ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (943-947), Month: April - June 2015, Available at: www.researchpublish.com

Data Pre-processing For Performance Analysis

Shobha Tyagi[1,] Komal Chaudhary[2]

[1]Assistant Professor (CSE), Faculty of Engineering & Technology Engineering, MRIU, Faridabad, Haryana, India
[2]Research Scholar, M. Tech (CSE), MRIU, Faridabad, Haryana, India

Abstract: **Pre-processing the data is a need in today's scenario. Data must be converted into a valid form so that it can be more useful and can assure great results. In this paper we are focusing on cleaning of data by filling in missing values and identifying the class basically known as classification. Results are compared in Weka for performance factor by taking data sets- first, with missing values and second with filling in those missing values with averaging and assigning class using Random Forest in terms of error rate. Two algorithms J48 and Random Forest are compared in weka in terms of performance (Accuracy and Error rate).**

***Keywords*: Data mining, preprocessing, random forest, decision tree, classification, data cleaning.**

I. INTRODUCTION

Classification [9] is a form of data analysis that extracts models describing important data classes. These type of models deals with class labels. For example, a model can be build for application of bank loan whether it should be approved or not depending on the person's present status and this application can be put in category of either 'safe' or 'risky'. This is how classes are defined. Classification [7] is done to tuples present in the dataset.

Data classification[12] is basically a two-step process, consisting of a learning step in which a classification model is constructed and second is a classification step where the model is used to predict class labels for given data.

Decision trees [2] are building with the help of two-step process of classification. For example, there is a tuple named **X** whose class is unknown. Now, the question is what is needed to do to predict its class. Here is the answer for using decision trees. The attribute value is tested against the decision tree. A path is traced from the root node to leaf node, which holds the class prediction for that tuple. Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes**.** Decision trees are basically used because they can handle multidimensional data.

Classification according to the *kinds of databases* mined: A data mining system can be classified according to the kinds of databases mined. Database systems can be classified according to different criteria (such as data models, or the types of data or applications involved), each of which may require its own data mining technique. Data mining systems can therefore be classified accordingly. For instance, if classifying according to data models, we may have a relational, transactional, object-relational, or data warehouse mining system. If classifying according to the special types of data handled, we may have a spatial, time-series, text, stream data, multimedia data mining system, or a World Wide Web mining system. Classification according to the *kinds of knowledge* mined: Data mining systems can be categorized according to the kinds of knowledge they mine, that is, based on data mining functionalities, such as characterization, discrimination, association and correlation analysis, classification, prediction, clustering, outlier analysis, and evolution analysis. A comprehensive data mining system usually provides multiple and/or integrated data mining functionalities.

II. DATA MINING

Data nowadays is expanding at exponential rate and it is very important to handle such data effectively so that important remains secure and unusable data may not be able to store much space. To solve this problem data mining [11] is used which extracts the important information from database. It is a stepwise process where preprocessing [5] of data is done, then patterns are generated. This is what data mining does.

Random Forest- Random forest (or random forests) is an ensemble classifier that consists of many decision trees [8] and outputs the class that is the mode of the class's output by individual trees. The method combines Breiman's "bagging" [1] idea and the random selection of features. Bagging [4] averages noisy and unbiased models to create a model with low variance. The random forest starts with a standard machine learning technique called a “decision tree” which, in ensemble terms, corresponds to our weak learner. In a decision tree, an input is entered at the top and as it traverses down the tree the data gets bucketed into smaller and smaller sets. For details see here, from which the figure below is taken.

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction [9].

J48- C4.5 is implemented as J48 in weka. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, \ldots$of already classified samples. Each sample $s_i$consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, \ldots, x_{p,i})$, where the $x_j$represent attributes or features of the sample, as well as the class in which $s_i$falls. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sublists.

III. RELATED WORK

A number of researchers have discussed the problem of finding relevant search results from the search engines. Relevant query recommendation research is mainly based on previous query log of the search engine, which contains the history of submitted query and the similarity of keywords [4]. and in this we used a information retrieval method by using term frequency, document frequency, inverse document frequency and the normalization factor for fast retrieval of result that means time reduction of user query retrieval reduces by using this method the formula which used in this paper helps in improving the efficiency, helps in mining of query log for giving a relevant result and reduces the query retrieval time. This formula includes many important factors on the basis of similarity score will be calculated which will be used in the query log. The factor included in this formula are similarity on the basis of keywords, weight factor which in turn includes the tf term frequency that is the number of time the term occurred in the document, df is the document frequency that is the number of documents in which that frequency is related, idf is the inverse of document frequency and the normalization factor is also included in this formula. By using this method or concept of information relevancy we are able to get the relevant information in minimum time. The resulting query log helps the user to find the relevant query easily and quickly. The history of queries stored in the query log helps the user. This method searches the related query based on the input query while the user searches so he can build a proper search query with the knowledge domain terminology which is important for search engine to get the related results[4].

IV. PROPOSED WORK

As described above, random forest is an ensemble of various decision trees. It uses “bagging” [1] which is basically used for classification. For example, you are a patient and would like to have a diagnosis made based on your symptoms. Instead of asking one doctor, you may choose to ask several. If a certain diagnosis occurs more than any other, you may choose this as a final result. This is chosen based on majority. Here each doctor gets an equal vote and now replace each doctor by a classifier and this is bagging.

The main disadvantage in using this Random Forest is:

i. The trees in Random Forest are grown to maximum size and are not pruned.

ii. For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

To overcome these problems Random Forest is classifying the data after filling missing values using **averaging** technique. This algorithm is very efficient. This algorithm makes the leveling equal and fills the missing values. It can handle both categorical and numeric data. Averaging is used to make the levels at one place. Classification is done to identify classes after filing in missing values [3].

In "Averaging", firstly the data taken is first preprocessed in order to search for any missing values. If any such missing values are found for any attribute, the data in a row is averaged using basic formula for averaging and the missing value is filled. If any such missing value persists again this is done. This is how averaging is done for making levels equal. The dataset after averaging will be used with Random Forest Algorithm.

This whole work is compared with the help of WEKA tool in which first data sets will be taken with random and missing values and one after applying the proposed work. Results will be compared and outcome will be shown in terms of less error rate by using the proposed technique.

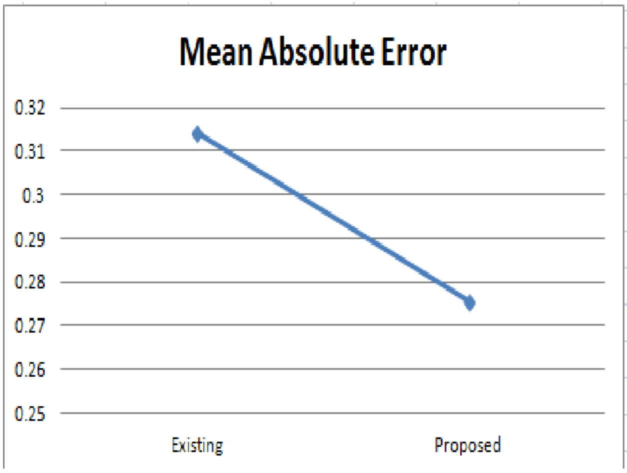
Two classifiers; J48 and Random Forest are taken to compare the respective result with the help of processed arff dataset in terms of performance (error rate and accuracy)

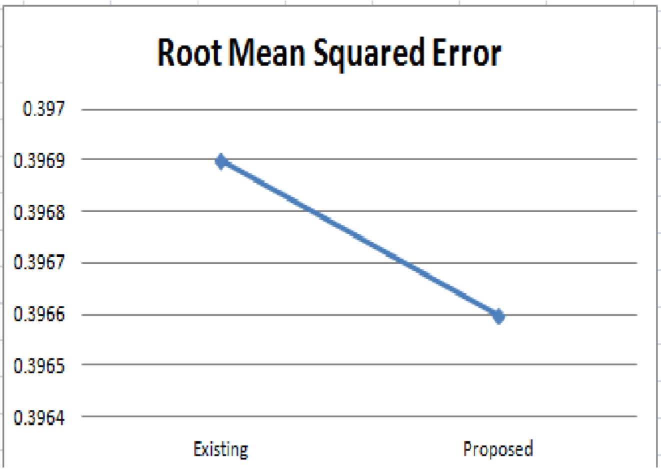
List of table:

Table 1:

Data Set	**Labor (missing)**	**Labor (modified)**
Mean Absolute Error	0.3143	0.2756
Root Mean Squared Error	0.3969	0.3966
Accuracy	70.1754%	77.193%

For dataset Labor (Note: Labor (Missing) shows dataset with missing value and Labor (Modified) shows data set after applying proposed work)





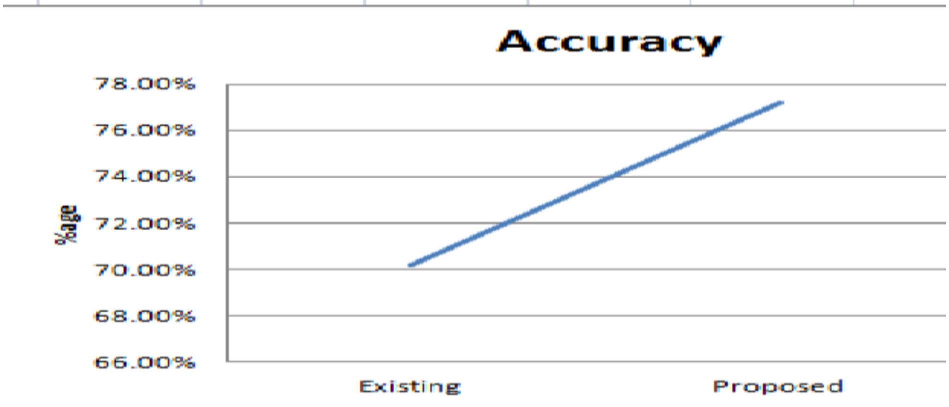
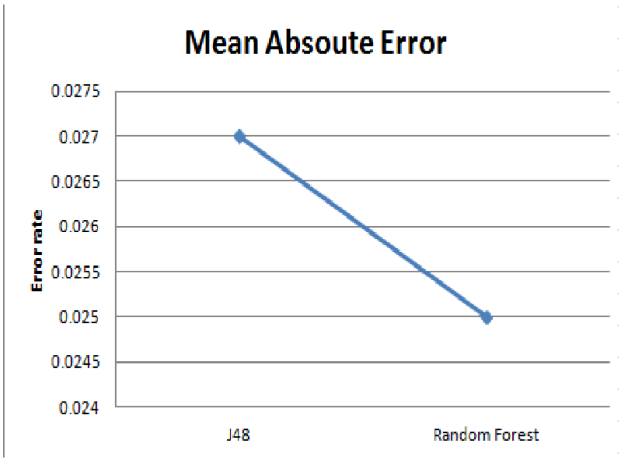
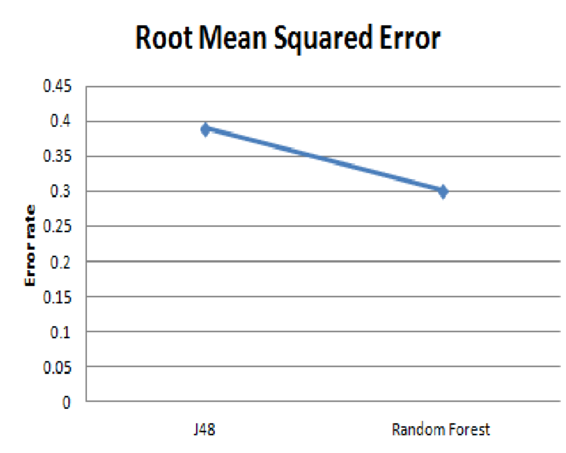


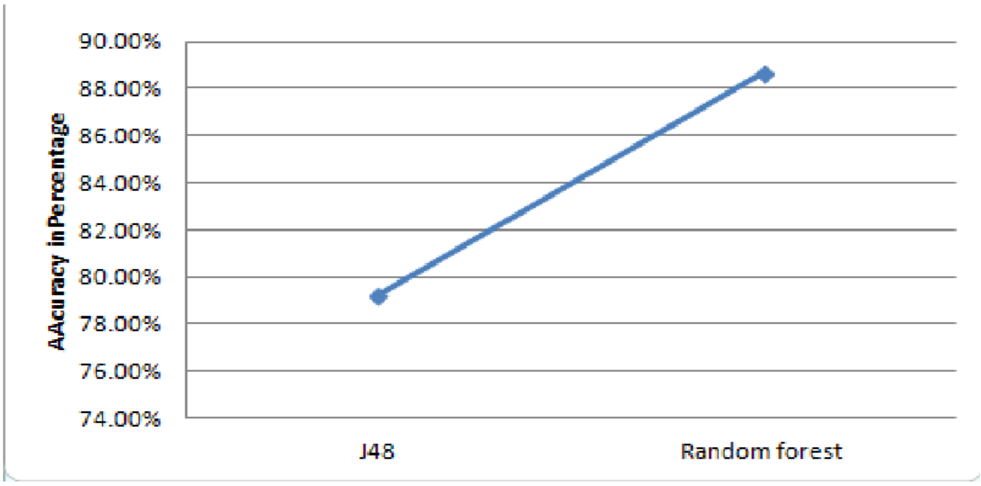
Table 2:

Criteria	**J48**	**Random Forest**
Mean Absolute Error	0.0271	0.0250
Relative Absolute Error	0.3396	0.302
Accuracy	78.65%	89.10%

For dataset Labor (Note: Both the algorithms are compared to know the best amongst these two in terms of error rate and accuracy)







V. CONCLUSION

As data preprocessing is a must task for any records to deal with. Nowadays data is available having certain discrepancies and fault; a technique must be included to handle this task effectively in all aspects. This dissertation shows how missing values are handled with the right procedure. Because for handling missing data there are many ways:

1. Using global constant:

This result in disadvantage of filling all the values with one unique number and it may result in filling some values greater than the value expected or less than the value expected.

2. Filling missing values randomly:

There is a disadvantage in this method because no values can be filled randomly. This results in various performance issues.

3. Performing Averaging:

This method proved out to be better than any method/technique that can be applied to handle missing value and result proved out to be true.

It can be concluded that random forest is better than J48 in all aspects whether; accuracy or error rate.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A.N. Swami, "Database Mining: A Performance Perspective," IEEE Trans. Knowledge and Data Eng., vol. 5, no. 6, pp. 914-925, Dec. 1993.

[2] J.R. Quinlan, "Induction of Decision Trees," IEEE Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.

[3] J.R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[4] C.L. Tsien, I.S. Kohane, and N. McIntosh, "Multiple Signal Integration by Decision Tree Induction to Detect Artifacts in the Neonatal Intensive Care Unit," IEEE Artificial Intelligence in Medicine, vol. 19, no. 3, pp. 189-202, 2000.

[5] G.L. Freed and J.K. Fraley, "25 Percent "Error Rate in Ear Temperature Sensing Device," IEEE Pediatrics, vol. 87, no. 3, pp. 414- 415, Mar. 1991.

[6] L.Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and Regression Trees. Wadsworth, 1984.

[7] R.Cheng, D.V. Kalashnikov, and S. Prabhakar, "Querying Imprecise Data in Moving Object Environments," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, pp. 1112-1127, Sept. 2004.

[8] R.R. Safavin and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Trans. Syst., Man, Cybern., vol. 21, pp. 660–674, May 1991.

[9] J.R. Quinlan, "Learning efficient classification procedures and their applications to chess end games," in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., IEEE Machine Learning: An Artificial Intelligence Approach, vol. 1. Palo Alto, CA: Tiogo, 1983.

[10] J.R. Quinlan, "Simplifying decision trees," IEEE Int. J. Man-Mach. Studies, vol. 27, no. 3, pp. 221–234, 1987.

[11] U.M. Fayyad and K. B. Irani, "On the handling of continuous valued attributes in decision tree generation," IEEE Mach. Learn., vol. 8, no. 1, pp. 87–102, 1992.

[12] L.Breiman, "Bagging Predictors", IEEE Machine Learning, vol. 24, pp. 123- 140,1996.

[13] L.Breiman, "Random Forests", IEEE Machine Learning, vol. 45, no. 1, pp. 5-32,2001.

[14] T.Dietterich, "An Experimental comparison of three methods for constructing ensembles of decision trees: Bagging, Boosting, and Randomization", vol. 45, no.2, pp. 139-157, 2000.