
**Ανακάλυψη Γνώσης σε Βάσεις Δεδομένων (ISE709)**

ΑΚΑΔ. ΕΤΟΣ 2024-25

**Διδάσκουσα: Γ. Κολωνιάρη**

# HW1

## Εξόρυξη Γνώσης σε Δεδομένα - Εργασία 1 (Classification)

Φοιτητής: Ερρίκος Ματεβοσιάν

ΑΜ: iis23018

# (A) Δέντρα Απόφασης με το χέρι

| Colour | Height | Stripes | Texture | Poisonous |
|---|---|---|---|---|
| Purple | Tall | Yes | Rough | Yes |
| Purple | Tall | Yes | Smooth | Yes |
| Red | Short | Yes | Hairy | No |
| Blue | Short | No | Smooth | No |
| Blue | Short | Yes | Hairy | Yes |
| Red | Tall | No | Hairy | No |
| Blue | Tall | Yes | Smooth | Yes |
| Blue | Short | Yes | Smooth | Yes |
| Blue | Tall | No | Hairy | No |
| Blue | Short | Yes | Rough | Yes |
| Red | Short | No | Smooth | No |
| Purple | Short | No | Hairy | Yes |
| Red | Tall | Yes | Hairy | No |
| Purple | Tall | Yes | Hairy | Yes |
| Purple | Tall | No | Rough | No |
| Purple | Tall | No | Smooth | No |

**i)**

Συνολικές τιμές = 16
Τιμή “Yes” στο χαρακτηριστικό “Poisonous” = 8
Τιμή “No” στο χαρακτηριστικό “Poisonous” = 8

**Gini Index** $$GINI(t) = 1 - \sum_{j=1}^{c} [p(j \mid t)]^2$$

**GiniSplit** $$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

- **Gini Index για Colour**

Χωρίζουμε σε Purple, Red, Blue

*Purple*

Έχουμε 4 Yes και 2 No στο χαρακτηριστικό Poisonous

$$\text{Gini(Purple)} = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 1 - 0.444 - 0.111 \cong 0.444$$

*Red*

Έχουμε 4 No στο χαρακτηριστικό Poisonous

$$\text{Gini(Red)} = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$$

*Blue*

Έχουμε 4 Yes και 2 No

$$\text{Gini(Blue)} = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 1 - 0.444 - 0.111 \cong 0.444$$

**Weighted Gini Index για Colour**

$$\text{Gini(Colour)} = \frac{6}{16} * 0.444 + \frac{4}{16} * 0 + \frac{6}{16} * 0.444 \cong 0.3334$$

- **Gini Index για Height**

Χωρίζουμε σε Tall και Short

Tall: έχουμε 5 Yes και 3 No στο χαρακτηριστικό Poisonous

$$\text{Gini(Tall)} = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2 \cong 0.468$$

Short: έχουμε 4 Yes και 4 No

$$\text{Gini(Short)} = 1 - \left(\frac{4}{8}\right)^2 - \left(\frac{4}{8}\right)^2 = 0.5$$

**Weighted Gini Index για Height**

$$\text{Gini(Height)} = \left(\frac{8}{16}\right) * 0.468 + \left(\frac{8}{16}\right) * 0.5 \cong 0.4844$$

- **Gini Index για Stripes**

Χωρίζουμε σε Yes και No

Yes: Έχουμε 6 Yes και 2 No

$$\text{Gini(Yes)} = 1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2 = 0.375$$

No: Έχουμε 2 Yes και 4 No

$$\text{Gini(No)} = 1 - \left(\frac{2}{6}\right)^2 - \left(\frac{4}{6}\right)^2 \cong 0.444$$

**Weighted Index για Stripes**

$$\text{Gini(Stripes)} = \left(\frac{8}{16}\right) * 0.375 + \left(\frac{6}{16}\right) * 0.444 \cong 0.3542$$

- **Gini Index για Texture**

Χωρίζουμε σε Rough, Smooth, Hairy

Rough: Έχουμε 3 Yes και 1 No

$$\text{Gini(Rough)} = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.375$$

Smooth: Έχουμε 3 Yes και 3 No

$$\text{Gini(Smooth)} = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

Hairy: Έχουμε 3 Yes και 3 No

$$\text{Gini(Hairy)} = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

**Weighted Index για Texture**

$$\text{Gini(Texture)} = \left(\frac{4}{16}\right) * 0.375 + \left(\frac{6}{16}\right) * 0.5 + \left(\frac{6}{16}\right) * 0.5 \cong 0.4688$$

**Το Colour έχει το μικρότερο Gini Index οπότε το επιλέγουμε ως ρίζα**

Ας δούμε το πιο εμφανιζόμενη τιμή του Poisonous για κάθε Colour

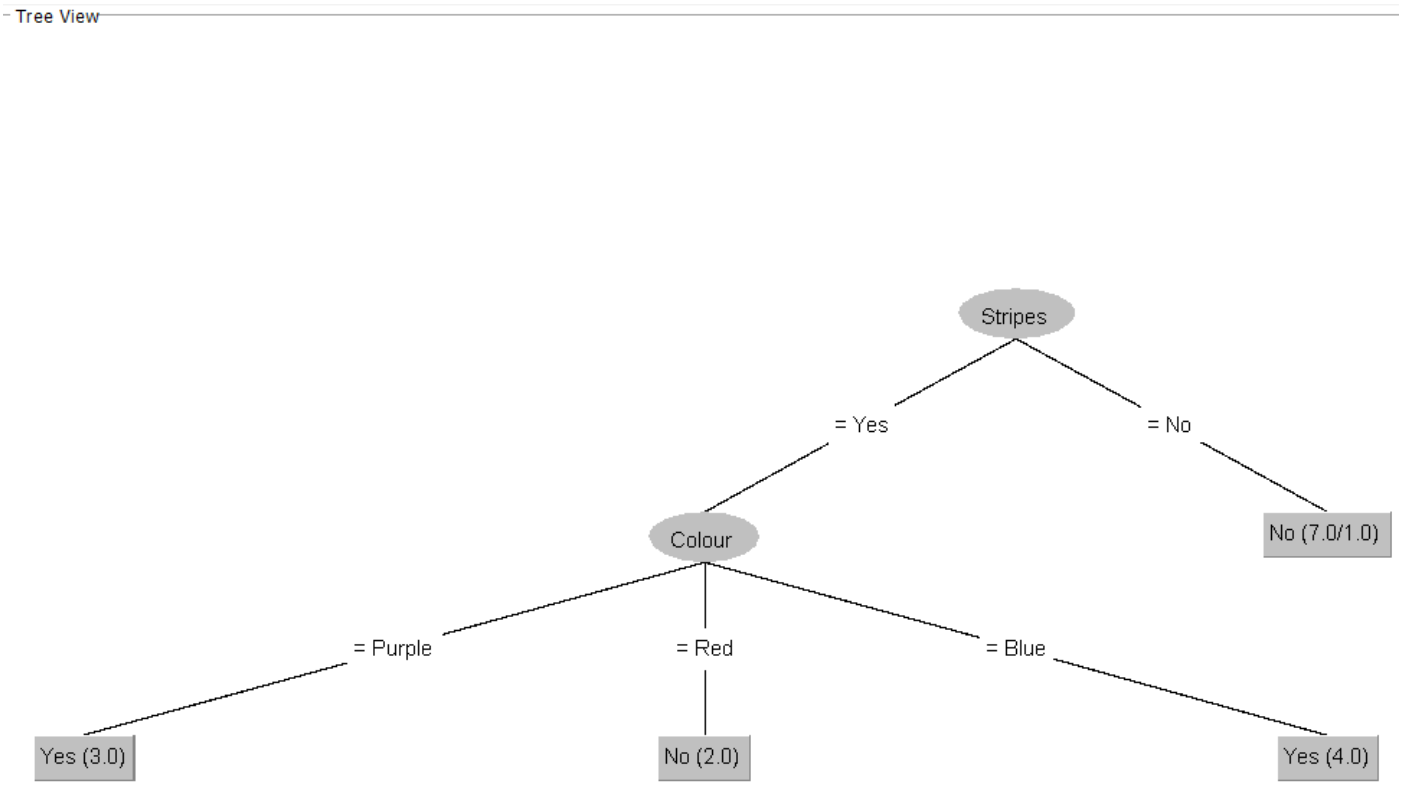
*Purple*
6 τιμές σύνολο, 4 Yes και 2 No
Πλειονότητα: Yes
Λάθος κατηγοριοποίηση: 2

*Red*
4 τιμές, 4 No
Πλειονότητα: No
Λάθος κατηγοριοποίηση: 0

*Blue*
6 τιμές, 4 Yes και 2 No
Πλειονότητα: Yes
Λάθος κατηγοριοποίηση: 2

**Συνολικά λάθος κατηγοριοποιήσεις = 2 + 0 + 2 = 4**

## ii) Weka




Το weka επιλέγει το χαρακτηριστικό Stripes ως ρίζα του δέντρου.

## iii)

| Colour | Height | Stripes | Texture | Poisonous |
|---|---|---|---|---|
| Purple | Tall | Yes | Rough | Yes |
| Red | Tall | Yes | Smooth | No |
| Red | Short | No | Hairy | Yes |
| Blue | Short | No | Smooth | No |

1. Purple, Tall, Yes, Rough → Πλειονότητα Purple = Poisonous → Σωστή κατηγοριοποίηση

2. Red, Tall, Yes, Smooth → Πλειονότητα Red = No Poisonous → Σωστή κατηγοριοποίηση

3. Red, Short, No, Hairy → Πλειονότητα Red = No Poisonous → Λάθος κατηγοριοποίηση, αφού στο παραπάνω dataset εμφανίζεται ως Poisonous Yes

4. Blue, Short, No, Smooth → Πλειονότητα Blue = Poisonous → Λάθος κατηγοριοποίηση, αφού στο παραπάνω dataset εμφανίζεται ως Non Poisonous

**Accuracy 2/4 = 0.5**

**Weka Results**

```
=== Summary ===

Correctly Classified Instances           2               50      %
Incorrectly Classified Instances         2               50      %
```

# (B) Μελέτη περίπτωσης με το WEKA

- **J48**

  ✓ minNumObj = 2

```
=== Detailed Accuracy By Class ===
```

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.983 | 0.014 | 0.994 | 0.983 | 0.989 | 0.963 | 0.989 | 0.992 | unacc |
| | 0.943 | 0.027 | 0.910 | 0.943 | 0.926 | 0.904 | 0.968 | 0.919 | acc |
| | 0.826 | 0.010 | 0.781 | 0.826 | 0.803 | 0.795 | 0.958 | 0.726 | good |
| | 0.846 | 0.003 | 0.917 | 0.846 | 0.880 | 0.876 | 0.958 | 0.836 | vgood |
| Weighted Avg. | 0.963 | 0.016 | 0.964 | 0.963 | 0.963 | 0.940 | 0.982 | 0.959 | |

  ✓ minNumObj = 5

```
=== Detailed Accuracy By Class ===
```

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.974 | 0.031 | 0.987 | 0.974 | 0.980 | 0.935 | 0.987 | 0.993 | unacc |
| | 0.919 | 0.036 | 0.880 | 0.919 | 0.899 | 0.870 | 0.975 | 0.915 | acc |
| | 0.783 | 0.012 | 0.730 | 0.783 | 0.755 | 0.745 | 0.986 | 0.794 | good |
| | 0.769 | 0.005 | 0.847 | 0.769 | 0.806 | 0.800 | 0.949 | 0.847 | vgood |
| Weighted Avg. | 0.946 | 0.030 | 0.947 | 0.946 | 0.947 | 0.908 | 0.983 | 0.962 | |

  ✓ minNumObj = 10

```
=== Detailed Accuracy By Class ===
```

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.971 | 0.035 | 0.985 | 0.971 | 0.978 | 0.928 | 0.987 | 0.994 | unacc |
| | 0.883 | 0.041 | 0.860 | 0.883 | 0.871 | 0.834 | 0.974 | 0.896 | acc |
| | 0.841 | 0.015 | 0.699 | 0.841 | 0.763 | 0.756 | 0.991 | 0.746 | good |
| | 0.708 | 0.007 | 0.793 | 0.708 | 0.748 | 0.740 | 0.962 | 0.822 | vgood |
| Weighted Avg. | 0.936 | 0.034 | 0.939 | 0.936 | 0.937 | 0.893 | 0.983 | 0.956 | |

  ✓ minNumObj = 20

```
=== Detailed Accuracy By Class ===
```

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.959 | 0.023 | 0.990 | 0.959 | 0.974 | 0.917 | 0.985 | 0.994 | unacc |
| | 0.914 | 0.058 | 0.818 | 0.914 | 0.863 | 0.824 | 0.971 | 0.868 | acc |
| | 0.565 | 0.011 | 0.672 | 0.565 | 0.614 | 0.602 | 0.986 | 0.648 | good |
| | 0.800 | 0.010 | 0.754 | 0.800 | 0.776 | 0.767 | 0.981 | 0.723 | vgood |
| Weighted Avg. | 0.927 | 0.030 | 0.930 | 0.927 | 0.928 | 0.878 | 0.982 | 0.942 | |

✓ minNumObj = 50

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.927 | 0.091 | 0.960 | 0.927 | 0.943 | 0.819 | 0.967 | 0.987 | unacc |
| | 0.852 | 0.102 | 0.705 | 0.852 | 0.771 | 0.703 | 0.938 | 0.740 | acc |
| | 0.000 | 0.000 | ? | 0.000 | ? | ? | 0.962 | 0.346 | good |
| | 0.785 | 0.026 | 0.537 | 0.785 | 0.638 | 0.633 | 0.958 | 0.422 | vgood |
| Weighted Avg. | 0.868 | 0.087 | ? | 0.868 | ? | ? | 0.960 | 0.885 | |

---

- **IBK**

✓ k = 1

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.934 | 0.091 | 0.960 | 0.934 | 0.947 | 0.829 | 0.923 | 0.945 | unacc |
| | 0.727 | 0.094 | 0.689 | 0.727 | 0.707 | 0.621 | 0.813 | 0.570 | acc |
| | 0.536 | 0.022 | 0.507 | 0.536 | 0.521 | 0.501 | 0.759 | 0.282 | good |
| | 0.769 | 0.014 | 0.685 | 0.769 | 0.725 | 0.715 | 0.870 | 0.528 | vgood |
| Weighted Avg. | 0.866 | 0.086 | 0.871 | 0.866 | 0.868 | 0.765 | 0.890 | 0.819 | |

✓ k = 5

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.945 | 0.052 | 0.977 | 0.945 | 0.961 | 0.874 | 0.991 | 0.995 | unacc |
| | 0.885 | 0.094 | 0.728 | 0.885 | 0.799 | 0.740 | 0.963 | 0.840 | acc |
| | 0.377 | 0.002 | 0.897 | 0.377 | 0.531 | 0.572 | 0.980 | 0.701 | good |
| | 0.708 | 0.010 | 0.742 | 0.708 | 0.724 | 0.714 | 0.993 | 0.804 | vgood |
| Weighted Avg. | 0.900 | 0.058 | 0.910 | 0.900 | 0.899 | 0.827 | 0.985 | 0.942 | |

✓ k = 10

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.952 | 0.042 | 0.981 | 0.952 | 0.966 | 0.893 | 0.993 | 0.996 | unacc |
| | 0.938 | 0.100 | 0.729 | 0.938 | 0.820 | 0.771 | 0.969 | 0.869 | acc |
| | 0.246 | 0.000 | 1.000 | 0.246 | 0.395 | 0.489 | 0.990 | 0.772 | good |
| | 0.615 | 0.002 | 0.930 | 0.615 | 0.741 | 0.749 | 0.995 | 0.873 | vgood |
| Weighted Avg. | 0.908 | 0.052 | 0.924 | 0.908 | 0.903 | 0.844 | 0.988 | 0.954 | |

✓ k = 20

```
=== Detailed Accuracy By Class ===
```

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.948 | 0.037 | 0.984 | 0.948 | 0.965 | 0.891 | 0.991 | 0.996 | unacc |
| | 0.932 | 0.111 | 0.706 | 0.932 | 0.804 | 0.750 | 0.962 | 0.835 | acc |
| | 0.087 | 0.000 | 1.000 | 0.087 | 0.160 | 0.289 | 0.985 | 0.729 | good |
| | 0.631 | 0.005 | 0.837 | 0.631 | 0.719 | 0.718 | 0.994 | 0.850 | vgood |
| Weighted Avg. | 0.898 | 0.051 | 0.917 | 0.898 | 0.888 | 0.829 | 0.985 | 0.944 | |

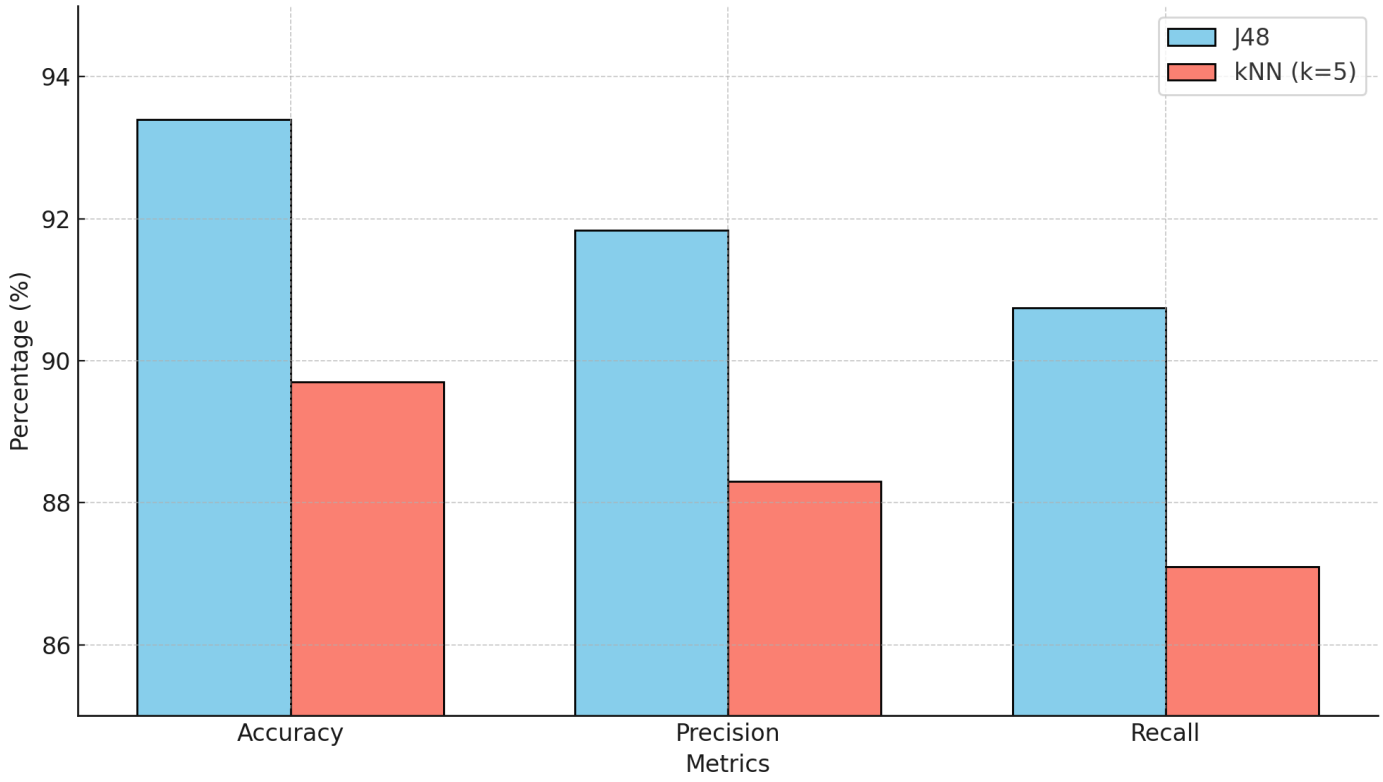
✓ k = 35

```
=== Detailed Accuracy By Class ===
```

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---|---|---|---|---|---|---|---|---|---|
| | 0.936 | 0.058 | 0.974 | 0.936 | 0.954 | 0.857 | 0.990 | 0.995 | unacc |
| | 0.917 | 0.135 | 0.660 | 0.917 | 0.768 | 0.704 | 0.957 | 0.833 | acc |
| | 0.014 | 0.000 | 1.000 | 0.014 | 0.029 | 0.118 | 0.983 | 0.695 | good |
| | 0.446 | 0.002 | 0.906 | 0.446 | 0.598 | 0.627 | 0.994 | 0.842 | vgood |
| Weighted Avg. | 0.876 | 0.071 | 0.903 | 0.876 | 0.863 | 0.785 | 0.982 | 0.942 | |

---

Διάγραμμα




Παρατηρούμε ότι ο J48 ξεπερνάει τον IBK στο Accuracy (93.4% σε αντίθεση με το 89.7%), στο Precision (91.84% έναντι 88.3%) και στο Recall (90.74% έναντι 87.1%). Συμπεραίνουμε, δηλαδή, πως ο J48 πέτυχε μεγαλύτερη ακρίβεια, είχε πιο αξιόπιστες προβλέψεις καθώς και ήταν καλύτερος στον εντοπισμό όλων των σχετικών περιπτώσεων.

Τα καλύτερο tree model του J48 ήταν για minNumObj = 10, καθώς είχε την καλύτερη απόδοση όσον αφορά το Accuracy, Precision και Recall για την πλειονότητα των κατηγοριών ειδικά των “unacc” και “acc”.

```
=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1175   33    2    0 |    a = unacc
   18  339   17   10 |    b = acc
    0    9   58    2 |    c = good
    0   13    6   46 |    d = vgood
```

Το καλύτερο IBK model παρατηρούμε πως είναι για k = 5 ειδικά για τις “unacc” και “acc” κατηγορίες, παρόλο που “χάνει” στην “good”.

```
=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1143   67    0    0 |    a = unacc
   26  340    3   15 |    b = acc
    1   41   26    1 |    c = good
    0   19    0   46 |    d = vgood
```

---

J48: Οι κατηγορίες «good» και «vgood» δεν προβλέπονται τόσο καλά όσο οι κατηγορίες «unacc» και «acc». Η κατηγορία «good» έχει σχετικά χαμηλό Recall (84,1%) και η κατηγορία «vgood» ακόμη χαμηλότερο (70,8%).

kNN: Ομοίως, η κατηγορία «good» περιέχει πολύ χαμηλό Recall (37,7%) και Precision (50,7%). Η «vgood» δεν προβλέπεται επίσης ικανοποιητικά, με Recall στο 70,8%.

---

Rules για κάθε κατηγορία για τον J48 με minNumObj = 10

- Unacceptable
  **If** (buying = high) **and** (maint = high) **then** unacc

- Acceptable
  **If** (buying = low) **and** (doors = 4) **then** acc

- Good
  **If** (buying = medium) **and** (maint = medium) **and** (doors = 4) **then** good

- Very Good
  **If** (buying = high) **and** (maint = low) **then** vgood