DOI: http://dx.doi.org/10.26483/ijarcs.v12i1.6685

Volume 12, No. 1, January-February 2021

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

A REVIEW OF SUPERVISED MACHINE LEARNING ALGORITHMS TO CLASSIFY DONORS FOR CHARITY

Pooja Mittal
Research Scholar, Department of Computer Science and Applications,
Baba Mastnath University,
AsthalBohar, Rohtak

Dr. V. K. Srivastava
Professor and Head, Department of Computer Science and Applications,
Baba Mastnath University,
AsthalBohar, Rohtak

Abstract: Machine Learning has several supervised algorithms which have the capability for potential prediction based on the data collected from external or internal sources. Different supervised algorithms are employed to find the best-chosen algorithms based on the preliminary results and then optimized further to find the best outcomes. Non-profit organization survives on donations and predicting the individual's income helps to identify how big a donation can be made by the individuals. Therefore, it helps whether to approach to the individuals or not based on their income. With the
help of this paper, different algorithms are constructed and discussed based on their accuracy, complexity, speed, and overfitting to choose the best candidate model. Best optimized model helps to predict the individual's income efficiently and help making decision whether to reach out to them or not which helps in the non-profit organization survival.

Keywords: Supervised Machine Learning, Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbor, XGBoost

I. INTRODUCTION

In today's world, many countries as well as many organizations are running with a motive to help any individual or the community which is in need currently. These kinds of working organization are term as non-profit organization. These non-profit organizations do not have any target, in terms of profit, to achieve. Instead they run to help other ones without earning a single penny for their own. They help with the amount; they receive as donation from various sources like high income group, social workers, big industries etc. They can serve the humanity in terms of monetary help, goods, services or any combination of two. Some of the non-profit organizations are universities, hospitals, churches, foundations or national charities. The objective of this study is to model accurately the income of an individual by using the data which have been collected from U.S. census, 1994. Various supervised machine learning algorithm have been proposed to identify such income bracket individual or community. By applying the various supervised machine learning algorithm, first, it is predicted whether or not an individual earn $50000 or more than that. Such type of prediction helps the organization whose surveillance is totally dependent on the donations. Next, the candidate algorithm with the best prediction is selected as the final one and then again it is optimized in order to best model data. This prediction is very crucial as in order to predict large donation source. Once the income of individual is identified, the non-profit organization can best estimate the amount of donation, to be received or to be requested. Such type of information is very difficult to extract from public sources. So, various public domains are used to fetch such kind of authentic information. UCI Machine Learning Repository has been used for retrieving the data set. Ron Kohavi published the data set in the article "Scaling Up the Accuracy of Naïve-Bayes Classifiers: A Decision Tree Hybrid"[1]. In today's world, the role of machine learning is increasing day by day. Machine learning is a field of computer science, which is derived from artificial intelligence or we can think it as a subset of artificial intelligence. It trains a machine without any explicit programming. It is comprised of a number of coding programs for a problem which further adjust themselves according to the external data provided to it and perform at their best. This all is managed by a model which must be parameterized and having tuned parameters. It must be designed in such a manner that it adjusts itself while handling different type of performance criteria. Therefore, the field of machine learning is divided into three important categories: (1) Supervised Learning, (2) Unsupervised Learning (3) Reinforcement Learning. With reference to this paper, we will study about supervised machine learning approach or algorithms. Simply saying, supervised machine learning is "learning by doing". In this approach, the model is trained with a labelled data set. The given data set is split into two parts: (1) training data set & (2) testing data set. Training data set is used to train the model, while on the other hand; testing data set is used to check the accuracy level of

© 2020-2022, IJARCS All Rights Reserved

prediction by the model. The approach says: first find out the final goal and then find out the best way to achieve that goal.

This supervised machine learning is followed in two situations: Classification & Regression.

Classification: Whenever output variable is categorized and we are able to label them then this situation is known as classification. Here, we are able to answer in the form of yes or no, red or blue, pass or fail [3][7].

Regression: Whenever output variable is predicted in terms of real-valued quantities then this situation is known as regression problem. Here, we are able to answer in Dollars, weight etc.

This supervised machine learning approach uses the previous history and experiences to drive new results and predictions and hence, is helpful in optimizing the performance [13]. Therefore, it is very beneficial in handling real world problems.

Various algorithms have been proposed for supervised machine learning approach:

1. Logistic regression
2. Decision Trees
3. Random Forest
4. XGBoost
5. K-NN etc.

II. METHODOLOGIES

Fig. 1 depicts the general scenario for identifying a donor which is based on the individual's income, then for charity. It has two components. The first will be to predict individual's income based on the publicly available features provided. In the second step, it will find the donor for charity, if the individual has an income of more than $50000. Finding the donor or to determine the donor is achieved by the various classification methods like Logistic Regression, Decision Tree, Random Forest, K Nearest Neighbors and XGBoost. Identifying the best performing algorithm, different methods are used in order to make the comparisons like confusion matrix, classification report and cross validation as a performance measure.

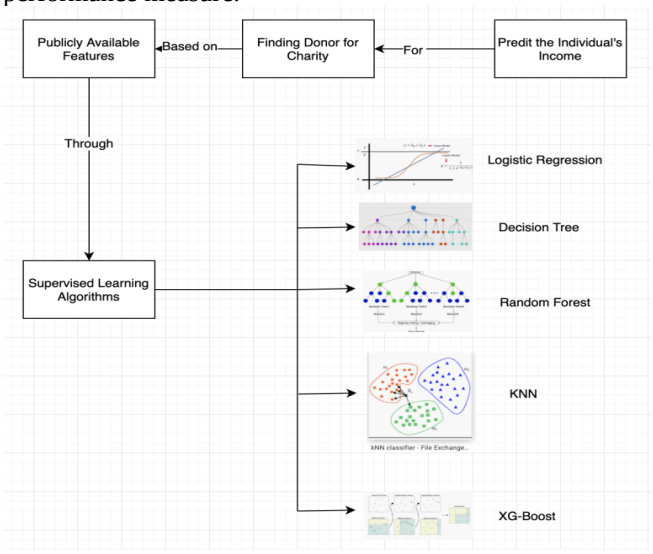

Figure1. General Scenario of Finding Donor Problem

To train models of Logistic Regression, Decision Tree, Random Forest, K Nearest Neighbors and XGBoost, a training dataset of which distribution of two classes corresponding to donor or not a donor were used. The data set has the data distribution with 34014 records for the individual having income less than $50000 whereas 11208 records are available having income of $50000 greater. The final model is evaluated with the help of confusion matrix, classification report and cross validation. Fig.2, depicts the scenario of solving the problem of Finding Donors.

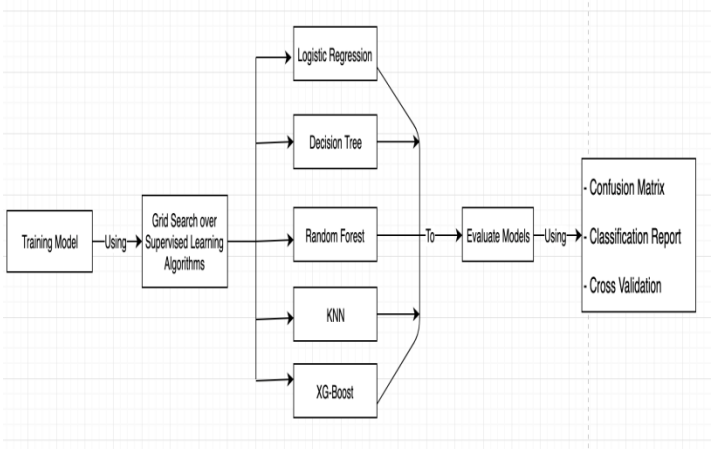

Figure2. Flow Diagram for Finding Donor Problem

A. Logistic Regression

It is one of the supervised machine learning classification algorithms which we use to predict target variable probability. It is a statistical model in which Logistic Curve is fitted to dataset. This model is applied where the target variable or dependent variable is dichotomous in nature. Dichotomous means the nature of target variable can be recorded as either 1 or 0. '1' is used to denote 'yes' or 'success' while '0' is used to denote 'no' or 'failure' the model is capable in terms of good probabilistic interpretation. Also, it can accommodate new data very easily. This can be done by using gradient descent method. This model returns probability. Therefore, the adjustment of classification threshold can be made easily. Discriminant analysis can be replaced by Logistic model. Some of the assumptions have to be followed for this:

1. No linear relationship between target variable and predictors will be assumed.
2. No assumption on Independent variable distribution.

Logistic regression model is able to handle power terms, nonlinear effect and interactive effect. The main prerequisite for Logistic regression model is large sample size which helps in predicting the stable results.

In order to procure Logistic Regression model with the help of GridSearch, hyperparameters defined in Table I, are used. Variables like penalty, solver, max_iter used by the model during prediction phase. Table I, represent the dictionary of defined hyperparameters for Logistic Regression.

© 2020-2022, IJARCS All Rights Reserved

Table I. Logistic Regression Hyperparameters

Hyper Parameters	Possible Values
Penalty	L1,L2
Solver	{'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}
Max_Iter	Range(50,200)

According to the hyperparameters defined as hyperparameters dictionary for Logistic Regression, total 70 models were trained. The best performance was achieved by taking 'Penalty' value as 5, 'Solver' value as 4 with 'maximum iteration' 1. Fig.3, represents the model venture along with its hyperparameters.

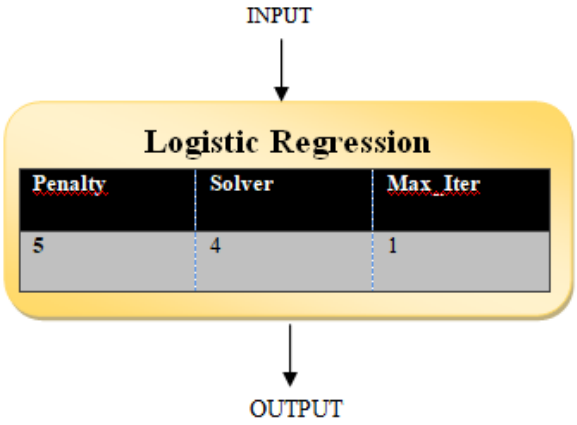

Figure3. Logistic Regression Plan of Action

B. Decision Tree

Decision tree is one of the famous supervised machine learning classification models. Here, the data is continuously split up based on certain parameters. It looks like a flowchart where every internal node or decision node exhibits a test on feature[4]. After computing the entire feature and taking decision, a leaf node is derived which represents the final outcome. It is also known as "class label". The branches are used for conjunction features that help in directing towards final output. The interaction between the features can be easily handled by decision trees. They are easy to build, interpret and define. Due to the nonparametric nature, it is not much affected due to the outliers. Some of the popular algorithms for decision tree are:C4.5, C5.0, ID3 and CART. The choice of algorithm really depends upon various splitting criteria like Gain Ratio, Gini Coefficient, and Information Gain etc. Some of the important features of Decision Tree which make it popular are:

- Ability of handling missing data
- Redundant features can be handled
- A variety of data like numerical data, nominal data or textual data can be handled by decision tree
- Ability to have good generalization
- It is very robust to noise
- With very small computational effort, achieve good results.

Apart from that, decision tree is sometime inefficient in handling high dimensional data. It's true that the computation time with DT is less but over all time to construct the tree maybe high. In case of some highly relevant attributes, the divide and conquer strategy followed by it gives its best. In case of complex structure, this strategy is somehow find itself fail to perform well. Apart from that as soon as the number of classes increases, the error starts propagating through the tree which is a key issue. Another problems associated with it, is "data fragmentation". As and when the tree grows, the number of records in each leaf nodes starts getting decrease. Due to this, statistical important decisions are almost impossible to conclude. To resolve this, a threshold value is decided after which the tree will not be split further. Over fitting is another problem that may arise in decision tree when proper pruning is not performed. Therefore, to handle all these problems another ensemble learning model was developed which is known as random forest.

In order to procure Decision Tree model with the help of GridSearch, hyperparameters defined in Table II, are used. Variables like Criterion, Max_Depth, Min_Samples_Split were used by the model during prediction phase. Table II, represent the dictionary of defined hyperparameters for Decision Tree.

Table II. Decision Tree Hyperparameters

Hyper Parameters	Possible Values
Criterion	Gini, entropy
Max_depth	Range(2 – 20)
Min_Samples_Split	Range(2-10)

As a result, total 320 models were trained based on hyperparameters defined in the hyperparameter dictionary for Decision Tree. The best performance was achieved with 'Criterion', which is used to measure the division quality is taken as 'Entropy', with 'Maximum Depth' value 5 having 'Minimum sample splits' 2. Fig. 4, represents the model venture along with its hyperparameters.

© 2020-2022, IJARCS All Rights Reserved

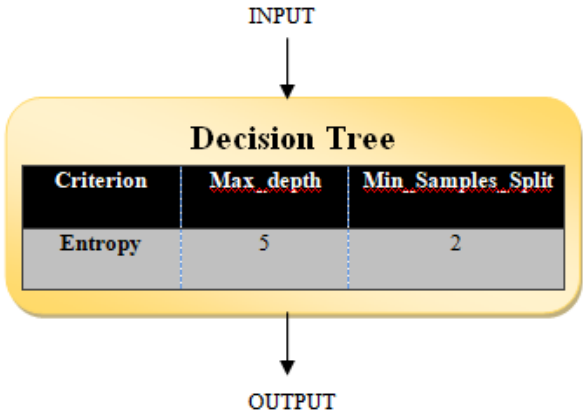

Figure4.Decision Tree Plan of Action

C. Random Forest

Random forest is a supervised machine learning model which is based on ensemble learning techniques. Here, the model performs its job in two phases. In first phase, a number of decision trees are constructed on different data set, derived from original data set. These DTs predict some results. After that, using voting mechanism, it is find out, which output is maximum time predicted. At last, the class which has maximum vote is declared actual prediction by the model. Random Forest is the favourite choice by most of the ML programmer to solve the classification problem as the results predicted by it are of high accuracy[8]. Various features like scalability, fast prediction, robust to noise, avoid overfitting, easily interpretable makes it popular among other ML model. But, as soon as the number of trees starts increasing, the model is slow down in predicting the problem with real world scenario. To solve this problem, some attempts are performed such as reducing the correlation that exists among the trees and for splitting; different measures for attribute evaluation can be used. Therefore, to improve the performance of Random Forest, some of the prerequisites are as follows:

1. The feature selection should be good in order to get better results rather than selecting randomly.
2. The correlation among the predictions of the trees should be less.

In order to procure Random Forest model with the help of GridSearch, hyperparameters defined in Table III., are used. Variables like Criterion, Max_Depth, N_Estimaors were used by the model during prediction phase. Table III, represents the dictionary of defined hyperparameters for Random Forest.

Table III. Random Forest Hyperparameters

Hyper Parameters	Possible Values
Criterion	Gini, entropy
Max_Depth	Range(2,20)
N_Estimators	Range(5,50)

Based on the hyperparameters defined, a total of 360 models were trained. By using the combination; 'Criterion' value as 'Gini' with 'maximum depth' value selected none having 'N_Estimators' 100; best performance was achieved.Fig. 5, represents the model venture along with its hyperparameters.

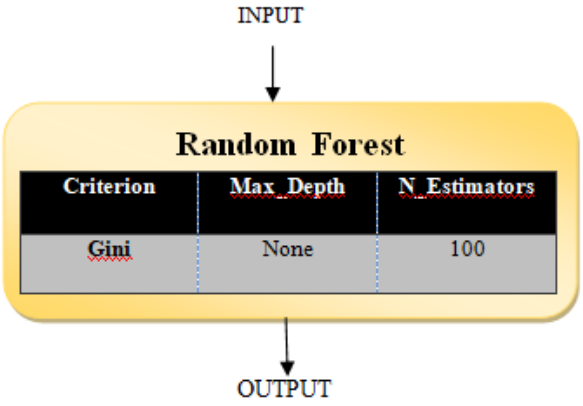

Figure5. Random Forest Plan of Action

D. K-NN

It is a supervised machine learning algorithm which is non parametric in nature. It is used to find out the solution for both regression as well as classification problems. The property of easy to build and simplicity makes it popular among other classification algorithm. It tries to find out the closeness between existing data and cases those are available now. After that, new cases are put into the classification class whichever is most closed. It does not have any assumption quality in it. We can call it a "Lazy Learner" model just because it does not derive from the existing data set instantly. It comes into action during classification time. The performance of KNN is totally dependent on how good we select "k", number of neighbors. In general, there is no standard rule for selecting the value of "k". We have to try different values of "k" introduces noise and may introduce outliers while on the other hand; very large value of "k" is good but difficult to manage. However, to choose the value of 'k' computationally is an expensive method using cross validation or any other process. It is very sensitive to features which are irrelevant and does not make any sense in predicting the results. The cost of computation is sometimes high. The performance depends upon the size of the training data set size, It is supposed to have a large training data set in order to achieve good prediction accuracy.

In order to procure K-Nearest Neighbors Model with the help of GridSearch, hyperparameters defined in table IV, are used. Variables like Leaf_Size, N_Neighbors, P (Probability) were used by the model during prediction phase. Table IV, represent the dictionary of defined hyperparameters for K-Nearest Neighbors.

© 2020-2022, IJARCS All Rights Reserved

Table IV. K-NN Hyperparameters

Hyper Parameters	Possible Values
Leaf_Size	range(1,50)
N_Neighbors	range(1,30)
P	[1,2]

According to the hyperparameters, defined in the hyperparameter dictionary in table IV, total 300 models were trained. The best performance was achieved by taking 'Leaf_Size' 30 with 'Number of Neighbors' 5 having 'p' as 2. Fig.6, represents the model venture along with its hyperparameters.

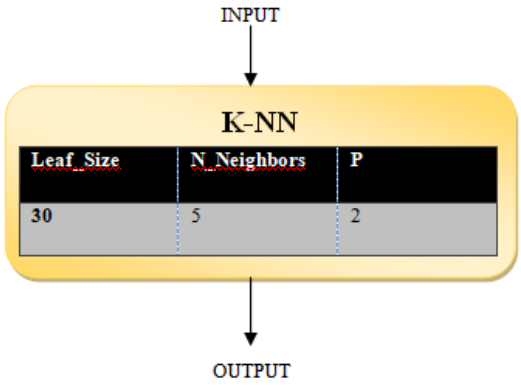

Figure6. K-NN Plan of Action

E. XGBoost

XGBoost is an efficient and popular implementation of gradient boosted trees. This implementation is open source in nature. Gradient Boosting technique is also a supervised machine learning approach in which the prediction is performed at various stages in incrementing order by taking the weakness of previous data set and adding them to the new one to make it stronger[2][14]. AT last, a desired prediction is achieved. It is called so because of its use of gradient descent approach in order to reducing the loss while new models are added. It is an ensemble learning approach. Now a day, XGBoost is becoming very popular because of its high prediction accuracy. It is a speedy algorithm. However, parallelizability makes it strong because data set can be run parallel on various GPU's and can take benefits of their computing capability. In case of XGBoost, no need of cross validation externally. It has the internal capability for regularization, cross validation, handling missing values; user defined objective function, tuned parameters etc.

In order to procure XGBoost model with the help of GridSearch, hyperparameters defined in table V, are used. Variables like Booster, Max_Depth, Estimators were used by the model during prediction phase.Table V, represent the dictionary of defined hyperparameters for XGBoost.

Table V. XGBoost Hyperparameters

Hyper Parameters	Possible Values
Booster	Gbtree, gblinear
max_depth	Range(5,20)
Estimators	Range(50,300)

A total of 180 models were trained by using the hyper parameters defined in the hyperparameter dictionary for XGBoost. The best combination was achieved by taking 'Booster' value 'Gbtree' with 'Maximum Depth ' value 6 and having 'Estimators value' 100. Fig. 7,represents the model venture along with its hyperparameters.

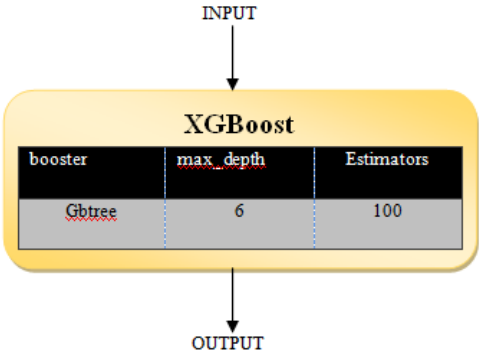

Figure7. XGBoost Plan of Action

III. EXPERIMENT RESULTS

A total of 9045 sample records were used in order to evaluate models' accuracy and comparing their performances[12]. However, model's accuracy is measured by various parameters likeconfusion matrix, classification report and cross validation methods among which classification report is comprised of precision, recall, f1 score and support parameters ans cross validation with the purpose to calculate average accuracy produced by the model. At last, final comparisions among the implemented models are performed in order to find out best one[5][9][15].

A. Confusion Matrix

Confusion Matrix for every model was obtained with the help of sk-learn library which is implemented in Python. This matrix was obtained with the help of defined test data set.

1. Logistic Regression

Table VI, represents the confusion matrix using Logistic Regression model during the prediction phase, with every class having 9045 samples. Here, 85% success rate was obtained.

© 2020-2022, IJARCS All Rights Reserved

Table VI. Confusion Matrix using LR

Actual / Predicted	0	1
0	6283	462
1	936	1364

2. *Decision Tree*

Table VII, represents the confusion matrix using Decision Tree model during the prediction phase, with every class having 9045 samples. Here, 84% success rate was obtained.

Table VII. Confusion Matrix using DT

Actual / Predicted	0	1
0	6462	283
1	1157	1143

3. *Random Forest*

Table VIII, represents the confusion matrix using Random Forest model during the prediction phase, with every class having 9045 samples. Here, 84% success rate was obtained.

Table VIII. Confusion Matrix using RF

Actual / Predicted	0	1
0	6174	571
1	841	1459

4. *K Nearest Neighbor*

Table IX, represents the confusion matrix using K Nearest Neighbor model during the prediction phase, with every class having 9045 samples. Here, 83% success rate was obtained.

Table IX. Confusion Matrix using KNN

Actual / Predicted	0	1
0	6154	591
1	974	1326

5. *XGBoost*

Table X, represents the confusion matrix using XGBoost model during the prediction phase, with every class having 9045 samples. Here, 87% success rate was obtained.

Table X. Confusion Matrix using XGB

Actual / Predicted	0	1
0	6366	379
1	778	1522

After observing the confusion matrices defined in above section, it can be viewed that the model XGBoost gives its best performancewhile solving the problem. The success rate of XGBoost is 87% which is greater than other algorithms, used to solve the problem.. Table XI shows the confusion matrix values analysis.

Table XI. Confusion Matrices Analysis

Method	**0**	**1**
Logistic Regression	0.85	0.15
Decision Tree	0.84	0.16
Random Forest	0.84	0.16
K-Nearest Neighbors	0.83	0.17
XGBoost	0.87	0.13

B. *Classification Report*

In order to obtain metrics like precision, recall, f1-score, sk-learn library was used in module of classification report[6][11]. This classification report module includes all the metrics defined above. For every trained model, the average of each metrics is represented by Table XII.

Table XII. Classification Report with Average Metrics

S. No.	Results based on Algorithm				
	Algorithm	Precision	Recall	F1-Score	Accuracy
1.	Logistic Regression	0.84	0.85	0.84	0.85
2.	Decision Tree	0.84	0.84	0.83	0.83
3.	Random Forest	0.84	0.84	0.84	0.84
4.	K-NN	0.82	0.83	0.82	0.83
5.	XGBoost	0.87	0.87	0.87	0.87

Fig. 8 is the graphical representation of classification report of all algorithms shows the comparison of their accuracy, Precision, Recall and F1 score. Metrics shows that the XGB model is the best performing model as compared to K-NN, RF, DT and LR models by ~3% on an average. Which depicts that XGB model is able to predict the maximum true positives

© 2020-2022, IJARCS All Rights Reserved

and prevents the false positives coming to the system whereas other models have more false positives compared to XGB model. Having more true positives will lead to get more accurate individual's income and further to make decision of finding right donor rather than spending effort on individuals who will not be able to donate.

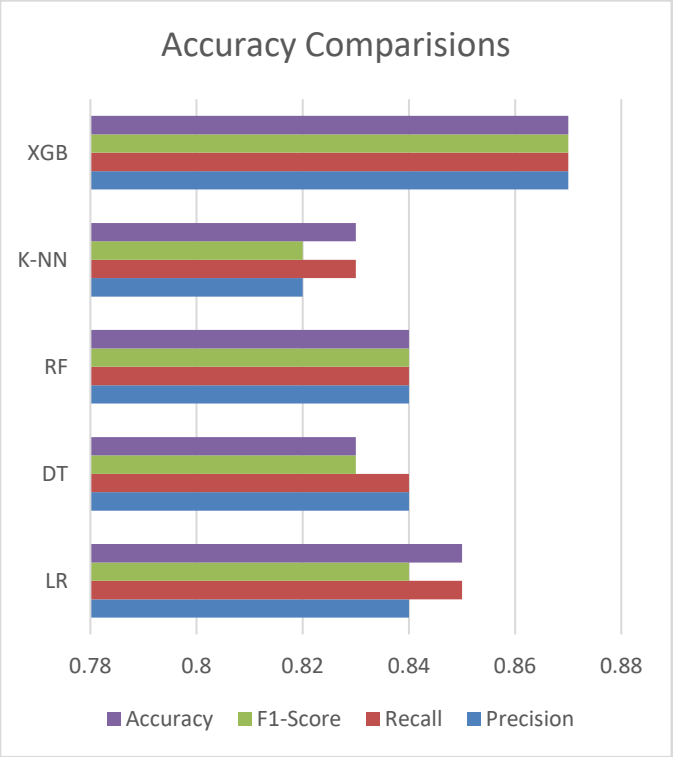

Figure8. Classification Report Values Comparison

C. Cross Validation

To perform cross validation phase, training dataset as well as test dataset were used. With the help of sk-learn library, various functions like cross_val_score, K-Fold were implemented in the module of module_selection. Table XIII represents the average cross validation score of individual model.

Table XIII. Average Score Values in Cross Validation

Model	Average Cross Validation Score
XGB	0.8873
KNN	0.8113
RF	0.8291
DT	0.8234
LR	0.8312

Fig. 9 is the representation of model comparison by their cross-validation score depicts that the XGB model has the highest cross validation score which is approx. 0.88%. XGB model score is an increase of 7% score than KNN, 5% against RF, 6% against DT and 5% more than LR model. Model exposing the cross-validation score around 88% is a promising score and provides a confident model. Cross validation score is calculated by the 10 divisions of dataset called k-folds and accuracy is calculated with the data variation which is considered quite stable. Therefore, XGB model with 88% cross validation score will remain stable and will not vary significantly even if the data variation is there in future.

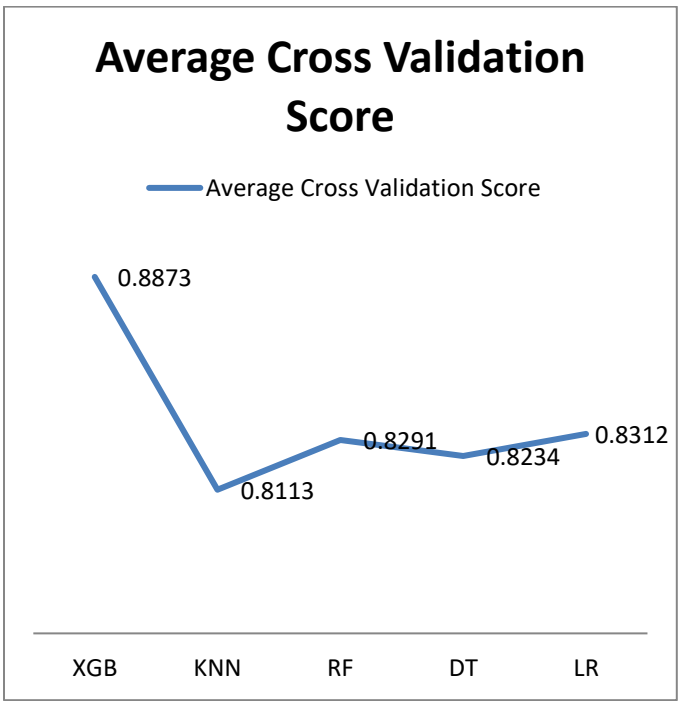

Figure9. Average Cross Validation Score Comparison

IV. CONCLUSIONS

This paper focuses on some of the commonly used supervised machine learning algorithm. Based on the study, it is concluded that the model XGBoost has higher accuracy as compared to other supervised machine learning algorithms like Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors to find the donor for charity. Best hyperparameters were used for each model in order to generate the best model. Here, we have discussed them to solve classification problem. This algorithm can be used to solve regression problem as well. Our main goal is to get into a thorough review of the most important ideas of the algorithm discussed. With the help of this review paper, we come to know that each and every algorithm works differently with different scenarios. A single algorithm cannot be treated as best one in every situation. The selection of algorithm depends upon various factors such as type of application, type and number of attributes, complexity of problem etc. By using ensemble techniques, we can increase the accuracy of the algorithm. Hopefully, the references which are covered here will help the researchers to deeply study the concepts of Supervised Machine Learning algorithms and will guide in their research in an interesting manner.

© 2020-2022, IJARCS All Rights Reserved

REFERENCES

[1] K. Ron, "Scaling Up the Accuracy of Naive - Bayes Classifiers: A decision Tree Hybrid," in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996, pp. 202–207, Accessed: Jan. 14, 2021. [Online]. Available: https://www.aaai.org/Papers/KDD/1996/KDD96-033.pdf.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016, vol. 13-17-Augu, pp. 785–794, doi: 10.1145/2939672.2939785.

[3] G. Cybenko and T. G. Allen, "Parallel Algorithms For Classification And Clustering," Adv. Algorithms Archit. Signal Process. II, vol. 0826, no. 4, p. 126, 1988, doi: 10.1117/12.942023.

[4] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers - A survey," IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 35, no. 4, pp. 476–487, 2005, doi: 10.1109/TSMCC.2004.843247.

[5] R.-M. Ștefan, "A Comparison of Data Classification Methods," Procedia Econ.Financ., vol. 3, no. 12, pp. 420–425, 2012, doi: 10.1016/s2212-5671(12)00174-8.

[6] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," Comput.Commun. Rev., vol. 36, no. 5, pp. 7–15, 2006, doi: 10.1145/1163593.1163596.

[7] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and P. S. Yu, "Active learning: A survey," Data Classif. Algorithms Appl., pp. 571–605, 2014, doi: 10.1201/b17320.

[8] M. Robnik-Šikonja, "Improving random forests," Lect. Notes Artif. Intell.(Subseries Lect. Notes Comput.Sci., vol. 3201, no. March, pp. 359–370, 2004, doi: 10.1007/978-3-540-30115-8_34.

[9] K. Wisaeng, "A Comparison of Different Classification Techniques for Bank Direct Marketing," Int. J. Soft Comput. Eng., no. 4, pp. 116–119, 2013, [Online]. Available: http://www.ijsce.org/wp-content/uploads/papers/v3i4/D1789093413.pdf.

[10] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," J. Mach. Learn. Res., vol. 5, pp. 1205–1224, 2004.

[11] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," J. Mach. Learn. Res., vol. 7, pp. 1–30, 2006.

[12] A. C. Lorena et al., "Comparing machine learning classifiers in potential distribution modelling," Expert Syst. Appl., vol. 38, no. 5, pp. 5268–5275, May 2011, doi: 10.1016/j.eswa.2010.10.031.

[13] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," Artif. Intell. Rev., vol. 26, no. 3, pp. 159–190, 2006, doi: 10.1007/s10462-007-9052-3.

[14] S. Dhaliwal, A.-A.Nahid, and R. Abbas, "Effective Intrusion Detection System Using XGBoost," Information, vol. 9, no. 7, p. 149, Jun. 2018, doi: 10.3390/info9070149.

[15] C. Rich and A. Niculescu-Mizil, "An Empirical Comparisions of Supervised Learning Algorithms," Icml, pp. 161–168, 2017, [Online]. Available: www.cs.cornell.edu.

© 2020-2022, IJARCS All Rights Reserved