STM JOURNALS

Journal of Artificial Intelligence Research & Advances
ISSN: 2395-6720 (Online)
Volume 4, Issue 3
www.stmjournals.com

A Review Paper on Comparison of Algorithms for Supervised Machine Learning

Abhishek Panjabi, Krupali H. Shah*, Vyshali Lasitha, Param Pandya
Department of Information Technology, BVM Engineering College, V.V. Nagar, Gujarat, India

Abstract

Supervised learning attempts to create calculations that help deliver general speculations, which is able to do forecasts about future occasions. The objective of supervised learning is to manufacture a model of the segregation of class names in terms of predictor components. The classifier produced accordingly is then used to allocate class marks to the testing data where the estimations of the indicator elements are known, yet the estimation of the class name is not known. This paper compares different supervised machine learning characterization strategies. The algorithms that are taken into considerations are K-Nearest Neighbor, Linear Regression, Logistic Regression, Naïve Bayes, Decision Trees, Random Forests and Neural Networks.

Keywords: *Supervised machine learning, algorithm, K-Nearest Neighbor, Linear Regression*

****Author for Correspondence*** E-mail: krupali.shah@bvmengineering.ac.in, panjabiabhishek@yahoo.com

INTRODUCTION

Most programmers consider learning as a parameter for clever machines. Profound understanding would help in taking choices in a more enhanced frame and furthermore help then to work in most proficient strategy. Learning is becoming a key to study of natural and non-natural vision. Instead of making high capacity machines with straight-forward programming now various algorithms are coming in market which will help the computer to interpret the implied environment and based on the perception the machine will produce the output based on the decision taken by it. This will help to reduce the programming concepts and machine will become self-reliant and do the settlement on its own.

Supervised learning is a method of creating a function from labeled training data [1]. The training data are having number of training tuples. Each and every tuple is a pair of an input object, i.e., a vector and a required output value. The algorithm inspects the training data and fabricate a deduced function, which can be useful for mapping new samples. It is a very well known process of classification. Classification is a systematic arrangement in categories or groups based on some established criteria. Classification uses a model to predict previously unknown values/attributes (output), using a number of already known values/attributes (input). The important function of Supervised Machine learning attempts is to tell machines how to find a good forecaster based on past encounters and it is done by satisfactory classifier.

COMPARISION

Problem Type

There are only two types of problems that all the models can have: Regression and Classification. K-nearest neighbor (KNN), random forests and neural networks can work with both the problem types, whereas, logistic regression and Naïve Bayes can only work with classification type of problems and linear regression can only handle regression type of problems. It is good to have an algorithm which can work with both types of problems. But before that we need to analyze other components also to choose a perfect algorithm.

Result is Interpretable by Human?

In KNN and linear regression results can generally be interpretable by humans because the output is based on weight of neighbors which is linear.

Table 1: *Comparing Supervised Learning Algorithms [2].*

Algorithm	Problem Type	Results Interpretable by you?	Easy to explain algorithm to others?	Average predictive accuracy	Training speed	Prediction speed	Amount of parameter tuning needed	Performs well with small number of observations?	Handles lots of irrelevant features well?	Automatically learns feature interactions?	Parametric	Features might need scaling?
KNN	Either	Yes	Yes	Lower	Fast	Depends on n	Minimal	No	No	No	No	Yes
Linear Regression	Regression	Yes	Yes	Lower	Fast	Fast	None (excluding regularization)	Yes	No	No	Yes	No (unless regularized)
Logistic Regression	Classification	Somewhat	Somewhat	Lower	Fast	Fast	None (excluding regularization)	Yes	No	No	Yes	No (unless regularized)
Naïve Bayes	Classification	Somewhat	Somewhat	Lower	Fast (excluding feature extraction)	Fast	Some for feature extraction	Yes	Yes	No	Yes	No
Decision Trees	Either	Somewhat	Somewhat	Lower	Fast	Fast	Some	No	No	Yes	No	No
Random Forests	Either	A little	No	Higher	Slow	Moderate	Some	No	Yes(unless noise ratio is high)	Yes	No	No
Neural Networks	Either	No	No	Higher	Slow	Fast	Lots	No	Yes	Yes	No	Yes

In linear regression we get weight of coefficient which is quite easy to work with but in logistic regression it is quite hard to understand the weights of coefficient. This is the case because in liner regression one can interpret a coefficient value of 2.5 as "keeping all other parameters fixed, a 1 unit increase in a parameter is associated with 2.5 unit increase in output" however, in logistic regression "a 1 unit increase in a parameter is connected with a 2.5 unit increase in log-odds in output". Naïve Bayes classification is also not very easy to interpret as it is based on the probabilities and multiplication of it and as we all know that probabilities are from 0 to 1 and multiplication of numbers less than 1 will make resulting number even smaller. Results of decision trees are somewhat hard to understand because as the number of training samples increases, depth of decision tree also increases. So, it becomes harder to understand the results. Result of neural networks cannot be interpreted as they are very complex in nature and consists of many stages.

Algorithm is Easy to Explain to Others?

KNN and linear regression are easy to explain because of their low complexity and linearity. Logistic regression Naïve Bayes and decision trees are little harder to explain as they are moderately complex in nature. They are not linear. However, random forests and neural networks are not easy to explain. As neural networks has high complexity and simple functions but with many hidden layers containing nodes.

What is Average Predictive Accuracy?

The former five algorithms have low predictive accuracy for a new sample. KNN is very susceptible to overfitting of data. If there are too few neighbors then it will lead to overfitting and if there are too many neighbors then it will lead to boundaries that may not locate structure required to separate the data. Linear regression has the solution linear in parameters, which means solution is flat hyperplane type which will fail to identify any curvature or non-linearity in data [3]. There are some more disadvantages of linear regression which makes it not so best algorithm. In logistic regression users are required to find out efficacious depictions of the predictor data that performs optimally. Naïve Bayes algorithm assumes predictors to be independent of others. Naïve Bayes algorithm also takes very unrealistic assumptions in consideration. If decision tree is having data which includes categorical variables at disparate levels, then information gain is biased on the side of those attributes with extra levels. Average predictive accuracy of random forest and neural network is higher. Random forest uses the shape of neighborhood which accommodates local significance of each and every feature [4]. If the learning algorithm, model and cost function is chosen properly then neural networks can be extremely vigorous.

Training Speed?

Training speed is important aspect to consider while choosing a supervised machine learning algorithm. KNN, linear regression, logistic regression, Naïve Bayes and decision trees consume very less to train whereas random forest and neural networks comparatively takes

more time to train the model. KNN, linear and logistic regression and Naïve Bayes are simple equation based algorithms and nowadays machines are becoming very fast in doing arithmetic calculations, that is why, they take so less time to train. Decision trees are built of conditions of like:
If condition1 and condition2 and … and condition n then output.

And the tree has only the expanding nodes, but no converging nodes. That is why training a decision tree and building one takes less time. With the increasing size of forest which means increasing number of trees which means more computational burden will incur to build and train the model [3]. Neural networks take more time in training compared to others because of hidden layers and back propagation between the nodes during training.

Prediction Speed of Algorithm?

Prediction speed is based on amount of time taken by a model to predict a sample using the trained model. It is again an important aspect to consider in choosing a best suitable model. The prediction speed of KNN is dependent on n because it finds out the distance between the sample and it's all the neighbors. So, more the neighbors, more time it will take to predict. Random forest has the moderate speed of prediction due to so many trees used during training. Except these two, all the rest of the algorithms have faster prediction speed. KNN, linear and logistic regression, Naïve Bayes and neural networks deals with not so complex arithmetic formulas. That is why the result can be calculated rapidly. Decision tree is also works with Boolean conditions based tree traversal, which can be evaluated quickly.

How much Parameter Tuning Needed?

Tuning a machine learning algorithm is a process in which optimization of parameters is done, which, influence the model in a positive way to empower the algorithm to give the best performance. Linear and logistic regression does not require any tuning parameters though it does require tuning parameters for regularization. KNN requires only the value of K as a tuning parameter for algorithm to perform best. Naïve Bayes algorithm requires few tuning parameters for feature extraction. With Naïve Bayes classifier, parameter tuning is limited, so to improve accuracy we can apply the tuning parameters for feature extraction. Decision tree and random forest requires some amount of tuning parameters. As these algorithms cannot handle all types of data by itself; one need to tune it by explicitly setting tuning parameters. Neural networks require many parameters for tuning the algorithm. In neural networks one should fine tune learning rate to reduce error rate, width and depth which is dependent on the type of problem.

Is Algorithm Performs Well with Small Number of Observations?

Linear regression, logistic regression and Naïve Bayes can perform well even with small number of observations supplied while training. This is the case, because, linear and logistic regression works with the arithmetic equations. One can estimate an arithmetic equation after few required observation. Naïve Bayes has the zero-one loss function which specifies error as the number of times the data were wrongly classified. Even if posterior probabilities may get changed, the class having greatest posterior probability is remains untouched. The rest of the algorithms perform worst with the small number of observations and KNN over fits the data if the neighbors are too few. If we have very few observations then at training time it may not split into more specialized nodes and may give wrong results while predicting. Random forests works with more than one tree and build a model. The problem in random forests is also same as of in decision trees. Neural networks need more observations to learn to do proper predictions. That is why it cannot work with few observations.

Is Model can Handle Many Irrelevant Features Well?

Handling many irrelevant features means model can separate signal from noise or not. The former three models stated in Table 1 and decision trees cannot manage lots of irrelevant attributes. We can increase the value of K so that outliers can nullify themselves but we cannot make K=N, because it will lead to oversimplified boundaries where always the majority class will be selected and wrong

prediction will be made. That is why we can only increase the value of K up to a certain extent. That puts barrier on the outlier handling capacity of KNN [5].

Linear and logistic regressions are highly susceptible to underfitting and overfitting of data. Decisions trees are supremely influenced by the training data [6]. A nominal change in data can result in whole different tree. Naïve Bayes, random forests and neural networks can easily handle the irrelevant features. In the case of Naïve Bayes the segregation of reliance between all the attributes of class that makes Naïve Bayes susceptible to outliers [7]. Noise while training model of random forests helps in reducing correlation within predictors which helps in prediction. Using some tuning parameters one can easily make the neural network that can withstand the irrelevant features.

Does Model Automatically Learn the Feature Interaction?

KNN, linear regression, logistic regression and Naïve Bayes algorithms cannot learn the feature interaction by themselves. KNN is based on the distance between the neighbors and based on the distance it classifies the data. It does not take the interaction into consideration. Linear and logistic regressions are based on the mathematical equations. They do not take inter-action of features into consideration. Naïve Bayes considers the hypothesis that the features are independent from each other.

The rest of the three algorithms can learn feature interaction by themselves. If there is any co-linearity in the two features then decision tree splits on the better node. This is how we can know about the feature interaction in decision trees. Trees are very good at finding co-linearity between features. Random forest is also based on the trees and is good at finding feature interaction. The example is shown in Figure 1 in which random forests performs best in fitting.

Is Algorithm Parametric?

A training model that gives an outline of data with a group of parameters of non-variable size which is independent of the quantity of training sample is called a parametric model [9]. Parametric algorithms generally make use of a function. Often this function is made up of linear mixture of input variables [10]. Linear regression, logistic regression and Naïve Bayes algorithms work with mathematical equations and hence they are parametric type of algorithms.

Non-parametric algorithms do not state the number of parameters. KNN, decision trees, random forests and neural networks are non-parametric type of algorithms. KNN algorithm does predictions which are dependent upon only the K most homogeneous training patterns for a new sample. Decision trees, random forests and neural networks also do not takes the number of parameters into account. They simply try to fit a sample based on the previously fitted data.

Features might need Scaling?

Feature scaling is a technique used to make the range of interdependent features of data consistent. One should do scaling when the scale is not meaningful. Main advantage of scaling is to stay away from considering attributes with higher range suppressing the importance of attributes having lower range. KNN and neural networks only required to do scaling of features. KNN uses Euclidean distance which is quite meaningful in nature. If a feature has a considerably big scale compared to another, then there arises a need to normalize the feature values. Same principal is also applied to neural networks. Despite these two, no other algorithm requires scaling. Linear and logistic regressions work with coefficients. So, scaling the coefficient might be a bad idea because after a bad scaling they might not be readable and may give false output. Naïve Bayes works with probabilities which are always in range from 0 to 1. So, practically there will be no advantage of scaling in Naïve Bayes. Tree based algorithms partition the tree by a node into two data sets by comparing a feature which splits the tree at its best to a threshold value. There is no need to regularize a threshold value. Therefore, tree based algorithms are not affected by different scales.

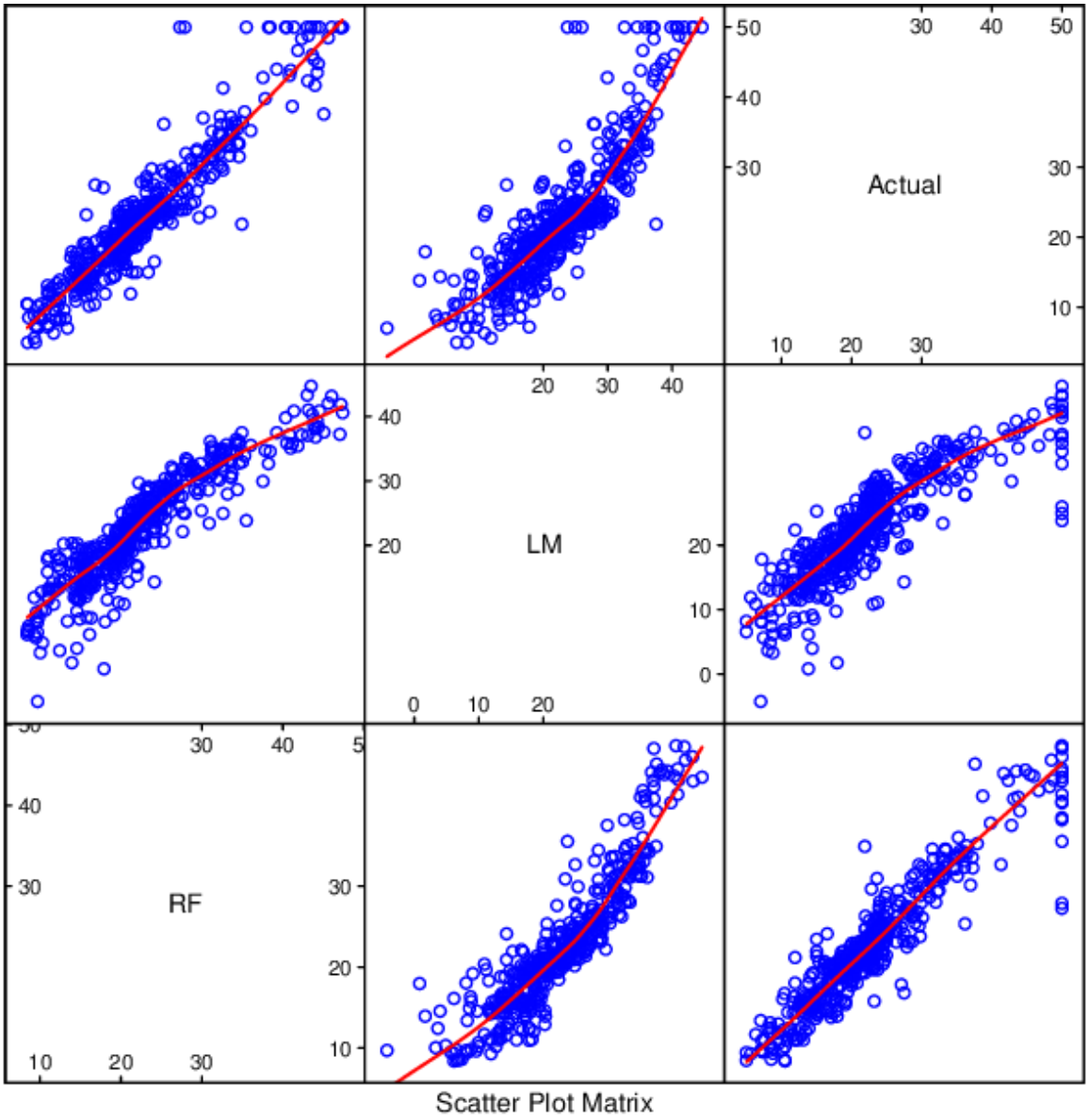

Fig. 1: *Fitting of Different Algorithms [8].*

CONCLUSION

Selecting a proper algorithm for a problem is very essential step in problems of machine learning. If it is done wrongly, then the performance will not be as expected. The comparison of different supervised machine learning algorithm based on a dozen parameters has been done in this paper. All these parameter plays a vital role in algorithm selection. We have described why the algorithm is having some particular characteristics. This paper can be very useful to someone who is unsure about which algorithm to use for one's problem. By reading this paper one can easily understand the different characteristics supervised learning algorithms have and can select the most appropriate algorithm based on one's need.

REFERENCES

1. Mehryar Mohri, Afshin Rostamizdeh, Ameet Talwalkar, *Foundations of Machine Learning*, MA: MIT Press, 2012.
2. Kevin Markham, *An article on Comparing Supervised Learning Algorithms*, machine learning, data school, February 27, 2015.
3. Max Kuhn, Kjell Johnson, *Applied Predictive Modelling*, Springer, 2013.
4. Lin Yi, Jeon Yongho. Random forests and adaptive nearest neighbors (Technical report), *Technical Report No. 1055*, University of Wisconsin, 2002.

5. Kumar Sricharan, Raviv Raich, Alfred O. HeroIII, K-nearest neighbor estimation of entropies with confidence, *IEEE International Symposium on Information Theory Proceedings*, 2011.
6. Lalit Sachan, A blog on *Logistic Regression vs decision Trees vs SVM: Part II*, 2015.
7. Harry Zhang, optimality of Naïve Bayes, flairs04-097, *AAAI Conference on Artificial Intelligence*, 2004.
8. Andy Liaw and Matthew Wiener, A paper on *Classification and Regression by RandomForest*, 2012.
9. Stuart Russell, Peter Norvig. *Artificial Intelligence: A Modern Approach*, 3rd edition, 2013.
10. Jason Brownlee, An article on *Parametric and Nonparametric Machine Learning Algorithms*, in Machine Learning Algorithms, 2016.

Cite this Article

Abhishek Panjabi, Krupali H. Shah, Vyshali Lasitha *et al.* A Review Paper on Comparison of Algorithms for Supervised Machine Learning. *Journal of Artificial Intelligence Research & Advances*. 2017; 4(3): 11–16p.