Project 1.6

ZeroR Classifier:

I used the dataset which includes 99999 instances to run the ZeroR classifier. Firstly, I removed some attributes and remained “Patient-Sex, Age, Patient-Ethnicity, Region-ID, Severity”. Then I changed the type of “Patient-Ethnicity” and “Severity” to nominal manually (Patient-Ethnicity,{1,2,9}, Severity{1,2,3,4,5,6,7,8,9}). At last I used the “Discretize” in Weka to discretize the “Age” (preprocessing).

I chose 10 folds cross-validation as the test mode. The output is as follows:

=== Run information ===

Scheme: weka.classifiers.rules.ZeroR
Relation: train-weka.filters.unsupervised.attribute.Remove-R1-4,7,9-13,15-77,79
Instances: 99999
Attributes: 5
Patient-Sex
Age
Patient-Ethnicity
Region-ID
Severity
Test mode: 10-fold cross-validation

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

ZeroR predicts class value: 3

Time taken to build model: 0.08 seconds

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

Correctly Classified Instances	22030	22.0302 %
Incorrectly Classified Instances	77969	77.9698 %
Kappa statistic	0	
Mean absolute error	0.1876	
Root mean squared error	0.3063	
Relative absolute error	100	%
Root relative squared error	100	%
Total Number of Instances	99999	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0	0	0	0	0	0.5	1
	0	0	0	0	0	0.5	2
	1	1	0.22	1	0.361	0.5	3
	0	0	0	0	0	0.5	4
	0	0	0	0	0	0.5	5
	0	0	0	0	0	0.5	6
	0	0	0	0	0	0.5	7
	0	0	0	0	0	?	8
	0	0	0	0	0	0.488	9
Weighted Avg.	0.22	0.22	0.049	0.22	0.08	0.5	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	<-- classified as
0	0	11821	0	0	0	0	0	0 |	a = 1
0	0	11618	0	0	0	0	0	0 |	b = 2
0	0	22030	0	0	0	0	0	0 |	c = 3
0	0	16752	0	0	0	0	0	0 |	d = 4
0	0	16604	0	0	0	0	0	0 |	e = 5
0	0	13578	0	0	0	0	0	0 |	f = 6
0	0	7528	0	0	0	0	0	0 |	g = 7
0	0	0	0	0	0	0	0	0 |	h = 8
0	0	68	0	0	0	0	0	0 |	i = 9

Discussion:

ZeroR classifier has a bad performance on our dataset (correctly classified instances is just 22.03%). The reason is mainly because ZeroR algorithm considers the class label which has the most proportion in the dataset as the default classification result. However, in our dataset, the default classification result should not be the class which is frequently seen.

Decision Tree

I chose SimpleCart as the algorithm for the decision tree. In this part, I used 4 attributes “Patient-Sex, Age, Patient-Ethnicity, Region-ID, Severity”, the output is as follows:

=== Run information ===

Scheme: weka.classifiers.trees.SimpleCart -S 1 -M 2.0 -N 5 -C 1.0
Relation: train-weka.filters.unsupervised.attribute.Remove-R1-4,6-7,9-13,15-77,79
Instances: 99999
Attributes: 4

Patient-Sex
Patient-Ethnicity
Region-ID
Severity

Test mode: 10-fold cross-validation

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

CART Decision Tree
: 3(22030.0/77969.0)

Number of Leaf Nodes: 1

Size of the Tree: 1

Time taken to build model: 37.23 seconds

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

Correctly Classified Instances	22030	22.0302 %
Incorrectly Classified Instances	77969	77.9698 %
Kappa statistic	0	
Mean absolute error	0.1876	
Root mean squared error	0.3063	
Relative absolute error	99.9995 %	
Root relative squared error	100 %	
Total Number of Instances	99999	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0	0	0	0	0	0.5	1
	0	0	0	0	0	0.5	2
	1	1	0.22	1	0.361	0.5	3
	0	0	0	0	0	0.5	4
	0	0	0	0	0	0.5	5
	0	0	0	0	0	0.5	6
	0	0	0	0	0	0.5	7
	0	0	0	0	0	?	8
	0	0	0	0	0	0.488	9
Weighted Avg.	0.22	0.22	0.049	0.22	0.08	0.5	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	<-- classified as
0	0	11821	0	0	0	0	0	0 |	a = 1
0	0	11618	0	0	0	0	0	0 |	b = 2
0	0	22030	0	0	0	0	0	0 |	c = 3
0	0	16752	0	0	0	0	0	0 |	d = 4
0	0	16604	0	0	0	0	0	0 |	e = 5
0	0	13578	0	0	0	0	0	0 |	f = 6
0	0	7528	0	0	0	0	0	0 |	g = 7
0	0	0	0	0	0	0	0	0 |	h = 8
0	0	68	0	0	0	0	0	0 |	i = 9

Discussion:

SimpleCart algorithm doesn’t have a good performance on our dataset either. In my opinion, the main reason is this algorithm adopts the way of binary split to split the attributes during making the decision tree. However, in our dataset the attributes are nominal rather than binary. They have more than two values. That kind of splitting method may result in some errors when making classification.