iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IX **Month of publication:** September 2021

DOI: https://doi.org/10.22214/ijraset.2021.37976

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

Hard Disk Drive Failure Detection Using Hybrid Algorithm

Vaibhav Umesh Mokal[1], Dr. Prashant Nitnaware[2]
[1, 2]*Pillai College of Engineering, Maharashtra*

Abstract: ***The data is the most valuable thing in this modern world of Information Technology. As we can see the day to day the data is increasing as each and every people using the World Wide Web. This all system generated data or may be the personal or informative data will get generated in a huge amount of size. That data will get stored at the data centers or on cloud. But those will get stored on the Hard Disk Drives in data centers. So in some situation if the HDD got crashed then we will have lost our data. This work proposes to develop the failure prediction of Hard disk drive. We have chosen the accuracy and review measurements, generally important to the issue, and tried a few learning strategies, Adaboost, Naive Bayes, Logistic Regression and Voting. Our investigation shows that while we can't accomplish close to 100% forecast precision utilizing ML with the present information we have accessible for HDDs, we can improve our expectation exactness over the standard methodology***
Keywords: Machine learning, Adaboost, Naive Bayes, Voting, Logistic Regression

I. INTRODUCTION

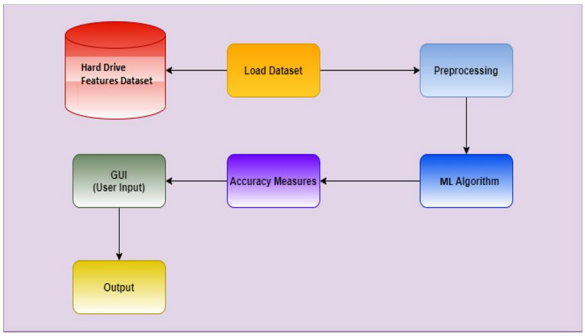
IT sector has become the major driving factor of today's world economy. Any business or work by any company may be it tech or non tech i.e. technical company that build software's or non tech companies that focus on non tech products like food, clothing etc. they all need computers i.e. they all need the IT department. No business, organization can work without computers in 21th century. Computers have affected us so much that even are quotidian life cannot function without it. It has become our day-to-day necessity. Humans cannot imagine life without computers. "Computing is not only computing anymore it's about human living". (Nicholas Negroponte, Being Digital, 1995, p.78) This Computer basically consist of two components i.e. software and hardware.

Hard drive is component where all the information is stored. Big organization performs a great amount of work using computers. Hence, a huge chunk of info in form of data are stored on these Hard Disk. These Hard-drives have a good history of malfunctioning. So what if these hard drives are damaged somehow and stop working suddenly, the important/critical data stored will be completely gone. There is almost 1 Exabyte of data stored on cloud according to the result of research done by Nasuni (A survey of security and privacy challenges in cloud computing: solutions and future directions, 2015).

One of the major threat considered in cloud computing is data loss and data breaches. (Jing Li, 2014). According to a recent survey conducted, 63% of customers will not purchase a cloud service if it has some history of sensitive data loss or data breached. " (Jing Li, 2014). 78% of hardware substitution has been recorded for hard drives by Microsoft in a study done by (Nicolas Aussel, 2017)

II. METHODOLOGY

The predefined Hard Drives features of dataset is available. Among that data set system will select the parameters for the detection of the HDD failures. We are using the data set that will load and it will go into the pre-processing stage. After that we are using the multiple algorithms to detect the failure of HDD. Result of this will be captured and based on that system will measure the accuracy. Intension behind the applying the multiple algorithms is to get the more accurate result.



©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

A. Pre-processing Data

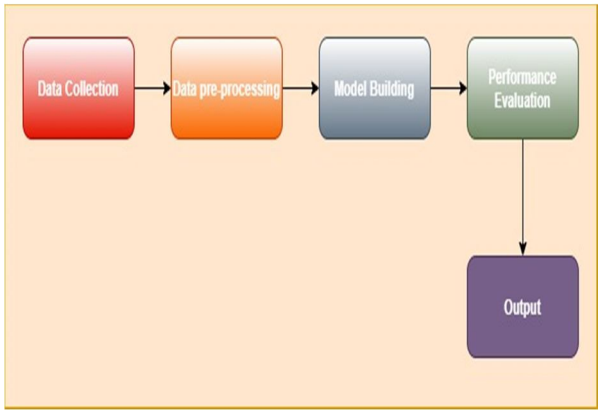
Raw data collected are susceptible to missing values, noisy data, incomplete data, inconsistent data and outlier data. So, it is important for these data to be processed. In pre-processing the collected dataset is divided into two parts, one for testing and other for training purpose. The dataset is labelled. The pre-processing stage involves feature extraction and scaling, feature selection, dimensionality reduction and sampling. Pre-processing helps in cleaning data to be made fit for further processing.

Steps involved in pre-processing

1) Feature extraction
2) Feature selection
3) Dimensionality reduction
4) Sampling

B. Model Building

In this step a model can be built using different machine learning techniques. We also use Ensemble Classifiers in the process to predict hard drive failure. Ensemble Classifier is a classifier formed by combining two or more different classifiers. This newly created model or more precisely the hybrid model can be far more efficient than each one them to be used alone. In this hybrid model different learning and methods are used to produce N-different models using a single dataset. In a nutshell, the system than combines the output of all these models to get the most efficient output. The output obtained is a weighted average of all inputs for each model.



III. EXPERIMENTATION

The Combination of Algorithm used by us are:

A. ADABOOST

One of the most widely used and studied the ADABoost algorithm by Freund and Schapire was the first practical boosting algorithm with a large number of application in numerous fields. (Robert E. Schapire). AdaBoost is a well-known boosting strategy that encourages you join numerous "feeble classifiers" into a solitary "solid classifier". A powerless classifier is essentially a classifier that performs inadequately yet performs superior to arbitrary speculating. AdaBoost can be utilized to any description calculation, so it's a technique that grows over different classifiers as a substitute of being a classifier itself.

We can prepare a lot of frail classifiers all alone and consolidate the outcomes. It helps you pick the preparation ready for each new classifier that you train reliant on the aftereffects of the past classifier. It chooses how much weight ought to be submitted to every classifier's proposed reply when joining the outcomes.

Each powerless classifier ought to be able on an arbitrary subset of the absolute preparing set. The subsets can include it's not comparable to, for instance, dividing the preparation set into ten bits. AdaBoost allocates a "weight" to each preparation model, which takes the likelihood that each model should to show up in the preparation set. Models with greater loads are bound to be remembered for the preparation set, and the other way around. Subsequent to preparing a classifier, AdaBoost constructs the weight on the misclassified models with the aim that these models will make up a larger piece of the next classifiers' preparation set, and preferably, the following classifier prepared will do better on them.

©IJRASET: All Rights are Reserved

After each single classifier is prepared, the classifier's weight is decided dependent on its accuracy. Gradually precise classifiers are given extra weight. A classifier along with half exactness is offered a load of zero, and a classifier with under half precision (sort of an amusing idea) is given a negative weight.
We should look at the situation for the last classifier.

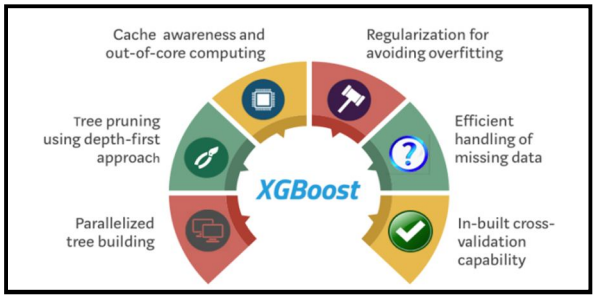
$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

The attribute D_t is a vector of loads, with one capacity for each preparation model in the preparation set. 'I' is the training model number. This situation tells you the finest way to restore the weight for the it is preparing model.

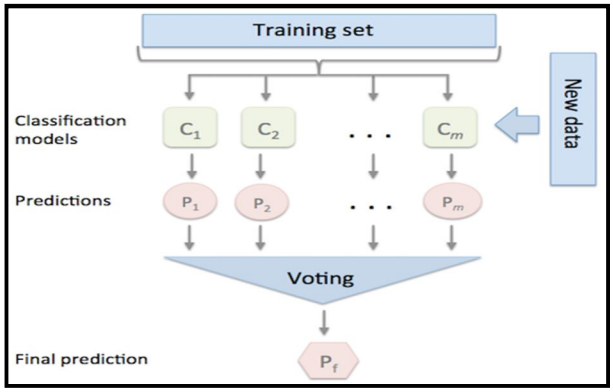
B. XGBOOST

XGBoost is a choice tree-based group Machine Learning calculation that uses an angle boosting structure. In expectation issues including unstructured information (pictures, content, and so on.) counterfeit neural systems will in general beat every other calculation or structures. In any case, with regards to little to-medium organized/forbidden information, choice tree-based calculations are viewed as top tier at this moment. XGBoost calculation was created as an exploration venture at the University of Washington. Tianqi Chen and Carlos Guestrin exhibited their paper at SIGKDD Conference in 2016 and discovered the ML world by fire. XGBoost and Gradient Boosting Machines (GBMs) are both group tree techniques that apply the standard of boosting powerless students (CARTs for the most part) utilizing the angle drop design. Be that as it may, XGBoost enhances the base GBM structure through frameworks streamlining and algorithmic improvements.



C. Voting

Voting is definitely the most straightforward method for consolidating expectations from numerous AI calculations. The democratic classifier isn't a genuine classifier yet a wrapper for a lot of various ones that are prepared and assessed in parallel so as to abuse the various quirks of every calculation.



We can train data set using different algorithms and ensemble then to predict the final output. The final output on a prediction is done by majority in voting according to two strategies: hard voting and soft voting.

©IJRASET: All Rights are Reserved

D. SVM

SVM is a classification approach for supervised learning. The goal of the SVM is to classify or separate the given data on basis of some feature selection using a hyperplane. For differentiating between two types of data point, the hyper plane can be a straight line, with both types of data point on either side of it. The structure of this hyperplane can be changed on the basis of different types of data points given. It's basically a type of classification algorithm which draws a line between different types of data given to classify and identify the given data. If the number of input data points or features is 2, then a line is used as a hyperplane for classification. If the number of input data points is 3, then the hyperplane transforms from a line to two dimensional plane.

E. Naïve Bayes

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the most simplified supervised learning algorithm. It is also one of the most accurate reliable and fast algorithm. It performs best when applied to large datasets in terms of accuracy and speed.

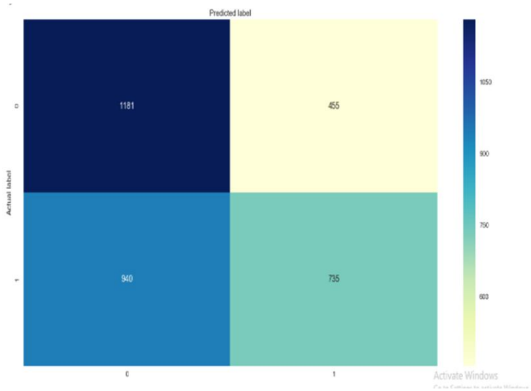
Naive Bayes classifier assumes that each and every feature's result is independent of every other feature in a particular class. It doesn't matter whether the features are dependent and independent, these features are considered independently. This assumption simplifies computation, and that's why it is considered as naive.

We have discussed all the algorithms used by us in creating our system. We have also discussed the drawbacks of previous papers. We have seen how much the strength of dataset can have an impact on having a high accuracy.

By keeping all these things in mind we will discuss the implementation of the combination of algorithms discussed by us here, in the next session.

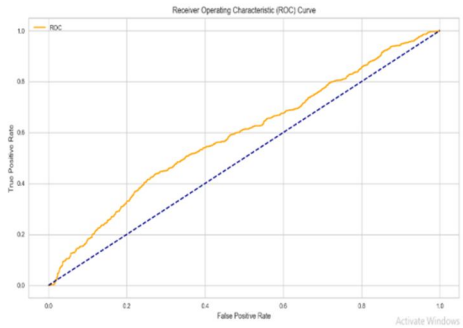
IV. RESULTS AND DISCUSSION

Logistic Regression: Accuracy: 57.86771368166717



Confusion Matrix(Logistic Regression)

As per the above fig there are 1181 cases that are been predicted correctly with its respective class (0) and 455 cases are predicted wrongly as class(1) of respective class(0). Our classifier has predicted 940 cases incorrect as class(0) of respective class(1) and 735 cases correctly as class(1).



AUC score : 0.58

©IJRASET: All Rights are Reserved

Logistic Regression: Classification report

precision	recall	f1-score	support	
Yes	0.56	0.72	0.63	1636
No	0.62	0.44	0.51	1675

accuracy			0.58	3311
macroavg	0.59	0.58	0.57	3311
weightedavg	0.59	0.58	0.57	3311

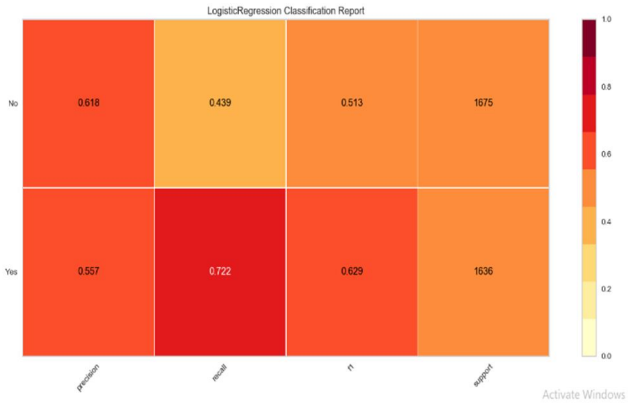


Figure 11: Classification report(Logistic Regression)

Report shows precision, recall and f1-score for the two resultant classes. For Class(0) they are 0.618, 0.439 and 0.513 and 0.557, 0.722, 0.629 for Class(1) respectively.

Naive Bayes: Accuracy: 57.95832074901842
Naive Bayes: Confusion Matrix

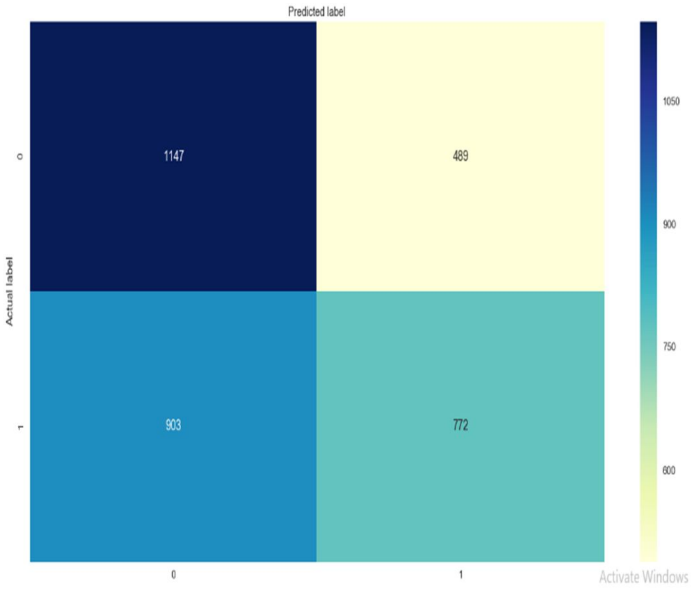


Figure 12: Confusion Matrix(Naive Bayes)

As per the above there are 1147 cases that are been predicted correctly with its respective class (0) and 489 cases are predicted wrongly as class(1) of respective class(0). Our classifier has predicted 903 cases incorrect as class(0) of respective class(1) and 772 cases correctly as class(1).

©IJRASET: All Rights are Reserved

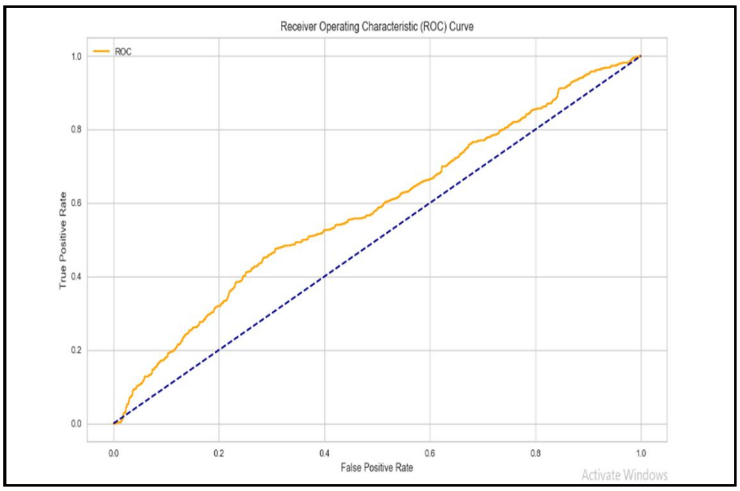


Figure 13: ROC-Curve(Naive Bayes)

AUC score : 0.58

Naïve Bayes: Classification report

	precision	recall	f1-score	support
Yes	0.56	0.70	0.62	1636
No	0.61	0.46	0.53	1675

accuracy			0.58	3311
Macro avg	0.59	0.58	0.57	3311

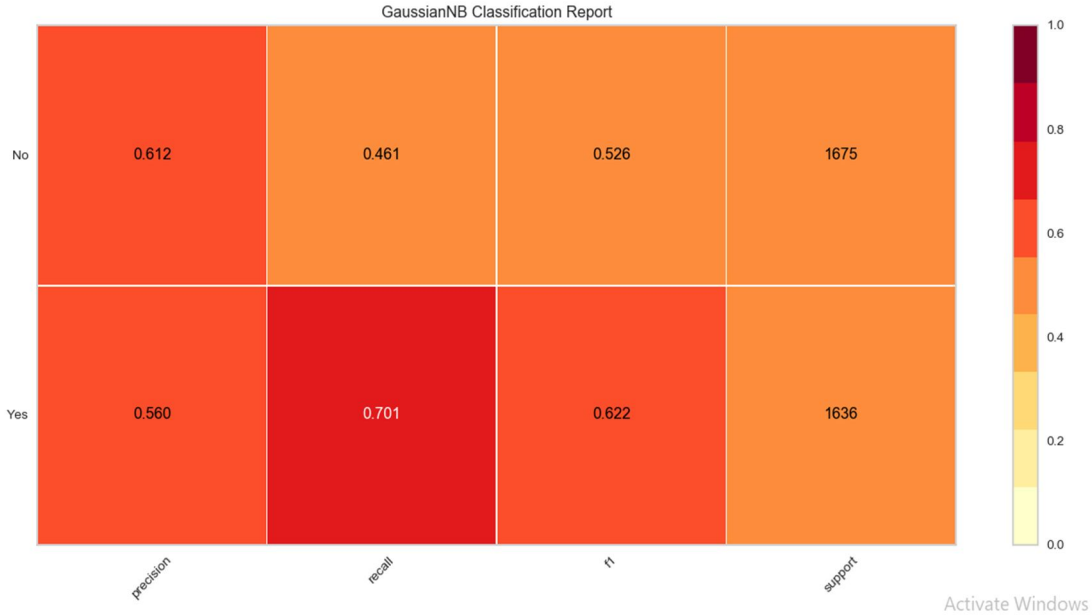
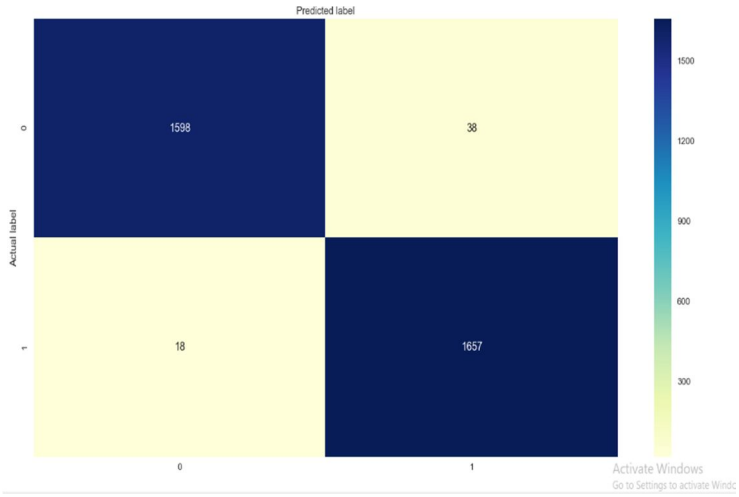


Figure 14: Classification Report(Naive Bayes)

Report shows precision, recall and f1-score for the two resultant classes. For Class(0) they are 0.612, 0.461 and 0.526 and 0.560, 0.701, 0.622 for Class(1) respectively.

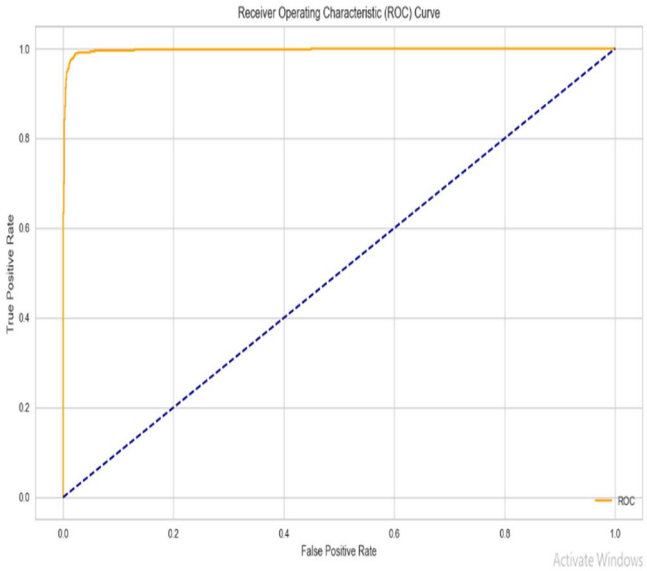
©IJRASET: All Rights are Reserved

Ada Boost: Accuracy: 98.30866807610994
Ada Boost: Confusion Matrix



Confusion Matrix(Adaboost)

As per figure 15 there are 1598 cases that are been predicted correctly with its respective class (0) and 38 cases are predicted wrongly as class(1) of respective class(0). Our classifier has predicted 18 cases incorrect as class(0) of respective class(1) and 1657 cases correctly as class(1).



ROC-Curve(Adaboost)
AUC score : 1.00
ADABoost Classification Report

	precision	recall	f1-score	support
Yes	0.99	0.98	0.98	1636
No	0.98	0.99	0.98	1675

accuracy			0.98	3311
Macro avg	0.98	0.98	0.98	3311
Weighted avg	0.98	0.98	0.98	3311

©IJRASET: All Rights are Reserved

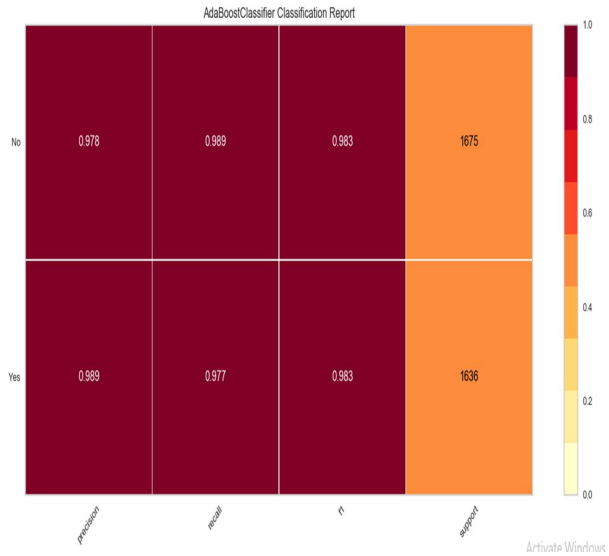


Figure 17: Classification report(ADABoost)

Report shows precision, recall and f1-score for the two resultant classes. For Class(0) they are 0.978, 0.989 and 0.983 and 0.989, 0.977, 0.983 for Class(1) respectively.

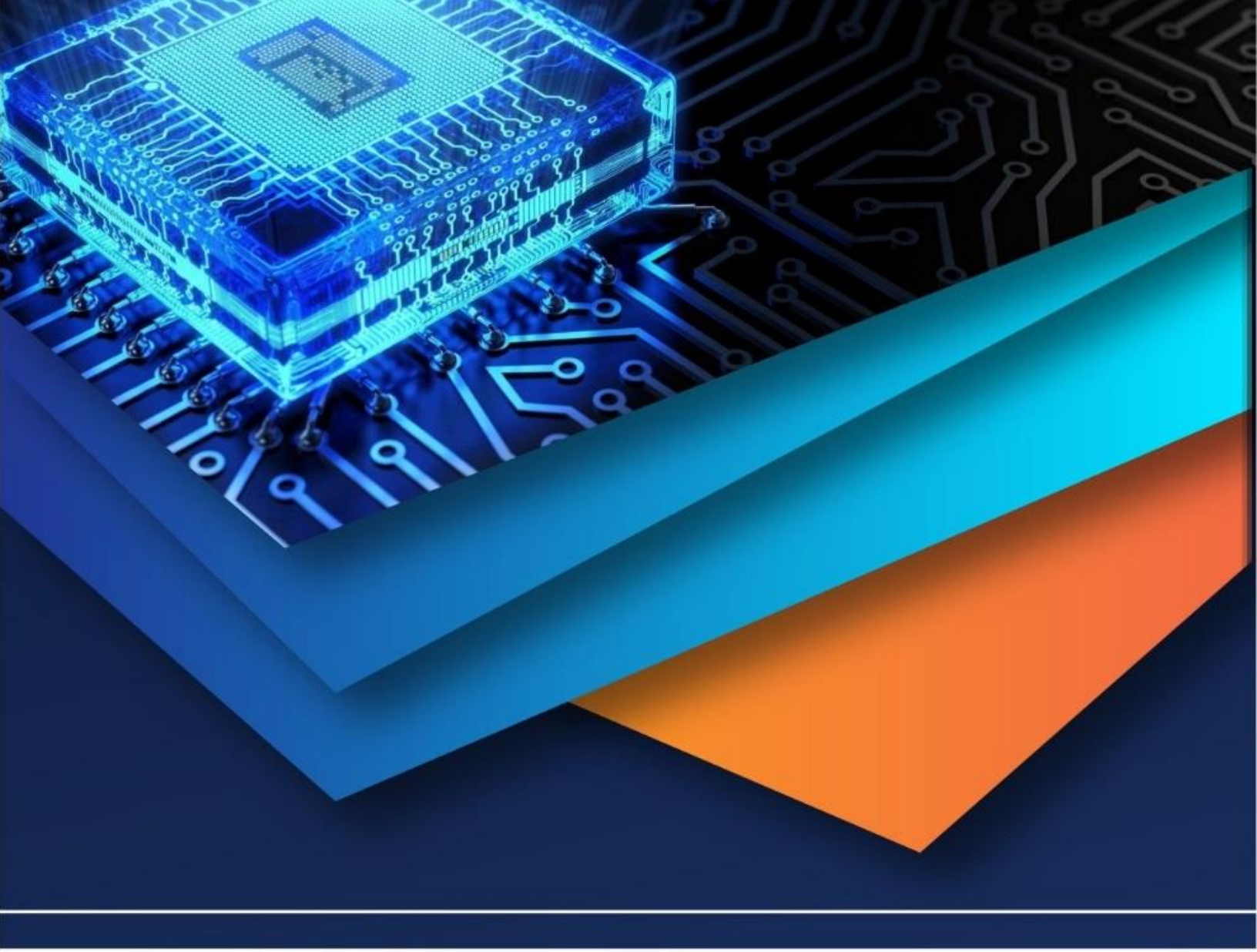
V. CONCLUSIONS

In this project, we have chosen the accuracy and review measurements, generally important to the issue, and tried a few learning strategies, Adaboost, Naive Bayes, Logistic Regression and Voting. Our investigation shows that while we can't accomplish close to 100% forecast precision utilizing ML with the present information we have accessible for HDDs, we can improve our expectation exactness over the standard methodology. ML calculations are equipped for giving progressively exact expectations of HDD disappointments, with promptly accessible information, then what is right now executed in the present business. From the Implementation we concluded that the Voting ensemble hybrid algorithm gives the best result as compared to the other algorithms.

REFERENCES

[1] Anantharaman, P., Qiao, M. & Jadav, D., 2018. Large Scale Predictive Analytics for Hard Disk Remaining Useful Life Estimation. IEEE.
[2] B.Y, V. & Borah, A., 2016. Enhanced Rules Framework for Predicting Disk Drives Failures. International Journal of Computer Science and Mobile Computing.
[3] Hamerly, G. & Elkan, C., 2003. Bayesian Approaches to Failure Prediction for Disk Drives. IEEE.
[4] He, X., Wang, Z. & Zhang, J., 2011. Research on security of hard disk firmware. IEEE.
[5] He, Z., Yang, H. & Xie, M., 2012. Statistical modeling and analysis of hard disk drives (HDDs) failure. IEEE.
[6] Hughes, G., 2005. Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. IEEE.
[7] Jing Li, X., 2014. Hard Drive Failure Prediction Using Classification and Regression Trees. IEEE, p. 12.
[8] Li, J., Li, Z. & Wang, G., 2014. Hard Drive Failure Prediction Using Classification and Regression Trees. 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).
[9] Li, W., 2017. Proactive Prediction of Hard Disk Drive Failure. IEEE.
[10] Mahdisoltani, F., Stefanovici, I. & Schroeder, B., 2017. Improving Storage System Reliability with Proactive Error Prediction. IEEE.
[11] Murray, J. F., Hughes, G. F. & Kreutz-Delgado, K., 2005. Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. IEEE.
[12] Nicolas Aussel, S. J. G. G. Y. P. E. F. S. C., 2017. Predictive Models of Hard Drive Failures Based on Operational Data. IEEE International Conference on Machine Learning and Application.
[13] Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. IEEE.
[14] Shen, J., Wan, J., Lim, S.-J. & Yu, L., 2018. Random-forest-based failure prediction for hard disk drives. SAGE journals.
[15] Strom, B. D., Lee , S. & Tyndall, G. W., 2007. Hard Disk Drive Reliability Modeling and Failure Prediction. IEEE.
[16] Suchatpong, T. & Bhumkittipich, K., 2014. Hard Disk Drive failure mode prediction based on industrial standard using decision tree learning. IEEE.
[17] Suchatpong, T. & Bhumkittipich, K., 2014. Hard Disk Drive Failure Mode Prediction based on Industrial Standard using Decision Tree Learning. IEEE.
[18] Sun, F.-b. & Zhang, S., 2007. Does Hard Disk Drive Failure Rate Enter Steady-State Afer One Year?.
[19] Wang, Y., Jiang , S., Long , H. & Peng, Y., 2019. Hard Disk Drives Failure Detection Using A Dynamic Tracking Method. IEEE.
[20] Xiao, J., Xiong, Z. & Wu, S., 2018. Disk Failure Prediction in Data Centers via Online Learning. ICPP.
[21] Yigit, I. O., Arslan, Ş. Ş. & Ze, E., 2018. A visualization platfom for disk failure analysis. IEEE.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)