ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Optimization of Naïve Bayes Data Mining Classification Algorithm

Maneesh Singhal[#1], Ramashankar Sharma[#2]

Department of Computer Engineering, University College of Engineering,
Rajasthan Technical University, Kota, Rajasthan, INDIA

Abstract— As a probability-based statistical classification method, the Naïve Bayesian classifier has gained wide popularity; however, the performance of Naive Bayes classification algorithm suffers in the domains (data set) that involve correlated features. [Correlated features are the features which have a mutual relationship or connection with each other. As correlated features are related to each other, they are measuring the same feature only, means they are redundant features]. This paper is focused upon optimization of Naive Bayes classification algorithms to improve the accuracy of generated classification results with reduced time to build the model from training dataset. The aim is to improve the performance of Naive Bayes algorithms by removing the redundant correlated features before giving the dataset to classifier. This paper highlights and discusses the mathematical derivation of Naive Bayes classifier and theoretically proves how the redundant correlated features reduce the accuracy of the classification algorithm. Finally, from the experimental reviews using WEKA data mining software, this paper presents the impressive results with significant improvement into the accuracy and time taken to build the model by Naive Bayes classification algorithm.

Keywords— classification, Naive Bayes, Correlated Redundant Features, CFS algorithm, Classifier Prediction Accuracy, WEKA

I. INTRODUCTION

There has been extensive research over the classification of data across multiple domains as it has the capabilities to predict the class of a new dataset with unknown class by analysing its structural similarity. Multiple classification algorithms have been implemented, used and compared for different data domains, however, there has been no single algorithm found to be superior over all others for all data sets for different domain.

Naive Bayesian classifier represents each class with a probabilistic summary and finds the most likely class for each example it is asked to classify. It is known that Naive Bayesian classifier works very well on some domains, and poorly on others. The performance of Naive Bayesian suffers in domains that involve redundant correlated and/or irrelevant features. If two or more attributes are highly correlated, they receive too much weight in the final decision as to which class an example belongs. This leads to a decline in accuracy of prediction in domains with correlated features. Several researchers have emphasized the issue of redundant attributes and it has been shown that Naive Bayesian classifier is extremely effective in practice and difficult to improve upon.

The primary motive of this paper is to understand the Naive Bayesian classifier, Conceptual understanding of redundant correlated and/or irrelevant features, performance impact of redundant correlated and/or irrelevant features over the Naive Bayesian classifier, To explore the various methods as suggested by multiple researchers to improve the performance of Naive Bayesian classifier, Identification of the best suitable approach towards optimization of Naive Bayesian classifier for the domains that involve redundant correlated and/or irrelevant features and Finally, performing different experiments to confirm the suitability of the proposed solution.

II. THEORETICAL EVALUATION

Naïve Bayes classifier function [1] [2] has been defined as below

$$\text{classify}(f_1, \ldots, f_n) = \underset{c}{\operatorname{argmax}}\; p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c).$$

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Based upon the above Naïve Bayes classifier function, we would present an approach to OPTIMIZE the Naive Bayes classification algorithm by removing the redundant correlated and irrelevant features so that algorithm can be applied/used with a significant improvement in the domain which involves correlated features.

A. Sample Classification Problem

Given a list of candidates for an interview process (Candidate's current designation and Years of experience), an university wanted to decide whether the candidate can be offered a permanent position (Tenured) OR Not!

TABLE 1

TRAINING DATASET

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Here, "Tenured" is a Group / Class to which each record (candidate) will be assigned to.

1) Constructing a Model and Classification of New Data: classification is a 2 step process.

Step 1

- Construct / Build a Model based upon the supplied data / training data set.
- Training data is a set of records where the Group / Class of each record is already KNOWN to us.

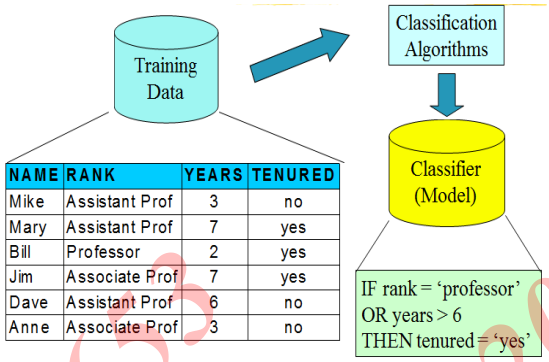

Fig. 1 classification model based upon the training data set from table 1

Step 2

- Use / Apply the model (Built in Step-1) to classify the new data / test data.
- Test data is a set of records where the Group / Class of each record is NOT KNOWN to us.
- Classification will help us to identify the Group / Class of records from test data.

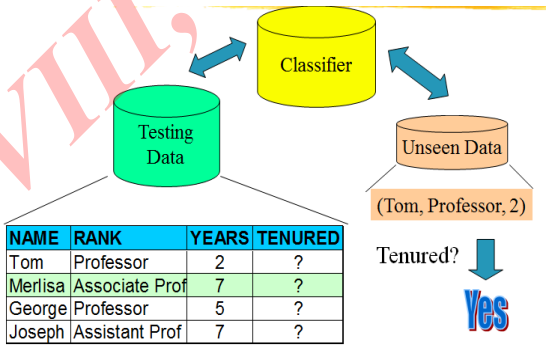

Fig. 2 classification of new data (Tom, Professor, 2) based upon the classification model from Fig. 1

B. Classification Approach Based Upon Mathematical Derivation

Revisiting the Naïve Bayes classifier function which has been defined as below -

$$\text{classify}(f_1, \ldots, f_n) = \text{argmax}\; p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c).$$

Total No of Features / Attributes: f1…fn

Total No of Class / Group to which a record can be assigned to: c

We can conclude that there are two features / attributes being used in the above training data set

Feature f1 = Rank
Feature f2 = Years

A new data record would be assigned to a Class TENURED, Whether Yes / Not.

Class: TENURED = Yes | NO

Calculating the probability of the Class – TENURED being Yes / No, based upon the existing records from training data set:

P (Tenured = Yes) = 3/6 = 0.5

P (Tenured = No) = 3/6 = 0.5

We wanted to identify the class to be assigned for a new data record: "*Tom, Professor, 2*"

As there are two classes (TENURED = Yes | NO) to which Tom can be assigned to, we will calculate the two probabilities as below. The greater probability will decide to which class Tom will be assigned to.

Probability of Tom being *TENURED = Yes*

P(Tenured = Yes) * P(Rank = Professor | Tenured=Yes) * P(Years <= 6 | Tenured=Yes)

= 3/6 * 1/3* 1/3

= 0.056

Probability of Tom being *TENURED = No*

P(Tenured = No) * P(Rank = Professor | Tenured=No) * P(Years <= 6 | Tenured=No)

=3/6 * 0 * 1

= 0

Probability of Tom being *Tenured = Yes* is 0.056 which is greater than another probability. Hence, Tom will be Tenured to Yes.

C. *Naïve Bayes Optimization Based Upon Mathematical Derivation*

Revisiting the probability of Tom being *TENURED = Yes* from previous section -

Probability of Tom being *TENURED = Yes*

= P(Tenured = Yes) * P(Rank = Professor | Tenured=Yes) * P(Years <= 6 | Tenured=Yes)

Above classification expression can be generalized as below -

Class "Yes" -> Has been replaced with class C1

Feature "Rank" has been replaced with f1

Feature "Years" has been replaced with f2

P(C1) * P(f1 | C1) * P(f2 | C1)

If there is another feature f3 in the training data, classification expression will become:

P(C1) * P(f1 | C1) * P(f2 | C1) * P(f3 | C1)

Now consider if feature f3 is correlated with feature f1, means both f1 and f3 are measuring the same underlying feature, say f0, hence, replacing f1/f3 with f0 will result into following classification expression

P(C1) * P(f0 | C1) * P(f2 | C1) * P(f0 | C1)

"Above mathematical classification expression Proves that the feature f0 has twice as much influence on the classification expression as feature f2 has, which is a strength not reflected in reality. The increased strength of f0 may make the classification algorithm to calculate the incorrect class and hence the total accuracy of the algorithm will get impacted when number of redundant correlated and irrelevant features are increased in the training data"

As feature f0 is redundant correlated, hence, removing the multiple instances of feature f0 from the classification expression as below -

P(C1) * P(f0 | C1) * P(f2 | C1)

Based upon the theoretical evaluation of the classification expression, we can conclude that removing of redundant correlated and irrelevant features from the

data set would result into AN IMPROVEMENT of Naïve Bayes algorithm.

Redundant correlated features wouldn't be included while constructing the classification model, resulting into TIME OPTIMISATION. Less time would be required to build the classification model as total number of features would be reduced.

Number of features to build the classification model =

Total features in training data set - redundant correlated features

Removing the redundant correlated features ensures that the remaining features which are used to build the classification model, would have an equal impact, hence, the *ACCURACY OF THE ALGORITHM WOULD BE IMPROVED SIGNIFICANTLY*.

From the classification expression, P(C1) * P(f1 | C1) * P(f2 | C1) * P(f3 | C1), feature f1 and f3 are redundant correlated features measuring the same underlying feature, hence, only a single feature (Either f1 OR f3) to be considered for building the classification model so that the remaining features (Here, f2) would have an equal impact over the classification result.

D. Theoretical Evaluation - Summary

Theoretical Evaluation, based upon the Naïve Bayes classification expression, proves that DEFINITE performance improvement can be achieved in the Naïve Bayes algorithm through the identification of the redundant correlated features from the training data set and excluding these redundant correlated features from the process of constructing the classification model.

Generated classification model would require less time due to the reduced features and when this classification model is applied for a new data set, would improve an overall accuracy of the classification results.

In the next chapter, we would go through an experimental exercise using the WEKA [3] [4] software for a sample data set to verify the theoretical conclusion we have summarized here. An analysis of statistical results from the experimental exercise would confirm that the classification approach as presented in this paper can be extended over the live classification problems.

III. EXPERIMENTAL EVALUATION AND STATISTICS

Theoretical evaluation from previous section, which was based upon the mathematical derivation of the Naive Bayes classification algorithm, has been evaluated using WEKA data mining software.

This section describes about the dataset to be used for building the classification model, WEKA data mining software, Different experimental reviews using WEKA software and Multiple statistical results through graphical representation to support the mathematical and experimental analysis for performance improvement of Naive Bayes classification algorithm.

A. Introduction to WEKA: Data Mining Software

WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. WEKA contains software for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

WEKA, a data mining software written in Java, is used extensively into research and academics and this is open source software issued under the GNU General Public License.

WEKA is capable to provide a practical evaluation of a classification algorithm based upon the different statistics, as follows:

- Classification accuracy (In %) [5]
- Time taken for classification (In minutes/seconds)
- Accuracy matrix [6]
- Multiple error statistics:
  Kappa, Mean Absolute Error, Root Mean Squared Error, Relative Absolute Error

B. Data Set Information

Experiments as presented through the next section have been executed on the sample dataset (Eucalyptus Soil Conservation [7]) which has been drawn from the TunedIT repository of machine learning databases.

The objective of this dataset was to determine which seed lots in a species are best for soil conservation in seasonally dry hill country. Determination is found by measurement of

height, diameter by height, survival, and other contributing factors.

This dataset includes 736 Instances with 19 Attributes describing each of the data record.

C. Experimental Evaluation Through WEKA - Execution of Naïve Bayes with Complete Feature Set

In this section, we would execute the Naïve Bayes considering all the features in the selected Eucalyptus Soil Conservation data set.

The execution process is divided into the following steps:

- Data loading to read the input dataset
- Selecting an appropriate classification algorithm
- Training and testing of selected classifier (Naïve Bayes)

1) Data Loading to Read the Input Dataset: WEKA tool displays the following details after reading the data from the input file.

Number of Instances: 736

Number of Attributes: 20

List of all Attributes

Distinct values a class can have: None / Low / Average / Good / Best

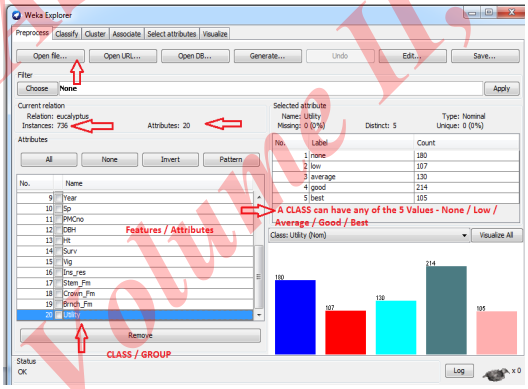

Fig. 3 Loading the data into WEKA for classification

2) Selecting an Appropriate Classification Algorithm: We have selected Naïve Bayes algorithm to be used for this experiment.

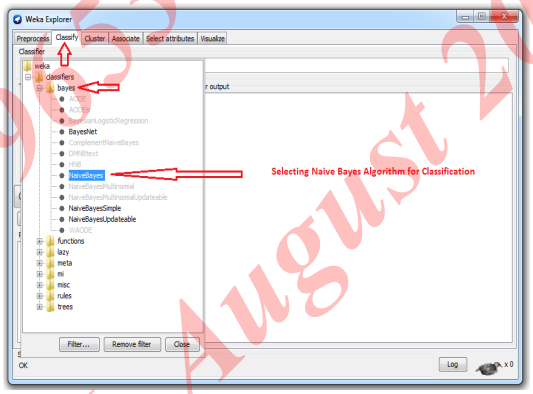

Fig. 4 Selecting an appropriate classification algorithm

3) Training and Testing of Selected Classifier (Naïve Bayes): After classification model is built, classifier will be tested to confirm the accuracy. Classifier will be tested according to the options that are set by clicking in the test options box.

We have selected the option - percentage split.

The classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing. The amount of data held out depends on the value entered in the % field.

Please note that we have specified the percentage split as 66%, means that 34% (250 Records) of the total 736 records would be held out to test the classification model built.

ISSN: 2321-9653

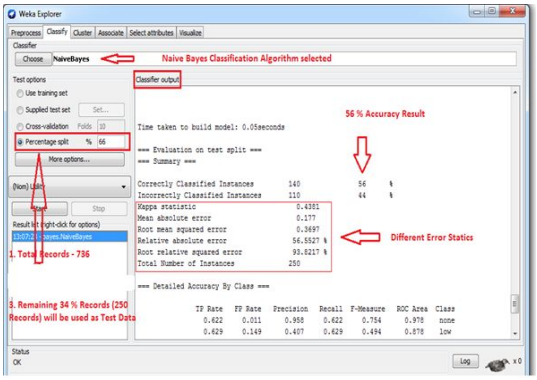

Fig. 5 Training and testing of selected classifier (Naïve Bayes)

Once the classifier, test options and class have all been set, the learning process is started by clicking on the start button.

When training is complete, the classifier output area to the right of the display is filled with text describing the results of training and testing. A textual representation of the classification model that was produced on the full training data is displayed in the classifier output area (Fig.6).

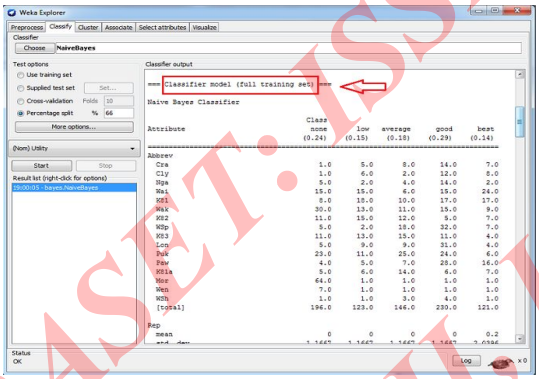

Fig. 6 Classifier output area displaying the classification model

generated on the full training data

The result of testing the Naïve Bayes classifier against the 250 records would be displayed in the classifier output area.

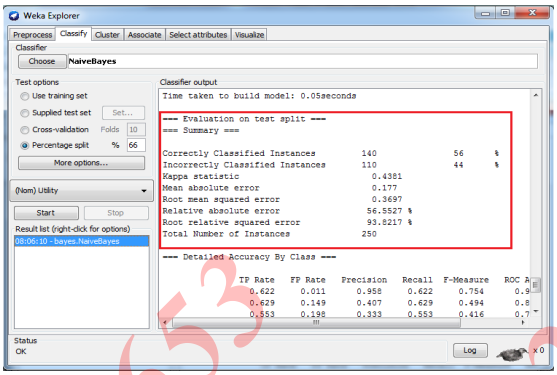

Fig. 7 Classifier output area displaying the summary of the result

Following are the list of statistics as displayed in the summary of the testing results:

Correctly Classified Instances	140	56 %
Incorrectly Classified Instances	110	44 %
Kappa statistic	0.4381	
Mean absolute error	0.177	
Root mean squared error	0.3697	
Relative absolute error	56.5527 %	
Root relative squared error	93.8217 %	
Total Number of Instances	250	

Please note that the accuracy of the Naïve Bayes classifier has been reported as 56% as 140 instances has been classified correctly against a total 250 instances in the test data.

D. Experimental Evaluation Through WEKA - Removing Correlated Redundant Features using CFS Algorithm

After executing the Naïve Bayes classification with all the features of the selected Eucalyptus Soil Conservation dataset in the previous section, in this section, we will now apply the correlation based feature selection (CFS) algorithm [8] to eliminate the correlated redundant features from the Eucalyptus Soil Conservation dataset.

We have selected CFS algorithm through WEKA Tool to eliminate the correlated redundant features. Result of applying

ISSN: 2321-9653

the CFS algorithm is displayed in attribute selection output area as presented in the following Fig.8.

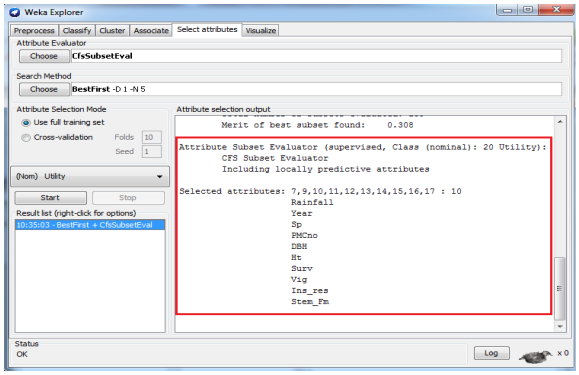

Fig.8 List of 10 selected attributes being displayed after applying the CFS algorithm

Attribute selection output area displays the list of 10 attributes selected (Out of total 19 attributes) after applying the CFS algorithm over Eucalyptus Soil Conservation dataset.

E. Experimental Evaluation Through WEKA - Execution of Naïve Bayes after Removing Correlated Redundant Features

In this section we would execute the Naïve Bayes classification again over the reduced Eucalyptus Soil Conservation dataset generated after eliminating the correlated redundant features using CFS algorithm in previous section.

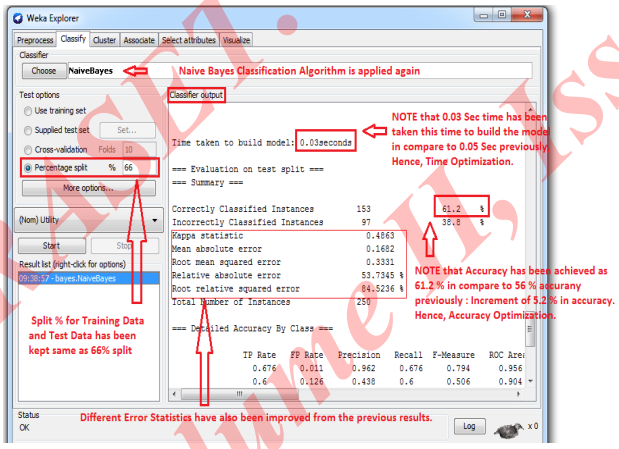

Fig. 9 Classifier output area displaying the summary of the testing result over the *reduced* data

Following are the list of statistics as displayed in the summary of the testing results over the reduced Eucalyptus Soil Conservation dataset:

Correctly Classified Instances	153	61.2 %
Incorrectly Classified Instances	97	38.8 %
Kappa statistic	0.4863	
Mean absolute error	0.1682	
Root mean squared error	0.3331	
Relative absolute error	53.7345 %	
Root relative squared error	84.5236 %	
Total Number of Instances	250	

F. Experimental Evaluation Through WEKA - Comparative Analysis of Naïve Bayes Performance Improvements

In this section we will go through the comparative analysis of Naïve Bayes classification algorithm's performance between the 2 datasets:
Eucalyptus Soil Conservation full / original dataset (With all 20 features)

Vs

Eucalyptus Soil Conservation reduced dataset (With 10 features only as selected by CFS algorithm. No correlated redundant features)

While making a comparative analysis for the 2 datasets, the following performance criteria would be selected:

- Classifier training time [Time taken to build the classification model]
- Classifier Prediction Accuracy Statistics
- Error Statistics

1) Comparative Analysis of Classifier Training Time [Time Taken to Build the Classification Model]:

TABLE 2

CLASSIFIER TRAINING TIME (IN SECONDS) FOR TWO DATASETS

	Naïve Bayes [With Correlated Features]	Naïve Bayes [No Correlated Features]
Classifier Training Time (in Seconds)	0.05	0.03

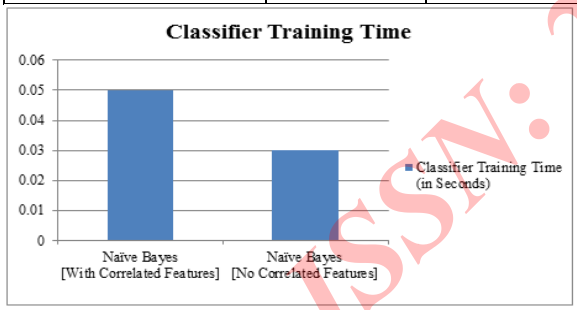

Fig. 10 Graph showing the comparative training time (In Seconds) taken

by Naïve Bayes classifier to build the model

Comparative analysis shows that time taken to build the Naïve Bayes classification model significantly reduces (From 0.05 to 0.03 Sec) when correlated features are removed from the dataset.

2) Comparative Analysis of Classifier Prediction Accuracy Statistics:

TABLE 3

CLASSIFIER PREDICTION ACCURACY (%) FOR TWO DATASETS

	Naïve Bayes [With Correlated Features]	Naïve Bayes [No Correlated Features]
Correctly Classified Instances	140	153
Incorrectly Classified Instances	110	97
Prediction Accuracy (%)	56	61.2

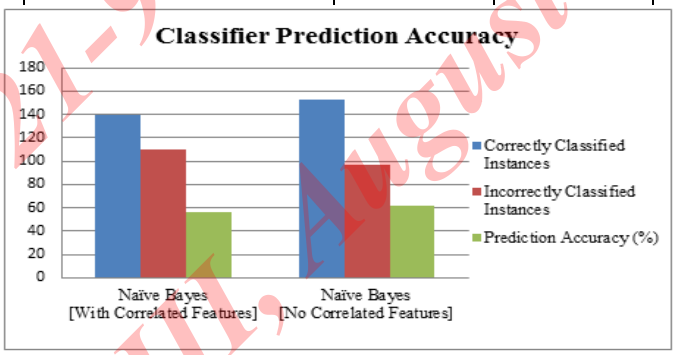

Fig. 11 Graph showing the comparative Prediction Accuracy statistics of Naïve Bayes classifier for two Datasets

Comparative analysis shows that predication accuracy of Naïve Bayes classification is increased (From 56% to 61.2%) when correlated features are removed from the dataset.

3) Comparative Analysis of Error Statistic:

ISSN: 2321-9653

TABLE 4

CLASSIFIER KAPPA STATS, MEAN ABSOLUTE AND ROOT MEAN SQUARED ERROR FOR TWO DATASET

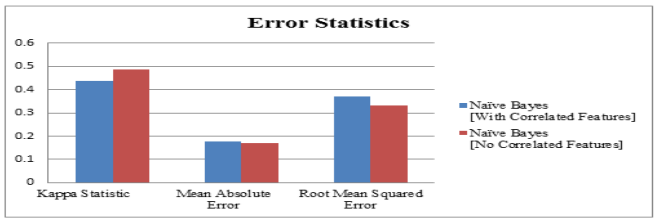

Fig.12 Graph showing the comparative Error statistics of Naïve Bayes classifier for two Datasets

TABLE 5

CLASSIFIER RELATIVE ABSOLUTE AND ROOT RELATIVE SQUARED ERROR (%) FOR TWO DATASETS

	Naïve Bayes [With Correlated Features]	Naïve Bayes [No Correlated Features]
Relative Absolute Error (%)	56.5527	53.7345
Root Relative squared Error (%)	93.8217	84.5236

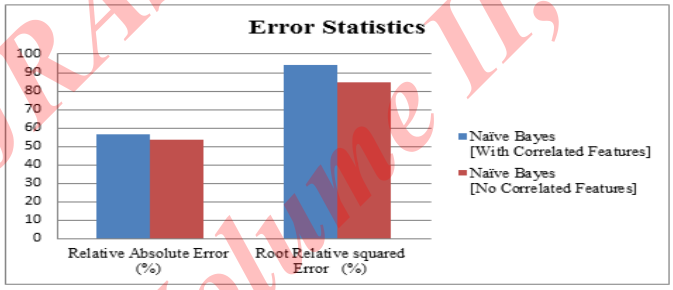

Fig. 13 Graph showing the comparative Error statistics (%) of Naïve Bayes classifier for two Datasets

Comparative analysis shows that different error statistics of Naïve Bayes classification are reduced when correlated features are removed from the dataset.

IV. CONCLUSIONS

The Naïve Bayesian classifier is a straight forward and frequently used method for supervised learning. It provides a flexible way for dealing with any number of attributes or classes, and is based on probability theory. It is the asymptotically fastest learning algorithm that examines all its training input.

	Naïve Bayes [With Correlated Features]	Naïve Bayes [No Correlated Features]
Kappa Statistic	0.4381	0.4863
Mean Absolute Error	0.177	0.1682
Root Mean Squared Error	0.3697	0.3331

It is known that Naïve Bayesian classifier (NB) works very well on some domains, and poorly on some. The performance of NB suffers in domains that involve correlated features. Naïve Bayes can suffer from oversensitivity to redundant and/or irrelevant attributes. If two or more attributes are highly correlated, they receive too much weight in the final decision as to which class an example belongs to. This leads to a decline in accuracy of prediction in domains with correlated features.

This paper illustrates that if those redundant and/or irrelevant attributes are eliminated, the performance of Naïve Bayesian classifier can significantly increase.

Based upon the comparative analysis of Naïve Bayes classification algorithm's performance on the basis of training time, prediction accuracy and multiple error statistics between the two datasets, we have observed a significant improvement in the Naive Bayes classification performance.

Testing results from WEKA tool have confirmed that the training time required by the Naive Bayes classifier to build

the classification model is also reduced after removing the correlated redundant features.

We can conclude that Naive Bayes can be applied in the domains (data set) that involve correlated redundant and irrelevant features with improved performance. This optimization is possible through the correlation based feature selection (CFS) algorithm which eliminates the correlated redundant and irrelevant features from the dataset before the dataset is passed to the Naive Bayes classifier for training purpose.

ACKNOWLEDGMENT

The authors would also like to express their sincere thanks to the TunedIT solutions for providing the Eucalyptus Soil Conservation dataset to execute the experiments. The TunedIT Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms.

Also, we would like to extend our thanks to the Machine Learning Group at the University of Waikato for WEKA tool, A data mining software written in Java, which is used extensively into Research and Academics and an open source software issued under the GNU General Public License. We have used WEKA tool to evaluate the performance of Naive Bayes algorithm.

REFERENCES

[1] Ioan Pop, "An approach of the Naive Bayes classifier for the document classification", General Mathematics Vol. 14, No. 4 (2006), 135–138

[2] I. Rish, An empirical study of the naive Bayes classifier, IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. (available online: PDF, PostScript).

[3] WEKA Data Mining software website. [Online].

Available: http://www.cs.waikato.ac.nz/ml/weka/

[4] Remco R. Bouckaert, Eibe Frank - University of Waikato, Hamilton, New Zealand, "WEKA Manual"

[5] JERZY STEFANOWSKI, "Data Mining - Evaluation of Classifiers", Institute of Computing Sciences, Poznan University of Technology, Poznan, Poland

[6] Roman Eisner, "Basic Evaluation Measures for Classifier Performance". [Online].

Available: http://webdocs.cs.ualberta.ca/~eisner/measures.html

[7] Machine Learning Repository of Dataset. [Online].

Available: http://tunedit.org/repo

[8] Mark A. Hall - Department of Computer Science University of Waikato, Hamilton, NewZealand, "Correlation-based Feature Selection for Machine Learning"

AUTHORS BIBLIOGRAPHY

Mr. Maneesh Singhal received his B.Tech (CS) from UP Technical University, Lucknow & M.Tech (CSE) from University College of Engineering, Rajasthan Technical University, Kota.

He has been working as a Lecturer in Department of Computer Science and Engineering, Arya College of Engineering and IT, Jaipur, Rajasthan.

His research interest includes Data Mining.

Mr Ramashankar Sharma, received his M.Tech from NIT kurukshetra, Haryana.

He has been working in Department of Computer Engineering, University College of Engineering, Rajasthan Technical University, Kota, Rajasthan, India since 1993.

At present he has been associated as Associate Professor and HOD of Department of Computer Engineering.

His research interest includes Distributed Systems.