



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

Τμήμα Πληροφορικής και  
Τηλεπικοινωνιών

## Waikato Environment for Knowledge Analysis (WEKA)

Εξαγωγή συχνών προτύπων από δεδομένα εστίασης.

### Συγγραφείς:

Λάμπρος Γραμματικόπουλος, 2022201800038, [dit18038@uop.gr](mailto:dit18038@uop.gr)

Κωσταντίνος Νικόλαος Κολοτούρος, 2022201800090, [dit18090@uop.gr](mailto:dit18090@uop.gr)

### Διδάσκων:

Χρήστος Τρυφωνόπουλος

Ιανουάριος 2020

## Πίνακας περιεχομένων

Ερώτημα 1 .....	2
Ερώτημα 2 .....	3
Ερώτημα 3 .....	3
Ερώτημα 4 .....	4
Μετρικές confidence, lift, leverage και conviction: .....	5
Ερώτημα 5 .....	7
Ερώτημα 6 .....	11
Ερώτημα 7 .....	11
Ερώτημα 8 .....	12
Συνοδευτικά αρχεία .....	13
Τρόπος εκτέλεσης .....	13
Πηγές .....	13

## Ερώτημα 1

Για την εύρεση των διαφορετικών ειδών που πουλάει το κατάστημα χρησιμοποιήθηκε κώδικας σε Java (που περιέχεται στα παραδοτέα αρχεία), ο οποίος διατρέχει το δοθέν αρχείο και βρίσκει τα διαφορετικά είδη που πουλάει το κατάστημα και τα εμφανίζει όλα από μία φορά. Το αποτέλεσμα του φαίνεται παρακάτω (σύνολο 24 διαφορετικά είδη):

γύρο\_χοιρινό, κοτομπέικον, χωριάτικη, σαγανάκι, ψωμί\_λευκό, αναψυκτικό, φρουτοσαλάτα, πατάτες, παγωτό, ψωμί\_μαύρο, χυμός, καίσαρα, αγγουροντομάτα, γύρο\_κοτόπουλο, μαρούλι, τυροκροκέτες, ψωμί\_πολύσπορο, μπύρα, μπιφτέκι\_λαχανικών, γιαούρτι-μέλι, νερό, καλαμάρι, μιλκσέικ, κοτομπουκιές.

## Ερώτημα 2

Τα δεδομένα χρειάστηκαν κάποια προεπεξεργασία για μπορούν να χρησιμοποιηθούν από το WEKA. Συγκεκριμένα, το αρχείο των δεδομένων έπρεπε από .txt αρχείο να μετατραπεί σε .arff αρχείο. Έπειτα, τα δεδομένα έπρεπε να μετασχηματιστούν σε αγγλικούς χαρακτήρες ώστε να μπορεί να τα διαβάσει το εργαλείο WEKA. Συγκεκριμένα, το είδος «γύρο\_χοιρινό» μετασχηματίστηκε σε «guroXoirino», το είδος «κοτομπέικον» μετασχηματίστηκε σε «kotompeikon» κ.τ.λ. Τέλος, από κάθε παραγγελία έπρεπε να αφαιρέσαμε τα διπλότυπα είδη σε μία παραγγελία για να λειτουργήσει ο αλγόριθμος Apriori (αν και αυτήν την διαδικασία την κάνει, από ότι διαπιστώθηκε, από μόνο του το πρόγραμμα WEKA).

## Ερώτημα 3

Για τον μετασχηματισμό του .txt αρχείου σε .arff ο κώδικας έκανε τις εξής λειτουργίες:

1. Πρόσθεσε τις απαραίτητες κεφαλίδες (Headers) που χρειάζεται ένα .arff αρχείο (@RELATION, @ATTRIBUTE, @DATA).
2. Παράλειψε όλα τα id και τις ηλικίες (που ήταν δεδομένα τύπου Numeric) από το αρχικό αρχείο κατά την προσθήκη τους στο .arff αρχείο. Αυτό έγινε προκειμένου να είναι όλα τα δεδομένα τύπου Nominal ώστε να μπορεί ο αλγόριθμος Apriori να τα δεχθεί, χωρίς περαιτέρω επεξεργασία.
3. Επίσης, ο κώδικας βρήκε από το αρχείο μας ότι ο μέγιστος αριθμός διαφορετικών ειδών σε κάποια παραγγελία ήταν 11. Οπότε, προστέθηκαν 11 κεφαλίδες τύπου @ATTRIBUTE με αναμενόμενη είσοδο, η κάθε μία, κάποιο από τα 24 διαφορετικά είδη.

4. Τέλος, για να γίνει το αρχείο τύπου .arff, ο κώδικας που δημιουργήθηκε πρόσθεσε σε κάθε γραμμή του αρχείου που δεν υπήρχαν και τα 11 διαφορετικά είδη ένα αγγλικό ερωτηματικό «?» για κάθε θέση που δεν ήταν συμπληρωμένη.

Έτσι το αρχείο .arff που προέκυψε είχε την εξής μορφή:

```
@RELATION Sasmos
@ATTRIBUTE class0 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class1 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class2 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class3 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class4 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class5 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class6 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class7 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class8 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class9 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}
@ATTRIBUTE class10 {guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,psw}

@DATA
guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,fROUTosalata,patates,pagwto,?,?
xwriatikh,saganaki,pswmiMauro,xumos,kaisara,aggourontomata,?,?,?,?
guroKotopoulo,xwriatikh,marouli,fROUTosalata,patates,turokroketes,?,?,?,?
guroKotopoulo,saganaki,marouli,pswmiLeuko,xumos,aggourontomata,patates,?,?,?,?
guroKotopoulo,xwriatikh,marouli,pswmiMauro,pswmiPolusporo,mpura,anapsuktiko,fROUTosalata,patates,?,?,?
mpiftekiLaxanikwn,pswmiPolusporo,anapsuktiko,fROUTosalata,turokroketes,?,?,?,?,?
anapsuktiko,aggourontomata,giaourtiMeli,?,?,?,?,?,?
guroXoirino,kotompeikon,xwriatikh,saganaki,pswmiLeuko,anapsuktiko,kaisara,aggourontomata,patates,giaourtiMeli,?
marouli,anapsuktiko,pagwto,?,?,?,?,?,?,?
mpiftekiLaxanikwn,xwriatikh,pswmiPolusporo,anapsuktiko,turokroketes,?,?,?,?,?,?
guroKotopoulo,xwriatikh,marouli,fROUTosalata,patates,turokroketes,pagwto,?,?,?,?
anapsuktiko,aggourontomata,turokroketes,pagwto,?,?,?,?,?,?
guroKotopoulo,saganaki,pswmiLeuko,xumos,aggourontomata,patates,turokroketes,?,?,?,?
```

## Ερώτημα 4

Η δυσκολία που αντιμετωπίσαμε κατά την εκτέλεση του αλγορίθμου Arriori ήταν πως στην αρχή δεν δεχόταν τα δεδομένα μας. Αυτό συνέβαινε, γιατί δεν ήταν όλα τύπου Nominal καθώς το id της παραγγελίας και η ηλικία ήταν τύπου Numeric. Έτσι, για να λύσουμε το πρόβλημα τα δύο αυτά πεδία αφαιρέθηκαν.

Μετρικές confidence, lift, leverage και conviction:

$$\text{confidence}(A \rightarrow C) = \frac{\text{support}(A \rightarrow C)}{\text{support}(A)}, \quad \text{range: } [0, 1]$$

$$\text{lift}(A \rightarrow C) = \frac{\text{confidence}(A \rightarrow C)}{\text{support}(C)}, \quad \text{range: } [0, \infty]$$

$$\text{leverage}(A \rightarrow C) = \text{support}(A \rightarrow C) - \text{support}(A) \times \text{support}(C), \quad \text{range: } [-1, 1]$$

$$\text{conviction}(A \rightarrow C) = \frac{1 - \text{support}(C)}{1 - \text{confidence}(A \rightarrow C)}, \quad \text{range: } [0, \infty]$$

- Το confidence ενός κανόνα  $A \rightarrow C$  είναι η πιθανότητα να υπάρχει το C σε μία συναλλαγή δεδομένου ότι υπάρχει επίσης και το A. Για το confidence το  $A \rightarrow C \neq C \rightarrow A$ . Η μέγιστη τιμή του confidence είναι το 1 για τον κανόνα  $A \rightarrow C$  αν το A εμφανίζεται πάντα με το C.
- Το lift χρησιμοποιείται συχνά για να μετρήσει το πόσο περισσότερο τα A και C προκύπτουν μαζί από ότι θα περιμέναμε αν ήταν στατιστικά ανεξάρτητα. Αν τα A και C είναι ανεξάρτητα τότε lift=1. Το lift βρίσκει πολύ ισχυρούς κανόνες για τα λιγότερα συχνά αντικείμενα.
- Το leverage υπολογίζει την διαφορά μεταξύ της παρατηρούμενης συχνότητας των A και C που εμφανίζονται μαζί και της συχνότητας που θα αναμενόταν αν τα A και C ήταν ανεξάρτητα. Αν leverage=0 τότε τα A και C είναι ανεξάρτητα. Το leverage στοχεύει να βρει κανόνες για αντικείμενα με πολλές εμφανίσεις και υψηλό support στην συλλογή.
- Μια υψηλή τιμή conviction σημαίνει ότι το C εξαρτάται σε μεγάλο βαθμό από το A. Παρόμοια με το lift, εάν τα αντικείμενα είναι ανεξάρτητα τότε conviction=1.

## Σύγκριση Confidence με Lift

Η μετρική Lift μπορεί να χρησιμοποιηθεί στην περίπτωση που ο μαγαζάτορας θέλει να μάθει πια προϊόντα αγοράζονται μαζί με τα λιγότερο συχνά. Με σκοπό να προωθήσει τα προϊόντα που αγοράζονται λιγότερο και να τα βάλει σε προσφορά.

Η μετρική Confidence από την άλλη, μπορεί να χρησιμοποιηθεί από τον μαγαζάτορα αν θελήσει να προωθήσει τα προϊόντα που αγοράζονται πιο συχνά και να κάνει τις κατάλληλες εκπτώσεις σε αυτά με σκοπό να γίνει πιο ανταγωνιστικός στα λοιπά μαγαζιά.

Παρακάτω φαίνονται τα αποτελέσματα του Apriori ταξινομημένα με βάση την μετρική Confidence. Συγκεκριμένα, φαίνονται οι καλύτεροι κανόνες (16 στο σύνολο) με το υψηλότερο Support (0.2 που βρέθηκε με δοκιμές) ξεκινώντας από το ελάχιστο που υπήρχε για αυτούς Confidence (0.5 που επίσης βρέθηκε με δοκιμές).

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Associator

Choose **Apriori** -N 100 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -c -1

Start Stop

Associator output

Result list (right-click...)

16:26:49 - Apriori  
16:32:22 - Apriori  
16:32:31 - Apriori  
16:32:47 - Apriori  
16:33:39 - Apriori  
16:34:08 - Apriori  
16:34:14 - Apriori  
16:34:21 - Apriori  
16:34:27 - Apriori  
16:34:38 - Apriori  
16:34:44 - Apriori  
16:44:12 - Apriori  
16:44:26 - Apriori  
16:44:31 - Apriori  
16:44:37 - Apriori  
16:44:44 - Apriori  
16:44:50 - Apriori  
16:45:19 - Apriori  
16:45:25 - Apriori  
16:45:32 - Apriori  
16:45:37 - Apriori  
16:46:06 - Apriori

Minimum support: 0.2 (238 instances)  
Minimum metric <confidence>: 0.5  
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6  
Size of set of large itemsets L(2): 5  
Size of set of large itemsets L(3): 1

Best rules found:

1. class2=xwriatikh 260 ==> class0=guroXoirino 260 <conf:(1)> lift:(4.42) lev:(0.17) [201] conv:(201.18)
2. class2=xwriatikh 260 ==> class1=kotompeikon 260 <conf:(1)> lift:(4.45) lev:(0.17) [201] conv:(201.61)
3. class1=kotompeikon class2=xwriatikh 260 ==> class0=guroXoirino 260 <conf:(1)> lift:(4.42) lev:(0.17) [201] conv:(201.18)
4. class0=guroXoirino class2=xwriatikh 260 ==> class1=kotompeikon 260 <conf:(1)> lift:(4.45) lev:(0.17) [201] conv:(201.61)
5. class0=guroXoirino class1=kotompeikon 260 ==> class2=xwriatikh 260 <conf:(1)> lift:(4.57) lev:(0.17) [203] conv:(203.15)
6. class2=xwriatikh 260 ==> class0=guroXoirino class1=kotompeikon 260 <conf:(1)> lift:(4.57) lev:(0.17) [203] conv:(203.15)
7. class1=kotompeikon 267 ==> class0=guroXoirino 260 <conf:(0.97)> lift:(4.3) lev:(0.17) [199] conv:(25.82)
8. class1=kotompeikon 267 ==> class2=xwriatikh 260 <conf:(0.97)> lift:(4.45) lev:(0.17) [201] conv:(26.08)
9. class1=kotompeikon 267 ==> class0=guroXoirino class2=xwriatikh 260 <conf:(0.97)> lift:(4.45) lev:(0.17) [201] conv:(26.08)
10. class0=guroXoirino 269 ==> class1=kotompeikon 260 <conf:(0.97)> lift:(4.3) lev:(0.17) [199] conv:(20.86)
11. class0=guroXoirino 269 ==> class2=xwriatikh 260 <conf:(0.97)> lift:(4.42) lev:(0.17) [201] conv:(21.02)
12. class0=guroXoirino 269 ==> class1=kotompeikon class2=xwriatikh 260 <conf:(0.97)> lift:(4.42) lev:(0.17) [201] conv:(21.02)
13. class2=marouli 318 ==> class1=xwriatikh 298 <conf:(0.94)> lift:(2.42) lev:(0.15) [174] conv:(9.27)
14. class0=guroKotopoulo 317 ==> class1=xwriatikh 248 <conf:(0.78)> lift:(2.02) lev:(0.11) [125] conv:(2.77)
15. class1=xwriatikh 461 ==> class2=marouli 298 <conf:(0.65)> lift:(2.42) lev:(0.15) [174] conv:(2.06)
16. class1=xwriatikh 461 ==> class0=guroKotopoulo 248 <conf:(0.54)> lift:(2.02) lev:(0.11) [125] conv:(1.58)

Status

OK

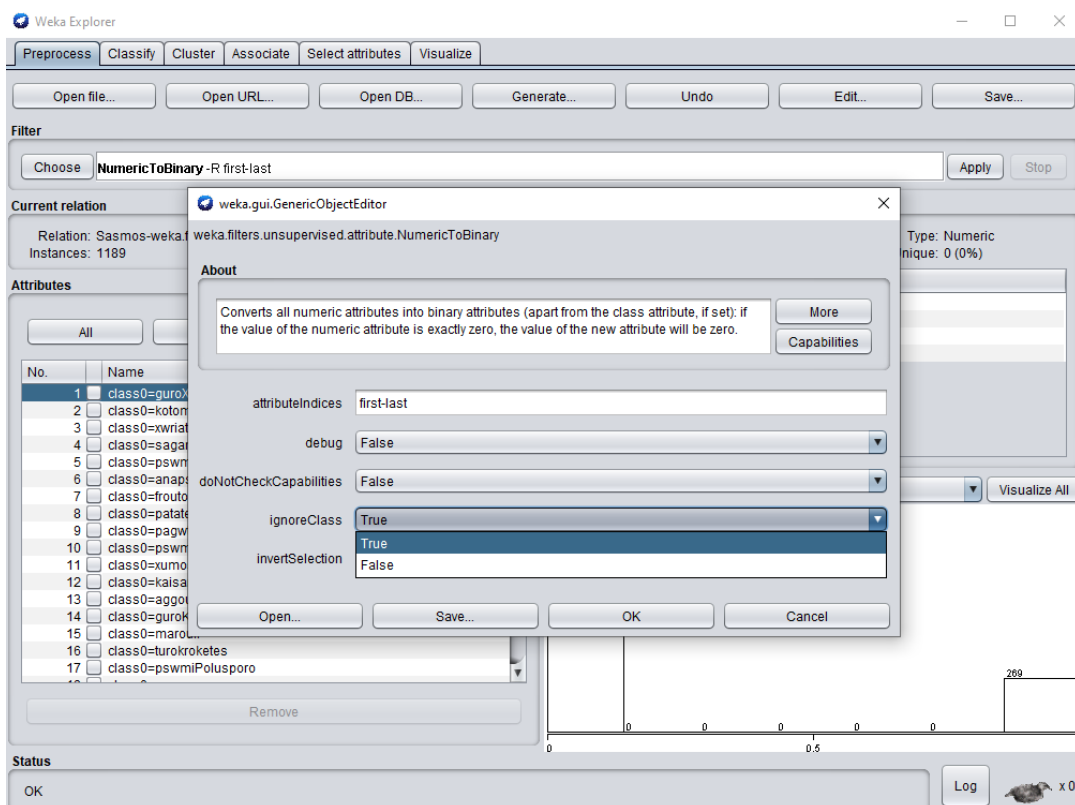
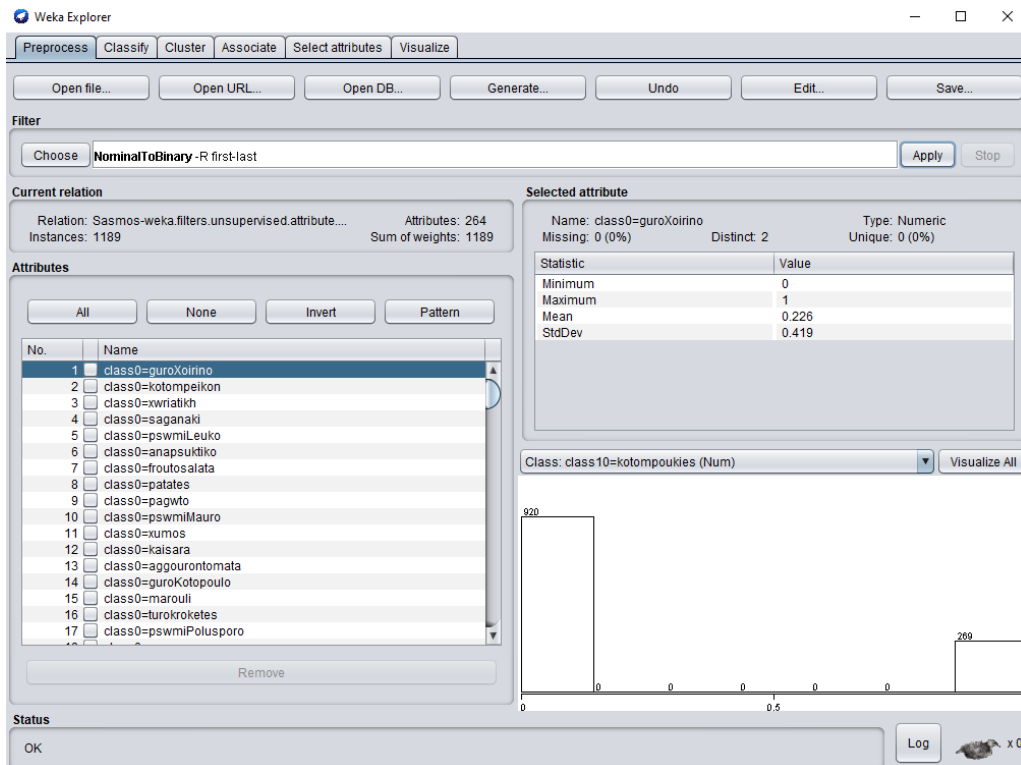
Log x0

## Ερώτημα 5

Προκειμένου τα δεδομένα μας να εισαχθούν στον αλγόριθμο FPGrowth, έπρεπε να τους εφαρμόσουμε την τεχνική της δυαδοποίησης (binarization) προκειμένου όλα τα είδη της κάθε κλάσης να μπορούν να χαρακτηριστούν από True ή False δηλαδή το αν περιέχονται ή όχι στην εκάστοτε παραγγελία. Η διαδικασία αυτή στο WEKA φαίνεται στις παρακάτω εικόνες:

The screenshot shows the Weka Explorer interface with the 'Preprocess' tab selected. The 'Filter' list on the left includes 'NominalToBinary', which is highlighted. A tooltip for this filter is displayed, stating: 'Converts all nominal attributes into binary numeric attributes'. The 'Selected attribute' panel shows 'Name: class0', 'Missing: 0 (0%)', 'Distinct: 16', and 'Type: Nominal'. Below this, a table lists the distinct values and their counts: 'guroxoirino' (269) and 'kotompeikon' (0). The 'Visualize All' button is visible at the bottom right of the filter panel.

The screenshot shows the Weka Explorer interface after applying the 'NominalToBinary' filter. The 'Filter' list now shows 'NominalToBinary -R first-last'. The 'Current relation' panel shows 'Relation: Sasmos' and 'Instances: 1189'. The 'Attributes' panel lists the attributes: 'class0', 'class1', 'class2', 'class3', 'class4', 'class5', 'class6', 'class7', 'class8', 'class9', and 'class10'. The 'Selected attribute' panel shows 'Name: class0', 'Missing: 0 (0%)', 'Distinct: 16', and 'Type: Nominal'. Below this, a table lists the distinct values and their counts: 'guroxoirino' (269), 'kotompeikon' (0), 'xwratih' (47), 'saganaki' (39), 'pswmiLeuko' (0), 'anapsukdiko' (5), 'frouosalata' (1), 'patates' (3), 'pagwio' (0), and 'sagwmiLeuko' (0). The 'Visualize All' button is visible at the bottom right of the filter panel.





Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Attributes: 264  
Sum of weights: 1189

Selected attribute

Name: class0=guroXoirino  
Missing: 0 (0%) Distinct: 2  
Type: Numeric  
Unique: 0 (0%)

Statistic Value

Converts all numeric attributes into binary attributes (apart from the class attribute, if set): if the value of the numeric attribute is exactly zero, the value of the new attribute will be zero

If the value of the numeric attribute is missing, the value of the new attribute will be missing. Otherwise, the value of the new attribute will be one. The new attributes will be nominal.

**CAPABILITIES**

Class -- Binary class, Date class, Empty nominal class, Missing class values, No class, Nominal class, Numeric class, Relational class, String class, Unary class

Attributes -- Binary attributes, Date attributes, Empty nominal attributes, Missing values, Nominal attributes, Numeric attributes, Relational attributes, String attributes, Unary attributes

Interfaces -- StreamableFilter, UnsupervisedFilter, WeightedAttributesHandler, WeightedInstancesHandler

Additional

Minimum number of instances: 0

Filter... Remove filter Close

Status

OK Log x 0

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose NumericToBinary -R first-last Apply Stop

Current relation

Relation: Sasmos-weka.filters.unsupervised.attribute...  
Instances: 1189 Attributes: 264  
Sum of weights: 1189

Attributes

All None Invert Pattern

No.	Name
1	class0=guroXoirino_binarized
2	class0=kotompeikon_binarized
3	class0=xwriatikh_binarized
4	class0=saganaki_binarized
5	class0=pswmiLeuko_binarized
6	class0=anapsuktiko_binarized
7	class0=frutosalata_binarized
8	class0=patates_binarized
9	class0=pagwto_binarized
10	class0=pswmiMauro_binarized
11	class0=xumos_binarized
12	class0=kaisara_binarized
13	class0=aggourontomata_binarized
14	class0=guroKotopoulo_binarized
15	class0=marouli_binarized
16	class0=turokroketes_binarized
17	class0=pswmiPolusporo_binarized

Remove

Selected attribute

Name: class0=guroXoirino\_binarized  
Missing: 0 (0%) Distinct: 2  
Type: Nominal  
Unique: 0 (0%)

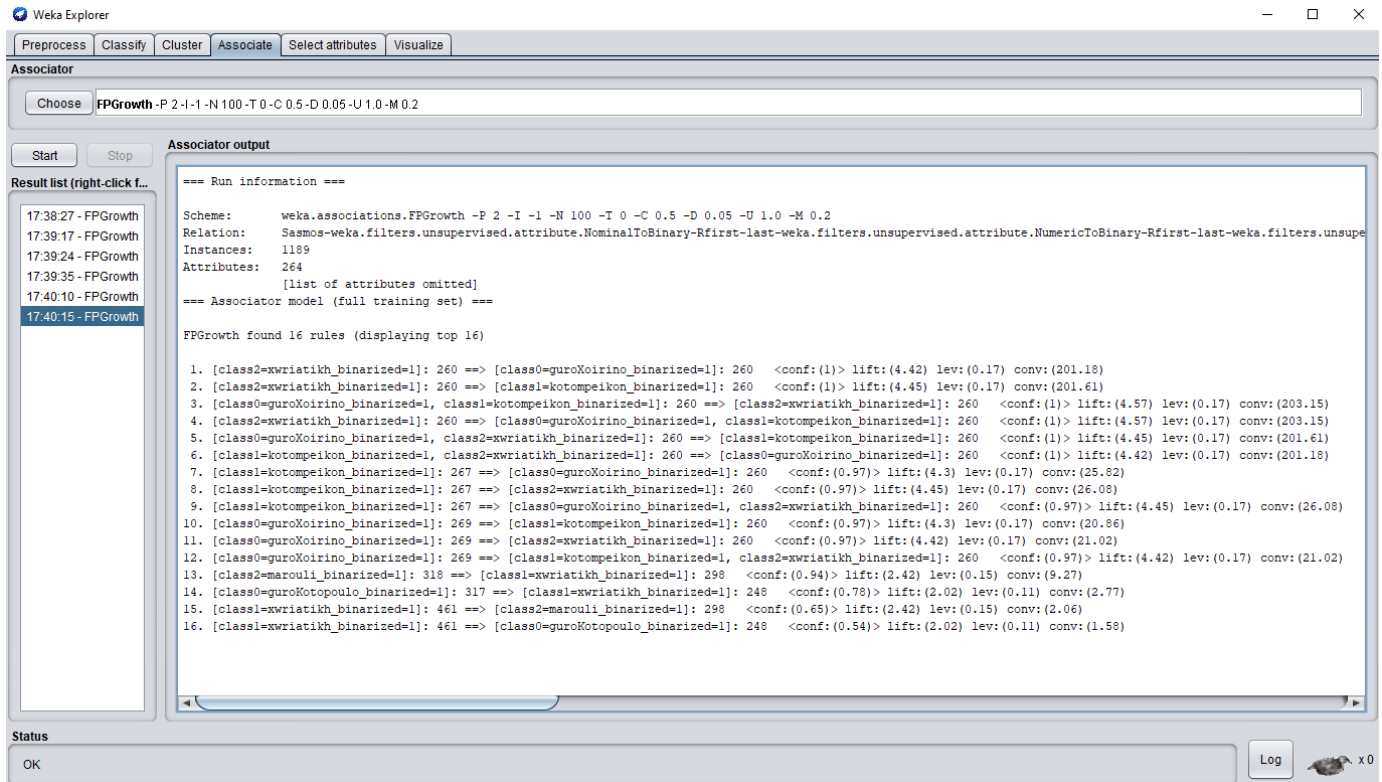
No.	Label	Count	Weight
1	0	920	920.0
2	1	269	269.0

Class: class10=kotompoukies\_binarized (Nom) Visualize All

Status

OK Log x 0

Παρακάτω φαίνονται τα αποτελέσματα του FPGrowth ταξινομημένα με βάση την μετρική Confidence. Συγκεκριμένα, φαίνονται οι καλύτεροι κανόνες (16 στο σύνολο) με το υψηλότερο Support (0.2 που βρέθηκε με δοκιμές) ξεκινώντας από το ελάχιστο που υπήρχε για αυτούς Confidence (0.5 που επίσης βρέθηκε με δοκιμές).



## Ερώτημα 6

Οι αλγόριθμοι αυτοί μας βοηθούν να βρούμε ποια προϊόντα αγοράζει ο πελάτης, μαζί με ποια άλλα, πιο συχνά. Συνεπώς, μπορούμε για παράδειγμα, να του κάνουμε προσφορές για συγκεκριμένους συνδυασμούς προϊόντων, προκειμένου να τον ωθήσουμε να τα αγοράσει μαζί και έτσι να μεγιστοποιούμε το κέρδος μας.

Προσφορές	
Προϊόντα	Προσφορά
Χωριάτικη + Γύρο Χοιρινό	-15%
Χωριάτικη + Κοτομπέικον	-15%
Χωριάτικη + Γύρο Χοιρινό + Κοτομπέικον	-10% και δώρο 1 Cola 1L
Γύρο Χοιρινό + Κοτομπέικον	-10%
Χωριάτικη + Μαρούλι	-10%
Χωριάτικη + Γύρο Κοτόπουλο	-5%
Χωριάτικη + Μαρούλι + Γύρο Κοτόπουλο	-10%

## Ερώτημα 7

Αν είχαμε κατά νου, τον χρόνο παρασκευής του κάθε είδους και πως αυτός μπορεί να μειωθεί, τότε με τα δεδομένα αποτελέσματα θα μπορούσαμε να κάνουμε αλλαγές στην προετοιμασία τους αλλά και στις τιμές του μενού. Για παράδειγμα: θα ήταν αποδοτικό, ο μάγειρας να έχει προετοιμάσει μεγαλύτερη ποσότητα «χωριάτικης» σαλάτας, αφού αυτή εμφανίζεται πολλές φορές και με πολλούς συνδυασμούς σε παραγγελίες. Το ίδιο ισχύει και για την προετοιμασία του «γύρου χοιρινού». Όμως, αυτό θα έχει σαν αποτέλεσμα να αφαιρεθούν από το μενού προσφορών κάποια προϊόντα όπως η σαλάτα «μαρούλι», αφού οι παραγγελίες με σαλάτα «μαρούλι» είναι λιγότερες από την σαλάτα «χωριάτικη» και η προετοιμασία της θα αργούσε την προετοιμασία της «χωριάτικης σαλάτας». Έτσι, ένα πιθανό μενού προσφορών θα ήταν το εξής:

Προσφορές	
Προϊόντα	Προσφορά
Χωριάτικη + Γύρο Χοιρινό	-15%
Χωριάτικη + Κοτομπέικον	-15%
Χωριάτικη + Γύρο Χοιρινό + Κοτομπέικον	-10% και δώρο 1 Cola 1L
Γύρο Χοιρινό + Κοτομπέικον	-10%
Χωριάτικη + Γύρο Κοτόπουλο	-10%

## Ερώτημα 8

Αν είχαμε και την τιμή για κάθε είδος, ο αλγόριθμος Apriori θα άλλαζε ώστε να βρίσκει μόνο τα συχνά πρότυπα που δεν ξεπερνούν ένα όριο τιμής συγκρίνοντας την τιμή του κάθε συνόλου ( που υπολόγισε αθροιστικά από το κάθε είδος) με το όριο της τιμής της επιλογής μας. Παρακάτω φαίνεται αναλυτικά ο ψευδοκώδικας:

```

L1={large1-itemsets}      count item frequency
for(k=2; Lk-1≠{}; k++) de begin
    Ck=apriori-gen(Lk-1);    new candidates
    ∀orders o∈D do begin
        Co=subset(Ck,o);      candidates in transaction
        ∀candidates c∈Co do
            c.count++;           determine support
            c.prize++;           determine candidate prize
        end
        Lk={c∈Ck | c.count≥minsup | c.prize≤maxPrize}  create new set
    End
Answer=UkLk;

```

## Συνοδευτικά αρχεία

Μαζί με την αναφορά παραδίδεται ολόκληρος ο φάκελος του Project που περιέχει τον κώδικα που υλοποιήσαμε και χρησιμοποιήθηκε για την εργασία. Ο φάκελος αυτός είναι φάκελος από τον IDE (IntelliJ IDEA) που χρησιμοποιήθηκε. Επίσης, παρατίθεται το αρχείο .arff που παρήγαγε ο κώδικας το οποίο και χρησιμοποιήθηκε. Τέλος, παραδόθηκε και το εκτελέσιμο αρχείο του κώδικα.

## Τρόπος εκτέλεσης

Τοποθετούμε τα αρχεία WEKA.jar και dataset.txt στον ίδιο φάκελο. Ανοίγουμε ένα terminal πλοηγούμεστε στον παραπάνω φάκελο, τρέχουμε την εντολή «`java -jar WEKA.jar`» και δημιουργείται το αρχείο dataset.arff. Εισάγουμε το αρχείο που παράχθηκε στο εργαλείο WEKA και κάνουμε τις απαραίτητες παραμετροποιήσεις για να «τρέξουν» οι αλγόριθμοι όπως φαίνεται στις παραπάνω εικόνες.

## Πηγές

<https://www.cs.waikato.ac.nz/ml/weka/>

[http://rasbt.github.io/mlxtend/user\\_guide/frequent\\_patterns/association\\_rules/](http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/)

<https://support.bigml.com/hc/en-us/articles/207310225-What-is-the-difference-between-lift-and-leverage->