



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ
ΒΙΟΙΑΤΡΙΚΗ»**

**Αυτοματοποιημένη Διαχείριση και Ενημέρωση
Συνόψεων σε Κατανεμημένα Αποθετήρια Δεδομένων**

ΘΕΟΧΑΡΗΣ ΚΟΥΚΑΡΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Επιβλέπων Καθηγητής: κ. Σταμούλης Γεώργιος
Επιστημονικός Σύμβουλος: κ. Κολομβάτσος Κωνσταντίνος**

Λαμία, 2018



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ
ΚΑΤΕΥΘΥΝΣΗ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**«ΠΛΗΡΟΦΟΡΙΚΗ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΣΦΑΛΕΙΑ,
ΔΙΑΧΕΙΡΙΣΗ ΜΕΓΑΛΟΥ ΟΓΚΟΥ ΔΕΔΟΜΕΝΩΝ ΚΑΙ
ΠΡΟΣΟΜΟΙΩΣΗ»**

**Αυτοματοποιημένη Διαχείριση και Ενημέρωση Συνόψεων
σε Κατανεμημένα Αποθετήρια Δεδομένων**

ΘΕΟΧΑΡΗΣ ΚΟΥΚΑΡΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Επιβλέπων Καθηγητής: κ. Σταμούλης Γεώργιος
Επιστημονικός Σύμβουλος: κ. Κολομβάτσος Κωνσταντίνος**

Λαμία, 2018

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο [«τίτλος εργασίας»] αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο ΔΗΛΩΝ

Ημερομηνία

Υπογραφή

Αυτοματοποιημένη Διαχείριση και Ενημέρωση Συνόψεων σε
Κατανεμημένα Αποθετήρια Δεδομένων

ΘΕΟΧΑΡΗΣ ΚΟΥΚΑΡΑΣ

Επιβλέπων Καθηγητής

Καθηγητής Γεώργιος Σταμούλης

Επιστημονικός Σύμβουλος:

Δρ. Κωνσταντίνος Κολομβάτσος

Περίληψη της εργασίας

Η παρούσα διπλωματική εργασία εκπονείται στο πλαίσιο του μεταπτυχιακού Προγράμματος Σπουδών "Πληροφορικής και Υπολογιστικής Βιοιατρικής" με κατεύθυνση την Πληροφορική και έχει την εξής δομή:

Στο πρώτο κεφάλαιο γίνεται μια σύντομη εισαγωγή και πως περνάμε από τα δεδομένα στην γνώση. Στο δεύτερο κεφάλαιο εξηγείται τι είναι εξόρυξη δεδομένων (Data Mining) και ποιες είναι οι τεχνικές της. Στο τρίτο κεφάλαιο αναλύονται οι αλγόριθμοι της κάθε τεχνικής, τα πλεονεκτήματα και τα μειονεκτήματα τους. Στο τέταρτο κεφάλαιο εξηγείται τι είναι σύνοψη δεδομένων και αναλύουμε την τεχνική και τον αλγόριθμο που θα χρησιμοποιήσουμε. Στο πέμπτο κεφάλαιο, της πειραματικής αποτίμησης, εξηγούνται τα λογισμικά που χρησιμοποιήθηκαν και το γενικότερο στήσιμο προκειμένου να προχωρήσουμε στην υλοποίηση του αλγορίθμου, καθώς και οι γραφικές παραστάσεις ώστε να αναδειχθεί η τεχνική.

Τέλος στο παράρτημα αναλύεται όλος ο κώδικας.

Σκοπός της παρούσας διπλωματικής είναι η δημιουργία κώδικα με τον αλγόριθμο των micro-clusters, ο οποίος θα τρέχει και θα βγάζει τις συστάδες για κάθε πίνακα.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κύριο Κωνσταντίνο Κολομβάτσο, Διδάσκων στο Μεταπτυχιακό Πρόγραμμα σπουδών, που μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, για την άμεση απόκριση του στα ερωτήματά μου, αλλά και για το χρόνο που αφιέρωσε για τη διεκπεραίωση της διπλωματικής μου εργασίας.

Πίνακας Περιεχομένων

Εικόνες	8
Πίνακες	8
1 Κεφάλαιο 1	
1.1 Εισαγωγή	10
1.2 Τι είναι Big Data “Μεγάλα Δεδομένα”	10
1.3 Βασικά χαρακτηριστικά Big Data	11
1.4 Παράγοντες κλειδί στα Big Data	12
1.5 Από τα δεδομένα στη γνώση	12
1.6 Data Streams “Ροές Δεδομένων”	14
1.6.1 Εφαρμογές Ροών Δεδομένων	14
2 Κεφάλαιο 2	
2.1 Εξόρυξη δεδομένων	16
2.2 Στόχοι	17
2.3 Μέθοδοι εξόρυξης δεδομένων	17
2.3.1 Ταξινόμηση - Classification	17
2.3.2 Συσχέτιση – Association	17
2.3.3 Συσταδοποίηση – Clustering	18
2.4 Εφαρμογές και συνεισφορά των Big Data	19
3 Κεφάλαιο 3	
3.1 Αλγόριθμοι εξόρυξης δεδομένων	23
3.2 Αλγόριθμοι Συσταδοποίησης	23
3.2.1 Ιεραρχική - συσταδοποίηση	24
3.2.2 Διαμεριστικοί αλγόριθμοι συσταδοποίησης	25
3.3 Αλγόριθμοι Κατηγοροποίησης	27
3.4 Αλγόριθμοι συσχέτισης	31
3.5 Πλεονεκτήματα και μειονεκτήματα αλγορίθμων	33
4 Κεφάλαιο 4	
4.1 Δημιουργία και Διατήρηση Συνόψεων	37

4.2	Επιλογή μεθόδου και αλγορίθμου	38
4.3	Δημιουργία συνόψεων και επιλογή μετρικών	39
5	Κεφάλαιο 5 (Πειραματική Αποτίμηση)	
5.1	Εγκατάσταση	42
5.2	Παραμετροποίηση	42
5.3	Δημιουργία Project	43
	5.3.1 Upload και εισαγωγή δεδομένων	43
	5.3.2 Παραγωγή τυχαίων πλειάδων 5 αριθμών	44
	5.3.3 Υπολογισμός μετρικών	45
5.4	Γραφικές παραστάσεις	45
	Παράρτημα 1(Κώδικας)	70
	Βιβλιογραφία	112
	<u>Εικόνες</u>	
	Εικόνα 1: Η πορεία προς την εξαγωγή γνώσης	13
	Εικόνα 2.1: Ανακάλυψη προτύπων σε δεδομένα	16
	Εικόνα 2.2: Market basket analysis	18
	Εικόνα 3: Διαδικασία εξόρυξης πληροφορίας	23
	Εικόνα 4 Σημεία σε 3 clusters	24
	Εικόνα 5: Δενδροδιάγραμμα ιεραρχικού αλγορίθμου	24
	Εικόνα 6: Σημεία DBScan	27
	Εικόνα 7: Εφαρμογή του αλγορίθμου C4.5 σε δένδρο απόφασης	29
	Εικόνα 8: Αλγόριθμος Apriori	32
	Εικόνα 9: Τύπος JACCARD	40
	Εικόνα 10: Τύπος Sorensen	40
	Εικόνα 11: Τύπος Kulczynski	41
	Εικόνα 12: Επιλογή φύλλο εργασίας excel	44
	Εικόνα 13: Παραγωγή τυχαίων αριθμών	44
	Εικόνα 14: Παραγόμενοι τυχαίοι αριθμοί (10000 παραγόμενες πεντάδες)	45
	<u>Πίνακες</u>	
	Πίνακας 1.1 : Μέσος όρος πίνακα AIR_1, 1 ^{ου} σετ πειραμάτων	46
	Πίνακας 1.2 : Μέσος όρος πίνακα AIR_2, 1 ^{ου} σετ πειραμάτων	46
	Πίνακας 1.3 : Μέσος όρος πίνακα AIR_3, 1 ^{ου} σετ πειραμάτων	47
	Πίνακας 1.4 : Μέσος όρος πίνακα AIR_4, 1 ^{ου} σετ πειραμάτων	48
	Πίνακας 1.5 : Μέσος όρος πίνακα AIR_5, 1 ^{ου} σετ πειραμάτων	48
	Πίνακας 1.6 : Τυπική απόκλιση πίνακα AIR_1, 1 ^{ου} σετ πειραμάτων	49

Κεφάλαιο 1

1.1 Εισαγωγή

Ο αιώνας που διανύουμε έχει χαρακτηριστεί από πολλούς ως αιώνας της πληροφορίας. Σημαντικό εφόδιο για να πρωταγωνιστήσει κάποιος, σε όποιον τομέα και αν δραστηριοποιείται, είναι να έχει την ικανότητα να συγκεντρώνει πληροφορίες και δεδομένα και στη συνέχεια να τα αξιολογεί. Η αξιολόγηση και η αξιοποίηση της πληροφορίας και των δεδομένων είναι τα στοιχεία αυτά που θα επιτρέψουν στον κάθε ενδιαφερόμενο να αποκτήσει ανταγωνιστικό πλεονέκτημα στο χώρο που δραστηριοποιείται και να λάβει τις βέλτιστες αποφάσεις, δεδομένων των εκάστοτε συνθηκών, σε θέματα που τον αφορούν. Τέτοιου είδους αναλύσεις, που λαμβάνουν χώρα σε ποιοτικά αλλά και σε αριθμητικά δεδομένα πραγματοποιούνται, μεταξύ άλλων, με τη βοήθεια της χρήσης των τεχνικών Εξόρυξης Δεδομένων (Data Mining), οι οποίες δίνουν την δυνατότητα εξαγωγής σχέσεων και κανόνων μέσω της χρήσης ηλεκτρονικών υπολογιστών [1].

1.2 Τι είναι Big Data - “Μεγάλα Δεδομένα”

Αρχικά θα αναφερθούμε στον όρο Big Data που είναι μια αόριστη έννοια για την οποία υπάρχει μεγάλος διάλογος, συμφωνίες και διαφωνίες πάνω στον ορισμό της από ακαδημαϊκούς και επαγγελματίες της πληροφορικής.

Σε γενικές γραμμές μπορούμε να ορίσουμε ως “Μεγάλα Δεδομένα” που περιγράφει τα μεγάλα συνόλα δεδομένων που δεν μπορούσαν να ληφθούν, να αποθηκευτούν, να διαχειριστούν ούτε να αναλυθούν χρησιμοποιώντας παραδοσιακές βάσεις δεδομένων. Ωστόσο, ο ορισμός έχει διευρυνθεί σημαντικά τα τελευταία χρόνια. Τα μεγάλα δεδομένα δεν αφορούν μόνο τα ίδια τα δεδομένα αλλά και το σύνολο των τεχνολογιών που εκτελούν όλες τις προαναφερόμενες λειτουργίες, καθώς και ποικίλες συλλογές που επιλύουν περίπλοκα προβλήματα [19] και καθιστούν την αξία ξεκλειδώματος από αυτά τα δεδομένα πιο οικονομικά [20].

Έχει αναπτυχθεί μια σειρά νέων τεχνολογιών γύρω από τη συλλογή, ταξινόμηση, ανάλυση, συνδυασμό και αξιοποίηση πληροφοριών που προκύπτουν από την επεξεργασία των δεδομένων αυτών οι οποίες συγκροτούν την Επιστήμη των Big Data.

Η ουσιαστική αξία των Big Data δεν βρίσκεται στο μέγεθος των δεδομένων αλλά στην συνδεσιμότητά τους με άλλα δεδομένα. Χάρη στις επιστήμες εξόρυξης και ανάλυσης δεδομένων (Data Mining), συσχετισμού και συνάθροισης τα Big Data μετατρέπονται ουσιαστικά σε ένα δίκτυο. Η αξία της προέρχεται από τις σχέσεις και τα μοτίβα που μπορεί να προκύψουν από την πραγματοποίηση των συνδέσεων μεταξύ των δεδομένων, για ένα άτομο, για το άτομο σε σχέση με μια ομάδα, για ομάδες ανθρώπων, ή απλά για τη δομή της ίδιας της πληροφορίας [21].

1.3 Βασικά χαρακτηριστικά των Big Data

Ένας άλλος, πιο ακριβής ορισμός για τον όρο “Μεγάλα Δεδομένα” είναι “δεδομένα τα οποία χαρακτηρίζονται από τα λεγόμενα 3Vs: Volume, velocity, variety” δηλαδή δεδομένα που χαρακτηρίζονται από μεγάλο όγκο, ταχύτητα επεξεργασίας και ποικιλία δεδομένων. Στον διάλογο για τον ορισμό των Big Data τους έχουν αποδοθεί και άλλα χαρακτηριστικά όπως Μεταβλητότητα (Variability), Ακρίβεια (Veracity), Απεικόνιση (Visualization) και Αξία (Value). Στην κυριότητά τους όμως όλοι συμφωνούν στα 3Vs στα οποία θα μείνουμε και περισσότερο [22].

Όγκος : είναι το πόσα δεδομένα που έχουμε. Τεράστιες ποσότητες δεδομένων είναι τώρα σε επεξεργασία με τη βασική μονάδα των «Big Data» να μεταβαίνει από τα Gigabytes και Terabytes σε Exabytes ή ακόμα και Zettabytes [23]. Το Ίντερνετ των πραγμάτων, τα Δίκτυα Κοινωνικής Δικτύωσης, οι online συναλλαγές και η κίνηση του παγκοσμίου δικτύου συνολικά έχει δημιουργήσει εκθετική αύξηση των δεδομένων.

Ταχύτητα: Το δεύτερο σημαντικότερο χαρακτηριστικό των Big Data είναι η ταχύτητα με την οποία αυτά συλλέγονται, επεξεργάζονται και είναι προσβάσιμα σε «πραγματικό χρόνο». Ποτέ στο παρελθόν σύνολα δεδομένων ήταν σε θέση να συλλεχθούν τόσο ευέλικτα και γρήγορα. Αυτό σημαίνει, επίσης, ότι η χρησιμότητα και η αξία των δεδομένων εξελίσσεται σε πραγματικό χρόνο, σύμφωνα με την εισροή νέων δεδομένων.

Ποικιλία: Ένα άλλο χαρακτηριστικό των Big Data είναι η ποικιλία τους και περιγράφει μια από τις μεγαλύτερες προκλήσεις των μεγάλων δεδομένων. Η ποικιλία των δεδομένων μπορεί να έχει πολλές κατηγορίες. Οι βασικότερες είναι:

1) Η ποικιλία των μορφών (κείμενο, εικόνες, ήχος, γεωγραφικά δεδομένα, δεδομένα κινητών συσκευών, κλπ),

2) Δομημένα ή αδόμητα δεδομένα, όπου δομημένα είναι τα δεδομένα των οποίων όλοι οι δυνατές τιμές καθορίζονται και γνωστά εκ των προτέρων, ενώ τα αδόμητα όχι. Π.χ. σε μια βάση δεδομένων e-mail client, ο συγγραφέας και η ημερομηνία είναι δομημένα δεδομένα, αλλά το σώμα του μηνύματος είναι αδόμητο.

3) Από την προέλευσή τους ή το πού παράγονται: Τα Hard data είναι δεδομένα που παράγονται από ιδρύματα και δημόσιες διοικητικές αρχές (μητρώα παντός τύπου, καταγγελίες και δικαστικές αποφάσεις, εκλογικούς καταλόγους, άδειες όλων των ειδών , κ.λπ.) Soft data δεδομένα που παράγονται από φυσικά πρόσωπα, είτε εκ προθέσεως μέσω των blogs, κοινωνικά δίκτυα και φόρουμ συζητήσεων, είτε αθέλητα από τις δραστηριότητές και αλληλεπιδράσεις, τις online και offline κινήσεις τους, ψηφιακά «ίχνη» που συχνά συλλέγονται από προεπιλεγμένους μηχανισμούς.

4) Σε όλα αυτά προστίθενται τα μεταδεδομένα (metadata) τα οποία - με την γενικότερη έννοια του όρου - είναι «δεδομένα για τα δεδομένα», δημιουργούνται αυτόματα από τα ίδια τα συστήματα ηλεκτρονικών υπολογιστών, τα οποία καθιστούν δυνατή την περιγραφή και τη δομή άλλων στοιχείων, ανεξάρτητα από το περιεχόμενό τους [27].

1.4 Παράγοντες Κλειδί στα Big Data

Η τεχνολογία των Big Data έχει τέσσερις βασικούς παράγοντες [24] [25]: Υποδομή, Αποθήκευση δεδομένων, Επεξεργασία/Διαχείριση και Ανάλυση Δεδομένων. Το 2012, οι πέντε πρώτες σε όλο τον κόσμο γεννήτριες σε έσοδα σχετιζόμενα με Big Data ήταν η IBM (USD 1,3 δισ), η HewlettPackard (USD 664m), Teradata (USD 435), Dell (USD 425), και η Oracle (USD 415) [26].

- **Υποδομές:** Το κλειδί για την υποδομή μεγάλων δεδομένων είναι η επεκτασιμότητα και η ευελιξία στο χειρισμό petabytes δεδομένων, έτσι το σύννεφο γίνεται μια φυσική επιλογή. Βασικά οι πάροχοι δημόσιων cloud περιλαμβάνουν: Amazon Web Services, GoGrid και RackSpace,

- **Αποθήκευση δεδομένων:** Τα παραδοσιακά συστήματα και οι μέθοδοι αποθήκευσης είναι μη ιδανικά λόγω των περιορισμών των τιμών και της επεκτασιμότητας. Νέες μέθοδοι αποθήκευσης, ιδιαίτερα οι τεχνολογίες των NoSQL και DFS (Distributed File System) αντιπροσωπεύουν τη παραδειγματική στροφή στην αρένα αποθήκευσης. Μεταξύ αυτών, το Hadoop είναι η πιο συχνά χρησιμοποιούμενη μέθοδος αποθήκευσης και διαχείρισης για Big Data.

Βασικοί παίκτες αποθήκευσης Big Data περιλαμβάνουν: MongoDB, Hadoop, Clustrix και Netezza

- **Επεξεργασία και Διαχείριση Δεδομένων:** Αυτό το πλαίσιο χρησιμοποιείται για την επεξεργασία μια αυξανόμενης μάζας των δεδομένων παράλληλα. Βασικά επεξεργαστές δεδομένων και διαχειριστές περιλαμβάνουν: Cloudera, Hadoop, και Greenplum.

- **Data Analytics:** Αυτή είναι η περιοχή που παρέχει οπτικοποίηση, προγνωστική ανάλυση και τις τεχνολογίες data analytics. Βασικά οι πάροχοι analytics δεδομένων περιλαμβάνουν: Splunk, Clickfox, Rainstor, Pervasive, MapR και Progress dataDirect [27].

1.5 Από τα δεδομένα στην γνώση

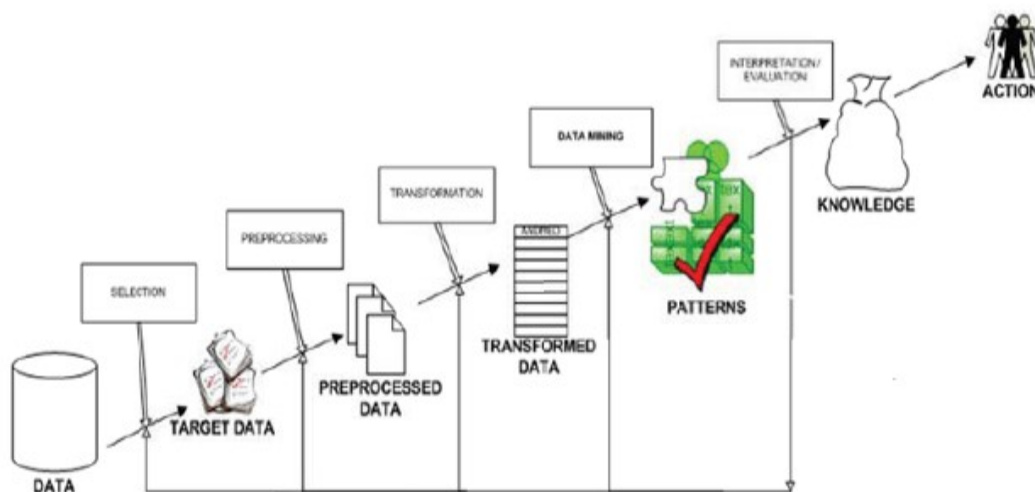
Ο όγκος των αποθηκευμένων δεδομένων παγκοσμίως αυξάνεται συνεχώς. Παντοδύναμοι ηλεκτρονικοί υπολογιστές καθιστούν την αποθήκευση οποιουδήποτε είδους δεδομένων μια πολύ απλή υπόθεση. Δίσκοι με αποθηκευτική ικανότητα πολλών gigabyte συνεισφέρουν στην συνεχιζόμενη αύξηση των αποθηκευμένων δεδομένων. Ο παγκόσμιος ιστός μας κατακλύζει καθημερινά με δεδομένα. Σύμφωνα με το νόμο του Moore, η ταχύτητα των υπολογιστών διπλασιάζεται κάθε 18 μήνες ενώ τα δεδομένα που αποθηκεύονται διπλασιάζονται μόλις κάθε 9 μήνες. Αν υποθέσει κανείς ότι χρόνος που δαπανάται για την επεξεργασία αυτών των δεδομένων παραμένει σταθερός είναι εύκολα αντιληπτό το αυξανόμενο χάσμα μεταξύ της παραγωγής δεδομένων και της επεξεργασίας και εξαγωγής πληροφορίας από αυτά.

Για να εξερευνηθούν επομένως όλες αυτές οι εγγραφές πολλών μεταβλητών ώστε να εξαχθούν χρήσιμα συμπεράσματα από αυτές, απαιτείται μια εκ βαθέων μελέτη που οδηγεί στην ανακάλυψη γνώσης από τα δεδομένα (Knowledge Discovery in Data - KDD), γνώσης που θα δίνει αξία και νόημα στα δεδομένα. Η KDD είναι μια μη τετριμμένη διαδικασία εύρεσης έγκυρων, πρωτότυπων, πιθανώς χρήσιμων και οπωσδήποτε κατανοητών προτύπων μέσα στα δεδομένα [12].

Η διαδικασία ανακάλυψης γνώσης, απεικονίζεται στην εικόνα 1, αποτελείται από την επαναλαμβανόμενη αλληλουχία των ακόλουθων γεγονότων:

1. Ολοκλήρωση δεδομένων (Data Intergration) , όπου συνδυάζονται δεδομένα από διάφορες πηγές.
2. Επιλογή δεδομένων (Data Selection), όπου ανακτούνται όσα δεδομένα είναι σχετικά με το αντικείμενο ανάλυσης.
3. Καθαρισμός δεδομένων (Data Cleaning), για την απομάκρυνση θορύβου ή ασυνεπών δεδομένων.
4. Μετασχηματισμός δεδομένων (Data Transformation), όπου τα δεδομένα μετασχηματίζονται ή ενώνονται σε μορφές κατάλληλες για εξόρυξη πληροφοριών από αυτά.
5. Εξόρυξη πληροφοριών από τα δεδομένα (Data Mining), μία απαραίτητη διαδικασία όπου εφαρμόζονται έξυπνες μέθοδοι για την εξαγωγή προτύπων.
6. Αξιολόγηση προτύπων (Pattern Evalutaion), για την αναγνώριση των πραγματικά ενδιαφερόντων προτύπων που αναπαριστούν γνώση, βάσει κάποιων μέτρων σύγκρισης
7. Παρουσίαση γνώσης (Knowledge representation), όπου χρησιμοποιούνται τεχνικές οπτικοποίησης και παρουσίασης συμπερασμάτων για την γνώση που εξορύχτηκε στον χρήστη.

Τα βήματα 1 έως 4 είναι τα διάφορα στάδια της προεπεξεργασίας των δεδομένων, ώστε αυτά να ετοιμασθούν για το βήμα 5 (Data Mining). Παρότι σύμφωνα με τα παραπάνω το Data Mining είναι μόνο ένα βήμα στην όλη διαδικασία, είναι το πιο σημαντικό και το πλέον απαραίτητο καθώς σε αυτό γίνεται η εξόρυξη των προτύπων και της γνώσης που κρύβεται μέσα στα δεδομένα. Λόγω αυτής της σημαντικότητας του κι επειδή έχει επικρατήσει και στο χώρο της βιομηχανίας και των επιχειρήσεων, ο όρος Data Mining χρησιμοποιείται πολλές φορές για να περιγράψει όλη τη διαδικασία ανακάλυψης προτύπων από δεδομένα.



Εικόνα 1: Η πορεία προς την εξαγωγή γνώσης

1.6 Data Streams - «Ροές Δεδομένων»

Οι πρόσφατες τεχνολογικές εξελίξεις έχουν καταστήσει αναγκαία τη συνεχή συλλογή δεδομένων. Απλές συναλλαγές της καθημερινής ζωής, όπως η χρήση πιστωτικής κάρτας, οι τραπεζικές συναλλαγές ή η αναζήτηση στο διαδίκτυο, οδηγούν σε αυτόματη παραγωγή δεδομένων. Σε πολλές περιπτώσεις αυτός ο τεράστιος όγκος δεδομένων μπορεί να μας δώσει ενδιαφέρουσες πληροφορίες, χρήσιμες σε ένα μεγάλο εύρος εφαρμογών.

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) σχεδιάστηκαν αρχικά για να υποστηρίξουν επιχειρηματικές εφαρμογές. Τα δεδομένα σε τέτοιου είδους εφαρμογές μεταβάλλονται μόνο ως αποτέλεσμα συναλλαγών που εφαρμόζονται από τους ανθρώπους / χρήστες. Παρόμοια, οι άνθρωποι / χρήστες θέτουν τα ερωτήματα στα υπάρχοντα δεδομένα. Τα ΣΔΒΔ συμπεριφέρονται παθητικά όσον αφορά στις μεταβολές των δεδομένων και τις ερωτήσεις στα δεδομένα. Τα παραδοσιακά ΣΔΒΔ δεν επαρκούν για ένα σύνολο νέων εφαρμογών που απαιτούν τη συνεχή αξιολόγηση ερωτημάτων σε δεδομένα πραγματικού – χρόνου [56].

1.6.1 Εφαρμογές των Ροών Δεδομένων

Μια ροή δεδομένων είναι μία απεριόριστη (μη - πεπερασμένη) ακολουθία τιμών. Οι τιμές αυτές μπορεί να είναι είτε μίας είτε πολλών διαστάσεων (εγγραφών - tuples). Οι ροές δεδομένων μπορούν να χωριστούν σε δύο τύπους: τις ροές συναλλαγών και τις ροές μετρήσεων.

Ροές συναλλαγών: Πρόκειται για τις ροές δεδομένων, οι οποίες προκύπτουν από τις συναλλαγές και την αλληλεπίδραση μεταξύ οντοτήτων. Παραδείγματα τέτοιων ροών είναι:

- Σε πολλές ιστοσελίδες, η αλληλεπίδραση των χρηστών με την ιστοσελίδα καταγράφεται και παρακολουθείται από εφαρμογές, όπως η εξατομίκευση και η δημιουργία προτιμήσεων. Οι αλληλεπιδράσεις συνεχώς καταγράφονται και αξιοποιούνται.
- Οι τράπεζες παρακολουθούν τις αγορές των πελατών τους μέσω των πιστωτικών τους καρτών και προσπαθούν να ανιχνεύσουν ανωμαλίες που υποδεικνύουν πιθανή κλοπή ή δόλια χρήση τους. Οι συναλλαγές με πιστωτικές κάρτες σχηματίζουν μια συνεχή ροή δεδομένων από συναλλαγές.

Ροές μετρήσεων: Πρόκειται για ροές δεδομένων, οι οποίες προκύπτουν από την παρακολούθηση διάφορων οντοτήτων. Παραδείγματα εφαρμογών είναι:

- Με την εξέλιξη της τεχνολογίας, οι αισθητήρες γίνονται ολοένα και φθηνότεροι με αποτέλεσμα να αναδεικνύεται ένα μεγάλο σύνολο εφαρμογών. Οι αισθητήρες κατανέμονται σε μία περιοχή και καταμετρούν την κατάσταση μιας συγκεκριμένης οντότητας. Οι μετρήσεις δημιουργούν ροές δεδομένων. Για παράδειγμα, η παρακολούθηση της κίνησης στους δρόμους, η καταμέτρηση θερμοκρασίας και ταχύτητας αέρα σε μετεωρολογικούς σταθμούς.
- Σε μεγάλου μεγέθους δίκτυα απαιτείται η παρακολούθησή τους προκειμένου να διεκπεραιωθούν διάφορα ζητήματα, όπως ο εντοπισμός σημείων συμφόρησης ή ο εντοπισμός επίθεσης στο δίκτυο.

Συνήθως παρακολουθείται η κεφαλίδα μέσω των δρομολογητών του δικτύου. Έτσι, μπορεί να θεωρηθεί ότι οι κεφαλίδες σχηματίζουν μια ροή δεδομένων [56].

Κεφάλαιο 2

2.1 Εξόρυξη Δεδομένων (Data Mining)

Με τον όρο εξόρυξη δεδομένων (Data Mining) περιγράφουμε την αυτόματη τετριμμένη διαδικασία, μέσω χρήσης ηλεκτρονικού υπολογιστή, εξαγωγής χρήσιμων πληροφοριών και προτύπων από μεγάλες βάσεις δεδομένων.



Εικόνα 2.1 Ανακάλυψη προτύπων σε δεδομένα (πηγή:

http://techpubs.sgi.com/library/dynaweb_docs/books/MiniSet_T/sgi_html/figures/dataprocess.gif)

Η Εξόρυξη Δεδομένων επιτρέπει στους χρήστες του τη δυνατότητα μέσω της εφαρμογής του να εξάγουν απαντήσεις σε πολύ δύσκολα ερωτήματα (στατιστικού και μη χαρακτήρα) αλλά και γενική γνώση πάνω στο υποκείμενο προς εξέταση. Γνώση την οποία δεν έχουν φανταστεί ότι μπορεί να προκύψει και απαντήσεις σε ερωτήματα που δεν έχουν ακόμα θέσει. Είναι χαρακτηριστικό ένα απόφθεγμα των Grey και Watson που συνοψίζει πλήρως την χρησιμότητα του "Το data mining επιτρέπει σε αναλυτές και διευθυντές να βρουν απαντήσεις χωρίς να έχουν καν ρωτήσει." [27]. Γενικά ο όρος εξόρυξη δεδομένων αναφέρεται σε υψηλού επιπέδου εφαρμογές και μεθόδους που χρησιμοποιούνται για να παρουσιάσουν και να αναλύσουν δεδομένα σε πεδία λήψης αποφάσεων. Η βασική ιδέα που κρύβεται πίσω από τον όρο εξόρυξη δεδομένων είναι η εύρεση εκείνης της μη μηδενικής λύσης, η οποία δίνει τη δυνατότητα εξαγωγής ουσιαστικών κανόνων από δεδομένα [2]. Η όλη διαδικασία βασίζεται στην χρησιμοποίηση αλγορίθμων οι οποίοι αναζητούν κανόνες μεταξύ των μεταβλητών των δεδομένων και στην συνέχεια καταχωρούν τα δεδομένα σε νέες βάσεις δεδομένων. [3]

2.2 Στόχοι

Οι κυριότεροι στόχοι της τεχνικής εξόρυξης δεδομένων είναι πρόβλεψη (Prediction) και περιγραφή (Description). Η πρόβλεψη χρησιμοποιεί τις υπάρχουσες μεταβλητές σε μία βάση δεδομένων, ώστε να προβλέψει άγνωστες ή μελλοντικές αξίες ενδιαφέροντος. Η περιγραφή από την άλλη μεριά επικεντρώνεται στην εύρεση προτύπων και κανόνων, περιγράφοντας τις γενικές ιδιότητες των δεδομένων [4].

2.3 Μέθοδοι εξόρυξης δεδομένων

Στην εξόρυξη δεδομένων χρησιμοποιούνται πολλά είδη μεθόδων που χρησιμοποιούν Μηχανικές μάθησης, όπως:

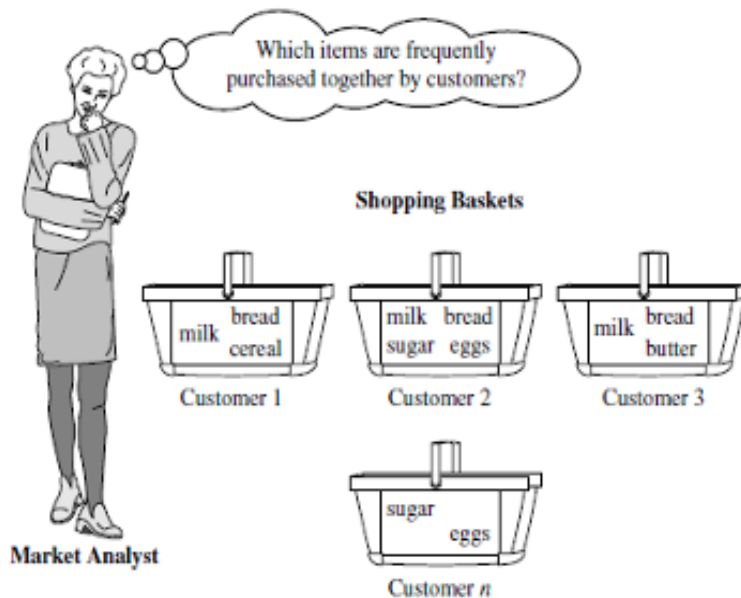
- Κατηγοροποίηση - classification
- Συσχέτιση - association
- Συσταδοποίηση – clustering
- Στατιστική ανάλυση - Statistical Analysis
- Στατιστική πρόβλεψη - Numerical Prediction
- Παλινδρόμηση - Regression Analysis

2.3.1 Κατηγοροποίηση - Classification

Εφαρμόζονται σε δεδομένα τα οποία έχουν ταξινομηθεί σε συγκεκριμένες κλάσεις, με σκοπό την εξαγωγή κανόνων οι οποίοι μπορούν μετέπειτα να χρησιμοποιηθούν για την ταξινόμηση νέων υποδειγμάτων στις ίδιες κλάσεις. Η λειτουργία ενός αλγορίθμου κατηγοροποίησης έχεις ως εξής: τροφοδοτούμε με ένα σύνολο από δεδομένα (dataset) έναν αλγόριθμο κατηγοροποίησης, στη συνέχεια ο αλγόριθμος κατανοεί τους κανόνες βάσει των οποίων ταξινομήθηκαν τα δεδομένα και τέλος βάσει αυτών των κανόνων ο αλγόριθμος έχει την ικανότητα να ταξινομεί τα νέα δεδομένα. Ανάλογα το είδος του ταξινομητή, οι αλγόριθμοι κατηγοροποίησης χωρίζονται σε αυτούς που παράγουν λίστες αποφάσεων και σε αυτούς που παράγουν δένδρα αποφάσεων [2]. Στο επόμενο κεφάλαιο θα αναφερθούν κάποιοι από τους κυριότερους αλγορίθμους της κατηγοροποίησης.

2.3.2 Συσχέτιση - Association

Σκοπός της συσχέτισης είναι η εύρεση των σημαντικότερων αλληλοεξαρτήσεων μεταξύ των διαφόρων πεδίων/χαρακτηριστικών του συνόλου εκπαίδευσης [5]. Η ουσιαστική διαφορά από τους κανόνες ταξινόμησης είναι ότι δύνανται να προβλέψουν την τιμή κάθε χαρακτηριστικού ή συνδυασμού χαρακτηριστικών και όχι μόνο της τάξης. Επίσης δεν χρησιμοποιούνται σύνολα κανόνων. Η πιο συνηθισμένη μορφή της μεθόδου είναι η ανάλυση του «καλαθιού της αγοράς» (market basket analysis, εικόνα 2.2), όπου σκοπός είναι να αναγνωρισθούν προϊόντα που αγοράζονται μαζί [3].



Εικόνα 2.2: market basket analysis (Jiawei Han and Micheline Kamber, “Data Mining: Concepts and Techniques”, 2 edition (4 Jun 2006))

Ένας κανόνας συσχέτισης είναι μια έκφραση της μορφής $X \rightarrow Y \{S, C\}$, όπου το X και Y είναι σύνολα τιμών των πεδίων, πχ σύνολα οικονομικών αγαθών. Το S , C αναφέρονται στην υποστήριξη (Support) και την εμπιστοσύνη (Confidence) αντίστοιχα του κάθε κανόνα. Η υποστήριξη είναι οι επιτυχείς προβλέψεις του κανόνα ως προς το σύνολο των δεδομένων και είναι καθαρός αριθμός, ενώ η εμπιστοσύνη είναι οι επιτυχείς προβλέψεις του κανόνα ως τα υποδείγματα στα οποία αυτός εφαρμόζεται και εκφράζεται με ποσοστό [6]. Στο επόμενο κεφάλαιο θα αναφερθούν κάποιοι από τους κυριότερους αλγορίθμους της συσχέτισης.

2.3.3 Συσταδοποίηση

Οι αλγόριθμοι συσταδοποίησης είναι ιδιαίτερα διαδεδομένοι και η λογική τους μοιάζει αρκετά με αυτήν των αλγορίθμων κατηγοροποίησης. Η ειδοποιός διαφορά είναι πως τα δεδομένα του συνόλου εκπαίδευσης δεν είναι προταξινομημένα. Αντίθετα το σύνολο των εγγραφών χωρίζονται σε ομάδες έτσι ώστε οι εγγραφές της ίδιας ομάδας να έχουν περισσότερες ομοιότητες μεταξύ τους με βάση κάποιου προκαθορισμένα κριτήρια, από ότι με εγγραφές άλλων ομάδων. Σε ορισμένους αλγορίθμους συσταδοποίησης υπάρχει η δυνατότητα μια εγγραφή να ανήκει σε περισσότερες της μια ομάδες ταυτόχρονα, ενώ άλλοι αλγόριθμοι δίνουν την πιθανότητα καταχώρησης του κάθε υποδείγματος σε μία ομάδα.

Η μέθοδος της συσταδοποίησης μπορεί να είναι στατιστική είτε αριθμητική (statistical/ numerical clustering), οπότε χρησιμοποιούνται διάφορα αριθμητικά κριτήρια ομοιότητας και οι ομάδες που προκύπτουν περιγράφονται από αριθμητικές τιμές, είτε εννοιολογική (conceptual clustering) οπότε σε αυτή την περίπτωση ο προσδιορισμός των ομάδων βασίζεται στο νόημα και στις έννοιες που τα διάφορα αριθμητικά στοιχεία αντιπροσωπεύουν και οι τιμές που προκύπτουν είναι κατηγορικές και όχι αριθμητικές [7]. Στο επόμενο

κεφάλαιο θα αναφερθούν κάποιοι από τους κυριότερους αλγορίθμους της συσταδοποίησης.

2.4 Εφαρμογές και συνεισφορά των Big Data

Τα Big Data είναι μια μεγάλη βιομηχανία. Μια έρευνα στο Ινστιτούτο Τεχνολογίας της Μασαχουσέτης δείχνει ότι οι εταιρείες που βασίζονται στην λήψη αποφάσεων στην επιστήμη των δεδομένων, απολαμβάνουν μια αύξηση της παραγωγικότητας στο 5-6%. Οι εταιρείες που χρησιμοποιούν τα δεδομένα πιο αποτελεσματικά ξεχωρίζουν από τις υπόλοιπες. Μια έκθεση του Ινστιτούτου McKinsey Global Institute (MGI) καταδεικνύει τη μετασχηματιστική επίδραση των big data σε ολόκληρους τομείς που κυμαίνονται από την υγειονομική περίθαλψη στο λιανικό εμπόριο έως και την κατασκευή πολιτικών εκστρατειών [29]. Ακριβώς όπως βοηθά τις επιχειρήσεις να αυξήσουν την παραγωγικότητα τους, τα big data επιτρέπουν στις κυβερνήσεις να βελτιώσουν τη διαχείριση του δημόσιου τομέα και βοηθά παγκόσμιους οργανισμούς στην ανάλυση των πληροφοριών για την εκπόνηση στρατηγικού σχεδιασμού. Η ζήτηση για μεγάλα στοιχεία επιταχύνεται. Σε αυτό το κεφάλαιο, παρουσιάζουμε μερικά τυπικά παραδείγματα με τα οφέλη που τα big data έχουν επιφέρει.

Υγεία:

Ο Δρ Russ Altman, καθηγητής της ιατρικής και βιοτεχνολογίας στο Πανεπιστήμιο του Στάνφορντ, και οι συνεργάτες του έκαναν μια πρωτοποριακή με τη χρήση των Big Data. Βρήκαν ότι όταν λαμβάνοντας μαζί, Paxil, το πολύ γνωστό αντικαταθλιπτικό που παρέχονται σε εκατομμύρια Αμερικανών, και το Pravachol, ένα πολύ δημοφιλές φάρμακο μείωσης της χοληστερόλης, έχουν φοβερές παρενέργειες. Αυξάνουν τα επίπεδα γλυκόζης στο αίμα των ασθενών σε επίπεδα διαβητικών. Κάθε φάρμακο που λαμβάνεται μόνο του δεν έχει τις διαβητικές παρενέργειες, ως εκ τούτου, η Υπηρεσία Τροφίμων και Φαρμάκων (Food and Drug Administration FDA) ενέκρινε τα φάρμακα για χρήση. Σίγουρα ο FDA δεν μπορούσε να δοκιμάσει κάθε φάρμακο για κάθε πιθανή αλληλεπίδραση. Ο Altman και η ομάδα του έκαναν την ανακάλυψή τους κατά την επιδίωξη στατιστικών αναλύσεων και τεχνικές εξόρυξης δεδομένων για την αναγνώριση προτύπων σε μεγάλα σύνολα 17 δεδομένων. Σχεδόν 15 εκατομμύρια Αμερικανοί χρησιμοποιούν είτε Paxil ή Pravachol. Υπολογίζεται ότι ένα εκατομμύριο χρησιμοποιούσε και τα δύο. Για αυτούς τους χρήστες, το έργο του Altman και των συνεργατών του ήταν δυνητικά σωτήριο [30]. Τα ευρήματα των Altman και της ομάδας του δεν είναι η μόνη σημαντική ανακάλυψη βασίζεται σε ανάλυση των big data στον τομέα της υγείας. Η ανακάλυψη των ανεπιθύμητων παρενεργειών του φαρμάκου Vioxx, το οποίο οδήγησε στην απόσυρσή του από την αγορά, κατέστη δυνατή με την ανάλυση των κλινικών και οικονομικών δεδομένων που συλλέχθηκαν από το Kaiser Permanente [31]. Αν η Kaiser Permanente δεν άθροιζε κλινικά και οικονομικά δεδομένα, οι ερευνητές μπορεί να μην ήταν σε θέση να αποδώσουν 27.000 καρδιακούς θανάτους από ανακοπή που συνέβησαν μεταξύ 1999 και 2003, με τη χρήση του φαρμάκου. Ένα άλλο συχνά αναφερόμενο παράδειγμα είναι το Google Flu Trends, το οποίο προβλέπει και εντοπίζει εστίες γρίπης και λοιπών λοιμώξεων της κάνοντας χρήση πληροφοριών – συσχέτιση ερωτημάτων αναζήτησης – τα οποία αρχικά δεν είχαν συλλεχθεί για αυτή την καινοτόμο εφαρμογή. Φυσικά, η έγκαιρη ανίχνευση της δραστηριότητας της νόσου, όταν ακολουθείται από ταχεία αντίδραση, μπορεί να μειώσει τον αντίκτυπο της εποχικής και της πανδημικής γρίπης [32].

Mobile Networks:

Ο τομέας της υγείας δεν είναι ο μοναδικός χώρος για την πρωτοποριακή χρήση των big data. Μια ομάδα επιστημόνων που εργάζονται σε ένα συνεργατικό έργο στο MIT, το Harvard και αντίστοιχα ερευνητικά πανεπιστήμια αναλύουν σε αυτή τη φάση δεδομένα επικοινωνιών κινητής τηλεφωνίας για να κατανοήσουν καλύτερα τις ανάγκες του 1 δισεκατομμυρίου ανθρώπων που ζουν σε οικισμούς ή φτωχογειτονίες στις αναπτυσσόμενες χώρες [33]. Ερευνούν τρόπους για την πρόβλεψη ελλείψεων τροφίμων χρησιμοποιώντας μεταβλητές όπως οι τιμές της αγοράς, η ξηρασία, οι μεταναστεύσεις, η προηγούμενη περιφερειακή παραγωγή και εποχιακές διακυμάνσεις [34]. Για να ποσοτικοποιήσουν τα κύματα εγκληματικότητας, παρακολουθώντας το χρόνο, τον τόπο και τη φύση των εγκληματικών δραστηριοτήτων σε τοποθεσίες σε όλη την πόλη [35]. Και για να αποφασίσουν ποια παρέμβαση λειτουργεί καλύτερα για τη βελτίωση των μαθησιακών αποτελεσμάτων στην ανάπτυξη των σχολείων της χώρας [36].

Έξυπνο δίκτυο:

Ένα άλλο παράδειγμα είναι το «έξυπνο δίκτυο», το οποίο αναφέρεται στον εκσυγχρονισμό του ισχύοντος ηλεκτρικού δικτύου για να εισαγάγει μια αμφίδρομη ροή πληροφοριών της ηλεκτρικής ενέργειας [37]. Το έξυπνο δίκτυο έχει σχεδιαστεί για να επιτρέπει στους φορείς παροχής υπηρεσιών ηλεκτρικής ενέργειας, στους χρήστες (καταναλωτές ηλ. ενεργείας), και τρίτους φορείς να παρακολουθούν και να ελέγχουν τη χρήση ηλεκτρικής ενέργειας. Οι βοηθητικές εφαρμογές βλέπουν το έξυπνο δίκτυο ως έναν τρόπο για να εντοπίσουν με ακρίβεια διακοπές ρεύματος ή άλλα προβλήματα, συμπεριλαμβανομένων των κυβερνο-επιθέσεων ή φυσικών καταστροφών, έτσι ώστε τεχνικοί να αποστέλλονται για την επίλυση των προβλημάτων. Οι καταναλωτές επωφελούνται από περισσότερες επιλογές για το πώς, πότε και πόση ηλεκτρική ενέργεια χρησιμοποιούν [38]. Οι φίλοπεριβαλλοντικοί φορείς βλέπουν το έξυπνο δίκτυο ως ένα κλειδί για την παροχή καλύτερης ποιότητας ισχύος, και την πιο αποτελεσματική κίνηση προς τις ανανεώσιμες πηγές ενέργειας.

Διαχείριση της κυκλοφορίας:

Μια παρακείμενη περιοχή για τα δεδομένα με γνώμονα την περιβαλλοντική καινοτομία είναι η διαχείριση και ο έλεγχος της κυκλοφορίας. Οι κυβερνήσεις ανά τον κόσμο εγκαθιστούν ηλεκτρονικά συστήματα τιμολόγησης των διοδίων, τα οποία καθορίζουν χρεώσεις και πληρωμές με βάση την κινητικότητα και την κυκλοφοριακή συμφόρηση [39]. Οι χρήστες πληρώνουν ανάλογα με τη χρήση των οχημάτων τους και των δρόμων. Οι πολιοδόμοι επωφελούνται από την ανάλυση των δεδομένων τοποθεσία για αποφάσεις που αφορούν την κατασκευή δρόμων και τις μαζικές διελεύσεις, τον μετριασμό της κυκλοφοριακής συμφόρησης, καθώς και το σχεδιασμό για την ανάπτυξη υψηλής πυκνότητας ροής οχημάτων [40]. Οι αποφάσεις αυτές μπορούν όχι μόνο να μειώσουν τη συμφόρηση, αλλά και να μετριάσουν τις εκπομπές ρύπων [41]. Παράλληλα, επωφελούνται οι οδηγοί ατομικά από έξυπνες δρομολόγησης, βασιζόμενες σε πληροφορίες για την κυκλοφορία σε πραγματικό χρόνο, συμπεριλαμβανομένων των εκθέσεων ατυχημάτων και πληροφορίες σχετικά με προγραμματισμένα οδικά έργα και περιοχές υψηλής κυκλοφοριακής συμφόρησης [42].

Λιανεμπόριο:

Τα big data μετασχηματίζουν και την λιανική αγορά. Το σύστημα απογραφής διαχείρισης της γνωστής αλυσίδας λιανικής πώλησης, που ονομάζεται Retail Link, ήταν το πρώτο το οποίο πρωτοστάτησε στην εποχή των big data, δίνοντας στους προμηθευτές τη δυνατότητα να βλέπουν το ακριβή αριθμό των προϊόντων τους, σε κάθε ράφι, σε κάθε κατάστημα, για κάθε συγκεκριμένη χρονική στιγμή. Πολλοί από εμάς χρησιμοποιούν «Οι πελάτες που αγόρασαν αυτό επίσης αγόρασαν» χαρακτηριστικό της Amazon (και πλέον σήμερα χρησιμοποιείτε από πολλά e-shop), προτρέποντας τους χρήστες να εξετάσει την αγορά επιπλέον προϊόντων που επιλέγονται από ένα συνεργατικό εργαλείο φιλτραρίσματος [43]. Το βασικό επιχειρηματικό μοντέλο του Διαδικτύου χρηματοδοτεί προϊόντα και υπηρεσίες με στοχευμένες διαφημίσεις η αξία των οποίων συσχετίζεται άμεσα με την ποσότητα των πληροφοριών που συλλέγονται από τους χρήστες [44]. Οι επιχειρήσεις δεν ενδιαφέρονται τόσο πολύ για την ταυτότητα του κάθε χρήστη, αλλά μάλλον για τα χαρακτηριστικά του προφίλ τους, το οποίο καθορίζει τη φύση των διαφημίσεων που του εμφανίζονται. Τα Analytics μπορεί επίσης να χρησιμοποιηθούν σε offline περιβάλλον για τη μελέτη της συμπεριφοράς των πελατών μέσα στο κατάστημα για τη βελτιώσει της διάταξης των καταστημάτων, το φάσμα προϊόντων, και την τοποθέτηση στο ράφι. Ο McKinsey αναφέρει σαφώς ότι «πρόσφατες καινοτομίες έχουν επιτρέψει στους λιανοπωλητές να παρακολουθούν τα αγοραστικά πρότυπα των πελατών (π.χ., μονοπάτι και το χρόνο που δαπανάται σε διάφορα μέρη σε ένα κατάστημα), αντλώντας δεδομένα θέσης πραγματικού χρόνου από smartphone εφαρμογές (π.χ., Shopkick), αναμεταδότες από το καλάθι αγορών, ή παθητικά την παρακολούθηση της θέσης των κινητών τηλεφώνων μέσα σε ένα περιβάλλον λιανικής πώλησης.» [45]. Ολοένα και περισσότερο, οι οργανισμοί επιδιώκουν να συνδέσουν την ηλεκτρονική δραστηριότητα με τη συμπεριφορά εκτός σύνδεσης, τόσο για να αξιολογήσουν την αποτελεσματικότητα των διαφημιστικών καμπανιών στο διαδίκτυο και να προωθήσουν τους πελάτες τους με διαφημίσεις στο διαδίκτυο.

Πληρωμές :

Ένα ακόμα σημαντικός χώρος για την αξιοποίηση των big data είναι η ανίχνευσης της απάτης στον τομέα των πληρωμών με κάρτα. Με το ηλεκτρονικό εμπόριο να κατακτά ολοένα και μεγαλύτερο μερίδιο της αγοράς λιανικής πώλησης, οι έμποροι, οι οποίοι φέρουν την τελική ευθύνη για τη δόλια και κακόβουλη χρήση πληρωμών με κάρτα [46], πρέπει να εφαρμόσουν ισχυρούς μηχανισμούς για τον εντοπισμό ύποπτων συναλλαγών που συχνά πραγματοποιούνται σε “πρωτάρηδες” πελάτες. Με το σκοπό αυτό, οι εταιρείες ανέπτυξαν λύσεις για τους εμπόρους συστήματα πρόγνωσης απάτης «Συναλλαγή Μη Παρούσας Κάρτας», προκειμένου να μετρηθεί σε πραγματικό χρόνο η πιθανότητα μιας συναλλαγής να είναι ύποπτη ως απατηλή. Για να γίνει αυτό, οι υπηρεσίες αναλύουν το ιστορικό του αγοραστή και παρέχουν αξιολογήσεις, σαν μια συνοπτική λίστα των αναφορών, αλλά με τη μορφή μιας ενιαίας βαθμολογίας.

Αστρονομία:

Έχει κατασκευαστεί ένα σύστημα με την ονομασία SKICAT από το JPL/Caltech και χρησιμοποιείται από τους αστρονόμους στο να αναγνωρίζουν αυτόματα τους διάφορους γαλαξίες και αστεροειδείς σε μία μεγάλη κλίμακα η οποία περιέχει διάφορα αστρονομικά μεγέθη.

Βιολογία:

Διάφορα συστήματα έχουν κατασκευαστεί στο να εξάγουν κανόνες που αφορούν την δομή των οργανισμών, την ανάλυση του DNA, καθώς και την δυνατότητα για εύρεση φαρμάκων για την καταπολέμηση ασθενειών.

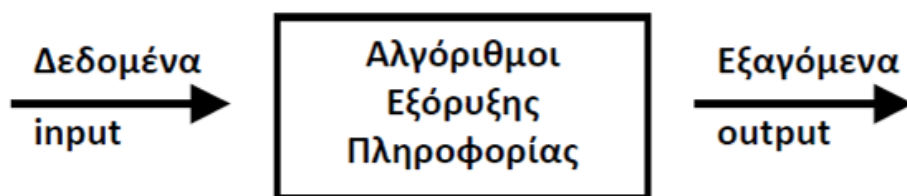
Διαδίκτυο:

Τέλος, σκεφτείτε τα ογκώδη δεδομένα που διατηρούν οι online giants της τεχνολογίας, το Google, το Facebook, η Microsoft, η Apple και το Amazon. Οι εταιρείες αυτές συγκεντρώνουν προηγουμένως αδιανόητες ποσότητες προσωπικών δεδομένων. Το Facebook, για παράδειγμα, έχει πάνω από 900 εκατομμύρια χρήστες που ανεβάζουν πάνω από 250 εκατομμύρια φωτογραφίες και κάνουν κλικ στο κουμπί "όπως" περισσότερο από 2,5 δισεκατομμύρια φορές την ημέρα [47]. Η Google προσφέρει μια πληθώρα προϊόντων και υπηρεσιών με μεγάλη ένταση δεδομένων, όπως η πανταχού παρούσα μηχανή αναζήτησης, το λειτουργικό σύστημα Android, το πρόγραμμα περιήγησης Chrome, το Gmail, το Youtube, οι Χάρτες Google, το Google Plus, το Google Analytics, το Google Apps και πολλά άλλα [48]. Επιπλέον, η Google κατέχει τη μεγαλύτερη εταιρεία προβολής διαφημίσεων σε απευθείας σύνδεση DoubleClick, την οποία αγόρασε το 2007, σε μεγάλο βαθμό για τη συντριβή των υπερασπιστών ιδιωτικού απορρήτου [49], καθώς και για την AdMob, την κορυφαία εταιρεία διαφήμισης για κινητά. Ως αποτέλεσμα, η Google έχει πλέον παρουσία σε πάνω από το 70% των ιστότοπων τρίτων [50]. Η Amazon και το Yahoo αναζητούν νέους τρόπους για να εκμεταλλευτούν και να αποκομίσουν έσοδα από το θησαυρό των δεδομένων των πελατών τους [51].

Κεφάλαιο 3

3.1 Αλγόριθμοι Εξόρυξης Δεδομένων (Data Mining)

Η ανάλυση συστάδων βρίσκει πρόσφορο έδαφος στη διερεύνηση και άντληση δεδομένων (data mining), στη μάθηση των μηχανών (machine learning), στην ανάπτυξη νευρωνικών δικτύων (neural networks), στην ιατρική (ανάλυση εικόνας, συσταδοποίηση ασθενών, θεραπειών ή συμπτωμάτων), στην ψυχιατρική (σωστή διάγνωση των συμπτωμάτων όπως, παράνοια, σχιζοφρένεια, ουσιαστική για επιτυχημένη θεραπεία) [8].



Εικόνα 3: Διαδικασία εξόρυξης πληροφορίας

3.2 Αλγόριθμοι Συσταδοποίησης

Για παράδειγμα ο παγκόσμιος ιστός αποτελείται από δισεκατομμύρια σελίδες και τα αποτελέσματα από ένα ερώτημα σε μια μηχανή αναζήτησης μπορεί να επιστρέψει χιλιάδες σελίδες σαν αποτέλεσμα, ως εικόνα 3. Η συσταδοποίηση μπορεί να χρησιμοποιηθεί για την ομοαδοποίηση αυτών των αποτελεσμάτων σε ένα μικρό αριθμό από ομάδες, καθεμιά από τις οποίες σκιαγραφεί μια διάσταση του ερωτήματος. Για παράδειγμα ένα ερώτημα για μια ταινία μπορεί να επιστρέψει σελίδες ομαδοποιημένες σε κατηγορίες όπως ανασκοπήσεις, τρέιλερ, ηθοποιούς και θέατρα. Κάθε κατηγορία (cluster) μπορεί να διασπαστεί σε υποκατηγορίες (subclusters), παράγοντας ιεραρχικές δομές που στη συνέχεια βοηθούν περισσότερο το χρήστη κατά την αναζήτηση που κάνει [9].

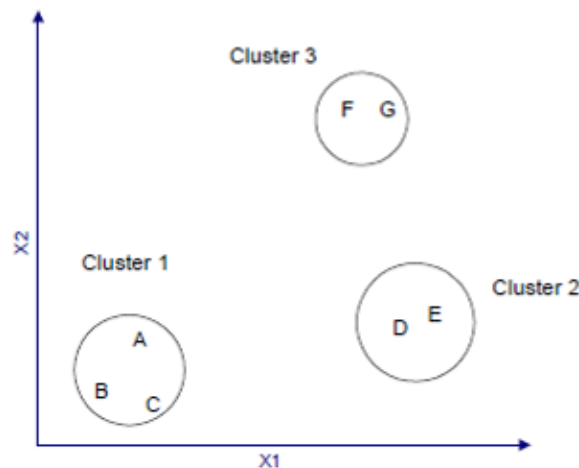
Οι κύριες κατηγορίες ομαδοποιήσεων είναι οι εξής:

- Ιεραρχική συσταδοποίηση, όπου παράγει εμφωλευμένες συστάδες που ανήκουν σε μία ιεραρχία.

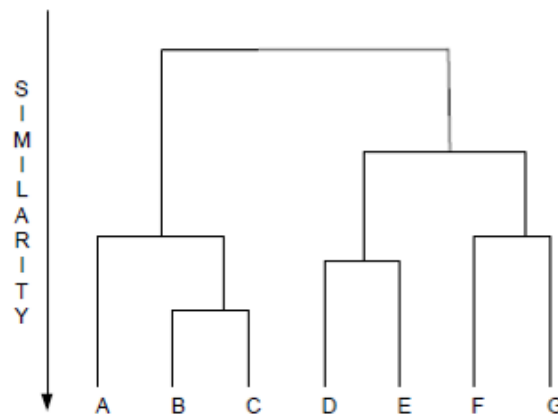
- Διαχωριστική ή διαμεριστική ή τμηματοποίηση, η οποία παράγει μη επικαλυπτόμενες συστάδες.

3.2.1 Ιεραρχική συσταδοποίηση

Ένας ιεραρχικός αλγόριθμος μπορεί να αποδοθεί με ένα δενδροδιάγραμμα το οποίο παρουσιάζει τις συγχωνεύσεις στοιχείων για την δημιουργία clusters και τα επίπεδα ομοιότητας με βάση τα οποία αλλάζουν οι ομάδες και διαμορφώνονται τα clusters. Το δενδροδιάγραμμα που αντιστοιχεί στα δεδομένα της εικόνας 4 παρουσιάζεται στην εικόνα 5. Χαρακτηριστικό του διαγράμματος είναι τα διάφορα επίπεδα στα οποία αποδίδονται διαφορετικές ομαδοποιήσεις [9].



Εικόνα 4 : Σημεία σε 3 clusters



Εικόνα 5 : Δενδροδιάγραμμα ιεραρχικού αλγορίθμου

Οι περισσότεροι ιεραρχικοί αλγόριθμοι είναι παραλλαγές των αλγορίθμων απλού-συνδέσμου (single-link) και του πλήρους-συνδέσμου (complete-link). Οι διαφορές μεταξύ των αλγορίθμων αυτών έχει να κάνει με τον τρόπο με τον οποίο ορίζουν την ομοιότητα μεταξύ στοιχείων και κατά συνέπεια clusters πριν την συγχώνευσή τους.

Με βάση το κριτήριο απλού συνδέσμου (single-link), η ομοιότητα μεταξύ δύο συστάδων βασίζεται στα δύο πιο όμοια (πιο γειτονικά) σημεία στις

διαφορετικές συστάδες. Με βάση το κριτήριο πλήρους συνδέσμου (complete-link), η ομοιότητα μεταξύ δύο συστάδων βασίζεται στα δύο πιο ανόμοια (πιο απόμακρα) σημεία στις διαφορετικές συστάδες, δηλαδή στα σημεία με τη μέγιστη απόσταση μεταξύ τους. Επίσης ο μέσος όρος συστάδων (group average) είναι ουσιαστικά η μέση τιμή των αποστάσεων μεταξύ κάθε πιθανού ζεύγους μεταξύ των σημείων των δύο συστάδων. Βρίσκεται κάπου ανάμεσα στην ελάχιστη και τη μέγιστη απόσταση.

Τέλος η συσταδοποίηση κατά Ward, βασίζεται στην εφαρμογή της ανάλυσης της διακύμανσης στις παρατηρήσεις των ομάδων με σκοπό την εκτίμηση των αποστάσεων μεταξύ των ομάδων. Ουσιαστικά, η μέθοδος αυτή αποσκοπεί στην ελαχιστοποίηση της μεταβλητότητας μεταξύ δύο εξεταζόμενων ομάδων που σχηματίζονται σε κάθε διαδοχικό στάδιο της ιεραρχικής ταξινόμησης των ομάδων. Η διασύνδεση του Ward στηρίζεται στην ελαχιστοποίηση του αθροίσματος των τετραγώνων του σφάλματος, το οποίο ορίζεται ως το άθροισμα των τετραγώνων των αποστάσεων από κάθε παρατήρηση προς το κέντρο βάρους της ομάδας του. Θεωρείται ως η πλέον αποτελεσματική μέθοδος, έχει όμως το μειονέκτημα να σχηματίζει ομάδες πολύ μικρού μεγέθους.

3.2.2 Διαμεριστικοί αλγόριθμοι συσταδοποίησης

Ένας διαμεριστικός αλγόριθμος έχει ως αποτέλεσμα μια διαμέριση του χώρου δεδομένων σε αντίθεση με τους ιεραρχικούς αλγορίθμους που δημιουργούν πιο πολύπλοκες δομές που περιγράφονται από τα δένδρογράμματα. Οι αλγόριθμοι αυτοί υπερτερούν σε περιπτώσεις όπου τα δεδομένα είναι πάρα πολλά και η δημιουργία δένδρογραμμάτων είναι αδύνατη. Το κυρίως πρόβλημα των αλγορίθμων αυτών είναι η απόφαση για τον αριθμό των τελικών clusters. Ο αριθμός αυτός καθορίζεται κυρίως από την προσπάθεια βελτιστοποίησης μιας συνάρτησης. Στην πραγματικότητα αυτό που γίνεται είναι να τρέχει ο αλγόριθμος για διάφορους αριθμούς από clusters και να επιλέγεται εκείνη η τελική κατάσταση η οποία βελτιστοποιεί την παραπάνω συνάρτηση.

Το κριτήριο που χρησιμοποιείται κυρίως σε διαμεριστικούς αλγορίθμους για την τελική απόφαση του αριθμού των clusters είναι το κριτήριο του τετραγωνικού λάθους ή η συνάρτηση τετραγωνικού λάθους (squared error function).

Ο αλγόριθμος k-means είναι ένας πολύ απλός και πολύ διαδεδομένος διαμεριστικός αλγόριθμος που χρησιμοποιεί το κριτήριο του τετραγωνικού λάθους. Ο αλγόριθμος ξεκινά με μια τυχαία διαμέριση σε clusters και συνεχώς τοποθετεί στοιχεία στα clusters με βάση την απόσταση των στοιχείων από το κεντροειδές του cluster. Αυτό σταματάει μέχρι να ικανοποιηθεί κάποιο κριτήριο το οποίο μπορεί να είναι η ελαχιστοποίηση της συνάρτησης τετραγωνικού λάθους ή η μη διαφοροποίηση των clusters από κάποια επανάληψη και μετά. Ο αλγόριθμος αυτός είναι δημοφιλής εξαιτίας της απλότητας υλοποίησης του και της πολυπλοκότητας του η οποία είναι της τάξης n ($O(n)$), όπου n είναι ο αριθμός των στοιχείων. Το μόνο πρόβλημα που έχει ο αλγόριθμος είναι στην αρχική επιλογή των clusters. Αν η επιλογή αυτή δεν είναι αρκετά προσεκτική τότε το κριτήριο τετραγωνικού λάθους συγκλίνει σε τοπικά ελάχιστο κάνοντας την τελική επιλογή cluster ανεπιτυχή.

Για να εξαλειφθούν οι αποκλίσεις αυτές του αλγορίθμου k-means, ένας άλλος αλγόριθμος ο k-Medoids, όπου αντί να λαμβάνουμε τη μέση τιμή των αντικειμένων σε μία συστάδα ως σημείο αναφοράς, μπορούμε να επιλέξουμε τα πραγματικά αντικείμενα που αντιπροσωπεύουν τις ομάδες - συστάδες, χρησιμοποιώντας ένα αντιπροσωπευτικό αντικείμενο ανά συστάδα. Τα υπόλοιπα αντικείμενα αντιστοιχούν στο σύμπλεγμα του οποίου το αντιπροσωπευτικό αντικείμενο είναι το πιο παρόμοιο. Στη συνέχεια, η μέθοδος κατανομής γίνεται με βάση την αρχή της ελαχιστοποίησης του αθροίσματος των αποκλίσεων μεταξύ κάθε αντικειμένου p και του αντίστοιχου αντιπροσωπευτικού αντικειμένου [11].

Επίσης προτείνεται ένας αλγόριθμος εξαντλητικής αναζήτησης, ο οποίος εγγυάται ότι θα δημιουργήσει μία βέλτιστη περίληψη ενός δεδομένου μεγέθους L για ένα δεδομένο σύνολο συναλλαγών (transactions) T . Το πρώτο βήμα του αλγορίθμου είναι η δημιουργία του συνόλου T (όλα τα πιθανά υποσύνολα του T) που υποδηλώνεται από το Z . Το μέγεθος του Z είναι $2^{|T|}$. Το δεύτερο βήμα είναι η αναζήτηση όλων πιθανών υποσυνόλων του Z $(2^{2^{|T|}})$. Ο αλγόριθμος παρουσιάζεται στη σελίδα 8 της αναφοράς [10].

Ο αλγόριθμος εξαντλητικής αναζήτησης που αναφέρθηκε στην προηγούμενη παράγραφο, απαιτεί αναζήτηση σε χώρο $(2^{2^{|T|}})$. Για παράδειγμα για 4 συναλλαγές (transactions) θα χρειαστεί αναζήτηση σε 65,536 υποσύνολα, το οποίο το καθιστά υπολογιστικά ανέφικτο ακόμη και για πολύ μικρά σύνολα δεδομένων.

Ένας ακόμη αλγόριθμος συσταδοποίησης, είναι ο αλγόριθμος Dbscan (Density-Based Spatial Clustering of Applications with Noise). Ανήκει στην κατηγορία πυκνότητας γράφων και όπως θα αναφερθεί αργότερα είναι αλγόριθμος όπου χρησιμοποιεί την προσέγγιση του MicroCluster [15].

Κατάλληλος για ομάδες που έχουν υψηλή πυκνότητα σημείων, οι οποίες μπορεί να είναι διαχωρισμένες από άλλα σημεία (θόρυβος) χαμηλότερης πυκνότητας. Προϋποθέτει ότι η πυκνότητα των ομάδων είναι παρόμοια, χωρίς μεγάλες διακυμάνσεις. $\text{πυκνότητα} = \frac{\text{\#σημείων}}{\text{σε ακτίνα Eps}}$.

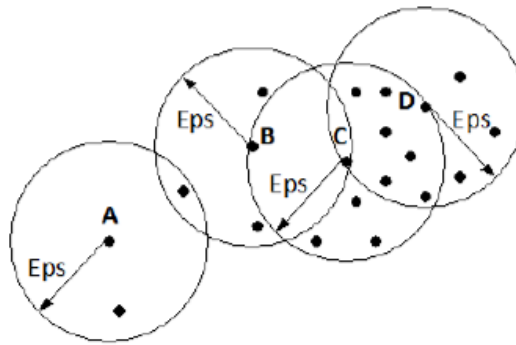
Κεντρικό σημείο: έχει πυκνότητα μεγαλύτερη ή ίση από μία τιμή MinPts (ανήκουν στο εσωτερικό των ομάδων).

•**Συνοριακό σημείο:** έχει πυκνότητα μικρότερη από MinPts, αλλά απέχει από ένα κεντρικό σημείο απόσταση μικρότερη ή ίση από Eps (βρίσκονται στα όρια των ομάδων).

•**Θορυβώδες σημείο:** κάθε άλλο σημείο (ανήκουν στις περιοχές χαμηλής

πυκνότητας).

- $\text{MinPts} = 10$
- C,D κεντρικά.
- B συνοριακό.
- A θορυβώδες.



Εικόνα 6: Σημεία DBScan

Τα βήματα του αλγορίθμου είναι τα εξής:

1. Χαρακτήρισε κάθε σημείο ως κεντρικό, συνοριακό ή θόρυβο.
2. Αγνόησε όλα τα σημεία θορύβου.
3. Δημιούργησε ένα γράφο με μια κορυφή για κάθε σημείο.
4. Τοποθέτησε μια ακμή μεταξύ όλων των κεντρικών σημείων που είναι σε απόσταση έως Eps μεταξύ τους.
5. Θέσε κάθε ομάδα συνδεδεμένων κεντρικών σημείων ως μια διαφορετική συστάδα.
6. Ανάθεσε κάθε συνοριακό σημείο σε μία από τις συστάδες που περιέχει το πιο κοντινό του κεντρικό σημείο.

Η επιλογή Eps , MinPts γίνεται ως εξής:

1. Για κάθε σημείο, βρίσκουμε το k πλησιέστερο προς αυτό, καθώς και τη μεταξύ τους απόσταση.
2. Ταξινομούμε τα σημεία ως προς την απόστασή τους από το k -οστό πλησιέστερο τους.
3. Καθορίζουμε τις τιμές των Eps και MinPts έτσι ώστε να διαχωρίζονται τα σημεία που ανήκουν σε ομάδες από τα θορυβώδη σημεία [13].

3.3 Αλγόριθμοι Κατηγοροποίησης

Συστήματα που δημιουργούν ταξινομητές είναι από τα πιο συχνά χρησιμοποιούμενα εργαλεία σε εφαρμογές data mining. Τέτοια συστήματα δέχονται σαν είσοδο ένα σύνολο δεδομένων (dataset) αποτελούμενο από υποδείγματα που ανήκουν σε κάποιες κλάσεις και στη συνέχεια παράγουν έναν ταξινομητή ικανό να προσδιορίσει με ακρίβεια την κλάση στην οποία ένα νέο υπόδειγμα ανήκει.

Στις αρχές του 1980 ο Quinlan, ένας ερευνητής στην επιστήμη της μηχανικής μάθησης ανέπτυξε έναν αλγόριθμο για δένδρα αποφάσεων τον Id3 (iterative dichotomiser). Στην συνέχεια παρουσίασε τον διάδοχο του id3 τον αλγόριθμο

C4.5, ο οποίος αποτελεί ακόμα και σήμερα ορόσημο με το οποίο συγκρίνονται πολλοί πρόσφατα ανεπτυγμένοι αλγόριθμοι μάθησης με επίβλεψη.

Δένδρα αποφάσεων

Ο αλγόριθμος C4.5 υιοθετεί μια άπληστη (greedy) προσέγγιση στην κατά την οποία τα δένδρα αποφάσεων δημιουργούνται από πάνω προς τα κάτω με αναδρομική διαδικασία διαίρει και βασίλευε. Με δεδομένο ένα σύνολο υποδειγμάτων ο αλγόριθμος δημιουργεί ένα αρχικό δένδρο χρησιμοποιώντας την τεχνική διαίρει και βασίλευε ως ακολούθως:

- Αν όλα τα υποδείγματα στο σύνολο ανήκουν στην ίδια τάξη ή το σύνολο υποδειγμάτων είναι πολύ μικρό, το δένδρο αποτελείται από ένα φύλλο με όνομα την τάξη με την μεγαλύτερη συχνότητα εμφάνισης.

- Αλλιώς γίνεται ένας έλεγχος με βάση μια μεταβλητή με δύο ή περισσότερες πιθανές τιμές. Ο έλεγχος αυτός γίνεται η βάση του δένδρου και από αυτή ξεκινούν κλαδιά για κάθε πιθανό αποτέλεσμα από τον έλεγχο. Έτσι το σύνολο των υποδειγμάτων "σπάει" σε μικρότερα σύνολα ανάλογα με το αποτέλεσμα του ελέγχου και για κάθε νέο σύνολο ακολουθείται αναδρομικά η παραπάνω διαδικασία.

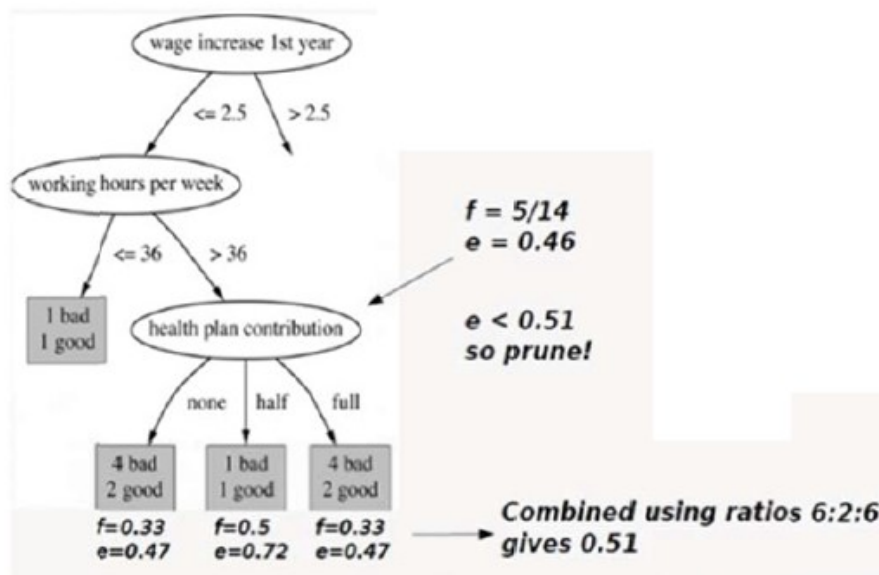
Συνήθως υπάρχουν πολλοί πιθανοί έλεγχοι που μπορούν να επιλεγούν στην παραπάνω διαδικασία. Ο C4.5 χρησιμοποιεί δύο ευρετικά (heuristic) κριτήρια για να βαθμολογήσει αυτούς τους ελέγχους και να επιλέξει τον προτιμότερο:

- Το κέρδος πληροφορίας, το οποίο ελαχιστοποιεί την συνολική εντροπία των παραγόμενων υποσυνόλων.

- Και τον λόγο κέρδους, ο οποίος προκύπτει από την διαίρεση του κέρδους πληροφορίας προς την πληροφορία που προκύπτει από τις πιθανές τιμές του ελέγχου.

Οι μεταβλητές που χρησιμοποιούνται μπορεί να είναι ονομαστικές ή αριθμητικές και αυτό καθορίζει την μορφή του ελέγχου. Για τις αριθμητικές μεταβλητές οι πιθανές μορφές είναι $\{A \leq h, A > h\}$, όπου A είναι η μεταβλητή και h ένα όριο που προκύπτει ταξινομώντας το σύνολο των υποδειγμάτων με βάση τις τιμές του A και επιλέγοντας το όριο μεταξύ δύο διαδοχικών τιμών που μεγιστοποιεί τα παραπάνω κριτήρια. Για τις μεταβλητές με διακριτές τιμές, υπάρχει εξ ορισμού ένα αποτέλεσμα από τον έλεγχο για κάθε διαφορετική τιμή του A . Ωστόσο υπάρχει και η δυνατότητα ομαδοποίησης των διαφορετικών τιμών του A σε υποσύνολα, επιτρέποντας έτσι έναν έλεγχο για κάθε υποσύνολο.

Το αρχικό δένδρο που δημιουργείται στη συνέχεια υπόκειται σε κλάδεμα για να αποφευχθεί η υπερπροσαρμογή. Το κλάδεμα γίνεται από τα φύλλα προς τη ρίζα του δένδρου. Για ένα τμήμα του δένδρου ο αλγόριθμος C4.5 βρίσκει το σταθμισμένο άθροισμα του εκτιμώμενου σφάλματος των κλαδιών και το συγκρίνει με το εκτιμώμενο σφάλμα που θα προέκυπτε αν το τμήμα αυτού του δένδρου το αντικαθιστούσε ένα φύλλο. Αν το δεύτερο δεν είναι μεγαλύτερο από το πρώτο, το τμήμα αυτό του δένδρου κλαδεύεται. Με ανάλογο τρόπο ο αλγόριθμος ελέγχει αν ένα τμήμα του δένδρου θα μπορούσε να αντικατασταθεί από ένα κλαδί αυτού του τμήματος, συγκρίνοντας το εκτιμώμενο σφάλμα. Η διαδικασία του κλαδέματος ολοκληρώνεται με ένα πέρασμα από ολόκληρο δένδρο.



Εικόνα 7: Εφαρμογή του αλγορίθμου C4.5 σε δένδρο απόφασης

Κανόνες Κατηγοροποίησης

Πολύπλοκα δένδρα αποφάσεων μπορεί να είναι αρκετά δυσνόητα πχ επειδή πληροφορίες για μια τάξη συνήθως είναι διανεμημένες σε ολόκληρο το δένδρο. Ο C4.5 εισήγαγε μια εναλλακτική διατύπωση αποτελούμενη από λίστες κανόνων της μορφής:

- IF A and B and C.... then class X

όπου κανόνες για κάθε τάξη ομαδοποιούνται. Ένα υπόδειγμα ταξινομείται βρίσκοντας τον πρώτο κανόνα που οι συνθήκες του ικανοποιούνται. Αν δεν βρεθεί κανένας κανόνας, τότε το υπόδειγμα τοποθετείται στην προεπιλεγμένη κλάση.

Οι κανόνες του C4.5 προκύπτουν από το αρχικό δένδρο αποφάσεων, πριν υποστεί κλάδεμα. Κάθε διαδρομή από τη βάση του δένδρου ως κάποιο φύλλο γίνεται ένας πρωτότυπος κανόνας του οποίου οι συνθήκες είναι τα αποτελέσματα ελέγχων κατά μήκος της διαδρομής και του οποίου η κλάση είναι το όνομα του φύλλου. Στην συνέχεια ο κανόνας απλοποιείται, καθορίζοντας την επίδραση της απόρριψης κάθε συνθήκης κατά σειρά. Η απόρριψη μιας συνθήκης μπορεί να αυξήσει τον αριθμό των υποδειγμάτων που καλύπτονται από τον κανόνα ή τον αριθμό των υποδειγμάτων που δεν ανήκουν στην κλάση που υποδεικνύεται από τον κανόνα ή να μειώσει το εκτιμώμενο σφάλμα του κανόνα. Ένας αλγόριθμος αναρρίχησης λόφων (hill-climbing) χρησιμοποιείται για την απόρριψη συνθηκών από τον κανόνα μέχρι να βρεθεί το μικρότερο δυνατό σφάλμα. Για την ολοκλήρωση της διαδικασίας, ένα υποσύνολο απλοποιημένων κανόνων επιλέγεται για κάθε κλάση κάθε φορά. Αυτά τα υποσύνολα κατατάσσονται για την ελαχιστοποίηση του σφάλματος στα δεδομένα εκπαίδευσης και επιλέγεται μια τάξη που ορίζεται ως η προεπιλεγμένη. Το τελικό σύνολο περιέχει πολύ λιγότερους κανόνες από τον αριθμό των φύλλων στο δένδρο μετά το κλάδεμα.

Το κυριότερο μειονέκτημα των κανόνων κατηγοροποίησης του C4.5 είναι η ποσότητα του υπολογιστικού χρόνου και της χωρητικότητας που απαιτούνται. Σε πείραμα, επιλέχθηκαν δείγματα που αποτελούνταν από 10000 ως 100000

υποδείγματα. Για τα δένδρα αποφάσεων, η μετακίνηση από τα δείγματα σε αυτά των 100000 υποδειγμάτων προκάλεσε αύξηση του υπολογιστικού χρόνου από 1,4 σε 61 δευτερόλεπτα, ένα συντελεστή της τάξης του 44. Ο χρόνος για την αντίστοιχη διαδικασία με σύνολα κανόνων αυξήθηκε από 32 σε 9.715 δευτερόλεπτα, συντελεστής της τάξης του 300 [1].

Δεδομένου ενός συνόλου υποδειγμάτων, το καθένα από τα οποία ανήκει σε μία γνωστή κλάση και έχει ένα γνωστό διάνυσμα χαρακτηριστικών, σκοπός μας είναι η δημιουργία ενός κανόνα που θα μας επιτρέψει την ανάθεση μελλοντικών υποδειγμάτων σε κάποια κλάση, γνωρίζοντας μόνο το διάνυσμα των χαρακτηριστικών του μελλοντικού υποδείγματος. Αυτού του είδους τα προβλήματα χαρακτηρίζονται προβλήματα ταξινόμησης υπό επίβλεψη, συναντώνται πολύ συχνά και έχουν οδηγήσει στην κατασκευή πολλών μεθόδων για την δημιουργία τέτοιων κανόνων. Μια πολύ σημαντική τέτοια μέθοδος είναι και ο απλοϊκός (naive) Bayes. Η μέθοδος αυτή είναι πολύ εύκολη στην κατασκευή, καθώς δεν απαιτεί ιδιαίτερα εξελιγμένα σχήματα επαναληπτικής εκτίμησης παραμέτρων. Είναι εύκολο να ερμηνευτεί, ώστε χρήστες χωρίς ιδιαίτερη γνώση σε ότι αφορά τους ταξινομητές να είναι εύκολο να κατανοήσουν για ποιο λόγο γίνεται μια συγκεκριμένη κατηγοροποίηση. Επίσης λόγω ότι έχει καλή απόδοση, μπορεί να βασιστεί κάποιος πάνω της λόγω της σταθερότητας της.

Θεωρούμε την ύπαρξη δύο τάξεων που ονομάζεται $i=0,1$. Στόχος μας είναι η χρήση του αρχικού συνόλου υποδειγμάτων με γνώση των τάξεων στις οποίες ανήκουν (το σύνολο εκπαίδευσης) για την δημιουργία μιας βαθμονόμησης ώστε υποδείγματα με μεγάλη βαθμολογία να σχετίζονται με τα υποδείγματα της κλάσης 1 και υποδείγματα με μικρή βαθμολογία να σχετίζονται με τα υποδείγματα της τάξης 0. Η κατηγοροποίηση στην συνέχεια επιτυγχάνεται συγκρίνοντας τη βαθμολογία με ένα όριο t . Αν ορίσουμε $P(i|x)$ την πιθανότητα ένα υπόδειγμα με διάνυσμα μέτρησης $x=(x_1, x_2, \dots, x_p)$ να ανήκει στην τάξη i , τότε οποιαδήποτε μονοτονική συνάρτηση του $P(i|x)$ θα ήταν μια κατάλληλη βαθμονόμηση. Συγκεκριμένα ο λόγος $P(1|x)/P(0|x)$ θα ήταν κατάλληλος. Από την θεωρία πιθανοτήτων γνωρίζουμε μπορούμε να αποσυνθέσουμε το $P(i|x)$ στο γινόμενο $f(x|i)P(i)$, όπου $f(x|i)$ είναι η κατανομή υπό συνθήκες του x στα υποδείγματα της κλάσης i και $P(i)$ η πιθανότητα ένα υπόδειγμα να ανήκει στην τάξη i αν δεν γνωρίζουμε τίποτα παραπάνω γι αυτό (η εκ των προτέρων πιθανότητα - a priori). Άρα ο λόγος γίνεται $P(1|x)/P(0|x)=f(x|1)P(1)/f(x|0)P(0)$. Για να το χρησιμοποιήσουμε αυτό για την ταξινόμηση, πρέπει να εκτιμήσουμε τα $f(x|i)$ και $P(i)$. Αν το σύνολο εκπαίδευσης ήταν ένα τυχαίο δείγμα από το συνολικό πληθυσμό, το $P(i)$ θα μπορούσε να εκτιμηθεί άμεσα από το ποσοστό των υποδειγμάτων της τάξης i στο σύνολο εκπαίδευσης. Για να εκτιμήσει το $f(x|i)$, η μέθοδος Naive Bayes υποθέτει τα στοιχεία του x είναι ανεξάρτητα, με $f(x|i)=\prod_{j=1}^p f(x_j|i)$ και στην συνέχεια εκτιμά κάθε μια από τις κατανομές μιας μεταβλητής $f(x_j|i)$, $j=1, \dots, p$, $i=0,1$ ξεχωριστά. Άρα το πρόβλημα P διατάσεων πολλών μεταβλητών έχει αναχθεί σε P προβλήματα εκτίμησης μιας μεταβλητής. Η εκτίμηση μας μεταβλητής είναι πιο απλή και απαιτεί μικρότερου μεγέθους σύνολα εκπαίδευσης για να επιτύχει ακριβείς εκτιμήσεις. Αυτό είναι ένα από τα μοναδικά χαρακτηριστικά της μεθόδου Naive Bayes: η εκτίμηση είναι απλή, γρήγορη και δεν απαιτεί εξελιγμένα σχήματα επαναληπτικής εκτίμησης.

Το μοντέλα που προκύπτουν από τον Naive Bayes είναι πολύ ελκυστικά λόγω της απλότητας, της κομψότητας αλλά και της σταθερότητας τους. Πρόκειται για έναν από τους παλιούς αλγορίθμους κατηγοροποίησης, που ακόμα και

στην πιο απλή του μορφή παρουσιάζει ανεπάντεχα υψηλή απόδοση. Χρησιμοποιείται ευρέως σε περιπτώσεις ταξινόμησης κειμένου και φιλτραρίσματος κακόβουλης αλληλογραφίας [1], [9], [11].

3.4 Αλγόριθμοι Συσχέτισης

Αλγόριθμος Apriori

Μία από πιο συχνές εφαρμογές data mining είναι η αναγνώριση συνόλων στοιχείων που συναντώνται μαζί σε δεδομένα συναλλαγών και η εξαγωγή αντίστοιχων κανόνων συσχέτισης. Η εύρεση τέτοιων συνόλων στοιχείων με μεγάλη συχνότητα (με τον όρο συχνότητα εννοούμε το να πληρούν ένα κριτήριο ελάχιστης υποστήριξης που θα έχει θέσει ο χρήστης) δεν είναι καθόλου ασήμαντη λόγω της συνδυαστικής της δυναμικής. Μόλις βρεθούν τέτοια σύνολα στοιχείων, είναι άμεση η μετάβαση στη δημιουργία κανόνων συσχέτισης με βαθμό εμπιστοσύνης ίσο ή μεγαλύτερο από αυτό που έχει θέσει ο χρήστης.

Ο Apriori είναι ένας πρωτότυπος αλγόριθμος για την εύρεση συνόλων στοιχείων με μεγάλη συχνότητα χρησιμοποιώντας μία γεννήτρια υποψηφίων. Χρησιμοποιεί κατά την εξερεύνηση των υποδειγμάτων την ιδέα πως αν ένα σύνολο στοιχείων δεν είναι "συχνό" τότε και κανένα υπερσύνολο αυτού δεν θα είναι. Συμβατικά ο Apriori υποθέτει ότι τα υποδείγματα σε ένα σετ δεδομένων είναι ταξινομημένα αλφαβητικά. Έστω πως το συχνό σύνολο k στοιχείων είναι F_k και τα υποψήφια C_k . Ο αλγόριθμος πρώτα εξερευνά την βάση δεδομένων και αναζητά συχνά σύνολα στοιχείων μεγέθους 1 (1 στοιχείο δηλαδή αποτελεί το σύνολο) αθροίζοντας τις εμφανίσεις κάθε στοιχείο και συγκεντρώνοντας εκείνα που ικανοποιούν μια ελάχιστη υποστήριξη. Στην συνέχεια επαναλαμβάνει τα επόμενα 3 βήματα και εξάγει όλα τα συχνά σύνολα στοιχείων:

1. Παράγει C_{k+1} υποψήφια συχνά σύνολα στοιχείων μεγέθους $k+1$ από τα συχνά σύνολα στοιχείων μεγέθους k .
2. Εξερευνά την βάση δεδομένων και υπολογίζει την υποστήριξη κάθε υποψηφίου συχνού συνόλου στοιχείων.
3. Προσθέτει αυτά τα σύνολα στοιχείων που ικανοποιούν την απαιτούμενη ελάχιστη υποστήριξη στο F_{k+1} .

Input: A database D of transactions and $minsup$.
Output: A set F that contains all frequent itemsets of D .
Algorithm:

```

( 1)       $F_1 \leftarrow \{\text{frequent 1-itemsets}\};$ 
( 2)       $k \leftarrow 2;$ 
( 3)      while  $F_{k-1} \neq \phi$  do begin
( 4)           $C_k \leftarrow \text{AprioriGen}(F_{k-1});$ 
( 5)          forall transactions  $t \in D$  do begin
( 6)               $C_t \leftarrow \text{Subset}(C_k, t);$ 
( 7)              forall candidates  $c \in C_t$  do
( 8)                   $c.count \leftarrow c.count + 1;$ 
( 9)          end;
(10)        $F_k \leftarrow \{c \in C_k | c.count \geq minsup\};$ 
(11)        $k \leftarrow k + 1;$ 
(12)       end;
(13)        $F \leftarrow \cup_{k \geq 1} F_k;$ 

```

Εικόνα 8: Ο αλγόριθμος Apriori

Ο αλγόριθμος φαίνεται στην παραπάνω εικόνα 8. Η συνάρτηση AprioriGen που εμφανίζεται στην 4η γραμμή παράγει C_{k+1} υποψήφιους συνδυασμούς από το σύνολο F_k μέσω των επόμενων δύο βημάτων:

1. Βήμα σύζευξης: Παραγωγή R_{k+1} , τα αρχικά υποψήφια συχνά σύνολα στοιχείων μεγέθους $k+1$, μέσω της ένωσης δύο συχνών συνόλων στοιχείων μεγέθους k , P_k και Q_k , που έχουν τα πρώτα $k-1$ στοιχεία κοινά.

$R_{k+1} = P_k \cup Q_k = \{\text{στοιχείο1}, \text{στοιχείο2}, \dots, \text{στοιχείο}k-1, \text{στοιχείο}k, \text{στοιχείο}k'\}$

$P_k = \{\text{στοιχείο}, \dots, \text{στοιχείο}k-1, \text{στοιχείο}k\}$

$Q_k = \{\text{στοιχείο}, \dots, \text{στοιχείο}k-1, \text{στοιχείο}k'\}$

όπου $\text{στοιχείο } 1 < \text{στοιχείο } 2 < \dots < \text{στοιχείο } k < \text{στοιχείο } k'$.

2. Βήμα κλαδέματος: έλεγχος αν όλα τα σύνολα στοιχείων μεγέθους k στο R_{k+1} μπορούν να θεωρηθούν συχνά και παραγωγή του C_{k+1} αφαιρώντας όσα από αυτά δεν ικανοποιούν τις απαιτήσεις από το R_{k+1} . Η λογική πίσω από αυτό είναι ότι όποιο σύνολο μεγέθους k στο C_{k+1} δεν θεωρείται συχνό, δεν μπορεί και να αποτελεί υποσύνολο ενός συχνού συνόλου στοιχείων μεγέθους $k+1$.

Ο αλγόριθμος σαρώνει την βάση δεδομένων το πολύ $k_{max}-1$ φορές όταν το μέγιστο μέγεθος των συχνών συνόλων στοιχείων έχει οριστεί ως k_{max} . Μέσω της μείωσης του μεγέθους των υποψηφίων συνόλων ο Apriori πετυχαίνει καλή απόδοση. Ωστόσο σε περιπτώσεις με πάρα πολλά σύνολα συχνών στοιχείων, σύνολα μεγάλου μεγέθους ή με ορισμό πολύ χαμηλής ελάχιστης υποστήριξης, υποφέρει από το τίμημα της παραγωγής πολύ μεγάλου αριθμού υποψηφίων συνόλων και της επανειλημμένης σάρωσης της βάσης δεδομένων για τον έλεγχο όλων αυτών. Στην πράξη, η παραγωγή ενός συνόλου συχνών στοιχείων μεγέθους 100 συνεπάγεται την δημιουργία 2^{100} υποψηφίων συνόλων στοιχείων.

Στην επόμενη υποενότητα θα γίνει μια σύντομη περίληψη των πλεονεκτημάτων και μειονεκτημάτων του κάθε αλγορίθμου όπου αναφέρθηκε πιο πάνω.

3.5 Πλεονεκτήματα και Μειονεκτήματα Αλγορίθμων

Πριν προχωρήσουμε αναλυτικά στα πλεονεκτήματα και τα μειονεκτήματα του κάθε έναν αλγορίθμου θα αναφέρουμε ότι οι μη ιεραρχικές τεχνικές συσταδοποίησης, απαιτούν γνώση για τον αριθμό των cluster, σε αντίθεση με τις ιεραρχικές μεθόδους, όπου δεν απαιτείται μια εκ των προτέρων γνώση του αριθμού των cluster ή της αρχικής διαμέρισης. Αυτό είναι το πλεονέκτημα των ιεραρχικών μεθόδων, έναντι των μη ιεραρχικών, καθώς επίσης κι ότι είναι υπολογιστικά γρηγορότερες. Όμως οι ιεραρχικές έχουν ένα μειονέκτημα, ότι άπαξ και μία παρατήρηση ανατεθεί σε ένα cluster, δεν μπορεί να ανατεθεί εκ νέου σε καινούριο cluster.

Αλγόριθμος k-means

Πλεονεκτήματα

- Είναι δημοφιλής εξαιτίας της απλότητας υλοποίησης του και της πολυπλοκότητας του η οποία είναι της τάξης n ($O(n)$) , όπου n είναι ο αριθμός των στοιχείων.
- Ο αλγόριθμος αυτός υπερτερεί σε περιπτώσεις όπου τα δεδομένα είναι πάρα πολλά και η κατασκευή δενδρογραμμάτων είναι αδύνατη.

Μειονεκτήματα

- Αρχική επιλογή των κεντροειδών. Αν η επιλογή αυτή δεν είναι αρκετά προσεκτική τότε το κριτήριο του τετραγωνικού λάθους συγκλίνει σε τοπικά ελάχιστο κάνοντας την τελική επιλογή cluster ανεπιτυχή.
- Ευαισθησία σε θόρυβο (outliers).
- Έχει προβλήματα όταν οι ομάδες είναι διαφορετικού μεγέθους, πυκνότητας, μη-σφαιρικού σχήματος.

Αλγόριθμος k-medoid (PAM –Partitioning Around Medoids)

Πλεονεκτήματα

- Μειώνει την ευαισθησία σε θόρυβο (outlier).
- Μπορεί να εφαρμοστεί σε δεδομένα οποιουδήποτε τύπου (π.χ. κατηγορικά δεδομένα)

Μειονεκτήματα

- Η πολυπλοκότητα κάθε επανάληψης είναι $O(k(n-k)^2)$. Για μεγάλους αριθμούς n, k ο υπολογισμός γίνεται δαπανηρός και μάλιστα πιο δαπανηρός από τον αλγόριθμο k-means.

Τόσο ο k-means όσο και ο k-medoids απαιτούν από τον χρήστη τον αριθμό k , δηλαδή τον αριθμό των κεντροειδών.

Αλγόριθμος εξαντλητικής αναζήτησης

Πλεονεκτήματα

- Εγγυάται ότι θα δημιουργήσει μία βέλτιστη περίληψη ενός δεδομένου μεγέθους L για ένα δεδομένο σύνολο συναλλαγών (transactions) T .

Μειονεκτήματα

- Απαιτεί αναζήτηση σε χώρο $(2^{2^{|T|}})$. Για παράδειγμα για 4 συναλλαγές (transactions) θα χρειαστεί αναζήτηση σε 65,536 υποσύνολα, το οποίο το καθιστά υπολογιστικά ανέφικτο ακόμη και για πολύ μικρά σύνολα δεδομένων.

Αλγόριθμος απλού συνδέσμου

Πλεονεκτήματα

- Τα προτερήματα αυτής της μεθόδου είναι ότι δημιουργούνται συνεχόμενες συστάδες,
- Μπορεί να χειριστεί μη ελλειπτικά σχήματα.

Μειονεκτήματα

- Το βασικό μειονέκτημα είναι η ευαισθησία στον θόρυβο και στις ακραίες τιμές (outliers)

Αλγόριθμος πλήρους συνδέσμου

Πλεονεκτήματα

- Το βασικό πλεονέκτημα αυτού του τρόπου σύνδεσης είναι η μικρή ευαισθησία στον θόρυβο και στις ακραίες τιμές (outliers).

Μειονεκτήματα

- Τείνει να διασπά μεγάλες συστάδες και ότι οδηγεί, συνήθως, σε κυκλικά σχήματα.

Αλγόριθμος μέσου όρου συστάδων

Πλεονεκτήματα

- Έχει μικρότερη ευαισθησία σε θόρυβο και σε ακραίες τιμές (outliers), αλλά ευνοεί τις συστάδες με κυκλικό σχήμα

Μειονεκτήματα

- Το πρόβλημα με αυτή την απόσταση είναι ότι δεν έχει μονότονη αύξηση. Έτσι, δύο συστάδες που συγχωνεύονται μπορεί να έχουν μικρότερη απόσταση από συστάδες, οι οποίες έχουν συγχωνευτεί σε προηγούμενα βήματα.

Αλγόριθμος Ward

Πλεονεκτήματα

- Ελαχιστοποίηση της μεταβλητότητας μεταξύ δύο εξεταζόμενων ομάδων

Μειονεκτήματα

- Σχηματίζει ομάδες πολύ μικρού μεγέθους

Αλγόριθμος DBScan (Density-Based Spatial Clustering of Applications with Noise)

Πλεονεκτήματα

- Κατάλληλος για ομάδες που έχουν υψηλή πυκνότητα σημείων , οι οποίες μπορεί να είναι διαχωρισμένες από άλλα σημεία (θόρυβος) χαμηλότερης πυκνότητας.
- Δεν επηρεάζεται από τον θόρυβο και μπορεί να χειριστεί συστάδες με διαφορετικά σχήματα και μεγέθη.

Μειονεκτήματα

- Πρόβλημα με διαφορετικές πυκνότητες.
- Ευαισθησία στις παραμέτρους.
- Πολυ-διάστατα δεδομένα: δύσκολος ορισμός πυκνότητας και δαπανηρός υπολογισμός γειτόνων.

Αλγόριθμος C4.5

Πλεονεκτήματα

- Δέχεται ως μεταβλητές εισόδου και ονομαστικά γνωρίσματα και γνωρίσματα με αριθμητικές τιμές.
- Μπορεί να χειριστεί δεδομένα με χαμένες τιμές.
- Διαθέτει ένα εξαιρετικά γρήγορο αλγόριθμο εκπαίδευσης.

Μειονεκτήματα

- Η ποσότητα του υπολογιστικού χρόνου και της χωρητικότητας που απαιτούνται.
- Λειτουργεί μόνο για διακριτές και όχι για συνεχόμενες τιμές κλάσης.
- Απαιτεί την εγκατάσταση ολόκληρου του δείγματος εκπαίδευσης στην κύρια μνήμη του υπολογιστή, κάτι που προκαλεί προβλήματα στον χειρισμό εξαιρετικά μεγάλων δειγμάτων.

Αλγόριθμος Apriori

Πλεονεκτήματα

- Μέσω της μείωσης του μεγέθους των υποψηφίων συνόλων πετυχαίνει καλή απόδοση.
- Εκμεταλλεύεται την ιδιότητα των συχνών itemsets.
- Υλοποιείται εύκολα (και σε παράλληλη μορφή)

Μειονεκτήματα

- Υποθέτει ότι η βάση των δοσοληψιών βρίσκεται στη μνήμη.
- Απαιτεί μέχρι και m σαρώσεις της βάσης (m το πλήθος των items).

Μοντέλα που βασίζονται στην μέθοδο Bayes

Πλεονεκτήματα

- Η εκτίμηση είναι απλή στην κατασκευή, γρήγορη και δεν απαιτεί εξελιγμένα σχήματα επαναληπτικής εκτίμησης παραμέτρων.
- Επίσης λόγω ότι έχει καλή απόδοση, μπορεί να βασιστεί κάποιος πάνω της λόγω της σταθερότητας της.
- Μπορούν να χειριστούν και αριθμητικές και ονομαστικές μεταβλητές.
- Το μοντέλο που προκύπτουν από τον Naive Bayes είναι πολύ ελκυστικά λόγω της απλότητας, της κομψότητας.

Μειονεκτήματα

- Το σημαντικότερο μειονέκτημα τους είναι το γεγονός ότι δεν υπάρχει ένας καθιερωμένος και γενικά αποδεκτός τρόπος εξαγωγής του γράφου από τα δεδομένα.
- Για τον υπολογισμό των πιθανοτήτων ενός κλάδου του δικτύου απαιτείται ο υπολογισμός όλων των άλλων κλάδων επιφέροντας σημαντικό υπολογιστικό κόστος.

Κεφάλαιο 4

4.1 Δημιουργία και Διατήρηση Συνόψεων

Ο μεγάλος όγκος ροών δεδομένων δημιουργεί μοναδικούς περιορισμούς χρόνου και χρόνου στην επεξεργασία των δεδομένων. Σε πολλές επεξεργασίες ερωτήσεων, οι αλγόριθμοι απαιτούν αποτελεσματική εκτέλεση που μπορεί να είναι δύσκολο να επιτευχθεί με μια γρήγορη ροή δεδομένων. Τα τελευταία χρόνια μια σειρά αλγορίθμων για την εξαγωγή συνόψεων έχουν αναπτυχθεί, οι οποίοι μπορούν να χρησιμοποιηθούν σε συνδυασμό με μια ποικιλία εξορύξεων και τεχνικών επεξεργασίας ερωτημάτων στην επεξεργασία ροών δεδομένων. Κάποια βασικές μέθοδοι περιλαμβάνουν τις μεθόδους δειγματοληψίας, wavelets και ιστογράμματα. Σε αυτό κεφάλαιο, θα παράσχουμε μια επισκόπηση των βασικών τεχνικών των συνόψεων και της εξόρυξης τεχνικών που υποστηρίζονται από τέτοιες μεθόδους και στην συνέχεια θα αναλύσουμε τον αλγόριθμό μας.

Θα μπορούσαμε να δώσουμε τον εξής ορισμό για την σύνοψη: Μια συμπιεσμένη δομή δεδομένων που έχει σχεδιαστεί για να μειώνει το μέγεθος του χώρου αποθήκευσης εισόδου δεδομένων (π.χ. σε μια βάση δεδομένων) προκειμένου να επιτευχθεί αποδοτικότητα κατά την επεξεργασία δεδομένων.

Οι ροές δεδομένων αποτελούν μοναδική πρόκληση σε πολλές βάσεις δεδομένων λόγω του υπολογιστικού και αποθηκευτικού κόστους που συνδέονται με τη μεγάλη ποσότητα της ροής δεδομένων. Σε πολλές περιπτώσεις, οι δομές δεδομένων και οι στατιστικές συνόψεων μπορούν να κατασκευαστούν από ροές που είναι χρήσιμες για μια ποικιλία από εφαρμογών. Μερικά παραδείγματα τέτοιων εφαρμογών είναι τα εξής:

Εκτίμηση κατά προσέγγιση ερωτήματος:

Το πρόβλημα της εκτίμησης ερωτήματος είναι ενδεχομένως η πιο ευρέως χρησιμοποιούμενη εφαρμογή των συνόψεων [56]. Το πρόβλημα είναι ιδιαίτερα σημαντικό από άποψη απόδοσης, δεδομένου ότι τα ερωτήματα συνήθως πρέπει να απαντηθούν σε πραγματικό χρόνο. Ως εκ τούτου, οι περισσότερες συνόψεις όπως η δειγματοληψία, τα ιστογράμματα, τα wavelets είναι συνήθως σχεδιασμένα για να είναι σε θέση να λύσουν το πρόβλημα εκτίμησης ερωτήματος.

Ο σχεδιασμός και η επιλογή μιας συγκεκριμένης μεθόδου συνόψεως εξαρτάται από το πρόβλημα. Επομένως, η σύνοψη πρέπει να κατασκευαστεί σε τρόπο που είναι φιλικό προς τον χρήστη.

Για παράδειγμα, μια δομή συνόψεως που χρησιμοποιείται για την εκτίμηση ερωτήματος είναι πιθανό να είναι διαφορετική από τη δομή της σύνοψης που χρησιμοποιείται για προβλήματα εξόρυξης δεδομένων όπως η αλλαγή ανίχνευσης και ταξινόμησης. Σε γενικές γραμμές, θα θέλαμε να κατασκευάσουμε τη σύνοψη κατά τρόπον ώστε να έχει ευρεία εφαρμογή σε ευρείες κλάσεις και προβλήματα [15].

Συγκεκριμένα, τα επιθυμητά για μία αποτελεσματική κατασκευή σύνοψης είναι οι εξής:

Ευρεία εφαρμογή:

Δεδομένου ότι οι δομές συνόψεων χρησιμοποιούνται για μια ποικιλία των εφαρμογών εξόρυξης δεδομένων, είναι επιθυμητό να έχουν ευρεία δυνατότητα εφαρμογής. Αυτό συμβαίνει επειδή κάποιος μπορεί να επιθυμεί να χρησιμοποιήσει τη ροή δεδομένων για πολλές διαφορετικές εφαρμογές.

Περιορισμοί one Pass:

Δεδομένου ότι οι ροές δεδομένων περιέχουν συνήθως έναν μεγάλο αριθμό σημείων, το περιεχόμενο της ροής δεν μπορεί να εξεταστεί περισσότερο από μία φορά κατά τη διάρκεια του υπολογισμού. Ως εκ τούτου, όλα οι αλγόριθμοι συνόψεων σχεδιάζονται με έναν περιορισμό ενός περάσματος.

Χρόνος και απόδοση:

Σε πολλές παραδοσιακές μεθόδους σύνοψης τα στατικά σύνολα δεδομένων (όπως ιστογράμματα), οι μεθοδολογίες απαιτούν υπερ-γραμμικό χώρο και χρόνο. Αυτό δεν είναι αποδεκτό για μια ροή δεδομένων.

Εξέλιξη:

Οι ροές δεδομένων σπάνια παρουσιάζουν σταθερές κατανομές, αλλά γρήγορα εξελίσσονται με την πάροδο του χρόνου. Οι μέθοδοι συλλογής στατικών συνόλων δεδομένων δεν έχουν σχεδιαστεί για να αντιμετωπίσουν την ταχεία εξέλιξη μιας ροής δεδομένων. Για αυτό στόχος, μέθοδοι όπως η ομαδοποίηση [15] χρησιμοποιούνται για τους σκοπούς της σύνοψης όπως καθώς η κατηγοροποίηση [16].

Υπάρχει μια ποικιλία τεχνικών που μπορούν να χρησιμοποιηθούν για την κατασκευή των συνόψεων σε ροές δεδομένων. Συνοψίζουμε αυτές τις μεθόδους παρακάτω:

Ιστογράμματα (Histograms):

Οι μέθοδοι που βασίζονται σε ιστογράμματα χρησιμοποιούνται ευρέως για στατικά σύνολα δεδομένων. Ωστόσο, οι πιο παραδοσιακοί αλγόριθμοι στα σύνολα στατικών δεδομένων απαιτούν υπερ-γραμμικό χρόνο και χώρο. Αυτό οφείλεται στη χρήση του δυναμικού προγραμματισμού των τεχνικών για τη βέλτιστη κατασκευή ιστογραμμάτων. Η προέκτασή τους στην περίπτωση ροής δεδομένων είναι ένα δύσκολο έργο. Ένας αριθμός πρόσφατων τεχνικών [54] συζητούν το σχεδιασμό των ιστογραμμάτων για τη δυναμική περίπτωση.

Σύνοψη βασισμένη σε μικρο-συστάδες:

Μια πρόσφατη μέθοδος μικροσυσσωμάτωσης [15] μπορεί να χρησιμοποιηθεί για να εκτελεστούν συνοπτικές κατασκευές ροών δεδομένων. Το πλεονέκτημα της περίληψης μικρο-συστάδων είναι ότι ισχύει για τη πολυδιάστατη περίπτωση και προσαρμόζεται καλά στην πορεία των ροών δεδομένων. Στην παρούσα εργασία θα δημιουργηθεί ένας αλγόριθμος βασισμένος σε μικρο-συστάδες (micro-cluster).

4.2 Επιλογή μεθόδου και αλγορίθμου

Ως συνέπεια των παραπάνω και μελετώντας τα [14], [15], [16] οι εφαρμογές εξόρυξης δεδομένων δεν απαιτούν τη χρήση μεμονωμένων σημείων δεδομένων αλλά απαιτούν μόνο μια προσωρινή σύνοψη που παρέχει μια γενική εικόνα της συμπεριφοράς των ροών δεδομένων (data stream). Μέθοδοι όπως η συσταδοποίηση μπορούν να χρησιμοποιηθούν για την αποτελεσματική ανίχνευση αλλαγών στις ροές δεδομένων.

Μία μέθοδος που μπορεί να χρησιμοποιηθεί για την εκτέλεση συνόψεων ροών δεδομένων είναι η σύνοψη βασισμένη σε μικρο - συστάδες (micro-cluster) [15 σελίδα 2]. Το πλεονέκτημα χρησιμοποίησης μικρο – συστάδων είναι ότι εφαρμόζεται ή καλύτερα ταιριάζει στην πολυδιάστατη περίπτωση και προσαρμόζεται στην εξέλιξη των ροών δεδομένων.

Με άλλα λόγια οι μικρο - συστάδες (micro-cluster) είναι μία τεχνική που υιοθετείται στην ομαδοποίηση των ροών δεδομένων. Είναι μια χρονική επέκταση της δυνατότητας ομαδοποίησης, η οποία συμπιέζει αποτελεσματικά τα δεδομένα.

Μπορούμε να ορίσουμε τα χαρακτηριστικά της συστάδας και την μικρο - συστάδα ως εξής: Τα χαρακτηριστικά μιας συστάδας (Cluster Feature CF) είναι μία τριπλή σύνοψη, η οποία περιλαμβάνει το πλήθος (count) των δεδομένων, το άθροισμα των δεδομένων και το τετραγωνικό άθροισμα αυτών. Σύμφωνα με τα [15], [17] μία μικρο – συστάδα είναι ένα σύμπλεγμα για ένα σύνολο 2-διαστάσεων σημείων $X_{i1}, \dots, X_{in}$ με CF1, CF2 και n. Μια παραλλαγή του micro-cluster στην δική μας περίπτωση είναι : όπου CF1 είναι το άθροισμα των τιμών της κάθε στήλης ενός πίνακα, CF2 είναι το άθροισμα των τετραγώνων για κάθε στήλη του κάθε πίνακα ενώ n είναι το πλήθος των δεδομένων. Όπως ορίζεται και στο [17] για την τεχνική BIRCH (balanced iterative reducing and clustering using hierarchies) θα χρησιμοποιήσουμε για τα πειράματά μας την Centroid μέθοδο $C=CF1/n$.

4.3 Δημιουργία συνόψεων και επιλογή μετρικών

Τα CF θα είναι ουσιαστικά μαθηματικοί υπολογισμοί για κάθε στήλη του κάθε πίνακα. Στην δική μας περίπτωση έχουμε χρησιμοποιήσει το συνολικό άθροισμα για κάθε στήλη του κάθε πίνακα.

Στα πειράματά μας χρησιμοποιήθηκαν 5 πίνακες όπου ο καθένας έχει 5 στήλες. Για κάθε στήλη του κάθε πίνακα υπολογίζεται αρχικά το συνολικό άθροισμά της καθώς και το πλήθος των δεδομένων της.

Αυτές θα είναι οι συνόψεις μας και τα αποτελέσματα αυτά θα μας βοηθήσουν στην συνέχεια να υπολογίσουμε τις μετρικές που θα αναφερθούν παρακάτω.

Όσον αφορά τις μετρικές αποστάσεων αυτές που θα χρησιμοποιήσουμε για τις μικρο-συστάδες (micro-cluster) θα είναι η Jaccard, Sorensen ή Bray-Curtis και η Kulczynski ή QSK(Quantitative symmetric dissimilarity) [18]. Πιο κάτω αναλύεται ο τύπος της καθεμίας και πως ακριβώς χρησιμοποιήθηκε στην περίπτωσή μας.

- **Jaccard**, σύμφωνα με [18 σελίδα 4] ο τύπος του dissimilarity είναι

$$JD_{ih} = \frac{2 \sum_{j=1}^p a_{ij} - a_{hj}}{\sum_{j=1}^p a_{ij} + \sum_{j=1}^p a_{hj} + \sum_{j=1}^p (a_{ij} - a_{hj})}$$

Εικόνα 9: Τύπος JACCARD

Στον αριθμητή το a_{ij} θα είναι δεδομένα για τα οποία θέλουμε να υπολογίσουμε την ομοιότητα με τις υπάρχουσες συνόψεις και το a_{hj} θα είναι η σύνοψη κάθε στήλης του πίνακα.

Στον παρανομαστή αρχικά θα είναι το άθροισμα των δεδομένων συν το άθροισμα της κάθε σύνοψης και συν το άθροισμα των δεδομένων μείον της κάθε σύνοψης, δηλαδή ότι είναι στον αριθμητή αλλά όχι πολλαπλασιασμένο με το 2.

- Σχετική **Sorensen** ή Bray-Curtis.

Η μετρική ανομοιότητα Bray-Curtis παρέχει ισχυρά και αξιόπιστα αποτελέσματα για ένα ευρύ φάσμα εφαρμογών. Είναι μια από τις πιο συχνά εφαρμοζόμενες μετρήσεις για την έκφραση σχέσεων στην οικολογία, τις περιβαλλοντικές επιστήμες και τους συναφείς τομείς.

Ο Bray-Curtis είναι μια τροποποιημένη μέτρηση της μετρικής Μανχάταν, όπου οι αθροισμένες διαφορές μεταξύ των μεταβλητών τυποποιούνται από τις αθροιστικές μεταβλητές των αντικειμένων.

Σύμφωνα με [18 σελίδα 4] ο τύπος της είναι:

$$D_{ih} = \frac{\sum_{j=1}^p a_{ij} - a_{hj}}{\sum_{j=1}^p a_{ij} + \sum_{j=1}^p a_{hj}}$$

Εικόνα 10: Τύπος Sorensen

Στον αριθμητή το a_{ij} θα είναι δεδομένα για τα οποία θέλουμε να υπολογίσουμε την ομοιότητα με τις υπάρχουσες συνόψεις και το a_{hj} θα είναι η σύνοψη κάθε πίνακα. Επομένως θα έχουμε το άθροισμα δεδομένων μείον της κάθε σύνοψης.

Στον παρανομαστή θα είναι το άθροισμα των δεδομένων συν το άθροισμα της κάθε στήλης.

- **Kulczynski** ή QSK(Quantitative symmetric dissimilarity).

Σύμφωνα με [18 σελίδα 4] ο τύπος της είναι:

$$QSK_{th} = 1 - \frac{1}{2} \left[\frac{\sum_{j=1}^p \text{MIN}(a_{ij}, a_{hj})}{\sum_{j=1}^p a_{ij}} + \frac{\sum_{j=1}^p \text{MIN}(a_{ij}, a_{hj})}{\sum_{j=1}^p a_{hj}} \right]$$

Εικόνα 11: Τύπος Kulczynski

Όπου a_{ij} θα είναι δεδομένα για τα οποία θέλουμε να υπολογίσουμε την ομοιότητα με τις υπάρχουσες συνόψεις και το a_{hj} θα είναι η σύνοψη κάθε στήλης του πίνακα. Αρχικά υπολογίζεται το ελάχιστο μεταξύ δεδομένων και σύνοψης της αντίστοιχης στήλης κάθε πίνακα και προσθέτουμε τα ελάχιστα. Στην συνέχεια διαιρείται την πρώτη φορά με το άθροισμα των δεδομένων και έπειτα διαιρείται με το άθροισμα των συνόψεων. Τέλος όλος ο αριθμός πολλαπλασιάζεται με το 0,5.

Με την βοήθεια των αποτελεσμάτων των μετρικών αυτών θα μπορέσουμε να πάρουμε το μέσο όρο ώστε να γίνει η σύγκριση μεταξύ των δεδομένων και της σύνοψης. Με τον όρο σύνοψη θα εννοείται στο project ο υπολογισμός του μέσου όρου και τυπικής απόκλισης κάθε στήλη κάθε πίνακα.

Εφόσον γίνει ο υπολογισμός των μετρικών για κάθε πίνακα υπολογίζεται ο μέσος όρος των τριών δηλαδή Jaccard, Sorensen και Kulczynski .

Στην συνέχεια γίνεται η εύρεση της ελάχιστης απόστασης ώστε να εξαχθεί το τμήμα των δεδομένων όπου η εισαρχόμενη πλειάδα θα πρέπει να τοποθετηθεί. Ο στόχος μας είναι να τοποθετήσουμε όμοιες πλειάδες στο ίδιο τμήμα δεδομένων ώστε να έχουμε γνώση για τα στατιστικά χαρακτηριστικά τους. Η προσέγγιση αυτή μπορεί να βοηθήσει σε μεταγενέστερο στάδιο όπου ένα σύνολο ερωτημάτων θα πρέπει να απαντηθούν. Με την προτεινόμενη τεχνική, έχουμε τη δυνατότητα να γνωρίζουμε εκ των προτέρων το τμήμα των δεδομένων που θα μπορούσε να μας δώσει τις πιο κατάλληλες απαντήσεις για κάθε εισερχόμενο ερώτημα. Συνεπώς, μέσω του προτεινόμενου μοντέλου προσπαθούμε να μεγιστοποιήσουμε το βαθμό ικανοποίησης κάθε ερωτήματος με θετικές επιπτώσεις στην επιστροφή των αποτελεσμάτων και την υποστήριξη των τελικών χρηστών.

Κεφάλαιο 5

Πειραματική Αποτίμηση

Στο κεφάλαιο αυτό θα αναφερθούν τα λογισμικά που χρησιμοποιήθηκαν για το στήσιμο, προκειμένου να προχωρήσουμε στην υλοποίηση του αλγορίθμου.

5.1 Εγκατάσταση

Αρχικά εγκαταστάθηκε το πρόγραμμα Microsoft Visual Studio 2010 από το Microsoft Imagine όπου παρέχεται λογισμικό στους φοιτητές. Επίσης από το επίσημο Site της Oracle εγκαταστάθηκε η Oracle 11gR2 database και προκειμένου να υπάρχει επικοινωνία μεταξύ Visual Studio και Oracle από το site της Oracle, ο provider Oracle Data Access Components για Windows ODTwithODAC1120320_32bit. Συγκεκριμένα για την Oracle χρησιμοποιήθηκε η default βάση *orcl* όπως κατά την εγκατάσταση καθώς και το σχήμα *hr* με τους πίνακες του. Για την διαχείριση της βάσης και γενικότερα της Oracle χρησιμοποιήθηκε το πρόγραμμα της Oracle Sqldeveloper. Τέλος από το Machine Learning Repository (<https://archive.ics.uci.edu/ml/index.php>) χρησιμοποιήθηκε το Data Set (σύνολο δεδομένων) Air Quality. Τα λογισμικά που αναφέρθηκαν εγκαταστάθηκαν σε 64 bit συστήματα.

5.2 Παραμετροποίηση

Δημιουργήθηκε ένας πίνακας με όνομα AIR_COPY στο σχήμα HR με τα ίδια πεδία όπως αυτού του Data Set (σύνολο δεδομένων) Air Quality και στην συνέχεια έγινε «σπάσιμο» σε 5 αντίστοιχους πίνακες (AIR_1,2,3,4,5) με τα ίδια πεδία του αρχικού πίνακα AIR_COPY. Η διαδικασία αυτή έγινε με το εργαλείο της Oracle Sqldeveloper.

Πιο κάτω βλέπουμε ενδεικτικές τιμές από το dataset, όπου θα δουλέψουμε μόνο με θετικούς αριθμούς.

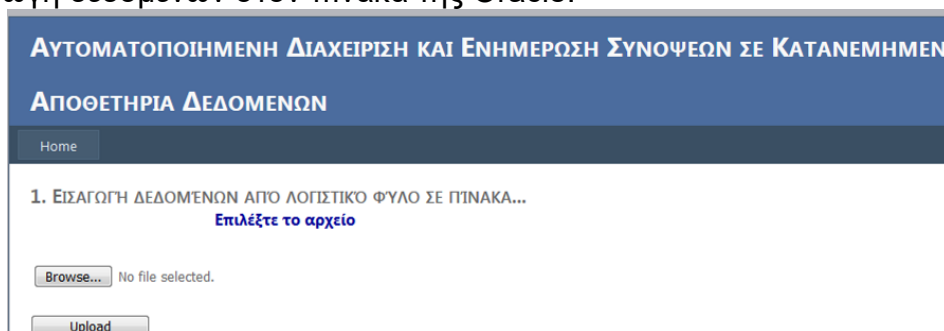
CO(GT)	NMHC(GT)	C6H6(GT)	NOx(GT)	NO2(GT)
2,6	150	11,9	166	113
2	112	9,4	103	92
2,2	88	9,0	131	114
2,2	80	9,2	172	122
1,6	51	6,5	131	116
1,2	38	4,7	89	96
1,2	31	3,6	62	77
1	31	3,3	62	76
0,9	24	2,3	45	60
0,6	19	1,7	200	200
200	14	1,3	21	34
0,7	8	1,1	16	28
0,7	16	1,6	34	48
1,1	29	3,2	98	82
2	64	8,0	174	112
2,2	87	9,5	129	101
1,7	77	6,3	112	98
1,5	43	5,0	95	92
1,6	61	5,2	104	95
1,9	63	7,3	146	112
2,9	164	11,5	207	128
2,2	79	8,8	184	126


```

και ένα Trigger για κάθε πίνακα όπου ο κώδικας είναι :
create or replace TRIGGER HR.AIR_COPY_TR
  BEFORE INSERT
  ON HR.air_copy REFERENCING OLD AS OLD NEW AS NEW
  FOR EACH ROW
BEGIN
  SELECT AIR_COPY_SEQ.nextval INTO :NEW.ID FROM dual;
END;

```

Έτσι λοιπόν αφού γίνει Build το project ανοίγει σελίδα browser και επιλέγεται το φύλλο εργασίας Air Quality.xls και στη συνέχεια πατώντας Upload γίνεται η εισαγωγή δεδομένων στον πίνακα της Oracle.



Εικόνα12: Επιλογή φύλλο εργασίας excel

5.3.2 Παραγωγή τυχαίων πλειάδων 5 αριθμών

Μετά την επιτυχή εισαγωγή δεδομένων ακολουθεί το δεύτερο μέρος όπου είναι η παραγωγή τυχαίων πλειάδων 5 αριθμών, διότι 5 είναι και τα πεδία που χρησιμοποιούνται.

2. ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΑΣ ΠΛΕΙΑΔΑΣ 5 ΑΡΙΘΜΩΝ

Δημιουργία τυχαίας πλειάδας αριθμών

CO_GT
NMHC_GT
C6H6_GT
NOX_GT
NO2_GT

Εικόνα 13: Παραγωγή τυχαίων αριθμών

Ο κώδικας παράγει τιμές σύμφωνα με το μέσο όρο και την μέση απόκλιση του κάθε πεδίου. Οι τιμές που δόθηκαν για κάθε σετ πειραμάτων ήταν οι εξής

- 1^ο σετ πειραμάτων: avg:25, stddev:10.
- 2^ο σετ πειραμάτων: avg:25, stddev:20.
- 3^ο σετ πειραμάτων: avg:50, stddev:20.
- 4^ο σετ πειραμάτων: avg:50, stddev:50.

Κάθε φορά που καλείται επιστρέφει και μια νέα τυχαία τιμή. Οι τιμές αυτές απεικονίζονται σε ένα GridView, αποθηκεύονται σε απλό δισ-δυάστατο πίνακα και ο καθένας σε μια μία μεταβλητή για να χρησιμοποιηθεί αργότερα για τους υπολογισμούς. Ο κώδικας μας παράγει 10000 πεντάδες τυχαίων αριθμών.

Στην κάτωθι εικόνα φαίνεται η παραγωγή των τυχαίων αριθμών

2. ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΑΣ ΠΛΕΙΔΑΣ 5 ΑΡΙΘΜΩΝ

Δημιουργία τυχαίας πλειάδας αριθμών	
Double number	
22,9502347119691	
12,3638134324043	
10,0020658109387	
3,45379421723345	
-13,5592964177425	
20,197693365177	
23,8204780588301	
18,4821931117177	
13,2484329418402	
4,26813866545579	
11,61856861701	
8,58962059019445	
23,6132494467893	
17,6077023684825	
17,2518411810997	
-7,20012491696311	
8,4875974398166	
11,7927125379279	
15,6985006443607	
10,0324397613514	
6,9420978273811	

Εικόνα 14: Παραγόμενοι τυχαίοι αριθμοί (10000 παραγόμενες πεντάδες)

5.3.3 Υπολογισμός μετρικών

Μετά την παραγωγή των τυχαίων αριθμών, γίνεται ο υπολογισμός αθροίσματος για κάθε στήλη του πίνακα. Για κάθε πίνακα έχει δημιουργηθεί μία λίστα για να είναι πιο εύκολα διαχειρίσιμη στην συνέχεια και οι υπολογισμοί του πίνακα γίνονται σε `foreach`.

Στη συνέχεια υπολογίζεται η μέθοδος jaccard, η μέθοδος Sorensen ή Bray – Curtis και η μέθοδος Kulczynski ή QSK (Quantitative symmetric dissimilarity), όπως αναφέρθηκε στο κεφάλαιο 4.

Έπειτα υπολογίζεται ο μέσος όρος για κάθε πίνακα και στη συνέχεια, αφού γίνουν όλοι οι υπολογισμοί για όλους τους 5 πίνακες, ελέγχεται ποια έχει την μικρότερη απόσταση (μικρότερο αριθμό) και μέσω μια μεταβλητής `v` δείχνει σε ποιον πίνακα θα γίνει η εισαγωγή των εγγραφών:

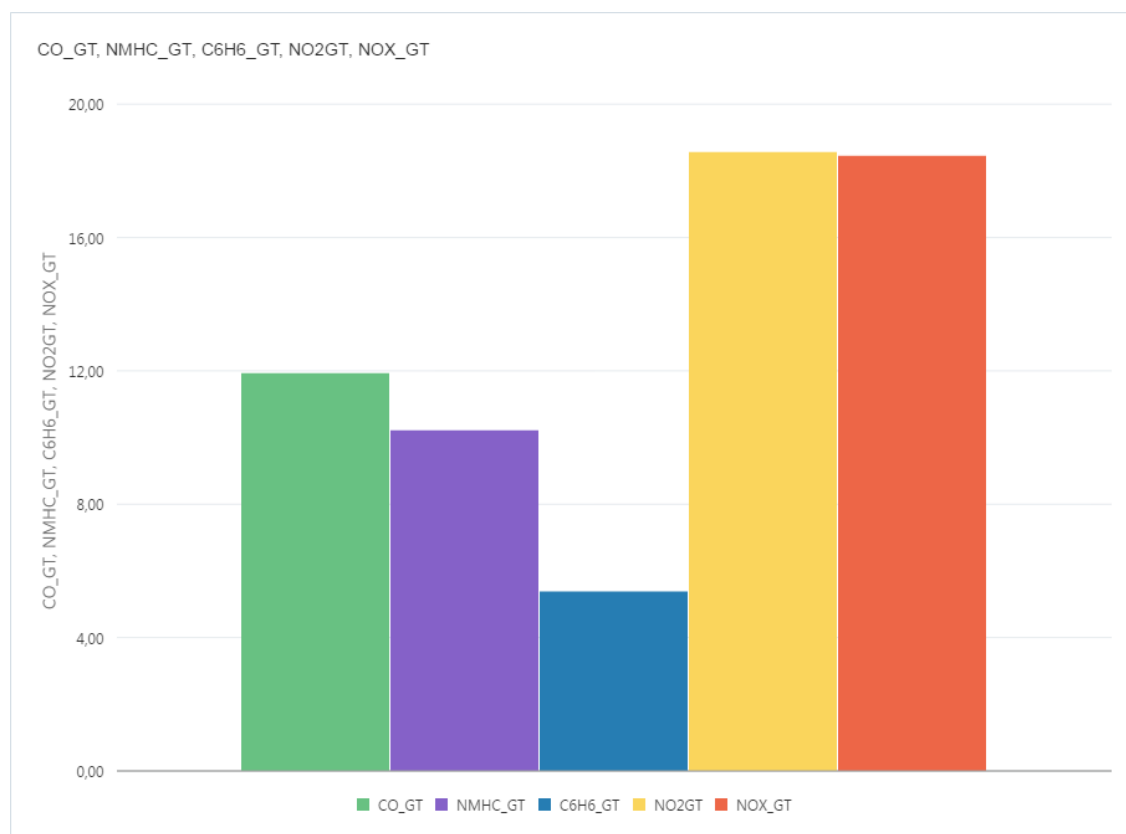
Μετά τον υπολογισμό αυτών ακολουθεί η εισαγωγή στον πίνακα αυτόν οι τιμές των 5 τυχαίων αριθμών (όσα είναι και τα πεδία που θέλουμε).

Ακολουθούν τα αποτελέσματα του μέσου όρου και της τυπικής απόκλισης, όπου για κάθε πίνακα δημιουργήθηκε μια λίστα. Υπολογίζεται αρχικά ξανά το πλήθος, έπειτα το άθροισμα της κάθε στήλης καθώς και το άθροισμα της κάθε στήλης πολλαπλασιασμένο με τον εαυτό του, που θα χρησιμοποιηθεί πιο κάτω για τον υπολογισμό του μέσου όρου και της τυπικής απόκλισης αντίστοιχα. Σύμφωνα με τα αποτελέσματα αυτά θα απεικονίσουμε στο τέλος γραφικές παραστάσεις.

5.4 Γραφικές παραστάσεις

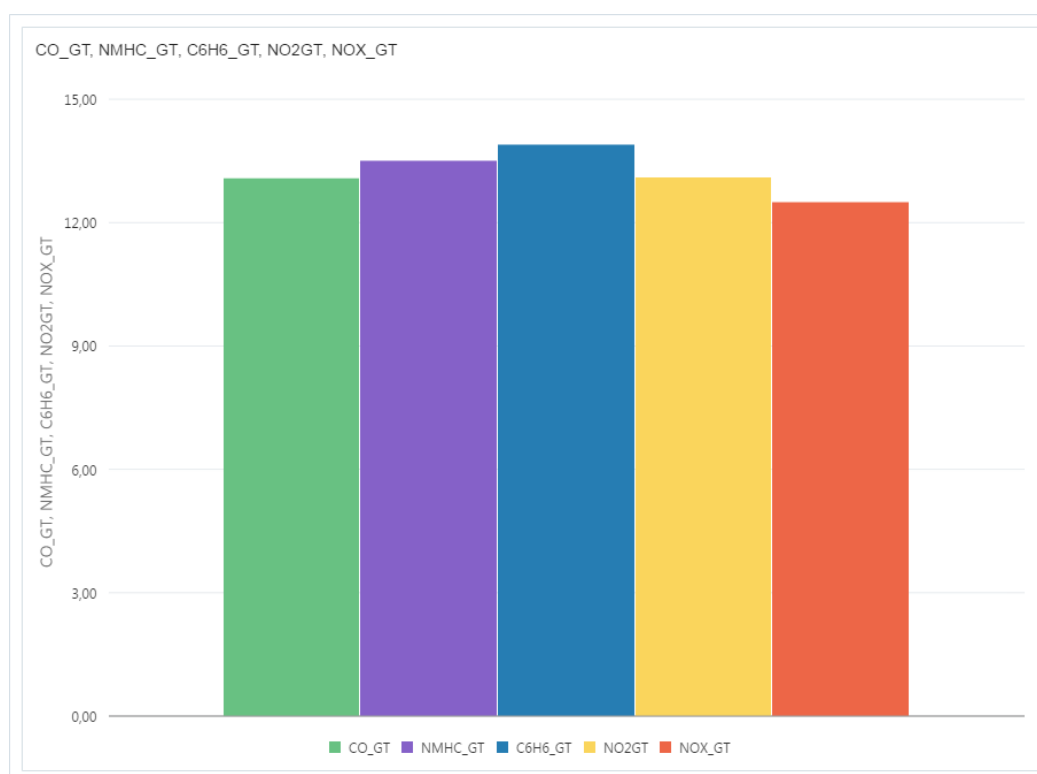
Τέλος σύμφωνα με τα αποτελέσματα του μέσου όρου και της τυπικής απόκλισης εξάγαμε γραφικές παραστάσεις για κάθε πίνακα και για κάθε σύνολο πειραμάτων. Για τον μέσο όρο χρησιμοποιήθηκε το εργαλείο της Oracle Data Visualization, ενώ για την τυπική απόκλιση το Excel

1^ο σετ πειραμάτων: avg:25, stddev:10



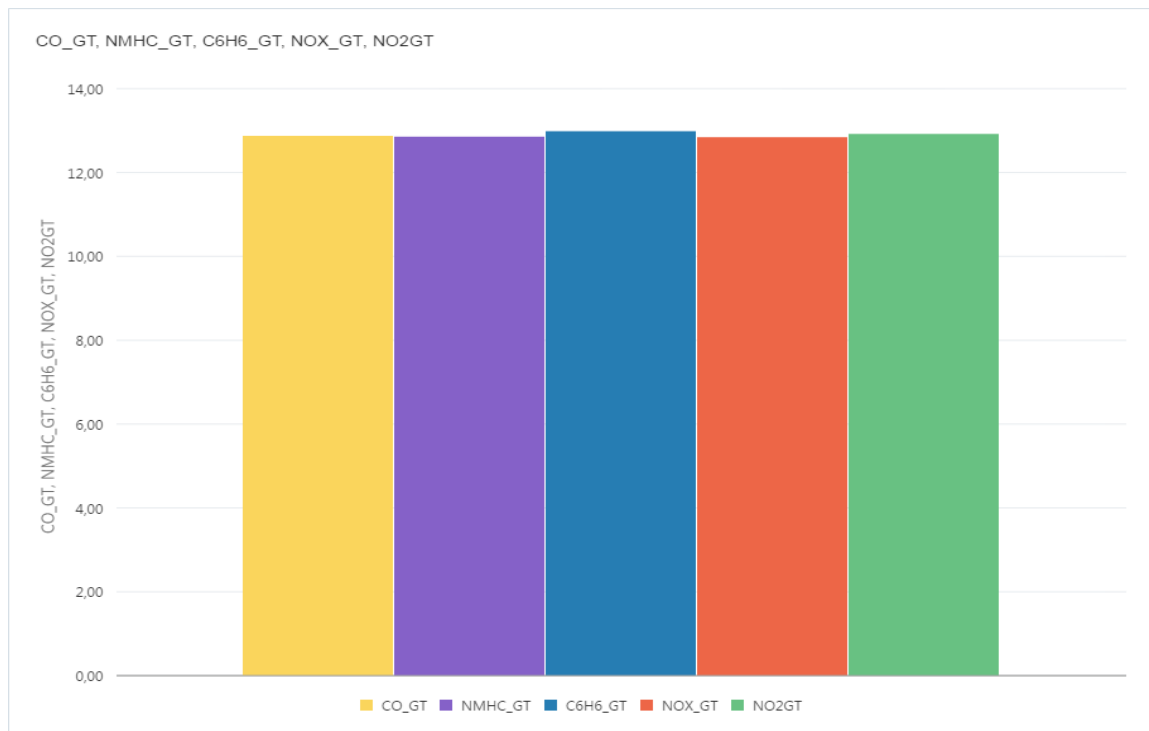
Πίνακας 1.1 : Μέσος όρος πίνακα AIR_1, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_1 εναλλάσσονται με μονη διαφορά του C6H6_GT που είναι στο 5. Πλήθος εγγραφών πίνακα AIR_1: 1585



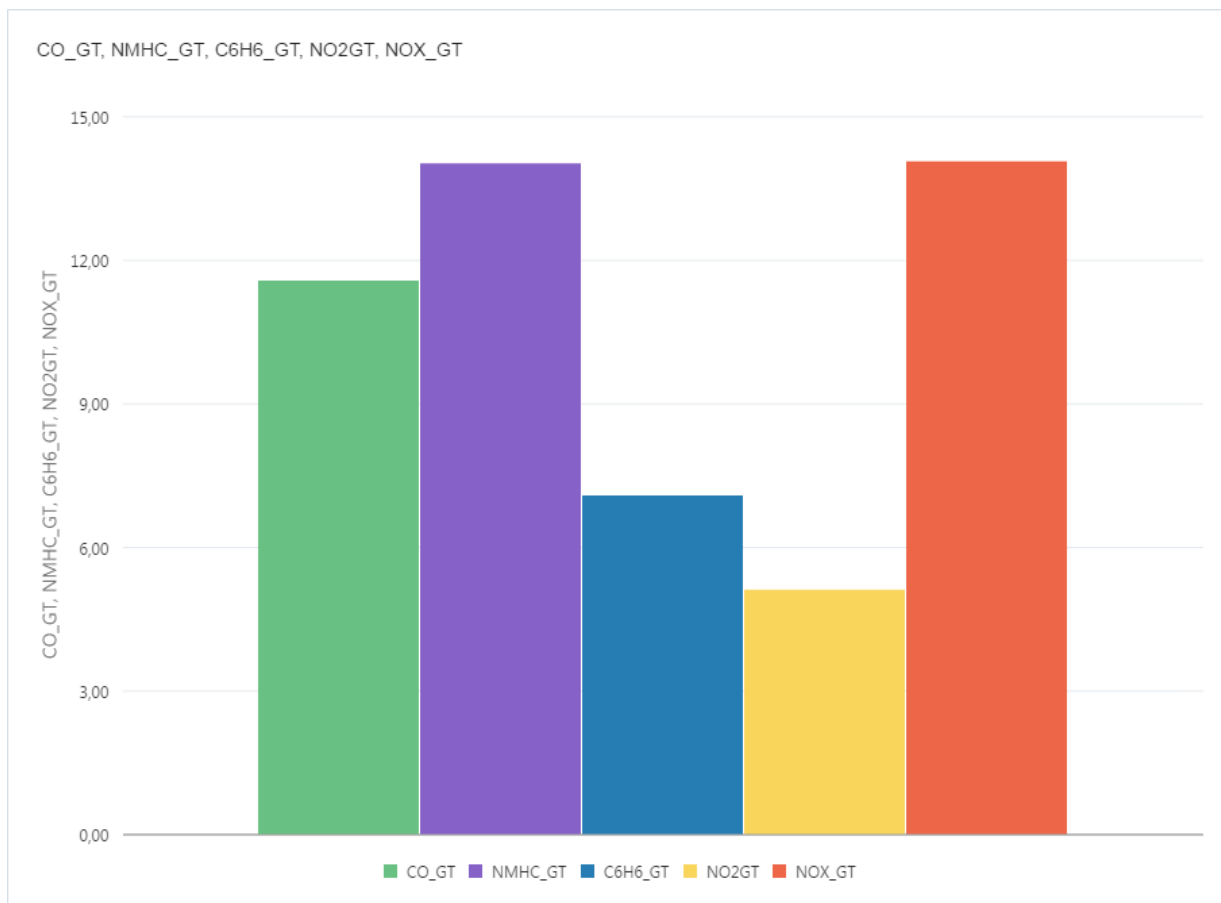
Πίνακας 1.2 : Μέσος όρος πίνακα AIR_2, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_2 είναι κοντά μεταξύ τους όλα γύρω στο 12. Πλήθος εγγραφών πίνακα AIR_2: 4368



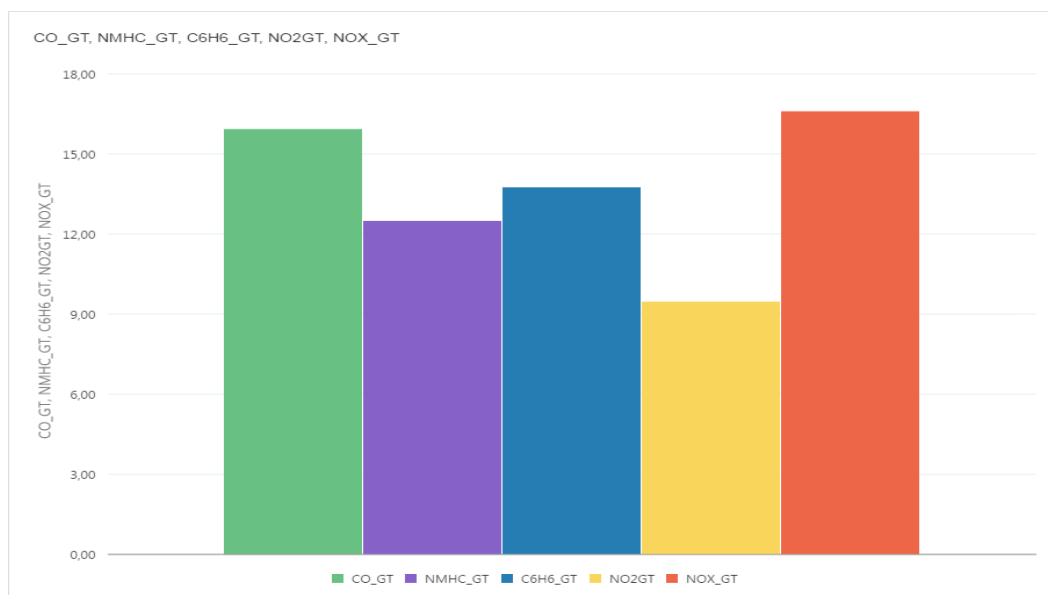
Πίνακας 1.3 : Μέσος όρος πίνακα AIR_3, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_3 είναι κοντά μεταξύ τους όλα γύρω στο 12. Πλήθος εγγραφών πίνακα AIR_3: 3345



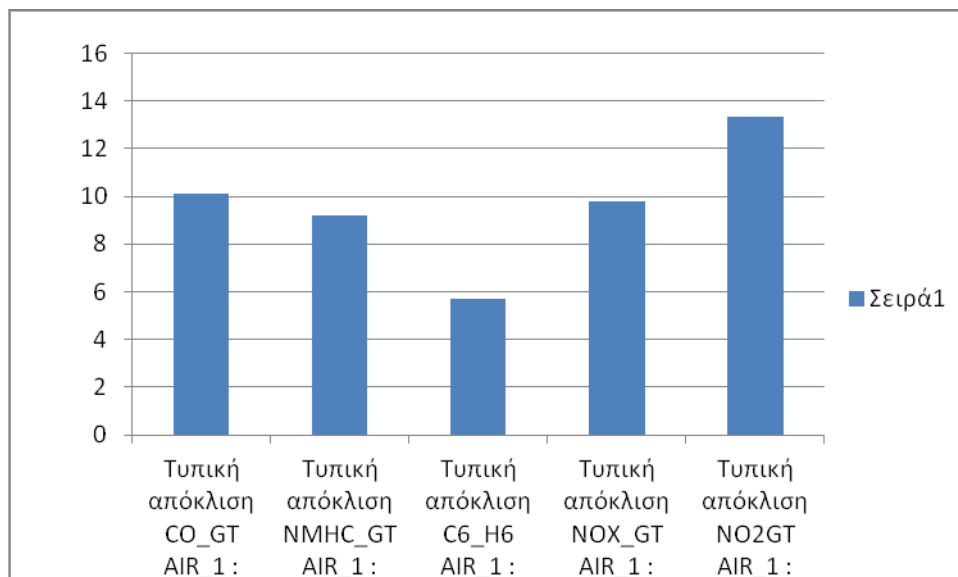
Πίνακας 1.4 : Μέσος όρος πίνακα AIR_4, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_4 εναλλάσσονται μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_4: 467



Πίνακας 1.5 : Μέσος όρος πίνακα AIR_5, 1^{ου} σετ πειραμάτων

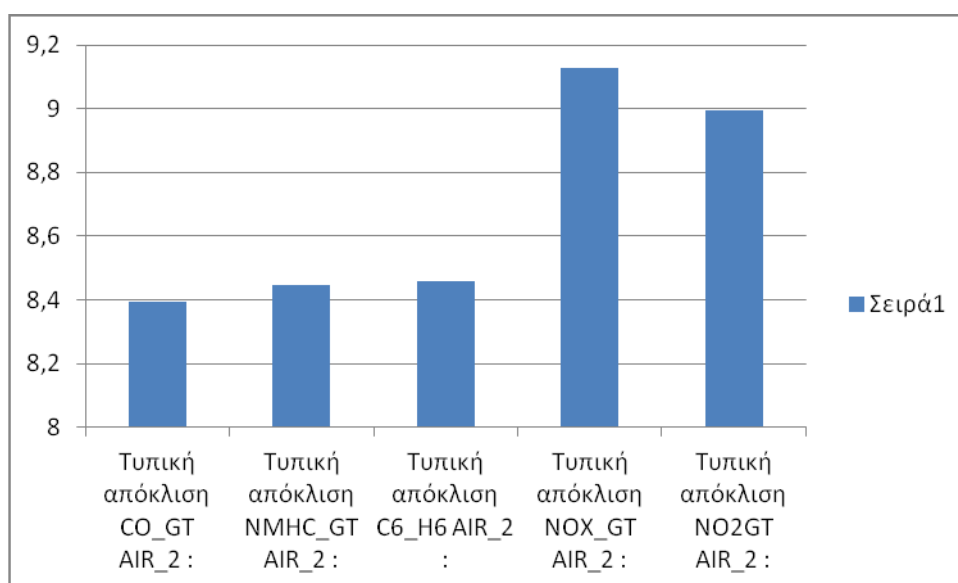
Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_5 εναλλάσσονται μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_5: 235



Πίνακας 1.6 : Τυπική απόκλιση πίνακα AIR_1, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_1 εναλλάσσονται μεταξύ τους.

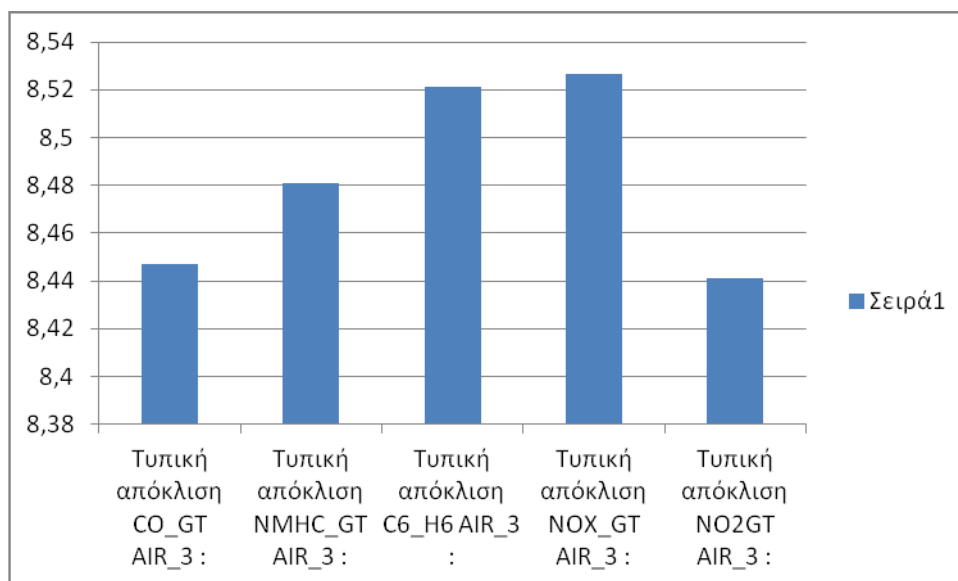
Σε σύγκριση με τον πίνακα 1.1 του μέσου όρου η τυπική απόκλιση είναι κοντά στα πεδία CO_GT, NMHC_GT και C6H6_GT, όχι όμως στα πεδία NOX_GT και NO2_GT



Πίνακας 1.7 : Τυπική απόκλιση πίνακα AIR_2, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_2 είναι κοντά μεταξύ τους, γύρω στο 8.

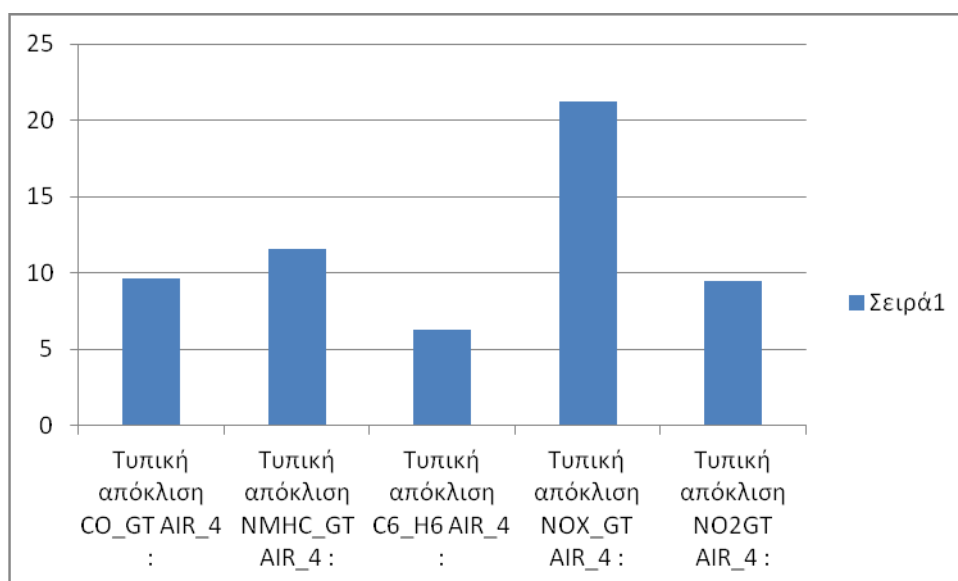
Σε σύγκριση με τον πίνακα 1.2 του μέσου όρου η τυπική απόκλιση και στα 5 πεδία διαφέρει κατά 5 μονάδες.



Πίνακας 1.8 : Τυπική απόκλιση πίνακα AIR_3, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_3 είναι κοντά μεταξύ τους, γύρω στο 8.

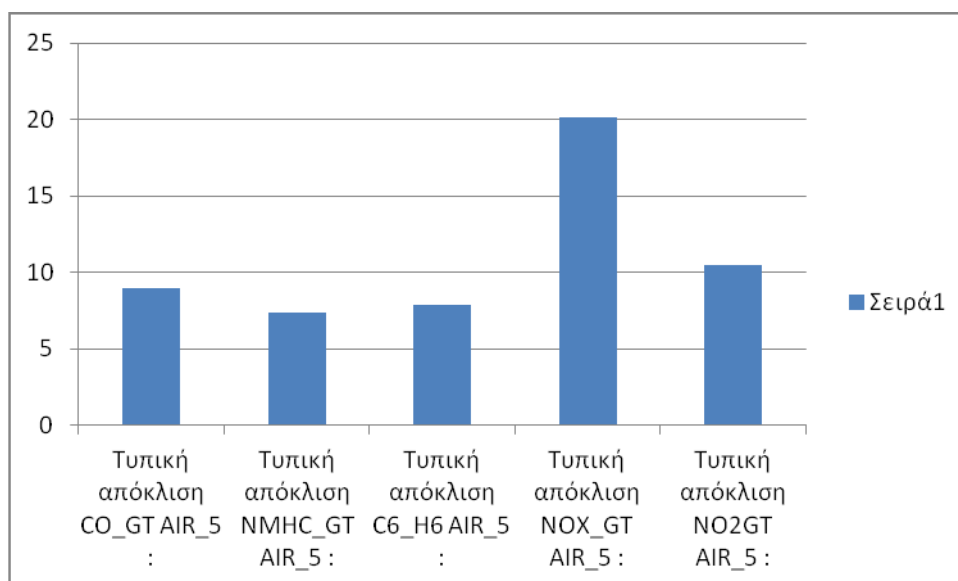
Σε σύγκριση με τον πίνακα 1.3 του μέσου όρου η τυπική απόκλιση και στα 5 πεδία διαφέρει κατά 4 μονάδες.



Πίνακας 1.9 : Τυπική απόκλιση πίνακα AIR_4, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_4 εναλλάσσονται μεταξύ τους.

Σε σύγκριση με τον πίνακα 1.4 του μέσου όρου η τυπική απόκλιση διαφέρει πολύ κυρίως στο πεδίο NOX_GT κατά 17 μονάδες.



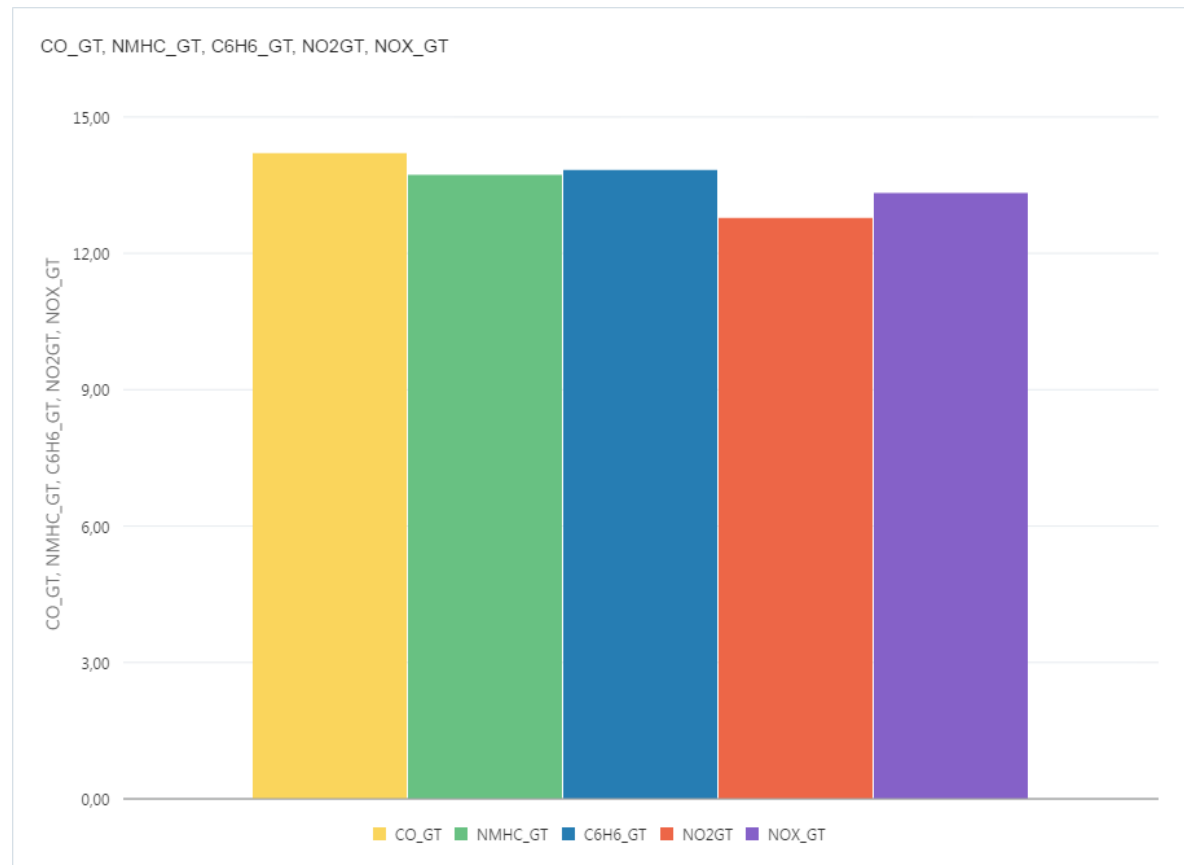
Πίνακας 1.10 : Τυπική απόκλιση πίνακα AIR_5, 1^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_5 εναλλάσσονται μεταξύ τους.

Σε σύγκριση με τον πίνακα 1.5 του μέσου όρου η τυπική απόκλιση διαφέρει πολύ κυρίως στο πεδίο NOX_GT κατά 11 μονάδες.

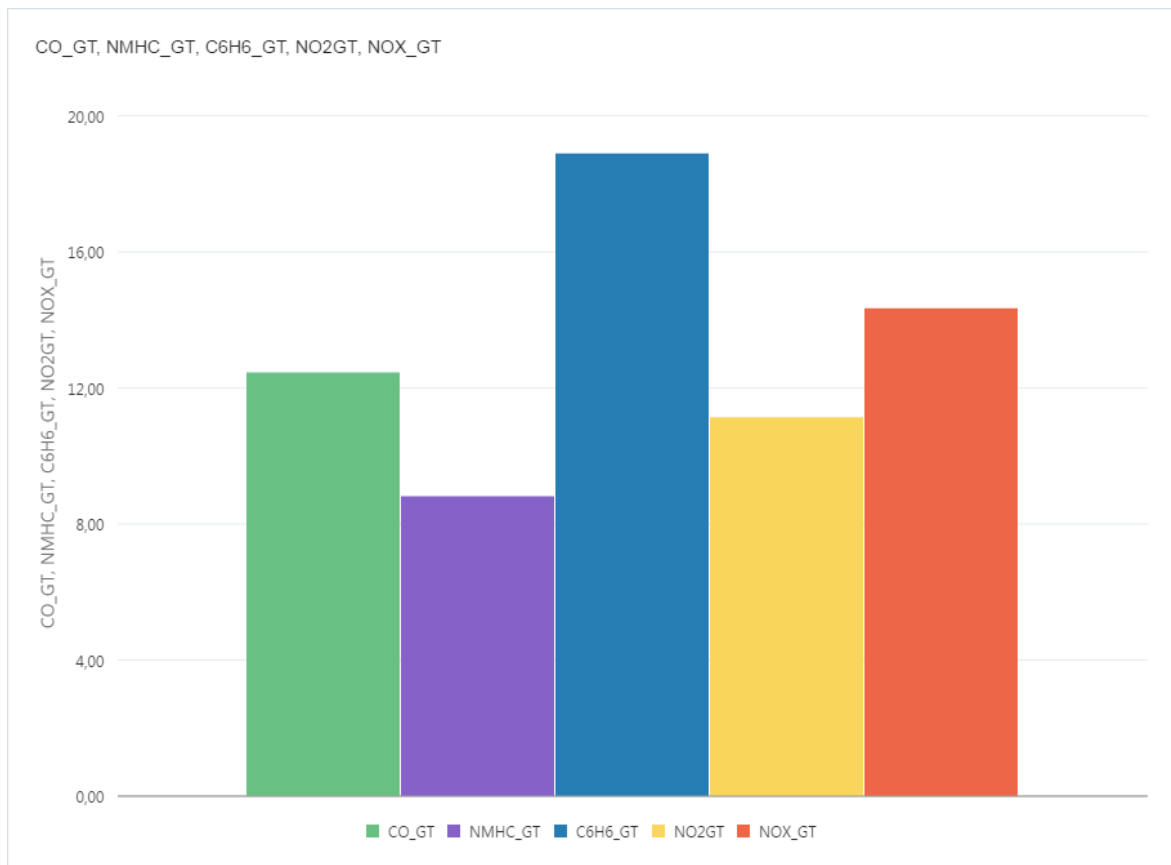
Από τους πιο πάνω πίνακες βλέπουμε ότι αυτοί με τις περισσότερες εγγραφές είναι ο AIR_2 και ο AIR_3 με 4368 και 3345 αντίστοιχα. Και στους 2 αυτούς πίνακες οι τιμές είναι κοντά μεταξύ τους τόσο στον μέσο όρο όσο και στην τυπική απόκλιση. Στον πίνακα AIR_2 στον μέσο όρο οι τιμές κυμαίνονται μεταξύ 12,49-13,89 και στην τυπική απόκλιση μεταξύ 8,39 - 9,12. Στον πίνακα AIR_3 στον μέσο όρο οι τιμές κυμαίνονται μεταξύ 12,84-12,98 και στην τυπική απόκλιση μεταξύ 8,44 - 8,52. Δεδομένου ότι οι τιμές αυτές είναι κοντά μεταξύ τους στους 2 από τους 5 πίνακες με τις περισσότερες εγγραφές έχουμε πετύχει τον στόχο μας για το 1^ο σετ πειραμάτων, που είναι να μαζέψουμε παραπλήσιες τιμές σε ένα σύνολο δεδομένων (Dataset).

2^ο σειτ πειραμάτων: avg:25, stddev:20



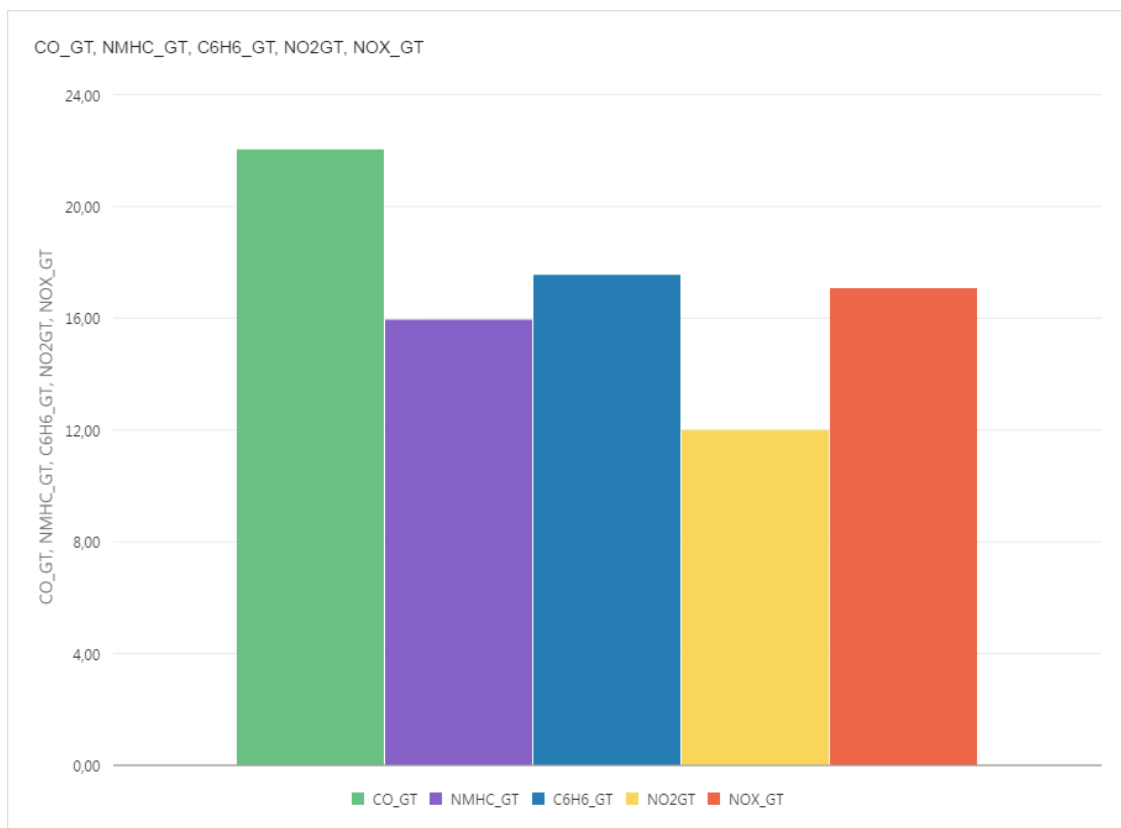
Πίνακας 2.1 : Μέσος όρος πίνακα AIR_1, 2^{ου} σειτ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_1 είναι κοντά μεταξύ τους, γύρω στο 13. Πλήθος εγγραφών πίνακα AIR_1: 3456



Πίνακας 2.2 : Μέσος όρος πίνακα AIR_2, 2^{ου} σερ πειραμάτων

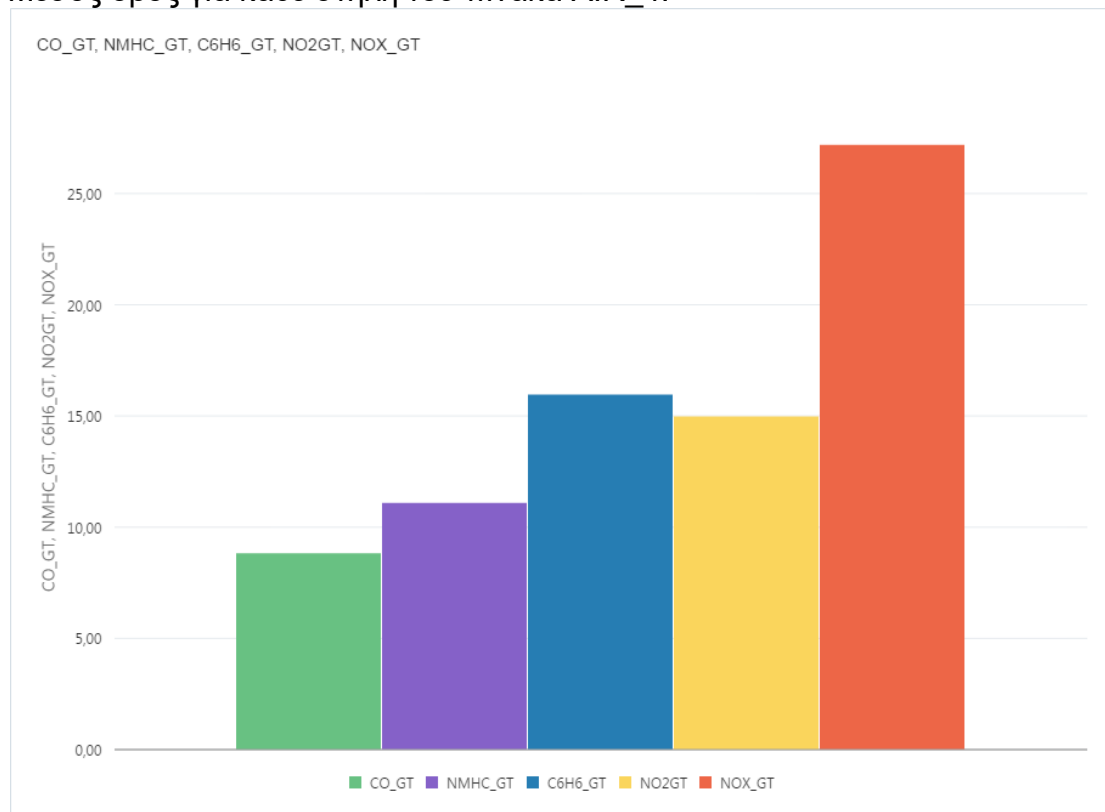
Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_2 εναλλάσσονται μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_2: 580



Πίνακας 2.3 : Μέσος όρος πίνακα AIR_3, 2^{ου} σερ πειραμάτων

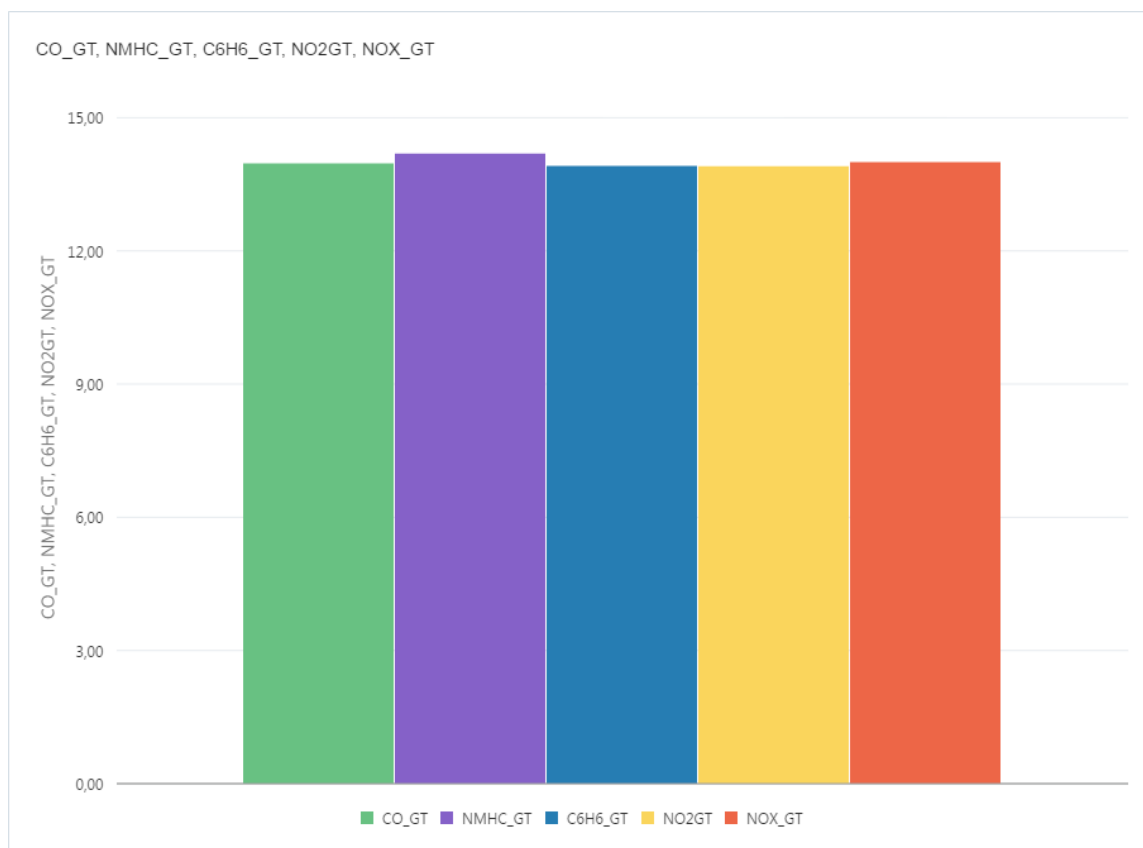
Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_3 εναλλάσσονται μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_3: 752

Μέσος όρος για κάθε στήλη του πίνακα AIR_4:



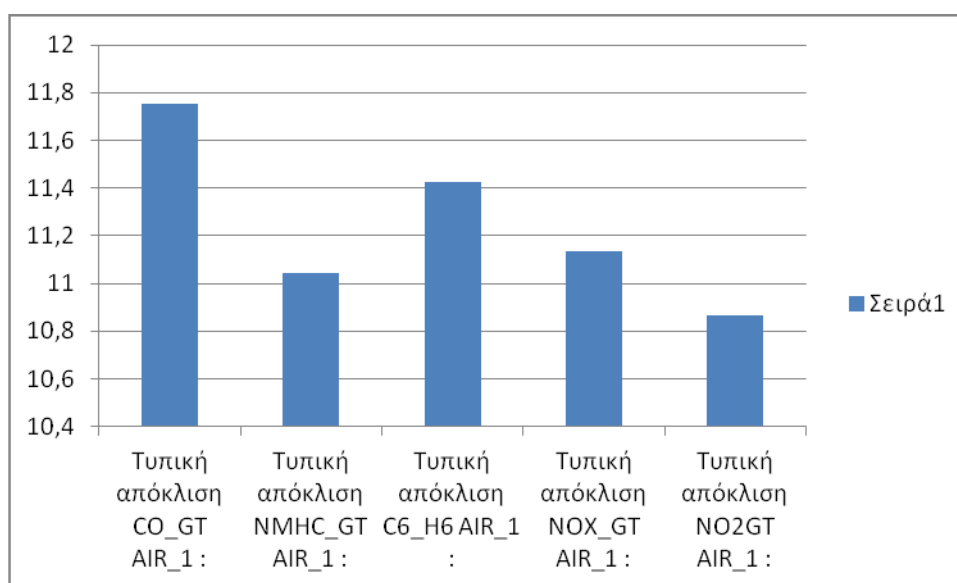
Πίνακας 2.4 : Μέσος όρος πίνακα AIR_4, 2^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_4 εναλλάσσονται μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_4: 1265



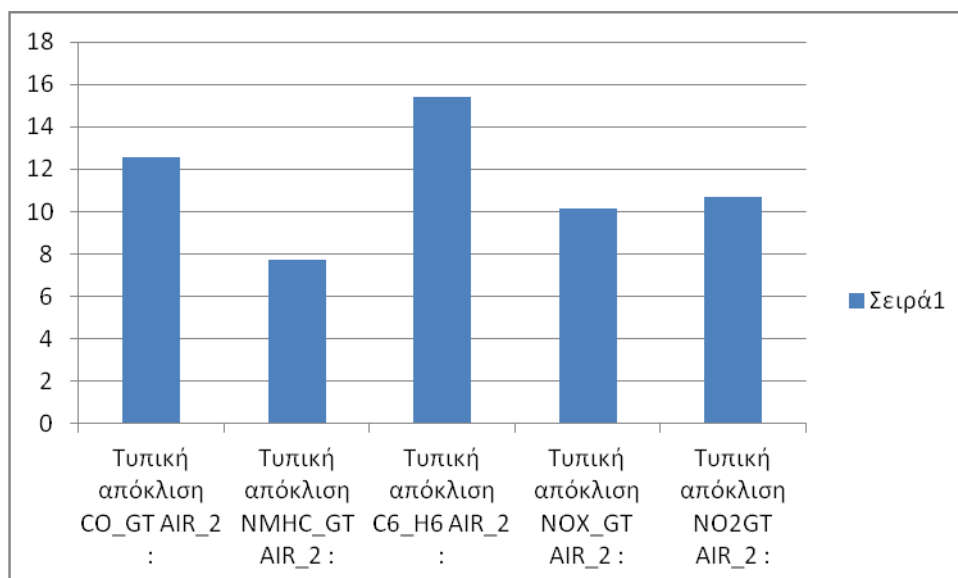
Πίνακας 2.5 : Μέσος όρος πίνακα AIR_5, 2^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_5 είναι κοντά μεταξύ τους, γύρω στο 13. Πλήθος εγγραφών πίνακα AIR_5: 3947



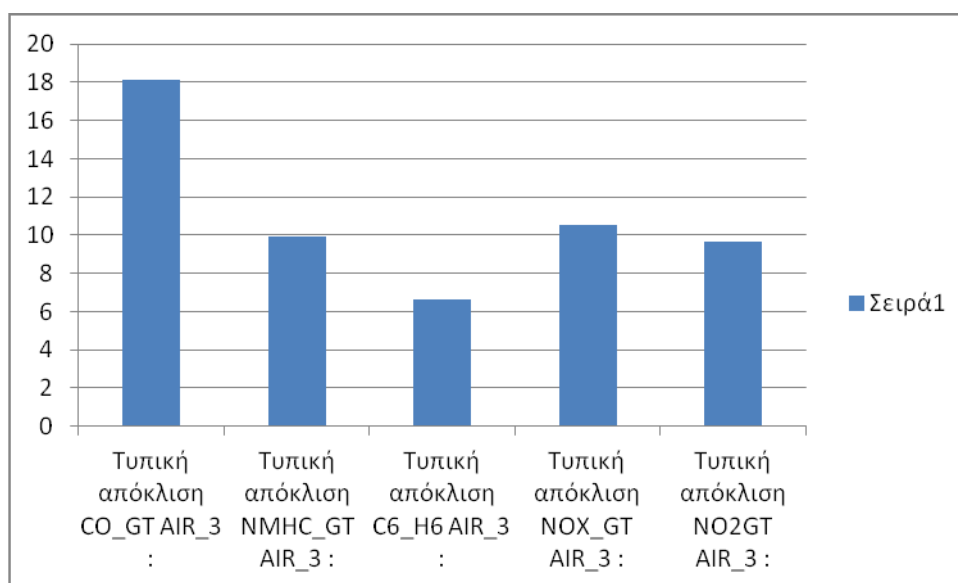
Πίνακας 2.6 : Τυπική απόκλιση πίνακα AIR_1, 2^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 2.1 του μέσου όρου η τυπική απόκλιση είναι αρκετά κοντά στις τιμές του μέσου όρου.



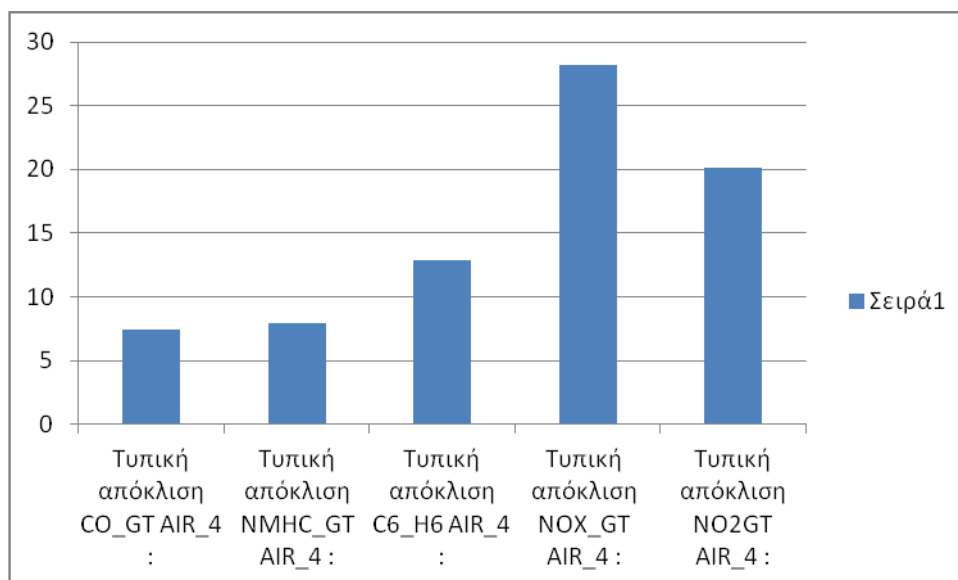
Πίνακας 2.7 : Τυπική απόκλιση πίνακα AIR_2, 2^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 2.2 του μέσου όρου η τυπική απόκλιση είναι αρκετά κοντά στις τιμές του μέσου όρου.



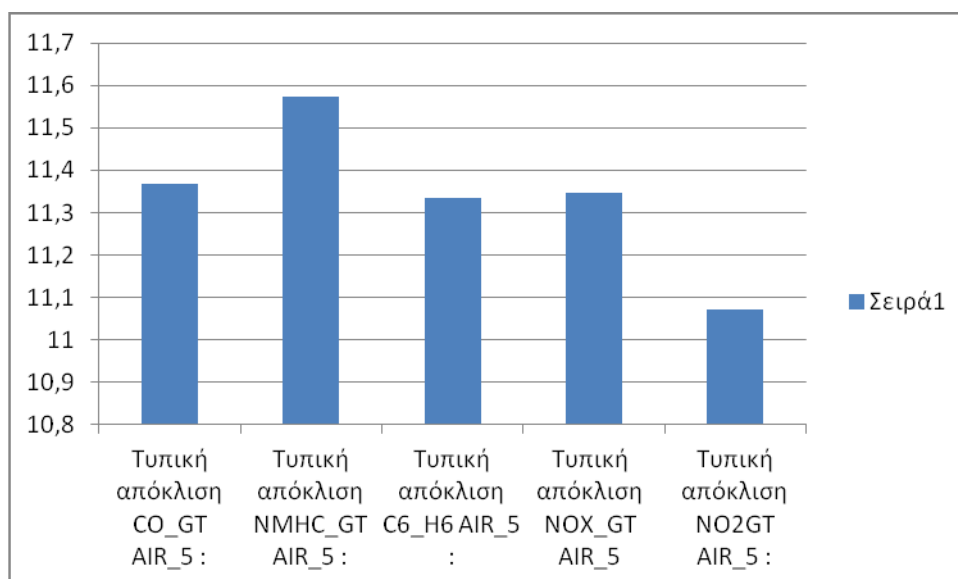
Πίνακας 2.8 : Τυπική απόκλιση πίνακα AIR_3, 2^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 2.3 του μέσου όρου η τυπική απόκλιση διαφέρει πολύ κυρίως στο πεδίο C6H6_GT κατά 11 μονάδες και στο NOX_GT κατά 17 μονάδες.



Πίνακας 2.9 : Τυπική απόκλιση πίνακα AIR_4, 2^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 2.4 του μέσου όρου η τυπική απόκλιση διαφέρει μόνο στο πεδίο NOX_GT κατά 13 μονάδες και στο πεδίο NO2_GT κατά 7 μονάδες και στο

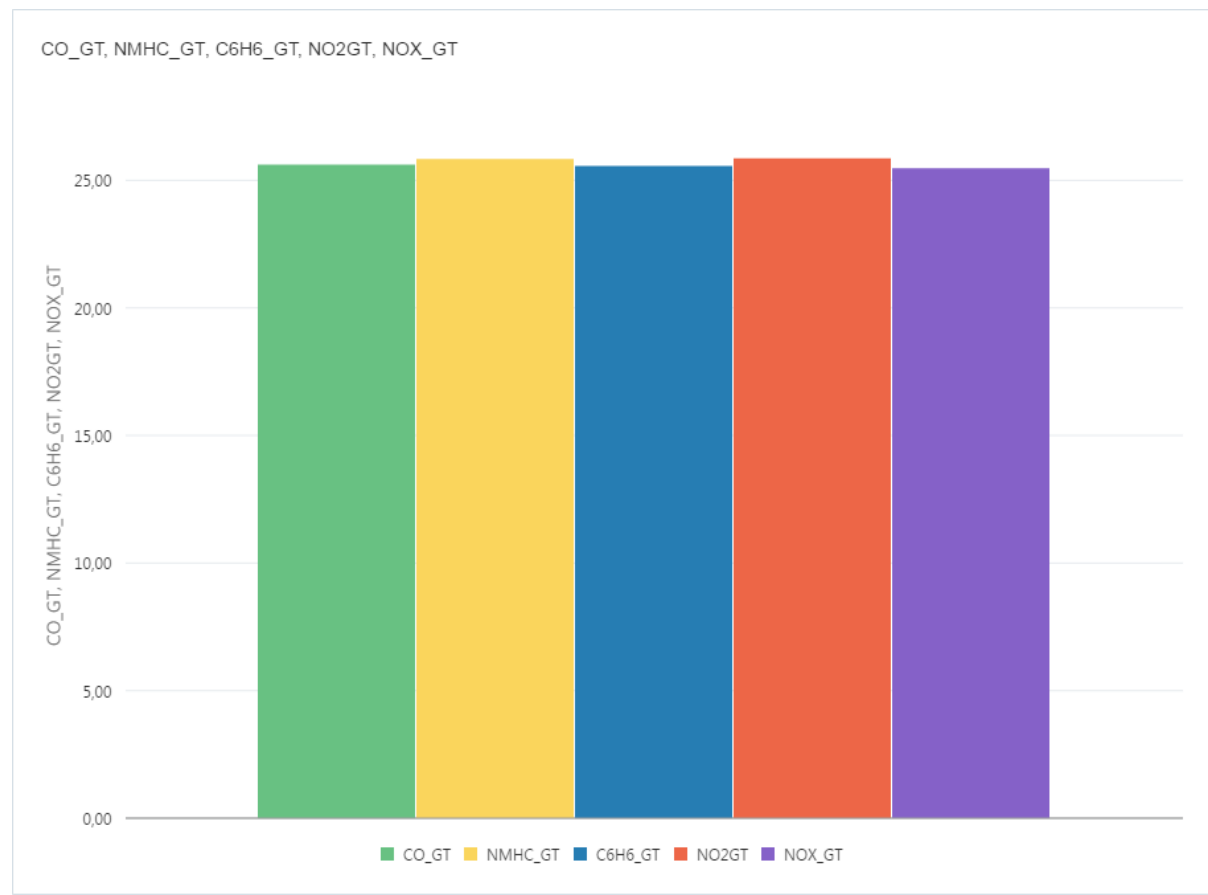


Πίνακας 2.10 : Τυπική απόκλιση πίνακα AIR_5, 2^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 2.5 του μέσου όρου η τυπική απόκλιση είναι αρκετά κοντά στις τιμές του μέσου όρου.

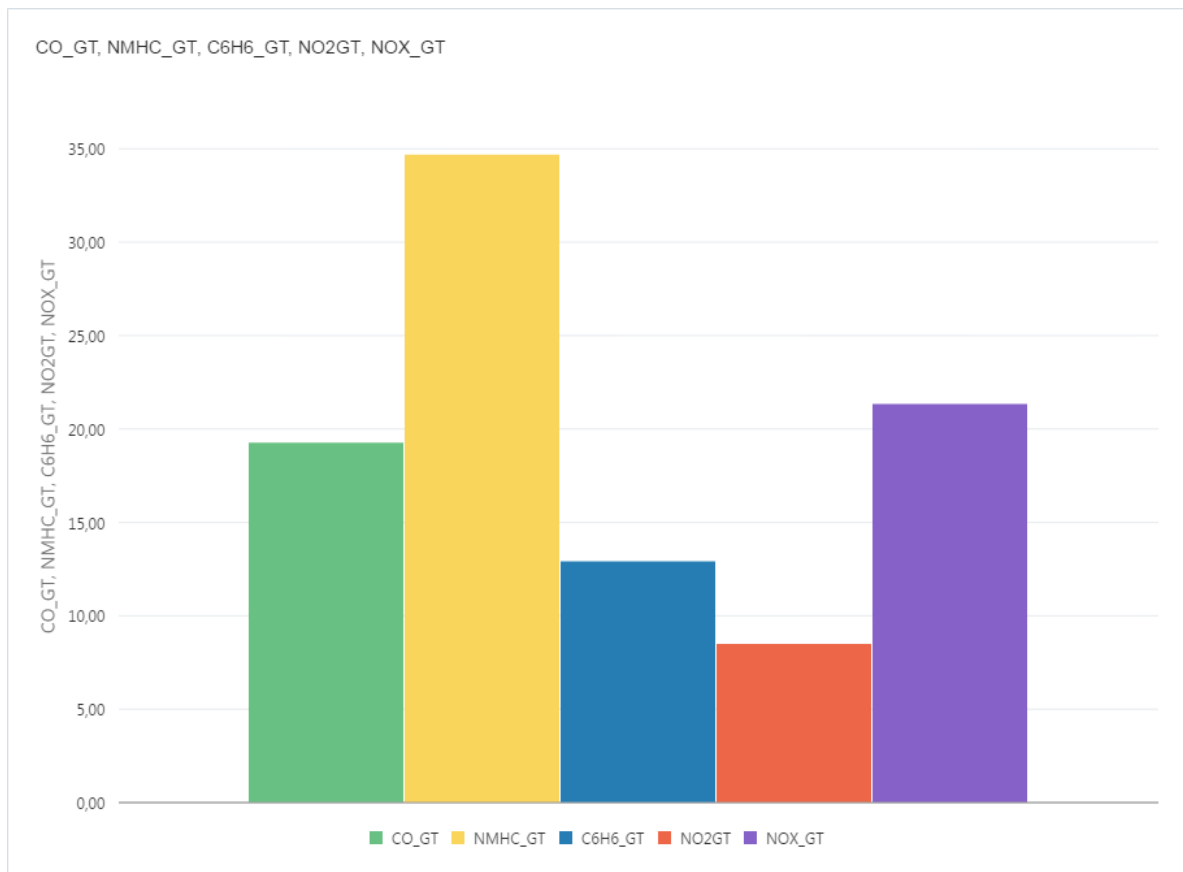
Από τους πιο πάνω πίνακες βλέπουμε ότι αυτοί με τις περισσότερες εγγραφές είναι ο AIR_1 και ο AIR_5 με 3456 και 3947 αντίστοιχα. Και στους 2 αυτούς πίνακες οι τιμές είναι κοντά μεταξύ τους τόσο στον μέσο όρο όσο και στην τυπική απόκλιση. Στον πίνακα AIR_1 στον μέσο όρο οι τιμές κυμαίνονται μεταξύ 12,77-14,02 και στην τυπική απόκλιση μεταξύ 10,86 -11,75. Στον πίνακα AIR_5 στον μέσο όρο οι τιμές κυμαίνονται μεταξύ 13,90-14,00 και στην τυπική απόκλιση μεταξύ 11,07 - 11,57. Δεδομένου ότι οι τιμές αυτές είναι κοντά μεταξύ τους στους 2 από τους 5 πίνακες με τις περισσότερες εγγραφές έχουμε πετύχει τον στόχο μας και για το 2^ο σετ πειραμάτων, που είναι να μαζέψουμε παραπλήσιες τιμές σε ένα σύνολο δεδομένων (Dataset).

3^ο σετ πειραμάτων: avg:50, stddev:20



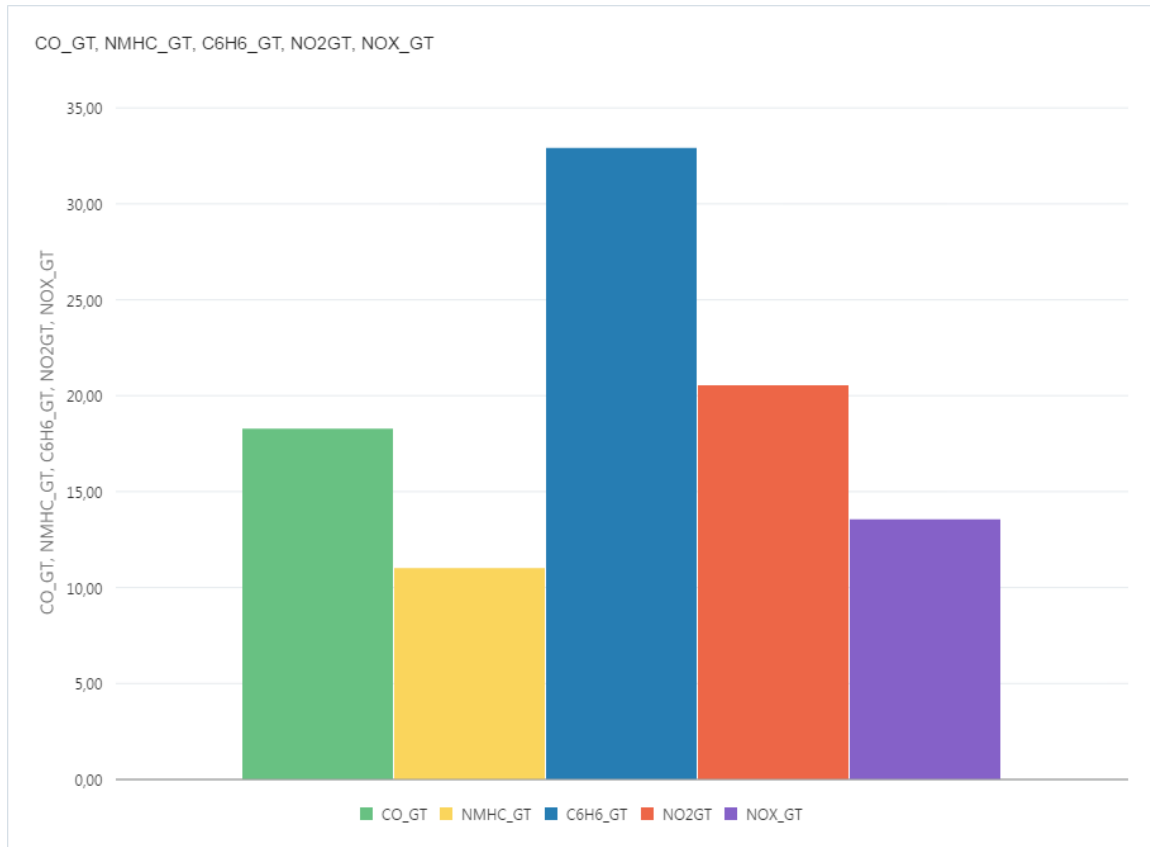
Πίνακας 3.1 : Μέσος όρος πίνακα AIR_1, 3^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_1 είναι κοντά μεταξύ τους, γύρω στο 25. Πλήθος εγγραφών πίνακα AIR_1: 4923



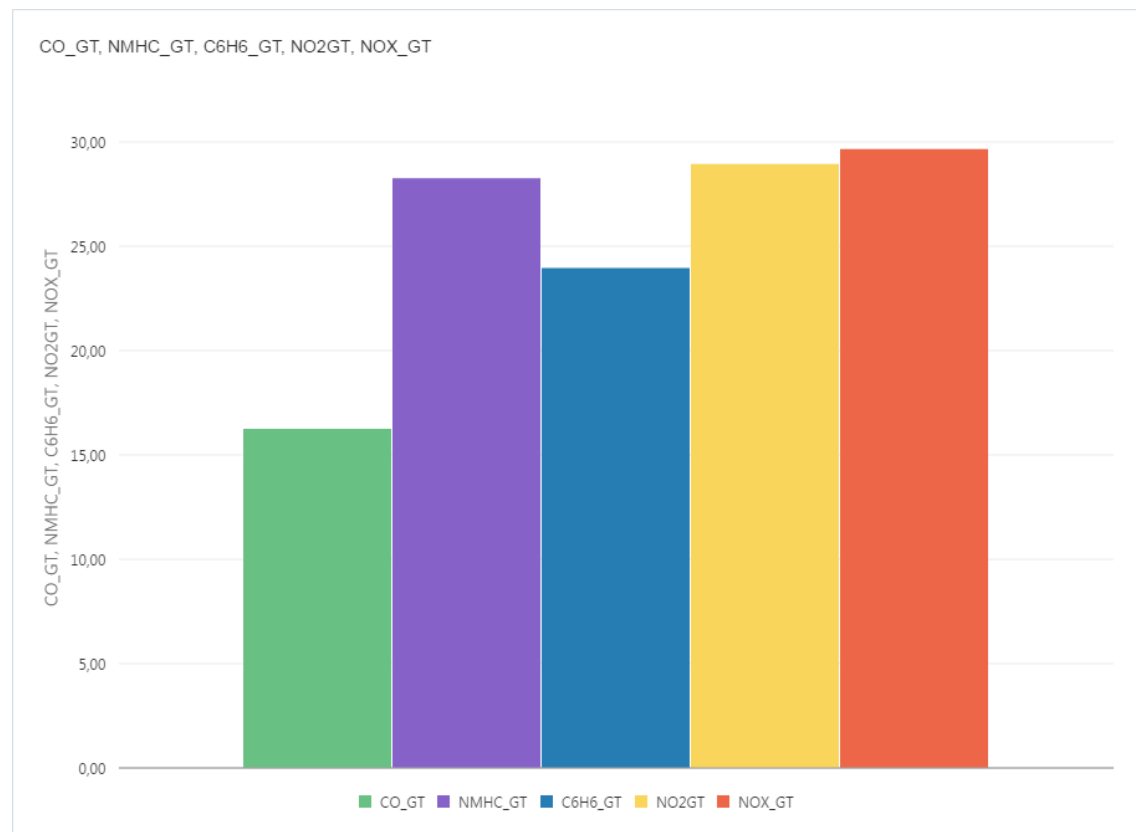
Πίνακας 3.2 : Μέσος όρος πίνακα AIR_2, 3^{ου} σερ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_2 εναλλάσσονται μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_2: 1876



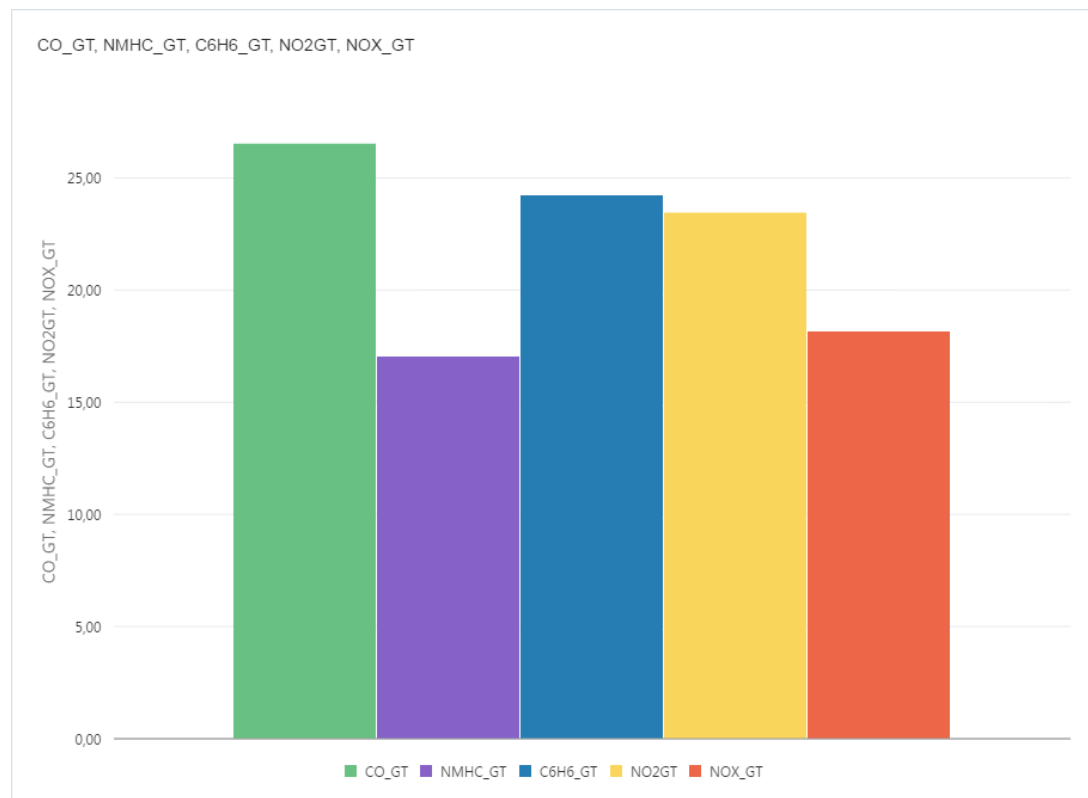
Πίνακας 3.3 : Μέσος όρος πίνακα AIR_3, 3^{ου} σερ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_3 εναλλάσσονται μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_3: 896



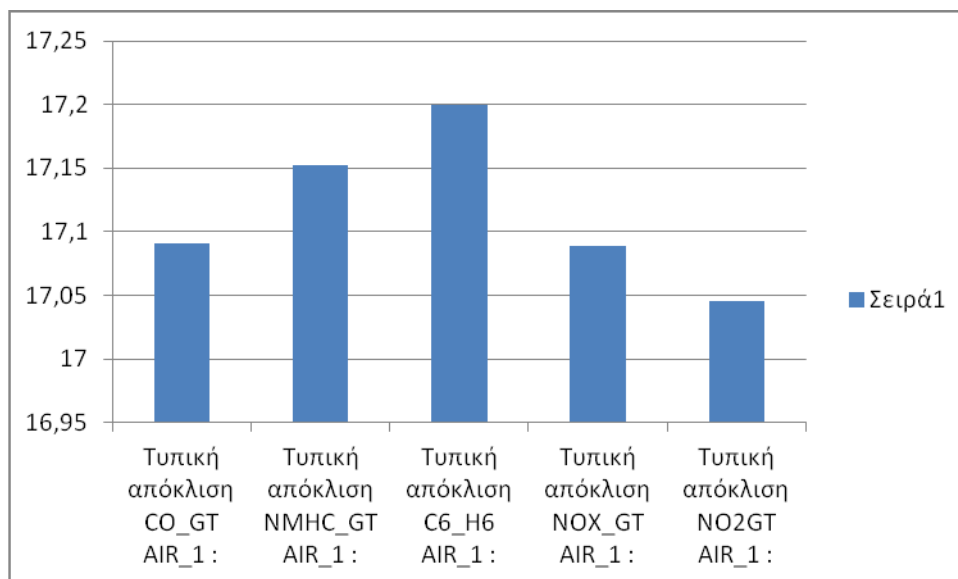
Πίνακας 3.4 : Μέσος όρος πίνακα AIR_4, 3^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_4 είναι κοντά μεταξύ τους εκτός από το CO_GT. Πλήθος εγγραφών πίνακα AIR_4: 1251



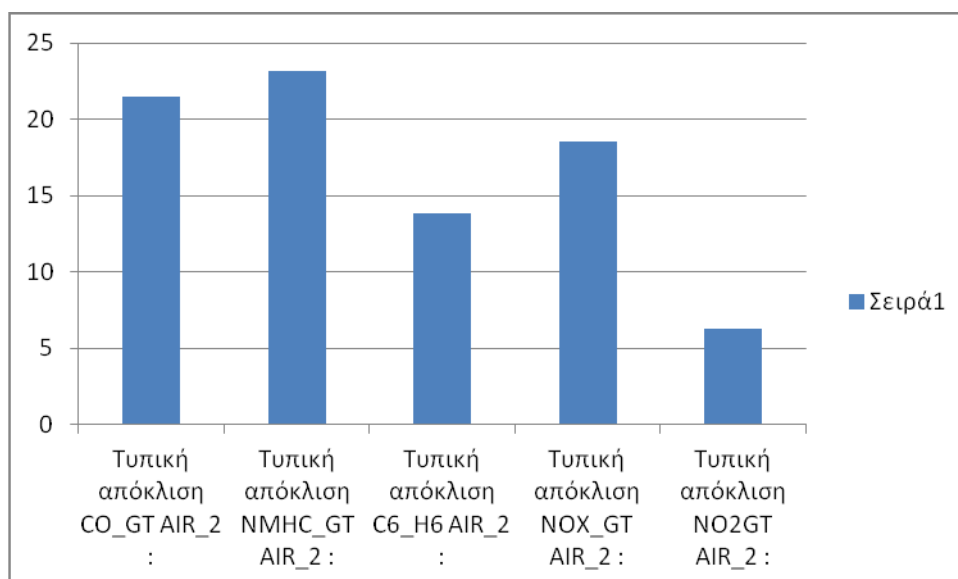
Πίνακας 3.5 : Μέσος όρος πίνακα AIR_5, 3^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_5 είναι κοντά μεταξύ τους εκτός από το NMHC_GT και NO2GT. Πλήθος εγγραφών πίνακα AIR_5: 1054



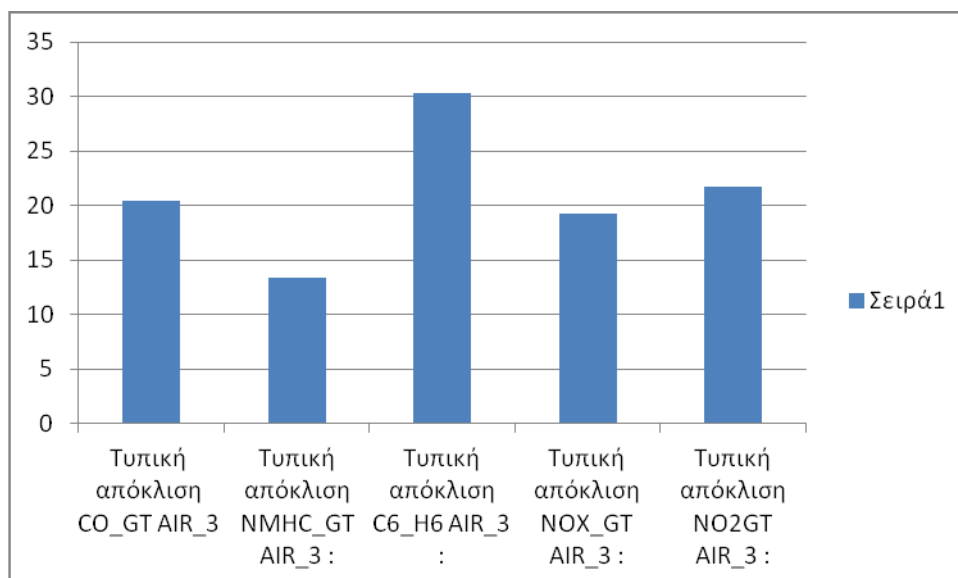
Πίνακας 3.6 : Τυπική απόκλιση πίνακα AIR_1, 3^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 3.1 του μέσου όρου η τυπική απόκλιση είναι κοντά μεταξύ τους, γύρω στο 17.



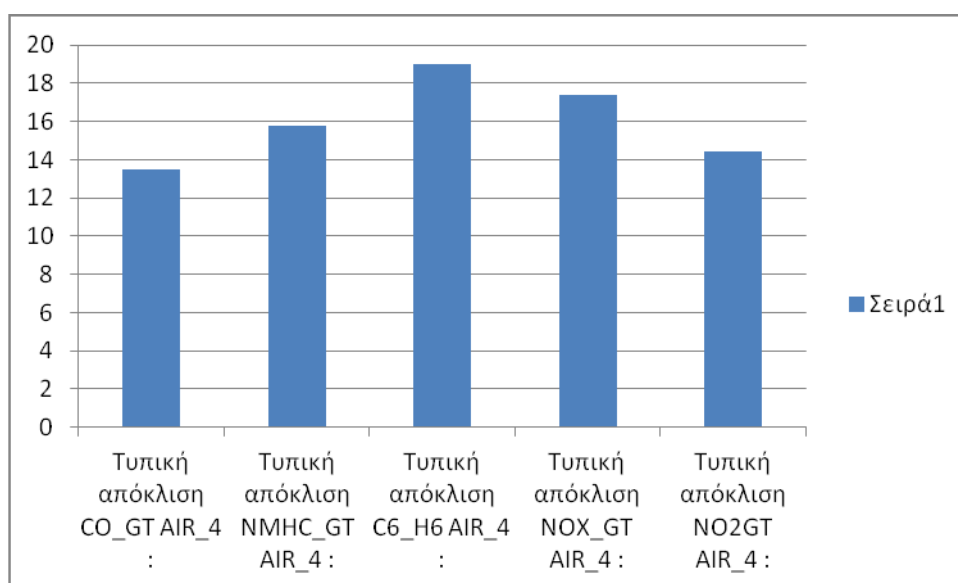
Πίνακας 3.7 : Τυπική απόκλιση πίνακα AIR_2, 3^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 3.2 του μέσου όρου η τυπική απόκλιση διαφέρει μόνο στο πεδίο NMHC_GT κατά 11 μονάδες, στο πεδίο NOX_GT κατά 10 μονάδες και στο NO2GT κατά 14 μονάδες.



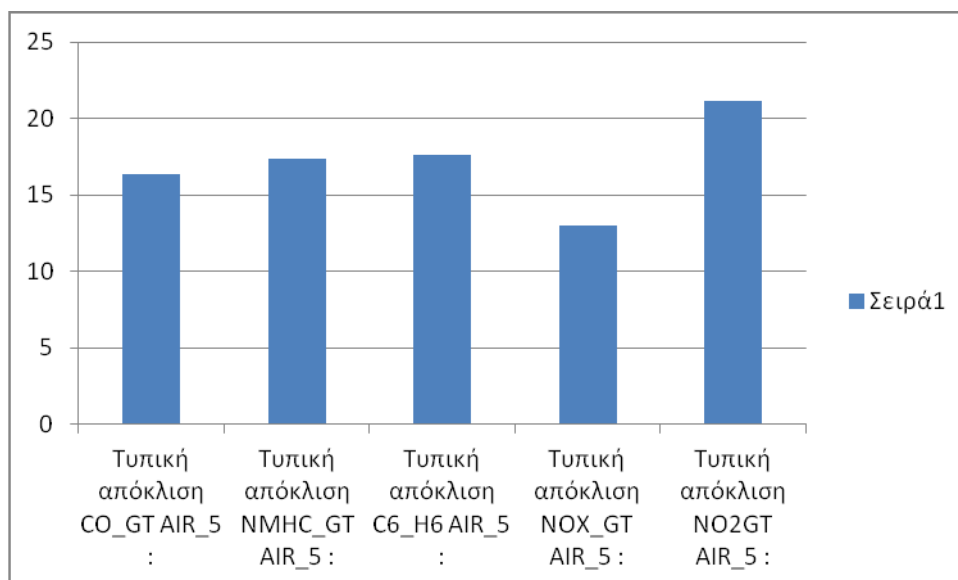
Πίνακας 3.8 : Τυπική απόκλιση πίνακα AIR_3, 3^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 3.3 του μέσου όρου η τυπική απόκλιση διαφέρει μόνο στο πεδίο NO2GT κατά 8 μονάδες.



Πίνακας 3.9 : Τυπική απόκλιση πίνακα AIR_4, 3^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 3.4 του μέσου όρου η τυπική απόκλιση διαφέρει αρκετά στα πεδία NMHC_GT, NOX_GT και στο NO2GT.

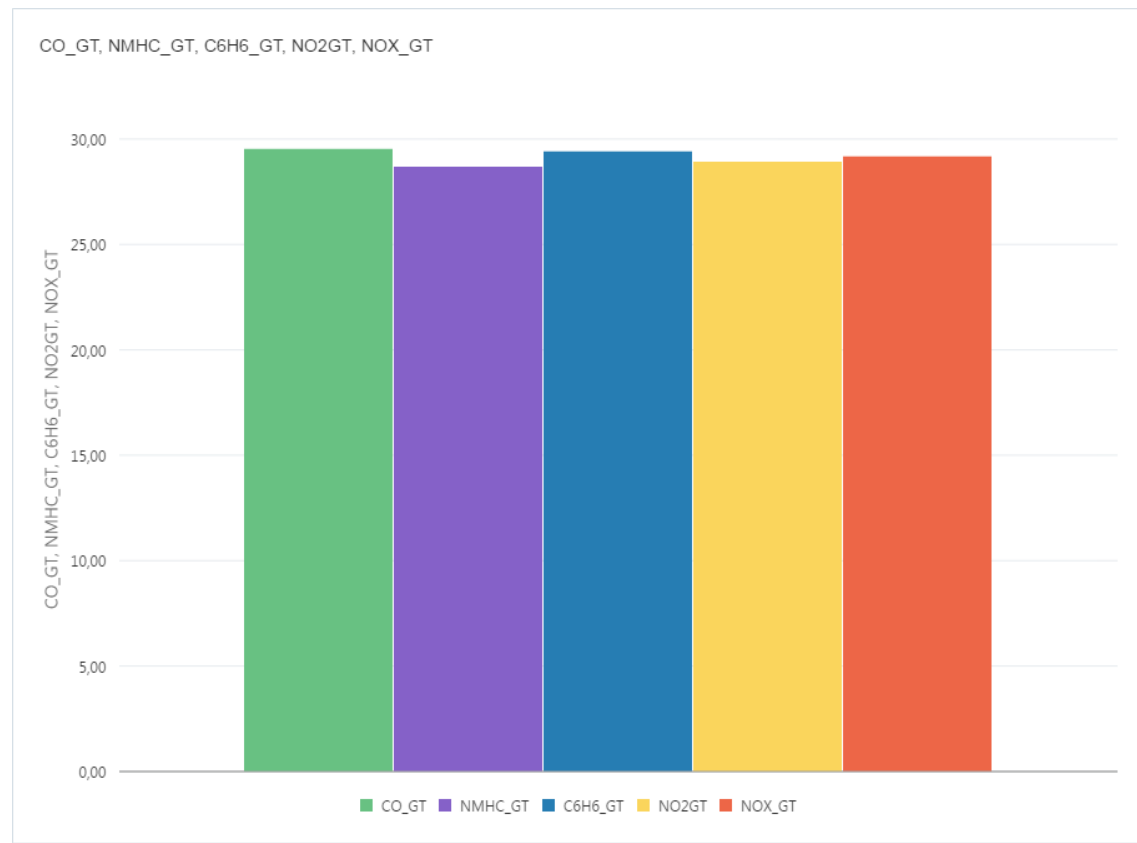


Πίνακας 3.10 : Τυπική απόκλιση πίνακα AIR_5, 3^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 3.5 του μέσου όρου η τυπική απόκλιση διαφέρει αρκετά σε όλα τα πεδία εκτός από το CO_GT.

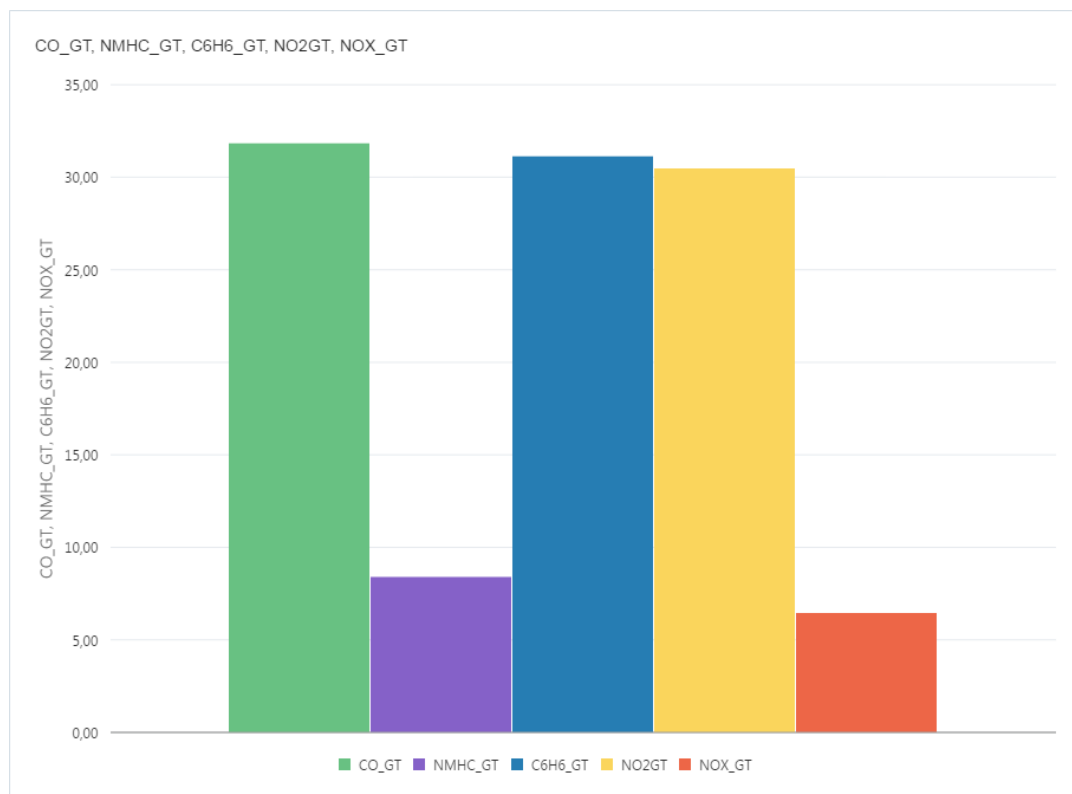
Από τους πιο πάνω πίνακες βλέπουμε ότι αυτός με τις περισσότερες εγγραφές είναι ο AIR_1 με 4923. Στον πίνακα αυτόν οι τιμές είναι κοντά μεταξύ τους τόσο στον μέσο όρο όσο και στην τυπική απόκλιση. Δηλαδή στον μέσο όρο οι τιμές κυμαίνονται μεταξύ 25,46-25,86 και στην τυπική απόκλιση μεταξύ 17,04 -17,89. Δεδομένου ότι οι τιμές αυτές είναι κοντά μεταξύ τους στον 1 από τους 5 πίνακες με τις περισσότερες εγγραφές έχουμε πετύχει τον στόχο μας και για το 3^ο σετ πειραμάτων, που είναι να μαζέψουμε παραπλήσιες τιμές σε ένα σύνολο δεδομένων (Dataset). Εδώ παρατηρούμε ότι λόγω της αύξησης της τιμής του μέσου όρου και της τυπικής απόκλισης που δώσαμε κατά την εκτέλεση δηλαδή 50 και 20 αντίστοιχα αυξήθηκαν και τα αποτελέσματα των παραπλήσιων τιμών, σε σχέση με τα 2 προηγούμενα πειράματά μας. Επίσης παρατηρούμε οι περισσότερες εγγραφές υπάρχουν σε έναν πίνακα και όχι σε 2 πίνακες όπως έγινε με τα 2 προηγούμενα πειράματά μας.

4^ο σετ πειραμάτων: avg:50, stddev:50



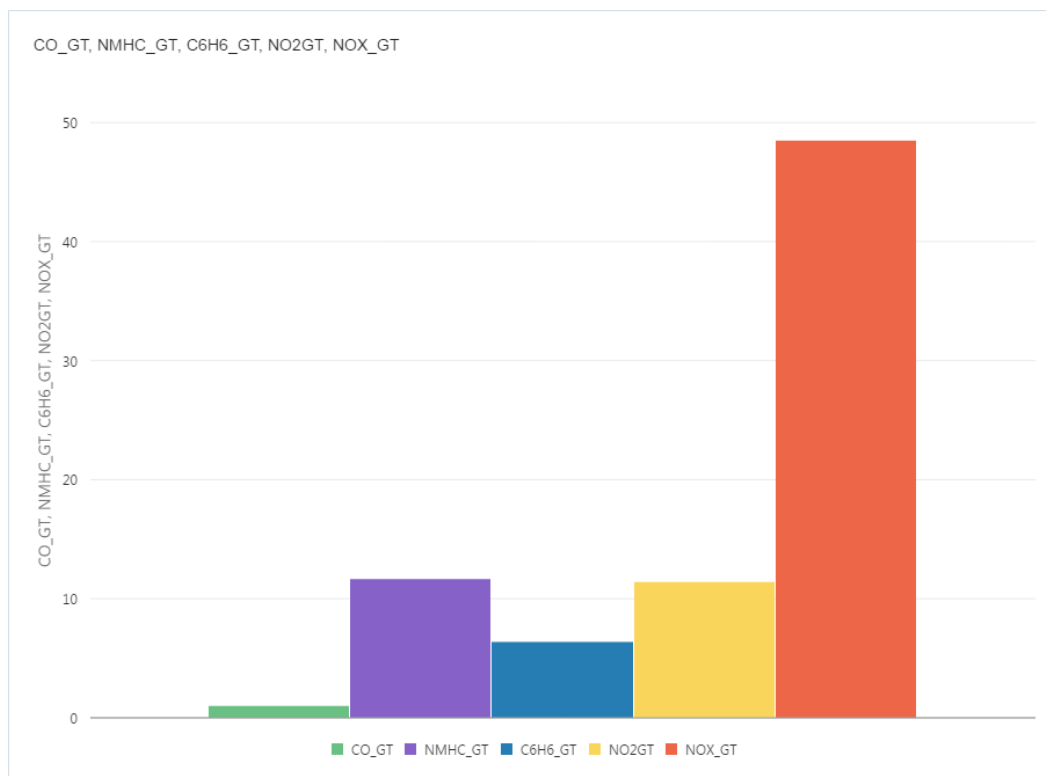
Πίνακας 4.1 : Μέσος όρος πίνακα AIR_1, 4^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_1 είναι κοντά μεταξύ τους, γύρω στο 29. Πλήθος εγγραφών πίνακα AIR_1: 4782.



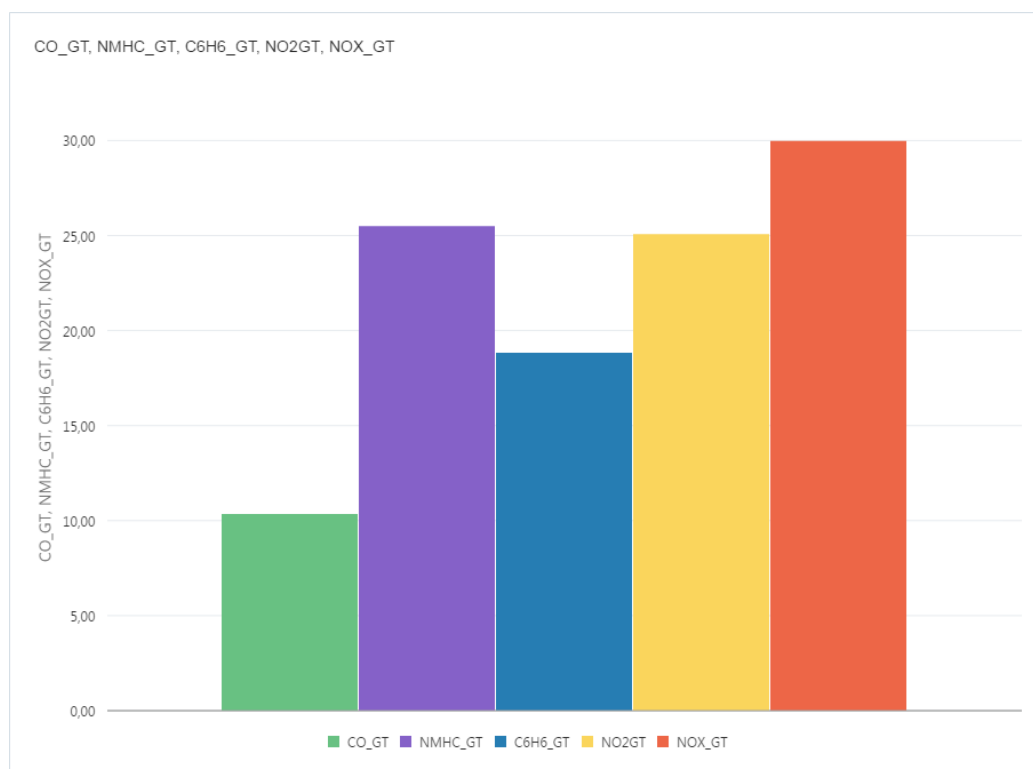
Πίνακας 4.2 : Μέσος όρος πίνακα AIR_2, 4^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_2 είναι κοντά μεταξύ τους εκτός από το NMHC_GT και NOX_GT. Πλήθος εγγραφών πίνακα AIR_2: 1532.



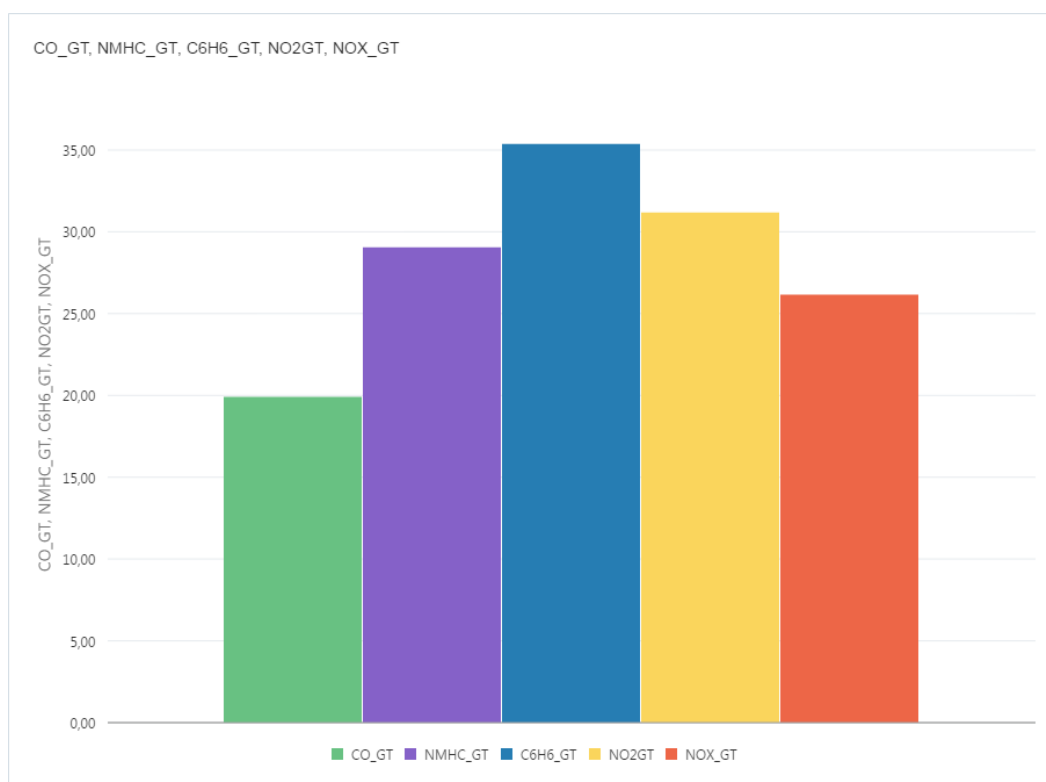
Πίνακας 4.3 : Μέσος όρος πίνακα AIR_3, 4^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_3 εναλλάσσονται αρκετά μεταξύ τους. Πλήθος εγγραφών πίνακα AIR_3: 946.



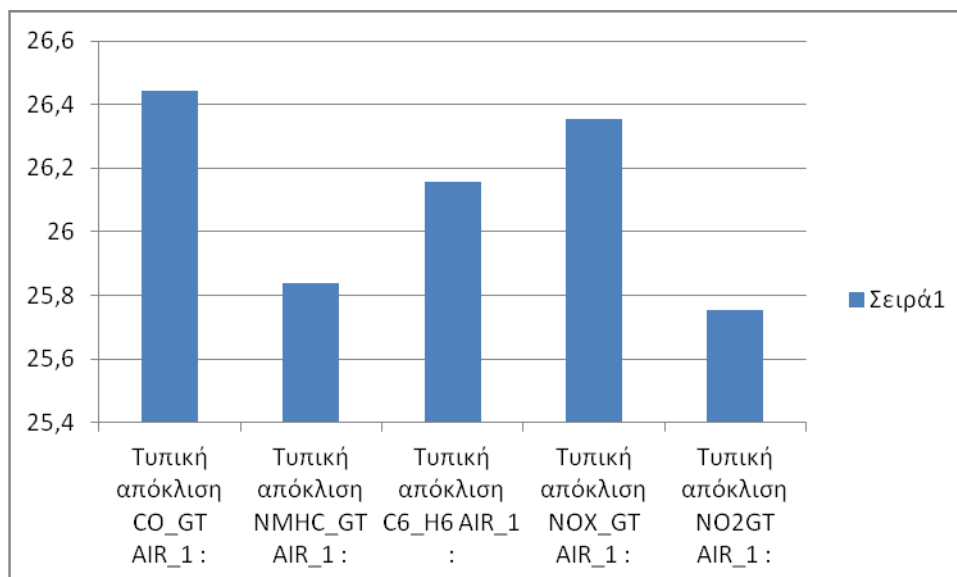
Πίνακας 4.4 : Μέσος όρος πίνακα AIR_4, 4^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_4 είναι κοντά μεταξύ τους εκτός από το CO_GT και το C6H6_GT. Πλήθος εγγραφών πίνακα AIR_4: 1391.



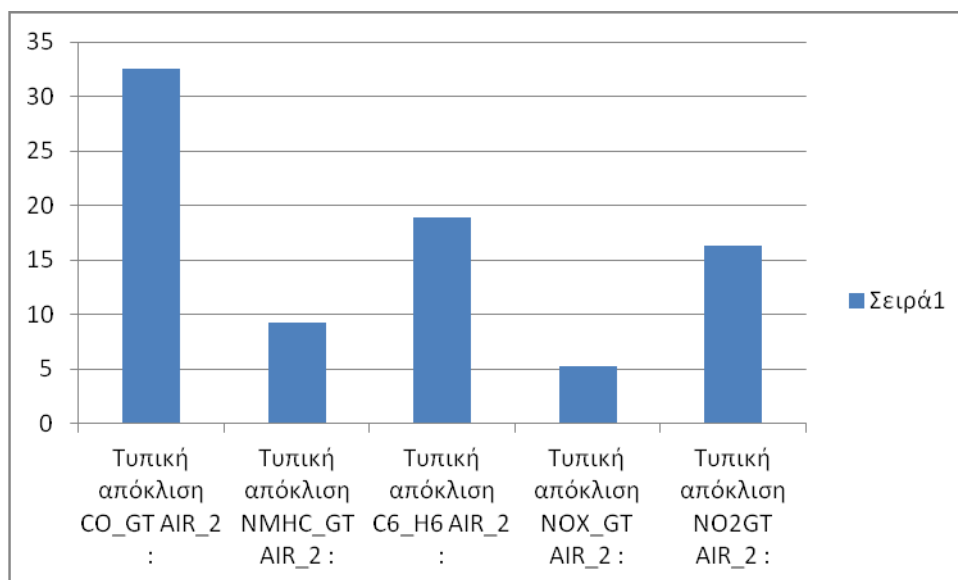
Πίνακας 4.5 : Μέσος όρος πίνακα AIR_5, 4^{ου} σετ πειραμάτων

Παρατηρούμε ότι τα αποτελέσματα του πίνακα AIR_5 είναι κοντά μεταξύ τους εκτός από το CO_GT. Πλήθος εγγραφών πίνακα AIR_5: 1349.



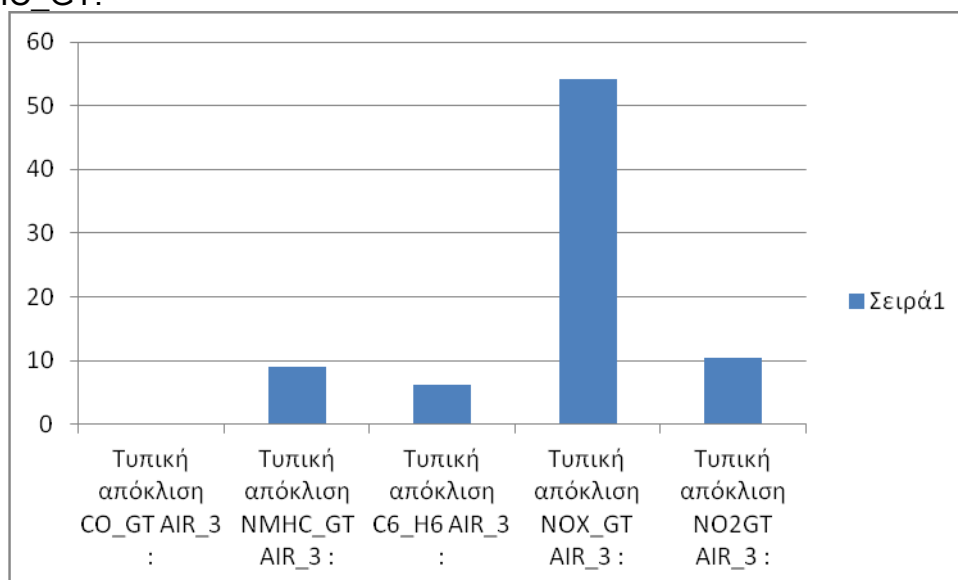
Πίνακας 4.6 : Τυπική απόκλιση πίνακα AIR_1, 4^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 4.1 του μέσου όρου η τυπική απόκλιση είναι αρκετά κοντά στις τιμές του μέσου όρου.



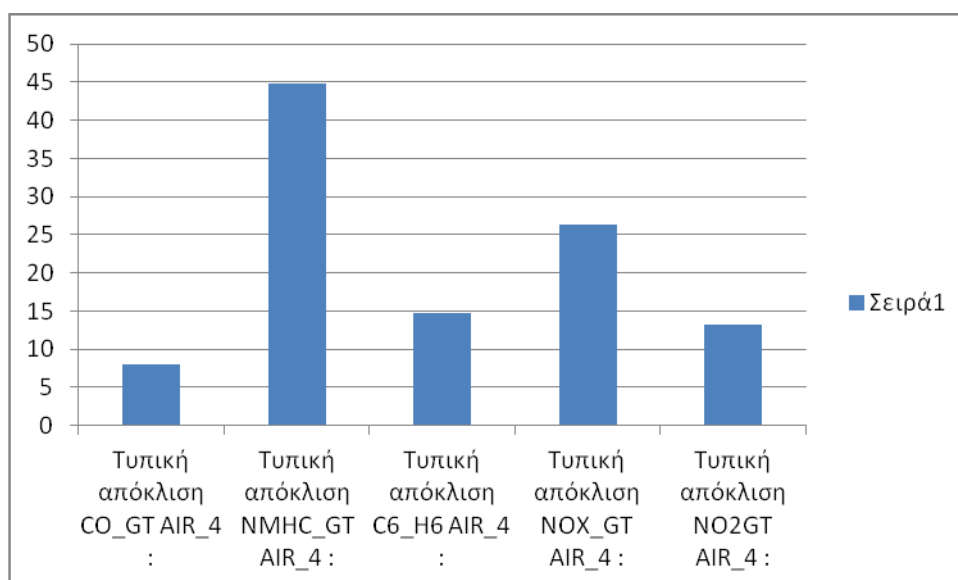
Πίνακας 4.7 : Τυπική απόκλιση πίνακα AIR_2, 4^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 4.2 του μέσου όρου η τυπική απόκλιση είναι αρκετά κοντά στις τιμές του μέσου όρου, μόνο στα πεδία CO_GT και NMHC_GT.



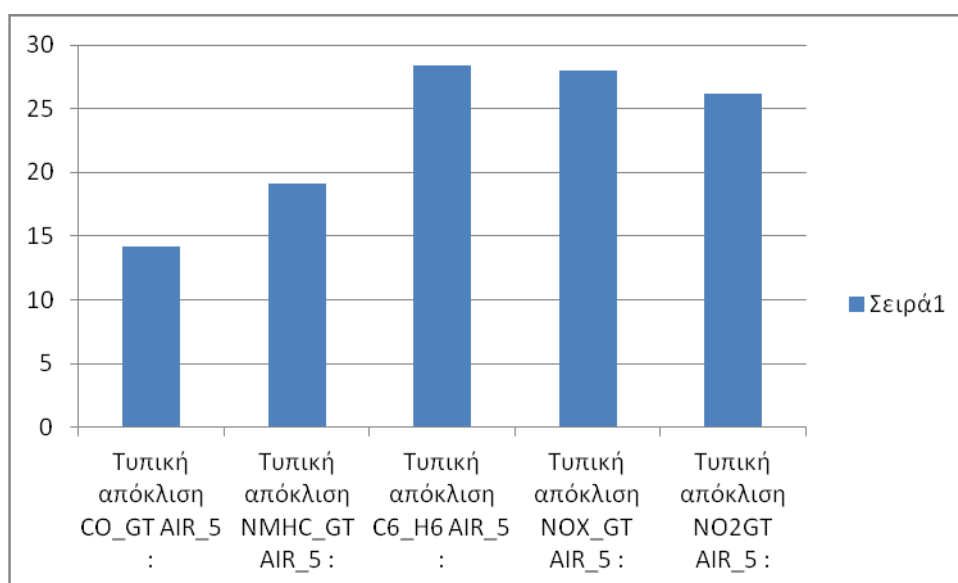
Πίνακας 4.8 : Τυπική απόκλιση πίνακα AIR_3, 4^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 4.3 του μέσου όρου η τυπική απόκλιση διαφέρει κατά πολύ στις τιμές NOX_GT και NO2GT.



Πίνακας 4.9 : Τυπική απόκλιση πίνακα AIR_4, 4^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 4.4 του μέσου όρου η τυπική απόκλιση διαφέρει κατά πολύ στις τιμές NMHC_GT και NO2GT.



Πίνακας 4.10 : Τυπική απόκλιση πίνακα AIR_5, 4^{ου} σετ πειραμάτων

Σε σύγκριση με τον πίνακα 4.5 του μέσου όρου η τυπική απόκλιση διαφέρει κατά πολύ στις τιμές NMHC_GT και C6H6_GT.

Από τους πιο πάνω πίνακες βλέπουμε ότι αυτός με τις περισσότερες εγγραφές είναι ο AIR_1 με 4782. Στον πίνακα αυτόν οι τιμές είναι κοντά μεταξύ τους τόσο στον μέσο όρο όσο και στην τυπική απόκλιση. Δηλαδή στον μέσο όρο οι τιμές κυμαίνονται μεταξύ 28,69 - 29,53 και στην τυπική απόκλιση μεταξύ 25,83 - 26,44. Δεδομένου ότι οι τιμές αυτές είναι κοντά μεταξύ τους στον 1 από τους 5 πίνακες με τις περισσότερες εγγραφές έχουμε πετύχει τον στόχο μας και για το 4^ο σετ πειραμάτων, που είναι να μαζέψουμε παραπλήσιες τιμές σε ένα σύνολο δεδομένων (Dataset). Εδώ παρατηρούμε ότι λόγω της αύξησης της τιμής του μέσου όρου και της τυπικής απόκλισης που δώσαμε κατά την εκτέλεση δηλαδή 50 και 50 αντίστοιχα αυξήθηκαν και τα αποτελέσματα των παραπλήσιων τιμών, σε σχέση με τα 3 προηγούμενα πειράματά μας. Επίσης παρατηρούμε οι περισσότερες εγγραφές υπάρχουν

σε έναν πίνακα και όχι σε 2 πίνακες όπως έγινε με τα 2 πρώτα πειράματά μας.

Τέλος μπορούμε τα 4 παραπάνω συμπεράσματα να συμπεριλάβουμε ως εξής:

Set Πειράματος	Μέσος όρος (AVG)	Τυπική απόκλιση (STDDEV)	Πίνακας με τις περισσότερες εγγραφές	Εύρος Τιμών Μέσου όρου	Εύρος Τιμών Τυπικής αποκλισης
1 ^ο	25	10	AIR_2 (4368)	12,49-13,89	8,39 - 9,12
			AIR_3 (3345)	12,84-12,98	8,44 - 8,52
2 ^ο	25	20	AIR_1 (3456)	12,77-14,02	10,86 -11,75
			AIR_5 (3947)	13,90-14,00	11,07 -11,57
3 ^ο	50	20	AIR_1 (4923)	25,46-25,86	17,04 -17,89
4 ^ο	50	50	AIR_1 (4782)	28,69 - 29,53	25,83 -26,44

Παρατηρούμε ότι όσο αυξάνονται οι δωθείσες τιμές στον μέσο όρο αυξάνονται και οι τιμές που υπολογίσθηκαν από τους πίνακες. Δηλαδή, για την δωθείσα τιμή μέσου όρου 25, παρατηρούμε στο 1^ο και στο 2^ο σετ πειραμάτων ότι οι τιμές κυμαίνονται μεταξύ 12,49 - 14,02. Για την δωθείσα τιμή μέσου όρου 50, παρατηρούμε στο 3^ο και στο 4^ο σετ πειραμάτων ότι οι τιμές κυμαίνονται μεταξύ 25,46 - 29,53, δηλαδή έναν διπλασιασμό σε σχέση με πριν, κάτι που αναμέναμε.

Επίσης παρατηρούμε ότι όσο αυξάνονται οι δωθείσες τιμές στην τυπική απόκλιση αυξάνονται και οι τιμές που υπολογίσθηκαν από τους πίνακες. Δηλαδή, για την δωθείσα τιμή τυπικής απόκλισης 10, παρατηρούμε στο 1^ο σετ πειραμάτων ότι οι τιμές κυμαίνονται μεταξύ 8,39 - 9,12. Για την δωθείσα τιμή τυπικής απόκλισης 20, παρατηρούμε στο 2^ο σετ πειραμάτων ότι οι τιμές κυμαίνονται μεταξύ 10,86 -11,75. Και τέλος για την δωθείσα τιμή τυπικής απόκλισης 50, παρατηρούμε στο 4^ο σετ πειραμάτων ότι οι τιμές κυμαίνονται μεταξύ 25,83 - 26,44, δηλαδή έναν διπλασιασμό σε σχέση με πριν, κάτι που αναμέναμε.

Παράρτημα

Ολόκληρος ο κώδικας μας φαίνεται πιο κάτω:

Κεντρική Σελίδα **Default.aspx.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Configuration;
using System.Data;
using System.Data.OleDb;
using System.IO;

namespace Insert_from_xls_to_Oracle
{
    public partial class _Default : System.Web.UI.Page
    {
        public static List<D> ListOfDoubles; //dhmiourgia listas D gia tous
        tyxaious arithmous
        public static List<double> doublelist; //orizoume th lista doublelist

        Entities1 ctx = new Entities1();//Oristhke modelo kai sthn synexeia
        entitie

        //orismos metablhtvn gia ton pinaka AIR_1
        double air1=0, air11=0, air12=0, air13=0, air14=0, air1d = 0, air1d1=
        0, air1d2 = 0, air1d3 = 0, air1d4 = 0, average1d=0, average11d, average12d,
        average13d, average14d, sumOfDerivationAverage1, sumOfDerivationAverage11,
        sumOfDerivationAverage12, sumOfDerivationAverage13, sumOfDerivationAverage14,
        aver1, aver11, aver12, aver13, aver14, stdev1, stdev11, stdev12, stdev13,
        stdev14;
        //orismos metablhtvn gia ton pinaka AIR_2
        double air2, air21, air22, air23, air24, air2d = 0, air2d1 = 0, air2d2
        = 0, air2d3 = 0, air2d4 = 0, average2d, average21d, average22d, average23d,
        average24d, sumOfDerivationAverage2, sumOfDerivationAverage21,
        sumOfDerivationAverage22, sumOfDerivationAverage23, sumOfDerivationAverage24,
        aver2, aver21, aver22, aver23, aver24, stdev2, stdev21, stdev22, stdev23,
        stdev24;
        //orismos metablhtvn gia ton pinaka AIR_3
        double air3, air31, air32, air33, air34, air3d = 0, air3d1 = 0, air3d2
        = 0, air3d3 = 0, air3d4 = 0, average3d, average31d, average32d, average33d,
        average34d, sumOfDerivationAverage3, sumOfDerivationAverage31,
        sumOfDerivationAverage32, sumOfDerivationAverage33, sumOfDerivationAverage34,
        aver3, aver31, aver32, aver33, aver34, stdev3, stdev31, stdev32, stdev33,
        stdev34;
        //orismos metablhtvn gia ton pinaka AIR_4
        double air4, air41, air42, air43, air44, air4d = 0, air4d1 = 0, air4d2
        = 0, air4d3 = 0, air4d4 = 0, average4d, average41d, average42d, average43d,
        average44d, sumOfDerivationAverage4, sumOfDerivationAverage41,
        sumOfDerivationAverage42, sumOfDerivationAverage43, sumOfDerivationAverage44,
        aver4, aver41, aver42, aver43, aver44, stdev4, stdev41, stdev42, stdev43,
        stdev44;
        //orismos metablhtvn gia ton pinaka AIR_5
```

```

        double air5, air51, air52, air53, air54, air5d = 0, air5d1 = 0, air5d2
= 0, air5d3 = 0, air5d4 = 0, average5d, average51d, average52d, average53d,
average54d, sumOfDerivationAverage5, sumOfDerivationAverage51,
sumOfDerivationAverage52, sumOfDerivationAverage53, sumOfDerivationAverage54,
aver5, aver51, aver52, aver53, aver54, stdev5, stdev51, stdev52, stdev53,
stdev54;

```

```

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        #region Upload kai Eisagvgh excel sthn bash
        //***** ARXH UPLOAD KAI EISAGVGH STH
BASH*****
        protected void upload_btn_Click(object sender, EventArgs e)
        {
            //kodikas gia to upload tou arxeiou, na sozetai prosorina sto temp
            kai meta ekteleitai me to button Import_To_DB
            if (FileUpload1.HasFile)
            {
                string FileName =
Path.GetFileName(FileUpload1.PostedFile.FileName);
                string Extension =
Path.GetExtension(FileUpload1.PostedFile.FileName);
                string FolderPath = Server.MapPath(@"\temp\");
                //ConfigurationManager.AppSettings["FolderPath"];

                string FilePath = FolderPath + FileName;
                FileUpload1.SaveAs(FilePath);
                Import_To_DB(FilePath, Extension, "YES");
            }
        }

        private void Import_To_DB(string FilePath, string Extension, string
isHDR)
        {
            string conStr = "";
            switch (Extension)
            {
                case ".xls": //Excel 97-03
                    conStr =
ConfigurationManager.ConnectionStrings["Excel03ConString"].ConnectionString;
                    break;
                case ".xlsx": //Excel 07
                    conStr =
ConfigurationManager.ConnectionStrings["Excel07ConString"].ConnectionString;
                    break;
            }
            conStr = String.Format(conStr, FilePath, isHDR);
            //dhmiourgia syndeshs me th bash
            OleDbConnection connExcel = new OleDbConnection(conStr);
            OleDbCommand cmdExcel = new OleDbCommand();
            OleDbDataAdapter oda = new OleDbDataAdapter();
            DataTable dt = new DataTable();
            cmdExcel.Connection = connExcel;

            //Get the name of First Sheet
            connExcel.Open();
            DataTable dtExcelSchema;
            dtExcelSchema =
connExcel.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, null);
            string SheetName = dtExcelSchema.Rows[0]["TABLE_NAME"].ToString();
            connExcel.Close();

```

```

//Read Data from First Sheet
connExcel.Open();
cmdExcel.CommandText = "SELECT * From [" + SheetName + "];";
oda.SelectCommand = cmdExcel;
oda.Fill(dt);
connExcel.Close();

//Diabasma dedomenavn apo to modelo tou entitie kai eisagvgh ston
pinaka

Entities1 dbentities = new Entities1();

foreach (DataRow row in dt.Rows)
{
    AIR_COPY tempObj = new AIR_COPY()
    {
        CO_GT = Convert.ToDecimal((row[0].Equals(null)) ? "" :
row[0].ToString()),
        NMHC_GT = Convert.ToDecimal((row[1].Equals(DBNull.Value))
? "" : row[1].ToString()),
        C6H6_GT = Convert.ToDecimal((row[2].Equals(DBNull.Value))
? "" : row[2].ToString()),
        NOX_GT = Convert.ToDecimal((row[3].Equals(DBNull.Value)) ?
"" : row[3].ToString()),
        NO2_GT = Convert.ToDecimal((row[4].Equals(DBNull.Value)) ?
"" : row[4].ToString()),
    };
    dbentities.AddToAIR_COPY(tempObj);
}

dbentities.SaveChanges(System.Data.Objects.SaveOptions.DetectChangesBeforeSave)
;

}

public object CO_GT { get; set; }

public object NMHC_GT { get; set; }

public object C6H6_GT { get; set; }

public object NOX_GT { get; set; }

public object NO2_GT { get; set; }

//***** TELOS UPLOAD KAI EISAGVGH STH
BASH*****
#endregion

#region Tyxaion(Random) Arithmon
//***** ARXH DHMIOURGIAS TYXAION
ARITHMON*****

public class D
{
    public double Doublenumber { get; set; }
}

static Random random = new Random();
public double GetRandomNumber(double minimum, double
maximum)//Dhmiourgia tyxaia arithmvn metaksu 2 double, minimum kai maximum.
{

    double mean = random.NextDouble() * 50; //dhlvsh meso orou

```

```

        double stdDev = random.NextDouble() * 50; //dhlvsh apoklishs
        double u1 = 1.0 - random.NextDouble(); //uniform(0,1] random
doubles
        double u2 = 1.0 - random.NextDouble(); //uniform(0,1] random
doubles
        double randStdNormal = Math.Sqrt(-2.0 * Math.Log(u1)) *
            Math.Sin(2.0 * Math.PI * u2); //random normal(0,1)
        double randNormal =
            mean + stdDev * randStdNormal; //random
normal(mean,stdDev^2)
        return Math.Max(randNormal,1); // Gia na mhn emfanizei arnhtiko
arithmo
    }

    protected void generatedRandom_Click(object sender, EventArgs e)
    {
        GetDs();
    }
    public void GetDs()//Dhmiourgia class
    {
        double[,] doubleArray = new Double[,] { { 1, 9.9 }, { 1, 99 }, { 1,
9.996 }, { 1, 999.4 }, { 1, 99.7 } }; //Dhmiourgia 2D pinaka me 5 grammes kai 2
sthles opou orizontai ta min kai max tou kathe pediou.

        ListOfDoubles = new List<D>(); //arxikopoihsh listas

        //micro-cluster

        doublelist = new List<double>();
        for (int exp = 1; exp <= 10000; exp++) // For gia na diabazontai
oi 10000 tyxaioi arithmoi
        {
            for (int i = 0; i <= 4; i++) // For gia na diabazontai h
pentada tyxaion arithmv
            {
                D tmpdouble = new D();
                tmpdouble.DoubleNumber = GetRandomNumber(doubleArray[i, 0],
doubleArray[i, 1]); //Se kathe keli tou pinaka gemizontai oi tyxaioi arithmoi

                ListOfDoubles.Add(tmpdouble);
                doublelist.Add(Math.Round((tmpdouble.DoubleNumber),
2)); //Ftiachv mia lista me onoma doublelist pou periexei ta stoixeia
ListOfDoubles me 2 dekadika
            }

            //oi 5 tyxaioi arithmoi (vector/tuple) pou paraxthkan
apothikeuontai se kathemia metablhth tou pinaka Array 5 thesevn.
            int[] Array = new int[5];

            for (int i = 0; i <= 4; i++)
            {
                Array[i] = (exp - 1) * 5 + i;
            }
            //Dhlonetai h pentada tyxaion arithmon se kathe keli toy
pinaka Array
            double xx1 =
Convert.ToDouble(ListOfDoubles[Array[0]].DoubleNumber);
            double xx2 =
Convert.ToDouble(ListOfDoubles[Array[1]].DoubleNumber);

```

```

        double xx3 =
Convert.ToDouble(ListOfDoubles[Array[2]].Doublenumber);
        double xx4 =
Convert.ToDouble(ListOfDoubles[Array[3]].Doublenumber);
        double xx5 =
Convert.ToDouble(ListOfDoubles[Array[4]].Doublenumber);

        //***** TELOS DHMIOURGIAS TYXAION
ARITHMON*****

        //***** START ΠΙΝΑΚΑΣ
AIR_1*****

        EntityDataSource1.DataBind();//syndesh tou modelou

        #region Table AIR_1

        List<AIR_1> ac1 = new List<AIR_1>();//dhmiourgia listas me
syndesh ston pinaka AIR_1 tvn dedomenvn
        ac1 = ctx.AIR_1.ToList();

        int avgac12=ac1.Count();
        foreach (AIR_1 k in ac1)//gia kathe sthlh tou pinaka pou einai
sthn lista
        {
            int avgac1 = ac1.Count();//to count kathe sthlhs
            double air1m = Convert.ToDouble(k.CO_GT);
            air1 = air1 + air1m * air1m; //ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air1d = air1d + air1m; //ginetai ypologismos athroismatos

            double air1m1 = Convert.ToDouble(k.NMHC_GT);
            air11 = air11 + air1m1 * air1m1;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air1d1 = air1d1 + air1m1;//ginetai ypologismos athroismatos

            double air1m2 = Convert.ToDouble(k.C6H6_GT);
            air12 = air12 + air1m2 * air1m2;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air1d2 = air1d2 + air1m2;//ginetai ypologismos athroismatos

            double air1m3 = Convert.ToDouble(k.NOX_GT);
            air13 = air13 + air1m3 * air1m3;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air1d3 = air1d3 + air1m3;//ginetai ypologismos athroismatos

            double air1m4 = Convert.ToDouble(k.NO2GT);
            air14 = air14 + air1m4 * air1m4;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air1d4 = air1d4 + air1m4;//ginetai ypologismos athroismatos

        }

        int ca1 = ac1.Count();//to count ths kathe sthlhs mpainei se
mia metablhth gia ton pinaka AIR_1

        clabelair_1.Text = string.Format("Πλήθος πίνακα AIR_1 :{0}",
ca1);

```

```

//*****
//Ypologismos arithmiti gia thn methodo jaccard me apolyth timh
//o kathe tyxaios arithmos afairetai me thn antistoixh synopsi
h opoia diareitai me to plththos ths gia na paroume ton meso
var jac1 = (Math.Abs(xx1 - (air1d / ca1)) + Math.Abs(xx2 -
(air1d1 / ca1)) + Math.Abs(xx3 - (air1d2 / ca1)) + Math.Abs(xx4 - (air1d3 /
ca1)) + Math.Abs(xx5 - (air1d4 / ca1)));
//Ypologimos methodos Jaccard dissimilarity
//arithmitis einai to jac1 pollaplasiasmeno me to 2 kai
paranomasths to athroisma tvn tyxaion metablhtvn pou orizetai os tyxaios
//+ to athroisma tvn synopseon tou pinaka pou orizetai par1 +
ksana to jac1
var tyxaios = xx1 + xx2 + xx3 + xx4 + xx5;
var par1 = air1d / ca1 + air1d1 / ca1 + air1d2 / ca1 + air1d3 /
ca1 + air1d4 / ca1 + jac1;
var resultAIRJAC1 = (2 * jac1) / (tyxaios + par1 + jac1);
jaclabel_1.Text = string.Format("Jaccard απόσταση πίνακα AIR_1
: {0}", resultAIRJAC1.ToString());

//*****
//Ypologimos methodos Sorensen dissimilarity
//arithmitis einai to jac1 kai paranomasths to athroisma tvn
tyxaion metablhtvn opoy orizetai os tyxaios
//+ to athroisma tvn synopseon tou pinaka opou orizetai par1
var resultAIRSOR1 = jac1 / (tyxaios + par1);

sorlabel_1.Text = string.Format("Sorensen απόσταση πίνακα AIR_1
: {0}", resultAIRSOR1.ToString());
//*****

//Ypologimos methodos Kulczynski similarity
//Prota ypologizetai to elaxisto gia kathe zeygos
double minkulair1 = Math.Min(xx1, air1d / ca1);
double minkulair11 = Math.Min(xx2, air1d1 / ca1);
double minkulair12 = Math.Min(xx3, air1d2 / ca1);
double minkulair13 = Math.Min(xx4, air1d3 / ca1);
double minkulair14 = Math.Min(xx5, air1d4 / ca1);
//Athroisma tvn elaxistvn
var kul1 = minkulair1 + minkulair11 + minkulair12 + minkulair13
+ minkulair14;

//Typos Kulczynski
var resultAIRKUL1 = 1 - 0.5 * (kul1 / tyxaios + kul1 / jac1);

kullabel_1.Text = string.Format("Kulczynski απόσταση πίνακα
AIR_1 : {0}", resultAIRKUL1.ToString());

//Ypologizetai o mesos oros tvn apotelesmaton apo Jaccard,
Sorensen kai Kulczynski
var resultAIRAVG1 = (resultAIRJAC1 + resultAIRSOR1 +
resultAIRKUL1) / 3;
airavg_1.Text = string.Format("Μέσος όρος Jaccard, Sorensen,
Kulczynski πίνακα AIR_1 : {0}", resultAIRAVG1.ToString());

//***** END ΠΙΝΑΚΑΣ
AIR_1*****

#endregion

#region Table AIR_2

//***** START ΠΙΝΑΚΑΣ
AIR_2*****

```

```

EntityDataSource2.DataBind();//syndesh tou modelou

List<AIR_2> ac2 = new List<AIR_2>();//dhmiourgia listas me
syndesh ston pinaka AIR_2 tvn dedomenvn
ac2 = ctx.AIR_2.ToList();
foreach (AIR_2 k1 in ac2)//gia kathe sthlh tou pinaka pou einai
sthn lista
{
    //ginetai ypologismos tou athroismatos ths sthlhs kai
pollaplasiasmos me ton eauto tou
    int avgac2 = ac2.Count();//to count kathe sthlhs
    double air2m = Convert.ToDouble(k1.CO_GT);
    air2 = air2 + air2m * air2m;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air2d = air2d + air2m;//ginetai ypologismos athroismatos

    double air2m1 = Convert.ToDouble(k1.NMHC_GT);
    air21 = air21 + air2m1 * air2m1;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air2d1 = air2d1 + air2m1;//ginetai ypologismos athroismatos

    double air2m2 = Convert.ToDouble(k1.C6H6_GT);
    air22 = air22 + air2m2 * air2m2;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air2d2 = air2d2 + air2m2;//ginetai ypologismos athroismatos

    double air2m3 = Convert.ToDouble(k1.NOX_GT);
    air23 = air23 + air2m3 * air2m3;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air2d3 = air2d3 + air2m3;//ginetai ypologismos athroismatos

    double air2m4 = Convert.ToDouble(k1.NO2GT);
    air24 = air24 + air2m4 * air2m4;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air2d4 = air2d4 + air2m4;//ginetai ypologismos athroismatos

}

int ca2 = ac2.Count();//to count ths kathe sthlhs mpainei se
mia metablhth gia ton pinaka AIR_2

clabelair_2.Text = string.Format("Πλήθος πίνακα AIR_2 :{0}",
ca2);

//*****
//Ypologismos arithmiti gia thn methodo jaccard me apolyth timh
h opoia diareitai me to plthos ths gia na paoume ton meso
var jac2 = (Math.Abs(xx1 - (air2d / ca2)) + Math.Abs(xx2 -
(air2d1 / ca2)) + Math.Abs(xx3 - (air2d2 / ca2)) + Math.Abs(xx4 - (air2d3 /
ca2)) + Math.Abs(xx5 - (air2d4 / ca2)));
//Ypologismos methodos Jaccard similarity
//arithmitis einai to jac2 pollaplasiasmeno me to 2 kai
paranomasths to athroisma tvn tyxaion metablhtvn pou orizetai os tyxaios
//+ to athroisma tvn synopseon tou pinaka pou orizetai par12 +
ksana to jac2
var par2 = air2d / ca2 + air2d1 / ca2 + air2d2 / ca2 + air2d3 /
ca2 + air2d4 / ca2 + jac2;
var resultAIRJAC2 = (2 * jac2) / (tyxaios + par2 + jac2);
jaclabel_2.Text = string.Format("Jaccard απόσταση πίνακα AIR_2
: {0}", resultAIRJAC2.ToString());

```

```

//*****
//Ypologimos methodos Sorensen similarity
//arithmitis einai to jac2 kai paranomasths to athroisma tvn
tyxaion metablhtvn opoy orizetai os tyxaios
//+ to athroisma tvn synopseon tou pinaka opou orizetai par2

var resultAIRSOR2 = jac2 / (tyxaios + par2);

sorlabel_2.Text = string.Format("Sorensen απόσταση πίνακα AIR_2
: {0}", resultAIRSOR2.ToString());
//*****
//Ypologimos methodos Kulczynski similarity
//Prota ypologizetai to elaxisto gia kathe zeygos
double minkulair2 = Math.Min(xx1, air2d / ca2);
double minkulair21 = Math.Min(xx2, air2d1 / ca2);
double minkulair22 = Math.Min(xx3, air2d2 / ca2);
double minkulair23 = Math.Min(xx4, air2d3 / ca2);
double minkulair24 = Math.Min(xx5, air2d4 / ca2);
//Athroisma tvn elaxistvn
var kul2 = minkulair2 + minkulair21 + minkulair22 + minkulair23
+ minkulair24;

//Typos Kulczynski
var resultAIRKUL2 = 1 - 0.5 * (kul2 / tyxaios + kul2 / jac2);
kullabel_2.Text = string.Format("Kulczynski απόσταση πίνακα
AIR_2 : {0}", resultAIRKUL2.ToString());

//Ypologizetai o mesos oros tvn apotelesmaton apo Jaccard,
Sorensen kai Kulczynski
var resultAIRAVG2 = (resultAIRJAC2 + resultAIRSOR2 +
resultAIRKUL2) / 3;
airavg_2.Text = string.Format("Μέσος όρος Jaccard, Sorensen,
Kulczynski πίνακα AIR_2 : {0}", resultAIRAVG2.ToString());

//***** END ΠΙΝΑΚΑΣ
AIR_2*****
#endregion

#region Table AIR_3
//***** START ΠΙΝΑΚΑΣ
AIR_3*****
EntityDataSource2.DataBind();//syndesh tou modelou

List<AIR_3> ac3 = new List<AIR_3>();//dhmiourgia listas me
syndesh ston pinaka AIR_3 tvn dedomenvn
ac3 = ctx.AIR_3.ToList();
foreach (AIR_3 k2 in ac3)//gia kathe sthlh tou pinaka pou einai
sthn lista
{
//ginetai ypologismos tou athroismatos ths sthlhs kai
pollaplasiasmos me ton eauto tou
int avgac3 = ac3.Count();//to count kathe sthlhs
double air3m = Convert.ToDouble(k2.CO_GT);
air3 = air3 + air3m * air3m;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
air3d = air3d + air3m;//ginetai ypologismos athroismatos

double air3m1 = Convert.ToDouble(k2.NMHC_GT);
air31 = air31 + air3m1 * air3m1;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
air3d1 = air3d1 + air3m1;//ginetai ypologismos athroismatos

```

```

        double air3m2 = Convert.ToDouble(k2.C6H6_GT);
        air32 = air32 + air3m2 * air3m2;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
        air3d2 = air3d2 + air3m2;//ginetai ypologismos athroismatos

        double air3m3 = Convert.ToDouble(k2.NOX_GT);
        air33 = air33 + air3m3 * air3m3;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
        air3d3 = air3d3 + air3m3;//ginetai ypologismos athroismatos

        double air3m4 = Convert.ToDouble(k2.NO2GT);
        air34 = air34 + air3m4 * air3m4;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
        air3d4 = air3d4 + air3m4;//ginetai ypologismos athroismatos

    }

    int ca3 = ac3.Count();//to count ths kathe sthlhs mpainei se
mia metablth gia ton pinaka AIR_3

    clabelair_3.Text = string.Format("Πλήθος πίνακα AIR_3 :{0}",
ca3);

    //*****
    //Ypologismos arithmiti gia thn methodo jaccard me apolyth timh
    //o kathe tyxaios arithmos afairetai me thn antistoixh synopsi
h opoia diareitai me to plththos ths gia na paoume ton meso
    var jac3 = (Math.Abs(xx1 - (air3d / ca3)) + Math.Abs(xx2 -
(air3d1 / ca3)) + Math.Abs(xx3 - (air3d2 / ca3)) + Math.Abs(xx4 - (air3d3 /
ca3)) + Math.Abs(xx5 - (air3d4 / ca3)));
    //Ypologismos methodos Jaccard similarity
    //arithmitis einai to jac3 pollaplasiasmeno me to 2 kai
paranomasths to athroisma tvn tyxaion metablhtvn pou orizetai os tyxaios
    //+ to athroisma tvn synopseon tou pinaka pou orizetai par3 +
ksana to jac3

    var par3 = air3d / ca3 + air3d1 / ca3 + air3d2 / ca3 + air3d3 /
ca3 + air3d4 / ca3 + jac3;
    var resultAIRJAC3 = (2 * jac3) / (tyxaios + par3 + jac3);
    jaclabel_3.Text = string.Format("Jaccard απόσταση πίνακα AIR_3
: {0}", resultAIRJAC3.ToString());

    //*****
    //Ypologismos methodos Sorensen similarity
    //arithmitis einai to jac3 kai paranomasths to athroisma tvn
tyxaion metablhtvn opoy orizetai os tyxaios
    //+ to athroisma tvn synopseon tou pinaka opou orizetai par3

    var resultAIRSOR3 = jac3 / (tyxaios + par3);

    sorlabel_3.Text = string.Format("Sorensen απόσταση πίνακα AIR_3
: {0}", resultAIRSOR3.ToString());
    //*****
    //Ypologismos methodos Kulczynski similarity
    //Prota ypologizetai to elaxisto gia kathe zeygos
    double minkulair3 = Math.Min(xx1, air3d / ca3);
    double minkulair31 = Math.Min(xx2, air3d1 / ca3);
    double minkulair32 = Math.Min(xx3, air3d2 / ca3);
    double minkulair33 = Math.Min(xx4, air3d3 / ca3);
    double minkulair34 = Math.Min(xx5, air3d4 / ca3);
    //Athroisma tvn elaxistvn
    var kul3 = minkulair3 + minkulair31 + minkulair32 + minkulair33
+ minkulair34;

```

```

        //Typos Kulczynski
        var resultAIRKUL3 = 1 - 0.5 * (kul3 / tyxaios + kul3 / jac3);
        kullabel_3.Text = string.Format("Kulczynski απόσταση πίνακα
AIR_3 : {0}", resultAIRKUL3.ToString());

        //Ypologizetai o mesos oros tvn apotelesmaton apo Jaccard,
        Sorensen kai Kulczynski
        var resultAIRAVG3 = (resultAIRJAC3 + resultAIRSOR3 +
resultAIRKUL3) / 3;
        airavg_3.Text = string.Format("Μέσος όρος Jaccard, Sorensen,
Kulczynski πίνακα AIR_3 : {0}", resultAIRAVG3.ToString());

        //***** END ΠΙΝΑΚΑΣ
AIR_3*****
        #endregion

        #region Table AIR_4
        //***** START ΠΙΝΑΚΑΣ
AIR_4*****
        EntityDataSource2.DataBind();//syndesh tou modelou

        List<AIR_4> ac4 = new List<AIR_4>();//dhmiourgia listas me
syndesh ston pinaka AIR_4 tvn dedomenvn
        ac4 = ctx.AIR_4.ToList();
        foreach (AIR_4 k3 in ac4)//gia kathe sthlh tou pinaka pou einai
sthn lista
        {
            //ginetai ypologismos tou athroismatos ths sthlhs kai
pollaplasiasmos me ton eauto tou
            int avgac4 = ac4.Count();//to count kathe sthlhs
            double air4m = Convert.ToDouble(k3.CO_GT);
            air4 = air4 + air4m * air4m;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air4d = air4d + air4m;//ginetai ypologismos athroismatos

            double air4m1 = Convert.ToDouble(k3.NMHC_GT);
            air41 = air41 + air4m1 * air4m1;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air4d1 = air4d1 + air4m1;//ginetai ypologismos athroismatos

            double air4m2 = Convert.ToDouble(k3.C6H6_GT);
            air42 = air42 + air4m2 * air4m2;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air4d2 = air4d2 + air4m2;//ginetai ypologismos athroismatos

            double air4m3 = Convert.ToDouble(k3.NOX_GT);
            air43 = air43 + air4m3 * air4m3;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air4d3 = air4d3 + air4m3;//ginetai ypologismos athroismatos

            double air4m4 = Convert.ToDouble(k3.NO2GT);
            air44 = air44 + air4m4 * air4m4;//ginetai ypologismos tou
athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
            air4d4 = air4d4 + air4m4;//ginetai ypologismos athroismatos

        }

        int ca4 = ac4.Count();//to count ths kathe sthlhs mpainei se
mia metablhth gia ton pinaka AIR_4

```

```

ca4);

        clabelair_4.Text = string.Format("Πλήθος πίνακα AIR_4 :{0}",

//*****
//Υπολογισμος arithmiti gia thn methodo jaccard me apolyth timh
//o kathe tyxaios arithmos afairetai me thn antistoixh synopsi
h opoia diareitai me to plththos ths gia na paoume ton meso
        var jac4 = (Math.Abs(xx1 - (air4d / ca4)) + Math.Abs(xx2 -
(air4d1 / ca4)) + Math.Abs(xx3 - (air4d2 / ca4)) + Math.Abs(xx4 - (air4d3 /
ca4)) + Math.Abs(xx5 - (air4d4 / ca4)));
        //Υπολογισμος methodos Jaccard similarity
        //arithmitis einai to jac4 pollaplasiasmeno me to 2 kai
paranomasths to athroisma tvn tyxaion metablhtvn pou orizetai os tyxaios
        //+ to athroisma tvn synopseon tou pinaka pou orizetai par4 +
ksana to jac4

        var par4 = air4d / ca4 + air4d1 / ca4 + air4d2 / ca4 + air4d3 /
ca4 + air4d4 / ca4 + jac4;
        var resultAIRJAC4 = (2 * jac4) / (tyxaios + par4 + jac4);
        jaclabel_4.Text = string.Format("Jaccard απόσταση πίνακα AIR_4
: {0}", resultAIRJAC4.ToString());

//*****

        //Υπολογισμος methodos Sorensen similarity
        //arithmitis einai to jac4 kai paranomasths to athroisma tvn
tyxaion metablhtvn opoy orizetai os tyxaios
        //+ to athroisma tvn synopseon tou pinaka opou orizetai par4

        var resultAIRSOR4 = jac4 / (tyxaios + par4);

        sorlabel_4.Text = string.Format("Sorensen απόσταση πίνακα AIR_4
: {0}", resultAIRSOR4.ToString());
//*****

        //Υπολογισμος methodos Kulczynski similarity
        //Prota ypologizetai to elaxisto gia kathe zeygos
double minkulair4 = Math.Min(xx1, air4d / ca4);
double minkulair41 = Math.Min(xx2, air4d1 / ca4);
double minkulair42 = Math.Min(xx3, air4d2 / ca4);
double minkulair43 = Math.Min(xx4, air4d3 / ca4);
double minkulair44 = Math.Min(xx5, air4d4 / ca4);
        //Athroisma tvn elaxistvn
var kul4 = minkulair4 + minkulair41 + minkulair42 + minkulair43
+ minkulair44;

        //Typos Kulczynski
        var resultAIRKUL4 = 1 - 0.5 * (kul4 / tyxaios + kul4 / jac4);
        kullabel_4.Text = string.Format("Kulczynski απόσταση πίνακα
AIR_4 : {0}", resultAIRKUL4.ToString());

        //Υπολογizetai o mesos oros tvn apotelesmaton apo Jaccard,
Sorensen kai Kulczynski
        var resultAIRAVG4 = (resultAIRJAC4 + resultAIRSOR4 +
resultAIRKUL4) / 3;
        airavg_4.Text = string.Format("Μέσος όρος Jaccard, Sorensen,
Kulczynski πίνακα AIR_4 : {0}", resultAIRAVG4.ToString());

//***** END ΠΙΝΑΚΑΣ
AIR_4*****

#endregion

#region Table AIR_5
//***** START ΠΙΝΑΚΑΣ
AIR_5*****

```

```

EntityDataSource2.DataBind();//syndesh tou modelou

List<AIR_5> ac5 = new List<AIR_5>();//dhmiourgia listas me
syndesh ston pinaka AIR_5 tvn dedomenvn
ac5 = ctx.AIR_5.ToList();

int air5dnew = ac5.Count();
foreach (AIR_5 k4 in ac5)//gia kathe sthlh tou pinaka pou einai
sthn lista
{
    //ginetai ypologismos tou athroismatos ths sthlhs kai
    pollaplasiasmos me ton eauto tou
    int avgac5 = ac5.Count();//to count kathe sthlhs
    double air5m = Convert.ToDouble(k4.CO_GT);
    air5 = air5 + air5m * air5m;//ginetai ypologismos tou
    athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air5d = air5d + air5m;//ginetai ypologismos athroismatos

    double air5m1 = Convert.ToDouble(k4.NMHC_GT);
    air51 = air51 + air5m1 * air5m1;//ginetai ypologismos tou
    athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air5d1 = air5d1 + air5m1;//ginetai ypologismos athroismatos

    double air5m2 = Convert.ToDouble(k4.C6H6_GT);
    air52 = air52 + air5m2 * air5m2;//ginetai ypologismos tou
    athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air5d2 = air5d2 + air5m2;//ginetai ypologismos athroismatos

    double air5m3 = Convert.ToDouble(k4.NOX_GT);
    air53 = air53 + air5m3 * air5m3;//ginetai ypologismos tou
    athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air5d3 = air5d3 + air5m3;//ginetai ypologismos athroismatos

    double air5m4 = Convert.ToDouble(k4.NO2GT);
    air54 = air54 + air5m4 * air5m4;//ginetai ypologismos tou
    athroismatos ths sthlhs kai pollaplasiasmos me ton eauto tou
    air5d4 = air5d4 + air5m4;//ginetai ypologismos athroismatos

}

int ca5 = ac5.Count();//to count ths kathe sthlhs mpainei se
mia metablth gia ton pinaka AIR_5

clabelair_5.Text = string.Format("Πλήθος πίνακα AIR_5 :{0}",
ca5);

//*****
//Ypologismos arithmiti gia thn methodo jaccard me apolyth timh
//o kathe tyxaios arithmos afairetai me thn antistoixh synopsi
h opoia diareitai me to plthos ths gia na paoume ton meso
var jac5 = (Math.Abs(xx1 - (air5d / ca5)) + Math.Abs(xx2 -
(air5d1 / ca5)) + Math.Abs(xx3 - (air5d2 / ca5)) + Math.Abs(xx4 - (air5d3 /
ca5)) + Math.Abs(xx5 - (air5d4 / ca5)));
//Ypologismos methodos Jaccard similarity
//arithmitis einai to jac5 pollaplasiasmeno me to 2 kai
paranomasths to athroisma tvn tyxaion metablhtvn pou orizetai os tyxaios
//+ to athroisma tvn synopseon tou pinaka pou orizetai par5 +
ksana to jac5

var par5 = air5d / ca5 + air5d1 / ca5 + air5d2 / ca5 + air5d3 /
ca5 + air5d4 / ca5 + jac5;
var resultAIRJAC5 = (2 * jac5) / (tyxaios + par5 + jac5);

```

```

        jaclabel_5.Text = string.Format("Jaccard απόσταση πίνακα AIR_5
: {0}", resultAIRJAC5.ToString());

        //*****

        //Ypologimos methodos Sorensen similarity
        //arithmitis einai to jac5 kai paranomasths to athroisma tvn
        tyxaion metablhtvn opoy orizetai os tyxaios
        //+ to athroisma tvn synopseon tou pinaka opou orizetai par5

        var resultAIRSOR5 = jac5 / (tyxaios + par5);

        sorlabel_5.Text = string.Format("Sorensen απόσταση πίνακα AIR_5
: {0}", resultAIRSOR5.ToString());

        //*****
        //Ypologimos methodos Kulczynski similarity
        //Prota ypologizetai to elaxisto gia kathe zeygos
        double minkulair5 = Math.Min(xx1, air5d / ca5);
        double minkulair51 = Math.Min(xx2, air5d1 / ca5);
        double minkulair52 = Math.Min(xx3, air5d2 / ca5);
        double minkulair53 = Math.Min(xx4, air5d3 / ca5);
        double minkulair54 = Math.Min(xx5, air5d4 / ca5);
        //Athroisma tvn elaxistvn
        var kul5 = minkulair5 + minkulair51 + minkulair52 + minkulair53
+ minkulair54;

        //Typos Kulczynski
        var resultAIRKUL5 = 1 - 0.5 * (kul5 / tyxaios + kul5 / jac5);
        kullabel_5.Text = string.Format("Kulczynski απόσταση πίνακα
AIR_5 : {0}", resultAIRKUL5.ToString());

        //Ypologizetai o mesos oros tvn apotelesmaton apo Jaccard,
        Sorensen kai Kulczynski
        var resultAIRAVG5 = (resultAIRJAC5 + resultAIRSOR5 +
resultAIRKUL5) / 3;
        airavg_5.Text = string.Format("Μέσος όρος Jaccard, Sorensen,
Kulczynski πίνακα AIR_5 : {0}", resultAIRAVG5.ToString());

        //***** END ΠΙΝΑΚΑΣ
AIR_5*****
        #endregion

        #region Min find

        //Elegxos me poio dataset moiazei h pleiada

        //Ευρεση του ελαχιστου απο τα 5 resultAIRAVG1,2,3,4,5
        double[] numbers = new[] { resultAIRAVG1, resultAIRAVG2,
resultAIRAVG3, resultAIRAVG4, resultAIRAVG5 };
        double minindex = numbers.Min();

        //Edv dhlvnetai h metablhtv v pou tha krataei poios pinakas tha
        exei thn mikroterh apostash, dhladh mikrotero arithmo
        int v = -1;
        for (int l = 0; l < numbers.Length; l++)
        {
            if (numbers[l] == minindex)
                v = l;
        }
        v++;
        minresult.Text = string.Format(" Πίνακας με την ελάχιστη τιμή
των μέσων όρων είναι ο : {0} και έχει {1}", v.ToString(),
minindex.ToString());

```

```

#endregion

//*****
//Topothethsh ths pleiadas se katalhlo dataset
#region Eisagvgh sth bash

DataSet ds = new DataSet();//Syndesh me thn bash

using (OleDbConnection con = new
OleDbConnection("Provider=MSDAORA;Data Source=localhost/orcl;Persist Security
Info=True;Password=hr;User ID=hr"))
{
    con.Open();
    //antikatastash toy , me thn teleia se ena string
    string XX1_dot = xx1.ToString().Replace(", ", ".");
    string XX2_dot = xx2.ToString().Replace(", ", ".");
    string XX3_dot = xx3.ToString().Replace(", ", ".");
    string XX4_dot = xx4.ToString().Replace(", ", ".");
    string XX5_dot = xx5.ToString().Replace(", ", ".");

    //Sto query pio katv fainetai h metablhth v pou deixnei se
    poio pinaka tha ginei eisagvgh
    //ta pedia opou ginetai h eggrafh kathos kai oi tyxaioi pou
    eisagontai
    string query = "INSERT INTO HR.AIR_" + v + " (CO_GT,
NMHC_GT, C6H6_GT, NOX_GT, NO2GT) VALUES (" + XX1_dot + ", " + XX2_dot + " , " +
XX3_dot + " , " + XX4_dot + " , " + XX5_dot + ")";

    Labelquery.Text = string.Format(query);
    OleDbCommand cmd = new OleDbCommand(query, con);
    int rowsUpdated = cmd.ExecuteNonQuery();
    if (rowsUpdated == 0)
        Labelinsert.Text = string.Format("Δεν έγινε εισαγωγή
εγγραφής!");
    else
        Labelinsert.Text = string.Format("Επιτυχής εγγραφή!!");

    con.Dispose();
}
#endregion

//*****

//Ypologismos statistikvn

#endregion

//Syndesh kai emfanish tou GridView me thn lista tvn tyxaion
arithvn.
GridView1.DataSource = ListOfDoubles;
GridView1.DataBind();

} //edv teleivnei to for

//Enhmerosh ths synopsis
#region Enhmerosh ths synopsis
//Prokeimenou na bgoun ta apotelesmata gia ton meso oro kai thn
typikh apoklish
//dhmiourgthike ksana gia kathe pinaka mia nea lista

```

```

        List<AIR_1> NEWac1 = new List<AIR_1>(); //dhmiourgia listas me
syndesh ston pinaka AIR_1 tvn dedomenvn
        NEWac1 = ctx.AIR_1.ToList();
        int countNewac1 = NEWac1.Count();

        double adderAIR1_cogt = 0, adderAIR1_nmhgt = 0, adderAIR1_c6gt =
0, adderAIR1_no2gt = 0, adderAIR1_noxgt = 0;
        double sumAIR1_cogt = 0, sumAIR1_nmhgt = 0, sumAIR1_c6h6gt = 0,
sumAIR1_no2gt = 0, sumAIR1_noxgt = 0;

        foreach (AIR_1 c1 in NEWac1)
        {
            //edv ginetai to athroisma ths kathe sthlhs
            adderAIR1_cogt = adderAIR1_cogt + (double)c1.CO_GT;
            adderAIR1_nmhgt = adderAIR1_nmhgt + (double)c1.NMHC_GT;
            adderAIR1_c6gt = adderAIR1_c6gt + (double)c1.C6H6_GT;
            adderAIR1_no2gt = adderAIR1_no2gt + (double)c1.NO2GT;
            adderAIR1_noxgt = adderAIR1_noxgt + (double)c1.NOX_GT;

            //edv ginetai ypologismos tou athroismatos ths sthlhs kai
pollaplasiasmos me ton eauto tou
            sumAIR1_cogt = sumAIR1_cogt + (double)c1.CO_GT *
(double)c1.CO_GT;
            sumAIR1_nmhgt = sumAIR1_nmhgt + (double)c1.NMHC_GT *
(double)c1.NMHC_GT;
            sumAIR1_c6h6gt = sumAIR1_c6h6gt + (double)c1.C6H6_GT *
(double)c1.C6H6_GT;
            sumAIR1_no2gt = sumAIR1_no2gt + (double)c1.NO2GT *
(double)c1.NO2GT;
            sumAIR1_noxgt = sumAIR1_noxgt + (double)c1.NOX_GT *
(double)c1.NOX_GT;

        }
        //Edv ypologizetai o mesos oros
        double mo_air1_cogt = adderAIR1_cogt / countNewac1;
        AVGCO_GT1.Text = string.Format("Μέσος όρος CO_GT AIR_1 :{0}",
mo_air1_cogt);

        double mo_air1_nmhgt = adderAIR1_nmhgt / countNewac1;
        AVGNMHC_GT1.Text = string.Format("Μέσος όρος NMHC_GT AIR_1
:{0}", mo_air1_nmhgt);

        double mo_air1_c6h6gt = adderAIR1_c6gt / countNewac1;
        AVGC6_H61.Text = string.Format("Μέσος όρος C6_H6 AIR_1 :{0}",
mo_air1_c6h6gt);

        double mo_air1_no2gt = adderAIR1_no2gt / countNewac1;
        AVGNOX_GT1.Text = string.Format("Μέσος όρος NOX_GT AIR_1 :{0}",
mo_air1_no2gt);

        double mo_air1_noxgt = adderAIR1_noxgt / countNewac1;
        AVGN02GT1.Text = string.Format("Μέσος όρος NO2GT AIR_1 :{0}",
mo_air1_noxgt);

        //edv ypologizetai h typikh apoklish
        double stddev_air1_cogt = sumAIR1_cogt / countNewac1;
        stdev1 = Math.Sqrt(stddev_air1_cogt - (mo_air1_cogt *
mo_air1_cogt));
        STDDEVCO_GT1.Text = string.Format("Τυπική απόκλιση CO_GT AIR_1
:{0}", stdev1);

        double stddev_air1_nmhgt = sumAIR1_nmhgt / countNewac1;
        stdev11 = Math.Sqrt(stddev_air1_nmhgt - (mo_air1_nmhgt *
mo_air1_nmhgt));
        STDDEVNMHC_GT1.Text = string.Format("Τυπική απόκλιση NMHC_GT
AIR_1 :{0}", stdev11);

```

```

        double stddev_air1_c6h6gt = sumAIR1_c6h6gt / countNewac1;
        stdev12 = Math.Sqrt(stddev_air1_c6h6gt - (mo_air1_c6h6gt *
mo_air1_c6h6gt));
        STDDEV_C6_H61.Text = string.Format("Τυπική απόκλιση C6_H6 AIR_1
:{0}", stdev12);

        double stddev_air1_noxgt = sumAIR1_noxgt / countNewac1;
        stdev13 = Math.Sqrt(stddev_air1_noxgt - (mo_air1_noxgt *
mo_air1_noxgt));
        STDDEVNOX_GT1.Text = string.Format("Τυπική απόκλιση NOX_GT AIR_1
:{0}", stdev13);

        double stddev_air1_no2gt = sumAIR1_no2gt / countNewac1;
        stdev14 = Math.Sqrt(stddev_air1_no2gt - (mo_air1_no2gt *
mo_air1_no2gt));
        STDDEVNO2GT1.Text = string.Format("Τυπική απόκλιση NO2GT AIR_1
:{0}", stdev14);
        //
        *****
        *

        List<AIR_2> NEWac2 = new List<AIR_2>(); //dhmiourgia listas me
syndesh ston pinaka AIR_2 tvn dedomenvn
        NEWac2 = ctx.AIR_2.ToList();
        int countNewac2 = NEWac2.Count();

        double adderAIR2_cogt = 0, adderAIR2_nmhgt = 0, adderAIR2_c6gt =
0, adderAIR2_no2gt = 0, adderAIR2_noxgt = 0;
        double sumAIR2_cogt = 0, sumAIR2_nmhgt = 0, sumAIR2_c6h6gt = 0,
sumAIR2_no2gt = 0, sumAIR2_noxgt = 0;

        foreach (AIR_2 c2 in NEWac2)
        {
            //edv ginetai to athroisma ths kathe sthlhs
            adderAIR2_cogt = adderAIR2_cogt + (double)c2.CO_GT;
            adderAIR2_nmhgt = adderAIR2_nmhgt + (double)c2.NMHC_GT;
            adderAIR2_c6gt = adderAIR2_c6gt + (double)c2.C6H6_GT;
            adderAIR2_no2gt = adderAIR2_no2gt + (double)c2.NO2GT;
            adderAIR2_noxgt = adderAIR2_noxgt + (double)c2.NOX_GT;

            //edv ginetai ypologismos tou athroismatos ths sthlhs kai
pollaplasiasmos me
            ton eauto tou
            sumAIR2_cogt = sumAIR2_cogt + (double)c2.CO_GT *
(double)c2.CO_GT;
            sumAIR2_nmhgt = sumAIR2_nmhgt + (double)c2.NMHC_GT *
(double)c2.NMHC_GT;
            sumAIR2_c6h6gt = sumAIR2_c6h6gt + (double)c2.C6H6_GT *
(double)c2.C6H6_GT;
            sumAIR2_no2gt = sumAIR2_no2gt + (double)c2.NO2GT *
(double)c2.NO2GT;
            sumAIR2_noxgt = sumAIR2_noxgt + (double)c2.NOX_GT *
(double)c2.NOX_GT;
        }
        //Edv ypologizetai o mesos oros
        double mo_air2_cogt = adderAIR2_cogt / countNewac2;
        AVGC_OGT2.Text = string.Format("Μέσος όρος CO_GT AIR_2 :{0}",
mo_air2_cogt);

        double mo_air2_nmhgt = adderAIR2_nmhgt / countNewac2;
        AVGNMHC_GT2.Text = string.Format("Μέσος όρος NMHC_GT AIR_2
:{0}", mo_air2_nmhgt);

        double mo_air2_c6gt = adderAIR2_c6gt / countNewac2;
        AVGC_C6_H62.Text = string.Format("Μέσος όρος C6_H6 AIR_2 :{0}",
mo_air2_c6gt);

        double mo_air2_no2gt = adderAIR2_no2gt / countNewac2;

```

```

        AVGNOX_GT2.Text = string.Format("Μέσος όρος NOX_GT AIR_2 :{0}",
mo_air2_no2gt);

        double mo_air2_noxgt = adderAIR2_noxgt / countNewac2;
        AVGN02GT2.Text = string.Format("Μέσος όρος NO2GT AIR_2 :{0}",
mo_air2_noxgt);

        //edv ypologizetai h typikh apoklish
        double stddev_air2_cogt = sumAIR2_cogt / countNewac2;
        stdev2 = Math.Sqrt(stddev_air2_cogt - (mo_air2_cogt *
mo_air2_cogt));
        STDDEVCO_GT2.Text = string.Format("Τυπική απόκλιση CO_GT AIR_2
:{0}", stdev2);

        double stddev_air2_nmhgt = sumAIR2_nmhgt / countNewac2;
        stdev21 = Math.Sqrt(stddev_air2_nmhgt - (mo_air2_nmhgt *
mo_air2_nmhgt));
        STDDEVNMHC_GT2.Text = string.Format("Τυπική απόκλιση NMHC_GT
AIR_2 :{0}", stdev21);

        double stddev_air2_c6h6gt = sumAIR2_c6h6gt / countNewac2;
        stdev22 = Math.Sqrt(stddev_air2_c6h6gt - (mo_air2_c6gt *
mo_air2_c6gt));
        STDDEVC6_H62.Text = string.Format("Τυπική απόκλιση C6_H6 AIR_2
:{0}", stdev22);

        double stddev_air2_noxgt = sumAIR2_noxgt / countNewac2;
        stdev23 = Math.Sqrt(stddev_air2_noxgt - (mo_air2_noxgt *
mo_air2_noxgt));
        STDDEVNOX_GT2.Text = string.Format("Τυπική απόκλιση NOX_GT AIR_2
:{0}", stdev23);

        double stddev_air2_no2gt = sumAIR2_no2gt / countNewac2;
        stdev24 = Math.Sqrt(stddev_air2_no2gt - (mo_air2_no2gt *
mo_air2_no2gt));
        STDDEVNO2GT2.Text = string.Format("Τυπική απόκλιση NO2GT AIR_2
:{0}", stdev24);
        //
        *****
        *

        List<AIR_3> NEWac3 = new List<AIR_3>();//dhmiourgia listas me
syndesh ston pinaka AIR_3 tvn dedomenvn
        NEWac3 = ctx.AIR_3.ToList();
        int countNewac3 = NEWac3.Count();

        double adderAIR3_cogt = 0, adderAIR3_nmhgt = 0, adderAIR3_c6gt =
0, adderAIR3_no2gt = 0, adderAIR3_noxgt = 0;
        double sumAIR3_cogt = 0, sumAIR3_nmhgt = 0, sumAIR3_c6h6gt = 0,
sumAIR3_no2gt = 0, sumAIR3_noxgt = 0;

        foreach (AIR_3 c3 in NEWac3)
        {
            //edv ginetai to athroisma ths kathe sthlhs
            adderAIR3_cogt = adderAIR3_cogt + (double)c3.CO_GT;
            adderAIR3_nmhgt = adderAIR3_nmhgt + (double)c3.NMHC_GT;
            adderAIR3_c6gt = adderAIR3_c6gt + (double)c3.C6H6_GT;
            adderAIR3_no2gt = adderAIR3_no2gt + (double)c3.NO2GT;
            adderAIR3_noxgt = adderAIR3_noxgt + (double)c3.NOX_GT;

            //edv ginetai ypologismos tou athroismatos ths sthlhs kai
pollaplasiasmos me ton eauto tou
            sumAIR3_cogt = sumAIR3_cogt + (double)c3.CO_GT *
(double)c3.CO_GT;
            sumAIR3_nmhgt = sumAIR3_nmhgt + (double)c3.NMHC_GT *
(double)c3.NMHC_GT;

```

```

sumAIR3_c6h6gt = sumAIR3_c6h6gt + (double)c3.C6H6_GT *
(double)c3.C6H6_GT;
sumAIR3_no2gt = sumAIR3_no2gt + (double)c3.NO2GT *
(double)c3.NO2GT;
sumAIR3_noxgt = sumAIR3_noxgt + (double)c3.NOX_GT *
(double)c3.NOX_GT;

}
//Edv ypologizetai o mesos oros
double mo_air3_cogt = adderAIR3_cogt / countNewac3;
AVGCO_GT3.Text = string.Format("Μέσος όρος CO_GT AIR_3 :{0}",
mo_air3_cogt);
double mo_air3_nmhgt = adderAIR3_nmhgt / countNewac3;
AVGNMHC_GT3.Text = string.Format("Μέσος όρος NMHC_GT AIR_3
:{0}", mo_air3_nmhgt);
double mo_air3_c6gt = adderAIR3_c6gt / countNewac3;
AVGC6_H63.Text = string.Format("Μέσος όρος C6_H6 AIR_3 :{0}",
mo_air3_c6gt);
double mo_air3_no2gt = adderAIR3_no2gt / countNewac3;
AVGNOX_GT3.Text = string.Format("Μέσος όρος NOX_GT AIR_3 :{0}",
mo_air3_no2gt);
double mo_air3_noxgt = adderAIR3_noxgt / countNewac3;
AVGNO2GT3.Text = string.Format("Μέσος όρος NO2GT AIR_3 :{0}",
mo_air3_noxgt);

//edv ypologizetai h typikh αποκλιση
double stddev_air3_cogt = sumAIR3_cogt / countNewac3;
stddev3 = Math.Sqrt(stddev_air3_cogt - (mo_air3_cogt *
mo_air3_cogt));
STDDEVCO_GT3.Text = string.Format("Τυπική απόκλιση CO_GT AIR_3
:{0}", stddev3);

double stddev_air3_nmhgt = sumAIR3_nmhgt / countNewac3;
stddev31 = Math.Sqrt(stddev_air3_nmhgt - (mo_air3_nmhgt *
mo_air3_nmhgt));
STDDEVNMHC_GT3.Text = string.Format("Τυπική απόκλιση NMHC_GT
AIR_3 :{0}", stddev31);

double stddev_air3_c6h6gt = sumAIR3_c6h6gt / countNewac3;
stddev32 = Math.Sqrt(stddev_air3_c6h6gt - (mo_air3_c6gt *
mo_air3_c6gt));
STDDEVCO6_H63.Text = string.Format("Τυπική απόκλιση C6_H6 AIR_3
:{0}", stddev32);

double stddev_air3_noxgt = sumAIR3_noxgt / countNewac3;
stddev33 = Math.Sqrt(stddev_air3_noxgt - (mo_air3_noxgt *
mo_air3_noxgt));
STDDEVNOX_GT3.Text = string.Format("Τυπική απόκλιση NOX_GT AIR_3
:{0}", stddev33);

double stddev_air3_no2gt = sumAIR3_no2gt / countNewac3;
stddev34 = Math.Sqrt(stddev_air3_no2gt - (mo_air3_no2gt *
mo_air3_no2gt));
STDDEVNO2GT3.Text = string.Format("Τυπική απόκλιση NO2GT AIR_3
:{0}", stddev34);
//
*****
*

List<AIR_4> NEWac4 = new List<AIR_4>(); //dhmiourgia listas me
syndesh ston pinaka AIR_4 tvn dedomenvn
NEWac4 = ctx.AIR_4.ToList();
int countNewac4 = NEWac4.Count();

double adderAIR4_cogt = 0, adderAIR4_nmhgt = 0, adderAIR4_c6gt =
0, adderAIR4_no2gt = 0, adderAIR4_noxgt = 0;

```

```

double sumAIR4_cogt = 0, sumAIR4_nmhgt = 0, sumAIR4_c6h6gt = 0,
sumAIR4_no2gt = 0, sumAIR4_noxgt = 0;

foreach (AIR_4 c4 in NEWac4)
{
    //edv ginetai to athroisma ths kathe sthlhs
    adderAIR4_cogt = adderAIR4_cogt + (double)c4.CO_GT;
    adderAIR4_nmhgt = adderAIR4_nmhgt + (double)c4.NMHC_GT;
    adderAIR4_c6gt = adderAIR4_c6gt + (double)c4.C6H6_GT;
    adderAIR4_no2gt = adderAIR4_no2gt + (double)c4.NO2GT;
    adderAIR4_noxgt = adderAIR4_noxgt + (double)c4.NOX_GT;

    //edv ginetai ypologismos tou athroismatos ths sthlhs kai
    pollaplasiasmos me ton eauto tou
    sumAIR4_cogt = sumAIR4_cogt + (double)c4.CO_GT *
(double)c4.CO_GT;
    sumAIR4_nmhgt = sumAIR4_nmhgt + (double)c4.NMHC_GT *
(double)c4.NMHC_GT;
    sumAIR4_c6h6gt = sumAIR4_c6h6gt + (double)c4.C6H6_GT *
(double)c4.C6H6_GT;
    sumAIR4_no2gt = sumAIR4_no2gt + (double)c4.NO2GT *
(double)c4.NO2GT;
    sumAIR4_noxgt = sumAIR4_noxgt + (double)c4.NOX_GT *
(double)c4.NOX_GT;
}
//Edv ypologizetai o mesos oros
double mo_air4_cogt = adderAIR4_cogt / countNewac4;
AVGCO_GT4.Text = string.Format("Μέσος όρος CO_GT AIR_4 :{0}",
mo_air4_cogt);
double mo_air4_nmhgt = adderAIR4_nmhgt / countNewac4;
AVGNMHC_GT4.Text = string.Format("Μέσος όρος NMHC_GT AIR_4
:{0}", mo_air4_nmhgt);
double mo_air4_c6gt = adderAIR4_c6gt / countNewac4;
AVGC6_H64.Text = string.Format("Μέσος όρος C6_H6 AIR_4 :{0}",
mo_air4_c6gt);
double mo_air4_noxgt = adderAIR4_no2gt / countNewac4;
AVGNOX_GT4.Text = string.Format("Μέσος όρος NOX_GT AIR_4 :{0}",
mo_air4_noxgt);
double mo_air4_no2gt = adderAIR4_noxgt / countNewac4;
AVGNO2GT4.Text = string.Format("Μέσος όρος NO2GT AIR_4 :{0}",
mo_air4_no2gt);

//edv ypologizetai h typikh apoklish
double stddev_air4_cogt = sumAIR4_cogt / countNewac4;
stddev4 = Math.Sqrt(stddev_air4_cogt - (mo_air4_cogt *
mo_air4_cogt));
STDDEVCO_GT4.Text = string.Format("Τυπική απόκλιση CO_GT AIR_4
:{0}", stddev4);

double stddev_air4_nmhgt = sumAIR4_nmhgt / countNewac4;
stddev41 = Math.Sqrt(stddev_air4_nmhgt - (mo_air4_nmhgt *
mo_air4_nmhgt));
STDDEVNMHC_GT4.Text = string.Format("Τυπική απόκλιση NMHC_GT
AIR_4 :{0}", stddev41);

double stddev_air4_c6h6gt = sumAIR4_c6h6gt / countNewac4;
stddev42 = Math.Sqrt(stddev_air4_c6h6gt - (mo_air4_c6gt *
mo_air4_c6gt));
STDDEVCO6_H64.Text = string.Format("Τυπική απόκλιση C6_H6 AIR_4
:{0}", stddev42);

double stddev_air4_noxgt = sumAIR4_noxgt / countNewac4;
stddev43 = Math.Sqrt(stddev_air4_noxgt - (mo_air4_noxgt *
mo_air4_noxgt));
STDDEVNOX_GT4.Text = string.Format("Τυπική απόκλιση NOX_GT AIR_4
:{0}", stddev43);

double stddev_air4_no2gt = sumAIR4_no2gt / countNewac4;

```

```

        stdev44 = Math.Sqrt(stddev_air4_no2gt - (mo_air4_no2gt *
mo_air4_no2gt));
        STDDEVNO2GT4.Text = string.Format("Τυπική απόκλιση NO2GT AIR_4
:{0}", stdev44);
        //
        *****
        *

        List<AIR_5> NEWac5 = new List<AIR_5>();//dhmiourgia listas me
syndesh ston pinaka AIR_5 tvn dedomenvn
        NEWac5 = ctx.AIR_5.ToList();
        int countNewac5 = NEWac5.Count();

        double adderAIR5_cogt = 0, adderAIR5_nmhgt = 0,
adderAIR5_c6gt=0, adderAIR5_no2gt=0, adderAIR5_noxgt=0;
        double sumAIR5_cogt = 0, sumAIR5_nmhgt = 0, sumAIR5_c6h6gt = 0,
sumAIR5_no2gt = 0, sumAIR5_noxgt = 0;

        foreach (AIR_5 c5 in NEWac5)
        {
            //edv ginetai to athroisma ths kathe sthlhs
            adderAIR5_cogt = adderAIR5_cogt + (double)c5.CO_GT;
            adderAIR5_nmhgt = adderAIR5_nmhgt + (double)c5.NMHC_GT;
            adderAIR5_c6gt = adderAIR5_c6gt + (double)c5.C6H6_GT;
            adderAIR5_no2gt = adderAIR5_no2gt + (double)c5.NO2GT;
            adderAIR5_noxgt = adderAIR5_noxgt + (double)c5.NOX_GT;

            //edv ginetai ypologismos tou athroismatos ths sthlhs kai
pollaplasiasmos me ton eauto tou
            sumAIR5_cogt = sumAIR5_cogt + (double)c5.CO_GT *
(double)c5.CO_GT;
            sumAIR5_nmhgt = sumAIR5_nmhgt + (double)c5.NMHC_GT *
(double)c5.NMHC_GT;
            sumAIR5_c6h6gt = sumAIR5_c6h6gt + (double)c5.C6H6_GT *
(double)c5.C6H6_GT;
            sumAIR5_no2gt = sumAIR5_no2gt + (double)c5.NO2GT *
(double)c5.NO2GT;
            sumAIR5_noxgt = sumAIR5_noxgt + (double)c5.NOX_GT *
(double)c5.NOX_GT;
        }
        //Edv ypologizetai o mesos oros
        double mo_air5_cogt = adderAIR5_cogt / countNewac5;
        AVGCO_GT5.Text = string.Format("Μέσος όρος CO_GT AIR_5 :{0}",
mo_air5_cogt);

        double mo_air5_nmhgt = adderAIR5_nmhgt / countNewac5;
        AVGNMHC_GT5.Text = string.Format("Μέσος όρος NMHC_GT AIR_5
:{0}", mo_air5_nmhgt);

        double mo_air5_c6gt = adderAIR5_c6gt / countNewac5;
        AVGC6_H65.Text = string.Format("Μέσος όρος C6_H6 AIR_5 :{0}",
mo_air5_c6gt);

        double mo_air5_noxgt = adderAIR5_no2gt / countNewac5;
        AVGNOX_GT5.Text = string.Format("Μέσος όρος NOX_GT AIR_5 :{0}",
mo_air5_noxgt);

        double mo_air5_no2gt = adderAIR5_noxgt / countNewac5;
        AVGN02GT5.Text = string.Format("Μέσος όρος NO2GT AIR_5 :{0}",
mo_air5_no2gt);

        //edv ypologizetai h typikh apoklish
        double stddev_air5_cogt = sumAIR5_cogt / countNewac5;
        stdev5 = Math.Sqrt(stddev_air5_cogt - (mo_air5_cogt *
mo_air5_cogt));
        STDDEVCO_GT5.Text = string.Format("Τυπική απόκλιση CO_GT AIR_5
:{0}", stdev5);

```

```

        double stddev_air5_nmhgt = sumAIR5_nmhgt / countNewac5;
        stdev51 = Math.Sqrt(stddev_air5_nmhgt - (mo_air5_nmhgt *
mo_air5_nmhgt));
        STDDEVNMHC_GT5.Text = string.Format("Τυπική απόκλιση NMHC_GT
AIR_5 :{0}", stdev51);

        double stddev_air5_c6h6gt = sumAIR5_c6h6gt / countNewac5;
        stdev52 = Math.Sqrt(stddev_air5_c6h6gt - (mo_air5_c6gt *
mo_air5_c6gt));
        STDDEVC6_H65.Text = string.Format("Τυπική απόκλιση C6_H6 AIR_5
:{0}", stdev52);

        double stddev_air5_noxgt = sumAIR5_noxgt / countNewac5;
        stdev53 = Math.Sqrt(stddev_air5_noxgt - (mo_air5_noxgt *
mo_air5_noxgt));
        STDDEVNOX_GT5.Text = string.Format("Τυπική απόκλιση NOX_GT
AIR_5 :{0}", stdev53);

        double stddev_air5_no2gt = sumAIR5_no2gt / countNewac5;
        stdev54 = Math.Sqrt(stddev_air5_no2gt - (mo_air5_no2gt *
mo_air5_no2gt));
        STDDEVNO2GT5.Text = string.Format("Τυπική απόκλιση NO2GT AIR_5
:{0}", stdev54);
        //
*****
*

        #endregion

    } //edv teleivnei h GetDs class pou trexei to generatedRandom_Click
button
}

}

```

Ο κώδικας της σελίδας **Web.config** φαίνεται πιο κάτω:

```

<?xml version="1.0" encoding="utf-8"?>
<!--
    For more information on how to configure your ASP.NET application, please
    visit
    http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <connectionStrings>
    <add name="Excel03ConString"
connectionString="Provider=Microsoft.Jet.OLEDB.4.0;Data Source={0};Extended
Properties='Excel 8.0;HDR={1}'" />
    <add name="Excel07ConString"
connectionString="Provider=Microsoft.ACE.OLEDB.12.0;Data Source={0};Extended
Properties='Excel 8.0;HDR={1}'" />
    <add name="ApplicationServices" connectionString="data
source=.\SQLEXPRESS;Integrated
Security=SSPI;AttachDBFilename=|DataDirectory|\aspnetdb.mdf;User Instance=true"
providerName="System.Data.SqlClient" />

```

```

    <add name="ConnectionString2" connectionString="DATA
SOURCE=localhost/orcl;PASSWORD=hr;PERSIST SECURITY INFO=True;USER ID=hr"
providerName="Oracle.DataAccess.Client" />
    <add name="ConnectionString1" connectionString="Provider=MSDAORA;Data
Source=localhost/orcl;Persist Security Info=True;Password=hr;User ID=hr"
providerName="System.Data.OleDb" />
    <add name="Entities"
connectionString="metadata=res://*/OModel.csdl|res://*/OModel.ssdl|res://*/OMod
el.msl;provider=Oracle.DataAccess.Client;provider connection string="DATA
SOURCE=localhost/orcl;PASSWORD=hr;PERSIST SECURITY INFO=True;USER ID=HR";"
providerName="System.Data.EntityClient" />
    <add name="Entities1"
connectionString="metadata=res://*/OModel.csdl|res://*/OModel.ssdl|res://*/OMod
el.msl;provider=Oracle.DataAccess.Client;provider connection string="DATA
SOURCE=LOCALHOST/ORCL;PASSWORD=hr;PERSIST SECURITY INFO=True;USER ID=HR";"
providerName="System.Data.EntityClient" />
</connectionStrings>
<system.web>
    <compilation debug="true" targetFramework="4.0">
        <assemblies>
            <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
        </assemblies>
    </compilation>
    <authentication mode="Forms">
        <forms loginUrl="~/Account/Login.aspx" timeout="2880" />
    </authentication>
    <membership>
        <providers>
            <clear />
            <add name="AspNetSqlMembershipProvider"
type="System.Web.Security.SqlMembershipProvider"
connectionStringName="ApplicationServices" enablePasswordRetrieval="false"
enablePasswordReset="true" requiresQuestionAndAnswer="false"
requiresUniqueEmail="false" maxInvalidPasswordAttempts="5"
minRequiredPasswordLength="6" minRequiredNonalphanumericCharacters="0"
passwordAttemptWindow="10" applicationName="/" />
        </providers>
    </membership>
    <profile>
        <providers>
            <clear />
            <add name="AspNetSqlProfileProvider"
type="System.Web.Profile.SqlProfileProvider"
connectionStringName="ApplicationServices" applicationName="/" />
        </providers>
    </profile>
    <roleManager enabled="false">
        <providers>
            <clear />
            <add name="AspNetSqlRoleProvider"
type="System.Web.Security.SqlRoleProvider"
connectionStringName="ApplicationServices" applicationName="/" />
            <add name="AspNetWindowsTokenRoleProvider"
type="System.Web.Security.WindowsTokenRoleProvider" applicationName="/" />
        </providers>
    </roleManager>
</system.web>
<system.webServer>
    <modules runAllManagedModulesForAllRequests="true" />
</system.webServer>
</configuration>

```

Τέλος ο κώδικας για το μοντέλο μας που ήταν απαραίτητο για να μπορέσουμε να συνδέσουμε τους πίνακες με την Oracle είναι ως εξής:

```

//-----
-
// <auto-generated>
//   This code was generated from a template.
//
//   Manual changes to this file may cause unexpected behavior in your
//   application.
//   Manual changes to this file will be overwritten if the code is
//   regenerated.
// </auto-generated>
//-----
-

using System;
using System.ComponentModel;
using System.Data.EntityClient;
using System.Data.Objects;
using System.Data.Objects.DataClasses;
using System.Linq;
using System.Runtime.Serialization;
using System.Xml.Serialization;

[assembly: EdmSchemaAttribute()]
namespace Insert_from_xls_to_Oracle
{
    #region Contexts

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    public partial class Entities1 :ObjectContext
    {
        #region Constructors

        /// <summary>
        /// Initializes a new Entities1 object using the connection string
        found in the 'Entities1' section of the application configuration file.
        /// </summary>
        public Entities1() : base("name=Entities1", "Entities1")
        {
            this.ContextOptions.LazyLoadingEnabled = true;
            OnContextCreated();
        }

        /// <summary>
        /// Initialize a new Entities1 object.
        /// </summary>
        public Entities1(string connectionString) : base(connectionString,
"Entities1")
        {
            this.ContextOptions.LazyLoadingEnabled = true;
            OnContextCreated();
        }

        /// <summary>
        /// Initialize a new Entities1 object.
        /// </summary>
        public Entities1(EntityConnection connection) : base(connection,
"Entities1")
        {
            this.ContextOptions.LazyLoadingEnabled = true;
            OnContextCreated();
        }

        #endregion

        #region Partial Methods

```

```

partial void OnContextCreated();

#endregion

#region ObjectSet Properties

/// <summary>
/// No Metadata Documentation available.
/// </summary>
public ObjectSet<AIR_1> AIR_1
{
    get
    {
        if ((_AIR_1 == null))
        {
            _AIR_1 = base.CreateObjectSet<AIR_1>("AIR_1");
        }
        return _AIR_1;
    }
}
private ObjectSet<AIR_1> _AIR_1;

/// <summary>
/// No Metadata Documentation available.
/// </summary>
public ObjectSet<AIR_2> AIR_2
{
    get
    {
        if ((_AIR_2 == null))
        {
            _AIR_2 = base.CreateObjectSet<AIR_2>("AIR_2");
        }
        return _AIR_2;
    }
}
private ObjectSet<AIR_2> _AIR_2;

/// <summary>
/// No Metadata Documentation available.
/// </summary>
public ObjectSet<AIR_3> AIR_3
{
    get
    {
        if ((_AIR_3 == null))
        {
            _AIR_3 = base.CreateObjectSet<AIR_3>("AIR_3");
        }
        return _AIR_3;
    }
}
private ObjectSet<AIR_3> _AIR_3;

/// <summary>
/// No Metadata Documentation available.
/// </summary>
public ObjectSet<AIR_4> AIR_4
{
    get
    {
        if ((_AIR_4 == null))
        {
            _AIR_4 = base.CreateObjectSet<AIR_4>("AIR_4");
        }
        return _AIR_4;
    }
}

```

```

    }
}
private ObjectSet<AIR_4> _AIR_4;

/// <summary>
/// No Metadata Documentation available.
/// </summary>
public ObjectSet<AIR_5> AIR_5
{
    get
    {
        if ((_AIR_5 == null))
        {
            _AIR_5 = base.CreateObjectSet<AIR_5>("AIR_5");
        }
        return _AIR_5;
    }
}
private ObjectSet<AIR_5> _AIR_5;

/// <summary>
/// No Metadata Documentation available.
/// </summary>
public ObjectSet<AIR_COPY> AIR_COPY
{
    get
    {
        if ((_AIR_COPY == null))
        {
            _AIR_COPY = base.CreateObjectSet<AIR_COPY>("AIR_COPY");
        }
        return _AIR_COPY;
    }
}
private ObjectSet<AIR_COPY> _AIR_COPY;

#endregion

#region AddTo Methods

/// <summary>
/// Deprecated Method for adding a new object to the AIR_1 EntitySet.
Consider using the .Add method of the associated ObjectSet<T> property
instead.
/// </summary>
public void AddToAIR_1(AIR_1 aIR_1)
{
    base.AddObject("AIR_1", aIR_1);
}

/// <summary>
/// Deprecated Method for adding a new object to the AIR_2 EntitySet.
Consider using the .Add method of the associated ObjectSet<T> property
instead.
/// </summary>
public void AddToAIR_2(AIR_2 aIR_2)
{
    base.AddObject("AIR_2", aIR_2);
}

/// <summary>
/// Deprecated Method for adding a new object to the AIR_3 EntitySet.
Consider using the .Add method of the associated ObjectSet<T> property
instead.
/// </summary>
public void AddToAIR_3(AIR_3 aIR_3)
{

```

```

        base.AddObject("AIR_3", aIR_3);
    }

    /// <summary>
    /// Deprecated Method for adding a new object to the AIR_4 EntitySet.
    Consider using the .Add method of the associated ObjectSet<T> property
    instead.
    /// </summary>
    public void AddToAIR_4(AIR_4 aIR_4)
    {
        base.AddObject("AIR_4", aIR_4);
    }

    /// <summary>
    /// Deprecated Method for adding a new object to the AIR_5 EntitySet.
    Consider using the .Add method of the associated ObjectSet<T> property
    instead.
    /// </summary>
    public void AddToAIR_5(AIR_5 aIR_5)
    {
        base.AddObject("AIR_5", aIR_5);
    }

    /// <summary>
    /// Deprecated Method for adding a new object to the AIR_COPY
    EntitySet. Consider using the .Add method of the associated ObjectSet<T>
    property instead.
    /// </summary>
    public void AddToAIR_COPY(AIR_COPY aIR_COPY)
    {
        base.AddObject("AIR_COPY", aIR_COPY);
    }

    #endregion

}

#endregion

#region Entities

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmEntityTypeAttribute(NamespaceName="Model", Name="AIR_1")]
    [Serializable()]
    [DataContractAttribute(IsReference=true)]
    public partial class AIR_1 : EntityObject
    {
        #region Factory Method

        /// <summary>
        /// Create a new AIR_1 object.
        /// </summary>
        /// <param name="id">Initial value of the ID property.</param>
        public static AIR_1 CreateAIR_1(global::System.Int32 id)
        {
            AIR_1 aIR_1 = new AIR_1();
            aIR_1.ID = id;
            return aIR_1;
        }

        #endregion

        #region Primitive Properties

        /// <summary>

```

```

/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> CO_GT
{
    get
    {
        return _CO_GT;
    }
    set
    {
        OnCO_GTChanging(value);
        ReportPropertyChanging("CO_GT");
        _CO_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("CO_GT");
        OnCO_GTChanged();
    }
}
private Nullable<global::System.Decimal> _CO_GT;
partial void OnCO_GTChanging(Nullable<global::System.Decimal> value);
partial void OnCO_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NMHC_GT
{
    get
    {
        return _NMHC_GT;
    }
    set
    {
        OnNMHC_GTChanging(value);
        ReportPropertyChanging("NMHC_GT");
        _NMHC_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NMHC_GT");
        OnNMHC_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NMHC_GT;
partial void OnNMHC_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNMHC_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> C6H6_GT
{
    get
    {
        return _C6H6_GT;
    }
    set
    {
        OnC6H6_GTChanging(value);
        ReportPropertyChanging("C6H6_GT");
        _C6H6_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("C6H6_GT");
        OnC6H6_GTChanged();
    }
}
}

```

```

private Nullable<global::System.Decimal> _C6H6_GT;
partial void OnC6H6_GTChanging(Nullable<global::System.Decimal> value);
partial void OnC6H6_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NOX_GT
{
    get
    {
        return _NOX_GT;
    }
    set
    {
        OnNOX_GTChanging(value);
        ReportPropertyChanging("NOX_GT");
        _NOX_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NOX_GT");
        OnNOX_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NOX_GT;
partial void OnNOX_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNOX_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NO2GT
{
    get
    {
        return _NO2GT;
    }
    set
    {
        OnNO2GTChanging(value);
        ReportPropertyChanging("NO2GT");
        _NO2GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NO2GT");
        OnNO2GTChanged();
    }
}
private Nullable<global::System.Decimal> _NO2GT;
partial void OnNO2GTChanging(Nullable<global::System.Decimal> value);
partial void OnNO2GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=true, IsNullable=false)]
[DataMemberAttribute()]
public global::System.Int32 ID
{
    get
    {
        return _ID;
    }
    set
    {
        if (_ID != value)
        {

```

```

        OnIDChanging(value);
        ReportPropertyChanging("ID");
        _ID = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("ID");
        OnIDChanged();
    }
}
}
private global::System.Int32 _ID;
partial void OnIDChanging(global::System.Int32 value);
partial void OnIDChanged();

#endregion

}

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmEntityTypeAttribute(NamespaceName="Model", Name="AIR_2")]
[Serializable()]
[DataContractAttribute(IsReference=true)]
public partial class AIR_2 : EntityObject
{
    #region Factory Method

    /// <summary>
    /// Create a new AIR_2 object.
    /// </summary>
    /// <param name="id">Initial value of the ID property.</param>
    public static AIR_2 CreateAIR_2(global::System.Int32 id)
    {
        AIR_2 aIR_2 = new AIR_2();
        aIR_2.ID = id;
        return aIR_2;
    }

    #endregion

    #region Primitive Properties

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
    [DataMemberAttribute()]
    public Nullable<global::System.Decimal> CO_GT
    {
        get
        {
            return _CO_GT;
        }
        set
        {
            OnCO_GTChanging(value);
            ReportPropertyChanging("CO_GT");
            _CO_GT = StructuralObject.SetValidValue(value);
            ReportPropertyChanged("CO_GT");
            OnCO_GTChanged();
        }
    }
    private Nullable<global::System.Decimal> _CO_GT;
    partial void OnCO_GTChanging(Nullable<global::System.Decimal> value);
    partial void OnCO_GTChanged();

    /// <summary>

```

```

/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NMHC_GT
{
    get
    {
        return _NMHC_GT;
    }
    set
    {
        OnNMHC_GTChanging(value);
        ReportPropertyChanging("NMHC_GT");
        _NMHC_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NMHC_GT");
        OnNMHC_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NMHC_GT;
partial void OnNMHC_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNMHC_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> C6H6_GT
{
    get
    {
        return _C6H6_GT;
    }
    set
    {
        OnC6H6_GTChanging(value);
        ReportPropertyChanging("C6H6_GT");
        _C6H6_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("C6H6_GT");
        OnC6H6_GTChanged();
    }
}
private Nullable<global::System.Decimal> _C6H6_GT;
partial void OnC6H6_GTChanging(Nullable<global::System.Decimal> value);
partial void OnC6H6_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NOX_GT
{
    get
    {
        return _NOX_GT;
    }
    set
    {
        OnNOX_GTChanging(value);
        ReportPropertyChanging("NOX_GT");
        _NOX_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NOX_GT");
        OnNOX_GTChanged();
    }
}
}

```

```

private Nullable<global::System.Decimal> _NOX_GT;
partial void OnNOX_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNOX_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NO2GT
{
    get
    {
        return _NO2GT;
    }
    set
    {
        OnNO2GTChanging(value);
        ReportPropertyChanging("NO2GT");
        _NO2GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanging("NO2GT");
        OnNO2GTChanged();
    }
}
private Nullable<global::System.Decimal> _NO2GT;
partial void OnNO2GTChanging(Nullable<global::System.Decimal> value);
partial void OnNO2GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=true, IsNullable=false)]
[DataMemberAttribute()]
public global::System.Int32 ID
{
    get
    {
        return _ID;
    }
    set
    {
        if (_ID != value)
        {
            OnIDChanging(value);
            ReportPropertyChanging("ID");
            _ID = StructuralObject.SetValidValue(value);
            ReportPropertyChanging("ID");
            OnIDChanged();
        }
    }
}
private global::System.Int32 _ID;
partial void OnIDChanging(global::System.Int32 value);
partial void OnIDChanged();

#endregion

}

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmEntityTypeAttribute(NamespaceName="Model1", Name="AIR_3")]
[Serializable()]
[DataContractAttribute(IsReference=true)]
public partial class AIR_3 : EntityObject

```

```

{
    #region Factory Method

    /// <summary>
    /// Create a new AIR_3 object.
    /// </summary>
    /// <param name="id">Initial value of the ID property.</param>
    public static AIR_3 CreateAIR_3(global::System.Int32 id)
    {
        AIR_3 aIR_3 = new AIR_3();
        aIR_3.ID = id;
        return aIR_3;
    }

    #endregion

    #region Primitive Properties

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
    [DataMemberAttribute()]
    public Nullable<global::System.Decimal> CO_GT
    {
        get
        {
            return _CO_GT;
        }
        set
        {
            OnCO_GTChanging(value);
            ReportPropertyChanging("CO_GT");
            _CO_GT = StructuralObject.SetValidValue(value);
            ReportPropertyChanged("CO_GT");
            OnCO_GTChanged();
        }
    }
    private Nullable<global::System.Decimal> _CO_GT;
    partial void OnCO_GTChanging(Nullable<global::System.Decimal> value);
    partial void OnCO_GTChanged();

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
    [DataMemberAttribute()]
    public Nullable<global::System.Decimal> NMHC_GT
    {
        get
        {
            return _NMHC_GT;
        }
        set
        {
            OnNMHC_GTChanging(value);
            ReportPropertyChanging("NMHC_GT");
            _NMHC_GT = StructuralObject.SetValidValue(value);
            ReportPropertyChanged("NMHC_GT");
            OnNMHC_GTChanged();
        }
    }
    private Nullable<global::System.Decimal> _NMHC_GT;
    partial void OnNMHC_GTChanging(Nullable<global::System.Decimal> value);
    partial void OnNMHC_GTChanged();

    /// <summary>

```

```

/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> C6H6_GT
{
    get
    {
        return _C6H6_GT;
    }
    set
    {
        OnC6H6_GTChanging(value);
        ReportPropertyChanging("C6H6_GT");
        _C6H6_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("C6H6_GT");
        OnC6H6_GTChanged();
    }
}
private Nullable<global::System.Decimal> _C6H6_GT;
partial void OnC6H6_GTChanging(Nullable<global::System.Decimal> value);
partial void OnC6H6_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NOX_GT
{
    get
    {
        return _NOX_GT;
    }
    set
    {
        OnNOX_GTChanging(value);
        ReportPropertyChanging("NOX_GT");
        _NOX_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NOX_GT");
        OnNOX_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NOX_GT;
partial void OnNOX_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNOX_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NO2GT
{
    get
    {
        return _NO2GT;
    }
    set
    {
        OnNO2GTChanging(value);
        ReportPropertyChanging("NO2GT");
        _NO2GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NO2GT");
        OnNO2GTChanged();
    }
}
}

```

```

private Nullable<global::System.Decimal> _NO2GT;
partial void OnNO2GTChanging(Nullable<global::System.Decimal> value);
partial void OnNO2GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=true, IsNullable=false)]
[DataMemberAttribute()]
public global::System.Int32 ID
{
    get
    {
        return _ID;
    }
    set
    {
        if (_ID != value)
        {
            OnIDChanging(value);
            ReportPropertyChanging("ID");
            _ID = StructuralObject.SetValidValue(value);
            ReportPropertyChanged("ID");
            OnIDChanged();
        }
    }
}
private global::System.Int32 _ID;
partial void OnIDChanging(global::System.Int32 value);
partial void OnIDChanged();

#endregion

}

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmEntityTypeAttribute(NamespaceName="Model", Name="AIR_4")]
[Serializable()]
[DataContractAttribute(IsReference=true)]
public partial class AIR_4 : EntityObject
{
    #region Factory Method

    /// <summary>
    /// Create a new AIR_4 object.
    /// </summary>
    /// <param name="id">Initial value of the ID property.</param>
    public static AIR_4 CreateAIR_4(global::System.Int32 id)
    {
        AIR_4 aIR_4 = new AIR_4();
        aIR_4.ID = id;
        return aIR_4;
    }

    #endregion

    #region Primitive Properties

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
    [DataMemberAttribute()]
    public Nullable<global::System.Decimal> CO_GT

```

```

{
    get
    {
        return _CO_GT;
    }
    set
    {
        OnCO_GTChanging(value);
        ReportPropertyChanging("CO_GT");
        _CO_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("CO_GT");
        OnCO_GTChanged();
    }
}
private Nullable<global::System.Decimal> _CO_GT;
partial void OnCO_GTChanging(Nullable<global::System.Decimal> value);
partial void OnCO_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NMHC_GT
{
    get
    {
        return _NMHC_GT;
    }
    set
    {
        OnNMHC_GTChanging(value);
        ReportPropertyChanging("NMHC_GT");
        _NMHC_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NMHC_GT");
        OnNMHC_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NMHC_GT;
partial void OnNMHC_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNMHC_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> C6H6_GT
{
    get
    {
        return _C6H6_GT;
    }
    set
    {
        OnC6H6_GTChanging(value);
        ReportPropertyChanging("C6H6_GT");
        _C6H6_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("C6H6_GT");
        OnC6H6_GTChanged();
    }
}
private Nullable<global::System.Decimal> _C6H6_GT;
partial void OnC6H6_GTChanging(Nullable<global::System.Decimal> value);
partial void OnC6H6_GTChanged();

/// <summary>

```

```

/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NOX_GT
{
    get
    {
        return _NOX_GT;
    }
    set
    {
        OnNOX_GTChanging(value);
        ReportPropertyChanging("NOX_GT");
        _NOX_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NOX_GT");
        OnNOX_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NOX_GT;
partial void OnNOX_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNOX_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NO2GT
{
    get
    {
        return _NO2GT;
    }
    set
    {
        OnNO2GTChanging(value);
        ReportPropertyChanging("NO2GT");
        _NO2GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NO2GT");
        OnNO2GTChanged();
    }
}
private Nullable<global::System.Decimal> _NO2GT;
partial void OnNO2GTChanging(Nullable<global::System.Decimal> value);
partial void OnNO2GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=true, IsNullable=false)]
[DataMemberAttribute()]
public global::System.Int32 ID
{
    get
    {
        return _ID;
    }
    set
    {
        if (_ID != value)
        {
            OnIDChanging(value);
            ReportPropertyChanging("ID");
            _ID = StructuralObject.SetValidValue(value);
            ReportPropertyChanged("ID");
            OnIDChanged();
        }
    }
}

```

```

    }
}

private global::System.Int32 _ID;
partial void OnIDChanging(global::System.Int32 value);
partial void OnIDChanged();

#endregion

}

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmEntityTypeAttribute(NamespaceName="Model", Name="AIR_5")]
[Serializable()]
[DataContractAttribute(IsReference=true)]
public partial class AIR_5 : EntityObject
{
    #region Factory Method

    /// <summary>
    /// Create a new AIR_5 object.
    /// </summary>
    /// <param name="id">Initial value of the ID property.</param>
    public static AIR_5 CreateAIR_5(global::System.Int32 id)
    {
        AIR_5 aIR_5 = new AIR_5();
        aIR_5.ID = id;
        return aIR_5;
    }

    #endregion

    #region Primitive Properties

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
    [DataMemberAttribute()]
    public Nullable<global::System.Decimal> CO_GT
    {
        get
        {
            return _CO_GT;
        }
        set
        {
            OnCO_GTChanging(value);
            ReportPropertyChanging("CO_GT");
            _CO_GT = StructuralObject.SetValidValue(value);
            ReportPropertyChanged("CO_GT");
            OnCO_GTChanged();
        }
    }

    private Nullable<global::System.Decimal> _CO_GT;
    partial void OnCO_GTChanging(Nullable<global::System.Decimal> value);
    partial void OnCO_GTChanged();

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
    [DataMemberAttribute()]
    public Nullable<global::System.Decimal> NMHC_GT

```

```

{
    get
    {
        return _NMHC_GT;
    }
    set
    {
        OnNMHC_GTChanging(value);
        ReportPropertyChanging("NMHC_GT");
        _NMHC_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NMHC_GT");
        OnNMHC_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NMHC_GT;
partial void OnNMHC_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNMHC_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> C6H6_GT
{
    get
    {
        return _C6H6_GT;
    }
    set
    {
        OnC6H6_GTChanging(value);
        ReportPropertyChanging("C6H6_GT");
        _C6H6_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("C6H6_GT");
        OnC6H6_GTChanged();
    }
}
private Nullable<global::System.Decimal> _C6H6_GT;
partial void OnC6H6_GTChanging(Nullable<global::System.Decimal> value);
partial void OnC6H6_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NOX_GT
{
    get
    {
        return _NOX_GT;
    }
    set
    {
        OnNOX_GTChanging(value);
        ReportPropertyChanging("NOX_GT");
        _NOX_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NOX_GT");
        OnNOX_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NOX_GT;
partial void OnNOX_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNOX_GTChanged();

/// <summary>

```

```

    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
    [DataMemberAttribute()]
    public Nullable<global::System.Decimal> NO2GT
    {
        get
        {
            return _NO2GT;
        }
        set
        {
            OnNO2GTChanging(value);
            ReportPropertyChanging("NO2GT");
            _NO2GT = StructuralObject.SetValidValue(value);
            ReportPropertyChanged("NO2GT");
            OnNO2GTChanged();
        }
    }
    private Nullable<global::System.Decimal> _NO2GT;
    partial void OnNO2GTChanging(Nullable<global::System.Decimal> value);
    partial void OnNO2GTChanged();

    /// <summary>
    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=true, IsNullable=false)]
    [DataMemberAttribute()]
    public global::System.Int32 ID
    {
        get
        {
            return _ID;
        }
        set
        {
            if (_ID != value)
            {
                OnIDChanging(value);
                ReportPropertyChanging("ID");
                _ID = StructuralObject.SetValidValue(value);
                ReportPropertyChanged("ID");
                OnIDChanged();
            }
        }
    }
    private global::System.Int32 _ID;
    partial void OnIDChanging(global::System.Int32 value);
    partial void OnIDChanged();

    #endregion
}

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmEntityTypeAttribute(NamespaceName="Model", Name="AIR_COPY")]
[Serializable()]
[DataContractAttribute(IsReference=true)]
public partial class AIR_COPY : EntityObject
{
    #region Factory Method

    /// <summary>
    /// Create a new AIR_COPY object.

```

```

/// </summary>
/// <param name="id">Initial value of the ID property.</param>
public static AIR_COPY CreateAIR_COPY(global::System.Decimal id)
{
    AIR_COPY aIR_COPY = new AIR_COPY();
    aIR_COPY.ID = id;
    return aIR_COPY;
}

#endregion

#region Primitive Properties

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> CO_GT
{
    get
    {
        return _CO_GT;
    }
    set
    {
        OnCO_GTChanging(value);
        ReportPropertyChanging("CO_GT");
        _CO_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("CO_GT");
        OnCO_GTChanged();
    }
}
private Nullable<global::System.Decimal> _CO_GT;
partial void OnCO_GTChanging(Nullable<global::System.Decimal> value);
partial void OnCO_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NMHC_GT
{
    get
    {
        return _NMHC_GT;
    }
    set
    {
        OnNMHC_GTChanging(value);
        ReportPropertyChanging("NMHC_GT");
        _NMHC_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NMHC_GT");
        OnNMHC_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NMHC_GT;
partial void OnNMHC_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNMHC_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> C6H6_GT

```

```

{
    get
    {
        return _C6H6_GT;
    }
    set
    {
        OnC6H6_GTChanging(value);
        ReportPropertyChanging("C6H6_GT");
        _C6H6_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("C6H6_GT");
        OnC6H6_GTChanged();
    }
}
private Nullable<global::System.Decimal> _C6H6_GT;
partial void OnC6H6_GTChanging(Nullable<global::System.Decimal> value);
partial void OnC6H6_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NOX_GT
{
    get
    {
        return _NOX_GT;
    }
    set
    {
        OnNOX_GTChanging(value);
        ReportPropertyChanging("NOX_GT");
        _NOX_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NOX_GT");
        OnNOX_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NOX_GT;
partial void OnNOX_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNOX_GTChanged();

/// <summary>
/// No Metadata Documentation available.
/// </summary>
[EdmScalarPropertyAttribute(EntityKeyProperty=false, IsNullable=true)]
[DataMemberAttribute()]
public Nullable<global::System.Decimal> NO2_GT
{
    get
    {
        return _NO2_GT;
    }
    set
    {
        OnNO2_GTChanging(value);
        ReportPropertyChanging("NO2_GT");
        _NO2_GT = StructuralObject.SetValidValue(value);
        ReportPropertyChanged("NO2_GT");
        OnNO2_GTChanged();
    }
}
private Nullable<global::System.Decimal> _NO2_GT;
partial void OnNO2_GTChanging(Nullable<global::System.Decimal> value);
partial void OnNO2_GTChanged();

/// <summary>

```

```

    /// No Metadata Documentation available.
    /// </summary>
    [EdmScalarPropertyAttribute(EntityKeyProperty=true, IsNullable=false)]
    [DataMemberAttribute()]
    public global::System.Decimal ID
    {
        get
        {
            return _ID;
        }
        set
        {
            if (_ID != value)
            {
                OnIDChanging(value);
                ReportPropertyChanging("ID");
                _ID = StructuralObject.SetValidValue(value);
                ReportPropertyChanged("ID");
                OnIDChanged();
            }
        }
    }
    private global::System.Decimal _ID;
    partial void OnIDChanging(global::System.Decimal value);
    partial void OnIDChanged();

    #endregion

}

#endregion

}

```

Βιβλιογραφία - Αναφορές

- (1) Παναγιωτάκος Θεόδωρος, Διπλωματική εργασία με θέμα «Τεχνικές Εξόρυξης Δεδομένων» Αθήνα 2012
- (2) Χαραλάμπους Κωνσταντίνος, Διπλωματική εργασία με θέμα «Οικονομικές Επιπτώσεις Ολυμπιακών Αγώνων και Τεχνικές Εξόρυξης Πληροφορίας: Απόπειρες Εξαγωγής Αποτελεσμάτων και Συμπεράσματα» Αθήνα Ιούλιος 2009
- (3) Χαλκίδη Μ, Βαρζιγιάννης Μ, Εξόρυξη γνώσης από βάσεις δεδομένων και τον παγκόσμιο ιστό, εκδόσεις Gutenberg, 2009
- (4) Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, Ramasary Uthurusamy, Advances in Knowledge Discovery and Data Mining <https://www.aaai.org/Papers/KDD/1996/KDD96-014.pdf>
- (5) Παγουρόπουλος Απόστολος, Διπλωματική εργασία με θέμα «Data Mining στη Χρηματοοικονομική ανάλυση» Πάτρα 17/10/2006
- (6) Τζιραλής Γεώργιος, «Αλγόριθμοι εξόρυξης πληροφορίας, Διάλεξη 4, απεικόνιση γνώσης, αξιοπιστία και αποτίμηση» http://sites.google.com/site/gtziralis/Lecture04_KnowledgeRepresentationCredibilityEvaluation.pdf
- (7) Richter M Michael, Advances in Data Mining: Theoretical Aspects and Applications, εκδ. Springer, 2007
- (8) Ελληνικά Ακαδημαϊκά και Ηλεκτρονικά Συγγράμματα και Βοηθήματα https://repository.kallipos.gr/bitstream/11419/2130/1/06_chapter05.pdf, https://repository.kallipos.gr/bitstream/11419/2972/1/02_chapter_06.pdf και https://repository.kallipos.gr/bitstream/11419/1236/2/Kef._9.pdf
- (9) Βρυώνης Ευάγγελος, Τσούτσας Δημήτριος, Πτυχιακή εργασία με θέμα «Γραφο-θεωρητικές μέθοδοι συσταδοποίησης και ο αλγόριθμος JARVIS-Patrick σε βάσεις δεδομένων Oracle» Σέρρες Ιούνιος 2011
- (10) Varun Chandola, Vipon Kumar Summarization – Compressing Data Into an Informative Representation, University of Minnesota
- (11) Jiawei Han, Micheline Kamber, Jian Pei Data Mining Concepts and Techniques
- (12) Θεοδωρίδης Γιάννης, Πελέκης Νίκος, «Αποθήκες Δεδομένων και εξόρυξη γνώσης»
- (13) Αναστάσιος Γούναρης, ΑΠΘ, Ανοικτά Μαθήματα «Αποθήκες Δεδομένων και εξόρυξη δεδομένων» Ενότητα 01-Ενότητα 12.
- (14) Charu C. Aggarwal , Philip S. Yu . , A Survey of Synopsis Construction in Data Streams.

- (15) Aggarwal C., Han J., Wang J., Yu P. (2003) A Framework for Clustering Evolving Data Streams. *VLDB Conference*.
- (16) Aggarwal C, Han J., Wang J., Yu P. (2004). On-Demand Classification of Data Streams. *ACM KDD Conference*.
- (17) BIRCH, <https://en.wikipedia.org/wiki/BIRCH>
- (18) McCune, Grace Distance Measures, chapter 6 2002.
- (19) "The Big Data Market: 2014 - 2020 - Opportunities, Challenges, Strategies, Industry Verticals and Forecasts." The Big Data Market: 2014 - 2020 - Opportunities, Challenges, Strategies, Industry Verticals and Forecasts. N.p., n.d. Web. 19 Aug. 2014.
- (20) "Big data in the banking sector -." Big data in the banking sector. N.p., n.d. Web. 19 Aug. 2014.
- (21) Six Provocations for Big Data
- (22) "OF DATA AND MEN" FUNDAMENTAL RIGHTS AND FREEDOMS IN A WORLD OF BIG DATA
 Gigabytes= 10^6 byte, Terabytes= 10^9 byte, Exabytes= 10^{12} byte, Zettabytes= 10^{15} byte
- (23). We shall not discuss here the indirect costs for the environment and health of the development of digital technologies
- (24) "Big Data" : Technologies and players. N.p., n.d. Web. 19 Aug. 2014. <<http://bigdatatutorial.blogspot.co.uk/2012/01/big-data-technologies-and-players.html>>.
- (25) Big Data How it became a Differentiator (Deutsche Bank)
- (26) "Big Data Vendor Revenue And Market Forecast 2012-2017 - Wikibon." Big Data Vendor Revenue And Market Forecast 2012-2017 - Wikibon. N.p., n.d. Web. 19 Aug. 2014. http://wikibon.org/wiki/v/Big_Data_Vendor_Revenue_and_Market_Forecast_2012-2017
- (27) Νίκος Κυραμάς ΕΛΕΓΧΟΣ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΣΕ ΠΑΡΟΧΟΥΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΗΡΕΣΙΩΝ Πανεπιστήμιο Πειραιώς Τμήμα Ψηφιακών Συστημάτων Π.Μ.Σ. 2015-16.
- (28) Customer Successes in your industry from http://www.spss.com/success/?source=homepage&hpzone=nav_bar
- (29) McKinsey Global Institute, Big data: The next frontier for innovation, competition, and productivity, June 2011 ("MGI Report"), http://www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation; Thomas Edsall, Let the Nanotargeting Begin, NY Times Campaignstops Blog, April 15, 2012, <http://campaignstops.blogs.nytimes.com/2012/04/15/let-the-nanotargeting-begin>.

- (30) Adverse Event Reporting System (AERS), <http://www.fda.gov/drugs/guidancecomplianceregulatoryinformation/surveillance/adversedrugeffects/default.htm>.
- (31) Rita Rubin, How did Vioxx debacle happen?, USA Today, October 12, 2004, http://www.usatoday.com/news/health/2004-10-12-vioxx-cover_x.htm.
- (32) Jeremy Ginsberg, Matthew Mohebbi, Rajan Patel, Lynnette Brammer, Mark Smolinski & Larry Brilliant, Detecting Influenza Epidemics Using Search Engine Query Data, 457 Nature 1012 (2009).
- (33) Big Data for Social Good Initiative, Engineering Social Systems, <http://www.hsph.harvard.edu/ess/bigdata.html>; see, e.g., Amy Wesolowski & Nathan Eagle, Parameterizing the Dynamics of Slums, AAAI Spring Symposium 2010 on Artificial Intelligence for Development (AI-D), <http://ai-d.org/pdfs/Wesolowski.pdf>.
- (34) Washington Okori & Joseph Obua, Machine Learning Classification Technique for Famine Prediction, Proceedings of the World Congress on Engineering 2011, <http://www.iaeng.org/publication/WCE2011/WCE2011pp991-996.pdf>
- (35) Jmeson Toole, Nathan Eagle and Joshua Plotkin, Quantifying Crime Waves, AAAI Spring Symposium 2010 on Artificial Intelligence for Development (AI-D), <http://ai-d.org/pdfs/Toole.pdf>.
- (36) Massoud Moussavi and Noel McGinn, A Model for Quality of Schooling, American Associations for the Advancement of Artificial Intelligence, 2010 AAAI Spring Symposium Series.
- (37) Information and Privacy Commissioner of Ontario & Future of Privacy Forum, Smart Privacy for the Smart Grid: Embedding Privacy into the Design of Electricity Conservation, November 2009, <http://www.ipc.on.ca/images/Resources/pbd-smartpriv-smartgrid.pdf>.
- (38) Katie Fehrenbacher, Introducing the Facebook Social Energy App, October 17, 2011, GigaOm <http://gigaom.com/cleantech/introducing-the-facebook-social-energy-app>.
- (39) In Europe: Directive 2004/52/EC of the European Parliament and of the Council of 29 April 2004 on the interoperability of electronic road toll systems in the Community; Commission Decision 2009/750/EC of 6 October 2009 on the definition of the European Electronic Toll Service and its technical element.
- (40) See, e.g., Carlo Ratti, Riccardo Maria Pulselli, Sarah Williams & Dennis Frenchman, Mobile Landscapes: Using Location Data from Cell-Phones for Urban Analysis, 33 Environment and Planning B: Planning and Design 727 (2006).
- (41) MGI Report, supra note 21.

- (42) Sastry Duri, Jeffrey Elliot, Marco Gruteser, Xuan Liu, Paul Moskowitz, Ronald Perez, Moninder Singh & Jung-Mu Tang, Data Protection and Data Sharing in Telematics, 9(6) Mobile Networks & Applications, 693 (2004).
- (43) See Article 29 Working Party, Opinion 2/2010 on online behavioral advertising (WP 171), 22 June 2010, http://ec.europa.eu/justice/home/fsi/privacy/docs/wpdocs/2010/wp171_en.pdf; FTC Staff Report, SelfRegulatory Principles for Online Behavioral Advertising, February 2009, <http://www.ftc.gov/os/2009/02/P085400behavadreport.pdf>.
- (44) Omer Tene & Jules Polonetsky, To Track or 'Do Not Track': Advancing Transparency and Individual Control in Online Behavioral Advertising, 13 Minn. J. L. Sci. & Tech. 282 (2012).
- (45) MGI Report, supra note 21, at p. 68.
- (46) Duncan Douglass, An Examination of the Fraud Liability Shift in Consumer Card-Based Payment Systems, Federal Reserve Bank of Chicago (2009), www.chicagofed.org/digital.../ep_1qtr2009_part7_douglass.pdf.
- (47) Melissa Fach, Stats on Facebook 2012, Search Engine Journal, February 17, 2012, <http://www.searchenginejournal.com/stats-on-facebook-2012-infographic/40301>; also see 10 Key Statistics About Facebook, Experian Hitwise Blog, February 2, 2012, <http://www.experian.com/blogs/hitwise/2012/2/2/10-key-statistics-about-facebook>.
- (48) <http://www.google.com/intl/en/about/products/index.html>.
- (49) See Statement of the Federal Trade Commission Concerning Google/DoubleClick, FTC File No. 071-0170, December 20, 2007, <http://www.ftc.gov/os/caselist/0710170/071220statement.pdf>; and see Dissenting Statement of Commissioner Pamela Jones Harbour, <http://www.ftc.gov/os/caselist/0710170/071220harbour.pdf>.
- (50) Balachander Krishnamurthy & Craig Wills, Privacy Diffusion on the Web: A Longitudinal Perspective, October 2009, submitted as public comment to Federal Trade Commission Exploring Privacy Roundtable Series, <http://www.ftc.gov/os/comments/privacvroundtable/544506-00009.pdf>.
- (51) Nicole Perlroth, Revamping at Yahoo to Focus on Its Media Properties and Customer Data, NY Times, April 11, 2012, <http://bit.ly/HUneMM>.
- (52) Chakrabarti K., Garofalakis M., Rastogi R., Shim K. (2001) Approximate Query Processing with Wavelets. VLDB Journal, 10(2-3), pp. 199–223.
- (53) Gehrke J., Korn F., Srivastava D. (2001) On Computing Correlated Aggregates Over Continual Data Streams. SIGMOD Conference.
- (54) Garofalakis M., Gehrke J., Rastogi R. (2002) Querying and mining data streams: you only get one look (a tutorial). SIGMOD Conference.

(55) Database Systems for Advanced Applications, 17th International Conference, DASFAA 2012, Busan, South Korea, April 15-19, 2012, Proceedings, Part II, https://link.springer.com/chapter/10.1007/978-3-642-29035-0_33

(56) Μέθοδοι εξόρυξης και επεξεργασίας ερωτημάτων σε ροές δεδομένων, Μαρία Κοντάκη, Διδακτορική Διατριβή, ΑΠΘ / Τμήμα Πληροφορικής, 2009.