Analysis of Machine Learning Algorithms using WEKA

Aaditya Desai
Ph.D. Scholar, NMIMS University.
TCET, Mumbai

Dr. Sunil Rai
Ph.D. Guide, NMIMS University.

ABSTRACT
The purpose of this paper is to conduct an experimental study of real world problems using the WEKA implementations of Machine Learning algorithms. It will mainly perform classification and comparison of relative performance of different algorithms under certain criteria.

General Terms
TreesJ48, TreesJ48graft, RandomTree, OneR, ZeroR, Decision Table, Naïve Bayes, Bayes Net, Naïve Bayes Simple, Bayes Updatable, Multilayer Perceptron, Logistic, RBF Network, Simple Logistic

Keywords
WEKA, Machine Learning

1. INTRODUCTION
WEKA is a collection of open source of many data mining and machine learning algorithms, including: pre-processing on data, classification, clustering, association rule extraction.[1] [2]
In this paper we have taken the real world problem of predicting whether it is going to rain or any other prediction of weather. Machine learning works on the concept of the way a human brain works the machine also uses logical steps to perform the decision or to predict an output.

2. Data Set
The Data Set consists of attributes related to weather conditions. These weather conditions are sunny, overcast and rainy. Temperature, humidity, windy will provide us the actual values to make a decision whether to play or not to play.

2.1 Description of attributes in the Data Set
Table 1.1 provides the description of the attributes in the data set. The selected attributes consists of discrete attribute type. Also Fig 1.1 shows the input format of the data set which is in ARFF form i.e. Attribute Relation File Format which is used as input to Weka.

outlook	temperatu	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

Table**: 1.1 Weather.csv file**

Code for Weather1.arff:

```
@relation weather

@relation outlook{sunny,overcast,rainy}@attribute temperature real

@attribute humidity real

@attribute windy{TRUE,FALSE}

@attribute play{yes,no}

@data

sunny,85,85,FALSE,no

sunny,80,90,TRUE,no

overcast,83,86,FALSE,yes

rainy,70,96,FALSE,yes

rainy,68,80,FALSE,yes

rainy,65,70,TRUE,no

overcast,64,65,TRUE,yes

sunny,72,95,FALSE,no

sunny,69,70,FALSE,yes

rainy,75,80,FALSE,yes

sunny,75,70,TRUE,yes

overcast,72,90,TRUE,yes

overcast,81,75,FALSE,yes

rainy,71,91,TRUE,no
```

3. Results and Discussion:

3.1 Implementation of Algorithms
Weka is chosen for implementation of algorithms. The objective of selecting this tool is to understand the basic concepts and also application of these algorithms in real time. Weka is helpful in learning the basic concepts of machine learning with different options and analyzes the output that is being produced.

Implementation Procedure Used in Weka:

The ARFF file is fed in to Weka and the classification algorithms are implemented as defined in the following steps:

a) In the Preprocess tab, Discretize filter is applied to discretize the attributes sunny, overcast and rainy.

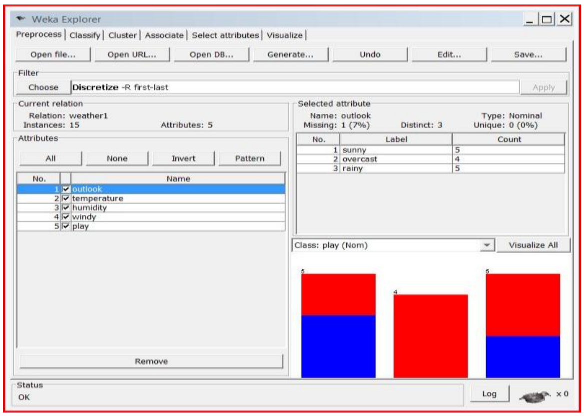

Fig. 1.2 a)Choosing filter

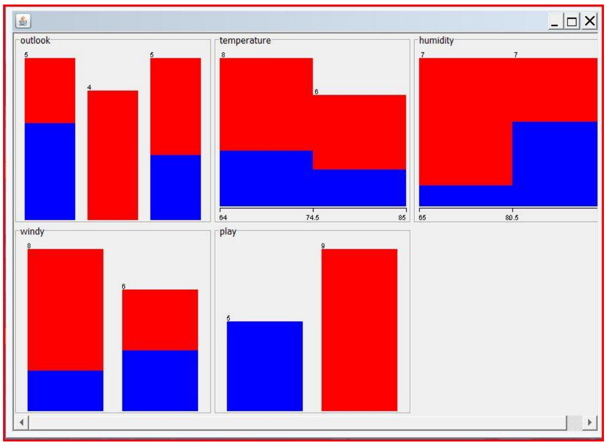

Fig.1.2 b) Output

b) In the Classify tab, choose the classification algorithm to be implemented and start the analysis to get results. In the WEKA experiment environment, the classifier at the top of the list in algorithms section is used as reference classifier. The table 1.2 below shows the classifiers that should be included. The *Attributes* field in the table states any change for the specified parameters of the classifier. Use 10-fold cross validation in the experiment.

Classifier	Attributes	Comments	Figure
TreesJ48	unprune=true	Without pruning, ID3 without pruning	1.3
TreesJ48graft	default	Uses pruning, C4.5 algorithm: an improvement over ID3.	1.4
RandomTree	default	Random Tree	1.5
OneR	default	Rules OneR	1.6
ZeroR	default	Rules ZeroR	1.7
Decision Table	default	Rules Decision Table	1.8
Naïve Bayes	default	Naïve Bayes	1.9
Bayes Net	default	Bayes Net	1.10
Naïve Bayes Simple	default	Naïve Bayes Simple	1.11
Bayes Updatable	default	Bayes Updatable	1.12
Multilayer Perceptron	default	Function	1.13
Logistic	default	Function Logistic	1.14
RBF Network	default	Function	1.15
Simple Logistic	default	Function Simple Logistic	1.16

3.1.1 Classifier Algorithm

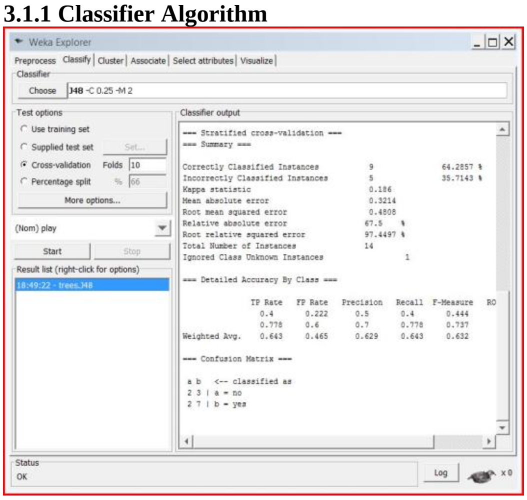

Fig. 1.3 J48

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 64.299%
Total number of instances = 14

3.1.2 Classifier Algorithm J48Graph

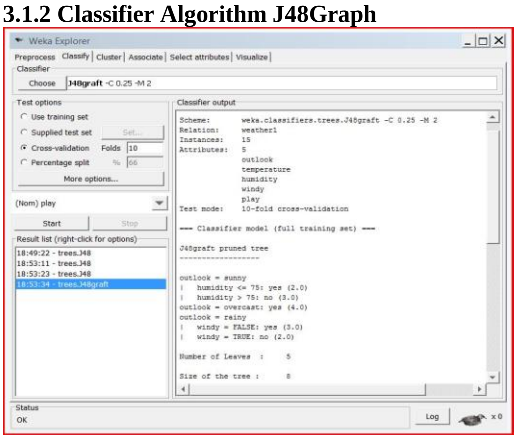

Fig. 1.4 J48graft

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 64.299%
Total number of instances = 14

3.1.3 Classifier Algorithm RandomTree

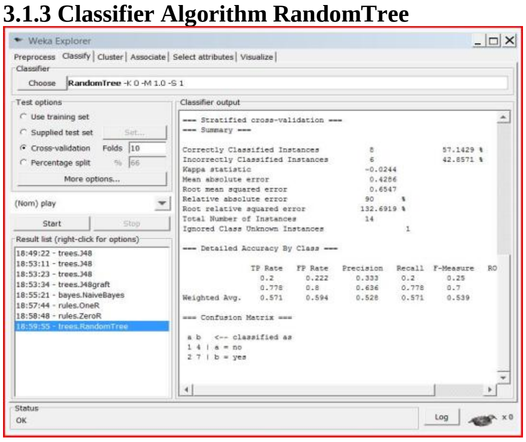

Fig.1.5 Random Tree

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 57.143%
Total number of instances = 14

3.1.4 Classifier Algorithm OneR

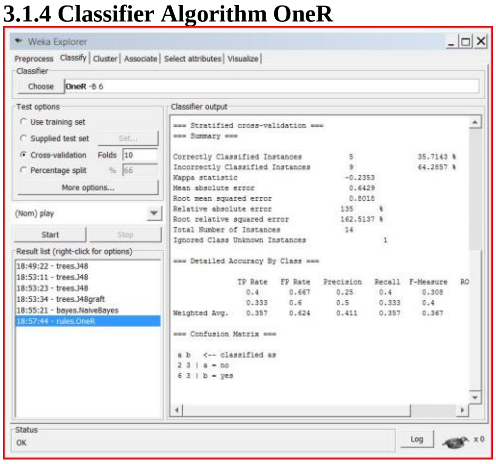

Fig. 1.6 One R

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 35.714%
Total number of instances = 14

3.1.5 Classifier Algorithm ZeroR

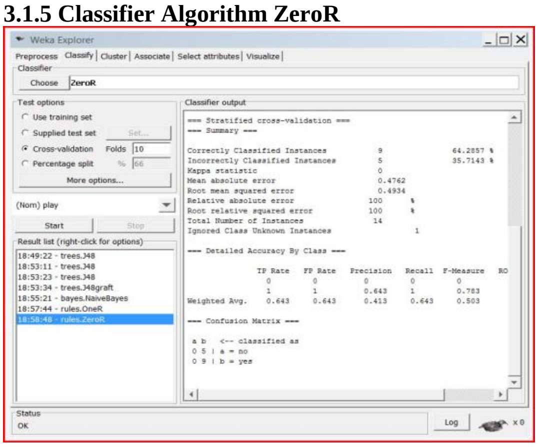

Fig. 1.7 ZeroR

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 64.286%
Total number of instances = 14

3.1.6 Classifier Algorithm Decision Table

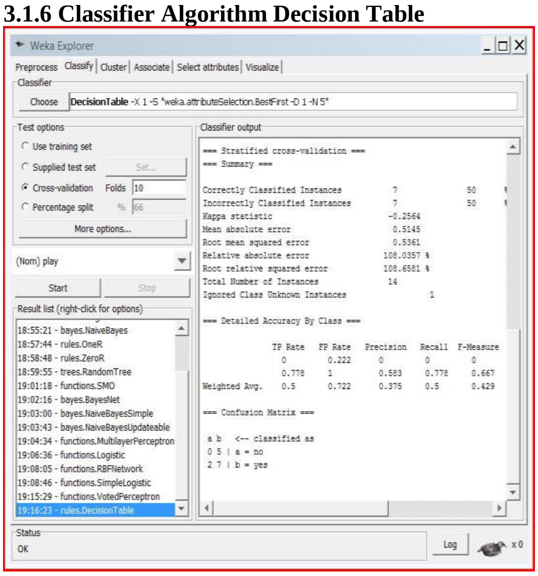

Fig. 1.8 Decision Table

Results:
Time taken to build the model = 0.01 seconds
Correctly classified instances (in %) = 50%
Total number of instances = 14

3.1.7 Classifier Algorithm Naïve Bayes

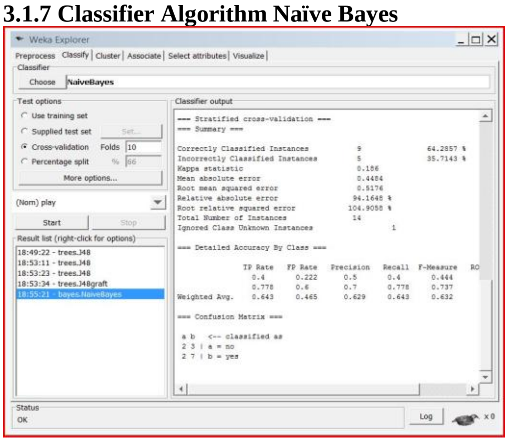

Fig 1.9 Naïve Bayes

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 64.286%
Total number of instances = 14

3.1.8 Classifier Algorithm Bayes Net

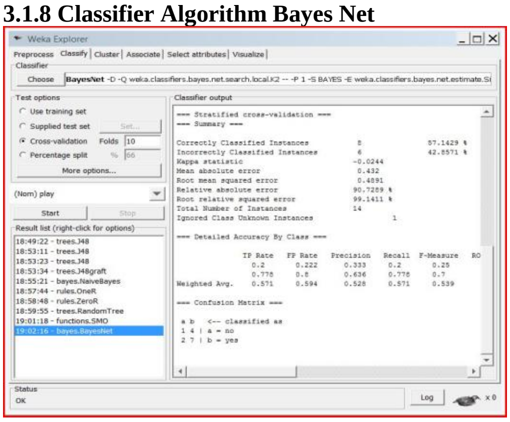

Fig.1.10 Bayes Net

Results:
Time taken to build the model = 0.01 seconds
Correctly classified instances (in %) = 57.143%
Total number of instances = 14

3.1.9 Classifier Algorithm Naïve Bayes Simple

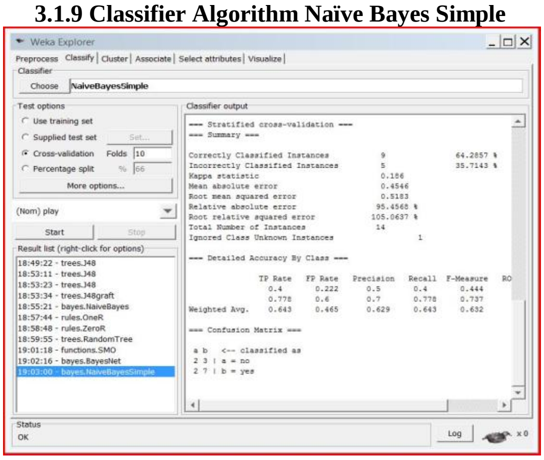

Fig 1.11 Naïve Bayes Simple

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 64.286%
Total number of instances = 14

3.1.10 Classifier Algorithm Naïve Bayes Updatable

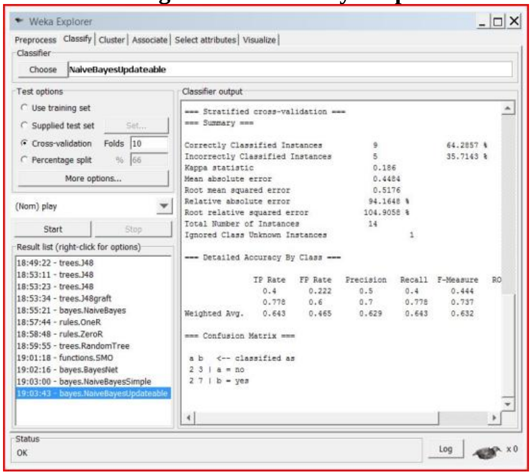

Fig. 1.12 Naïve Bayes Updatable

Results:
Time taken to build the model = 0 seconds
Correctly classified instances (in %) = 64.286%
Total number of instances = 14

3.1.11 Classifier Algorithm Multilayer Perceptron

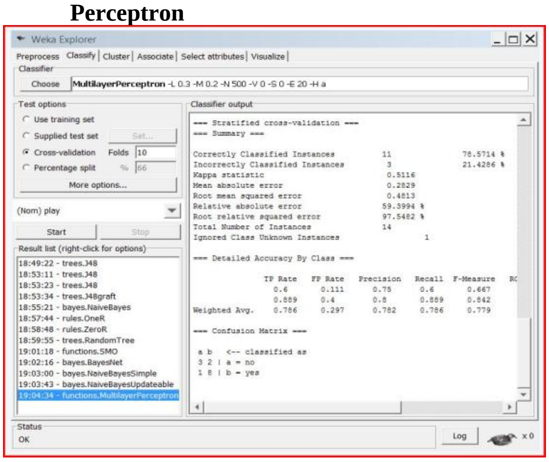

Fig. 1.13 Multilayer Perceptron

Results:
Time taken to build the model = 0.08 seconds
Correctly classified instances (in %) = 78.5714%
Total number of instances = 14

3.1.12 Classifier Algorithm Logistic

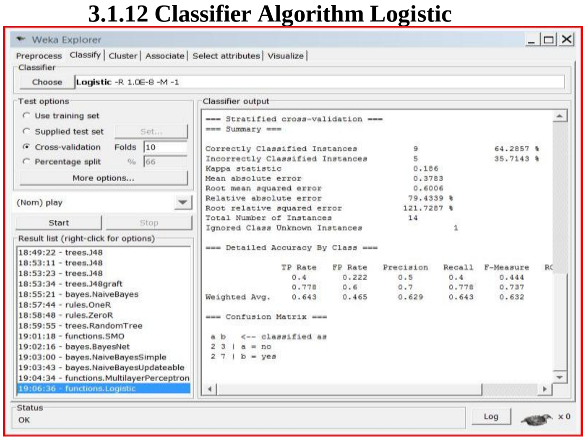

Fig. 1.14 Logistic

Results:
Time taken to build the model = 0.03 seconds
Correctly classified instances (in %) = 64.286%
Total number of instances = 14

3.1.13 Classifier Algorithm RBF Network

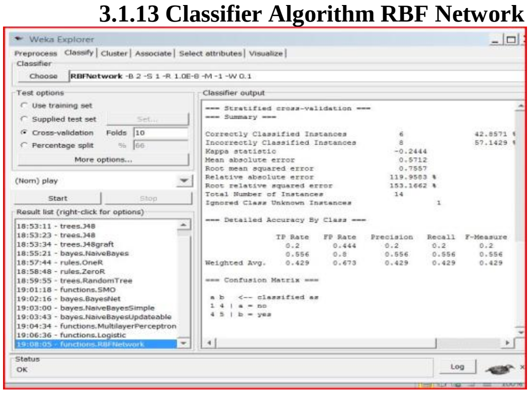

Fig. 1.15 RBF Network

Results:
Time taken to build the model = 0.03 seconds
Correctly classified instances (in %) = 42.857%
Total number of instances = 14

3.1.14 Classifier Algorithm Simple Logistic

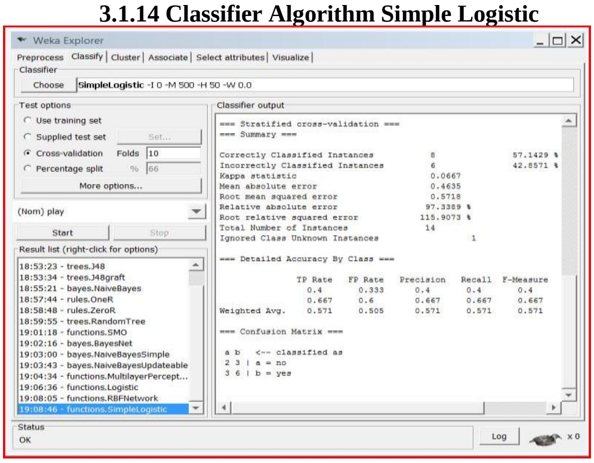

Fig. 1.16 Simple Logistic

Results:
Time taken to build the model = 0.08 seconds
Correctly classified instances (in %) = 57.143%
Total number of instances = 14

4. CONCLUSION:

A)Table 1.3 shows the comparison of time taken to build the model with different algorithms.

	TreesJ48	Trees J48graft	Random Tree	OneR	ZeroR	Decision Table	Naïve Bayes	BayesNet	Naïve Bayes Simple	Bayes Update able	Multilayer Perceptron	Logistic	RBF Network	Simple Logistic
Time taken to build the model(seconds)	0	0	0	0	0	0.01	0	0.01	0	0	0.08	0.03	0.03	0.08

Table 1.3 Time taken (in seconds) to build the model

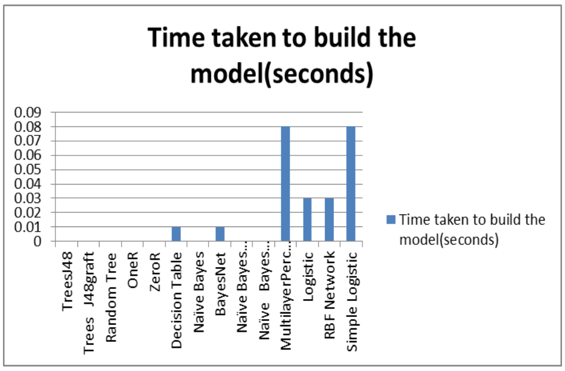

Fig. 4.1 Comparison of different algorithms with respect to time

Fig. 4.1 shows that Mulitlayer Perceptron and Simple Logistic takes 0.8 seconds to build the model whereas Logistic and RBF net take 0.3 seconds each. Decision table and Bayes Net take about 0.1 seconds each. Rest of the algorithms take 0 seconds to build.

B) Table 1.4 shows the performance of classifier filters based on the identification of correct instances

	Trees			Rules			Bayes				Functions			
	TreesJ48	Trees J48graft	Random Tree	OneR	ZeroR	Decision Table	Naïve Bayes	BayesNet	Naïve Bayes Simple	Bayes Update able	Multilayer Perceptron	Logistic	RBF Network	Simple Logistic
Identification of Current Instances (%)	64.299	64.299	57.143	35.714	64.286	50	64.286	57.143	64.286	64.286	78.5714	64.286	42.857	57.143

Table 1.4 Performance of classifiers based on identification of correct instances

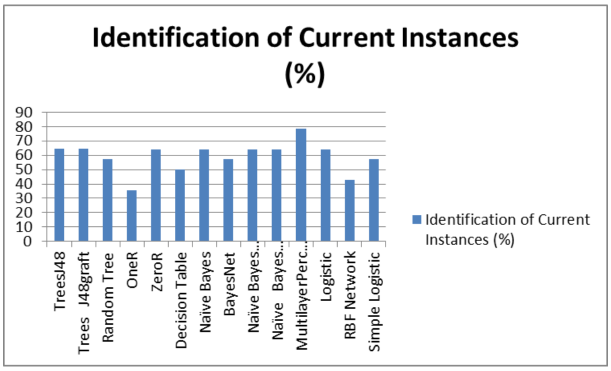

Fig 4.2 Comparison of identification of correct instances

Fig. 4.2 shows that Multilayer Perceptron has the highest identification of correct instances i.e. it is 78% for MLP and the least is that of OneR which is about 36%.

Now, if the comparison of classifier filters is done with respect to time taken to build model and identification of correct instances, then it is concluded that **Multilayer Perceptron** gives better results (time to build model = 0.08 sec, Identification of correct instances = 78.571%) compared to other classifier filters.

REFERENCES:

[1] Wikipedia.org

http://en.wikipedia.org/wiki/Weka_(machine_learning)

[2] WEKA Tutorial

http://www.cs.utexas.edu/users/ml/tutorials/Weka-tut/