Experiment No: 06	TE AI&DS
Date of Performance:	Roll No: 9696
Aim: : To analyze and evaluate the performance of different classification techniques using WEKA tool	
Related CO5: To analyze and evaluate perform of data mining techniques applied on large dataset using open-source tool for data mining	
Objective: To learn WEKA(Data Mining tool) and analyze and evaluate classification algorithm performance.	

Rubrics for assessment of Experiment:

Sr. No	**Parameters**	**Exceed Expectations(EE)**	**Meet Expectations (ME)**	**Below Expectations (BE)**
1	Timeline (2)	Early or on time (2)	One session late (1)	More than one session late (0)
2	Preparedness (2)	Knows the basic theory related to the experiment very well. (2)	Managed to explain the theory related to the experiment. (1)	Not aware of the theory to the point. (1)
3	Effort (3)	Done expt on their own. (3)	Done expt with help from other. (2)	Just managed. (1)
4	Documentation(2)	Lab experiment is documented in proper format and maintained neatly. (2)	Documented in proper format but some formatting guidelines are missed. (1)	Experiments not written in proper format (0.5)
5	Result (1)	Specific conclusion.(1)	Partially specific conclusion. (0.5)	Not specific at all. (0)

Assessment Marks:

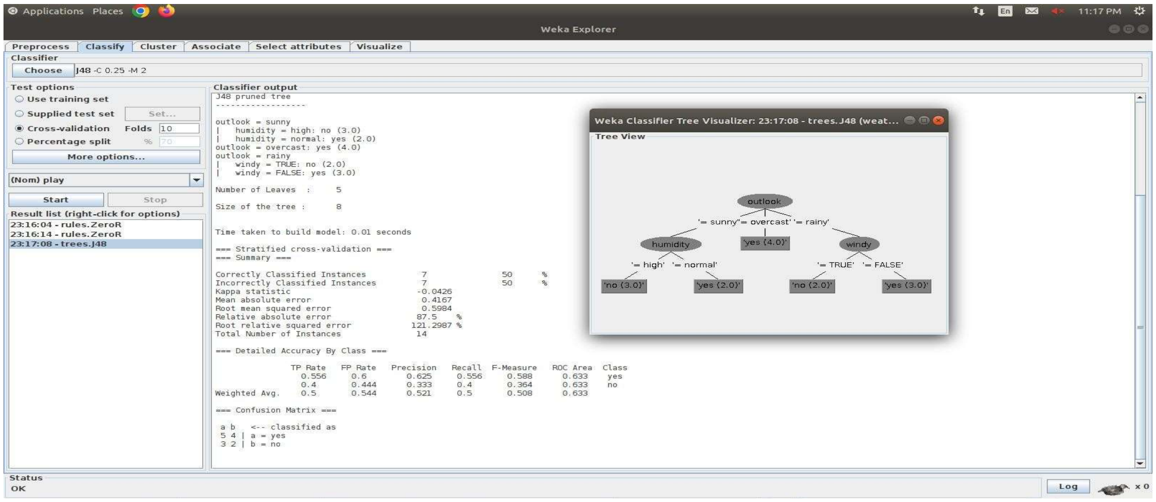
Timeline(2)	Preparedness(2)	Effort(3)	Documentation(2)	Result(1)	Total(10)

Theory:

WEKA is a data mining system developed by the University of Waikato in New Zealand that implements data mining algorithms. It is a collection of machine learning algorithms for data mining tasks. The algorithms are applied directly to a dataset.
WEKA implements algorithms for data preprocessing, classification, regression, clustering, association rules; it also includes a visualization tools. When 'WEKA GUI Chooser' window appears on the screen, we can select one of the four options.
The options are 1. Simple CLI : provides a simple command line interface and allows direct execution of WEKA commands. 2. Explorer : is an environment for exploring data. 3. Experimenter : is an environment for performing experiments and conducting statistical tests between learning schemes. 4. Knowledge Flow : is a Java-Beans-based interface for setting up and running machine learning experiments. Classifiers in WEKA are the models for predicting nominal or numeric quantities. The learning schemes available in WEKA include decision trees, Bayes net, neural network, support vector machine and so on. In this experiment analysis and evaluation of three classification algorithm: Naive Bayesian algorithm, C4.5 algorithm, zero R is done. Before running the classification algorithm it is required to set test options. The selected test options were : 1. Use training set : Evaluates the classifier on how well it predicts the class of the instances it was trained on. 2. Cross-validation : Evaluates the classifier by cross-validation, using the number of folds (10)
3. Percentage split : Evaluates the classifier on how well it predicts a certain percentage of the data, which is held out for testing. In the classifier evaluation options following options are checked : 1. Output model : The output is the classification model on the full training set. 2. Output per-class stats : The precision/recall and true/false statistics for each class output 3. Output confusion matrix : The confusion matrix of one classifier's prediction is included in the output When training set is complete, the 'classifier' output area on the right panel of the classifier window is filled with text describing the results of training and testing.

Practical Exercise:

Apply and evaluate the result for different classification techniques on various datasets using WEKA.



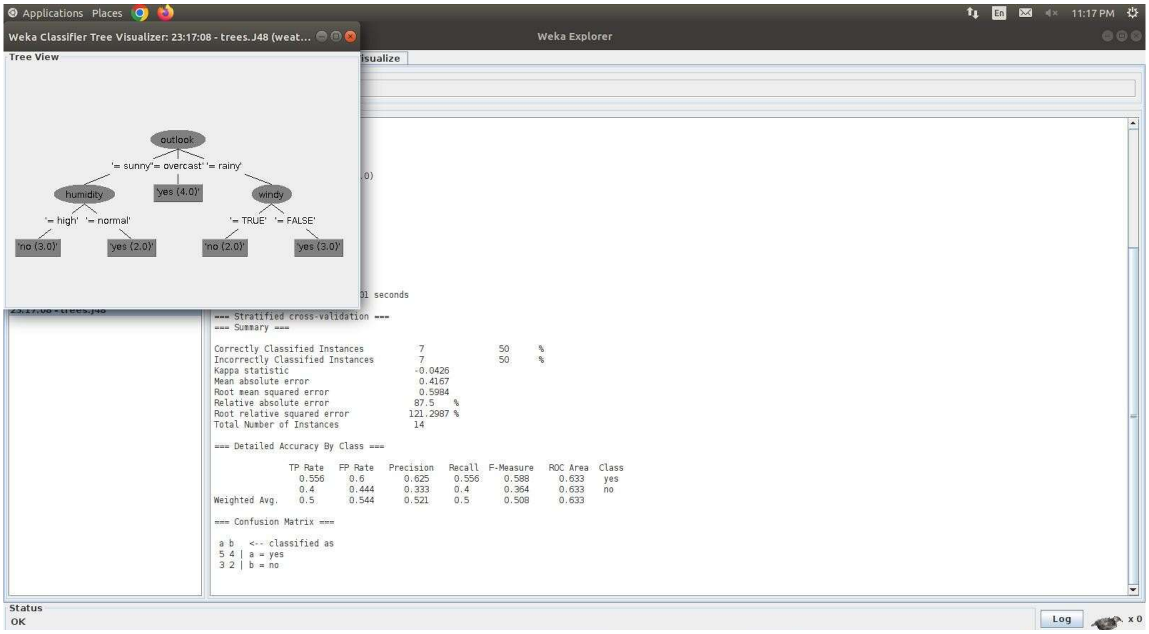
Classifier: J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 70

(Nom) play

23:16:04 - rules.ZeroR
23:16:14 - rules.ZeroR
23:17:08 - trees.J48

```
J48 pruned tree
------------------

outlook = sunny
|   humidity = high: no (3.0)
|   humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :      5

Size of the tree :       8


Time taken to build model: 0.01 seconds

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

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.556     0.6       0.625      0.556     0.588      0.633    yes
                 0.4       0.444     0.333      0.4       0.364      0.633    no
Weighted Avg.    0.5       0.544     0.521      0.5       0.508      0.633

=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```



```
.0)

01 seconds

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

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5     %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.556     0.6        0.625     0.556     0.588      0.633    yes
                 0.4       0.444      0.333     0.4       0.364      0.633    no
Weighted Avg.    0.5       0.544      0.521     0.5       0.508      0.633

=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```

Post lab Questions:

1. **Explain with example different measures to check the performance of a classifier.**

To check how well a classifier is doing, we use different measures:

<u>Accuracy:</u> It's the percentage of correct predictions. For instance, if it's 80%, it means the classifier gets things right 80% of the time.

<u>Precision:</u> This tells us how many of the positive predictions are actually correct. If it's 90%, it means 90% of the times the classifier says something is positive, it's right.

<u>Recall:</u> This is about finding out how many of the actual positive cases the classifier gets right. A 70% recall means the classifier misses 30% of the actual positive cases.

<u>F1 Score:</u> This is a combination of precision and recall. It's useful when we want to balance getting things right and not missing out on positive cases.

<u>AUC-ROC:</u> This measures how well the classifier can distinguish between good and bad cases.

For example, let's say we have a classifier that predicts if a customer will cancel their subscription. When we tested it, we found:

Accuracy: 80% (It's right 80% of the time).
Precision: 90% (When it predicts cancellations, it's right 90% of the time).
Recall: 70% (It misses 30% of actual cancellations).
F1 Score: 80% (A good balance of precision and recall).
AUC-ROC: 0.90 (It's good at telling good from bad).
Overall, this classifier seems pretty good, but it's not perfect. It's great at predicting cancellations,

but it does miss some actual cancellations, which could be due to various reasons, like the types of customers or the test data. So, while it's good, it's not perfect.

2. **Explain the result of any one classifier.**

Here's an illustration of what happens when a classifier is trained to determine if an email is spam:

The following metrics can be computed using the confusion matrix:

- Accuracy: (TP + TN) / (TP + TN + FP + FN) = 80%
- Precision: TP / (TP + FP) = 90%
- Recall: TP / (TP + FN) = 70%
- F1 score: 2 * (Precision * Recall) / (Precision + Recall) = 80%

These metrics suggest that the classifier is performing quite well, with an accuracy of 80%, precision of 90%, and recall of 70%. However, it is important to keep in mind thatthe recall is slightly lower, so the classifier may be missing some of the actual spam emails.

Choosing the right metric

The particular problem we are attempting to solve will determine which metric is most appropriate to use when assessing a classifier.

To ensure that you don't overlook any fraudulent transactions, for instance, having a high recall is crucial when developing a classifier for fraud detection.

To avoid misclassifying any legitimate emails as spam, it is more crucial to have a high precision when developing a classifier to filter spam emails.

It is also important to consider the class imbalance of your data. If the data is imbalanced, then some metrics, such as accuracy, can be misleading. In this case, it is better to use metrics that take into account the class imbalance, such as the F1 score or AUC-ROC.

Tuple	class	Prob	TP	FP	TPR	FPR
1	P	0.9.	1	0	0.2	0
2	P	0.8	2	0	0.4	0
3	N	0.7	2	1	0.4	0.2
4	P	0.6	3	1	0.6	0.2
5	P	0.55	4	1	0.8	0.2
6	N	0.54	4	2	0.8	0.4
7	N	0.53	4	3	0.8	0.6
8	N	0.51	4	4	0.8	0.8
9	P	0.5	5	4	1.0	0.8
10	N.	0.4.	5	5	1.0	1.0

TP = actual P predicted P

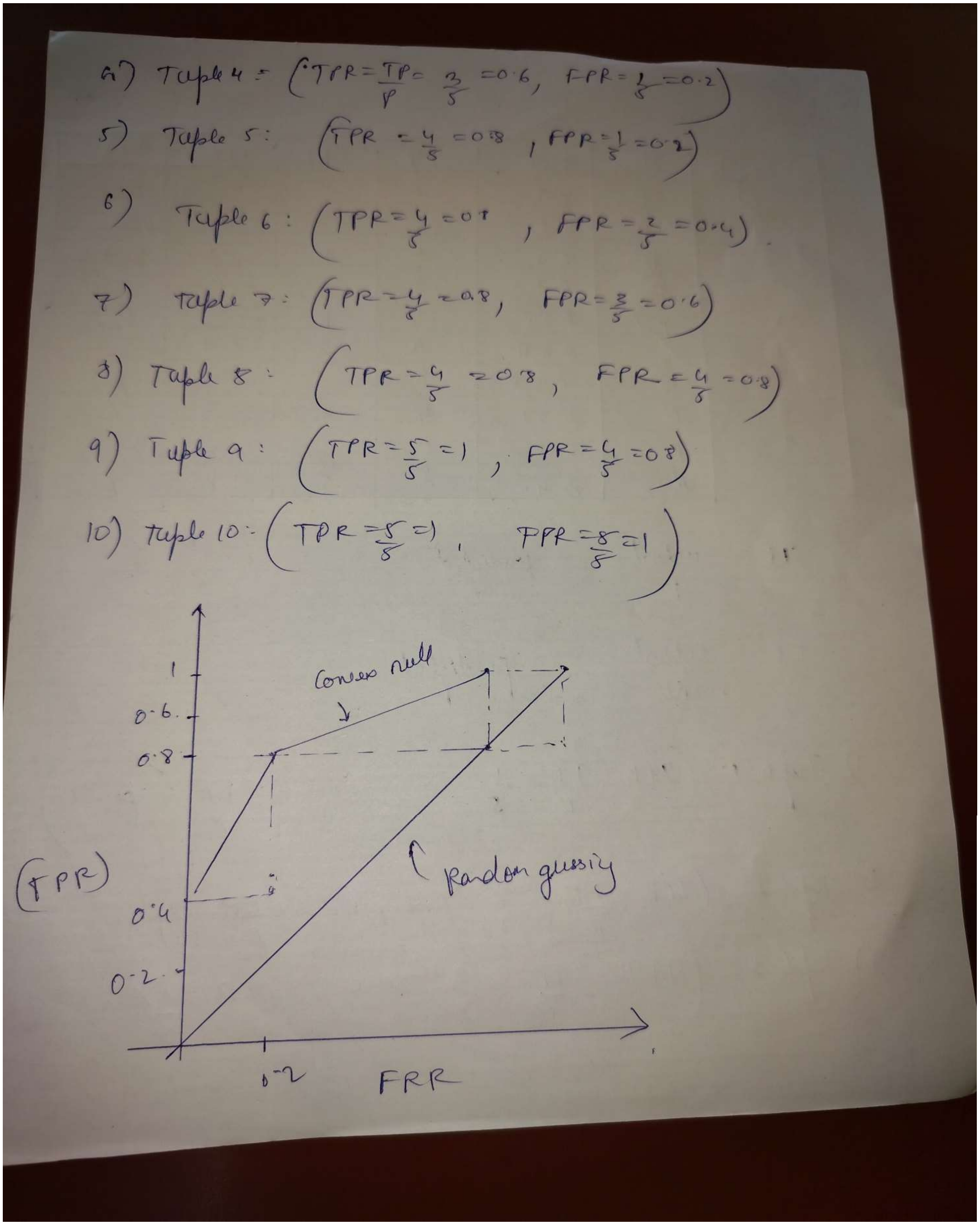
FP = actual N predicted P

1) Tuple 1: $\left(TPR = \frac{TP}{P} = \frac{1}{5} = 0.2 , FPR = \frac{FP}{P} = \frac{0}{5} = 0 \right)$

2) Tuple 2: $\left(TPR = \frac{TP}{P} = \frac{2}{5} = 0.4 , FPR = \frac{FP}{P} = 0 \right)$

3) Tuple 3: $\left(TPR = \frac{TP}{P} = \frac{2}{5} = 0.4 , FPR = \frac{FP}{P} = \frac{1}{5} = 0.2 \right)$

We learnt to analyze and evaluate the performance of different classification techniques using WEKA tool



4) Tuple 4 = ($TPR = \frac{TP}{P} = \frac{3}{5} = 0.6$, $FPR = \frac{1}{5} = 0.2$)

5) Tuple 5: ($TPR = \frac{4}{5} = 0.8$, $FPR = \frac{1}{5} = 0.2$)

6) Tuple 6: ($TPR = \frac{4}{5} = 0.8$, $FPR = \frac{2}{5} = 0.4$)

7) Tuple 7: ($TPR = \frac{4}{5} = 0.8$, $FPR = \frac{3}{5} = 0.6$)

8) Tuple 8: ($TPR = \frac{4}{5} = 0.8$, $FPR = \frac{4}{5} = 0.8$)

9) Tuple 9: ($TPR = \frac{5}{5} = 1$, $FPR = \frac{4}{5} = 0.8$)

10) Tuple 10: ($TPR = \frac{5}{5} = 1$, $FPR = \frac{5}{5} = 1$)

Convex hull
1
0.6.
0.8
(TPR)
0.4
0.2
Random guessing
0.2
FRR

Conclusion:

We learnt to analyze and evaluate the performance of different classification techniques using WEKA tool