ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΑ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Καθαρισμός δεδομένων: Επισκόπηση και Εφαρμογή σε Πραγματικά Δεδομένα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

Τσικάκη Δημητρίου
Μπαλαμώτη Αλεξάνδρας

Επιβλέπουσα : Βλάχου Ακριβή

Μέλη εξεταστικής επιτροπής: Κωστούλας Θεόδωρος, Μαλιάτσος Κωνσταντίνος

Σάμος, Φεβρουάριος 2023

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Θα θέλαμε να εκφράσουμε τις θερμές μας ευχαριστίες σε όλους όσους συνέβαλαν στην εκπόνησή της. Συγκεκριμένα, θα θέλαμε να ευχαριστήσουμε την επιβλέπουσα καθηγήτρια του τμήματος μηχανικών και πληροφοριακών συστημάτων κα. Ακριβή Βλάχου καθώς δίχως την πολύτιμη βοήθειά της θα ήταν αδύνατη η ολοκλήρωση της διπλωματικής μας εργασίας όπως επίσης και όλο το διδακτικό προσωπικό του Πανεπιστημίου Αιγαίου για την συνεχή υποστήριξη και βοήθειά τους. Τέλος θα θέλαμε να εκφράσουμε την ευγνωμοσύνη στους οικείους μας για όλη την στήριξη, την συμπαράσταση και την κατανόησή τους καθ' όλη τη διάρκεια των σπουδών μας.

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

Λίστα Σχημάτων

Λίστα Τύπων

Λίστα Πινάκων

Ακρωνύμια

ΒΔ	Βάση Δεδομένων
IC	Integrity constraint
ETL	Extract, transform, load
PDF	Probability density function
DC	Denial constraints
FD	Functional dependencies
CFD	Conditional functional dependencies)
MD	Matching dependencies
ML	Machine learning
csv	Comma Separated Values- Τύπος αρχείων κειμένου με τον χαρακτήρα «,» ως διαχωριστικό τιμών
lat	latitude - γεωγραφικό πλάτος
long	longitude - γεωγραφικό μήκος
vin	Vehicle Identification Number - Αναγνωριστικός αριθμός οχήματος
Avg	Average – Μέση τιμή
Query	Επερώτημα
Null	Τιμή που δηλώνει τη μη ύπαρξη αντικειμένου ή τιμής
String	Τύπος τιμών κειμένου
Int	Τύπος αριθμητικών ακέραιων τιμών
Float	Τύπος αριθμητικών δεκαδικών τιμών

Περίληψη

Η διαδικασία της εύρεσης και της διόρθωσης προβλημάτων στην ανάλυση δεδομένων είναι γνωστή ως καθαρισμός δεδομένων. Μια ουσιαστική, συχνά δαπανηρή και πάντα δύσκολη ευθύνη προκύπτει κατά τη συλλογή και συγχώνευση δεδομένων από διαφορετικές πηγές σε μια αποθήκη δεδομένων: η διασφάλιση καλής ποιότητας και συνέπειας των δεδομένων. Στην πραγματικότητα, μία από τις μόνιμες δυσκολίες στην ανάλυση δεδομένων είναι η εύρεση και ο καθορισμός των βρώμικων δεδομένων- η αποτυχία αυτή μπορεί να οδηγήσει σε λανθασμένες αναλύσεις και αναξιόπιστες κρίσεις. Το ενδιαφέρον για θέματα καθαρισμού δεδομένων έχει αυξηθεί πρόσφατα τόσο στη βιομηχανία όσο και στην πανεπιστημιακή κοινότητα. Για να κατανοήσουμε καλύτερα τις εξελίξεις σε αυτό το κομμάτι, στο πρώτο μέρος της διπλωματικής παρουσιάζονται διάφορες τεχνικές, μέθοδοι και προσεγγίσεις καθαρισμού δεδομένων. Το δεύτερο μέρος της μελέτης εφαρμόζει τεχνικές καθαρισμού δεδομένων σε τρεις συλλογές δεδομένων που σχετίζονται με τον κλάδο της εστίασης, τις τιμές της εταιρίας Uber και τα μεταχειρισμένα αυτοκίνητα. Αναλύονται διαφορετικά σενάρια που μπορούν να προκύψουν από την ανάλυση συλλογών δεδομένων τα οποία είναι διαθέσιμα δημόσια, αφού πρώτα γίνει καθαρισμός των δεδομένων με Excel και Python. Παρουσιάζονται τα αποτελέσματα της διαδικασίας καθαρισμού των δεδομένων, τα οποία δείχνουν ότι οι τεχνικές καθαρισμού των δεδομένων βελτιώνουν σημαντικά την ποιότητα και την ακρίβεια των συνόλων δεδομένων.

Λέξεις Κλειδιά: *καθαρισμός δεδομένων, ανάλυση δεδομένων, επιστήμη των δεδομένων, συλλογή δεδομένων*

Abstract

The process of finding and correcting problems in data analysis is known as data cleaning. A substantial, often costly and always difficult responsibility arises when collecting and merging data from different sources into a data warehouse: ensuring good quality and consistency of data. In fact, one of the enduring difficulties in data analysis is finding and fixing dirty data; this failure can lead to incorrect analyses and unreliable judgments. Interest in data cleaning issues has recently increased in both industry and academia. To better understand the developments in this area, the first part of the thesis presents different techniques, methods and approaches to data cleaning. The second part of the study applies data cleaning techniques to three datasets related to the restaurant industry, Uber fares, and used cars. Different scenarios that can result from the analysis of data collections that are publicly available are analyzed, after they first have been cleaned using Excel and Python. The results of the data cleaning process are presented, which show that the data cleaning techniques significantly improve the quality and accuracy of the datasets. The study also evaluates the impact of data cleaning on subsequent data analysis, demonstrating that the cleaned datasets provide more reliable and meaningful insights.

Keywords: *data cleaning, data analytics, data science, dataset*

1

Εισαγωγή

1.1 Τα δεδομένα στη σημερινή εποχή και ο καθαρισμός τους

Στην εποχή της οικονομίας των (μεγάλων) δεδομένων, τα υψηλής ποιότητας δεδομένα αποτελούν μακράν το σημαντικότερο στοιχείο κάθε επιχείρησης ή οργανισμού. Οι μεγάλες επιχειρήσεις συλλέγουν πλέον συστηματικά δεδομένα, όχι μόνο για την τήρηση αρχείων, αλλά και για την υποστήριξη μιας σειράς εργασιών ανάλυσης δεδομένων, που είναι απαραίτητες για την επίτευξη των εταιρικών στόχων. Η ανάλυση δεδομένων συχνά βοηθάει σε διαδικασίες λήψης αποφάσεων και βελτίωσης της αποδοτικότητας και χρησιμεύει ως θεμέλιο σε ολόκληρους οργανισμούς ή επιχειρήσεις. Τα ανακριβή ή ασυνεπή δεδομένα μπορούν να αλλοιώσουν δραστικά τα συμπεράσματα των ερευνών, υπονομεύοντας συχνά τα δυνητικά πλεονεκτήματα των προσεγγίσεων που βασίζονται στις πληροφορίες. Εξαιτίας αυτού, ο καθαρισμός δεδομένων είναι μια τεχνική, που χρησιμοποιείται για τον εντοπισμό λανθασμένων, ελλιπών ή ακατάλληλων δεδομένων και προσφέρει μια λύση σε αυτά τα προβλήματα και τα τυχόν εντοπισμένα σφάλματα και οι παραλείψεις διορθώνονται, για τη βελτίωση της ποιότητας. Σε ένα τυπικό σενάριο χρήσης, μια εταιρεία διαχειρίζεται και αναλύει πολλά δεδομένα προκειμένου να εξάγει σχετικές πληροφορίες, οι οποίες είναι απαραίτητες για την προσφορά υπηρεσιών ή τη λήψη αποφάσεων. Λόγω της αυξανόμενης ποσότητας, ταχύτητας και ποικιλίας δεδομένων σε πολλές εφαρμογές, ο καθαρισμός δεδομένων θεωρείται, μια σημαντική πρόκληση στην εποχή που ζούμε. Παρόλο που μπορεί να είναι χρονοβόρα και επίπονη διαδικασία, η διόρθωση των λαθών στα δεδομένα και η αφαίρεση εσφαλμένων πληροφοριών, θα πρέπει να γίνεται σε κάθε περίπτωση. Μια πρόσφατη έρευνα έδειξε ότι τα "βρώμικα" και διπλά δεδομένα κοστίζουν στην οικονομία των ΗΠΑ περισσότερα από 3 τρισεκατομμύρια δολάρια ετησίως (Tibbets, 2011), ενώ το Data Warehousing Institute υπολόγισε

ότι τα προβλήματα ποιότητας δεδομένων κοστίζουν στις αμερικανικές επιχειρήσεις περισσότερα από 600 δισεκατομμύρια δολάρια ετησίως (Eckerson, 2002).

Σε πολλές περιπτώσεις ακόμη και αν τα δεδομένα είναι αξιόπιστα, ακριβή και πλήρη, μπορεί να μην αντικατοπτρίζουν τις πραγματικές πρόσφατες τιμές. Σε μια βάση δεδομένων με 500 χιλιάδες εγγραφές πελατών, 10 χιλιάδες εγγραφές μπορεί να αλλοιώνονται κάθε μήνα, 120 χιλιάδες εγγραφές ετησίως, και μέσα σε δύο χρόνια, περίπου το 50% όλων των εγγραφών μπορεί να είναι αλλοιωμένο, σύμφωνα με εκτιμήσεις ειδικών που αναφέρουν ότι το 2% των εγγραφών σε ένα αρχείο πελατών παλιώνουν (Fan κ.ά., 2014)

Υπάρχουν πολλοί παράγοντες που συμβάλλουν σε αυτά τα ζητήματα. Στην κορυφή του καταλόγου βρίσκονται τα ανθρώπινα λάθη, συμπεριλαμβανομένης της εσφαλμένης καταχώρησης δεδομένων και των ελλιπών τιμών. Στη συνέχεια, υπάρχουν τεχνολογικοί παράγοντες, όπως ανακριβείς ενδείξεις αισθητήρων ή συσκευών, ανακριβείς εφαρμογές που συμπληρώνουν βάσεις δεδομένων, εσφαλμένος σχεδιασμός βάσεων δεδομένων που δεν επιβάλλουν τις κατάλληλες απαιτήσεις ποιότητας ή συνδυασμός δεδομένων από διάφορες ετερογενείς πηγές. Επιπλέον, επειδή πληροφορίες όπως διευθύνσεις, αριθμοί τηλεφώνου και ώρες ραντεβού αλλάζουν συνεχώς, οι βάσεις δεδομένων μπορεί να θεωρηθούν παρωχημένες δίνοντας αναληθή αναπαράσταση του πραγματικού κόσμου.

1.2 Αντικείμενο διπλωματικής

Στην παρούσα διπλωματική εργασία θα μελετηθούν οι προσεγγίσεις καθαρισμού δεδομένων, καθώς ο αντίκτυπος μιας καθαρής βάσης έχει μεγάλη σημασία εφαρμογής σε διάφορους τομείς όπως στην επιστήμη των δεδομένων.

Ο καθαρισμός δεδομένων γίνεται στην παρούσα εργασία με την βοήθεια μικρών script σε Python και με τη βοήθεια του προγράμματος excel σε κάποιες περιπτώσεις. Και στις τρεις συλλογές δεδομένων γίνεται ένας βασικός καθαρισμός ώστε να μπορέσουν τα αρχεία να εισαχθούν και να διαβαστούν από μία βάση δεδομένων.

Μερικές από τις βασικές μεθόδους καθαρισμού που χρησιμοποιήθηκαν και χρησιμοποιούνται ευρέως είναι η απαλοιφή διπλότυπων τιμών, η εκκαθάριση από null τιμές και ο τύπος δεδομένων σε κάθε στήλη να είναι αυτός που απαιτείται σε κάθε περίσταση.

1.3 Στόχος διπλωματικής

Αρχικός στόχος της διπλωματικής εργασίας είναι να μπορέσει ο αναγνώστης να κατανοήσει μερικούς από τους βασικούς τρόπους και μεθόδους καθαρισμού δεδομένων και να αποκτήσει μια γενική εικόνα σχετικά με τις προκλήσεις και τις δυσκολίες στον τομέα αυτόν. Επίσης, στόχος της διπλωματικής είναι ο καθαρισμός τριών συλλογών δεδομένων ώστε να επιτευχθεί πιο εύκολα η ανάγνωση και η ανάλυση αυτών από μια βάση δεδομένων καθώς και η παρουσίαση μερικών σεναρίων χρήσης των πλέον καθαρισμένων δεδομένων και σε συνέχεια ο ρόλος που παίζει ο

καθαρισμός τους προτού αυτά χρησιμοποιηθούν ώστε να πάρουμε τις σωστές απαντήσεις σε ερωτήματα που θέλουμε να θέσουμε.

1.4 Δομή της διπλωματικής

Στο δεύτερο κεφάλαιο γίνεται μια επισκόπηση καθαρισμού δεδομένων και αναλύονται οι πηγές των σφαλμάτων, τα προβλήματα που δημιουργούνται και οι λύσεις που προτείνονται. Επίσης γίνεται μια επισκόπηση προσεγγίσεων στα προβλήματα μίας πηγής και πολλαπλών πηγών καθώς και μερικές μεθοδολογίες για τον καθαρισμό δεδομένων. Στο τρίτο κεφάλαιο γίνεται αναφορά στην ανίχνευση ακραίων τιμών και ταξινομούμε τις μεθόδους ανίχνευσης ανάλογα με τον τύπο τους καθώς και στην απαλοιφή των διπλότυπων δεδομένων. Επίσης, γίνεται μια αναφορά στους συντακτικούς μετασχηματισμούς δεδομένων καθώς αναλύουμε βάσει κανόνων και με την βοήθεια της μηχανικής μάθησης τον καθαρισμό δεδομένων. Τέλος αναφερόμαστε στον καθαρισμό δεδομένων με την βοήθεια της ανθρώπινης συμμετοχής. Στο τέταρτο κεφάλαιο γίνεται εφαρμογή καθαρισμού δεδομένων σε πραγματικά δεδομένα σε τρεις συλλογές δεδομένων που μας δόθηκαν και παρουσιάζονται διαφορετικά σενάρια που έχουν εφαρμογή στην πραγματικότητα, χρησιμοποιώντας και αναλύοντας τις προαναφερθείσες συλλογές δεδομένων.

2

Επισκόπηση καθαρισμού δεδομένων

2.1 Γενικά για τις μεθοδολογίες και λύσεις καθαρισμού δεδομένων

Όπως αναφέρθηκε και προηγουμένως, όταν αναφερόμαστε στον καθαρισμό δεδομένων, εννοούμε τον εντοπισμό και την εξάλειψη των ελαττωμάτων και των ασυνεπειών από τα δεδομένα προκειμένου να αυξηθεί η ποιότητα τους. Ο καθαρισμός των δεδομένων γίνεται όλο και πιο σημαντικός, όταν πρέπει να συνδυαστούν πολλές πηγές δεδομένων, όπως από αποθήκες δεδομένων ή ομοσπονδιακά συστήματα βάσεων δεδομένων ή διεθνή συστήματα πληροφοριών, που βασίζονται στο διαδίκτυο. Αυτό συμβαίνει επειδή στις πηγές αυτές υπάρχουν συχνά περιττά δεδομένα σε διάφορες μορφές. Κατά την ενσωμάτωση πολλών πηγών δεδομένων και κατά τη χρήση μιας μόνο πηγής δεδομένων, θα πρέπει να εντοπίζονται και να διορθώνονται όλα τα σημαντικά λάθη και οι ασυνέπειες. Η ενοποίηση των διαφόρων μορφών δεδομένων και η αφαίρεση των πλεοναζόντων δεδομένων καθίσταται σημαντική, προκειμένου να παρέχεται πρόσβαση σε ορθά και συνεπή δεδομένα.

Πολλά σφάλματα δεδομένων εντοπίζονται ακούσια κατά τη διεξαγωγή ερευνητικών δραστηριοτήτων πέρα από τη διαδικασία του καθαρισμού. Ωστόσο, είναι πιο αποτελεσματική η εύρεση σφαλμάτων με σκόπιμη αναζήτηση. Ορισμένες φορές είναι δύσκολο να διαπιστώσει κανείς αμέσως, αν ένα σημείο δεδομένων είναι ανακριβές. Ένα αμφισβητήσιμο σημείο δεδομένων ή μοτίβο είναι συχνά αυτό που εντοπίζεται και απαιτεί περαιτέρω έρευνα. Ομοίως, οι τιμές που λείπουν απαιτούν πρόσθετη έρευνα. Οι τιμές που λείπουν μπορεί να οφείλονται σε διαταραχές της ροής δεδομένων ή στην απουσία των προβλεπόμενων δεδομένων (van den Broeck κ.ά., 2005).

Ως εκ τούτου, είναι βέλτιστη πρακτική να υπάρχουν προκαθορισμένες κατευθυντήριες γραμμές για το χειρισμό σφαλμάτων, πραγματικών ελλιπών τιμών και ακραίων τιμών.

2.1.1 Λύσεις καθαρισμού δεδομένων

Σύμφωνα με μια μελέτη των Abedjan κ.ά., (2016a) οι υπάρχουσες λύσεις καθαρισμού δεδομένων χωρίζονται σε τέσσερις κατηγορίες:

- **Ανίχνευση ποσοτικών σφαλμάτων** για την εύρεση ανωμαλιών και σφαλμάτων δεδομένων. Αυτό είναι ιδιαίτερα χρήσιμο για τον εντοπισμό τιμών που αποκλίνουν από την κατανομή

των δεδομένων σε μια στήλη ή έναν πίνακα κατά την εργασία με αισθητηριακά δεδομένα ή ροές δεδομένων.

- **Η συσχέτιση εγγραφών** (επίσης γνωστή ως αντιγραφή δεδομένων) χρησιμοποιείται για την εύρεση διαφορετικών πλειάδων δεδομένων, που σχετίζονται με τις ίδιες οντότητες του πραγματικού κόσμου και η συγχώνευση δεδομένων χρησιμοποιείται για τον συνδυασμό αυτών των διπλών εγγραφών σε μια ενιαία εγγραφή.
- **Επιβολή και μετασχηματισμός προτύπων** για την εύρεση συντακτικών ή σημασιολογικών προτύπων στα δεδομένα και τον εντοπισμό καταχωρίσεων που δεν ακολουθούν αυτά τα πρότυπα. Αυτή η συλλογή εργαλείων μπορεί να διορθώσει προβλήματα μορφοποίησης, ορθογραφικά λάθη, εσφαλμένους τύπους δεδομένων κ.λπ.
- **Καθαρισμός δεδομένων βάσει κανόνων** όταν ένα σύνολο κανόνων εφαρμόζεται στα δεδομένα. Προκειμένου να δημιουργηθεί ένα συνεπές παράδειγμα βάσης δεδομένων που συμμορφώνεται με τους κανόνες, τα εργαλεία καθαρισμού δεδομένων προσπαθούν να εντοπίσουν τις παραβιάσεις αυτών των κανόνων και να διορθώσουν τις εγγραφές δεδομένων. Οι απλοί περιορισμοί μη-μηδενικού, οι περιορισμοί ακεραιότητας, οι εξαρτήσεις πολλαπλών χαρακτηριστικών, οι λειτουργίες που ορίζονται από τον χρήστη και οι μοναδικοί επιχειρηματικοί κανόνες είναι μερικά μόνο παραδείγματα των κανόνων που μπορούν να χρησιμοποιηθούν.

2.1.2 Πηγές σφαλμάτων στα δεδομένα

Ένα στοιχείο δεδομένων συχνά περνάει από διάφορες φάσεις, συμπεριλαμβανομένης της ανθρώπινης συμμετοχής και του υπολογισμού, προτού καταλήξει σε μια βάση δεδομένων. Κάθε στάδιο της διαδικασίας, από την αρχική συλλογή δεδομένων μέχρι την αποθήκευση στο αρχείο, είναι επιρρεπές στο να υποστεί λάθη. Η κατανόηση των αιτιών των λαθών στα δεδομένα μπορεί να βοηθήσει στην καθιέρωση μεθόδων συλλογής και επιμέλειας δεδομένων που μειώνουν την εισαγωγή λαθών, καθώς και μεθόδων εκκαθάρισης δεδομένων που μπορούν να εντοπίσουν και να διορθώσουν τα προβλήματα. Οι ακόλουθες κατηγορίες περιλαμβάνουν ένα μεγάλο μέρος των πηγών ανακρίβειας στις βάσεις δεδομένων:

- **Σφάλματα καταχώρησης δεδομένων:** Οι άνθρωποι εξακολουθούν να χρησιμοποιούνται συχνά σε διάφορες καταστάσεις για την εισαγωγή δεδομένων, είτε αντλώντας πληροφορίες από την ομιλία (όπως στα τηλεφωνικά κέντρα) είτε πληκτρολογώντας πληροφορίες από γραπτές ή έντυπες πηγές. Τα τυπογραφικά λάθη ή η έλλειψη κατανόησης της πηγής δεδομένων είναι συνήθεις αιτίες αλλοίωσης των δεδομένων σε αυτά τα πλαίσια κατά τη στιγμή της εισαγωγής. Ένας άλλος πολύ συχνός λόγος για τον οποίο οι άνθρωποι εισάγουν "βρώμικα" δεδομένα σε φόρμες είναι για να παρέχουν αυτό που αναφέρουμε ως ψευδή ακεραιότητα: πολλές φόρμες απαιτούν τη συμπλήρωση συγκεκριμένων πεδίων, και όταν ένας χρήστης που εισάγει δεδομένα δεν έχει πρόσβαση σε τιμές για ένα από αυτά τα πεδία, συχνά επινοεί μια προεπιλεγμένη τιμή που είναι απλό να πληκτρολογηθεί ή που φαίνεται να είναι μια τυπική τιμή.
- **Σφάλματα μέτρησης**: Τα δεδομένα χρησιμοποιούνται συχνά για τη μέτρηση μιας φυσικής διαδικασίας, όπως για παράδειγμα την ταχύτητα ενός αυτοκινήτου, το μέγεθος του

πληθυσμού, την ανάπτυξη μιας οικονομίας κ.λπ. Οι μετρήσεις αυτές πραγματοποιούνται ενίοτε με τη χρήση ανθρώπινων τεχνικών, οι οποίες ενδέχεται να οδηγήσουν σε λάθη, τόσο κατά την εφαρμογή τους (π.χ. χρήση ακατάλληλων μεθόδων δειγματοληψίας ή έρευνας), όσο και κατά το σχεδιασμό τους (π.χ., με τα όργανα). Καθώς όλο και περισσότερες συσκευές χρησιμοποιούνται για τη μέτρηση φυσικών ιδιοτήτων, η τεχνολογία των αισθητήρων έχει παράγει τεράστιες ποσότητες δεδομένων, ώστε να μην αλλοιώνονται ποτέ από τον ανθρώπινη παρέμβαση. Ως αποτέλεσμα, υπάρχουν λιγότερες πιθανότητες για ανθρώπινα λάθη κατά τη συλλογή και καταχώρηση δεδομένων. Τα λάθη εξακολουθούν να είναι αρκετά τυπικά στην ανάπτυξη αισθητήρων που έχουν σχεδιαστεί από ανθρώπους, όπως για παράδειγμα η εσφαλμένη βαθμονόμηση και οι παρεμβολές σήματος από ανεπιθύμητες πηγές.

- **Σφάλματα "απόσταξης":** Πριν προστεθούν σε μια βάση δεδομένων, τα ακατέργαστα δεδομένα συχνά υποβάλλονται σε προεπεξεργασία και συνοψίζονται. Αυτή η "απόσταξη" δεδομένων πραγματοποιείται για διάφορους λόγους, μεταξύ άλλων για να μειωθεί η πολυπλοκότητα ή ο θόρυβος στα ακατέργαστα δεδομένα και σε ορισμένες περιπτώσεις απλώς για να μειωθεί ο όγκος των αποθηκευμένων δεδομένων. Όλες αυτές οι διαδικασίες ενέχουν τον κίνδυνο εισαγωγής σφαλμάτων στα δεδομένα που "αποστάζονται" ή στον τρόπο με τον οποίο η μέθοδος "απόσταξης" αλληλεπιδρά με την τελική ανάλυση.
- **Σφάλματα ενσωμάτωσης δεδομένων**: Μια βάση δεδομένων περιλαμβάνει δεδομένα που έχουν συγκεντρωθεί με την πάροδο του χρόνου από πολλές πηγές με διάφορους τρόπους σε όλα σχεδόν τα πλαίσια. Επιπλέον, πολλές βάσεις δεδομένων εξελίσσονται στην πράξη συνδυάζοντάς τες με άλλες βάσεις δεδομένων που προϋπάρχουνε- αυτή η διαδικασία συνδυασμού απαιτεί σχεδόν πάντα μια προσπάθεια αντιμετώπισης των αποκλίσεων μεταξύ των βάσεων δεδομένων όσον αφορά τις μορφές δεδομένων, τις μονάδες, τα διαστήματα μέτρησης κ.λπ. Κάθε διαδικασία που συνδυάζει δεδομένα από διάφορες πηγές έχει τη δυνατότητα να κάνει λάθη.

2.1.3 Προβλήματα με τον καθαρισμό δεδομένων

Οι κανόνες ποιότητας δεδομένων ή οι περιορισμοί ακεραιότητας (ICs) έχουν προταθεί ως δηλωτική τεχνική για την έκφραση νόμιμων ή ορθών περιπτώσεων δεδομένων προκειμένου να ανιχνεύονται προβλήματα δεδομένων. Τα λανθασμένα δεδομένα, γνωστά και ως παραβίαση, είναι οποιοδήποτε υποσύνολο δεδομένων που δεν ακολουθεί το σύνολο των κανόνων (Ilyas και Chu, 2012)

Έχουν αναπτυχθεί διάφοροι τύποι μεθόδων επιδιόρθωσης δεδομένων με διάφορους στόχους. Χρησιμοποιούνται αλγόριθμοι για την εύρεση υποσυνόλων δεδομένων που παραβιάζουν τους δηλωμένους περιορισμούς ακεραιότητας και ακόμη και για τη σύσταση αλλαγών στη βάση δεδομένων ώστε η νέα περίπτωση βάσης δεδομένων να συμμορφώνεται με αυτούς. Ενώ ορισμένοι από αυτούς τους αλγορίθμους επιδιώκουν να μεταβάλλουν τη βάση δεδομένων όσο το δυνατόν λιγότερο, άλλοι επιστρατεύουν τη βοήθεια ανθρώπινων ειδικών ή βάσεων γνώσεων για την επικύρωση των διορθώσεων που προτείνουν οι αυτόματοι αλγόριθμοι επανάληψης (Ilyas και Chu 2012).

2.2 Επισκόπηση προσεγγίσεων

Σύμφωνα με τους Ilyas και Chu (2017) ανεξάρτητα από το είδος των σφαλμάτων δεδομένων που πρέπει να διορθωθούν, οι διαδικασίες καθαρισμού δεδομένων περιλαμβάνουν συνήθως δύο φάσεις, την ανίχνευση και την επιδιόρθωση σφαλμάτων. Στην ανίχνευση εντοπίζονται και ίσως επαληθεύονται από εμπειρογνώμονες διάφορα ελαττώματα και παραβιάσεις ενώ στην επιδιόρθωση υλοποιούνται ενημερώσεις της βάσης δεδομένων για να καθαριστούν τα δεδομένα και να καταστούν κατάλληλα για αναλύσεις και μεταγενέστερες εφαρμογές. Οι δύο κύριες κατηγορίες προσεγγίσεων ανίχνευσης σφαλμάτων μπορεί να είναι ποσοτικές ή ποιοτικές. Για την αναγνώριση μη φυσιολογικών συμπεριφορών και σφαλμάτων, οι στατιστικές προσεγγίσεις και οι προσεγγίσεις μηχανικής μάθησης χρησιμοποιούνται συχνά στις ποσοτικές στρατηγικές ανίχνευσης σφαλμάτων (Hellerstein, 2008). Η ανίχνευση ακραίων τιμών (outlier detection) ήταν ο κύριος τομέας μελέτης για τα συστήματα ποσοτικής ανίχνευσης σφαλμάτων. Από την άλλη πλευρά, οι περιγραφικές προσεγγίσεις χρησιμοποιούνται από τα ποιοτικά συστήματα ανίχνευσης σφαλμάτων για να προσδιορίσουν τα πρότυπα ή τους περιορισμούς μιας περίπτωσης νομικών δεδομένων. Η χρήση προτύπων ποιότητας δεδομένων που ορίζονται σε ορισμένες γλώσσες περιορισμών ακεραιότητας είναι μια τυπική μέθοδος καθορισμού αυτών των προτύπων ή περιορισμών. Οι κανόνες για την ποιότητα των δεδομένων παραβιάζονται όταν χρησιμοποιούνται μεθοδολογίες για ποιοτικό εντοπισμό λαθών βάσει κανόνων (Aggarwal, 2013). Παράδειγμα, έστω ότι "δύο άτομα με τον ίδιο ταχυδρομικό κώδικα ζουν στην ίδια πολιτεία" είναι ένα κριτήριο ποιότητας δεδομένων για ένα στιγμιότυπο νόμιμου υπαλλήλου (legal employee instance). Μπορούμε να είμαστε βέβαιοι ότι τουλάχιστον μία από τις τιμές δεδομένων ταχυδρομικού κώδικα και πολιτείας για δύο υπαλλήλους σε μια περίπτωση δεδομένων είναι εσφαλμένη, εάν τους αναγνωρίσουμε ότι έχουν τον ίδιο ταχυδρομικό κώδικα αλλά διαφορετική πολιτεία.

Ειδικές πτυχές της ποιότητας των δεδομένων και του καθαρισμού των δεδομένων καλύπτονται σε διάφορες έρευνες και δημοσιεύσεις. Οι Rahm και Do (2000), για παράδειγμα, κατηγοριοποιούν τα πολλά προβλήματα που μπορούν να προκύψουν κατά τη διάρκεια μιας διαδικασίας εξαγωγής-μετασχηματισμού-φόρτωσης (ETL: extract, transform, load) και εξετάζουν τις διαθέσιμες λύσεις για τον καθαρισμό των δεδομένων κατά τη διάρκεια μιας διαδικασίας ETL.

Κατά τη φάση επιδιόρθωσης σφαλμάτων του καθαρισμού δεδομένων εφαρμόζονται επίσης script μετασχηματισμού δεδομένων, τα οποία συνήθως αναπτύσσονται σύμφωνα με τη διαδικασία που χρησιμοποιείται για τον εντοπισμό σφαλμάτων. Η άλλη πιθανότητα είναι να εμπλακούν ανθρώπινοι ειδικοί με τρόπο που βασίζεται σε αρχές, ή ένας συνδυασμός των δύο μεθόδων που αναφέρθηκαν.

Οι Rahm και Do 2018) κατηγοριοποιούν τα πρωταρχικά ζητήματα με την ποιότητα των δεδομένων που πρέπει να επιλυθούν μέσω του καθαρισμού και του μετασχηματισμού των δεδομένων. Τα ζητήματα αυτά είναι αλληλένδετα και θα έπρεπε να αντιμετωπίζονται με παρόμοιο τρόπο. Για να υποστηριχθούν οποιεσδήποτε αλλαγές στη δομή, την αναπαράσταση ή το περιεχόμενο των δεδομένων, απαιτούνται μετασχηματισμοί δεδομένων.

Αυτοί οι μετασχηματισμοί απαιτούνται σε διάφορες περιπτώσεις, όπως η αντιμετώπιση της εξέλιξης του σχήματος, η μετακίνηση ενός παλαιού συστήματος σε ένα νέο πληροφοριακό σύστημα ή η ενσωμάτωση διαφορετικών πηγών δεδομένων. Διακρίνουμε χονδρικά μεταξύ προβλημάτων

που σχετίζονται με σχήματα και περιπτώσεις, καθώς και μεταξύ προβλημάτων μίας και πολλαπλών πηγών. Φυσικά, τα ζητήματα σε επίπεδο σχήματος (schema-level) αντικατοπτρίζονται επίσης στα παραδείγματα- τα ζητήματα αυτά μπορούν να επιλυθούν σε επίπεδο σχήματος με τη βελτίωση του σχεδιασμού του σχήματος (εξέλιξη του σχήματος), τη μετάφραση του σχήματος και την ενσωμάτωση του σχήματος. Από την άλλη πλευρά, τα ζητήματα σε επίπεδο παραδείγματος (instance-level) αφορούν λάθη και αποκλίσεις στο περιεχόμενο των πραγματικών δεδομένων που δεν είναι εμφανή στο επίπεδο σχήματος. Η εκκαθάριση δεδομένων επικεντρώνεται κυρίως σε αυτά. Τα προβλήματα μίας πηγής, εκτός από τα μοναδικά προβλήματα πολλαπλών πηγών, υπάρχουν (με μεγαλύτερη πιθανότητα) και στο σενάριο πολλαπλών πηγών.

2.2.1 Προβλήματα μίας πηγής

Ο βαθμός στον οποίο τα δεδομένα μιας πηγής περιορίζονται από τους περιορισμούς ακεραιότητας και τους περιορισμούς σχήματος που διέπουν τις αποδεκτές τιμές δεδομένων καθορίζει την ποιότητα των δεδομένων που περιέχει. Υπάρχουν περιορισμένοι περιορισμοί ως προς τα δεδομένα που μπορούν να εισαχθούν και να διατηρηθούν για πηγές χωρίς σχήμα, όπως τα αρχεία, γεγονός που αυξάνει την πιθανότητα σφαλμάτων και ασυνέπειας. Αντίθετα, τα συστήματα βάσεων δεδομένων επιβάλλουν τόσο περιορισμούς ακεραιότητας που αφορούν συγκεκριμένες εφαρμογές όσο και περιορισμούς που σχετίζονται με μια συγκεκριμένη αρχιτεκτονική δεδομένων (π.χ. η σχεσιακή προσέγγιση απαιτεί απλές τιμές χαρακτηριστικών, αναφορική ακεραιότητα κ.λπ.)

Επομένως, τα προβλήματα ποιότητας δεδομένων που σχετίζονται με το σχήμα προκύπτουν από την έλλειψη κατάλληλων περιορισμών ακεραιότητας για συγκεκριμένο μοντέλο ή εφαρμογή, οι οποίοι μπορεί να οφείλονται σε διάφορους παράγοντες, όπως περιορισμοί του μοντέλου δεδομένων, ανεπαρκής σχεδιασμός του σχήματος ή ορισμός μικρού αριθμού περιορισμών ακεραιότητας για τη μείωση του φόρτου του ελέγχου ακεραιότητας. Τα ζητήματα που αφορούν μόνο μια περίπτωση περιλαμβάνουν λάθη και ασυμφωνίες που δεν μπορούν να αποφευχθούν σε επίπεδο σχήματος (π.χ. ορθογραφικά λάθη).

Μπορούμε να διακρίνουμε διάφορα πεδία προβλημάτων, όπως το χαρακτηριστικό (πεδίο), η εγγραφή, ο τύπος εγγραφής και η πηγή, τόσο για θέματα σε επίπεδο σχήματος όσο και σε επίπεδο παραδείγματος. Θα πρέπει να σημειωθεί ότι οι περιορισμοί μοναδικότητας που τίθενται σε επίπεδο σχήματος δεν εμποδίζουν την εμφάνιση διπλών περιπτώσεων, για παράδειγμα, εάν οι πληροφορίες για το ίδιο πράγμα του πραγματικού κόσμου εισαχθούν ξανά με διαφορετικές τιμές χαρακτηριστικών.

Η αποτροπή της καταχώρησης βρώμικων δεδομένων είναι προφανώς ένα κρίσιμο βήμα προκειμένου να μειωθεί το πρόβλημα του καθαρισμού, δεδομένου ότι ο καθαρισμός των πηγών δεδομένων είναι μια δαπανηρή διαδικασία. Αυτό απαιτεί τον κατάλληλο σχεδιασμό των εφαρμογών εισαγωγής δεδομένων, του σχήματος της βάσης δεδομένων και των περιορισμών ακεραιότητας. Επιπλέον, η ανακάλυψη των κατευθυντήριων γραμμών καθαρισμού δεδομένων κατά τον σχεδιασμό της αποθήκης μπορεί να προσφέρει προτάσεις για την ενίσχυση των περιορισμών που επιβάλλονται από τα τρέχοντα σχήματα.

2.2.2 Προβλήματα πολλαπλών πηγών

Όταν πρέπει να συνδυαστούν πολλές πηγές, τα προβλήματα που είναι εμφανή στις μεμονωμένες πηγές επιδεινώνονται. Κάθε πηγή μπορεί να έχει ανακριβείς πληροφορίες και οι πληροφορίες που περιέχονται σε αυτήν μπορεί να παρουσιάζονται με διάφορους τρόπους, να επικαλύπτονται ή να έρχονται σε σύγκρουση. Αυτό οφείλεται στο γεγονός ότι οι πηγές συχνά δημιουργούνται, υλοποιούνται και διατηρούνται ανεξάρτητα για την ικανοποίηση συγκεκριμένων απαιτήσεων. Ως αποτέλεσμα, τα συστήματα διαχείρισης δεδομένων, τα μοντέλα δεδομένων, τα σχέδια σχημάτων και τα πραγματικά δεδομένα είναι εξαιρετικά ετερογενή.

Τα βήματα της μετάφρασης σχήματος και της ενοποίησης σχήματος, αντίστοιχα, έχουν ως στόχο να αντιμετωπίσουν τις διαφορές στο μοντέλο δεδομένων και στο σχεδιασμό σχήματος σε επίπεδο σχήματος. Όσον αφορά το σχεδιασμό του σχήματος, η ονοματολογία και οι δομικές συγκρούσεις είναι τα βασικά ζητήματα. Οι συγκρούσεις ονοματοδοσίας συμβαίνουν όταν δύο ή περισσότερα αντικείμενα έχουν το ίδιο όνομα (ομώνυμα) ή όταν το ίδιο αντικείμενο έχει πολλαπλά ονόματα (συνώνυμα). Οι δομικές συγκρούσεις αφορούν τις διαφορετικές αναπαραστάσεις του ίδιου αντικειμένου σε διαφορετικές πηγές και μπορούν να λάβουν πολλές διαφορετικές μορφές, όπως αναπαράσταση χαρακτηριστικών έναντι πίνακα, διαφορετική δομή συστατικών, διαφορετικοί τύποι δεδομένων, διαφορετικοί περιορισμοί ακεραιότητας κ.λπ.

Εκτός από τις συγκρούσεις σε επίπεδο σχήματος, πολλές συγκρούσεις εμφανίζονται μόνο σε επίπεδο στιγμιοτύπου (instance) (συγκρούσεις δεδομένων). Διαφορετικές αναπαραστάσεις σε διάφορες πηγές μπορούν να προκαλέσουν όλα τα προβλήματα από την κατάσταση μιας και μόνο πηγής (π.χ. αντιφατικές εγγραφές, διπλές εγγραφές). Επιπλέον, ακόμη και μεταξύ πηγών που χρησιμοποιούν τα ίδια ονόματα χαρακτηριστικών και τύπους δεδομένων, μπορεί να υπάρχουν διαφορές στις αναπαραστάσεις τιμών (όπως για την κατάσταση γάμου) ή στις ερμηνείες των τιμών (όπως όταν η μέτρηση γίνεται σε λίρες σε αντίθεση με τα ευρώ). Επιπλέον, τα δεδομένα στις πηγές μπορεί να παρουσιάζονται σε διάφορα επίπεδα συνάθροισης (π.χ. πωλήσεις ανά προϊόν έναντι πωλήσεων ανά κατηγορία προϊόντος) ή να αφορούν διάφορες χρονικές στιγμές (π.χ. τρέχουσες πωλήσεις από χθες για την πηγή 1 έναντι πωλήσεων της περασμένης εβδομάδας για την πηγή 2).

Η εύρεση επικαλυπτόμενων δεδομένων, ιδίως καταχωρίσεων που ταιριάζουν και αφορούν το ίδιο αντικείμενο του πραγματικού κόσμου, αποτελεί σημαντική πρόκληση κατά τον καθαρισμό δεδομένων από πολλές πηγές (π.χ. πελάτη). Το πρόβλημα της ταυτότητας αντικειμένου, η εξάλειψη της επικάλυψης και τα προβλήματα συγχώνευσης/καθαρισμού είναι άλλα ονόματα για αυτό το ζήτημα. Οι πληροφορίες είναι συχνά μόνο εν μέρει περιττές και οι πηγές μπορεί να προσθέτουν η μία στη γνώση της άλλης για μια οντότητα δίνοντας περισσότερες λεπτομέρειες. Προκειμένου να επιτευχθεί μια συνεπής εικόνα των πραγμάτων του πραγματικού κόσμου, οι περιττές πληροφορίες πρέπει να εξαλειφθούν και οι συμπληρωματικές πληροφορίες πρέπει να συνδυαστούν.

2.2.3 Προσεγγίσεις καθαρισμού δεδομένων

Υπάρχουν διάφορες προσεγγίσεις καθαρισμού δεδομένων σύμφωνα με τους Rahm και Do. Αυτές αναλύονται παρακάτω:

Ανάλυση δεδομένων: Είναι απαραίτητη η ενδελεχής ανάλυση των δεδομένων για να καθοριστούν τα είδη των σφαλμάτων και των ασυνεπειών που πρέπει να εξαλειφθούν. Για τη συλλογή πληροφοριών σχετικά με τις ιδιότητες των δεδομένων και τον εντοπισμό ζητημάτων με την ποιότητα των δεδομένων, θα πρέπει να χρησιμοποιούνται προγράμματα ανάλυσης εκτός από τη χειροκίνητη αξιολόγηση των δεδομένων ή των δειγμάτων δεδομένων.

Επαλήθευση: Για να διασφαλιστεί ότι μια ροή εργασίας μετασχηματισμού είναι σωστή και αποτελεσματική, οι ορισμοί για τον μετασχηματισμό θα πρέπει να δοκιμαστούν και να αξιολογηθούν, για παράδειγμα σε ένα δείγμα ή αντίγραφο των δεδομένων προέλευσης. Τα βήματα ανάλυσης, σχεδιασμού και επαλήθευσης μπορεί να χρειαστεί να επαναληφθούν αρκετές φορές, για παράδειγμα, καθώς ορισμένα σφάλματα ανακαλύπτονται μόνο μετά την πραγματοποίηση ορισμένων μετασχηματισμών.

Ορισμός της ροής εργασίας μετασχηματισμού και των κανόνων αντιστοίχισης: Ενδέχεται να χρειαστεί να πραγματοποιηθεί σημαντικός αριθμός σταδίων μετασχηματισμού και καθαρισμού δεδομένων, ανάλογα με την ποσότητα των πηγών δεδομένων, τον βαθμό ετερογένειάς τους και το πόσο «βρώμικα» είναι τα δεδομένα. Οι πηγές περιστασιακά αντιστοιχίζονται σε ένα κοινό μοντέλο δεδομένων χρησιμοποιώντας μια μετάφραση σχήματος- οι αποθήκες δεδομένων συχνά χρησιμοποιούν μια σχεσιακή αναπαράσταση. Οι έγκαιρες διαδικασίες καθαρισμού δεδομένων μπορούν να επιλύσουν προβλήματα με περιπτώσεις μίας πηγής και να καταστήσουν τα δεδομένα έτοιμα για ενσωμάτωση. Οι μεταγενέστερες διαδικασίες ενσωματώνουν το σχήμα και τα δεδομένα και διορθώνουν ζητήματα με πολλές πηγές δεδομένων, όπως τα αντίγραφα. Για την αποθήκευση δεδομένων, θα πρέπει να χρησιμοποιείται μια ροή εργασίας που περιγράφει τη διαδικασία εξαγωγής, μετασχηματισμού και φόρτωσης για να προσδιορίζεται ο έλεγχος και η ροή δεδομένων για αυτές τις φάσεις μετασχηματισμού και καθαρισμού.

Για να καταστεί δυνατή η αυτόματη ανάπτυξη του κώδικα μετασχηματισμού, οι μετασχηματισμοί δεδομένων που σχετίζονται με το σχήμα καθώς και οι διαδικασίες καθαρισμού θα πρέπει να παρέχονται από μια δηλωτική γλώσσα ερωτημάτων και αντιστοίχισης. Επιπλέον, καθ' όλη τη διάρκεια μιας διαδικασίας μετασχηματισμού δεδομένων, θα πρέπει να επιτρέπεται η κλήση κώδικα καθαρισμού γραμμένου από τον χρήστη και εξειδικευμένων εργαλείων. Εάν δεν υπάρχει ενσωματωμένη λογική καθαρισμού για μια συγκεκριμένη περίπτωση δεδομένων, τα βήματα μετασχηματισμού μπορούν να ζητούν από τον χρήστη σχόλια.

Μετασχηματισμός: η διαδικασία εκτέλεσης εξαγωγής, μετασχηματισμού και φόρτωσης για τη φόρτωση και την ανανέωση μιας αποθήκης δεδομένων ή κατά την απάντηση σε ερωτήματα από διάφορες πηγές.

Αντίστροφη ροή καθαρισμένων δεδομένων: Μετά την αφαίρεση των λαθών (μίας πηγής), τα καθαρισμένα δεδομένα θα πρέπει επίσης να αντικαταστήσουν τα βρώμικα δεδομένα των αρχικών πηγών, προκειμένου να παρέχονται στις παλαιότερες εφαρμογές τα βελτιωμένα δεδομένα, καθώς και για να αποφεύγεται η επανάληψη της διαδικασίας καθαρισμού για μεταγενέστερες εξορύξεις δεδομένων. Τα καθαρισμένα δεδομένα είναι προσβάσιμα μέσω της περιοχής αποθήκευσης δεδομένων για την αποθήκευση δεδομένων.

3

Ροή εργασιών καθαρισμού δεδομένων

Συχνά προκειμένου να καθαρίσουμε ένα "βρώμικο" σύνολο δεδομένων, χρειάζεται να αναπαραστήσουμε διάφορες πτυχές αυτών, όπως σχήματα, μοτίβα, κατανομές πιθανοτήτων και άλλες πληροφορίες. Το στάδιο ανίχνευσης σφαλμάτων μπορεί να αποκαλύψει μια ποικιλία λαθών, συμπεριλαμβανομένων των ακραίων τιμών, των παραβιάσεων και των αντιγράφων. Στο στάδιο αυτό γίνονται αλλαγές δεδομένων που εφαρμόζονται με τη σειρά τους στο "βρώμικο" σύνολο δεδομένων, με βάση τα σφάλματα που ανακαλύφθηκαν και τα μεταδεδομένα που προκαλούν αυτά τα σφάλματα. Προκειμένου να διασφαλιστεί η ακρίβεια της ροής εργασίας καθαρισμού, ανθρώπινο δυναμικό που αποτελείται από ειδικούς προσφέρουν την βοήθεια τους, όποτε αυτό είναι πρακτικό, καθώς η διαδικασία αυτή συνεπάγεται χρόνο και κόστος.

3.1 Ανίχνευση ακραίων τιμών

Όταν χρησιμοποιούμε τη φράση ανίχνευση ακραίων τιμών, μπορεί να αποδοθούν διαφορετικοί ορισμοί αναλόγως με την εκάστοτε εφαρμογή σε κάθε περίσταση. Παρά ταύτα, υπάρχουν ορισμένοι αποδεκτοί ορισμοί ευρέως, όπως "ακραία είναι μια παρατήρηση που αποκλίνει τόσο πολύ από άλλες παρατηρήσεις, ώστε να προκαλεί υποψίες ότι δημιουργήθηκε από έναν διαφορετικό μηχανισμό" (Hawkins, 1980) και "ακραία είναι μια παρατήρηση που φαίνεται να αποκλίνει σημαντικά από άλλα μέλη του δείγματος στο οποίο εμφανίζεται" (Barnett και Lewis, 1988). Για παράδειγμα όταν ο μισθός ενός υπαλλήλου παρεκκλίνει από το μέσο μηνιαίο μισθό των υπόλοιπων υπαλλήλων, τότε αυτό μπορεί να θεωρηθεί περίεργο.

Ένας τρόπος ανίχνευσης ακραίων τιμών μπορεί να γίνει με τη χρήση μεθόδων ομαδοποίησης, όπως για παράδειγμα η εφαρμογή μιας μεθόδου ιεραρχικής ομαδοποίησης που έχει σαν σκοπό την ανίχνευση ακραίων τιμών. Η ανομοιόμορφη κατανομή των ακραίων έναντι των "κανονικών" περιπτώσεων στα σύνολα δεδομένων, είναι αυτή που οδηγεί στη χρήση της ιεραρχικής ομαδοποίησης. Γενικότερα οι ακραίες τιμές είναι τιμές που δεν ακολουθούν τη στατιστική κατανομή του μεγαλύτερου μέρους των δεδομένων και κατά συνέπεια μπορεί να οδηγήσουν σε εσφαλμένα αποτελέσματα, όσον αφορά τη στατιστική ανάλυση. Υπάρχουν δύο βασικά εμπόδια στην ανίχνευση ακραίων τιμών. Πρώτον, δεδομένου ότι τα διάφορα δεδομένα και οι εφαρμογές

έχουν διαφορετικούς ορισμούς του τι είναι τυπικό, μπορεί να είναι δύσκολο να διακρίνουμε μεταξύ μιας κανονικής τάσης δεδομένων και μιας ακραίας τιμής. Για τον ορισμό της κανονικής συμπεριφοράς, έχει προταθεί ένα ευρύ φάσμα μεθόδων ανίχνευσης. Δεύτερον, πολλές στρατηγικές εντοπισμού ακραίων τιμών γίνονται λιγότερο αποτελεσματικές, καθώς αυξάνονται οι τιμές – χαρακτηριστικά του συνόλου δεδομένων. Το φαινόμενο αυτό αναφέρεται και ως "κατάρα της διαστατικότητας" (curse of dimensionality).

Υπάρχουν πολυάριθμες χρήσεις για την ανίχνευση ακραίων τιμών. Διαφορετικοί τύποι δεδομένων, συμπεριλαμβανομένων των κλήσεων του λειτουργικού συστήματος και της δικτυακής κίνησης, συγκεντρώνονται σε μεγάλες ποσότητες στο πλαίσιο των δικτύων υπολογιστών. Στα λαμβανόμενα δεδομένα, η αναγνώριση ακραίων τιμών μπορεί να βοηθήσει στην ανίχνευση πιθανών εισβολών και επιβλαβών δραστηριοτήτων. Τα μη εξουσιοδοτημένα άτομα εμφανίζουν συχνά μοναδικά μοτίβα αγορών όταν διαπράττουν κλοπή πιστωτικών καρτών, όπως π.χ. το να πηγαίνουν για ψώνια από διαφορετική πόλη. Η ανίχνευση απάτης είναι η διαδικασία εντοπισμού παράνομης δραστηριότητας σε αυτού του είδους τις οικονομικές συναλλαγές. Τέλος, ο αυτόματος εντοπισμός ανώμαλων μοτίβων ή εγγραφών σε αρχεία ιατρικών δεδομένων, όπως σαρώσεις μαγνητικής τομογραφίας, σαρώσεις PET, συνήθως σηματοδοτεί την παρουσία μιας ασθένειας και μπορεί να βοηθήσει στην έγκαιρη διάγνωση.

Ο εντοπισμός των ακραίων τιμών δεν είναι εύκολη υπόθεση. Πρώτα απ' όλα, μπορεί να είναι δύσκολο να καθοριστούν τα κανονικά πρότυπα δεδομένων για ένα κανονιστικό πρότυπο. Για παράδειγμα, οι αιχμές στις καταγεγραμμένες ενδείξεις από φορητές συσκευές μπορούν να θεωρηθούν ως φυσιολογικές εάν εμπίπτουν σε ένα προκαθορισμένο εύρος που καθορίζεται από τους χρησιμοποιούμενους αισθητήρες. Από την άλλη πλευρά, κατά την εξέταση δεδομένων προσωπικού, οι αιχμές τιμών μισθών είναι πιθανώς ενδιαφέρουσες ακραίες τιμές. Επομένως, η επιλογή του κατάλληλου εργαλείου για μια συγκεκριμένη εργασία εφαρμογής απαιτεί επίγνωση των προϋποθέσεων και των περιορισμών κάθε μεθόδου αναγνώρισης ακραίων τιμών.

3.1.1 Ταξινόμηση των μεθόδων ανίχνευσης ακραίων τιμών

Η κύρια διαφορά μεταξύ των μεθόδων εντοπισμού ακραίων τιμών είναι ο τρόπος με τον οποίο ορίζουν την τυπική συμπεριφορά. Αυτές οι μέθοδοι κατηγοριοποιούνται σε τρεις κύριες ομάδες από τους Ilyas και Chu (2019): στρατηγικές εντοπισμού ακραίων τιμών με βάση τη στατιστική, τεχνικές εντοπισμού ακραίων τιμών με βάση την απόσταση και τεχνικές εντοπισμού ακραίων τιμών με βάση το μοντέλο (Aggarwal, 2013, Chandola κ.ά., 2009, Hodge και Austin, 2004).

3.1.1.1 Μέθοδοι ανίχνευσης ακραίων τιμών με βάση τη στατιστική

Οι τεχνικές αναγνώρισης ακραίων τιμών που βασίζονται στη στατιστική βασίζονται στην ιδέα ότι ενώ οι ακραίες τιμές θα εμφανίζονταν σε περιοχές χαμηλής πιθανότητας ενός στοχαστικού μοντέλου, τα κανονικά σημεία δεδομένων θα εμφανίζονταν σε περιοχές υψηλής πιθανότητας (Chandola κ.ά. 2009). Για τον εντοπισμό ακραίων τιμών με βάση τη στατιστική, υπάρχουν δύο ομάδες τεχνικών που χρησιμοποιούνται συχνά. Το Grubbs Test (Grubbs, 1969) και το Tietjen-

Moore Test (Tietjen και Moore, 1972) είναι δύο παραδείγματα τεχνικών ελέγχου υποθέσεων που βασίζονται σε παρατηρούμενα σημεία δεδομένων. Αυτές οι τεχνικές συνήθως υπολογίζουν ένα στατιστικό ελέγχου με βάση τα παρατηρούμενα σημεία δεδομένων, το οποίο χρησιμοποιείται για να αποφασιστεί εάν η μηδενική υπόθεση (ότι δεν υπάρχει ακραία τιμή στο σύνολο δεδομένων) πρέπει να απορριφθεί. Η δεύτερη ομάδα στατιστικών μεθόδων ανίχνευσης ακραίων τιμών επικεντρώνεται στην προσαρμογή μιας κατανομής στα παρατηρούμενα δεδομένα ή στην εξαγωγή μιας συνάρτησης πυκνότητας πιθανότητας (probability density function). Ως ακραίες τιμές ορίζονται τα σημεία δεδομένων με χαμηλή πιθανότητα σύμφωνα με το pdf. Υπάρχουν δύο ακόμη κατηγορίες μεθόδων προσαρμογής κατανομής: παραμετρικές προσεγγίσεις και μη παραμετρικές προσεγγίσεις. Στόχος των παραμετρικών τεχνικών προσαρμογής κατανομών είναι ο προσδιορισμός των παραμέτρων της κατανομής από τα παρατηρούμενα δεδομένα με την υπόθεση ότι τα δεδομένα ακολουθούν μια υποκείμενη κατανομή. Για παράδειγμα, εάν τα δεδομένα έχουν κανονική κατανομή, οι παραμετρικές τεχνικές θα πρέπει να βρουν τη μέση τιμή και τη διακύμανση της κανονικής κατανομής. Οι μη παραμετρικές μέθοδοι, από την άλλη πλευρά, συμπεραίνουν την κατανομή από τα δεδομένα χωρίς να κάνουν υποθέσεις σχετικά με την κατανομή που παρήγαγε τα δεδομένα.

Η χρήση μεθόδων με βάση τη στατιστική έχει τα εξής πλεονεκτήματα: Πρώτον, οι στατιστικές προσεγγίσεις εντοπισμού ακραίων τιμών μπορούν να προσφέρουν μια στατιστική εξήγηση για τις κρυμμένες ακραίες τιμές, εάν τα υποκείμενα δεδομένα ακολουθούν μια συγκεκριμένη κατανομή. Δεύτερον, αντί να αποδίδουν μια δυαδική κρίση, οι στατιστικές διαδικασίες συνήθως αποδίδουν σε κάθε σημείο δεδομένων μια βαθμολογία ή ένα διάστημα εμπιστοσύνης. Κατά την επιλογή ενός σημείου δεδομένων δοκιμής, η βαθμολογία μπορεί να χρησιμοποιηθεί ως συμπληρωματική πληροφορία. Τρίτον, οι στατιστικές μέθοδοι εκτελούνται συνήθως χωρίς επίβλεψη, χωρίς να απαιτούν επισημασμένα δεδομένα εκπαίδευσης.

Παρόλα αυτά, υπάρχουν ορισμένα μειονεκτήματα στις μεθόδους που βασίζονται στη στατιστική: Η υπόθεση ότι τα δεδομένα παράγονται από μια συγκεκριμένη κατανομή είναι μια υπόθεση στην οποία βασίζονται συνήθως οι στατιστικές προσεγγίσεις. Αυτή η υπόθεση είναι συχνά εσφαλμένη, ιδίως για πραγματικά σύνολα δεδομένων με πολλές διαστάσεις. Επίσης, υπάρχουν διάφορα στατιστικά στοιχεία ελέγχου υποθέσεων που μπορούν να χρησιμοποιηθούν για την ανακάλυψη ανωμαλιών, ακόμη και όταν η στατιστική υπόθεση μπορεί να δικαιολογηθεί σωστά- ωστόσο, η επιλογή του σωστού στατιστικού στοιχείου δεν είναι συχνά εύκολη υπόθεση. Είναι δύσκολο να κατασκευαστούν έλεγχοι υποθέσεων για τις περίπλοκες κατανομές που απαιτούνται για την προσαρμογή συνόλων δεδομένων υψηλών διαστάσεων.

3.1.1.2 Μέθοδοι ανίχνευσης ακραίων τιμών με βάση την απόσταση.

Η απόσταση μεταξύ των σημείων δεδομένων που χρησιμοποιείται για τον ορισμό μιας κανονικής συμπεριφοράς ορίζεται συχνά με τη χρήση τεχνικών ανίχνευσης ακραίων τιμών που βασίζονται στην απόσταση. Ένα κανονικό σημείο δεδομένων, για παράδειγμα, θα έπρεπε να είναι κοντά σε πολλά άλλα σημεία δεδομένων, και τα σημεία δεδομένων που αποκλίνουν από αυτή την κανονική συμπεριφορά χαρακτηρίζονται ως ακραία (Knorr και Ng, 1998, 1999, Breunig κ.ά., 2000). Ανάλογα

με τον πληθυσμό αναφοράς που χρησιμοποιείται για να καθοριστεί αν ένα σημείο είναι ακραίο, οι προσεγγίσεις για την ανίχνευση ακραίων σημείων με βάση την απόσταση μπορούν να ταξινομηθούν περαιτέρω ως παγκόσμιες ή τοπικές μέθοδοι. Με βάση την απόσταση μεταξύ ενός σημείου δεδομένων και κάθε άλλου σημείου δεδομένων στο σύνολο δεδομένων, μια προσέγγιση εντοπισμού ακραίων σημείων με βάση την παγκόσμια απόσταση αξιολογεί εάν ένα σημείο είναι ακραίο. Από την άλλη πλευρά, μια τοπική τεχνική λαμβάνει υπόψη την απόσταση μεταξύ ενός σημείου και των σημείων που το περιβάλλουν κατά τον εντοπισμό των ακραίων τιμών.

Η χρήση προσεγγίσεων που βασίζονται στην απόσταση έχει κάποια πλεονεκτήματα: πρώτον, το γεγονός ότι οι αλγόριθμοι που βασίζονται στην απόσταση είναι από τη φύση τους μη επιβλεπόμενοι και δεν κάνουν καμία υπόθεση σχετικά με τη γενεσιουργό κατανομή των δεδομένων αποτελεί σημαντικό πλεονέκτημα. Αντιθέτως, είναι εξ ολοκλήρου καθοδηγούμενοι από τα δεδομένα. Δεύτερον, είναι απλή η προσαρμογή των τεχνικών που βασίζονται στην απόσταση σε διάφορους τύπους δεδομένων- το μόνο που χρειάζεται είναι να οριστεί το κατάλληλο μέτρο απόστασης για τα δεδομένα που βρίσκονται υπό εξέταση.

Τα μειονεκτήματα της χρήσης προσεγγίσεων που βασίζονται στην απόσταση είναι: Πρώτον, οι προσεγγίσεις που βασίζονται στην απόσταση δεν θα κατηγοριοποιήσουν σωστά τα κανονικά παραδείγματα ή τις ανωμαλίες εάν δεν υπάρχουν αρκετά κοντινά κανονικά παραδείγματα ή ανωμαλίες στα δεδομένα, αντίστοιχα. Δεύτερον, δεδομένου ότι ο υπολογισμός της απόστασης μεταξύ κάθε ζεύγους σημείων δεδομένων είναι ένα απαραίτητο βήμα στη φάση της δοκιμής, παρουσιάζει ένα σημαντικό εμπόδιο υπολογιστικής πολυπλοκότητας. Τρίτον, πρέπει να αναπτυχθεί ένα μέτρο απόστασης που να μπορεί να διακρίνει αποτελεσματικά μεταξύ κανονικών και ανώμαλων παραδειγμάτων μεταξύ ενός ζεύγους περιπτώσεων δεδομένων, προκειμένου μια τεχνική που βασίζεται στον πλησιέστερο γείτονα να έχει καλή απόδοση. Όταν τα δεδομένα είναι πολύπλοκα, όπως σε γραφήματα, ακολουθίες και άλλους τύπους δεδομένων, ο καθορισμός μέτρων απόστασης μεταξύ των περιπτώσεων μπορεί να είναι δύσκολος.

3.1.1.3 Μέθοδοι ανίχνευσης ακραίων τιμών βασισμένες σε μοντέλα.

Προκειμένου να εκτιμηθεί εάν ένα σημείο δεδομένων δοκιμής είναι ακραίο, οι αλγόριθμοι ανίχνευσης ακραίων σημείων που βασίζονται σε μοντέλα δημιουργούν πρώτα ένα μοντέλο ταξινομητή χρησιμοποιώντας ένα σύνολο επισημασμένων σημείων δεδομένων. Οι τεχνικές που βασίζονται σε μοντέλα προϋποθέτουν ότι, δεδομένου ενός χώρου χαρακτηριστικών, ένας ταξινομητής μπορεί να διδαχθεί να διακρίνει μεταξύ κανονικών και ανώμαλων σημείων δεδομένων. Εάν κανένα από τα μαθημένα μοντέλα δεν κατηγοριοποιεί ένα δεδομένο σύνολο σημείων δεδομένων ως φυσιολογικά, αυτά χαρακτηρίζονται ως ακραία. Οι προσεγγίσεις που βασίζονται σε μοντέλα μπορούν να διαχωριστούν περαιτέρω σε δύο υποκατηγορίες: τεχνικές που βασίζονται σε μοντέλα πολλαπλών κατηγοριών και τεχνικές που βασίζονται σε μοντέλα μίας κατηγορίας, ανάλογα με τις ετικέτες που είναι διαθέσιμες για την εκπαίδευση του ταξινομητή. Οι τεχνικές που βασίζονται σε μοντέλα πολλαπλών κλάσεων βασίζονται στην υπόθεση ότι τα σημεία δεδομένων εκπαίδευσης περιλαμβάνουν επισημασμένες περιπτώσεις από διάφορες κανονικές κλάσεις (De Stefano κ.ά., 2000). Ωστόσο, οι μέθοδοι που βασίζονται σε μοντέλα μίας κλάσης κάνουν την υπόθεση ότι κάθε σημείο δεδομένων εκπαίδευσης ανήκει σε μία μόνο κανονική κλάση.

Ορισμένα πλεονεκτήματα των τεχνικών που βασίζονται σε μοντέλα είναι ότι πρώτον οι βασισμένες στο μοντέλο προσεγγίσεις, ιδίως εκείνες με πολλές κλάσεις, μπορούν να χρησιμοποιήσουν ισχυρούς αλγόριθμους για να ξεχωρίσουν περιπτώσεις από διάφορες κλάσεις και δεύτερον, η φάση δοκιμής των προσεγγίσεων που βασίζονται σε μοντέλα είναι γρήγορη, καθώς κάθε περίπτωση δοκιμής πρέπει να συγκρίνεται με το προ-υπολογισμένο μοντέλο.

Ένα από τα κύρια μειονεκτήματα των προσεγγίσεων που βασίζονται σε μοντέλα είναι ότι συχνά δεν μπορούν να χρησιμοποιηθούν, καθώς εξαρτώνται από την ύπαρξη ακριβών ετικετών για τις διάφορες κανονικές κλάσεις.

3.2 Απαλοιφή διπλότυπων δεδομένων

Πολλοί είναι οι παράγοντες που μπορεί να οδηγήσουν σε διπλές εγγραφές. Η απαλοιφή διπλότυπων δεδομένων, γνωστή και ως ανίχνευση αντιγράφων ή ανάλυση οντοτήτων, αναφέρεται στη διαδικασία εντοπισμού πλειάδων σε μία ή περισσότερες σχέσεις, που αναφέρονται στην ίδια οντότητα του πραγματικού κόσμου. Για παράδειγμα, εάν ένας πελάτης πραγματοποίησε πολλαπλές αγορές χρησιμοποιώντας διαφορετικά ονόματα, ο πελάτης μπορεί να έχει πολλαπλές εγγραφές σε μια βάση δεδομένων πελατών. Ο σχεδιασμός μετρικών ομοιότητας για την αξιολόγηση της ομοιότητας για ένα ζεύγος εγγραφών, η ομαδοποίηση όλων των εγγραφών για την απόκτηση ομάδων εγγραφών που αντιπροσωπεύουν την ίδια οντότητα του πραγματικού κόσμου, η ενοποίηση ομάδων εγγραφών για τη δημιουργία μοναδικών αναπαραστάσεων και ο σχεδιασμός στρατηγικών αποκλεισμού ή κατανεμημένων στρατηγικών για την κλιμάκωση της διαδικασίας απαλοιφής διπλότυπων δεδομένων είναι τυπικά βήματα και αποφάσεις σε τέτοια διαδικασία.

Για την αντιμετώπιση αυτού του προβλήματος έχουν αναπτυχθεί πολυάριθμα προγράμματα απαλοιφής διπλότυπων δεδομένων ανοικτού κώδικα και κερδοσκοπικά προγράμματα, όπως το Data Tamer (Stonebraker κ.ά., 2014) και το Magellan (Konda κ.ά., 2016).

Στην απαλοιφή διπλότυπων δεδομένων, είναι δύσκολο να επιτευχθεί ταυτόχρονα καλή ακρίβεια και ανάκληση. Η ανακήρυξη όλων των ζευγών ως αντιγράφων, έχει ως αποτέλεσμα τέλεια ανάκληση αλλά κακή ακρίβεια, ενώ η ανακήρυξη κανενός ζεύγους ως αντίγραφου έχει ως αποτέλεσμα τέλεια ακρίβεια αλλά κακή ανάκληση. Επιπλέον, η διαδικασία αυτή είναι από τη φύση της ένα συνδυαστικό πρόβλημα τετραγωνικής πολυπλοκότητας. Για παράδειγμα, η σύγκριση όλων των ζευγών για 1000 εγγραφές, απαιτεί 499.500 συγκρίσεις (Elsner και Schudy, 2009).

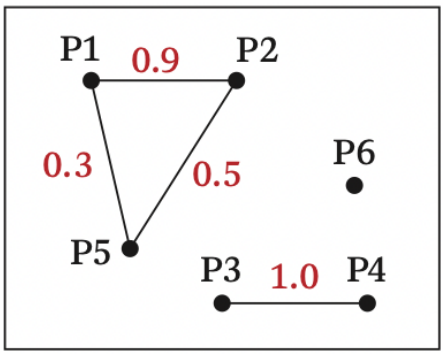
Για να μπορέσουμε να κατανοήσουμε καλύτερα τα διπλότυπα δεδομένα, μπορούμε να πάρουμε σαν παράδειγμα τον πίνακα 3.1 που φαίνεται παρακάτω.

ID	Name	ZIP	Income
P1	Green	51519	30k
P2	Green	51518	32k
P3	Peter	30528	40k
P4	Peter	30528	40k
P5	Gree	51519	55k
P6	Chuck	51519	30k

Πίνακας 3.1 Διπλότυπα δεδομένα πριν τον καθαρισμό
(Ilyas και Chu, 2019, σελ. 48)

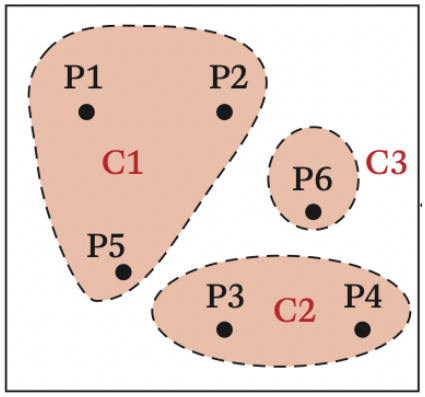
Παρατηρώντας αρχικά τον πίνακα, βλέπουμε πως δεν έχει υποστεί καθαρισμό, καθώς περιέχει διπλότυπες εγγραφές στις πλειάδες του, όπως για παράδειγμα το όνομα ή ο ταχυδρομικός κώδικας.

Οι εγγραφές αυτού του πίνακα μπορούν να αναπαραχθούν και σχηματικά, αναλόγως με το αν είναι διπλότυπες ή όχι. Συγκεκριμένα όπως φαίνεται και στο παρακάτω σχήμα (3.1), οι διπλές εγγραφές που φαίνονται να υπάρχουν μέσα στον πίνακα, ενώνονται στο σχήμα μεταξύ τους με γραμμές. Κάθε γραμμή έχει έναν αριθμό, που υποδηλώνει το μέτρο ομοιότητας μεταξύ των εγγραφών που ενώνει. Για παράδειγμα όπως φαίνεται και στο σχήμα, το μέτρο της ομοιότητας μεταξύ των εγγραφών P1 και P2 είναι 0.9. Για το συγκεκριμένο παράδειγμα ας ορίσουμε , πως το μέτρο ομοιότητας μπορεί να πάρει τιμές από 0 έως και 1. Όσο πιο κοντά είναι στο 1 το μέτρο ομοιότητας, τόσο περισσότερο οι εγγραφές αυτές μοιάζουν μεταξύ τους. Αντίθετα οι εγγραφές που δεν είναι διπλότυπες, δεν ενώνονται καθόλου μεταξύ τους.



Σχήμα 3.1 Μέτρο ομοιότητας
(Ilyas και Chu, 2019, σελ. 48)

Σύμφωνα με το παραπάνω σχήμα (3.1), οι εγγραφές που είναι διπλότυπες μπορούν να ομαδοποιηθούν και συναρτήσει του μέτρου ομοιότητας όπως αναφέρθηκε προηγουμένως και αυτό φαίνεται στο παρακάτω σχήμα 3.2. Πιο πριν είπαμε πως το μέτρο ομοιότητας μπορεί να πάρει τιμές από 0 έως και 1, τότε μπορούμε να ορίσουμε πως οι εγγραφές που έχουν μέτρο ομοιότητας μεγαλύτερο του 0.5, θα θεωρούνται διπλότυπες. Παρόλο που οι εγγραφές P1 και P5 έχουν τιμή 0.3 θεωρούνται διπλότυπες, γιατί συσχετίζονται και οι δύο με την εγγραφή P2, γι' αυτό και είναι μαζί σε ένα σύνολο (C1) στο παρακάτω σχήμα.



Σχήμα 3.2 Συστάδα παρόμοιων εγγραφών
(Ilyas και Chu, 2019, σελ. 48)

Τελικά από τις συστάδες που έχουν δημιουργηθεί από το παραπάνω σχήμα (3.2), δημιουργείται ένας νέος πίνακας καθαρισμένος, χωρίς διπλότυπες εγγραφές, όπως φαίνεται και παρακάτω στον πίνακα (3.2).

ID	Name	ZIP	Income
C1	Green	51519	39k
C2	Peter	30528	40k
C3	Chuck	51519	30k

Πίνακας 3.2 Καθαρισμένα δεδομένα χωρίς διπλότυπες εγγραφές
(Ilyas και Chu, 2019, σελ. 48)

3.2.1 Μετρικές Ομοιότητας

Προκειμένου να προσδιοριστεί αν δύο εγγραφές είναι διπλότυπες, είναι ζωτικής σημασίας να συγκριθεί η ομοιότητα μεταξύ δύο τιμών στην ίδια στήλη. Έχουν αναπτυχθεί διάφορες μετρικές ομοιότητας για την αντιμετώπιση διαφόρων τύπων λαθών και τύπων δεδομένων, όπως για παράδειγμα για αριθμητικά δεδομένα και για δεδομένα συμβολοσειράς. Για τα αριθμητικά δεδομένα για παράδειγμα είναι αυτονόητο ότι η ομοιότητα μεταξύ δύο αριθμητικών τιμών θα πρέπει να εξαρτάται άμεσα από το πόσο απέχουν μεταξύ τους. Για παράδειγμα, ο ακόλουθος ορισμός μπορεί να χρησιμοποιηθεί για να περιγράψει την ομοιότητα μεταξύ δύο αριθμητικών τιμών **v1** και **v2** για ένα χαρακτηριστικό **A**:

$$NumSim(v_1, v_2) = 1.0 - \frac{|v_1 - v_2|}{max(A) - min(A)} \quad (1)$$

Υπάρχουν πολυάριθμες επιλογές για τον καθορισμό ομοιοτήτων μεταξύ δύο τιμών συμβολοσειράς σε αντίθεση με τις αριθμητικές τιμές. Οι τρεις κατηγορίες μετρικών ομοιότητας που εξετάζουμε

παρακάτω αντιμετωπίζουν τρεις διαφορετικούς τύπους κοινών σφαλμάτων. Αυτές οι κατηγορίες περιλαμβάνουν μετρικές ομοιότητας που βασίζονται σε χαρακτήρες, σε σύμβολα και σε φωνητικά.

3.2.1.1 Μετρικές ομοιότητας με βάση το χαρακτήρα.

Τα τυπογραφικά λάθη είναι συχνά και μπορούν να αντιμετωπιστούν με τη χρήση διαφόρων μετρικών ομοιότητας με βάση τον χαρακτήρα. Το λιγότερο δαπανηρό σύνολο πράξεων επεξεργασίας που απαιτείται για την αλλαγή μιας συμβολοσειράς **s1**, σε μια άλλη, είναι γνωστό ως απόσταση επεξεργασίας μεταξύ των δύο συμβολοσειρών **s1** και **s2**. Η εισαγωγή ενός χαρακτήρα, η αφαίρεση ενός χαρακτήρα και η αντικατάσταση ενός χαρακτήρα με έναν άλλο είναι τα τρία είδη πράξεων επεξεργασίας που συνήθως λαμβάνονται υπόψη. Η απόσταση **Levenshtein** (Levenshtein, 1966), όπου το κόστος κάθε λειτουργίας θεωρείται μοναδιαίο κόστος, αναφέρεται γενικά ως "απόσταση επεξεργασίας".

Οι τρεις τύποι βασικών πράξεων επεξεργασίας που επιτρέπονται από την απόσταση επεξεργασίας είναι οι εξής:

- Εισαγωγή
- Διαγραφή
- Αντικατάσταση

Δεδομένου ότι χρειάζεται μόνο μία πράξη διαγραφής για να αλλάξει το "iPhone 6s" σε "iPhone 6", η απόσταση επεξεργασίας μεταξύ αυτών των δύο συμβολοσειρών είναι 1, υποδεικνύοντας ότι είναι συγκρίσιμες. Δεδομένου ότι το "Kim" πρέπει να αλλάξει σε "Jack" με τρεις αντικαταστάσεις και μία εισαγωγή, η απόσταση επεξεργασίας μεταξύ των δύο συμβολοσειρών είναι 4.

Οι βασικές αποστάσεις επεξεργασίας παρουσιάζουν μεγάλη ποικιλία και γι’ αυτό θα αναφερθούμε ενδεικτικά σε μερικές. Η απόσταση **Hamming**, για παράδειγμα, μπορεί να θεωρηθεί ως ένας τύπος απόστασης επεξεργασίας στον οποίο επιτρέπεται μόνο η πράξη αντικατάστασης. Ως αποτέλεσμα, μπορεί να οριστεί μόνο για δύο συμβολοσειρές ίδιου μήκους. Συγκεκριμένα, ο αριθμός των θέσεων στις οποίες οι αντίστοιχοι χαρακτήρες είναι διαφορετικοί είναι η απόσταση **Hamming** για δύο συμβολοσειρές ίδιου μήκους. Η πράξη της μετάθεσης, η οποία αλλάζει δύο γειτονικούς χαρακτήρες σε μια συμβολοσειρά, είναι μια πράξη επεξεργασίας που προστίθεται στους τρεις θεμελιώδεις τύπους πράξεων με την απόσταση **Levenshtein** (Bard, 2007). Μία άλλη μετρική βάσει του χαρακτήρα, είναι η απόσταση **Jaro–Winkler**, που επιτρέπει μόνο μεταθέσεις και είναι ειδικά σχεδιασμένη για την αντιστοίχιση ονομάτων (Winkler, 1990).

3.2.1.2 Μετρικές ομοιότητας με βάση τα σύμβολα.

Η ομοιότητα μεταξύ δύο συμβολοσειρών που χρησιμοποιούν το ίδιο σύνολο συμβόλων, αλλά με διαφορετική σειρά για παράδειγμα, "James Smith" έναντι "Smith James" συχνά διαφεύγει από τις μετρικές ομοιότητας, με βάση τους χαρακτήρες οι οποίες είναι χρήσιμες για την επίλυση τυπογραφικών σφαλμάτων. Για τη διαχείριση τέτοιων ανακριβειών, μπορούν να χρησιμοποιηθούν

διάφορα μέτρα ομοιότητας με βάση τα σύμβολα, όπως ο συντελεστής επικάλυψης, ο συντελεστής **Jaccard** και ο συντελεστής **Dice**.

3.2.1.3 Μετρικές ομοιότητας με βάση τη φωνητική.

Ακόμη και αν δύο συμβολοσειρές δεν είναι παρόμοιες όταν συγκρίνονται χρησιμοποιώντας μετρικές ομοιότητας με βάση τους χαρακτήρες ή τα σύμβολα για παράδειγμα "Clair" έναντι "Clare", οι μετρικές ομοιότητας με βάση τη φωνητική μπορούν να χρησιμοποιηθούν για να καθορίσουν πόσο παρόμοιες είναι.

Το πιο γνωστό σύστημα φωνητικής κωδικοποίησης, το Soundex (Russell, 1918 όπως αναφέρεται στους Ilyas και Chu, 2019) δημιουργήθηκε για τη φωνητική κωδικοποίηση αγγλικών επωνύμων για σκοπούς απογραφής. Τα βήματα που χρησιμοποιεί για να λειτουργήσει είναι τα εξής: Πρώτον, διατηρεί το πρώτο γράμμα μιας συμβολοσειράς και μετατρέπει τα υπόλοιπα γράμματα σε αριθμούς σύμφωνα με τον πίνακα (*3.3*), όπως φαίνεται παρακάτω. Στη συνέχεια, αφαιρούνται όλα τα μηδενικά (η πρώτη γραμμή του πίνακα 3.3) και οι ακολουθίες του ίδιου αριθμού μειώνονται σε έναν μόνο αριθμό (για παράδειγμα, το "222" αντικαθίσταται με το "2"). Τέλος, το αρχικό πρώτο γράμμα και οι επόμενοι τρεις αριθμοί αποτελούν την τελική συμβολοσειρά κωδικοποίησης. Η ακολουθία του αριθμού τερματίζεται εάν είναι μεγαλύτερη από τρεις, εάν είναι μικρότερη από τρεις συμπληρώνεται με μηδενικά.

a,e,h,i,o,u,w,y	0
b,f,p,v	1
c,g,j,k,q,s,x,z	2
d,t	3
l	4
m,n	5
r	6

Πίνακας 3.3 Πίνακας μετατροπής Soundex
(Ilyas και Chu, 2019, σελ. 54)

Για παράδειγμα χρησιμοποιώντας το σύστημα φωνητικής κωδικοποίησης για το όνομα Peter το όνομα κωδικοποιείται ως εξής: Αρχικά το γράμμα "P" διατηρείται και τα υπόλοιπα γράμματα μετατρέπονται σε αριθμούς "0306" σύμφωνα με τον πίνακα (*3.3*), στη συνέχεια τα 0 στο "0306" αφαιρούνται για να προκύψει το "36" και το πρώτο γράμμα "P" ενώνεται με το "36" και συμπληρώνεται με ένα επιπλέον 0 για να προκύψει η τελική κωδικοποίηση "P360".

Τόσο το "**Paul**" όσο και το "**Pual**", σύμφωνα με το **Soundex**, μοιράζονται την ίδια κωδικοποίηση Soundex "**P400**" και, επομένως, θεωρείται ότι έχουν το ίδιο όνομα.

Το σύστημα κωδικοποίησης **Soundex** έχει διάφορες μορφές. Το **Phonix** (Gadd, 1990) είναι μια έκδοση του **Soundex** που επεξεργάζεται εκ των προτέρων τα ονόματα σύμφωνα με περισσότερους από **100** κανόνες, όπως η αντικατάσταση των "**kn**" και "**wr**" με "**n**" και "**r**", για να προσπαθήσει να βελτιώσει την ποιότητα της κωδικοποίησης. Η διαδικασία κωδικοποίησης **Soundex** χρησιμοποιείται στη συνέχεια για την κωδικοποίηση της τροποποιημένης συμβολοσειράς. Το **Soundex** βελτιώθηκε με το **Daitch-Mokotoff Soundex (D-M Soundex**) (Steuart, 1994) για την καλύτερη αντιστοίχιση επωνύμων με σλαβικές και γερμανικές ρίζες.

3.2.2 Πρόβλεψη διπλών ζευγών

Σε αυτό το κομμάτι, θα εξετάσουμε πώς να προσδιορίσουμε την πιθανότητα ενός ζεύγους πλειάδων να είναι διπλότυπα με βάση μία ή περισσότερες βαθμολογίες ομοιότητας. Το αποτέλεσμα της πρόβλεψης μπορεί να είναι ένας πραγματικός αριθμός μεταξύ 0 και 1, ο οποίος υποδεικνύει την πιθανότητα ένα ζεύγος πλειάδων να είναι διπλότυπο, ή μια δυαδική μεταβλητή, η οποία δείχνει αν ένα ζεύγος πλειάδων είναι διπλότυπο ή όχι. Για την πρόβλεψη διπλών ζευγών μπορούν να χρησιμοποιηθούν τόσο εποπτευόμενες όσο και μη εποπτευόμενες μέθοδοι.

Χωρίς ένα σύνολο δεδομένων εκπαίδευσης, οι αλγόριθμοι χωρίς επίβλεψη επιλέγουν ζευγάρια που ταιριάζουν. Παραδείγματα περιλαμβάνουν την εφαρμογή ενός προκαθορισμένου κατωφλίου σε μια συνάρτηση απόστασης (Monge και Elkan, 1996, Chaudhuri κ.ά., 2005) και τη χρήση κατευθυντήριων γραμμών για συγκεκριμένο κλάδο (Hernandez και Stolfo, 1998, Doan κ.ά., 2003, Weis κ.ά., 2008), όπως: Εάν το όνομα και το επώνυμο δύο ατόμων είναι παρόμοια, τότε οι δύο εγγραφές είναι αντίγραφα.

Ένα σύνολο δεδομένων εκπαίδευσης που περιέχει ζεύγη εγγραφών με ετικέτες που προσδιορίζουν αντιστοιχίες ή αναντιστοιχίες χρησιμεύει ως βάση για τις προσεγγίσεις μάθησης με επίβλεψη. Όταν ένας ταξινομητής εκπαιδεύεται για να χρησιμοποιηθεί στα υπόλοιπα δεδομένα, οι βαθμολογίες ομοιότητας μεταξύ ζευγών εγγραφών χρησιμοποιούνται ως χαρακτηριστικά. Ο Naive Bayes (Winkler, 1999), τα δέντρα απόφασης (Chaudhuri κ.ά., 2007) και οι support vector machines (SVM) (Bilenko και Mooney, 2003) είναι μερικά παραδείγματα μοντέλων ταξινομητών. Η πρώτη επιβλεπόμενη πιθανολογική μέθοδος για την κατάργηση διπλότυπων δεδομένων περιγράφεται στις παραγράφους που ακολουθούν. Η πρώτη φορά που προσδιορίστηκε ένα πρόβλημα Μπεϋζιανής συμπερασματολογίας (Bayesian inference) ως πρόβλημα εντοπισμού διπλών εγγραφών ήταν από τους Newcombe κ.ά., το 1959. Αργότερα, η ιδέα κωδικοποιήθηκε από τους Fellegi και Sunter (1969).

Έστω A και B για τους πίνακες που θέλουμε να αντιστοιχίσουμε (A και B μπορεί να είναι ο ίδιος πίνακας). Κάθε ζεύγος εγγραφών $\langle \alpha, \beta \rangle (\alpha \in A, \beta \in B)$ θα τοποθετείται σε μία από τις δύο κλάσεις (M ή U), με την κλάση M να περιέχει ζεύγη εγγραφών που αντιπροσωπεύουν τις ίδιες οντότητες του πραγματικού κόσμου και την κλάση U να περιέχει ζεύγη εγγραφών που αντιπροσωπεύουν άλλες οντότητες του πραγματικού κόσμου. Η ομοιότητα μεταξύ ενός ζεύγους εγγραφών σε διάφορες διαστάσεις αντιπροσωπεύεται από ένα διάνυσμα σύγκρισης $\gamma = [\gamma 1, \gamma 2, \ldots, \gamma k]$, το οποίο υπολογίζεται. Το διάνυσμα σύγκρισης χρησιμοποιείται για να καθοριστεί αν ένα ζεύγος εγγραφών είναι ταιριαστό ή μη ταιριαστό. Ο ακόλουθος είναι ο απλούστερος κανόνας απόφασης που βασίζεται στην απλή πιθανότητα:

$$\langle \alpha, \beta \rangle = \begin{cases} M & \text{if } p(M|\gamma) \geq p(U|\gamma) \\ U & \text{otherwise.} \end{cases}$$

Αυτός ο κανόνας απόφασης δηλώνει ότι, δεδομένου του διανύσματος σύγκρισης γ για ένα ζεύγος εγγραφών ⟨α, β⟩, εάν η πιθανότητα της κλάσης M είναι μεγαλύτερη ή ίση με την πιθανότητα της κλάσης U, τότε ταξινομεί τα ⟨α, β⟩ ως αντιστοιχισμένο ζεύγος. Χρησιμοποιώντας το θεώρημα του Bayes, οι παραπάνω κανόνες απόφασης μπορούν να αναδιατυπωθούν ως εξής:

$$\langle \alpha, \beta \rangle = \begin{cases} M & \text{if } \frac{p(\gamma|M)}{p(\gamma|U)} \geq \frac{p(U)}{p(M)} \\ U & \text{otherwise.} \end{cases}$$

Ο λόγος πιθανότητας συμβολίζεται με τον λόγο p(γ |M) και ο λόγος p(U) είναι η τιμή κατωφλίου του λόγου πιθανότητας, p(γ|U) p(M), για τον κανόνα απόφασης. Εάν οι κατανομές των p(γ|M) και p(γ|U) και οι εκ των προτέρων p(U) και p(M) είναι γνωστές (Friedman κ.ά., 2001), πράγμα που δεν είναι συνηθισμένο στην πράξη, μπορεί να αποδειχθεί ότι ο προαναφερόμενος κανόνας απόφασης θα παράγει το μικρότερο σφάλμα.

Η παραδοχή εξαρτημένης ανεξαρτησίας μεταξύ των διαστάσεων στο διάνυσμα σύγκρισης είναι μια κοινή μέθοδος για τον εξορθολογισμό του υπολογισμού των δύο κατανομών, p(γ|M) και p(γ|U). Εάν υπάρχουν διαθέσιμα δεδομένα εκπαίδευσης, δηλαδή ζεύγη εγγραφών με ετικέτες M ή U, μπορούμε εύκολα να εκτιμήσουμε τα p(U), p(M), p(γi|M) και p(γi|U). Οι αλγόριθμοι μεγιστοποίησης της προσδοκίας χρησιμοποιούνται συνήθως για την εκτίμηση των παραμέτρων του μοντέλου Bayes όταν δεν υπάρχουν επαρκή ή καθόλου διαθέσιμα δεδομένα εκπαίδευσης (Dempster κ.ά., 1977, Winkler, 1993). Μόνο στην περίπτωση που οι παράμετροι της κατανομής είναι γνωστές, οι κανόνες απόφασης Bayes παράγουν τα καλύτερα αποτελέσματα. Ωστόσο, στην πραγματικότητα, οι εκτιμώμενες παράμετροι είναι συχνά ατελείς, οδηγώντας σε μεγάλη συχνότητα σφαλμάτων ταξινόμησης.

3.3 Μετασχηματισμός Δεδομένων

Ο μετασχηματισμός δεδομένων είναι η διαδικασία μετατροπής δεδομένων από μια μορφή σε μια άλλη. Για παράδειγμα, η εισαγωγή ενός "-" μεταξύ των ψηφίων ενός τηλεφωνικού αριθμού θα το μετατρέψει σε τυποποιημένη μορφή. Οι μετασχηματισμοί δεδομένων, οι οποίοι χρησιμοποιούνται σε διάφορες φάσεις της ανάλυσης δεδομένων, μπορούν να θεωρηθούν ως ενέργειες που διορθώνουν τα σφάλματα. Για παράδειγμα, οι μετασχηματισμοί χρησιμοποιούνται συχνά για την τυποποίηση μορφών δεδομένων, την επιβολή τυποποιημένων προτύπων ή τη συντόμευση μεγάλων συμβολοσειρών πριν από την εκτέλεση ενός έργου ολοκλήρωσης δεδομένων.

Ένα πρόσθετο του Excel που ονομάζεται Transform-Data-by-Example καθιστά απλό τον προσδιορισμό του επιθυμητού μετασχηματισμού για μια συγκεκριμένη εργασία μετασχηματισμού. Το Transform-Data-by-Example εντοπίζει αυτόματα τις σχετικές λειτουργίες μετασχηματισμού δεδομένων από μια μεγάλη συλλογή που έχει ήδη ευρετηριάσει, όταν απαιτούνται μόνο μερικά δείγματα της επιθυμητής εξόδου. Για τη σύνταξη αυτής της συλλογής χρησιμοποιήθηκαν αποσπάσματα κώδικα του Stackoverflow, πηγαίοι κώδικες του Github και άλλες πηγές και

βιβλιοθήκες .Net. Συμβάλλοντας τις δικές τους συναρτήσεις μετασχηματισμού, οι χρήστες μπορούν επιπλέον να επεκτείνουν τη βιβλιοθήκη ώστε να συμπεριλάβει δυνατότητες μετασχηματισμού για συγκεκριμένο τομέα.

Μπορεί να υπάρχει ένας ατελείωτος αριθμός χρήσιμων προγραμμάτων μετασχηματισμού, επειδή είναι συχνά αδύνατο να εκπαιδευτούν ή να αναπτυχθούν ακριβείς λύσεις μετασχηματισμού χρησιμοποιώντας την αλήθεια του εδάφους. Ακόμη και αν η επιθυμητή μεταβολή μπορεί να είναι προφανής σε έναν άνθρωπο εμπειρογνώμονα, η διερεύνηση κάθε δυνατού αποτελέσματος είναι δαπανηρή. Προκειμένου να μειωθεί το κόστος, οι πρακτικές λύσεις μετασχηματισμού δεδομένων πρέπει να κλαδέψουν με επιτυχία τον χώρο διαδραστικά.

Τόσο ο εντοπισμός διπλότυπων όσο και ο εντοπισμός ακραίων τιμών απαιτούν τα δεδομένα εισόδου να έχουν τη "σωστή μορφή". Για παράδειγμα, προκειμένου να εντοπιστούν οι ακραίες τιμές, ένας κατάλογος θερμοκρασιών σε διάφορες μονάδες (όπως Κελσίου ή Φαρενάιτ) πρέπει να μετατραπεί στην ίδια μονάδα, και προκειμένου να εντοπιστούν τα διπλά δεδομένα, οι ιδιότητες των διαφόρων εγγραφών πρέπει να ευθυγραμμιστούν κατάλληλα. Η διαδικασία μετατροπής δεδομένων από μια μορφή ή δομή σε μια άλλη με τη χρήση προγραμμάτων, κανόνων ή σεναρίων που ορίζονται από τον χρήστη αναφέρεται ως μετασχηματισμός δεδομένων. Οι μετασχηματισμοί χρησιμοποιούνται σε μια σειρά από δραστηριότητες και σε διαφορετικές φάσεις του κύκλου ζωής του ETL, εκτός από την ανίχνευση ακραίων τιμών και την απαλοιφή διπλότυπων δεδομένων.

Για παράδειγμα, μετασχηματισμοί χρησιμοποιούνται συχνά για την τυποποίηση μορφών δεδομένων, την επιβολή τυποποιημένων προτύπων ή τη σύντμηση μεγάλων συμβολοσειρών πριν από την εκτέλεση ενός έργου ολοκλήρωσης δεδομένων. Κατά την ολοκλήρωση της διαδικασίας ETL, οι μετασχηματισμοί χρησιμοποιούνται επίσης, για παράδειγμα, για να συνδυάσουν ομάδες διπλών εγγραφών, να προσδιορίσουν μια μοναδική αναπαράσταση για μια ομάδα εγγραφών (μερικές φορές γνωστή ως "χρυσή εγγραφή") ή να καταστήσουν τα δεδομένα έτοιμα για κατανάλωση από λογισμικό ανάλυσης. Δεδομένου ότι μπορεί να χρησιμοποιηθεί για να "μετασχηματίσει" ανακριβή δεδομένα, ο μετασχηματισμός δεδομένων μπορεί επίσης να θεωρηθεί ως μια μέθοδος για την επισκευή δεδομένων.

Οι εργασίες μετασχηματισμού δεδομένων μπορούν να χωριστούν σε δύο κατηγορίες: συντακτικοί μετασχηματισμοί (Singh και Gulwani, 2012, Kandel κ.ά. 2011, Gulwani, 2011, Jin κ.ά., 2017) και σημασιολογικοί μετασχηματισμοί (Abedjan κ.ά., 2016). Οι συντακτικοί μετασχηματισμοί προσπαθούν να αλλάξουν τη συντακτική μορφή ενός πίνακα, συχνά χωρίς την ανάγκη εξωτερικής γνώσης ή υλικού αναφοράς. Παραδείγματα συντακτικών μετασχηματισμών περιλαμβάνουν τη μετατροπή τηλεφωνικών αριθμών σε μια τυποποιημένη μορφή, την ένωση δύο στηλών με ονόματα και επώνυμα για την παραγωγή μιας ενιαίας στήλης ονόματος και την αλλαγή του σχεδιασμού ενός πίνακα για να γίνει πιο ευανάγνωστος. Προκειμένου να κατανοηθεί το νόημα/η σημασιολογία ή η τυπική εφαρμογή των δεδομένων, οι σημασιολογικοί μετασχηματισμοί συνήθως απαιτούν πρόσβαση σε άλλες πηγές δεδομένων. Για παράδειγμα, η αλλαγή του "SIGMOD" σε "Special Interest Group on Management of Data" από τη συντομογραφία του θα απαιτούσε πρόσβαση σε μια εξωτερική πηγή δεδομένων που περιέχει τόσο τις συντομογραφίες όσο και τους πλήρεις τίτλους των συνεδρίων.

3.3.1 Συντακτικοί μετασχηματισμοί δεδομένων

Σύμφωνα με τους Ilyas και Chu (2019), ένα σύστημα συντακτικού μετασχηματισμού δεδομένων έχει τρία κύρια μέρη: γλώσσα, συγγραφή και εκτέλεση. Ένα σύνολο δεδομένων μπορεί να μετασχηματιστεί με διάφορους τρόπους, επομένως οι λύσεις συντακτικού μετασχηματισμού δεδομένων χρησιμοποιούν συχνά μια γλώσσα μετασχηματισμού για να περιορίσουν το εύρος των πιθανών μετασχηματισμών. Η γλώσσα μπορεί να περιλαμβάνει ένα περιορισμένο σύνολο λειτουργιών που μπορούν να εκτελεστούν σε έναν πίνακα (Raman και Hellerstein, 2001- Kandel κ.ά., 2011), όπως η διάσπαση και η συγχώνευση στηλών, ή μπορεί να περιλαμβάνει ένα σύνολο λειτουργιών για τον χειρισμό συμβολοσειρών (Gulwani, 2011), όπως η εξαγωγή μιας υποσυμβολοσειράς και η συνένωση δύο υποσυμβολοσειρών. Μια γλώσσα πρέπει να είναι αρκετά εκφραστική και περιορισμένη ώστε να επιτρέπει την αποτελεσματική και απλή συγγραφή μετασχηματισμών, ενώ παράλληλα να αποτυπώνει πολυάριθμες λειτουργίες μετασχηματισμών του πραγματικού κόσμου. Τα εργαλεία συντακτικού μετασχηματισμού διαθέτουν διάφορα μοντέλα αλληλεπίδρασης με τον χρήστη ανάλογα με την εκάστοτε εργασία για τη συγγραφή ενός προγράμματος μετασχηματισμού στην επιλεγμένη γλώσσα. Οι τρεις κύριοι τύποι μοντέλων αλληλεπίδρασης είναι τα δηλωτικά (declarative), τα καθοδηγούμενα από παραδείγματα (example-driven) και τα προορατικά (proactive). Οι χρήστες εκφράζουν ρητά τους μετασχηματισμούς στο δηλωτικό μοντέλο, συνήθως χρησιμοποιώντας μια οπτική διεπαφή. Οι χρήστες καλούνται να δώσουν μερικά παραδείγματα εισόδου-εξόδου στο μοντέλο που καθοδηγείται από παραδείγματα, έτσι ώστε το εργαλείο να μπορεί να συμπεράνει αυτόματα τους πιθανούς μετασχηματισμούς που αντιστοιχούν στα παραδείγματα. Στο προορατικό μοντέλο αλληλεπίδρασης, το εργαλείο μετασχηματισμού κάνει προτάσεις για πιθανούς μετασχηματισμούς δεδομένων και ακόμη και προτάσεις σχετικά με το ποια δεδομένα πρέπει να μετασχηματιστούν. Τέλος, το σύνολο δεδομένων υποβάλλεται στους δεδομένους μετασχηματισμούς κατά την εκτέλεση του μετασχηματισμού. Δεδομένου ότι το προηγούμενο βήμα μπορεί να έχει καθορίσει πολλαπλούς μετασχηματισμούς, τα εργαλεία μετασχηματισμού συνήθως βοηθούν στην επιλογή του επιθυμητού μετασχηματισμού. Για παράδειγμα, μπορεί να δείχνουν τον άμεσο αντίκτυπο του καθορισμένου μετασχηματισμού σε δεδομένα δείγματος ή να προσφέρουν ερμηνείες των καθορισμένων μετασχηματισμών. Σε αυτές τις τρεις διαστάσεις, τα διάφορα εργαλεία μετασχηματισμού προσφέρουν διαφορετικές δυνατότητες. Παραδείγματα συστημάτων συντακτικού μετασχηματισμού είναι το Data Wrangler (Kandel κ.ά., 2011, Guo κ.ά., 2011, Heer κ.ά., 2015) το οποίο βασίζεται στο Potter's Wheel (Raman & Hellerstein, 2001) και είναι κατασκευασμένο για τον μετασχηματισμό δεδομένων σε πίνακες, το QuickCode (Gulwani 2011) το οποίο είναι κατασκευασμένο για τον χειρισμό συμβολοσειρών και το KNIME[1] το οποίο είναι ένα εργαλείο συγγραφής ροών εργασίας ανοικτού κώδικα που μπορεί να συνθέτει δηλωτικά προγράμματα μετασχηματισμού δεδομένων.

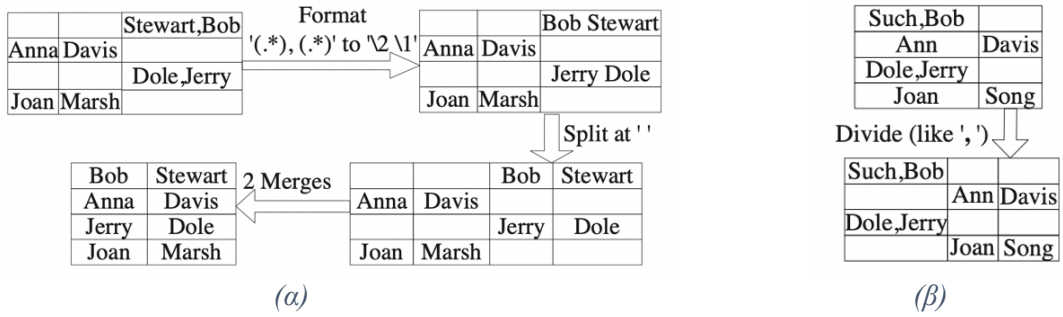
[1] https://www.knime.com/

3.3.1.1 Γλώσσα μετασχηματισμού

Η συλλογή των επιτρεπόμενων τελεστών για ένα εργαλείο συντακτικού μετασχηματισμού ορίζεται από μια γλώσσα μετασχηματισμού. Ακολουθεί μια συζήτηση σχετικά με τις γλώσσες που χρησιμοποιούν το QuickCode και το Data Wrangler.

Για την υποστήριξη κοινών εργασιών χειρισμού δεδομένων, όπως η διάσπαση ή η εξαγωγή τιμών από συμβολοσειρές, η διαγραφή ή η συγχώνευση στηλών, η παρεμβολή τιμών και η αναδιαμόρφωση πινάκων με αναδίπλωση και αναδίπλωση, το Data Wrangler (Kandel κ.ά., 2011, Guo κ.ά., 2011, Heer κ.ά., 2015) υιοθετεί μια γλώσσα που αποτελείται από μια ακολουθία επιμέρους τελεστών. Η αρχιτεκτονική της γλώσσας Data Wrangler είναι μικρή, επιτρέποντας στους αναλυτές να εκτελούν ένα ευρύ φάσμα λειτουργιών μετασχηματισμού δεδομένων με μόλις δώδεκα τελεστές. Έχει διαπιστωθεί ότι η γλώσσα μπορεί να εκφράσει οποιουσδήποτε μετασχηματισμούς ένα προς ένα ή ένα προς πολλά των τιμών των κελιών (Raman και Hellerstein, 2001), επειδή βασίζεται στην SchemaSQL (Lakshmanan κ.ά., 2001). Ο δηλωτικός σχεδιασμός της γλώσσας μετασχηματισμού δεδομένων καθιστά ευκολότερη την υλοποίησή της σε διάφορες πλατφόρμες, για παράδειγμα με την παραγωγή εκτελέσιμου κώδικα για Python και JavaScript, μεταξύ άλλων χρόνων εκτέλεσης. Οι χρήστες χρησιμοποιούν συχνά τη διεπαφή για να "χειριστουν" (“wrangle”) ένα υποσύνολο των δεδομένων πριν εξάγουν το προκύπτον σενάριο Python για να αλλάξουν το πολύ ευρύτερο σύνολο δεδομένων σε έναν διακομιστή.

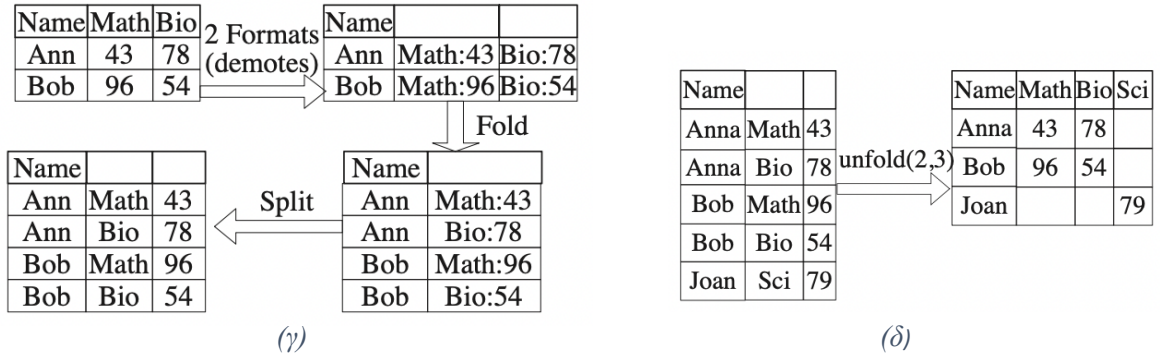
Τιμές του ίδιου είδους μπορεί να υπάρχουν σε πολλαπλές μορφές κατά την ενσωμάτωση δεδομένων από διάφορες πηγές. Μια απεικόνιση του τρόπου διόρθωσης των ασυνεπειών μορφής ονομάτων με τη χρήση των τελεστών Format, Split και Merge παρουσιάζεται στο σχήμα 3.3(α). Ο τελεστής Divide μπορεί να χρησιμοποιηθεί για το διαχωρισμό διαφορετικών μορφών ονομάτων, όπως φαίνεται στην σχήμα 3.3 (β), και τα δεδομένα του πίνακα μπορούν στη συνέχεια να τροποποιηθούν ώστε να μοιάζουν με το σχήμα 3.3 ως αποτέλεσμα (α).



Σχήμα 3.3 Παραδείγματα μετασχηματισμών στο Data Wrangler.
(Raman και Hellerstein, 2001, σελ. 8)

Όταν οι πληροφορίες διατηρούνται εν μέρει στο σχήμα και εν μέρει ως τιμές δεδομένων, οι αντιστοιχίσεις πολλών προς πολλές γραμμές, όπως το Fold και το Unfold, είναι χρήσιμες για την αντιμετώπιση της ετερογένειας των σχημάτων υψηλότερης τάξης. Ο τελεστής Folding επιδεικνύεται στο σχήμα 3.4 (γ), όπου τα ονόματα των θεμάτων υποβιβάζονται πρώτα σε γραμμές χρησιμοποιώντας τον τελεστή Format, ακολουθούμενο από την αναδίπλωση των δύο τελευταίων στηλών, ενώ η πρώτη στήλη αναπαράγεται. Τέλος, το θέμα και ο βαθμός διαχωρίζονται

χρησιμοποιώντας τον τελεστή Split. Αντίθετα, ο τελεστής Unfold ξεδιπλώνει τον πίνακα επιλέγοντας δύο στήλες, συγκεντρώνοντας γραμμές με ίδιες τιμές σε όλες τις άλλες στήλες και στη συνέχεια ξεδιπλώνοντας τις δύο επιλεγμένες στήλες. Για την ευθυγράμμιση των τιμών στην άλλη επιλεγμένη στήλη, οι τιμές σε μία από τις επιλεγμένες στήλες χρησιμοποιούνται ως ονόματα στήλης. Στο σχήμα 3.4 (δ) παρουσιάζεται ένα παράδειγμα ξεδιπλώματος της δεύτερης και της τρίτης στήλης:



Σχήμα 3.4 Παραδείγματα μετασχηματισμών στο Data Wrangler.
(Raman και Hellerstein, 2001, σελ. 9

Σε αντίθεση με το Data Wrangler, το οποίο επιδιώκει να προσφέρει ένα σύνολο τελεστών μετασχηματισμού για την εργασία με δεδομένα σε πίνακες, το QuickCode (Gulwani, 2011) προσφέρει μια γλώσσα μετασχηματισμού που στοχεύει στην εργασία με συμβολοσειρές. Το QuickCode, μια γλώσσα επεξεργασίας συμβολοσειρών ειδικού τομέα, επιτρέπει την αντιστοίχιση κανονικών εκφράσεων, περιορισμένες συνθήκες και βρόχους, καθώς και συντακτικές πράξεις συμβολοσειρών όπως η υποσειρά και η συνένωση.

3.3.2 Σημασιολογικοί μετασχηματισμοί δεδομένων

Η κατανόηση του νοήματος/της σημασιολογίας ή της κανονικής χρήσης των δεδομένων απαιτείται συνήθως για σημασιολογικούς μετασχηματισμούς και όχι απλώς για την εξέταση των δεδομένων ως μια σειρά χαρακτήρων. Οι σημασιολογικοί μετασχηματισμοί, σε αντίθεση με τους συντακτικούς μετασχηματισμούς, δεν μπορούν να ολοκληρωθούν με την απλή εξέταση των τιμών εισόδου- αντίθετα, συνήθως απαιτούν τη χρήση εξωτερικών πηγών δεδομένων για τον εντοπισμό των τιμών εισόδου που πρέπει να τροποποιηθούν. Η πλειονότητα των σημασιολογικών αλλαγών έχει πραγματοποιηθεί με τη χρήση τεχνικών που βασίζονται σε παραδείγματα (Arasu κ.ά., 2009, Singh & Gulwani, 2012, Abedjan κ.ά., 2016b, Jin κ.ά., 2017). Χρησιμοποιώντας το DataXFormer (Abedjan κ.ά., 2016b), το οποίο χρησιμοποιεί εξωτερικούς διαδικτυακούς πίνακες και διαδικτυακές φόρμες για την αντιστοίχιση παραδειγμάτων που παρέχει ο χρήστης. Η διαδικασία ανταλλαγής δεδομένων (Fagin κ.ά., 2005α), η οποία μετατρέπει δομημένα δεδομένα βάσει ενός σχήματος πηγής

σε δομημένα δεδομένα βάσει ενός σχήματος στόχου με βάση σημασιολογικές αντιστοιχίσεις μεταξύ των δύο σχημάτων, μπορεί επίσης να θεωρηθεί ως σημασιολογικός μετασχηματισμός.

3.3.2.1 Σημασιολογικοί μετασχηματισμοί με παράδειγμα

Το DataXFormer (Abejdan κ.ά., 2016β) είναι μια τεχνική σημασιολογικού μετασχηματισμού με παράδειγμα το οποίο βασίζεται σε δύο εξωτερικές πηγές δεδομένων, δηλαδή τοπικά αποθηκευμένους στατικούς πίνακες ιστού και δυναμικά ανακαλυπτόμενες φόρμες ιστού. Ο χρήστης ξεκινά στέλνοντας ένα ερώτημα μετασχηματισμού με μερικά δείγματα εισόδου-εξόδου, όπως το ερώτημα που μετατρέπει τους κωδικούς αεροδρομίων σε ονόματα πόλεων. Για να εξαγάγει πιθανούς πίνακες ιστού και πιθανές φόρμες ιστού, το DataXFormer μετατρέπει το ερώτημα του χρήστη σε μια εσωτερική φόρμα. Το κομμάτι υπεύθυνο για την ανάκτηση φορμών ψάχνει σε ολόκληρο τον ιστό για σχετικές διαδικτυακές φόρμες, σε αντίθεση με το κομμάτι υπεύθυνο για την ανάκτηση πινάκων, το οποίο λειτουργεί πάνω σε ένα τοπικό αποθετήριο. Ο χρήστης αναμένεται να δώσει στο DataXFormer ένα σύνολο X με n τιμές και ένα σύνολο E με m ζεύγη παραδειγμάτων, όπου $E=\{(xi,yi)|xi \in X, 1\leq i\leq m\}$. Η εύρεση των τιμών που λείπουν Y είναι ο στόχος και η λύση δίνεται από την $F,F=\{(xi,yi)|xi \in X, 1\leq i\leq n\}$.

Η τεχνική ανάκτησης πινάκων ανακτά έναν κατάλογο σχετικών υποψήφιων πινάκων για το καθορισμένο ερώτημα του χρήστη χρησιμοποιώντας ένα ανεστραμμένο ευρετήριο. Το συστατικό βελτίωσης εξετάζει περαιτέρω τα υποψήφια αποτελέσματα επιβεβαιώνοντας την κάλυψη των υποψηφίων πινάκων στα παραδείγματα του ερωτήματος. Το DataXFormer εξάγει τις υπόλοιπες αναγκαίες τιμές μετασχηματισμού εάν η κάλυψη είναι μεγαλύτερη από το όριο που ορίζει ο χρήστης. Το DataXFormer χρησιμοποιεί μια μηχανή αναζήτησης ιστού για την εύρεση σχετικών διευθύνσεων URL όταν ασχολείται με φόρμες ιστού. Το DataXFormer ανακαλύπτει πιθανές φόρμες ιστού εξετάζοντας τα αποτελέσματα της αναζήτησης ιστού. Στη συνέχεια, το DataXFormer δημιουργεί ένα "περιτύλιγμα" για κάθε δυνητική φόρμα, ώστε να μπορεί να υποβάλει ερώτημα στη φόρμα και να ανακτήσει τις τιμές μετασχηματισμού. Στη συνέχεια, δεδομένων των δειγμάτων που παρέχονται από το ερώτημα του χρήστη, γίνεται αναζήτηση στις υποψήφιες φόρμες ιστού. Οι υπόλοιπες τιμές εισόδου χρησιμοποιούνται στη συνέχεια για την ενεργοποίηση εκείνων που έχουν επαρκή κάλυψη. Η συνιστώσα ενοποίησης λύσεων εξάγει στη συνέχεια τον επιθυμητό μετασχηματισμό μετά την ενοποίηση και την κατάταξη των διαφόρων λύσεων μετασχηματισμού για ένα συγκεκριμένο ερώτημα, όπως αυτές λαμβάνονται από τα δύο υποσυστήματα.

3.4 Καθαρισμός δεδομένων βάσει κανόνων

Η επιβολή κανονισμών για την ποιότητα των δεδομένων, οι οποίοι μερικές φορές αντιπροσωπεύονται ως απαιτήσεις ακεραιότητας (integrity constraints), είναι μια άλλη τυπική μέθοδος εντοπισμού και επιδιόρθωσης λαθών στις βάσεις δεδομένων (ΒΔ) (Chu κ.ά., 2013b, Kolahi & Lakshmanan, 2009, Fan & Geerts, 2012). "Για δύο υπαλλήλους που εργάζονται στο ίδιο υποκατάστημα μιας εταιρείας, ο ανώτερος υπάλληλος δεν μπορεί να έχει λιγότερο χρόνο διακοπών από τον κατώτερο υπάλληλο" είναι ένα παράδειγμα ποιότητας δεδομένων. Εάν υποθέσουμε λοιπόν ότι έχουμε μια "βρώμικη" συλλογή δεδομένων *I* με schema *R* και ένα σύνολο από ICs, οι τεχνικές

που βασίζονται σε κανόνες καθαρίζουν την *I* σε δύο βασικά βήματα: πρώτο, την ανίχνευση παραβιάσεων, η οποία βρίσκει λάθη σε βρώμικες βάσεις δεδομένων εντοπίζοντας παραβιάσεις στο σύνολο δεδομένων σε σχέση με τα προκαθορισμένα πρότυπα ποιότητας δεδομένων, και δεύτερον, την επιδιόρθωση σφαλμάτων, η οποία ενημερώνει τα δεδομένα εισόδου για να διορθώσει τις παραβιάσεις.

Τα πρότυπα ποιότητας δεδομένων μπορούν είτε να δημιουργηθούν χειροκίνητα από εμπειρογνώμονες που είναι εξοικειωμένοι με τη σημασιολογία των δεδομένων, είτε να εξορύσσονται αυτόματα από τα δεδομένα δεδομένου ενός συνόλου δεδομένων (Chu κ.ά., 2013a, Chiang and Miller, 2008). Σε αυτό το πλαίσιο, η επιδιόρθωση σφαλμάτων είναι η διαδικασία τροποποίησης της βάσης δεδομένων έτσι ώστε οι παραβιάσεις να διορθωθούν και τα νέα δεδομένα να συμμορφώνονται με τα παρεχόμενες απαιτήσεις ακεραιότητας. Η (ποιοτική) ανίχνευση σφαλμάτων είναι η διαδικασία ανακάλυψης των παραβιάσεων των ICs, δηλαδή υποσύνολα εγγραφών που δεν μπορούν να συνυπάρχουν.

Οι τεχνικές καθαρισμού δεδομένων βάσει κανόνων με τη χρήση περιορισμών άρνησης (Chu κ.ά., 2013a, Chu κ.ά., 2013b) έχουν αποδειχθεί εξαιρετικά χρήσιμες στον εντοπισμό και την επιδιόρθωση σφαλμάτων σε πραγματικά δεδομένα. Ένα τέτοιο παράδειγμα είναι η περίπτωση που μελέτησαν οι Tetje κ.ά. (2018) σε συνεργασία με επιστήμονες ζώων στο UC Berkeley, οι οποίοι ερευνούν τις επιπτώσεις της κοπής καυσόξυλων σε μικρά χερσαία σπονδυλωτά. Κάθε καταχώρηση στο σύνολο δεδομένων περιλαμβάνει λεπτομέρειες σχετικά με την αλίευση ενός συγκεκριμένου ζώου, όπως η ημερομηνία και ο τόπος αλίευσης, η ταυτότητα του ζώου με την ετικέτα του και τα χαρακτηριστικά του ζώου κατά τη στιγμή της αλίευσης, συμπεριλαμβανομένου του βάρους, του είδους, του φύλου και της ηλικιακής ομάδας. Κατά τη διάρκεια των προηγούμενων 20 ετών συγκεντρώθηκε το σύνολο των δεδομένων. Κάθε σύλληψη καταχωρήθηκε αρχικά σε ένα φύλλο Excel ως μεταγραφή από ένα ημερολόγιο, επομένως το σύνολο δεδομένων περιέχει αριθμούς που λείπουν, λάθη και σφάλματα μεταγραφής. Για να βρεθούν και να διορθωθούν τα λάθη δεδομένων, εφαρμόζονται δώδεκα πρόσφατα ανακαλυφθέντες κανόνες ποιότητας δεδομένων ως περιορισμοί άρνησης (Baudinet κ.ά., 1999).

Αποτελεί πρόκληση η εξαγωγή ενός πλήρους συνόλου περιορισμών ακεραιότητας που να αντικατοπτρίζει με ακρίβεια τη σημασιολογία του τομέα και τις πολιτικές ενός οργανισμού. Τα δεδομένα είναι συχνά διασκορπισμένα σε αποθήκες δεδομένων- για παράδειγμα, διάφορα τμήματα μπορεί να διατηρούν τα δεδομένα προσωπικού τους ξεχωριστά και να έχουν διαφορετικές πολιτικές μισθολογικών κλιμάκων.

Επιπλέον, ένας οργανισμός μπορεί να έχει κριτήρια ποιότητας δεδομένων ποικίλης εκφραστικότητας, από απλούς ελέγχους ορθότητας έως ad-hoc σενάρια υπολογιστή. Οι τεχνικές για την αυτόματη ανακάλυψη των ICs μπορούν να χρησιμοποιηθούν αντί της διαβούλευσης με ειδικούς του τομέα, αλλά πρέπει να βρίσκουν ισορροπία μεταξύ της εκφραστικότητας των ICs και της αποτελεσματικότητας της ανακάλυψης. Δεδομένου ενός συνόλου δηλωμένων IC και των παραβιάσεών τους σε μια συλλογή βρώμικων δεδομένων, υπάρχουν συχνά πάρα πολλές πιθανές ενημερώσεις των δεδομένων που δεν μπορούν να εξεταστούν διεξοδικά από ανθρώπους. Προκειμένου να προτείνουν τις πιο πιθανές ενημερώσεις, οι αλγόριθμοι καθαρισμού δεδομένων πρέπει να είναι σε θέση να αναζητούν αποτελεσματικά στο τεράστιο σύμπαν των πιθανών αλλαγών.

3.4.1 Ανίχνευση παραβιάσεων

Κάθε διαδικασία εκκαθάρισης δεδομένων πρέπει να ξεκινά με τον εντοπισμό των λαθών. Ο όρος "παραβιάσεις" χρησιμοποιείται για να περιγράψει τα σφάλματα που ανακαλύπτονται με τη χρήση κανόνων ποιότητας δεδομένων. Όσον αφορά τα κριτήρια ποιότητας δεδομένων χωρίς υπαρκτικούς ποσοδείκτες, ο ακόλουθος είναι ένας τυπικός ορισμός μιας παραβίασης: Μια ελάχιστη ομάδα κελιών που είναι ασύμβατα μεταξύ τους αποτελεί παραβίαση. Από αυτόν τον ορισμό μπορούν να συναχθούν οι ακόλουθες πληροφορίες. Αρχικά, δεν είναι όλα τα κελιά μιας παραβίασης στην πραγματικότητα εσφαλμένα. Ωστόσο, ένα από αυτά είναι εσφαλμένο. Δεύτερον, ο αριθμός των κελιών που αποτελούν την παραβίαση πρέπει να είναι μέτριος- αν αφαιρεθούν κάποια κελιά, ο αριθμός των κελιών που μπορούν να συνυπάρξουν στην παραβίαση θα μειωθεί. Τρίτον, η παραβίαση πρέπει να διορθωθεί με την αλλαγή τουλάχιστον ενός κελιού από το σύνολο των κελιών που συνθέτουν την παραβίαση.

Σύμφωνα με τους Ilyas και Chu (2019), στην βιβλιογραφία της ανίχνευσης παραβιάσεων υπάρχουν ορισμένες δυσκολίες και προσκλήσεις: Πρώτον, μια παραβίαση δηλώνει μόνο ότι μια ομάδα κελιών δεν μπορεί να συνυπάρξει μαζί- ορισμένα από τα κελιά της ομάδας μπορεί να είναι ακριβή, ενώ άλλα μπορεί να είναι ανακριβή. Δεύτερον, τα δεδομένα συνήθως πρέπει να περάσουν από πολλά στάδια επεξεργασίας προκειμένου να δημιουργήσουν αξία, επομένως η αντιμετώπιση κάθε παραβίασης ξεχωριστά και αποσπασματικά δεν θα αποκαλύψει ακριβώς ποια κελιά είναι εσφαλμένα. Παρά το γεγονός ότι οι παραβιάσεις δεδομένων ξεκινούν από τα πηγαία δεδομένα, τα σφάλματα συνήθως εντοπίζονται πολύ αργότερα στον αγωγό επεξεργασίας δεδομένων, όταν είναι προσβάσιμη πρόσθετη επιχειρησιακή λογική. Προκειμένου να βρεθούν σφάλματα στην πηγή δεδομένων, οι παραβιάσεις που εντοπίζονται σε μεταγενέστερα στάδια πρέπει να εντοπιστούν και τρίτον, η απαρίθμηση όλων των συνδυασμών k πλειάδων από n πλειάδες εισόδου είναι απαραίτητη για τον εντοπισμό παραβιάσεων των κανόνων ποιότητας δεδομένων που αφορούν k πλειάδες. Αυτό το πολυωνυμικό κόστος μπορεί να αυξηθεί σημαντικά όταν το n είναι μεγάλο.

Για να αντιμετωπίσουν αυτές τις προκλήσεις οι Chu κ.ά., (2013b, 2019) εισάγουν έναν ολιστικό καθαρισμό δεδομένων, ο οποίος συγκεντρώνει όλες τις ανιχνεύσεις παραβιάσεων σε μια ενιαία ομοιογενή αναπαράσταση γνωστή ως υπεργράφημα σύγκρουσης (conflict hypegraph), ανεξάρτητα από το ποια πρότυπα ποιότητας δεδομένων παραβιάζουν. Στη συνέχεια, μπορούμε να προσδιορίσουμε ποια κελιά έχουν μεγαλύτερη πιθανότητα να είναι λανθασμένα με βάση τον υπεργράφο σύγκρουσης.

Επίσης, το γεγονός ότι τα σφάλματα συνήθως εντοπίζονται πολύ αργότερα στην διαδικασία επεξεργασίας δεδομένων καθιστά την πρόκληση της ανίχνευσης σφαλμάτων πιο δύσκολη. Είναι ζωτικής σημασίας η διάδοση των σφαλμάτων που εντοπίζονται στα αποτελέσματα του μετασχηματισμού στις πηγές δεδομένων, προκειμένου να διορθωθούν και να μην επαναληφθούν. Ανάλογα με τον τύπο των αναμενόμενων μετασχηματισμών δεδομένων, χρησιμοποιούνται διαφορετικές τεχνικές διάδοσης σφαλμάτων, όπως Μπουλιανές εκφράσεις (Meliou κ.ά., 2011) και συνάθροιση σε αριθμητικά χαρακτηριστικά (Wu κ.ά., 2012)

3.4.2 Επιδιόρθωση σφαλμάτων

Δεδομένου ότι έχουμε ένα παράδειγμα μιας σχεσιακή βάση δεδομένων I με schema R το οποίο παραβιάζει ένα ορισμένο σύνολο κανόνων ποιότητας δεδομένων, στόχος είναι η εύρεση μιας διαφορετικής περίπτωσης βάσης δεδομένων I′ που συμμορφώνεται με το συγκεκριμένο σύνολο απαιτήσεων ποιότητας δεδομένων. Αυτή η διαδικασία είναι γνωστή ως διόρθωση σφάλματος. Έχουν προταθεί πολυάριθμες μέθοδοι διόρθωσης σφαλμάτων. Μερικές μέθοδοι αναφέρονται παρακάτω.

Οι ολιστικοί αλγόριθμοι διόρθωσης λαμβάνουν υπόψη τις παραβιάσεις που προέρχονται από διάφορους τύπους ολοκληρωμένων κυκλωμάτων ταυτόχρονα και προσφέρουν ενημερώσεις για τη διόρθωση των υποκείμενων δεδομένων. Οι Chu κ.ά. (2013b), για παράδειγμα, λαμβάνουν υπόψη τους μια ποικιλία ICs, όπως FDs, CFDs και DCs, υπό την προϋπόθεση ότι οι παραβιάσεις των ICs μπορούν να αναπαρασταθούν ως ένα "hyperedge" στον υπεργράφο συγκρούσεων. Το NADEEF των Dallachiesa κ.ά., (2013), ένα σύστημα καθαρισμού δεδομένων ανοικτού κώδικα το οποίο χρησιμοποιεί τις μεθόδους των Chu κ.ά., (2013b) για την ολιστική επίλυση των παραβιάσεων, παρέχοντας στους χρήστες μια διεπαφή μέσω της οποίας μπορούν να καθορίζουν τους δικούς τους κανόνες ποιότητας δεδομένων. Όλοι οι περιορισμοί που μπορούν να δηλωθούν ως εξαρτήσεις που δημιουργούν ισότητα λαμβάνονται υπόψη από το LLUNATIC (Geerts κ.ά., 2013). Η ενσωμάτωση της αντιστοίχισης εγγραφών με βάση τα MDs και της αποκατάστασης δεδομένων με βάση τα CFDs από τους Fan κ.ά., (2011b) αποδεικνύει τα συνεργατικά οφέλη του συνδυασμού αυτών των δύο δραστηριοτήτων. Προκειμένου να συγκεντρώσει όλα τα σήματα και να κάνει προβλέψεις σχετικά με πιθανές επισκευές, το HoloClean (Rekatsinas κ.ά., 2017a) χρησιμοποιεί πιθανολογικά γραφικά μοντέλα.

3.5 Καθαρισμός δεδομένων με την βοήθεια Μηχανικής Μάθησης

Η μηχανική μάθηση είναι ένα φυσικό εργαλείο που μπορεί να χρησιμοποιηθεί για ποικίλες εργασίες, όπως ο εντοπισμός διπλών ζευγών κατά την απαλοιφή διπλότυπων, ο προσδιορισμός της πιο πιθανής τιμής για μια ελλιπή τιμή, η πρόβλεψη ενός μετασχηματισμού και η κατηγοριοποίηση τιμών ως κανονικών ή ακραίων τιμών, όπως φαίνεται από τη μέχρι τώρα συζήτηση. Στην πραγματικότητα, η μηχανική μάθηση ενσωματώνεται στην προτεινόμενη διαδικασία καθαρισμού σε αρκετές λύσεις.

Ο καθαρισμός δεδομένων μπορεί να θεωρηθεί ως ένα πιθανολογικό (probabilistic) πρόβλημα βάσεων δεδομένων, όπου η στατιστική και πιθανολογική ερμηνεία των λαθών δεδομένων μπορεί να οδηγήσει σε πιο ολοκληρωμένες και γενικές λύσεις για τον εντοπισμό και τη διόρθωση λαθών. Οι ασυνέπειες στις βάσεις δεδομένων θεωρούνται από την πλειονότητα των μεθόδων επιδιόρθωσης δεδομένων που συζητήθηκαν σε προηγούμενα κεφάλαια ως παραβιάσεις των λογικών νόμων. Αυτές οι μέθοδοι αντιμετωπίζουν τις παραβιάσεις των σκληρών περιορισμών ακεραιότητας με τη λήψη μέτρων για να φέρουν τα δεδομένα σε συμμόρφωση. Για παράδειγμα, μπορούν να τροποποιήσουν ευριστικά την τιμή χαρακτηριστικού μιας πλειάδας για να διορθώσουν ένα πρόβλημα εξάρτησης από μια συνάρτηση ή να αφαιρέσουν εντελώς μια συλλογή πλειάδων.

Αν και στατιστικές και πιθανολογικές προσεγγίσεις χρησιμοποιούνται εδώ και δεκαετίες για τον καθαρισμό των δεδομένων, περιορίζονται σε μεγάλο βαθμό σε αριθμητικές μεθόδους ανίχνευσης ακραίων τιμών (Barnett και Lewis, 1994, Hawkins, 1980, Hawkins κ.ά., 2002) και στην ανάπτυξη δυαδικών ταξινομητών για την απαλοιφή διπλότυπων δεδομένων (Fellegi και Sunter, 1969, Sarawagi και Bhamidipaty, 2002). Η αξιοποίηση τεχνικών μηχανικής μάθησης για τον καθαρισμό δεδομένων αναδεικνύεται σε ένα δημοφιλές και πολλά υποσχόμενο πεδίο, καθώς οι λύσεις μηχανικής μάθησης μεγάλης κλίμακας γίνονται όλο και πιο δημοφιλείς και πλούσιες σε πόρους. Απαιτείται βαθύτερη κατανόηση της σχέσης μεταξύ του καθαρισμού και της πιθανολογικής μοντελοποίησης δεδομένων, επειδή αυτά τα μοντέλα ML βασίζονται σε μια πιθανολογική ερμηνεία των υποκείμενων δεδομένων.

Η υιοθέτηση μιας πιθανολογικής προοπτικής για τον καθαρισμό δεδομένων έχει πολλά πλεονεκτήματα, ένα από τα οποία είναι ότι επιτρέπει την ολοκληρωμένη αντιμετώπιση μιας σειράς λαθών δεδομένων σε μια ενιαία πλατφόρμα. Πολλά παραδείγματα λαθών και ανωμαλιών δεδομένων, συμπεριλαμβανομένων των ακραίων τιμών, των αντιγράφων, των παραβιάσεων των περιορισμών ακεραιότητας και των λανθασμένων ευθυγραμμίσεων, καλύφθηκαν προηγουμένως. Παρόλο που είναι απλούστερο να ορίζονται και να αντιμετωπίζονται αυτά τα προβλήματα δεδομένων ξεχωριστά, τα περισσότερα «βρώμικα» σύνολα δεδομένων έχουν έναν συνδυασμό των περισσότερων ή όλων αυτών των προβλημάτων. Επιπλέον, υπάρχουν σημαντικές αλληλεπιδράσεις μεταξύ αυτών των πολλών σφαλμάτων.

Για παράδειγμα, χωρίς την αντιστοίχιση σχημάτων (mapping schemas), τον χειρισμό πηγών δεδομένων, τον υπολογισμό ελλιπών τιμών και την εξάλειψη ακραίων τιμών, δεν μπορούμε να ανακαλύψουμε αποτελεσματικά τα αντίγραφα. Λόγω των αλληλεξαρτήσεών τους, μελέτες έχουν δείξει ότι δεν υπάρχει μια βέλτιστη σειρά με την οποία να εκτελούνται οι ασκήσεις καθαρισμού. Ακόμη και όταν χρησιμοποιούνται διάφορα εργαλεία, οι επιδόσεις είναι συχνά υποδεέστερες, επειδή κανένα από αυτά δεν έχει καταφέρει να αποτυπώσει πλήρως την ολιστική φύση της διαδικασίας καθαρισμού δεδομένων (Abedjan κ.ά., 2016a, Stonebraker και Ilyas, 2018). Μια ηθική και πολλά υποσχόμενη προσέγγιση για την επίλυση αυτών των προβλημάτων είναι η προσέγγιση της πρόκλησης του καθαρισμού δεδομένων ως πρόβλημα ML μεγάλης κλίμακας.

Η ανάπτυξη λύσεων καθαρισμού δεδομένων έχει χρησιμοποιήσει ή προτείνει σήμερα μια ποικιλία αρχών και μεθοδολογιών ML. Προκειμένου να αποτυπωθεί η συσχέτιση μεταξύ των πολυάριθμων μεταβλητών και σημάτων που εμπλέκονται στην πρόβλεψη της πιο πιθανής τιμής ενός λανθασμένου κελιού της βάσης δεδομένων (τιμή χαρακτηριστικού μιας πλειάδας), χρησιμοποιούνται οι γράφοι παραγόντων, ένα είδος πιθανολογικού γραφικού μοντέλου (Rekatsinas κ.ά., 2017a). Κατά την ενσωμάτωση διαφόρων συνόλων δεδομένων, πιθανολογικά γραφικά μοντέλα έχουν επίσης αξιοποιηθεί για τον προσδιορισμό της πιστότητας και της ακρίβειας των πηγών (Rekatsinas κ.ά., 2017). Προκειμένου να δημιουργηθούν επισημασμένα δεδομένα για το πρόβλημα της σύνδεσης εγγραφών, χρησιμοποιήθηκε με επιτυχία η ενεργητική μάθηση για την ενσωμάτωση ανθρώπινων εμπειρογνωμόνων (Sarawagi και Bhamidipaty, 2002).

3.6 Καθαρισμός δεδομένων με ανθρώπινη συμμετοχή

Συχνά ζητείται η συμβουλή ανθρώπινων ειδικών σε μια διαδικασία καθαρισμού δεδομένων για την επίλυση διαφόρων ειδών αβεβαιότητας, επειδή είναι συνήθως αδύνατο για τις μηχανές να καθαρίσουν τα προβλήματα δεδομένων με ακρίβεια 100%. Δεδομένου ότι τα μεταδεδομένα ανακαλύπτονται με βάση μια περίπτωση του συνόλου δεδομένων και μπορεί να μην ισχύουν πάντα σε οποιαδήποτε άλλη περίπτωση δεδομένων του ίδιου σχήματος, πρέπει πρώτα να επαληθευτούν από ανθρώπους προτού χρησιμοποιηθούν για τον εντοπισμό και τη διόρθωση σφαλμάτων. Ένα παράδειγμα είναι τα μεταδεδομένα (όπως τα κλειδιά και οι περιορισμοί ακεραιότητας) που παράγονται από το βήμα ανακάλυψης και σκιαγράφησης (π.χ. μελλοντικά δεδομένα). Μια άλλη περίπτωση "συνετής" ανθρώπινης παρέμβασης εμφανίζεται κατά το στάδιο της ανίχνευσης σφαλμάτων. Για να δημιουργήσουμε έναν ταξινομητή υψηλής ποιότητας που μπορεί να αποφασίσει με ακρίβεια αν ένα ζεύγος εγγραφών είναι διπλότυπα, χρειαζόμαστε αρκετές επισημειωμένες εγγραφές ως δεδομένα εκπαίδευσης (ως διπλότυπα ή ως μη διπλότυπα). Ένα άνισο σύνολο παραδειγμάτων εκπαίδευσης θα προέκυπτε αν ζητούσαμε από τους ανθρώπους να ταξινομήσουν ένα τυχαίο σύνολο ζευγών εγγραφών, δεδομένου ότι υπάρχουν πολύ περισσότερα ζεύγη μη διπλών εγγραφών από τα ζεύγη διπλών εγγραφών σε ένα σύνολο δεδομένων. Ως εκ τούτου, απαιτείται ευφυής ανθρώπινη διαβούλευση για την απόκτηση εκπαιδευτικών παραδειγμάτων που είναι πιο συμφέροντα για τον ταξινομητή.

Στη φάση της επιδιόρθωσης σφαλμάτων απαιτούνται επίσης άνθρωποι, επειδή οι ενημερώσεις δεδομένων που παράγονται από τους αυτόματους αλγορίθμους επιδιόρθωσης δεδομένων συχνά βασίζονται σε ευρετικές ή/και στατιστικές μεθόδους, όπως η εκτίμηση της μέγιστης πιθανότητας ή η ελάχιστη απόσταση επιδιόρθωσης, και συνεπώς δεν είναι πάντα ακριβείς επιδιορθώσεις. Κατά συνέπεια, πριν χρησιμοποιηθούν σε μια συλλογή δεδομένων, πρέπει να επικυρωθούν από ανθρώπους.

4

Καθαρισμός δεδομένων: Εφαρμογή σε πραγματικά δεδομένα

Δόθηκαν τρεις συλλογές δεδομένων προς καθαρισμό. Με μια σειρά από εντολές σε Excel και Python έγινε καθαρισμός στις συλλογές δεδομένων έτσι ώστε να μπορούν να εισαχθούν σε μια βάση δεδομένων χωρίς να παρουσιάζονται προβλήματα.

4.1 Συλλογή δεδομένων: Εστιατόριο

4.1.1 Περίπτωση Διακύμανσης Τιμών αναλόγως τον Τύπο Κουζίνας

Ξεκινάμε αρχικά με μία μικρή σε μέγεθος συλλογή δεδομένων που μας έχει δοθεί για καθαρισμό, το *Zomato.csv*. Το αρχείο αυτό περιέχει αρχικά 21 στήλες και 9.551 πλειάδες. Σε αυτή τη συλλογή δεδομένων αναφέρονται διάφορες πληροφορίες για εστιατόρια που βρίσκονται ανά τον κόσμο, όπως για παράδειγμα η πόλη στην οποία βρίσκονται, ο τύπος της κουζίνας του εστιατορίου, η βαθμολογία των πελατών, η διεύθυνση του καταστήματος κ.α.

Οι στήλες που υπάρχουν αρχικά μέσα στο αρχείο είναι οι εξής:

- Restaurant Id
- Restaurant Name
- Country Code
- City
- Address
- Locality
- Locality Verbose
- Longitude
- Latitude
- Cuisines

- Average Cost for two
- Currency
- Has Table booking:
- Has Online delivery
- Is delivering.
- Switch to order menu
- Price range
- Aggregate Rating
- Rating color
- Rating text
- Votes

Πρώτο βήμα για τον καθαρισμό της συλλογής δεδομένων, είναι να κρατήσουμε τις επιθυμητές στήλες που χρειαζόμαστε, ανάλογα με τα δεδομένα που θέλουμε να αναλύσουμε στη συνέχεια. Για παράδειγμα δεν χρειάζεται να κρατήσουμε τη στήλη *Is delivering*, αν δεν μας ενδιαφέρει η πολιτική του εστιατορίου σχετικά με τη διανομή φαγητού. Οπότε με γνώμονα αυτό το κριτήριο κρατάμε από τις 21 αρχικές στήλες που περιέχει η συλλογή δεδομένων μόνο τις παρακάτω 7, για την αρχική περίπτωση που θέλουμε να κάνουμε μια σχετική έρευνα, στην οποία να αναφέρεται το είδος της κουζίνας των εστιατορίων που προσφέρει το καθένα, η μέση τιμή κόστους για δύο άτομα, το νόμισμα της κάθε χώρας που βρίσκεται το εστιατόριο, η πόλη στην οποία βρίσκεται και το εύρος τιμής του εστιατορίου. Είναι λογικό στις παρακάτω στήλες να περιλαμβάνονται και οι στήλες *Restaurant Id*, καθώς είναι το Primary Key στη συλλογή δεδομένων και πρέπει να αναφέρεται καθώς και η στήλη *Restaurant name*. Σκοπός της ανάλυσης αυτής είναι να δούμε τις διακυμάνσεις κόστους, ανάλογα με τον τύπο κουζίνας του εστιατορίου και να βγάλουμε μερικά αποτελέσματα σχετικά με το ποια κουζίνα θεωρείται πιο ακριβή σε σχέση με κάποια άλλη, όταν δύο ή περισσότερα εστιατόρια βρίσκονται στην ίδια πόλη.

Ο καθαρισμός στηλών στη συλλογή δεδομένων γίνεται με τη βοήθεια ενός script, που "τρέχει" στην Python. Έτσι προκύπτουν οι εξής στήλες για την πρώτη μας ανάλυση:

- Restaurant Id
- Restaurant Name
- City
- Cuisines
- Average Cost for two
- Currency
- Price range

Δεύτερο βασικό βήμα στον καθαρισμό της συλλογής δεδομένων, είναι να το καθαρίσουμε τις κενές (null) τιμές, ώστε να μην λείπουν δεδομένα από το αρχείο μας, για να είναι αξιόπιστο ως προς την περαιτέρω ανάλυση που θα γίνει στη συνέχεια. Ο καθαρισμός και σε αυτό το σημείο, γίνεται με την βοήθεια ενός script γραμμένο σε Python και παρατηρούμε πως υπάρχουν συνολικά 9 τιμές κενές στη στήλη *Cuisines*. Από τη στιγμή που δεν επιθυμούμε κενές τιμές στη συλλογή δεδομένων, οι πλειάδες που περιέχουν τις κενές αυτές τιμές θα αφαιρεθούν τελείως. Έτσι λοιπόν προκύπτει μια νέα συλλογή δεδομένων με 9.542 πλειάδες και 7 στήλες, χωρίς κανένα απολύτως null δεδομένο.

Ο καθαρισμός της συλλογής δεδομένων συνεχίζεται με τον έλεγχο διπλότυπων τιμών στις στήλες του. Αρχικά ο βασικός έλεγχος για διπλότυπες τιμές στη συλλογή δεδομένων γίνεται στη στήλη που αποτελεί το Primary Key. Είναι απαγορευτικό να είναι περασμένο το ίδιο *Restaurant Id* παραπάνω από μία φορά και γι' αυτό πραγματοποιούμε τον έλεγχο αυτό με τη βοήθεια του προγράμματος Excel. Στις υπόλοιπες στήλες της συλλογής δεδομένων δεν μας απασχολεί αν υπάρχει κάποιο δεδομένο περασμένο παραπάνω από μία φορά, οπότε δεν πραγματοποιείται και περαιτέρω έλεγχος. Για παράδειγμα στη στήλη *Restaurant Name* μπορεί να υπάρχει ένα δεδομένο δύο ή περισσότερες φορές, αλλά μπορεί να πρόκειται για αλυσίδα εστιατορίων στην ίδια ή διαφορετικές πόλεις, οπότε δεν επηρεάζει κάπου την ανάλυσή μας.

Στη συνέχεια πρέπει αν ελέγξουμε αν οι τύποι των δεδομένων είναι σωστοί στη συλλογή δεδομένων που θέλουμε να κρατήσουμε, ώστε τα αποτελέσματα που θα εξάγουμε να είναι σωστά. Για παράδειγμα δεν γίνεται στη στήλη *Restaurant Id* να υπάρχει κάποιο άλλο δεδομένο εκτός από τύπο int. Ο έλεγχος πραγματοποιείται με τη βοήθεια της Python και όπως φαίνεται και στο παρακάτω σχήμα 4.1, όλες οι στήλες περιέχουν το σωστό τύπο δεδομένων που πρέπει.

```
0   Restaurant ID         9542 non-null   int64
1   Restaurant Name       9542 non-null   object
2   City                  9542 non-null   object
3   Cuisines              9542 non-null   object
4   Average Cost for two  9542 non-null   int64
5   Currency              9542 non-null   object
6   Price range           9542 non-null   int64
```

Σχήμα 4.1

Συνεχίζουμε τον καθαρισμό μας με τις ποιοτικές τιμές που περιλαμβάνει η συλλογή δεδομένων. Όπως έχει αναφερθεί και στα προηγούμενα κεφάλαια, πρέπει πάντα η συλλογή δεδομένων που μελετάμε να περιέχει τιμές που αρμόζουν στα δεδομένα, ώστε να μην αλλοιώνεται από λανθασμένες τιμές που μπορεί να περιέχονται σε αυτό. Για παράδειγμα παρατηρούμε πως στη συλλογή δεδομένων που μέχρι στιγμής επεξεργαζόμαστε στη στήλη *Average Cost for two* υπάρχουν 15 πλειάδες συνολικά με τιμή 0 στη στήλη αυτή. Από τη στιγμή που πρόκειται για εστιατόρια που η τιμή τους διαμορφώνεται ανάλογα με την κουζίνα που προσφέρουν στους πελάτες, δεν γίνεται κατά μέσο όρο για δύο άτομα η χρέωση να ισούται με 0, οπότε αυτές τις πλειάδες που περιέχουν τις τιμές αυτές τις αφαιρούμε τελείως από τη συλλογή δεδομένων.

Τέλος στη στήλη *Restaurant Name* για να είναι πιο ευανάγνωστη με τη βοήθεια της Python αφαιρέσαμε τους ειδικούς χαρακτήρες, αντικαθιστώντας τους με κενά, έτσι ώστε να μπορεί να μελετηθεί πιο εύκολα η συλλογή δεδομένων.

Αφού ολοκληρωθούν όλες οι διαδικασίες καθαρισμού, όπως αναφέρθηκαν παραπάνω, παίρνουμε το τελικό αρχείο zomato_1.csv με 9.527 πλειάδες και 7 στήλες και πλέον μπορούμε να το διαβάσουμε μέσω MySQL Workbench και να το μελετήσουμε ποιοτικότερα και με περισσότερη ευκολία, ώστε να εξάγουμε τα αποτελέσματα που θέλουμε.

Από την τελική συλλογή δεδομένων που προκύπτει ύστερα από τον καθαρισμό που έχει γίνει σε αυτή, μπορούμε να δούμε με τη βοήθεια του MySQL Workbench διάφορα αποτελέσματα. Για παράδειγμα μπορούμε να δούμε τα διαφορετικά *Price range*, για τα εστιατόρια που βρίσκονται στην πόλη *Athens* και τι τύπος κουζίνας αντιστοιχεί στο ανάλογο *Price range*.

Τρέχοντας το παρακάτω query βλέπουμε πως στην πόλη *Athens* με *Price range* 3 (40$) μπορεί κάποιος να βρει τις εξής κουζίνες, όπως φαίνεται και στο παρακάτω σχήμα 4.2:

- American, Southern, Southwestern
- American
- Italian
- International, Southern

Restaurant_ID	Restaurant_Name	City	Cuisines	Average_Cost_for_two	Currency	Price_range
17293281	Last Resort Grill	Athens	American, Southern, Southwestern	40	Dollar($)	3
17293205	Five Ten	Athens	American	40	Dollar($)	3
17293273	La Dolce Vita Ristorante	Athens	Italian	40	Dollar($)	3
17293890	The National	Athens	International, Southern	40	Dollar($)	3

Σχήμα 4.2

Ενώ στην ίδια πόλη με αντίστοιχο *Price range 1* (10$) μπορεί να βρει αυτές τις κουζίνες, όπως φαίνεται και στο σχήμα 4.3:

- Southern
- Mexican
- Japanese, Korean
- Breakfast, Burger, Sandwich
- Breakfast, Sandwich
- American, Burger, Sandwich
- International, Southern, Vegetarian
- Sandwich
- Mexican, Spanish
- Bar Food
- Italian, Pizza, Sandwich
- Desserts, Latin American, Argentine

Restaurant_ID	Restaurant_Name	City	Cuisines	Average_Cost_for_two	Currency	Price_range
17293301	Mama s Boy Restaurant	Athens	Southern	10	Dollar($)	1
17293409	Sr Sol 1	Athens	Mexican	10	Dollar($)	1
17293163	Choo Choo Eastside	Athens	Japanese, Korean	10	Dollar($)	1
17293228	The Grill	Athens	Breakfast, Burger, Sandwich	10	Dollar($)	1
17293880	Big City Bread Cafe	Athens	Breakfast, Sandwich	10	Dollar($)	1
17293169	Clocked	Athens	American, Burger, Sandwich	10	Dollar($)	1
17293229	Grit	Athens	International, Southern, Vegetarian	10	Dollar($)	1
17293897	Ike Jane	Athens	Sandwich	10	Dollar($)	1
17293877	Taqueria Del Sol	Athens	Mexican, Spanish	10	Dollar($)	1
17293915	The Royal Peasant	Athens	Bar Food	10	Dollar($)	1
17293422	Transmetropolitan	Athens	Italian, Pizza, Sandwich	10	Dollar($)	1
17294014	Viva Argentine Cuisine	Athens	Desserts, Latin American, Argentine	10	Dollar($)	1

Σχήμα 4.3

Τα query που χρησιμοποιήθηκε ήταν τα εξής:

- select * from zomato_1.zomato_1 where City = 'Athens' and Price_range = '3';

- select * from zomato_1.zomato_1 where City = 'Athens' and Price_range = '1';

Σκοπός λοιπόν από την τελική συλλογή δεδομένων, είναι να μπορεί κάποιος να βγάλει τα επιθυμητά αποτελέσματα, όπως για παράδειγμα τον τύπο της κουζίνας των εστιατορίων σε μία πόλη, χωρίς να κινδυνεύουν να αλλοιωθούν τα αποτέλεσματά του από ανεπιθύμητα ή λανθασμένα δεδομένα, καθώς η βάση του θα είναι καθαρή και βασισμένη σε πληροφορίες που χρειάζεται.

4.1.2 Περίπτωση Γεωγραφικής Κατανομής των Εστιατορίων αναλόγως τη Συνολική Βαθμολογία

Η συλλογή δεδομένων είναι η ίδια που μας έχει δοθεί με την προηγούμενη περίπτωση (Zomato.csv) με τις 21 στήλες και τις 9.551 πλειάδες, μόνο που σε αυτή την περίπτωση θα κρατήσουμε διαφορετικές στήλες, γιατί θέλουμε να βγάλουμε διαφορετικά αποτελέσματα.

Οι στήλες που υπάρχουν αρχικά μέσα στο αρχείο είναι οι εξής:

- Restaurant Id
- Restaurant Name
- Country Code
- City
- Address
- Locality
- Locality Verbose
- Longitude
- Latitude
- Cuisines
- Average Cost for two
- Currency
- Has Table booking:
- Has Online delivery
- Is delivering.
- Switch to order menu
- Price range
- Aggregate Rating
- Rating color
- Rating text
- Votes

Οι στήλες που θα χρειαστεί να κρατήσουμε αυτή τη φορά για να κάνουμε τη μελέτη μας διαφέρουν, σε σχέση με την προηγούμενη τελική συλλογή δεδομένων που κρατήσαμε και διαμορφώνονται ως εξής με τη βοήθεια της Python:

- Restaurant Id
- Restaurant Name
- Country Code
- Longitude
- Latitude
- Cuisines
- Aggregate Rating
- Votes

Όπως αναφέρθηκε και προηγουμένως οι στήλες *Restaurant Id* που είναι το Primary Key στη συλλογή δεδομένων, καθώς και η στήλη Restaurant name περιλαμβάνονται στην τελική συλλογή δεδομένων. Σε αυτή την περίπτωση θα χρειαστούμε και τις στήλες *Longitude* και *Latitude*, καθώς πρόκειται για το γεωγραφικό μήκος και πλάτος του εστιατορίου, που είναι απαραίτητα για τον γεωγραφικό εντοπισμό στο χάρτη. Επίσης κρατάμε και τις στήλες *Aggregate Rating, Votes* γιατί τα αποτελέσματα που θέλουμε να πάρουμε βασίζονται στη συνολική βαθμολογία και τη στήλη *Country Code,* για να ξέρουμε σε ποια χώρα βρίσκεται το εστιατόριο που αναζητάμε πριν κάνουμε τον γεωγραφικό εντοπισμό.

Δεύτερο βασικό βήμα στον καθαρισμό της συλλογής δεδομένων, είναι να το καθαρίσουμε τις κενές (null) τιμές, ώστε να μην λείπουν δεδομένα από το αρχείο μας, για να είναι αξιόπιστο ως προς

περαιτέρω ανάλυση που θα γίνει στη συνέχεια. Ο καθαρισμός και σε αυτό το σημείο, γίνεται με την βοήθεια ενός script γραμμένο σε Python και παρατηρούμε πως δεν υπάρχουν επιπλέον null τιμές πέρα από τις 9 στη στήλη Cuisines που αναφέρθηκαν και στη προηγούμενη περίπτωση, οπότε αφαιρούνται και η συλλογή δεδομένων διαμορφώνεται με 8 στήλες και 9.527 πλειάδες.

Έλεγχος διπλότυπων τιμών δεν γίνεται στη συγκεκριμένη περίπτωση, καθώς στη στήλη *Restaurant Id* που μας ενδιαφέρει κάναμε προηγουμένως και είδαμε πως δεν υπάρχει διπλή τιμή και στις άλλες στήλες, όπως στη *Longitude* και *Latitude* δεν μας επηρεάζει κάπου στα αποτελέσματα που θέλουμε να εξάγουμε, καθώς αν δύο διαφορετικά εστιατόρια βρίσκονται στις ίδια συντεταγμένες, τότε μπορεί να βρίσκονται μέσα σε κάποιο πολυκατάστημα ή σε διαφορετικούς ορόφους.

Και σε αυτή την περίπτωση οι τύποι δεδομένων στις στήλες είναι οι σωστοί που πρέπει, όπως φαίνεται και παρακάτω στο σχήμα 4.4 που πήραμε από την Python.

```
0   Restaurant ID     9542 non-null   int64
1   Restaurant Name   9542 non-null   object
2   Country Code      9542 non-null   int64
3   Longitude         9542 non-null   float64
4   Latitude          9542 non-null   float64
5   Cuisines          9542 non-null   object
6   Aggregate rating  9542 non-null   float64
7   Votes             9542 non-null   int64
```

Σχήμα 4.4

Συνεχίζουμε τον καθαρισμό μας με τις ποιοτικές τιμές που περιλαμβάνει η συλλογή δεδομένων. Όπως έγινε και στην προηγούμενη περίπτωση, πρέπει πάντα η συλλογή δεδομένων που μελετάμε να περιέχει τιμές που αρμόζουν στα δεδομένα, ώστε να μην αλλοιώνεται από λανθασμένες τιμές που μπορεί να περιέχονται σε αυτό. Για παράδειγμα παρατηρούμε πως στη συλλογή δεδομένων που μέχρι στιγμής επεξεργαζόμαστε, στη στήλη *Votes* υπάρχουν δεδομένα με τιμές 0, που σημαίνει πως κανένας δεν έχει ψηφίσει, ώστε να διαμορφωθεί η συνολική βαθμολογίας αξιολόγησης του εστιατορίου. Επομένως οι πλειάδες που περιλαμβάνουν αυτές τις τιμές πρέπει να αφαιρεθούν τελείως, ώστε να μην επηρεάζεται λανθασμένα το ανάλογο αποτέλεσμα. Έτσι με τη βοήθεια της Python, διώχνουμε αυτές τις πλειάδες και η συλλογή δεδομένων διαμορφώνεται με 8 στήλες και 8.433 πλειάδες.

Αφού ολοκληρωθούν όλες οι διαδικασίες καθαρισμού, όπως αναφέρθηκαν παραπάνω, παίρνουμε το τελικό αρχείο *zomato_2.csv* με 8.433 πλειάδες και 8 στήλες και πλέον μπορούμε να το διαβάσουμε μέσω MySQL Workbench και να το μελετήσουμε ποιοτικότερα και με περισσότερη ευκολία, ώστε να εξάγουμε τα αποτελέσματα που θέλουμε.

Από την τελική συλλογή δεδομένων που προκύπτει ύστερα από τον καθαρισμό που έχει γίνει σε αυτό, μπορούμε να δούμε με τη βοήθεια του MySQL Workbench διάφορα αποτελέσματα. Για παράδειγμα μπορούμε να δούμε οι 3 υψηλότερες συνολικές βαθμολογίες εστιατορίων (4.7, 4.8, 4.9), σε ποιες χώρες (κωδικό χώρας) αντιστοιχούν τρέχοντας το παρακάτω query:

- select * from zomato_1.zomato_2 where Aggregate_rating >= 4.7;

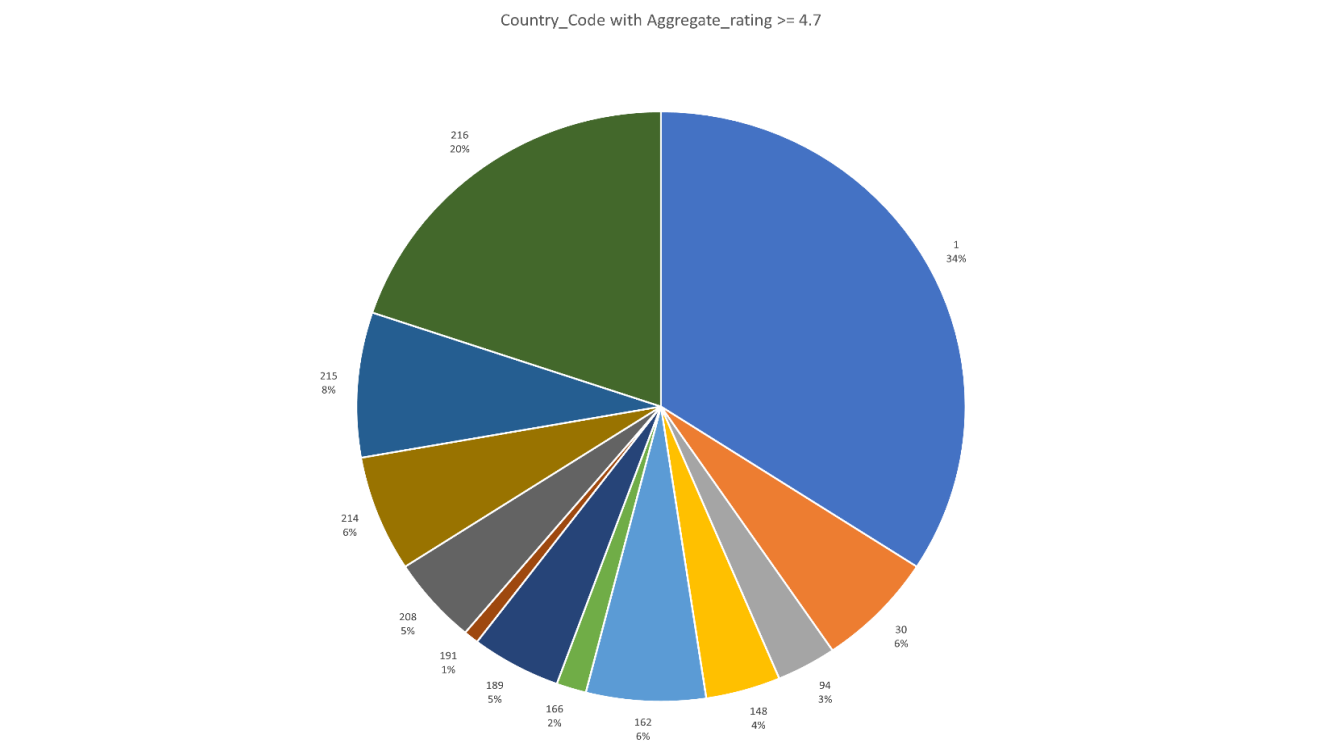
Τα αποτελέσματα που παίρνουμε σύμφωνα με το παραπάνω query φαίνονται και στην παρακάτω εικόνα 4.5. Όπως φαίνεται και από το chart pie στο σχήμα 4.5 οι χώρες με τα περισσότερα εστιατόρια με βαθμολογία μεγαλύτερη ή ίση του 4.7 είναι οι χώρες που αντιστοιχούν στους κωδικούς 1,216 και 215.

Στην περίπτωση που θελήσουμε να εντοπίσουμε τα εστιατόρια που αντιστοιχούν σε αυτούς τους κωδικούς χωρών, μπορούμε μέσω του MySQL Workbench να γράψουμε το αντίστοιχο query και να μας φέρει τα ανάλογα αποτελέσματα. Για παράδειγμα αν θέλουμε να βρούμε τις γεωγραφικές συντεταγμένες για τα εστιατόρια που έχουν συνολική βαθμολογία μεγαλύτερη ή ίση του 4.7 και ανήκουν στον κωδικό χώρας 1, τότε αρκεί να γράψουμε το ακόλουθο query:

- select Longitude, Latitude from zomato_1.zomato_2 where Aggregate_rating >= 4.7 and Country_Code = 1;

Στην περίπτωση που θέλουμε να βρούμε τις γεωγραφικές συντεταγμένες για κάποια άλλη χώρα, αρκεί να αλλάξουμε το *Country Code*, με ένα άλλο *Country Code* που έχουμε βρει από το προηγούμενο query.

Σκοπός της συγκεκριμένης συλλογής δεδομένων είναι να δημιουργηθεί ένα αρχείο στο οποίο να βρίσκονται συγκεντρωμένες οι συνολικές βαθμολογίες για τα εστιατόρια μαζί με τις αντίστοιχες συντεταγμένες και τους κωδικούς χωρών, χωρίς επιπλέον στήλες με περιττές πληροφορίες, ώστε αυτός που μελετάει η συγκεκριμένη συλλογή δεδομένων, να μπορεί να εντοπίζει εύκολα, γρήγορα και ορθά τα αποτελέσματα που θέλει, χωρίς να επηρεάζεται από λανθασμένα δεδομένα.



Σχήμα 4.5

4.2 Συλλογή δεδομένων: Υπηρεσία παροχής μεταφοράς προσώπων Uber

4.2.1 Κόστος κούρσας βασισμένο σε χιλιομετρική απόσταση

Μία δεύτερη συλλογή δεδομένων που μας έχει δοθεί για καθαρισμό, είναι *το uber.csv* στο οποίο περιλαμβάνονται δεδομένα για την γνωστή εταιρεία ταξί Uber και μερικά από αυτά είναι το κόστος κάθε κούρσας, ο αριθμός των επιβατών, οι γεωγραφικές συντεταγμένες έναρξης του ταξίμετρου, καθώς και οι γεωγραφικές συντεταγμένες που λήξης αυτού. Η συλλογή δεδομένων αποτελείται αρχικά από 200.000 πλειάδες και 9 στήλες με τις τελευταίες να φαίνονται παρακάτω:

- unnamed: 0
- key
- fare_amount
- pickup_datetime
- pickup_longitude
- pickup_latitude
- dropoff_longitude
- dropoff_latitude
- passenger_count

Αρχικά σαν πρώτο βήμα για τον καθαρισμό της συλλογής δεδομένων, θα πρέπει να κρατήσουμε τις στήλες που χρειαζόμαστε, για να μπορέσουμε στη συνέχεια να πάρουμε τα αποτελέσματα που θέλουμε εύκολα χωρίς περιττή πληροφορία. Στην περίπτωση μας, θέλουμε να καταλήξουμε σε μία συλλογή δεδομένων, για να βλέπουμε το κόστος της κάθε κούρσας, ανάλογα την χιλιομετρική απόσταση που έχει διανύσει. Για να μπορέσουμε να υπολογίσουμε την χιλιομετρική απόσταση που έχει διανύσει ένα ταξί, θα χρησιμοποιήσουμε την Haversine formula (ο τύπος της αναφέρεται παρακάτω, τύπος (2), γραμμένη σε Python, η οποία μας υπολογίζει την απόσταση μεγάλου κύκλου δύο σημείων σε μία σφαίρα, με δεδομένα το γεωγραφικό μήκος και πλάτος. Θα χρειαστούμε στήλες όπως *pickup_longitude, pickup_latitude*, στις οποίες αναφέρονται οι συντεταγμένες που ξεκίνησε να μετράει το ταξίμετρο, καθώς και τις στήλες *dropoff_longitude* και *dropoff_latitude*, που είναι οι συντεταγμένες στις οποίες σταμάτησε να μετράει αυτό, ώστε να μπορέσουμε να υπολογίσουμε τις αποστάσεις. Επιπλέον απαραίτητη στήλη είναι η *fare_amount*, στην οποία αναγράφεται το κόστος της κάθε κούρσας. Η πρώτη στήλη που δεν έχει κάποιο γνώρισμα, πιθανότητα πρέπει να είναι το Primary Key στην συλλογή δεδομένων, καθώς φαίνεται να περιέχει ξεχωριστό αριθμό (ID) για κάθε κούρσα και στην περίπτωση μας, η συγκεκριμένη στήλη κρίνεται αναγκαία να κρατηθεί. Έτσι με τη βοήθεια της Python τρέχουμε το script που μας κρατάει τις επιθυμητές στήλες που αναφέρθηκαν παραπάνω και καταλήγουμε στις εξής 7, καθώς την 7η στήλη την δημιουργούμε εμείς:

- unnamed: 0
- fare_amount

- pickup_longitude
- pickup_latitude
- dropoff_longitude
- dropoff_latitude
- dist_travel_km

Haversine formula:

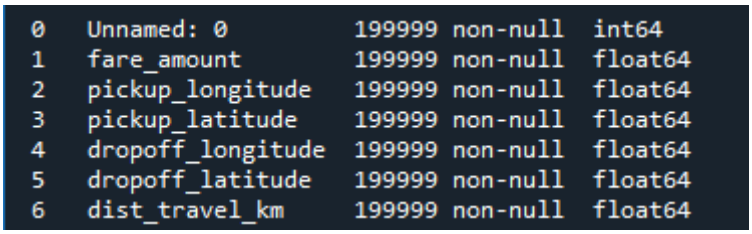
$$\mathrm{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right)$$

(2)

Δεύτερο βασικό βήμα στον καθαρισμό της συλλογής δεδομένων, είναι να το καθαρίσουμε τις κενές (null) τιμές, ώστε να μην έχουμε ελλιπή δεδομένα από το αρχείο μας, για να είναι αξιόπιστο ως προς τη περεταίρω ανάλυση που θα γίνει στη συνέχεια. Ο καθαρισμός και σε αυτό το σημείο, γίνεται με την βοήθεια της Python και παρατηρούμε πως υπάρχουν συνολικά 2 τιμές κενές στις στήλες *dropoff_longtitude* και *dropoff_latitude*, που τυχαίνει να βρίσκονται και στην ίδια πλειάδα. Από τη στιγμή που δεν επιθυμούμε κενές τιμές στη συλλογή δεδομένων, οι πλειάδες που περιέχουν τις κενές αυτές τιμές θα αφαιρεθούν τελείως. Έτσι λοιπόν προκύπτει μια νέα συλλογή δεδομένων με 199.999 πλειάδες και 7 στήλες, χωρίς κανένα απολύτως null δεδομένο.

Τον καθαρισμό της συλλογής δεδομένων τον συνεχίζουμε με τον έλεγχο διπλότυπων τιμών στις στήλες του αρχείου. Αρχικά ο βασικός έλεγχος για διπλότυπες τιμές στη συλλογή δεδομένων γίνεται στη στήλη που αποτελεί το Primary Key. Είναι απαγορευτικό το Primary Key να είναι το ίδιο για περισσότερες από μια πλειάδες. Στη συγκεκριμένη περίπτωση η πρώτη στήλη του αρχείου που δεν έχει κάποιο γνώρισμα αποτελεί το πρωτεύον κλειδί και ύστερα από έλεγχο που πραγματοποιούμε με τη βοήθεια του προγράμματος Excel δεν εντοπίζουμε κάποια τιμή περασμένη παραπάνω από μια φορά. Στις υπόλοιπες στήλες της συλλογής δεδομένων, δεν μας επηρεάζει αν υπάρχει κάποιο δεδομένο περασμένο παραπάνω από μια φορά, καθώς πρόκειται για συντεταγμένες και κόστος διαδρομής, οπότε δεν πραγματοποιείται και περαιτέρω έλεγχος.

Συνεχίζουμε τον καθαρισμό του αρχείου, με τον έλεγχο των τύπων από τα δεδομένα. Θα πρέπει σε κάθε στήλη να υπάρχουν δεδομένα με τον αντίστοιχο σωστό τύπο τους, ώστε τα αποτελέσματα που θα εξάγουμε να είναι σωστά. Για παράδειγμα στις στήλες που περιλαμβάνουν τις γεωγραφικές συντεταγμένες θα πρέπει να είναι αποκλειστικά και μόνο τύπου float. Ο έλεγχος πραγματοποιείται με τη βοήθεια της Python και όπως φαίνεται στο παρακάτω σχήμα 4.6, όλες οι στήλες περιέχουν το σωστό τύπο δεδομένων που πρέπει.



0	Unnamed: 0	199999	non-null	int64
1	fare_amount	199999	non-null	float64
2	pickup_longitude	199999	non-null	float64
3	pickup_latitude	199999	non-null	float64
4	dropoff_longitude	199999	non-null	float64
5	dropoff_latitude	199999	non-null	float64
6	dist_travel_km	199999	non-null	float64

Σχήμα 4.6

Σημαντικό σημείο στον καθαρισμό της συλλογής δεδομένων, είναι οι ποιοτικές τιμές που περιλαμβάνονται σε αυτό, καθώς από αυτές εξαρτάται η ποιότητα και ορθότητα των αποτελεσμάτων που θέλουμε να εξάγουμε. Αρχικά παρατηρούμε πως στη στήλη *dist_travel_km* που δημιουργήσαμε, υπάρχουν μηδενικές τιμές. Από τη στιγμή που θέλουμε να υπολογίσουμε την χιλιομετρική απόσταση που έχει διανύσει ένα ταξί uber, αναλόγως τις συντεταγμένες από τις οποίες ξεκίνησε και τις συντεταγμένες στις οποίες κατέληξε, συμπεραίνουμε πως οι πλειάδες που περιλαμβάνουν αυτές τις τιμές θα πρέπει να αφαιρεθούν τελείως. Μελετώντας τη συλλογή δεδομένων με τη βοήθεια της Python και του Excel, παρατηρούμε πως στο σύνολο τους οι πλειάδες αυτές με μηδενικές τιμές στη στήλη *dist_travel_km*, ανέρχονται σε 35.894. Έτσι προκύπτει ένα νέο dataset με 7 στήλες και 164.105 πλειάδες στο σύνολο του. Με παρόμοια λογική εξετάζοντας τη στήλη *fare_amount* στην οποία αναφέρονται οι συνολικές αξίες της κούρσας θα πρέπει να αφαιρεθούν τυχόν μη αποδεκτές τιμές, όπως μηδενικές ή τιμές που αντιστοιχούν σε παράλογα ποσά ή ποσά με αρνητική αξία. Ύστερά από εκτενέστατο έλεγχο παρατηρούμε πως αφαιρέθηκαν συνολικά 94 πλειάδες με τιμές που δεν θα έπρεπε να περιέχονται στη στήλη και έτσι η νέα συλλογή δεδομένων περιέχει τελικά 7 στήλες και 164.011 πλειάδες. Το εύρος τιμών για τη στήλη *fare_amount* εκτείνεται τελικά από 10 έως και 99,2 (USD).

Αφού έχουν ολοκληρωθεί όλες οι διαδικασίες, όπως αναφέρθηκαν παραπάνω, παίρνουμε την τελική καθαρισμένη συλλογή δεδομένων *uber_1.csv* με 164.011 πλειάδες και 7 στήλες. Παρατηρούμε πως το μέγεθος του τελικού καθαρισμένου αρχείου είναι 11,4 mb σε αντίθεση με το αρχικό που έφτανε τα 22,3 mb. Επόμενο βήμα είναι να το διαβάσουμε με το MySQL Workbench και να τρέξουμε τα αντίστοιχα query για να πάρουμε τα αποτελέσματα που θέλουμε. Από τη στιγμή που το διαβάζουμε με το MySQL Workbench παρατηρούμε πως φορτώνεται κανονικά, με όλες τις στήλες και τις πλειάδες που περιέχει, οπότε δεν υπάρχει κάποιο πρόβλημα και μπορούμε να συνεχίσουμε με την μελέτη της συλλογής δεδομένων. Επόμενο βήμα είναι να φτιάξουμε το αντίστοιχο query που είναι απαραίτητο για τα αποτελέσματα που θέλουμε να πάρουμε, ώστε να μπορέσουμε να βγάλουμε τα ανάλογα συμπεράσματα. Αρχικά μπορούμε να φτιάξουμε ένα query που να μας φέρνει σαν αποτέλεσμα το κόστος της κάθε κούρσας, για τις περιπτώσεις που το ταξίμετρο έχει γράψει από 3 έως και 5 χιλιόμετρα και να μπορέσουμε να βγάλουμε ένα μέσο όρο. Το query που χρησιμοποιούμε για το συγκεκριμένο σενάριο είναι το ακόλουθο:

- select fare_amount , dist_travel_km from uber.uber_1 where dist_travel_km >=3 and dist_travel_km <5;

Μελετώντας τα δεδομένα που μας εξάγει το παραπάνω query, μπορούμε να υπολογίσουμε το μέσο όρο της συνολικής αξίας, όταν το ταξίμετρο καταγράφει από 3 έως 5 χιλιόμετρα. Υπολογίζοντας το μέσο όρο, παρατηρούμε πως η μέση τιμή για μια τέτοια διαδρομή είναι περίπου ίση με 12,29 (USD).

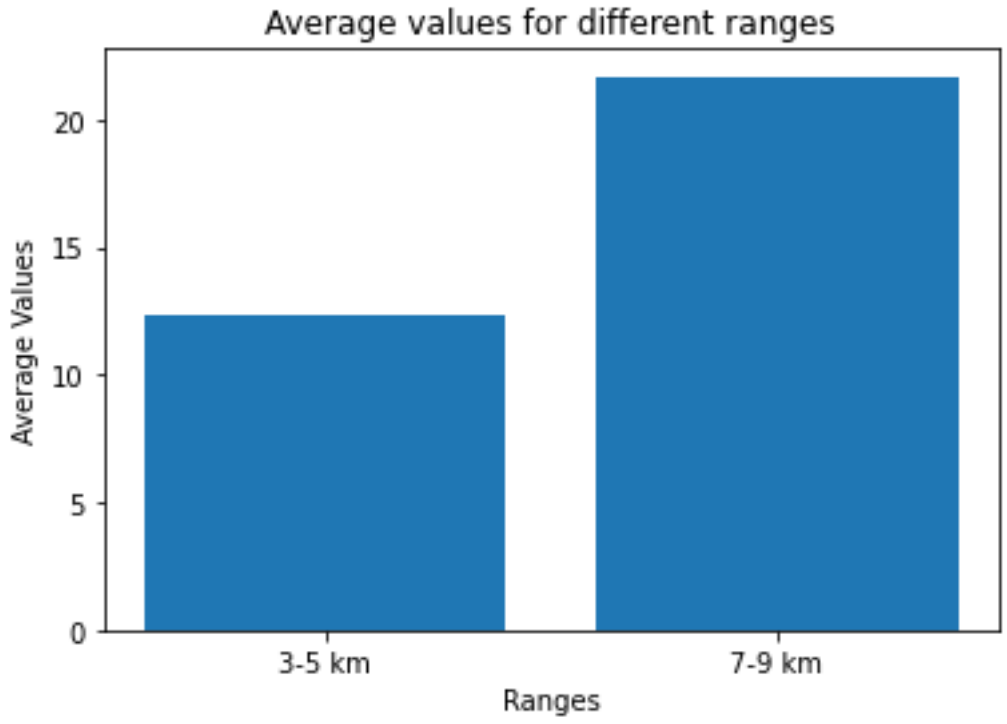
Μπορούμε να βρούμε με ένα δεύτερο αντίστοιχο query τη συνολική αξία κάθε κούρσας για διαφορετικό εύρος χιλιομετρικών αποστάσεων και να συγκρίνουμε τα δύο αποτελέσματα, ώστε να βγάλουμε διάφορα συμπεράσματα. Το query που χρησιμοποιούμε για τον δεύτερο υπολογισμό είναι το παρακάτω και πρόκειται να μας φέρει τη συνολική αξία κάθε κούρσας ξεχωριστά για το χιλιομετρικό διάστημα 7 έως 9.

- select fare_amount , dist_travel_km from uber.uber_1 where dist_travel_km >=7 and dist_travel_km <9;

Μελετώντας τα δεδομένα από το δεύτερο query και υπολογίζοντας το μέσο όρο της συνολικής αξίας για ταξίμετρο με τιμές 7 έως 9 χιλιόμετρα για ένα ταξί uber, τη μέση τιμή τη βρίσκουμε ίση περίπου με 21,68 (USD).

Η διαφορά στις μέσες τιμές της συνολικής αξίας της κούρσας για τα παραπάνω διαστήματα χιλιομετρικής απόστασης αναπαρίσταται και στην παρακάτω εικόνα με τη βοήθειας της Python.

Σκοπός της τελικής συλλογής δεδομένων είναι να δημιουργηθεί ένα αρχείο στο οποίο να μπορεί να εξάγει ο ενδιαφερόμενος πληροφορίες σχετικά με τις συνολικές χιλιομετρικές αποστάσεις και αξίες της κάθε κούρσας, ώστε να μπορεί να κάνει μια εύκολη εκτίμηση της αξίας, με πληροφορίες όπως τις γεωγραφικές συντεταγμένες έναρξης και λήξης.



Σχήμα 4.7

4.2.2 Υπολογισμός μέσης τιμής κόστους κούρσας ανά άτομο σε σχέση με την χιλιομετρική απόσταση

Έχουμε ένα δεύτερο υποθετικό σενάριο που κάποιος θέλει να υπολογίσει από τη συλλογή δεδομένων *uber.csv* που μας έχει δοθεί, το κόστος κούρσας που αναλογεί σε ένα άτομο, όταν στο ταξί uber υπάρχουν παραπάνω από ένα επιβάτες. Η αρχική συλλογή δεδομένων, όπως είχαμε δει και πιο πριν περιλαμβάνει 200.000 πλειάδες και τις εξής 9 στήλες:

- Unnamed: 0
- Key
- Fare_amount
- Pickup_datetime
- Pickup_longitude
- Pickup_latitude
- Dropoff_longitude
- Dropoff_latitude
- Passenger_count

Αρχικά αυτό που πρέπει να κάνουμε είναι να κρατήσουμε τις στήλες που θα μας βοηθήσουν να υπολογίσουμε και να πάρουμε τα δεδομένα που θέλουμε. Οπότε από τις παραπάνω 9 στήλες θα χρειαστούμε τη στήλη *fare_amount* που αναφέρεται το ποσό της κούρσας, τις στήλες με τις γεωγραφικές συντεταγμένες, καθώς και τη στήλη *passenger count* στην οποία αναφέρεται ο αριθμός των επιβατών μέσα στο ταξί. Την 6η στήλη *dist_travel_km* την δημιουργούμε εμείς, όπως και στην προηγούμενη συλλογή δεδομένων με τη βοήθεια της Haversine formula που έχει αναφερθεί και παραπάνω. Ο βασικός αυτό καθαρισμός των στηλών γίνεται με τη βοήθεια της Python και του excel και έτσι καταλήγουμε σε μια συλλογή δεδομένων με 200.000 πλειάδες και 7 στήλες, όπως φαίνεται παρακάτω.

- fare_amount
- pickup_longtitude
- pickup_latitude
- dropoff_longtitude
- dropoff_latitude
- passenger_count
- dist_travel_km

Το πιο σημαντικό βήμα για τον καθαρισμό της βάσης μας είναι να καθαρίσουμε τη συλλογή δεδομένων από κενές (null) τιμές, ώστε να μην έχουμε ελλιπή δεδομένα από το αρχείο μας, για να είναι αξιόπιστο ως προς τη περεταίρω ανάλυση που θα γίνει στη συνέχεια. Ο καθαρισμός και σε

αυτό το σημείο, γίνεται με την βοήθεια της Python και παρατηρούμε πως υπάρχουν συνολικά 2 τιμές κενές στις στήλες *dropoff_longtitude* και *dropoff_latitude*, που τυχαίνει να βρίσκονται και στην ίδια πλειάδα. Από τη στιγμή που δεν επιθυμούμε κενές τιμές στη συλλογή δεδομένων, οι πλειάδες που περιέχουν τις κενές αυτές τιμές θα αφαιρεθούν τελείως. Έτσι λοιπόν προκύπτει μια νέα συλλογή δεδομένων με 199.999 πλειάδες και 7 στήλες, χωρίς κανένα απολύτως null δεδομένο.

Έλεγχος διπλότυπων τιμών δεν χρειάζεται να γίνει στη συγκεκριμένη συλλογή δεδομένων, καθώς δεν μας επηρεάζει κάπου αν υπάρχουν τυχόν διπλότυπες τιμές σε κάποιες από τις στήλες που έχουμε κρατήσει. Για παράδειγμα δεν επηρεάζει κάπου την μελέτη μας, αν στη στήλη *passenger_count* υπάρχει παραπάνω από μία φορά ένα δεδομένο, καθώς πρόκειται για αριθμό επιβατών.

Σημαντικό στάδιο στον καθαρισμό της συλλογής δεδομένων όμως, είναι ο ποιοτικός έλεγχος των τιμών που περιλαμβάνονται στο αρχείο, καθώς θα πρέπει να περιέχονται δεδομένα που ανταποκρίνονται σε ρεαλιστικά σενάρια. Για αυτό το λόγο γίνεται προσεκτικός έλεγχος των δεδομένων στις στήλες της συλλογής δεδομένων. Για παράδειγμα όπως και στη προηγούμενη περίπτωση στη στήλη *fare_amount* θα πρέπει να αφαιρεθούν τιμές μηδενικές ή με αρνητικό πρόσημο, καθώς και τιμές που τείνουν στο μηδέν ή σε εξωφρενικά υψηλά ποσά. Σύμφωνα με τα παραπάνω και με τη βοήθεια της Python αφαιρούνται συνολικά 111 πλειάδες που περιέχουν τέτοιες τιμές στη στήλη fare amount και η νέα συλλογή δεδομένων που προκύπτει περιέχει 7 στήλες και 199.888 πλειάδες. Επόμενη στήλη που πρέπει να γίνει έλεγχος ποιοτικών τιμών είναι η στήλη *passenger_count*. Στη στήλη πρέπει να αφαιρεθούν τιμές που ισούνται με το μηδέν, καθώς αναφερόμαστε σε αριθμό επιβατών μέσα σε ταξί, καθώς και τιμές με πολλά ψηφία, καθώς δεν γίνεται σε ένα αμάξι να χωράνε περισσότερα από 6-7 άτομα , ανάλογα τον τύπο του αμαξιού (βαν, απλό επιβατικό κτλ.). Σύμφωνα με τα παραπάνω αφαιρούνται οι πλειάδες που περιέχουν τέτοιες τιμές και η νέα τελική συλλογή δεδομένων περιλαμβάνει 7 στήλες και 199.179 πλειάδες. Από τη στήλη *dist_travel_km* θα πρέπει να αφαιρεθούν τιμές με αρνητικό πρόσημο, μηδενικές, καθώς και τιμές που είναι αρκετά μεγάλες και δεν αντιστοιχούν σε πραγματικά δεδομένα. Η νέα τελική συλλογή δεδομένων που προκύπτει περιέχει 7 στήλες και 163.431 πλειάδες. Ο παραπάνω καθαρισμός έγινε με τη βοήθεια του excel και της Python.

Ένα ακόμη απαραίτητο βήμα για τον καθαρισμό της συλλογής δεδομένων είναι να ελέγξουμε τον τύπο των δεδομένων που βρίσκονται στη κάθε στήλη. Για παράδειγμα στη στήλη *passenger_count* δεν γίνεται να υπάρχουν τιμές float, παρά μόνο ακέραιες. Ο έλεγχος γίνεται με τη βοήθεια της Python και όπως βλέπουμε και στην παρακάτω εικόνα 4.8 τα δεδομένα μας έχουν τον σωστό τύπο που πρέπει.

```
1   fare_amount        199999 non-null  float64
2   pickup_longitude   199999 non-null  float64
3   pickup_latitude    199999 non-null  float64
4   dropoff_longitude  199999 non-null  float64
5   dropoff_latitude   199999 non-null  float64
6   passenger_count    199999 non-null  int64
7   dist_travel_km     199999 non-null  float64
```

Σχήμα 4.8

Αφού έχουν ολοκληρωθεί όλες οι διαδικασίες, όπως αναφέρθηκαν και παραπάνω παίρνουμε τη τελική συλλογή δεδομένων με ονομασία *uber_2.csv* και παρατηρούμε πως το μέγεθός του είναι 10,4 mb σε αντίθεση με το αρχικό που ήταν 22,3 mb. Φορτώνοντας το τελικό .csv αρχείο στο MySQL Workbench παρατηρούμε πως φορτώνεται κανονικά χωρίς κανένα πρόβλημα, φέρνοντας όλες τις στήλες και τις πλειάδες που περιέχει. Επόμενο βήμα είναι να φτιάξουμε το αντίστοιχο query για τα αποτελέσματα που θέλουμε να εξαγάγουμε και τα συμπεράσματα που θέλουμε να βγάλουμε. Αρχικά θα υπολογίσουμε τη μέση τιμή του κόστους της διαδρομής ανά άτομο, όταν μέσα στο ταξί βρίσκεται ένα άτομο και στη συνέχεια θα υπολογίσουμε τη μέση τιμή του κόστους ανά άτομο, όταν βρίσκονται μέσα στο ταξί παραπάνω από ένα άτομα. Τέλος θα συγκρίνουμε τις τιμές αυτές και θα βγάλουμε ένα συμπέρασμα για το πότε μια κούρσα είναι πιο οικονομική αναλόγως τον αριθμό των επιβατών για τη συγκεκριμένη χιλιομετρική απόσταση. Από τη στιγμή που θέλουμε να υπολογίσουμε το μέσο όρο του κόστους ανά άτομο αναλόγως την χιλιομετρική απόσταση, θα πρέπει να ορίσουμε ένα εύρος τιμών στην απόσταση. Για αρχή θα μελετήσουμε για χιλιομετρικό διάστημα από 1 έως 3 χιλιόμετρα και όταν μέσα στο αμάξι υπάρχει ένας επιβάτης. Έτσι λοιπόν το αντίστοιχο query θα είναι αυτό, όπως φαίνεται και παρακάτω:

- select fare_amount from uber.uber_2 where dist_travel_km >=1 and dist_travel_km <3 and passenger_count =1;

Μελετώντας το data grid από το παραπάνω query με τη βοήθεια του excel και για την περίπτωση που μέσα στο αμάξι υπάρχει ένα άτομο η μέση τιμή του κόστους υπολογίζεται σε 10,61 (USD).

Στην περίπτωση που μέσα στο αμάξι υπάρχουν δύο επιβάτες το query γράφεται ως εξής:

- select fare_amount from uber.uber_2 where dist_travel_km >=1 and dist_travel_km <3 and passenger_count =2;

Μελετώντας το data grid από το παραπάνω query με τη βοήθεια του excel και για την περίπτωση που μέσα στο αμάξι υπάρχουν δύο άτομα η μέση τιμή του κόστους υπολογίζεται σε 11,30 (USD), άρα ανά άτομο 5,65 (USD).

Στην περίπτωση που μέσα στο αμάξι υπάρχουν τρεις επιβάτες το query γράφεται ως εξής:

- select fare_amount from uber.uber_2 where dist_travel_km >=1 and dist_travel_km <3 and passenger_count =3;

Μελετώντας το data grid από το παραπάνω query με τη βοήθεια του excel και για την περίπτωση που μέσα στο αμάξι υπάρχουν τρία άτομα η μέση τιμή του κόστους υπολογίζεται σε 10,96 (USD), άρα ανά άτομο 3,65 (USD).

Στην περίπτωση που μέσα στο αμάξι υπάρχουν τέσσερις επιβάτες το query γράφεται ως εξής:

- select fare_amount from uber.uber_2 where dist_travel_km >=1 and dist_travel_km <3 and passenger_count =4;

Μελετώντας το data grid από το παραπάνω query με τη βοήθεια του excel και για την περίπτωση που μέσα στο αμάξι υπάρχουν τέσσερα άτομα η μέση τιμή του κόστους υπολογίζεται σε 11,18 (USD), άρα ανά άτομο 2,80 (USD).

Στην περίπτωση που μέσα στο αμάξι υπάρχουν πέντε επιβάτες το query γράφεται ως εξής:

- select fare_amount from uber.uber_2 where dist_travel_km >=1 and dist_travel_km <3 and passenger_count =5;

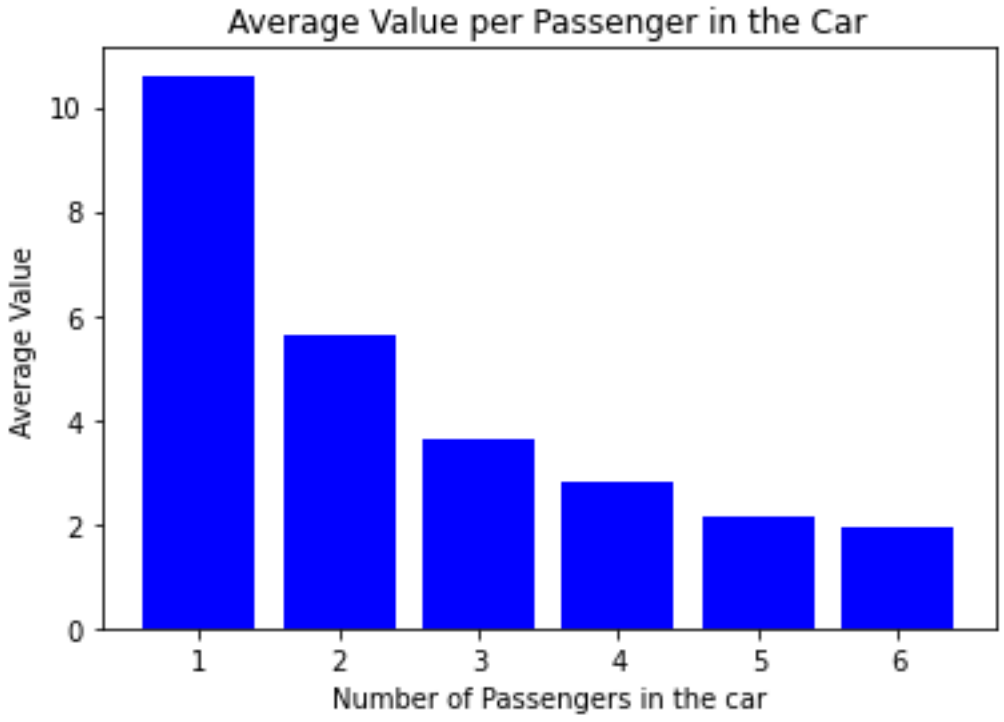
Μελετώντας το data grid από το παραπάνω query με τη βοήθεια του excel και για την περίπτωση που μέσα στο αμάξι υπάρχουν πέντε άτομα η μέση τιμή του κόστους υπολογίζεται σε 10,70 (USD), άρα ανά άτομο 2,14 (USD).

Στην περίπτωση που μέσα στο αμάξι υπάρχουν έξι επιβάτες το query γράφεται ως εξής:

- select fare_amount from uber.uber_2 where dist_travel_km >=1 and dist_travel_km <3 and passenger_count =6;

Μελετώντας το data grid από το παραπάνω query με τη βοήθεια του excel και για την περίπτωση που μέσα στο αμάξι υπάρχουν έξι άτομα η μέση τιμή του κόστους υπολογίζεται σε 11,76 (USD), άρα ανά άτομο 1,96 (USD).

Τελικά συγκρίνοντας τα τελικά αποτελέσματα από τα data grid, που έχουμε εξάγει από τα παραπάνω query, βλέπουμε πως για χιλιομετρική απόσταση από 1 έως 3 χιλιόμετρα η πιο οικονομική κούρσα ανά άτομο είναι όταν υπάρχουν 6 επιβάτες και η πιο δαπανηρή, όταν υπάρχει μόνο ένας επιβάτης μέσα στο ταξί, όπως φαίνεται και από το παρακάτω σχήμα.



Σχήμα 4.9

4.3 Συλλογή δεδομένων: Μεταχειρισμένα αυτοκίνητα

4.3.1 Υπολογισμός μέσης τιμής κόστους των οχημάτων ανά κατασκευαστή και χιλιομετρική απόσταση

Μια τρίτη περίπτωση καθαρισμού συλλογής δεδομένων που μας έχει δοθεί, είναι το αρχείο vehicles.csv, στο οποίο περιλαμβάνονται δεδομένα για το Craigslist.org, που περιλαμβάνει την μεγαλύτερη συλλογή μεταχειρισμένων αυτοκίνητων στον κόσμο. Το αρχικό μέγεθος του αρχείου είναι αρκετά μεγάλο 1,34 Gb, καθώς περιλαμβάνει 426.880 πλειάδες και 26 στήλες, όπως φαίνεται και παρακάτω:

- id
- url
- region
- region_url
- price
- year
- manufacturer
- model
- condition
- cylinders
- fuel
- odometer
- title_status
- transmission
- VIN
- drive
- size
- type
- paint_color
- image_url
- description
- country
- state
- lat
- long
- posting_date

Αρχικά σαν πρώτο βήμα για τον καθαρισμό της συλλογής δεδομένων, όπως και στις προηγούμενες περιπτώσεις είναι να κρατήσουμε τις στήλες που χρειαζόμαστε, για να μπορέσουμε στη συνέχεια να πάρουμε τα αποτελέσματα που θέλουμε και να βγάλουμε εύκολα τα συμπεράσματα που πρέπει χωρίς περιττή πληροφορία. Στην περίπτωση μας, θέλουμε να καταλήξουμε σε μία συλλογή δεδομένων, στο οποίο να περιέχονται οι ανάλογες πληροφορίες, ώστε να μπορεί κάποιος να

υπολογίζει της μέση τιμή, αναλόγως τον κατασκευαστή του αυτοκινήτου, την χρονιά που κατασκευάστηκε και τα συνολικά χιλιόμετρα που είναι περασμένα στο κοντέρ. Οπότε με τη βοήθεια της Python θα κρατήσουμε τις αντίστοιχες στήλες που περιέχουν τις πληροφορίες αυτές. Η στήλη που θα χρειαστούμε για αυτό το σενάριο αρχικά είναι η στήλη *id* η οποία αποτελεί το Primary κλειδί για τη συλλογή δεδομένων, καθώς αναφέρεται σε αυτή το μοναδικό αναγνωριστικό που δίνεται για κάθε εγγραφή. Η στήλη *price* που περιλαμβάνει την τιμή σε (USD), καθώς και οι στήλες *manufacturer* και *odometer* που έχουν δεδομένα για τον κατασκευαστή του αυτοκινήτου καθώς και τα χιλιόμετρα που έχει διανύσει το αυτοκίνητο από την στιγμή που αγοράστηκε. Έτσι λοιπόν προκύπτει μια νέα συλλογή δεδομένων με 5 στήλες και 426.880 πλειάδες με τις στήλες να φαίνονται παρακάτω:

- id
- price
- year
- manufacturer
- odometer

Αρχικό βήμα για τον καθαρισμό της συλλογής δεδομένων που έχουμε, είναι όπως γνωρίζουμε να καθαρίσουμε τις κενές (null) τιμές, ώστε να μην έχουμε ελλιπή δεδομένα από το αρχείο μας, για να είναι αξιόπιστο ως προς τη περεταίρω ανάλυση που θα γίνει στη συνέχεια. Την εύρεση null τιμών την πραγματοποιούμε με την βοήθεια της Python και βρίσκουμε αρκετές κενές τιμές. Συγκεκριμένα βρίσκουμε 1.205 κενές τιμές στη στήλη *year*, 17.646 στη στήλη manufacturer και 4.400 στη στήλη *odometer*. Έτσι λοιπόν προκύπτει μια νέα συλλογή δεδομένων με 5 στήλες και 405.077 πλειάδες.

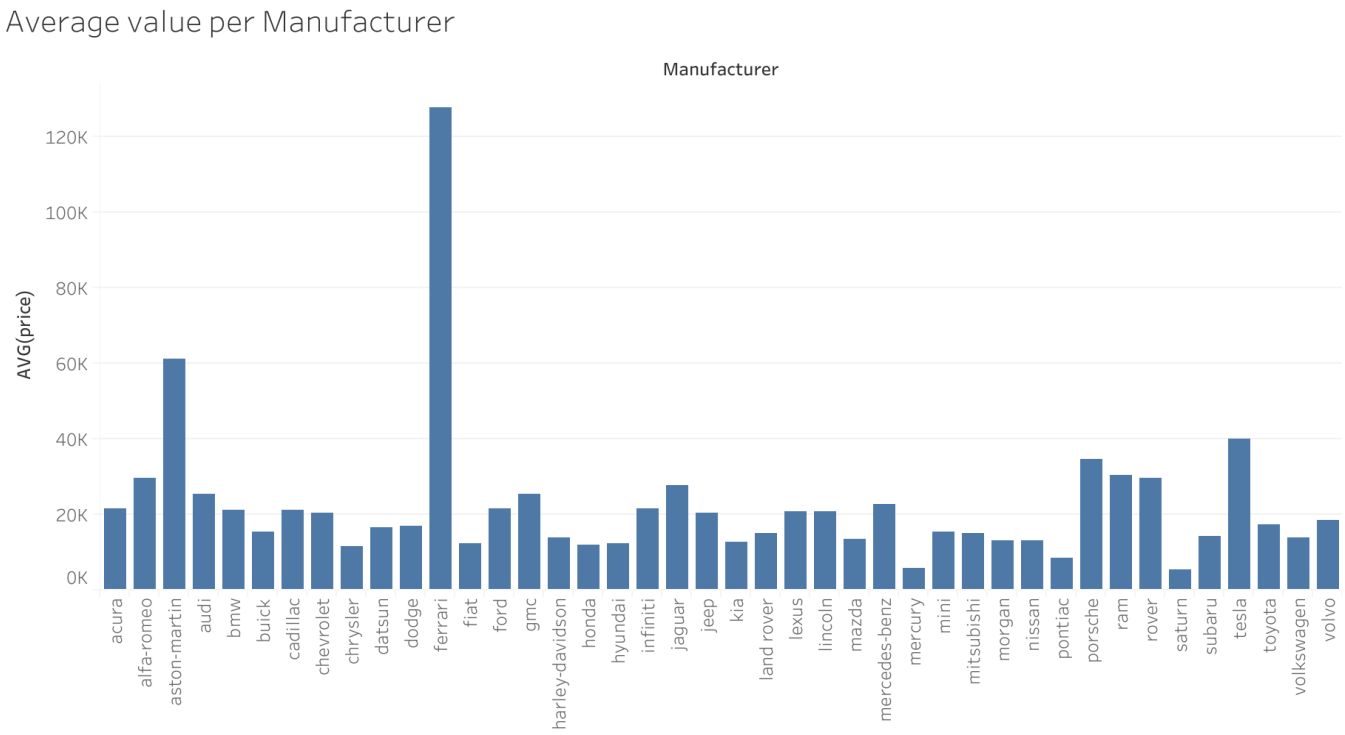
Επόμενο βήμα είναι ο καθαρισμός διπλότυπων τιμών στις στήλες της συλλογής δεδομένων. Η στήλη που μας ενδιαφέρει να μην περιέχει διπλότυπες τιμές είναι η στήλη που αποτελεί το Primary Key στη συλλογή δεδομένων, δηλαδή η στήλη με το *id* που περιλαμβάνει το μοναδικό αναγνωριστικό για κάθε εγγραφή. Πραγματοποιώντας τον έλεγχο διπλότυπων τιμών με τη βοήθεια του excel παρατηρούμε, πως δεν υπάρχει κάποια τιμή δύο ή περισσότερες φορές, οπότε η στήλη μας είναι καθαρή. Περαιτέρω έλεγχος για διπλότυπες τιμές στις υπόλοιπες στήλες δεν χρειάζεται, καθώς δεν επηρεάζονται από κάπου τα αποτελέσματα που θέλουμε να πάρουμε, αν κάποια δεδομένα είναι περασμένα παραπάνω από μία φορά στις στήλες *price, year και odometer*.

Σημαντικό βήμα στον καθαρισμό του αρχείου μας, είναι να ελέγξουμε αν τα δεδομένα που περιέχονται σε αυτό είναι ποιοτικά, αν δηλαδή αντιστοιχούν σε ρεαλιστικά δεδομένα και αν χρειάζεται να μην συμπεριλάβουμε μερικά από αυτά. Ξεκινώντας από τη στήλη *odometer* παρατηρούμε πως υπάρχουν τιμές μηδενικές, καθώς και τιμές που τείνουν στο μηδέν. Οι τιμές αυτές δεν θα πρέπει να περιλαμβάνονται, γιατί θα αλλοιώσουν τα τελικά αποτελέσματα. Θέτοντας ένα κατώτατο όριο τα 100 χιλιόμετρα, αφαιρούμε από το αρχείο μας συνολικά 3.767 πλειάδες. Με την ίδια λογική από τη στήλη *price* πρέπει να αφαιρέσουμε τυχόν μηδενικές τιμές ή τιμές που τείνουν στο μηδέν. Θέτοντας και εδώ ένα κατώτατο όριο τα 500 (USD) και ένα ανώτατο όριο τα 1.000.000 (USD), αφαιρούνται αυτές οι πλειάδες που περιέχουν τιμές εκτός των ορίων που θέσαμε και η τελική συλλογή δεδομένων που προκύπτει περιέχει 5 στήλες και 364.438 πλειάδες. Ποιοτικό έλεγχος τιμών γίνεται και στη στήλη *year*, χωρίς να αφαιρείται κάποιο δεδομένο, καθώς το εύρος τιμών το ορίζουμε από το 1900 αν πρόκειται για κάποια αντίκα, έως το 2022 χρονολογικά. Ο παραπάνω καθαρισμός έγινε με τη βοήθεια της Python και του Excel.

Αφού έχουν ολοκληρωθεί όλες οι διαδικασίες, όπως αναφέρθηκαν και παραπάνω παίρνουμε τη τελική συλλογή δεδομένων με ονομασία *vehicles_1.csv* και παρατηρούμε πως το μέγεθός του είναι 13,8 mb σε αντίθεση με το αρχικό που ήταν 1,34 Gb. Φορτώνοντας το τελικό .csv αρχείο στο MySQL Workbench παρατηρούμε πως φορτώνεται κανονικά χωρίς κανένα πρόβλημα, φέρνοντας όλες τις στήλες (5) και τις πλειάδες (364.438) που περιέχει. Επόμενο βήμα είναι να φτιάξουμε το αντίστοιχο query για τα αποτελέσματα που θέλουμε να εξαγάγουμε για τα συμπεράσματα που θέλουμε να βγάλουμε. Αρχικά θα πρέπει να βγάλουμε τον μέσο όρο κόστους για κάθε κατασκευαστή ξεχωριστά. Έτσι το αντίστοιχο query που φτιάχνουμε για αυτή την περίπτωση είναι το ακόλουθο:

- SELECT manufacturer, AVG(price) FROM vehicles.vehicles_1 GROUP BY manufacturer;

Μελετώντας το data grid που παίρνουμε από το παραπάνω query προσπαθούμε να αναπαραστήσουμε γραφικά τις μέσες τιμές ανάλογα με τον κατασκευαστή, όπως φαίνεται και στο παρακάτω σχήμα 4.10.



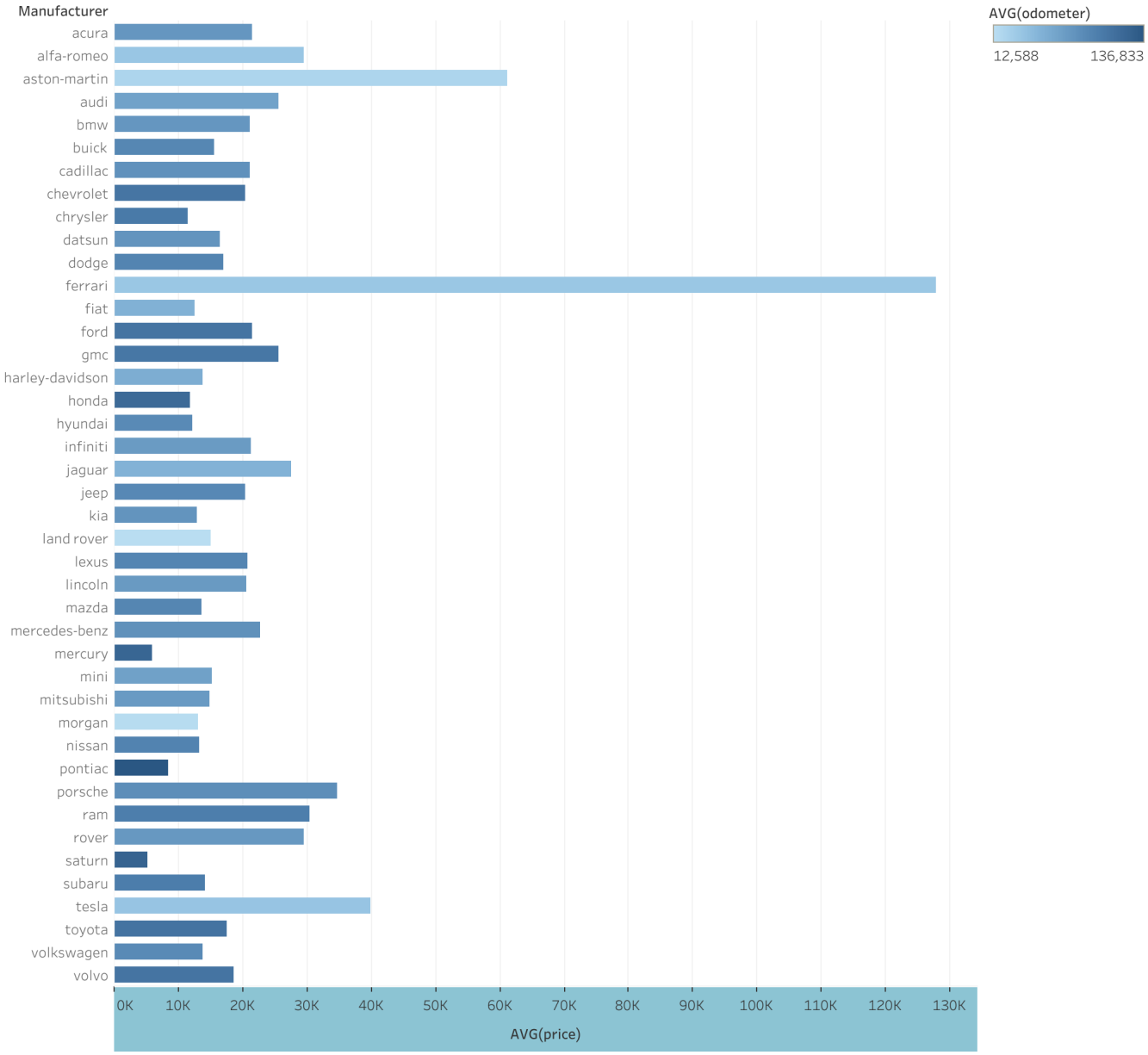
Σχήμα 4.10

Παρατηρούμε από την μελέτη της εικόνας και του data grid, πως για τις μάρκες αυτοκινήτων Ferrari, Aston – Martin και Tesla ο μέσος όρος της τιμής πώλησης είναι υψηλότερος σε σχέση με τις υπόλοιπους κατασκευαστές αυτοκινήτων. Παράλληλα παρατηρούμε πως οι χαμηλότερες τιμές αντιστοιχούν στους κατασκευαστές Saturn, Mercury και Pontiac.

Με τη βοήθεια της Python μπορούμε να φτιάξουμε το παρακάτω query για να μας φέρει σαν αποτελέσματα το μέσο όρο τιμής πώλησης σε σχέση με τον μέσο όρο χιλιομετρικής απόστασης ανά κατασκευαστή αυτοκινήτων:

- SELECT manufacturer, AVG(price), AVG(odometer) FROM vehicles.vehicles_1 group by manufacturer;

Αυτό το απεικονίσαμε στο παρακάτω σχήμα 4.11 με την βοήθεια του Tableau:



Σχήμα 4.11

5

Συμπεράσματα

Αν και η έρευνα για τον καθαρισμό δεδομένων έχει προχωρήσει, οι ακαδημαϊκές τεχνικές εξακολουθούν να χρησιμοποιούνται σπάνια σε πραγματικές καταστάσεις. Πρώτον, ενώ έχουν αρχίσει να εμφανίζονται έννοιες για ολοκληρωμένο καθαρισμό δεδομένων, η πλειονότητα των σημερινών λύσεων καθαρισμού δεδομένων εξακολουθεί να λαμβάνει υπόψη έναν τύπο ανακρίβειας κάθε φορά. Συνήθως δεν επαρκούν για να χειριστούν καταστάσεις του πραγματικού κόσμου, όπου τα σφάλματα αλληλοεπιδρούνε μεταξύ τους με περίπλοκους τρόπους. Η πλειονότητα των τεχνικών, δεύτερον, δεν κλιμακώνεται σωστά. Είτε χρειάζονται πολυάριθμα περάσματα σε ολόκληρο το σύνολο δεδομένων είτε έχουν τετραγωνική πολυπλοκότητα. Τρίτον, οι παράμετροι της πλειονότητας αυτών των εργαλείων είναι είτε ανύπαρκτες είτε εξαιρετικά δύσκολο να ρυθμιστούν. Για παράδειγμα, ο καθαρισμός βάσει κανόνων απαιτεί ένα ακριβό και περιεκτικό σύνολο ολοκληρωμένων ICs. Τέταρτον, υπάρχει συνήθως μια χωρική αποσύνδεση μεταξύ του σημείου δημιουργίας σφαλμάτων και του σημείου ανίχνευσης σφαλμάτων.

Μια αποτελεσματική λύση πρέπει να αποθηκεύει και να διαχειρίζεται περίπλοκες πληροφορίες προέλευσης σε ετερογενή συστήματα, καθώς, ενώ τα σφάλματα εντοπίζονται γρήγορα στα επόμενα στάδια κοντά στις τελικές αναφορές, οι πηγές δεδομένων χρειάζονται στην πραγματικότητα διόρθωση. Και τέλος, η πρόσληψη ανθρώπων κοστίζει ακριβά. Βασική προϋπόθεση για την εφαρμογή λύσεων καθαρισμού δεδομένων για τη λεπτομερή ρύθμιση, την επιβεβαίωση και την έγκριση διαφόρων αυτόνομων κρίσεων εξακολουθεί να είναι η συμπερίληψη ανθρώπων στη διαδικασία λήψης αποφάσεων (Ilyas and Chu, 2019).

Η έλλειψη εφαρμόσιμων και ισχυρών λύσεων καθαρισμού δεδομένων μπορεί να εξηγηθεί από αυτές τις δυσκολίες καθώς και από διάφορους άλλους πρακτικούς παράγοντες. Η απόκλιση μεταξύ των παραδοχών, της εγκατάστασης και της δυνατότητας εφαρμογής των μεθόδων καθαρισμού δεδομένων που χρησιμοποιούνται σε μεγάλες εταιρείες και των όσων δημοσιεύει η επιστημονική κοινότητα αυξάνεται καθημερινά. Οι Ilyas και Chu (2019) ισχυρίζονται πως μετά από εμπειρία συνεργασίας που είχαν με πραγματικούς πελάτες που χρειάζονταν τον καθαρισμό των δεδομένων τους, τα σενάρια των τμημάτων πληροφορικής με εκατομμύρια γραμμές κώδικα και ειδικούς για κάθε περίπτωση κανόνες επεξεργασίας είναι οι γνήσιοι αντίπαλοι των ακαδημαϊκών ιδεών. Το θετικό είναι ότι οι επιχειρήσεις αρχίζουν να αναγνωρίζουν την ανάγκη για κλιμακούμενες και ηθικές λύσεις καθαρισμού σε αντίθεση με τα εσωτερικά τους συστήματα που βασίζονται σε

κανόνες, τα οποία γίνονται όλο και πιο δύσκολα στη διαχείριση και δημιουργούν σημαντικό τεχνικό χρέος.

Εκτός από τις διάφορες λύσεις και μεθοδολογίες στον καθαρισμό δεδομένων είδαμε στην πράξη την κατάδειξη της σημασία του καθαρισμού των δεδομένων για τη διασφάλιση της ποιότητας και της αξιοπιστίας των συνόλων δεδομένων που χρησιμοποιούνται για ανάλυση. Μέσω της εφαρμογής διαφόρων τεχνικών καθαρισμού δεδομένων σε τρία διαφορετικά σύνολα δεδομένων που αφορούν εστιατόρια, την υπηρεσία μεταφοράς Uber και μεταχειρισμένα αυτοκίνητα, η παρούσα μελέτη έδειξε πώς τα σφάλματα, οι ασυνέπειες και οι ελλείπουσες τιμές μπορούν να επηρεάσουν σημαντικά τα αποτελέσματα της ανάλυσης δεδομένων. Με την εφαρμογή κατάλληλων μεθόδων καθαρισμού δεδομένων, οι ερευνητές μπορούν να μειώσουν τα σφάλματα, να βελτιώσουν την ακρίβεια των δεδομένων και να αυξήσουν την εμπιστοσύνη στα ευρήματα της ανάλυσής τους. Ως εκ τούτου, συνιστάται ο καθαρισμός των δεδομένων να θεωρηθεί ως βασικό βήμα στη διαδικασία ανάλυσης δεδομένων και οι ερευνητές θα πρέπει να δώσουν τη δέουσα προσοχή σε αυτό το στάδιο για να επιτύχουν ισχυρά και ακριβή αποτελέσματα.

Βιβλιογραφία

Abedjan, Z., Chu, X., Deng, D., Fernandez, R. C., Ilyas, I. F., Ouzzani, M., Papotti, P., Stonebraker, M., & Tang, N. (2016a). Detecting data errors. *Proceedings of the VLDB Endowment*, *9*(12), 993–1004. https://doi.org/10.14778/2994509.2994518

Abedjan, Z., Chu, X., Deng, D., Fernandez, R. C., Ilyas, I. F., Ouzzani, M., Papotti, P., Stonebraker, M., & Tang, N. (2016b). Detecting data errors. *Proceedings of the VLDB Endowment*, *9*(12), 993–1004. https://doi.org/10.14778/2994509.2994518

Aggarwal, C. C. (2012). An Introduction to Outlier Analysis. *Outlier Analysis*, 1–40. https://doi.org/10.1007/978-1-4614-6396-2_1

AnHai Doan, Ying Lu, Yoonkyong Lee, & Jiawei Han. (2003). Profile-based object matching for information integration. *IEEE Intelligent Systems*, *18*(5), 54–59. https://doi.org/10.1109/mis.2003.1234770

Arasu, A., Chaudhuri, S., & Kaushik, R. (2009). Learning string transformations from examples. *Proceedings of the VLDB Endowment*, *2*(1), 514–525. https://doi.org/10.14778/1687627.1687686

Bard, G. (2007). *Spelling-Error Tolerant, Order-Independent Pass-Phrases via the Damerau-Levenshtein String-Edit Distance Metric*. Retrieved January 23, 2023, from https://eprint.iacr.org/2006/364.pdf

Barnett, V. & Lewis, T. (1994). *Outliers in Statistical Data*. Wiley, New York.

Baudinet, M., Chomicki, J., & Wolper, P. (1999). Constraint-Generating Dependencies. *Journal of Computer and System Sciences*, *59*(1), 94–115. https://doi.org/10.1006/jcss.1999.1632

Bilenko, M., & Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '03*. https://doi.org/10.1145/956750.956759

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. *ACM SIGMOD Record*, *29*(2), 93–104. https://doi.org/10.1145/335191.335388

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection. *ACM Computing Surveys*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Chaudhuri, S., Ganti, V., & Motwani, R. (2005). *Robust identification of fuzzy duplicates*. IEEE Xplore. https://doi.org/10.1109/ICDE.2005.125

Chaudhuri, S. Chen, B.-C., Ganti, V., & Kaushik, R. (2007). Example-driven design of efficient record matching queries. *Proceedings of the 33rd International Conference on Very Large Data Bases*, pp. 327–338

Chiang, F., & Miller, R. J. (2008). Discovering data quality rules. *Proceedings of the VLDB Endowment*, *1*(1), 1166–1177. https://doi.org/10.14778/1453856.1453980

Chu, X., Ilyas, I. F., & Papotti, P. (2013a). Discovering denial constraints. *Proceedings of the VLDB Endowment*, *6*(13), 1498–1509. https://doi.org/10.14778/2536258.2536262

Chu, X. Ilyas, I. F., & Papotti, P. (2013b). Holistic data cleaning: Putting violations into context. *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. https://doi.org/10.1109/icde.2013.6544847

Chu, X., Ilyas, I. F. (2017). *Scalable and holistic qualitative data cleaning* (dissertation).

Dallachiesa, M., Ebaid, A., Eldawy, A., Elmagarmid, A., Ilyas, I. F., Ouzzani, M., & Tang, N. (2013). NADEEF. *Proceedings of the 2013 International Conference on Management of Data - SIGMOD '13*. https://doi.org/10.1145/2463676.2465327

De Stefano, C., Sansone, C., & Vento, M. (2000). To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man and Cybernetics, Part c (Applications and Reviews)*, *30*(1), 84–94. https://doi.org/10.1109/5326.827457

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Eckerson, W. W. (2002). (rep.). *Data quality and the bottom line: Achieving business success through a commitment to high quality data* (pp. 1–36). Chatsworth, CA: The Data Warehousing Institute.

Elsner, M., & Schudy, W. (2009). Bounding and comparing methods for correlation clustering beyond ILP. *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing - ILP '09*. https://doi.org/10.3115/1611638.1611641

Fan, W., Li, J., Ma, S., Tang, N., & Yu, W. (2011). Interaction between record matching and data repairing. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. https://doi.org/10.1145/1989323.1989373

Fan, W., & Geerts, F. (2012). Foundations of Data Quality Management. *Synthesis Lectures on Data Management*, *4*(5), 1–217. https://doi.org/10.2200/s00439ed1v01y201207dtm030

Fan, W., Geerts, F., Ma, S., Tang, N., & Yu, W. (2013). Data quality problems beyond consistency and deduplication. *In Search of Elegance in the Theory and Practice of Computation*, 237–249. https://doi.org/10.1007/978-3-642-41660-6_12

Fagin, R., Kolaitis, P. G., Miller, R. J., & Popa, L. (2002). Data Exchange: Semantics and Query Answering. *Lecture Notes in Computer Science*, 207–224. https://doi.org/10.1007/3-540-36285-1_14

Fellegi, I. P., & Sunter, A. B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*, *64*(328), 1183–1210. https://doi.org/10.1080/01621459.1969.10501049

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning* , volume 1. Springer.

Gadd, T. N. (1990). PHONIX: The algorithm. *Program: electronic library and information systems*, *24*(4), 363–366. https://doi.org/10.1108/eb047069

Geerts, F., Mecca, G., Papotti, P., & Santoro, D. (2013). The LLUNATIC data-cleaning framework. *Proceedings of the VLDB Endowment*, *6*(9), 625–636. https://doi.org/10.14778/2536360.2536363

Gladitz, J. (1988). Barnett, V. & Lewis, T.: Outliers in statistical data, 2nd ed., John Wiley & Sons, Chi-Chester – New York – Brisbane – Toronto – Singapore, 1984, XIV, 463 S., 26 abb., £ 29.95, ISBN 0471905070. *Biometrical Journal*, *30*(7), 866–867. https://doi.org/10.1002/bimj.4710300725

Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*, *11*(1), 1–21. https://doi.org/10.1080/00401706.1969.10490657

Gulwani, S. (2011). Automating string processing in spreadsheets using input-output examples. *ACM SIGPLAN Notices*, *46*(1), 317. https://doi.org/10.1145/1925844.1926423

Guo, P. J., Kandel, S., Hellerstein, J. M., & Heer, J. (2011). Proactive wrangling. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. https://doi.org/10.1145/2047196.2047205

Hawkins D. (1980). Identification of outliers (monographs on statistics and applied probability). 1st ed. Netherlands: Springer.

Hawkins, S., He, H., Williams, G., & Baxter, R. (2002). Outlier Detection Using Replicator Neural Networks. *Data Warehousing and Knowledge Discovery*, 170–180. https://doi.org/10.1007/3-540-46145-0_17

Heer, J., Hellerstein, J.M., & Kandel, S. (2015). Predictive Interaction for Data Transformation. *Conference on Innovative Data Systems Research*.

Hernández, M. A., & Stolfo, S. J. (1995). The merge/purge problem for large databases. *ACM SIGMOD Record*, *24*(2), 127–138. https://doi.org/10.1145/568271.223807

Hodge, V., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, *22*(2), 85–126. https://doi.org/10.1023/b:aire.0000045502.10941.a9

Ilyas, I. F., & Xu, C. (2019). *Data Cleaning*. Association for Computing Machinery.

Jin, W., Tung, A. K. H., Han, J., & Wang, W. (2006). Ranking Outliers Using Symmetric Neighborhood Relationship. *Advances in Knowledge Discovery and Data Mining*, 577–593. https://doi.org/10.1007/11731139_68

Jin, Z., Anderson, M.R., Cafarella, M.J., & Jagadish, H.V. (2017). Foofah: Transforming Data By Example. *Proceedings of the 2017 ACM International Conference on Management of Data*.

Kandel, S., Paepcke, A., Hellerstein, J., & Heer, J. (2011). Wrangler: Interactive visual specification of data transformation scripts. *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11*. https://doi.org/10.1145/1978942.1979444

Knorr, E. M. & Ng, R. T. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets. In A. Gupta, O. Shmueli & J. Widom (eds.), *VLDB* (p./pp. 392-403), : Morgan Kaufmann. ISBN: 1-55860-566-5

Knorr, E.M., & Ng, R.T. (1999). Finding Intensional Knowledge of Distance-Based Outliers. *Very Large Data Bases Conference*.

Kolahi, S., & Lakshmanan, L. V. S. (2009). On approximating optimum repairs for functional dependency violations. *Proceedings of the 12th International Conference on Database Theory*. https://doi.org/10.1145/1514894.1514901

Meliou, A., Gatterbauer, W., Nath, S., & Suciu, D. (2011b). Tracing data errors with view-conditioned causality. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. https://doi.org/10.1145/1989323.1989376

Monge, Alvaro E. & Elkan, Charles P. (1996). The field matching problem: Algorithms and applications. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96).* AAAI Press, 267–270.

Newcombe, H. B., Kennedy, J. M., Axford, S. J., & James, A. P. (1959). Automatic Linkage of Vital Records. *Science*, *130*(3381), 954–959. https://doi.org/10.1126/science.130.3381.954

Rahm, E., & Do, H. H. (2000). Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull.*, *23*(4), 3–13.

Raman, V., & Hellerstein, J.M. (2001). Potter's Wheel: An Interactive Data Cleaning System. *Very Large Data Bases Conference*.

Rekatsinas, T., Chu, X., Ilyas, I. F., & Ré, C. (2017a). HoloClean. *Proceedings of the VLDB Endowment*, *10*(11), 1190–1201. https://doi.org/10.14778/3137628.3137631

Rekatsinas, T., Joglekar, M., Garcia-Molina, H., Parameswaran, A., & Ré, C. (2017b). SLiMFast: Guaranteed results for data fusion and source reliability. *Proceedings of the 2017 ACM International Conference on Management of Data*. https://doi.org/10.1145/3035918.3035951

Sarawagi, S., & Bhamidipaty, A. (2002). Interactive deduplication using active learning. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '02*. https://doi.org/10.1145/775047.775087

Singh, R., & Gulwani, S. (2012). Learning semantic string transformations from examples. *Proceedings of the VLDB Endowment*, *5*(8), 740–751. https://doi.org/10.14778/2212351.2212356

Steuart, B.W. and Precision Indexing Staff (1994). The Daitch-Mokotoff Soundex Reference Guide. Heritage Quest.

Stonebraker, M., Bruckner, D., Ilyas, I. F., Beskales, G., Cherniak, M., Zdonik, S. B., Pagan, A., & Xu, S. (2013). Data curation at scale: The data tamer system. [Review of *Data curation at scale: The data tamer system.*]. In *In Proceedings of the 6th Biennial Conf. on Innovative Data Systems Research.*

Stonebraker, M., & Ilyas, I.F. (2018). Data Integration: The Current Status and the Way Forward. *IEEE Data Eng. Bull., 41*, 3-9.

Tibbets, H. (11, September 11). *Dirty data costs the US economy $3.1 trillion yearly*. Newswire. Retrieved January 03, 2023, from https://www.newswire.com/news/dirty-data-costs-the-us-economy-3-1-trillion-yearly-93846.

Tietjen, G. L., & Moore, R. H. (1972). Some Grubbs-Type Statistics for the Detection of Several Outliers. *Technometrics*, *14*(3), 583–597. https://doi.org/10.1080/00401706.1972.10488948

Van den Broeck, J., Argeseanu Cunningham, S., Eeckels, R., & Herbst, K. (2005). Data Cleaning: Detecting, diagnosing, and editing data abnormalities. *PLoS Medicine*, *2*(10). https://doi.org/10.1371/journal.pmed.0020267

Konda, P., Das, S., C., P. S., Doan, A. H., Ardalan, A., Ballard, J. R., Li, H., Panahi, F., Zhang, H., Naughton, J., Prasad, S., Krishnan, G., Deep, R., & Raghavendra, V. (2016). Magellan. *Proceedings of the VLDB Endowment*, *9*(13), 1581–1584. https://doi.org/10.14778/3007263.3007314

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals.. *Soviet Physics Doklady*, 10, 707--710.

Weis, M., Naumann, F., Jehle, U., Lufter, J., & Schuster, H. (2008). Industry-scale duplicate detection. *Proceedings of the VLDB Endowment*, *1*(2), 1253–1264. https://doi.org/10.14778/1454159.1454165

Wikipedia Contributors. (2018, December 1). *Car*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Car

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *ERIC*. https://eric.ed.gov/?id=ED325505

Winkler, W. E. (1993). *Improved decision rules in the Fellegi-Sunter model of record linkage*. Bureau of the Census. Retrieved January 03, 2023, from http://www.asasrms.org/Proceedings/papers/1993_042.pdf

Winkler, W.E. (1999). The State of Record Linkage and Current Research Problems. *Statistical Research Division,* U.S. Census Bureau. Retrieved January 03, 2023, from https://www.census.gov/library/working-papers/1999/adrm/rr99-04.html

Wu, E., Madden, S., & Stonebraker, M. (2012). A demonstration of DBWipes. *Proceedings of the VLDB Endowment*, *5*(12), 1894–1897. https://doi.org/10.14778/2367502.2367531