International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 06 | Jun 2023 www.irjet.net p-ISSN: 2395-0072

Classification and Prediction Based Data Mining Algorithm in Weka Tool

Renu[1], Kanika[2]

---***---

Abstract-*Process of extract unseen and hidden information from large set of data is Data Mining. Different techniques and algorithm are used to get the meaningful information from the large set of data. Different classification algorithm are used just like J48, SMO, REP tree, Naïve Bayes, Multilayer perception to extract meaning information from large set of dataset. Predictive data mining that use historical data, statistical modeling, data mining technique and machine learning to make prediction about future outcomes. Predictive analytics used in different area to identify risks and opportunities. Weka tool are use to predict new data using classification and different classifier J48,SMO,REPTree,Naive Bayes, Multilayer Perception are classify with dataset and find accuracy of Multilayer perception is more efficient in accuracy.*

Keywords: Data mining, Weka tool, J48 algorithm classification, Naïve Bayes

1. Introduction

Huge amount of data is collected daily in this information era. Analyzing huge amount of data and extract information from that data is necessity to achieve goals. In data mining data cleaning, incorporating earlier knowledge on data set and interpreting perfect solution from the pragmatic results. Data mining[1] tool weka use to predict new data using selling house dataset. Efficiency of different classifier is calculated using confusion matrix and finds multilayer perception classifier has higher accuracy.

2. Related Technique in data mining

Different data mining techniques [3] to extract insights in data but type of data mining technique used depends on their data and goals. To extract information from data a wide variety of data mining technique are employed.

➢ Descriptive Modeling

- Clustering
- Association
- Sequential Analysis.

➢ Predictive Data mining Technique

- Classification
  1. Decision Tree
  2. Neural network.
  3. Rule Induction.
- Regression.
  - ➢ Prescriptive Modeling
  - ➢ Pattern Mining.
  - ➢ Anomaly Detection.

3. Methodology

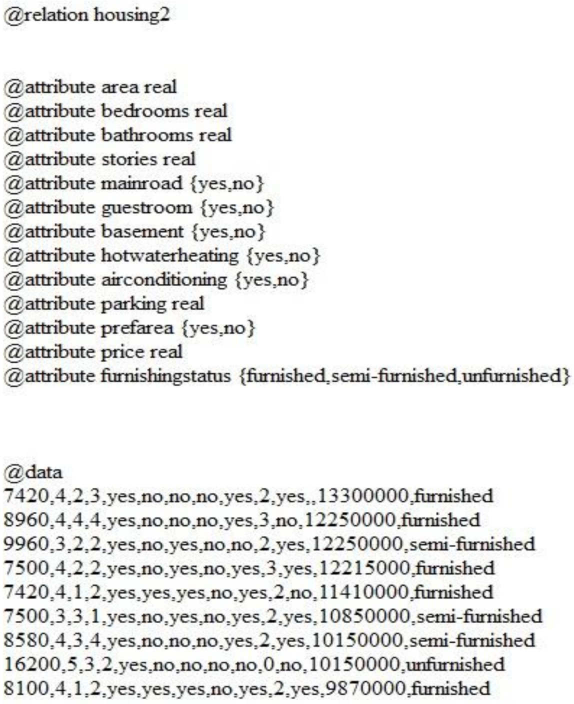
Weka contains a collection of classifier for data analysis with graphical user interface for easy access. Original non-Java version of weka was a Tel/TK front-end to modeling algorithms implemented in other programming languages plus data preprocessing utilities in C and a make file based system.Orignal version was design as a tool for analyzing data from agriculture domains. Weka3 java based version developed in 1997 is used in different application areas particularly for education purposes and research. Several standard data mining tasks data preprocessing, clustering, classification, regression, visualization and feature selection supported by weka.Input to weka is expected to be formatted according to the attributed relational file format.

Waikato Environment for Knowledge Analysis
Version 3.8.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

Figure 1 Weka Data Mining Tool

4. Collect Dataset and preprocessing

Collection of related items of related data accessed individually is dataset. Process of preparing the raw data and making it suitable for a machine learning model just like apply filter and convert file into arff, handling missing data etc is data preprocessing. Used data in the paper is collected from kaggle.com.



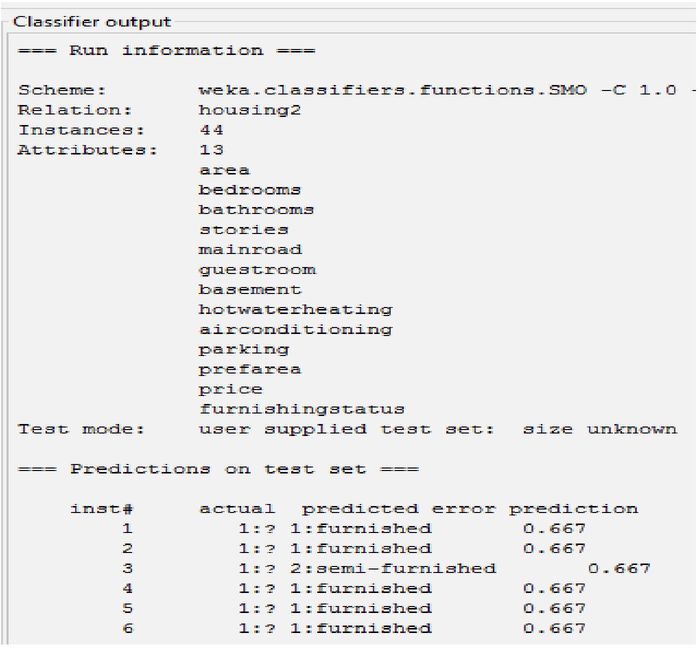
```
@relation housing2


@attribute area real
@attribute bedrooms real
@attribute bathrooms real
@attribute stories real
@attribute mainroad {yes,no}
@attribute guestroom {yes,no}
@attribute basement {yes,no}
@attribute hotwaterheating {yes,no}
@attribute airconditioning {yes,no}
@attribute parking real
@attribute prefarea {yes,no}
@attribute price real
@attribute furnishingstatus {furnished,semi-furnished,unfurnished}


@data
7420,4,2,3,yes,no,no,no,yes,2,yes,,13300000,furnished
8960,4,4,4,yes,no,no,no,yes,3,no,12250000,furnished
9960,3,2,2,yes,no,yes,no,no,2,yes,12250000,semi-furnished
7500,4,2,2,yes,no,yes,no,yes,3,yes,12215000,furnished
7420,4,1,2,yes,yes,yes,no,yes,2,no,11410000,furnished
7500,3,3,1,yes,no,yes,no,yes,2,yes,10850000,semi-furnished
8580,4,3,4,yes,no,no,no,yes,2,yes,10150000,semi-furnished
16200,5,3,2,yes,no,no,no,no,0,no,10150000,unfurnished
8100,4,1,2,yes,yes,yes,no,yes,2,yes,9870000,furnished
```

Figure 2 Dataset of house

5. Predict new data based on Dataset and Classifier

In prediction [4] use Dataset housing and classifier J48 by supplied

Training data as dataset and Supplied test data to predict unknown attribute.



```
=== Run information ===

Scheme:       weka.classifiers.functions.SMO -C 1.0 -
Relation:     housing2
Instances:    44
Attributes:   13
              area
              bedrooms
              bathrooms
              stories
              mainroad
              guestroom
              basement
              hotwaterheating
              airconditioning
              parking
              prefarea
              price
              furnishingstatus
Test mode:    user supplied test set:  size unknown

=== Predictions on test set ===

    inst#     actual  predicted error prediction
        1        1:? 1:furnished       0.667
        2        1:? 1:furnished       0.667
        3        1:? 2:semi-furnished       0.667
        4        1:? 1:furnished       0.667
        5        1:? 1:furnished       0.667
        6        1:? 1:furnished       0.667
```

Figure 3 Predict new data j48 Classifier

6. Performance evaluation

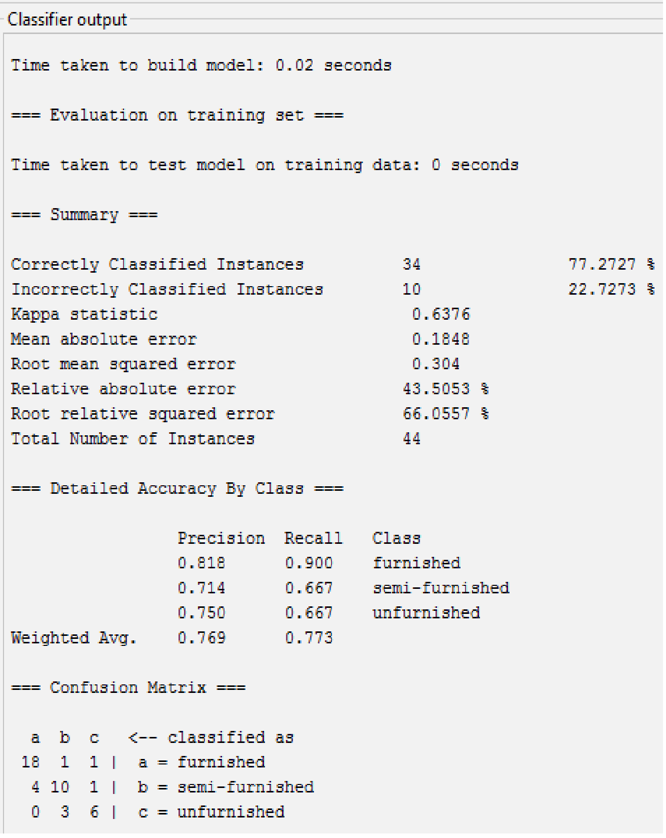
Different machine and deep learning measurement can be applied on the various classifier models. The measurements are Accuracy, Recall and Precision is the important criterion used to assess a model performance. The value of the confusion matrix which is generated during the testing of the model is considered to calculate those measurements. A confusion matrix is N*N matrix used for evaluating the performance of classification model. After classification confusion matrix compares the actual target values with predicted by the machine learning model. Confusion matrices give a better idea of a model performance.

Accuracy=Total correctly classified/Actual

Precision=Corrected predicted/Total predicted

Recall=correctly classified/Actual

6.1. Classifier J48



```
Time taken to build model: 0.02 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances          34               77.2727 %
Incorrectly Classified Instances        10               22.7273 %
Kappa statistic                          0.6376
Mean absolute error                      0.1848
Root mean squared error                  0.304
Relative absolute error                 43.5053 %
Root relative squared error             66.0557 %
Total Number of Instances               44

=== Detailed Accuracy By Class ===

                 Precision  Recall   Class
                 0.818      0.900    furnished
                 0.714      0.667    semi-furnished
                 0.750      0.667    unfurnished
Weighted Avg.    0.769      0.773

=== Confusion Matrix ===

  a  b  c   <-- classified as
 18  1  1 |  a = furnished
  4 10  1 |  b = semi-furnished
  0  3  6 |  c = unfurnished
```

Figure 4 Classifier J48

Accuracy, precision, recall of Classifier J48 using confusion matrix

Table 1 Confusion matrix J48

a	b	c	Total
18	1	1	20
4	10	1	15
0	3	6	9
22	14	8	44

Accuracy=Total correctly classified/Actual

= ((18+10+6)/44)*100=77.27%

Precision=Corrected predicted/Total predicted

A=18/22=0.818

B=10/14=0.714

C=6/8=0.75

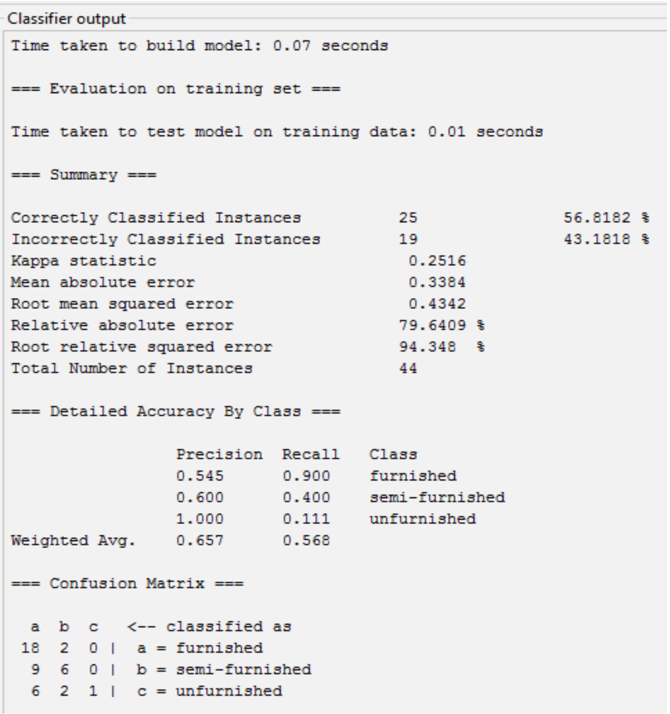
Recall=correctly classified/Actual

A=18/20=0.9

B=10/15=0.667

C=6/9=0.667

6.2. Classifier SMO



```
Classifier output
Time taken to build model: 0.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances          25               56.8182 %
Incorrectly Classified Instances        19               43.1818 %
Kappa statistic                          0.2516
Mean absolute error                      0.3384
Root mean squared error                  0.4342
Relative absolute error                 79.6409 %
Root relative squared error             94.348  %
Total Number of Instances               44

=== Detailed Accuracy By Class ===

                 Precision  Recall   Class
                 0.545      0.900    furnished
                 0.600      0.400    semi-furnished
                 1.000      0.111    unfurnished
Weighted Avg.    0.657      0.568

=== Confusion Matrix ===

  a  b  c   <-- classified as
 18  2  0 |  a = furnished
  9  6  0 |  b = semi-furnished
  6  2  1 |  c = unfurnished
```

Figure 5 Classifier SMO

Accuracy, precision, recall of Classifier SMO using confusion matrix

Table 2 Confusion Matrix SMO

a	b	c	Total
18	2	0	20
9	6	0	15
6	2	1	9
33	10	1	44

Accuracy=Total correctly classified/Actual

= ((18+6+1)/44)*100 =56.81%

Precision=Corrected predicted/Total predicted

A=18/33=0.545

B=6/10=0.6

C=1/1=1

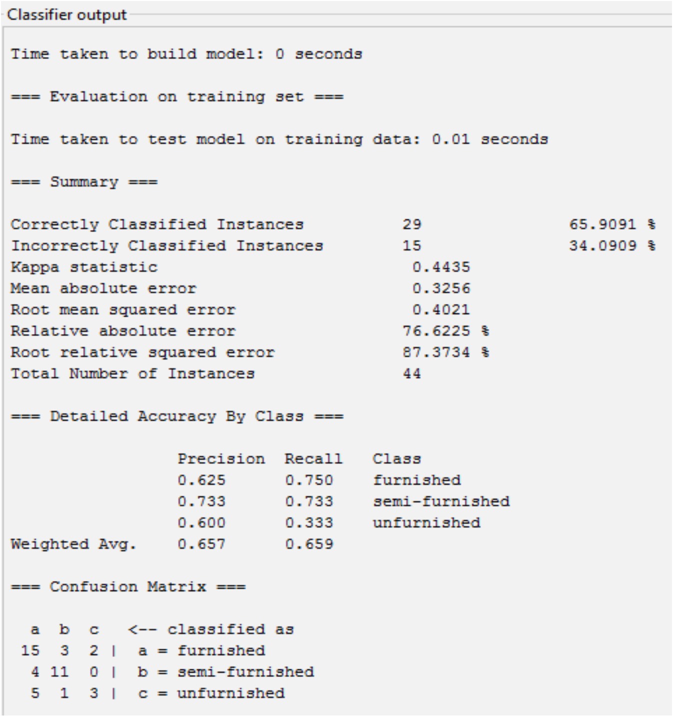
Recall=correctly classified/Actual

A=18/20=0.9

B=6/15=0.4

C=1/9=0.1

6.3. Classifier Naïve Bayes



```
Classifier output
Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances          29               65.9091 %
Incorrectly Classified Instances        15               34.0909 %
Kappa statistic                          0.4435
Mean absolute error                      0.3256
Root mean squared error                  0.4021
Relative absolute error                 76.6225 %
Root relative squared error             87.3734 %
Total Number of Instances               44

=== Detailed Accuracy By Class ===

                 Precision  Recall   Class
                 0.625      0.750    furnished
                 0.733      0.733    semi-furnished
                 0.600      0.333    unfurnished
Weighted Avg.    0.657      0.659

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  3  2 |  a = furnished
  4 11  0 |  b = semi-furnished
  5  1  3 |  c = unfurnished
```

Figure 6 Classifier Naive Bayes

Accuracy, precision, recall of Classifier Naïve Bayes using confusion matrix

Table 3 Confusion Matrix Naive Bayes

a	b	c	Total
15	3	2	20
4	11	0	15
5	1	3	9
24	15	5	44

Accuracy=Total correctly classified/Actual

= ((15+11+3)/44)*100

=65.90%

Precision=Corrected predicted/Total predicted

A=15/24 =0.625

B=11/15 =0.733

C=3/5 =0.6

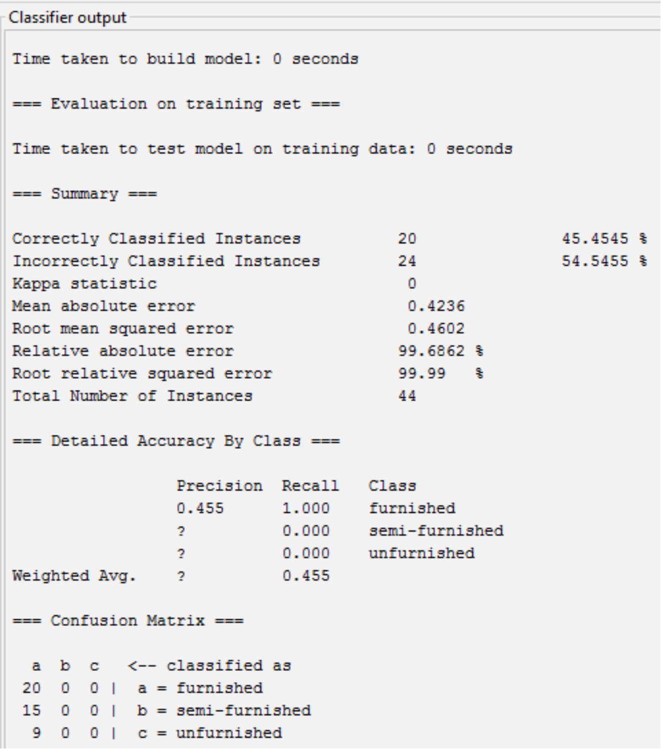
Recall=correctly classified/Actual

A=15/20 =0.75

B=11/15 =0.733

C=3/9=0.33

6.4. Classifier REPTree



```
Classifier output

Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances          20               45.4545 %
Incorrectly Classified Instances        24               54.5455 %
Kappa statistic                          0
Mean absolute error                      0.4236
Root mean squared error                  0.4602
Relative absolute error                 99.6862 %
Root relative squared error             99.99   %
Total Number of Instances               44

=== Detailed Accuracy By Class ===

                 Precision  Recall   Class
                 0.455      1.000    furnished
                 ?          0.000    semi-furnished
                 ?          0.000    unfurnished
Weighted Avg.    ?          0.455

=== Confusion Matrix ===

  a  b  c   <-- classified as
 20  0  0 |  a = furnished
 15  0  0 |  b = semi-furnished
  9  0  0 |  c = unfurnished
```

Figure 7 classifier REPTree

Accuracy, precision, recall of Classifier REPTree using confusion matrix

Table 4 Confusion Matrix REPTree

a	b	c	Total
20	0	0	20
15	0	0	15
9	0	0	9
44	0	0	44

Accuracy=Total correctly classified/Actual

= ((20+0+0)/44)*100=45.45%

Precision=Corrected predicted/Total predicted

A=20/44=0.455

B=0/0

C=0/0

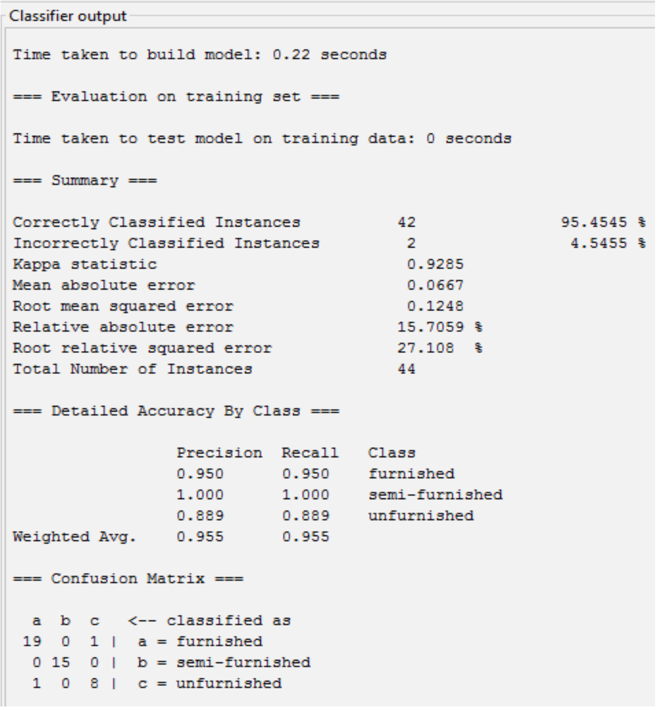
Recall=correctly classified/Actual

A=20/20 =1

B=0/15 =0

C=0/9=0

6.5. Classifier Multilayer perception



```
Classifier output

Time taken to build model: 0.22 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances          42               95.4545 %
Incorrectly Classified Instances         2                4.5455 %
Kappa statistic                          0.9285
Mean absolute error                      0.0667
Root mean squared error                  0.1248
Relative absolute error                 15.7059 %
Root relative squared error             27.108  %
Total Number of Instances               44

=== Detailed Accuracy By Class ===

                 Precision  Recall   Class
                 0.950      0.950    furnished
                 1.000      1.000    semi-furnished
                 0.889      0.889    unfurnished
Weighted Avg.    0.955      0.955

=== Confusion Matrix ===

  a  b  c   <-- classified as
 19  0  1 |  a = furnished
  0 15  0 |  b = semi-furnished
  1  0  8 |  c = unfurnished
```

Figure 8 Classifier Multilayer Perception

Accuracy, precision, recall of Classifier Multilayer perception using confusion matrix

Table 5 Confusion Matrix Multilayer Perception

a	b	c	Total
19	0	1	20
0	15	0	15
1	0	8	9
20	15	9	44

Accuracy=Total correctly classified/Actual

= ((19+15+8)/44)*100=95.45%

Precision=Corrected predicted/Total predicted

A=19/20=0.95

B=15/15 =1

C=8/9=0.88

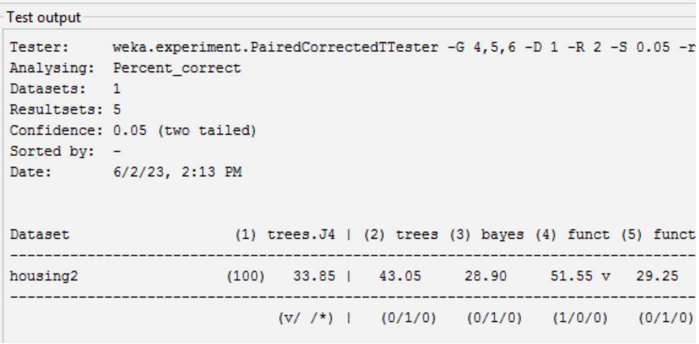
Recall=correctly classified/Actual

A=19/2 =0.95

B=15/15 =1

C=8/9 =0.88

6.6. Different Classifier Analysis



```
Test output
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -r
Analysing:  Percent_correct
Datasets:   1
Resultsets: 5
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       6/2/23, 2:13 PM


Dataset                   (1) trees.J4 | (2) trees (3) bayes (4) funct (5) funct
----------------------------------------------------------------------------------
housing2                 (100)   33.85 |   43.05     28.90     51.55 v   29.25
----------------------------------------------------------------------------------
                               (v/ /*) |   (0/1/0)   (0/1/0)   (1/0/0)   (0/1/0)
```

Figure 9 Different Classifiers Analysis

7. Accuracy of Different Classifier

The dataset is tested and analyze with classification algorithm [6] those are Multilayer perception, J48, Naïve Bayes, SMO, J48 and REPTree. Comparison of accuracy of all classifier is done it has been find that Multilayer Perception classifier perform best with accuracy. Accuracy is metric for evaluating classification models.

To increase the accuracy of model various method are used. Easiest way to improve the accuracy of model is to handle missing values. These some methods are to increase accuracy

- Acquire more data.
- Missing value treatment.
- Outlier treatment.
- Feature Engineering.
- Applying different model.
- Cross validation.
- Ensembling methods.
- Hyperparameter tuning.

Table 6 Different classifier Accuracy

Classifier	**Accuracy**
Multilayer Perception	95.45%
J48	77.27%
Naïve Bayes	65.90%
SMO	56.81%
Reptree	45.45%

As above Figure10 show that accuracy of Multilayer Perception classifier is high that is 95.45% as compare to the other classifier.

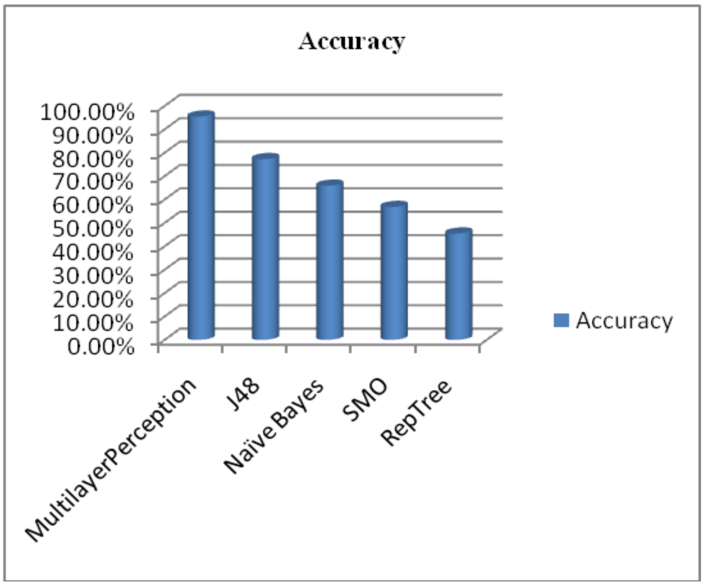


Figure 10 Accuracy of Classifier

Conclusion

In this paper classification technique J48 is used to predict the data using housing dataset and also analysis the various classifiers and find that multilayer perception perform best with high accuracy.Weka data mining tool is easy to understand and interfaced with various technique. Hence future of data mining is promising for further research and can be applied in different areas due to the availability of huge databases.

References

[1] https://en.wikipedia.org/wiki/Data_mining

[2] https://medium.com

[3] Jiawei Han Michelin Kamber,"Data Mining Concepts and Techniques", Morgan Kaufmann Publishers

[4] M.Ramaswami and R.Bhaskaran,"A CHAID Based performance prediction model in educational data mining,"Journal of computer science Issues

[5] Mansi Gera Shivani goel,"Data mining techniques methods and algorithms

[6] A Michal,"IPM developers works: IBM resource for developers and IT"