Available online at http://www.advancedscientificjournal.com
http://www.krishmapublication.com
IJMASRI, Vol. 2, issue10, pp. 690-695,Oct -2022
https://doi.org/10.53633/ijmasri

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY ADVANCED SCIENTIFIC RESEARCH AND INNOVATION (IJMASRI)

ISSN: 2582-9130 **IBI IMPACT FACTOR 1.5** **DOI: 10.53633/IJMASRI**

RESEARCH ARTICLE

COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS ON HEART ATTACK DATASET

Mohd Talib Akhtar
School of Engineering, Science and Technology
Jamia Hamdard University, New Delhi, India
mohdtalibakhtar147@gmail.com

Abstract

The goal of this research paper is to compare different Machine Learning Algorithms on heart attack dataset. In this Project, some popular Machine Learning Algorithms were used like Logistic Regression, Kernel Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest, Naïve Bayes, & XG Boost and conclude the best working algorithm by comparing the accuracies.

Keywords*:* Machine Learning, heart dataset, algorithms, comparative analysis

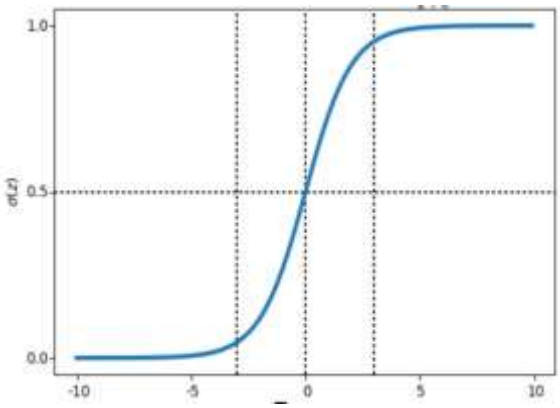
Introduction

Machine Learning is a type of Artificial Intelligence (AI) that gives computers the capability to learn without being explicitly programmed (Burns 2021; Education 2020). Machine Learning is actively being used today, perhaps in every field like medical, business, weather forecast, etc. We programmed the machine by giving the precursory data and Machine predicts the forthcoming. Machine Learning term was initially used by Arthur Samuel, A colonist of Artificial Intelligence at IBM in 1959. There is plethora of ML Algorithms which are often designated by their type of output i.e., classification & regression, and their type of data i.e., supervised & unsupervised. Researchers have used different algorithms in ML according to expertise, availability, and the dataset (Sethi 2017). In this project, our problem is classification problem, and we use supervised data. The most predominant & perplex part of ML project is to choose the Algorithm for the model. To compare our Algorithms, we cleave the dataset into 80% training data and 20% test data and compare the accuracy, sensitivity & specificity.

2. Classifiers

2.1 Logistic Regression

Logistic Regression is very primary yet very constructive ML Algorithm. It works on the on the concept of probability (Saxena 2021) It is similar to Linear regression, but the Logistic Regression uses a complex cost function, which is 'Sigmoid Function' or also known as 'Logistic Function' instead of linear function (Pant 2019).



$$f(x) = \frac{1}{1 + e^{-(x)}}$$

Formula for sigmoidal function

Sigmoid Function uses to map predicted values to probabilities, It maps any real value into another value between 0 to 1 (Saxena 2021)

While mapping the value between 0 to 1, we reduced the cost function and in order to do that gradient descent comes into the picture, its goal is to minimize the cost value.

2.2 K-Nearest Neighbour (KNN)

K Nearest Neighbour is one of the most simplest Machine Learning algorithm. It is a non-parametric algorithm, which states that it does not make any assumption on underlying data (Harrison, O. (2018). It stores the dataset and when a new data point comes, it classifies it based on its similarity.

How does KNN works:

- Load the dataset.
- Select the number of K of the neighbours.
- Calculate the distance of K neighbours of new data point.
- Amidst these K neighbours, count the number of data points in each category.
- Assign the new data point to the category where sum of all the neighbours is more than the other one.
- Model is ready (Srivastava 2018)

2.3 Support Vector Machine (SVM)

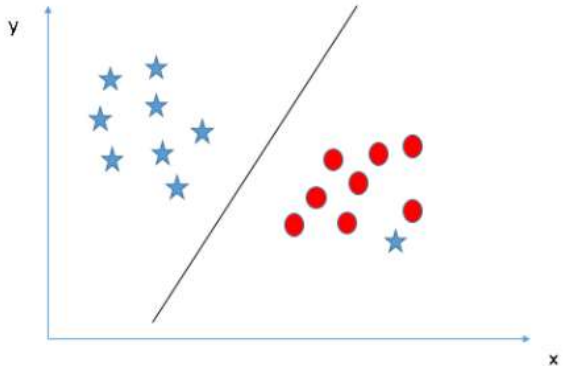
Support Vector Machine is a supervised ML algorithm, which can be brought to bear both classification and regression problems (Stecanella 2017). The goal of SVM is to find a hyperplane, a hyperplane is a best line or decision that can set apart N-dimensional space into classes to situate a new data point in the unerring category.

Types of Support Vector Machine (SVM):

2.3.1 Linear SVM

It is used for data that can be classified into two classes by the help of straight line, it is also known as linearly separable data and classifier is called Linear SVM. Here we use two dimensions i.e., x & y and disparate the data by using a single line.

$$k(x, xi) = sum(\dot{x} * xi)$$

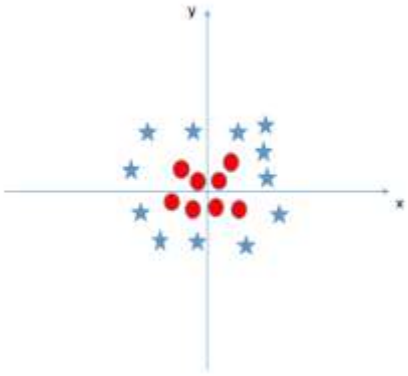


2.3.2 Non-Linear SVM

As the name suggest, this algorithm used for Non-linearly separable data and the classifier known as Non-linear SVM. Here we can't use a single line to distinct the data points hence, we generate one dimension 'z' and compute its value by squaring both the dimensions and add (Gandhi 2018).

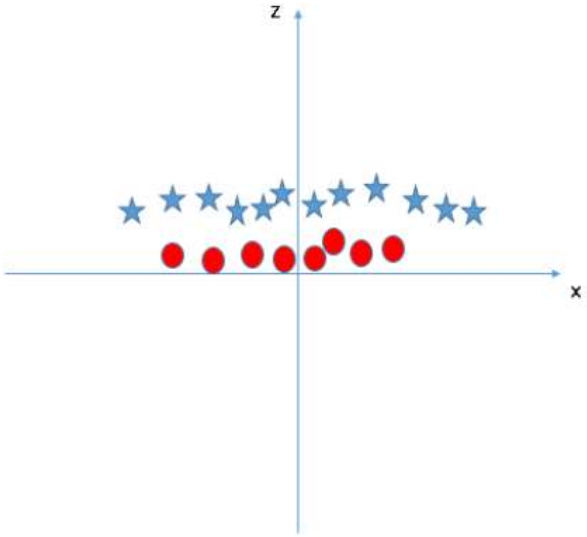
$$z = x^2 + y^2$$

For e.g.,



$$F(x, x_j) = \exp\left(-gamma^* \|x - x_j\|^\Lambda 2\right)$$

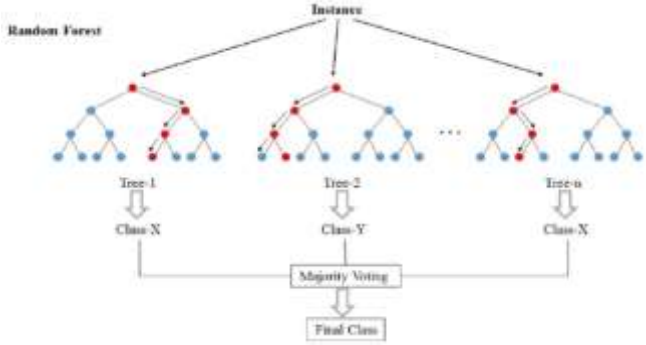
Here we can't categorize the data into divergent categories. So, in order to do that we have to initiate an additional feature, z by squaring both the dimensions and add.



2.4 Random Forest

Random forest algorithm is a group of decision trees or in the other words decision trees are the building blocks of the random forest model, where each tree proffers a class prediction (Géron 2017). The class which contains most number of plebiscite can be infer as our prediction model.

$$E(s) = -\sum p(x)^* \log 2\big(p(x)\big)$$



In the above picture, there are three decision tree and every decision transfire with a class, in the same way there are n number of decision trees and the class with majority of votes will be our output (Pawar 2020).

2.5 Naïve Bayes

Naïve Bayes is a probabilistic machine learning model, works on Bayes Theorem. It conjectures the existence of a feature in a class is unallied to the existence of any other feature.

$$P_{(C|x)} = \frac{P_{(x|C)}P_{(C)}}{P(x)}$$

Above, $P_{(C|x)}$ is posterior probability.

$P_{(x|C)}$ is the likelihood which is the probability of predictor class.

$P_{(C)}$ is the prior probability of class.

$P(x)$ is the prior probability of predictor (Ray 2017).

2.6 XG Boost

XG Boost is an ensemble tree method that apply the principle of boosting weak learnings using the gradient descend architecture. It also optimizes standard Gradient Boosting Machines (GBM) Algorithm (Morde 2019).

3. Proposed Methadology

In this section we will elucidate the steps in detail that we will follow to build this project.

3.1 Dataset

We took this heart dataset from UCI repository which contains 13 columns like chest pain, blood pressure, cholesterol etc.

3.2 Data Preprocessing

Data preprocessing is exceedingly important step as it gives the insight of our dataset because in healthcare datasets, there are chances that data might be missing and other impurities that can cause effectiveness of data.

3.3 Splitting the dataset

After apprehension the data, we divide it into 80% training data, to train our ML model and 20% test data to test the trained model.

3.4 Standardization

Standardization rescales the data from 0 mean to standard deviation of 1 unit. We use this method to make sure that data are scaled on a same factor.

3.5 Algorithms

Now we use different algorithms on training dataset to train our model.

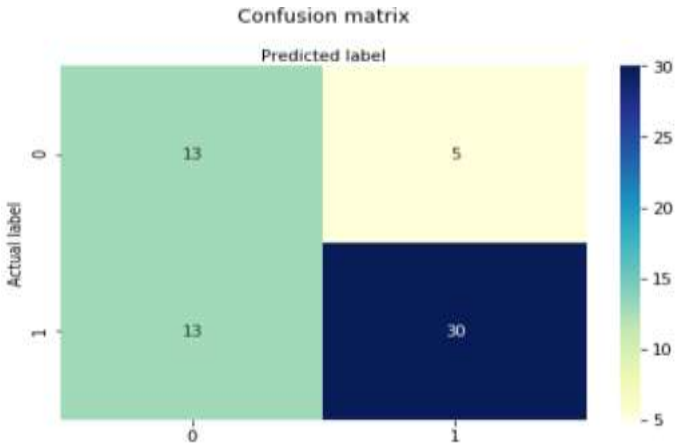
3.6 Comparison

After building ML models, its time to check their accuracies and find out which one is best working algorithm.

4. Result & Comparison

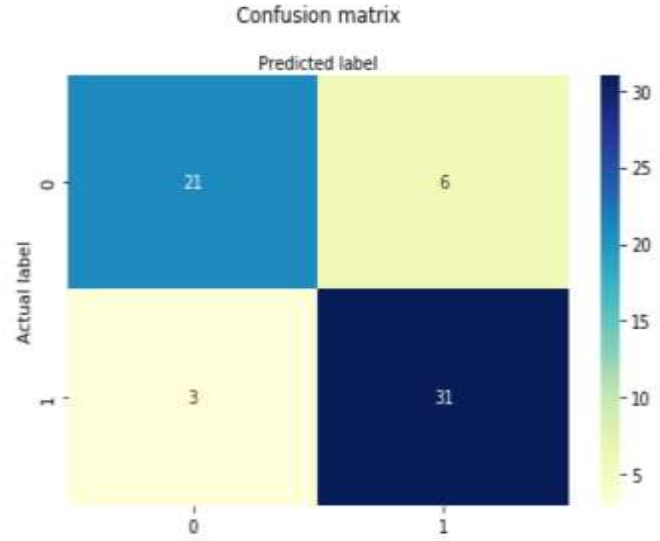
The most interesting part of this study is here, after applying different ML algorithm on the dataset, which algorithm outnumbered every other algorithm and able to get the highest accuracy. Let's start with Naïve Bayes

Naïve bayes gave us the accuracy of 70.49%, Sensitivity of 72.22% & Specificity of 69.76%. As our first model, it shows some decent results.



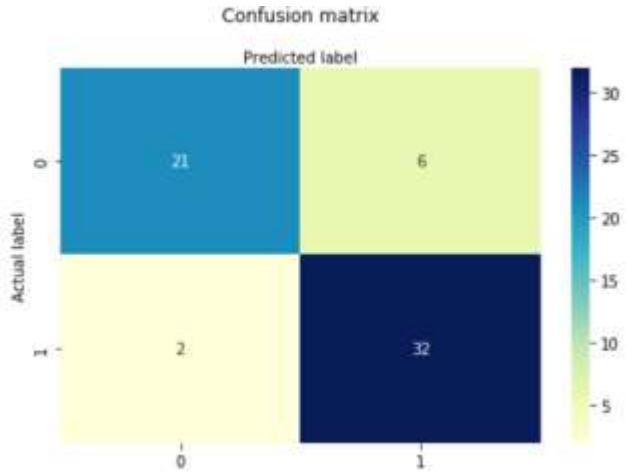
Logistic regression:

This gave the accuracy of 85.24%, sensitivity of 77.77% and specificity of 97.17%, This shows great improvement than Naïve Bayes.



Kernel SVM:

This model worked way better than Logistic regression, with an accuracy of 86.88%, sensitivity of 77.77% and specificity of 94.11%. This model also shows some improved result and considerable amount of satisfaction.

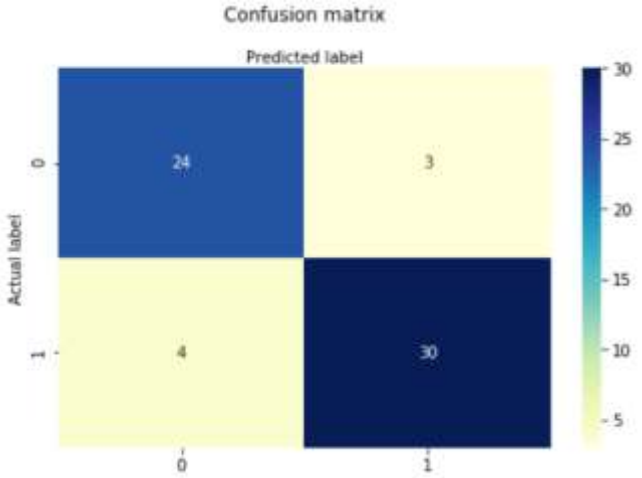


Random Forest:

In this model we used 5 numbers of decision trees with the criterion of entropy and got the accuracy of 85.24%, sensitivity of 85.18% and specificity of 85.29%. This model also worked really well with great results.

K-Nearest Neighbor (KNN)

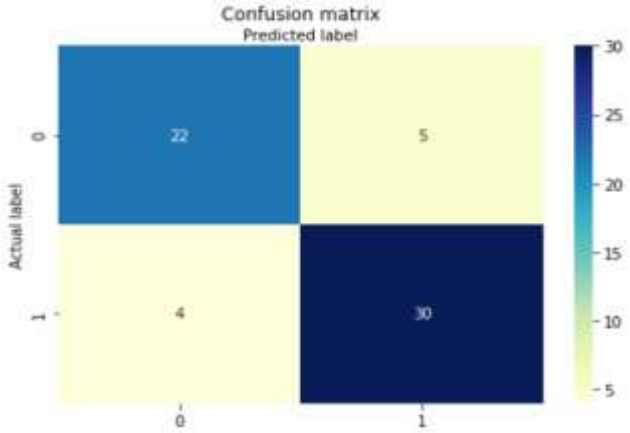
In this model, we use 10 neighbors with metric of minkowski, and this shows significant results with an accuracy of 88.52%, sensitivity of 88.88% and specificity of 88.23%. So far KNN is pre-eminent algorithm with these results.



XG Boost

Conclusion

XG Boost comes up with the accuracy of 85.24%, Sensitivity of 81.48% and Specificity of 88.23%. This result is nearly same as Random Forest and the reason of it is that both are based on ensemble



Confusion Matrix

It consists of 4 sections:

True Positive : These are the cases where we predicted true and its actually true.

True Negative : These are cases where we predicted is true but its false

False Positive: Where we predicted False but it's true.

True Negative: Where we predicted false and it's actually false.

Table 1. Accuracy, Sensitivity & Specificity of dataset.

	Accuracy	Sensitivity	Specificity
Naïve Bayes	70.49%	72.22%	69.76%
Logistic	85.24%	77.77%	91.17%
Kernel SVM	86.88%	77.77%	94.11%
Random Forest	85.24%	85.18%	85.29%
KNN	88.52%	88.88%	88.23%
XG Boost	85.24%	81.48%	88.23%

So far, we explore the models and their working, and without a doubt we can say that K-

Nearest Neighbor overcome every other model with the great accuracy of 88.52% but we can't ignore the fact that Logistic regression, Random Forest & XG Boost gave the same accuracy of 85.24% so we can say that the average accuracy is around 85%.

These models are really useful in the field of science, law, medical fields, etc.

References

1. Burns, E. (2021). *Machine Learning*. Retrieved from Search Enterprise AI: https://searchenterpriseai.techtarget.com/definition/machine-learning-ML.
2. Education, I. C. (2020). *Machine Learning* Retrieved from IBM: https://www.ibm.com/in-en/cloud/learn/machine-learning
3. Sethi, K. (2017). Comparative Analysis of MachineLearning Algorithms on Different Datasets. *International Conference on Innovations in Computing*, 87-91.
4. Saxena, S. (2021). *Data Scientist's Guide to Logistic regression*. Retrieved from Analytics Vidhya : https://www.analyticsvidhya.com/blog/2021/03/logistic -regression/
5. Pant, A. (2019). *Introduction to Logistic Regression*. Retrieved from towards data science: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148
6. Saxena, S. (2021). *Data Scientist's Guide to Logistic regression*. Retrieved from Analytics Vidhya : https://www.analyticsvidhya.com/blog/2021/03/logistic -regression/
7. Harrison, O. (2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Retrieved from Towards Data Science: https://towardsdatascience.com/machine-learning basics-with-the-k-nearest-neighbors-algorithm- 6a6e71d01761
8. Srivastava, T (2018). *Introduction to k-Nearest Neighbors: A powerful Machine Learning Algorithm (with implementation in Python & R)*.Retrieved from Analytics Vidhya - Learn everything about Analytics: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
9. Stecanella, B. (2017). *An Introduction to Support Vector Machines (SVM)*. Retrieved from MonkeyLearn: https://monkeylearn.com/blog/introduction-to-support -vector-machines-svm/
10. Gandhi, R. (2018). *Support Vector Machine — Introduction to Machine Learning Algorithms*. Retrieved from Towards Data Science: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.
11. Géron, A. (2017). Random Forests. In A. Géron, *Hands- On Machine Learning with Scikit-Learn* (pp. 240-242).Sebastopol, California: O'Reilly Media, Inc.
12. Pawar, U. (2020). *Lets Open the Black Box of Random Forests*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/12/lets-open-the-black-box-of-random-forests/
13. Ray, S. (2017). *6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/09/naive- bayes-explained/
14. Morde, V. (2019). *XGBoost Algorithm: Lo May She Reign!* Retrieved from Towards Data Sciecne: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-

*****