Συστήματα Ανακάλυψης Γνώσης από Βάσεις Δεδομένων

Εργασία 03 – Classification και Clustering (με το WEKA)

Ονοματεπώνυμο: Παναγιώτης Γιαννουτάκης

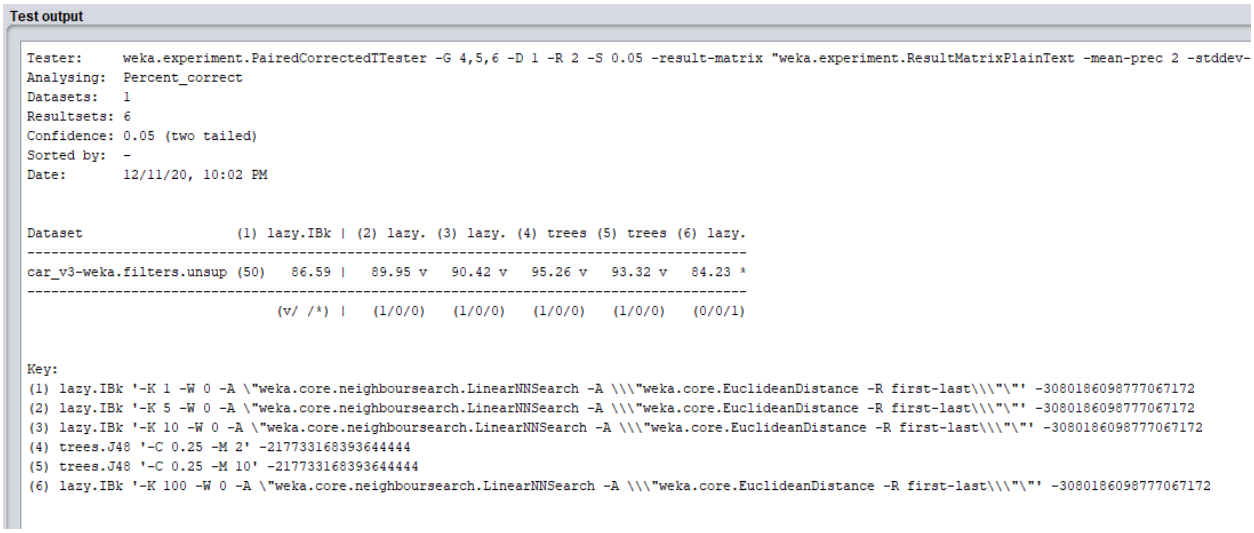
ΑΜ: 1238

Email: it1238@uom.edu.gr

Κατηγοριοποίηση

(1)

Τα αποτελέσματα που προέκυψαν μετά από το Paired T-Test φαίνονται στην παρακάτω εικόνα:



Test output

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-
Analysing:  Percent_correct
Datasets:   1
Resultsets: 6
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       12/11/20, 10:02 PM


Dataset                   (1) lazy.IBk | (2) lazy. (3) lazy. (4) trees (5) trees (6) lazy.
-----------------------------------------------------------------------------------------
car_v3-weka.filters.unsup (50)   86.59 |   89.95 v   90.42 v   95.26 v   93.32 v   84.23 *
-----------------------------------------------------------------------------------------
                            (v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (0/0/1)


Key:
(1) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(2) lazy.IBk '-K 5 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(3) lazy.IBk '-K 10 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(4) trees.J48 '-C 0.25 -M 2' -217733168393644444
(5) trees.J48 '-C 0.25 -M 10' -217733168393644444
(6) lazy.IBk '-K 100 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
```

```
     (1) lazy.IBk | (2) lazy. (3) lazy. (4) trees (5) trees (6) lazy.
-----------------------------------------------------------------------
ısup (50)   86.59 |   89.95 v   90.42 v   95.26 v   93.32 v   84.23 *
-----------------------------------------------------------------------
          (v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (0/0/1)
```

Βλέπουμε πως ο kNN με K=1 έχει 4 ήττες από τους πρώτους 4 αλγορίθμους και μια νίκη σε σχέση με τον τελευταίο αλγόριθμο. Το καλύτερο μοντέλο το βρίσκουμε όταν συγκρίνουμε τον αλγόριθμο IBk με τις Default ρυθμίσεις σε σχέση με τους υπόλοιπους. Όπως φαίνεται και από την παρακάτω εικόνα βλέπουμε πως επικρατεί όλων τον αλγορίθμων, έχει 5 νίκες.

```
  (4) trees.J4 | (1) lazy. (2) lazy. (3) lazy. (5) trees (6) lazy.
-----------------------------------------------------------------
p (50)   95.26 |   86.59 *   89.95 *   90.42 *   93.32 *   84.23 *
-----------------------------------------------------------------
       (v/ /*) |   (0/0/1)   (0/0/1)   (0/0/1)   (0/0/1)   (0/0/1)
```

(2)

Στην παρακάτω εικόνα βλέπουμε την ακρίβεια και το κόστος που πέτυχε το μοντέλο με τον αλγόριθμο kNN με k=1.

```
Correctly Classified Instances        1496               86.5741 %
Incorrectly Classified Instances       232               13.4259 %
Kappa statistic                          0.7129
Total Cost                             552
```

Στην κάτω εικόνα ο αλγόριθμος είναι πάλι ο kNN με k=100. Εδώ βλέπουμε πως παρόλο που το συνολικό κόστος είναι κατά πολύ μικρότερο σε σχέση με τον kNN με k=1, τα 69 αυτοκίνητα που θα έπρεπε να ανήκουν στην τρίτη γραμμή και τρίτη στήλη, κατανεμήθηκαν όλα σε λάθος στήλες.

```
Correctly Classified Instances        1461               84.5486 %
Incorrectly Classified Instances       267               15.4514 %
Kappa statistic                          0.6451
Total Cost                             165
```

```
=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1155   55    0    0 |    a = unacc
   81  303    0    0 |    b = acc
    3   66    0    0 |    c = good
    0   62    0    3 |    d = vgood
```

Στην εικόνα που ακολουθεί εφαρμόζεται στο μοντέλο μας ο αλγόριθμος J48.

```
Correctly Classified Instances        1664               96.2963 %
Incorrectly Classified Instances        64                3.7037 %
Kappa statistic                          0.9196
Total Cost                             150
```

Και τέλος στην τελευταία εικόνα φαίνονται τα αποτελέσματα που πήραμε με τον αλγόριθμο J48 και minNumObj=10.

```
Correctly Classified Instances        1618               93.6343 %
Incorrectly Classified Instances       110                6.3657 %
Kappa statistic                          0.8624
Total Cost                             264
```

Το συμπέρασμα που βγαίνει είναι ότι με τον αλγόριθμο J48 με τις default ρυθμίσεις πετυχαίνουμε την μεγαλύτερη ακρίβεια με το μικρότερο κόστος.

Συσταδοποίηση

(3)

Με τον αλγόριθμο kMeans και seed=10 οι συστάδες που δημιουργήθηκαν είναι οι εξής:

```
Clustered Instances

0        91 (  5%)
1       670 ( 39%)
2       557 ( 32%)
3       410 ( 24%)
```

Με seed=100 παίρνουμε τα εξής αποτελέσματα.

```
Clustered Instances

0       511 ( 30%)
1       384 ( 22%)
2       368 ( 21%)
3       465 ( 27%)
```