$RMSE = \sqrt{MSE} \tag{7}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2}{n}} \tag{8}$$

Therefore, the following code snippet reveals how to determine the metric:

```
Metric <- "Accuracy"
```

4. Constructing Models

To spot-check algorithms during denominating methods, three points should be taken into consideration. First, which methods should be selected? Second, how to arrange and construct methods parameters? Last, data pre-processing for the algorithm.

According to algorithms, it is crucial to have a combination of algorithms. There are different methods that can be utilized as shown below:

- Linear algorithm: It is used with logistic regression and linear inequity analysis.
- Non-linear algorithm: It includes Naïve Bayes, support vector machine, K-nearest neighbour, and neural network, etc. algorithms.
- Rules and Tree: PART, J48, and CART are examples algorithms.
- Tree ensembles: stochastic gradient boosting, random forest, C5.0, and bagged CART algorithms.

The complicatedness of algorithms is different. If the algorithms with less sophisticated is desired, then KNN as an example is a choice. Besides, the algorithm can be developed. On the other hand, if an algorithm with more complicatedness is desired, then random forest RF is a choice to figure out if the difficulty could be solved and also to start making accuracy anticipations.

For configuring algorithms, the entire ML algorithms have nearly parameterized. Thus, the algorithms parameters argument are required to be specified. In addition, the algorithms that are heuristic can be utilized to provide the first past algorithm configuration to begin. Therefore, during spot-checking, it is not appropriate to attempt different parameters in algorithms which appear during the developing consequences. To do this, R package under name Caret is better to be utilized because it supports tuning parameters of algorithms.

Data pre-processing is another point that several algorithms execute the entire data pre-processing instead of performing fundamental pre-processing. To provide equality in chances of selecting algorithms, it is vitally important that necessary pre-processing is included in the training data for those algorithms that are mandatory to be conducted. For instance, those algorithms that depend on many instances perform better, where the entire input attribute have the same scale. To conduct pre-processing, the train () method inside Caret package was used. This allows to determine pre-processing the data to conduct preceding to training. The necessary changes are provided to the pre-processing parameters argument list, and they are performed sequentially on the data according to (Wu et al, 2008). These algorithms are the most popular algorithms to be conducted in data mining and pre-processing.

1. C4.5: it is a decision tree algorithm that contains ancestor methods for instance C5.0 and ID3 algorithms.
2. K-Means: it is a clustering algorithm.
3. SVM: it is mostly utilized in classification.
4. Apriori: It is used for rule extraction.
5. EM: It is a clustering algorithm.
6. AdaBoost: it is for ensemble methods.
7. KNN: it is an effective technique that is instance-based technique.
8. Naive Bayes: it utilizes Bayes theorem.
9. CART: it is working on trees of classification and regression.

This can be done by utilizing the following code snippet:

```
Pre-Processing=c("center", "scale")
```

To execute spot-checking for algorithms, the following code snippet is performed:

```
# Linear Analysis
set.seed(seed)
fit.lda <- train(biomedicaldata~., data=biodata, method="lda", metric=metric, preProc=c("center", "scale"),
trControl=control)
# Log Regression
set.seed(seed)
fit.glm <- train(biomedicaldata ~., data= biodata, method="glm", metric=metric, trControl=control)
# GLMNET
set.seed(seed)
fit.glmnet <- train(biomedicaldata ~., data= biodata, method="glmnet", metric=metric, preProc=c("center",
"scale"), trControl=control)
# SVM
set.seed(seed)
fit.svmRadial <- train(biomedicaldata ~., data= biodata, method="svmRadial", metric=metric, pre-
Processing=c("center", "scale"), trControl=control, fit=FALSE)
# KNN
set.seed(seed)
fit.knn <- train(biomedicaldata ~., data= biodata, method="knn", metric=metric, preProc=c("center", "scale"),
trControl=control)
# Naive Bayes
set.seed(seed)
fit.nb <- train(biomedicaldata ~., data= biodata, method="nb", metric=metric, trControl=control)
# CART
set.seed(seed)
fit.cart <- train(biomedicaldata ~., data= biodata, method="rpart", metric=metric, trControl=control)
```

```
# C5.0
set.seed(seed)
fit.c50 <- train(biomedicaldata ~., data= biodata, method="C5.0", metric=metric, trControl=control)
# Bagged CART
set.seed(seed)
fit.treebag <- train(biomedicaldata ~., data= biodata, method="treebag", metric=metric, trControl=control)
# Random Forest
set.seed(seed)
fit.rf <- train(biomedicaldata ~., data= biodata, method="rf", metric=metric, trControl=control)
# Stochastic Gradient Boosting
set.seed(seed)
fit.gbm <- train(biomedicaldata ~., data= biodata, method="gbm", metric=metric, trControl=control,
verbose=FALSE)
```

5. Selecting Appropriate Model

After the data has been trained with various algorithms, they are necessary to be assessed and compared. At this step, the best algorithm is not searched for because they are not tuned which provides better results than the current results. The main aim is to select the well performed algorithm. To perform this task, the following code snippet was conducted:

```
consequences <- resamples(list(lda=fit.lda, logistic=fit.glm, glmnet=fit.glmnet,
            svm=fit.svmRadial, knn=fit.knn, nb=fit.nb, cart=fit.cart, c50=fit.c50,
            bagging=fit.treebag, rf=fit.rf, gbm=fit.gbm))
# Table comparison
summary(consequences)
1
2
3
4
5
results <- resamples(list(lda=fit.lda, logistic=fit.glm, glmnet=fit.glmnet,
            svm=fit.svmRadial, knn=fit.knn, nb=fit.nb, cart=fit.cart, c50=fit.c50,
            bagging=fit.treebag, rf=fit.rf, gbm=fit.gbm))
# Table comparison
summary(results)
```

The summary of the consequences of accuracy is shown in Table 1.

Table 1. Consequences of Algorithms Accuracy.

Algorithm	Average	Accuracy
LDA	0.775	0.870
Logistic	0.778	0.870
Glmnet	0.777	0.870
SVM	0.765	0.896
KNN	0.746	0.896
NB	0.756	0.857
CART	0.738	0.844
C50	0.758	0.883
Bagging	0.753	0.857
RF	0.761	0.857
GBM	0.770	0.883

According to Table 1, the outcome shows that linear algorithms, such as LDA, Log, Glmnet and GBM perform better to be conducted on biomedical data on diabetes for women. Figure 3 shows the consequences of algorithms accuracy:

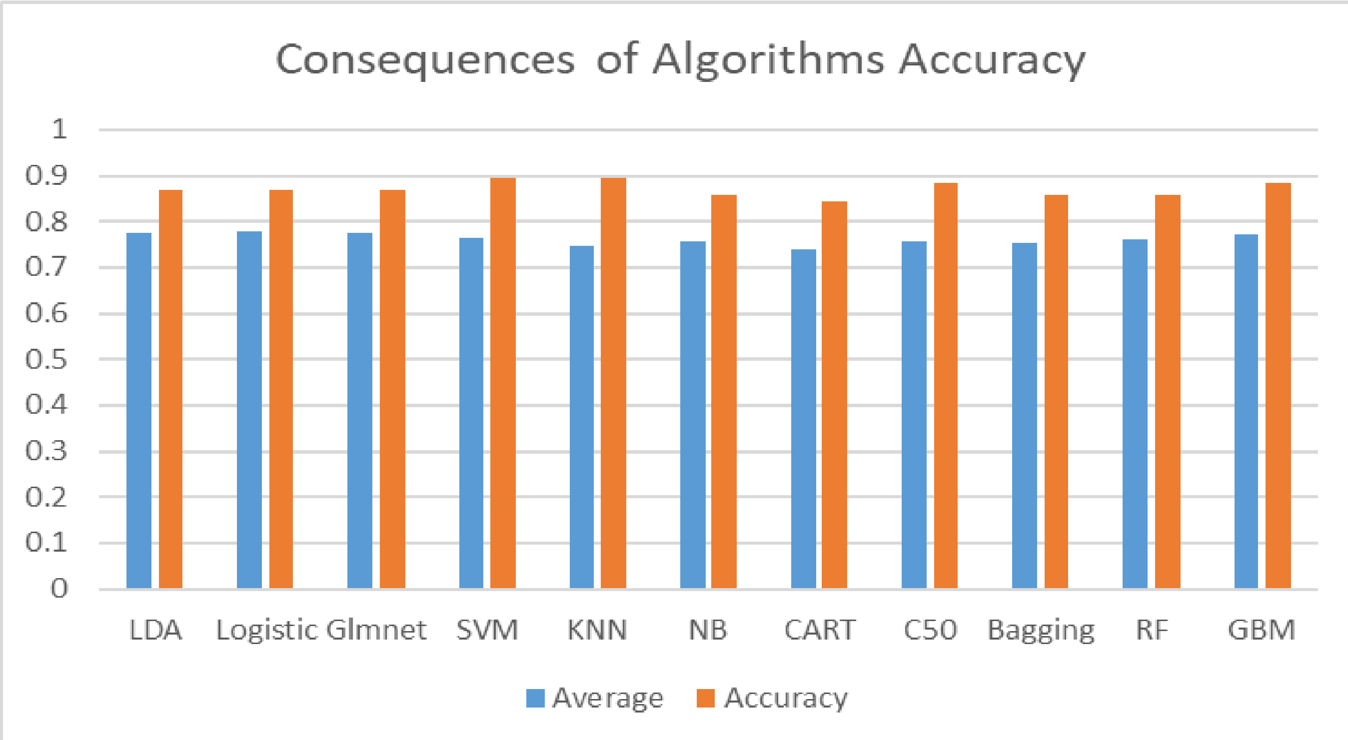

Figure 3. Consequences of Algorithms Accuracy.

5.1 Spot-checking for Better Algorithm

In this research paper, several recommendations were addressed to select or spot a significant algorithm. These following suggestions are supportive at assessing machine-learning algorithms by utilizing R programming:

1. Pace or speed: It is important to obtain consequences fast, utilize small amount of sample data and utilize basic anticipates for parameters of the algorithm. This might take a minute to an hour depending on the sample dataset.
2. Variety: It is useful to utilize various algorithms for the representations. Besides, for the same representation, utilize various learning algorithms.
3. Scale-Up: Researchers could follow scaling up with spot-checking algorithms for greater amount of sample data. Nevertheless, this needs more time. Besides, it might need a more powerful computer, but it helps to obtain algorithms that perform better with greater amount of sample data.
4. A brief list: One of the purposes of spot-checking algorithms is to make a brief list of algorithms to find out more, not optimal accuracy.
5. Heuristic: It is the best form to experience algorithms configuration.

6. Conclusion

In conclusion, this research paper helps to find out the significance of spot-checking machine-learning ML algorithms for a problem. In addition, it reveals that the spot-checking is the best option to discover a good ML algorithm for a provided dataset. A case study on diabetes patient for women dataset was considered by utilizing R programming. Besides, various algorithms were assessed on a class variable which is a binary class variable. Moreover, a better algorithm has been selected based on the tests conducted and the results obtained and compared. Consequently, this work answers the questions: which algorithm should be utilized on a dataset and how to investigate ML algorithm. We have noticed that in recent years, machine learning algorithms are still considered one of the most promising models in terms of being integrated with other technologies. This means that the changes and developments in it are continuing to come up with new methods that can be more effective and capable of yielding more satisfying results. For future reading, the authors advise the reader could optionally read the following research works (Hassan and Rashid, 2019, 2020, 2021; Hassan, 2020; Hassan et al, 2016, 2021a, 2021b).

References

Smola, A., & Vishwanathan, S. V. N. (2010). Introduction to machine learning. *Cambridge University, UK*, *32*(34), 2010.

Hassan, B. A. (2021). CSCF: a chaotic sine cosine firefly algorithm for practical application problems. *Neural Computing and Applications*, *33*(12), 7011-7030.

Ahmed, A. M., Hassan, B., Saeed, S., & Saeed, A. A. (2016). Evaluating e-Government Services in Kurdistan Institution for Strategic Studies and Scientific Research Using the EGOVSAT Model. *Journal of Applied Research*, *1*(2).

Hassan, B. A., & Rashid, T. A. (2020). Operational framework for recent advances in backtracking search optimisation algorithm: A systematic review and performance evaluation. *Applied Mathematics and Computation*, *370*, 124919.

Hassan, B. A., & Rashid, T. A. (2020). Datasets on statistical analysis and performance evaluation of backtracking search optimisation algorithm compared with its counterpart algorithms. *Data in brief*, *28*, 105046.

Hassan, B. A., & Rashid, T. A. (2021). A multidisciplinary ensemble algorithm for clustering heterogeneous datasets. *Neural Computing and Applications*, *33*(17), 10987-11010.

Hassan, B. A., Rashid, T. A., & Mirjalili, S. (2021). Formal context reduction in deriving concept hierarchies from corpora using adaptive evolutionary clustering algorithm star. *Complex & Intelligent Systems*, *7*(5), 2383-2398.

Hassan, B. A., Rashid, T. A., & Mirjalili, S. (2021). Performance evaluation results of evolutionary clustering algorithm star for clustering heterogeneous datasets. *Data in Brief*, *36*, 107044.

Hassan, B. A., Rashid, T. A., & Hamarashid, H. K. (2021). A novel cluster detection of COVID-19 patients and medical disease conditions using improved evolutionary clustering algorithm star. *Computers in biology and medicine*, *138*, 104866.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*, 381-386.

Kumar, C. (2018). Artificial intelligence: Definition, types, examples, technologies. *Epub: medium. com*, *31*.

EDUCBA (2020). Introduction to Types of Machine Learning Algorithms. Retrieved on 30 December 2020 from URL: https://www.educba.com/types-of-machine-learning-algorithms/.

Expert System Team (2020). What is Machine Learning?. Retrieved on 05 January 2021 from URL: https://www.expert.ai/blog/machine-learning-definition/.

Hamarashid, H. K., Saeed, S. A., & Rashid, T. A. (2021). Next word prediction based on the N-gram model for Kurdish Sorani and Kurmanji. *Neural Computing and Applications*, *33*(9), 4547-4566.

Hamarashid, H. K., Saeed, S. A., & Rashid, T. A. (2022). A comprehensive review and evaluation on text predictive and entertainment systems. *Soft Computing*, 1-22.

Hamarashid, H. K. (2021). Utilizing statistical tests for comparing machine learning algorithms. *Kurd J Appl Res*, *6*(1), 69-74.

Hamarashid, H. K., Saeed, M. H. R., & Saeed, S. (2017). Designing a smart traffic light algorithm (HMS) based on modified round robin algorithm. *Kurdistan Journal of Applied Research*, *2*(1), 27-30.

H. Mujtaba (2020). Learn Machine Learning with Python. Retrieved on 05 January 2021 from URL: Machine Learning Tutorial for Beginners | Machine Learning with Python (mygreatlearning.com).

Reddy, Y. C. A. P., Viswanath, P., & Reddy, B. E. (2018). Semi-supervised learning: A brief review. *Int. J. Eng. Technol*, *7*(1.8), 81.

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373-440.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, *11*(3-4), 219-354.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., ... & Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, *14*(1), 1-37.