

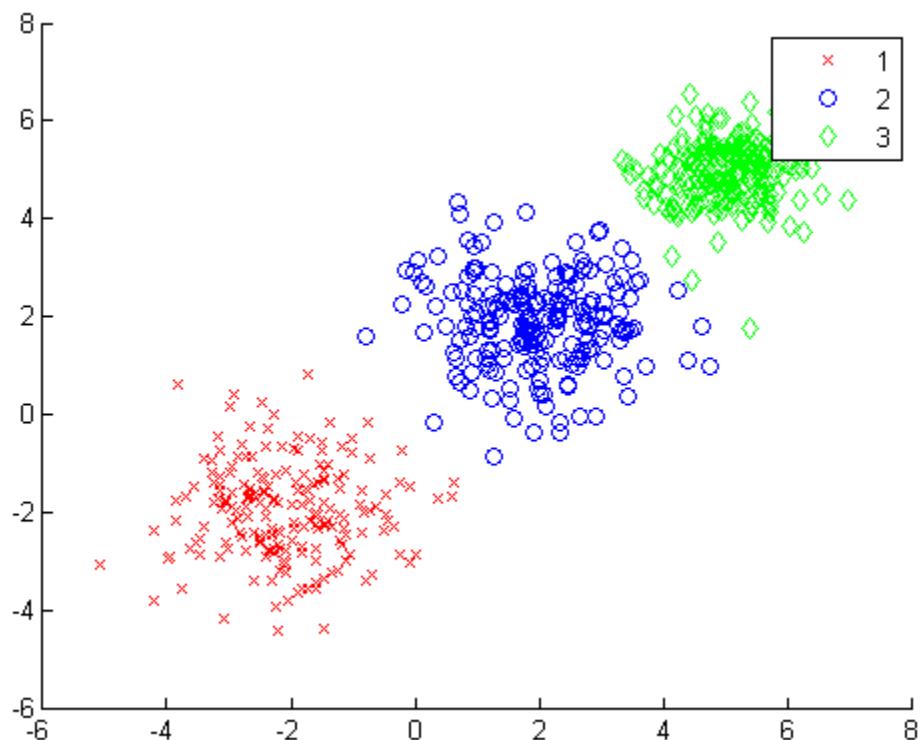
Aristotle University of Thessaloniki, Department of Computer Science and Electronics

Αναγνώριση Προύπτων

Εργασία 2 – Ομαδοποίηση

Δημανίδης Αναστάσιος 7422 (dhmtasos@gmail.com)

Κλειτσιώτης Ιωάννης 7447 (ikleitsi@auth.gr)



1. Εισαγωγή

Στην εργασία αυτή κληθήκαμε να εφαρμόσουμε τεχνικές αναγνώρισης προτύπων σε ένα σετ δεδομένων κατανάλωσης ηλεκτρικής ενέργειας. Οι μετρήσεις του σετ δεδομένων έχουν πραγματοποιηθεί σε ένα σπίτι στην Γαλλία την περίοδο Δεκέμβριος 2006 – Νοέμβριος 2010. Συνολικά το σετ περιέχει περίπου 2×10^6 εγγραφές με τα εξής χαρακτηριστικά:

- *Date* – Ημερομηνία Μέτρησης
- *Time* – Λεπτό Μέτρησης
- *Global_active_power* – Συνολική ενεργός ισχύς σπιτιού (kW)
- *Global_reactive_power* – Συνολική άεργος ισχύς σπιτιού (kW)
- *Voltage* – Τάση σπιτιού (Volt)
- *Global_intensity* – Ένταση σπιτιού (Ampere)
- *Sub_metering_1* – Κατανάλωση σε συσκευές κουζίνας (W/h)
- *Sub_metering_2* – Κατανάλωση σε συσκευές πλυσταριού (W/h)
- *Sub_metering_3* – Κατανάλωση θερμοσίφωνα και κλιματιστικού (W/h)

Όλες οι τιμές που αφορούν ενέργεια είναι minute-averaged. Τα τρία Sub_metering είναι υποσύνολο του Global_active_power. Μάλιστα ο τύπος

`global_active_power * 1000/60 - sub_metering_1 - sub_metering_2 - sub_metering_3`

δίνει την υπόλοιπη κατανάλωση ισχύος στο σπίτι. Επίσης δεν έχουμε καμία πληροφορία σχετικά με το πόσοι άνθρωποι δραστηριοποιούνται στο σπίτι, τι ηλικίας είναι, τι ρουτίνες δραστηριοτήτων έχουν κ.λ.π.

Σκοπός μας ήταν να ομαδοποιήσουμε την καταναλωτική συμπεριφορά μέσα σε αυτό το σπίτι βάσει μετρικών όπως χρόνος, max, avg και sum. Η ομαδοποίηση πραγματοποιήθηκε με γνωστούς στην επιστήμη αλγορίθμους ομαδοποίησης, όπως kmeans και DBSCAN. Σε αυτό το έγγραφο θα αναλύσουμε όλα τα στάδια της διαδικασίας αναγνώρισης προτύπων μέχρι το τελικό μοντέλο.

2. Τυπογραφικές παραδοχές

Το έγγραφο είναι γραμμένο σε γραμματοσειρά Constantia 11. Οι επικεφαλίδες είναι σε **έντονη γραφή** και έχουν μέγεθος 14. Οι υποκεφαλίδες είναι επίσης σε **έντονη γραφή** και έχουν μέγεθος 12. Τα χαρακτηριστικά του σετ δεδομένων επισημαίνονται με πλάγια γραφή. Οι λεζάντες των εικόνων είναι μεγέθους 10 και έχουν κεντρική στοιχίση. Τα χαρακτηριστικά που δημιουργήσαμε εμείς και δεν υπάρχουν στο σετ δεδομένων επισημαίνονται με πλάγια υπογραμμισμένη γραφή. Οι φράσεις και οι λέξεις που είναι σημαντικές επισημαίνονται με **έντονη πλάγια γραφή**. Η παράθεση κώδικα γίνεται με την παρακάτω μορφοποίηση

Code line 1

Code line 2

...

3. Γενικές παραδοχές

- Το πρόβλημα του outlier detection, η ανάλυση και οι λύσεις που δώσαμε σε αυτό παρατίθενται στην ενότητα 5.2. Εξωκείμενες τιμές. Η πρακτική εφαρμογή του όμως βρίσκεται στην ενότητα 6.Clustering.
- Σε όλα τα διαγράμματα του Weka έχουμε βάλει στην μέγιστη τιμή το option Jitter (αναδεικνύει την πυκνότητα), για καλύτερο οπτικό αποτέλεσμα.

4. Προγράμματα που χρησιμοποιήθηκαν

- Weka – Απεικόνιση δεδομένων, εφαρμογή αλγορίθμων ομαδοποίησης
- Γλώσσα προγραμματισμού Ruby – Προεπεξεργασία πάσης φύσεως στο σετ δεδομένων. Συγκεκριμένα συγγράφηκε το πρόγραμμα csv_reader.rb.
- Matlab – Εντοπισμός outlier, περιστασιακή εφαρμογή αλγορίθμων ομαδοποίησης, αξιολόγηση ομαδοποίησης.
- Excel – Μερικός έλεγχος ορθότητας αποτελεσμάτων του csv_reader.rb, ελάχιστη προεπεξεργασία (γρήγορη αναζήτηση μηδενικών τιμών, ταξινόμηση)

5. Preprocessing

Η διαδικασία του preprocessing αποδείχτηκε ως η πιο χρονοβόρα τόσο λόγω του μεγέθους του σετ δεδομένων, όσο και λόγω της ανάγκης για δημιουργία πάρα πολλών υποσετ προκειμένου να καταλήξουμε σε συμπεράσματα. Παρ' όλο που στην προηγούμενη εργασία η χρήση του excel και περιστασιακών script αρκούσε, στην συγκεκριμένη ήταν επιτακτικό να αυτοματοποιηθεί η διαδικασία. Συνεπώς, καθ' όλη την διάρκεια της προεπεξεργασίας αλλά και του clustering, συγγράφονταν πρόγραμμα ανάλογα με τις ανάγκες που προέκυπταν. Το πρόγραμμα αυτό ονομάζεται csv_reader.rb και επιτελεί λειτουργίες όπως:

- Dataset splitting
- Dataset merging
- Attribute selection
- Attribute conversion
- Avg,min,max,sum calculations
- Outlier detection preparation

5.1 Ανάλυση σετ δεδομένων

Το αρχικό σετ δεδομένων είχε την παρακάτω μορφή:

```

1 Date;Time;Global_active_power;Global_reactive_power;Voltage;Global_intensity;Sub_metering_1;Sub_metering_2;Sub_metering_3
2 16/12/2006;17:24:00;4.216;0.418;234.840;18.400;0.000;1.000;17.000
3 16/12/2006;17:25:00;5.360;0.436;233.630;23.000;0.000;1.000;16.000
4 16/12/2006;17:26:00;5.374;0.498;233.290;23.000;0.000;2.000;17.000
5 16/12/2006;17:27:00;5.388;0.502;233.740;23.000;0.000;1.000;17.000
6 16/12/2006;17:28:00;3.666;0.528;235.680;15.800;0.000;1.000;17.000
7 16/12/2006;17:29:00;3.520;0.522;235.020;15.000;0.000;2.000;17.000
8 16/12/2006;17:30:00;3.702;0.520;235.090;15.800;0.000;1.000;17.000
9 16/12/2006;17:31:00;3.700;0.520;235.220;15.800;0.000;1.000;17.000
10 16/12/2006;17:32:00;3.668;0.510;233.990;15.800;0.000;1.000;17.000
11 16/12/2006;17:33:00;3.662;0.510;233.860;15.800;0.000;2.000;16.000
12 16/12/2006;17:34:00;4.448;0.498;232.860;19.600;0.000;1.000;17.000
13 16/12/2006;17:35:00;5.412;0.470;232.780;23.200;0.000;1.000;17.000
14 16/12/2006;17:36:00;5.224;0.478;232.990;22.400;0.000;1.000;16.000
15 16/12/2006;17:37:00;5.268;0.398;232.910;22.600;0.000;2.000;17.000
16 16/12/2006;17:38:00;4.054;0.422;235.240;17.600;0.000;1.000;17.000
17 16/12/2006;17:39:00;3.384;0.282;237.140;14.200;0.000;0.000;17.000
18 16/12/2006;17:40:00;3.270;0.152;236.730;13.800;0.000;0.000;17.000
19 16/12/2006;17:41:00;3.430;0.156;237.060;14.400;0.000;0.000;17.000
20 16/12/2006;17:42:00;3.266;0.000;237.130;13.800;0.000;0.000;18.000
21 16/12/2006;17:43:00;3.728;0.000;235.840;16.400;0.000;0.000;17.000
22 16/12/2006;17:44:00;5.894;0.000;232.690;25.400;0.000;0.000;16.000
23 16/12/2006;17:45:00;7.706;0.000;230.980;33.200;0.000;0.000;17.000
24 16/12/2006;17:46:00;7.026;0.000;232.210;30.600;0.000;0.000;16.000

```

Εικόνα 1 Αρχική, raw μορφή του σετ (household_power_consumption.txt)

Οι τιμές στο dataset διαχωρίζονται με semicolon και είναι συνολικά 9. Επίσης το σετ περιέχει missing values τα οποία αναπαρίστανται με απουσία τιμής. Όταν μετατρέψαμε το αρχείο σε csv τα missing values αντικαταστάθηκαν με αγγλικά ερωτηματικά '?'. Διαπιστώσαμε όμως ότι τα missing values του τελευταίου χαρακτηριστικού δεν είχαν ερωτηματικό, το οποίο είναι λογικό:

«a missing value is represented by the absence of value between two consecutive semi-colon attribute separators.»

Αποφασίσαμε λοιπόν καταρχήν να αντικαταστήσουμε τα semi-colons με commas ώστε να έχουμε μία across the board συμβατότητα (Weka – Matlab – Excel) χωρίς περεταίρω παραμετροποίηση. Επιπλέον αποφασίσαμε να συμπληρώσουμε τα missing missing values του τελευταίου χαρακτηριστικού. Έτσι δώσαμε την παρακάτω εντολή

```
/> ruby csv_reader.rb -t household_power_consumption.csv
```

και το dataset ήταν έτοιμο.

Το επόμενο πρόβλημα που αντιμετωπίσαμε ήταν ότι το dataset δεν χωρούσε λόγω μεγέθους ούτε στο Weka ούτε στο Excel. Συνεπώς αποφασίσαμε αρχικά να το χωρίσουμε με βάση την λογική σε:

- 5 Χρόνια

```
/> ruby csv_reader.rb -x year household_power_consumption.csv
```

- 12 Μήνες

```
/> ruby csv_reader.rb -x month household_power_consumption.csv
```

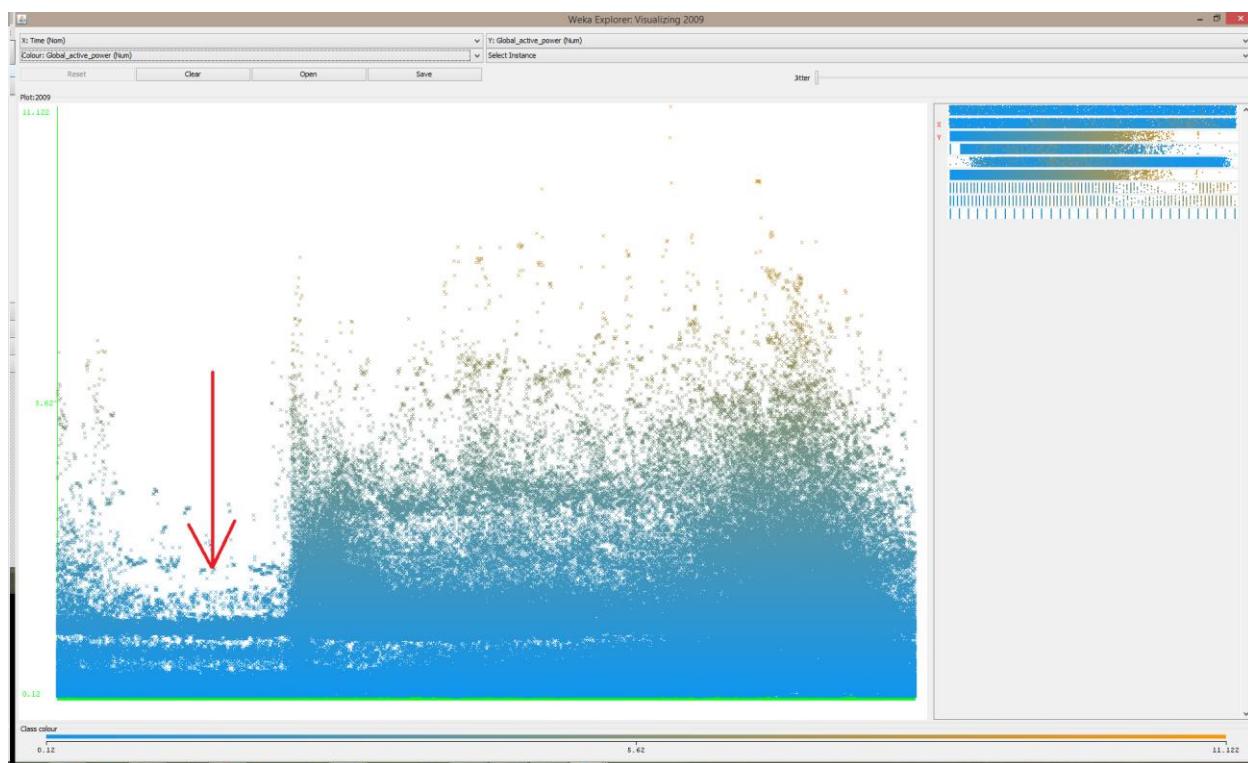
- 4 Εποχές

```
/> ruby csv_reader.rb -x season household_power_consumption.csv
```

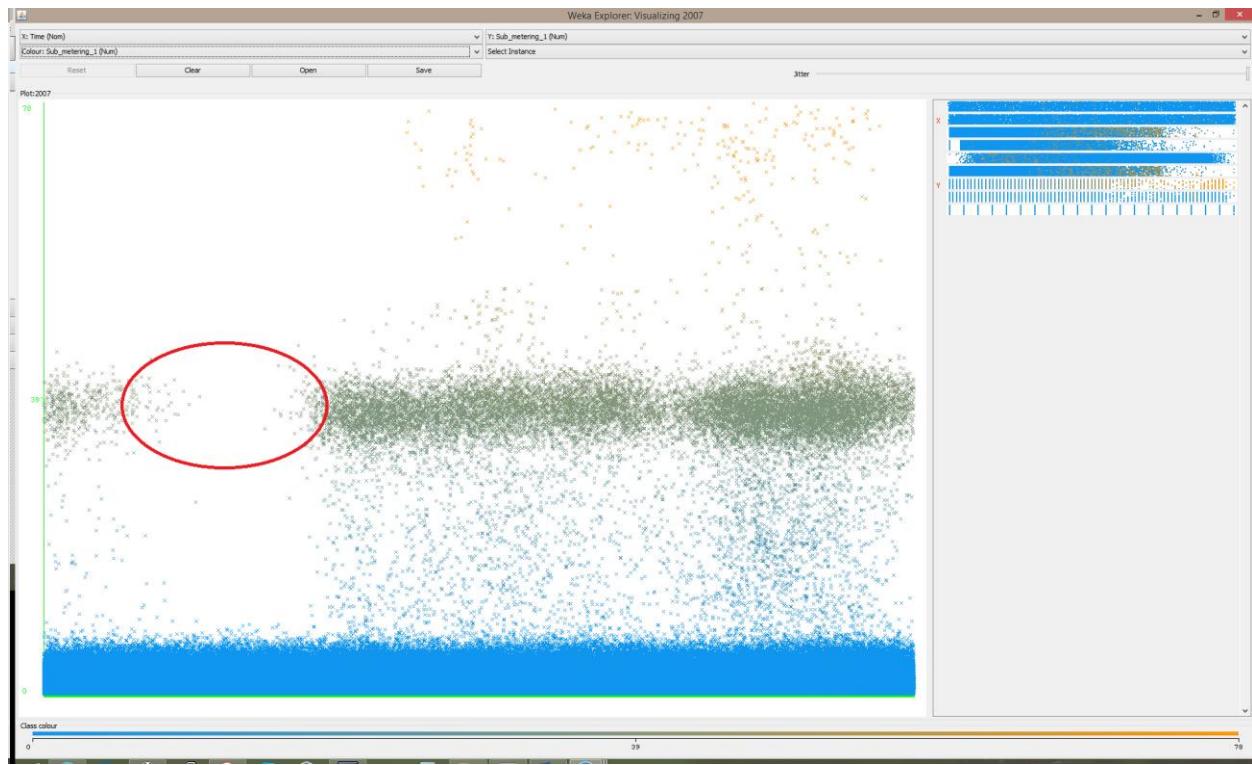
Τα 21 dataset αυτά αποτέλεσαν τον κορμό της βασικής μας ανάλυσης και περαιτέρω δειγματοληψία ή εφαρμογή μετρικών γινόταν κυρίως σε αυτά.

Στην συνέχεια θα επισημάνουμε τις παρατηρήσεις μας στις διάφορες απεικονίσεις δεδομένων.

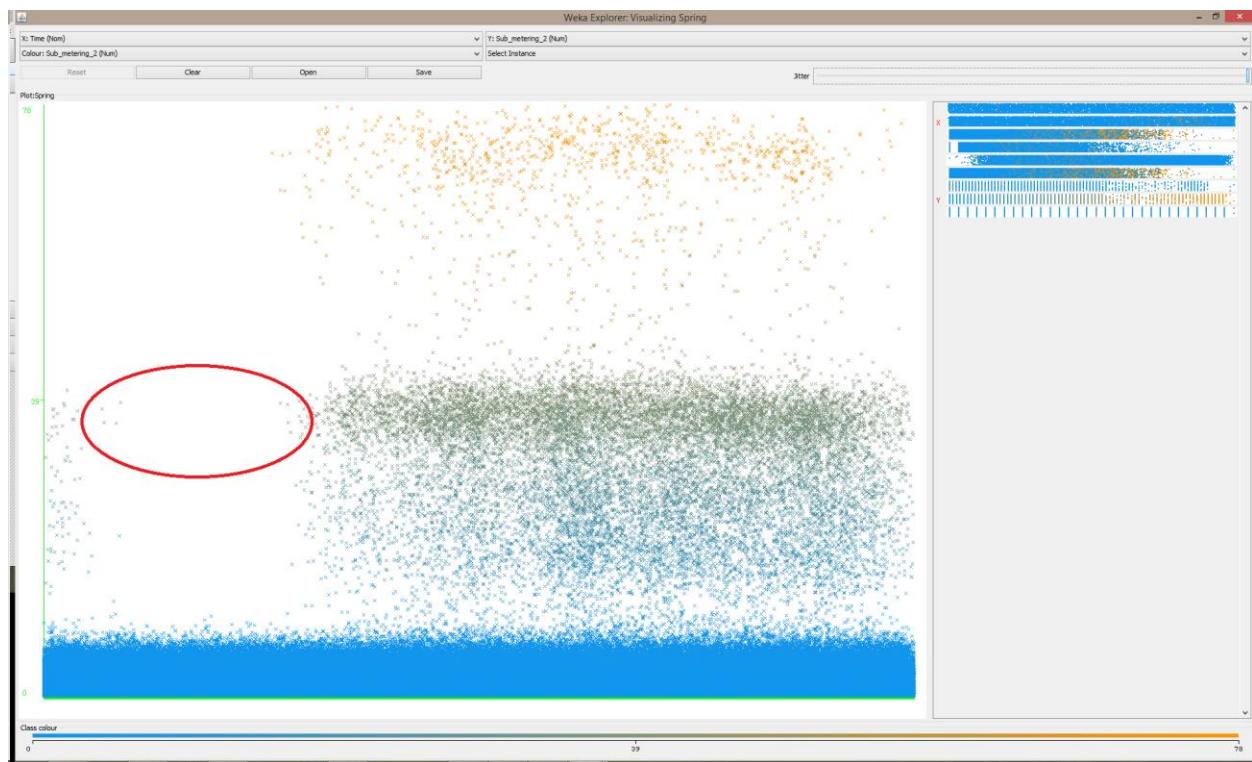
- To *Global_active_power* και τα *Sub_metering_1,2* παρουσιάζουν αισθητά μειωμένη έως καθόλου κατανάλωση, όλα τα χρόνια και όλες τις εποχές το χρονικό διάστημα 2 το πρωί με 6 το πρωί. Ενδεικτικά παραθέτουμε 3 απεικονίσεις που αναδεικνύουν αυτό το φαινόμενο:



Εικόνα 2 2009 – Time - *Global_active_power*

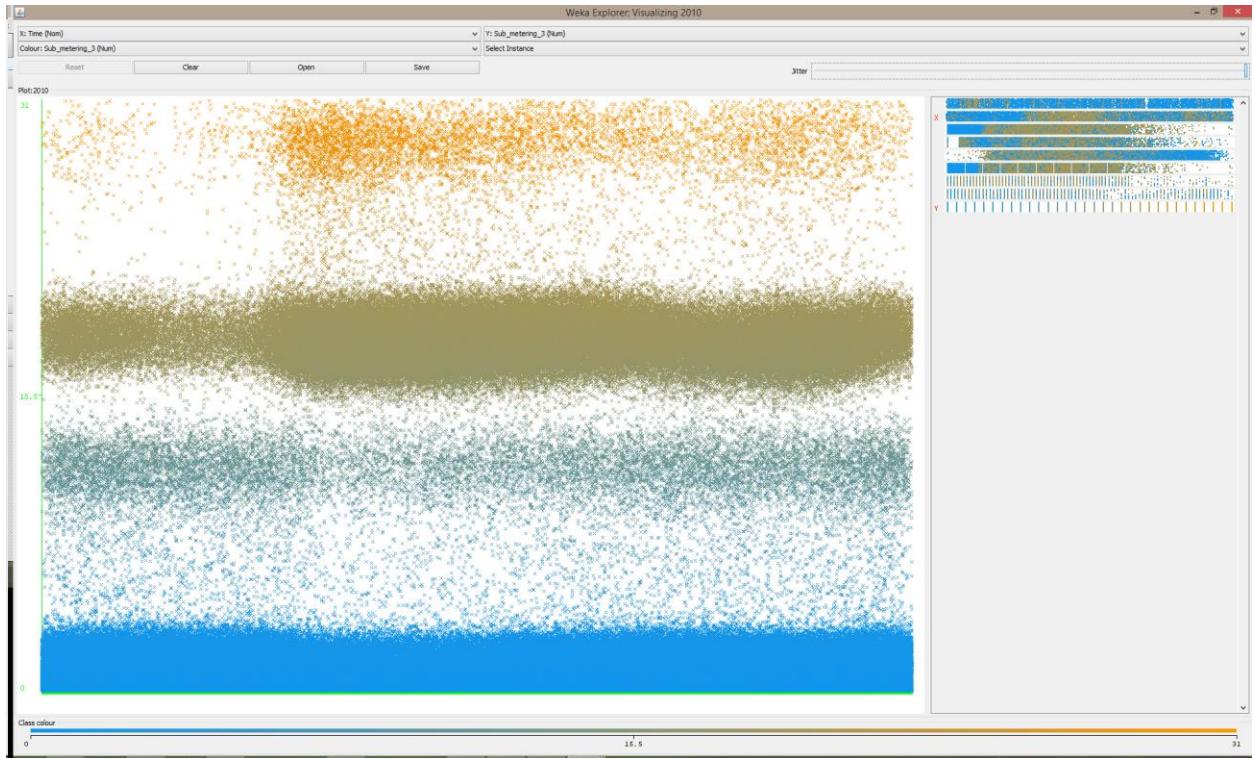


Εικόνα 3 2007-Time-Sub_metering_1



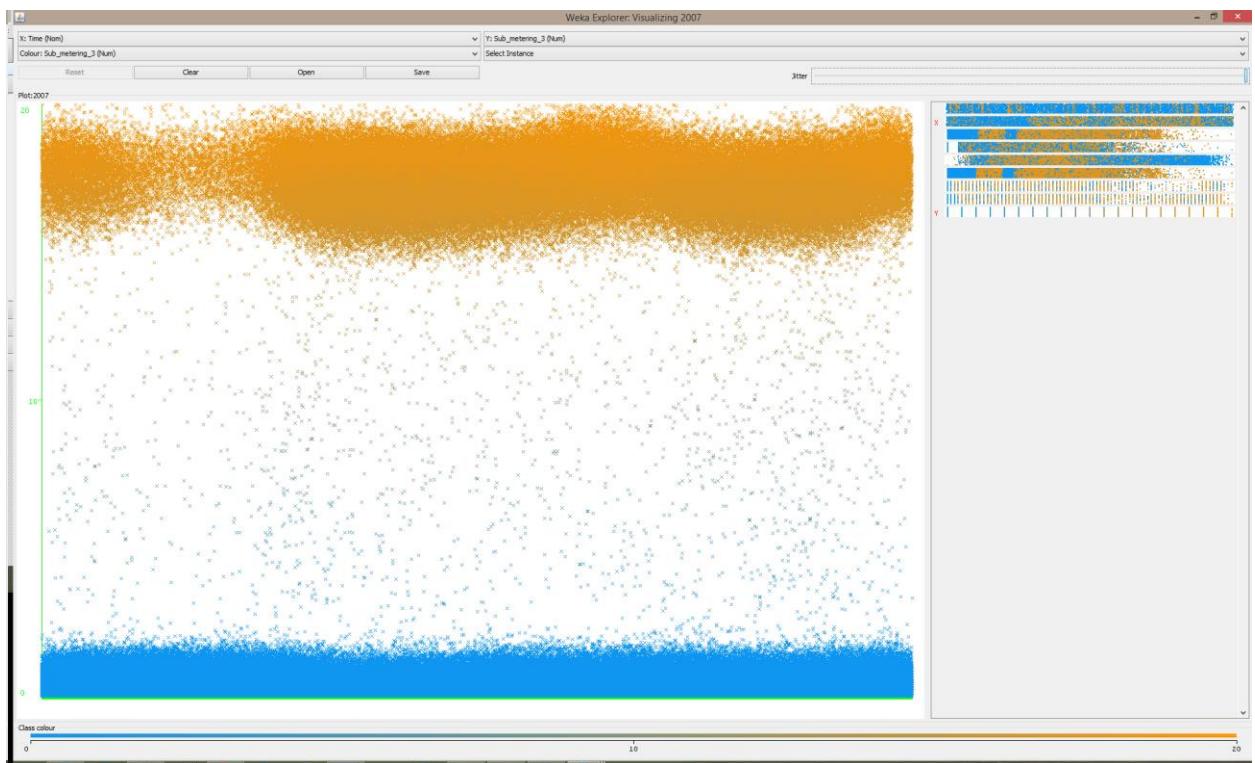
Εικόνα 4 Spring-Time-Sub_metering_2

- Για το *Sub_metering_3* γνωρίζουμε ότι αποτελείται από δύο συσκευές το κλιματιστικό και το θερμοσίφωνο. Μία ενδεικτική απεικόνιση του *Sub_metering_3* είναι η παρακάτω:

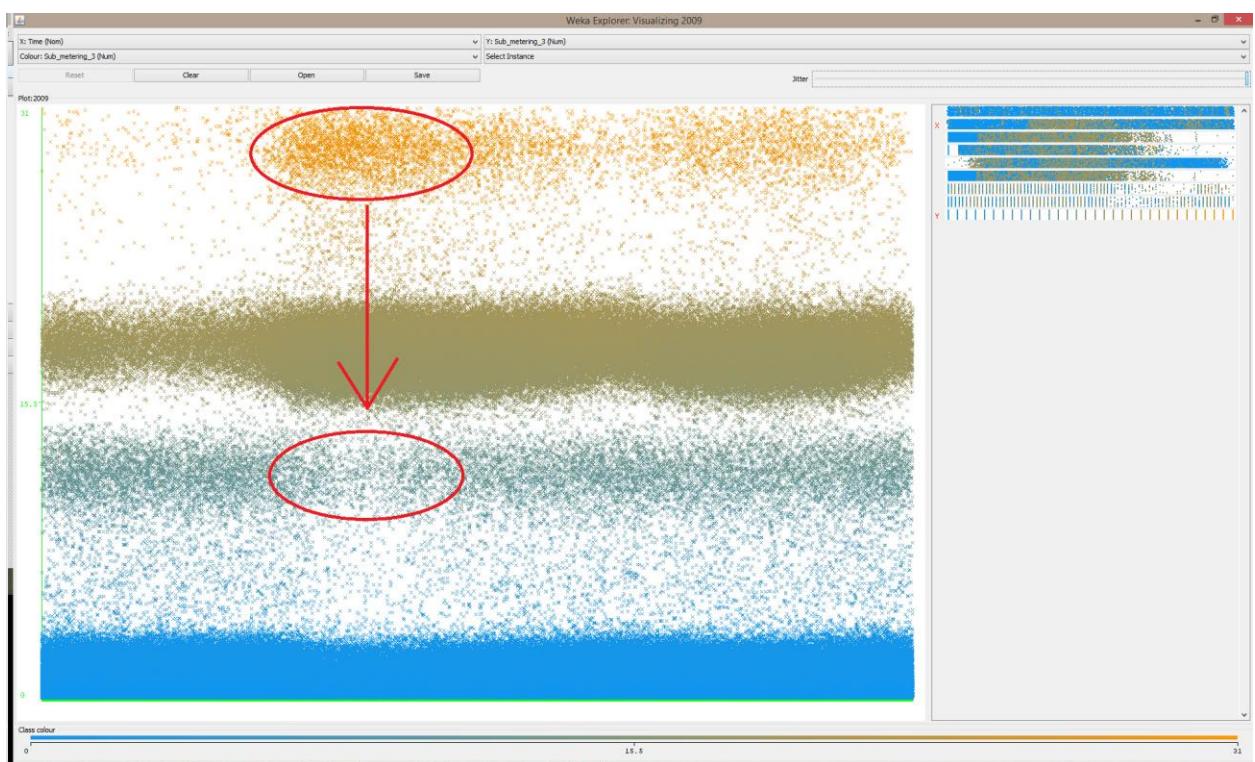


Εικόνα 5 2010-Time- *Sub_metering_1*

Έχοντας από ανάλυση καταλήξαμε στο συμπέρασμα ότι η πυκνή μπεζ ζώνη αποτελεί την μία συσκευή και οι δύο αραιές αποτελούν την άλλη συσκευή. Ο ισχυρισμός αυτός επιβεβαιώνεται από τις παρακάτω απεικονίσεις:



Εικόνα 6 2007-Time-Sub_metering_3



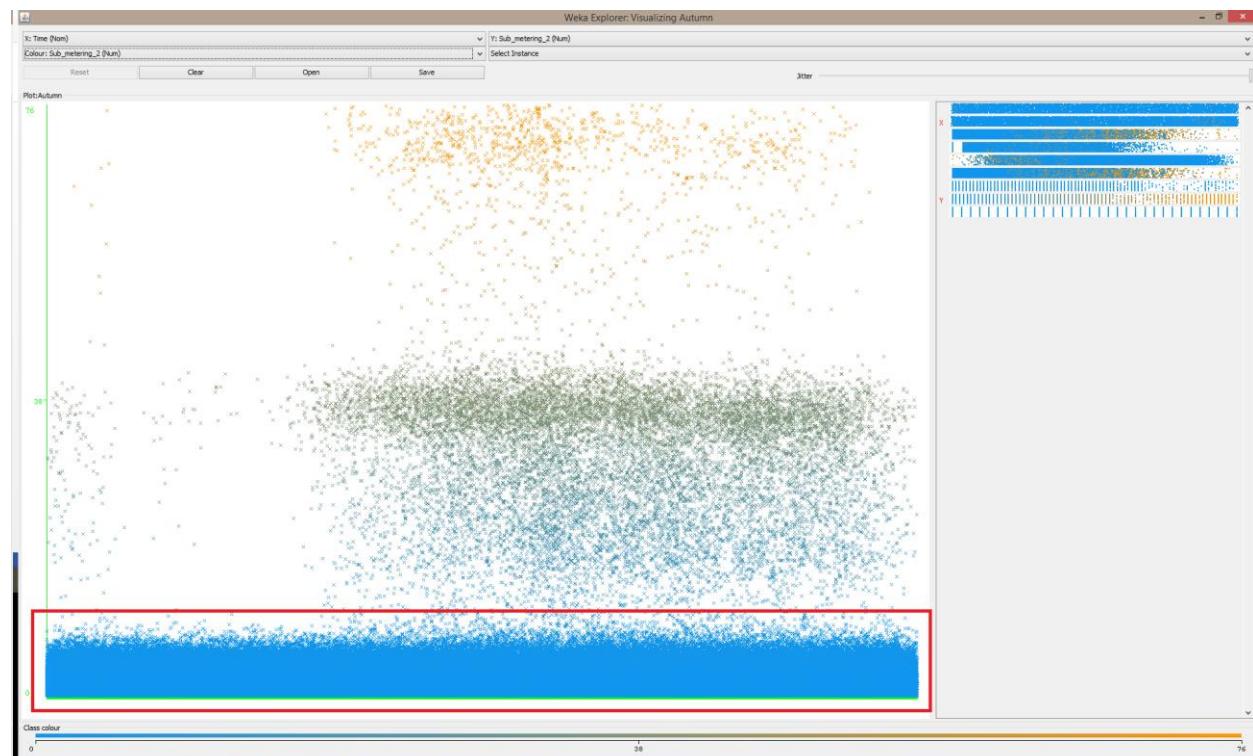
Εικόνα 7 2009-Time-Sub_metering_3

Στην πρώτη εικόνα παρατηρούμε ότι υπάρχει μόνο η πυκνή περιοχή και όχι οι δύο αραιές. Αυτό σημαίνει ότι η μία συσκευή το 2007 δεν υπήρχε. Το ότι οι δύο περιοχές αποτελούν μία συσκευή, αναδεικνύεται στην δεύτερη εικόνα, όπου η πορτοκαλί πυκνή περιοχή συμπληρώνει ύποπτα το αραιό κομμάτι της κυανής.

Επίσης κάναμε τον ισχυρισμό, χωρίς αυτό να επηρεάζει την ανάλυση, ότι σχεδόν σίγουρα η πυκνή περιοχή αποτελεί το θερμοσίφωνο και η αραιά το κλιματιστικό.

- Και τα 3 *Sub_metering* περιέχουν μία περιοχή «idle λειτουργίας»

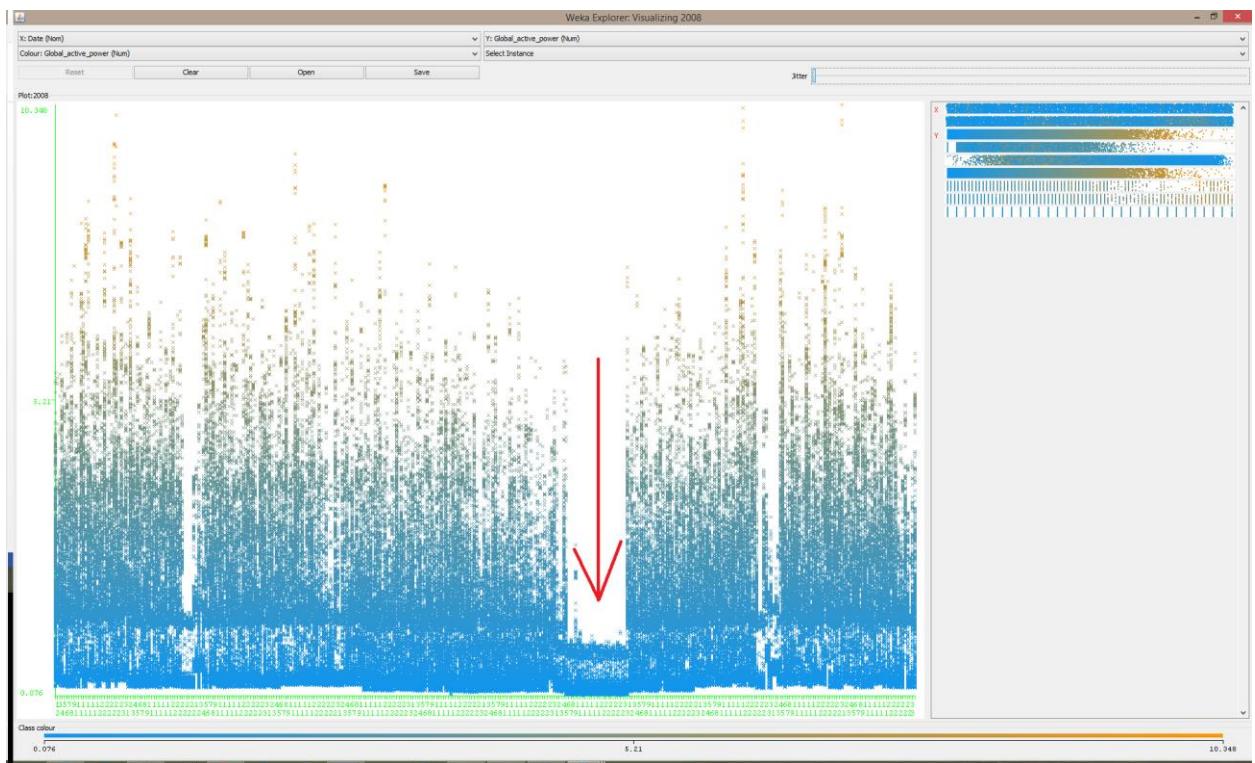
Πρόκειται για την χαρακτηριστική μπλε πυκνή ζώνη στην κάτω περιοχή των γραφημάτων *Time-Sub_metering_1,2,3*:



Εικόνα 8 Autumn-Time-Sub_metering_2

Οι τιμές αυτές γενικώς δεν προσφέρουν κάποια ουσιαστική πληροφορία χρονικά και γενικώς αντιμετωπίζονται με αφαίρεση. Κυμαίνονται ανάμεσα στα 0 και στα 2 W/h.

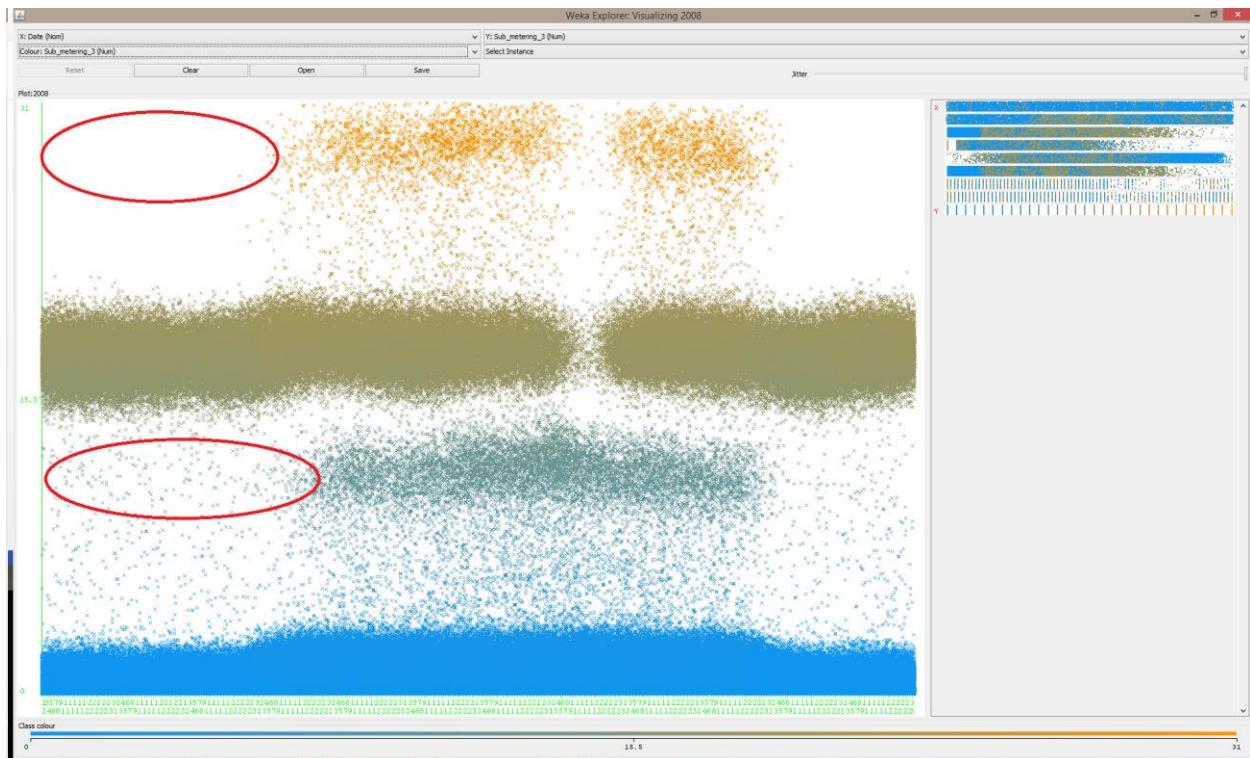
- Το *Global_active_power* και τα 3 *Sub_metering* στην διάρκεια του έτους εμφανίζουν πτώση στην κατανάλωση για όλα τα χρόνια την περίοδο του καλοκαιριού και κυρίως τους μήνες Ιούλιο – Αύγουστο. Αυτό φαίνεται με έμφαση το 2008:



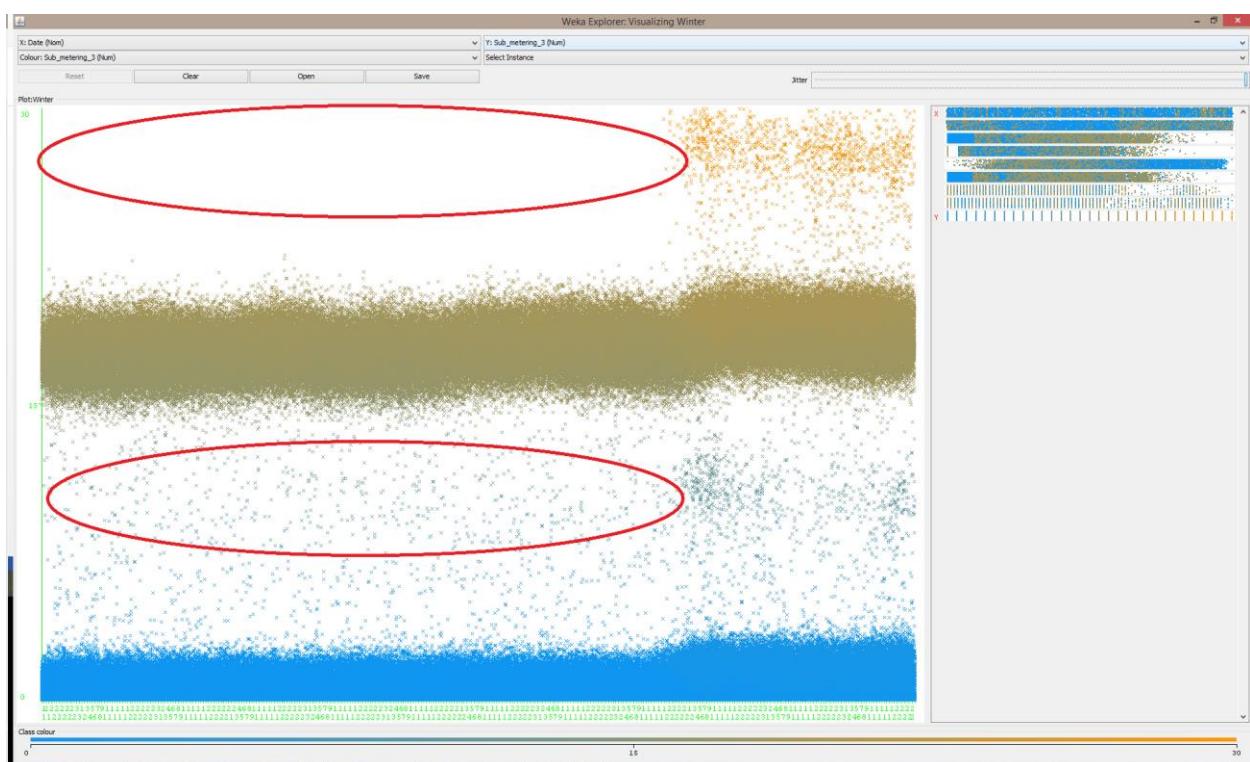
Εικόνα 9 2008-*Date*-*Global_active_power*

Η καταναλωτική αυτή συμπεριφορά πολύ πιθανόν να οφείλεται στο γεγονός ότι οι κάτοικοι του σπιτιού πηγαίνουν διακοπές.

- Στο *Sub_metering_3*, το χειμώνα γενικώς δεν λειτουργεί η μία από τις δύο συσκευές:

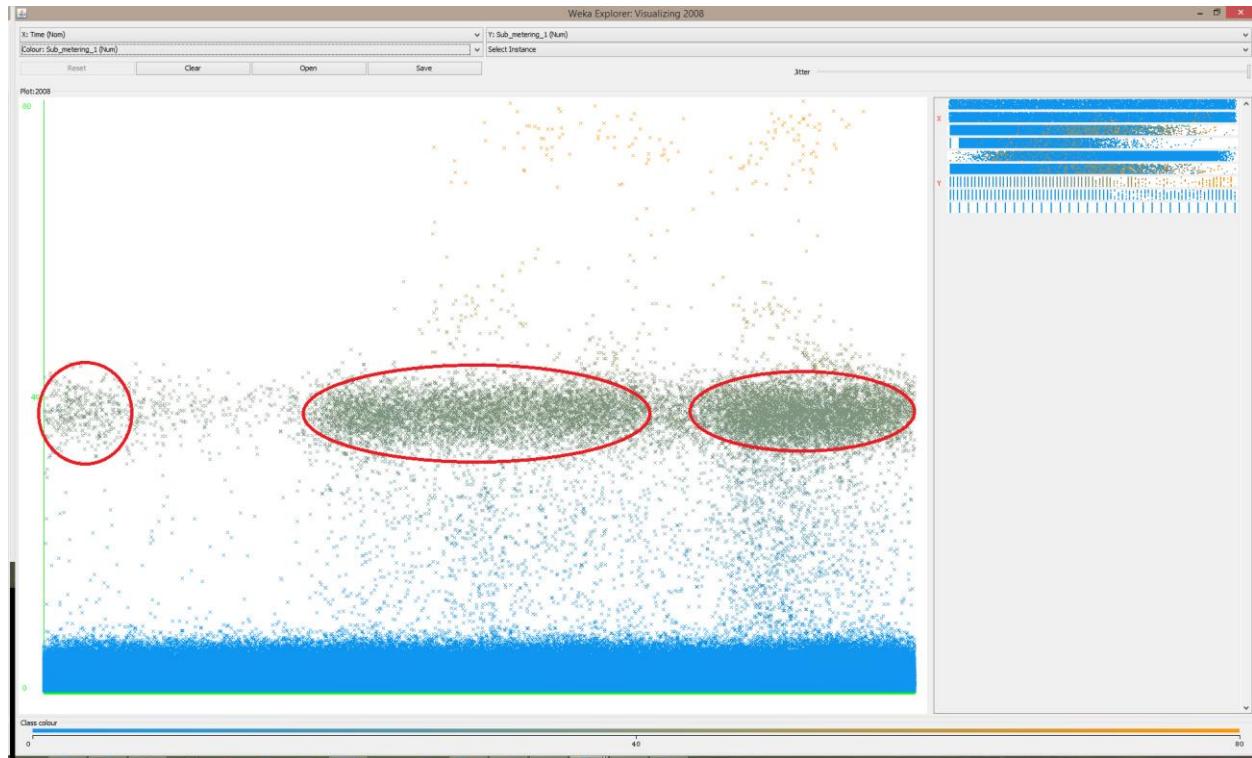


Εικόνα 10 2008-Date-Sub_metering_3



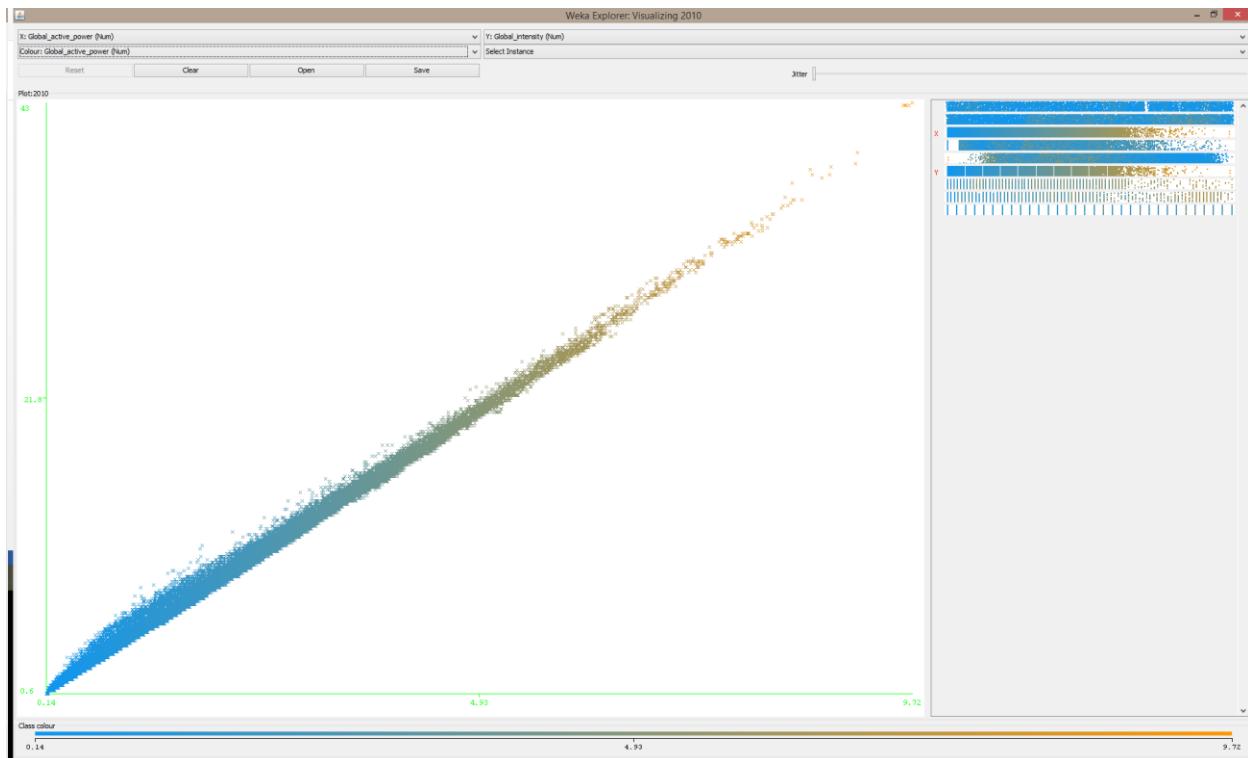
Εικόνα 11 Winter-Date-Sub_metering_3

- Τα *Sub_metering_1,2* παρουσιάζουν ωράρια λειτουργίας στο 24ωρο:
1. 12 το βράδυ με 2 το πρωί
 2. 7 το πρωί με 4 το μεσημέρι
 3. 6 το απόγευμα με 12 παρά το βράδυ



Εικόνα 12 – 2008 - *Sub_metering_1* - Ζώνες

- Το *Global_active_power* και το *Global_intensity* έχουν γραμμική σχέση:



Εικόνα 13 2010-*Global_active_power*-*Global_intensity*

Το ότι έχουν γραμμική συσχέτιση σημαίνει ότι δεν παρέχουν κάποια επιπλέον πληροφορία όταν βρίσκονται ταυτόχρονα στο dataset.

5.2 Εξωκείμενες τιμές

Η διαδικασία εντοπισμού και αφαίρεσης εξωκείμενων τιμών μας απασχόλησε επίσης σε μεγάλο βαθμό. Εξετάσαμε και την επιλογή του one-class svm και την επιλογή του knn.

1. One-class SVM

Η διαδικασία αφαίρεσης εξωκείμενων τιμών με την τεχνική one-class svm λειτουργεί ως εξής:

- Αναθέτουμε σε όλες τις εγγραφές μία nominal τιμή, π.χ. NOT-OUTLIER
- Κάνουμε ταξινόμηση με 1 κλάση με τον αλγόριθμο one-class svm για συγκεκριμένα νι και gamma.

Η προετοιμασία του dataset για one-class svm γίνεται με την παρακάτω εντολή στον csv_reader.rb:

```
ruby csv_reader -o weka <dataset>
```

Το αποτέλεσμα που παίρνουμε ύστερα από την εκτέλεση του αλγορίθμου είναι ένα ποσοστό unclassified εγγραφών τις οποίες το svm δεν μπορεί να ενσωματώσει στην γεωμετρία των εγγραφών της nominal τιμής. Οι μη ταξινομημένες εγγραφές αποτελούν τα outliers του σετ δεδομένων. Το πρόβλημα εδώ είναι ο ορισμός των παραμέτρων νι και gamma. Συγκεκριμένα για το νι βρήκαμε σε πολλές πηγές ότι:

“...the trade-off parameter ν is an upper bound on the fraction of outliers (training points outside the estimated region) and a lower bound on the fraction of support vectors... The optimal γ minimizes

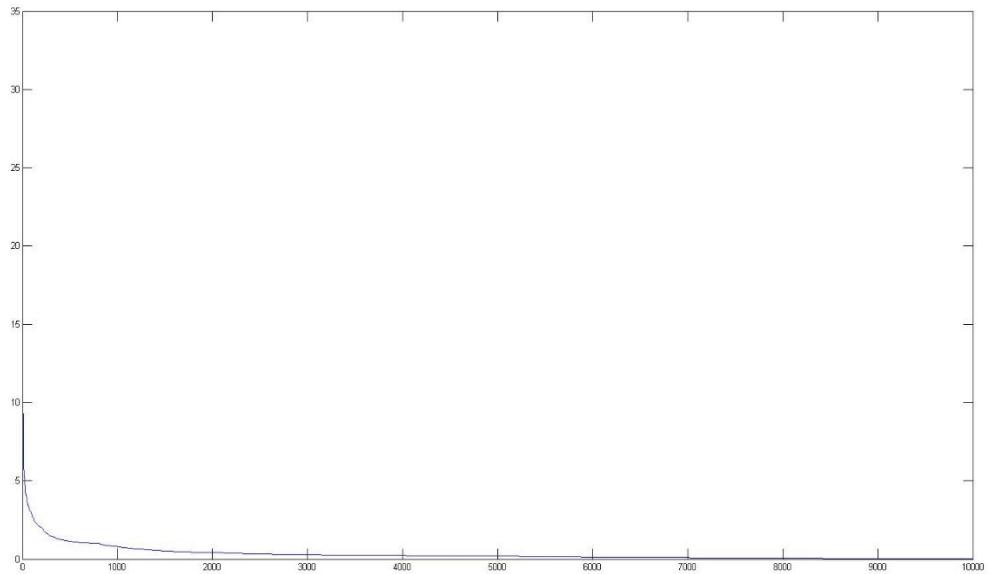
$$\sqrt{(\nu - F_{sv})^2 - (\nu - F_{out})^2}^1$$

Χωρίς να θέλουμε να εφαρμόσουμε αποκλειστικά τον παραπάνω τύπο, το πρόβλημά μας ήταν ότι δεν γνωρίζαμε εξ’ ορισμού το ποσοστό των outlier στο dataset (και φυσικά το πλήθος των support vectors). Συνεπώς ο ορισμός της παραμέτρου που που λειτουργεί ως άνω όριο στο πλήθος των outlier ενέχει ένα βαθμό αυθαιρεσίας. Με διάφορες δοκιμές στην περιοχή 0.01 – 0.1 διαπιστώσαμε με επισκόπηση στο Excel ότι οι εγγραφές που βρίσκει ως outliers οντως έχουν ιδιομορφίες. Παρ’ όλ’ αυτά η αποθήκευση τους από το Weka και η αφαίρεσή τους από το αρχικό dataset είναι μία τεχνικά δύσβατη διαδικασία: το Weka 3.6 δεν υποστηρίζει αποθήκευση prediction και το 3.7 που υποστηρίζει δεν έχει libsvm στους classifiers. Αυτά σε συνδυασμό με τον βαθμό προσωπικής ελευθερίας στην επιλογή των παραμέτρων, μας οδήγησε στην εξέταση της δεύτερης επιλογής.

2. knn-distance

Η διαδικασία αφαίρεσης εξωκείμενων τιμών με την μέθοδο του knn-distance είναι η εξής: αρχικά, βρίσκουμε για κάθε σημείο i την απόστασή του από το k -στο κοντινότερο σημείο του, έστω $d(i)$. Στην συνέχεια, ταξινομούμε το διάνυσμα $d(i)$ κατά φθίνουσα σειρά και επιλέγουμε με βάση κάποιον κανόνα τα n πρώτα σημεία. Παράμετροι που πρέπει να καθορίσει ο χρήστης είναι η τιμή του k και ο κανόνας απόφασης της τιμής του n . Από την εκφώνηση της εργασίας, έγιναν δοκιμές με τις τιμές $k=1\dots 5$, και τελικά επιλέχθηκε η τιμή 5. Όσον αφορά την επιλογή του n , εξετάσθηκαν 2 προσεγγίσεις. Η πρώτη έχει να κάνει με την έννοια του “γονάτου” το οποίο εμφανίζεται σε ένα γράφημα του $d(i)$ συναρτήσει των σημείων, όπως φαίνεται παρακάτω:

¹ <http://www.joint-research.org/wp-content/uploads/2011/07/lukashevich2009Using-One-class-SVM-Outliers-Detection.pdf>



Εικόνα 14 knn-distance graph

Η έννοια αυτή είναι δύσκολο να ορισθεί με σαφήνεια, αλλά παρόλα αυτά μπορεί να πει κανείς ότι το "γόνατο" βρίσκεται στο σημείο όπου η παράγωγος της συνάρτησης $d(i)$ (η ακριβολογώντας η συνάρτηση διαφορών $y(i) = d(i) - d(i-1)$) παρουσιάζει μία απότομη αύξηση. Αυτή η απότομη αύξηση θα μπορούσε να ορισθεί με στατιστικούς όρους, ως π.χ. η τιμή στην οποία η παράγωγος γίνεται μεγαλύτερη από την μέση τιμή της συν την διακύμανσή της, $\mu + \sigma$.

Η δεύτερη απλούστερη προσέγγιση είναι να θεωρήσει κανείς ότι τα outliers αποτελούν ένα σταθερό ποσοστό του dataset π.χ 1%, και να αφαιρέσει το ποσοστό αυτό των outliers από την αρχή του $d(i)$.

Η πρώτη προσέγγιση δοκιμάστηκε και έδινε γενικά καλά αποτελέσματα, αλλά τελικά δεν επιλέχθηκε, καθώς και αυτή τελικά θα έπρεπε να έχει μια στατιστική παραμετροποίηση του τύπου "το γόνατο βρίσκεται σε μία τιμή ρ φορές την διακύμανση + την μέση τιμή", και άρα να είναι ισοδύναμη με την δεύτερη.

Η πρώτη προσέγγιση βρίσκεται στο αρχείο knn_outliers_der.m, ενώ η δεύτερη που ακολουθήθηκε τελικά στο knn_outliers_stat.m .

3. Κατακλείδα

Τελικά και με τις δύο επιλογές πρέπει να αποφασίζουμε εμείς σε ένα σημαντικό βαθμό το πλήθος των outliers με τον έναν ή τον άλλον τρόπο. Η δεύτερη επιλογή όμως διαπιστώθηκε ότι μπορεί να αυτοματοποιηθεί πλήρως με την βοήθεια του csv_reader. Δηλαδή η προετοιμασία του dataset για αφαίρεση εξωκείμενων, η παραμετροποίηση των Matlab script, ο εντοπισμός των outlier μέσω Matlab και η αφαίρεσή τους από το dataset συνοψίζονται στην εντολή

```
/> ruby csv_reader.rb -o matlab <dataset>
```

Έτσι αποφασίσαμε να ακολουθήσουμε τον δρόμο του knn-distance outlier detection για λόγους αυτοματοποίησης.

6. Clustering

Ομαδοποίηση: Ορισμός, Τεχνικές, Αλγόριθμοι

Ομαδοποίηση ονομάζεται η διαδικασία μηχανικής μάθησης που στοχεύει στην μη επιβλεπόμενη ταξινόμηση. Ενώ δηλαδή στην επιβλεπόμενη ταξινόμηση προσφέρονται εξωτερικά στον αλγόριθμο οι κλάσεις ή ομάδες, στην ομαδοποίηση ο αλγόριθμος καλείται ο ίδιος να βρει τις ομάδες που δημιουργούνται, χρησιμοποιώντας μόνο την πληροφορία που δίνεται από τις τιμές του dataset. Ο γενικός στόχος ενός αλγορίθμου ομαδοποίησης είναι να δημιουργήσει ομάδες οι οποίες είναι όσο το δυνατόν “πυκνότερες” στο εσωτερικό τους, και όσο το δυνατόν πιο απομακρυσμένες μεταξύ τους. Αμέσως καταλαβαίνουμε ότι για να πετύχει τον στόχο του, ο αλγόριθμος πρέπει να διαθέτει μια μετρική ομοιότητας ή διαφοράς μεταξύ των στοιχείων, όπως και κάποια κριτήρια που θα καθορίζουν πόσο καλή είναι η συνοχή των ομάδων και ο διαχωρισμός μεταξύ τους.

Γενικές τεχνικές για καλή ομαδοποίηση δεν υπάρχουν. Η ποιότητα της ομαδοποίησης καθορίζεται από πολλούς παράγοντες, όπως η ο τύπος των clusters, η επιστημονική περιοχή στην οποία θα εφαρμόσουμε μηχανική μάθηση (π.χ. ιατρική), η χρονική και χωρική πολυπλοκότητα των αλγορίθμων, οι υποκειμενικές εκτιμήσεις ειδικών κ.τ.λ.

Σημαντικά είδη clusters είναι οι καλώς διαχωρισμένοι, αυτοί που χαρακτηρίζονται από κάποιο αντιπροσωπευτικό σημείο τους και αυτοί που καθορίζονται με βάση την πυκνότητά τους. Γενικά χαρακτηριστικά που μπορεί να συναντήσει κάποιος σε clusters είναι ότι ακολουθούν μια κατανομή πυκνότητας πιθανότητας, ότι έχουν κάποιο συγκεκριμένο σχήμα, ότι δεν είναι καλά διαχωρισμένοι μεταξύ τους κ.α.

Χαρακτηριστικά του dataset που επηρεάζουν την διαδικασία ομαδοποίησης είναι οι διαστάσεις των attributes, το μέγεθος του dataset, εξωκείμενες τιμές, ανάγκη για κανονικοποίηση, μαθηματικές ιδιότητες των χαρακτηριστικών, ώστε οι μετρικές απόστασης να έχουν νόημα κ.λ.π.

Οι διάφοροι αλγόριθμοι ομαδοποίησης μπορούν να χωρισθούν σε διάφορες κατηγορίες, ανάλογα με την προσέγγιση που ακολουθούν για να δημιουργήσουν τις ομάδες. Οι τρεις κατηγορίες που χρησιμοποιήθηκαν στην εργασία είναι οι αλγόριθμοι Διαχωρισμού, οι Ιεραρχικοί και οι Πυκνωτικοί.

Οι αλγόριθμοι Διαχωρισμού αναθέτουν κάθε σημείο του dataset σε ένα cluster, δηλαδή χωρίζουν τον χώρο των δειγμάτων σε σαφώς καθορισμένες περιοχές. Αντιπρωσοπευτικό παράδειγμα τέτοιου είδους αλγορίθμου είναι ο kmeans. Για τον kmeans πρέπει να ορισθεί μία μετρική απόστασης ανάμεσα στα σημεία (συνήθως ευκλείδεια), καθώς και ένας ορισμός για το τί σημαίνει αντιπροσωπευτικό δείγμα ενός cluster (συνήθως η μέση τιμή για αριθμητικά δεδομένα). Δηλαδή, ο kmeans είναι Prototype-based. Έχοντας τους παραπάνω ορισμούς, ο αλγόριθμος επιλέγει κάποια σημεία ως κέντρα, αναθέτει σε αυτά τα σημεία που τους αντιστοιχούν, υπολογίζει τα νέα κέντρα, αναθέτει εκ νέου σημεία σε αυτά κοκ., έως ότου να επέλθει σύγκλιση.

Μειονεκτήματα του kmeans είναι

- A) Διαφορετική αρχικοποίηση των σημείων οδηγεί σε διαφορετικές ομαδοποιήσεις. (Υπάρχουν διάφοροι ευρετικοί τρόποι για την αντιμετώπιση του προβλήματος)
- B) Διαφορετικές πυκνότητες ή μεγέθη ομάδων επηρεάζουν πολύ το αποτέλεσμα
- Γ) Δεν ανιχνεύει μη σφαιρικού σχήματος ομάδες
- Δ) Επηρεάζεται πολύ από εξωκείμενες τιμές
- Ε) Πρέπει να καθορισθεί εξαρχής ο αριθμός των clusters.

Πλεονεκτήματά του είναι η ευκολία υλοποίησής του, καθώς και το ότι έχει ισχυρή μαθηματική βάση, καθώς αποδεικνύεται ότι το αποτέλεσμά του θα είναι πάντα ένα τοπικό ελάχιστο του αθροίσματος των τετραγώνων των αποστάσεων μεταξύ των σημείων.

Οι Ιεραρχικοί αλγόριθμοι χωρίζονται σε 2 κατηγορίες: σε αυτούς που διαιρούν συνεχώς τον δειγματοχώρο έως ότου καταλήξουν σε στοιχειώδη δείγματα, είτε σε αυτούς που ξεκινούν από στοιχειώδη δείγματα και τα συνενώνουν δημιουργώντας επαναληπτικά όλο και μεγαλύτερα clusters, έως ότου καλύψουν ολόκληρο τον δειγματοχώρο. Εμείς θα ασχοληθούμε με την δεύτερη προσέγγιση (agglomerative), που είναι και η πιο συνηθισμένη. Κάθε agglomerative ιεραρχικός αλγόριθμος βασίζεται στην ίδια κεντρική ιδέα: υπολόγισε ομοιότητα ή απόσταση ανάμεσα σε όλους τους clusters, συνένωσε τους δύο κοντινότερους, υπολόγισε ξανά την απόσταση ανάμεσα σε όλους τους clusters, συνένωσε κτλπ. Έως ότου μείνει μόνο μία ομάδα. Η αρχικοποίηση είναι όλα τα σημεία του dataset να αποτελούν μία ομάδα. Το βασικό πρόβλημα σε αυτή την διαδικασία είναι, πως θα ορίσουμε απόσταση μεταξύ clusters, και αυτός ο ορισμός είναι αυτό που ουσιαστικά διαφοροποιεί τους ιεραρχικούς αλγορίθμους μεταξύ τους. Οι συνηθέστερες προσεγγίσεις είναι να θεωρήσουμε απόσταση μεταξύ 2 clusters την απόσταση των κοντινότερων σημείων τους – single link (προφανώς έχουμε ορίσει κάποια μετρική για απόσταση δειγματοσημείων, πχ ευκλείδεια), την απόσταση των μακρινότερων σημείων τους – complete link και την μέση απόσταση όλων των σημείων μεταξύ των δύο clusters - group average. Υπάρχουν φυσικά και άλλες μέθοδοι, όπως αυτή του Ward, η οποία μετράει την αύξηση του SSE κατά την ένωση 2 clusters, έχοντας ορίσει αρχικά την έννοια του κεντρικού-αντιπροσωπευτικού σημείου ενός cluster.

Κάθε ορισμός απόστασης clusters έχει και τα δικά του πλεονεκτήματα και μειονεκτήματα, πχ το complete link συμπεριφέρεται καλά ακόμη και όταν υπάρχουν εξωκείμενες τιμές, αλλά τείνει να διασπά μεγάλα clusters σε μικρότερα.

Γενικά οι ιεραρχικοί αλγόριθμοι έχουν το πλεονέκτημα, του ότι μπορούν να χρησιμοποιηθούν τόσο για μια ιεραρχική ταξινόμηση δεδομένων, όσο και για ομαδοποίηση, χωρίς να είναι απαραίτητο ν α καθοριστεί εκ των προτέρων ο αριθμός των clusters. Παρατηρώντας την απόσταση στην οποία συνενώθηκε κάθε cluster, μπορούμε βρούμε “με το μάτι” την βέλτιστη ομαδοποίηση. Βασικό μειονέκτημα αυτής της οικογένειας αλγορίθμων είναι η ακριβή πολυπλοκότητα χώρου-χρόνου.

Τελευταία σημαντική κατηγορία είναι οι πυκνωτικοί αλγόριθμοι, με αντιπροσωπευτικότερο παράδειγμα τον DBSCAN. Αυτού του είδους οι αλγόριθμοι ορίζουν την έννοια της πυκνότητας στον δειγματοχώρο, και με βάση αυτήν διαχωρίζουν περιοχές υψηλής πυκνότητας (τα clusters) που διαχωρίζονται από άλλες περιοχές υψηλής πυκνότητας μέσω περιοχών χαμηλής πυκνότητας. Συγκεκριμένα ο DBSCAN έχει 2 παραμέτρους που ορίζονται από τον χρήστη, Eps και MinPts. Κάθε

σημείο που έχει στην Eps-γειτονιά του περισσότερα από MinPts σημεία, θεωρείται core. Αν έχει λιγότερα από MinPts σημεία στην γειτονία του και βρίσκεται στην γειτονία ενός core σημείου θεωρείται border, αλλιώς σε κάθε άλλη περίπτωση θεωρείται noise και διαγράφεται. Τα core σημεία που έχουν αλληλεπικαλυπτόμενες Eps-γειτονιές δημιουργούν τα clusters, ενώ τα border ανατίθενται στα κοντινότερα clusters.

Οι πυκνωτικοί αλγόριθμοι γενικά μπορούν να βρουν ομάδες με πολύπλοκο γεωμετρικό σχήμα, αρκεί να είναι μικρών διαστάσεων, και στο dataset να μην υπάρχουν ομάδες με αρκετά διαφορετική πυκνότητα.

Οι αλγόριθμοι ομαδοποίησης του Weka

Cobweb: Ιεραρχικός αλγόριθμος με πιθανοτική προσέγγιση (υποθέτει στατιστική ανεξαρτησία)

DBSCAN: Πυκνωτικός αλγόριθμος 2 παραμέτρων Eps και MinPts

EM: Πιθανοτική προσέγγιση, υπολογίζει τις παραμέτρους του υποτιθέμενου στατιστικού μοντέλου με επαναληπτικό τρόπο. Μπορεί να δοθεί ο υποτιθέμενος αριθμός των ομάδων ώστε να επιταχυνθεί η διαδικασία

FarthestFirst: Παραλλαγή του Kmeans, τοποθετεί τα κέντρα των ομάδων σε όσο το δυνατόν μεγαλύτερη απόσταση μεταξύ τους. Παράμετροι ο αριθμός των clusters και ένα seed για την τυχαία αρχική επιλογή.

HierarchcalClusterer: Υλοποιεί agglomerative iεραρχικού τύπου αλγορίθμους, με πολλές δυνατότητες επιλογής της μετρικής απόστασης των σημείων και των clusters. Έχει την δυνατότητα κανονικοποίησης των δεδομένων.

OPTICS: Πυκνωτικός αλγόριθμος, παρόμοιος με το DBSCAN, ίδιες παράμετροι.

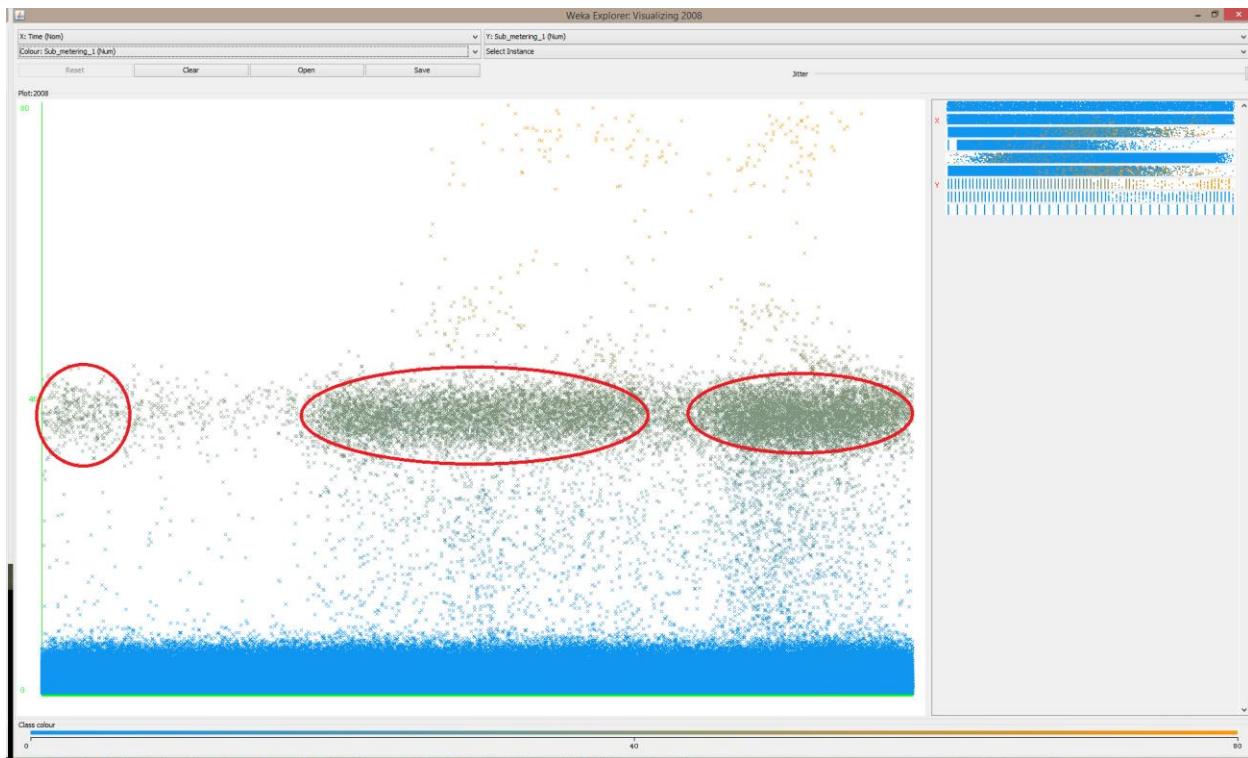
SimpleKmeans: Αλγόριθμος διαχωρισμού, prototype-based, βρίσκει τοπικό ελάχιστο σε objective function, για Euclidean distance αυτή είναι η SSE. Παράμετροι - αριθμός ομάδων, seed για τυχαία αρχικοποίηση, distance metric, κανονικοποίηση των δεδομένων.

Σημείωση για το outlier detection:

Προκειμένου να μην διαμαρτύρεται η csv_read() του Matlab για string values, οι nominal τιμές των Date και Time μετατρέπονται σε numeric με το csv_reader.rb ή αφαιρούνται εντελώς. Ακόμη αφαιρούνται τα missing values μιας και αποτελούν αμελητέο ποσοστό του εκάστοτε dataset.

6.1 2008 – Time - Sub_metering_1

Το Sub_metering_1 αντιστοιχεί στην κουζίνα του σπιτιού η οποία περιέχει κυρίως φούρνο, φούρνο μικροκυμάτων και πλυντήριο πιάτων. Σκοπός μας είναι να βρούμε τουλάχιστον 3 ζώνες ωρών τις οποίες λειτουργεί η κουζίνα, όπως φαίνονται στο διάγραμμα:



Εικόνα 15 – 2008 - Sub_metering_1 – Ζώνες

- Μετά τις 12 το βράδυ μέχρι τις 2 το πρωί
- 7 το πρωί με 4 το μεσημέρι
- 6 το απόγευμα με 12 παρά το βράδυ

Τώρα η υπολογιστική πολυπλοκότητα είναι πολύ μεγάλη (γύρω στις 2^{20} τιμές). Μία επιλογή ήταν να κάνουμε δειγματοληψία. Παρατηρήσαμε όμως ότι η μπλε περιοχή περιέχει τιμές ισχύος, στις οποίες οι συσκευές τις κουζίνας είναι σε idle λειτουργία. Οι τιμές αυτές αποτελούν πολύ μεγάλο ποσοστό του δείγματος (σχεδόν 90%). Συνεπώς **επηρεάζουν σε μεγάλο βαθμό το clustering** μιας και οι υπόλοιπες τιμές είναι ελάχιστες. Διαπιστώσαμε ύστερα από πολλά πειράματα ότι ήταν αδύνατο να χωρίσουμε την κάτω ζώνη idle λειτουργίας από την πάνω και ταυτόχρονα να χωρίσουμε τις χρονικές ζώνες (οριζόντιο και κάθετο clustering). Επίσης οι idle τιμές δεν δίνουν κάποια πληροφορία ως προς την καταναλωτική συμπεριφορά στο 24ωρο. Συνεπώς αποφασίσαμε να αφαιρέσουμε αυτές και 10% εξωκείμενες. Δώσαμε λοιπόν τις παρακάτω εντολές στον csv_reader:

```

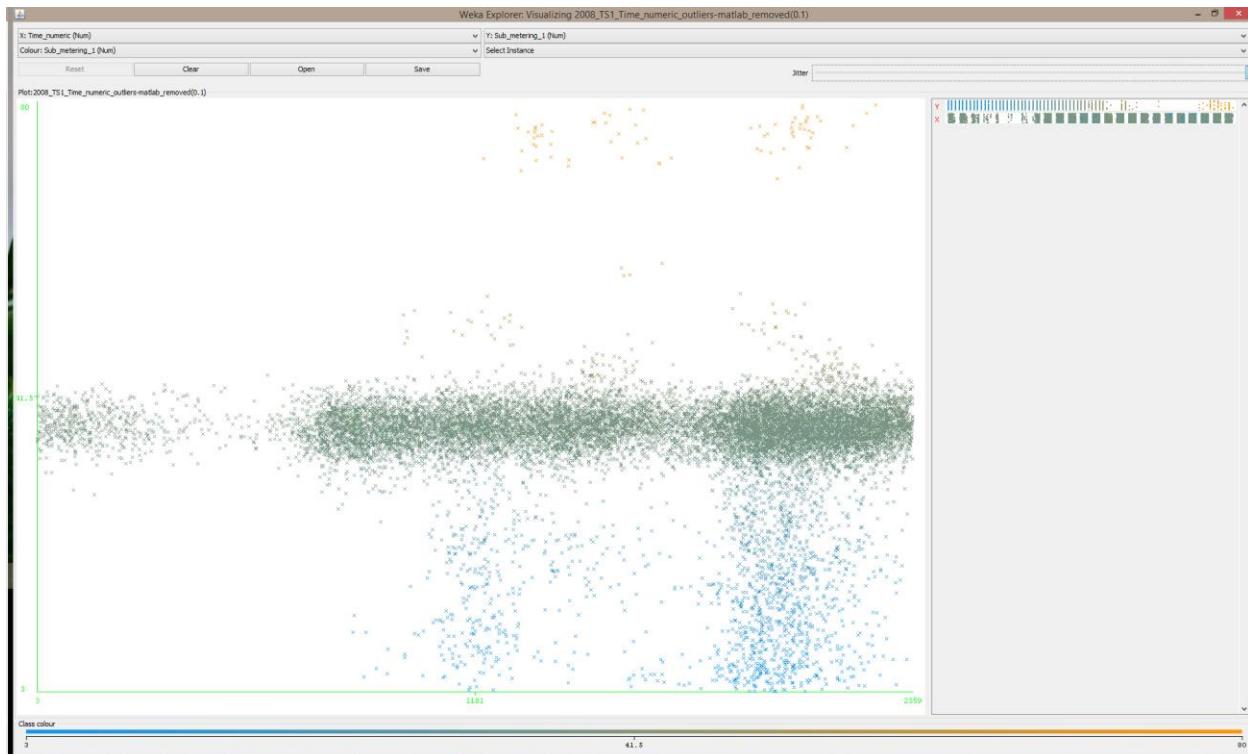
\> ruby csv_reader.rb -l Time,Sub_metering_1 2008.csv // Επιλογή στηλών

\> ruby csv_reader.rb -c Time_numeric 2008.csv // Μετατροπή Time σε numeric

\> ruby csv_reader.rb -o matlab 2008.csv // Αφαίρεση outlier μέσω matlab
Remove idles/zeros?
y
Idle limit?
2
Outlier percentage?
0.1

```

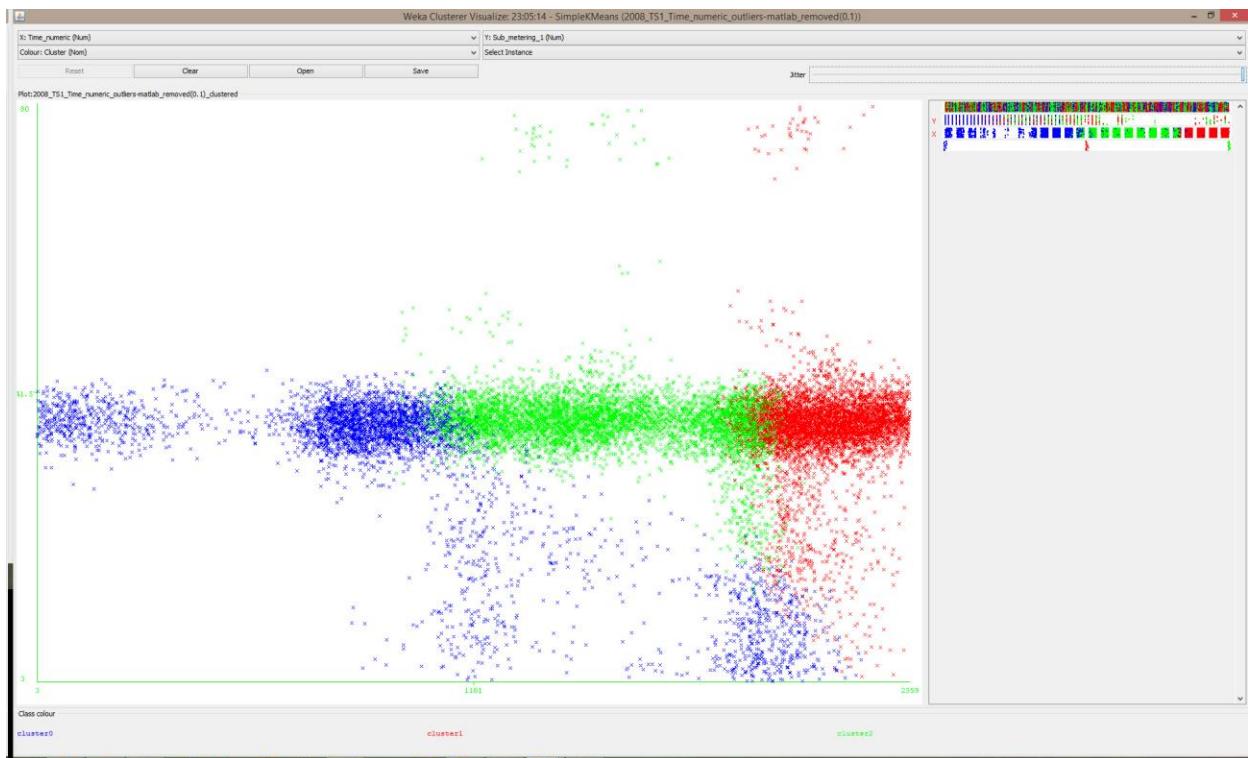
Να σημειωθεί ότι η διαδικασία εντοπισμού εξωκείμενων στην Matlab ξεκινάει **αφού αφαιρεθούν οι idle τιμές** από το csv_reader.rb,
 Έτσι δημιουργήθηκε το παρακάτω dataset:



Εικόνα 16 – 2008 - Sub_metering_1 – no idles, 10% outliers removed

1. Kmeans

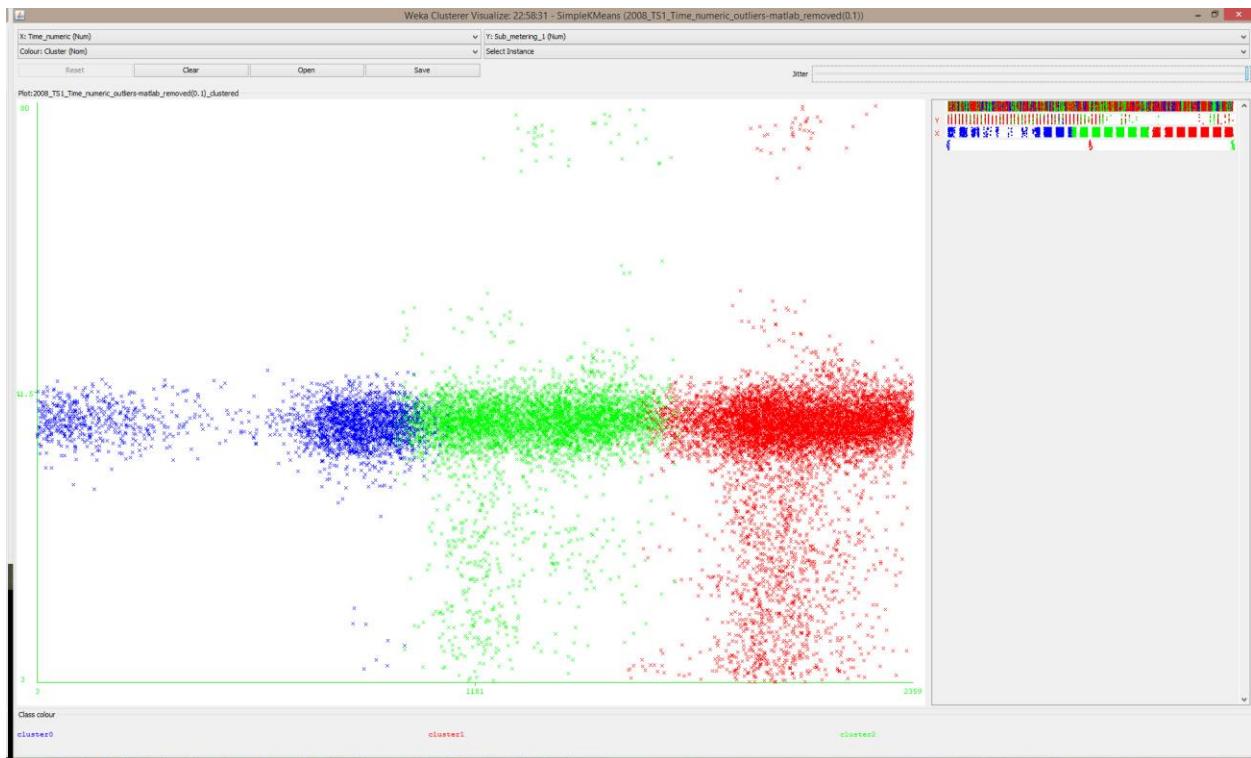
Ξεκινάμε με αλγόριθμο kmeans για 3 γείτονες και 1 επανάληψη:



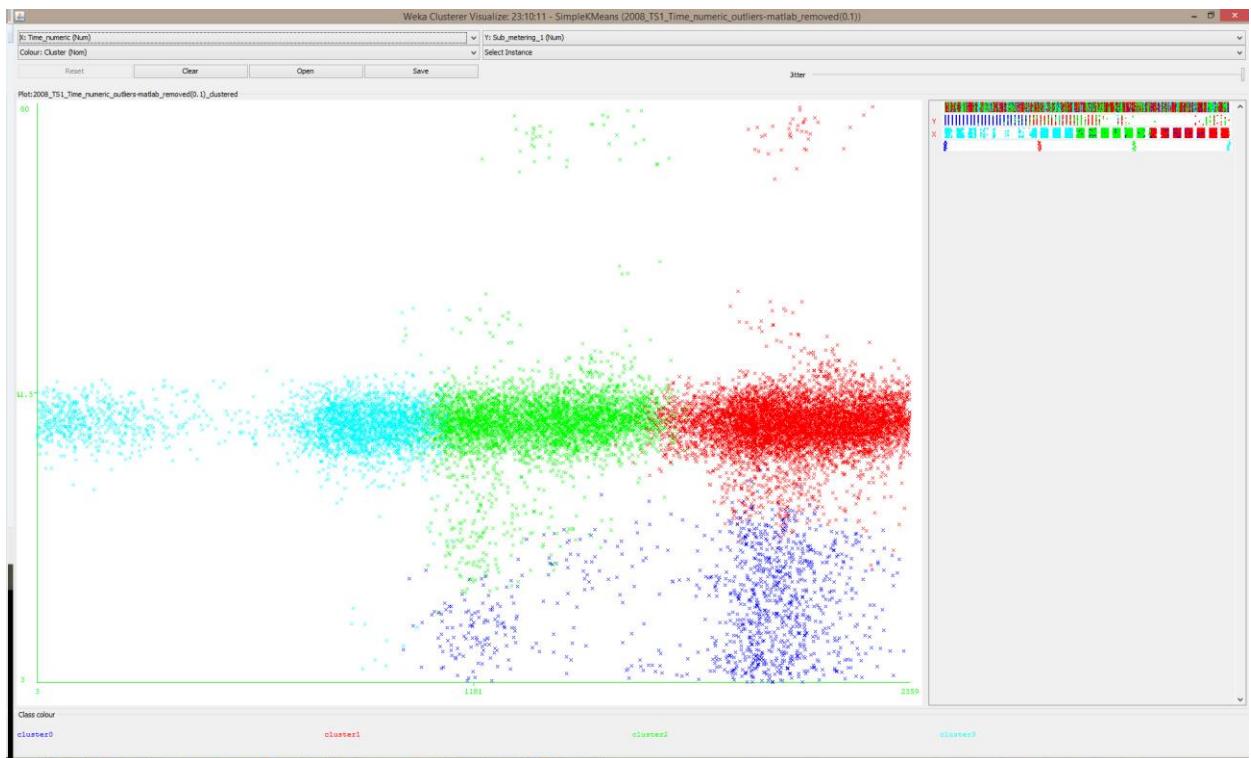
Εικόνα 17 – 2008 - Sub_metering_1 kmeans-N₃S₁₀I₁(0.1)

Η μία επανάληψη προφανώς δεν επαρκεί

Με δοκιμές είδαμε ότι από 10 επαναλήψεις και πάνω η ομαδοποίηση δεν αλλάζει. Για 10 επαναλήψεις και 3/4 γείτονες:

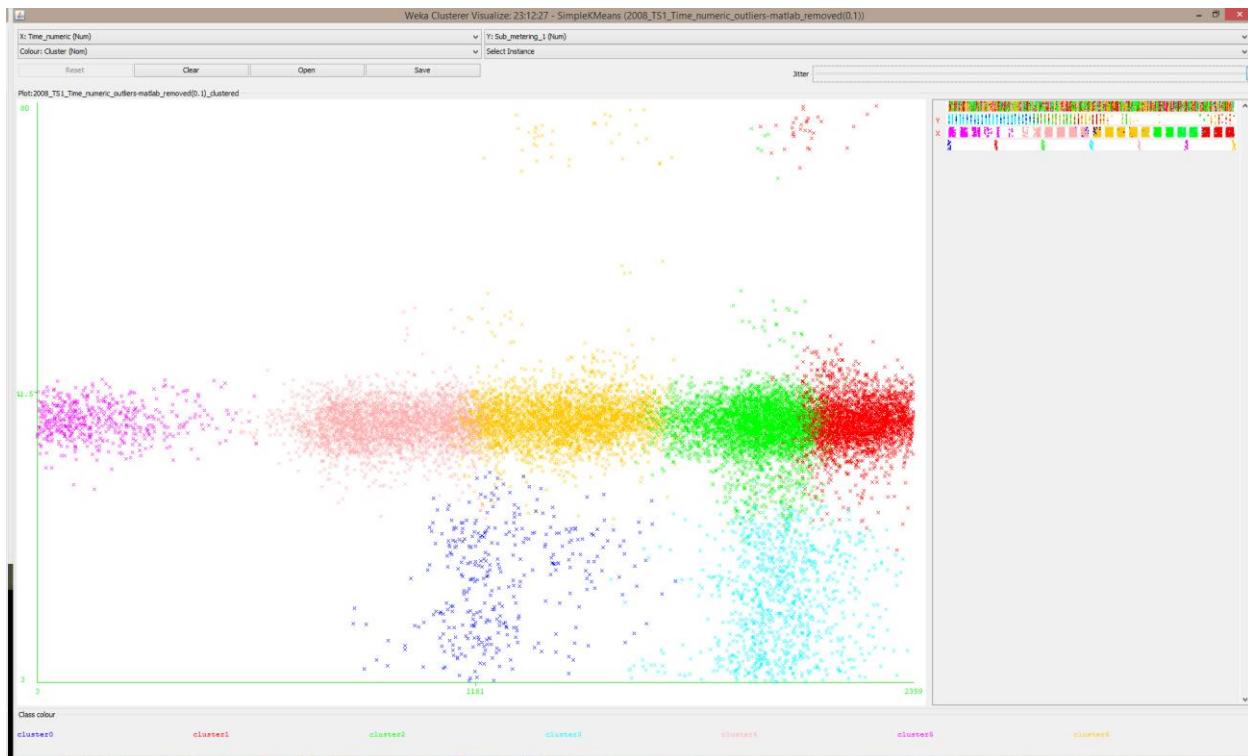


Εικόνα 18 – 2008 - Sub_metering_1 kmeans-N3S1oI1o_(0.1)



Εικόνα 19 - 2008 - Sub_metering_1 kmeans-N4S1oI1o_(0.1)

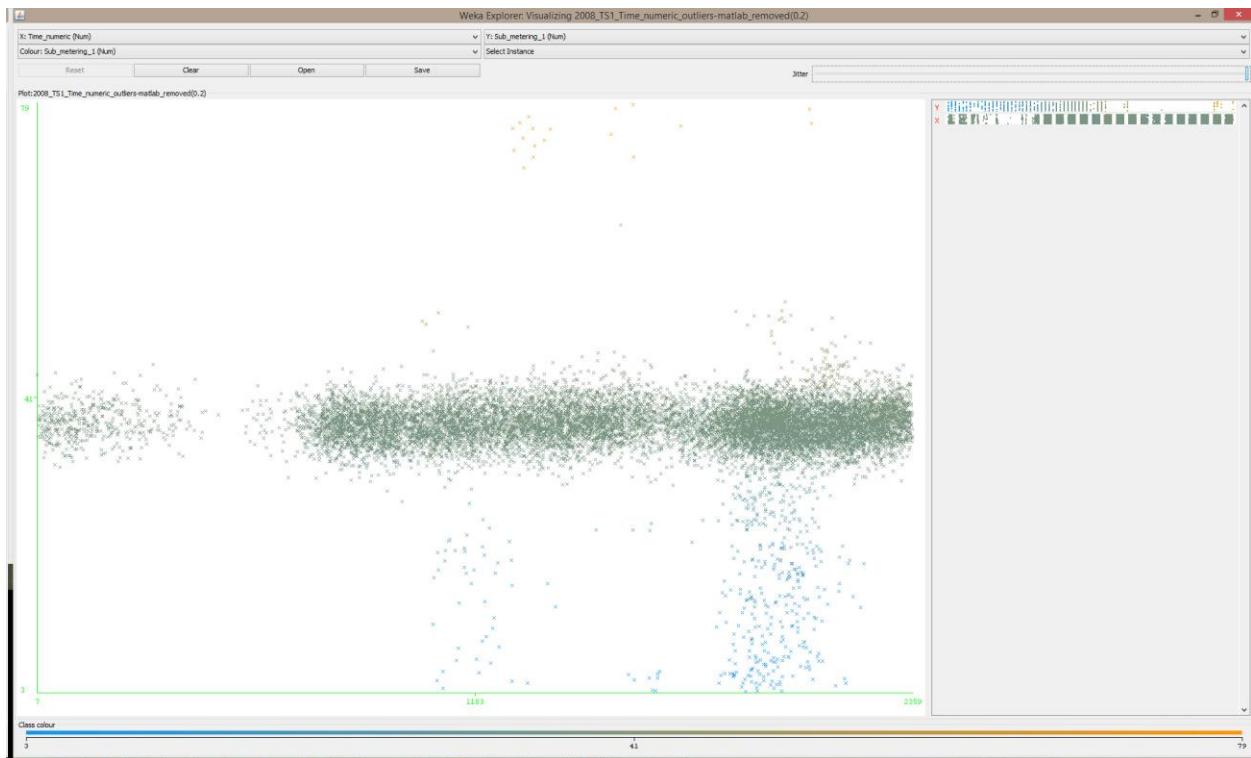
Βλέπουμε ότι το kmeans δυσκολεύεται να δημιουργήσει τις ομάδες που θέλουμε. Αυτό είναι πολύ πιθανό να οφείλεται στις εξωκείμενες τιμές που εξακολουθούν να υπάρχουν. Προσπαθώντας να κάνουμε τον k-means να δει τις εξωκείμενες ως ομάδες παίρνουμε το παρακάτω αποτέλεσμα για 7 γείτονες:



Εικόνα 20 - 2008 - Sub_metering_1 1.4.kmeans-N7S1oI500_(o.1)

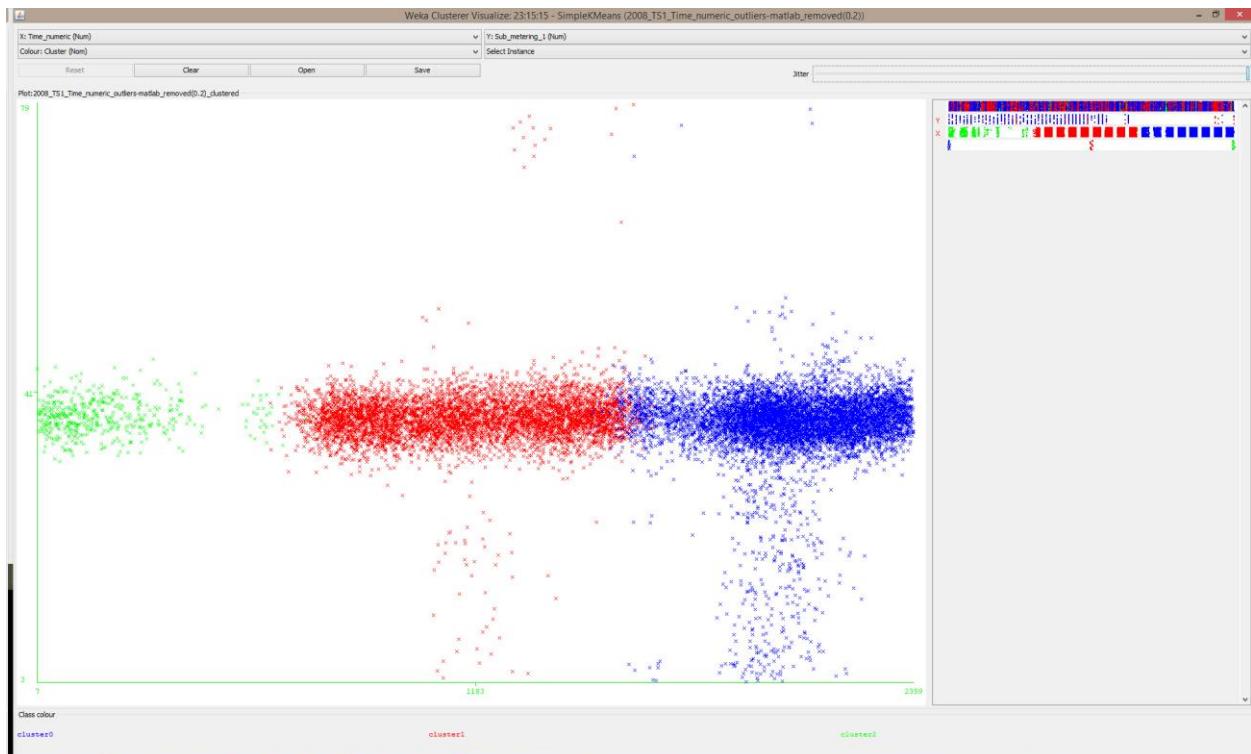
Για πρώτη φορά χωρίσαμε την πρώτη ζώνη με την δεύτερη, αλλά το συνολικό αποτέλεσμα δεν είναι αποδεκτό. Δοκιμάζουμε λοιπόν να βγάλουμε 20% outliers:

```
ruby csv_reader.rb -o matlab <dataset>
Remove idles/zeros?
y
Idle limit?
2
Outlier percentage?
0.2
```



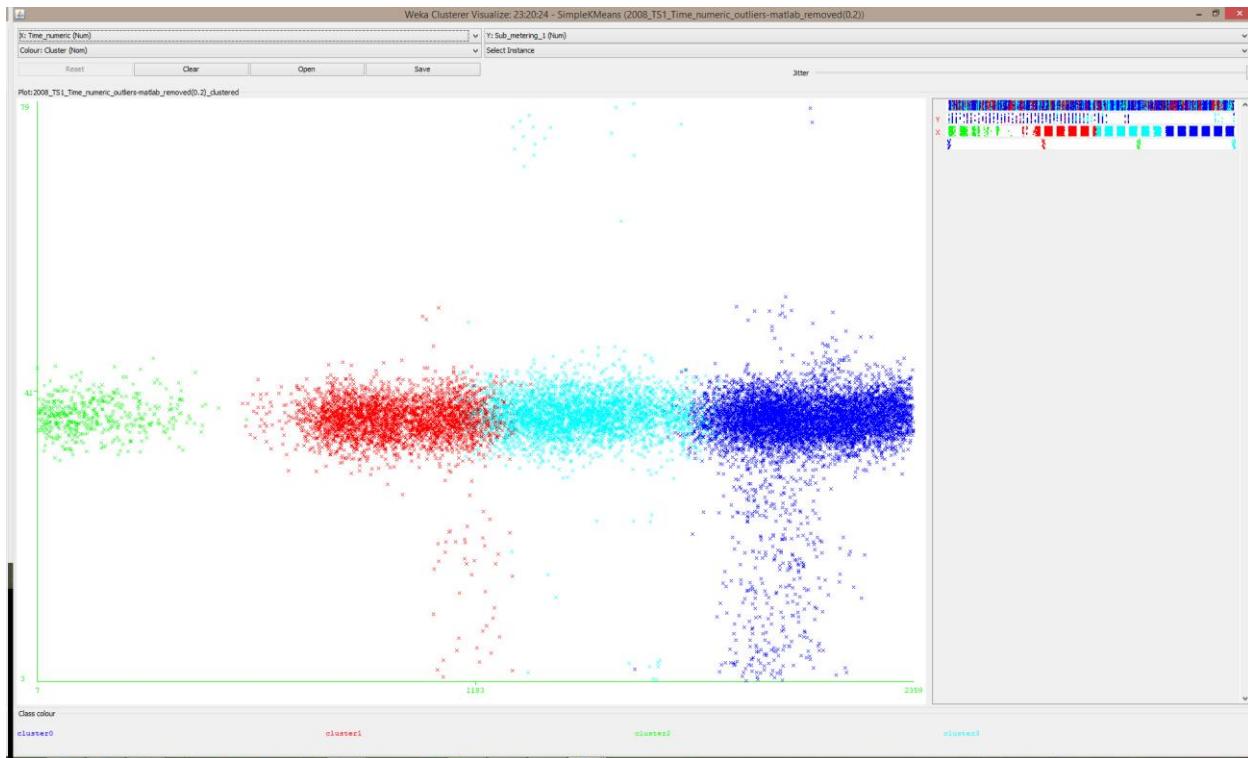
Εικόνα 21 – 2008 - *Sub_metering_1* – no idles, 20% outliers removed

Δοκιμάζουμε πάλι για 3 γείτονες και 500 επαναλήψεις αυτή την φορά (default value):



Εικόνα 22 - 2008 - *Sub_metering_1* kmeans-N₃SioI500_(0.2)

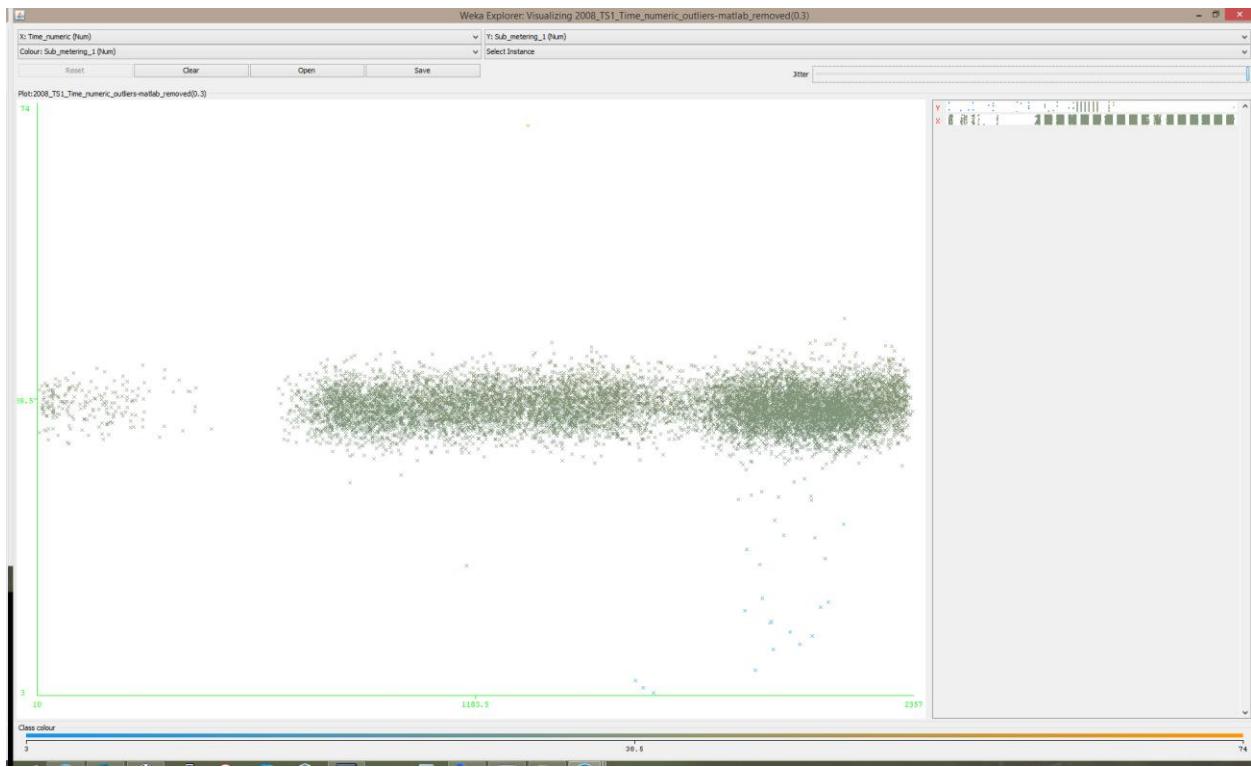
Φαίνεται ότι το k-means βρήκε τις ζώνες που θέλαμε, αλλά στην τελευταία ζώνη συμπεριέλαβε και τις τιμές της ενδιάμεσης αραιής περιοχής. Δηλαδή τις ώρες 4-6 το απόγευμα. Προσπαθώντας να αποσπάσουμε αυτές τις τιμές από αυτή την ομάδα δοκιμάζουμε με 4 γείτονες:



Εικόνα 23 - 2008 - Sub_metering_1 kmeans-N4S1oI5oo_(0.2)

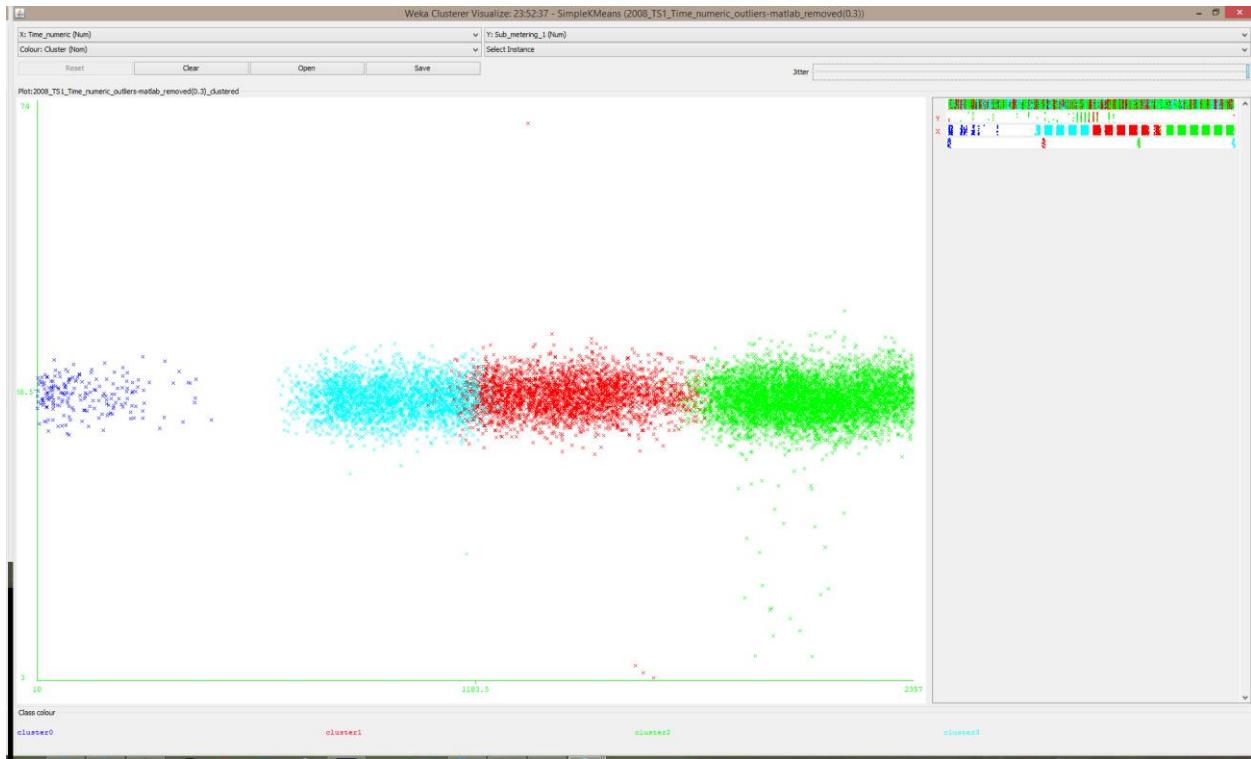
Χωρίς όμως να έχουμε αποτέλεσμα. Οι δοκιμές με περισσότερους γείτονες δεν απέδωσαν.
Δοκιμάσαμε να αφαιρέσουμε και άλλα outliers:

```
ruby csv_reader.rb -o matlab <dataset>
Remove idles/zeros?
y
Idle limit?
2
Outlier percentage?
0.3
```



Εικόνα 24 – 2008 - *Sub_metering_1* – no idles, 30% outliers removed

Στην συνέχεια εκτελέσαμε τα ίδια πειράματα χωρίς αποτέλεσμα.

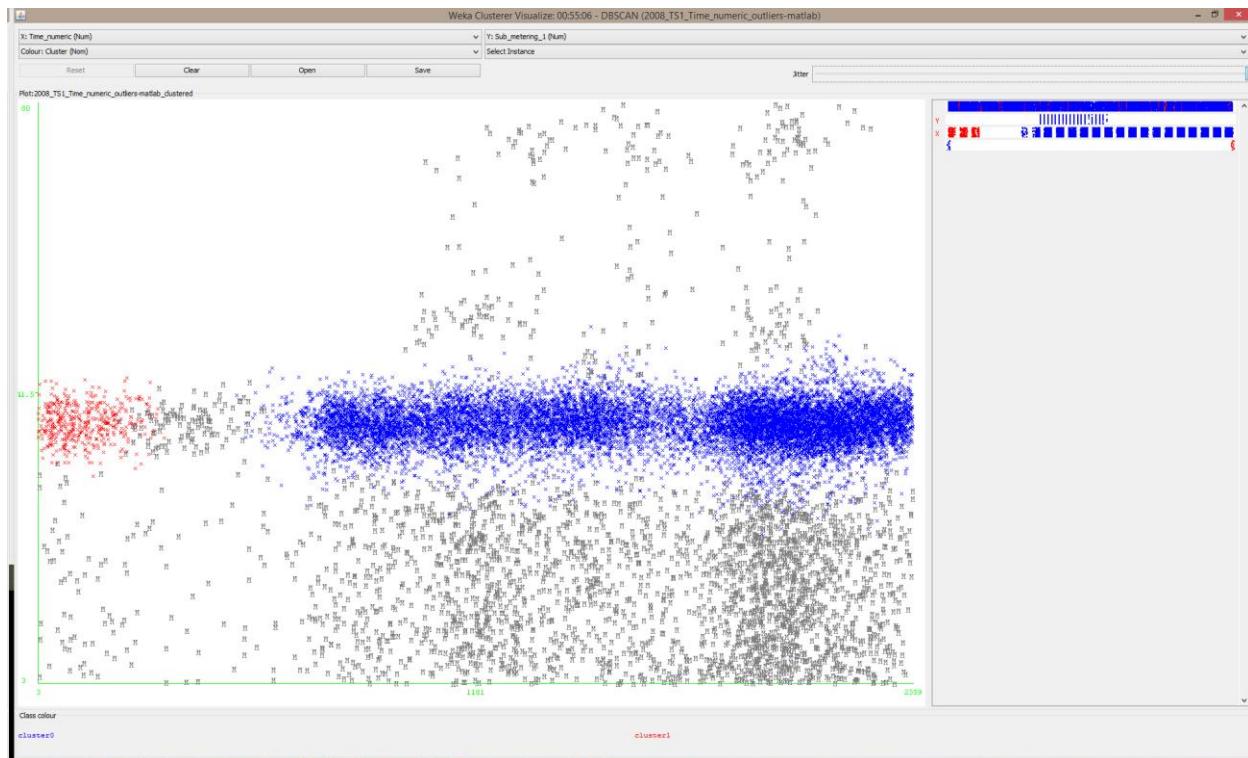


Εικόνα 25 - 2008 - *Sub_metering_1* kmeans-N4S1oI5oo_(0.3)

Επειδή η ζώνη 4-6 έχει ίση ή μεγαλύτερη πυκνότητα με την ζώνη 12 με 2 το πρωί, περαιτέρω αφαίρεση outlier οδηγούσε σε αφαίρεση της τελευταίας. Στους επόμενους αλγορίθμους πειραματίζομασταν με το σετ δεδομένων όπου είχαν αφαιρεθεί 30% των outliers μιας και διαπιστώσαμε ότι έχει καλύτερη απόδοση.

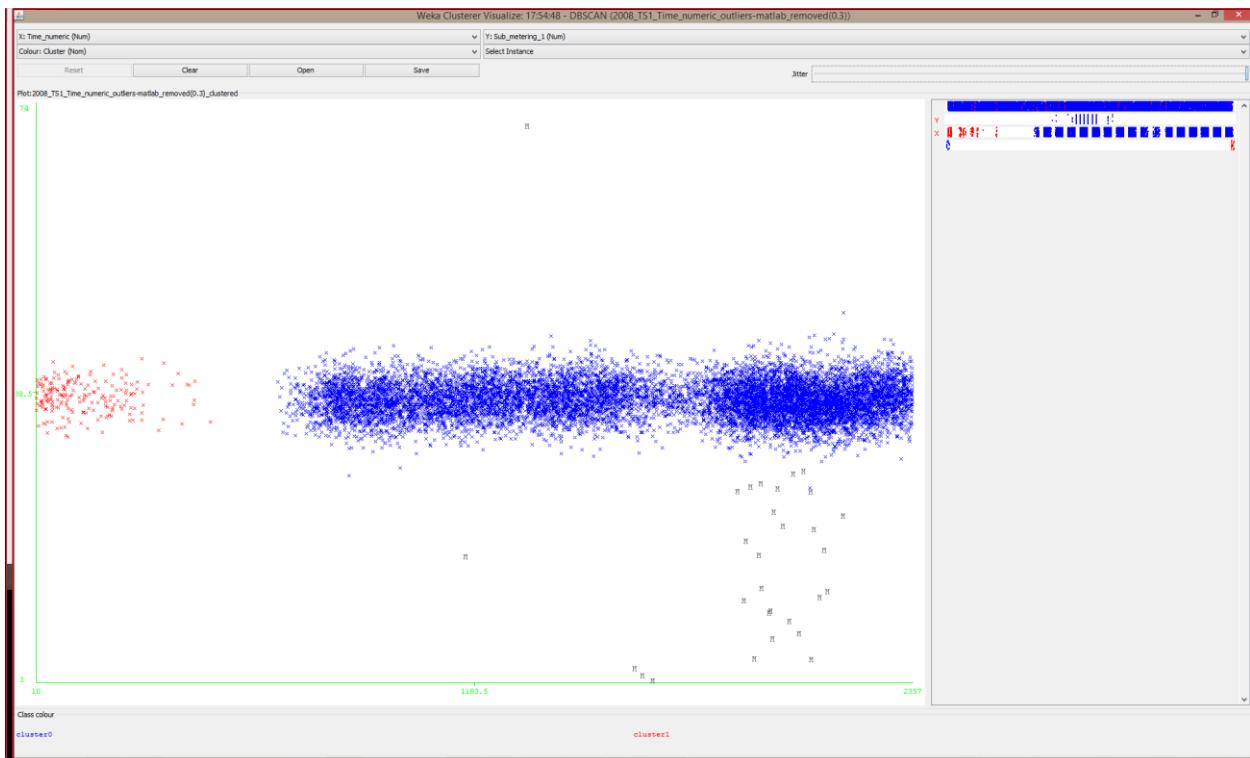
2. DBSCAN

Ο επόμενος αλγόριθμος που δοκιμάστηκε σε αυτό το σετ είναι ο DBSCAN. Στην αρχή δοκιμάσαμε χωρίς να αφαιρέσουμε outliers, γνωρίζοντας ότι ο DBSCAN είναι ανθεκτικός στο noise. Αφού δοκιμάσαμε αρκετές τιμές για χαμηλό eps και διάφορα MinPts, καταλήξαμε στις τιμές 0.06 και 350 αντίστοιχα. Τρέχοντάς τον με αυτές τις παραμέτρους πήραμε το εξής αποτέλεσμα:



Εικόνα 26 - 2008 - Sub_metering_1 DBSCAN-Eo.06M350_(raw)

Παρ' όλο που δεν βρήκε τις 3 ομάδες όπως θέλαμε, κατάφερε και χώρισε το βράδυ από την υπόλοιπη ημέρα. Λίγο καλύτερο αποτέλεσμα πήραμε όταν αφαιρέσαμε το 30% των outliers και ορίσαμε MinPts = 20:

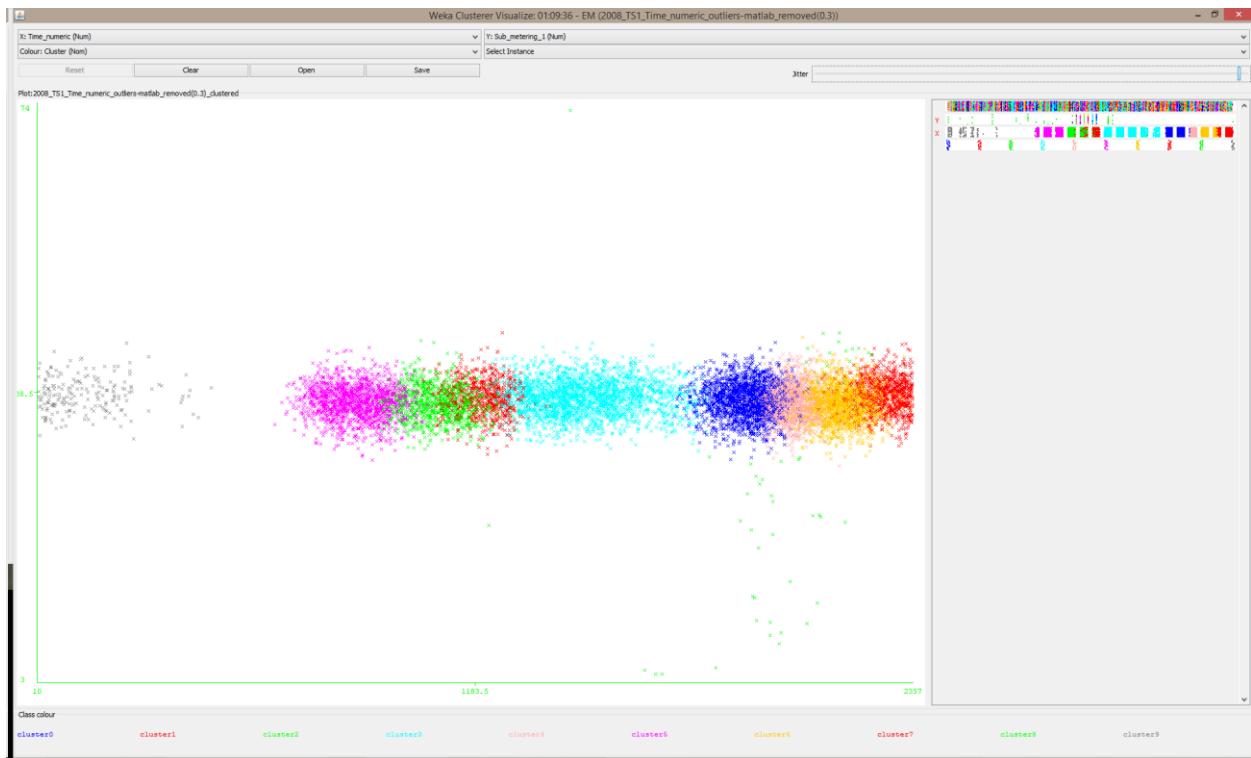


Εικόνα 27 - 2008 - Sub_metering_1 DBSCAN-Eo.o6M35o_(0.3)

Γενικώς η αφαίρεση των outliers παρέχει μία ευελιξία στον ορισμό του MinPts και του eps. Από την άλλη όσο και να αλλάζουμε το eps και το MinPts για να χωρίσουμε τις δύο τελευταίες ζώνες, δεν μπορεί να τις βρει. Απ' ότι φαίνεται η διαφορά στις πυκνότητες των δύο ομάδων δεν επαρκεί για να τις διαχωρίσει το DBSCAN.

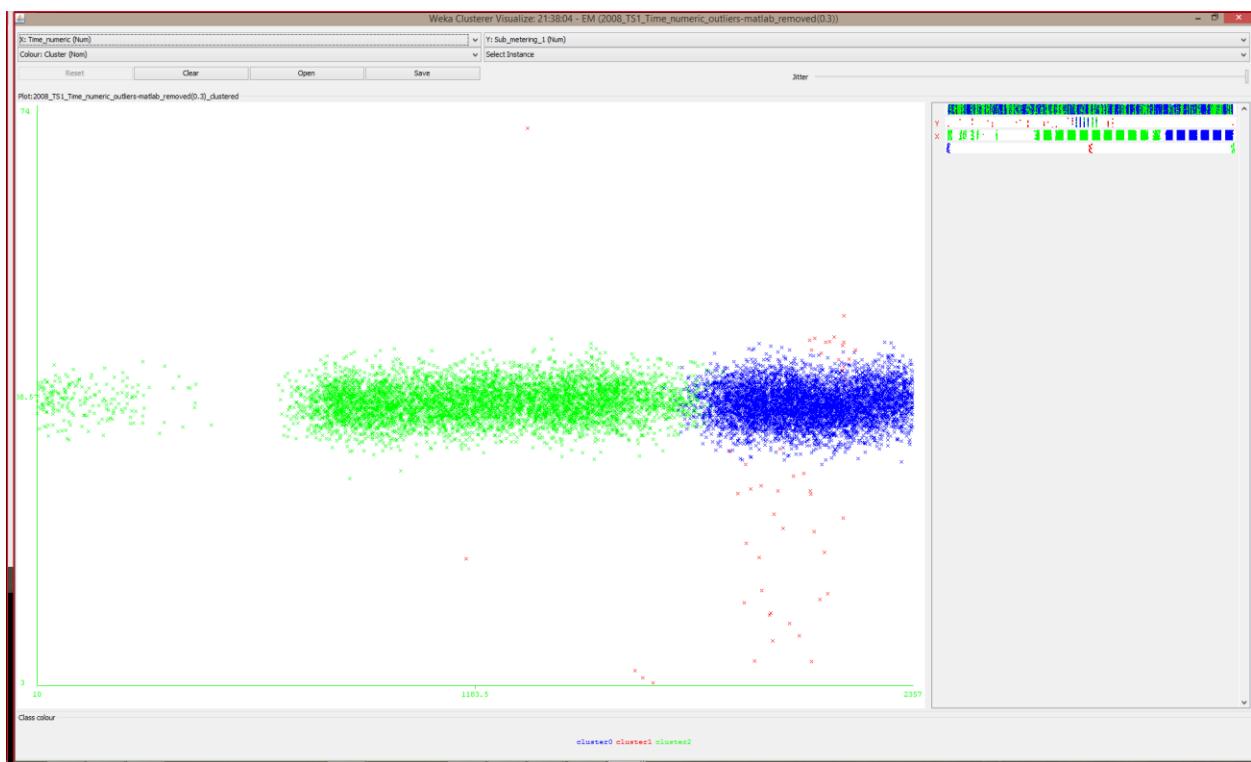
3. EM

Αρχικά τρέξαμε τον EM με την default ρύθμιση του Weka ώστε να προσπαθήσει να βρει μόνος του τα cluster:

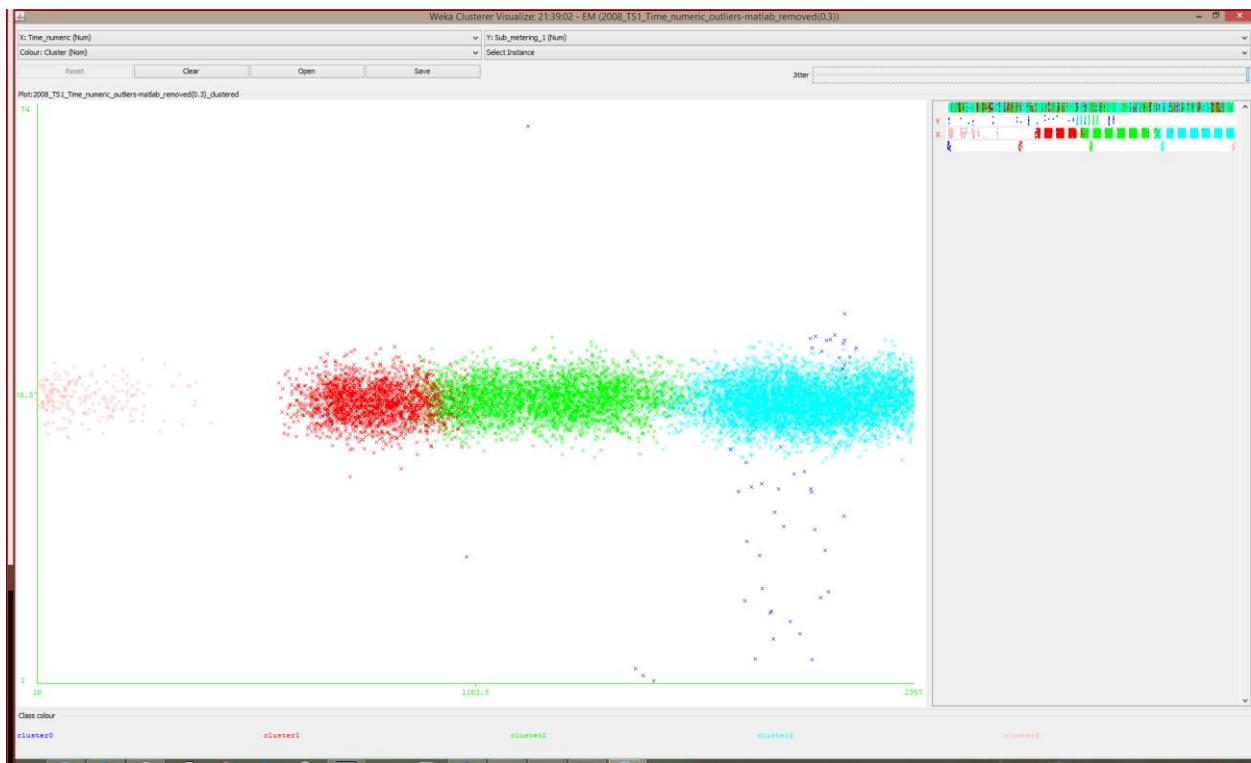


Εικόνα 28 - 2008 - Sub_metering_1 EM_(0.3)

Όπως φαίνεται το αποτέλεσμα δεν είναι επιθυμητό και γι' αυτό δοκιμάσαμε για 3 και 5 clusters (μιας και οι 4 δεν απέδωσαν):



Εικόνα 29 - 2008 - Sub_metering_1 EM_N-3(0.3)

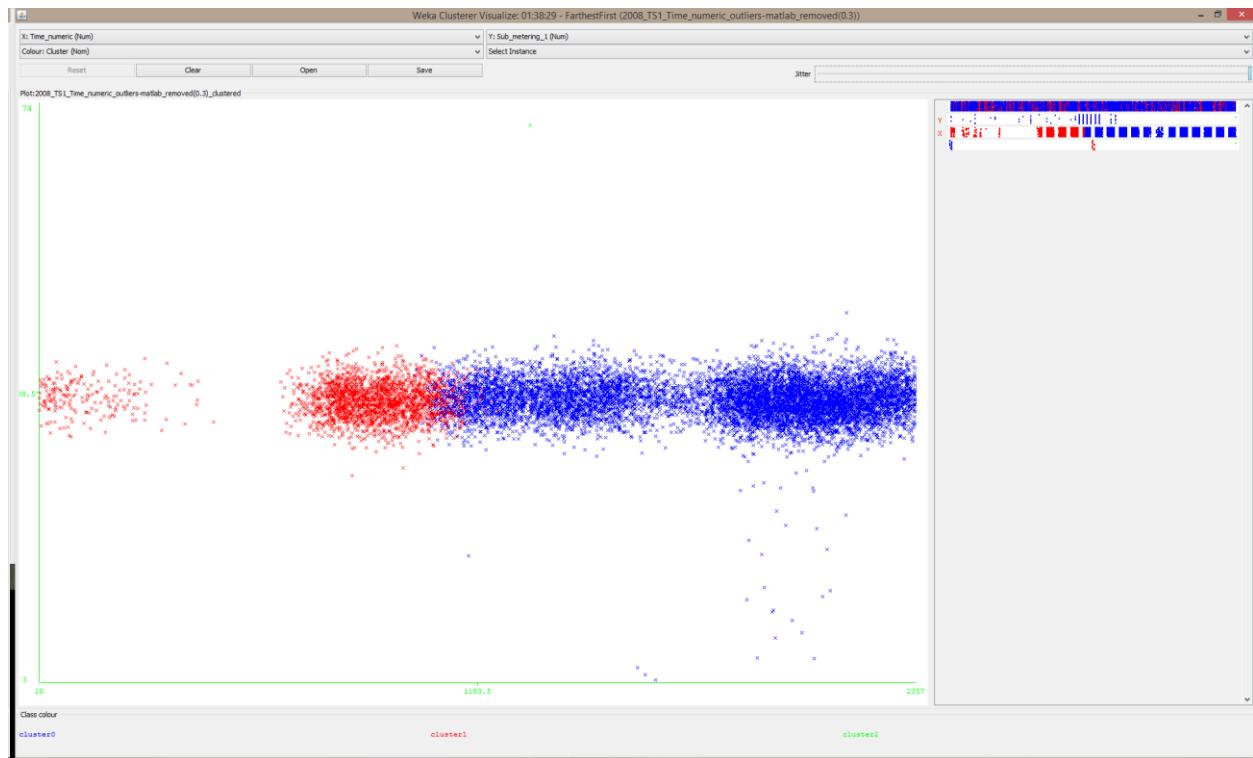


Εικόνα 30 - 2008 - Sub_metering_1 EM_N-5(0.3)

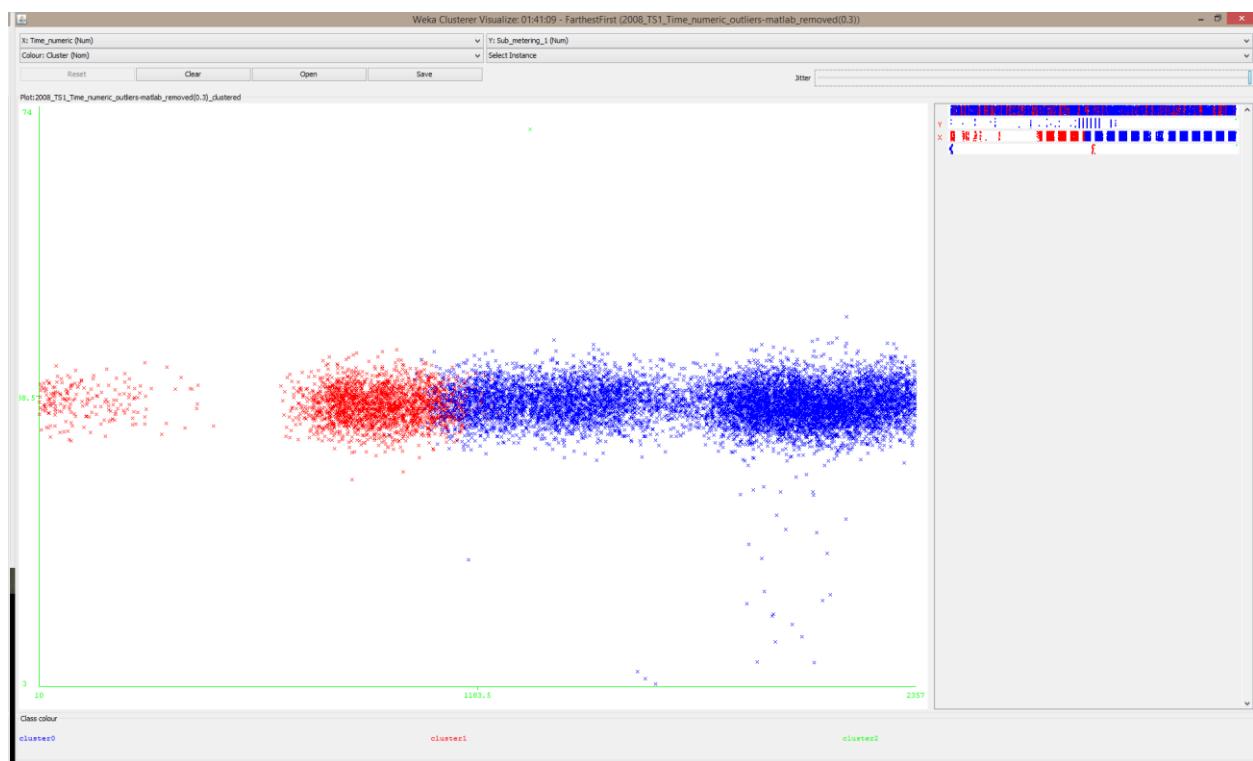
Το αποτέλεσμα με 4 γείτονες είναι αρκετά ικανοποιητικό.

4. FarthestFirst

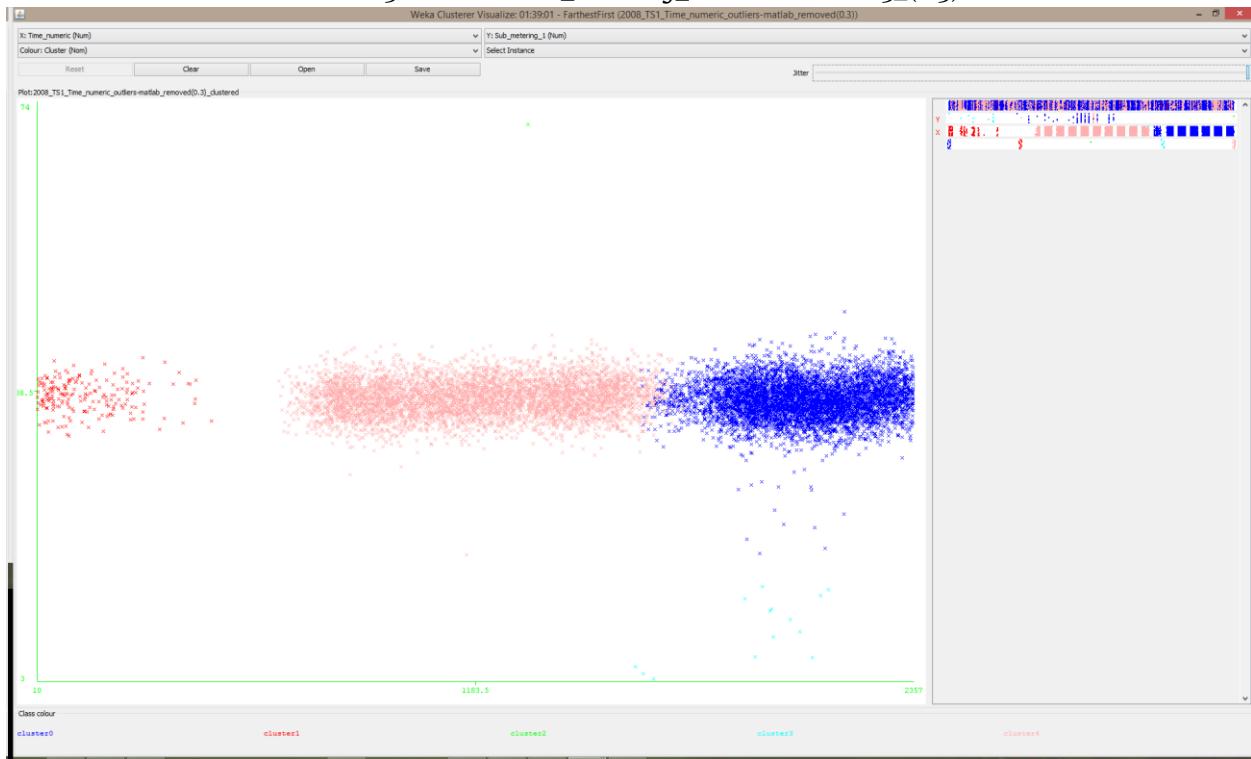
Δοκιμάσαμε για 2,3 και 5 γείτονες:



Εικόνα 31 - 2008 - Sub_metering_1 FarthestFirst-N₂(0.3)



Εικόνα 32 - 2008 - Sub_metering_1 FarthestFirst-N3_(o.3)



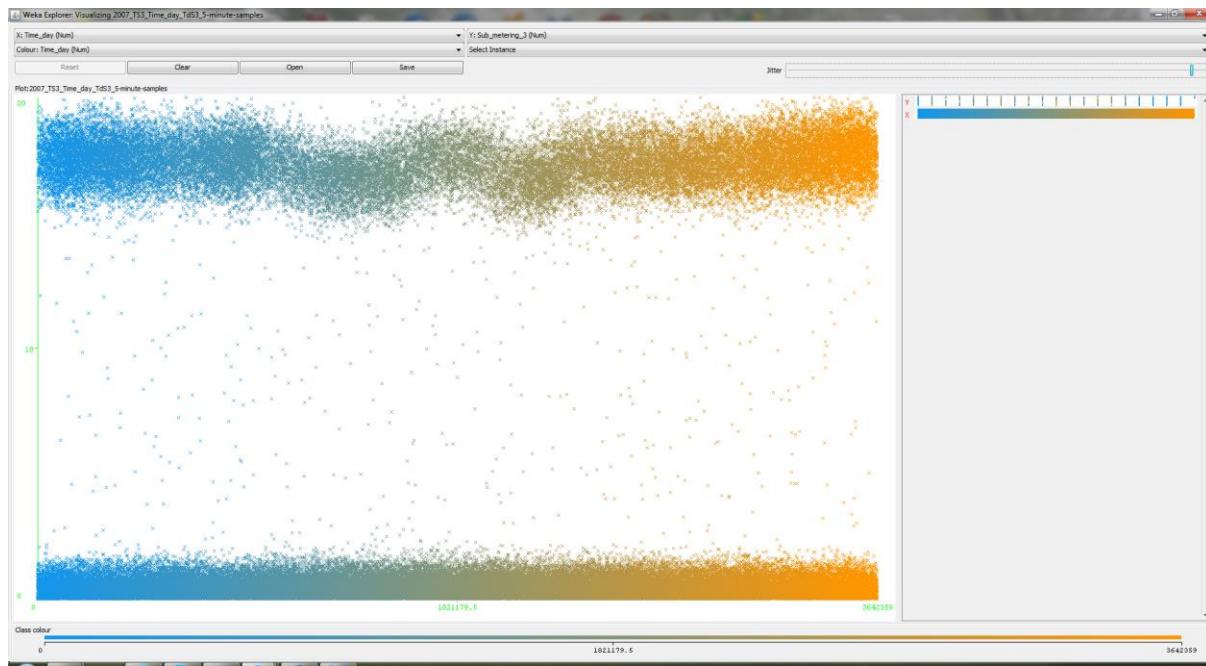
Εικόνα 33 - 2008 - Sub_metering_1 FarthestFirst-N5_(o.3)

Το καλύτερο αποτέλεσμα το πήραμε για 5 γείτονες μιας και ο FarthestFirst είναι ευαίσθητος στις εξωκείμενες τιμές και δημιουργεί ομάδες με αυτές.

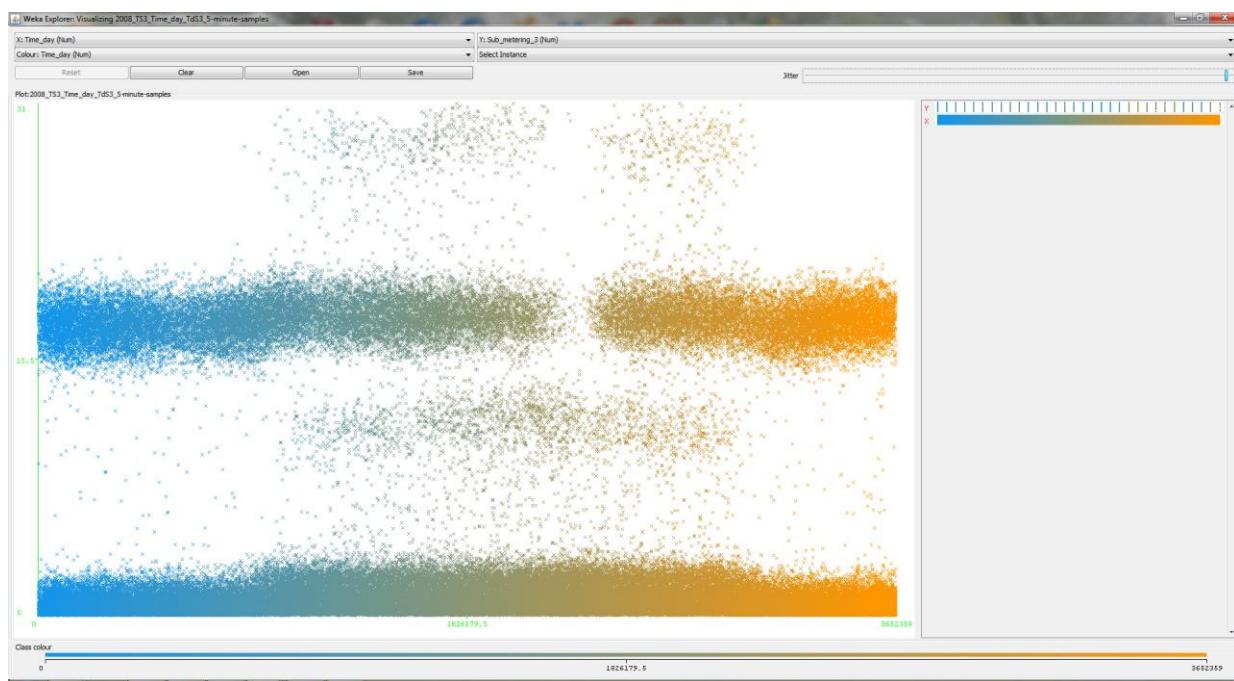
6.2 2010 – Day - Sub_metering_3

Στην παρακάτω ανάλυση χρησιμοποιούμε dataset των οποίων τα attributes έχουν δύο τιμές, τον χρόνο ως έναν ακέραιο αριθμό από ο έως 365^*24^*60 , και την τιμή της αθροιστικής ενεργού ισχύος που καταναλώνεται στον θερμοσίφωνα και στο air-conditioner. Το μέγεθος του dataset αντιστοιχεί σε έναν ολόκληρο χρόνο (π.χ έτος 2008), ενώ έγινε υποδειγματοληψία για να μειωθεί η απαιτούμενη υπολογιστική ισχύς, ανά 5 λεπτά.

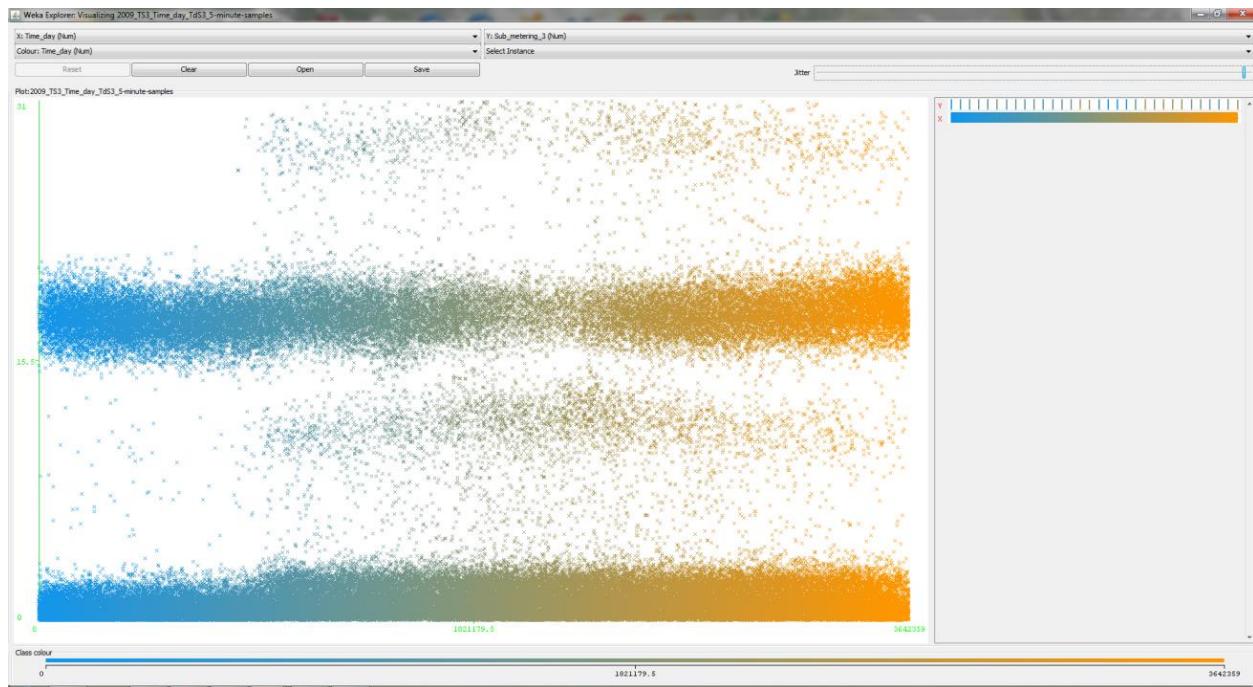
Παρουσιάζονται στη συνέχει τα 2-D γραφήματα των dataset για κάθε έτος.



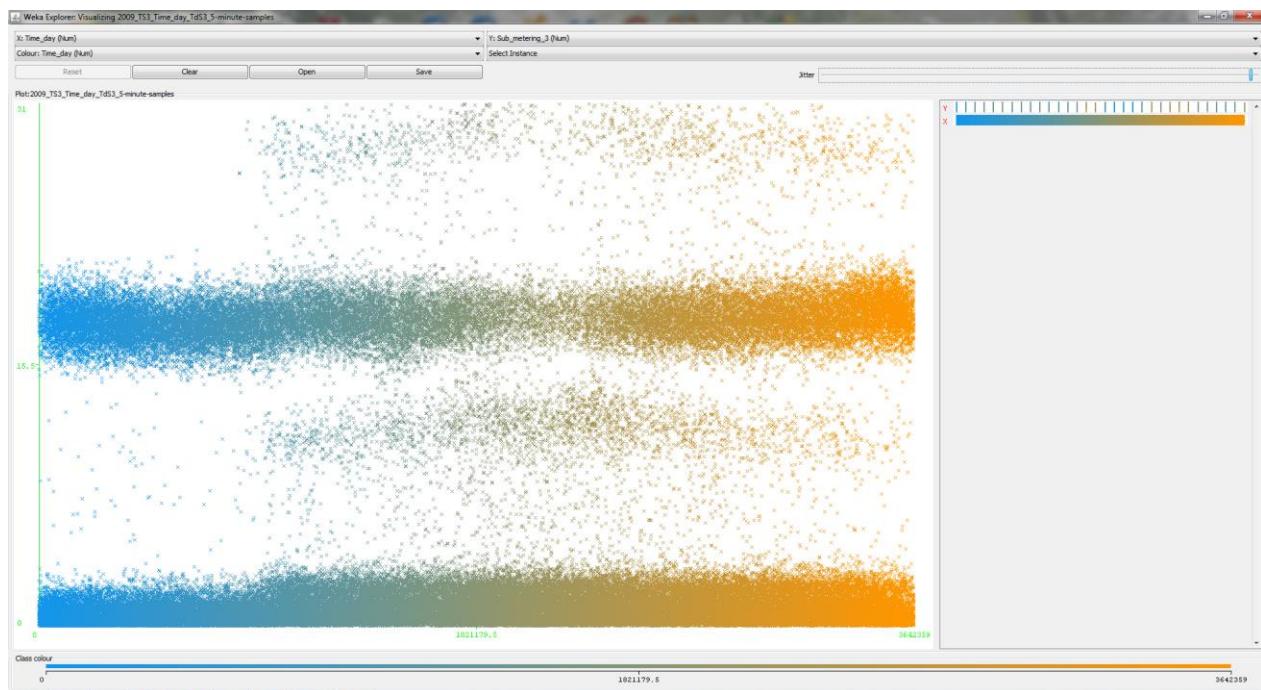
Εικόνα 34 2007 - Time - Sub_metering_3



Εικόνα 35 2008 - Time - Sub_metering_3



Εικόνα 36 2009- Time - Sub_metering_3

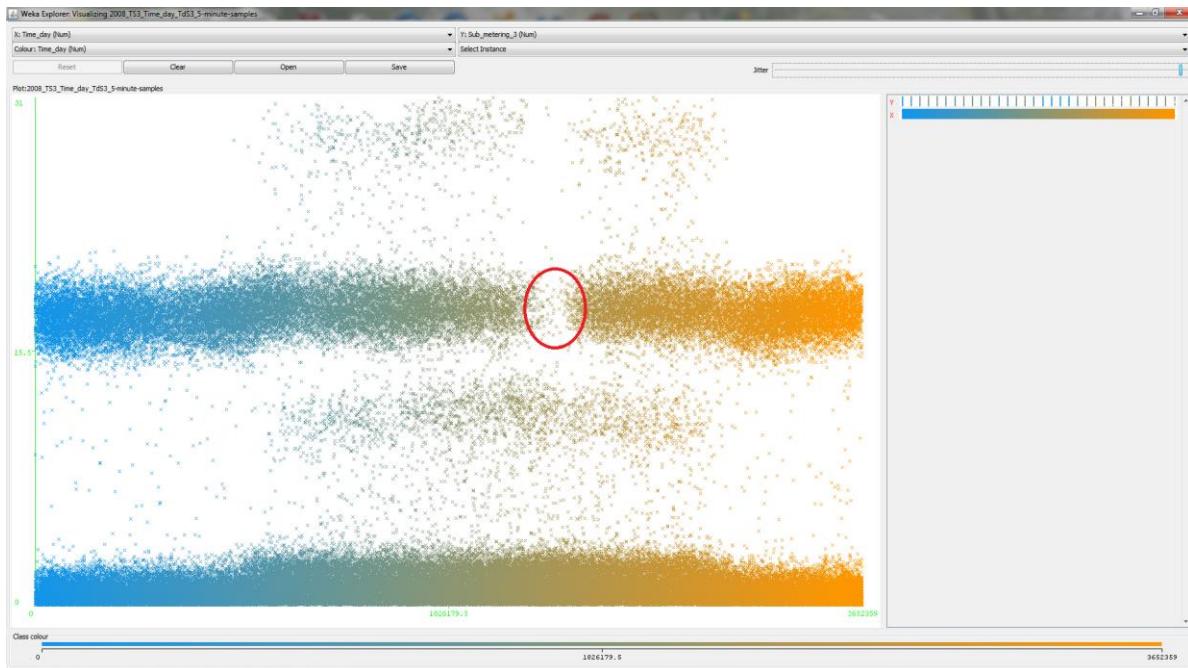


Εικόνα 36 2010 - Time - Sub_metering_3

Παρατηρούμε ότι στα έτη 2008-2010 έχουμε παρόμοια συμπεριφορά, ενώ στο έτος 2007 εμφανίζεται μία μόνο συσκευή με σταθερή κατανάλωση περίπου 18 Wh, η οποία λειτουργεί για όλο το έτος. Στα υπόλοιπα 3 έτη συνεχίζεται η λειτουργία αυτής της συσκευής, σε συνδυασμό όμως και με μία ακόμη, η οποία έχει 2 επίπεδα λειτουργίας: 11 και 29 Wh. Το επίπεδο λειτουργίας 29 Wh έχει 2 ζώνες έντονης δραστηριότητας, ενώ το 11 μία, η οποία (λογικό) βρίσκεται εκεί όπου το

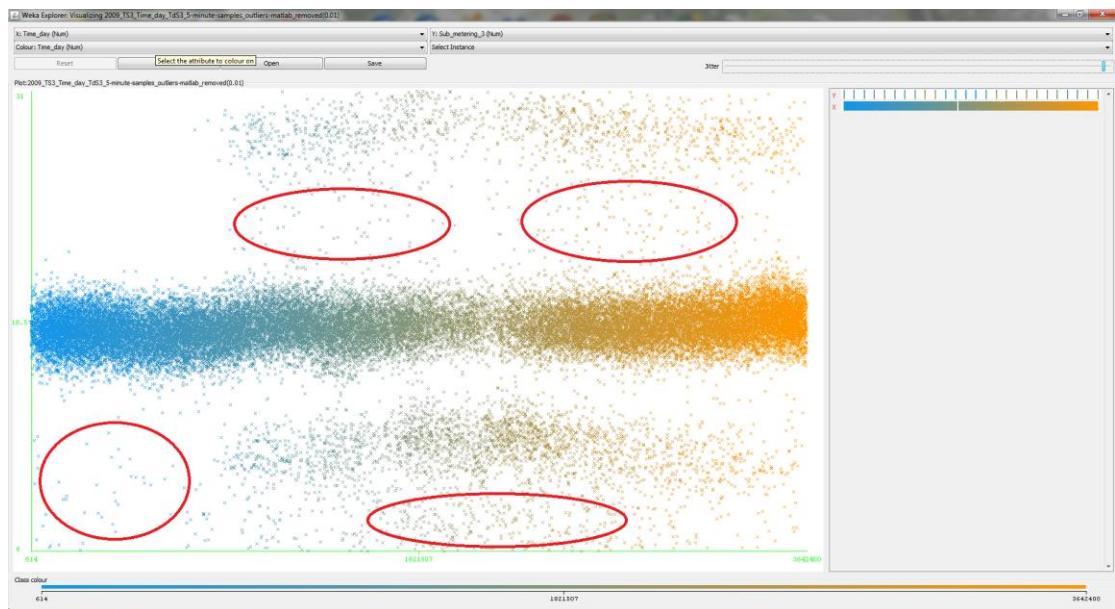
υψηλότερο επίπεδο λειτουργίας είναι αραιό. Επίσης, είναι ξεκάθαρο ότι σε ολόκληρη την διάρκεια της χρονιάς υπάρχει μια idle κατάσταση των συσκευών, η οποία φτάνει και μέχρι τα 5Wh κατανάλωση.

Θεωρήσαμε ότι το έτος 2007 δεν περιέχει αρκετά σημαντική πληροφορία, ώστε να επιχειρήσουμε ομαδοποίηση (το μόνο που θα βρίσκαμε είναι πότε χρησιμοποιείται ή όχι η συσκευή των 18 Wh). Πολύ μεγαλύτερο ενδιαφέρον έχουν τα 3 επόμενα έτη. Από αυτά, επιλέξαμε να κάνουμε ομαδοποίηση στο έτος 2009, επειδή α) το 2010 έχει αρκετά περισσότερο ποίηση και β) επειδή στο 2008 υπάρχει μια περίοδος μέσα στον Αύγουστο που οι συσκευές δεν χρησιμοποιούνται, και η οποία θα εμπόδιζε την εύρεση των πραγματικών ομάδων ενδιαφέροντος.



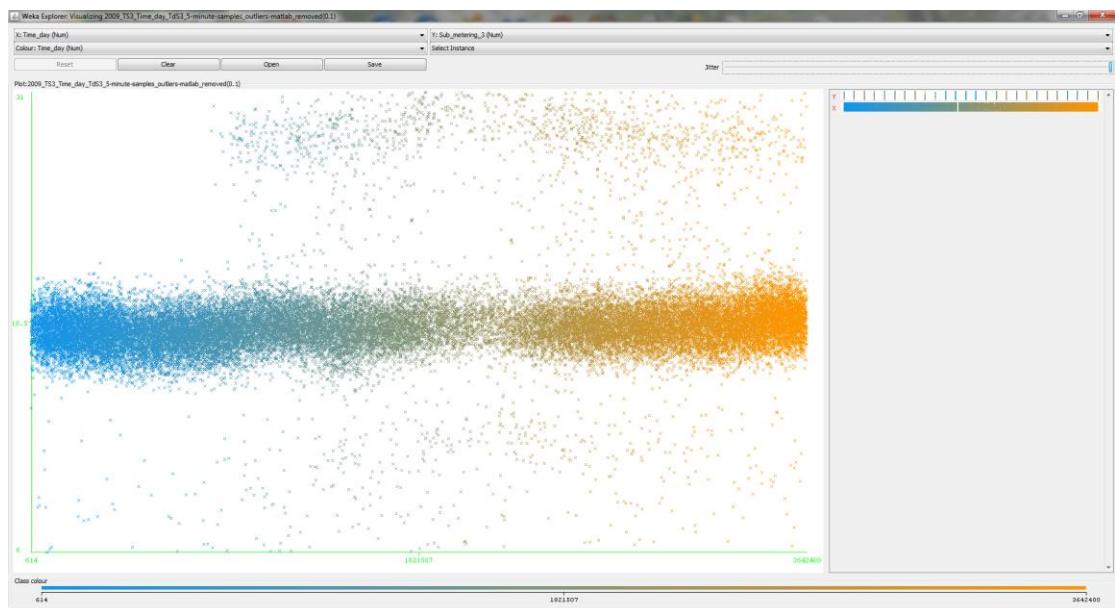
Εικόνα 37 *Sub_metering_3*

Αρχικά αφαιρούμε από το έτος 2009 τις idle τιμές και το 1% των outliers, και έχουμε το παρακάτω αποτέλεσμα:



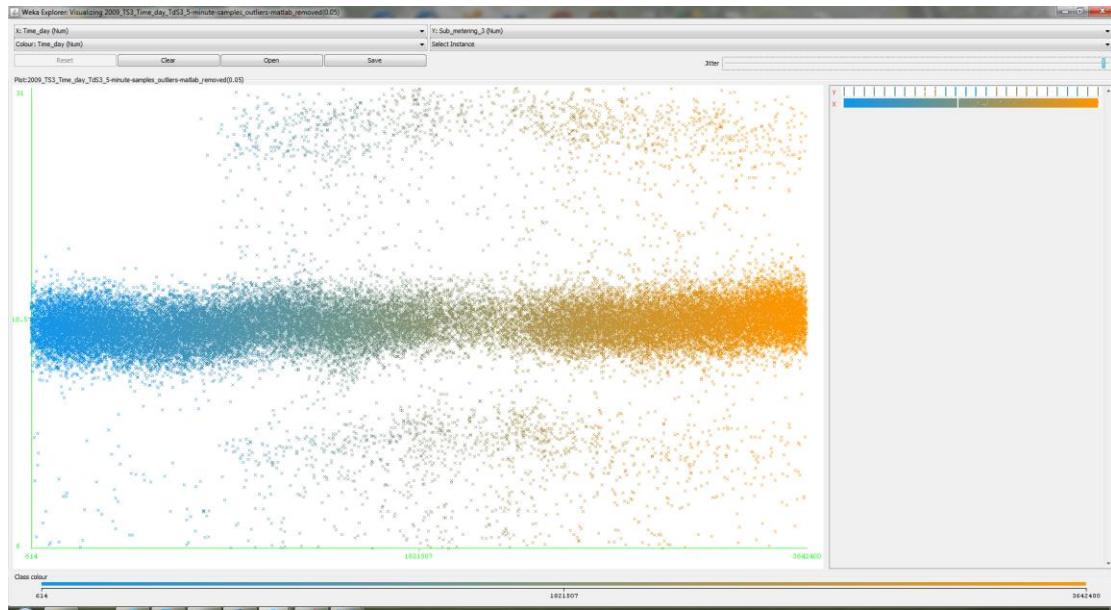
Εικόνα 38 *Sub_metering_3*

Όπως φαίνεται από το γράφημα, οι idle τιμές έχουν αφαιρεθεί, αλλά παραμένουν ακόμη κάποια outliers, τα οποία σημειώνονται μέσα σε κόκκινο κύκλο. Οπότε σε επόμενο στάδιο επιχειρούμε να αφαιρέσουμε το 10% των εξωκείμενων τιμών, και παίρνουμε το παρακάτω σχήμα:



Εικόνα 39 *Sub_metering_3*

Δυστυχώς μαζί με την αφαίρεση των outliers, αφαιρείται σε μεγάλο βαθμό και το κάτω επίπεδο λειτουργίας των η Wh, το οποίο είναι αρκετά αραιό. Δοκιμάζουμε λοιπόν 5%, και έχουμε το εξής αποτέλεσμα:

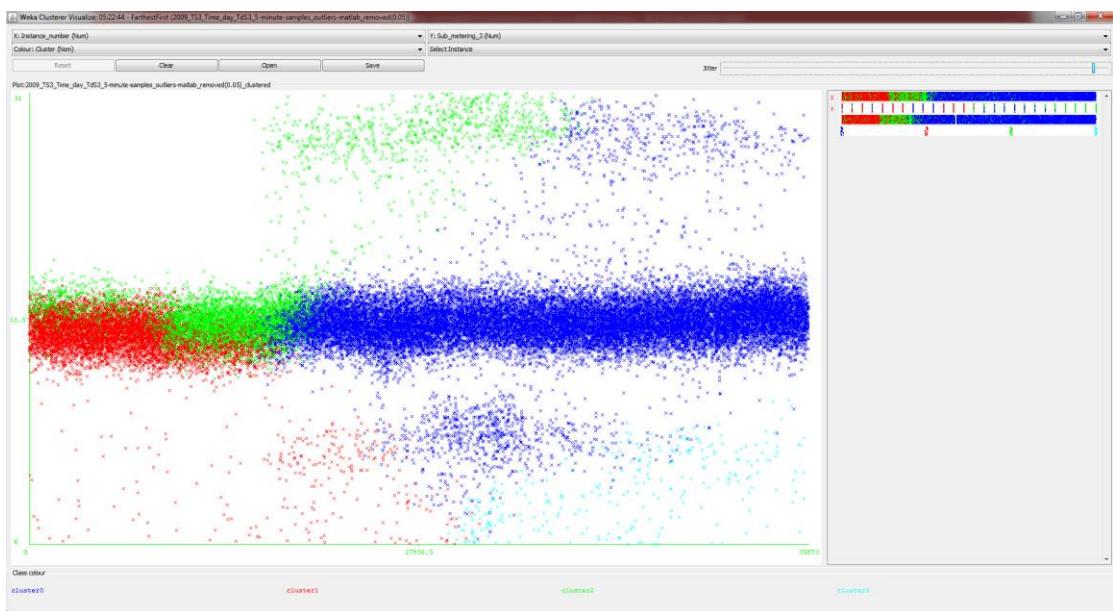


Εικόνα 4ο Sub_metering_3

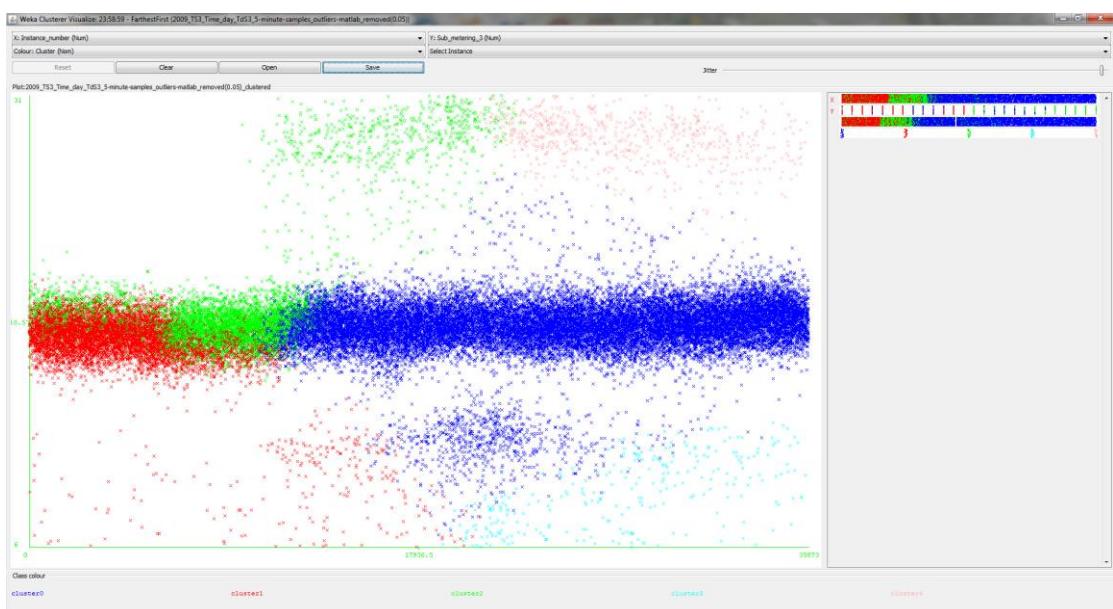
Γενικά έχει μειωθεί η πυκνότητα της κάτω ομάδας, το οποίο λογικά θα αποτελέσει πρόβλημα για την αναγνώρισή της. Για χρήση σε μη πυκνωτικούς αλγορίθμους όμως, αυτό το ποσοστό αποτελεί τον συμβιβασμό ανάμεσα στην ύπαρξη μιας σχετικά καλά ορισμένης ομάδας, και στην αφαίρεση των outliers. Σε αλγορίθμους όπως ο DBSCAN μπορεί να χρησιμοποιηθεί και η αφαίρεση του 1%.

Θα εφαρμόσουμε στο dataset τους αλγορίθμους FarthestFirst, SimpleKMeans (Διαχωρισμού), DBSCAN (Πυκνωτικός), HierarchicalClusterer για single, complete και average link (Ιεραρχικοί). Για κάθε αλγόριθμο θα δώσουμε σχηματικά το clustering, τις παραμέτρους που χρησιμοποιήθηκαν, και τον (μέσο) συνολικό συντελεστή silhouette (s).

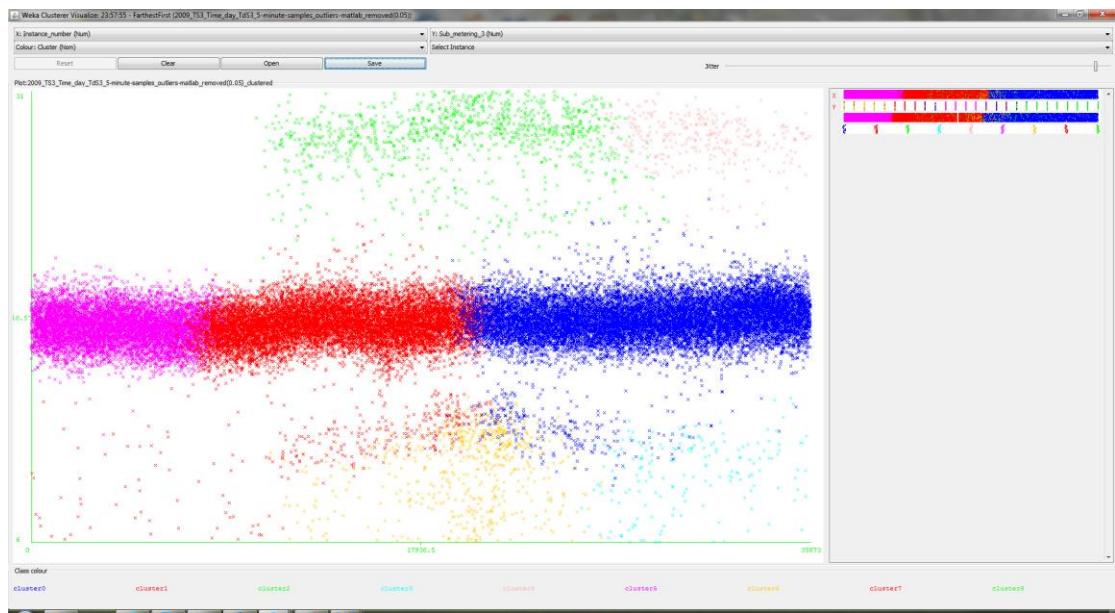
1. FarthestFirst



Εικόνα 41 Farthest_First-K4

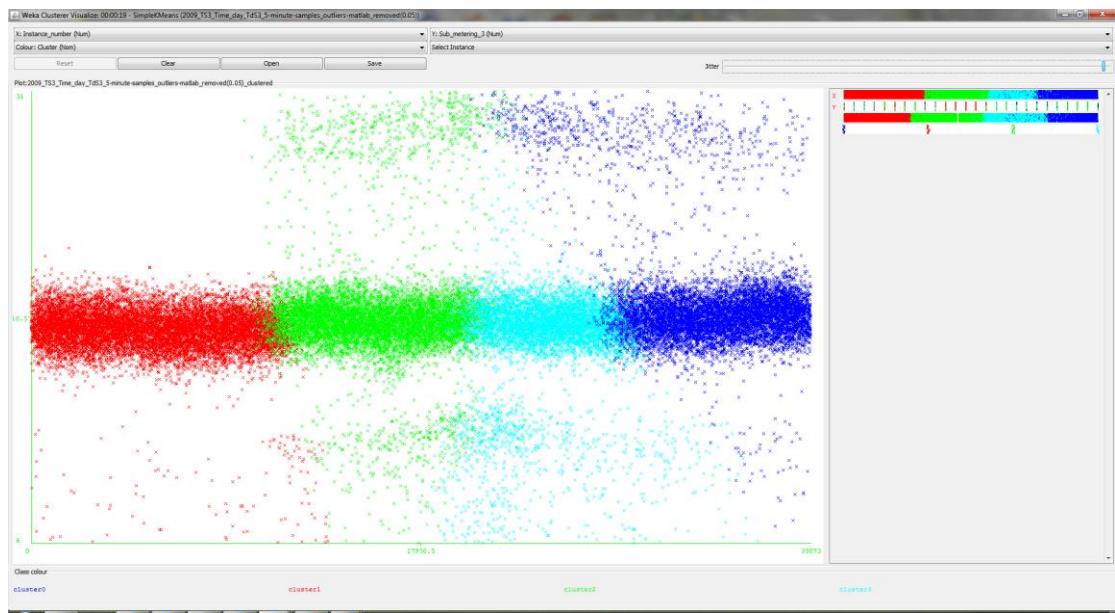


Εικόνα 42 Farthest_First-K5

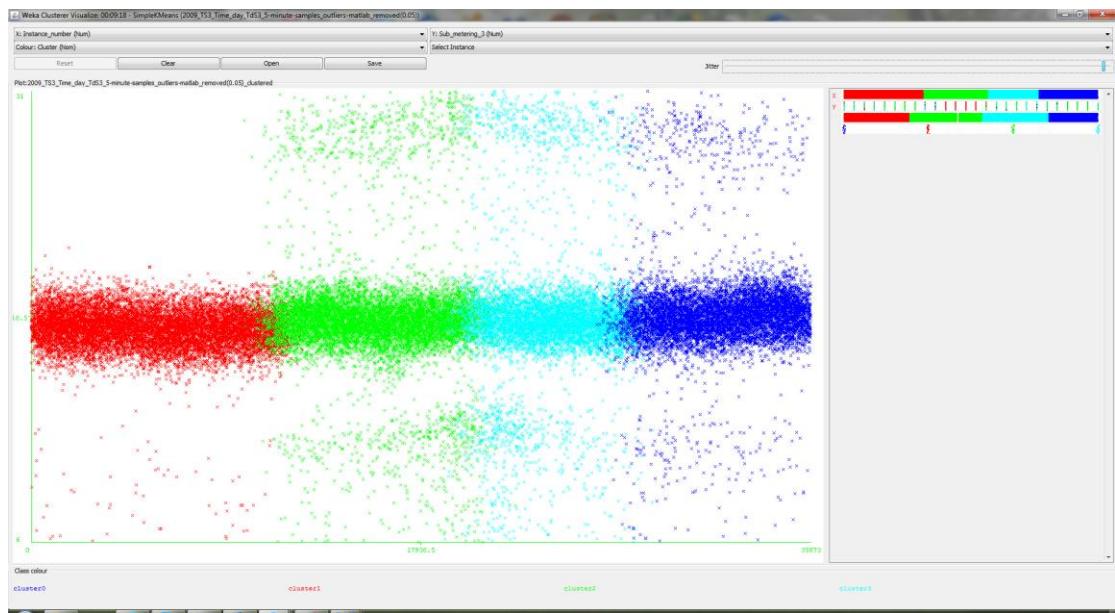


Εικόνα 43 FarthestFirst-K9

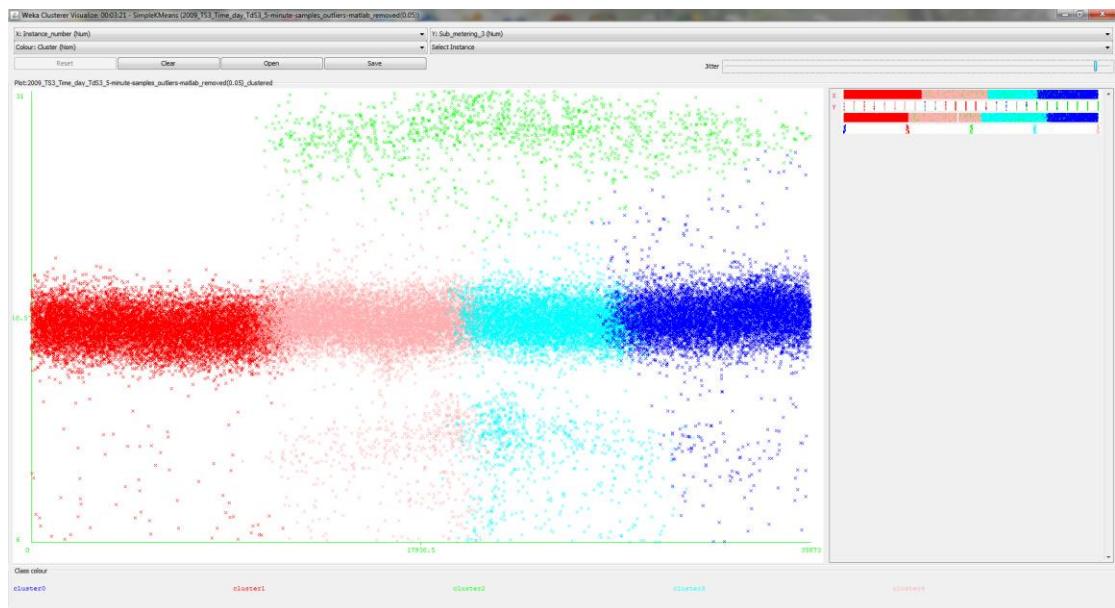
2. kmeans



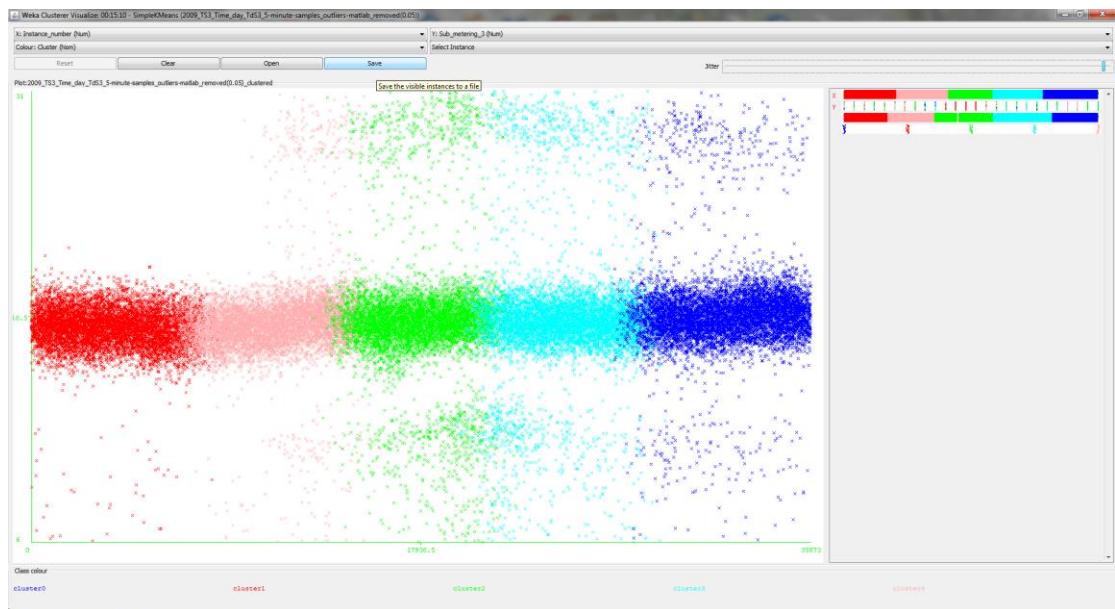
Εικόνα 44 kmeans-K4Ntrue



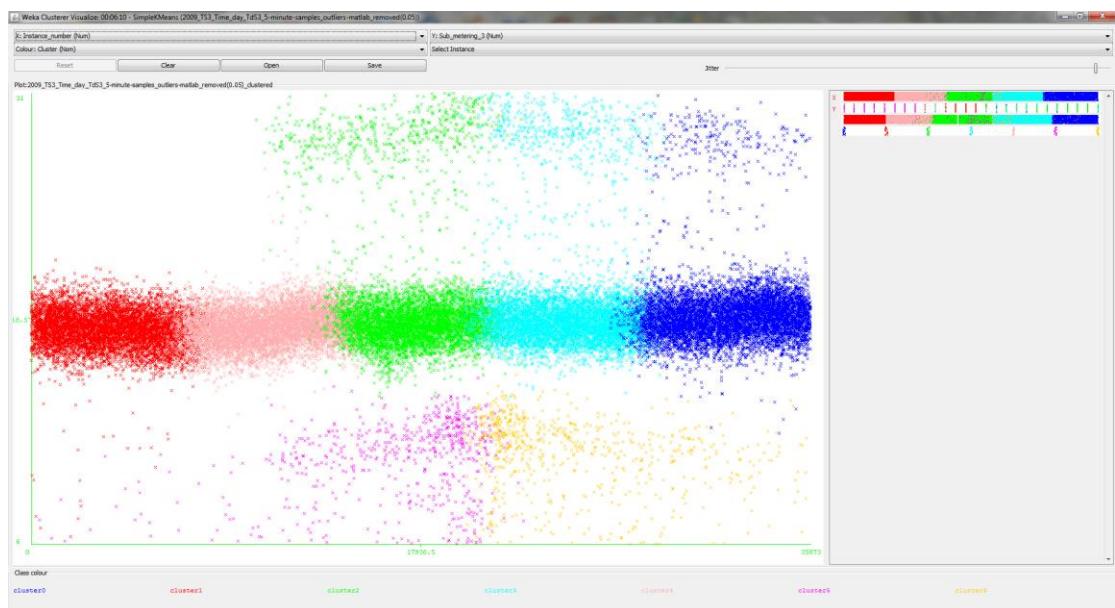
Eικόνα 45 kmeans-K4Nfalse



Eικόνα 46 kmeans-K5Ntrue

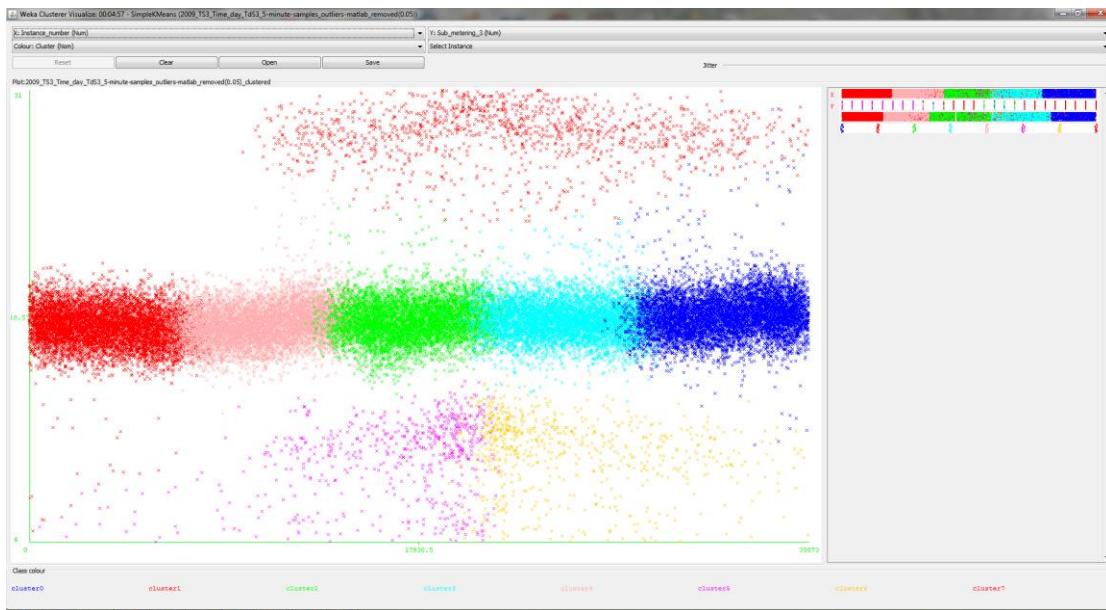


Eικόνα 47 kmeans-K5Nfalse



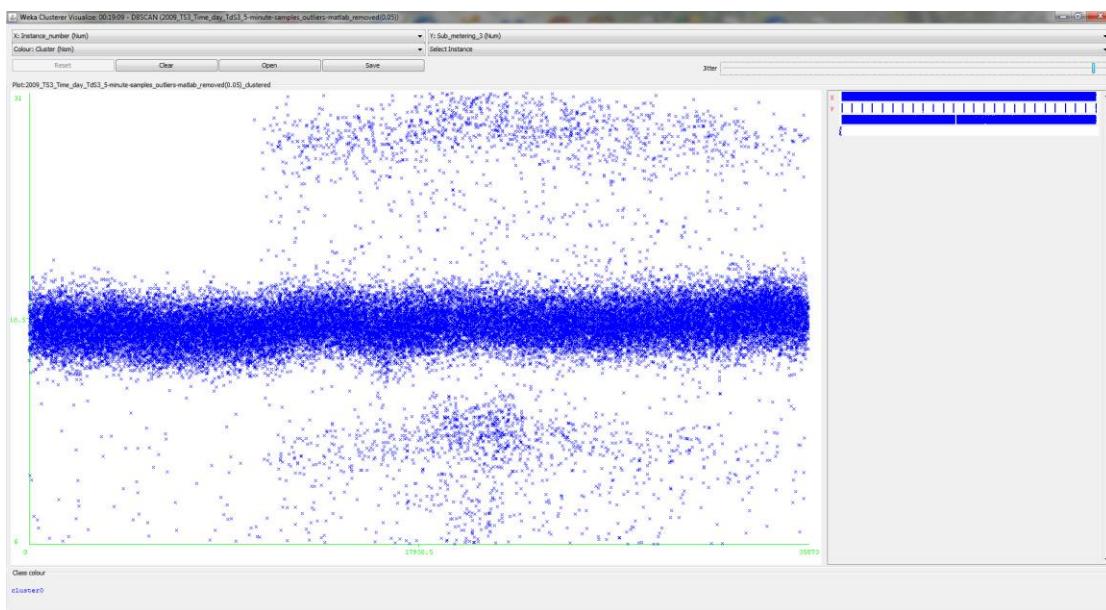
Eικόνα 48 kmeans-K7Ntrue

K=8, normalized true s=0.16427

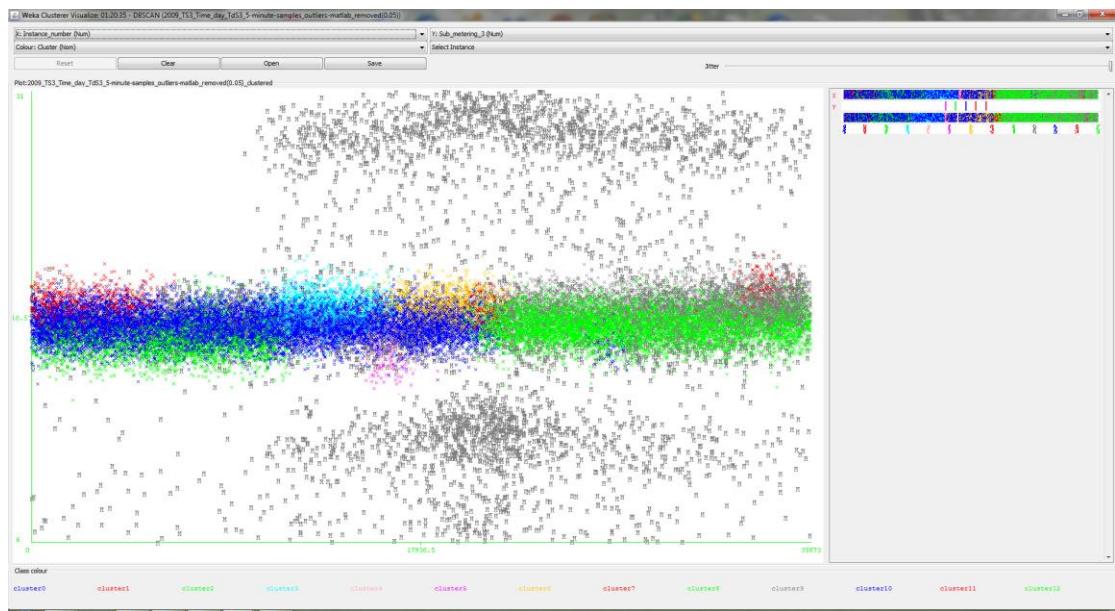


Εικόνα 49 kmeans-K8Ntrue

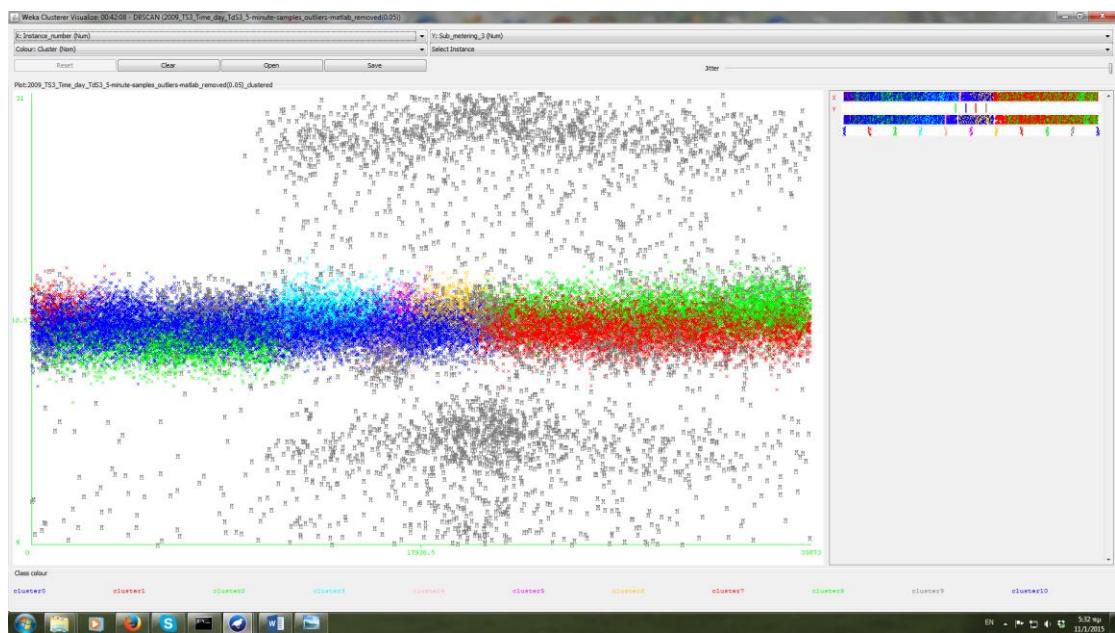
3. DBSCAN



Εικόνα 50 DBSCAN-Eps0.09MinPts6

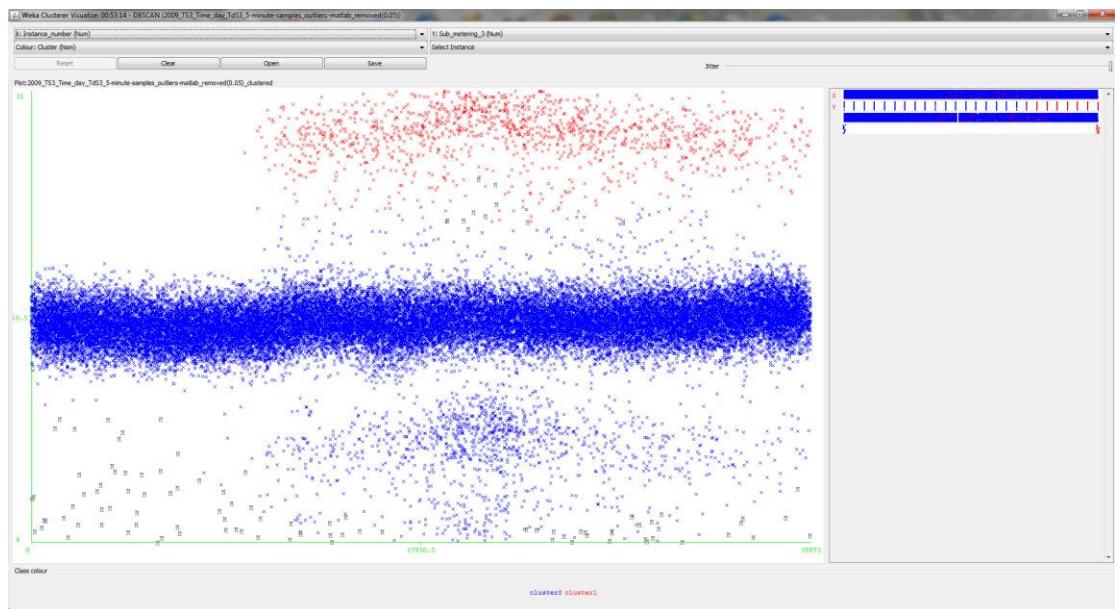


Εικόνα 51 DBSCAN-Eps=0.01MinPts=80

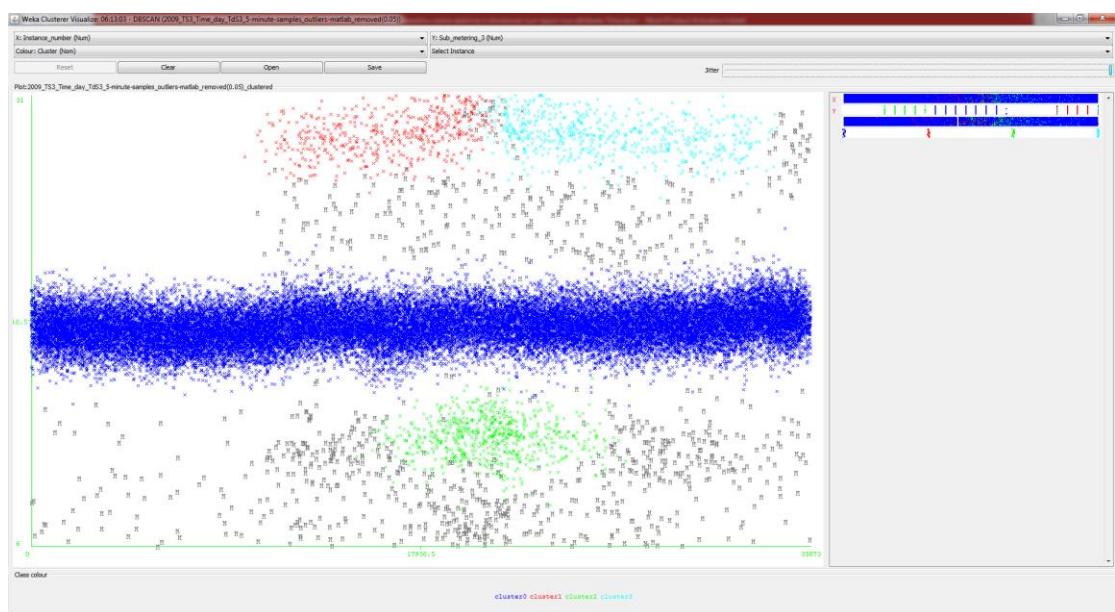


Εικόνα 52 DBSCAN-Eps=0.01MinPts=100

Eps = 0.1 MinPts = 100 s=0.10546



Εικόνα 53 DBSCAN-Eps=0.1MinPts=100

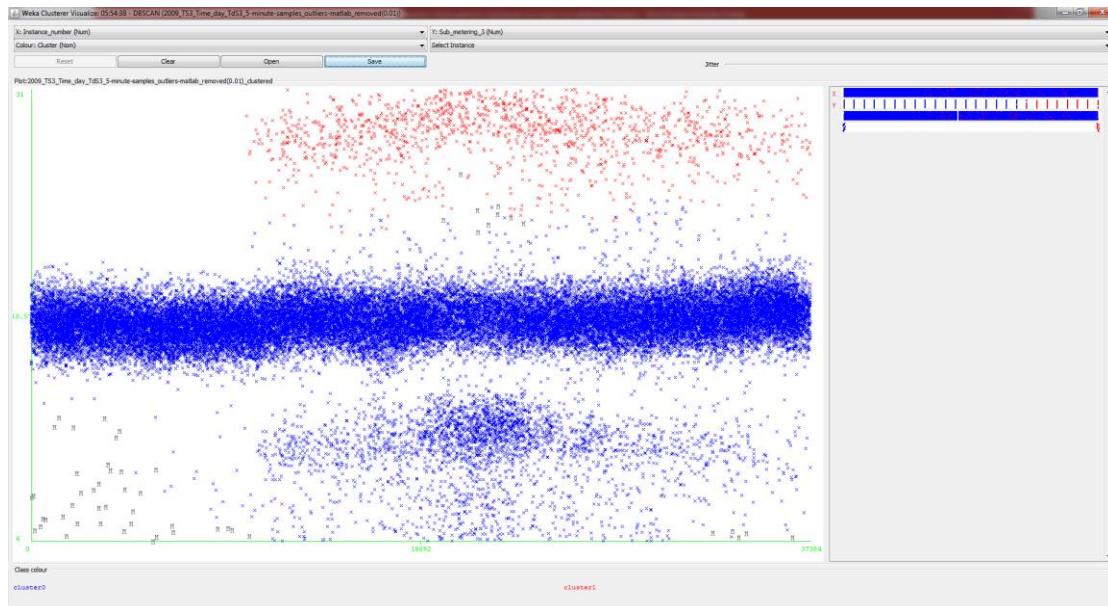


Εικόνα 54 DBSCAN-Eps=0.05MinPts=100

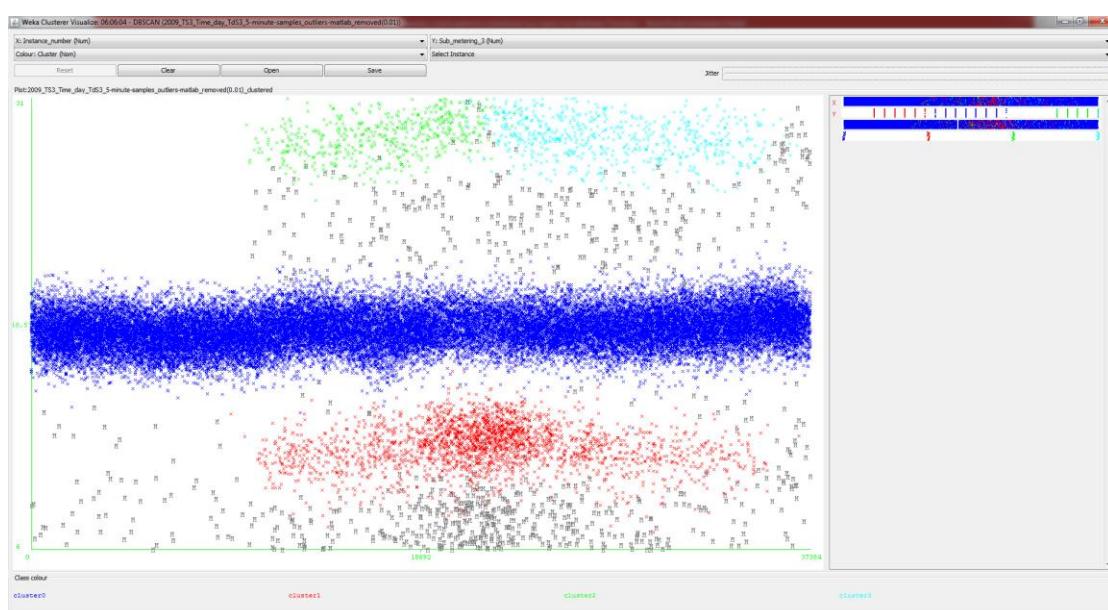
Η τελευταία ομαδοποίηση είναι ακριβώς αυτή που περιμέναμε: μία κεντρική ομάδα της συσκευής 18 Wh, μία ομάδα για την δεύτερη συσκευή σε λειτουργία 11 Wh στην χρονική ζώνη που είναι πιο έντονη και 2 ακόμη ομάδες για την δεύτερη συσκευή σε λειτουργία 29 Wh, στις 2 ζώνες που αυτή είναι πιο έντονη. Ο συντελεστής silhouette είναι .

Γενικά εξετάστηκαν συνδυασμοί για τις 2 παραμέτρους του DBSCAN στις τάξεις μεγέθους $1\text{-}10^{-3}$ για Eps, $1\text{-}10^{-3}$ για MinPts, και οι παραπάνω ομαδοποιήσεις ήταν αυτές που είχαν το μεγαλύτερο νόημα οπτικά.

Επιπλέον ο αλγόριθμος DBSCAN δοκιμάστηκε και σε dataset όπου έχει αφαιρεθεί ποσοστό outliers 1%, και έδωσε τα παρακάτω αποτελέσματα (δίνονται τα πιο χαρακτηριστικά):



Εικόνα 55 DBSCAN-Eps0.1MinPts100_(0.01)



Εικόνα 56 DBSCAN-Eps0.1MinPts100_(0.01)

Η μόνη διαφορά στις 2 τελευταίες ομαδοποιήσεις είναι ότι ο αλγόριθμος βρήκε περισσότερα noise σημεία, το οποίο είναι λογικό.

Από την παραπάνω ομαδοποίηση συμπεραίνουμε ότι ο πυκνωτικός αλγόριθμος λειτούργησε καλύτερα για τα δεδομένα, μετά όμως από αρκετή παραμετροποίηση, ενώ οι prototype-based διαχωρισμού απέτυχαν να ομαδοποιήσουν ικανοποιητικά τα δεδομένα, εκτός και αν χρησιμοποιήσουμε έναν μεγάλο αριθμό από clusters όπως π.χ το K=8 που δοκιμάσθηκε, και θεωρήσουμε ότι η ένωση μικρότερων ομάδων αποτελεί ένα πραγματικό cluster.

7. Αξιολόγηση

7.1 Εσωτερικές Μετρικές Αξιολόγησης Ομαδοποίησης

Η αξιολόγηση αλγορίθμων clustering είναι γενικά δύσκολη διαδικασία, καθώς στηρίζεται πολλές φορές σε υποκειμενικά κριτήρια. Στην δυσμενέστερη περίπτωση που δεν έχουμε καμία εξωτερική πληροφορία για την ιδιότητα των instances να ανήκουν σε κάποια κλάση, χρησιμοποιούμε μετρικές αξιολόγησης οι οποίες βασίζονται αποκλειστικά και μόνο στις τιμές των attributes, δηλαδή σε πληροφορία εσωτερική του dataset. Οι κυριότερες από αυτές είναι οι Cohesion, Separation, SSE και Silhouette.

Έχοντας ορίσει κάποια μετρική ομοιότητας ανάμεσα σε σημεία, το Cohesion του cluster C ισούται με το άθροισμα αυτών των μετρικών, για κάθε συνδυασμό σημείων μέσα στον C. Ενναλακτικά, ισούται με την ομοιότητα κάθε σημείου του cluster με το “χαρακτηριστικό” σημείο του cluster. Οπότε είναι ένας δείκτης που μετράει την συνολική συνοχή ενός cluster.

Έχοντας ορίσει κάποια μετρική ομοιότητας ανάμεσα σε σημεία, το Separation των clusters Ci, Cj ισούται με το άθροισμα αυτών των μετρικών, για κάθε συνδυασμό σημείων (x,y), όπου x ανήκει στον Ci και y ανήκει στον Cj. Ενναλακτικά, ισούται με την ομοιότητα των “χαρακτηριστικών” σημείων των Ci, Cj. Οπότε είναι ένας δείκτης που μετράει τον διαχωρισμό μεταξύ 2 cluster.

Η ομοιότητα που αναφέρεται παραπάνω μπορεί να είναι οποιοσδήποτε συνδυασμός μετρικών εγγύτητας και απόστασης, σταθμισμένων και μη. Γενικά επιθυμούμε όσο το δυνατόν καλύτερο cohesion και separation, όπου το αν το καλύτερο είναι μέγιστο ή ελάχιστο εξαρτάται από τον ορισμό της μετρικής. Σε μερικές περιπτώσεις αποδεικνύεται ότι η βελτιστοποίηση του Cohesion συνεπάγεται και βελτιστοποίηση του Separation, όπως πχ η ελαχιστοποίηση του SSE, που είναι δείκτης συνοχής και ισούται με το άθροισμα των τετραγώνων των αποστάσεων των σημείων ενός cluster από το κέντρο του (αν έχει νόημα να ορισθεί), ισοδυναμεί με την μεγιστοποίηση του SSB, το οποίο είναι δείκτης διαχωρισμού cluster και ισούται με το άθροισμα των τετραγώνων των αποστάσεων των κέντρων των cluster.

Για την αξιολόγηση των αλγορίθμων ομαδοποίησης επιλέξαμε τον δείκτη silhouette, ο οποίος συνδυάζει τόσο την έννοια της συνοχής και διαχωρισμού σε έναν δείκτη, είναι πιο ευέλικτος στον χειρισμό του, καθώς μπορεί να υπολογιστεί για κάθε σημείο, για καθένα cluster και για όλη την ομαδοποίηση, είναι υπολογιστικά αποδοτικότερος, καθώς υπάρχει έτοιμη συνάρτηση στην Matlab, και δεν χρειάζεται την έννοια του κέντρου ενός cluster, η οποία συχνά μπορεί να μην υπάρχει.

Για τον υπολογισμό του δείκτη για το σημείο ρ υπολογίζουμε την μέση απόσταση του ρ από όλα τα σημεία του cluster του, έστω a, και την ελάχιστη των μέσων αποστάσεων από κάθε άλλο cluster, έστω b. Τότε ο συντελεστής silhouette θα ισούται με $s = 1 - a/b$, για $a < b$. Αν $a > b$ το σημείο μας είναι πιο κοντά κατά μέσο όρο σε άλλα clusters παρά στο δικό του και το s προκύπτει αρνητικό. Γενικά καλή ομαδοποίηση ισοδυναμεί με s κοντά στην μονάδα.

7.2. Αξιολόγηση Clustering

Για την αξιολόγηση του clustering συγγράφηκε script στην matlab το οποίο κάνει αξιολόγηση με βάση το silhouette (evaluate_cluster.m). Η διαδικασία ήταν να αποθηκεύουμε τα αρχεία σε arff μορφή από το weka, να τα μετατρέπουμε σε csv με την εντολή

```
Ruby csv_reader.rb -e Folder
```

η οποία μετέτρεπε όλα τα αρχεία του Folder σε csv αρχεία. Στην συνέχεια τρέχαμε το script στην Matlab. Τα αποτελέσματα που πήραμε είχαν ως εξής:

Clustering	Silhouette
2008_TS1_DBSCAN-Eo.o6MP2o_(0.3)	0.82591
2008_TS1_EM_(0.3)	0.51449
2008_TS1_EM-N3_(0.3)	0.18944
2008_TS1_EM-N5_(0.3)	0.46556
2008_TS1_FarthestFirst-N2_(0.3)	0.73092
2008_TS1_FarthestFirst-N3_(0.3)	0.51056
2008_TS1_FarthestFirst-N5_(0.3)	0.3487
2008-TS1_kmeans-N3S1oI500_(0.2)	0.84636
2008-TS1_kmeans-N4S1oI1o_(0.1)	0.2855
2008-TS1_kmeans-N4S1oI500_(0.2)	0.82523
2008-TS1_kmeans-N7S1oI1o_(0.1)	0.21362
2009-TdS3_FarthestFirst-N4_(0.05)	0.12577
2009-TdS3_FarthestFirst-N5_(0.05)	0.012297
2009-TdS3_FarthestFirst-N9_(0.05)	-0.07485
2009-TdS3_kmeans-N4S1oI500_(0.05)	0.70314
2009-TdS3_kmeans-N4-RS1oI500_(0.05)	0.748
2009-TdS3_kmeans-N5S1oI500_(0.05)	0.42191
2009-TdS3_kmeans-N5-RS1oI500_(0.05)	0.74059
2009-TdS3_kmeans-N7S1oI500_(0.05)	0.30667
2009-TdS3_kmeans-N5-RS1oI500_(0.05)	0.16427
2009-TdS3_DBSCAN-Eo.9MP6_(0.05)	NaN
2009-TdS3_DBSCAN-Eo.1MP8o_(0.05)	0.11116
2009-TdS3_DBSCAN-Eo.01MP100_(0.05)	0.21781
2009-TdS3_DBSCAN-Eo.1MP100_(0.05)	0.10546
2009-TdS3_DBSCAN-Eo.05MP100_(0.05)	0.048281
2009-TdS3_DBSCAN-Eo.1MP100_(0.01)	0.097794

8. Dimensionality Reduction

Στην προσπάθειά μας να βρούμε εύλογες ομαδοποιήσεις του αρχικού dataset, αντιμετωπίσαμε το πρόβλημα της πολυδιαστατικότητας των δεδομένων (multidimensionality). Επιχειρώντας να το ξεπεράσουμε, διερευνήσαμε την δημοφιλή τεχνική μείωσης των διαστάσεων ενός πολυδιάστατου dataset PCA (Principal Component Analysis). Η κεντρική ιδέα της τεχνικής αυτής είναι η εξής: με γνωστά εργαλεία της γραμμικής άλγεβρας, απεικονίζουμε τα n -διάστατα διανύσματά μας σε έναν νέο n -διάστατο χώρο, τέτοιων ώστε

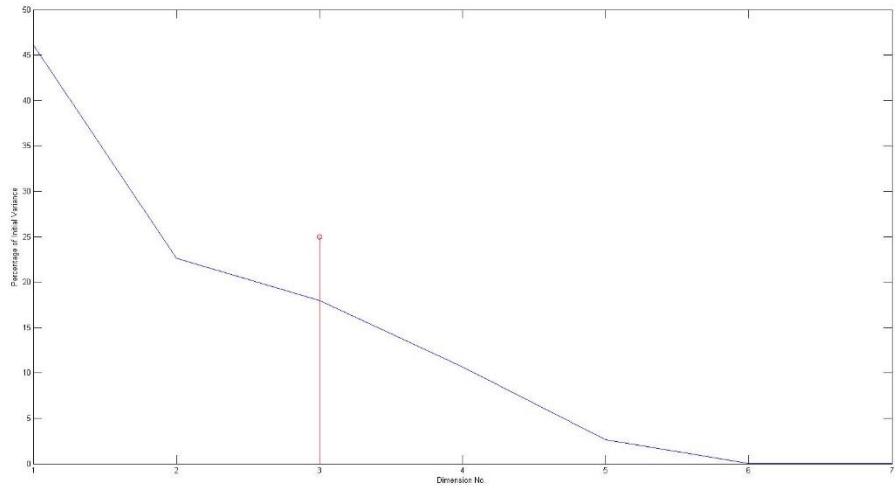
- Οι βάσεις του να είναι ορθογώνιες μεταξύ τους
- Κατά την διεύθυνση της πρώτης κατά σειρά βάσης να υπάρχει μεγαλύτερη διακύμανση στα δεδομένα από αυτήν που υπάρχει στις υπόλοιπες $n-1$ βάσεις, κατά την διεύθυνση της δεύτερης μεγαλύτερη διακύμανση στα δεδομένα από ότι σε όλες τις εναπομείναντες (πλην της πρώτης προφανώς) κοκ.

Έτσι, δημιουργούμε ουσιαστικά ένα καινούργιο dataset, το οποίο συνδέεται με μια απλή γραμμική σχέση με το αρχικό, και του οποίου το πρώτο στοιχείο αντιπροσωπεύει το 1-στο στοιχείο του αρχικού, όπως θα αναμέναμε. Για να μειώσουμε λοιπόν τις διαστάσεις του αρχικού dataset μας, αρκεί να πάρουμε τα λ πρώτα attributes του νέου dataset, με το λ να έχει επιλεχθεί έτσι ώστε το μεγαλύτερο ποσοστό της αρχικής διακύμανσης να έχει διατηρηθεί και στα καινούργια δεδομένα (κάθε νέο attribute αντιστοιχεί και σε μια βάση).

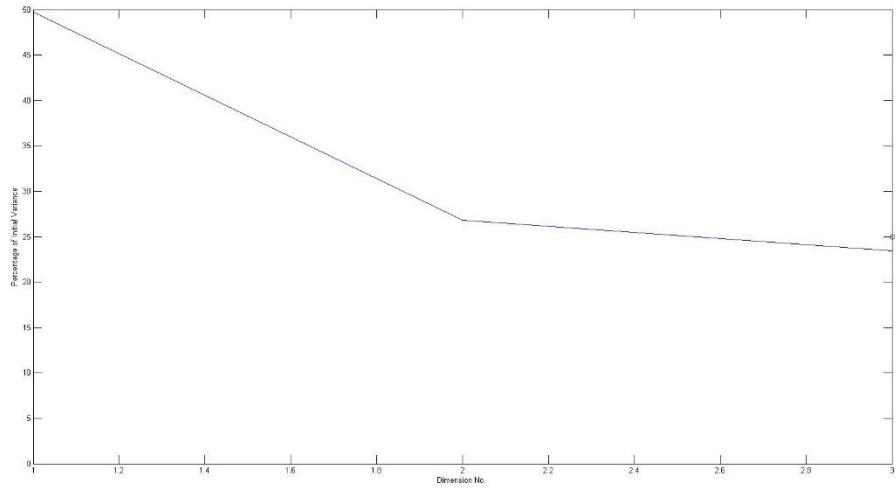
Μειονέκτημα του PCA για την εφαρμογή μας είναι ότι πρώτον η λειτουργία του επισκιάζεται από χαρακτηριστικά τα οποία έχουν πολύ μεγαλύτερη διακύμανση από όλα τα υπόλοιπα, και δεύτερον ότι για να τον χρησιμοποιήσουμε πρέπει να αφαιρέσουμε τις μέσες τιμές από κάθε attribute, κάτι που συχνά στερείται νοήματος. Δυστυχώς, ο χρόνος στο dataset μας έχει και τα δύο παραπάνω μειονεκτήματα, οπότε η παρακάτω ανάλυση έγινε χωρίς την στήλη του χρόνου.

Για την διερεύνηση του PCA χρησιμοποιήθηκε ο κώδικας `pca2_explore.m`, ο οποίος εκτυπώνει ένα γράφημα τις διακύμανσης συναρτήσει των διαστάσεων των principal components, καθώς και με κόκκινο χρώμα το σημείο όπου υπάρχει ένα επιθυμητό ποσοστό της αρχικής διακύμανσης. Το αρχικό dataset είναι το έτος 2008 με δειγματοληψία ανά 15 λεπτά.

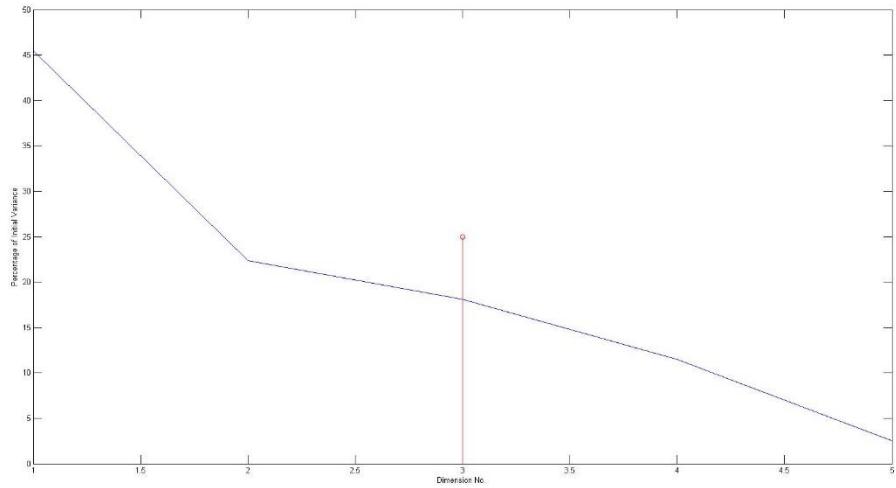
Για ποσοστό 85% :



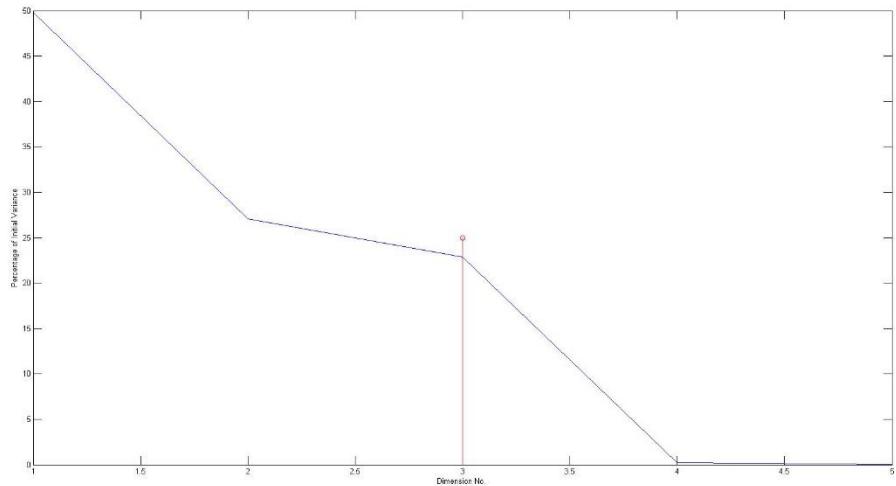
Εικόνα 57 Ολα τα attributes



Εικόνα 58 Τα 3 Sub_metering



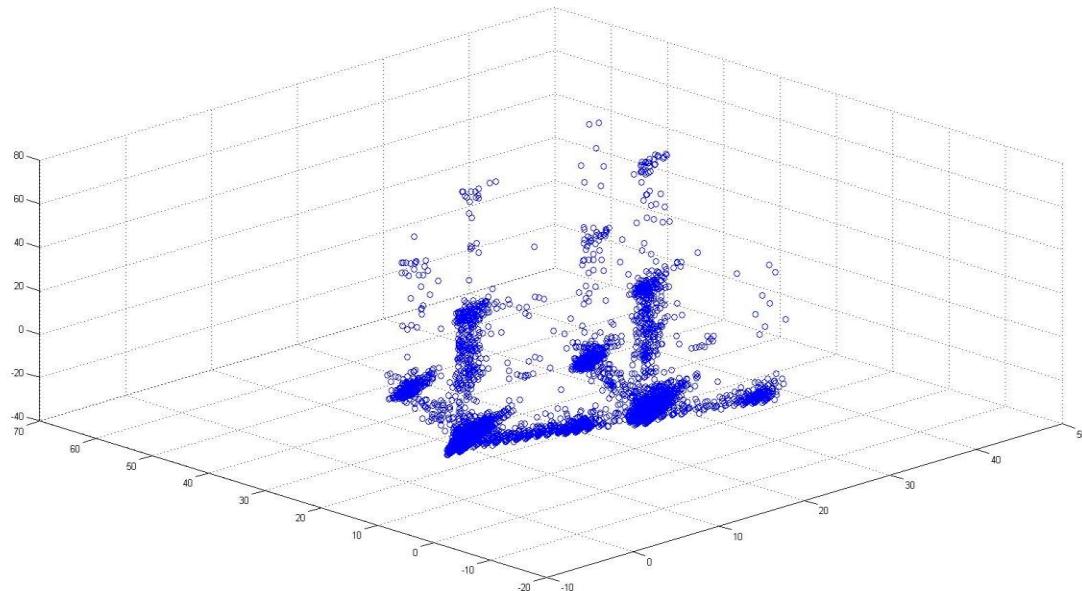
Εικόνα 60 Voltage intensity και τα 3 Sub_metering



Εικόνα 59 Global_active_power, Global_reactive_power και τα 3 Sub_metering

Μάλιστα, το *Global_active_power*, *Global_reactive_power* και τα 3 *Sub_metering* έχουν την ίδια συμπεριφορά ακόμη και για 99% ποσοστό διακύμανσης. Για μεγαλύτερα ποσοστά η εφαρμογή του PCA σε όλα τα attributes επέστρεφε 4 διαστάσεις, οι οποίες δεν θεωρήθηκαν πρακτικές στον χειρισμό τους.

Ενδεικτικά χρησιμοποιήσαμε τον αλγόριθμο στο προαναφερθέν dataset (για όλα τα attributes και ποσοστό 85%), τρέχοντας τον κώδικα *pca2.m* της Matlab. Η απεικόνιση των 3 πρώτων διαστάσεων του dataset, που περιέχουν το μεγαλύτερο ποσοστό διακύμανσης, είναι η παρακάτω:



Εικόνα 6: 3D Αναπαράσταση των δεδομένων ύστερα από PCA

Γενικά, βλέπουμε ότι σχηματίζονται ευδιάκριτες ομάδες, και ότι υπάρχει έντονη συμμετρία. Τρέχοντας ορισμένους αλγορίθμους clustering, βρήκαμε τον συντελεστή silhouette, ο οποίος δίνεται στον παρακάτω πίνακα:

Clustering Algorithm	Silhouette Coefficient
Kmeans 2	0.60875
Kmeans 4	0.81368
Kmeans 6	0.59549
DBSCAN eps=0.001 MinPts=200	0.69616
DBSCAN eps=0.001 MinPts=300	0.79734

9. Ανοιχτά Θέματα

Οι διαδικασίες της ομαδοποίησης και αξιολόγησης δεν έχουν ολοκληρωθεί. Υπάρχουν πολλά σετ δεδομένων που παρουσιάστηκαν στην ενότητα του preprocessing τα οποία δεν έχουν περάσει από

clustering και υπάρχουν και 3 ακόμα μεγάλα dataset τουλάχιστον. Στα 3 τελευταία αυτά dataset έχουν εφαρμοσθεί οι μετρικές avg,sum και max σε διάρκεια 1 μέρας σε όλο το αρχικό dataset. Προκειμένου να γίνει αυτό, το αρχικό dataset έπρεπε να σπάσει στα δύο, να εφαρμοσθούν οι μετρικές τοπικά:

```
ruby csv_reader.rb -m avg,day,1 <dataset>
ruby csv_reader.rb -m sum,day,1 <dataset>
ruby csv_reader.rb -m max,day,1 <dataset>
```

και στην συνέχεια να ενωθούν τα dataset:

```
ruby csv_reader.rb -j <dataset2> <dataset1>
```

Τέλος κάναμε sampling για να μειώσουμε το μέγεθος του dataset, με δείγματα μιας μέρας:

```
ruby csv_reader.rb -s day,1 <dataset>
Whole day (1440 minutes) or 1 minute (11:59P.M.)? w/o
o
```

Επίσης το ίδιο το csv_reader.rb επιδέχεται πολλές βελτιώσεις από άποψη καλύτερης διαχείρισης πόρων.

Τα καλύτερα μοντέλα προέκυψαν με βάση τις οπτικές αναπαραστάσεις των ομαδοποιήσεων, η αξιολόγησή τους δεν έχει ολοκληρωθεί.