Article can be accessed online at http://www.publishingindia.com

Experimental Study of Data Mining Classification Algorithms in Establishing Indian Agricultural Commodity Patterns

Gulledmath Sangayya*

Abstract

This paper presents novel idea of how to establish various products pattern of Indian agricultural commodity using Data Mining Classification Algorithms. Generally when we talk about Data Mining we come across several basics and advance technique's to incorporate for the broader applications of usage. Now we want to use Data Mining algorithms to extract some very interesting patterns by detailed study of agricultural data sets. As we all know computing and information has vast scope to deal for commercial usage but picture changes when it comes to medium profitable segments. In this paper I have tried experimental basis of Data Sets using agricultural products category by extracting from local APMC (Agricultural Product Market Cooperation). This organization helps locally to guide farmers to know the best price for selling and buying. If we incorporate new trend setting solution that makes more transparent and predictable solution patterns for farmers of local community and compare the price segments with national monitoring markets like Agmark(Agricultural market of India). The research establishes various classifications based on given class of market by using Naïve Bayes and Bayes Net algorithms and comparing with Rules one R [1R] and Trees.J48.

Keywords: Data Mining, Classification, Algorithms, Knowledge and Data Engineering Tools and Techniques, Agricultural Products, Agricultural Markets.

1. Introduction

Data mining is the process or methodology of processing large quantum of data (usually stored in a database in the form of repositories or tools like Data warehouses), searching for interesting patterns and relationships within that data. Retail outlets those who are using data mining as a tool might discover that many customers who buy beer also buy diapers *[Mining Generalized Association Rules –Ramkrishna Srikant and Rakesh Agrawal 21 VLDB conferences Zurich Switzerland, 1995]*. They may then increase sales by positioning the two together. Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a in this classification model price could be used to identify what constraint farmer can take up the risk to sale or hold for higher price or negotiate the price with comparing the national markets.

Classification if we look into conventional text books it clearly defines a task of assigning objects to one of several predefined categories it may be pervasive in its existence problem that encompasses many diverse applications. Here we are going to analyze the problem of agricultural data sets and its behavior perception of various concept based algorithms. Generally, classification works on the mode of attribute set of x, which can be mapped into class label y using classification model.

* Assistant Professor and HOD, Department of Computer Science, Government First Grade College, Yelahanka, Bangalore, Karnataka, India. E-mail: gsswamy@gmail.com

The input data for classification task is collection of records as in this paper we are referring to agricultural data sets. Each record also known as instance or example, is characterized by tuple (x, y).Where x is the attribute set and y is special attribute, designated as class label also known as category or target attribute. In this paper Table 1 represents attribute set includes property of data sets. One more thing that we should make a note here is attributes presented here are mostly discrete; attribute set can also contain continuous features. The class label which we generally referred as output must be discrete attribute. This attentive distinct character makes classification from regression. We go by the definition of classification is the task of learning target function f that maps each attribute set x to one of the predefined class label y. One more interesting fact is target function is also known as classification model. A classification model is useful in various patterns to extract either in descriptive modeling or predictive modeling.

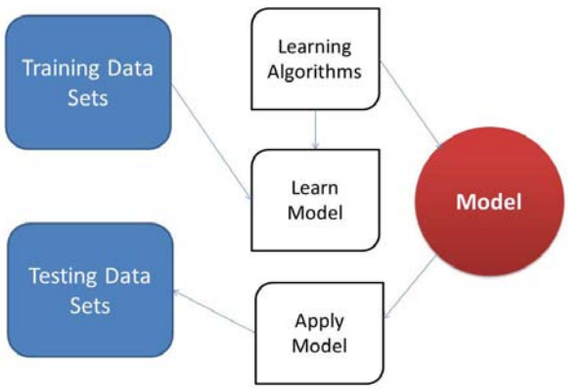
A classification model can serve as an explanatory tool to distinguish between objects of different classes known to be descriptive modeling. A classification model can also use to serve as predictive of unknown records of class label y. Classification techniques are most suited for predicting or describing data sets with either binary or nominal categories. They are less effective for ordinal categories like high, medium or low. In the paper *"Classification methods" by Aijun An of York University, Canada* makes point classification is the action of assigning an object to the specified category according to the characteristics of the object. Classification has wider scope in real time applications like in medical science predicting patient's behavior, diseased based symptoms, analyzing the credit card fraudulent transactions etc. In this paper we are experimenting the various patterns of agricultural data sets and its outcome incase if market is volatile.

2. Procedure for Solving Classification Model

2.1. General Approach

As we are aware that classification or generally we call as classifier is discipline that shows the systematic approach to build model of user choice from input data sets. There are various techniques which include decision tree classifiers, or we can consider rule based classifiers, or neural networks or it may even support vector machines along with techniques like naive based classifier makes more prominent in handling the approach of classifiers or we can say deployment of classifiers in real time implementation. *In conventional text of Introduction to Data Mining by Tan, Steinbach and Kumar* indicated each technique employs learning algorithm to identify a model that fits according to the need of situation that fits the relationship between the attribute set and class label of input data.

Figure 1: General Approach for Building a 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 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 weightage. However in this situation run the instance for the exploratory data outcomes rather being instrumental in predictive.

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.

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

2.2. Data Stage

In this paper to experiment we have taken following sets of data attributes

Table 1: Attribute Table of Data Sets

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

2.3. 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. Taking common template which weka provides for data transformation

```
//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)
{
System.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

2.4. Weka Data Format

Weka permits the input data set to be in numerous file formats like for users to load data like CSV (comma separted values: *.csv-taking MS-Excel tool help to convert as CSV), Binary Serialized Instances (*.bsi) etc. However, the most preferred and the most convenient input file format is the attribute relation file format (arff). So the first step in Weka always is taking an input file and making sure that it is in ARFF. It looks like as follows other wise using editor tool we can acompolish the above task.

2.5. Data Loading into Weka for Experiment Mode

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.

2.6. Working Principles of Selected Algorithms

A 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 effects 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. The problem can be optimized using the "Bayes Theorem" Its basically statistical principle that combines prior knowledge of the classes with the new collective records sets that gathered from current data. What exactly Bayes theorem states is let X and y be two pair of random variables. Their joint probability which represents P(X=x and Y=y) refers to the probability of X will approach to x and Y approach to y. If conditional probability P(Y=y | X=x) refers to the probability that the variable Y will take on y given in the circumstances where X is observed to have the value of x that these can be related as

Figure 2: ARFF file format-Source.weka.com

ARFF File Format Looks like

- Requisite declaration of @RELATION @ATTRIBUTE AND @ DATA
- FOLLOWED BY @ RELATION decleration associates a name with the dataset and relation-name.
- @ ATTRIBUTE decleration specifies the name and type of an attribute and data type
- @ DATA decleration is a single line denoting the starat of the data segment and missing values are represented by the special character?

P(X, Y) =P(Y/X) x P(X) =P (X|Y) x P(Y) in simple manner we can write the equation as

P (Y|X) =P (X|Y) P(Y)/P(X) (A)

Suppose there are m no of classes' c1, c2, c3……cm. The classifier predicts an unseen example of X as belonging to the class having the highest posterior probability condition on X. In other words we can say that if X is assigned to class Ci if and only if statement A is true. As P(X) is constant for all classes, only P(X/Ci) or P (Ci) needs to be maximized. Given a set of training data, P (Ci) can be estimated by counting how frequently each class in the sets of training data is occurring. To reduce the computational expenses is estimating P(X/Ci) for all possible of X, the classification makes a naïve assumption that the given attributes used in explaining X are fully conditionally independent of each other given the class of X. As a given attribute we have

P(X|Ci)=value of Pi ranges to j=1 to n of P(Xi|Ci)

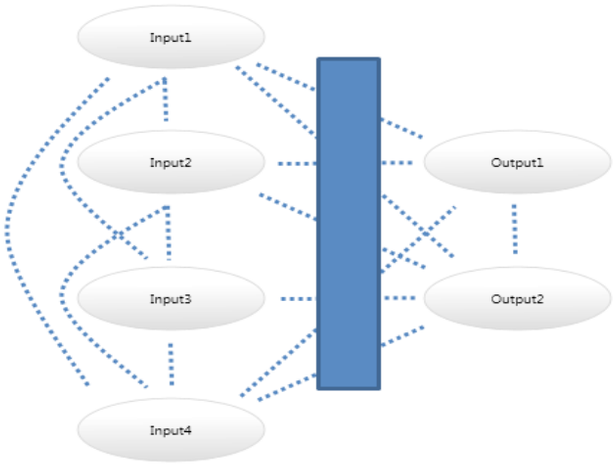
The Probabilities P(x1|C1),P(x2|C2), P(x3|C3) P(xn|Cn) can be estimated from the given training data sets. As in many situation Bayesian classifier is simple to use and very efficient to learn. It generally requires only one scan of the given training data. In some situtations we know the fact that independence assumption is often violated in practice for impliemenation of naïve bayes. In some cases Bayesian classifier is so robust (Domingos and Pazzani, 1997; Rish, 2001)

2.7. Bayes Belief Networks

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, logistics regression models, decision tress 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.

The listed drawback can be overcome some extent using statistically sounded alternative technique known as Bayesian networks. This technique is comprised with well-structured probability distributions concisely and comprehensively. They are drawn as network of nodes that represents one for each attribute in addition, connected by directed edges in such way that there are no redundant cycles only directed acyclic graphs.

Figure 3: Generic Structure of a Bayesian Network Classifier



Here we have to make assumptions that all attributes either training sets or test sets are all nominal and there are no missing values present in both cases. Some algorithms presents hidden attributes with values cannot be observed. A Bayesian network provides better way of using them at prediction time. Only one disadvantage is that predictions make more complex and time consuming.

Figure 3 depicts the possible structure of a Bayesian network used for classification in general. The dotted lines denote potential links, and the blue box is used to indicate that additional nodes and links can be added to the model, usually between the input and output nodes.

In order to perform classification with a Bayesian network such as the one depicted, first evidence must be set on the input nodes, and then the output nodes can be queried using standard Bayesian network inference. The result will be a distribution for each output node, so that you can not only determine the most probable state for each output, but also see the probability assigned to each output state.

2.8. 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.

2.9. OneR Algorithm General Interpretation

For each predictor in its class,

For each value of that predictor, make a rule as it follows;

Count how often each value of target (class) appears

Find the most frequent class

Make the rule assign that class to this value of the predictor

Calculate the total error of the rules of each predictor

Choose the predictor with the smallest total error.

Source: Data Mining Description of Saed Saad at http://www.saedsayad.com/oner.htm

2.10. Decision Trees with J48 Algorithm

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, 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.

Table 2: Comparative Runs using Training Sets

1) Used Training Sets	*NaiveBayes*	*BayesNet*	*OneR*	*Trees.J48*
Time taken to build model	0.01 Seconds	0.01 Seconds	0.02 Seconds	0.05 Seconds
Correctly classified instances	64 55.6522 %	39 33.913 %	36 31.3043 %	61 53.0435 %
Incorrectly classified instances	51 44.3478 %	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	48.6872 %	94.8255 %	70.1289 %	48.2321 %
Root Relative Absolute Error	78.415 %	95.5183 %	118.5218 %	69.5029 %

Table 3: Comparative Runs using Cross Validation of 10 Fold

1) Cross Validation	*NaiveBayes*	*BayesNet*	*OneR*	*Trees.J48*
Time taken to build model	0.002 Seconds	0.001 Seconds	0.002 Seconds	0.001 Seconds
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 %	100.8314 %	101.6067 %	101.3785 %
Root Relative Absolute Error	121.3516 %	101.5638 %	142.3195 %	121.2644 %

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.

It is important before we deploy this technique in constructing decision tree as variety of options available when using this algorithm, as they can make a significant difference in the quality of outcomes. In many situation, the default settings itself will prove adequate, but in others, each choice may require some consideration for higher results.

The J48 algorithm or Quinlan's c4.5 gives several options related to tree pruning. Various other algorithms attempt to "Prune", or simplify, their outcomes. Pruning generally produces fewer, more easily interpreted results. More importantly, pruning can be used as a tool to correct for potential over fitting of values.

The basic algorithm of J48 described above recursively classifies until each leaf of data is pure in its nature, meaning that the data has been categorized as close to perfectly as much as possible. This process proximity ensures maximum accuracy on the training data of our experiments, but it may create excessive rules that only describe particular idiosyncrasies of the given data. When tested on new data sets, the rules may be less effective. Pruning always reduces the accuracy of a model on training data. This is because pruning employs various means to relax the specificity of the decision tree or exemption nature of decision tree, hopefully improving its performance on test data. The overall concept is to gradually generalize a decision tree until it gains equal balance of flexibility and accuracy.

J48 generally employs two pruning methods. The first is known as sub tree replacement. This means that every node in a decision tree may be replaced with a leaf -- basically reducing the number of tests. This process starts from the leaves of the fully formed tree as decision tree, and works backwards toward the root of its original. Another important type of pruning used in J48 is termed sub tree rising. In this case, a node may be moved upwards towards the root of the tree, replacing other existing nodes along the way. Sub tree rising often has a negligible effect on decision tree models of any type. There is often no

Table 4: Comparative Runs using Split Test at 66 % Sets

1) Split Test of 66 %	*NaiveBayes*	*BayesNet*	*OneR*	*Trees.J48*
Time taken to build model	0.002 Seconds	0.001 Seconds	0.002 Seconds	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 %	101.5472 %	101.197 %
Root Relative Absolute Error	121.8041 %	101.7704 %	142.1861 %	121.305 %

clear way to predict the utility of the will help in effective manner, though it may be suggestive step to try turning it off if the induction process is taking a long time for execution. This is due to the fact that sub tree rising can be somewhat computationally complex as it moves backward towards the root.

The most basic parameter to employ is the tree pruning option. If we decide to employ tree pruning, then will need to consider the options above. Be aware that depending on how the training and test data have been defined in its class that the performance of unpruned tree may superficially appear better than a pruned one in its category. What we have seen in our results, because of over fitting. It is important to experiment with models by intelligently adjusting these parameters for our usage. Often, only repeated experiments can cause the familiarity with the data that will tease out the best set of options for creating optimal decision tree.

2.11. 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 experiments 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

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 referred:

http://www.pmean.com/definitions/kappa.htm

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

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

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

The MAE and the RMSE 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 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:

$$Ei = \sqrt{\frac{\sum_{j=1}^{n}\left(P_{(i)} - T_i\right)^2}{\sum_{j=1}^{n}\left(T_i - \bar{T}\right)^2}}$$

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.

3. Conclusion

Market information is an important aspect of Agricultural Marketing which is handy now days. The importance of sound agricultural marketing policies for ensuring fair returns to the farmers of its profit can hardly be over-emphasized. It, therefore, becomes necessary on the part of regulatory agencies and other policy holders to ensure remunerative prices to the farmers or retailers for the sale of their produce, to boost up their efforts for increasing and sustaining the agricultural production in India. A number of measures have been taken by the Government to protect and safeguard the interests of farmers, like regulation of price markets, grading of agricultural produce in India, cooperative marketing segment etc. Still the benefits are not percolating down factor to the farmers, as they are unable to plan their strategies for sale of their produce at remunerative prices, in this context correct and timely market information and advice about arrivals, prices, market trend, and other required information are crucial.

Here we are going to analyze the problem of agricultural data sets and its negotiating observation of various concept based algorithms. Generally classification works on the mode of attribute set of x. Further it can be mapped into class label y using classification model.

A classification model can serve as an explanatory tool to distinguish between objects of different classes known to be descriptive modeling. A classification model can also use to serve as predictive of unknown records of class label y. Classification techniques are most suited for predicting or describing data sets with either binary or nominal categories. They are less effective for ordinal categories like high, medium or low. In the paper "Classification methods" by Aijunan of York University, Canada makes point classification is the action of assigning an object to the specified category according to the characteristics of the object. Classification has wider scope in real time applications like in medical science predicting patients behavior, diseased based symptoms, analyzing the credit card fraudulent transactions etc. In this paper we are

experimenting the various patterns of agricultural data sets and its outcome incase if market is volatile.

References

1. Baik, S. & Bala, J. (2004). *A Decision Tree Algorithm for Distributed Data Mining: Towards Network Intrusion Detection, Lecture Notes in Computer Science*, 3046, 206-212.
2. Bouckaert, R. (2004). *Naive Bayes Classifiers That Perform Well with Continuous Variables, Lecture Notes in Computer Science*, 3339, 1089-1094.
3. Bouckaert, R. R. (2008). *Bayesian Network Classifiers in Weka*. Retrieved from remco@ cs.waikato.ac.nz.
4. Breslow, L. A. & Aha, D. W. (1997). Simplifying decision trees: A survey. *Knowledge Engineering Review* 12(1), 1–40.
5. Brighton, H. & Mellish, C. (2002). Advances in Instance Selection for Instance-Based Learning Algorithms. *Data Mining and Knowledge Discovery,* 6(2), 153-172.
6. 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.
7. Cheng, J., Greiner, R., Kelly, J., Bell, D. & Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence,* 137(1-2), 43-90.
8. Clark, P. & Niblett, T. (1989). The CN2 Induction Algorithm. *Machine Learning*, 3(4), 261-283.
9. Cover, T. & Hart, P. (1967). *Nearest Neighbor Pattern Classification*. IEEE Transactions on Information Theory, 13(1), 21-7.
10. Cowell, R. G. (2001). *Conditions under Which Conditional Independence and Scoring Methods Lead to Identical Selectionof Bayesian Network Models*. Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence.
11. Domingos, P. & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3), 103-130.
12. Elomaa, T. (1999). The biases of decision tree pruning strategies. *Lecture Notes in Computer Science* (1642, pp. 63-74). Springer.
13. Friedman, N. & Koller, D. (2003). Being Bayesian about Network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1), 95-125.
14. Friedman, N., Geiger, D. & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29(2-3), 131-163.
15. Han, J. & Kamber, M. (2011). *Data Mining Concepts and Techniques* (2nd Ed.). Elsevier.
16. Jensen, F. (1996). *An Introduction to Bayesian Networks*. Springer.
17. Kubat, M. & Cooperson M. (2001). A reduction technique for nearest-neighbor classification: Small groups of examples. Intelligent Data Analysis, 5(6), 463-476.
18. Madden, M. (2003). *The Performance of Bayesian Network Classifiers Constructed using Different Techniques*. Proceedings of European Conference on Machine Learning,
19. McSherry, D. (1999). Strategic induction of decision trees. *Knowledge-Based Systems*, 12(5- 6), 269-275.
20. Vivarelli, F. & Williams, C. (2001). Comparing Bayesian neural network algorithms for classifying segmented outdoor images. *Neural Networks* 14(4-5), 427-437.
21. Weka: Data Mining Software in JAVA (2013). Retrieved from http://www.cs.waikato.ac.nz/ml/weka.
22. Wilson, D. R. & Martinez, T. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38,257-286.
23. Workshop on Probabilistic Graphical Models for Classification, pp. 59-70.
24. Written, I. H. & Frank, E. (2007). *Data Mining Practical Machine Learning Tools and Techniques* (2ndEd.). Elsevier.
25. Yang, Y. & Webb, G. (2003). On why discretisization Works for Naive-Bayes Classifiers. *Lecture Notes in Computer Science*, 2903, 440-452.
26. Zhang, G. (2000). *Neural Networks for Classification: A Survey*. IEEE transactions on systems, man, and cybernetics, 30(4), 451-462.
27. Zheng, Z. (2000). Constructing *X*-of-*N* attributes for decision tree learning. *Machine Learning*, 40(1), 35-75.