International Engineering Journal For Research & Development

Vol.5
Issue 2

E-ISSN NO:-2349-0721

Impact factor : 6.03

SCRUTINIZE PERFORMANCE COMPARISON OF RANDOM FOREST AND NAIVE BAYES FOR DIFFERENT TEST MODE USING WEKA ON THE DATASET OF CAR REVIEWS

Prof. Sushilkumar R. Kalmegh
sushil.kalmegh@gmail.com
Associate Professor, Dept. of Computer Science, Sant Gadge Baba Amravati University,
Amravati, M.S., India

ABSTRACT-

The amount of data in the world and in our lives seems ever-increasing and there's no end to it. We are overwhelmed with data. The WWW overwhelms us with information. The Size of information base is increasing day by day with fast speed. The WEKA is data processing tool contain equipped series of state of art machine learning algorithm. The basic way of interacting with these methods is by invoking them from the command line. However, convenient interactive graphical user interfaces are provided for data exploration, for setting up large-scale experiments on distributed computing platforms, and for designing configurations for streamed data processing. This paper has been carried out to make a performance evaluation of Random Forest from Trees Classifier and Naive Bayes from Bayes Classifier algorithm with different test modes. The test mode used in this research work is Use Training set, 10-folds cross validation. The paper sets out to make comparative evaluation of Random Forest and Naive Bayes in the context of dataset of car reviews to maximize true positive (TP) rate and minimize false positive (FP) rate. The WEKA tools used for result processing. The result in the paper on dataset of car reviews shows that the efficiency and accuracy of Random Forest is excellent as compared to Naive Bayes.

Index Terms----**Classification, Data mining, Naive Bayes, Random Forest, WEKA.**

I. INTRODUCTION

The development of various applications and demand of internet is the main source of information generation. Today Computers make it too easy to save things. Inexpensive disks and online storage make it too easy to postpone decisions about what to do with all this stuff, we simply get more memory and keep it all. The data mining help us to store such type of data in computerized form. Data mining is a topic that involves learning in a practical, non theoretical sense. The researchers are interested in techniques for finding pattern from data. The new tools are also available to find the prediction from such huge data. Such available data is also called as machine learning tool. Recently various ecommerce platform software and application provide data in the form of product review it is available in the textual format provided by expert, user and customer. A product rating on the other hand represents the customer's and expert opinion on a sampling scale. In this given research paper car review data set [1] was used. Comparative analysis of RandomForest and Naive Bayes with different test mode such as Use Training set, 10-folds cross validation has been carry out.

As the volume of data increases, inexorably, the proportion of it that people understand decreases alarmingly. Lying hidden in all this data is information. In data mining, the data is stored electronically and the search is automated or at least augmented by computer. Even this is not particularly new. Economists, statisticians, and communication engineers have long worked with the idea that patterns in data can be sought automatically, identified, validated, and used for prediction. What is new is the staggering increase in opportunities for finding patterns in data. Data mining is a topic that involves learning in a practical, non theoretical sense. We are interested in techniques for finding and describing structural patterns in data, as a tool for helping to explain that data and make predictions from it. Experience shows that in many applications of machine learning to data mining, the explicit knowledge structures that are acquired, the structural descriptions, are at least as important as the ability to perform well on new examples. People frequently use data mining to gain knowledge, not just predictions.

Data Mining is an incredible innovation with extraordinary capacity to assist associations with concentrating on the most significant data in their data focus. It additionally foresee future patterns, conduct and with result. It likewise contains assortment of diagnostic tools that utilized for data investigation. It enables clients to dissect the data from various perspectives, sort it, and outline the distinguished connections. There are numerous Data Mining tools are available, for example the WEKA, KNIME, Orange, SPSS, MATLAB, and NeuroShell and so on. These tools give a lot of Data Mining strategies and calculations that help in better execution of data and data accessible to clients. The accessible Data Mining tools can be partitioned into two kinds which are open source/non-business programming and business programming. These kinds of tools have their very own qualities and shortcomings regarding data types and the application techniques. From the given arrangement of tool in investigation work WEKA tool have utilized. This paper is organized into Six parts. First part discusses the Introduction followed by the literature required for analysis of methods implemented. Third one is System Design followed by datasets used for analysis. Fifth is the Performance Analysis and then Conclusion.

E-ISSN NO:2349-0721

II LITERATURE SURVEY

A. WEKA

Weka was developed at the University of Waikato in New Zealand; the name stands for Waikato Environment for Knowledge Analysis The system is written in Java and distributed under the terms of the GNU General Public License. It runs on almost any platform and has been tested under Linux, Windows, and Macintosh operating systems and even on a personal digital assistant. It provides a uniform interface to many different learning algorithms, along with methods for pre and post processing and for evaluating the result of learning schemes on any given dataset. Weka provides implementations of learning algorithms that can be easily apply to dataset. It also includes a variety of tools for transforming datasets, such as the algorithms.

The Weka workbench is a collection of state-of-the-art machine learning algorithms and data pre processing tools. It is designed so that we can quickly try out existing methods on new datasets in flexible ways.

It provides extensive support for the whole process of experimental data mining, including preparing the input data, evaluating learning schemes statistically, and visualizing the input data and the result of learning. As well as a variety of learning algorithms, it includes a wide range of pre processing tools. This diverse and

comprehensive toolkit is accessed through a common interface so that its users can compare different methods and identify those that are most appropriate for the problem at hand. All algorithms take their input in the form of a single relational table in the ARFF format. The easiest way to use Weka is through a graphical user interface called Explorer as shown in figure I. This gives access to all of its facilities using menu selection and form filling.

Fig. I: WEKA GUI Explorer

The Weka contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality. Advantages of Weka include:

- Free availability under the GNU General Public License
- Portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform.
- A comprehensive collection of data preprocessing and modeling techniques.
- Ease of use due to its graphical user interfaces.

Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported). Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. Weka's main user interface is the Explorer, but essentially the same functionality can be accessed through the component-based Knowledge Flow interface and from the command line. There is also the Experimenter, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets. The Explorer interface features several panels providing access to the main components of the workbench. Figure II shows Opening of file Car_Review.arff file by Weka Explorer and Figure III shows processing of arff file for RandomForest Classifier. [1],[2],[3].

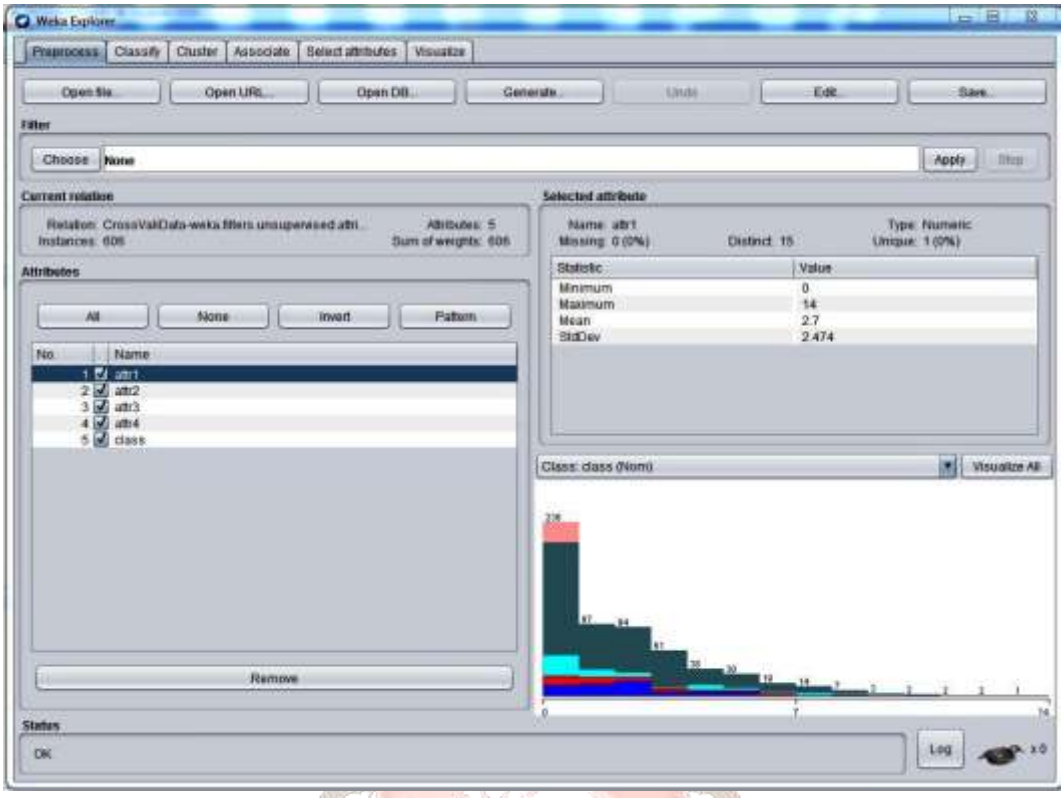


Fig. II: Opening of Car_Review.arff file by Weka Explorer

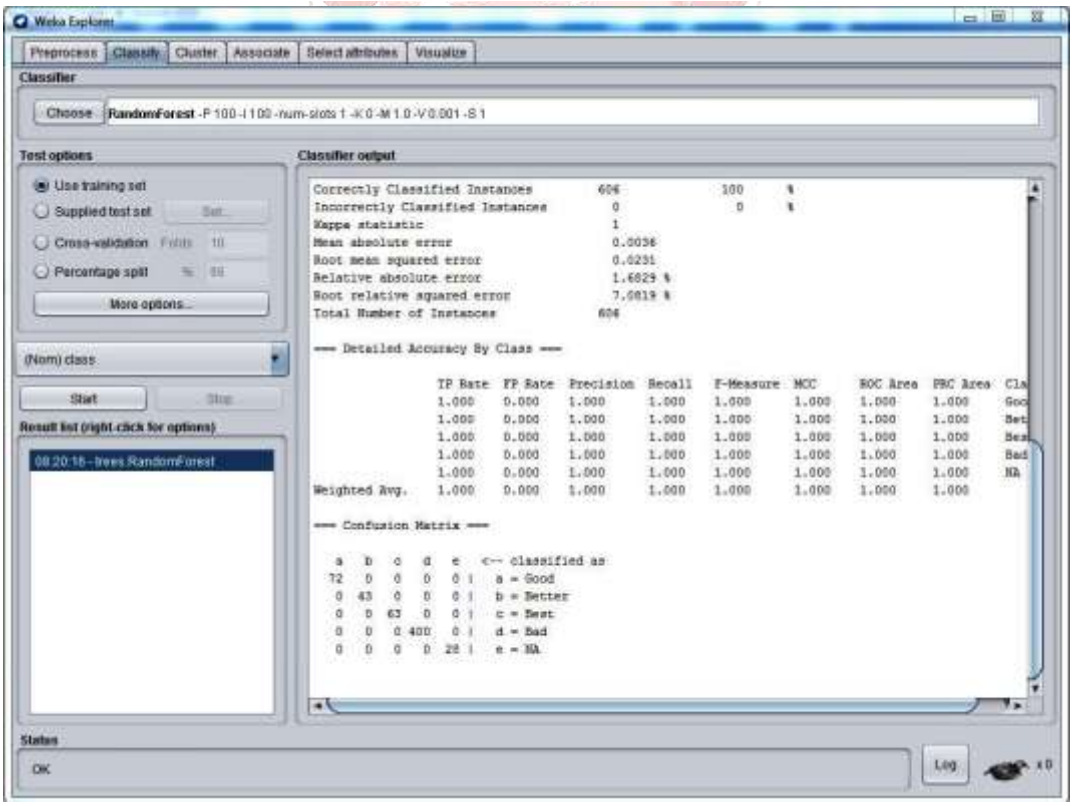


Fig. III : Processing of arff file by RandomForest Classifier on Test Mode Use Training Set

Classification

Classification may refer to categorization, the process in which ideas and objects are recognized, differentiated, and understood. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering or cluster analysis, and involves grouping data into categories based on some measure of inherent similarity. Classification is a data mining algorithm that creates a step-by-step guide for how to determine the output of a new data instance. The tree it creates is exactly that: a tree whereby each node in the tree represents a spot where a decision must be made based on the input, and to move to the next node and the next until one reach a leaf that tells the predicted output. Sounds confusing, but it's really quite straightforward. There is also some argument over whether classification methods that do not involve a statistical model can be considered "statistical". Other fields may use different terminology: e.g. in community ecology, the term "classification" normally refers to cluster analysis, i.e. a type of unsupervised learning, rather than the supervised learning [1],[2],[3],[4].

i. Random Forest Classifiers:

Random Forest is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large. Random Forest (RF) is a special kind of ensemble learning techniques and robust concerning the noise and the number of attributes. Random Forest is an ensemble learning method for classification (and regression) that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. The algorithm for inducing a Random Forest was developed by Leo Breiman and Adele Cutler, and "Random Forests" is their trademark. The term came from random decision forest that was first proposed by Tin Kam Ho of Bell Labs in 1995. The method combines Breiman's "bagging" idea and the random selection of features, introduced independently by Ho and Amit and Geman in order to construct a collection of decision trees with controlled variation. RF builds an ensemble of CART tree classifications using bagging mechanism. By using bagging, each node of trees only selects a small subset of features for the split, which enables the algorithm to create classifiers for high dimensional data very quickly. This somewhat counterintuitive strategy turns out to perform very well compared to the state-of-the-art methods in classification and regression. Also, RF runs efficiently on large data sets with many features and its execution speed is fast. RF produces additional facilities, especially the variable importance by numerical values.

The key idea of the regularization framework is to penalize selecting a new feature for splitting when its gain (e.g. information gain) is similar to the features used in previous splits. The regularization framework is applied on Random Forest and boosted trees here, and can be easily applied to other tree models. Experimental studies show that the regularized trees can select high-quality feature subsets with regard to both strong and weak classifiers. Because tree models can naturally deal with categorical and numerical variables, missing values, different scales between variables, interactions and nonlinearities etc., the tree regularization framework provides an effective and efficient feature selection solution for many practical problems. [2],[3],[5],[6],[7],[8].

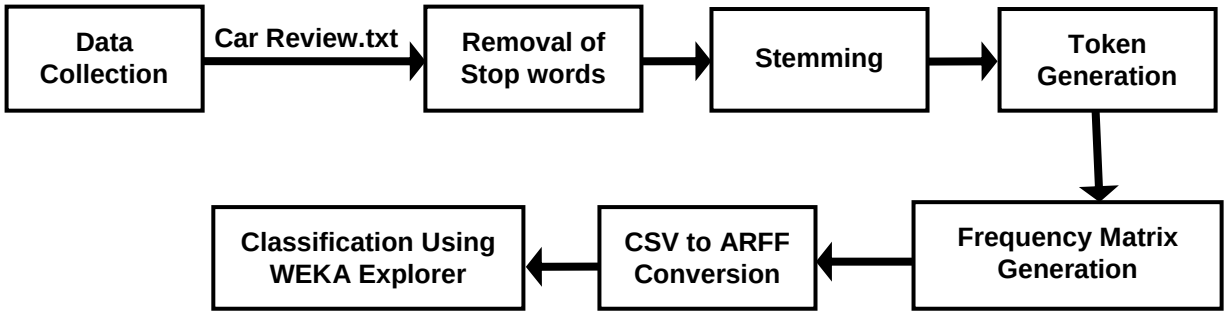
ii. Naive Bayes Classifiers:

Naive Bayes implements the probabilistic Naive Bayes classifier. Naive Bayes Simply uses the normal distribution to model numeric attributes. Naive Bayes can use kernel density estimators, which improve performance if the normality assumption is grossly incorrect; it can also handle numeric attributes using supervised discretization. The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. Its assumption that attributes are conditionally independent given a particular class value means that the overall class probability is obtained by simply multiplying the per-attribute conditional probabilities together (and taking into account the class prior probabilities as well). By default, Weka's Naive Bayes classifier assumes that the attributes are normally distributed given the class. It is particularly suited when the dimensionality of the inputs is high. Parameter estimation for Naive Bayes models uses the method of maximum likelihood. In spite oversimplified assumptions, it often performs better in many complex real world situations. Naive Bayes has been denigrated as the punching bag of classifiers, and has earned the dubious distinction of placing last or near last in numerous head-to-head. Still, it is frequently used for text classification because it is fast and easy to implement. Less erroneous algorithms tend to be slower and more complex. Naive Bayes selects poor weights for the decision boundary. This is due to an under-studied bias effect that shrinks weights for classes with few training examples. Another systemic problem with Naive Bayes is that features are assumed to be independent. As a result, even when words are dependent, each word contributes evidence individually. Thus the magnitude of the weights for classes with strong word dependencies is larger than for classes with weak word dependencies. To keep classes with more dependencies from dominating, we normalize the classification weights.

Naive Bayes has advantages (i) Fast to train (single scan). Fast to classify, (ii) Not sensitive to irrelevant features, (iii) Handles real and discrete data, (iv) Handles streaming data well and the disadvantage, assumes independence of features [2],[6],[9],[10],[11],[12].

III. System Design

In order to co-relate Reviews with the categories, a model based on the machine learning was designed. As an input to the model, various quality car reviews are considered which are available online. Around 606 car reviews samples were collected on above repository using internet. In order to extract context from the car reviews, the car reviews was process with stop word removal, stemming and tokenization on the car reviews contents. The car reviews then separated into 5 categories GOOD, BETTER, BEST, BAD, NA (not applicable) and then converted into the term frequency matrix for further analysis purpose. Frequency matrix then converted to arff file using Java Programming. Finally classification is processed using WEKA Explorer; this can be seen in following figure IV. Due to classification in above 5 categories we are also able to find the GOOD, BETTER, BEST, BAD, NA count on every data set which help for market analysis, product rating and much more purposes. Based on this data, features (i.e. metadata) were extracted so that contextual assignment of the car reviews to the appropriate content can be done. This process is known as metadata processing [1],[8].



IV. Data Collection

Hence, it was proposed to generate car reviews data. Consequently the national and international resources were used for the research purpose. Data for the purpose of research has been collected from the various online resources using internet. They are downloaded and after reading the car reviews they are manually classified into 12 (Twelve) categories. There are 606 car reviews in total. The details are as shown in following table I. The attributes consider for this classification is based on GOOD, BETTER, BEST, BAD, NA count each classification having their own data dictionary and based on this they are classified, the review are made by expert and user. Hence, there will be drastic enhancement in e-Contents when we refer to the latest material available in this regards [1],[8].

Table I: Categorization of Car Review Dataset

Sr.. No.	Car Companies	Numbers of Reviews
1	Chevrolet	38
2	Fiat	27
3	Ford	36
4	Honda	47
5	Hyundai	59
6	Mahindra & Mahindra	63
7	Maruti Suzuki	95
8	Renault	53
9	Skoda	23
10	Tata Motors	90
11	Toyota	41
12	Volkswagan	34
Total		**606**

V. Performance Analysis

The Data so collected needed a processing. Hence as given in the system design phase, all the 606 data were processed for stop word removal, stemming, tokenization and ultimately generated the frequency matrix based on GOOD, BETTER, BEST, BAD and NA count. Stemming is used as many times when Car Review Data is printed, for a same there can be many variants depending on the tense used or whether it is singular or plural.

Such words when processed for stemming, generates a unique word. Stop words needs to be removed as they do not contribute much in the decision making process. The dictionary of words is checked and removed such word from it. Frequency matrix thus generated can be processed for generating a model by converting CSV to ARFF file and the model so generated was used in further decision process. The two different test mode i.e. i) Use Training set and ii) 10-folds cross validation used for RandomForest and Naive Bayes. For processing WEKA APIs were used. The following tables shows the Confusion Matrix and True positive (TP) and False Positive (FP) rate of RandomForest and Naive Bayes. In the given result the 1.0 represent the BEST, whereas the WORST is 0.0. The following table II shows the summary of Classification.

The following tables III, V, VII and IX show the result for the Confusion Matrix and the Tables IV, VI, VIII, and X show True Positive and False Positive rate of RandomForest and Naive Bayes for test mode: i) Use Training set and ii) 10-folds cross validation.

Table II: Summary of Classification

Classifier	**RandomForest**		**NaïveBayes**	
Test Mode	**Use Training Set**	**10-Fold Cross Validation**	**Use Training Set**	**10-Fold Cross Validation**
Correctly Classified Instances	**606 (100%)**	**597 (98.51%)**	**553 (91.25%)**	**540 (89.11%)**
Incorrectly Classified Instances	**00 (0%)**	**09 (1.49%)**	**53 (8.75%)**	**66 (10.89%)**

Table III: Confusion Matrix for RandomForest for Test Mode: Use Training Set

Classified As ➡	**GOOD**	**BETTER**	**BEST**	**BAD**	**NA**
GOOD	**72**	0	0	0	0
BETTER	0	**43**	0	0	0
BEST	0	0	**63**	0	0
BAD	0	0	0	**400**	0
NA	0	0	0	0	**28**

Table IV: TP and FP Rate of RandomForest for Test Mode: Use Training Set

Class ⬇	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**
GOOD	1.000	0.000	1.000	1.000	1.000	1.000
BETTER	1.000	0.000	1.000	1.000	1.000	1.000
BEST	1.000	0.000	1.000	1.000	1.000	1.000
BAD	1.000	0.000	1.000	1.000	1.000	1.000
NA	1.000	0.000	1.000	1.000	1.000	1.000
Weighted Avg. ➡	1.000	0.000	1.000	1.000	1.000	1.000

Table V: Confusion Matrix for RandomForest for Test Mode: 10-Fold Cross Validation

Classified As ➡	GOOD	BETTER	BEST	BAD	NA
GOOD	**70**	0	0	2	0
BETTER	0	**42**	1	0	0
BEST	0	0	**60**	3	0
BAD	0	0	2	**398**	0
NA	1	0	0	0	**27**

Table VI: TP and FP Rate of RandomForest for Test Mode: 10-Folds Cross Validation

Class ⬇	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
GOOD	0.972	0.002	0.986	0.972	0.979	0.985
BETTER	0.977	0.000	1.000	0.977	0.988	0.988
BEST	0.952	0.006	0.952	0.952	0.952	0.999
BAD	0.995	0.024	0.988	0.995	0.991	1.000
NA	0.964	0.000	1.000	0.964	0.982	0.982
Weighted Avg. ➡	0.985	0.017	0.985	0.985	0.985	0.996

Table VII: Confusion Matrix for Naive Bayes for Test Mode: Use Training Set

Classified As ➡	GOOD	BETTER	BEST	BAD	NA
GOOD	**70**	0	0	2	0
BETTER	0	**36**	4	3	0
BEST	0	0	**28**	35	0
BAD	0	7	1	**392**	0
NA	0	0	0	1	**27**

Table VIII: TP and FP Rate of Naive Bayes for Test Mode: Use Training Set

Class ⬇	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
GOOD	0.972	0.000	1.000	0.972	0.986	0.989
BETTER	0.837	0.012	0.837	0.837	0.837	0.977
BEST	0.444	0.009	0.848	0.444	0.583	0.986
BAD	0.980	0.199	0.905	0.980	0.941	0.979
NA	0.964	0.000	1.000	0.964	0.982	0.994
Weighted Avg. ➡	0.913	0.133	0.910	0.913	0.904	0.981

Table IX: Confusion Matrix for Naive Bayes for Test Mode: 10-Folds Cross Validation

Classified As ➡	GOOD	BETTER	BEST	BAD	NA
GOOD	69	1	0	2	0
BETTER	0	27	4	12	0
BEST	0	0	26	37	0

BAD	0	6	3	391	0
NA	0	0	0	1	27

Table X: TP and FP Rate of Naive Bayes for Test Mode: 10-Folds Cross Validation

Class ⬇	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**
GOOD	0.958	0.000	1.000	0.958	0.979	0.988
BETTER	0.628	0.012	0.794	0.628	0.701	0.958
BEST	0.413	0.013	0.788	0.413	0.542	0.951
BAD	0.978	0.252	0.883	0.978	0.928	0.967
NA	0.964	0.000	1.000	0.964	0.982	0.994
Weighted Avg. ➡	0.891	0.169	0.886	0.891	0.880	0.969

IV . CONCLUSION

In this paper as per the previous performance analysis, Table II Summary of Classification shows that the Classifier Random Forest has the accuracy for test mode evaluate on training data is 100% & for 10-Fold Cross validation is: 98.51% and the Classifier Naive Bayes has accuracy for test mode evaluate on training data is 91.25% & for 10-Fold Cross validation is 89.11%. This 100% accuracy for test mode evaluate on training data for the Classifier Random Forest is achieved due to Random Forest is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Overall Performance of Naive Bayes algorithm is acceptable, except some of News from every category are classified into other category. This is because Naive Bayes Simply uses the normal distribution to model numeric attributes so the accuracy for test mode evaluate on training data is 91.25%.

For 10-Fold Cross validation in both the Classifier the accuracy decreases. The reason for this is that, in 10-fold cross-validation, the original sample is randomly partitioned into 10 subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 10 – 1 (i.e. 9) sub samples are used as training data. The cross-validation process is then repeated 10 times (the folds), with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds then can be averaged (or otherwise combined) to produce a single estimation. From all the above result in the Table II to Table X, it is observed that performance of Classifier Random Forest is Excellent as compared to Classifier Naive Bayes.

REFERENCES

[1] S.A.Ghogare and Dr.S.R.Kalmegh,-- Comparative analysis of J48 and LMT classifier using WEKA data mining tool on car Review Data, 'Research Journey' International E- Research Journal, ISSN: 2348-7143, Special Issue 110 (C)-Computer Science, pp-99-105, February 2019.

[2] Ian H. Witten, Eibe Frank & Mark A. Hall, Data Mining Practical Machine Learning Tools and Techniques, (Third Edition, Morgan Kaufmann Publishers is an imprint of Elsevier, 2016).

[3] Sushilkumar Rameshpant Kalmegh,-- Comparative Analysis of WEKA Data Mining Algorithm RandomForest, RandomTree and LADTree for Classification of Indigenous News Data,-- International Journal of Emerging Technology and Advanced Engineering (IJETAE), ISSN 2250-2459, Vol. 5, Iss. 1, pp- 507-517, January 2015.

[4] http://en.wikipedia.org/wiki/Classification.

[5] en.wikipedia.org/wiki/Random_forest.

[6] Tarannum A Bloch, Prof. V. B. Vaghela, Dr. K. H. Wandra,-- Applied Taxonomy Techniques Intended for Strenuous Random Forest Robustness, Int. J. Comp. Tech. Appl. (IJCTA), ISSN:2229-6093, Vol. 2 (6), pp-2061-2065, NOV-DEC 2011

[7] Sneh Lata Pundir, Amrita,-- FEATURE SELECTION USING RANDOM FOREST IN INTRUSION DETECTION SYSTEM, International Journal of Advances in Engineering & Technology (IJAET), ISSN: 2231-1963, Vol. 6, Issue 3, pp--1319-1324, July 2013.

[8] S.A.Ghogare, Dr.S.R.Kalmegh, Performance Comparison of RandomForest and Hoeffding Tree classifier using WEKA data mining tool on Car reviews data, International Journal of Engineering Research and Applications (IJERA), ISSN:2248-9622, Vol 09, Iss 03, pp-37-42, March 2019.

[9] Sushilkumar Rameshpant Kalmegh,-- Effective Classification Of Indian News Using Classifier Hyperpipe and Naivebayes From Weka, International Journal of Pure and Applied Research in Engineering and Technology (IJPRET), ISSN: 2319-507X, Vol. 4, Iss. 9, pp- 364-378, March 2016.

[10] Pat Langley, Stephanie Sage,-- Tractable Average Case Analysis of Naive Bayesian Classifiers, ICML 99, Proceedings of the Sixteenth International Conference on Machine Learning, pp-220-228, June 1999

[11] Uzair Bashir & Manzoor Chachoo,-- PERFORMANCE EVALUATION OF J48 AND BAYES ALGORITHMS FOR INTRUSION DETECTION SYSTEM, International Journal of Network Security & Its Applications (IJNSA), Vol 9, No. 4. pp-1-11, July 2017.

[12] Mahendra Tiwari, Manu Bhai Jha and OmPrakash Yadav,-- Performance analysis of Data Mining algorithms in WEKA, IOSR Journal of Computer Engineering(IOSRJCE), 2012, ISSN: 2278-0661, ISBN: 2278-8727, Vol 6, Iss 3, pp- 32-41, Sep-Oct. 2012.