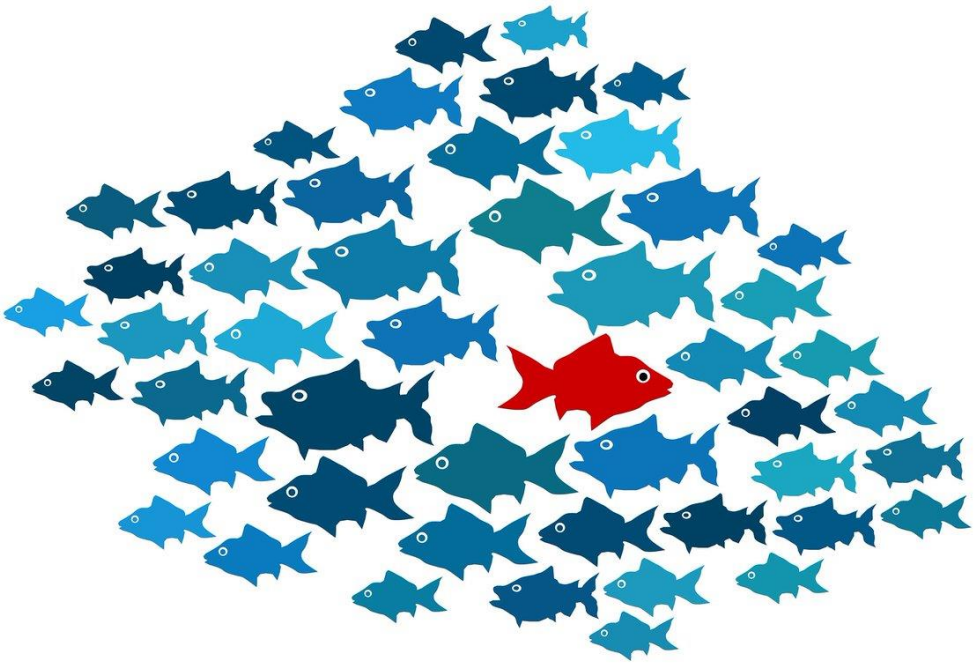
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

SCHOOL OF INFORMATICS AUTH

Πτυχιακή εργασία

Τεχνικές εντοπισμού ανωμαλιών σε δεδομένα: Μελέτη, αξιολόγηση και εφαρμογές μεθόδων σε σύγχρονα συστήματα

Anomaly detection techniques: study, evaluation and application of methods in modern systems



Κιζιρίδης Κωνσταντίνος – Α.Μ. 3566

Επιβλέπων καθηγητής

Γούναρης Αναστάσιος, Αναπληρωτής καθηγητής

Το παρακάτω κείμενο αποτελεί προϊόν προσωπικής έρευνας και συγγραφής. Οποιαδήποτε χρήση παραπομπών από συγγράμματα, επιστημονικά άρθρα και πηγές αναφέρεται σε διακριτικά σημεία, καθώς και συγκεντρωτικά στο τέλος του παρόντος κειμένου.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Αναστάσιο Γούναρη, αναπληρωτή καθηγητή του τμήματος πληροφορικής του Α.Π.Θ. για την ευκαιρία που μου έδωσε να μελετήσω και να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα καθώς και για την καλή συνεργασία μας στο πλαίσιο της πτυχιακής μου εργασίας.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου για την στήριξη και την βοήθεια που μου παρείχαν στην ακαδημαϊκή μου πορεία.

Πίνακας περιεχομένων:

1.Εισαγωγή

Το επιστημονικό πεδίο του εντοπισμού ανωμαλιών σε δεδομένα αποτελεί ένα συνεχώς εξελισσόμενο πεδίο για τους επιστήμονες της πληροφορικής. Παλαιότερα κατά την ανάλυση δεδομένων, όταν οι αναλυτές συναντούσαν τιμές στα δεδομένα, οι οποίες θεωρούνταν περίεργες και μη συμβατές σε σχέση με άλλες τιμές του συνόλου συνήθως τις απέρριπταν κατά το στάδιο της προ – επεξεργασίας. Αυτό γινόταν, καθώς τέτοιες τιμές δυσχεραίναν τα μοντέλα εκμάθησης που αναπτύσσονταν και δεν θεωρούταν χρήσιμα.

Όμως, με το πέρασμα του χρόνου όπου οι πηγές των δεδομένων άρχισαν να αυξάνονται ραγδαία εμφανίστηκε ιδιαίτερο ενδιαφέρον στην μελέτη των ανώμαλων τιμών και συγκεκριμένα για το τι είδους γνώση μπορούν αυτά να προσφέρουν. Έτσι, με την εξέλιξη των υπολογιστικών πόρων που έδωσαν την δυνατότητα της επεξεργασίας μεγάλης ποσότητας δεδομένων ξεκίνησε η ανάπτυξη μεθόδων εντοπισμού και αξιολόγησης των τιμών αυτών.

Στο πλαίσιο του θέματος της παρούσης εργασίας ο σκοπός είναι να μελετηθούν οι υπάρχουσες μέθοδοι εντοπισμού ανωμαλιών σε σύνολα δεδομένων, να προταθούν υλοποιήσεις των μεθόδων αυτών με ανάπτυξη κώδικα καθώς και αναδειχθεί το πως σύγχρονα εργαλεία μπορούν να ενσωματώσουν τα ευρήματα του πεδίου αυτού για την δημιουργία χρήσιμων ερφαρμογών που μπορούν να χρησιμοποιηθούν σε πραγματικά προβλήματα.

Δομή του κειμένου

Προτού μεταβούμε στο κύριο σώμα του κειμένου παρατίθεται η δομή που αυτό θα ακολουθήσει.

Αρχικά, προκειμένου να γίνει αντιληπτή η σημαντικότητα του πεδίου αυτού θα παρατεθούν ορισμένα παραδείγματα εφαρμογών που απορρέουν απ' τον εντοπισμό ανωμαλιών σε συνθήκες πραγματικού κόσμου. Έπειτα, θα δοθούν βασικές ορολογίες, οι οποίες είναι απαραίτητες για την κατανόηση των εννοιών που θα αναλυθούν.

Παρακάτω, αναλύονται οι 2 βασικές κατηγορίες μεθόδων εντοπισμού ανωμαλιών σε δεδομένα (Μέθοδοι εντοπισμού σε στατικά σύνολα δεδομένων και μέθοδοι εντοπισμού σε χρονοσειρές δεδομένων, δηλαδή σε δεδομένα στα οποία λαμβάνεται υπόψιν η έννοια του χρόνου.) Αφού ολοκληρωθούν τα παραπάνω τμήματα του κειμένου θα δοθούν υλοποιήσεις μεθόδων εντοπισμού ανωμαλιών ανεπτυγμένες σε γλώσσα προγραμματισμού Python. Οι υλοποιήσεις αυτές αποτελούν μία προσέγγιση για το πως θα μπορούσε το Anomaly Detection να χρησιμοποιηθεί σε ένα πραγματικό σύστημα.

Τέλος θα παρουσιαστεί το σύστημα Apache Pinot – Thirdeye, το οποίο αποτελεί πραγματικό σύστημα που ενσωματώνει τέτοιες τεχνολογίες. Ο σκοπός της παρουσίασης του συστήματος είναι η εύρεση των μεθόδων που αξιοποιούνται σε πραγματικά συστήματα, η συγκρισή τους με τις προηγούμενες υλοποιήσεις και η παράθεση προτάσεων για το πως αυτά μπορούν να γίνουν καλύτερα.

2.Εφαρμογές εντοπισμού ανωμαλιών

Προτού αρχίσει η βαθύτερη μελέτη του θέματος κρίθηκε σκόπιμη η αναφορά της προσφοράς του πεδίου στην καθημερινότητα. Όπως θα διαπιστώσετε, το πεδίο βρίσκει εφαρμογές σε ποικίλες πτυχές της καθημερινότητας. Παρακάτω παρατίθενται μερικές απ' αυτές:

Επιστήμη της Ιατρικής: Ο εντοπισμός ανώμαλων τιμών μπορεί να προσφέρει σημαντικές πληροφορίες σε γιατρούς τόσο στο επίπεδο διάγνωσης ασθενειών όσο και στην θεραπεία ασθενών. Μέσω της ανάλυσης δεδομένων που προέρχονται από εξετάσεις ασθενών μπορούν να αναγνωριστούν μοτίβα που σχετίζονται με συγκεκριμένες ασθένειες γεγονός, το οποίο μπορεί να ενισχύσει την πρόληψη. Επίσης κατά τη διάρκεια θεραπειών η χρήση τεχνικών εντοπισμού ανωμαλιών μπορεί να συνεισφέρει στην διαπίστωση εάν οι ασθενείς ανταποκρίνονται θετικά στις αγωγές δίνοντας καίριες πληροφορίες στους θεράποντες ιατρούς.

Επιστήμη Περιβάλλοντος: Οι εφαρμογές του πεδίου μπορούν να συνεισφέρουν στην διατήρηση οικοσυστημάτων και στην παροχή βοήθειας σε γεωργούς μέσω ανάλυσης μετρήσεων από αισθητήρες (π.χ. θερμοκρασίας, υγρασίας, αέρα κ.τ.λ.). Η αναγνώριση ανώμαλων τιμών μπορεί να δώσει πληροφορίες για επερχόμενα καιρικά φαινόμενα για τα οποία μπορεί να χρειαστεί ανθρώπινη παρέμβαση για την προστασία οικοσυστημάτων ή σοδειών.

Κυβερνο – ασφάλεια / Αναγνώριση απάτης: Ένας ακόμη τομέας όπου βρίσκονται ιδιαίτερα χρήσιμες εφαρμογές είναι η αναγνώριση ηλεκτρονικής απάτης και της κυβερνο – ασφάλειας. Πλέον, όλο και περισσότερες χρηματικές συναλλαγές πραγματοποιούνται μέσω του διαδικτύου. Το γεγονός αυτό επιφέρει πολλούς κινδύνους για τους χρήστες, διότι ευαίσθητα δεδομένα μπορούν να διαρρεύσουν (π.χ. στοιχεία τραπεζικών λογαριασμών). Συνεπώς, μέσω ανάλυσης αρχείων τραπεζικών συναλλαγών οι πάροχοι μπορούν να εντοπίζουν ύποπτες κινήσεις και να προστατεύουν τους πελάτες τους από επιτήδειους που επιχειρούν να τους βλάψουν.

3.Εισαγωγή στις έννοιες και βασική ορολογία

Στο παρόν τμήμα του κειμένου θα παρουσιαστούν οι βασικές έννοιες και ορολογίες του πεδίου, οι οποίες θα χρησιμοποιηθούν κατά κόρον στην συνέχεια.

Ορισμός ανώμαλης τιμής

Προς το παρόν έχουν γίνει πολλές αναφορές στον όρο «ανώμαλη τιμή», ωστόσο δεν έχει δοθεί ένας σαφής ορισμός ως προς το τι καθιστά μία εγγραφή ενός συνόλου δεδομένων ανώμαλη. Πιο συγκεκριμένα, μία τιμή για ένα χαρακτηριστικό σε ένα σύνολο δεδομένων μπορεί να κριθεί ως ανώμαλη, όταν αυτή παρουσιάζει μεγάλη διαφοροποίηση σε σχέση με αντίστοιχες τιμές άλλων εγγραφών. Η διαφοροποίηση μεταξύ των εγγραφών μπορεί να βρεθεί μέσω διαφόρων τεχνικών και προσεγγίσεων που θα αναλυθούν παρακάτω. Ένας εναλλακτικός χαρακτηρισμός των τιμών αυτών είναι «ασυνεπής τιμή», ενώ στην αγγλική βιβλιογραφία αναφέρονται ως “anomaly” ή “outlier”.

Μοντέλα εντοπισμού ανωμαλιών

Γενικά, υπάρχουν τρία κύρια μοντέλα εκμάθησης αναγνώρισης ανωμαλιών, που ακολουθούν τις τάσεις του πεδίου της μηχανικής μάθησης. Πιο συγκεκριμένα:

1. **Μοντέλο μάθησης με επίβλεψη/ supervised learning:** Στο συγκεκριμένο μοντέλο μάθησης χρησιμοποιείται ένα σύνολο εκπαίδευσης, το οποίο περιέχει δεδομένα που ανήκουν τόσο σε κανονικές όσο και ανώμαλες εγγραφές και είναι γνωστό για κάθε εγγραφή σε ποια εκ των δύο κλάσεων ανήκει. Στην συνέχεια, τα νέα δεδομένα κατηγοριοποιούνται με βάση το σύνολο εκπαίδευσης.
2. **Μοντέλο μάθησης με ημι-επίβλεψη /semi – supervised learning:** Στο συγκεκριμένο μοντέλο χρησιμοποιείται ένα σύνολο εκπαίδευσης, το οποίο περιέχει μόνο στιγμιότυπα της φυσιολογικής κλάσης και άρα το μοντέλο μαθαίνει να αναγνωρίζει τα φυσιολογικά στιγμιότυπα. Έτσι, σε νέα δεδομένα βρίσκεται σε θέση να βρει τα ανώμαλα.

3. **Μοντέλο μάθησης χωρίς επίβλεψη /unsupervised learning:** Στο συγκεκριμένο μοντέλο δεν υπάρχει σύνολο εκπαίδευσης και τα δεδομένα κατηγοριοποιούνται με αλγοριθμικό τρόπο. Οι αλγόριθμοι που χρησιμοποιούνται έχουν διάφορες προσεγγίσεις και είναι αυτοί που θα μας απασχολήσουν κυρίως στην συνέχεια του κειμένου.

Αποτελέσματα μεθόδων εντοπισμού

Κατά την εφαρμογή των μεθόδων εντοπισμού ανωμαλιών υπάρχουν 2 διαφορετικά αποτελέσματα που μπορούν να επιστραφούν και σχετίζονται άμεσα με το μοντέλο εκμάθησης που ακολουθεί η μέθοδος. Οι μέθοδοι επιστρέφουν είτε ετικέτες (labeling) όπου καθορίζεται ξεκάθαρα εάν μία εγγραφή θεωρείτε ανώμαλη ή όχι είτε βαθμολογία (score) όπου σε κάθε εγγραφή αποδίδεται με μαθηματικό υπολογισμό μία τιμή που καθορίζει κατά πόσο είναι μία εγγραφή ανώμαλη. Στην δεύτερη περίπτωση τίθεται μία τιμή κατωφλίου για την οποία θεωρούμε, πως όταν ξεπεραστεί τότε η εγγραφή είναι ανώμαλη.

Η χρήση ετικετών σχετίζεται συνήθως με μεθόδους βασισμένες σε επιβλεπόμενη ή ήμι-επιβλεπόμενη μάθηση καθώς θυμίζουν περισσότερο μεθόδους κατηγοριοποίησης. Αντίθετα, σε μεθόδους που ανήκουν σε μάθηση χωρίς επίβλεψη επειδή δεν μπορεί να δοθεί σίγουρη απάντηση σύμφωνα με κάποιο σύνολο εκμάθησης χρησιμοποιείται η βαθμολόγηση.

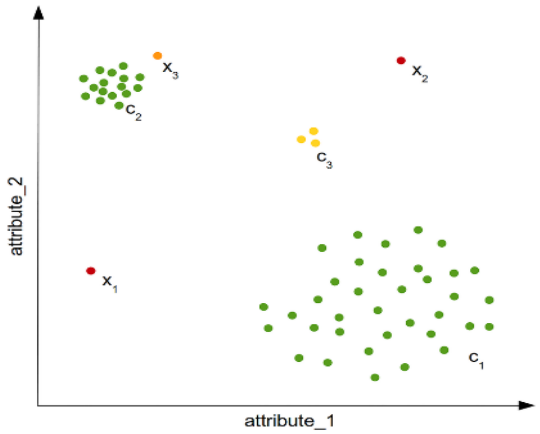
Τύποι ανώμαλων τιμών

Οι ανώμαλες τιμές ανάλογα με τα χαρακτηριστικά τους μπορούν να κατηγοριοποιηθούν στις παρακάτω κατηγορίες.

- **Global anomaly/ Καθολική ανωμαλία:** Ονομάζεται μία τιμή, η οποία διακρίνεται πως είναι ασυνεπής συγκριτικά με όλες τις υπόλοιπες εγγραφές του συνόλου.
- **Local anomaly/ Τοπική ανωμαλία:** Ονομάζεται μία τιμή, η οποία εάν τη δεις σε σχέση με ολόκληρο το σύνολο δεν μπορεί να κριθεί ως ανώμαλη, αλλά εάν παρατηρηθεί σε σχέση με τα πιο «κοντινά» της σημεία, τότε μπορεί να χαρακτηριστεί ως ανώμαλη.

- **Micro-cluster anomaly/ Ανωμαλία μικρο-συστάδας:** Η συγκεκριμένη κατηγορία αφορά σε ανωμαλίες οι οποίες εμφανίζονται πολύ κοντά μαζί τους και σχηματίζουν μικρού μεγέθους συστάδες.

Οι παραπάνω κατηγορίες μπορούν να γίνουν καλύτερα κατανοητές μέσω του παρακάτω διαγράμματος.



(Εικόνα 1 απ' το άρθρο νούμερο 1)

Παραπάνω μπορούμε να κάνουμε τις εξής παρατηρήσεις. Τα σημεία Χ1, Χ2 αποτελούν καθολικές ανωμαλίες. Το σημείο Χ3 αποτελεί τοπική ανωμαλία καθώς διακρίνεται ανώμαλη σε σχέση με την συστάδα C2 που βρίσκεται κοντά της και όχι σε όλο το σύνολο. Η συστάδα C3 ανήκει στην Τρίτη κατηγορία καθώς πολλές ανώμαλες τιμές σχηματίζουν μαζί μία ομάδα.

Επίσης, από τις παραπάνω κατηγορίες προκύπτουν και κατηγορίες για τις μεθόδους αναζήτησης. Αναλυτικότερα:

- **Point anomaly detection:** Η point anomaly detection αφορά σε μεθόδους, που αναζητούν ανωμαλίες που αποτελούνται από ανεξάρτητες εγγραφές.
- **Collective anomaly detection:** Η collective anomaly detection αφορά σε μεθόδους που αναζητούν ανωμαλίες που αποτελούνται από πολλαπλές εγγραφές, όπως είναι οι ανωμαλίες μικρό-συστάδων

- **Contextual anomaly detection:** Η contextual anomaly detection εισάγει στις μεθόδους και την έννοια της σημασιολογίας των μεταβλητών. Για παράδειγμα, έστω ότι σε ένα σύνολο δεδομένων καταχωρούνται πληροφορίες για τον πληθυσμό συγκεκριμένων δήμων της Ελλάδας. Εάν έχουμε μία καταχώρηση ότι σε μία περιοχή υπάρχουν 5.000 κάτοικοι, αυτό μπορεί να κριθεί λάθος ανάλογα με τον δήμο που αφορά. Αν η καταχώρηση αναφέρεται σε δήμο κάποιου ορεινού χωριού, τότε η τιμή φαίνεται ρεαλιστική, αλλά εάν αφορούσε στον δήμο Θεσσαλονίκης που γνωρίζουμε ότι είναι απ' τους πολυπληθέστερους της χώρας, τότε η συγκεκριμένη εγγραφή είναι σίγουρα λανθασμένη. Η κατηγορία αυτή τονίζει το πόσο σημαντικό είναι πέρα απ' την ανάλυση δεδομένων να υπάρχει και η κατανόησή τους.

<u>Τύποι δεδομένων</u>

Τα δεδομένα, τα οποία καλούνται να αναλύσουν οι μέθοδοι εντοπισμού έχουν και αυτά τα δικά τους χαρακτηριστικά. Τα χαρακτηριστικά τους έχουν να κάνουν με τις διαστάσεις που μπορούν να έχουν, αλλά και τον τρόπο που αυτά έρχονται.

Όσον αφορά τις διαστάσεις, τα δεδομένα μπορεί να είναι από μονοδιάστατα έως και πολλών διαστάσεων. Όταν είναι πολλών διαστάσεων σημαντικό ρόλο κατέχει το στάδιο της προεπεξεργασίας τους, καθώς οι πολλές διαστάσεις συνεπάγονται με αυξημένη δυσκολία ανάλυσης.

Όσον αφορά τον τρόπο που λαμβάνονται τα δεδομένα, υπάρχει ένας μεγάλος διαχωρισμός με βάση των οποίο γίνεται η λήψη τους. Ειδικότερα, τα δεδομένα μπορεί να είναι είτε στατικά (π.χ. μία βάση δεδομένων συναλλαγών μίας εταιρείας) ή χρονικά μεταβαλλόμενα real – time δεδομένα που τροφοδοτούνται μέσω κάποιου συστήματος.

Ο τρόπος χειρισμού των δεδομένων ανάλογα με τον τρόπο που αυτά γίνονται διαθέσιμα διαφέρει πολύ και θα αποτελέσει βάση για την συνέχεια του κειμένου.

4.Μέθοδοι εντοπισμού ανωμαλιών σε στατικά δεδομένα

(Στην ενότητα 4 χρησιμοποιήθηκαν πληροφορίες απ' τα 1, 2)

Στην ενότητα αυτή θα μελετηθούν μέθοδοι που αφορούν σε στατικά σύνολα δεδομένων. Οι μέθοδοι που θα παρουσιαστούν υπάγονται σε κατηγορίες βασισμένες ανάλογα με τη φύση των αλγορίθμων και των σκεπτικών πάνω στα οποία έχουν αναπτυχθεί.

Οι κατηγορίες:

1. Nearest Neighbor based methods / Μέθοδοι βασισμένες στο σκεπτικό των πλησιέστερων γειτόνων
2. Clustering based / Μέθοδοι που βασίζονται στο σκεπτικό αλγορίθμων συσταδοποίησης
3. Statistical based methods / Μέθοδοι που βασίζονται στη στατιστική
4. Subspace based methods / Μέθοδοι που βασίζονται σε subspacing
5. Classifier based methods / Μέθοδοι που βασίζονται σε κατηγοριοποίηση

4.1 Nearest Neighbor based methods

Υπάρχουν διάφορες μέθοδοι βασισμένες στο Nearest Neighbor. Γενικά, σε όλες αυτές υπάρχει μία ευαισθησία ως προς τον προσδιορισμό του κατάλληλου αριθμού γειτόνων (k), καθώς και της τιμής κατωφλίου απ' την οποία θα καθορίζεται αν μία εγγραφή είναι outlier ή όχι. Το γεγονός αυτό απαιτεί πολλαπλές δοκιμές με διάφορες τιμές για τις μεταβλητές αυτές με σκοπό να υπάρχει όσο το δυνατόν μεγαλύτερη βεβαιότητα για τα αποτελέσματα.

Global anomaly methods

Στις μεθόδους που βασίζονται στην λογική των πλησιέστερων γειτόνων υπάρχουν 2 που αφορούν σε εντοπισμό καθολικών ανωμαλιών. Αυτοί είναι οι k-NN και kth-NN. Και οι δύο ακολουθούν το ίδιο σκεπτικό, αλλά κατά τον υπολογισμό της βαθμολογίας υπάρχει μια διαφοροποίηση.

Πιο συγκεκριμένα οι μέθοδοι ξεκινούν με τον εντοπισμό των k πλησιέστερων γειτόνων για κάθε μία από τις εγγραφές του εξεταζόμενου συνόλου. Έπειτα αποδίδεται η βαθμολογία ως εξής.

Στην k-NN: Αποδίδεται ως βαθμολογία ο μέσος όρος των αποστάσεων του σημείου από τους πλησιέστερους του γείτονες.

Στην kth-NN: Αποδίδεται ως βαθμολογία η απόσταση του σημείου από τον kth γείτονα

Local anomaly methods

Για τον εντοπισμό των τοπικών ανωμαλιών οι κυριότερες μέθοδοι είναι οι εξής:

Local Outlier Factor (LOF)

Η συγκεκριμένη μέθοδος χρησιμοποιεί έναν μαθηματικό τρόπο με σκοπό να υπολογίσει τοπικούς λόγους πυκνοτήτων των εγγραφών του συνόλου δεδομένων.

Βήμα 1ο : Αρχικά, για κάθε σημείο βρίσκονται οι k πλησιέστεροι γείτονες.

Βήμα 2ο :Στην συνέχεια για κάθε σημείο υπολογίζεται το Local Reachability Density ή αλλιώς LRD με βάση τον τύπο:

$$LRK_k(X) = \frac{1}{\frac{\sum_{o \in N_k(x)} d_k(x,o)}{|N_k(x)|}}$$

Ερμηνεία τύπου: Το σύνολο $N_k(x)$ αποτελεί το σύνολο των πλησιέστερων γειτόνων ενός σημείου χ. Για κάθε σημείο βρίσκουμε το άθροισμα των αποστάσεων των γειτόνων του απ' αυτό διαιρεμένο με το πλήθος των γειτόνων. Έπειτα αντιστρέφουμε αυτή την τιμή.

Σημείωση: Στην μέθοδο αυτή χρησιμοποιείται η ευκλείδεια απόσταση.

Βήμα 3ο : Υπολογίζω την τιμή του LOF ως εξής:

$$LOF_x = \frac{\sum_{o \in N_k(x)} \frac{LRK_k(o)}{LRK_k(X)}}{|N_k(x)|}$$

Ερμηνεία τύπου: Για τον υπολογισμό του LOF βρίσκω το άθροισμα των λόγων του LRK των γειτόνων ενός σημείου προς το LRK του ίδιου του σημείου και στην συνέχεια το άθροισμα αυτό διαιρείται με το πλήθος των γειτόνων.

Ερμηνεία αποτελεσμάτων της μεθόδου:

Παρατηρώντας τον υπολογισμό των τιμών γίνεται κατανοητό, ότι η μέθοδος υπολογίζει τοπικούς λόγους πυκνοτήτων των σημείων με βάση τους πλησιέστερους γείτονές τους. Όταν μία εγγραφή είναι κανονική το αναμενόμενο LOF score της είναι μία τιμή που βρίσκεται κοντά στην μονάδα. Αυτό γίνεται καθώς εάν μία τιμή είναι φυσιολογική συνήθως βρίσκεται ανάμεσα σε άλλες κανονικές εγγραφές. Άρα και οι πλησιέστεροι γείτονές της θα είναι κατά πάσα πιθανότητα κανονικοί και θα έχουν παραπλήσιες πυκνότητες. Συνεπώς το άθροισμα των λόγων προς το πλήθος των γειτόνων που υπολογίζεται στο LOF πρέπει να είναι κοντά στην μονάδα. Αντιθέτως, εγγραφές που είναι ανώμαλες και δεν περιτριγυρίζονται από άλλα σημεία που έχουν παρόμοια χαρακτηριστικά θα έχουν μικρότερες τοπικές πυκνότητες και άρα μεγαλύτερα LOF scores.

Πολυπλοκότητα: Η χρονική πολυπλοκότητα της μεθόδου είναι $O(n^2)$, όπου n το πλήθος των δεδομένων. Επιπλέον, το γεγονός ότι χρειάζονται δοκιμές για τον προσδιορισμό του κατάλληλου αριθμού γειτόνων k σημαίνει ότι η μέθοδος είναι ιδιαίτερα κοστοβόρα για μεγάλο πλήθος δεδομένων.

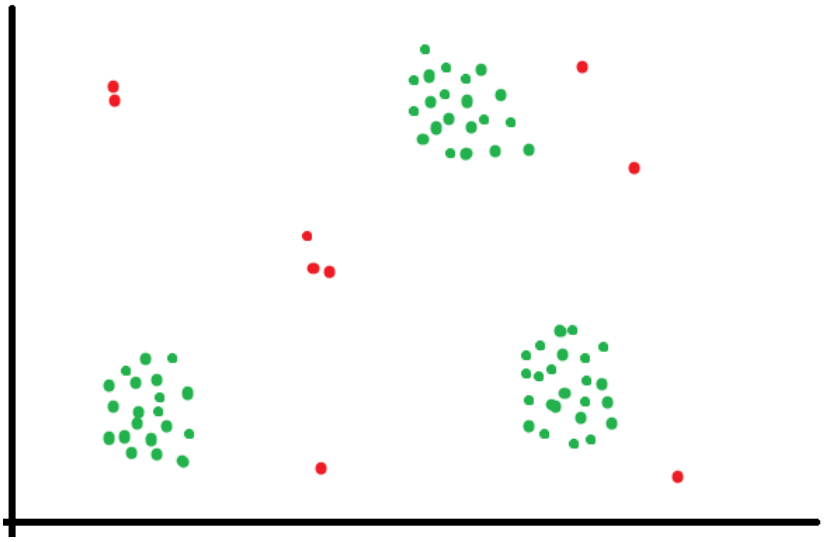
Σύνοψη: Η μέθοδος LOF αποτελεί προφανέστατα μέθοδος εντοπισμού τοπικών ανωμαλιών καθώς βασίζεται στις πυκνότητες των σημείων με τους γείτονές του . Η γνωστή αδυναμία που υπάρχει εξ αιτίας της μεταβλητής k παραμένει και προκύπτουν και νέες αδυναμίες που αντιμετωπίζονται παρακάτω.

Conectivity – Based Outlier Factor (COF)

Η μέθοδος εντοπισμού COF αποτελεί διαφοροποίηση της προηγούμενης (LOF). Στην μέθοδο LOF προκειμένου να υπολογιστεί η απόσταση ενός σημείου από τους γείτονές του χρησιμοποιήθηκε η Ευκλείδεια απόσταση. Το γεγονός αυτό υπονοεί και υποθέτει, ότι τα δεδομένα ακολουθούν σφαιρική κατανομή. Όταν όντως υπάρχει η σφαιρική κατανομή δεν υπάρχει πρόβλημα, αλλά εάν αυτό δεν ισχύει (π.χ. τα δεδομένα έχουν κάποια γραμμική εξάρτηση), τότε η Ευκλείδεια απόσταση δεν είναι η κατάλληλη για τους υπολογισμούς και η LOF αποτυγχάνει να βρει σωστά αποτελέσματα.

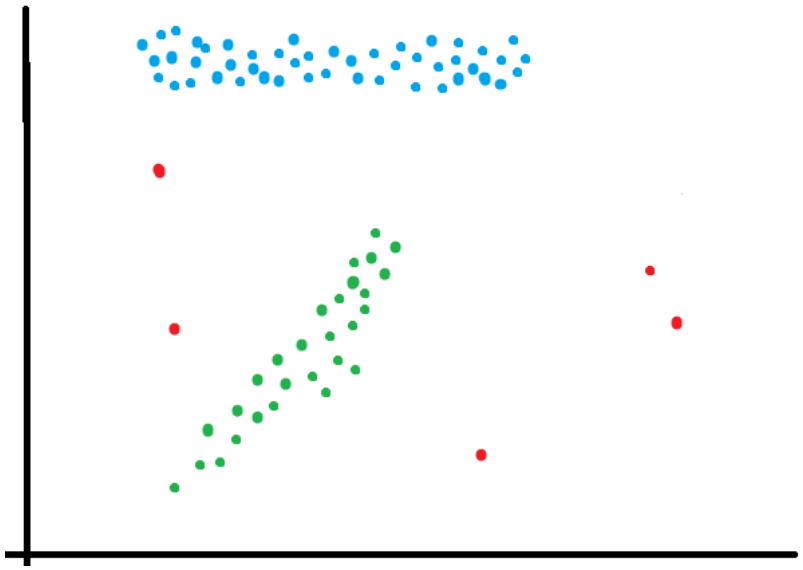
Παρατηρώντας το διάγραμμα:

1. Περίπτωση που η LOF μπορεί να βρει σωστά αποτελέσματα:



(Περίπτωση που τα δεδομένα ακολουθούν σφαιρική κατανομή)

2. Περίπτωση που η LOF δεν μπορεί να βρει σωστά αποτελέσματα, αλλά η COF μπορεί:



(Περίπτωση που τα δεδομένα ακολουθούν γραμμική συσχέτιση)

Η μέθοδος COF αντιμετωπίζει την αδυναμία της LOF χρησιμοποιώντας έναν εναλλακτικό τρόπο υπολογισμού της απόστασης. Πιο συγκεκριμένα, χρησιμοποιεί μία εναλλακτική απόσταση που ονομάζεται αλυσιδωτή (chaining distance) και είναι βασισμένη στη φιλοσοφία των ελάχιστων μονοπατιών. Για τον υπολογισμό της απόστασης αυτής δεν χρησιμοποιείται μόνο η άμεση απόσταση σημείου και πλησιέστερου γείτονα, αλλά λαμβάνεται υπόψιν και η διασύνδεση των γειτόνων μεταξύ τους. Έτσι ως απόσταση ενός σημείου από ένα γείτονα ορίζεται η ελάχιστη τιμή ενός συνόλου που περιέχει το άθροισμα όλων των αποστάσεων που ενώνει τον γείτονα με τους υπόλοιπους γείτονες και το σημείο.

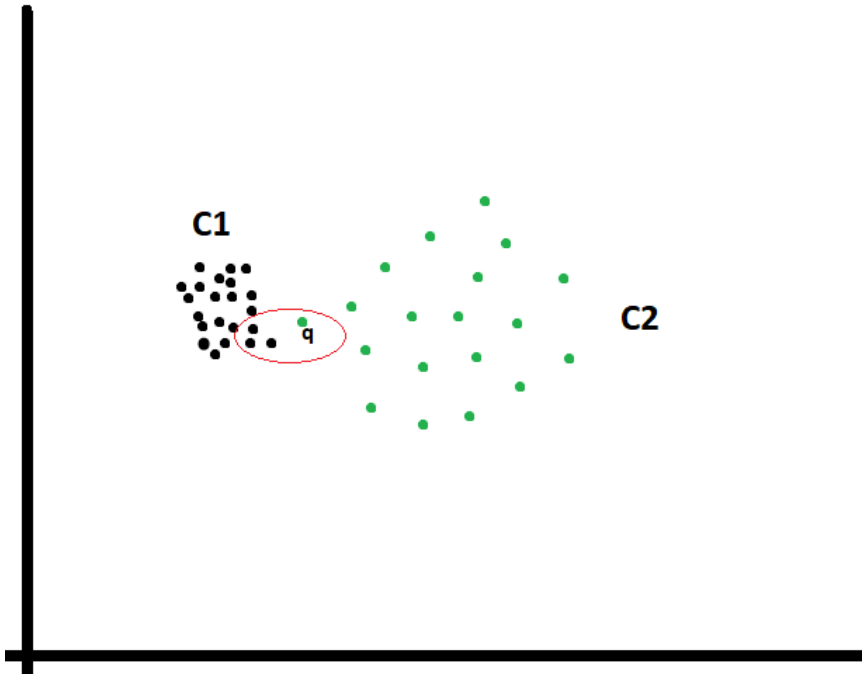
Ο εναλλακτικός αυτός τρόπος υπολογισμού της απόστασης δίνει την δυνατότητα αποδοτικού εντοπισμού ανωμαλιών ακόμα και αν τα δεδομένα έχουν πιο ασυνήθιστες κατανομές.

Πολυπλοκότητα: Όπως και η προηγούμενη μέθοδος έτσι και η μέθοδος COF έχει χρονική πολυπλοκότητα ίση με $O(n^2)$. Οι δύο μέθοδοι μοιάζουν μεταξύ τους ως προς την προσέγγιση και η κύρια διαφορά τους είναι διαφορετική απόσταση που χρησιμοποιούν στους υπολογισμούς.

Influenced Outlierness (INFLO)

(Για την μέθοδο αυτή χρησιμοποιήθηκαν πληροφορίες από το 4)

Η μέθοδος INFLO είναι βασισμένη στην μέθοδο LOF και αναπτύχθηκε προκειμένου να αντιμετωπίσει μία ειδική κατηγορία εντοπισμού ανωμαλιών. Πιο συγκεκριμένα βρίσκει εφαρμογή στον εντοπισμό ανωμαλιών σε σύνολα δεδομένων που παρατηρούνται συστάδες διαφορετικών πυκνοτήτων που βρίσκονται κοντά μεταξύ τους.



(Διάγραμμα συστάδων διαφορετικών πυκνοτήτων σε κοντινή απόσταση)

Στην παραπάνω εικόνα παρατηρώντας το σημείο q βλέπουμε, ότι οι κοντινότεροί του γείτονες (με έστω k = 3) ανήκουν στην συστάδα c1 ενώ το q ανήκει στην συστάδα c2. Τότε το σημείο αυτό θα κριθεί ως outlier με βάση την πυκνότητα της c1, καθώς οι γείτονές του ανήκουν σ' αυτή ενώ στην πραγματικότητα αυτό δεν είναι outlier, αλλά μέλος της c2.

Συνεπώς, η μέθοδος INFLO προτείνει μία επέκταση της LOF με σκοπό την καλύτερη αποτύπωση της τοπικής πυκνότητας των σημείων.

Αυτό γίνεται με την εισαγωγή και χρήση ενός ακόμη συνόλου σημείων πέρα των Nearest Neighbors και συγκεκριμένα το Reverse Nearest Neighbors, δηλαδή των σημείων που έχουν γείτονα το εξεταζόμενο σημείο.

Πιο ξεκάθαρα το σύνολο περιγράφεται απ' το παρακάτω:

$$RNN_k(q) = \{ p \mid p \in Z , p \in NN_k(q) \}$$

Έτσι, απ' το νέο αυτό σύνολο που ορίστηκε παραπάνω και μέσω ένωσης με το σύνολο των πλησιέστερων γειτόνων προκύπτει ένα νέο εκτεταμένο σύνολο με βάση το οποίο γίνονται οι υπολογισμοί του Local Reachability Density και το INFLO score. Οι υπολογισμοί είναι ακριβώς ίδιοι με την μέθοδο LOF σε μεθοδολογία και επίσης η ερμηνεία των αποτελεσμάτων παραμένει ίδια.

Πολυπλοκότητα: Και η μέθοδος INFLO έχει χρονική πολυπλοκότητα $O(n^2)$, καθώς το κυριότερο τμήμα της μεθόδου μοιάζει αρκετά με την LOF. Η ουσιώδης διαφορά είναι η χρήση των επιπλέων σημείων στον υπολογισμό της βαθμολογίας.

<u>Μέθοδοι που χρησιμοποιούν θεωρία πιθανοτήτων</u>

Παρακάτω θα παρουσιαστούν μέθοδοι, οι οποίες χρησιμοποιούν στοιχεία απ' την θεωρία πιθανοτήτων προκειμένου να εντοπίσουν ανώμαλες τιμές. Αυτές αναπτύχθηκαν, καθώς δεν είναι πάντα εύκολος ο καθορισμός της σωστής τιμής κατωφλίου που θεωρούμε ως σημείο διαχωρισμού κανονικών τιμών και outlier. Ενώ αντίθετα, εκφράζοντας τα αποτελέσματα μεθόδων με όρους πιθανοτήτων σε διαστήματα εμπιστοσύνης δίνει ένα πιο κατανοητό και εύκολο τρόπο ερμηνείας των αποτελεσμάτων.

<u>Local Outlier – Propability (loOP)</u>

Όπως προϊδεάζει το όνομα, πρόκειται για μία μέθοδο εντοπισμού τοπικών ανωμαλιών, η οποία χρησιμοποιεί (όπως και οι προηγούμενες μέθοδοι) ένα σύνολο σημείων – γειτονίας, για να υπολογίσει την τοπική πυκνότητα των σημείων. Μόνο που για να υπολογιστεί η πυκνότητα χρησιμοποιείται ένας διαφορετικός τρόπος.

Πιο συγκεκριμένα, το γενικό σκεπτικό είναι πως τα σημεία που ανήκουν στην γειτονιά ενός εξεταζόμενου σημείου θα ακολουθούν κανονική (ή αλλιώς Γκαουσιανή) κατανομή όσων αφορά την απόστασή τους. Η κατανομή που χρησιμοποιείται δεν είναι ακριβώς κανονική αλλά μία λίγο διαφοροποιημένη με ονομασία “Half Gaussian Distribution” στην οποία οι τιμές είναι μεγαλύτερες ή ίσες του μηδενός (η απόσταση δεν μπορεί να είναι αρνητική).

Στην συνέχεια, υπολογίζεται ο λόγος της απόκλισης της απόστασης ενός σημείου προς την τυπική απόκλιση της κατανομής και έπειτα μέσω κανονικοποίησης των λόγων αυτών παράγονται τα τελικά αποτελέσματα εκφρασμένα με πιθανότητες για το αν μία εγγραφή είναι κανονική ή όχι.

4.2 Clustering based methods

Οι μέθοδοι που θα παρουσιαστούν παρακάτω βασίζονται σε αλγορίθμους συσταδοποίησης προκειμένου να προσδιορίσουν τα ανώμαλα σημεία ενός συνόλου δεδομένων. Γενικά, μπορεί να χρησιμοποιηθεί οποιοσδήποτε εκ των γνωστών αλγορίθμων συσταδοποίησης, αλλά συνήθως χρησιμοποιείται ο k – means ή κάποια παραλλαγή του, καθώς λόγω της χαμηλής του πολυπλοκότητας μπορεί να πετύχει καλούς χρόνους εκτέλεσης σε πραγματικές εφαρμογές.

Παρατήρηση: Όπως και στην προηγούμενη κατηγορία μεθόδων (Nearest Neighbor based methods) , έτσι κι εδώ υπάρχει αδυναμία στον ορισμό της παραμέτρου k (πλήθος κέντρων του k – means). Συνεπώς, και στην κατηγορία αυτή απαραίτητη είναι η δοκιμή και η επαλήθευση των αποτελεσμάτων των μεθόδων.

Global anomaly methods

Cluster based local outlier factor (CBLOF)

Η μέθοδος ξεκινάει με εφαρμογή του k – means στο εξεταζόμενο σύνολο δεδομένων απ’ την οποία προκύπτουν k ομάδες. Στην συνέχεια, για κάθε εγγραφή του συνόλου δίνεται βαθμολογία που ισούται με την απόστασή του από το κέντρο της ομάδας στην οποία ανήκει πολλαπλασιασμένη με το πλήθος των εγγραφών που ανήκουν στην

συγκεκριμένη συστάδα. Στην συνέχεια, τίθεται όπως και στις προηγούμενες μεθόδους μία τιμή κατωφλίου, που όταν ξεπεραστεί θεωρούμε ότι υπάρχει outlier.

Σημείωση: Εάν δημιουργηθεί κάποια συστάδα της οποίας το μέγεθος είναι εξαιρετικά μικρό (micro – cluster), τότε χρησιμοποιούμε ως σημείο αναφοράς για τους υπολογισμούς των αποστάσεων το κέντρο της κοντινότερης μεγάλης συστάδας που προκύπτει απ' τον k – means.

Πολυπλοκότητα: Η πολυπλοκότητα στην συγκεκριμένη κατηγορία σχετίζεται άμεσα με τον αλγόριθμο που επιλέγεται για την συσταδοποίηση. Όταν χρησιμοποιείται ο k – means η μέθοδος CBLOF επιτυγχάνει πολυπλοκότητα $O(n^2)$ και ίσως και λίγο χαμηλότερη ανάλογα με τις συνθήκες του εκάστοτε συνόλου.

Unweighted cluster based local outlier factor (uCBLOF)

Η συγκεκριμένη μέθοδος είναι παρόμοια με την παραπάνω με μία σημαντική διαφορά. Ως βαθμολογία ορίζεται η απόσταση που έχει το κάθε σημείο από το κέντρο της ομάδας στην οποία υπάγεται. Στην ουσία, δεν λαμβάνεται υπόψιν το πλήθος των σημείων, που περιέχει η κάθε ομάδα στον υπολογισμό της βαθμολογίας. Εξού και το "unweighted" στην ονομασία της μεθόδου (δεν χρησιμοποιείται το πλήθος των στοιχείων ως βάρος).

Σημείωση: Ισχύει η ίδια σύμβαση για τις συστάδες μικρού μεγέθους.

Local anomaly methods

Local Density Cluster based Outlier Factor (LDCOF)

Η συγκεκριμένη μέθοδος όπως και οι προηγούμενες ξεκινάει κάνοντας συσταδοποίηση με χρήση του k – means και στην συνέχεια κατηγοριοποιεί τις συστάδες σε μεγάλες και μικρές (για τον ίδιο λόγο με την σημείωση της μεθόδου CBLOF).

Έπειτα, για κάθε ομάδα υπολογίζεται η απόσταση όλων των σημείων από το εκάστοτε κέντρο, καθώς και η μέση απόσταση των σημείων απ' το κέντρο της ομάδας στην οποία ανήκουν. Στην συνέχεια, η βαθμολογία κάθε σημείου υπολογίζεται διαιρώντας της απόστασή του απ' το κέντρο της ομάδας δια την μέση απόσταση.

Απ' τον τρόπο υπολογισμού της βαθμολογίας είναι αντιληπτό, ότι η μέθοδος αυτή χρησιμοποιείται για να εντοπίζει τοπικά outliers, γιατί στον υπολογισμό της βαθμολογίας λαμβάνεται υπόψιν η απόσταση απ' το κέντρο των συστάδων και κατά συνέπεια η τοπική πυκνότητα των ομάδων.

Ερμηνεία: Τα αποτελέσματα της μεθόδου και η ερμηνεία τους ταυτίζεται με αυτά της μεθόδου LOF, που αναλύθηκε προηγουμένως. Δηλαδή, χαμηλές βαθμολογίες που βρίσκονται κοντά στο 1 σημαίνει, ότι μία εγγραφή είναι φυσιολογική (η απόστασή της απ' το κέντρο δεν έχει μεγάλη απόκλιση απ' τον μέσο όρο των αποστάσεων), ενώ οι μεγάλες τιμές σηματοδοτούν ανώμαλες τιμές με αποστάσεις που αποκλίνουν απ' τον μέσο όρο.

Πολυπλοκότητα: Ισχύουν όσα ίσχυαν και στις προηγούμενες μεθόδους βασισμένες στην συσταδοποίηση. Δηλαδή, $O(n^2)$ όταν χρησιμοποιείται ο k – means.

4.3 Statistical based methods

Στην κατηγορία αυτή υπάρχουν μέθοδοι που χρησιμοποιούν καθαρά στατιστική προσέγγιση για τον προσδιορισμό των outlier. Η διαφορά με προηγούμενες μεθόδους που περιείχαν στοιχεία στατιστικής (π.χ. LoOP) είναι ότι αυτές βασίζονταν σε άλλους αλγορίθμους και απλά εισήγαγαν βοηθητικές έννοιες σε επίπεδο υπολογισμών και αποτελεσμάτων.

Histogram – Based Outlier Score (HBOS)

(Για την μέθοδο αυτή χρησιμοποιήθηκαν πληροφορίες από 5)

Η HBOS αποτελεί μέθοδος εντοπισμού outliers που βασίζεται στην στατιστική. Το σκεπτικό της ταυτίζεται αρκετά με αυτό του Naïve Bayes, όπου για διάφορα γνωρίσματα / χαρακτηριστικά ενός συνόλου δεδομένων αγνοούνται οι συσχετίσεις που μπορεί να έχουν μεταξύ προκειμένου μέσω πολλαπλασιασμού να καθοριστεί η πιθανότητα εμφάνισης ενός γεγονότος.

Στην μέθοδο HBOS για κάθε ένα γνώρισμα των δεδομένων δημιουργείται ένα ιστόγραμμα και μέσω μελέτης των δεδομένων κάθε εγγραφή τοποθετείται στον «κάδο» που ανήκει (ξεχωριστά για κάθε ιδιότητα). Έπειτα για κάθε εγγραφή υπολογίζουμε το γινόμενο των αντίστροφων υψών του κάδου που ανήκει σε κάθε ιστόγραμμα κι έτσι προκύπτει μία βαθμολογία.

Ερμηνεία αποτελεσμάτων: Όταν μία εγγραφή ανήκει σε κάποιον «κάδο» που είναι πολυπληθής (δηλαδή ένα σχετικά μεγάλο ποσοστό των εγγραφών ανήκει στον ίδιο «κάδο») θεωρητικά αποτελεί κανονικό σημείο. Όταν, λοιπόν πρόκειται να γίνει ο υπολογισμός του γινομένου που αναφέρθηκε παραπάνω θα παρατηρηθεί το εξής:

Καθώς οι πολυπληθείς «κάδοι» έχουν μεγάλο ύψος, τότε το αντίστροφό τους θα είναι ένα μικρό κλάσμα (μικρότερο της μονάδας). Πολλαπλασιάζοντας πολλά τέτοια κλάσματα (ίσα με το πλήθος των διαφορετικών ιστογραμμάτων) το γινόμενο θα είναι ακόμη μικρότερο. Αντίθετα, οι «κάδοι» που περιέχουν ελάχιστες εγγραφές και δεν έχουν μικρό ύψος θα έχουν ως αντίστροφο ένα κλάσμα που εξακολουθεί να είναι μικρό (κάτω της μονάδας), αλλά αρκετά μεγαλύτερο απ' τους πολυπληθείς. Έτσι οι βαθμολογίες που θα αποδοθούν στις ανώμαλες εγγραφές θα είναι σημαντικά μεγαλύτερες από αυτές των κανονικών.

Προσεγγίσεις τις μεθόδου: Η μέθοδος αυτή μπορεί να εφαρμοστεί με 2 κύριες μορφές.

Μορφή 1η : Οι κάδοι έχουν στατικό πλάτος (διάστημα το οποίο καλύπτουν για ένα χαρακτηριστικό), το οποίο ορίζεται στην αρχή. Είναι πιο γρήγορη απ' την μορφή 2, αλλά μπορεί ο αρχικός ορισμός του πλάτους να μην είναι ο βέλτιστος.

Μορφή 2η : Οι κάδοι έχουν δυναμικό πλάτος που μπορεί να αλλάξει κατά την εκτέλεση . Είναι πιο αργή απ' την μορφή 1, αλλά μπορεί να μεταβάλει το πλάτος κατά την εκτέλεση εάν παρατηρηθεί, ότι ο αρχικός ορισμός δεν ήταν σωστός για το σύνολο δεδομένων (π.χ. η ανάθεση σημείων στους κάδους είναι πολλή αραιή ή όλα τα σημεία συγκεντρώνονται σε ένα κάδο).

Πολυπλοκότητα:

- Η **μορφή 1** επιτυγχάνει χρονική πολυπλοκότητα O(n).
- Η **μορφή 2** επιτυγχάνει χρονική πολυπλοκότητα O(n*logn).

Πλεονεκτήματα: Η χαμηλή πολυπλοκότητα της μεθόδου συγκριτικά με τις προηγούμενες των άλλων κατηγοριών.

Μειονεκτήματα: Το γεγονός ότι μπορεί να χάνεται πληροφορία, επειδή αγνοούνται οι συσχετίσεις μεταξύ των γνωρισμάτων (όπως και στο Naïve Bayes).

4.4 Subspace based methods

Robust Principal Component Analysis (rPCA)

(Για την μέθοδο αυτή χρησιμοποιήθηκαν πληροφορίες από το 3)

Η παραπάνω μέθοδος αποτελεί μία επέκταση του αλγορίθμου PCA. Ο αλγόριθμος PCA από μόνος του αποτελεί σημαντικό εργαλείο στην ανάλυση και μελέτη δεδομένων, καθώς μέσω της εφαρμογής του ο αναλυτής μπορεί να πετύχει τρεις πολύ βασικούς στόχους:

1. Μείωση διαστάσεων (αντιμετωπίζεται η «κατάρα της διαστασιμότητας»).
2. Γραφική απεικόνιση δεδομένων, καθώς μπορεί να μειώσει τις διαστάσεις σε αριθμό ικανό για απεικόνιση.
3. Απαλοιφή χαρακτηριστικών (π.χ. κάποια χαρακτηριστικά μπορεί να εμπεριέχονται σε άλλα και να είναι περιττά).

Η επέκταση της PCA, rPCA, βρίσκει εφαρμογές σε ποικίλα θέματα μηχανικής μάθησης (π.χ. αναγνώριση προσώπων / face recognition) και ανάλυσης δεδομένων (outlier detection). Ο τρόπος με τον οποίο τα πετυχαίνει αυτά είναι ο εξής:

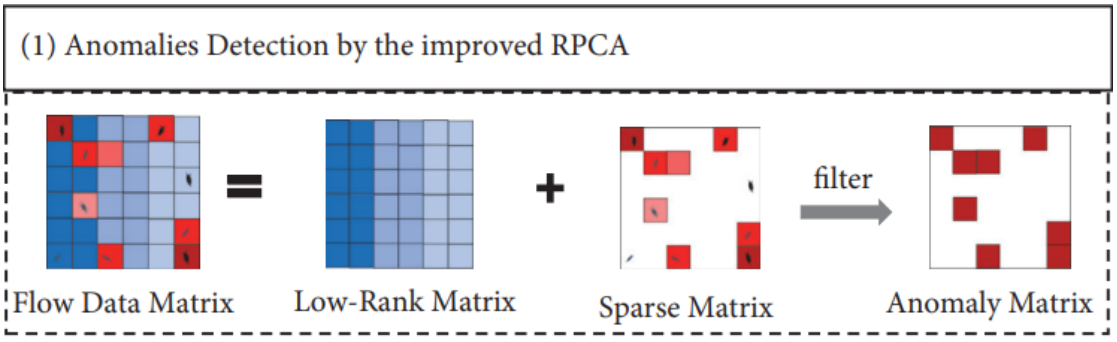
Αρχικά, τα δεδομένα αναπαρίστανται σε μορφή πίνακα. Χρησιμοποιώντας σύνολα εκπαίδευσης έχει την δυνατότητα να διαχωρίσει τον αρχικό πίνακα σε άθροισμα 2 πινάκων.

Πίνακας 1 : Ένας Low – Rank πίνακας που περιέχει τα φυσιολογικά δεδομένα

(Low – Rank πίνακας: Ένας πίνακας για τον οποίο ισχύει ότι οι γραμμικά ανεξάρτητες στήλες είναι πολλή λιγότερες από τον συνολικό αριθμό των στηλών)

Πίνακας 2 : Ένας Sparse πίνακας που περιέχει τα δεδομένα που δεν φαίνεται να ταιριάζουν με τα υπόλοιπα και πιθανότητα είναι outliers.

(Sparse πίνακας: Ένας πίνακας που έχει τα δεδομένα του πολύ αραιά μεταξύ τους και τα περισσότερα στοιχεία του είναι μηδέν).



(Εικόνα απ' το άρθρο 3)

Πολυπλοκότητα: Η μέθοδος rPCA εξαρτάται σημαντικά από τον αρχικό αριθμό των διαστάσεων. Η πολυπλοκότητά της εκφράζεται ως $O(d^2*n + d^3)$. Όταν, λοιπόν, ο αριθμός των διαστάσεων είναι πολύ μεγάλος η εφαρμογή της κρίνεται αρκετά κοστοβόρα, αλλά αντίθετα όταν βρίσκεται σε λογικά πλαίσια αποτελεί μία από τις γρηγορότερες μεθόδους που είναι διαθέσιμες για την αναγνώριση ανωμαλιών.

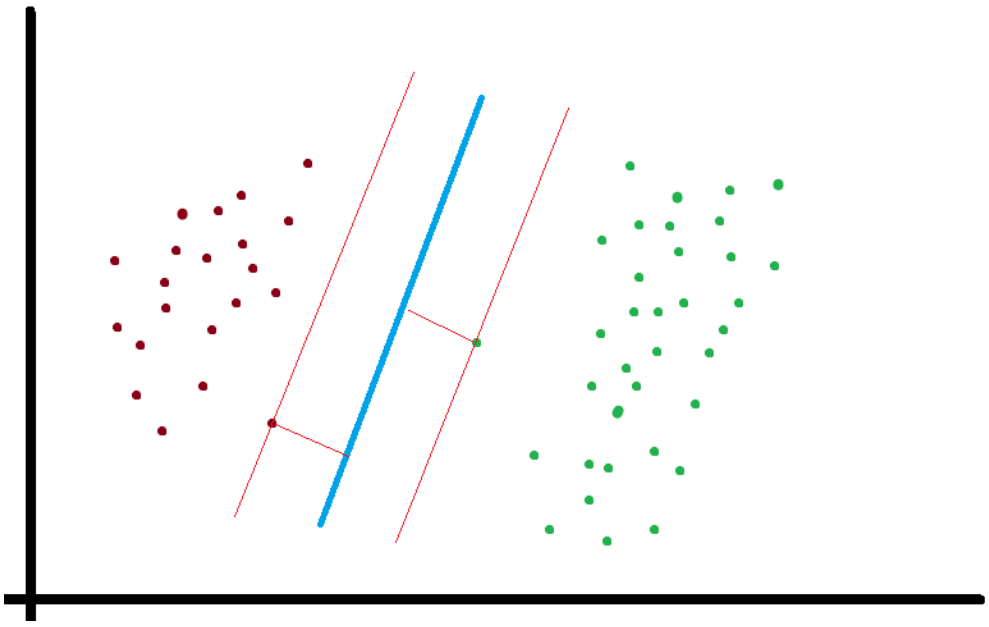
4.5 Classifier Based methods

(Για την μέθοδο αυτή χρησιμοποιήθηκαν πληροφορίες από το 6)

Η κατηγορία που θα αναλυθεί στην συνέχεια αφορά σε μεθόδους που προκειμένου να αποφασίσουν ποιες εγγραφές αποτελούν outliers και ποιες είναι κανονικές χρησιμοποιούν αλγορίθμους κατηγοριοποίησης.

One Class Support Vector Machines (One Class SVM)

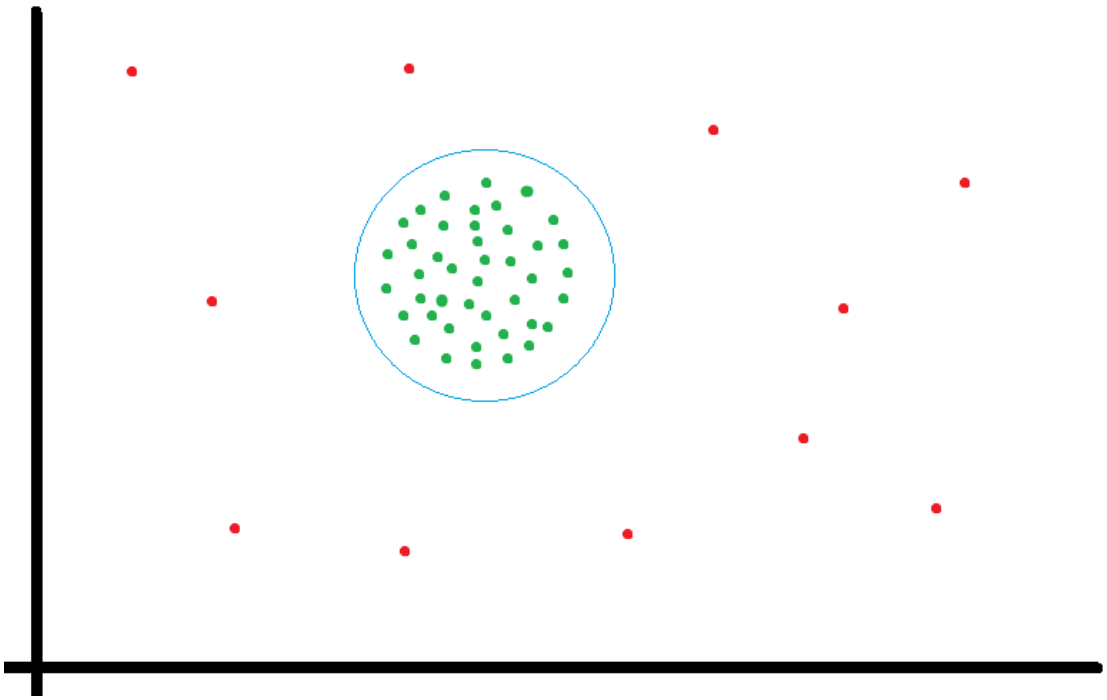
Για να γίνει κατανοητός ο τρόπος λειτουργίας της μεθόδου σκόπιμο είναι πρωτίστως να αποδοθεί η περιγραφή των απλών SVMs. Πιο συγκεκριμένα τα SVMs αποτελούν μέθοδος κατηγοριοποίησης δεδομένων σε κλάσεις. Στην απλή τους μορφή αποτελεί μέθοδος ημί – επιβλεπόμενης μάθησης, καθώς ένα μοντέλο εκπαιδεύεται με ένα σύνολο εκμάθησης που περιέχει μόνο αντικείμενα μίας κλάσης. Στην συνέχεια, εξετάζοντας το σύνολο δεδομένων που ενδιαφέρει τους αναλυτές καλείται να διαχωρίσει τα δεδομένα σε 2 κλάσεις βρίσκοντας ένα διάνυσμα υποστήριξης (support vector) που χωρίζει τις δύο κλάσεις. Ζητούμενο είναι επίσης το διάνυσμα αυτό να μεγιστοποιεί την απόσταση που έχουν τα κοντινότερα (προς αυτό) σημεία της κάθε κλάσης. Ο τρόπος λειτουργίας είναι πιο κατανοητός στο παρακάτω διάγραμμα:



Επειδή στην κλασική μορφή η κατηγοριοποίηση γίνεται μεταξύ δύο κλάσεων, το διαχωριστικό (το διάνυσμα υποστήριξης) έχει την μορφή ευθείας.

Όσον αφορά τώρα τα One Class SVM, αυτά όπως προϊδεάζει η ονομασία τους καλούνται να κατηγοριοποιήσουν τα σημεία σε μία κλάση και να ελέγξουν αν άλλα σημεία ταιριάζουν στην κλάση αυτή. Η διαδικασία αυτή δεν είναι ακριβώς outlier detection, αλλά novelty detection. Η διαφορά των δύο διαδικασιών είναι ότι στο outlier detection γνωρίζουμε ότι το σύνολο δεδομένων περιέχει ανώμαλα σημεία, ενώ αντίθετα στο novelty detection αυτό δεν είναι γνωστό και ψάχνουμε αν οι εγγραφές ταιριάζουν σε μία κλάση. Επίσης, εφόσον στο εξεταζόμενο σύνολο δεδομένων δεν γνωρίζουμε εάν υπάρχουν outliers δεν μπορούμε να έχουμε ημί – επιβλεπόμενη μάθηση και το One Class SVM αποτελεί περίπτωση μάθησης χωρίς επίβλεψη. Η διαδικασία γίνεται πιο κατανοητή στο παρακάτω διάγραμμα:

Διάγραμμα one – class SVM



(Πράσινα σημεία -> Κλάση κατηγοριοποίησης, Κόκκινα σημεία-> Ανωμαλίες)

Επειδή στα One Class SVM ο κατηγοριοποιητής προσπαθεί να περιορίσει μία μόνο κλάση απ' τα υπόλοιπα στοιχεία η μορφή του είναι κυκλική/σφαιρική (ανάλογα με τις διαστάσεις του προβλήματος), όπως φαίνεται στο παραπάνω σχήμα (μπλε κύκλος). Με την χρήση τους μπορεί να γίνει διαχωρισμός της κλάσης των φυσιολογικών εγγραφών και όσες μένουν εκτός της κλάσης θεωρούνται outliers.

Πολυπλοκότητα: Η πολυπλοκότητα των SVM είναι πολύ δύσκολο να προσδιοριστεί ,διότι εξαρτώνται κατά κόρον απ' τον αριθμό των διαστάσεων, τον αριθμό των κλάσεων που πρέπει να διαχωριστούν και κατά συνέπεια τον αριθμό των διανυσμάτων υποστήριξης που πρέπει να βρεθούν.

4.6: Σύνοψη και παρατηρήσεις

Πλέον, αφού έχουν παραστεί και αναλυθεί οι πτυχές του εντοπισμού ανωμαλιών σε στατικά δεδομένα, κρίσιμο είναι να γίνει αναφορά στις δυνατότητές τους και τις αδυναμίες τους.

Για να γίνει αυτό, θα κριθούν με βάση τα εξής χαρακτηριστικά:

- Την ακρίβεια
- Τον ντετερμινισμό
- Την ευαισθησία
- Την ταχύτητα
- Την δυνατότητα εύρεσης καθολικών ανωμαλιών

k-NN: Η μέθοδος k-NN έχει πολύ καλά επίπεδα ακρίβειας, ντετερμινισμού. Η ευαισθησία του επηρεάζεται απ' τον καθορισμό του πλήθους γειτόνων , ενώ έχει μέτρια ταχύτητα. Επίσης μπορεί να εντοπίσει καθολικές ανωμαλίες.

LOF: Η μέθοδος LOF έχει παρόμοιες επιδόσεις με την k-NN.

COF: Η μέθοδος COF έχει μέτρια ακρίβεια ενώ στα υπόλοιπα γνωρίσματα ταυτίζεται με την LOF. Αρνητικό στοιχείο, ότι δεν εντοπίζει καθολικές ανωμαλίες.

INFLO: Η μέθοδος αυτή έχει μέτρια ακρίβεια και στα υπόλοιπα επίσης ταυτίζεται με την LOF. Αρνητικό στοιχείο, ότι δεν εντοπίζει καθολικές ανωμαλίες.

LoOP: Η μέθοδος αυτή έχει πολύ καλά επίπεδα ακρίβειας, ντετερμινισμού, μικρή ευαισθησία και μέτρια ταχύτητα. Αρνητικό στοιχείο, ότι δεν εντοπίζει καθολικές ανωμαλίες.

CBLOF: Η μέθοδος αυτή έχει γενικά πολύ κακά επίπεδα ακρίβειας, μέτρια επίπεδα ντετερμινισμού και ευαισθησίας, αλλά και μέτρια επίπεδα ταχύτητας. Αρνητικό στοιχείο, ότι δεν εντοπίζει καθολικές ανωμαλίες.

uCBLOF: Η μέθοδος αυτή έχει πολύ καλά επίπεδα ακρίβειας, μέτρια επίπεδα ντετερμινισμού και ευαισθησίας, καλή ταχύτητα. Θετικό στοιχείο η δυνατότητα εύρεσης καθολικών ανωμαλιών.

LDCOF: Η μέθοδος αυτή έχει κακά επίπεδα ακρίβειας, μέτρια επίπεδα ντετερμινισμού και ευαισθησίας, καλή ταχύτητα, ενώ μπορεί να χρησιμοποιηθεί και για εντοπισμό καθολικών ανωμαλιών με καλές επιδόσεις.

HBOS: Η HBOS είναι από τις μεθόδους με τις καλύτερες επιδόσεις, καθώς έχει για όλα τα χαρακτηριστικά καλές επιδόσεις με εξαίρεση την ευαισθησία που κυμαίνεται σε μέτρια επίπεδα.

rPCA: Η μέθοδος rPCA γενικά έχει καλές επιδόσεις για τα χαρακτηριστικά του ντετερμινισμού, της ευαισθησίας και της ταχύτητας. Στα υπόλοιπα έχει μέτριες επιδόσεις.

One class -SVM: Η μέθοδος αυτή έχει μέτρια ακρίβεια και δυνατότητα εντοπισμού καθολικών outliers, καλά επίπεδα ντετερμινισμού και ευαισθησίας. Το μεγάλο της μειονέκτημα είναι η εξαιρετικά χαμηλή της ταχύτητα.

5. Μέθοδοι εντοπισμού ανωμαλιών σε χρονοσειρές δεδομένων

(Για την ενότητα 5 χρησιμοποιήθηκαν πληροφορίες από τα 2, 7)

Ολοκληρώνοντας την μελέτη μεθόδων που αφορούν σε στατικά δεδομένα, το ενδιαφέρον μεταβαίνει στην διαχείριση δεδομένων που εισάγουν και την έννοια του χρόνου στην μελέτη. Τέτοιου είδους δεδομένα είναι και αυτά, τα οποία συνήθως καλούνται να διαχειριστούν σύγχρονα συστήματα, όπως το Apache Pinot που θα αναλυθεί παρακάτω. Αυτό συμβαίνει, καθώς στις περισσότερες περιπτώσεις δεδομένα που περιέχουν και την έννοια του χρόνου φέρουν περισσότερη πληροφορία και η ανάλυσή τους έχει πιο πολύτιμα αποτελέσματα.

Ο τρόπος διαχείρισης τέτοιων δεδομένων διαφέρει κατά πολύ τόσο ως προς τον τρόπο ορισμού της ανώμαλης τιμής όσο και στον διαχωρισμό που υπόκεινται οι μέθοδοι εντοπισμού. Αναλυτικότερα ο διαχωρισμός των μεθόδων γίνεται με τα παρακάτω κριτήρια:

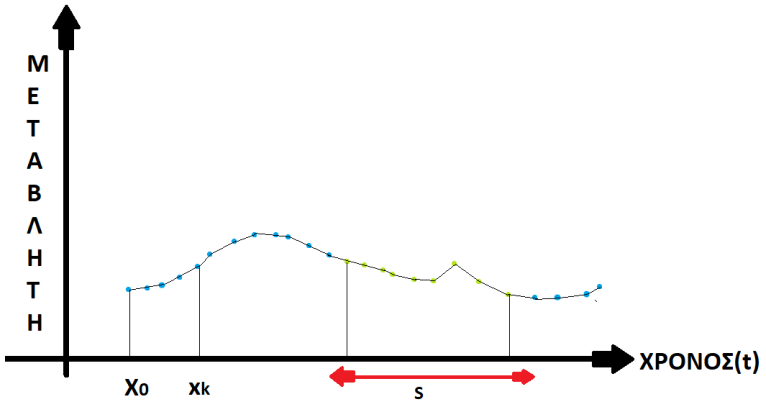
1. Τον τύπο των δεδομένων: Εάν είναι μονής μεταβλητής (univariate) ή πολλαπλών μεταβλητών (multivariate).
2. Τον τύπο της ανώμαλης τιμής: Εάν είναι ένα συγκεκριμένο σημείο, ένα υποσύνολο της χρονοσειράς, ή ολόκληρη χρονοσειρά (κι δω υπάρχει περαιτέρω διαχωρισμός για univariate και multivariate).
3. Η φύση της μεθόδου: Αν μπορεί να εφαρμοστεί σε univariate ή multivariate δεδομένα.

Στην συνέχεια θα παρατεθούν επιπλέον στοιχεία και ορισμοί για καθένα απ’ τα παραπάνω κριτήρια τα φανούν χρήσιμα κατά την ανάλυση.

Τύποι δεδομένων:

Τα δεδομένα, τα οποία μπορεί να χρήζουν ανάλυσης μπορούν να είναι δύο ειδών. Είτε μονής μεταβλητής είτε πολλαπλών μεταβλητών.

Δεδομένα μονής μεταβλητής (univariate data): Μία χρονοσειρά δεδομένων μονής μεταβλητής έστω X, αποτελεί ένα διατεταγμένο σύνολο τιμών όπου κάθε στοιχείο του έχει καταγραφεί σε μία ορισμένη χρονική στιγμή t. Έπειτα, στο σύνολο αυτό μπορούν να οριστούν υποσύνολα, έστω S τα οποία αναπαριστούν τις τιμές της μεταβλητής σε ένα χρονικό διάστημα. Παρακάτω δίνεται και ένα διάγραμμα για να είναι πιο κατανοητός ο ορισμός:



Δεδομένα πολλαπλών μεταβλητών (multivariate data): Μια χρονοσειρά δεδομένων πολλαπλών μεταβλητών, έστω X αποτελεί ένα σύνολο k διατεταγμένων συνόλων (όπου k το πλήθος των μεταβλητών). Το I – οστο σύνολο αντιπροσωπεύει την I – οστή μεταβλητή. Όπως και στην univariate κατηγορία, έτσι και σ' αυτήν τα στοιχεία κάθε συνόλου αναπαριστούν την τιμή κάποιας εκ των μεταβλητών σε συγκεκριμένη χρονική στιγμή.

Τύποι ανωμαλιών:

Όσον αφορά στους τύπους ανωμαλιών που μπορούν να παρατηρηθούν υπάρχουν 3 κατηγορίες:

Ανώμαλο σημείο (Point anomaly): Ως ανώμαλο σημείο λογίζεται η τιμή μίας έγγραφής, η οποία παρεκκλίνει σημαντικά είτε από τους γείτονές της (local anomaly) είτε απ' όλο το σύνολο (global anomaly).

Το ανώμαλο σημείο μπορεί να αφορά είτε σε univariate είτε σε multivariate data. Πιο συγκεκριμένα, στα univariate αρκεί η εγγραφή να αποκλίνει σε σχέση με τιμές της ίδιας της μεταβλητής , ενώ για να έχουμε multivariate point anomaly πρέπει να εμφανίζεται point anomaly σε παραπάνω από μία μεταβλητές την ίδια χρονική στιγμή t.

Ανώμαλο υποσύνολο (Subsequence anomaly): Η κατηγορία αυτή έχει παρόμοια χαρακτηριστικά με την προηγούμενη και η ουσιαστική διαφορά είναι ότι οι αποκλίνουσες εγγραφές εμφανίζονται διαδοχικά σε ένα χρονικό διάστημα.

Ανώμαλη χρονοσειρά (Time series anomaly) : Η κατηγορία αυτή μπορεί να εμφανιστεί μόνο σε περιπτώσεις όπου πραγματεύονται multivariate δεδομένα. Συνήθως, η ανωμαλία που παρατηρείται είναι ότι η χρονοσειρά δεδομένων που αναπαριστά μία συγκεκριμένη μεταβλητή αποκλίνει αρκετά σε σχέση με τις υπόλοιπες μεταβλητές.

Φύση της μεθόδου:

Το κριτήριο της φύσης της μεθόδου έχει να κάνει με το εάν αυτή είναι σχεδιασμένη με τέτοιο τρόπο ώστε να μπορεί να χειριστεί univariate ή multivariate data. Πιο συγκεκριμένα μία μέθοδος, που είναι σχεδιασμένη για δεδομένα μίας μεταβλητής μπορούν να χειριστούν και πολλαπλών μεταβλητών (με διαδοχική μελέτη των επιμέρους μεταβλητών), ενώ αντίθετα οι μέθοδοι που χειρίζονται δεδομένα πολλαπλών μεταβλητών δεν μπορούν να χειριστούν αυτά της μονής μεταβλητής.

5.1 Εντοπισμός point outliers σε χρονοσειρές

5.1.1 Εντοπισμός σε univariate data

Όσον αφορά των εντοπισμό των point outliers υπάρχουν 2 χαρακτηριστικά που διακρίνονται στις μεθόδους και είναι σημαντικό να αναφερθούν.

Το πρώτο εξ αυτών αποτελεί εάν οι μέθοδοι λαμβάνουν υπόψιν την παροδικότητα των μεθόδων, δηλαδή την σειρά με την οποία ελήφθησαν οι εγγραφές. Οι μέθοδοι που δεν λαμβάνουν την παροδικότητα υπόψιν θα δώσουν ίδια αποτελέσματα με όποια σειρά και εάν επεξεργαστούν τα δεδομένα, σε αντίθεση με τις άλλες.

Το δεύτερο έχει να κάνει με το κατά πόσο μία μέθοδος έχει την ικανότητα να εφαρμοστεί σε δεδομένα που τροφοδοτούνται σε κανονικό χρόνο και να προσδιορίσει εάν μία εισερχόμενη εγγραφή είναι ανωμαλία ή όχι. Έτσι οι μέθοδοι με βάση το χαρακτηριστικό αυτό χωρίζονται σε streaming , non – streaming .

Κατηγορίες μεθόδων εντοπισμού point outliers σε univariate data

Κατηγορία 1η: Model Based.

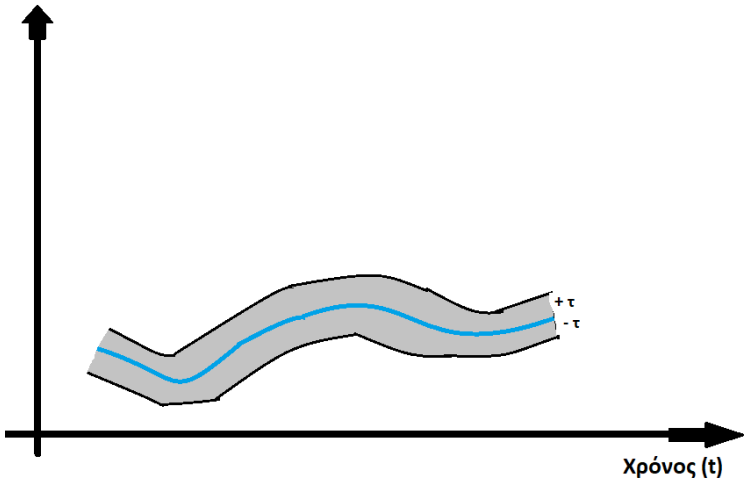
Η συγκεκριμένη κατηγορία αφορά σε μεθόδους που βασίζονται σε μοντέλα για την εύρεση και την κατηγοριοποίηση εγγραφών ως outliers. Οι δύο βασικές προσεγγίσεις που υπάρχουν για τα μοντέλα αυτά είναι τα μοντέλα πρόβλεψης δεδομένων (prediction models) και τα μοντέλα εκτίμησης δεδομένων (estimation models). Τα μοντέλα που λειτουργούν με βάση την πρόβλεψη δεδομένων χρησιμοποιούν δεδομένα του παρελθόντος προκειμένου να χαρακτηρίσουν τις εγγραφές, γεγονός το οποίο σημαίνει ότι μπορούν να χρησιμοποιηθούν σε streaming data. Αντιθέτως, τα μοντέλα που βασίζονται στην εκτίμηση χρησιμοποιούν παρελθοντικά, τωρινά καθώς και μελλοντικά δεδομένα για την απόδοση χαρακτηρισμών και άρα δεν μπορούν να χρησιμοποιηθούν για real – time εισερχόμενα δεδομένα.

Γενικά, οι μέθοδοι που ανήκουν στην κατηγορία αυτή προσανατολίζονται στο να προσδιορίσουν ποια είναι μία πιθανή τιμή για τις επόμενες εγγραφές με βάση υπάρχοντα δεδομένα και στην συνέχεια κρίνουν τις εισερχόμενες εγγραφές μελετώντας την απόκλιση της πιθανής τιμής που προσδιορίστηκε απ’ την πραγματική τιμή που λήφθηκε. Στην συνέχεια, εφόσον η απόκλιση αυτή βρίσκεται εντός ορισμένων ορίων κάποιας τιμής κατωφλίου ‘τ’ λογίζονται ως κανονικά δεδομένα, ενώ εάν η απόκλιση ξεπερνάει τα όρια, τότε λογίζονται ως outliers.

$$|x_t - x'_t| > \tau,$$

Όπου: x_t η τιμή μίας μεταβλητής την χρονική στιγμή t, x'_t η πιθανή τιμή που έδωσε το μοντέλο και τ η τιμή κατωφλίου. Παρακάτω παρατίθεται και διαγραμματική απεικόνιση.

Διάγραμμα:



Η σκιαγραφημένη περιοχή καθώς και οι 2 μαύρες καμπύλες γραμμές δεικτοδοτούν τα όρια στα οποία οι τιμές που θα δώσει το μοντέλο θα θεωρηθούν αποδεκτές και η εγγραφή θα λογιστεί ως φυσιολογική, ενώ πέρα απ' αυτά έχουμε ανώμαλη τιμή.

Εκεί όπου παρατηρείται διαφοροποίηση μεταξύ των μεθόδων είναι ως προς τον προσδιορισμό της εκτιμώμενης / προβλεπόμενης τιμής και στον προσδιορισμό της τιμής κατωφλίου.

Κατηγορία 2η : Density based.

Η δεύτερη κατηγορία μεθόδων αφορά σε μεθόδους που βασίζονται στην έννοια της πυκνότητας και των γειτόνων των εγγραφών των εξεταζόμενων συνόλων δεδομένων. Ο έλεγχος και ο καθορισμός του εάν ένα σημείο αποτελεί outlier ή όχι γίνεται με τον εξής τρόπο:

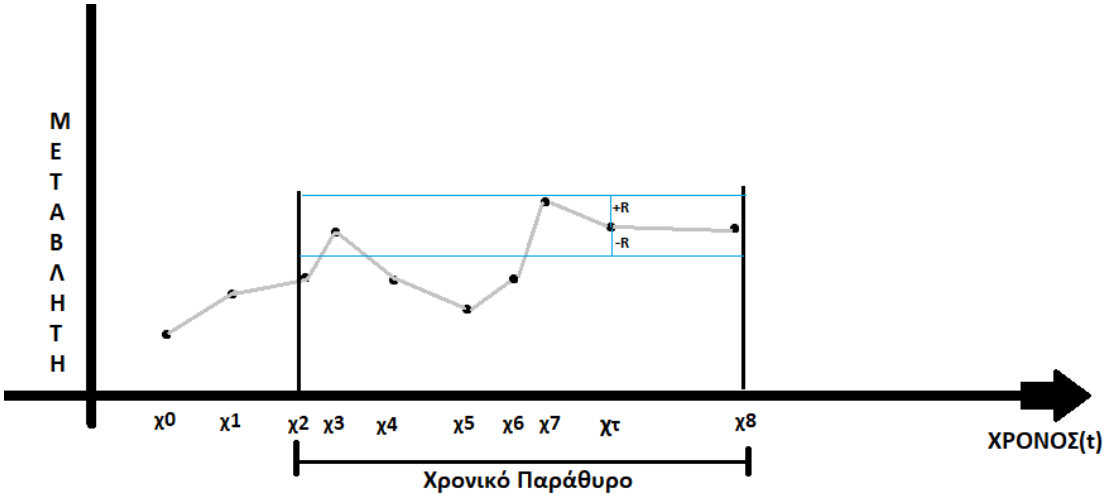
Εγγραφές του συνόλου, οι οποίες περιέχουν λιγότερες από 'τ' εγγραφές εντός μίας ορισμένης απόστασης R απ' αυτές θεωρούνται outliers.

Η πρόταση αυτή μπορεί να οριστεί και ως εξής:

$$Η\ εγγραφή\ \chi_\tau\ \ είναι\ outlier\ αν - ν\ |\{\chi\ \in\ X\ |d(\chi, \chi_\tau\} < R| < \tau$$

Στην πρόταση αυτή πολλές μέθοδοι προσθέτουν και την έννοια των χρονικών παραθύρων μέσα στα οποία γίνεται ο έλεγχος των γειτόνων, καθώς σε πολλές περιπτώσεις η σημασιολογική ερμηνεία του χρόνου λήψης των δεδομένων είναι καίριας σημασίας και η απουσία κατανόησής της μπορεί να οδηγήσει σε λανθασμένα αποτελέσματα. Για παράδειγμα, εάν το υπό μελέτη σύνολο αφορά σε αισθητήρα μέτρησης θερμοκρασίας μπορεί κατά τις μεσημβρινές ώρες του καλοκαιριού, όπου υπάρχουν μεγάλες και απότομες διακυμάνσεις θερμοκρασίας να μην καλύπτεται ο απαραίτητος αριθμός γειτόνων, αλλά αν ληφθεί υπόψιν ο χρόνος λήψης των δεδομένων και η αναζήτηση τους γίνει σε χρονικό παράθυρο που περιλαμβάνει μόνο τις μεσημβρινές ώρες, τότε πιθανόν η μέτρηση να αποτελεί κανονικό στιγμιότυπο και όχι ανώμαλο.

Οι παραπάνω προτάσεις μπορούν να αποδοθούν και οπτικά στο παρακάτω διάγραμμα:

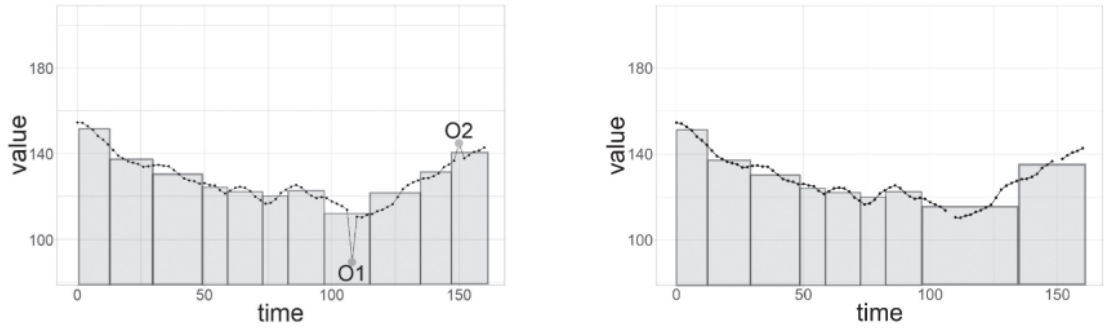


<u>Κατηγορία 3η :</u> Histogram based.

Η Τρίτη και τελευταία κατηγορία στην οποία θα γίνει αναφορά αφορά στις μεθόδους που βασίζονται στην χρήση ιστογραμμάτων. Σ' αυτήν οι ανώμαλες τιμές βρίσκονται με έναν ιδιαίτερο και ενδιαφέρον τρόπο.

Αρχικά , ορίζεται το πλήθος, καθώς και το εύρος (στον άξονα του χρόνου) των «κάδων» που πρόκειται να σχηματιστούν. Στην συνέχεια, το ύψος των επιμέρους «κάδων» υπολογίζεται βρίσκοντας την μέση τιμή των εγγραφών που τους ανήκουν. Μετά, ελέγχουμε τα ύψη των κάδων που προκύπτουν και ψάχνουμε να βρούμε ποιοι εξ' αυτών είναι προβληματικοί. Προβληματικός μπορεί να θεωρηθεί ένας κάδος που έχει απόκλιση ύψους μεγαλύτερη από κάποια τιμή κατωφλίου σε σχέση με τους υπόλοιπους. Έπειτα, ο σκοπός είναι να βρεθούν οι εγγραφές οι οποίες προκαλούν τις μεγαλύτερες αποκλείσεις μεταξύ των υψών των κάδων. Εφόσον, κάθε εγγραφή έχει την ίδια βαρύτητα στον υπολογισμό του ύψους του κάδου που την περιέχει, αυτό σημαίνει, ότι αντιστοίχως η εγγραφή που προκαλεί την απόκλιση των υψών θα είναι και αποκλίνουσα από τις υπόλοιπες εγγραφές. Οπότε, για τους «προβληματικούς» κάδους προσθαφαιρούμε εγγραφές απ' τον υπολογισμό της μέσης τιμής (και άρα του ύψους) μέχρι να εντοπιστεί ποια εξ' αυτών προκαλεί την απόκλιση. Τότε, έχει βρεθεί και η ανώμαλη τιμή.

Παράδειγμα σε μορφή διαγράμματος:



(a) Optimal histogram with eleven buckets.

(b) Optimal histogram with nine buckets and O1 and O2 removed.

(Εικόνα απ' το άρθρο 7)

5.1.2. Εντοπισμός σε multivariate data

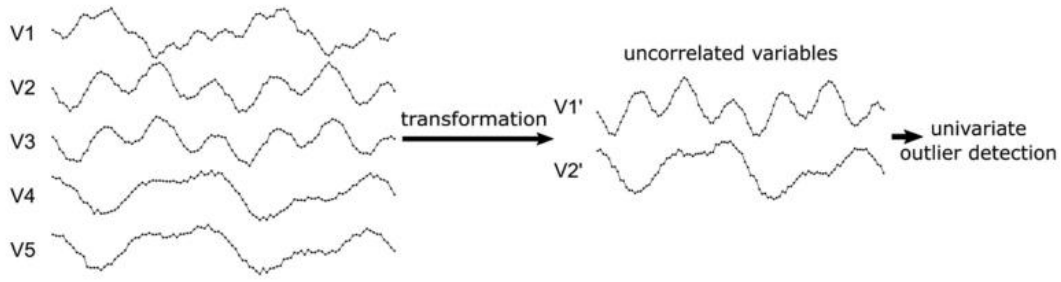
Κατά τον χειρισμό δεδομένων πολλαπλών μεταβλητών, υπάρχουν κοινά σημεία με αυτά των μονών μεταβλητών, αλλά και αρκετές διαφορές που προκύπτουν απ' την φύση των δεδομένων. Όπως αναφέρθηκε και προηγουμένως τα multivariate data μπορούν να

αναλυθούν χρησιμοποιώντας είτε τεχνικές ανεπτυγμένες για univariate data (μέσω διαδοχικής μελέτης της κάθε μεταβλητής), είτε από τεχνικές που αφορούν αποκλειστικά multivariate δεδομένα. Στην πρώτη εκδοχή υπάρχουν κάποια σημεία που χρίζουν ιδιαίτερη προσοχή.

Πιο συγκεκριμένα, όταν οι μεταβλητές που ανήκουν σε multivariate δεδομένα εξετάζονται ξεχωριστά υπάρχει μεγάλη πιθανότητα να μην λαμβάνεται υπόψιν κάποια τυχόν συσχέτιση μεταξύ των μεταβλητών αυτών, γεγονός το οποίο θα οδηγήσει σε απώλεια πληροφορίας και καταληκτικά σε λανθασμένα συμπεράσματα. Προκειμένου να αντιμετωπιστεί η ιδιαιτερότητα αυτή χρειάζεται να γίνουν ορισμένες πράξεις στο στάδιο της προ – επεξεργασίας δεδομένων και συγκεκριμένα να εφαρμοστούν μέθοδοι μείωσης της διαστασιμότητας με σκοπό την απαλοιφή της συσχέτισης των μεταβλητών. Η μείωση της διαστασιμότητας μπορεί να έχει ως αποτέλεσμα είτε μία νέα univariate μεταβλητή που στην συνέχεια επεξεργάζεται όπως έχει αναλυθεί παραπάνω, είτε σε μία νέα multivariate μεταβλητή που είναι απαλλαγμένη απ’ την συσχέτιση των μεταβλητών και εμπεριέχει όλη την πληροφορία.

Για την μείωση της διαστασιμότητας οι πιο διαδεδομένες χρησιμοποιούμενες τεχνικές είναι αλγόριθμοι που βασίζονται σε Principal Component Analysis (PCA), Autoregressive Models (AR) και Independent Component Analysis (ICA).

Παράδειγμα:



(Εικόνα απ’ το άρθρο 7)

Κατηγορίες μεθόδων εντοπισμού point outliers σε multivariate data

Οι κατηγορίες που ορίζονται στα multivariate data ταυτίζονται σε μεγάλο ποσοστό με αυτές των univariate. Κι εδώ απαντώνται model based (estimation, prediction models) και histogram based για τις οποίες η ουσιαστική διαφορά είναι ότι οι εγγραφές των συνόλων αντιμετωπίζονται ως διανύσματα που έχουν μέγεθος ίσο με το πλήθος των μεταβλητών.

5.2 Εντοπισμός subsequence outliers σε χρονοσειρές

Τα subsequence outliers είναι το δεύτερο είδος ανώμαλων τιμών που μπορεί να παρατηρηθεί σε χρονοσειρές δεδομένων. Όπως έχει προαναφερθεί, αυτά αποτελούνται από πολλαπλές εγγραφές που όταν εμφανίζονται μαζί δίνουν την αίσθηση, ότι είναι ανώμαλα. Λόγω της φύσης τους είναι πιο δύσκολο να γίνουν αντιληπτά και οι μέθοδοι που αναπτύσσονται για την εύρεσή τους λαμβάνουν υπόψιν κάποια βασικά χαρακτηριστικά τους.

Το πρώτο χαρακτηριστικό αφορά στο μέγεθός τους (δηλαδή από πόσες εγγραφές απαρτίζονται). Οι μέθοδοι προσαρμόζονται στο χαρακτηριστικό αυτό με 2 τρόπους. Ο πρώτος εξ' αυτών είναι η μέθοδος να κατασκευάζεται με τρόπο που εντοπίζει subsequence outliers συγκεκριμένου μεγέθους. Εάν η μέθοδος έχει αυτή την προσέγγιση, τότε πρέπει ο χρήστης να ορίζει το μέγεθος, που θα κληθεί να αναζητηθεί κατά την εκτέλεση. Ο δεύτερος τρόπος αφορά σε μεθόδους που το μέγεθος του subsequence δεν είναι ορισμένο και είναι μεταβλητό κατά την εκτέλεση. Επιπλέον, πρέπει να αναφερθεί, πως με βάση το μέγεθος που ορίζεται στην αναζήτηση διαμορφώνεται και η ταχύτητα εκτέλεσης. Είναι ευκόλως κατανοητό, ότι έχοντας ένα μικρό μέγεθος subsequence δημιουργούνται περισσότερα υποψήφια υποσύνολα προς μελέτη.

Το δεύτερο χαρακτηριστικό αφορά στον τρόπο αναπαράστασης των δεδομένων και το τι μορφή πρέπει να έχει το σύνολο, για να γίνεται η διαδικασία της μελέτης του πιο εύκολη.

Για παράδειγμα, πολλές μέθοδοι επιλέγουν να εφαρμόζουν τεχνικές διακριτοποίησης των μεταβλητών που αντιστοιχούν στα χαρακτηριστικά των δεδομένων. Η διαδικασία αυτή κρίνεται αναγκαία, καθώς προκειμένου να κατηγοριοποιηθεί ένα υποσύνολο ως outlier, η συνήθης προσέγγιση είναι η σύγκριση διαφόρων υποσυνόλων μεταξύ τους. Η σύγκριση αυτή είναι πολύ πιο εύκολη, όταν γίνεται με διακριτές μεταβλητές παρότι με συνεχείς.

Το τρίτο χαρακτηριστικό αφορά στην περιοδικότητα εμφάνισης κάποιου υποσυνόλου. Το χαρακτηριστικό αυτό ίσως είναι και το σημαντικότερο, διότι εισάγει στην μελέτη και την σημασιολογία του χρόνου εμφάνισης. Ένα υποσύνολο με μία πρώτη ματιά μπορεί να μην κινεί το ενδιαφέρον για κατηγοριοποίηση ως outlier, αλλά εάν παρατηρηθεί γι' αυτό ότι ακολουθεί κάποιο περιοδικό μοτίβο (δηλαδή εμφανίζεται αυτούσιο με κάποια συγκεκριμένη συχνότητα), τότε μπορεί όντως να αποτελεί ανωμαλία που πρέπει να αξιολογηθεί.

Επιπλέον, αξίζει να αναφερθεί, πως στα subsequence outliers η παροδικότητα (χρόνος λήψης των δεδομένων) έχει σημασία και γι' αυτό όλες οι μέθοδοι που τα πραγματεύονται την λαμβάνουν υπόψιν.

5.2.1 Εντοπισμός subsequence outliers σε univariate data

Κατηγορία 1η : Discord detection

Η πρώτη κατηγορία εύρεσης subsequence outliers σε χρονοσειρές είναι αυτή που βασίζεται στον εντοπισμό discord στο σύνολο δεδομένων. Με τον όρο discord εκφράζεται η ασυμφωνία ή η έλλειψη αρμονίας στα δεδομένα. Προκειμένου να εφαρμοστούν μέθοδοι της κατηγορίας αυτής πρέπει αρχικά να σχηματιστεί ένα σύνολο που περιέχει όλες τα δυνατά υποσύνολα που μπορούν να σχηματιστούν από το σύνολο εγγραφών. Καλή τακτική για την εύρεση των υποσυνόλων είναι η χρήση τεχνικής ολισθαίνοντος παραθύρου με κάποιο συγκεκριμένο βήμα.

Στην συνέχεια εφόσον έχει οριστεί το σύνολο με τα subsequences πρέπει να γίνει σύγκριση του καθενός εκ των υποσυνόλων με όλα τα υπόλοιπα.

Η σύγκριση γίνεται χρησιμοποιώντας μετρικές αποστάσεων (πιο συνηθισμένη είναι η ευκλείδεια απόσταση).

Όπως γίνεται αντιληπτό, η προσέγγιση αυτή έχει μεγάλη πολυπλοκότητα, καθώς πρέπει να ελεγχθούν όλα τα ζεύγη, γεγονός το οποίο απ' την έναρξη της επίλυσης έχει τουλάχιστον τετραγωνική πολυπλοκότητα. Υπάρχουσες μέθοδοι εντάσσουν στην διαδικασία heuristic μηχανισμούς και τεχνικές pruning κάποιων άσκοπων ελέγχων προκειμένου να περιορίσουν τον αριθμό των ελέγχων με σκοπό την μείωση της πολυπλοκότητας, για να είναι «βιώσιμη» επιλογή σε πραγματικές εφαρμογές.

Κατηγορία 2η: Dissimilarity based

Η κατηγορία αυτή έχει ιδιαίτερο ενδιαφέρον, καθώς προκειμένου να γίνει η κατηγοριοποίηση μεταξύ outlier και κανονικών subsequences γίνονται συγκρίσεις μεταξύ των υπαρχόντων υποσυνόλων και υποσυνόλων που θεωρούνται φυσιολογικά. Αυτό που κάνει την κατηγορία και τις μεθόδους που εντάσσονται σ' αυτήν ενδιαφέρουσα είναι οι τρόποι με τους οποίους λαμβάνονται οι θεωρητικές «φυσιολογικές» εκφάνσεις των υποσυνόλων. Υπάρχουν 3 τρόποι:

1. Χρησιμοποιείται ως «φυσιολογική» έκφανση η ίδια η χρονοσειρά που περιέχει τα υποσύνολα.
2. Χρησιμοποιείται μία εξωτερική χρονοσειρά
3. Χρησιμοποιείται το προηγούμενο υποσύνολο απ' το εξεταζόμενο.

Στην συνέχεια, εφόσον υπάρχουν τα υποσύνολα και έχει γίνει επιλογή του θεωρητικού φυσιολογικού στιγμιότυπου υπολογίζεται το dissimilarity τους ή αλλιώς το πόσο διαφορετικά είναι μεταξύ τους. Ταυτοχρόνως, ορίζεται και μία τιμή κατωφλίου, που όταν ξεπεραστεί θεωρούμε ότι βρέθηκε subsequence outlier.

Ο τρόπος ορισμού της τιμής κατωφλίου, καθώς και ο τρόπος υπολογισμού της διαφορετικότητας μεταξύ των φυσιολογικών και των πραγματικών υποσυνόλων διαφέρει από μέθοδο σε μέθοδο.

Κατηγορία 3η: Prediction based.

Η κατηγορία αυτή ταυτίζεται με την αντίστοιχη, που είχε αναλυθεί και στα point outliers. Πιο συγκεκριμένα, επιχειρείται να δημιουργηθεί ένα μοντέλο πρόβλεψης νέων υποσυνόλων από γεγονότα του παρελθόντος και στην συνέχεια με βάση το μοντέλο αυτό να μετρηθεί το κατά πόσο αποκλίνουν οι πραγματικές τιμές από τις προβλεπόμενες.

Βέβαια, καθώς εξετάζουμε subsequences υπάρχουν και κάποιες διαφορές απ' την αντίστοιχη των point outliers. Αναλυτικότερα, το μοντέλο πρόβλεψης που κατασκευάζεται αντί να παράγει μία προβλεπόμενη τιμή παράγει ένα προβλεπόμενο υποσύνολο. Το υποσύνολο αυτό έχει ίδιο μέγεθος με το υποσύνολο που πρόκειται να συγκριθεί. Στην συνέχεια, γίνεται ένα προς ένα σύγκριση όλων των εγγραφών που εμπεριέχονται στα υποσύνολα και εξετάζεται η απόκλισή τους με βάση την τιμή κατωφλίου. Εάν παρατηρηθεί απόκλιση μεγαλύτερη του κατωφλίου σε κάποια εκ των συγκρίσεων τότε θεωρείτε, ότι βρέθηκε ανώμαλο subsequence.

Κατηγορία 4η : Frequency based

Η κατηγορία που βασίζεται στην συχνότητα μελετάει το πόσο συχνά συναντάται ένα subsequence σε ένα σύνολο δεδομένων. Ο τρόπος που λειτουργούν οι μέθοδοι της κατηγορίας αυτής μοιάζουν σε μερικά σημεία με άλλες που έχουν αναφερθεί παραπάνω. Πρακτικά, στις μεθόδους αυτές υπάρχει πάλι μία χρονοσειρά με βάση την οποία ορίζεται η «κανονική» μορφή των δεδομένων.

Στην συνέχεια, γίνεται έλεγχος για το εάν η συχνότητα εμφάνισης ενός υποσυνόλου αποκλίνει από την αναμενόμενη εντός ορίων κάποιας τιμής κατωφλίου.

Η «κανονικότητα» των δεδομένων μπορεί πολλές φορές να σχετίζεται και με την σημασιολογία των χαρακτηριστικών των εγγραφών, οπότε ως χρονοσειρά που ορίζει την κανονικότητα θεωρείται η ίδια η εξεταζόμενη χρονοσειρά. Για παράδειγμα, εάν η χρονοσειρά δεδομένων αφορούσε σε τραπεζικές συναλλαγές ενός πελάτη, ο οποίος πληρώνει κάποιες σταθερές δόσεις (π.χ. δάνειο) που έχουν σταθερή μηνιαία εμφάνιση, τότε η εκτιμώμενη συχνότητα θα

ήταν ο ένας μήνας και αν μία πληρωμή γινόταν σε λάθος χρόνο (απόκλιση απ' τον ένα μήνα), τότε θα είχαμε υποψήφιο υποσύνολο outlier προς αξιολόγηση.

Κατηγορία 5η : Information theory based.

Η κατηγορία αυτή συνδέεται νοητικά με την προηγούμενη, διότι και σ' αυτήν παίζει ρόλο η περιοδικότητα των υποσυνόλων σε μία χρονοσειρά. Όμως, εισάγεται και η έννοια της πληροφορίας που «κουβαλάει» ένα υποσύνολο.

Πιο συγκεκριμένα, με βάση το προηγούμενο παράδειγμα των τραπεζικών συναλλαγών. Subsequences που αφορούν σε πληρωμές δόσεων που επαναλαμβάνονται σταθερά στον χρόνο, δεν δίνουν ιδιαίτερη πληροφορία. Συνεπώς, στην κατηγορία 5 εισάγεται εκτός απ' την συχνότητα και μία συνάρτηση υπολογισμού της πληροφορίας (I(S)).

Όπως είναι ευκόλως κατανοητό, όταν για ένα subsequence η συχνότητα εμφάνισής του είναι μεγάλη, τότε δεν φέρει πολλή πληροφορία, ενώ αντιθέτως όταν αυτό έχει σχετικά μικρή συχνότητα εμφάνισης είναι πιο πιθανό να περιέχει χρήσιμες πληροφορίες. Η σχέση αυτή μεταξύ συχνότητας και πληροφορίας μπορεί να αναπαρασταθεί στην παρακάτω σχέση.

$$I(S)\ x\ f(S) > \tau$$

Όπου S το εξεταζόμενο subsequence, *I(S)* η πληροφορία που μεταφέρει, f(s) η συχνότητα εμφάνισής του και τ μία τιμή κατωφλίου όπου όταν το γινόμενο
I(S) x f(s) την ξεπεράσει θεωρούμε ότι βρέθηκε outlier.

5.2.2 Εντοπισμός subsequence outliers σε multivariate data

Όπως και στα point outliers σε multivariate data, έτσι κι εδώ χρησιμοποιούνται τεχνικές που είτε είναι κατασκευασμένες για univariate δεδομένα και εκτελείται ξεχωριστός έλεγχος στις k μεταβλητές των δεδομένων ή υπάρχουν μέθοδοι που είναι ανεπτυγμένες ξεκάθαρα για multivariable data.

Οι κατηγορίες και οι τρόποι λειτουργίας τους βρίσκουν αντιστοιχία με προηγούμενες. Αναλυτικότερα:

Κατηγορία 1η : Model based – Estimation based.

Οι τεχνικές της κατηγορίας αυτής λειτουργούν με τον ίδιο τρόπο που αναφέρθηκε και προηγουμένως με την ιδιαιτερότητα ότι πλέον οι εκτιμήσεις και οι προβλέψεις που γίνονται αφορούν subsequences. Κατά τ' άλλα τα χαρακτηριστικά των μεθόδων είναι παρόμοια.

Κατηγορία 2η : Dissimilarity based.

Η dissimilarity based κατηγορία είναι η λιγότερο ανεπτυγμένη όσον αφορά τον εντοπισμό subsequence ανώμαλων τιμών σε multivariate data. Και σ ' αυτήν την κατηγορία τα γνωρίσματα των μεθόδων ταυτίζονται με τα προηγούμενα, αλλά καθώς δεν χρησιμοποιείται σε μεγάλο βαθμό σε πρακτικές εφαρμογές

5.3 Εντοπισμός time series outliers

Ο τελευταίος τύπος outlier, που μπορεί να βρεθεί σε χρονοσειρές δεδομένων είναι τα time series outliers. Ο τύπος αυτός μπορεί να απαντηθεί μόνο σε δεδομένα πολλαπλών μεταβλητών, διότι αυτό που διαφοροποιεί μία χρονοσειρά και την καθιστά ανώμαλη είναι το πως αυτή συσχετίζεται και πως διαφέρει από άλλες χρονοσειρές άλλων μεταβλητών.

Στα time series outliers, καθώς χρησιμοποιείται ολόκληρο το σύνολο της χρονοσειράς η παροδικότητα των δεδομένων μπορεί και να μην λαμβάνεται υπόψιν, δίχως αυτό να συνεπάγεται απώλεια πληροφορίας.

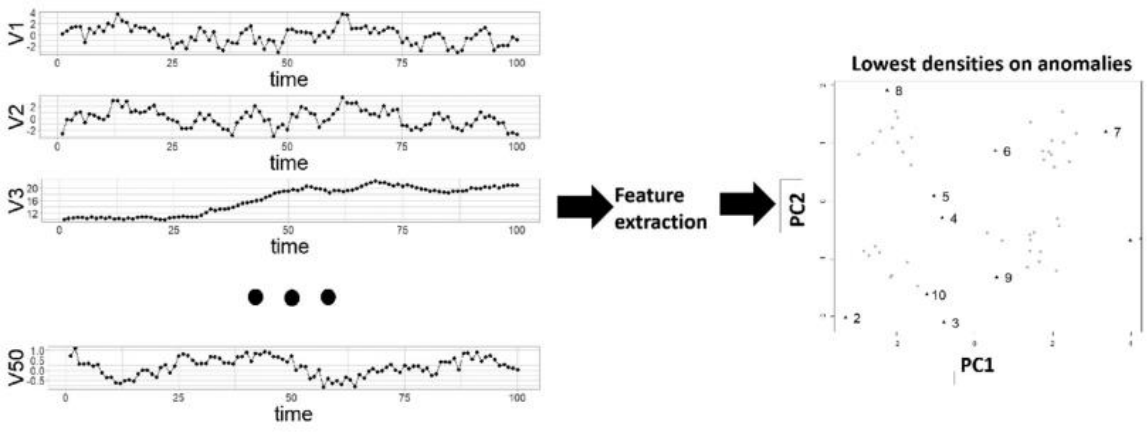
Για την εύρεση των ανώμαλων χρονοσειρών ορίζεται η εξής κατηγορία μεθόδων.

Κατηγορία 1η: Dimensionality reduction

Η κατηγορία που χρησιμοποιεί μείωση της διαστασιμότητας έχει το ίδιο σκεπτικό με την μέθοδο που παρουσιάστηκε στο κεφάλαιο 4.4, όπου με χρήση Principal Component Analysis μεταφερόμαστε από ένα πρόβλημα πολλαπλών διαστάσεων σε μία αναπαράσταση με λιγότερες και πιο εύκολα διαχειρίσιμες διαστάσεις.

Στην συνέχεια, για τον εντοπισμό των time series outliers εφαρμόζονται τεχνικές συσταδοποίησης στο sparse matrix που προκύπτει απ' την μέθοδο PCA. Απ' την συσταδοποίηση προκύπτουν τα κέντρα των συστάδων τα οποία χρησιμοποιούνται για να βρεθούν τα πιο απομακρυσμένα σημεία, τα οποία όμως αναπαριστούν τις χρονοσειρές του αρχικού προβλήματος.

Μ' αυτόν τον τρόπο επιτυγχάνεται η εύρεση των ανώμαλων χρονοσειρών. Παρακάτω δίνεται και διαγραμματική απεικόνιση της πορείας της μεθόδου.



(Εικόνα απ' το άρθρο 7)

5.4 Σύνοψη ενότητας και παρατηρήσεις

Φτάνοντας στο τέλος της παρούσης ενότητας γίνεται αντιληπτή η σημαντική συνεισφορά που μπορεί να έχει ο εντοπισμός ανωμαλιών σε χρονοσειρές. Στην ουσία, τέτοιου είδους μεθόδους και η περαιτέρω επέκτασή τους είναι αυτές που δίνουν την δυνατότητα να επιτυγχάνονται οι στόχοι και οι εφαρμογές που παρατέθηκαν στην αρχή του κειμένου.

Παρατηρήσαμε, ότι τα δεδομένα στα οποία εισάγεται η έννοια του χρόνου έχουν πολλές ιδιαιτερότητες σε σχέση με τον ορισμό της ανώμαλης τιμής, του τρόπου εντοπισμού της αλλά και τα διάφορα χαρακτηριστικά, προκύπτουν ανά περίπτωση, για τα οποία πρέπει να δίνεται ιδιαίτερη προσοχή. Επίσης, είδαμε πως χρησιμοποιούνται σε πολλές περιπτώσεις συνδυαστικές προσεγγίσεις σε διάφορα στάδια της εκτέλεσης που επιτρέπουν στις μεθόδους να εφαρμόζονται σε πραγματικές εφαρμογές (π.χ. διακριτοποίηση και μείωση της διαστασιμότητας στο στάδιο της προ – επεξεργασίας, μετρική της απόστασης, στοιχεία στατιστικής κ.α.).

Στην συνέχεια του κειμένου, θα παρουσιαστούν υλοποίησεις τέτοιων μεθόδων και θα γίνει λόγος στο πως όλα όσα αναφέρθηκαν παραπάνω δίνουν την δυνατότητα σε επιστήμονες της πληροφορικής, αλλά και σε άλλου τύπου χρήστες να αξιοποιούν το outlier detection με σκοπό την βελτίωση δραστηριοτήτων τους, είτε αφορά τις επαγγελματικές τους ανάγκες, είτε άλλες καθημερινές ενέργειες.

6. Υλοποίηση τεχνικών εντοπισμού ανωμαλιών σε γλώσσα προγραμματισμού Python

Στην συγκεκριμένη ενότητα θα παρατεθούν ενδεικτικές υλοποίησεις μεθόδων εντοπισμού ανωμαλιών που αναπτύχθηκαν σε γλώσσα προγραμματισμού Python. Οι μέθοδοι και ο τρόπος λειτουργίας τους αφορούν στις πιο χαρακτηριστικές κατηγορίες που αναφέρθηκαν στις προηγούμενες ενότητες. Για την καλύτερη κατανόηση των μεθόδων θα παρατεθεί και ο σχετικός πηγαίος κώδικας με λεπτομερή σχολιασμό.

6.1 Χρησιμοποιούμενο σύνολο δεδομένων

Πριν ξεκινήσει η παράθεση των μεθόδων θα γίνει λόγος για τα σύνολα δεδομένων, που θα χρησιμοποιηθούν στις δοκιμές. Για τις ανάγκες της εργασίας αναπτύχθηκε ένα απλό πρόγραμμα δημιουργίας τυχαίων σημείων (X,Y). Επιλέχθηκε η συγκεκριμένη προσέγγιση, καθώς η χρήση παραδειγμάτων με αριθμητικές τιμές είναι πιο ευκολονόητη για την ανάδειξη των μεθόδων, καθώς είναι πιο εύκολο να χρησιμοποιηθούν απαραίτητες έννοιες όπως π.χ. η απόσταση δύο σημείων.

Κατασκευαστής σημείων – Point generator

Για την κατασκευή των συνόλων δεδομένων χρησιμοποιήθηκαν οι βιβλιοθήκες : Random, csv, pandas.

Αρχικά αναπτύχθηκε η μέθοδος random_points(start, end, num).

```
def random_points(start, end, num):
    random_list = []
    for i in range(num):
        random_element = []
        for j in range(2):
            random_element.append(random.randint(start, end))
        random_list.append(random_element)

    return random_list
```

Η συγκεκριμένη μέθοδος χρησιμοποιεί μεθόδους της βιβλιοθήκης random προκειμένου να φτιάξει δύο λίστες με τυχαίους αριθμούς μέσα σε ένα εύρος τιμών (ορίσματα start, end). Η κάθε λίστα έχει αριθμούς ίσους με το όρισμα num.

Οι δύο αυτές λίστες επισυνάπτονται σε μία, η οποία επιστρέφεται σαν αποτέλεσμα.

Στην συνέχεια με χρήση των βιβλιοθηκών csv και pandas η λίστα των τυχαίων σημείων λαμβάνει την τελική της μορφοποίηση και αποθηκεύεται σε μορφή csv αρχείου για χρήση στις μεθόδους που θα αναπτυχθούν.

```
def main():
    header = ['X', 'Y']

    random_list = random_points(1, 100, 100)
    #for i in range(len(random_list)):
        #print(random_list[i])

    with open("datasetSmall.csv", 'w', encoding='UTF8') as f:
        writer = csv.writer(f)
        writer.writerow(header)
        writer.writerows(random_list)

    df = pd.read_csv('datasetSmall.csv')
    modifieddf = df.dropna()
    modifieddf.to_csv('datasetSmall.csv', index=False)
```

6.2 Αρχείο κλάσεων

Επιπλέον, κρίθηκε ορθό να χρησιμοποιηθούν έννοιες του αντικειμενοστραφή προγραμματισμού, για να οριστούν κλάσεις και αντικείμενά τους κατάλληλα προς χρήση για τις χρησιμοποιούμενες εφαρμογές.

Η βασικότερη κλάση που θα χρησιμοποιηθεί είναι η κλάση Data Point, η οποία αναπαριστά τα σημεία αριθμών και περιέχει τις παρακάτω ιδιότητες:

```
class DataPoint:

    def __init__(self, dimensions, values):
        self.neighbors = []
        self.dimensions = dimensions
        self.distances = []
        self.values = []
        self.isOutlier = False
        for i in range(dimensions):
            self.values.append(values[i])
        self.score = 0
        self.bin_x = 0
        self.bin_y = 0
```

- Neighbors: Λίστα που θα κρατάει τους γείτονες των σημείων (στις σχετικές μεθόδους)
- Dimensions : Διαστάσεις των σημείων (Για να μπορεί να υπάρξει γενίκευση αν χρειαστεί να προστεθούν διαστάσεις)
- Distances: Αποστάσεις ενός σημείου απ' τους γείτονές του.
- Values: Οι τιμές (X,Y) των σημείων.
- isOutlier: Boolean μεταβλητή που τίθεται True, αν κάποιο σημείο κριθεί ανώμαλο.
- Score: Μεταβλητή που θα χρησιμοποιηθεί για να κρατήσει την βαθμολογία ενός σημείου κατά την διαδικασία καθορισμού του.
- bin_x, bin_y: Μεταβλητές που θα χρησιμοποιηθούν σε στατιστικές μεθόδους. Θα εξηγηθούν σε παρακάτω σκέλος.

Οι υπόλοιπες κλάσεις θα παρατεθούν σε σημεία του κειμένου που χρησιμοποιούνται.

6.3 Υλοποίηση Nearest Neighbor based μεθόδων

Στην κατηγορία των Nearest Neighbor based μεθόδων υλοποιήθηκαν οι global k_NN , k^{th}_NN. Για την υλοποίησή τους χρειάστηκαν οι παρακάτω βιβλιοθήκες: numpy – array , csv, math, time (Για υπολογισμό του χρόνου εκτέλεσης) και η κλάση Data Point απ' το αρχείο κλάσεων.

Υλοποιημένες συναρτήσεις:

Συνάρτηση **get_data()**: Συνάρτηση, η οποία χρησιμοποιείται για την λήψη των δεδομένων από το αρχείο σημείων που δημιουργήθηκε προηγουμένως.

```
def get_data():

    with open('datasetSmall.csv', newline='') as f:
        reader = csv.reader(f)
        data = list(reader)

    return data
```

Συνάρτηση **calculate_distance(a: DataPoint, b: DataPoint):** Συνάρτηση η οποία χρησιμοποιείται για την εύρεση της απόστασης μεταξύ δύο αντικειμένων της κλάσης DataPoint (Για τον υπολογισμό της απόστασης χρησιμοποιήθηκε η Ευκλείδεια).

```
def calculate_distance(a: DataPoint, b: DataPoint):
    sum_of_values = 0
    for i in range(a.dimensions):
        sum_of_values += pow(a.values[i] - b.values[i], 2)

    return math.sqrt(sum_of_values)
```

Μέθοδος k_NN:

```
# Global method -Outlier score is set to  be the average to distance to the average distance
# of the data point to its k-nearest neighbors.
def k_nn(list_of_data_points, k):

    scores = []
    sum_of_dist = 0
    for obj in list_of_data_points:
        for i in range(k):
            sum_of_dist += obj.distances[i]
        avg_distance = sum_of_dist/k
        scores.append([list_of_data_points.index(obj), avg_distance])
        sum_of_dist = 0

    sorted_scores = sorted(scores, key=lambda x: x[1], reverse=True)
    print(sorted_scores)
    print(scores)
```

Για την μέθοδο αυτή ορίζουμε μία λίστα βαθμολογιών. Σ’ αυτήν για κάθε σημείο του συνόλου δεδομένων υπολογίζουμε την μέση απόσταση απ’ τους γείτονές του και την αποθηκεύουμε. Η λίστα βαθμολογιών πέρα από την βαθμολογία του κάθε σημείου κρατάει και τον δείκτη της θέσης στην οποία αυτό βρίσκεται στο αρχικό σύνολο δεδομένων. Στην συνέχεια οι βαθμολογίες αυτές ταξινομούνται έτσι ώστε να βρεθούν αυτές με την μεγαλύτερη βαθμολογία. Στην συνέχεια με βάση τις βαθμολογίες μπορεί να καθοριστεί ο τύπος των σημείων (φυσιολογικό/ ανώμαλο).

Μέθοδος k^{th}_NN : Με την ίδια φιλοσοφία υλοποιείται και η μέθοδος k^{th}_NN.

```
# Global method - Outlier score is set to be the distance to the k-th neighbor
def k_nn_th(list_of_data_points, k):

    scores = []
    i = 0
    for obj in list_of_data_points:
        scores.append([list_of_data_points.index(obj), obj.distances[k-1]])
        i += 1
    scores.sort(reverse=True)
    return scores
```

Η βασική διαφορά είναι ότι ως βαθμολογία ορίζεται η απόσταση απ' τον μακρινότερο γείτονα.

Και στις δύο μεθόδους ο υπολογισμός των αποστάσεων και η κατηγοριοποίηση των σημείων γίνεται σε τμήμα κώδικα μία μεθόδου main.

Παράδειγμα εκτέλεσης της μεθόδου k_nn:

```
def main():
    # Setting the start time
    start_time = time.time()

    # Reading data file to retrieve data points/
    data = get_data()
    d = array(data)
    dimensions = d.shape[1]

    # Change data type for latter use
    for i in range(len(data) - 1):
        for j in range(2):
            data[i+1][j] = int(data[i+1][j])

    # Creating data points objects
    list_of_data_points = []
```

```
for i in range(len(data) - 1):
    list_of_data_points.append(DataPoint(dimensions, data[i+1]))

# for obj in list_of_data_points:
    # print(obj.values)

for i in range(len(list_of_data_points)):
    for j in range(len(list_of_data_points)):
        list_of_data_points[i].distances.append(calculate_distance
                                                (list_of_data_points[i], list_of_data_points[j]))

for obj in list_of_data_points:
    obj.distances.sort()

#print(list_of_data_points[0].distances)
k_nn(list_of_data_points, 5)

# Getting the end time
end_time = time.time()
print("Time needed for code execution: ", end_time - start_time, " seconds.")
```

Αποτελέσματα:

```
[[96, 14.770332954278407], [97, 13.328428128526781], [46, 12.350989281683171], [29, 11.959136366432073], [89, 11.876607184356274]]
Time needed for code execution:  0.014004230499267578  seconds.
```

Τα 5 σημεία που είχαν την μεγαλύτερη βαθμολογία ήταν τα :

- Σημείο 96 (35,66)
- Σημείο 97 (34,73)
- Σημείο 46 (24,17)
- Σημείο 29 (15,57)
- Σημείο 89 (21,32)

Η μέθοδος χρειάστηκε 0.014 δευτερόλεπτα για να εντοπίσει τα πιθανά ανώμαλα σημεία. Με τον ίδιο τρόπο μπορεί να εκτελεστεί και η άλλη μέθοδος με αποτελέσματα:

```
[[46, 23.08679276123039], [89, 20.024984394500787], [97, 19.849433241279208], [96, 19.6468827043885], [32, 18.601075237738275]]
Time needed for code execution:  0.011987209320068836  seconds.
```

- Σημείο 46 (24,17)
- Σημείο 89 (21,32)
- Σημείο 97(34,73)
- Σημείο 96 (35,66)
- Σημείο 32 (68,91)

Η μέθοδος χρειάστηκε 0.011 δευτερόλεπτα για να εντοπίσει τα πιθανά ανώμαλα σημεία.

Παρατήρηση: Παρόλο που οι μέθοδοι έχουν διαφοροποίηση στον καθορισμό της βαθμολογίας, τα αποτελέσματά τους βρίσκουν «κοντινά» αποτελέσματα και χρειάζονται περίπου τον ίδιο χρόνο.

Σημείωση: Οι παραπάνω υλοποιήσεις αν και μπορούν να εντοπίσουν με επιτυχία outliers δεν είναι βέλτιστες και επιδέχονται βελτιώσεων.

6.4 Υλοποίηση statistical based μεθόδου

Στην κατηγορία των μεθόδων που βασίζονται σε στατιστικές προσεγγίσεις επιλέχθηκε η υλοποίηση της μεθόδου HBOS, η οποία χρησιμοποιεί ιστογράμματα. Χρησιμοποιήθηκαν οι βιβλιοθήκες: csv, time, numpy – array και το αρχείο των κλάσεων.

Για την υλοποίηση της μεθόδου αυτής πέρα απ' την κλάση DataPoint χρειάστηκε να οριστούν οι παρακάτω:

Κλάση Bin: Κλάση η οποία αναπαριστά τους «κάδους» του ιστογράμματος.

```
class Bin:

    def __init__(self):
        self.points = []
        self.height = 0
```

Ένα αντικείμενο αυτής της κλάσης περιέχει τα εξής χαρακτηριστικά:

- Points: Λίστα η οποία περιέχει όλα τα Data Points που ανήκουν στον συγκεκριμένο «κάδο»
- Height: Μεταβλητή που αναπαριστά το ύψος του «κάδου», όπου το ύψος ισούται με το πλήθος εγγραφών που αντιστοιχούν σ' αυτόν.

Κλάση Histogram: Κλάση η οποία αναπαριστά το ιστόγραμμα.

```
class Histogram:
    def __init__(self):
        self.bins = []
```

Ένα αντικείμενο αυτής της κλάσης έχει το εξής χαρακτηριστικό:

- Bins: Μία λίστα η οποία περιέχει αντικείμενα της κλάσης Bins

Σημείωση: Όσον αφορά το πλάτος που θα έχουν οι κάδοι του ιστογράμματος, επιλέχθηκε να είναι σταθερό και όχι μεταβλητού μεγέθους. Το πλάτος επιλέγεται κατά την εκτέλεση του κώδικα και για τις ανάγκες του παραδείγματος εκτέλεσης επιλέχθηκε πλάτος ίσο με 10.

Υλοποιημένες συναρτήσεις:

Αρχικά χρησιμοποιήθηκε η ίδια συνάρτηση get_data() που χρησιμοποιήθηκε και προηγουμένως για λήψη των δεδομένων. Στην συνέχεια υλοποιήθηκε μία συνάρτηση με όνομα check_bin(num).

check_bin(num): Συνάρτηση, η οποία δέχεται έναν αριθμό ως όρισμα και βρίσκει σε ποιον κάδο πρέπει να τοποθετηθεί με βάση τον αριθμό αυτόν. Η συνάρτηση είναι προσαρμοσμένη για το μέγεθος του χρησιμοποιούμενου συνόλου δεδομένων και για το πλάτος των «κάδων», που έχει οριστεί παραπάνω.

```
def check_bin(num):

    if 0 < num < 10:
        return 1
    elif 10 <= num < 20:
        return 2
    elif 20 <= num < 30:
        return 3
    elif 30 <= num < 40:
        return 4
    elif 40 <= num < 50:
        return 5
    elif 50 <= num < 60:
        return 6
    elif 60 <= num < 70:
        return 7
    elif 70 <= num < 80:
        return 8
    elif 80 <= num < 90:
        return 9
    elif 90 <= num <= 100:
        return 10
```

Συνάρτηση **create_histogram(w, data_points):** Συνάρτηση, η οποία χρησιμοποιείται για την κατασκευή των ιστογραμμάτων. Δέχεται ως ορίσματα μία μεταβλητή w, η οποία αφορά στο πλάτος των «κάδων» και μία μεταβλητή data_points που είναι τα δεδομένα. Αρχικά, για κάθε διάσταση δημιουργούνται άδειοι κάδοι ίσοι με το πλήθος των εγγραφών διαιρεμένο με το πλάτος που έχει τεθεί.

Έπειτα οι κάδοι αυτοί δίνονται σε μορφή λίστας σε αντικείμενα τύπου Histogram. Πιο συγκεκριμένα, για κάθε μία απ' τις διαστάσεις των σημείων δημιουργείται ένα ξεχωριστό ιστόγραμμα. Μέσω προσπέλασης όλων των σημείων βρίσκεται σε ποιον κάδο ανήκει ο κάθε αριθμός, οι αριθμοί τοποθετούνται στις λίστες κάδων που δημιουργήθηκαν πριν. Επίσης στο σημείο αυτό χρησιμοποιούνται οι μεταβλητές bin_x, bin_y των αντικειμένων DataPoint, που δεν αναλύθηκαν προηγουμένως. Στις μεταβλητές αυτές αποθηκεύεται ο δείκτης του κάδου που ανήκει η κάθε μεταβλητή του σημείου, ώστε να χρησιμοποιηθεί η πληροφορία αυτή στον υπολογισμό της βαθμολογίας.

```
def create_histogram(w, data_points):

    histogram_x = Histogram()
    histogram_y = Histogram()

    for i in range(int(100/w)):
        histogram_x.bins.append(Bin())
        histogram_y.bins.append(Bin())

    for i in range(len(data_points)):
        for j in range(2):
            if j == 0:
                number = data_points[i].values[0]
                bin_num = check_bin(number)
                print(bin_num)
                histogram_x.bins[bin_num-1].points.append(number)
                histogram_x.bins[bin_num-1].height += 1
                data_points[i].bin_x = bin_num
```

```
            elif j == 1:
                number = data_points[i].values[1]
                bin_num = check_bin(number)
                histogram_y.bins[bin_num - 1].points.append(number)
                histogram_y.bins[bin_num - 1].height += 1
                data_points[i].bin_y = bin_num

    list_of_histograms = [histogram_x, histogram_y]
    return list_of_histograms
```

Τελικά επιστρέφεται μία λίστα που περιέχει τα ιστογράμματα που δημιουργήθηκαν για να χρησιμοποιηθούν για τους υπολογισμούς των βαθμολογιών.

Συνάρτηση **calculcate_scores(data_points, histograms):** Συνάρτηση στην οποία υπολογίζονται οι βαθμολογίες των σημείων σύμφωνα με την μέθοδο HBOS. Αναλυτικότερα, για κάθε σημείο του συνόλου εγγραφών βρίσκεται σε ποιο «κάδο» υπάγεται κάθε μία απ' τις συντεταγμένες του και υπολογίζεται η βαθμολογία όπως είχε αναλυθεί στην ενότητα 4. Μετά τον υπολογισμό της βαθμολογίας, αυτή αποθηκεύεται στην μεταβλητή score του κάθε data point. Στην συνέχεια οι βαθμολογίες και ο δείκτης του κάθε σημείου αποθηκεύονται σε έναν πίνακα αποτελεσμάτων. Ο πίνακας αυτός μετά από κατάλληλη ταξινόμηση επιστρέφεται ως αποτέλεσμα.

```
def calculate_scores(data_points, histograms):

    histogram_x = histograms[0]
    histogram_y = histograms[1]
    for i in range(len(data_points)):
        bin_x = data_points[i].bin_x
        bin_y = data_points[i].bin_y
        score = (1/histogram_x.bins[bin_x - 1].height)*(1/histogram_y.bins[bin_y - 1].height)
        data_points[i].score = score

    results = []
    for i in range(len(data_points)):
        element = [i, data_points[i].score]
        results.append(element)

    sorted_results = sorted(results, key=lambda x: x[1], reverse=True)
    print(sorted_results)
```

Παράδειγμα εκτέλεσης μεθόδου :

Στην μέθοδο main, υπάρχουν τα ίδια αρχικά βήματα που ακολουθήθηκαν και στις προηγούμενες μεθόδους. Έπειτα, γίνονται κατάλληλες κλήσεις στις υλοποιημένες συναρτήσεις όπως διαπιστώνεται παρακάτω.

```
def main():

    # Setting start time
    start_time = time.time()

    # Reading data file to retrieve data points/
    data = get_data()
    d = array(data)
    dimensions = d.shape[1]

    # Change data type for latter use
    for i in range(len(data) - 1):
        for j in range(2):
            data[i+1][j] = int(data[i+1][j])
```

```
    # Creating data points objects
    list_of_data_points = []

    for i in range(len(data) - 1):
        list_of_data_points.append(DataPoint(dimensions, data[i+1]))

    histograms = create_histogram(10, list_of_data_points)
    calculate_scores(list_of_data_points, histograms)

    #Getting end time
    end_time = time.time()
    print("Time needed for code execution: ", end_time - start_time, " seconds.")
```

Λαμβάνονται τα παρακάτω αποτελέσματα:

```
[[25, 0.02040816326530612], [52, 0.02040816326530612], [96, 0.018518518518518517], [53, 0.017857142857142856], [46, 0.016666666666666666]
Time needed for code execution:  0.000997304916381836  seconds.
```

- Σημείο 25 (71,7)
- Σημείο 52 (82,60)
- Σημείο 96 (35,66)
- Σημείο 53 (60,34)
- Σημείο 46 (24,17)

Η μέθοδος χρειάστηκε 0.00099 δευτερόλεπτα για να ολοκληρώσει την μέθοδο. Η στατιστική μέθοδος παρουσιάζει σημαντική χρονική βελτίωση σε σχέση με τις προηγούμενες πράγμα, το οποίο ήταν αναμενόμενο με βάση τα όσα αναφέρθηκαν στην ενότητα 4.

6.5 Density based method σε χρονοσειρά δεδομένων

Στην συγκεκριμένη υπο-ενότητα παρατίθεται υλοποίηση μεθόδου για εντοπισμό outlier σε χρονοσειρά δεδομένων. Καθώς, το προηγούμενο σύνολο δεδομένων χρησιμοποιούσε δεδομένα χωρίς την έννοια του χρόνου χρειάστηκε η δημιουργία ενός νέου κατασκευαστή δεδομένων αλλά και μίας νέας κλάσης για να αναπαριστά τον χρόνο. Πιο συγκεκριμένα ορίστηκαν:

- TimeObj Class: Μία κλάση, η οποία αναπαριστά τον χρόνο έχοντας δύο ιδιότητες κλάσης. Οι ιδιότητες είναι η ώρα και τα λεπτά.

```
class TimeObj:
    def __init__(self, h, m):
        self.hour = h
        self.minute = m
```

- TimePointGenerator: Ένας νέος κατασκευαστής σημείων, ο οποίος λειτουργεί όπως και ο πρωταρχικός, αλλά με την διαφορά, ότι δημιουργεί σημεία με την μορφή (Τιμή, Αντικείμενο κλάσης TimeObj). Στο αρχείο αποθήκευσης τα δεδομένα αποθηκεύονται με μορφή τριπλέτας (Τιμή, Ώρα, Λεπτά) και με κατάλληλη μορφοποίηση κατά την λήψη τους κατασκευάζονται εκ νέου τα αντικείμενα.

Η μέθοδος που υλοποιήθηκε αναφέρεται στην ενότητα 5.1.1. – Κατηγορία 2^{η} . Για την υλοποίησή της χρειάστηκε να φτιαχτούν πρώτα οι παρακάτω βοηθητικές συναρτήσεις:

Check_r(a, b, r): Η μέθοδος αυτή δέχεται δύο αριθμούς a,b και ελέγχει εάν οι αριθμοί αυτοί βρίσκονται σε απόσταση μικρότερη από r μεταξύ τους. Δηλαδή ελέγχει την παρακάτω ανισότητα:

$$|a - b| < r$$

```python
def check_r(a, b, r):

    if abs(a - b) < r:
        return True
    else:
        return False
```

Calculate_time_diff(a: TimeObj, b: TimeObj): Η μέθοδος αυτή υπολογίζει την διαφορά ώρας μεταξύ δύο αντικειμένων της κλάσης TimeObj σε λεπτά. Η χρήση της αφορά στον έλεγχο, εάν τα δύο αντικείμενα βρίσκονται χρονικά εντός ενός χρονικού παραθύρου.

```python
def calculate_time_diff(a: TimeObj, b: TimeObj):

    time_diff = 0

    # Finding starting and ending time
    if a.hour > b.hour:
        starting = b
        ending = a
    elif a.hour < b.hour:
        starting = a
        ending = b
    else:
        time_diff = abs(a.minute - b.minute)
        return time_diff

    # Calculating the difference
    if ending.minute > starting.minute:
        time_diff = abs(a.hour - b.hour) * 60 + abs(a.minute - b.minute)
    elif ending.minute < starting.minute:
        ending.hour -= 1
        ending.minute += 60
        time_diff = abs(a.hour - b.hour) * 60 + abs(a.minute - b.minute)

    # Returning the result
    return time_diff
```

Έχοντας τις συγκεκριμένες συναρτήσεις μπορεί να προχωρήσει η υλοποίηση της μεθόδου που είναι σχετικά απλή για κατανόηση.

Density_based_outlier_detection(r, time_window, t, data_points): Η μέθοδος αυτή δέχεται ως ορίσματα μία μεταβλητή r που αφορά στον υπολογισμό της απόστασης μεταξύ δύο τιμών, μία μεταβλητή time_window που αφορά στο χρονικό παράθυρο που πρέπει να ελεγχθεί εάν βρίσκονται εντός τα TimeObj αντικείμενα (η μεταβλητή είναι αριθμός που μεταφράζεται σε λεπτά), μία μεταβλητή t που αφορά στο ελάχιστο πλήθος εγγραφών που πρέπει να βρίσκονται εντός χρονικού παραθύρου και σε απόσταση μικρότερη από r για ένα εξεταζόμενο σημείο ώστε αυτό να μην θεωρείται outlier. Τέλος η μεταβλητή data_points είναι η μεταβλητή που περιέχει τα δεδομένα προς εξέταση. Οι πρώτες 3 μεταβλητές εισόδου λαμβάνουν τιμές που συνδέονται άμεσα με τις τιμές των εγγραφών στο σύνολο δεδομένων.

```
def density_based_outlier_detection(r, time_window, t, data_points):

    outliers = []

    for i in range(len(data_points)):
        entries = []
        for j in range(len(data_points)):
            if check_r(data_points[i][0], data_points[j][0], r):
                if calculate_time_diff(data_points[i][1], data_points[j][1]) < time_window:
                    entries.append(data_points[j])
                else:
                    pass
            else:
                pass

        if len(entries) < t:
            outliers.append([i, data_points[i]])
        else:
            pass

    return outliers
```

Η υλοποίηση ακολουθεί την εξής λογική. Για κάθε σημείο του συνόλου δεδομένων ελέγχουμε για όλα τα υπόλοιπα εάν α) Η τιμή τους βρίσκεται εντός της απόστασης r με χρήση της check_r και β) Εάν επαληθευτεί το (α) ελέγχουμε εάν το σημείο βρίσκεται εντός του χρονικού παραθύρου. Εφόσον ισχύει και το β το σημείο προστίθεται σε έναν πίνακα entries. Αφού γίνει έλεγχος για όλα τα σημεία, μετρώντας

το πλήθος σημείων του πίνακα entries και συγκρίνοντάς τον με την μεταβλητή t μπορεί να γίνει ο χαρακτηρισμός outlier ή κανονικό σημείο. Εάν το πλήθος είναι μικρότερο του t το σημείο προστίθεται σε έναν πίνακα outliers μαζί με τον δείκτη του.
Όταν ολοκληρωθεί η διαδικασία για όλα τα σημεία ο πίνακας outliers επιστρέφεται, περιέχοντας όλα τα σημεία που κρίθηκαν ως ανώμαλα.

Παράδειγμα εκτέλεσης μεθόδου:

```
def main():

    # Setting the start time
    start_time = time.time()

    data = get_data()

    # Change data type for latter use
    for i in range(len(data) - 1):
        for j in range(3):
            data[i + 1][j] = int(data[i + 1][j])

    list_of_data_points = []
    for i in range(len(data)-1):
        list_of_data_points.append([data[i+1][0], TimeObj(data[i+1][1], data[i+1][2])])

    outliers = density_based_outlier_detection(10, 250, 3, list_of_data_points)

    for i in range(len(outliers)):
        print("Index: ", outliers[i][0], "Value: ", outliers[i][1][0], "Time: ",
              outliers[i][1][1].hour, "Minute: ", outliers[i][1][1].minute)

    # Getting the end time
    end_time = time.time()
    print("Time needed for code execution: ", end_time - start_time, " seconds.")
```

Το σύνολο δεδομένων περιέχει τιμές απ’ το μηδέν έως το 100 και έχει συνολικά 100 εγγραφές. Επιλέχθηκαν μετά από δοκιμές οι παρακάτω μεταβλητές εισόδου: r=10, time_window=250, t=3.

Προέκυψαν τα παρακάτω αποτελέσματα:

```
Index:  6 Value:  4 Time:  4 Minute:  64
Index:  19 Value:  3 Time:  14 Minute:  136
Index:  63 Value:  3 Time:  10 Minute:  110
Index:  69 Value:  16 Time:  0 Minute:  5
Index:  95 Value:  6 Time:  0 Minute:  31
Time needed for code execution:  0.003013134002685547  seconds.
```

6.6 Παρατηρήσεις και σύνοψη ενότητας

Έχοντας, λοιπόν, υλοποιήσει τις παραπάνω μεθόδους έχουν προκύψει κάποια ενδιαφέροντα πορίσματα.

Πιο συγκεκριμένα, όσον αφορά στους χρόνους εκτέλεσης επιβεβαιώνονται όσα ειπώθηκαν στις προηγούμενες ενότητες. Δηλαδή, ότι οι μέθοδοι που βασίζονται σε μεθόδους κοντινότερων γειτόνων είναι πιο αρκετά πιο αργές, καθώς σε ένα μικρό σύνολο δεδομένων χρειάστηκαν περίπου 0.011 δευτερόλεπτα, ενώ η πιο γρήγορη που υλοποιήθηκε (η HBOS) χρειάστηκε περίπου 0.001 για το ίδιο σύνολο δεδομένων. Διαφορά περίπου της τάξεως του 10^{-1}. Μπορεί η διαφορά αυτή να φαίνεται αμελητέα για σύνολα δεδομένων με μικρά μεγέθη, αλλά μπορεί να χει μεγάλο αντίκτυπο σε μεγάλα σύνολα.

Επιπλέον, παρατηρήθηκαν θέματα κλιμάκωσης στις μεθόδους. Για παράδειγμα όταν οι μέθοδοι βασισμένες σε k-NN κλήθηκαν να εντοπίσουν outliers σε ένα σύνολο δεδομένων με μόλις 10.000 εγγραφές χρειάστηκαν 173 δευτερόλεπτα. Η αύξηση αυτή είναι τρομακτικά μεγάλη εάν αναλογιστεί κανείς, ότι σύγχρονα συστήματα μπορεί να κληθούν να χειριστούν εκατομμύρια εγγραφές σε σύντομα χρονικά διαστήματα. Βέβαια, τα θέματα κλιμάκωσης σχετίζονται άμεσα και με την προσέγγιση της υλοποίησης. Στις παραπάνω υλοποίησης η προσέγγιση ήταν προσανατολισμένη σε μεθόδους ωμής – βίας που κατά βάση δεν έχουν και την καλύτερη συμπεριφορά σε θέματα κλιμάκωσης.

Η αδυναμία αυτή είναι που οδήγησε στην δημιουργία εργαλείων όπως το Apache Pinot, που θα αναλυθεί στην συνέχεια. Τα εργαλεία αυτά είναι φτιαγμένα με τέτοιο τρόπο, όπου ο φόρτος εργασίας διαμοιράζεται σε πολλούς πόρους πετυχαίνοντας έτσι καλύτερους χρόνους εκτέλεσης, καθιστώντας τα βιώσιμα σε πραγματικές συνθήκες

7. Αξιοποίηση τεχνικών εντοπισμού σε πραγματικές εφαρμογές

Την σημερινή εποχή υπάρχουν πολλά εργαλεία και εφαρμογές τα οποία προσφέρουν την δυνατότητα της ανάλυσης δεδομένων στους χρήστες τους. Ωστόσο, κάποια εξ' αυτών έχουν ορισμένες ιδιαίτερες δυνατότητες, που δίνουν την ευκαιρία υλοποίησης εξειδικευμένης ανάλυσης (π.χ. anomaly detection). Στο πλαίσιο της πτυχιακής εργασίας θα αναλυθεί το "Apache Pinot", ένα εργαλείο ανεπτυγμένο απ' το open source software της Apache και το "Third Eye", ένα συμπληρωματικό εργαλείο του πρώτου που παρέχει την δυνατότητα αναζήτησης ανωμαλιών.

7.1 Εισαγωγή στο Apache Pinot

- **Τι είναι το Apache Pinot ;**

Το Apache Pinot αποτελεί ένα κατανεμημένο σύστημα πραγματικού χρόνου το οποίο έχει την δυνατότητα να πραγματοποιεί OLAP (Online Analytical Processing) ερωτήματα σε αποθήκες δεδομένων. Πιο συγκεκριμένα, είναι ένα εργαλείο, το οποίο έχει την δυνατότητα επεξεργασίας ροών δεδομένων σε πραγματικό χρόνο. Επιπλέον, το γεγονός, ότι αποτελεί κατανεμημένο σύστημα δίνει την δυνατότητα εκτέλεσης των ερωτημάτων σε πολύ χαμηλό χρόνο.

- **Πώς δουλεύει το Apache Pinot;**

Το Pinot είναι κατασκευασμένο, ώστε να δουλεύει σε UNIX based συστήματα, όπως είναι το Linux και το MacOS. Χρησιμοποιεί μία σειρά από άλλα εργαλεία της Apache προκειμένου να υποστηρίξει τις λειτουργίες του. Ειδικότερα χρησιμοποιεί:

1. Apache Kafka: Εργαλείο, το οποίο χρησιμοποιείται για τροφοδότηση δεδομένων σε streaming περιβάλλον
2. Apache Zookeeper: Εργαλείο, το οποίο χρησιμοποιείται για συντονισμό cloud εφαρμογών.

Με την χρήση αυτών δημιουργεί ένα τοπικό server στον οποίο μπορεί να γίνει οπτικοποίηση του σχήματος μίας αποθήκης δεδομένων, να υλοποιηθούν ερωτήματα και άλλες λειτουργίες που θα αναφερθούν

στην συνέχεια. Επιπρόσθετα, μπορεί να στηθεί σε κάποια υπηρεσία cloud server για να παρέχει συνεχόμενα τις δυνατότητές του στις εφαρμογές που χρησιμοποιείται.

- **Τι δυνατότητες δίνει το Pinot στον χρήστη ;**

Το εργαλείο αυτό παρέχει μία μεγάλη ποικιλία εφαρμογών στους χρήστες του. Αναλυτικότερα:

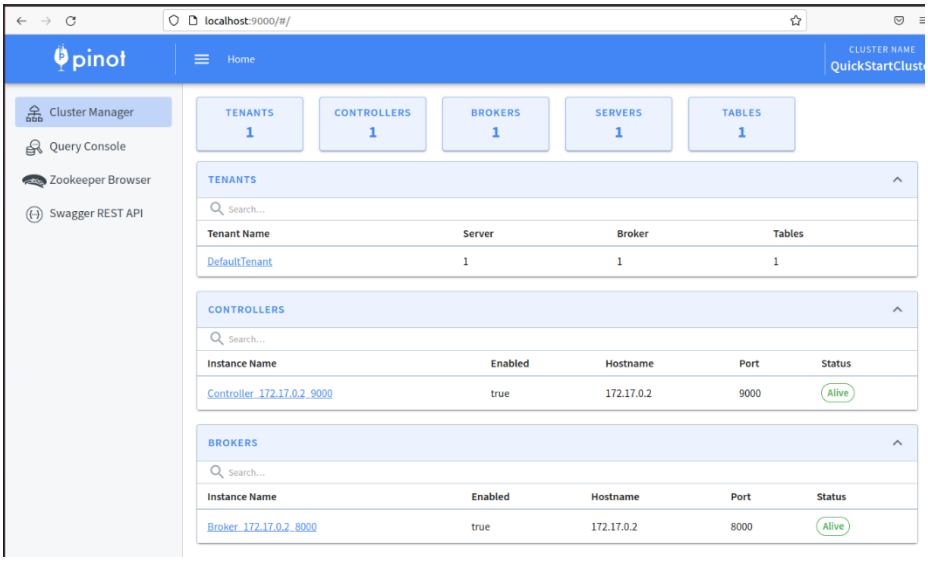
Business Intelligence: Μπορεί να χρησιμοποιηθεί για εφαρμογές στην επιχειρηματική ευφυΐα, καθώς μέσω ανάλυσης μπορεί να παρέχει χρήσιμες συμβουλές όσον αφορά επαγγελματικές αποφάσεις. Μ’ αυτόν τον τρόπο συνεισφέρει στην ανάπτυξη των επιχειρήσεων.

Εποπτεία τάσεων πελατών: Μπορεί να χρησιμοποιηθεί για μελέτη των τάσεων του αγοραστικού κοινού σε τομείς που η ανάλυση πραγματικού χρόνου είναι καίριας σημασίας (για παράδειγμα υπηρεσίας delivery, taxi κ.α.). Μέσω αυτού μπορούν να προβλεφθούν αυξομειώσεις στην ζήτηση των υπηρεσιών δίνοντας χρόνο κατάλληλης προετοιμασίας.

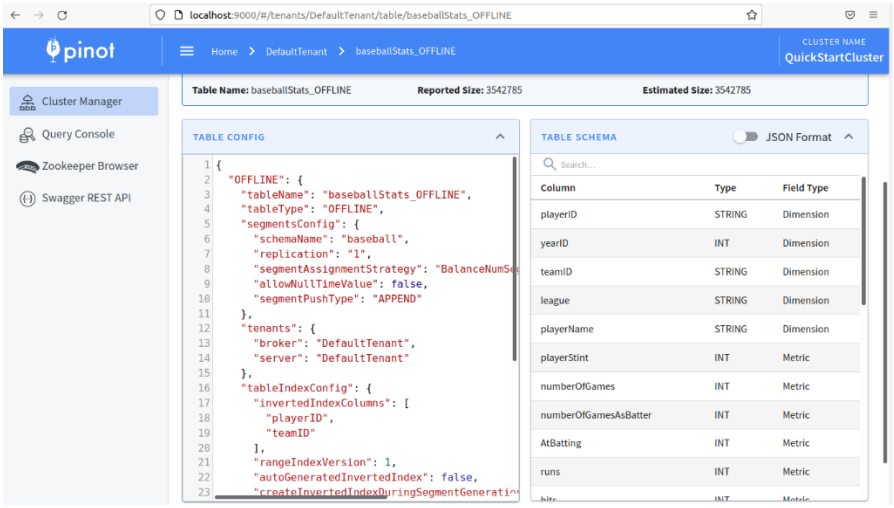
Anomaly detection – Third Eye : Η εφαρμογή που θα απασχολήσει κατά κύριο λόγο την συνέχεια του κειμένου είναι ο εντοπισμός ανωμαλιών σε δεδομένα. Καθώς το Pinot μπορεί να δουλέψει με real time streaming δεδομένα μπορούν να ενσωματωθούν σ’ αυτό οι τεχνικές που αναλύθηκαν στα προηγούμενα τμήματα του κειμένου. Έτσι, δίνεται η δυνατότητα στους χρήστες να γνωρίζουν ανά πάσα στιγμή εάν παρατηρείται κάποια ανωμαλία στα δεδομένα τους, να την αξιολογούν και να λαμβάνουν ενδεχομένως δράση για την εύρεση της αιτίας εμφάνισής της ή να λάβουν πληροφορία μέσω αυτής.

Περιβάλλον του Apache Pinot

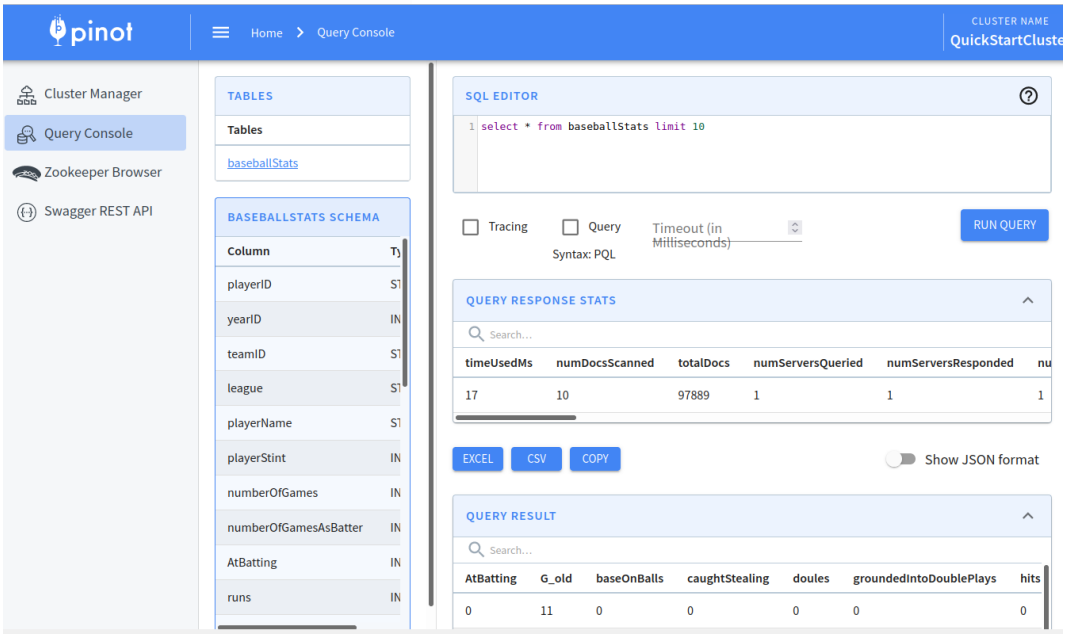
Παρακάτω παρατίθεται μία όψη του περιβάλλοντος εργασίας με σκοπό την κατανόηση του τρόπου λειτουργίας. Στο παράδειγμα, χρησιμοποιείται ένα από τα έτοιμα παραδείγματα του Pinot.



Στην παραπάνω φωτογραφία φαίνεται το βασικό dashboard του Pinot και περιφερειακά οι επιλογές πλοήγησης.



Στην παραπάνω φωτογραφία φαίνεται ο χώρος προεπισκόπησης των περιεχομένων του σχήματος (στο παράδειγμα – η δομή ενός πίνακα με δεδομένα παικτών του baseball).



Στην παραπάνω φωτογραφία φαίνεται ο χώρος υποβολής ερωτημάτων στα περιεχόμενα του σχήματος. Το Pinot παρέχει μία μεγάλη ποικιλία έτοιμων παραδειγμάτων που ως σκοπό έχουν την εκπαίδευση και την εξοικείωση του χρήστη με το περιβάλλον. Περισσότερες πληροφορίες για την χρήση του μπορούν να βρεθούν στην ιστοσελίδα του Pinot, που περιέχονται αναλυτικές οδηγίες χρήσης ή στο GitHub repository.

7.2 Εισαγωγή στο Third Eye

Όπως αναφέρθηκε και προηγουμένως το Third Eye αποτελεί ένα εργαλείο, το οποίο χρησιμοποιείται για την διαδικασία του εντοπισμού ανωμαλιών σε δεδομένα. Πιο συγκεκριμένα, είναι υλοποιημένο με τέτοιο τρόπο, ώστε να μπορεί να διαχειρίζεται χρονοσειρές δεδομένων πραγματικού χρόνου και να τις αναλύει με σκοπό να εντοπίσει ανωμαλίες όπως αυτές που αναλύθηκαν στην ενότητα 5.

Πηγές δεδομένων

Το Third Eye μπορεί να τροφοδοτηθεί με δεδομένα από διάφορες πηγές, όπως:

- Presto – Κατανεμημένο εργαλείο SQL ερωτημάτων για μεγάλες βάσεις δεδομένων.
- MySQL server
- Pinot

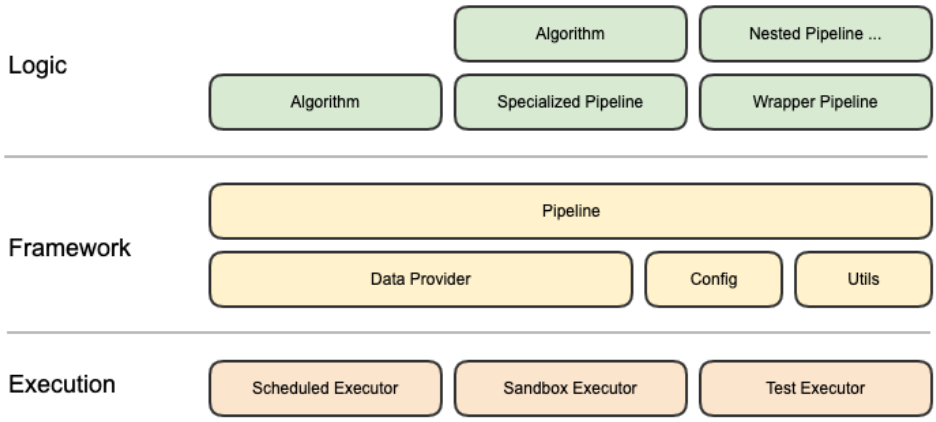
Η δυνατότητα λήψης δεδομένων απ’ αυτές τις πηγές (που χρησιμοποιούνται κατά κόρον) το καθιστά ιδανική επιλογή για πολλές εφαρμογές.

Τρόπος λειτουργίας

Ο τρόπος λειτουργίας του ακολουθεί την εξής προσέγγιση. Αφού δοθεί στο Third Eye το σύνολο δεδομένων προς επιτήρηση/αναζήτηση, ο χρήστης πρέπει να θέσει κάποια alerts/ειδοποιήσεις. Τα alerts αυτά αφορούν σε μετρικές που εξάγονται μέσα από την ανάλυση του συνόλου δεδομένων (οι μετρικές αυτές μπορεί να είναι κάποια εξ’ αυτών που μελετήθηκαν στις προηγούμενες ενότητες του κειμένου). Τα alerts θα ενεργοποιούνται όταν παραβιαστούν οι τιμές των μετρικών με βάση τις τιμές που έχει θέση ο χρήστης ότι δεν πρέπει να ξεπεραστούν. Τότε το σύστημα θεωρεί ότι έχει βρεθεί κάποια εγγραφή στο σύνολο, που θεωρείται ανώμαλη και χρήζει προσοχής.

Αρχιτεκτονική του συστήματος

Το Third Eye ακολουθεί μία πολλή συγκεκριμένη αρχιτεκτονική που φαίνεται στην παρακάτω εικόνα:



(Εικόνα απ' το documentation της εφαρμογής)

Το σύστημα χωρίζεται σε 3 επίπεδα: Επίπεδο Λογικής, Επίπεδο Δομής και Επίπεδο Εκτέλεσης.

Επίπεδο Λογικής: Στο επίπεδο αυτό περιλαμβάνονται όλα τα στοιχεία της λογικής του εργαλείου. Αναλυτικότερα σ' αυτό ενσωματώνονται οι αλγόριθμοι εντοπισμού των ανωμαλιών και ό,τι αυτοί χρησιμοποιούν (π.χ. αλγόριθμοι προ – επεξεργασίας δεδομένων), καθώς και τα pipelines του συστήματος. Τον όρο pipelines θα μπορούσαμε να τον παρομοιάσουμε με μία γραμμή παραγωγής σε ένα εργοστάσιο. Όπως σε μία γραμμή παραγωγής έχει τα στάδιά της που συγκεντρώνει προϊόντα κ.τ.λ. , έτσι κι εδώ τα pipelines είναι ο τρόπος σύνθεσης όλων των τμημάτων της λογικής προς εκτέλεση.

Επίπεδο Δομής: Στο επίπεδο αυτό περιλαμβάνονται στοιχεία που αφορούν σε πιο διαδικαστικά χαρακτηριστικά της εφαρμογής, όπως η παροχή των δεδομένων, η ενσωμάτωση εξωτερικών εργαλείων (π.χ. βιβλιοθήκες και πακέτα κάποιας γλώσσας) και τα δεδομένα διαμόρφωσης (π.χ. διαμόρφωση αρχείων με αποτελέσματα). Και στο επίπεδο αυτό υπάρχει κάποιο pipeline για να υποστηρίξει την σύνθεση.

Επίπεδο Εκτέλεσης: Στο επίπεδο αυτό οργανώνονται πραγματοποιείται ο χρονοπρογραμματισμός διεργασιών του συστήματος καθώς και η εκτέλεση τους.

7.3 Σύνδεση τεχνικών εντοπισμού – αξιοποιημένες τεχνικές στο Third Eye

Συνεχίζοντας, ιδιαίτερο ενδιαφέρον έχει η σύνδεση των τεχνικών που παρουσιάστηκαν στις προηγούμενες ενότητες με τις τεχνικές που χρησιμοποιεί το Third Eye (στο back – end του), ώστε να εντοπίσει τα outliers και να εξυπηρετήσει τον χρήστη. Για να διαπιστωθούν τα παραπάνω χρειάζεται βαθύτερη αναζήτηση στον πηγαίο κώδικά του.

Σημείωση: Το Third Eye εφαρμόζει τις τεχνικές σε χρονοσειρές δεδομένων. Γι’ αυτόν τον λόγο οι τεχνικές θα βρίσκουν ταύτιση με αυτές που αναφέρθηκαν στην ενότητα 5, αλλά όπως παρατηρήσαμε ήδη αυτές έχουν πολλά κοινά σημεία από άποψη προσέγγισης μ’ αυτές τις ενότητας 4.

Αρχικές παρατηρήσεις

Με μία πρώτη ματιά παρατηρείται, ότι δίνεται μεγάλη σημασία στο στάδιο της προ – επεξεργασίας δεδομένων, το οποίο κατέχει σημαντικό ρόλο για την απόκτηση ποιοτικών και ορθών πορισμάτων. Πιο συγκεκριμένα, το Third Eye υλοποιεί, πριν την έναρξη της ανάλυσης μία σειρά από «φιλτραρίσματα» στα δεδομένα. Υλοποιεί αρχικά διαδικασίες μείωσης των διαστάσεων, είτε μέσω πλήρους αγνόησης χαρακτηριστικών που δεν λαμβάνονται υπόψιν στην διαδικασία της αναζήτησης ανωμαλιών είτε μέσω άλλων τεχνικών που αναφέρθηκαν προηγουμένως (π.χ. PCA).

Χρησιμοποιούμενες κατηγορίες μεθόδων:

Η πρώτη κατηγορία μεθόδων που διακρίνεται αφορά στις model based μεθόδους τόσο για prediction, όσο και για estimation. Όπως διακρίνεται github repository (Ιστοσελίδα 4), το Third Eye λαμβάνει ένα training set απ’ την πηγή δεδομένων και το χρησιμοποιεί προκειμένου να εκπαιδεύσει ένα μοντέλο για αναγνώριση των outliers στα δεδομένα που θα κληθεί να μελετήσει.

Στην συνέχεια, με βάση το μοντέλο αυτό δίνεται η δυνατότητα εύρεσης των ανωμαλιών στα νέα δεδομένα. Η κατηγορία αυτή εφαρμόζεται τόσο σε point όσο και σε subsequence outliers. Η αξιοποίηση της συγκεκριμένης κατηγορίας δεν προκαλεί έκπληξη, καθώς οι εφαρμογές της τεχνητής νοημοσύνης και της επιβλεπόμενης μάθησης είναι ιδιαίτερα διαδεδομένες σε σύγχρονες εφαρμογές.

Η δεύτερη κατηγορία τεχνικών που εντοπίζεται και βρίσκει εφαρμογή στο Third Eye είναι αυτή η “Discord Based” στα subsqecuence outliers. Αναλυτικότερα το Third Eye δίνει την δυνατότητα ορισμού του μήκους του ολισθαίνοντος παραθύρου που θα χρησιμοποιηθεί για την κατασκευή του συνόλου των subsequences. Στην συνέχεια, υλοποιεί την διαδικασία ελέγχου των υποσυνόλων μεταξύ τους με σκοπό την εύρεση ασυμφωνιών μεταξύ τους. Εάν βρεθεί κάποια ασυμφωνία, τότε ειδοποιείται και καλείται να αξιολογήσει τα ευρήματα.

Η τρίτη κατηγορία τεχνικών αφορά στην “Histogram Based”, που χρησιμοποιεί στοιχεία απ’ την επιστήμη της στατιστικής και αφορά στον εντοπισμό των point outliers. Μελετώντας την υλοποίηση την υλοποίηση που υιοθετεί το Third Eye παρατηρείται ότι δίνεται η δυνατότητα ορισμού του μεγέθους των «κάδων» των ιστογραμμάτων και δεν δίνεται η επιλογή της δυναμικής εκχώρησης του μεγέθους τους. Οι μέθοδοι που ανήκουν στην κατηγορία αυτή είναι ιδιαίτερα γρήγορες, αποτελεσματικές και η παρουσία τους είναι ιδιαίτερα χρήσιμη για την παροχή υπηρεσιών στους χρήστες.

7.4 Προτάσεις βελτίωσης συστήματος

Στην ενότητα αυτή θα παρατεθούν προτάσεις και ιδέες που αναπτύχθηκαν έπειτα από εκτεταμένη μελέτη του αντικειμένου. Οι προτάσεις αυτές είναι ανεπτυγμένες με στόχο την βελτίωση τόσο των αποτελεσμάτων των αναλύσεων, καθώς και της εμπειρίας του χρήστη μέσω της διάδρασής του με το Third Eye.

Πρόταση 1η : Το Third Eye αυτή την στιγμή αξιοποιεί τεχνικές, οι οποίες αφορούν σε εντοπισμό point και subsequence outliers. Δεν υπάρχουν όμως ακόμη αξιοποιημένες τεχνικές για τον εντοπισμό time series outliers. Αν και η κατηγορία αυτή είναι η πιο σπάνια εξ' αυτών που αναλύθηκαν στις τεχνικές εντοπισμού σε χρονοσειρές και η εμφάνισή τους οφείλεται συνήθως λάθη ή αστοχίες υλικών (π.χ. ελλαττωματικός αισθητήρας) ο εντοπισμός και η αξιολόγησή τους είναι εξίσου σημαντική και πρέπει να είναι διαθέσιμη.

Πρόταση 2η : Ενσωμάτωση περισσότερων μεθόδων εντοπισμού που ανήκουν και σε άλλες κατηγορίες. Οι αξιοποιήσιμες μέθοδοι αυτήν την στιγμή ανήκουν στις discord based, model based και histogram based. Αν και αυτές οι κατηγορίες είναι ιδιαίτερα χρήσιμες και παράγουν αξιόπιστα αποτελέσματα θα έπρεπε να υπάρχουν περισσότερες επιλογές για τον χρήστη. Ένα χαρακτηριστικό παράδειγμα αποτελεί η 5η κατηγορία (Information Theory Based) στα subsequence outliers, η οποία πέρα από την συχνότητα των φαινομένων εισήγαγε και την έννοια της πληροφορίας. Όταν το εργαλείο δίνεται για χρήση, αυτό πρόκειται να χρησιμοποιηθεί για εφαρμογές πραγματικού κόσμου, οπότε η σημασιολογία των δεδομένων και η πληροφορία που κρύβουν είναι σημαντικός παράγοντας που πρέπει να λαμβάνεται υπόψιν κατά την ανάλυση.

Πρόταση 3η : Υλοποίηση μεθόδων που βασίζονται στην συσταδοποίηση δεδομένων. Όπως προαναφέρθηκε στην ενότητα 4, οι μέθοδοι που προσεγγίζουν τον εντοπισμό outliers με βάση την συσταδοποίηση αποτελούν ένα καλά ανεπτυγμένο πεδίο που μπορεί να εφαρμοστεί με καλά αποτελέσματα.

Συνεπώς, η υλοποίηση μεθόδων που ανήκουν στην κατηγορία αυτή είναι καίριας σημασίας. Επιπλέον, ως μελλοντική επέκταση της πρότασης αυτής θα μπορούσε να προσφέρεται ως επιλογή να καθορίζει ο χρήστης τον αλγόριθμο συσταδοποίησης που χρησιμοποιεί το Third Eye. Μπορεί ο k – means, που είναι ο πιο συνηθισμένος να χρησιμοποιείται, γιατί προσφέρει ικανοποιητικούς χρόνους εκτέλεσης, αλλά δεν παύει να αποτελεί αλγόριθμος που είναι αρκετά αδύναμος σε ορισμένους τύπους συστάδων (π.χ. να μην ακολουθούν σφαιρική/καμπυλωτή κατανομή όπως συμβαίνει σε δεδομένα με γραμμική συσχέτιση, να έχουν πολλές διαστάσεις τα δεδομένα κ.τ.λ.)

Πρόταση 4η : Μία τελευταία πρόταση που δεν έχει τόσο να κάνει με το πως το εργαλείο χειρίζεται την διαδικασία της ανάλυσης, αλλά με το εργαλείο καθαυτό είναι η μελλοντική προσφορά του σαν ένα πιο ολοκληρωμένο σύστημα. Αναλυτικότερα, το Pinot και το Third Eye προσφέρονται σε UNIX συστήματα και προκειμένου ο χρήστης να τα χρησιμοποιήσει πρέπει να έχει ένα σχετικά καλό υπόβαθρο χειρισμού τέτοιων συστημάτων. Ο μέσος χρήστης κατά πάσα πιθανότητα δεν έχει τέτοια εμπειρία και η χρήση του καθίσταται δύσκολη. Οπότε για να είναι πιο ανοικτό, προσβάσιμο και αξιοποιήσιμο θα ήταν ορθό να αναπτυχθεί κάποιο desktop application ή web application για να είναι πιο εύκολο στον μέσο χρήστη (π.χ. υπάλληλος σε τμήμα marketing μίας επιχείρησης.

8.Τελικά συμπεράσματα και αποφώνηση

Το πεδίο της αναγνώρισης και αξιολόγησης ανωμαλιών στα δεδομένα αποτελεί ένα ιδιαίτερο και ενδιαφέρον αντικείμενο μελέτης, το οποίο έχει μεγάλα περιθώρια ανάπτυξης μπροστά του. Προς το παρόν κατέχει καλά μελετημένες μεθόδους εφαρμογής, οι οποίες μπορούν να συνεισφέρουν σημαντικά σε εφαρμογές πραγματικού κόσμου.

Οι μέθοδοι που έχουν αναπτυχθεί καλύπτουν μία ευρεία γκάμα περιπτώσεων ανάλυσης που μπορεί να φανούν χρήσιμες σε αναλυτές δεδομένων. Αυτές βασίζονται σε μεγάλο βαθμό σε ήδη γνωστούς αλγορίθμους και έννοιες και μπορούν να αντιμετωπίσουν στατικά και χρονικά μεταβαλλόμενα δεδομένα εξίσου καλά παρέχοντας χρήσιμες πληροφορίες.

Επιπλέον, ήδη υπάρχουν συστήματα που ενσωματώνουν τα πορίσματα του πεδίου σε εφαρμογές αξιοποιήσιμες απ’ το κοινό. Τα συστήματα αυτά βέβαια βρίσκονται σε πρώιμο στάδιο και έχουν πολλά περιθώρια βελτίωσης όπως αναφέρθηκε στην ενότητα 6. Μελλοντικά θα μπορέσουν να προσφέρουν ακόμα πιο ισχυρά εργαλεία βελτιώνοντας τον τρόπο λειτουργίας πολλών τομέων της καθημερινότητας όπως την ασφάλεια, την υγεία και την επιχειρηματικότητα.

Εν κατακλείδι, το πεδίο αυτό συνδυάζει στοιχεία που πηγάζουν απ’ την επιστήμη της πληροφορικής και των μαθηματικών και χρήζει περαιτέρω μελέτης απ’ τους ειδικούς των αντικειμένων αυτών.

9. Πηγές – άρθρα και παραπομπές

Παρακάτω θα βρείτε αναλυτικά τα όλες τις πηγές, άρθρα απ' τα οποία αντλήθηκαν πληροφορίες για την εκπόνηση της πτυχιακής εργασίας.

Άρθρα:

1. A comparative evaluation of Unsupervised Anomaly Detection Algorithms for multivariate data - Markus Goldstein, Seiichi Uchida https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0152173
2. Outlier detection: Methods, Models and Classification -Azzedine Boukerche, Lining Zheng, Omar Alfandi https://dl.acm.org/doi/abs/10.1145/3381028
3. An improved Robust Principal Component Analysis Model for anomalies detection of subway passenger flow - Xuehui Wang, Yong Zhang, Hao Liu , Yang Wang , Lichun Wang , Baocai Yin https://www.hindawi.com/journals/jat/2018/7191549/
4. Improving Influenced Outlierness Outlier Detection Method – Suman, Shaswat http://ethesis.nitrkl.ac.in/5130/
5. Histogram Based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm – Markus Goldstein, Andreas Dengel (PDF) Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm (researchgate.net)
6. Support Vector Machine – Wikipedia https://en.wikipedia.org/wiki/Support-vector_machine
7. A review on Outlier/Anomaly Detection in Time series data -Ane Blázquez-García, Angel Conde, Usue Mori, Jose A. Lozano https://dl.acm.org/doi/abs/10.1145/3444690
8. Efficient Anomaly Detection by Isolation Using Nearest Neighrour Ensemble - Tharindu Bandaragoda, Kai Ming Ting, David W. Albrechet, Fei Tony Liu (PDF) Efficient Anomaly Detection by Isolation Using Nearest Neighbour Ensemble (researchgate.net)

9. From Cluster-Based Outlier Detection to Time series Discord Discovery - Nguyen Huy Kha , Duong Tuan Anh
   https://www.researchgate.net/publication/315544676_From_Cluster-Based_Outlier_Detection_to_Time_Series_Discord_Discovery

Ιστοσελίδες εργαλείων:

1. Apache - https://www.apache.org/
2. Apache Pinot - https://pinot.apache.org/
3. Apache Pinot GitHub - https://github.com/apache/pinot
4. Third Eye – GitHub - https://github.com/project-thirdeye/thirdeye
5. Third Eye – documentation - https://thirdeye.readthedocs.io/en/latest/
6. Apache Zookeper - https://zookeeper.apache.org/
7. Apache Kafka - https://kafka.apache.org/

Βιβλία:

Επιπλέον, χρησιμοποιήθηκε το παρακάτω βιβλίο για μελέτη και εξοικείωση με τις έννοιες.

An Introduction to Outlier Analysis – Charu C. Aggarwal

https://link.springer.com/chapter/10.1007/978-3-319-47578-3_1

Εικόνα εξωφύλλου:

Άρθρο της ιστοσελίδας An Introduction to Anomaly Detection and its importance in Machine Learning

https://thedatascientist.com/anomaly-detection-why-you-need-it/