Analysis of a Population of Cataract Patients Databases in Weka Tool

C.sugandhi, P.Yasodha, M.Kannan

ABSTRACT: Data mining refers to extracting knowledge from large amount of data. Real life data mining approaches are interesting because they often present a different set of problems for data miners. The process of designing a model helps to identify the different Cataract Diseases. A cataract can cause a decrease in visual function, which in turn can be classified as a visual disability. Thus, cataract can be defined in three ways. The first definition is an objective lens change. The second is a lens opacity that is associated with a defined level of visual acuity loss. The third relates to the functional consequences of lens opacification. This guideline focuses on the last definition. It deals with care of the patient with functional impairment due to cataract and improvement in function as a result of treatment for the condition. Taking into account the prevalence of cataract among men and women the study is aimed at finding out the characteristics that determine the presence of cataract and to track the maximum number of men and women suffering from cataract with 790 population using weka tool. In this paper the data classification is cataract patients data set is developed by collecting data from hospital repository consists of 790 instances with 11 different attributes. The instances in the Dataset are pertaining to the categories of Kerato meter reading (RE), Kerato meter reading (LE), Axil Length(RE), Axil Length(LE), Power(RE), Power(LE), Cataract Disease . WEKA tool is used to classify the data and the data is evaluated using 10-fold cross validation and the results are compared.

Keywords: Data Mining, Classification, Decision Tree Algorithm, Weka tool

1 INTRODUCTION

The main focus of this paper is the classification of different types of datasets that can be performed to determine if a person has cataract disease. For this reason, the goal of this research is classifier in order to correctly classify the datasets. The major motivation for this work is that Cataract affects a large number of the world population and it's a hard disease to diagnose.

Experimental statistical investigations performed with signals divided in to five groups- mature cataract, immature form of cataract, incipience cataract phase, healthy lenses and human eye phantom. Investigations have showed that value of specific quality in the test groups vary in the wide range from 1 to 60. This feature allows theoretically differentiating eye lenses, cataract in different classes with defined boundaries. Presented results show that we with high reliability can differentiate lenses in to three groups: healthy lenses (Q£ >50), lenses with incipient or immature cataract (Q£ = 2-20) and lenses with mature cataract (Q£ < 1). The investigated method can be used for eye lens classification and for early cataract detection. This Guideline will assist optometrists in achieving the following goals:

- Identify patients at risk of developing cataracts
- Accurately diagnose cataracts
- Improve the quality of care rendered to patients with cataracts
- Effectively manage patients with cataracts
- Identify and manage postoperative complications
- Inform and educate patients and other health care practitioners about the visual complications and functional disability from cataracts and the availability of treatment

The dissertation is used to classify the cataract based on the Op.no, Name, Age, Sex, Kerato meter reading (RE), Kerato meter reading (LE), Axil Length(RE), Axil Length(LE),, Power(RE), Power(LE), Cataract Disease. This may be achieved through collecting the data necessary for the system, determining the data mining technique to be used for the system, and choosing the most suitable implementation tool for the domain.

This is our main concern, to optimize the task of correctly selecting the set of medical tests that a patient must perform to have the best, the less expensive and time consuming diagnosis possible. This paper will focus on the analysis of data from a data set called cataract data set.

2. Related Work

Iskander et al [1] developed a method of modeling corneal shape using Zernike polynomials with a goal of determining the optimal number of coefficients to use in constructing the model. In a follow-up study, the same group concluded that a 4th order Zernike polynomial was adequate in modeling the majority of corneal surfaces [2]. In a earlier study, our group showed that with Zernike polynomials, lower order transformations were able to capture the general shape of the cornea and were unaffected by noise [3]. We also found that higher order transformations were able to provide a better model fit, but were also more susceptible to noise.

Forbes et al [4] reviewed the recent literature on the surgical management of cataracts in children. They report that the success of surgery and long-term management of infants and children with cataracts has improved. They cite progress in getting infants in earlier for cataract surgery, better forms of optical correction including more use of the intraocular lens, and the employment of many surgical techniques in children which were developed for older people with cataracts. One major topic of discussion was the need for early surgery, at or before 6 weeks of age, before the development of binocular vision; that is, before the two eyes start to work together for single vision and depth perception.

Ruth et al [5] presented a case report of an infant with a congenital cataract in one eye. The patient underwent cataract extraction and intraocular lens implantation at 8 weeks of age. The patient ended up with better vision in the eye that had a cataract than in the other "normal" eye.

Harris et al [6] is nearest equivalent sphere since SE, incorrectly, implies that spherical power exists that is equivalent to the spherocylindrical power. Because of its simplicity, SE is much used in clinical observance and research. The SE satisfies certain basic requirements and can, therefore, be used in statistical analyses to provide means, variances and so forth (Kaye & Harris 2002). An analysis done in SE alone loses, however, information about the other component of refractive power, the astigmatic component.

3. TOOLS AND TECHNIQUES

This paper proposes to use an open source data mining tool-Weka. Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato. WEKA is free software available under the GNU General Public License. The Weka workbench contains a collection of visualization tools and algorithms for data analysis and predictive modelling, together with graphical user interfaces for easy access to this functionality. However, as is mentioned above, Weka is an open source data mining tool which can be extended by the users, that helps users a lot, when tools Weka provides that can not meet the users requirement, they can develop new tool kits and add them to Weka. Therefore, Weka is a very good data mining tool which could be used in the field of education. In that we are going to use classification technique. The classification techniques of data mining help to classify the data on the basis of certain rules. This helps to frame policies for the future.

3.1 Data Mining by WEKA Engine

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. It is also wellsuited for developing new machine learning schemes. Weka is developed by the University of Waikato. In our paper we used the Weka as data mining engine, and made a bridge between the and Weka. No user needs to install Weka in his workstation, but it do already enough to be installed on server machine

3.1.1 Classification

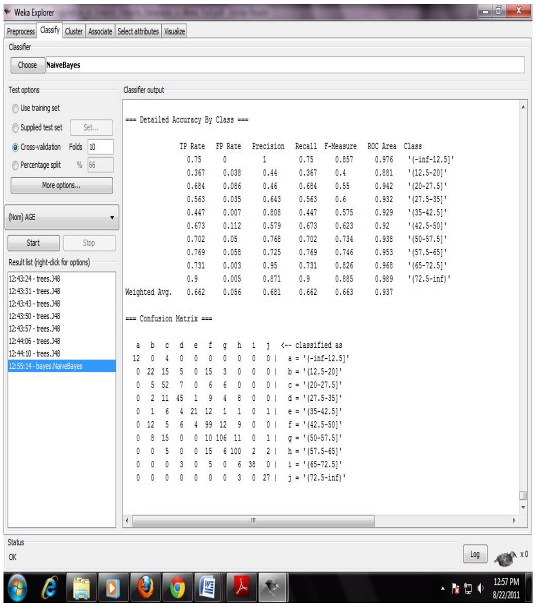
Classification is a data mining (machine learning) technique used to predict group membership for data instances. For example, you may wish to use classification to predict whether the weather on a particular day will be “sunny”, “rainy” or “cloudy”. Popular classification techniques include decision trees and neural networks.

NAIVEBAYES CLASSIFIER

A NaiveBayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from

Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

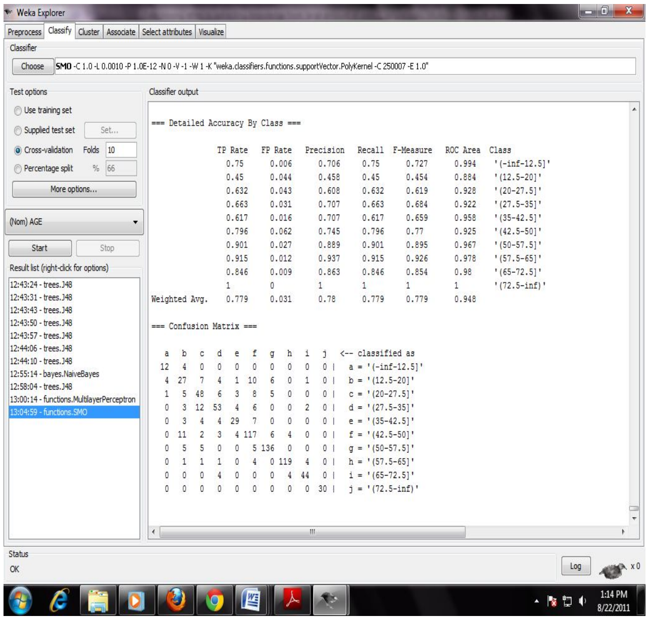


```
Classifier: NaiveBayes
Test options: Cross-validation  Folds 10
(Nom) AGE

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.75      0          1         0.75      0.857      0.976    '(-inf-12.5]'
                 0.367     0.038      0.44      0.367     0.4        0.881    '(12.5-20]'
                 0.684     0.086      0.46      0.684     0.55       0.942    '(20-27.5]'
                 0.563     0.035      0.643     0.563     0.6        0.932    '(27.5-35]'
                 0.447     0.007      0.808     0.447     0.575      0.929    '(35-42.5]'
                 0.673     0.112      0.579     0.673     0.623      0.92     '(42.5-50]'
                 0.702     0.05       0.768     0.702     0.734      0.938    '(50-57.5]'
                 0.769     0.058      0.725     0.769     0.746      0.953    '(57.5-65]'
                 0.731     0.003      0.95      0.731     0.826      0.968    '(65-72.5]'
                 0.9       0.005      0.871     0.9       0.885      0.989    '(72.5-inf)'
Weighted Avg.    0.662     0.056      0.681     0.662     0.663      0.937

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
  12   0   4   0   0   0   0   0   0   0 |   a = '(-inf-12.5]'
   0  22  15   5   0  15   3   0   0   0 |   b = '(12.5-20]'
   0   5  52   7   0   6   6   0   0   0 |   c = '(20-27.5]'
   0   2  11  45   1   9   4   8   0   0 |   d = '(27.5-35]'
   0   1   6   4  21  12   1   1   0   1 |   e = '(35-42.5]'
   0  12   5   6   4  99  12   9   0   0 |   f = '(42.5-50]'
   0   8  15   0   0  10 106  11   0   1 |   g = '(50-57.5]'
   0   0   5   0   0  15   6 100   2   2 |   h = '(57.5-65]'
   0   0   0   3   0   5   0   6  38   0 |   i = '(65-72.5]'
   0   0   0   0   0   0   0   3   0  27 |   j = '(72.5-inf)'
```

Figure 1: Naïve Bayes

SMO

This is a standard algorithm that is widely used for practical machine learning. Part is a more recent scheme for producing sets of rules called "decision lists"; it works by forming partial decision trees and immediately converting them into the corresponding rule. SMO implements the "sequential minimal optimization" algorithm for support vector machines, which are an important new paradigm in machine learning

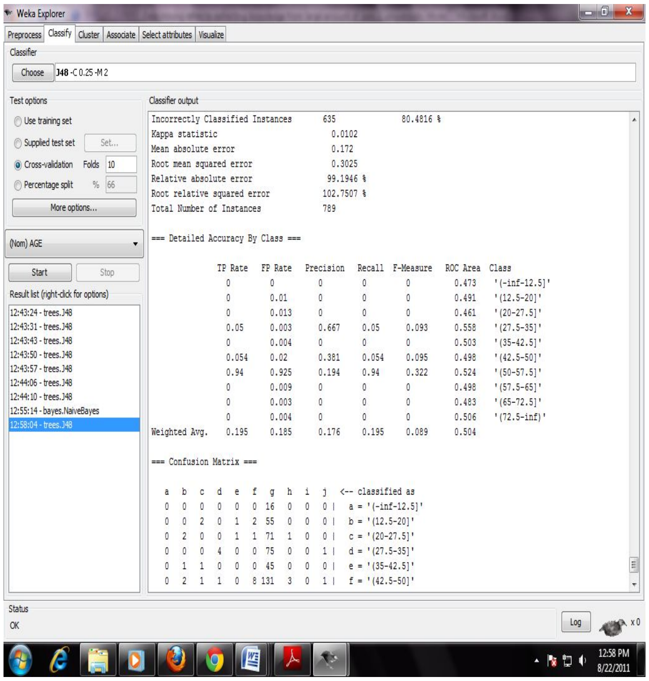


```
Classifier: SMO -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"
Test options: Cross-validation  Folds 10
(Nom) AGE

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.75      0.006      0.706     0.75      0.727      0.994    '(-inf-12.5]'
                 0.45      0.044      0.458     0.45      0.454      0.884    '(12.5-20]'
                 0.632     0.043      0.608     0.632     0.619      0.928    '(20-27.5]'
                 0.663     0.031      0.707     0.663     0.684      0.922    '(27.5-35]'
                 0.617     0.016      0.707     0.617     0.659      0.958    '(35-42.5]'
                 0.796     0.062      0.745     0.796     0.77       0.925    '(42.5-50]'
                 0.901     0.027      0.889     0.901     0.895      0.967    '(50-57.5]'
                 0.915     0.012      0.937     0.915     0.926      0.978    '(57.5-65]'
                 0.846     0.009      0.863     0.846     0.854      0.98     '(65-72.5]'
                 1         0          1         1         1          1        '(72.5-inf)'
Weighted Avg.    0.779     0.031      0.78      0.779     0.779      0.948

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
  12   4   0   0   0   0   0   0   0   0 |   a = '(-inf-12.5]'
   4  27   7   4   1  10   6   0   1   0 |   b = '(12.5-20]'
   1   5  48   6   3   8   5   0   0   0 |   c = '(20-27.5]'
   0   3  12  53   4   6   0   0   2   0 |   d = '(27.5-35]'
   0   3   4   4  29   7   0   0   0   0 |   e = '(35-42.5]'
   0  11   2   3   4 117   6   4   0   0 |   f = '(42.5-50]'
   0   5   5   0   0   5 136   0   0   0 |   g = '(50-57.5]'
   0   1   1   1   0   4   0 119   4   0 |   h = '(57.5-65]'
   0   0   0   4   0   0   0   4  44   0 |   i = '(65-72.5]'
   0   0   0   0   0   0   0   0   0  30 |   j = '(72.5-inf)'
```

Figure 2: SMO

J48 Pruned Tree

J48 is a module for generating a pruned or unpruned C4.5 decision tree. When we applied J48 onto refreshed data.

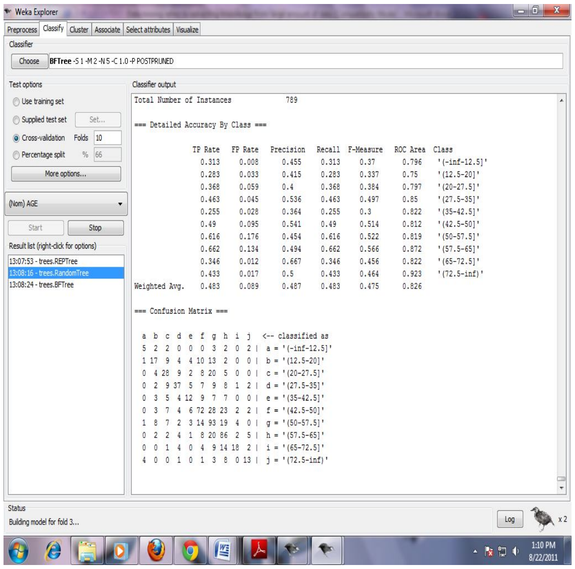


```
Classifier: J48 -C 0.25 -M 2
Test options: Cross-validation  Folds 10
(Nom) AGE

Incorrectly Classified Instances       635               80.4816 %
Kappa statistic                          0.0102
Mean absolute error                      0.172
Root mean squared error                  0.3025
Relative absolute error                 99.1946 %
Root relative squared error            102.7507 %
Total Number of Instances              789

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0          0         0         0          0.473    '(-inf-12.5]'
                 0         0.01       0         0         0          0.491    '(12.5-20]'
                 0         0.013      0         0         0          0.461    '(20-27.5]'
                 0.05      0.003      0.667     0.05      0.093      0.558    '(27.5-35]'
                 0         0.004      0         0         0          0.503    '(35-42.5]'
                 0.054     0.02       0.381     0.054     0.095      0.498    '(42.5-50]'
                 0.94      0.925      0.194     0.94      0.322      0.524    '(50-57.5]'
                 0         0.009      0         0         0          0.498    '(57.5-65]'
                 0         0.003      0         0         0          0.483    '(65-72.5]'
                 0         0.004      0         0         0          0.506    '(72.5-inf)'
Weighted Avg.    0.195     0.185      0.176     0.195     0.089      0.504

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
   0   0   0   0   0   0  16   0   0   0 |   a = '(-inf-12.5]'
   0   0   2   0   1   2  55   0   0   0 |   b = '(12.5-20]'
   0   2   0   0   1   1  71   1   0   0 |   c = '(20-27.5]'
   0   0   0   4   0   0  75   0   0   1 |   d = '(27.5-35]'
   0   1   1   0   0   0  45   0   0   0 |   e = '(35-42.5]'
   0   2   1   1   0   8 131   3   0   1 |   f = '(42.5-50]'
```

Figure 3: J48

REPTREE

Fast decision tree learner. Builds a decision/regression tree using information gain/variance reduction and prunes it using reduced-error pruning (with backfitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5). The table below describes the options available for REPTree.



Classifier: BFTree -S 1 -M 2 -N 5 -C 1.0 -P POSTPRUNED

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

(Nom) AGE

Result list (right-click for options)
13:07:53 - trees.REPTree
13:08:16 - trees.RandomTree
13:08:24 - trees.BFTree

Classifier output

```
Total Number of Instances              789

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.313     0.008      0.455     0.313     0.37       0.796    '(-inf-12.5]'
                 0.283     0.033      0.415     0.283     0.337      0.75     '(12.5-20]'
                 0.368     0.059      0.4       0.368     0.384      0.797    '(20-27.5]'
                 0.463     0.045      0.536     0.463     0.497      0.85     '(27.5-35]'
                 0.255     0.028      0.364     0.255     0.3        0.822    '(35-42.5]'
                 0.49      0.095      0.541     0.49      0.514      0.812    '(42.5-50]'
                 0.616     0.176      0.454     0.616     0.522      0.819    '(50-57.5]'
                 0.662     0.134      0.494     0.662     0.566      0.872    '(57.5-65]'
                 0.346     0.012      0.667     0.346     0.456      0.822    '(65-72.5]'
                 0.433     0.017      0.5       0.433     0.464      0.923    '(72.5-inf)'
Weighted Avg.    0.483     0.089      0.487     0.483     0.475      0.826

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
  5  2  2  0  0  0  3  2  0  2 |  a = '(-inf-12.5]'
  1 17  9  4  4 10 13  2  0  0 |  b = '(12.5-20]'
  0  4 28  9  2  8 20  5  0  0 |  c = '(20-27.5]'
  0  2  9 37  5  7  9  8  1  2 |  d = '(27.5-35]'
  0  3  5  4 12  9  7  7  0  0 |  e = '(35-42.5]'
  0  3  7  4  6 72 28 23  2  2 |  f = '(42.5-50]'
  1  8  7  2  3 14 93 19  4  0 |  g = '(50-57.5]'
  0  2  2  4  1  8 20 86  2  5 |  h = '(57.5-65]'
  0  0  1  4  0  4  9 14 18  2 |  i = '(65-72.5]'
  4  0  0  1  0  1  3  8  0 13 |  j = '(72.5-inf)'
```

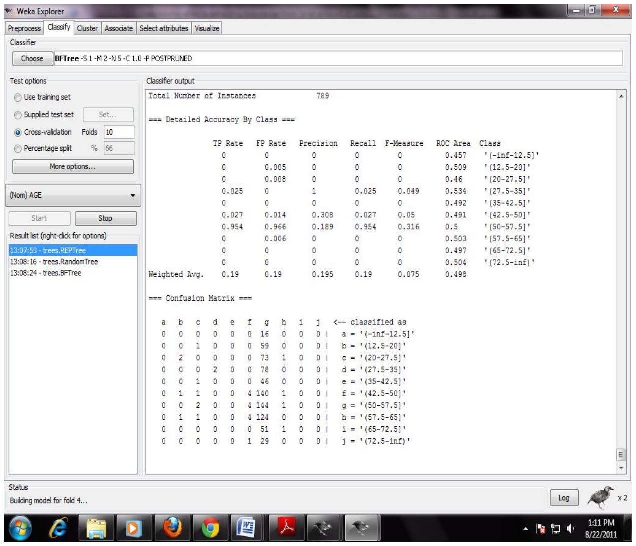
Status
Building model for fold 3...

Figure 4: REP TREE

RANDOM TREE

Algorithm	Correctly classified Instances	Mean Absolute Error
NaïveBayes	66.159%	0.0898
SMO	77.946%	0.1618
J48	19.518%	0.172
REPTree	48.289%	0.1143
Random Tree	84.8739%	0.231

Class for constructing a tree that considers K randomly chosen attributes at each node. Performs no pruning. Also has an option to allow estimation of class probabilities based on a hold-out set (backfitting).



Classifier: BFTree -S 1 -M 2 -N 5 -C 1.0 -P POSTPRUNED

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

(Nom) AGE

Result list (right-click for options)
13:07:53 - trees.REPTree
13:08:16 - trees.RandomTree
13:08:24 - trees.BFTree

Classifier output

```
Total Number of Instances              789

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0          0         0         0          0.457    '(-inf-12.5]'
                 0         0.005      0         0         0          0.509    '(12.5-20]'
                 0         0.008      0         0         0          0.46     '(20-27.5]'
                 0.025     0          1         0.025     0.049      0.534    '(27.5-35]'
                 0         0          0         0         0          0.492    '(35-42.5]'
                 0.027     0.014      0.308     0.027     0.05       0.491    '(42.5-50]'
                 0.954     0.966      0.189     0.954     0.316      0.5      '(50-57.5]'
                 0         0.006      0         0         0          0.503    '(57.5-65]'
                 0         0          0         0         0          0.497    '(65-72.5]'
                 0         0          0         0         0          0.504    '(72.5-inf)'
Weighted Avg.    0.19      0.19       0.195     0.19      0.075      0.498

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
   0   0   0   0   0   0  16   0   0   0 |   a = '(-inf-12.5]'
   0   0   1   0   0   0  59   0   0   0 |   b = '(12.5-20]'
   0   2   0   0   0   0  73   1   0   0 |   c = '(20-27.5]'
   0   0   0   2   0   0  78   0   0   0 |   d = '(27.5-35]'
   0   0   1   0   0   0  46   0   0   0 |   e = '(35-42.5]'
   0   1   1   0   0   4 140   1   0   0 |   f = '(42.5-50]'
   0   0   2   0   0   4 144   1   0   0 |   g = '(50-57.5]'
   0   1   1   0   0   4 124   0   0   0 |   h = '(57.5-65]'
   0   0   0   0   0   0  51   1   0   0 |   i = '(65-72.5]'
   0   0   0   0   0   1  29   0   0   0 |   j = '(72.5-inf)'
```

Status
Building model for fold 4...

Figure 5: Random Tree

4. RESULTS & DISCUSSIONS

This research is a starting attempt to use data mining functions to analyze and evaluate Cataract patient data. we classify data set with five different classification algorithms: Naïve Bayes, SMO, J48, REP Tree and Random Tree. In our study we are going to compare the correctly classified instances as well as mean absolute error with different algorithms they are, Naive Bayes, SMO, J48, and REP Tree and Random Tree. The following table shows this comparison.

This will help to improve the performance of such cataract patient in the early stage. The future work will be focused on using the other classification algorithms of data mining. It is a known fact that the performance of an algorithm is dependent on the domain and the type of the data set.

5. CONCLUSION

This paper illustrates how well different classification techniques are used as predictive tools in the data mining domain and after comparing their performances. From the results it is proven that Random Tree algorithm is most appropriate for cataract performance. Random Tree gives 84% which is relatively higher than other algorithms. This study is an attempt to use classification algorithms for the cataract performance and comparing the performance of NaiveBayes, SMO, J48, and REPTree and Random Tree .

REFERENCES

[1] D. R. Iskander, M. J. Collins, and B. Davis, "Optimal modeling of corneal surfaces with zernike polynomials," IEEE Trans Biomed Eng, vol. 48, no. 1, pp. 87–95., 2001.

[2] D. R. Iskander, M. J. Collins, B. Davis, and R. Franklin, "Corneal surface characterization: How many zernike terms should be used? (arvo abstract)," Invest Ophthalmol Vis Sci, vol. 42, no. 4, p. S896, 2001.

[3] M. D. Twa, S. Parthasarathy, T. W. Raasch, and M. A. Bullimore, "Automated classification of keratoconus: A case study in analyzing clinical data," in SIAM Int'l Conference on Data Mining, San Francisco, CA, 2003.

[4] Forbes "The deficit in cataract surgery in England and Wales and the escalating problem of visual impairment": epidemiological modelling of the population dynamics of cataract. Br J Ophthalmol 2000; 84: 4-8.

[5] Ruth W. Age-related utilisation of cataract surgery in Sweden during 1992-1999. A retrospective study of cataract surgery rate in one-year age groups based on the Swedish National Cataract Register. Acta Ophthalmol Scand 2001; 79: 342-349.

[6] Harris WF. Direct, vec and other squares, and sample variance-covariance of dioptric power.Harris WF:Astigmatism. Ophthal Physiol Opt. 2000;20:11–30.

[11] N. Maeda, S. D. Klyce, M. K. Smolek, and H. W. Thompson, "Automated keratoconus screening with corneal topography analysis," Invest Ophthalmol Vis Sci, vol. 35, no. 6, pp. 2749–57, 1994.

[7] Y. S. Rabinowitz and K. Rasheed, "KISA% - minimal topographic criteria for diagnosing keratoconus," J Cataract Refract Surg, vol. 25, no. 10, pp. 1327–35, 1999.

[8] M. K. Smolek and S. D. Klyce, "Screening of prior refractive surgery by a wavelet-based neural network," J Cataract Refract Surg, vol. 27, no. 12, pp. 1926–31., 2001.

[9] N. Maeda, S. D. Klyce, and M. K. Smolek, "Neural network classification of corneal topography. preliminary demonstration," Invest Ophthalmol Vis Sci, vol. 36, no. 7, pp. 1327–35., 1995.

[10] K. Marsolo and S. Parthasarathy, "Alternate representation of distance matrices for characterization of protein structure," in 5th IEEE International Conference on Data Mining (ICDM05), 2005.

AUTHOR PROFILE

C.Sugandhi received her Master Degree from Jaya College , Arakonam, and currently doing M.Phil Research at SCSVMV University, Kanchipuram. Her research interest lies in the area of Data Mining.

P.Yasodha Mphil Research Scholar in the Department of Computer Science & Applications in Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya University, Enathur, Kanchipuram. She received the degree in Master of Computer Applications from SCSVMV University in 2010. At present she is working as Professor at Department of Computer Science in Pachaiyappa's College for Women, kanchipuram, Tamil Nadu. She has published 3 research papers in National, International Journals and conferences . Her research interest lies in the area of Data Mining,Neural Networks.

M.Kannan has been working as a Assistant Professor and Ph.D Research Scholar in the Department of Computer Science & Applications in Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya University, Enathur, Kanchipuram –631 561. He received the degree in Master of Computer Applications from Bharathidasan University in 2001 and M.Phil(Computer Science) from Madurai Kamaraj University in 2005. His research interest includes Software Engineering & Data Mining.