Agricultural Data Mining – Exploratory and Predictive Model for Finding Agricultural Product Patterns

Gulledmath Sangayya¹, Yethiraj N.G²

¹Research Scholar, Anna University Chennai-600 025
²Assistant Professor with Department of Computer Science, FGGC, Yelahanka Bangalore-64
E-mail: gsswamy@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. 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. Our finding clearly indicates which algorithm does what and how to use in effective and appropriate manner.

Index Terms: Agriculture, Agriculture Marketing, Knowledge Management, Data Mining, and Data Mining Algorithms

I. 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]

![Fig1: Steps of Data Mining Process](image)

[Picture cite source: PhD thesis byEng. Camelia Lemnaru (Vidrighin Bratu) Titled: STRATEGIES FOR DEALING WITH REAL WORLD CLASSIFICATION PROBLEMS-Scientific Advisor: Prof.Dr.Eng.Sergiu NEDEVSCHI]

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. 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. Subsequently, 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.

**II. 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 Fig3.

![Fig3. Basic Training Data Set Exploration](image)

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.

### III. 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] We can build classification model based on its core concept of structural methodology like, Frequency Table-Algorithms are

- Zero R
- One R
- Naïve Bayesian
- Decision Tree

2) Covariance Matrix
- Linear Discriminate Analysis
- Logistic Regression

3) Similarity Functions
- K-Nearest Neighbors

**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

- Artificial Neural Network
- Others in Tray are Support Vector Machines

**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

```java
public class MyCSV2Arff{
    public static void main(String[] args) throws Exception
    {
        if(args.length!=2){System.out.println("Usage: MyCSV2Arff <input.csv> <output.arff.in>");}
        // 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();
    }
}
```

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

<table>
<thead>
<tr>
<th>S/No</th>
<th>Name of Attribute</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name of Market</td>
<td>Nominal</td>
</tr>
<tr>
<td>2</td>
<td>Name of Commodity</td>
<td>Nominal</td>
</tr>
<tr>
<td>3</td>
<td>Arrivals</td>
<td>Numerical</td>
</tr>
<tr>
<td>4</td>
<td>Unit of Arrivals</td>
<td>Nominal</td>
</tr>
<tr>
<td>5</td>
<td>Variety</td>
<td>Nominal</td>
</tr>
<tr>
<td>6</td>
<td>Minimum Price</td>
<td>Numerical</td>
</tr>
<tr>
<td>7</td>
<td>Maximum Price</td>
<td>Numerical</td>
</tr>
</tbody>
</table>

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 classifier followed by Bayes Net and compared with Rules one R[1R] and Trees.J48.

IV. WORKING BASICS OF VARIOUS ALGORITHMS

A. Naïve Bayes classifier

which we 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 short 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]

V. 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

<table>
<thead>
<tr>
<th>1) Used Training Sets</th>
<th>Naïve Bayes</th>
<th>BayesNet</th>
<th>OneR</th>
<th>Trees.J48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken to build model</td>
<td>0.01 Seconds</td>
<td>0.01 Seconds</td>
<td>0.02 Seconds</td>
<td>0.05 Seconds</td>
</tr>
<tr>
<td>Correctly classified instances</td>
<td>64</td>
<td>55.6522%</td>
<td>39</td>
<td>33.913%</td>
</tr>
<tr>
<td>Incorrectly classified instances</td>
<td>51</td>
<td>44.3478%</td>
<td>76</td>
<td>66.087%</td>
</tr>
<tr>
<td>Kappa Statistics</td>
<td>0.5482</td>
<td>0.3138</td>
<td>0.2888</td>
<td>0.516</td>
</tr>
<tr>
<td>Mean Absolute Error</td>
<td>0.0154</td>
<td>0.03</td>
<td>0.0222</td>
<td>0.0152</td>
</tr>
<tr>
<td>Root Mean Squared Error</td>
<td>0.0985</td>
<td>0.12</td>
<td>0.1489</td>
<td>0.0873</td>
</tr>
<tr>
<td>Relative Absolute Error</td>
<td>48.6872%</td>
<td>94.8255%</td>
<td>70.1289%</td>
<td>48.2321%</td>
</tr>
<tr>
<td>Root Relative Absolute Error</td>
<td>78.415%</td>
<td>95.5183%</td>
<td>118.5218%</td>
<td>69.5029%</td>
</tr>
</tbody>
</table>

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.
The RMSE can be used together with the MAE to diagnose the variation in the errors in a set of forecasts here dependent.

RMSD serves to aggregate the magnitudes of the errors in the data sample that was used for estimation, and are related to the data.

Mean absolute error (MAE): In statistics, the mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes as reult. The mean absolute error is given by the following equation:

\[MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|\]

where \(\hat{y}_i\) 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 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 RMSD can be used together to diagnose the variation in the errors in a set of forecasts here we refered 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 RMSD can range from 0 to \(\infty\). 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

<table>
<thead>
<tr>
<th>1 ) Cross Validation</th>
<th>NaiveBayes</th>
<th>BayesNet</th>
<th>OneR</th>
<th>Trees.J48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken to build model</td>
<td>0.002 Seconds</td>
<td>0.001 Seconds</td>
<td>0.002 Seconds</td>
<td>0.001 Seconds</td>
</tr>
<tr>
<td>Correctly classified instances</td>
<td>3</td>
<td>2.6087 %</td>
<td>0 %</td>
<td>0 %</td>
</tr>
<tr>
<td>Incorrectly classified instances</td>
<td>112 %</td>
<td>97.3913 %</td>
<td>115 %</td>
<td>100 %</td>
</tr>
<tr>
<td>Kappa Statistics</td>
<td>0.0029</td>
<td>-0.0433</td>
<td>-0.0389</td>
<td>-0.034</td>
</tr>
<tr>
<td>Mean Absolute Error</td>
<td>0.0313</td>
<td>0.032</td>
<td>0.0323</td>
<td>0.0322</td>
</tr>
<tr>
<td>Root Mean Squared Error</td>
<td>0.1531</td>
<td>0.1282</td>
<td>0.1796</td>
<td>0.153</td>
</tr>
<tr>
<td>Relative Absolute Error</td>
<td>98.533 %</td>
<td>100.8314 %</td>
<td>101.6067 %</td>
<td>101.3785 %</td>
</tr>
<tr>
<td>Root Absolute Error</td>
<td>121.3516 %</td>
<td>101.5638 %</td>
<td>142.3195 %</td>
<td>121.2644 %</td>
</tr>
</tbody>
</table>

Table 3: Comparative Runs using at 10 fold Prediction

<table>
<thead>
<tr>
<th>1 ) Split Test of 66 %</th>
<th>NaiveBayes</th>
<th>BayesNet</th>
<th>OneR</th>
<th>Trees.J48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time taken to build model</td>
<td>0.002 Seconds</td>
<td>0.001 Seconds</td>
<td>0.002 Seconds</td>
<td>0.001 Seconds</td>
</tr>
<tr>
<td>Correctly classified instances</td>
<td>1</td>
<td>0 %</td>
<td>0 %</td>
<td>0 %</td>
</tr>
<tr>
<td>Incorrectly classified instances</td>
<td>38</td>
<td>97.4359 %</td>
<td>39</td>
<td>100 %</td>
</tr>
<tr>
<td>Kappa Statistics</td>
<td>0.0146</td>
<td>-0.0194</td>
<td>-0.0291</td>
<td>-0.0188</td>
</tr>
<tr>
<td>Mean Absolute Error</td>
<td>0.0312</td>
<td>0.032</td>
<td>0.0323</td>
<td>0.0321</td>
</tr>
<tr>
<td>Root Mean Squared Error</td>
<td>0.1539</td>
<td>0.1286</td>
<td>0.1796</td>
<td>0.1532</td>
</tr>
<tr>
<td>Relative Absolute Error</td>
<td>98.2779 %</td>
<td>100.747 %</td>
<td>101.5472 %</td>
<td>101.197 %</td>
</tr>
<tr>
<td>Root Relative Absolute Error</td>
<td>121.8041 %</td>
<td>101.7704 %</td>
<td>142.1861 %</td>
<td>121.305 %</td>
</tr>
</tbody>
</table>

Table 4: Comparative Runs at split level of 66%

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

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.

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 reult. The mean absolute error is given by following equation: As the name suggests, the mean absolute error is an average of the absolute error \(E_i = |\hat{y}_i - y_i|\), where \(\hat{y}_i\) is the prediction and \(y_i\) the true value. Note that alternative formulations may follow:

- Various types of measurements 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 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 RMSD 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 RMSD can range from 0 to \(\infty\). 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:

$$E_i = \sqrt{\frac{\sum (P_{ij} - T_j)^2}{\sum (T_j - \bar{T})^2}}$$

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.

Concluding remark on facts of algorithms we can justify that algorithms work best when used training sets with Naïve Bayes and Tree J48 algorithms show 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

[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
[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.