

**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING**

**KATHMANDU ENGINEERING COLLEGE
KALIMATI, KATHMANDU**

DEPARTMENT OF COMPUTER ENGINEERING



**REPORT ON
ANALYSIS OF SAMPLE DATASET USING
CLASSIFICATION**

SUBMITTED BY:

MAUSAM GURUNG

(KAT076BEI014)

SUBMITTED TO:

ER. SHARAD CHANDRA JOSHI

DATA MINING

KATHMANDU, NEPAL

2080

INTRODUCTION

Data mining is defined as a process used to extract usable data from a larger set of raw data. It implies analyzing data patterns in large batches of data using one or more software. Data mining has applications in multiple fields, like science and research. As an application of data mining, businesses can learn more about their customers and develop more effective strategies related to various business functions and in turn leverage resources more optimally and insightfully. This helps businesses be closer to their objectives and make better decisions. Data mining involves effective data collection and warehousing as well as computer processing. For segmenting the data and evaluating the probability of future events, data mining uses sophisticated mathematical algorithms. Data mining is also known as Knowledge Discovery in Data (KDD).

This lab illustrates the basic data pre-processing that is done using WEKA.

CLASSIFICATION

Classification is a data analysis task, i.e., the process of finding a model that describes and distinguishes data classes and concepts. Classification is the problem of identifying to which of a set of categories (subpopulations), a new observation belongs, based on a training set of data containing observations and whose categories membership is known.

DESCRIPTION OF ROUGH DATABASE

The database we have used consists of more than 1300 datasets with 12 attributes. Each attribute further has its own values. There are some datasets with either null value or value written as nothing. Also, each of the values of attributes are represented by numbers like 0 for no and 1 for yes. During ARFF file generation, we either use their nominal values or real values.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	cast	1. bahun	2. chhetri	3. gurun	4. newar	5. tamang	6. limbhu	7. magar	8. muslim	9. sherpa	10. rai	11. damai	12. bishworkarma	14. sarki	
2															
3															
4															
5	cast	type	health Check (0: no, 1: yes)	child death (0: no, 1: yes)	smoke male (2: no, 1: yes)	smoke female (2: no, 1: yes)	special chulo (0: no, 1: yes)	self home (0: no, 1: yes)	toilet (0: no, 1: yes)	animal (0: no, 1: yes)	self field value in ropani	drinking water filter 0: don't filter, 1: filter			
6															
7	2	3	0	0			0	1	1	1	15	1			
8	2	1	0	0	1	1	0	1	1	1	1	1			
9	11	1	0	0			0	1	1		16	0			
10	2	1	nothing	nothing			nothing	nothing	nothing	nothing		nothing			
11	2	1	0	0			1	1	0	1	3	0			
12	2	3	0	0			1	1	1	1	2	1			
13	2	3	0	0	1		1	1	1	1	6	0			
14	2	1	0	0		2	0	1	1	1	4	1			
15	2	3	1	0			0	1	1	1	8	0			
16	2	2	nothing	0	1	1	0	1	1	1	8	0			
17	13	1	nothing	1			1	1	1	1	9	0			
18	2	1	nothing	0	1		0	0	0	1	3	0			
19	2	1	nothing	0			0	1	0	0	1	0			
20	2	1	nothing	0	1	1	0	1	1	1	3	0			
21	2	3	nothing	0	1		0	1	0	1	1	0			
22	2	1	nothing	0		1	1	1	1	1	7	0			
23	2	1	nothing	0	1		0	1	1	1		0			
24	2	4	nothing	0			0	1	nothing	1	3	nothing			
25	2	1	nothing	0		1	0	1	1	1	3.5	0			
26	2	3	nothing	0			0	1	1	1	2.5	0			
27	2	1	nothing	0			0	1	1	1	3	0			
28	2	1	nothing	0	1		0	1	1	1	6	0			
29	2	1	1	0			0	1	nothing	0	1	nothing			
30	2	1	nothing	0			0	1	1	1	1	1			
31	2	3	nothing	0	1		0	1	1	1	3	0			
32	2	1	1	0			1	1	1	1	3	0			
33	2	1	nothing	0			0	1	1	1	1	0			
34	2	1	nothing	0	1		1	1	1	1	0.5	1			
35	2	1	nothing	0	1		1	1	1	1	5	1			
36	2	3	nothing	nothing			1	1	1	1	2	0			
37	13	1	nothing	0	1		0	1	1	1	0.5	0			
38	14	1	nothing	0			0	1	1	1	4	0			
39	2	1	0	0	1		0	1	1	1		1			
40	2	1	0	0			0	1	0	1	6	0			
41	2	3	0	0			0	nothing	1	1	1	0			

KDD STEPS FOLLOWED IN DATABASE

The KDD steps we have used in the used database are as follows:

i. Data Selection:

- It is the first stage of KDD process in which we collect and select the data set or database required to work with.
- From the sets of more than 1300 datasets with 12 attributes each of which having their own value is used to select 100 datasets with 12 attributes and their respective values.

ii. Data Cleaning:

- This is the second stage of KDD where we try to eliminate all the defects such as: human errors, not available when collected, not entered due to misunderstanding by the stage of de-duplication, domain consistency, and disambiguation.
- In this step, we removed duplicate data and replaced null value with some value.

iii. Coding:

- Here, we converted attributes yes/no value to 1/0 in attributes like health check, animals, child death.

iv. Data Mining:

- It consists of different rules, techniques, and algorithms used for mining purpose.

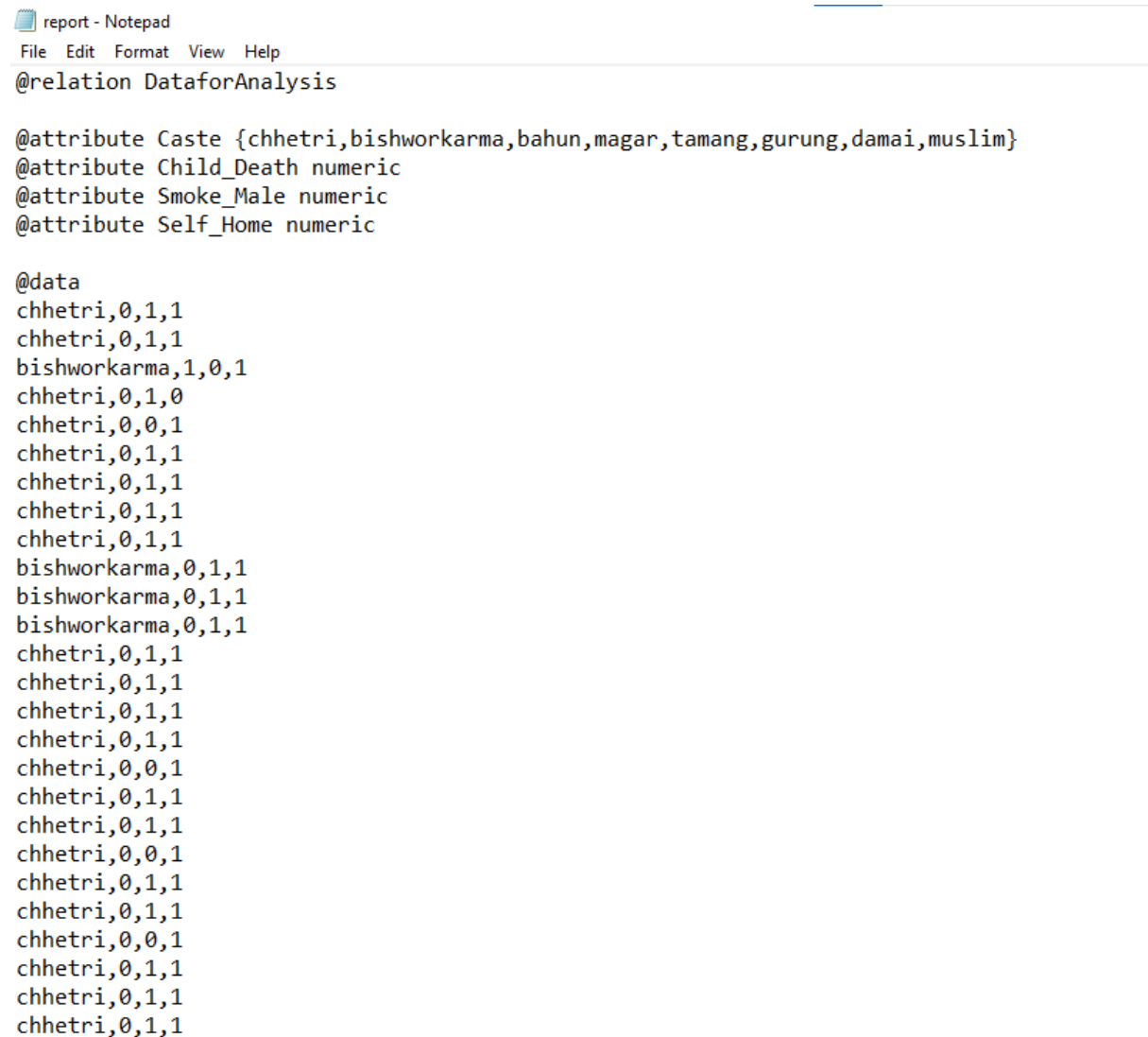
v. Reporting:

- This stage involves documenting the results obtained from learning algorithm.
- For this, we have analyzed result obtained from mining.

Now after data pre-processing we convert the file into arff format.

ARFF FILE DESCRIPTION

The converted file format is:



```
report - Notepad
File Edit Format View Help
@relation DataforAnalysis

@attribute Caste {chhetri,bishworkarma,bahun,magar,tamang,gurung,damai,muslim}
@attribute Child_Death numeric
@attribute Smoke_Male numeric
@attribute Self_Home numeric

@data
chhetri,0,1,1
chhetri,0,1,1
bishworkarma,1,0,1
chhetri,0,1,0
chhetri,0,0,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,1,1
bishworkarma,0,1,1
bishworkarma,0,1,1
bishworkarma,0,1,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,0,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,0,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,0,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,0,1
chhetri,0,1,1
chhetri,0,1,1
chhetri,0,1,1
```

In order to convert csv file into arff file we have to mention each attribute and its type after '@attribute'. '@relation' refers to the name of the folder of dataset. We have to mention the type of attribute after its name. Here all the attributes not numeric. '@data' refers to the beginning of the data needed for drawing conclusion using various classification or clustering algorithms.

ALGORITHM

Here, the algorithms used are two classification algorithm and two clustering algorithms. The classification algorithms used are Naive Bayes, Random Forest and clustering algorithm used are K-mean Clustering and Hierarchical clustering.

NAIVE BAYES:

In statistics, naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features (see Bayes classifier). They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve high accuracy levels. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. In the statistics literature, naïve Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naïve Bayes is not (necessarily) a Bayesian method. **Random Forest:**

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. It establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

K-MEANS CLUSTERING:

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. K-means clustering minimizes within-

cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

HIERARCHICAL CLUSTER:

A Hierarchical clustering method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes the subsequent steps:

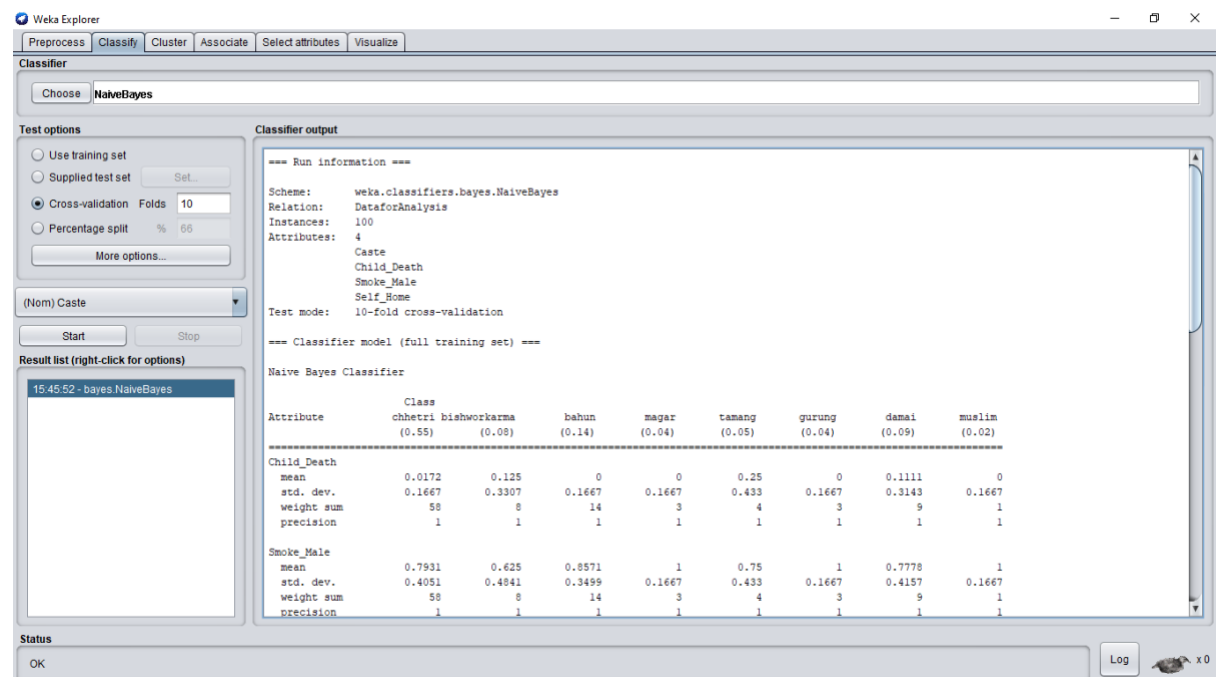
- Identify the 2 clusters which can be closest together
- Merge the 2 maximum comparable clusters.

We need to continue these steps until all the clusters are merged together. In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters.

ALGORITHM'S OUTPUT

• FOR CLASSIFICATION

1. NAÏVE BAYES



The screenshot shows the Weka Explorer interface with the Naive Bayes classifier selected. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' section displays the following information:

```

=== Run information ===

Scheme:      weka.classifiers.bayes.NaiveBayes
Relation:    DataforAnalysis
Instances:   100
Attributes:  4
  Caste
  Child_Death
  Smoke_Male
  Self_Home

Test mode:   10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Attribute    Class
              chhetri  bishworkarma  bahun  megar  tamang  gurung  damai  muslim
              (0.55)   (0.08)   (0.14)  (0.04)  (0.05)  (0.04)  (0.09)  (0.02)
-----
Child_Death
mean         0.0172   0.125    0       0       0.25    0       0.1111   0
std. dev.    0.1667   0.3307   0.1667  0.1667  0.433   0.1667  0.3143  0.1667
weight sum   58        8       14      3       4       3       9       1
precision    1         1       1       1       1       1       1       1

Smoke_Male
mean         0.7931   0.625    0.8571   1       0.75    1       0.7778   1
std. dev.    0.4051   0.4841   0.3459   0.1667  0.433   0.1667  0.4157  0.1667
weight sum   58        8       14      3       4       3       9       1
precision    1         1       1       1       1       1       1       1
  
```

The 'Result list' on the left shows '15:45:52 - bayes.NaiveBayes'.

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier

Choose **NaiveBayes**

Test options

☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds **10**
☐ Percentage split % **66**

(Nom) Caste

Result list (right-click for options)

15:45:52 - bayes.NaiveBayes

Classifier output

```

Self_Home
mean      0.931      0.875      0.9286      1      1      1      1      1
std. dev. 0.2534      0.3307      0.2575      0.1667      0.1667      0.1667      0.1667
weight sum 58      8      14      3      4      3      9      1
precision 1      1      1      1      1      1      1      1

Time taken to build model: 0 seconds

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

Correctly Classified Instances      56      56      %
Incorrectly Classified Instances    44      44      %
Kappa statistic                    0.0216
Mean absolute error                 0.164
Root mean squared error             0.2928
Relative absolute error             101.2599 %
Root relative squared error         104.3005 %
Total Number of Instances          100

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
0.966  0.905  0.596  0.966  0.737  0.126  0.474  0.552  chhetri
0.000  0.033  0.000  0.000  0.000  -0.052  0.211  0.054  bishworkarma
0.000  0.000  0.000  0.000  0.000  0.000  0.414  0.116  behun
0.000  0.000  0.000  0.000  0.000  0.000  0.330  0.028  magari
0.000  0.000  0.000  0.000  0.000  0.000  0.137  0.027  tamang
  
```

Status

OK x 0

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier

Choose **NaiveBayes**

Test options

☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds **10**
☐ Percentage split % **66**

(Nom) Caste

Result list (right-click for options)

15:45:52 - bayes.NaiveBayes

Classifier output

```

Root mean squared error            0.2928
Relative absolute error            101.2599 %
Root relative squared error        104.3005 %
Total Number of Instances          100

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
0.966  0.905  0.596  0.966  0.737  0.126  0.474  0.552  chhetri
0.000  0.033  0.000  0.000  0.000  -0.052  0.211  0.054  bishworkarma
0.000  0.000  0.000  0.000  0.000  0.000  0.414  0.116  behun
0.000  0.000  0.000  0.000  0.000  0.000  0.330  0.028  magari
0.000  0.000  0.000  0.000  0.000  0.000  0.137  0.027  tamang
0.000  0.000  0.000  0.000  0.000  0.000  0.352  0.028  gurung
0.000  0.033  0.000  0.000  0.000  -0.055  0.267  0.062  damai
0.000  0.000  0.000  0.000  0.000  0.000  0.040  0.010  muslim

Weighted Avg. 0.560  0.530  0.346  0.560  0.427  0.064  0.400  0.350

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  <-- classified as
56  1  0  0  0  0  1  0 | a = chhetri
 7  0  0  0  0  0  1  0 | b = bishworkarma
13  1  0  0  0  0  0  0 | c = bahun
 3  0  0  0  0  0  0  0 | d = magari
 3  0  0  0  0  1  0  0 | e = tamang
 3  0  0  0  0  0  0  0 | f = gurung
 8  1  0  0  0  0  0  0 | g = damai
 1  0  0  0  0  0  0  0 | h = muslim
  
```

Status

OK x 0

2. RANDOM FOREST

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) Caste

Start Stop

Result list (right-click for options)

15:45:52 - bayes.NaiveBayes

15:50:38 - trees.RandomForest

Classifier output

```
=== Run information ===
Scheme:      weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:    DateforAnalysis
Instances:    100
Attributes:   4
  Caste
  Child_Death
  Smoke_Male
  Self_Home
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      56          56 %
Incorrectly Classified Instances    44          44 %
Kappa statistic                    0.0101
Mean absolute error                 0.1554
Root mean squared error            0.286
Relative absolute error            95.97 %
```

Status

OK Log x 0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) Caste

Start Stop

Result list (right-click for options)

15:45:52 - bayes.NaiveBayes

15:50:38 - trees.RandomForest

Classifier output

```
Root mean squared error      0.286
Relative absolute error      95.97 %
Root relative squared error  101.8722 %
Total Number of Instances    100

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.966   0.929   0.589    0.966   0.732    0.084   0.523    0.640    chhetri
      0.000   0.000   0.000    0.000   0.000    0.000   0.147    0.050    bishworkarma
      0.000   0.012   0.000    0.000   0.000    -0.041   0.433    0.122    bahun
      0.000   0.000   0.000    0.000   0.000    0.000   0.337    0.028    magari
      0.000   0.010   0.000    0.000   0.000    -0.021   0.328    0.033    tamang
      0.000   0.000   0.000    0.000   0.000    0.000   0.359    0.029    gurung
      0.000   0.033   0.000    0.000   0.000    -0.055   0.357    0.071    damai
      0.000   0.000   0.000    0.000   0.000    0.000   0.141    0.010    muslim
Weighted Avg.    0.560   0.544   0.342    0.560   0.425    0.037   0.444    0.402

=== Confusion Matrix ===
  a  b  c  d  e  f  g  h  <-- classified as
56  0  1  0  0  0  1  0 | a = chhetri
 7  0  0  0  0  0  1  0 | b = bishworkarma
14  0  0  0  0  0  0  0 | c = bahun
 3  0  0  0  0  0  0  0 | d = magari
 3  0  0  0  0  0  1  0 | e = tamang
 3  0  0  0  0  0  0  0 | f = gurung
 8  0  0  1  0  0  0  0 | g = damai
 1  0  0  0  0  0  0  0 | h = muslim
```

Status

OK Log x 0

CONCLUSION OF ALGORITHM'S OUTPUT

CLASSIFICATION

1. NAÏVE BAYES

Correctly Classified Instances: 56(56%)

Incorrectly Classified Instances: 44(44%)

Kappa statistic: 0.0216

Mean absolute error: 0.164

Root mean squared error: 0.2928

Relative absolute error: 101.2599 %

Root relative squared error: 104.3005 %

Total Number of Instances :100

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	<-- classified as
56	1	0	0	0	0	0	1	0	a = chhetri
7	0	0	0	0	0	0	1	0	b = bishworkarma
13	1	0	0	0	0	0	0	0	c = bahun
3	0	0	0	0	0	0	0	0	d = magar
3	0	0	0	0	0	0	1	0	e = tamang
3	0	0	0	0	0	0	0	0	f = gurung
8	1	0	0	0	0	0	0	0	g = damai
1	0	0	0	0	0	0	0	0	h = muslim

2. RANDOM FOREST

Correctly Classified Instances: 56(56%)

Incorrectly Classified Instances: 44(44%)

Kappa statistic: 0.0101

Mean absolute error: 0.1554

Root mean squared error: 0.286

Relative absolute error: 95.97 %

Root relative squared error: 101.8722 %

Total Number of Instances 100

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	<-- classified as
56	0	1	0	0	0	0	1	0	a = chhetri
7	0	0	0	0	0	0	1	0	b = bishworkarma
14	0	0	0	0	0	0	0	0	c = bahun
3	0	0	0	0	0	0	0	0	d = magar
3	0	0	0	0	0	0	1	0	e = tamang
3	0	0	0	0	0	0	0	0	f = gurung
8	0	0	0	1	0	0	0	0	g = damai
1	0	0	0	0	0	0	0	0	h = muslim

DECISION FROM THE OUTPUT:

On observing above two classification algorithms we got to know that the mean absolute error of Random Forest is less than that of Naïve Bayes and the root mean square error of Random Forest is less than Naïve Bayes algorithm. The precision, recall, TP rate and so on is greater of Random Forest than Naïve Bayes so the best algorithm according to performance for sample dataset is Random Forest.

• FOR CLUSTERING

1) SIMPLE K MEANS

The screenshot shows the Weka GUI with the SimpleKMeans clustering algorithm selected. The 'Cluster mode' tab is active, and the 'Use training set' option is selected. The 'Store clusters for visualization' checkbox is checked. The 'Clusterer output' window displays the following information:

```

=== Run information ===

Scheme:      weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A
Relation:    DataforAnalysis
Instances:    100
Attributes:   4
              Caste
              Child_Death
              Smoke_Male
              Self_Home
Test mode:    evaluate on training data

=== Clustering model (full training set) ===

KMeans
=====

Number of iterations: 2
Within cluster sum of squared errors: 61.25531914893611

Initial starting points (random):

Cluster 0: chhetri,0,1,1
Cluster 1: chhetri,0,1,0

Missing values globally replaced with mean/mode
  
```


CLUSTERING

1. K-MEANS CLUSTERING

Clustered Instances

0	94 (94%)
1	6 (6%)

2. HIERARCHICAL CLUSTER

Clustered Instances

0	99 (99%)
1	1 (1%)

DECISION FROM THE OUTPUT

With the observation from the above two clustering algorithms their instances are mentioned above. We can say that from Hierarchical cluster k-means we are able to get cluster 99% in class1 i.e., chance of no child death whereas k-means shows 94% in class 1. Hence with all that information we get to know that hierarchical clustering is best among these two for the sample dataset.

CONCLUSION:

In this lab, we learned about how to preprocessing the given sample of data and how to convert them into arff format. We also learned how to use WEKA tool to use classification algorithms and clustering algorithms on arff format database. We also compare the classification and clustering algorithms and found out which one is better for our given sample.