

Agricultural Data Mining –Exploratory and Predictive model for finding Agricultural Product Patterns

Gulledmath Sangayya¹, Yethiraj N G², Dadavalli S P³

¹Research Scholar
Anna University
Chennai, India

² Assistant Professor
Dept of Computer Science, Maharani's Science college for women
Bangalore, India

³ Assistant Professor
Dept of Computer Science, GFGC
Bangalore, India

¹gsswamy@gmail.com, ²rajsbmjce@gmail.com, ³dadavali@gmail.com

Abstract

In India the agriculture was practiced as subsistence basis; farmers products used to exchange with his needs with other commodities on barter system in olden days. Now in better usage of agricultural technology and timely inputs, the agriculture becomes commercial in nature. The current scenario is totally different farmers want remunerative prices on his produced commodity and increased awareness marketing becomes part of agricultural system. Now situation is much more demanding if people are not competent enough then survival becomes difficult. Still in India dream of technology reaching to poor farmers is distant issue. However government is taking initiative to empower them with new ICT tools [1]. The small effort of this paper is intended to give insight of one such technology called as Data Mining. In this paper we have taken certain conceptual Data Mining Techniques and Algorithms to implement and incorporate various methodologies to create concurrent result for decision making and creating favorable policy for farmers. We used data sets from APMC market source and run using the open source software tool of Weka [2]. Our finding clearly indicates which algorithm does what and how to use in effective and appropriate manner.

Keywords: Agriculture, Agriculture Marketing, Knowledge Management, Data Mining, and Data Mining Algorithms.

1. Introduction

Agricultural Data Mining which is an application part of Data Mining [3]. Recently we have coined this word thinking that a use of Data Mining in Agricultural arena can be referred as Agricultural Data Mining (ADM). The conceptual frame and working architecture of data mining remains same.

The search for required patterns in data is a human nature that is as old as it is ubiquitous, and has witnessed a dramatic and transitive transformation in strategy throughout the years when we compared various associated set of data. Whether we refer to hunters seeking to understand the animals' migration patterns, or farmers attempting to model harvest evolution in realistic, or turn to more current concerns practically, like sales trend analysis, assisted medical diagnosis, or building models of the surrounding world from scientific data, we reach the same conclusion: hidden within raw data we could

find important new pieces of information and knowledge. This piece of information makes more profitable when we convert into knowledge.

Traditional and conventional approaches for deriving knowledge from data depend strongly on manual analysis and interpretation of results. For any domain – scientific process, marketing, finance, health, business, etc. – the success of a traditional analysis purely depends on the capabilities of one or more specialists to read into the data: scientists go through remote images of planets and asteroids to mark interest objects, such as impact craters; bank analysts go through credit applications to determine which are prone to end in defaults. Such an approach is slow, expensive and with limited results, depending upon strongly on experience, state of mind and specialist know-how and when

Moreover, the quantum of generated data through various repositories is increasing dramatically, which in turn makes traditional approaches impractical in most domains. Within the large volumes of data derived or we can say extracted hidden strategic pieces of information for fields such as science, health or business. Besides the possibility to collect and store large volumes of data, the information era has also provided us with an increased computational and logical decision making power. The natural attitude is to employ this power to automate the process of discovering interesting models and patterns in the raw data. Thus, the purpose of the knowledge discovery methods is to provide solutions to one of the problems triggered by the information era: “data overload” [Fay96].

A formal definition of Data Mining (DM), also known – historically – as data fishing, data dredging (1960–Survey on Data Mining), knowledge discovery in databases (1990–Survey on Data Mining), or – depending on the domain, as business intelligence, information discovery, information harvesting or data pattern processing – is [4]:

Definition: Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.[5]

By data the definition refers to a set of real facts (e.g. records in a database), whereas pattern represents an expression which describes a subset of the data or modeled out comes, i.e. any structured representation or higher level description of a subset of the data. The term process designates a complex activity, comprised of several steps, while non-trivial implies that some search or inference or logical engine is necessary, the straightforward derivation of the patterns is not possible. The resulting models or patterns should be valid on new data, with a certain level of confidence. Also, we wish that the patterns be novel – at least for the system and, ideally, for the analyst – and potentially useful, i.e. bring some kind of benefit to the analyst or the goal oriented task. Ultimately, they need to be interpretable, even if this requires some kind of result transformation.

Generic Model for DM Process: In below Fig1 shows the various steps involved in Data Mining process which comprises of following steps of major like Pre-processing, Processing and Post –processing.[5]

Generic DM process presents it as the development of various computer programs as software tools which automatically examine raw data as inputted data, in the search for models or designed models. In practically, performing data mining implies undergoing an entire process, and requires various techniques from a series of

domains, such as: statistics, machine learning, artificial intelligence, visualization. Essentially, the DM process is iterative and semi-automated, and may require human intervention in several key points. These key points enhance the approach to simulate the various process of Data mining.

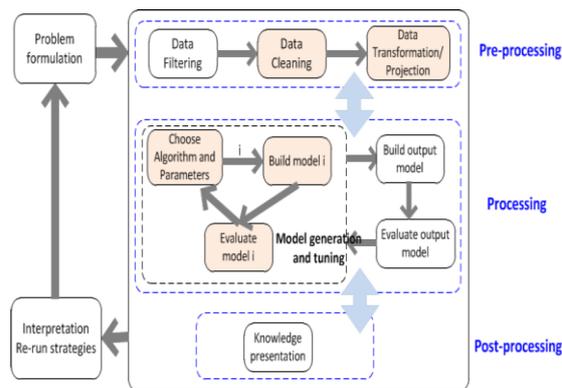


Fig1: Steps of Data Mining Process

Data filtering generally called as filter is responsible with the selection of relevant data for the intended analysis, according to the problem formulation. Data cleaning is responsible for handling missing values or discrete values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies, such as to compensate for the learning algorithms’ inability to deal with such data irregularities from the source.

Data transformation activities include various aggregation, normalization and solving syntactic incompatibilities, such as unit conversions or data format synchronization in case algorithms needs such conversion as intent

Data projection translates the input space into an alternative space, (generally) of lower dimensionality. The benefits of such an activity include processing speed-up, increased performance and/or reduced complexity for the resulting models. This also makes as catalytic model to increase the speed of ETL (Extract, Transform and Load) process.

During the processing steps, learning data models/ or patterns which we are looking are inferred, by applying the appropriate learning scheme on the pre-processed data. The processing activities are included in an iterative process, during which the most appropriate algorithm and associated parameter values are established (model generation and tuning). The appropriate selection of the learning algorithm, given the established goals and data characteristics, is essential which makes goal oriented task. In some situations in which it is required to adapt existing algorithms, or to develop new algorithms or methods in order to satisfy all requirements. Subse-

quently, the output model is built using the results from the model tuning loop, and its expected performance is assessed and analyzed for decision making purpose.

Knowledge presentation employs visualization methods to display the extracted knowledge in an intuitive, accessible and easy to understand manner. Decisions on how to proceed with future iterations are made based on the conclusions reached at this point.

DM process modeling represents an active challenge, through their diversity and uniqueness within a certain application. All process models contain activities which can be conceptually grouped, into the three types: pre-processing, processing and post-processing. Several standard process models which are discussed here, the most important being: William's model, Reinartz' model, CRISP-DM, I-MIN or Red path's model [Bha08]. Each model specifies the same process steps and data flow; they differ in the control flow. Essentially, they all are try to achieve the maximum automation and essential outcomes.

2. Methodology Adopted

There are various methods which can be adopted on the mode of either exploratory or predictive. However exploratory data models comes with various optional and readily designing patterns like Univariate and Bivariate models by conglomeration of data either categorical or numerical or grouped with categorical and numerical data components and results like Graphical charts, statistical outcome, histogram or correlation etc. Analyzed data after exploring looks like in Fig2.

In this paper we are tried to experiment various data mining prediction model to see how exactly data behaves to get some concurrent data. Generally speaking predictive modeling is the process in which model is created to predict an outcome. If the outcome of the model is categorical data then it as classification and outcome is numerical it is called as regression. In other case descriptive modeling or clustering is the observation for finding pattern of similar cluster. In last association provides the some interesting rules for mining termed as "Association Rule Miner". The limitation of our paper is we worked only on how various classification model works.

3. Classification

Classification is a data mining task of predicting the value of categorical variable in turns it should return to either target or class by constructing model based on one or more numerical or categorical variable. Here we assume categorical variables may be either predictors or attributes in general.[5]

No	Name of Commodity	Arrivals	Unit of Arrival	Variety	Minimum Price	Maximum Price	Model Price	Market
	Nominal	Numeric	Nominal	Nominal	Numeric	Numeric	Numeric	Nominal
1	Bajra	2.0	Tonnes	Hybrid	1040.0	1380.0	1260.0	Bangalore
10	Maize	46.0	Tonnes	Local	950.0	1020.0	1000.0	Arasikere
100	Rice(Sona Masori)	5.0	Tonnes	Coarse	2000.0	3000.0	2500.0	Ranebe...
101	Rice(Sona Masori)	2.0	Tonnes	Coarse	1400.0	1700.0	1500.0	Rachen...
102	Rice(Sona Masori)	6.0	Tonnes	Fine	1900.0	2900.0	2600.0	Surden...
103	Rice(Sona Masori)	4.0	Tonnes	Medium	1700.0	1800.0	1750.0	Sorabha...
104	Rice(Sona Masori)	103.0	Tonnes	Broke...	1300.0	1320.0	1320.0	Salale...
105	Rice(Sona Masori)	13.0	Tonnes	Pusa B...	2715.0	2715.0	2715.0	Shimoga
106	Rice(Sona Masori)	123.0	Tonnes	Pusa B...	2150.0	2300.0	2180.0	Srirang...
107	Rice(Sona Masori)	9.0	Tonnes	Pusa B...	1980.0	1980.0	1980.0	Sarjapura
108	Rice(Sona Masori)	6.0	Tonnes	Sarbit...	1670.0	1670.0	1670.0	Sindhapur
109	Rice(Sona Masori)	5.0	Tonnes	Kachh...	1800.0	1950.0	1900.0	Udupi
11	Maize	1126.0	Tonnes	Hybrid...	970.0	1042.0	1041.0	Badami
110	Same -Savi	1.0	Tonnes	Same/...	1900.0	2200.0	2000.0	Huliyar
111	Wheat	98.0	Tonnes	Local	1500.0	1900.0	1700.0	Bangalore
112	Wheat	3.0	Tonnes	Bans	1200.0	1800.0	1500.0	Bijapur
113	Wheat	10.0	Tonnes	Sona	1200.0	1500.0	1400.0	Bagemedi
114	Wheat	30.0	Tonnes	Jawari	1270.0	1270.0	1270.0	Hospet
115	Wheat	20.0	Tonnes	Super ...	1700.0	2050.0	1875.0	Shimoga
12	Maize	28.0	Tonnes	Hybrid...	1050.0	1100.0	1080.0	Bangalore
13	Maize	2.0	Tonnes	Local	930.0	1035.0	1001.0	Bellary
14	Maize	8.0	Tonnes	Hybrid...	952.0	952.0	952.0	Bijapur
15	Maize	8.0	Tonnes	Local	700.0	800.0	750.0	Channar...
16	Maize	23.0	Tonnes	Hybrid...	980.0	1000.0	1000.0	Channa...
17	Maize	11.0	Tonnes	Local	1040.0	1180.0	1160.0	Chikka...
18	Maize	6.0	Tonnes	Local	880.0	1000.0	920.0	Chenta...
19	Maize	385.0	Tonnes	Local	980.0	1025.0	1000.0	Davang...
2	Bajra	2.0	Tonnes	Hybrid	920.0	1282.0	1100.0	Bellary

Fig2. Basic Training Data Set Exploration

We can build classification model based on its core concept of structural methodology like,

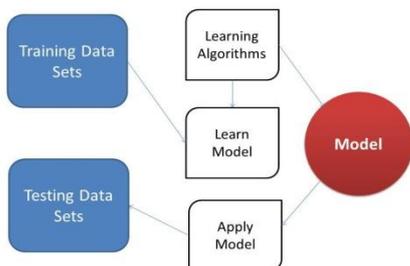
- 1) Frequency Table-Algorithms are
 - a. Zero R
 - b. One R
 - c. Naïve Bayesian
 - d. Decision Tree
- 2) Covariance Matrix
 - a. Linear Discriminate Analysis
 - b. Logistic Regression
- 3) Similarity Functions
 - a. K-Nearest Neighbors
- 4) Others in Tray are
 - a. Artificial Neural Network
 - b. Support Vector Machines

4. General Approach Building Classification Model

In this paper training set consists of records whose class labels are markets of agricultural data sets. Here we are considering part of data as input test sets so what the volatile behavior can be measured by observing various outcomes and how the learning model [Look at Fig 3. For general approach of classification] is behaving when we run the algorithm using machine learning tool like Weka. Conditional criteria of selecting data which we want to use as training and test sets needs to be parametric and obey the logical constraint of weight age. Whatever may be model outcomes in most of the cases the performance of classification is generally depend on the total counts of records which are correctly and incorrectly placed and predicted by the model. These counts later tabulated in a table known as confusion matrix. Here confusion matrix provides specific information needed to determine how well a classification model performs in any situation of data sets, summarizing this

information with single number or multiple results would make it more convenient to compare the performance of various other models for optimization. This can generally be done with performance with Accuracy and Error rate by the definition.

Fig. 3: General Approach frame work for Classification



Accuracy: It's the ratio of correct predictions to the total number of predictions.

Error rate: It's the ratio of Number of wrong predictions to the total number of predictions

Data Stage: In this experiment we used data sets from APMC market repositories and attributes are [Ref:Table1]

Table 1: Attribute of Data sets for our Experiments

Sl.No	Name of Attribute	Data type
1	Name of Market	Nominal
2	Name of Commodity	Nominal
3	Arrivals	Numerical
4	Unit of Arrivals	Nominal
5	Variety	Nominal
6	Minimum Price	Numerical
6	Maximum Price	Numerical
7	Modal Price	Numerical

Data Transformation: There are various support systems to convert either Microsoft Excel sheets into csv [Comma Separated Values] or load csv into Weka machine learning for experiment or convert csv into ARFF [Attributed Related File Format]. In case if familiar with java code running using java run environment or any IDE like Eclipse utility can be used. Then use following code snippet for conversion. Here we used common template for data conversion code. The algorithm looks like

```

//Common classes to be imported
import weka.core.Instances;
import weka.core.converters.ArffSaver;
import weka.core.converters.CSVLoader;
import java.io.File;
  
```

//Two public classes that dictates the logic of definitions

```

public class MyCSV2Arff{
    public static void main(String[] args) throws Exception
    {
        If(args.length!=2)
        {
            Sys-
            tem.out.println("\nUsage:MyCSV2Arff<input.csv><output.arff.\n");
            // In above usage file is mentioned as the source to convert
            System.exit(1);
        }
        // sources of loader to direct compiler
        CSVLoader loader=new CSVLoader();
        Loader.setSource(new File(args[0]));
        Instance data=loader.getDataSet();

        ArffSaver saver=new ArffSaver();
        // All Arff are saved in new file format that takes as arguments.
        Saver.setInstances(data);
        Saver.setFile(new File(args[1]));
        Saver.setDesination(new File(args[1]));
        Saver.writeBatch();
    }
}
  
```

Source: Weka open source tool for Data Mining

5. Data Loading into Weka for Experiment

Once the data loaded into Weka we started exploring various possibilities to explore the mechanisms of identifying the right algorithm to run our data. Initially we have chosen the Naïve Bayes followed by Bayes Net and compared with Rules one R[1R] and Trees.J48.

5.1 Working Basics of Various Algorithms

A Naïve Bayes classifier

This classifier adopt in many applications that generally creates the relationship among two important aspects of attribute set and class variable is totally non deterministic. In other words if we define the class label of each test record cannot be predicted with the amount of certainty even though its attribute set is similar to some of the given training examples. This sometimes arises due to the various noisy components exists. In some situations certain confounding factors that effect over all classification. For example if price varied due to volatile situation of market that effects the possibility of more input into the risk segmentation while fixing the price. Determine whether market is only condition or other hidden reasons to be claimed is more challenging. Here general interpretation is introducing uncertainties into the learning problem.[6]

B. Naive Bayes Belief Network

In General we can say probability estimates are often more useful than any plain predictions. They allow logically in predictive ranks and their expected cost of execution to be minimized. In some situations research community arguments for treating classification learning as the task of learning class probability problems estimated from given data. What is being estimated is the conditional probability distribution of the values for the given class of attributes and their values. There are many variants like Bayes classifiers, logistic regression models, decision trees and so on are the just good click to represent a conditional probability distribution of course each of technique differs in their representational powers. However Naïve Bayes classifiers and logistic regression in many situations represents only simple representations, whereas Decision Tree can represent at least approximate or sometimes arbitrary distributions. In practice these techniques have some drawbacks which in returns results as less reliable probability estimates.[8]

C. Trees.J48

Basically J48 algorithm is the Weka implementation of the C4.5 top-down decision tree learner proposed by Quinlan. This algorithm uses the greedy technique and its categorical variant of ID3[7], this algorithm determines at each step the most predictive attribute of data sets, and splits a node based on this attribute. Each node commonly represents a decision point over the value of some attribute. J48 also addresses to account for noise and missing values in a given datasets. It also deals with values which are numeric attributes by determining where exactly thresholds for decision splits should be placed. The main parameters that set for this algorithm are the confidence level threshold, the minimum number of instances per leaf and the number of folds for reduced error pruning.[9]

The algorithm used by Weka Team and the MONK project is known as J48. J48 is a version of an earlier algorithm developed by J. Ross Quinlan, this is very popular C4.5. Decision trees are a classical way to represent information from a machine learning algorithm, and offer a fast and powerful way to express structures that are needed in data.[10]

D. Rules one R

OneR, this algorithm shortly titled as for "One Rule", or "1 R" is a simple in action, yet accurate, classification algorithm that generates one rule for each one predictor in the data sets, then selects the rule with the smallest total error as its "One Rule". To create a rule for a predictor, we generally construct a frequency table for

each predictor value against the target function that evaluates the performance of algorithm. It has been shown that 1R produces rules only slightly less accurate than the other classification algorithms while producing rules that are simple for humans to interpret and analyze the results.[11]

6. Results and Discussions

In this section I am trying to explain what exactly we have done by incorporating various Data Mining classification algorithms using the above said agricultural data sets. We run the experiment in Weka open source learning environment using explorer menu. The test method we used are three mode of variants 1) Use Training Sets 2) Cross validation of 10 fold 3) Percentage split at 66

Table 2: Comparative Runs using Training Sets

1) Used Training Sets	Naive Bayes	BayesNet	OneR	Trees.J48
Time taken to build model	0.01 Se-conds	0.01 Seconds	0.02 Seconds	0.05 Seconds
Correctly classified instances	64 55.652 2 %	39 33.913 %	36 31.3043 %	61 53.0435 %
Incorrectly classified instances	51 44.347 8 %	76 66.087 %	79 68.6957 %	54 46.9565 %
Kappa Statistics	0.5482	0.3138	0.2888	0.516
Mean Absolute Error	0.0154	0.03	0.0222	0.0152
Root Mean Squared Error	0.0985	0.12	0.1489	0.0873
Relative Absolute Error	48.687 2 %	94.8255 %	70.1289 %	48.2321 %
Root Relative Absolute Error	78.415 %	95.5183 %	118.521 8 %	69.5029 %

Explanation of above table: In above commendable fact which is emerged from the experiment is that Naïve Bayes and Tree classification has more accurate instance classification the others.

Table 3:Comparative Runs using at 10 fold Prediction

1) Cross Validation	NaiveBayes	BayesNet	OneR	Trees.J48
Time taken to build model	0.002 Seconds	0.001 Se-conds	0.002 Se-conds	0.001 Se-conds
Correctly classified instances	3 2.6087 %	0 %	0 %	0 %
Incorrectly classified instances	112 97.3913 %	115 100 %	115 100 %	115 100 %
Kappa Statistics	0.0029	-0.0433	-0.0389	-0.034
Mean Absolute Error	0.0313	0.032	0.0323	0.0322
Root Mean Squared Error	0.1531	0.1282	0.1796	0.153
Relative Absolute Error	98.533 %	100.831 4 %	101.60 67 %	101.378 5 %
Root Absolute Error	121.3516 %	101.563 8 %	142.31 95 %	121.264 4 %

Explanation of above table: In above commendable fact which is emerged from the experiment is that Naïve Bayes is only algorithm which works for better usage of algorithm then any others.

Table 4: Comparative Runs at split level of 66%

1) Split Test of 66 %	NaiveBayes	BayesNet	OneR	Trees.J48
Time taken to build model	0.002 Seconds	0.001 Seconds	0.002 Se-conds	0.001 Seconds
Correctly classified instances	1	0 %	0 %	2.5641 %
Incorrectly classified instances	38 97.4359 %	39 100 %	39 100 %	39 100 %
Kappa Statistics	0.0146	-0.0194	-0.0291	-0.0188

Mean Absolute Error	0.0312	0.032	0.0323	0.0321
Root Mean Squared Error	0.1539	0.1286	0.1796	0.1532
Relative Absolute Error	98.2779 %	100.747 %	101.547 2 %	101.197 %
Root Relative Absolute Error	121.8041 %	101.770 4 %	142.186 1 %	121.305 %

Discussion: In above we used certain calculation to test the parametric justifications of used algorithms. Those are as listed follows

Kappa Statistics

Cohen’s kappa coefficient statistical measures among inter rater agreement which deals for qualitative items. It’s observed as more robust then measuring simple percent agreement calculation. When two binary variables which are attempts by two individuals to measure the same thing, we can use Cohen's Kappa (often simply called Kappa) as a measure of agreement between the two individuals items. Kappa measures the percentage of data items in the main diagonal of the table and then adjusts these values for the amount of agreement that could be expected due to chance alone.

Here is one possible interpretation of Kappa.

- Poor agreement = Less than 0.20
- Fair agreement = 0.20 to 0.40
- Moderate agreement = 0.40 to 0.60
- Good agreement = 0.60 to 0.80
- Very good agreement = 0.80 to 1.00

More details can be on he which we listed and referred:[15]

Mean absolute error (MAE)

In statistics, the mean absolute error (MAE) is a quantity used to measure how close for each forecasts or predictions are to the eventual outcomes as result. The mean absolute error is given by following equations

As the name suggests, the mean absolute error is an average

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|.$$

erage of the absolute errors $e_i = |f_i - y_i|$, where f_i is the prediction and y_i the true value. Note that alternative formulations may include relative frequencies as weight factors for calculating MAE.

Root mean squared error (RMSE)

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation, and are called prediction errors when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSD is a good measure of accuracy, but only to compare forecasting errors of different models for a particular variable and not between variables, as it is scale-dependent.[13]

The MAE and the RMSE can be used together to diagnose the variation in the errors in a set of forecasts here we referred to prediction. The RMSE will always be larger or equal to the MAE; the greater difference between them, the greater the variance in the individual errors in the sample. If the RMSE=MAE, then all the errors are of the same magnitude

Both the MAE and RMSE can range from 0 to ∞ . They are negatively-oriented scores: Lower values are better.

The root relative squared error is relative to what it would have been if a simple predictor had been used. More specifically, this simple predictor is just the average of the actual values in data sets. Thus, the relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the simple predictor.

By taking the square root of the relative squared error one reduces the error to the same dimensions as the quantity being predicted.

Mathematically, the root relative squared error E_i of an individual program i is evaluated by the equation:

$$E_i = \sqrt{\frac{\sum_{j=1}^n (P_{(ij)} - T_j)^2}{\sum_{j=1}^n (T_j - \bar{T})^2}}$$

Where $P_{(ij)}$ is the value predicted by the individual program i for sample case j (out of n sample cases); T_j is the target value for sample case j ; and \bar{T} is given by the formula:

$$\bar{T} = \frac{1}{n} \sum_{j=1}^n T_j$$

For a perfect fit, the numerator is equal to 0 and $E_i = 0$.

So, the E_i index ranges from 0 to infinity, with 0 corresponding to the ideal.[14]

Concluding remark on facts of algorithms we can justify that algorithms works when used training sets with Naïve Bayes and Tree J48 algorithms shows good response.

Acknowledgments

The great work can't be achieved unless team of members with coherent ideas should be matched. I take an opportunity to my coauthors for their painstaking work and Shri.Devaraj Librarian UAS, GKVK, Bangalore-65 for assisting me to get clear many issues for preparing papers.

References

- [1] Jac Stienen with Wietse Bruinsma and Frans Neuman, How ICT can make a difference in Agricultural livelihoods, The Commonwealth Ministers Reference Book – 2007
- [2] WEKA: Data Mining Software in JAVA: <http://www.cs.waikato.ac.nz/ml/weka>.
- [3] Sally Jo Cunningham and Geoffrey Holmes, Developing innovative applications in agriculture using data mining, Department of Computer Science, University of Waikato Hamilton, New Zealand
- [4] Ian H. Witten and Eibe Frank. Data Mining Practical Machine Learning Tools and Techniques, Second Edition, Elsevier.
- [5] Jiawei Han, Micheline Kamber, Data Mining Concepts and Techniques, Elsevier.
- [6] Remco R. Bouckaert, Bayesian Network Classifiers in Weka, remco@cs.waikato.ac.nz.
- [7] Baik, S. Bala, J. (2004), A Decision Tree Algorithm for Distributed Data Mining: Towards Network Intrusion Detection, Lecture Notes in Computer Science, Volume 3046, Pages 206 – 212.
- [8] Bouckaert, R. (2004), Naive Bayes Classifiers That Perform Well with Continuous Variables, Lecture Notes in Computer Science, Volume 3339, Pages 1089 – 1094.
- [9] Breslow, L. A. & Aha, D. W. (1997). Simplifying decision trees: A survey. Knowledge Engineering Review 12: 1–40.
- [10] Brighton, H. & Mellish, C. (2002), Advances in Instance Selection for Instance-Based Learning Algorithms. Data Mining and Knowledge Discovery 6: 153–172.

- [11] Cheng, J. & Greiner, R. (2001). Learning Bayesian Belief Network Classifiers: Algorithms and System, In Stroulia, E.& Matwin, S. (ed.), AI 2001, 141-151, LNAI 2056,
- [12] Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach. Artificial Intelligence 137: 43-90.
- [13] Clark, P., Niblett, T. (1989), The CN2 Induction Algorithm. Machine Learning, 3(4):261-283.
- [14] Cover, T., Hart, P. (1967), Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1): 21-7.
- [15] Internet source on Kappa statistics; <http://www.pmean.com/definitions/kappa.htm>