Extemporizing the Data Trait

[1]Sakshi Jolly, *Research Scholar*
[2]Dr. Neha gupta, *Assistant Professor*
Faculty of Computer Applications (FCA)
MRIU, Faridabad

Abstract- *Information quality is a focal issue for some data arranged associations. Late advances in the information quality field mirror the view that a database is the result of an assembling procedure. While routine blunders, for example, non-existent postal divisions can be recognized and amended utilizing conventional information purifying apparatuses, numerous mistakes systemic to the assembling procedure can't be tended to. Thusly, the result of the information producing process is a loose recording of data about the substances of intrigue (i.e. clients, exchanges or resources). Thusly, the database is just a single (imperfect) adaptation of the elements it should represent. There are numerous arrangement calculations yet choice tree is the most normally utilized calculation due to its simplicity of execution and less demanding to comprehend contrasted with other grouping calculations. In this paper we are actualizing a calculation utilizing weka information mining apparatus utilizing freely accessible datasets of various sizes. This paper additionally gives bits of knowledge into the rate of precision it gives when a dataset contains missing esteems, missing information and vast measure of information*.

Keywords: *Data quality (DQ), Data Warehouse (DW), ETL(Extraction Transformation Loading)*

I. INTRODUCTION

The ETL significance originated from the meaning of its usefulness and the advancement exertion. ETL is in charge of getting the information from the heterogeneous sources frameworks into the DW so every disappointment in the ETL usefulness prompts stacking erroneous information in DW, which thusly prompts furnish administrators with mistaken information that prompting wrong decisions. "Data distribution center undertakings come up short for some reasons, all of which can be followed to a solitary cause: non quality"[2]. This raises the need to guarantee that the information in the source is reliable with the information that came to the DW. Singh and Singh [6], Wayne [9], and Kimball [10] consider ETL organize as the most pivotal stage in DW handle since the greatest duty of information quality endeavors lives in this stage. This prompts considering ETL arrange as bounty territory of DQ issues. Expecting to robotize ETL testing on DQ originated from the significance to test the general information that originated from the information sources and guarantee that it's stacked accurately into the goal DW. Serial execution of choice calculation is anything but difficult to actualize and attractive when little medium informational indexes are included. In this paper we will execute Hybrid K Means Decision Tree Algorithm (HKMDT) using weka, serially. The point in testing the information quality in the ETL is to guarantee the accuracy of ETL strategies and regardless of whether it should be re-intended to moderate the issues. The point of this paper is to mechanize test schedules that check the information quality parameters (fulfillment, consistency, uniqueness, legitimacy, convenience, and precision)

II. HKMDT ALGORITHM

Choice tree learning is a strategy ordinarily utilized as a part of information mining. The objective is to make a model that predicts the estimation of an objective variable in light of a few info factors. Every inside hub compares to one of the information factors; there are edges to youngsters for each of the conceivable estimations of that info variable. Each leaf speaks to an estimation of the objective variable given the estimations of the info factors spoke to by the way from the root to the leaf.

A choice tree is a straightforward portrayal for characterizing cases. For this area, expect that the greater part of the info highlights have limited discrete spaces, and there is a solitary target include called the "arrangement". Every component of the area of the order is known as a class. A choice tree or an order tree is a tree in which each interior (non-leaf) hub is named with an information highlight. The circular segments originating from a hub named with an information include are marked with each of the conceivable estimations of the objective or yield highlight or the curve prompts a subordinate choice hub on an alternate info highlight. Each leaf of the tree is marked with a class or likelihood dispersion over the classes.

A tree can be "educated" by part the source set into subsets in view of a quality esteem test. This procedure is rehashed on each inferred subset in a recursive way called recursive apportioning. The recursion is finished when the subset at a hub has all a similar estimation of the objective variable, or while part never again increases the value of the forecasts. In information mining, choice trees can be depicted likewise as the mix of numerical and computational strategies to help the portrayal, classification and speculation of a given arrangement of information.

*Our Novel Algorithm is **Hybrid K Means Decision Tree (HKMDT).***

Hybrid K Means Decision Tree Algorithm(HKMDT) Proposed Approach
HKMDT=K Means+C4.5
Data Set are Trained by K Means then Tested By C4.5
Accuracy Rate: 98.00% to 99.50%

Algorithm: K-Means Clustering Build HKMDT Decision Tree

Input: K- the number of clusters and R the records of the dataset, the training data T, the attributes_available for computing the next branch

Output: A HKMDT decision tree

Method:

Step 1: Randomly choose K objects and make them the K cluster centroids

Step 2: Do

Step 3: For each record in R

Step 4: Calculate distance between each cluster centroid and the record.

Step 5: Assign the record to the cluster that has the minimum distance.

Step 6: Recalculate the cluster means (the values of attributes in the cluster / number of records in the cluster).

Step 7: End for loop

Step 8: While records assignment to clusters do not change

Step 9: End function

Step 10: create a node N.

Step 11: if all records in T have same target class

Step 12: return N as a leaf node with target class.

Step 13: if attributes_available is empty

Step 14: return N as leaf node with maximum target class for the records.

Step 15: Get best_attribute (T, attributes_available).

Step 16: attributes_available = attributes_available – best_attribute.

Step 17: Split the records based on best_attribute(best_attribute, T) //for each split, grown a subtree by calling the //Build HKMDT Decision Tree function

Step 18: for each split Ti of T on best_attribute

Step 19: attach a new node returned by build HKMDT DecisionTree(split records Ti , attributes_available)

Step 20: end for

Step 21: end function

III. IMPLEMENTATION

The Implementation Plan depicts how the data framework will be conveyed, introduced and changed into an operational framework. The arrangement contains a diagram of the framework, a short depiction of the real undertakings associated with the usage, the general assets expected to help the execution exertion, (for example, equipment, programming. offices, materials, and staff), and any site-particular usage necessities. The arrangement is produced amid the Design Phase and is refreshed amid the Development Phase; the last form is given in the Integration and Test Phase and is utilized for direction amid the Implementation Phase.

Implementation Plan:

1. Load the dataset using HKMDT.

2. K means reads every row of the data set. Instead of this we need to track the missing values, replicate values, cryptic values in every row & columns.

3. Fixing all the pre- processing issues of trained data set.

4. Test the data set to C4.5 Decision Tree to get the Better Accuracy Rate

In order to classify our data, first we need to load the dataset. This will be done in wekaexplorer window.

Stage 1 Process:

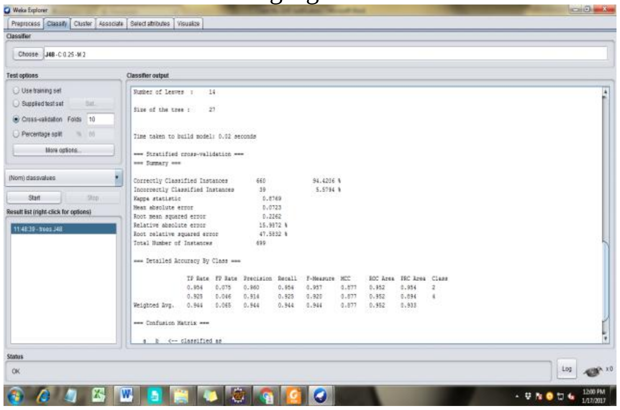
Example: Breast Cancer Data Set, Initially process the dataset to C4.5 Decision Tree

Output:

Correctly Classified Instances 660
94.4206 %
Incorrectly Classified Instances 39
5.5794 %
Kappa statistic
0.876 9
Mean absolute error
0.0723
Root mean squared error
0.2262
Relative absolute error
15.9872 %
Root relative squared error
47.5832 %
Total Number of Instances 699
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure
MCC ROC Area PRC Area Class
0.954 0.075 0.960 0.954 0.957 0.877
0.952 0.954 2
0.925 0.046 0.914 0.925 0.920 0.877
0.952 0.894 4
Weighted Avg. 0.944 0.065 0.944 0.944
0.944 0.877 0.952 0.933

=== Confusion Matrix ===
a b <-- classified as
437 21 | a = 2
18 223 | b = 4

As per our research is concerned we need to click on classify tab. This window consists of various classifiers like bays, functions, lazy, meta and tree etc. available in weka. We first click on trees, then choose J48 (c4.5 is termed as J48 in weka software) which results in following figure.



J48 -C 0.25 -M 2

Folds 10

11:48:39 - trees.J48

```
Number of Leaves  :     14

Size of the tree :      27


Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         660               94.4206 %
Incorrectly Classified Instances        39                5.5794 %
Kappa statistic                          0.8769
Mean absolute error                      0.0723
Root mean squared error                  0.2262
Relative absolute error                 15.9872 %
Root relative squared error             47.5832 %
Total Number of Instances              699

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.954    0.075    0.960      0.954    0.957      0.877    0.952     0.954     2
                 0.925    0.046    0.914      0.925    0.920      0.877    0.952     0.894     4
Weighted Avg.    0.944    0.065    0.944      0.944    0.944      0.877    0.952     0.933

=== Confusion Matrix ===

   a   b   <-- classified as
```

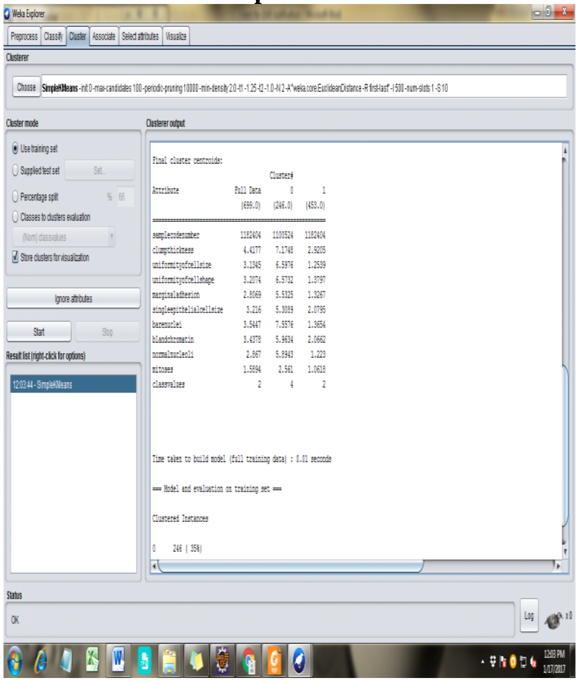
OK

Fig1. Classification Panel and Weka run information

Stage 2:

Initially Dataset will trained by K means and Tested by C4.5 Decision Tree Algorithm.

K Means Trained Output:



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

12:03:44 - SimpleKMeans

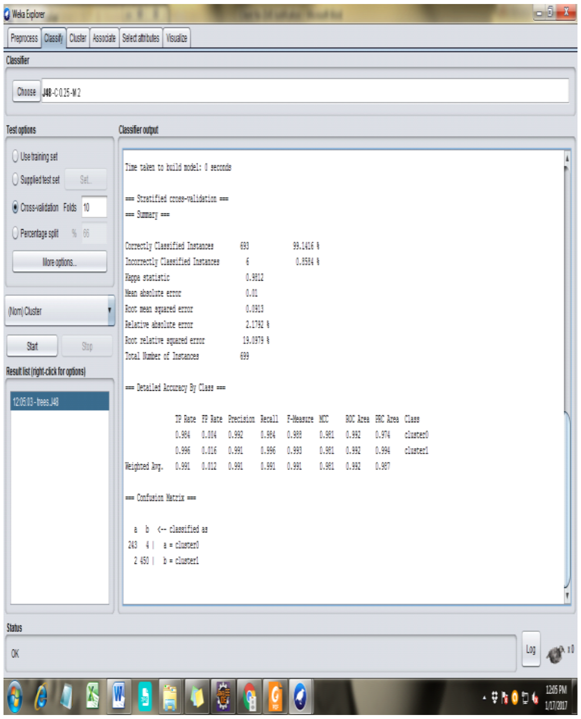
```
Final cluster centroids:
                                     Cluster#
Attribute                Full Data          0          1
                           (699.0)    (246.0)    (453.0)
=========================================================
samplecodenumber           1182404    1100524    1182404
clumpthickness              4.4177     7.1748     2.9205
uniformityofcellsize        3.1345     6.5976     1.2539
uniformityofcellshape       3.2074     6.5732     1.3797
marginaladhesion            2.8069     5.5325     1.3267
singleepithelialcellsize     3.216     5.3089     2.0795
barenuclei                  3.5447     7.5576     1.3654
blandchromatin              3.4378     5.9634     2.0662
normalnucleoli               2.867     5.8943      1.223
mitoses                     1.5894      2.561     1.0618
classvalues                      2          4          2




Time taken to build model (full training data) : 0.01 seconds


=== Model and evaluation on training set ===


Clustered Instances


0      246 ( 35%)
```

OK

Fig 2.Resultant

C4.5 Tested Output:

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 693
99.1416 %
Incorrectly Classified Instances 6
0.8584 %
Kappa statistic 0.9812
Mean absolute error 0.01
Root mean squared error 0.0913
Relative absolute error 2.1792 %
Root relative squared error 19.0979 %
Total Number of Instances 699
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure
MCC ROC Area PRC Area Class
0.984 0.004 0.992 0.984 0.988 0.981
0.992 0.974 cluster0
0.996 0.016 0.991 0.996 0.993 0.981
0.992 0.994 cluster1
Weighted Avg. 0.991 0.012 0.991 0.991
0.991 0.981 0.992 0.987
== Confusion Matrix ===
a b <-- classified as
243 4 | a = cluster0
2 450 | b = cluster1



Choose J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Cluster

12:05:03 - trees.J48

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         693               99.1416 %
Incorrectly Classified Instances         6                0.8584 %
Kappa statistic                          0.9812
Mean absolute error                      0.01
Root mean squared error                  0.0913
Relative absolute error                  2.1792 %
Root relative squared error             19.0979 %
Total Number of Instances              699

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.984    0.004    0.992      0.984    0.988      0.981    0.992     0.974     cluster0
                 0.996    0.016    0.991      0.996    0.993      0.981    0.992     0.994     cluster1
Weighted Avg.    0.991    0.012    0.991      0.991    0.991      0.981    0.992     0.987

=== Confusion Matrix ===

   a   b   <-- classified as
 243   4 |   a = cluster0
   2 450 |   b = cluster1
```

OK

Fig 3: Tested Output

IV. RESULTS ACHIEVED

- Filling the missing esteems from the heterogeneous information sources,
- Discovering conceivable answers for stay away from the spurious esteems,
- Recognizing obscure esteems
- Distinguishing repudiating information.
- Keeping up the exactness, uprightness, consistency, non – excess of information quality in a convenient way.

V. CONCLUSION AND LIGHTS TO THE FUTURE

In this paper we first show implementation of HKM decision tree algorithm. After that rate of precision it gives when dataset contains commotion, when there is some missing information in a dataset and when a . dataset contains number of occurrences in it. The exploratory outcomes demonstrate that HKMDT gives more prominent exactness in each above said case. In this examination we concentrated on serial usage of choice tree calculation which is memory occupant, quick and simple to execute. In future we will go for its parallel execution which is relatively mind boggling and assess how much exactness this calculation gives all things considered.

REFERENCES

1. Anyanwu, M., and Shiva, S. (2009). Application of Enhanced Decision Tree Algorithm toChurn Analysis. 2009 International Conference on Artificial Intelligence and PatternRecognition (AIPR-09), Orlando Florida

2. Jiawei han and micheline kamber. Data mining concepts and techniques, second edition,285-291

3. Matthew N.anyanwu, Sajjan g. shiva. Comparative analysis of serial decision tree classification algorithms

4. Mehdi piroozma, Youping deng, jack y yang and mary qu yang. A comparative study of different machine learning methods on microarray gene expression data, BMC genomics

5. Tzung-I tang,Gang Zheng, Yalou huang,Guangfu Shu,Pengtao wang. A comparative study of medical data classification methods based on decision tree and system reconstruction analysis. IEMS vol.4,no.1,pp-102-108,june 2005

6. Xu,M, wang, J. Chen, T. (2006). Improved decision tree algorithm: ID3+, intelligent computing in signal Processing and pattern recognition, Vol. 345, PP.141-149.

7. Wayne W. E. (2004) "*Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data* ",The Data warehouse Institute (TDWI) report , available at www.dw-institute.com .

8.The Standish Group (1999), "Migrate Headaches," available at *www.it-cortex.com/start_failure_rate.htm*

9. Ralaph Kimball, The Data Warehouse ETL Toolkit, Wiley India (P) Ltd (2004).

10. Tech Notes (2008), Why Data Warehouse Projects Fail: Using Schema Examination Tools to Ensure Information Quality, Schema Compliance, and Project Success. Embarcadero Technologies. Available at www.embarcadero.com.

11. Markus Helfert, Gregor Zellner, Carlos Sousa, "Data Quality Problems and Proactive DataQuality Management in Data-Warehouse-Systems"