

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΉΣ

Μελέτη μεγάλων αρχείων δεδομένων και εκπόνηση καθαρισμού αυτών μέσω διαφορετικών framework **στην** Python

Πτυχιακή εργασία

Κωνσταντίνος Κοντογιάννης

Αθήνα, Σεπτέμβριος 2022



HAROKOPIO UNIVERSITY

SCHOOL OF DIGITAL TECHNOLOGY

DEPARTMENT OF INFORMATICS AND TELEMATICS

Large data files analysis and cleaning using frameworks in Python

Bachelor thesis

Konstantinos Kontogiannis

Athens, September 2022



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΉΣ

Τριμελής Εξεταστική Επιτροπή

Ιωάννης Βιόλος (Επιβλέπων)
Ερευνητής, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο

Κωνσταντίνος Τσερπές
Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο

Ηρακλής Βαρλάμης
Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο

Ο Κωνσταντίνος Κοντογιάννης

δηλώνω υπεύθυνα ότι:

1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

3) Όπου υφίστανται δικαιώματα άλλων δημιουργών έχουν διασφαλιστεί όλες οι αναγκαίες άδειες χρήσης ενώ το αντίστοιχο υλικό είναι ευδιάκριτο στην υποβληθείσα εργασία.

Περίληψη

Στην εποχή μας η ραγδαία αύξηση της ποιότητας και της ποσότητας των τεχνολογικών μέσων έχει φέρει την επιστήμη των υπολογιστών σε περίοπτη θέση. Όλοι οι επιστημονικοί κλάδοι πρέπει να πλησιάσουν όσο πιο κοντά γίνεται στην επιστήμη των υπολογιστών για να αναπτυχθούν πιο γρήγορα και πιο αποτελεσματικά. Αυτό συμβαίνει διότι ένας άνθρωπος χρησιμοποιεί πολλές ηλεκτρονικές συσκευές μέσα στην μέρα του, γεγονός που έχει ως αποτέλεσμα την μεγάλη παραγωγή δεδομένων. Το γεγονός αυτό έχει δημιουργήσει την τάση των μεγάλων δεδομένων, μια φράση που ακούγεται πολύ τα τελευταία πέντε χρόνια. Η υπερβάλλουσα επιθυμία οργανισμών, εταιρειών και κυβερνήσεων να αναλύσουν όλα αυτά τα δεδομένα, δημιούργησε την επιστήμη των δεδομένων. Η επιστήμη αυτή δεν αποτελείται από ένα αντικείμενο εργασιών, αλλά μοιάζει περισσότερο με μια γραμμή παραγωγής ενός εργοστασίου, η οποία ξεκινάει από την ακατέργαστη πρώτη ύλη και φτάνει στο τελικό προϊόν. Ίσως το πιο σημαντικό κομμάτι στην επιστήμη των δεδομένων να αποτελεί ο καθαρισμός δεδομένων ο οποίος δημιουργεί τις βάσεις για μια καλή ανάλυση δεδομένων, εκμάθηση μοντέλου ή εξόρυξη γνώσης. Ο κύριος στόχος του καθαρισμού των δεδομένων είναι να παρέχει στο επόμενο βήμα δεδομένα υψηλότερης ποιότητας από αυτά που έλαβε. Η επιστήμη των δεδομένων βασίζεται στον προγραμματισμό μέσω γλωσσών υψηλού επιπέδου. Μια από τις πιο γνωστές είναι Python η οποία ταιριάζει πολύ καλά στις διαδικασίες που εμπεριέχουν επεξεργασία δεδομένων, διότι διαθέτει μεγάλο αριθμό βιβλιοθηκών που βοηθούν στην διεκπεραίωσή του. Σε αυτήν την εργασία θα γίνει μια ανάλυση ως προς το καινούργιο πεδίο της επιστήμης των δεδομένων που αφορά στα Big Data, ενώ θα αναλυθούν διεξοδικά οι ορισμοί της ποιότητας των δεδομένων και της διαδικασίας καθαρισμού δεδομένων. Τέλος, θα υλοποιηθούν με την σειρά οι εργασίες καθαρισμού σε ένα σετ δεδομένων, με διαφορετικά modules με σκοπό την σύγκρισή τους.

Λέξεις κλειδιά: Ποιότητα Δεδομένων, Καθαρισμός Δεδομένων, βιβλιοθήκες της Python

Abstract

Nowadays the rapid growth of the quality and quantity of the digital tools has brought computer science to a prominent position. All scientific disciplines need to get as close as possible to computer science to develop faster and more efficiently. This is explained by the fact that a man uses many electronic devices during the day and as a result they produce great amount of data. In this way, the term 'big data' has derived, and this is a term that has been heard a lot during the last years. The intense desire of organizations, companies, and governments to analyze all these data led to the nascence of computer data science. The data science does not consist of a unique remit but is more like a production line of a factory which starts working with the raw material and then reaches the finished product. Maybe the most important field in data science is the data cleaning which is fundamental for a qualitative data analysis, a machine learning, and a data mining . The main goal of data cleaning is to provide a higher quality of data than the one it received. The data science bases on the programming via a wide range of high-level languages. One of the most known is Python which fits very well on the procedures regarding the data processing, as it contains a great number of libraries that help in its processing. In this paper, an analysis will be made regarding the new field of data science related to Big Data. Definitions of data quality and the data cleaning process will also be thoroughly discussed. Finally, the cleaning tasks will be implemented in order, on a set of data with different modules to compare them quantitively and qualitatively.

Keywords: Data cleaning, Data Quality, Python modules

Περιεχόμενα

ΠΕΡΙΕΧΟΜΕΝΑ 7

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ 10

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ 11

1. **ΕΙΣΑΓΩΓΗ** 13

1.1 **Περιγραφη Μελετης-Προβληματος** 13
1.2 **Αντικειμενο-Σκοπος Εργασιας** 13

2. **ΤΙ ΕΙΝΑΙ ΤΑ 'ΜΕΓΑΛΑ' ΔΕΔΟΜΕΝΑ** (BIG DATA) 15

2.1 **Ορισμος** 15
2.2 **Χαρακτηριστικα** 16
2.3 **Τυποι** 19
2.4 **Πεδια εφαρμογης** 21

3. **ΠΟΙΟΤΗΤΑ ΔΕΔΟΜΕΝΩΝ** 25

3.1 **Ορισμος** 25
3.2 **Κατηγοριες Διαστασεων Ποιοτητας Δεδομενων** 25
3.3 **Διστασεις Ποιοτητας Δεδομενων** 31
3.4 **Στρατηγικες Βελτιωσης Ποιοτητας Δεδομενων** 40

4. **ΚΑΘΑΡΙΣΜΟΣ ΔΕΔΟΜΕΝΩΝ** 43

4.1 Outliers (**Έκτοπες τιμες**) 43
4.1.1 **Τυποι** Outlier 44
4.1.2 **Κατηγοριες Ανιχνευσης** Outlier 45
4.1.3 **Τεχνικες Ανιχνευσης** Outlier 46
4.2 Missing Data (**Ελλιπη Δεδομενα**) 51
4.2.1 **Κατηγοριες Ελλιπων Δεδομενων** 51
4.2.2 **Προσεγγισεις αντιμετωπισης ελλιπων δεδομενων** 52
4.3 Duplicate Data (**Διπλοτυπες εγγραφες δεδομενων**) 57
4.3.1 **Τυποι διπλοτυπων δεδομενων** 57
4.3.2 **Κατηγοριες προσεγγισης προβληματος** 59

4.3.3 **ΤΥΠΟΙ ΑΠΟΔΙΠΛΑΣΙΑΣΜΟΥ ΠΛΗΡΟΦΟΡΙΩΝ ΣΤΟ** CLOUD COMPUTING....62

5. PYTHON FRAMEWORKS **ΚΑΤΑΛΛΗΛΑ ΓΙΑ** DATA CLEANING....65

5.1 MISSINGNO....66
5.2 DATACLEANER....66
5.3 DORA....67
5.4 SCRUBADUB....68
5.5 ARROW....69
5.6 DATAPREP....70
5.7 SEABORN....71
5.8 FUZZYWUZZY....72
5.9 PYOD....72
5.10 STATSMODELS....73
5.11 OUTLIER-UTILS....73

6. **ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ**....75

6.1 **ΥΛΟΠΟΙΗΣΗ** DATA OVERVIEW....77
6.1.1 PANDAS,MATPLOTLIB,MISSINGO....77
6.1.2 DATAPREP.EDA....82
6.1.3 **ΣΥΓΚΡΙΣΗ**....85
6.2 **ΥΛΟΠΟΙΗΣΗ** FIX DATA SHAPE....85
6.2.1 **ΜΟΡΦΟΠΟΙΗΣΗ ΟΝΟΜΑΤΩΝ ΚΑΙ ΤΥΠΩΝ ΓΝΩΡΙΣΜΑΤΩΝ**....86
6.2.1.1 Pandas....86
6.2.1.2 dataprep.clean....87
6.2.2 **ΜΟΡΦΟΠΟΙΗΣΗ ΣΥΝΤΕΤΑΓΜΕΝΩΝ**....88
6.2.2.1 Geopandas....88
6.2.2.2 dataprep.clean....88
6.2.3 **ΜΟΡΦΟΠΟΙΗΣΗ ΗΜΕΡΟΜΗΝΙΑΣ**....89
6.2.3.1 Pandas....89
6.2.3.2 Arrow....90
6.2.3.3 dataprep.clean....91
6.2.4 **ΣΥΓΚΡΙΣΗ**....92
6.3 **ΥΛΟΠΟΙΗΣΗ** DEDUPLICATION OF DATA....94
6.3.1 PANDAS....94
6.3.2 CROSS CHAIN ALGORITHM WITH FUZZYWUZZY....94
6.3.3 DATAPREP.CLEAN....97
6.3.4 **ΣΥΓΚΡΙΣΗ**....97
6.4 REMOVE IRRELEVANT DATA....98
6.4.1 FILTERING **ΜΕ** PANDAS....98
6.4.2 FILTERING **ΜΕ** NUMPY....99
6.4.3 **ΣΥΓΚΡΙΣΗ**....100
6.5 DEAL WITH MISSING VALUES....100

6.5.1 MISSINGNO 100
6.5.2 DATACLEANER 102
6.5.3 PANDAS 103
6.5.4 **ΣΥΓΚΡΙΣΗ** 104
6.6 DETECTION OF DATA OUTLIERS 105
6.6.1 STATSMODELS-SEABORN 105
6.6.2 PYOD 106
6.6.3 OUTLIER-UTILS 108
6.6.4 **ΣΥΓΚΡΙΣΗ** 108

ΒΙΒΛΙΟΓΡΑΦΙΑ 110

Ευρετήριο Πινάκων

Πίνακας 1: Data overview, σύγκριση των modules. 85
Πίνακας 2: Fix Data shape, σύγκριση των modules 1..... 92
Πίνακας 3: Fix Data shape, σύγκριση των modules 2..... 93
Πίνακας 4: Fix Data shape, σύγκριση των modules 3 93
Πίνακας 5:Deduplication of data, σύγκριση των modules 98
Πίνακας 6: Remove irrelevant data, σύγκριση των modules..... 100
Πίνακας 7: Deal with missing data, σύγκριση των modules..... 104
Πίνακας 8: Detection of data outliers, σύγκριση των modules 109

Ευρετήριο Εικόνων

Εικόνα 1:Ορισμοί Big Data σύμφωνα με μια διαδικτυακή έρευνα που πήραν μέρος 154 στελέχη εταιρειών το 2012 από την εταιρεία SAP..... 16
Εικόνα 2: Κατηγορίες Δεδομένων..... 21
Εικόνα 3:Κοινά δημοσιευμένες διαστάσεις ποιότητας. Πηγή: (Wang, Storey, & Firth, A framework for analysis of quality research, 1995) 29
Εικόνα 4 Δενδρόγραμμα με τις διαστάσεις ποιότητας δεδομένων, Πηγή: (Mahanti, 2019).. 40
Εικόνα 5: Scatterplot με έκτοπη τιμή. Πηγή: Scatterplots (2 of 5) | Concepts in Statistics | | Course Hero 44
Εικόνα 6: Collective outliers Πηγή: (Babu & Divya, 016)..... 45
Εικόνα 7: : Φυσιολογικό σημείο - Ακραίο σημείο Πηγή: Outlier Detection with Isolation Forest | by Eryk Lewinson | Towards Data Science..... 49
Εικόνα 8: Παράδειγμα διαδικασίας ανίχνευσης έκτοπων τιμών μέσω autoencoder Πηγή: : (Mishra, 5 Outlier Detection Techniques that every “Data Enthusiast” Must Know, 2021) ... 50
Εικόνα 9: Παράδειγμα πίνακα με διπλότυπες στήλες 58
Εικόνα 10: Παράδειγμα πίνακα με παρόμοιες πλειάδες 58
Εικόνα 11: Προεπισκόπηση δεδομένων του dataframe μας..... 77
Εικόνα 12: εφαρμογή συνάρτησης info() 78
Εικόνα 13: αλλαγή format προβολής float τιμών και εφαρμογή συνάρτησης describe() με τα αποτελέσματα της δεύτερης..... 78
Εικόνα 14: εφαρμογή συνδυασμού των συναρτήσεων isnull() και sum() στα δεδομένα μας79
Εικόνα 15: Δεύτερο μέρος αποτελεσμάτων της συνάρτησης..... 79
Εικόνα 16: εφαρμογή συνάρτησης value_counts() και το πρώτος μέρος των αποτελεσμάτων 79
Εικόνα 17: Σχεδιάγραμμα της Pandas για την καταμέτρηση των 10 συχνότερων πολιτειών 80
Εικόνα 18: Εφαρμογή της συνάρτησης duplicated για την εύρεση διπλότυπων και προβολή αυτών..... 80
Εικόνα 19: : Ιστόγραμμα που παρουσιάζει την κατανομή των αριθμών ‘Total Death’..... 81
Εικόνα 20: Σχεδιάγραμμα 10 πιο συχνών FIPS Code στα δεδομένα μας..... 81
Εικόνα 21: : Σχεδιάγραμμα 10 πιο συχνών συντεταγμένων στα δεδομένα μας 81
Εικόνα 22: Εφαρμογή συνάρτηση matrix και αποτέλεσμα αυτής..... 82
Εικόνα 23: Εφαρμογή correlation σε συνδυασμό με την οπτικοποίηση των αποτελεσμάτων του σε σχεδιάγραμμα 82
Εικόνα 24: Overview πίνακας αποτελεσμάτων συνάρτησης create_report, τμήμα 1 83
Εικόνα 25:Variables πίνακας αποτελέσματα και γραφήματα συνάρτησης create_report, τμήμα 2 83
Εικόνα 26: Interactions γράφημα συνάρτησης create_report, τμήμα 3 84
Εικόνα 27: Correlations γράφημα συνάρτησης create_report, τμήμα 4..... 84
Εικόνα 28::Variables πίνακας αποτελέσματα και γραφήματα συνάρτησης create_report, τμήμα 2 84
Εικόνα 29: Μετονομασία στηλων και μετατροπή τύπων στήλης με Pandas 86
Εικόνα 30: Μετονομασία στηλων και μετατροπή τύπων στήλης με dataprep 87
Εικόνα 31: Δημιουργία geodataframe 88

Εικόνα 32: Διαχωρισμός συντεταγμένων μέσω dataprep 89
Εικόνα 33: Μορφοποίηση ημερομηνίας με Pandas 90
Εικόνα 34: Μορφοποίηση ημερομηνιών με Arrow 90
Εικόνα 35: Μορφοποίηση ημερομηνίας με την χρήση dataprep 91
Εικόνα 36: Τελική εικόνα του σετ δεδομένων μετά από το βήμα μορφοποίησης 91
Εικόνα 37: Σφάλμα pandas με την downcasting 92
Εικόνα 38: Εφαρμογή drop_duplicates() στο σετ δεδομένων covid_df 94
Εικόνα 39: δεύτερο μέρος αλγόριθμου cross chain 95
Εικόνα 40:πρώτο μέρος αλγόριθμου cross chain 95
Εικόνα 41: τρίτο μέρος αλγόριθμου cross chain μαζί με Fuzzywuzzy βιβλιοθήκη 96
Εικόνα 42: Εφαρμογή αλγορίθμου στα dataset covid_df 96
Εικόνα 43: Έλεγχος αποδιπλασιασμού δεδομένων με την βοήθεια της pandas 96
Εικόνα 44: Αποδιπλασιασμός dataset με την χρήση της dataprep.clean 97
Εικόνα 45: Φιλτράρισμα δεδομένων με την χρήση Pandas συναρτήσεων 99
Εικόνα 46: Φιλτράρισμα δεδομένων με την χρήση NumPy συναρτήσεων 99
Εικόνα 47: Διάγραμμα matrix της missingno για ελλιπείς τιμές 101
Εικόνα 48: Διάγραμμα correlation της missingno 101
Εικόνα 49: Καθαρισμός ελλιπών τιμών με datacleaner 102
Εικόνα 50: Εμφάνιση των πλειάδων με null από το covid_df 102
Εικόνα 51: Προβολή των ίδιων πλειάδων με εικόνα 50, μετά την εφαρμογή datacleaner στην covid_df 103
Εικόνα 52: Καθαρισμός σετ δεδομένων με Pandas 103
Εικόνα 53: Προβολή ελλιπών τιμών dataset πριν και μετά την χρήση pandas 104
Εικόνα 54: Οπτική ανίχνευση outliers συντεταγμένων με την χρήση statsmodels και seaborn βιβλιοθηκών 105
Εικόνα 55: Οπτική ανίχνευση outliers αριθμητικών στηλών ('TotalDeath' και 'TotalConfrimed) με την χρήση statsmodels και seaborn βιβλιοθηκών 106
Εικόνα 56: Επανεκτέλεση με την χρήση contamination παραμέτρου 107
Εικόνα 57: Εφαρμογή KNN αλγόριθμου για την εύρεση έκτοπων τιμών 107
Εικόνα 58: Εφαρμογής βιβλιοθήκη Outlier-utils για την εύρεση έκτοπων τιμών ανά στήλη 108

1. Εισαγωγή

1.1 Περιγραφή Μελέτης-Προβλήματος

Παρατηρώντας την τεράστια αύξηση του όγκου των δεδομένων, καθώς και τον καταλυτικό ρόλο τους στην ανάπτυξη όλων των επιστημονικών πεδίων είναι εύκολο να αντιληφθεί κανείς την σπουδαιότητα της διαδικασίας καθαρισμού των δεδομένων. Λόγω του τεράστιου όγκου δεδομένων που συλλέγεται καθημερινά ανακύπτει ως επιτακτική ανάγκη η ανάλυση τους με όσο το δυνατόν πιο αποτελεσματικά και γρήγορο τρόπο γίνεται. Ίσως το μεγαλύτερο εμπόδιο στην ανάλυσή τους είναι η ποιότητα των δεδομένων την στιγμή που συλλέγονται, καθώς χωρίς την κατάλληλη διαδικασία καθαρισμού ενδέχεται να βρίσκεται σε χαμηλά επίπεδα. Αυτό είναι πιθανό να έχει καταστροφικά αποτελέσματα στις αναλύσεις των δεδομένων αυτών.

Στην παρούσα εργασία συλλέγονται όλες οι απαραίτητες πληροφορίες που θα πρέπει να έχει ένας επιστήμονας των δεδομένων πριν ξεκινήσει την διαδικασία τους καθαρισμού ενός σετ δεδομένων. Επίσης, αναφέρεται αναλυτικά ο τρόπος καθορισμού της ποιότητας της πληροφορίας που περιέχει ένα σετ δεδομένων. Τέλος, υλοποιείται μια ολοκληρωμένη διαδικασία καθαρισμού ενός σετ δεδομένων με διαφορετικές βιβλιοθήκες της γλώσσας προγραμματισμού Python, καθώς και η αξιολόγηση αυτών ως προς τις λειτουργίες που προσφέρουν στον χρήστη.

1.2 Αντικείμενο-Σκοπός Εργασίας

Αντικείμενο της εργασίας αυτής είναι η επισκόπηση των μεγάλων δεδομένων ως προς τα χαρακτηριστικά, τους τύπους και τα πεδία που εφαρμόζονται. Επιπλέον, έχει σκοπό την συλλογή όσο το δυνατόν περισσότερων πληροφοριών που αφορούν στην ποιότητα των δεδομένων, όπως είναι οι διαστάσεις που την ορίζουν και οι προσεγγίσεις βελτίωσης τους. Επιπρόσθετα, παρουσιάζονται σε θεωρητικό επίπεδο τα τρία βασικότερα αντικείμενα που απασχολούν τον τομέα του καθαρισμού καθώς και οι προσεγγίσεις που χρησιμοποιούνται για την αντιμετώπισή τους. Στο πρακτικό κομμάτι της εργασίας εφαρμόζονται όσο το δυνατόν περισσότερες βιβλιοθήκες της Python που στοχεύουν στον καθαρισμό ενός αρχείου

με δεδομένα, αλλά και στην σύγκριση και αξιολόγηση αυτών ως προς την λειτουργικότητα και την ταχύτητά τους.

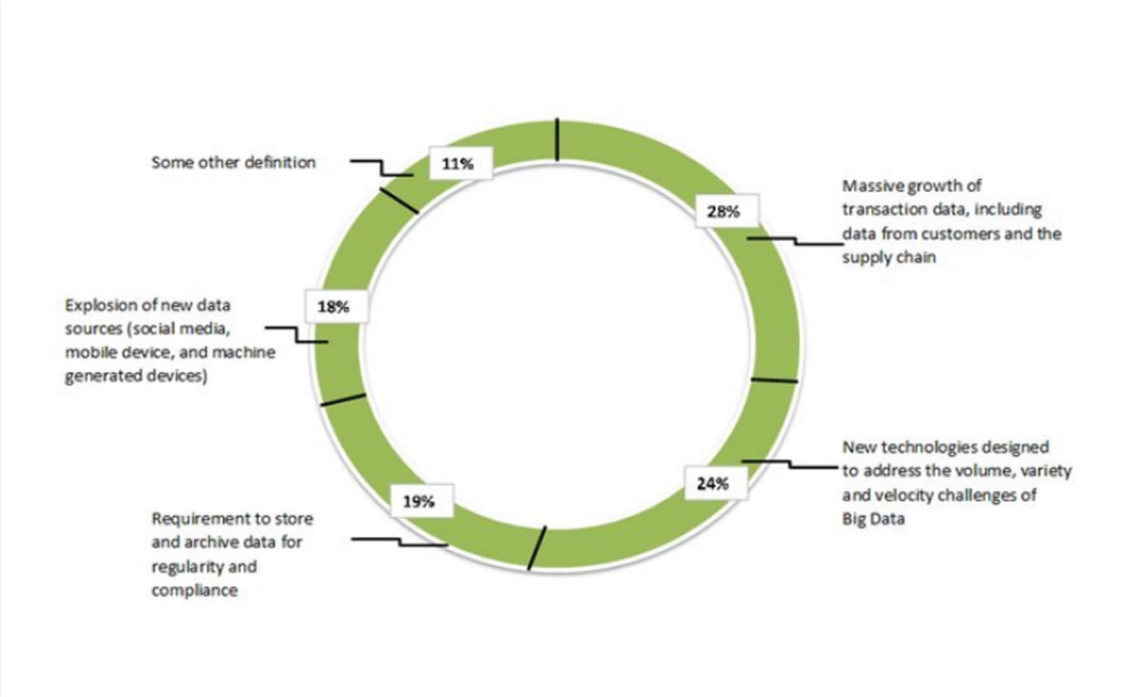
2. **Τι είναι τα** ‘**Μεγάλα**’ **Δεδομένα** (Big Data)

Τα μεγάλα δεδομένα απασχολούν την επιστημονική κοινότητα κατά κύριο λόγο τις τελευταίες δύο δεκαετίες. Αυτό βέβαια δεν σημαίνει πως δεν υπήρξαν και παλαιότερα επιστήμονες ή οργανισμοί που ασχολήθηκαν με το συγκεκριμένο ζήτημα ακόμα και στη δεκαετία του ενενήντα. Η δημοφιλία αυτών ανεβαίνει ανάλογα με την συνολική παραγωγή δεδομένων σε όλο τον κόσμο. Αρκεί να σκεφτούμε πως το 2010 η συνολική παραγωγή δεδομένων από όλες τις δυνατές πηγές ήταν 2 Zetabytes, ενώ το 2020 64,2 Zetabytes (**πηγή** Statista). Συνεπώς, μια συζήτηση μεταξύ επιστημόνων για τα μεγάλα δεδομένα το 1990 δεν θα ήταν κατανοητή σε έναν καθημερινό άνθρωπο εκείνης της εποχής, αφού αποτελούσε καθαρά επιστημονικό ζήτημα. Με την πάροδο του χρόνου και την ραγδαία αύξηση των πηγών δεδομένων, των αποθηκευτικών μέσων και των λογισμικών επεξεργασίας δεδομένων η επιστημονική κοινότητα μεγάλωσε και προσπάθησε να ορίσει τα ‘Μεγάλα’ Δεδομένα με διάφορους τρόπους. Μέσω μια ηλεκτρονικής μελέτης 2012 την οποία πραγματοποίησε η Harris Interactive, εκ μέρους της εταιρείας SAP (εικ.1), μας δείχνει πόσο διαφέρουν οι ορισμοί που θα έδιναν διάφορα στελέχη εταιρειών όσον αφορά στα Μεγάλα Δεδομένα, αφού κάποιοι προσπαθούσαν να εξηγήσουν τι είναι, ενώ άλλοι επικεντρώθηκαν στο τι κάνουν. (Oussous, Benjelloun, Lahcen, & Belfkih, 2017)

2.1 **Ορισμός**

Σε αντίθεση με τα παραδοσιακά δεδομένα που συναντάμε όλοι ως απλοί χρήστες υπολογιστή, με τον όρο μεγάλα δεδομένα (Big Data) αναφερόμαστε σε ευρέως αναπτυσσόμενα δεδομένα τα οποία μπορούν να λαμβάνουν ποικίλες μορφές. Οι μορφές ή αλλιώς τύποι δεδομένων (παραδοσιακών και «μεγάλων») είναι τρεις: τα δομημένα (structured), τα ημι-δομημένα (semi-structured) και τα αδόμητα (unstructured). Οι περισσότερες πηγές παράγουν αδόμητα δεδομένα, με αποτέλεσμα η ανάλυση τους μέσω των συμβατικών τεχνολογιών να καταλήγει προβληματική ως και ακατόρθωτη.

Ο ορισμός που προέρχεται από τους ειδικούς και τους επιστήμονες του πεδίου που μελετάμε διατυπώνεται μέσω τριών βασικών χαρακτηριστικών , τα τρία Vs όπως αναφέρονται συχνά στην αγγλική γλώσσα. Η επεξήγηση αυτών γίνεται παρακάτω (Furht & Vilanustre, 2016):



***Εικόνα 1**:Ορισμοί Big Data σύμφωνα με μια διαδικτυακή έρευνα που πήραν μέρος 154 στελέχη εταιρειών το 2012 από την εταιρεία SAP.*

2.2 **Χαρακτηριστικά**

- Volume/**Όγκος**: αναφέρεται στο μέγεθος των δεδομένων. Τα ψηφιακά δεδομένα παράγονται χωρίς διακοπή καθημερινά και σε τεράστιο όγκο από διαφορετικές συσκευές και εφαρμογές, όπως για παράδειγμα στα μέσα κοινωνικής δικτύωσης, στα καταστήματα που πραγματοποιούνται συναλλαγές πελατών, στα ‘έξυπνα’ κινητά, σε βιομηχανίες κτλ (Punn, Agarwal, Adiyarta, & Syafrullah, 2019). Αυτή η τεράστια παραγωγή δεδομένων είναι πολύ πιο εύκολη στις μέρες μας, αν σκεφτούμε ότι μόνο μια πλατφόρμα κοινωνικής δικτύωσης στην οποία πραγματοποιούνται σχόλια, αγορές, ενημέρωση επικαιρότητας , συζήτηση μέσω προσωπικών μηνυμάτων, διαθέτει 2 δισεκατομμύρια χρήστες οι οποίοι καθημερινά έμμεσα αποθηκεύουν τις

διαδικτυακές τους ενέργειες. Μόνο το Facebook αποθηκεύει 4 petabyte την ημέρα από τις ενέργειες των χρηστών του. Ακόμα, ο όγκος των δεδομένων δεν μπορεί να οριστεί με ένα συγκεκριμένο αριθμό για πάντα, διότι ένα όριο bytes το 2010 που θα μπορούσε να ορίζει ένα dataset ως big data dataset σήμερα θα είχε ξεπεραστεί (Gandomi & Haider, 2014). Επιπλέον, το ίδιο θα ίσχυε όσον αφορά στον τύπο ενός dataset. Η επεξεργασία μια δομημένης μορφής δεδομένων είναι πολύ πιο εύκολη και γρήγορη από την επεξεργασία αμέτρητων emails από έναν οργανισμό. Συμπεραίνεται, λοιπόν, ότι ο όγκος των big data είναι κάτι ευμετάβλητο και εξαρτάται απόλυτα από την εξέλιξη των τεχνολογιών, και θα ήταν άσκοπο να οριστεί με έναν συγκεκριμένο αριθμό (Gandomi & Haider, 2014).

- Velocity/**Ταχύτητα**: αναφέρεται στον ρυθμό παραγωγής των δεδομένων καθώς και στην ταχύτητα με την οποία θα πρέπει να επεξεργάζονται και να αναλύονται για να δίνουν όσο το δυνατόν πιο αξιόπιστα αποτελέσματα (Oussous, Benjelloun, Lahcen, & Belfkih, 2017). Η αυξανόμενη δημιουργία δεδομένων από ψηφιακές συσκευές, όπως είναι οι σένσορες σε συστήματα IOT καθιστούν την ταχύτητα αναγκαία για να μπορούν τα δεδομένα να ερμηνευτούν σωστά την στιγμή που παράγονται και να αποφεύγουν τον «θόρυβο» που δημιουργούν οι μεγάλες ροές δεδομένων (Gandomi & Haider, 2014). Στις μέρες μας ακόμα και απλά εμπορικά καταστήματα έχουν ανάγκη την ταχύτητα της επεξεργασίας δεδομένων, αρκεί να σκεφτούμε πως στην Αμερική τα καταστήματα Wall-Mart επεξεργάζονται κάθε ώρα ένα εκατομμύριο συναλλαγές οι οποίες προέρχονται από τους καταναλωτές τους (Cukier, 2010). Συμπεραίνεται, δηλαδή ότι όσο οι ψηφιακές συσκευές καταφέρνουν να απορροφούν περισσότερες πληροφορίες για τους χρήστες τους τόσο και ο ρυθμός παραγωγής δεδομένων θα αυξάνεται, και η ανάγκη για ταχεία επεξεργασία δεδομένων θα μεγαλώνει.
- Variety/**Ποικιλία**: αναφέρεται στην δομική ποικιλομορφία των δεδομένων που υπάρχουν μέσα σε datasets (Gandomi & Haider, 2014). Τα δεδομένα στις μέρες προέρχονται από πολλά και διαφορετικά τεχνολογικά μέσα, με αποτέλεσμα να διαφέρει και η μορφή τους (Punn, Agarwal, Adiyarta, & Syafrullah, 2019). Δηλαδή, τα δεδομένα που συλλέγονται μπορεί να είναι μουσική, βίντεο, εικόνες, αρχεία, σχόλια κτλ. Επίσης, η ποικιλομορφία υπάρχει και στους τύπους δεδομένων που μπορεί να έχει ένα dataset, όπως δομημένα / αδόμητα / ημιδομημένα, ιδιωτικά / δημόσια, τοπικά / απομακρυσμένα, κοινόχρηστα / εμπιστευτικά, πλήρη/.ατελή (Oussous,

Benjelloun, Lahcen, & Belfkih, 2017). Δεν είναι κρυφό πια ότι μεγάλοι οργανισμοί και πολυεθνικές εταιρείες διαχειρίζονται δεδομένα των πελατών τους που προέρχονται από διαφορετικά πηγές με σκοπό να παράγουν χρήσιμα αποτελέσματα για τα καταναλωτικά τους προφίλ και να τους προβάλουν προϊόντα ή υπηρεσίες που θα ήταν πιθανότερο να προτιμήσουν. Ο υψηλός βαθμός ποικιλομορφίας ως χαρακτηριστικό των μεγάλων δεδομένων δεν είναι κάτι καινούργιο, αλλά είναι σίγουρο ότι θα συνεχίσει να αυξάνει την ζήτηση κατάλληλων λογισμικών για την διαχείρισή τους όσο τα τεχνολογικά μέσα εξελίσσονται.

Είναι σημαντικό να αναφέρουμε ότι τα τρία Vs που παρουσιάστηκαν παραπάνω είναι τα πρώτα τρία χαρακτηριστικά που αναγνωρίστηκαν από τους περισσότερους επιστήμονες του κλάδου. Αυτό δεν σημαίνει πως δεν έχουν γίνει αρκετά αξιόλογες αναφορές σε άλλα χαρακτηριστικά που ορίζουν τα «Μεγάλα» Δεδομένα. Η ανάγκη για την διαχείρισή τους από κυβερνήσεις και πολυεθνικές εταιρείες έχει συγκεντρώσει την επιστημονική κοινότητα γύρω από τα Μ.Δ. και πλέον έχουμε ακόμα περισσότερα χαρακτηριστικά που βοηθούν στην κατανόηση τους. Κάποια από τα επιπλέον είναι τα εξής:

- Veracity/**Εγκυρότητα**: αναφέρεται στην αναξιοπιστία που έχουν κάποιες πηγές δεδομένων από την γέννηση τους (Gandomi & Haider, 2014). Σήμερα υπάρχουν πολλές πηγές που αθροιστικά παράγουν τεράστια ποσότητα δεδομένων. Ένα σύνηθες φαινόμενο σε αυτή την μαζική δημιουργία δεδομένων είναι ότι δημιουργούνται έκτοπες τιμές ή τιμές οι οποίες δημιουργούν «θόρυβο», δηλαδή επαναλαμβανόμενες τιμές που είναι παρόμοιες με άλλες. Τέτοια δεδομένα είναι εκείνα που προέρχονται από τα κοινωνικά δίκτυα τα οποία είναι αμφιβόλου αξιοπιστίας από την φύση τους, εφόσον εμπεριέχουν την ανθρώπινη κρίση. Η ανάγκη για την αντιμετώπιση τέτοιου είδους δεδομένων έχει φέρει μεγάλη ανάπτυξη σε εφαρμογές μηχανικής μάθησης, οι οποίες μπορούν να εντοπίσουν μέσω αλγορίθμων τον «θόρυβο», ανωμαλίες, ανθρώπινα λάθη και λάθη λογισμικών.
- Variability-Complexity/**Μεταβλητότητα-Πολυπλοκότητα**: η μεταβλητότητα αναφέρεται σε διακυμάνσεις που έχουν κάποιες ροές δεδομένων (Gandomi & Haider, 2014). Το χαρακτηριστικό αυτό παρατηρείται όταν τα δεδομένα έχουν αρκετά μεγάλη διαφορά, δηλαδή είναι απλωμένα. Ακόμα και οι διακυμάνσεις στην ταχύτητα παραγωγής δεδομένων (πλήθος) είναι ένα σημάδι υψηλής μεταβλητότητας. Η πολυπλοκότητα των δεδομένων αναφέρεται στο πλήθος διαφορετικών πηγών που

μπορούν να τα παράγουν. Αυτό αποτελεί πρόκληση για την επιστημονική κοινότητα, διότι πολλές διαφορετικές πηγές σημαίνει και διαφορετικές ενέργειες που θα πρέπει πραγματοποιηθούν έτσι ώστε να υπάρχει σύνδεση-ταίριασμα-καθάρισμα-μετατροπή των δεδομένων.

- Value/**Αξία**: αναφέρεται στην ποιότητα πληροφορίας που περιέχει ένα dataset (Gandomi & Haider, 2014). Η αξία δεν έχει καμία σχέση με την χρηματική αξία που θα μπορούσε να κοστίσει ένα dataset για να δημιουργηθεί. Αντίθετα, η αξία των δεδομένων έχουν να κάνουν με το πόσο πυκνή είναι η πληροφορία που διαθέτει ένα dataset. Στην πρώτη και ανεπεξέργαστη τους μορφή τα περισσότερα «Μεγάλα» Δεδομένα είναι αραιά, οπότε και έχουν μικρή αξία. Οι αναλυτές που τα επεξεργάζονται, μέσω της εξόρυξης δεδομένων (data mining) τους, αυξάνουν την ποιότητα και συνεπώς την αξία.

Τέλος, αρκετές ομάδες επιστημόνων προσπαθώντας να τα ορίσουν με μεγαλύτερη ακρίβεια, έχουν προσθέσει κάποια χαρακτηριστικά πιο μικρής σημασίας, όπως έχει . Κάποια από αυτά είναι (Oussous, Benjelloun, Lahcen, & Belfkih, 2017):

- Vision/Προοπτική
- Verification/Επαλήθευση
- Validation/Ευστάθεια
- Immutability/Σταθερότητα
- Visualization/Απεικόνιση

2.3 **Τύποι**

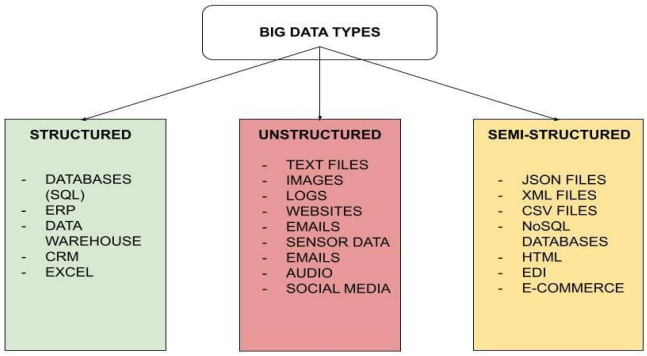
Για την καλύτερη κατανόηση και επεξεργασία μια συλλογής δεδομένων (παραδοσιακών ή μεγάλων), συχνά διαχωρίζονται σε τρεις διαφορετικές κατηγορίες σύμφωνα με τον τρόπο που αποθηκεύεται σε έναν υπολογιστή:

- Δομημένα (Structured): Τα δομημένα δεδομένα αποτελούνται από έναν ορισμένο σχεδιασμό ο οποίος είναι οργανωμένος σε μεγάλο βαθμό έτσι ώστε να μπορούν να χρησιμοποιηθούν εύκολα στην επεξεργασία και στην ανάλυση. Ως παράδειγμα αναφέρεται μια δομή σχεσιακής βάσης δεδομένων στην οποία αποθηκεύονται

πληροφορίες με συγκεκριμένες προδιαγραφές, παρόμοιες με εκείνες που θα χρησιμοποιούνταν σε περίπτωση ανάκτησης των δεδομένων από τον χρήστη. Η παραπάνω μορφή δεδομένων διαθέτει ένα σχεσιακό κλειδί (relational key) το οποίο μπορεί εύκολα να αντιστοιχηθεί στα προ σχεδιαστικά πεδία. Ακόμα, πιο συνήθη παραδείγματα παρόμοιων δομών δεδομένων είναι οι Βάσεις Δεδομένων (SQL Databases) , συστήματα ERP (Σύστημα Επιχειρησιακού Σχεδιασμού - SAP) καθώς και CRMs δηλαδή Συστήματα Διαχείρισης Πελατειακών Σχέσεων (Punn, Agarwal, Adiyarta, & Syafrullah, 2019).

- Αδόμητα (Unstructured): ο συγκεκριμένος τύπος δεδομένων δεν ακολουθεί κανένα είδος σχήματος. Αυτό τον καθιστά αρκετά δύσκολο στην ανάλυση των περιεχομένων του και τι περισσότερες φορές απαιτεί την συνεργασία κάποιων εργαλείων για να αποσπαστούν οι επιθυμητές πληροφορίες προς ανάλυση. Τέτοια είναι για παράδειγμα τα δεδομένα μεγάλης ροής (streaming data), log αρχεία, οι ιστοσελίδες , η ηλεκτρονική αλληλογραφία, τα βίντεο ή και αρχεία ανοικτού κειμένου χωρίς διαχωριστικό στηλών. Η ελλιπής οργάνωση των δεδομένων αυτού του τύπου καθιστούν την επεξεργασία τους μια δουλειά με υψηλό χρόνο και κόστος για τους οργανισμούς. Οι περισσότερες ροές δεδομένων πραγματικού χρόνου έχουν μορφή αδόμητων δεδομένων. Για να βρεθεί η λύση στο πρόβλημα της επεξεργασίας, τους συνήθως πρέπει να αναγνωριστεί κάποιο κρυφό μοτίβο που υπάρχει, αλλά όπως γίνεται κατανοητό αυτό δεν αποτελεί μια εύκολη εργασία. Άξιο αναφοράς είναι ότι τα περισσότερα δεδομένα που παράγονται καθημερινά σήμερα, είναι αδόμητα (Punn, Agarwal, Adiyarta, & Syafrullah, 2019).
- Ημιδομημένα (Semi Structured): αυτός ο τύπος δεδομένων μπορεί να οργανωθεί εφόσον εφαρμοστούν κάποιες επιθυμητές διαδικασίες όπως η μετατροπή ή μετατόπιση των δεδομένων. Υπάρχει μεγάλη ποικιλία λογισμικού που διαχειρίζεται αυτού του είδους τα δεδομένα όπως για παράδειγμα το Apache Hadoop. Μερικές φορές αναφέρεται αυτός ο τύπος δεδομένων και ως δομημένος με έλλειψη ακολουθίας ή μοτίβου που είναι διαθέσιμα με ανοργάνωτο τρόπο. Τέτοιου τύπου δεδομένα είναι σε μορφή αρχείων ανοιχτού κειμένου με διαχωριστικό tab, CSV, BibTex, XML, JSON ή και άλλα αρχεία γλωσσών προγραμματισμού με ετικέτες. Αυτός ο τύπος είναι πιο εύκολος στην διαχείριση συγκριτικά με τα εντελώς ανοργάνωτα δεδομένα και χρειάζεται αρκετά λιγότερος χρόνος και προσπάθεια για να

αναγνωριστούν τα μοτίβα και να επεξεργαστούν τις πληροφορίες οι αναλυτές (Punn, Agarwal, Adiyarta, & Syafrullah, 2019).



***Εικόνα 2**: Κατηγορίες Δεδομένων*

2.4 **Πεδία εφαρμογής**

Τα μεγάλα δεδομένα συνεχώς εξαπλώνονται σε πολλά και διαφορετικά πεδία της οικονομίας και της επιστήμης, όπως και τα συστήματα πληροφορικής. Η εφαρμογή των μεγάλων δεδομένων βοηθούν τις επιχειρήσεις να πάρουν σωστότερες αποφάσεις. Μέσω την ανάλυσης μεγάλου όγκου δεδομένων που προκύπτουν από τους καταναλωτές τους, ανακαλύπτουν κάποια κρυφά μοτίβα στον τρόπο που αγοράζουν προϊόντα ή υπηρεσίες. Αυτά τα δεδομένα προέρχονται από τα κοινωνικά δίκτυα, σένσορες συσκευών, logs ιστοσελίδων. γνώμες πελατών, browser cookies κτλ. Τελευταία πολλοί οργανισμοί ξοδεύουν αρκετά χρήματα σε λογισμικά διαχείρισης μεγάλων δεδομένων για να βρίσκουν μοτίβα, άγνωστες συσχετίσεις, προτιμήσεις της αγοράς , προτιμήσεις καταναλωτών και άλλες πολύτιμες πληροφορίες πελατών. Ακολούθως αναφέρονται κάποια σημαντικά πεδία όπου τα μεγάλα δεδομένα έχουν εφαρμογή όπως έχουν αποδοθεί από τους (Sunagar, Hanumantharaju, Siddesh, Kanavalli, & Srinivasa, 2020).

Υγεία: υπάρχει αξιοσημείωτη βελτίωση στο πεδία της υγείας μέσω της δημιουργίας προσωποποιημένων φαρμάκων και συνταγογραφικών αναλύσεων εξαιτίας του σημαντικού

όγκου μεγάλων δεδομένων. Οι ερευνητές αναλύουν διάφορα δεδομένα που έχουν στην κατοχή τους προτού αποφασίσουν την θεραπεία μιας συγκεκριμένης ασθένειας, όπως τις παρενέργειες των φαρμάκων ή να προβλέψουν πιθανούς κινδύνους για την υγεία. Εφαρμογές κινητών τηλεφώνων και συσκευές που φοριούνται στο ανθρώπινο σώμα παράγουν επιπλέον δεδομένα κάτι που συμβάλει στην ραγδαία ανάπτυξη τους. Είναι πολύ πιθανό πια, να προβλεφθεί το ξέσπασμα μια αρρώστιας από την χαρτογράφηση των γεωγραφικών και υγειονομικών δεδομένων μιας περιοχής. Με την έγκαιρη πρόβλεψη, γίνεται πιο εύκολος ο περιορισμός της εξάπλωσης και δίνεις περισσότερο χρόνο στους επιστήμονες να οργανώσουν το σχέδιο εξάλειψης της.

ΜΜΕ και Ψυχαγωγικά θεάματα: η βιομηχανία των ΜΜΕ και των τηλεοπτικών θεαμάτων διαφημίζουν και διανέμουν το περιεχόμενο τους χρησιμοποιώντας νέα επιχειρηματικά μοντέλα. Αυτό συμβαίνει λόγω της ανάγκης των καταναλωτών να βλέπουν ψηφιακό περιεχόμενο σε διάφορα μέρη κάθε στιγμή. Η εισαγωγή των διαδικτυακών σειρών σε πλατφόρμες όπως το Netflix, αποδεικνύει ότι η νέοι πελάτες ενδιαφέρονται για πρόσβαση σε δεδομένα από διαφορετικές τοποθεσίες. Αυτά τα media houses στοχεύουν το κοινό με το να προβλέπουν τις θα τους άρεσε να παρακολουθήσουν , τι διαφημίσεις θα προτιμούσαν και πως να διαχωρίσουν το περιεχόμενο που βλέπουν συχνά. Τα συστήματα μεγάλων δεδομένων που διαθέτουν οι εν λόγω εταιρείες τις βοηθούν να αυξήσουν τους πελάτες και κατά συνέπεια τον τζίρο τους, αναλύοντας τα κρυφά μοτίβα τους.

Internet of Things (IoT): Οι συσκευές IoT παράγουν συνεχείς ροές δεδομένων και τα αποστέλλουν σε έναν διακομιστή καθημερινά. Αυτά τα δεδομένα εξορύσσονται με σκοπό να εξασφαλίσουν την διασυνδεσιμότητα των συσκευών που τα δημιουργούν. Η απεικόνιση τους μπορεί να χρησιμοποιηθεί για καλό σκοπό από κυβερνητικές υπηρεσίες ή από ιδιωτικές εταιρείες για να διευρύνουν τις γνώσεις τους. Μέσω του IoT έχουν εφευρεθεί έξυπνα συστήματα για πότισμα γεωργικών εκτάσεων γης, διαχείριση οδικής κυκλοφορίας για την διευκόλυνση πολιτών και επειγόντων περιστατικών και διαχείριση πληθυσμού.

Παραγωγικός Τομέας : Η πρόβλεψη της παραγωγής βοηθάει στο να αυξηθεί η αποδοτικότητα σε βιομηχανίες με το να παράγονται περισσότερα αγαθά ελαχιστοποιώντας τις διακοπές λειτουργίας των μηχανημάτων. Αυτό περιλαμβάνει μεγάλη ποσότητα

δεδομένων από τέτοιες βιομηχανίες. Προηγμένα εργαλεία πρόβλεψης ακολουθούν μια οργανωμένη διαδικασία για να εξερευνήσουν πολύτιμες πληροφορίες για τέτοια δεδομένα. Μερικά από τα πιο σημαντικά πλεονεκτήματα χρησιμοποίησης τέτοιον εφαρμογών μεγάλων δεδομένων σε βιομηχανίες παραγωγής είναι η αύξηση ποιότητας των προϊόντων, έγκαιρη αναγνώριση σφαλμάτων, δημιουργία πλάνου προσφοράς, πρόβλεψη αποτελέσματος παραγωγής, αύξηση αποδοτικότητας σχετικά με την διαχείριση της ενέργειας που ξοδεύεται, δυνατότητα δοκιμής και προσομοίωσης νέων προϊόντων νέων παραγωγικών διαδικασιών και μεγάλος βαθμός εξατομίκευσης της παραγωγής.

Κυβερνήσεις : Με την υιοθέτηση συστημάτων μεγάλων δεδομένων, οι κυβερνήσεις μπορούν να επιτύχουν την βέλτιστη αποτελεσματικότητα σε όρους όπως κόστος, παραγωγή και καινοτομία. Λόγω του ότι αρκετά δεδομένα που χρησιμοποιούνται σε εφαρμογές είναι ίδια, πολλές υπηρεσίες των κυβερνήσεων έχουν την δυνατότητα να συνεργαστούν με στόχο την καλύτερη αποδοτικότητα και ακρίβεια. Οι κυβερνήσεις έχουν τον σημαντικότερο ρόλο στην καινοτομία αρκετών επιστημονικών πεδίων και τα συστήματα αυτά βοηθούν στην δραστηριοποίηση τους.

Τα μεγάλα δεδομένα είναι αναπόφευκτο να χρησιμοποιηθούν κάποια στιγμή σε όλα τα πεδία της οικονομίας ή της επιστήμης, αφού η αξιοποίηση τους με σωστό και ηθικό τρόπο φέρνει αρκετά πλεονεκτήματα στους οργανισμούς, στους υπαλλήλους αλλά και στους καταναλωτές. Επιπρόσθετα με τα παραπάνω υπάρχουν και κάποια ακόμα πεδία που τα μεγάλα δεδομένα υποβοηθούν τους ανθρώπους με το να τα δημιουργούν και μετέπειτα να τα αξιοποιούν:

- Γεωργία – Κτηνοτροφία
- Αεροπλοΐα
- Κυβερνοασφάλεια
- Πρόληψη εγκληματικότητας
- Ηλεκτρονικό εμπόριο
- Ανίχνευση fake news
- Ανίχνευση οικονομικής απάτης
- Πρόβλεψη καιρού

- Αξιολόγηση Φαρμακευτικών θεραπειών
- Επιστημονική έρευνα
- Λογιστική συμμόρφωση

3. **Ποιότητα Δεδομένων**

Με την πάροδο του χρόνου οι εταιρείες και οι οργανισμοί έχουν στη κατοχή τους όλο και περισσότερα δεδομένα να διαχειριστούν. Το επίπεδο ποιότητας των δεδομένων έχει μεγάλη σημασία για αυτές. Η χαμηλή ποιότητα δεδομένων είναι πιθανό να αποφέρει επιχειρησιακό χάος μέσω ανακριβών αναλύσεων και κακοσχεδιασμένων στρατηγικών. Επιπλέον τα προβληματικά δεδομένα που συνεπάγονται με την χαμηλή ποιότητα αυτών, προκαλούν οικονομική δυσχέρεια στις επιχειρήσεις, όταν για παράδειγμα παραδίδονται ηλεκτρονικές παραγγελίες σε λάθος διευθύνσεις ή όταν επιβάλλονται πρόστιμα εξαιτίας προβληματικών οικονομικών καταστάσεων. Σύμφωνα με την εταιρεία Gartner και τη έρευνα της το μέσο οικονομικό αντίκτυπο λόγω κακής ποιότητας δεδομένων είναι 15 εκατομμύρια δολάρια τον χρόνο (Moore, 2018), ενώ μια παρόμοια έρευνα της IBM αναφέρει ότι μόνο στις Ηνωμένες Πολιτείες οι επιχειρήσεις συνολικά χάνουν 3,1 τρισεκατομμύρια δολάρια τον χρόνο (Redman T. C., 2016).

3.1 **Ορισμός**

Η ποιότητα δεδομένων (Data Quality) έχει διαφορετικούς ορισμούς σε διαφορετικά πεδία και περιόδους. Σύμφωνα με την διαχείριση ποιότητας, η ποιότητα των δεδομένων είναι η καταλληλόλητα για χρήση ή για να καλύπτει τις ανάγκες του χρήστη ή ακόμα μπορεί να είναι η καταλληλόλητα για τις ανάγκες του κάθε πελάτη που τα χρησιμοποιεί. Ακόμα μια φράση που χρησιμοποιείται ευρέως από αρκετούς αναλυτές του χώρου είναι η «fitness for use» δηλαδή ικανότητα χρήσης όταν θέλουν να εξηγήσουν τι σημαίνει η ποιότητα δεδομένων. Πράγματι, η ποιότητα των δεδομένων αποτελεί σημαντικό κομμάτι της βελτίωσης της δραστηριότητας επεξεργασίας, ενώ μπορεί να απευθυνθεί σε διαφορετικά πεδία όπως στην φαρμακευτική, την στατιστική, την πληροφορική και την διαχείριση οικονομικών (Sidi, et al., 2012).

3.2 **Κατηγορίες Διαστάσεων Ποιότητας Δεδομένων**

Στο βιβλίο του Data Quality ο Rupa Mahanti (Mahanti, 2019) έχει παραθέσει σχεδόν όσα γνωρίζουμε από διάφορες πηγές και ερευνητές περί του θέματος της κατηγοριοποίησης

των διαστάσεων ποιότητας δεδομένων όπως θα δούμε στην συνέχεια. Ανά καιρούς αρκετοί ερευνητές έχουν προσπαθήσει να κατηγοριοποιήσουν και να προσεγγίσουν με τον δικό τους τρόπο το θέμα των κατηγοριών των διαστάσεων ποιότητας των δεδομένων. Παρατηρούμε ότι δεν έχει υπάρξει μέχρι στιγμής κάποια γενική σύμβαση περί κατηγοριοποίησης όπως είχαν αναφέρει οι (Wang & Wand, Anchoring data quality dimensions in ontological foundations, 1996). Βέβαια σε αρκετές δημοσιεύσεις βρίσκονται ποικίλες αναφορές από διαφορετικούς ερευνητές ως προς το θέμα αυτό.

Στο βιβλίο του Data Quality for the information Age ο Redman (Redman T. , Data Quality for the Information Age, 1996) έχει δώσει την οπτική ως προς την κατηγοριοποίηση μέσω τριών γενικών διαστάσεων:

- *Data modeling*: αφορά τις διαστάσεις ποιότητας που έχουν να κάνουν με το σχήμα
- *Data Value*: αναφέρεται στις τιμές αυτές καθαυτές, ανεξάρτητα από την εσωτερική αναπαράσταση τους.
- *Data presentation*: σχετίζεται με την μορφή που έχουν τα δεδομένα.

Με την παραπάνω οπτική, ένα στοιχείο μπορεί να οριστεί πιο αφηρημένα ως τριπλή αναπαράσταση: μια τιμή από το πεδίο χαρακτηριστικών μεταξύ μια οντότητας. Ο Redman αναγνωρίζει είχε αναγνωρίσει 27 διαφορετικές διαστάσεις ποιότητας οι οποίες είχαν στόχο την κατανόηση των δομών δεδομένων. Σχετικά με την κατηγορία data model ο Redman αναφέρει 15 χαρακτηριστικά που σχετίζονται με αυτήν και τα διαχωρίζει ως εξής: content(περιεχόμενο), level of detail (επίπεδο λεπτομέρειας), composition (σύνθεση), consistency (σταθερότητα), reaction to change (συμπεριφορά στις αλλαγές). Αυτά τα χαρακτηριστικά είναι διασυνδεδεμένα και αντικατοπτρίζουν τις επιλογές που μπορεί να παρθούν κατά την διάρκεια που δημιουργείται ένα μοντέλο. Επίσης παρουσιάζει 8 διαστάσεις που αφορούν την data presentation κατηγορία. Οι 7 από αυτές αφορούν την μορφοποίηση των δεδομένων: appropriateness (καταλληλότητα), interpretability (ερμηνευτικότητα), portability (φορητότητα), format precision (ακρίβεια μορφής), format flexibility (ευελιξία μορφής), ability to represent null values (ικανότητα αναπαράστασης κενών τιμών), efficient of use storage (αποτελεσματικότητα στην χρήση του αποθηκευτικού χώρου). Η τελευταία διάσταση της κατηγορίας data representation είναι η representational consistency που σχετίζεται με την φυσική υπόσταση των δεδομένων, που τις κάνει να

αλληλοστηρίζονται με τις μορφές τους. Κλείνοντας ο Redman την περιγραφή του για τις διαστάσεις με το ότι η συνέπεια των οντοτήτων, των τιμών και του τρόπου αναπαράστασης μπορούν να γίνονται ευκολότερα κατανοητές μέσω κάποιων όρων περιορισμού. Σε μεταγενέστερο βιβλίο του, ο (Redman T. , Data Quality: The Field Guide, 2001), διευρύνει σημαντικά τον αριθμό των διαστάσεων, συμπεριλαμβάνοντας συνολικά 55 από 7 διαφορετικές κατηγορίες. Οι 4 νέες κατηγορίες που προστέθηκαν είναι: data accessibility, privacy, commitment και improvement. Στην δεύτερη διατύπωση του συνεχίζει `να προσθέτει διαστάσεις με κύριο γνώμονα το data model ενώ έχει συμπεριλάβει μια νέα κατηγορία μια νέα κατηγορία που αφορά την αρχιτεκτονική.

Στο βιβλίο του ο Larry English (English, Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits, 1999) υπάρχουν δυο μεγάλες κατηγορίες που χωρίζουν τις διαστάσεις:

- Inherent Quality Characteristics: Περιεχόμενο πληροφοριών ή εγγενής ποιοτικά χαρακτηριστικά.
- Pragmatic Quality Characteristics: Παρουσίαση πληροφοριών ή ρεαλιστικά ποιοτικά χαρακτηριστικά

Στην πρώτη κατηγορία παρουσιάζονται τα ποιοτικά χαρακτηριστικά που υπάρχουν έμφυτα στα δεδομένα. Αναφέρει 9 τέτοια χαρακτηριστικά τα οποία είναι: definitional conformance (*συμμόρφωση στον ορισμό*), completeness of values (*πληρότητα τιμών*), business rule conformance (*συμμόρφωση στους κανόνες του οργανισμού*), accuracy to a surrogate source (*ακρίβεια υποκατάστατης πηγής*), accuracy to reality (*ακρίβεια στην πραγματικότητα*), precision (*ακρίβεια*), non-duplication (*απουσία διπλότυπων εγγραφών*), equivalence of redundant / distributed data (*ισοδυναμία περιττών / διαμοιρασμένων δεδομένων*), concurrency of redundant / distributed data (*συναίνεση περιττών / διαμοιρασμένων δεδομένων*).

Τα ρεαλιστικά ποιοτικά χαρακτηριστικά που συχνά αναφέρονται και ως dynamic quality characteristics (δυναμικά ποιοτικά χαρακτηριστικά) σχετίζονται με το πόσο ευκολονόητα είναι τα δεδομένα στην υπάρχουσα μορφή τους και κατά πόσο είναι ικανά να υποστηρίξουν επιχειρηματικές διαδικασίες ώστε να διευκολύνουν τους εργαζομένους να πετύχουν τους στόχους τους. Τα χαρακτηριστικά αυτά αφορούν κυρίως την αλληλεπίδραση ανθρώπου μηχανής, δηλαδή τα ποιοτικά χαρακτηριστικά παρουσίασης γίνονται δεκτά μόνο για την

αλληλεπίδραση των ανθρώπου και δεδομένων. Ο English παρέχει 6 ρεαλιστικά χαρακτηριστικά: accessibility (*προσβασιμότητα*), derivation integrity (ακεραιότητα προέλευσης), timeliness (*κατάλληλα χρονικά*). usability (χρηστικότητα), contextual clarity (σαφήνεια συμφραζόμενων), fact completeness (πληρότητα δεδομένων). Στο επόμενο βιβλίο του (English, Information Quality Applied: Best Practices for Improving Business Information, Processes and Systems, 2009) κατηγοριοποιεί εκ νέου τα χαρακτηριστικά ποιότητας. Αυτή την φορά τα χωρίζει σε χαρακτηριστικά περιεχομένου και χαρακτηριστικά παρουσίασης και προσθέτει 12 ακόμα διαστάσεις και να τις αθροίσει σε 27.

Στο βιβλίο του ο David Loshin (Loshin, 2010) διαχωρίζει τις ποιοτικές διαστάσεις των δεδομένων στις εξής κατηγορίες :

- Intrinsic
- Contextual
- Qualitative

Ο Loshin αναγνωρίζει 16 διαστάσεις που ανήκουν στις παραπάνω τρεις κατηγορίες. Η κατηγορία intrinsic περιέχει διαστάσεις που επικεντρώνονται κυρίως στις τιμές που έχουν τα δεδομένα και στις μετρήσεις τους χωρίς να σημαίνει ότι αξιολογούνται οι συνθήκες των τιμών. Ακόμα παρουσιάζει 4 εγγενής ποιοτικές διαστάσεις: accuracy (*ακρίβεια*), lineage (*ευθυγράμμιση*), structural consistency (*δομική συνέπεια*), semantic consistency (*εννοιολογική συνέπεια*), που περιγράφουν την δομή, την μορφή, το νόημα και την απαρίθμηση των δεδομένων.

Οι contextual διαστάσεις αφορούν την σχέση μεταξύ των στοιχείων δεδομένων, με έξι ακόμα διαστάσεις: completeness (*πληρότητα*), timeliness (*κατάλληλα χρονικά*), currency (*επίκαιρα*), consistency (*σταθερότητα*), reasonableness (*λογικά*), and identifiability (*αναγνωσιμότητα*).

Οι qualitative διαστάσεις επικεντρώνονται στο κατά πόσο οι προσδοκίες που έχουν προκαθοριστεί ταυτίζονται με τα δεδομένα. Όμως αυτές οι διαστάσεις εξαιτίας της υποκειμενικότητας που διαθέτουν δεν μπορούν εύκολα να μετρηθούν. Αναγνωρίζονται 6 τέτοιες διαστάσεις από τον Loshin: authoritative sources (έγκυρες πηγές), trust (εμπιστοσύνη), anonymity/privacy (ανωνυμία/απόρρητο), quality indicator (δείκτης

ποιότητας), edit and imputation (επεξεργασία και κατηγοριοποίηση), standards and policies (πρότυπα και πολιτικές).

Accuracy	25	Flexibility	5	Sufficiency	3	Informativeness	2
Reliability	22	Precision	5	Useableness	3	Level of detail	2
Timeliness	19	Format	4	Usefulness	3	Quantitativeness	2
Relevance	16	Interpretability	4	Clarity	2	Scope	2
Completeness	15	Content	3	Comparability	2	Understandability	2
Currency	9	Efficiency	3	Conciseness	2		
Consistency	8	Importance	3	Freedom from bias	2		

***Εικόνα 3**:Κοινά δημοσιευμένες διαστάσεις ποιότητας. Πηγή: (Wang, Storey, & Firth, A framework for analysis of quality research, 1995)*

Ο Wang (Wang, Storey, & Firth, A framework for analysis of quality research, 1995) στην έρευνα του έχει αθροίσει τις πιο συχνές διαστάσεις ποιότητας που υπάρχουν σε δημοσιεύσεις ερευνητών (εικόνα 2). Ο ίδιος έχει ομαδοποιήσει / κατηγοριοποιήσει όλες αυτές τις διαστάσεις σε 4 γενικές κατηγορίες.

1. *Intrinsic Data Quality:*

   Η εγγενής ποιότητα δεδομένων υποδηλώνει ότι τα δεδομένα έχουν ποιότητα από μόνα τους χωρίς την παρέμβαση μηχανής ή ανθρώπου. Τα «καλά» δεδομένα που προκύπτουν από αυτή την κατηγορία είναι accurate (ακριβή), correct (ορθά), objective (αντικειμενικά) και προέρχονται από κάποια αξιόπιστη πηγή.

2. *Contextual Data Quality:*

   Η ποιότητα των δεδομένων που αφορά τα συμφραζόμενα παραπέμπει στην προϋπόθεση ότι η ποιότητα των δεδομένων πρέπει να λαμβάνεται υπόψη στο πλαίσιο της αντίστοιχης εργασίας, κατανοητή σε μεγάλο βαθμό ως τον βαθμό σχετικότητας των δεδομένων με αυτό που χρειάζεται ο καταναλωτής των δεδομένων. Κάποιες διαστάσεις της contextual κατηγορίας είναι: relevancy (σχετικότητα), value-added (προστιθέμενη αξία), timeliness (κατάλληλα χρονικά), completeness (πληρότητα) και amount of information (ποσότητα πληροφορίας). Συνήθως οι διαστάσεις ποιότητας της κατηγορίας αφορούν το σχεδιασμό πριν ακόμα την δημιουργηθεί και την διεξαγωγή της έρευνας. Αντιθέτως είναι αρκετά πολύπλοκο να βελτιωθεί η ποιότητα των δεδομένων αφού συλλεχθούν.

3. *Representational Data Quality:*

   Αναφέρεται στην μορφή των δεδομένων και στον βαθμό που μπορεί κάποιος να αντλήσει νόημα από αυτά (Tress, 2017). Σύμφωνα με τον Fischer (Fisher, 2011) η συγκεκριμένη κατηγορία είναι βασισμένη στην άμεση χρηστικότητα των δεδομένων. Οι Wang και Strong (Wang & Strong, Beyond Accuracy: What Data Quality Means to Data Consumers, 1996) περιγράφουν την representational κατηγορία ως δεδομένα τα οποία παρουσιάζονται συνεχόμενα με πανομοιότυπη μορφή, συνεπή και μορφοποιημένα, καθώς και συμβατά με δεδομένα που έχουν παρουσιαστεί νωρίτερα. Κάποιες από τις διαστάσεις που σχετίζονται με την συγκεκριμένη κατηγορία είναι: interpretability (ερμηνευτικότητα), ease of understanding (ευκολία κατανόησης), concise representation (συνοπτική αναπαράσταση) και consistent representation (συνεπής αναπαράσταση).

4. *Accessibility Data Quality:*

   Επισημαίνει την σημασία των ρόλων και των αδειών που χρειάζονται για την πρόσβαση στα συστήματα. Η προσβασιμότητα είναι το εύρος στο οποίο τα δεδομένα είναι διαθέσιμα, καθώς και η ευκολία με την οποία μπορούν να διατεθούν σε έναν καταναλωτή δεδομένων. Αντίθετα η ασφάλεια πρόσβασης έχει να κάνει με τον περιορισμό και τον έλεγχο πρόσβασης στα δεδομένα από τρίτους. Εφόσον ένας καταναλωτής δεν έχει πρόσβαση στα δεδομένα τότε προφανώς δεν έχει την δυνατότητα να αξιολογήσει άλλες ποιοτικές διαστάσεις, όπως accuracy, completeness κτλ. Το σύστημα θα πρέπει να μπορεί να αποφεύγει περιστατικά κατάχρησης των δεδομένων του από καταναλωτές.

Συνοψίζοντας έχουμε ότι η κατηγορία intrinsic βασίζεται στην επιλογή σωστής πηγής δεδομένων, η κατηγορία contextual στον σχεδιασμό της μελέτης πολύ διεξοδικά, η κατηγορία representational στην συλλογή των δεδομένων με κατάλληλο τρόπο και στην κατηγορία accessibility να μεταδίδονται τα δεδομένα με τον σωστό τρόπο στους σωστούς χρήστες. Μέχρι στιγμής δεν έχει παρατηρηθεί συνέπεια στην ορολογία των διαστάσεων ποιότητας δεδομένων. Παρακάτω θα δούμε κάποιές από τις πιο σημαντικές και γνωστές διαστάσεις.

3.3 **Διστάσεις Ποιότητας Δεδομένων**

Οι διαστάσεις ποιότητας των δεδομένων χρησιμοποιούνται με σκοπό να αξιολογήσουν τα δεδομένα ενός οργανισμού ή επιχείρησης, ως προς την ποιότητα τους με όρους όπως, συμφωνία με τις προδιαγραφές, καταλληλότητα για χρήση, παράδοση των δεδομένων την κατάλληλη χρονική στιγμή. Οι διαστάσεις ποιότητας που περιέχει κάθε κατηγορία που αναφέρθηκαν προηγουμένως, βοηθούν στην κατανόηση διαφορετικών οπτικών που μπορούν να προσεγγίσουν την ποιότητά των δεδομένων. Παρακάτω θα δούμε κάποιες από τι πιο γνωστές και επαναλαμβανόμενες σε δημοσιεύσεις διαστάσεις ποιότητας όπως αναφέρονται στον βιβλίο του Rupa Mahanti (Mahanti, 2019).

Accuracy

Η ποιοτική διάσταση accuracy (ακρίβεια) αναφέρεται στο κατά πόσο σωστά αποθηκεύονται τα δεδομένα σε ένα σύστημα έτσι ώστε να αντικατοπτρίζουν την πραγματικότητα. Είναι ο βαθμός με τον οποίο τα δεδομένα περιγράφουν σωστά ένα ρεαλιστικό αντικείμενο, μια οντότητα, μια κατάσταση, ένα φαινόμενο ή ένα γεγονός που θα μπορούσε να περιγραφεί. Επίσης είναι η μέτρηση την ορθότητας του περιεχομένου των δεδομένων (τα οποία απαιτούν μια έγκυρη πηγή αναφοράς για να αναγνωριστούν και να γίνουν διαθέσιμα). Λάθη όπως τυπογραφικά σε ονόματα προϊόντων, φυσικών προσώπων, προμηθευτών ή και διευθύνσεων είναι πιθανό να έχουν άσχημο αντίκτυπο σε επιχειρησιακές εφαρμογές καθώς και σε εφαρμογές αναλύσεων. Ο Mahanti αναφέρει κάποιες ερωτήσεις οι οποίες βοηθούν στην κατανόηση της συγκεκριμένης διάστασης:

- Τα αντικείμενα δεδομένων χαρακτηρίζουν με ακρίβεια τις πραγματικές τιμές που προσδοκούν να παρουσιάσουν και τις πραγματικές οντότητες που διαμορφώνουν;
- Πόσο κοντά είναι η σωστή αποτύπωση των δεδομένων από αυτό που έχουν σχεδιαστεί να αποτυπώνουν.
- Τα δεδομένα αντιπροσωπεύουν με ακρίβεια την πραγματικότητα ή μια επαληθεύσιμη πηγή;

Ο Jack Olson στο βιβλίο του (Olson, January 2003) αναφέρει πως η ακρίβεια των δεδομένων αναφέρεται στο αν οι τιμές των δεδομένων είναι σωστές. Επεξηγεί μάλιστα, ότι για να είναι σωστά τα δεδομένα πρέπει να έχουν αληθινή τιμή και να απεικονίζονται με

συνεπή και ξεκάθαρη μορφή. Ο Olsen δηλώνει πως η συγκεκριμένη διάσταση αποτελείται από δύο χαρακτηριστικά την μορφή και το περιεχόμενο. Όσον αφορά την μορφή, είναι ο τρόπος αναπαράστασης των τιμών των δεδομένων. Για την καλύτερη κατανόηση μπορούμε φέρουμε ως παράδειγμα μια καταχώρηση μιας ημερομηνίας σε ένα database, όπως 12/10/1993. Αν βρισκόμαστε στην Αμερική η ημερομηνία μεταφράζεται ως 10 Δεκεμβρίου 1993, ενώ με τα Ευρωπαϊκά πρότυπα σημαίνει 12 Οκτωβρίου 1993. Σε αυτή την περίπτωση ένα χρήστης υπάρχει η πιθανότητα να κάνει ένα λάθος που σχετίζεται με την μορφή αν υπάρχει ασάφεια ως προς τους κανόνες την που περιέχει μία βάση δεδομένων. Όσον αφορά το περιεχόμενο ο Olsen λέει πως δυο τιμές δεδομένων μπορεί να είναι ακρινής και σαφής αλλά ακόμα να δημιουργούν προβλήματα. Όπως για παράδειγμα οι τιμές για μια περιοχή, «Άγιος Δημήτριος» και «Αγ. Δημήτριος», μπορεί να αναφέρονται στην ίδια αλλά δεν υπάρχει συνέπεια, επομένως μια από τις δύο πρέπει να θεωρηθεί ανακριβής. Επομένως η συνέπεια είναι ένα σημαντικό κλειδί ώστε να επιτευχθεί η ακρίβεια των δεδομένων.

Timeliness

Όλα αλλάζουν σχετικά με τον χρόνο, και τα δεδομένα δεν εξαιρούνται. Ο χρόνος επηρεάζει τα δεδομένα. Η διάσταση timeliness (κατάλληλη χρονική στιγμή) είναι μια από τις αντιπροσωπευτικές όσο έχει να κάνει με διαστάσεις ποιότητας σύμφωνα με τον χρόνο. Για τον έλεγχο χρονικής καταλληλότητας των δεδομένων μας θα πρέπει να κάνουμε κάποιες σημαντικές ερωτήσεις:

- Πόση είναι η χρονική διαφορά μεταξύ την πραγματοποίηση ενός γεγονότος στον πραγματικό κόσμο και στην καταγραφή των δεδομένων αυτού;
- Πόση είναι η χρονική διαφορά μεταξύ την πραγματοποίηση ενός γεγονότος στον πραγματικό κόσμο και στην στιγμή που τα τέθηκαν, τα σχετικά δεδομένα, διαθέσιμα στον χρήστη της επιχείρησης;
- Τα δεδομένα είναι προσβάσιμα όταν τα χρειάζονται;

Η timeliness διάσταση αναφέρεται στο αν τα δεδομένα είναι διαθέσιμα την χρονική στιγμή που προσδοκούσαμε ή και τα είχαμε ανάγκη. Τα δεδομένα θα πρέπει να συλλέγονται όσο το δυνατόν συντομότερα γίνεται με το που πραγματοποιηθεί ένα συμβάν και να γίνονται διαθέσιμα προς χρήση για προβλεπόμενες δραστηριότητες που βοηθούν τις επιχειρησιακές ανάγκες και αποφάσεις. Ο αποδεκτός χρόνος καθυστέρησης ανάμεσα στην στην πραγματοποίηση ενός γεγονότος και στην διαθεσιμότητα των σχετικών δεδομένων

εξαρτάται από το περιεχόμενο αυτών. Για παράδειγμα η καθυστέρηση σε online διαθεσιμότητα προϊόντων μπορεί να αποφέρει επιχειρησιακό χάος σε αρκετά τμήματα ενός οργανισμού, από ακυρώσεις παραγγελιών μέχρι πρόστιμο από οικονομικούς ελέγχους. Η timeliness διάσταση ποιότητας καθοδηγείται από το γεγονός ότι υπάρχει πιθανότητα να έχουμε διαθέσιμα δεδομένα τα οποία δεν εξυπηρετούν κανένα σκοπό επειδή έχουν αργήσει να εξαχθούν για κάποια συγκεκριμένη χρήση (Batini, Cappiello, Francalanci, & Maurino, 2009). Για να εξασφαλιστεί το timeliness των δεδομένων θα πρέπει να μετρηθεί η καθυστέρηση χρόνου από την στιγμή που θα ζητηθεί ένα κομμάτι δεδομένων μέχρι την στιγμή που θα είναι διαθέσιμο προς χρήση.

Relevance

Η διάσταση relevance (συνάφεια) ορίζεται ως ο βαθμός που το περιεχόμενο και η κάλυψη των δεδομένων είναι σχετικά με τον σκοπό με τον οποίο θα χρησιμοποιηθούν, και τον βαθμό τον οποίο εκπληρώνουν τις υπάρχουσες ή τις μελλοντικές ανάγκες. Κάποιες ερωτήσεις που θα πρέπει να γίνουν για την συγκριμένη ποιοτική διάσταση είναι:

- Είναι η έκταση που καλύπτουν τα δεδομένα σχετική με τον σκοπό;
- Τα δεδομένα εκπληρώνουν τις ανάγκες των χρηστών της επιχείρησης;
- Επαρκούν τα δεδομένα σε βάθος και έκταση για την σκοπό που χρειάζονται;

Για να εκτιμηθεί η relevance διάσταση των δεδομένων, είναι χρήσιμο να κατανοηθεί η αξία που έχουν τα εν λόγω δεδομένα, η αξία χαρακτηρίζεται από τον βαθμό με τον οποίο τα δεδομένα μπορούν να εξυπηρετήσουν του στόχους τους οποίους επιδιώκουν οι καταναλωτές και οι χρήστες. Οι στόχοι αυτοί μπορεί να είναι η απάντηση σε ένα ερώτημα ενός χρήστη, ή η εύρεση λύσης σε ένα πρόβλημα, αναφορά απαραίτητων προϋποθέσεων και ούτω κάθε εξής. Για να είναι τα δεδομένα σχετιζόμενα θα πρέπει να έχουν την κατάλληλη έκταση. Για παράδειγμα, για μια πολυεθνική εταιρεία εμπορίου, αν θα πρέπει να αξιολογηθούν οι πελάτες της με βάση κάποιες συγκεκριμένες παραμέτρους όπως ηλικία, φύλλο, γεωγραφική τοποθεσία, απασχόληση και ετήσιο εισόδημα, αλλά τα δεδομένα για το ετήσιο εισόδημα είναι διαθέσιμα μόνο στο πίνακα της βάσης δεδομένων για τους πελάτες της Β. Αμερικής, έτσι το όριο της συσχέτισης δεν εκπληρώνεται και θα πρέπει να αντιμετωπιστεί με το να γίνουν οι απαραίτητες αλλαγές στη δομή της βάσης δεδομένων και των πινάκων της με σκοπό να περιέχει επιπρόσθετα πεδία όπως είναι το ετήσιο εισόδημα. Η διάσταση relevance είναι αρκετά σημαντική για τα δεδομένα, αφού αν τα δεδομένα μας δεν

έχουν συνάφεια η αξία τους πέφτει ακόμα και αν είναι ακριβή, έγκυρα, πλήρη, συνεπή και επίκαιρα.

Completeness

Η διάσταση completeness (πληρότητα) αποτελεί την πιο θεμελιώδη από όλες τις διαστάσεις ποιότητας δεδομένων. Αποτελεί το μέτρο για το αν τα δεδομένα υπάρχουν ή απουσιάζουν. Σε μια σχεσιακή βάση δεδομένων το ότι υπάρχουν τα δεδομένα ισοδυναμεί με το δεν υπάρχουν κενές τιμές σε κάποιο πεδία του πίνακα μας, ενώ αν έχουμε απουσία δεδομένων σημαίνει ότι έχουμε κενές ή null τιμές σε κάποιο από τα πεδία του πίνακα μας. Ακόμα και οι τιμές «unknown» (άγνωστος) και «not applicable» (χωρίς ισχύ) θεωρούνται ως τιμές που λείπουν. Κάποιες ερωτήσεις που μπορούν να γίνουν είναι οι εξής :

- Είναι διαθέσιμα όλα τα απαραίτητα δεδομένα;
- Υπάρχουν ελλείψεις τιμών σε σημαντικά πεδία δεδομένων;
- Είναι όλα τα σετ δεδομένων καταγεγραμμένα;
- Είναι όλα τα αντικείμενα δεδομένων καταγεγραμμένα;

Σε κάποιες περιπτώσεις, η έλλειψη κάποιων δεδομένων δεν είναι σημαντική. Βέβαια υπάρχουν και άλλες περιπτώσεις που τα ελλιπή δεδομένα είναι σημαντικά για τις επιχειρησιακές διαδικασίες, οπότε αποτελούν λόγο ανησυχίας. Είναι σημαντικό υπάρχει επίγνωση γιατί μια τιμή λείπει από το σετ δεδομένων (Scannapieco, Missier, & Batin, 2005). Μια τιμή μπορεί να λείπει για τις ακόλουθες διαφορετικές περιπτώσεις:

I. Η τιμή υπάρχει αλλά δεν είναι γνωστή
II. Η τιμή δεν υπάρχει για να δοθεί ως τιμή
III. Η ύπαρξη της τιμής δεν είναι γνωστή (Scannapieco, Missier, & Batin, 2005)
IV. Το γνώρισμα δεν εφαρμόζεται
V. Η τιμή παράγεται σύμφωνα με κάποιες συγκεκριμένες συνθήκες

Στην πρώτη περίπτωση όπου έχουμε τιμή που λείπει από τα δεδομένα μας αλλά γνωρίζουμε ότι υπάρχει, είναι ένα αντιπροσωπευτικό σενάριο έλλειψης πληρότητας. Για παράδειγμα, ένα τηλέφωνο επικοινωνίας ενός οργανισμού-πελάτη λείπει, είναι περίπτωση έλλειψης δεδομένων αφού ένα οργανισμός υποχρεούται να έχει τηλέφωνο επικοινωνίας. Βέβαια στην δεύτερη περίπτωση που μια τιμή δεν υπάρχει ούτε στον πραγματικό κόσμο, όπως για παράδειγμα ένας πελάτης να μην έχει καθόλου κινητό τηλέφωνο, τότε δεν αποτελεί έλλειψη πληρότητας της τιμή. Στην τρίτη περίπτωση που δεν υπάρχει τιμή στα δεδομένα αλλά επίσης

δεν γνωρίζουμε αν υπάρχει και στον φυσικό κόσμο θα πρέπει να γίνονται επιπλέον διερευνήσεις και αναλύσεις έτσι ώστε να ανακαλυφθεί αν η έλλειψη τιμής μπορεί να γίνει αποδεκτή ή θα πρέπει να θεωρηθεί έλλειψη πληρότητας. Βέβαια πριν γίνει η οποιαδήποτε έρευνα περί του θέματος, θα πρέπει να αξιολογείται το όφελος σε συνάρτηση με το κόστος ενός τέτοιου εγχειρήματος. Στο επόμενο σενάριο που έχουμε τις περιπτώσεις που δεν μπορεί να εφαρμοστεί μια τιμή στο συγκεκριμένο πεδίο (N/A), είναι φανερό ότι δεν υπάρχει θέμα έλλειψη πληρότητας. Η τελευταία περίπτωση στην οποία μια τιμή παράγεται μόνο κάτω από κάποιες συγκεκριμένες συνθήκες, η έλλειψη πληρότητας μπορεί να θεωρηθεί μόνο αν έχουν εκπληρωθεί οι συνθήκες που χρειάζονταν για να παραχθεί μια τιμή. Για παράδειγμα ένα αυτοκίνητο εισέρχεται σε ένα ιδιωτικό χώρο στάθμευσης το οποίο διαθέτει βάση δεδομένων που αποθηκεύει όλα τα στοιχεία μια συναλλαγής χρήστης, ώρα εισόδου, ώρα εξόδου, συνολικός χρόνος , τελικό ποσό. Αν δεν έχει χτυπήσει το barcode στην έξοδο επειδή ακόμα χρησιμοποιεί την υπηρεσία, δεν μπορεί να παραχθεί η ώρα εξόδου και το ποσό που θα πληρώσει ο οδηγός.

Η διάσταση completeness είναι πολύ σημαντική για τα δεδομένα, και ακόμα περισσότερο για τα καίρια δεδομένα ενός σετ δεδομένων, αφού η έλλειψη τιμών σε αυτά μπορούν να αποφέρουν μεγάλα κόστη στις επιχειρήσεις η και ακόμη χειρότερα να χαθούν σημαντικές ευκαιρίες ανάπτυξης της γνώσης.

Consistency

Η διάσταση consistency (συνέπεια - σταθερότητα) σημαίνει ότι οι τιμές δεδομένων πρέπει να είναι όμοιες για όλες τις περιπτώσεις μια εφαρμογής, και επίσης ότι τα δεδομένα σε όλο το εύρος ενός οργανισμού θα πρέπει να είναι συγχρονισμένα μεταξύ τους.. Η μορφή και η εμφάνιση των δεδομένων θα πρέπει να είναι συνεπή σε όλο το σετ δεδομένων όταν αφορά μια οντότητα. Μερικές ερωτήσεις που μπορούν να γίνουν είναι:

- Υπάρχει σταθερή αναπαράσταση των δεδομένων;
- Διαφορετικές πηγές δεδομένων παρέχουν αντικρουόμενες πληροφορίες για το ίδιο αντικείμενο δεδομένων;
- Ποιες είναι οι τιμές δεδομένων που μα δίνουν αντιφατικές πληροφορίες;
- Τα δεδομένα αποθηκεύονται σε παρόμοια μορφή σε όλους τους πίνακες και συστήματα;
- Τα χαρακτηριστικά που είναι αμοιβαία εξαρτώμενα αντικατοπτρίζουν την προβλεπόμενη συνέπεια;

- Υπάρχουν διαφορετικά στοιχεία δεδομένων σε διαφορετικά περιβάλλοντα που σχετίζονται με το ίδιο αντικείμενο δεδομένων;

Κάποια από τα συνήθη προβλήματα ασυνέπειας των δεδομένων:

- Η κατάσταση της σύμβασης ενός εργαζομένου είναι «ληγμένο» ή «τερματισμένο» αλλά η κατάσταση πληρωμής είναι ενεργή.
- Υπάρχουν καταχωρήσεις πωλήσεων για έναν συγκεκριμένο μήνα, αλλά δεν υπάρχουν καταγεγραμμένες παραγγελίες στον ίδιο μήνα.
- Καταχώρησης παραγγελίας ενός κωδικού προϊόντος, ο οποίος δεν υπάρχει στην γενική λίστα κωδικών.
- Ένας τραπεζικός λογαριασμός έχει κλείσει, αλλά η τράπεζα συνεχίζει να στέλνει τις καταστάσεις του λογαριασμού στον πελάτη.
- Μια πιστωτική κάρτα έχει κλείσει, αλλά η κατάσταση χρέωσης λέει «ληξιπρόθεσμη».
- Ο αριθμό εργαζομένων ενός τμήματος μια εταιρείας είναι μεγαλύτερος από τον συνολικό αριθμό εργαζομένων.

Οι τιμές δεδομένων θα πρέπει να συνάδουν με τιμές διαφορετικών σετ δεδομένων .Στο πεδίο των βάσεων δεδομένων αυτό σημαίνει ότι ίδια δεδομένα τα οποία υπάρχουν σε διαφορετικούς αποθηκευτικούς χώρους θα πρέπει να θεωρούνται ότι θα ισοδύναμα (Silberschatz, Korth, & Sudarshan, 2006). Με τον όρο ισοδύναμα εννοούμε ότι έχουν ίδια ακριβώς τιμή και ερμηνεία, το οποίο είναι βασικό για να συμφωνούν μεταξύ τους οι διάφορες εφαρμογές και συστήματα. Η εργασία της συμφωνίας διαφορετικών συστημάτων που έχουν προβλήματα συνέπειας είναι πολύ απαιτητική. Με την διαδικασία του data synchronization μπορεί να λυθεί αυτό το πρόβλημα και τα δεδομένα να είναι ισοδύναμα. Ακόμα και η διάσταση accuracy να εκπληρώνεται σε όλα τα διαφορετικά συστήματα ενός οργανισμού, δεν σημαίνει ότι θα υπάρχει συνέπεια μεταξύ των τιμών τους.

Μπορούμε να ορίσουμε την consistency μέσω τριών διαφορετικών κατηγοριών: record level consistency (συνέπεια στο επίπεδο της εγγραφής), cross record consistency (συνέπεια διασταυρωμένων εγγραφών) και temporal consistency (χρονική συνέπεια). Η πρώτη κατηγορία αφορά τον βαθμό συνέπειας μέσα στην ίδια εγγραφή. Για παράδειγμα σε μία εγγραφή που έχει τα πεδία Ηλικία και Ημερομηνία Γέννησης, η τιμή ηλικία θα πρέπει να είναι ο χρόνος που έχουμε εκείνη την στιγμή πλην τον χρόνο γέννησης του χρήστη, ειδάλλως έχουμε πρόβλημα ασυνέπειας μέσα στην ίδια εγγραφή. Η δεύτερη κατηγορία αφορά την συνοχή ενός σετ γνωρισμάτων τιμών με ένα διαφορετικό σετ γνωρίσματων σε διαφορετικές

εγγραφές. Οι συγκεκριμένες ασυνέπειες μπορούν να προκύψουν στην ίδια βάση δεδομένων ή και σε διαφορετικές. Όσο περισσότερες ετερογενείς και αυτόνομες πηγές δεδομένων διαθέτει ένας οργανισμός τόσο μεγαλύτερες οι πιθανότητες να παράγονται ασυνεπή δεδομένα. Οι ασυνέπειές που αφορούν τις πηγές χωρίζονται σε τρεις υποκατηγορίες:

i. Schema Level (Σχηματικό επίπεδο), οι πηγές βρίσκονται σε διαφορετικά μοντέλα, ή έχουν διαφορετικό σχήμα αλλά ανήκουν στο ίδιο μοντέλο.
ii. Data Representation Level (Επίπεδο αναπαράστασης δεδομένων), τα δεδομένα των πηγών αναπαρίστανται με διαφορετικές μορφές ή διαφορετικά συστήματα μετρήσεων.
iii. Data Value Level (Επίπεδο τιμών δεδομένων), υπάρχουν πραγματικές διαφορές μεταξύ τιμών δεδομένων από διαφορετικές πηγές που περιγράφουν ένα ίδιο αντικείμενο

Η τρίτη κατηγορία ασυνέπειας αφορά ένα συγκεκριμένο γνώρισμα των δεδομένων και την τιμή του, αλλά σε διαφορετικές χρονικές στιγμές. Για παράδειγμα, αν η πελάτισσα με όνομα «Μαρία Παναγιώτου» παντρευτεί και θελήσει να πάρει το όνομα του συζύγου της και γίνει «Μαρία Ελευθεράκη» τότε θα έχουμε μια ασυνέπεια θα οφείλεται στον χρόνο, διότι ο εξαγόμενος πίνακας πριν την ημερομηνία που παντρεύτηκε θα έχει άλλο όνομα από αυτόν μετά.

Είναι σημαντικό να σημειωθεί ότι η εξασφάλιση consistency δεν εξασφαλίζει accuracy (ακρίβεια) ή correctness (ορθότητα). Είναι πολύ πιθανό να έχουμε συνέπεια και συγχρονισμό των δεδομένων μεταξύ διαφορετικών βάσεων δεδομένων, αλλα να έχουμε λανθασμένες τιμές σε αυτά, επειδή δεν αναπαριστούν σωστά γεγονότα του πραγματικού κόσμου, αντικείμενα και καταστάσεις. Αντίθετα όταν έχουμε ασυνεπής τιμές τότε σίγουρα μία από αυτές είναι και ανακριβής.

Ο Rupa Mahanti αναφέρει ακόμα 24 διστάσεις ποιότητας, πλην τις 5 που προαναφέρθηκαν, οι οποίες παρουσιάζονται εν συντομία στην συνεχεία.

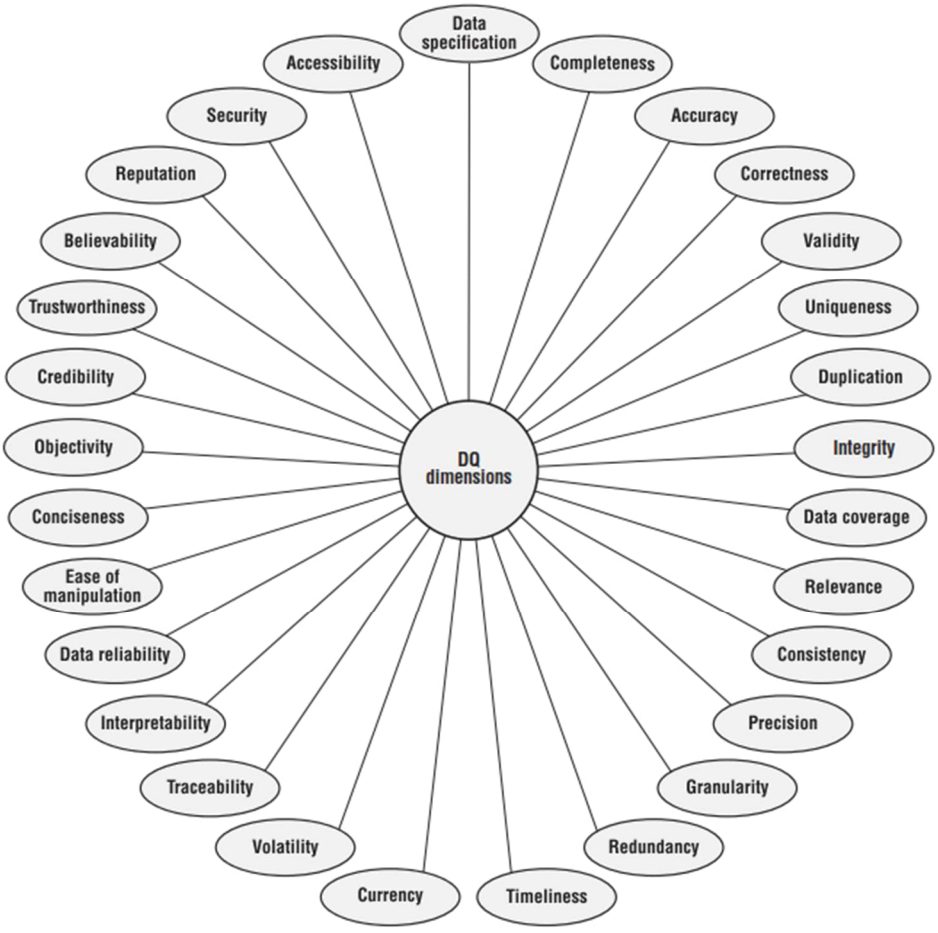
- *Data Specifications (Προδιάγραφες Δεδομένων)*: είναι το σύστημα αξιολόγησης των όρων existence (ύπαρξη), completeness (πληρότητα), quality (ποιότητας), documentation of data standards (οδηγίες των προδιαγραφών των δεδομένων), data models (μοντέλα δεδομένων), data definitions (ορισμοί δεδομένων) και metadata (μεταδεδομένα). Είναι σημαντικές διότι παράγουν το πρότυπο με το οποίο πρέπει να γίνεται η σύγκριση με τις άλλες διαστάσεις.

- *Correctness (ορθότητα):* σημαίνει ότι τα δεδομένα είναι απαλλαγμένα από λάθη. Η ορθότητα συνδέεται αρκετά με την διάσταση accuracy (ακρίβεια). Ενώ η ακρίβεια μπορεί να μετριέται με βαθμό, αυτό στην ορθότητα δεν ισχύει αφού έχει δυαδική φύση, δηλαδή έχουμε ορθά ή λανθασμένα δεδομένα.
- *Validity (εγκυρότητα)*: σημαίνει ότι τα δεδομένα συμφωνούν με ένα σύνολο εσωτερικών ή εξωτερικών κανόνων, ή κατευθυντήριων γραμμών, ή καθιερωμένων ορισμών δεδομένων, συμπεριλαμβανομένων των ορισμών των μεταδεδομένων.
- *Uniqueness (μοναδικότητα)*: η συγκεκριμένη ποιοτική διάσταση αφορά το γεγονός ότι δεν θα πρέπει να υπάρχουν διπλότυπες εγγραφές για την ίδια οντότητα ή γεγονός στο ίδιο σετ δεδομένων ή πίνακα δεδομένων.
- *Duplication (διπλότυπες εγγραφές)*: είναι η ακριβώς αντίστροφη διάσταση από την uniqueness, δηλαδή μετράει πόσες εγγραφές είναι διπλότυπες.
- *Integrity (ακεραιότητα):* αναφέρεται στις σχέσεις μεταξύ των αντικειμένων ή των οντοτήτων δεδομένων, στην εγκυρότητα αυτών σε όλο το εύρος των σχέσεων τους και στην διασφάλιση ότι όλα τα δεδομένα που περιέχουν οι βάσεις δεδομένων μπορούν να εντοπιστούν και να συνδεθούν με άλλα σχετικά δεδομένα.
- *Data Coverage (Έκταση Δεδομένων):* ορίζεται η έκταση της διαθεσιμότητας και της πληρότητας των δεδομένων σε σύγκριση με τον γενικό πληθυσμό των δεδομένων ή του πληθυσμού που χρειάζεται (McGilvray, 2021).
- *Precision (Ακρίβεια):* μπορεί να οριστεί και ως ο βαθμός επιδεξιότητας της διάστασης και ο βαθμός της λεπτομέρειας ως προς την περιγραφή ενός στοιχείου, όπως παραδείγματος χάρη ο αριθμός των δεκαδικών ψηφίων στα δεξιά της υποδιαστολής.
- *Granularity (Βαθμός λεπτομέρειας):* αναφέρεται στον βαθμό στον οποίο τα δεδομένα παρουσιάζονται συστατικά μέρη ή ακόμα και τον βαθμό στον οποίο τα δεδομένα μπορούν να υποδιαιρεθούν (Sebastian-Coleman, 2013).
- *Redundancy (Πλεονασμός):* ο πλεονασμός αφορά τα δεδομένα τα οποία έχουν αναπαραχθεί και συλλεχθεί σε δύο διαφορετικά συστήματα αποθήκευσης, συνήθως ως backup.
- *Currency (Επικαιρότητα), Volatility (μεταβλητότητα):* οι δυο αυτές διαστάσεις αφορούν στον χρόνο γι' αυτό δίδονται μαζί. Η διάσταση currency αναφέρεται στο βαθμό στον οποίο τα δεδομένα είναι ενημερωμένα για τις επιχειρησιακές

απαιτήσεις. Η διάσταση volatility μετράει την συχνότητα που τα δεδομένα αλλάζουν στον χρόνο.

- *Traceability (Ανιχνευσιμότητα):* αναφέρεται στον βαθμό που τα δεδομένα μπορούν να επαληθευτούν σε σχέση με την προέλευση, το ιστορικό, την ημερομηνία της πρώτης εγγραφής, την επικαιρομένη ημερομηνία και ώρα και την διαδρομή ελέγχου μέσω τεκμηριωμένης ταυτοποίησης.
- *Interpretability (Ερμηνευσιμότητα):* ορίζεται ο βαθμός στον οποίο ο χρήστης μπορεί εύκολα να κατανοήσει, να χρησιμοποιήσει και να αναλύσει τα δεδομένα.
- *Data Reliability (Αξιοπιστία Δεδομένων):* αναφέρεται στις διαστάσεις completeness, relevance, accuracy, uniqueness και consistency του σετ δεδομένων για τους επιθυμητούς σκοπούς χρήσης και στην ικανότητα ανίχνευσης μια πηγής που θεωρείται αξιόπιστη
- *Ease of manipulation (Ευκολία Παραποίησης):* ορίζεται ο βαθμός στον οποίο τα δεδομένα μπορούν να παραποιηθούν για διάφορες εργασίες, όπως τροποποίηση, συνένωση, προσαρμογή και συγκέντρωση δεδομένων.
- *Conciseness (Συνοπτικότητα):* αναφέρεται στον βαθμό στον οποίο τα δεδομένα είναι πυκνά χωρίς «φορτωμένα», δηλαδή συνοπτικά ως προς την αναπαράσταση, αλλά πλήρη και στενευμένα. (Wang & Strong, Beyond Accuracy: What Data Quality Means to Data Consumers, 1996).
- *Objectivity (Αντικειμενικότητα):* δίνει τον βαθμό στον οποίο τα δεδομένα αναπαριστούν την πραγματικότητα, χωρίς την διαστρέβλωση τους από προσωπικά συναισθήματα, στερεότυπα, κρυφά κίνητρα και προκαταλήψεις.
- *Credibility (Φερεγγυότητα), Trustworthiness (Αξιοπιστία), Believability (Αληθοφάνεια):* Οι τρείς παραπάνω διαστάσεις αφορούν την εμπιστοσύνη στα δεδομένα ή στα συστήματα των δεδομένων. Η Credibility αναφέρεται στο βαθμό στον οποίο ο πάροχος δεδομένων ή η πηγή δεδομένων έχει καλές προθέσεις έτσι ώστε τα δεδομένα να αναπαριστούν τα δεδομένα της πραγματικότητας. Η trustworthiness ορίζεται ως ο βαθμός στον οποίο τα δεδομένα προέρχονται από αξιόπιστες πηγές. Τέλος η διάσταση believability αναφέρεται στον βαθμό που τα δεδομένα θεωρούνται trustworthy και credible.

- *Reputation (Φήμη):* ορίζεται ως ο βαθμός στον οποίο τα δεδομένα έχουν υψηλή εκτίμηση από το κοινό που τα χρησιμοποιεί ως προς την πηγή και το περιεχόμενο τους.
- *Security (Ασφάλεια):* αφορά τον βαθμό όπου η πρόσβαση των δεδομένων είναι περιορισμένη και ελέγχεται κατάλληλα έτσι ώστε να αποφεύγεται η μη εξουσιοδοτημένη πρόσβαση.
- *Accessibility (Προσβασιμότητα):* είναι η ευκολία με την οποία μπορεί να αναγνωριστεί η ύπαρξη των εν λόγω δεδομένων και η καταλληλότητα της μορφής τους ή ακόμα η ευκολία πρόσβασης και ανάκτησης των δεδομένων.



Εικόνα 4 *Δενδρόγραμμα με τις διαστάσεις ποιότητας δεδομένων, Πηγή: (Mahanti, 2019)*

3.4 **Στρατηγικές Βελτίωσης Ποιότητας Δεδομένων**

Υπάρχουν δυο μέθοδοι που εφαρμόζονται με σκοπό την βελτίωση της ποιότητας δεδομένων. Ονομαστικά υπάρχουν η μέθοδος data driven και η process driven με την κάθε μέθοδο να ακολουθεί τις δικές της τεχνικές. Βέβαια αν και διαφέρουν έχουν τον ίδιο στόχο που είναι η βελτίωση της ποιότητας των δεδομένων (Sidi, et al., 2012).

A. Data Driven: αυτή η στρατηγική βελτίωσης βασίζεται στην μετατροπή των δεδομένων απευθείας. Κάποιες από τις τεχνικές που η στρατηγική data driven είναι:
    a. *Acquisition of new data*: Βελτιώνει τα δεδομένα μέσω της απόκτησης δεδομένων καλύτερης ποιότητας και αντικαθιστώντας τα σε αυτά που προκαλούν το πρόβλημα.
    b. *Standardization (Normalization)*: Αντικαθιστά τα δεδομένα τα οποία είναι μη τυπικά με αντίστοιχες τιμές οι οποίες συμμορφώνονται σε ένα συγκεκριμένο πρότυπο. Παραδείγματος χάρη το όνομα που έχει δοθεί ως Πάνος θα αντικατασταθεί με το Παναγιώτης.
    c. *Record Linkage*: Ξεχωρίζει τις αναπαραστάσεις των δεδομένων σε δυο ή και παραπάνω πίνακες, οι οποίοι μπορεί να απεικονίζουν το ίδιο αντικείμενο στην πραγματική ζωή. Όπως για παράδειγμα δυο εγγραφές πελατών που έχουν τον ίδιο ταχυδρομικό κώδικα στα στοιχεία τους.
    d. *Data and Schema Integration*: Ορίζει μια κοινή οπτική στα δεδομένα που προέρχονται από διαφορετικές πηγές. Ως κύριο στόχο η ενοποίηση (integration) έχει να επιτρέπει στον χρήστη την πρόσβαση στα αποθηκευμένα δεδομένα τα οποία μπορεί να είναι από διαφορετική προέλευση, αλλά τα παρουσιάζει υπό μια συγκεκριμένη οπτική. Οι πηγές δεδομένων έχουν μεγάλη ποικιλία ετερογενειών όπως είναι οι τεχνολογικές, οι σχηματικες και οι instance level. Η ενοποίηση θα πρέπει να είναι σε θέση να αντιμετωπίσει όλα αυτά τα είδη για να πετύχει τον σκοπό της.
    e. *Source Trustworthiness*: Εεπιλέγει πηγές δεδομένων ανάλογα με την ποιότητα των δεδομένων που περιέχει.
    f. *Error localization and correction*: Αναγνωρίζει τα προβλήματα ποιότητας και τα αποκλείει, με τον να διακρίνει τις εγγραφές οι οποίες δεν διαθέτουν κάποια βασικά πρότυπα ποιότητας. Αυτές οι τεχνικές υιοθετούνται κυρίως από το στατιστικό πεδίο. Αυτό συμβαίνει διότι τα στατιστικά δεδομένα είναι

αθροιστικά (average, sum, min-max) και για αυτό είναι λιγότερο ευαίσθητα. Τεχνικές για error localization and correction σκοπεύουν σε λάθη τύπου ασυνέπειας δεδομένων, ελλιπών δεδομένων ή και έκτοπων τιμών.

g. *Cost optimization*: Χαρακτηρίζει τις δραστηριότητες βελτίωσης δεδομένων σύμφωνα με κάποιες διαστάσεις έτσι ώστε να μειώσει το κόστος.

B. Process Driven: Είναι μια ακόμη στρατηγική η οποία βασίζεται στον επανασχεδιασμό της διαδικασίας που παράγει ή τροποποιεί τα δεδομένα με στόχο να βελτιώσει την ποιότητα τους. Υπάρχουν δυο βασικές μέθοδοι που πραγματοποιούν αυτήν την στρατηγική (Sidi, et al., 2012):

a. *Process Control:* Εισάγει διαδικασίες επαλήθευσης και ελέγχου στην διεργασία της παραγωγής δεδομένων όταν: (1) δημιουργούνται νέα δεδομένα, (2) κατά την επικαιροποίηση των dataset, (3) νέα dataset αποκτούν πρόσβαση στην διεργασία. Κατά αυτόν τον τρόπο η αντιδραστική αυτή στρατηγική εφαρμόζεται στην τροποποίηση των δεδομένων και έτσι αποφεύγονται τα επαναλαμβανόμενα λάθη και η υποβάθμιση της ποιότητας των δεδομένων.

b. *Process redesign*: Επανασχεδιάζει τις διαδικασίες με σκοπό να αφαιρέσει τα αίτια που προκαλούν τα κακής ποιότητας δεδομένα και εισάγει νέες δραστηριότητες που τα βελτιώνουν. Εάν η μέθοδος process redesign είναι ραγδαία τότε ονομάζεται και ως business process reengineering [Hammer and Champy 2001]

4. **Καθαρισμός Δεδομένων**

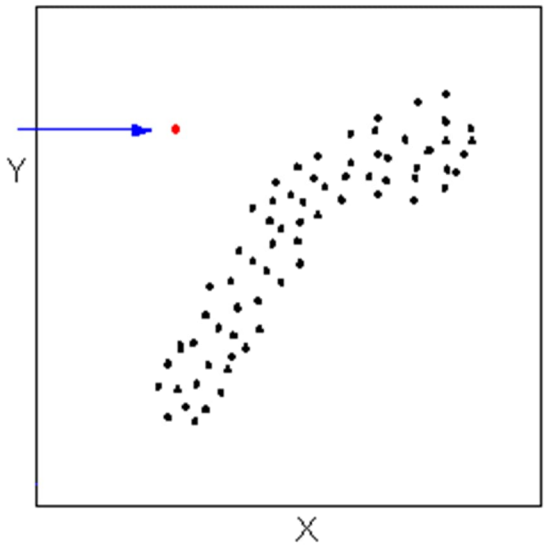
Ο καθαρισμός δεδομένων αποτελεί αναπόσπαστο κομμάτι της επιστήμης των δεδομένων. Αυτό συμβαίνει διότι οι οργανισμοί συλλέγουν υπέρογκα δεδομένα από διάφορες πηγές δεδομένων τα οποία περιέχουν λαθεμένες ή περιττές πληροφορίες. Η διαδικασία του καθαρισμού δεδομένων διαφέρει ανάλογα με τα ποιοτικά προβλήματα που μπορεί να έχει ένα σετ δεδομένων. Ο κύριος στόχος του καθαρισμού δεδομένων είναι να ανιχνεύσει τις λάθος τιμές ενός σετ δεδομένων και να τις διορθώσει, έτσι ώστε να ανεβάσει την ποιότητα των δεδομένων. Ένα σετ δεδομένων μπορεί να έχει λάθη όπως: ελλιπείς τιμές, τυπογραφικά λάθη, αναμειγμένες μορφές δεδομένων, επαναλαμβανόμενες οντότητες για το ίδιο αντικείμενο, έκτοπες τιμές και παραβίαση των επιχειρησιακών κανόνων. Παρακάτω θα εξεταστούν τα προβλήματα που απαιτούν τεχνικές καθαρισμού δεδομένων καθώς και οι διάφορες τεχνικές που προσπαθούν να τα επιλύσουν.

4.1 Outliers (**Έκτοπες τιμές**)

Από τα πιο συνηθισμένα ευρήματα ενός επιστήμονα δεδομένων σε ένα πίνακα δεδομένων είναι κάποιες τιμές οι οποίες απέχουν από το σύνολο των υπολοίπων κατά μεγάλο βαθμό. Οι τιμές αυτές οι οποίες, επί της ουσίας, δεν ακολουθούν την συμπεριφορά των τιμών των υπολοίπων συνόλων, τα οποία έχουν καθορίσει την έννοια της κανονικής συμπεριφοράς, ονομάζονται έκτοπες τιμές. Σύμφωνα με τον (Hawkins, 1980) έκτοπη τιμή είναι ένα σημείο παρατήρησης το οποίο αποκλίνει από τα άλλα σημεία παρατήρησης, ώστε να εγείρει τις υποψίες ότι μπορεί να δημιουργήθηκε από έναν τελείως διαφορετικό μηχανισμό . Ας εξεταστεί ως παράδειγμα η εικόνα 5, στην οποία υπάρχει ένα σύνολο με μαύρες κουκίδες οι οποίες αν και απέχουν μεταξύ τους ακολουθούν τους κανόνες μια συγκεκριμένης συμπεριφοράς και αποτελούν ένα σύνολο τιμών. Η κόκκινη κουκίδα απέχει κατά πολύ από την συμπεριφορά των υπολοίπων, γι' αυτό θεωρείται έκτοπη τιμή και θα πρέπει να διερευνηθεί περαιτέρω από τον εκάστοτε αναλυτή ή επιστήμονα δεδομένων. Η περαιτέρω διερεύνηση αναφέρεται, διότι εξαρτάται από τον εκάστοτε διαχειριστή των δεδομένων το πώς θα τις αξιοποιήσει. Οι έκτοπες τιμές που προέρχονται για παράδειγμα από τραπεζικά συστήματα μπορούν να ανιχνεύσουν μια υποκλοπή λογαριασμού και γενικότερα ύποπτες κινήσεις από εσωτερικούς ή και εξωτερικούς παράγοντες. Ωστόσο, σε

αρκετές περιπτώσεις οι έκτοπες τιμές είναι λαθεμένες ή σπάνιες τιμές, οι οποίες δεν βοηθούν τους αναλυτές στην εξαγωγή συμπερασμάτων, αντιθέτως μάλιστα πολλές φορές επηρεάζουν αρνητικά τα αποτελέσματα. Αν για παράδειγμα μελετώνται οι τιμές ενός συγκεκριμένου προϊόντος σε διαφορετικές ονομασίες καταστημάτων και η τιμή κατά πλειοψηφία κυμαίνεται κοντά στα 10 ευρώ, ενώ σε ένα κατάστημα είναι 50 ευρώ, τότε γίνεται αντιληπτό πως η τιμή 50 είναι μία έκτοπη τιμή.

4.1.1 **Τύποι** Outlier



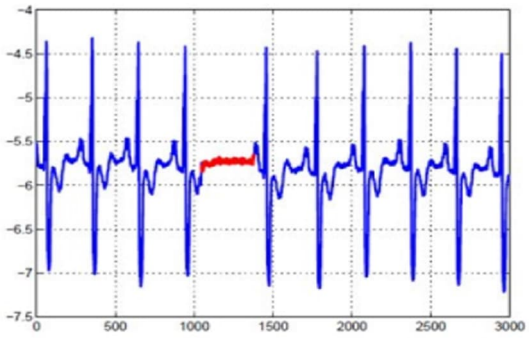
***Εικόνα 5**: Scatterplot με έκτοπη τιμή. Πηγή: Scatterplots (2 of 5) | Concepts in Statistics | | Course Hero*

Πριν την διαδικασία ανίχνευσης έκτοπων είναι σημαντικό να υπάρχει η γνώση του τύπου της έκτοπης τιμής. Οι έκτοπες τιμές χωρίζονται σε τρεις διαφορετικούς τύπους (Babu & Divya, 016):

- *Point Outliers:* Όταν ένα μεμονωμένο σημείο ενός συνόλου δεδομένων διαφέρει αρκετά από τα υπόλοιπα μπορεί να θεωρηθεί ως έκτοπο/ακραίο σημείο (εικόνα 5). Αυτή είναι η πιο συνήθης περίπτωση ακραίων τιμών και βρίσκεται στο επίκεντρο των επιστημονικών ερευνών και άρθρων με στόχο την ανίχνευσή τους από διάφορους αλγορίθμους.
- *Contextual Outliers*: Όταν μια τιμή φαίνεται να είναι ακραία βάσει ενός συγκεκριμένου πλαισίου αλλά όχι εν γένει ακραία, τότε ορίζεται ως contextual outlier (έκτοπο σημείο εξαρτώμενο από τα συμφραζόμενα/συνθήκες). Για παράδειγμα, ας

θεωρηθεί ότι ένας χρήστης μια χρεωστικής κάρτας ξοδεύει κατά μέσο όρο 120 ευρώ την εβδομάδα, εκτός από τα Χριστούγεννα που ξοδεύει κατά μέσο όρο 900 ευρώ ανά εβδομάδα. Ένα εβδομαδιαίο σύνολο εξόδων του ύψους των 800 ευρώ, κατά τον μήνα Ιούνιο, θα αποτελούσε contextual outlier.

- Collective Outliers: Όταν υπάρχει ένα σύνολο περιπτώσεων από δεδομένα τα οποία είναι σχετιζόμενα και έχουν διαφορετική συμπεριφορά από τα υπόλοιπα τότε ανήκουν στην κατηγορία collective outliers (Εικόνα 6). Είναι πολύ πιθανό τα σημεία μεμονωμένα να μην αποτελούν ακραίες τιμές, αλλά με την ομαδοποίηση δημιουργείται μια κολεκτίβα από τέτοιες. Τέτοιου είδους έκτοπες τιμές μπορεί να βρει κανείς σε σετ δεδομένων στα οποία τα δεδομένα συσχετίζονται



***Εικόνα 6**: Collective outliers Πηγή: (Babu & Divya, 016)*

4.1.2 **Κατηγορίες Ανίχνευσης** Outlier

Η Ανίχνευση Έκτοπων Τιμών είναι η διαδικασία που πρέπει να ακολουθηθεί για την εύρεσή τους. Σύμφωνα με τον (Chu, 2019) υπάρχουν τρεις ξεχωριστές κατηγορίες ανίχνευσης οι οποίες θα παρουσιαστούν παρακάτω:

- *Statistics based outlier detection (Ανίχνευση έκτοπων τιμών βασισμένη στην στατιστική):* Σύμφωνα με τις τεχνικές που ανήκουν σε αυτήν την κατηγορία, τα φυσιολογικά σημεία δεδομένων βρίσκονται σε περιοχές υψηλής πιθανότητας ενός στοχαστικού μοντέλου, ενώ αντίθετα οι έκτοπες τιμές θα εμφανίζονται σε περιοχές μικρής πιθανότητας του στοχαστικού μοντέλου (Grubbs, 1969). Αυτές οι τεχνικές δημιουργούν μια στατιστική ερμηνεία για τις ανιχνευμένες έκτοπες τιμές ή ακόμα και ένα μεσοδιάστημα βαθμολογίας/εμπιστοσύνης για ένα συγκεκριμένο σημείο το

οποίο θεωρείται ακραίο, κάτι που δεν θα συνέβαινε αν λάμβαναν δυαδική απόφαση (φυσιολογικό ή έκτοπο σημείο).

- *Distance based outlier detection (Ανίχνευση έκτοπων τιμών βασισμένη στην απόσταση):* Οι τεχνικές που ανήκουν στην κατηγορία αυτή ορίζουν μια τιμή απόσταση μεταξύ των σημείων δεδομένων η οποία χρησιμοποιείται για να χαρακτηρίσει την φυσιολογική απόσταση. Παραδείγματος χάρη, ένα φυσιολογικό σημείο δεδομένων θα πρέπει να είναι κοντά με τα υπόλοιπα σημεία δεδομένων, ενώ όσα σημεία διαφέρουν από αυτή την φυσιολογική συμπεριφορά θα πρέπει να θεωρούνται ως έκτοπα σημεία (Knorr & Ng, 1998). Η συγκεκριμένη κατηγορία ανίχνευσης έχει ένα πολύ θετικό χαρακτηριστικό, ότι δεν εξαρτάται από θεωρίες που πρέπει να εξάγει σχετικά με την γενική κατανομή των δεδομένων. Αντιθέτως, βασίζεται στα ίδια τα δεδομένα οπότε η ανίχνευση καθορίζεται πάντα από αυτά, είναι δηλαδή data-driven.
- *Model based outlier detection (Ανίχνευση έκτοπων τιμών βασισμένη στο μοντέλο):* Οι τεχνικές ανίχνευσης της κατηγορίας αυτής αρχικά εκπαιδεύονται σε ένα μοντέλο ταξινόμησης (classifier) από ένα σετ δεδομένων με ετικέτες στα σημεία δεδομένων. Έπειτα, εφαρμόζεται ο ήδη εκπαιδευμένος ταξινομητής σε ένα δοκιμαστικό σημείο δεδομένων, για να αποφασίσει αν αποτελεί έκτοπη ή φυσιολογική τιμή (Stefano, Sansone, & Vento, 2000). Οι προσεγγίσεις αυτές θεωρούν ότι ένας ταξινομητής αφού εκπαιδευτεί κατάλληλα μπορεί να διαχωρίσει τις τιμές ως φυσιολογικές ή έκτοπες χρησιμοποιώντας τα δοσμένα χαρακτηριστικά. Όταν κανένα από τα εκπαιδευμένα μοντέλα δεν μπορούν να ταξινομήσουν ένα σημείο ως φυσιολογικό τότε το ορίζουν ως έκτοπο.

4.1.3 **Τεχνικές Ανίχνευσης** Outlier

Οι έκτοπες τιμές τις οποίες ψάχνουμε σε ένα σετ δεδομένων, έναν πίνακα ή μια βάση δεδομένων μπορεί να προέρχεται από διαφορετικούς παράγοντες. Κάποιοι από αυτούς σχετίζονται με το ανθρώπινο λάθος την στιγμή της καταχώρησης των δεδομένων, όπως για παράδειγμα όταν ένας χρήστης να συμπληρώσει την ημερομηνία γέννησης του ως 1899 αντί για 1999. Επιπλέον, μια λάθος μέτρηση ενός μηχανισμού ή εργαλείου είναι πιθανή. Ένα ενδεικτικό παράδειγμα θα ήταν ένα θερμόμετρο που είναι

προγραμματισμένο να εξάγει την θερμοκρασία στην επιφάνεια της γης στις 12 η ώρα το μεσημέρι καθημερινά, και λόγω βραχυπρόθεσμου προβλήματος εξάγει την τιμή 5 βαθμούς κελσίου για τον μήνα Αύγουστο στην πόλη της Πάτρας. Ακόμα, πολύ συχνό είναι τα δεδομένα που έχουν εξαχθεί σωστά από μια πηγή να έχουν δεχθεί χειραγώγηση, όπως η αφαίρεση ψηφίων κατά την μορφοποίηση μιας στήλης που περιέχει γεωγραφικό πλάτος και μήκος. Τα παραπάνω αποτελούν χαρακτηριστικά παραδείγματα πηγής έκτοπων τιμών και ένας επιστήμονας δεδομένων όταν θα εφαρμόσει κάποια τεχνική εύρεσης έκτοπων τιμών θα του εμφανιστούν στην οθόνη των αποτελεσμάτων. Παρακάτω θα αναλυθούν 6 γνωστές τεχνικές ανίχνευσης έκτοπων τιμών ενός σετ δεδομένων.

- *Z-score:* Η διαφορά μεταξύ μιας τιμής ενός συνόλου δεδομένων και του μέσου όρου των δεδομένων του συνόλου, όταν υπολογίζεται σε μονάδες τυπικής απόκλισης, ονομάζεται Z-score ή αλλιώς standard score (deepblade, 2021). Ο βαθμός του Z μπορεί να είναι οποιοσδήποτε, θετικός ή αρνητικός ή και 0. Όταν ένα σημείο έχει τιμή 0 σημαίνει ότι βρίσκεται ακριβώς πάνω στο σημείο του μέσου όρου, ενώ αν είναι θετικός σημαίνει ότι είναι πιο πάνω και αρνητικός αντίστοιχα πιο κάτω. Η εξίσωση που ακολουθείται για την εύρεση του Z-score είναι η εξής: **Z**= (x – **μ**) / **σ**, όπου έχουμε x: ακατέργαστη βαθμολογία, μ: ο μέσος όρος του πληθυσμού/συνόλου, σ: η τυπική απόκλιση του πληθυσμού. Η τεχνική Z-score θεωρεί ότι τα στοιχεία δεδομένων υπάρχουν σε μια κανονική κατανομή[1] για να λειτουργήσει σωστά. Στην γλώσσα προγραμματισμού Python η Z-score υλοποιείται εύκολα μέσω των βιβλιοθηκών NumPy και pandas.
- *Local Outlier Factor (LOF):* Η τεχνική του LOF βασίζεται στην έννοια των τοπικών περιοχών. Υπολογίζει την τοπική πυκνότητα ενός συγκεκριμένου σημείου και την συγκρίνει με τις τοπικές πυκνότητες των γειτόνων του. Σε περίπτωση που η τιμή συγκριτικά με τα γειτονικά είναι χαμηλή αυτό πολύ πιθανά σημαίνει ότι αποτελεί έκτοπη τιμή. Ο αλγόριθμος LOF που μας δίνει αν κάποιο σημείο είναι ακραίο εξαρτάται από την παράμετρο K, η οποία δίνει τον αριθμό των σημείων που θεωρούνται γειτονικά με το

[1] Η κανονική κατανομή (γνωστή και ως γκαουσιανή κατανομή) αναφέρεται σε συνεχείς μεταβλητές αποτελώντας μία συνεχή συνάρτηση πυκνότητας πιθανότητας. Χρησιμοποιείται ως μία πρώτη προσέγγιση για να περιγραφούν τυχαίες μεταβλητές πραγματικών τιμών, οι οποίες τείνουν να συγκεντρώνονται γύρω από μια μέση τιμή.

αρχικό σημείο. Αυτός ο αριθμός μπορεί να είναι από 0 έως τον αριθμό των συνολικών σημείων που υπάρχουν στα δεδομένα (Mishra, 5 Outlier Detection Techniques that every "Data Enthusiast" Must Know, 2021). Για τον υπολογισμό του LOF θα πρέπει να ακολουθηθούν τρία βήματα:

i. Οι κ-πλησιέστεροι γείτονες θα πρέπει να βρεθούν για κάθε σημείο δεδομένων. Αν η απόσταση του κ-στου γείτονες ισοβαθμεί, τότε θα πρέπει να χρησιμοποιηθούν παραπάνω από κ γείτονες.

ii. Χρησιμοποιώντας του κ-πλησιέστερους γείτονες ενός σημείου, η τοπική πυκνότητά του υπολογίζεται βρίσκοντας την τοπική πυκνότητα προσιτότητας (LRD: Local Reachability Density).

iii. Στο τελευταίο βήμα από όπου και προκύπτει ο βαθμός LOF συγκρίνεται η τοπική πυκνότητα προσιτότητας του σημείου που εξετάζεται με την τοπική πυκνότητα προσιτότητας των κ-γειτόνων.

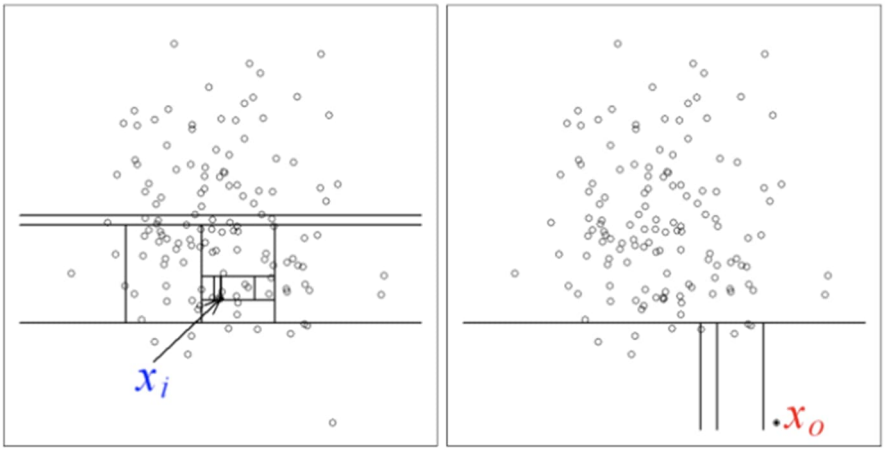
Όταν το LOF είναι λιγότερο από 1 τότε το σημείο του θεωρείται φυσιολογικό, ενώ αν είναι μεγαλύτερο τότε είναι ακραίο.

- *Angle Outlier Detection (ABOD):* Η συγκεκριμένη τεχνική έχει βασιστεί στην ιδέα της παρατήρησης της γωνίας που σχηματίζουν τρία σημεία στον χώρο πολλών διαφορετικών γνωρισμάτων. Η απόκλιση του μεγέθους της γωνίας διαφέρει για τα κανονικά και για τα έκτοπα σημεία, αφού για τα έκτοπα η απόκλιση είναι πολύ μεγαλύτερη απ' ότι στα κανονικά, κάτι που βοήθα στην ομαδοποίηση τους. Αυτή η βασισμένη στην γωνιακή απόκλιση τεχνική λειτουργεί πολύ καλά σε χώρο πολλών διαστάσεων, αντίθετα με αυτές που βασίζονται στην απόσταση, οι οποίες δυσκολεύονται αρκετά σε αυτό το κομμάτι (Mishra, outlier-detection-with-abod, 2022). Ο αλγόριθμος είναι αρκετά κατανοητός:

  i. Αρχικά υπολογίζει για κάθε σημείο την γωνία που σχηματίζει με όλα τα άλλα ζεύγη δεδομένων και τα αποθηκεύει σε μια λίστα.

  ii. Υπολογίζει την τυπική απόκλιση της λίστας που δημιουργήθηκε

  iii. Έχοντας ορίσει μια τιμή απόκλισης που θεωρείται ως όριο, επισημαίνει τα σημεία τα οποία θα μπορούσαν να θεωρηθούν ως ακραία. Η μικρή απόκλιση θεωρεί ότι το σημείο είναι πιθανό για να είναι έκτοπο, ενώ αντίστοιχα η μεγάλη απόκλιση σημαίνει ότι είναι φυσιολογικό σημείο.



Εικόνα 7: *: Φυσιολογικό σημείο - Ακραίο σημείο Πηγή: Outlier Detection with Isolation Forest | by Eryk Lewinson | Towards Data Science*

- *Isolation Forest:* Η ιδέα αυτής της τεχνικής διαφέρει αρκετά από άλλες γνωστές τεχνικές εντοπισμού έκτοπων τιμών. Προσπαθεί να ανιχνεύσει τις ακραίες τιμές χωρίς να δημιουργεί κάποιο προφίλ για τις φυσιολογικές τιμές. Είναι μια τεχνική που χρησιμοποιεί δένδρα απόφασης, τα οποία δημιουργούν τα κλαδιά τους αρχικά με βάση ένα τυχαίο χαρακτηριστικό και εν συνεχεία επιλέγοντας μια τυχαία τιμή μεταξύ της ελάχιστης και της μέγιστης του χαρακτηριστικού. Βάσει της συγκεκριμένης θεωρίας οι τιμές που είναι έκτοπες είναι λιγότερες και διαφέρουν αρκετά σε επίπεδο τιμών από τις φυσιολογικές. Για τον λόγο αυτό, χρησιμοποιώντας τυχαίο καταμερισμό θα πρέπει να βρίσκονται πιο κοντά στην ρίζα του δένδρου, με πιο λίγες διασπάσεις. Όπως φαίνεται και από την εικόνα 7, ένα σημείο το οποίο είναι ακραίο δέχεται πολύ λιγότερες κατατμήσεις, από ότι ένα φυσιολογικό σημείο. Για να ληφθεί η απόφαση αν ένα σημείο είναι ακραίο απαιτείται βαθμολόγηση των σημείων μέσω μιας εξίσωσης: $s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$, όπου h(x) είναι το μήκος της διαδρομής του σημείου που μελετάται, ενώ c(n) είναι το μέσο μήκος διαδρομής της ανεπιτυχούς αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης και n είναι ο αριθμός των εξωτερικών σημείων. Η βαθμολογία κοντά στο 1 συνήθως σημαίνει ότι το σημείο είναι πιθανά έκτοπο, ενώ μικρότερο από 0,5 αποτελεί φυσιολογικό σημείο του συνόλου (Liu, Ting, & Zhou, 2008).
- *Autoencoder:* Η τεχνική της αυτόματης κωδικοποίησης ανήκει στην οικογένεια των Νευρωνικών Δικτύων και γίνεται αντικείμενο εκπαίδευσης με σκοπό να αναπαραγάγει την είσοδο (Mishra, 5 Outlier Detection Techniques that every “Data Enthusiast” Must Know, 2021). Αποτελείται από κωδικοποιητή (encoder), ο οποίος



***Εικόνα 8**: Παράδειγμα διαδικασίας ανίχνευσης έκτοπων τιμών μέσω autoencoder Πηγή: : (Mishra, 5 Outlier Detection Techniques that every "Data Enthusiast" Must Know, 2021)*

έχει στόχο να μάθει μια λανθάνουσα αναπαράσταση εισόδου και αποκωδικοποιητή (decoder), ο οποίος προσπαθεί να αναδομήσει την είσοδο από την λανθάνουσα αναπαράσταση. Για να λειτουργεί σωστά αυτή η τεχνική θα πρέπει και οι δύο να βελτιστοποιούνται ως προς τις εργασίες που έχουν να κάνουν. Για να διαπιστωθεί πώς λειτουργεί αυτή η τεχνική θεωρείται ότι ένα έκτοπο σημείο απέχει κατά πολύ από ένα φυσιολογικό, διότι έχει διαφορετικές ιδιότητες. Όπως παραδείγματος χάρη ένα σύνολο εικόνων από σκύλους με μία εικόνα μιας αγελάδας απέχει κατά πολύ. Σε μία τέτοια περίπτωση ο autoencoder έχει μάθει, εξαιτίας του μεγαλύτερου αριθμού εικόνων από σκυλιά, την κατανομή της εικόνας ενός σκύλου. Με αυτόν τον τρόπο, η αναπαραγωγή της αρχικής εισόδου θα έχει λιγότερες απώλειες τιμών σε ότι αφορά τις εικόνες των σκύλων σε αντιδιαστολή με αυτή της αγελάδας. Αυτές οι βαθμολογίες ανακατασκευής επί της ουσίας χρησιμοποιούνται ως βαθμολογίες και ως βαθμολογίες ανωμαλίας τιμών. Όσο μεγαλύτερη η βαθμολογία τόσο το πιθανότερο ένα σημείο να είναι έκτοπο. Όπως φαίνεται και στην εικόνα 7 ένα σύνολο εισέρχεται στον encoder και εξάγεται από τον decoder. Επειδή το σύνολο έχει αρκετά κίτρινα σημεία οι encoder και decoder έχουν μάθει να τα μετατρέπουν σωστά. Αντίθετα ,όμως ,το μπλε σημείο δεν μπορεί να μετατραπεί το ίδιο σωστά και έχει υψηλή πιθανότητα να αποτελεί έκτοπο σημείο (μπορεί να ληφθεί υπόψη το προηγμένο παράδειγμα, με τα κίτρινα σημεία να αποτελούν τις φωτογραφίες σκύλων και η μπλε της αγελάδας).

- *Outlier Detection using In-degree Number (ODIN):* Βασίζεται στον υπολογισμό του in-degree, ο οποίος είναι ο βαθμός πλησιέστερων γειτονικών συνόλων που ανήκει το σημείο. Όσο μεγαλύτερος είναι αυτός ο αριθμός τόσο το πιθανότερο να μην αποτελεί έκτοπο σημείο, αφού βρίσκεται σε περιοχή υψηλής πυκνότητας σημείων. Αντίθετα η χαμηλή βαθμολογία θα σήμαινε ότι το σημείο είναι πιο απομονωμένο

στον χώρο από τα υπόλοιπα. Η τεχνική αυτή θυμίζει να είναι η αντίστροφη από την k-nearest neighbor τεχνική ανίχνευσης έκτοπων σημείων δεδομένων.

4.2 Missing Data (**Ελλιπή Δεδομένα**)

Το πρόβλημα των ελλιπών/ χαμένων δεδομένων υπάρχει για τους αναλυτές αυτών από την έναρξη του όρου data collection, δηλαδή της διαδικασίας συλλογής δεδομένων. Από την καταγραφή των δεδομένων μέχρι την εξαγωγή και συλλογή τους αρκετοί παράγοντές μπορούν να τα επηρεάσουν με αποτέλεσμα κάποια δεδομένα να μην συμπληρωθούν ποτέ ή να χαθούν λόγω στιγμιαίας αδυναμίας ενός υπολογιστικού συστήματος.

4.2.1 **Κατηγορίες Ελλιπών Δεδομένων**

Παρακάτω θα εξεταστούν οι διαφορετικές κατηγορίες ελλιπών δεδομένων που έχουν καταγραφεί σύμφωνα με τους (Brown & Kros, 2003).

- *Missing At Random (MAR):* Ο Rubin D. όρισε την τα ελλιπή δεδομένα ως MAR, «με δοθείσες τις μεταβλητές Χ και Ψ, η πιθανότητα ανταπόκρισης εξαρτάται από το Χ και όχι από το Ψ.» (Rubin, 1978) .Οι περιπτώσεις στις οποίες τα δεδομένα μας δεν είναι πλήρη θα πρέπει να αντιμετωπίζονται διαφορετικά με αυτές που είναι. Το μοτίβο που μπορεί να ακολουθούν τα ελλιπή δεδομένα μπορεί να τα κάνουν πιο ανιχνεύσιμα από άλλες μεταβλητές του σετ δεδομένων παρά να οφείλονταν σε μια μεταβλητή που τα δεδομένα λείπουν. Παραδείγματος χάρη στην περίπτωση που κάποιοι ερευνητές κάνουν ένα τεστ αναγνωστικής κατανόησης πριν δώσουν στους συμμετέχοντες το κανονικό τεστ έτσι ώστε να βρουν αυτούς με τη χαμηλότερη βαθμολογία. Οι συμμετέχοντες είναι πιο πιθανό να έχουν αφήσει τις ερωτήσεις προς το τέλος της έρευνας αναπάντητες από ότι αυτές στην αρχή. Παρόμοια η πιθανότητα κάποιος να απαντήσει στην ερώτηση για το βάρος του βασίζεται στην πιθανότητα κάποιος να μην έχει απαντήσει για την ηλικία του, τότε τα ελλιπή δεδομένα χαρακτηρίζονται Missing At Random.
- *Missing Completely At Random (MCAR):* ο Rubin (Rubin, 1978) ξεχωρίζει τα δεδομένα ως MCAR, όταν «η πιθανότητα της απάντησης δείχνει ότι η ανεξαρτησία υπάρχει μεταξύ Χ και Ψ». Τα MCAR δεδομένα παρουσιάζουν πολύ μεγαλύτερο επίπεδο

τυχαιότητας σε σχέση με τα MAR. Επί της ουσίας οι τιμές Ψ είναι τυχαία δείγματα για όλες τις τιμές Ψ, και κανένας παράγοντας δεν μεροληπτεί όσον αφορά τις παρατηρηθείσες τιμές Ψ. Για παράδειγμα στην περίπτωση που ένα χημικό εργαστήριο παρέχει αποτελέσματα από δοκιμές αποσυνθέσεων χημικών ενώσεων στις οποίες αναζητούνται σημαντική ποσότητα σιδήρου. Εάν σε μερικές περιπτώσεις πληρούνται τα επίπεδα σιδήρου ή σε άλλες λείπει εντελώς και δεν έχουν εντοπιστεί άλλα στοιχεία στην ένωση ή λείπουν δεδομένα για τον σίδηρο τότε αυτά τα ελλιπή δεδομένα λέγονται MCAR.

- *Non-ignorable missing data:* Αντίθετα από την MAR περίπτωση όπου η έλλειψη τιμών εξηγείται από άλλες μετρήσιμες μεταβλητές, τα non-ignorable ελλιπή δεδομένα προκύπτουν εξαιτίας του μοτίβου έλλειψης δεδομένων που εξηγείται από την ίδια μεταβλητή από την οποία λείπουν τα δεδομένα. Δηλαδή εάν έχουμε δυο μεταβλητές Χ και Ψ, τα δεδομένα κρίνονται non-ignorable όταν η πιθανότητα ανταπόκρισης εξαρτάται από την μεταβλητή Χ και πιθανόν από την Ψ. Τα non-ignorable ελλιπή δεδομένα είναι τα πιο δύσκολα αντιμετωπίσιμα και μάλιστα είναι αυτά τα οποία συμβαίνουν πιο συχνά.
- *Outliers treated as missing data:* Οι δοκιμές των δεδομένων καθώς και υπολογισμός των ορίων των τιμών είναι αναγκαία κατά την προεπεξεργασία των δεδομένων με σκοπό να χαρακτηριστούν κάποιες τιμές ως ελλιπείς. Κάποιες τιμές που ανήκουν στα δεδομένα και βγαίνουν από τα προκαθορισμένα όρια μπορεί να στρεβλώνουν τα αποτελέσματα. Αυτές οι τιμές θα πρέπει να χαρακτηρίζονται ως δεδομένα που δεν υπάρχουν. Για παράδειγμα στην περίπτωση που ένα χημικό εργαστήριο παρέχει αποτελέσματα από δοκιμές αποσυνθέσεων χημικών ενώσεων και έχει ορίσει ότι το ανώτερο όριο σιδήρου που μπορεί να βρεθεί σε μια τέτοια αποσύνθεση είναι 500 μέρη ανά εκατομμύριο, δεν θα πρέπει ποτέ αυτό το όριο να ξεπεραστεί. Στην περίπτωση που ξεπεραστεί θα πρέπει μέσω κάποιας τεχνικής απεικόνισης να απομονωθεί και να αναγνωριστεί έτσι ώστε να δοθεί στον τελικό χρήστη.

4.2.2 **Προσεγγίσεις αντιμετώπισης ελλιπών δεδομένων**

Η έλλειψη τιμών σε ένα σετ δεδομένων είναι από τα πιο συνήθη προβλήματα που καλείται να αντιμετωπίσει ένας επιστήμονας δεδομένων. Σε οποιαδήποτε από τις

κατηγορίες που προαναφέρθηκαν και να ανήκει θα πρέπει να αντιμετωπιστούν με κάποιο τρόπο έτσι ώστε να βελτιωθεί η ποιότητα των δεδομένων και στην συνέχεια και ακρίβεια των αποτελεσμάτων που θα εξαχθούν από αυτά. Κάποιες από τις προσεγγίσεις που έχουν αναφέρει αρκετοί επιστήμονες όπως θα παρουσιαστούν παρακάτω. Κάθε μια από αυτές τις προσεγγίσεις αποτελεί μια ξεχωριστή εργασία του καθαρισμού δεδομένων.

Deletion (Διαγραφή): Σε αυτή την προσέγγιση όλες οι εγγραφές που έχουν ελλιπή τιμές διαγράφονται κατά την διάρκεια της ανάλυσης δεδομένων. Θεωρείται ως η πιο απλή λύση αφού δεν χρειάζεται να γίνει κάποια προσπάθεια υπολογισμού των ελλιπών τιμών. Η διαδικασία της διαγραφής μπορεί να διαχωριστεί σε δυο διαφορετικούς τρόπους που μπορεί να πραγματοποιηθεί:

- *List-wise deletion (διαγραφή κατά λίστα):* Κάθε εγγραφή δεδομένων η οποία έχει μία ή περισσότερες τιμές κενές αφαιρείται ολόκληρη. Η συγκεκριμένη μέθοδος έχει γίνει η προεπιλεγμένη επιλογή αντιμετώπισης τέτοιων περιπτώσεων σε αρκετά στατιστικά λογισμικά για ανάλυση δεδομένων (Soley-Bori, 2013). Όμως στην υποθετική περίπτωση που τα δεδομένα είναι MCAR, τότε αυτος ο τρόπος οδηγεί σε μεροληψία (Williams, 2021). Ενώ αντίθετα στην περίπτωση που τα δεδομένα είναι μεγάλα και MCAR τότε αυτή η προσέγγιση θεωρείται λογική. Η διαγραφή κατά λίστα μπορεί να αποφέρει ανεπιθύμητα αποτελέσματα με την διαγραφή σημαντικών πληροφοριών, και ακόμα περισσότερο στην περίπτωση που ο αριθμός διαγεγραμμένων εγγραφών είναι μεγάλος.
- *Pairwise deletion (διαγραφή κατά ζεύγος):* Η διαγραφή κατά ζεύγος μειώνει αρκετά την απώλεια χρήσιμων δεδομένων σε σχέση με την διαγραφή κατά λίστα που παρουσιάστηκε προηγουμένως. Αυτό πραγματοποιείται με το να διαγράφει τις εγγραφές που έχουν ελλιπείς τιμές, αφού ελεγχθεί ότι αυτές λείπουν στην πραγματικότητα για το συγκεκριμένο σημείο δεδομένων (Allison, 2002). Η αδυναμία της συγκεκριμένης προσέγγισης είναι ότι υπάρχει περίπτωση να οδηγήσει σε έναν πίνακα διασυσχέτισης ο οποίος δεν είναι θετικά καθορισμένος, κάτι που αποτρέπει την περεταίρω ανάλυση όπως ο υπολογισμός εκτιμήσεων συντελεστών (coefficients) (JO & J., 1977). Αυτός ο τρόπος διαγραφής λειτουργεί με χαμηλά ποσοστά σφάλματος για δεδομένα MCAR και MAR.

Imputation (Αντικατάσταση): Η διαδικασία της αντικατάστασης αντικαθιστά τις ελλιπείς τιμές, με τιμές που έχουν προβλεφθεί. Οι τιμές του συνόλου δεδομένων

χρησιμοποιούνται με κάποιους τρόπους έτσι ώστε να αντικαταστήσουν τις τιμές που λείπουν.

- *Simple Imputation (Απλή αντικατάσταση):* Η απλή προσέγγιση αντικατάστασης αντικαθιστά τις τιμές που λείπουν για κάθε μεμονωμένη τιμή σύμφωνα με κάποιο ποιοτικό ή ποσοτικό χαρακτηριστικό των τιμών που υπάρχουν (GARCÍA-LAENCINA, SANCHO-GÓMEZ, & FIGUEIRAS-VIDAL, 2010). Η διαχείριση των χαμένων τιμών γίνεται με μεθόδους όπως τυχαία τιμή, μέσος όρος, διάμεσος τιμή των διαθέσιμων τιμών. Η απλή αντικατάσταση φαίνεται να χρησιμοποιείται πιο συχνά εξαιτίας της απλότητας των μεθόδων και της ευκολίας των τεχνικών της. Ωστόσο θα πρέπει να ειπωθεί ότι δημιουργούνται σφάλματα ή μη ρεαλιστικά αποτελέσματα σε σύνολα δεδομένων που έχουν υψηλές διαστάσεις όταν εφαρμόζονται τεχνικές απλή αντικατάστασης.
- *Regression Imputation (Αντικατάσταση με παλινδρόμηση):* Η παλινδρόμηση είναι μια στατιστική τεχνική η οποία προτιμάται αρκετά για την αντικατάσταση τιμών που λείπουν. Ο τρόπος αυτός χρησιμοποιεί μια τιμή την οποία προβλέπει μέσω ενός μοντέλου παλινδρόμησης και την αντικαθιστά σε περίπτωση που τα δεδομένα είναι MAR. Η διαδικασία παλινδρόμησης είναι έχει δυο φάσεις, πρώτα χρησιμοποιεί όλα τα δεδομένα που είναι συμπληρωμένα με σκοπό να δημιουργήσει το μοντέλο παλινδρόμησης και έπειτα σύμφωνα με αυτό το μοντέλο αντικαθιστά τις τιμές που λείπουν (Song & Shepperd, 2007). Αποτελεί μία αξιόλογή μέθοδος, διότι διατηρεί το μέγεθος του σετ δεδομένων παρά τις τιμές που λείπουν. Η αδυναμία βρίσκεται στο ότι χρειάζεται αρκετά μεγάλο δείγμα συμπληρωμένων τιμών για να μπορέσει να δημιουργήσει σταθερά αποτελέσματα. Επιπλέον, η απλή παλινδρόμηση (single regression) ακολουθείται από μονή καμπύλη για όλες τις τιμές που επρόκειτο να αντικατασταθούν χωρίς να παρουσιάζεται καμία εγγενής παραλλαγή στα δεδομένα (Little & Rubin, 2019). Ας υποθέσουμε ένα χαρακτηριστικό που περιέχει τιμές που λείπουν ενώ όλα τα υπόλοιπα χαρακτηριστικά είναι πλήρη, το μοντέλο παλινδρόμησης προσεγγίζει τις ελλιπείς τιμές μέσω των τιμών που υπάρχουν. Το πρώτο στάδιο είναι ο υπολογισμός ενός σετ από εξισώσεις παλινδρόμησης που θα έχουν την δυνατότητα να προβλέψουν τα ημιτελή δεδομένα μέσω των διαθέσιμων τιμών χρησιμοποιώντας μια πλήρη περίπτωση. Εν συνεχεία δημιουργούνται κάποιες τιμές για να αντικαταστήσουν τις τιμές που λείπουν. Στην περίπτωση που υπάρχουν ελλιπείς τιμές σε παραπάνω από ένα χαρακτηριστικά του σετ δεδομένων μας θα πρέπει να χρησιμοποιηθεί η παλινδρόμηση πολυμεταβλητών, η

οποία μετράει τον βαθμό για οποίο παραπάνω από μια ανεξάρτητη πρόβλεψη και παραπάνω από μια εξαρτημένες απαντήσεις σχετίζονται γραμμικά (Alexopoulos, 2010).

- *Hot-deck Imputation:* Η Hot-deck αντικατάσταση τιμών διαχειρίζεται τις τιμές που λείπουν αντιστοιχίζοντας ‘τες με άλλες τιμές από το σετ δεδομένων σύμφωνα με τις μεταβλητές «κλειδιά» οι οποίες είναι πλήρεις (Andridge & Little, 2010). Η μέθοδος αυτή έχει αρκετές παραλλαγές, αλλά εκείνη που επιτρέπει την μεταβλητότητα, στα δεδομένα που λείπουν, επιλέγει μια ομάδα από όλες τις περιπτώσεις. Η ομάδα αυτή ονομάζεται ομάδα δότη, και είναι πανομοιότυπη με τις περιπτώσεις που λείπουν δεδομένα σε αρκετές μεταβλητές και επιλέγει μια τυχαία περίπτωση από αυτή την ομάδα. Εν συνεχεία η τιμή που λείπει αντικαθίστανται από δεδομένα των τυχαίων αυτών περιπτώσεων. Επιπλέον υπάρχει μια άλλη τεχνική Hot-deck που χρησιμοποιεί αντικατάσταση από γειτονικό δότη, αντί να επιλέγει από μια ομάδα δοτών (Cheema, 2014). Αυτή η τεχνική αγνοεί την μεταβλητότητα που μπορεί να έχουν οι τιμές που λείπουν. Αξίζει να αναφερθούν δυο ακόμα τεχνικές Hot-deck, η weighted random Hot-deck και η weighted sequential Hot-deck. Σύμφωνα με την τεχνική weighted random hot-deck δεν υπάρχει περιορισμός ως προς τις φορές που ένα δότης επισημαίνεται, όμως οι δότες είναι επιλέγονται τυχαία από την ομάδα δοτών. Αντίθετα στην περίπτωση του weighted sequential Hot-deck τίθεται ένας περιορισμός φορών που μπορεί να χρησιμοποιηθεί ένας δότης για να αποτραπεί η ίδια τιμή δότη να υπάρχει σε αρκετές περιπτώσεις αντικατάστασης (Andridge & Little, 2010). Παρ’όλο που η συγκεκριμένη μέθοδος αντικατάστασης χρησιμοποιείται ευρέως ερευνητικά δεν έχει εδραιωθεί ως τεχνική αντιμετώπισης των ελλιπών τιμών συγκριτικά με άλλες μεθόδους.
- *Expectation maximization (Μεγιστοποίηση προσδοκιών):* Η συγκεκριμένη τεχνική είναι μια μέθοδος επανάληψης με σκοπό την διαχείριση των τιμών λείπουν από ένα αριθμητικό σετ δεδομένων, και ο αλγόριθμος χρησιμοποιεί την προσέγγιση «υπολογισμός – εκτίμηση – επανάληψη μέχρι την σύγκλιση». Σε κάθε επανάληψη υπάρχουν δυο στάδια, το στάδιο προσδοκίας και το στάδιο μεγιστοποίησης. Το πρώτο στάδιο εκτιμά τις τιμές που λείπουν βάσει των υπόλοιπων δεδομένων , ενώ στο δεύτερο στάδιο οι τρέχουσες εκτιμώμενες τιμές χρησιμοποιούνται για να μεγιστοποιήσουν την πιθανότητα όλων των δεδομένων (Lin & Tsai, 2020). Είναι και αυτή single imputation μέθοδος όπως και οι προηγούμενες που έχουν αναφερθεί και αποτελεί απλή μέθοδος αντιμετώπισης των τιμών που λείπουν με στόχο την εξοικονόμηση χρόνου.

Imputation methods inspired by machine learning: Είναι κάποιες μέθοδοι αντικατάστασης ελλιπών τιμών που βασίζονται στην μηχανική μάθηση, και για αυτό θεωρούνται εξελιγμένες τεχνικές που χρησιμοποιούν την ανάπτυξη προγνωστικών προσεγγίσεων για να διαχειριστούν τις τιμές που λείπουν από το σετ δεδομένων. Οι τεχνικές αυτές χρησιμοποιούν τα μη-ελλιπή δεδομένα για να υπολογίσουν αυτά που λείπουν χρησιμοποιώντας labelled ή unlabeled δεδομένα. Ειδικότερα, εάν τα διαθέσιμα δεδομένα περιέχουν χρήσιμες πληροφορίες για αυτές που λείπουν, μπορεί να υποστηριχτεί μια αντικατάσταση υψηλής προγνωστικής ακρίβειας. Παρακάτω παρουσιάζονται κάποιες από τι πιο ερευνημένες τεχνικές της κατηγορίας αυτής.

1. *K nearest neighbor classification:* ο αλγόριθμος KNN δουλεύει κατηγοριοποιώντας τους κοντινότερους γείτονες των τιμών που λείπουν και χρησιμοποιώντας τους για την αντικατάσταση των τιμών με παράμετρο την απόσταση των μεταξύ τους περιπτώσεων. Αρκετές υπολογισμοί απόστασης όπως Minkowski distance, Manhattan distance, Cosine distance, Hamming distance και Euclidean distance μπορούν να χρησιμοποιηθούν στην KNN μέθοδο αντικατάστασης, αν και η Ευκλείδεια απόσταση θεωρείται η πιο αποτελεσματική και για αυτό χρησιμοποιείται ευρέως στον υπολογισμό της απόστασης (Amirteimoori & Kordrostami, 2010). Η KNN μέθοδος είναι ευέλικτη τόσο σε συνεχή δεδομένα όσο και σε διακριτά καθώς μπορεί να εφαρμοστεί και ως χειριστής πολλαπλών ελλιπών δεδομένων. Ωστόσο, έχει και μειονεκτήματα όπως η χαμηλή ακρίβεια κατά των τιμών που υπολογίζει προς αντικατάσταση καθώς δημιουργεί ψευδής συσχετισμούς με τα υπάρχοντα δεδομένα (Beretta & Santaniello, 2016). Ακόμα το γεγονός ότι πραγματοποιεί αναζήτηση σε όλο το σετ δεδομένων, αυξάνει σημαντικά το υπολογιστικό χρόνο (Acuña & Rodriguez, 2004).
2. *Support Vector Machine (SVM):* Ένας ακόμα κοινός αλγόριθμος μηχανικής μάθησης είναι ο SVM που χρησιμοποιείται σε ευρέως για τιμές που λείπουν. Ο SVM, για ένα labeled δείγμα εκμάθησης, προσπαθεί να ανιχνεύσει κάποιο βέλτιστο υπέρ-επίπεδο έτσι ώστε η απόσταση από το υπέρ-επίπεδο αυτό μέχρι των κοντινότερων σημείων δεδομένων να μεγιστοποιείται (Stewart, Zeng, & Wu, 2018). Τα υπέρ-επίπεδα αυτά ορίζονται ως εξής:

- $w \cdot x1 + b \geq +1$ όταν το $yi = +1$
- $w \cdot x1 + b \leq -1$ όταν το $yi = -1$

w: παράγοντας βάρους, x: παράγοντας εισόδου, b:

3. Decision Tree: Ο αλγόριθμος δένδρου απόφασης είναι ένας ακόμα αλγόριθμος μηχανικής μάθησης που απεικονίζει όλα τα πιθανά αποτελέσματα και τις πιθανές διαδρομές που οδηγούν σε αυτά με την μορφή μιας δομής δένδρου. Η αντικατάσταση των τιμών που λείπουν μέσω αυτής της μεθόδου γίνεται με την δημιουργία δένδρων απόφασης με σκοπό την παρατήρηση των ελλιπών τιμών των μεταβλητών και μετέπειτα να γεμίσει τις τιμές που λείπουν από τις μεταβλητές χρησιμοποιώντας το αντίστοιχο δένδρο (Twala, 2009). Η πρόβλεψη τιμών που λείπουν εμφανίζεται στην συνέχεια στον κόμβο φύλλου. Αυτός ο αλγόριθμος διαχειρίζεται τόσο αριθμητικές όσο και κατηγορικές μεταβλητές, να εντοπίσει αρκετές μεταβλητές και να εξαλείψει τις υπόλοιπες. Ωστόσο αξίζει να σημειωθεί ότι υπάρχει πιθανότητα να παράγουν δένδρο μεγάλης πολυπλοκότητας που τείνει να είναι χρονοβόρο υπολογιστικά, αλλά με χαμηλή πιθανότητα σφάλματος.

4.3 Duplicate Data (**Διπλότυπες εγγραφές δεδομένων**)

Η ανίχνευση διπλότυπων εγγραφών (duplicate detection) συχνά ονομάζεται και αποδιπλασιασμός δεδομένων (data deduplication) αναφέρεται στην διαδικασία της αναγνώρισης πλειάδων που έχουν μια ή περισσότερες συσχετίσεις και αναφέρονται σε ένα αντικείμενο του πραγματικού κόσμου. Η διαδικασία αφαίρεσης των διπλότυπων εγγραφών περιέχει πολλά βήματα και επιλογές, όπως ο σχεδιασμός μετρικών ομοιότητας για την αξιολόγηση ενός ζεύγους εγγραφών, εκμάθηση κατηγοριοποιητών που αποφασίζουν αν ένα ζεύγος εγγραφών είναι διπλότυπο ή όχι, ομαδοποίηση όλων των εγγραφών για δημιουργία συστάδων εγγράφων που παρουσιάζουν το ίδιο πραγματικό αντικείμενο, ενοποίηση συστάδων σε μοναδικές αναπαραστάσεις, σχεδιασμός στρατηγικών αποκλεισμού ή κατανομής για την κλιμάκωση της διαδικασίας αποδιπλασιαμού και εμπλοκή ανθρώπινου παράγοντα για την λήψη αποφάσεων αν ένα ζεύγος είναι διπλότυπο όταν το μηχάνημα δεν μπορεί να το προσδιορίσει (Chu, 2019).

4.3.1 **Τύποι διπλότυπων δεδομένων**

Σε ένα σετ δεδομένων είναι πολύ πιθανό να βρεθούν διπλότυπα αντικείμενα, εξαιτίας των διαφόρων πηγών που τα παράγουν οι οποίες μπορεί να φέρουν κοινά αντικείμενα ή και

λάθη κατά την εισαγωγή των δεδομένων από ανθρώπινο παράγοντα, όπως ένας χρήστης να έχει λογαριασμό δύο φορές σε μια πλατφόρμα με μικρή απόκλιση του ονόματος του (Κώστας Ξυδιάς και Κωνσταντίνος Ξυδιάς). Παρακάτω αναγνωρίζονται οι δύο τύποι διπλότυπων που μπορούμε σε ένα σετ δεδομένων:

i. *Διπλότυπες στήλες*: Όταν δυο χαρακτηριστικά ενός πίνακα έχουν τι ίδιες τιμές ακριβώς, όπως φαίνεται και στην εικόνα 9. Στο παράδειγμα πού παρουσιάζεται στην εικόνα δυο στήλες έχουν ακριβώς τις ίδιες ημερομηνίες, οπότε είναι πολύ πιθανό μετά από αυτή την παρατήρηση να αφαιρεθεί μια από τις δύο αφού με την μια μόνο τα αποτελέσματα των αναλύσεων θα είναι ακριβώς τα ίδια. Κάτι τέτοιο μπορεί να έχει προκύψει εξαιτίας κάποιας ένωσης πινάκων δεδομένων από διαφορετικά databases τα οποία έχουν ένα χαρακτηριστικό κοινό, απλά με διαφορετική ονομασία.

#	Car Make	Car Model	Car Year	...	...	Sell Year	...
1	Toyota	Camry	2018	...	...	2018	...
2	Toyota	Corolla	2019	...	...	2019	...
3	Toyota	Camry	2018	...	...	2018	...
4	Toyota	Corolla	2019	...	...	2019	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
9999	Toyota	Camry	2018	...	...	2018	...
10000	Toyota	Camry	2018	...	...	2018	...

***Εικόνα 9**: Παράδειγμα πίνακα με διπλότυπες στήλες*

ii. *Διπλότυπες σειρές*: Όταν δυο χαρακτηριστικά δεν έχουν ίδιες τιμές στην ίδια σειρά, αλλά επαναλαμβάνονται σε άλλες σειρές με ακριβώς τον ίδιο τρόπο, όπως φαίνεται

#	Car Make	Car Model	Car Year	...	...	Sell Year	...
1	Toyota	Camry	2018	...	...	2018	...
2	Toyota	Corolla	2019	...	...	2019	...
3	Toyota	Camry	2018	...	...	2018	...
4	Toyota	Corolla	2019	...	...	2019	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
9999	Toyota	Camry	2018	...	...	2018	...
10000	Toyota	Camry	2018	...	...	2018	...

***Εικόνα 10:** Παράδειγμα πίνακα με παρόμοιες πλειάδες*

στην παρακάτω εικόνα. Στο παράδειγμα της εικόνας 10 παρατηρούμε ότι κάποια μοντέλα αυτοκινήτων έχουν ακριβώς την ίδια ημερομηνία, οπότε μετά από αυτή την

παρατήρηση ένας αναλυτής θα αφαιρούσε τις σειρές που είναι ακριβώς ίδιες αφού δεν εξυπηρετούν κάπως τις αναλύσεις εκτός από την καταγραφή της ποσότητας αυτών.

4.3.2 **Κατηγορίες προσέγγισης προβλήματος**

Η αντιμετώπιση των διπλότυπων είναι πολύ βασικό κομμάτι του καθαρισμού των δεδομένων, αφού μπορούν να οδηγήσουν τις αναλύσεις σε εντελώς διαφορετικά αποτελέσματα, ή ακόμα να λαμβάνουν χώρο και χρόνο και συνεπώς χρηματικός κόστος σε οργανισμούς που δουλεύουν με αυτά. Η διαδικασία αντιμετώπισης δηλαδή ανίχνευσης των διπλότυπων παρουσιάζεται παρακάτω μέσω διαφορετικών μεθόδων που μπορεί να πραγματοποιηθεί. Μετά την ανίχνευση αυτών είναι λογικό επόμενο η απομάκρυνση τους από τα δεδομένα, που προορίζονται να δώσουν τις αναφορές των αναλύσεων, με σκοπό την μεγαλύτερη ακρίβεια και παραγωγικότητα. Οι μέθοδοι που προαναφέρθηκαν χωρίζονται και αυτές σε δυο κατηγορίες προσέγγισης του προβλήματος (Elmagarmid, Verykios, & Ipeirotis, 2007):

- Προσεγγίσεις που βασίζονται στην εκμάθηση αλγορίθμων μέσω δεδομένων εκμάθησης για να μπορεί να αντιστοιχίζει εγγραφές. Αυτή η κατηγορία περιέχει προσεγγίσεις με βάση τις πιθανότητες και κάποιες supervised μηχανικές μάθησης.
- Προσεγγίσεις που βασίζονται στο γνωσιακό πεδίο από το οποίο έχουμε τα δεδομένα ή σε γενικές μετρικές απόστασης με σκοπό την αντιστοίχιση πανομοιότυπων εγγραφών. Η συγκεκριμένη κατηγορία περιλαμβάνει προσεγγίσεις που χρησιμοποιούν declarative γλώσσες προγραμματισμού για την αντιστοίχιση, και προσεγγίσεις που αναπτύσσουν κατάλληλες μετρικές απόστασης με σκοπό την ανίχνευση διπλότυπων.

- *Notation*: Η λογική που ακολουθείται με την μέθοδο notation είναι η εξής, έστω δύο πίνακες Α και Β που ελέγχεται ο βαθμός αντιστοιχίας ενώ έχουν n συγκρίσιμα πεδία ο καθένας. Κάθε ζευγάρι πλειάδων {α, β}, όπου το κάθε ένα ανήκει στον Α πίνακα και στον Β πίνακα αντίστοιχα, ανήκει σε μια ομάδα πλειάδων Μ ή U. Στην Μ ομάδα ανήκουν τα ζευγάρια που είναι πανομοιότυπα ενώ στην U αυτά που δεν είναι. Κάθε ζευγάρι παρουσιάζεται ως ένα τυχαίο διάνυσμα χ=[χ1,χ2,….,χn] με n στοιχεία που αντιστοιχούν στα n συγκρίσιμα πεδία του κάθε πίνακα. Τέλος κάθε χi δείχνει το

βαθμό συμφωνίας του i πεδίου για κάθε ζευγάρι εγγραφών α και β. Αρκετές προσεγγίσεις τέτοιας μεθοδολογίας χρησιμοποιούν δυαδικές τιμές όπως χi=1 αν ταιριάζουν και χi=0 αν όχι.

- *Probabilistic Matching Methods:*
    - *Bayes decision rule for minimum error:* Αναφέρεται στην μέθοδο στην οποία πιθανολογικά αποφασίζεται αν δυο εγγραφές είναι ίδιες, οπότε και διπλότυπες, ή αν δεν είναι. Έστω χ ένα τυχαίο διάνυσμα σύγκρισης πεδίων ενός ζευγαριού εγγραφών (α, β) και ο στόχος είναι να ληφθεί η απόφαση σε ποια ομάδα ανήκουν M ή U. Ο κανόνας απόφασης βασισμένος στις πιθανότητες μπορεί να είναι ότι (α, β) κατηγοριοποιείται στο M αν η πιθανότητα το χ να ανήκει στο M είναι μεγαλύτερη από αυτή του να ανήκει στο U. Χρησιμοποιώντας το θεώρημα Bayes ο προηγούμενος κανόνας γίνεται, το (α, β) ανήκει στο M αν η αναλογία l(x) είναι μεγαλύτερη από το όριο τιμής της πιθανολογικής αναλογίας για την απόφαση. Το l(x) είναι αναλογία της πιθανότητας χ να ανήκει στο M δια της πιθανότητας του χ να ανήκει στο U. Αυτή η μέθοδος του Bayes με σκοπό την μικρότερη πιθανότητα λάθους, αποτελεί έναν βέλτιστο κατηγοριοποιητή, να και δεν χρησιμοποιείται συχνά εξαιτίας του ότι πρέπει να είναι γνωστές οι κατανομές που προαναφέρθηκαν.
    - *Bayes decision rule for minimum cost:* Στην πραγματικότητα, η μέγιστη μείωση του λάθους δεν αποτελεί το καλύτερο κριτήριο απόφασης, αφού η λάθος ταξινόμηση ενός ζευγαριού εγγραφών στις M και U. Σε αυτή την μέθοδο αναθέτεται το κόστος Cij, για κάθε περίπτωση στην οποία το χ ανήκει στην κλάση i στην πραγματικότητα ανήκει στην j. Ο κανόνας είναι ο εξής, ένα ζευγάρι (α, β) ανήκει στην M αν το προσδοκώμενο κόστος του χ να ανήκει στο M είναι μικρότερο από το να ανήκει στο U, και το αντίθετο για να ανήκει στο U.
    - *Decision with a Reject Region :*Χρησιμοποιώντας την μέθοδο Bayes κανόνα απόφασης, όταν οι παράμετροι των κατανομών είναι γνωστοί επιστρέφουν τα βέλτιστα αποτελέσματα. Όμως ακόμα και στο ιδανικό σενάριο όπου η l(x) αναλογία βρίσκεται πολύ κοντά στο όριο που έχει τεθεί, το λάθος ή το κόστος μπορεί να είναι υψηλό. Οι Fellegi και Sunter (Fellegi & Sunter, 1969), πρότειναν να προστεθεί ακόμα μια κλάση ονομαζόμενη reject μαζί με τις M

και U. Αυτή η προστιθέμενη κατηγορία θα αφορά τις περιπτώσεις που δεν μπορεί να παρθεί μια ξεκάθαρη απόφαση, και χρειάζεται επανεξέταση. Αυτά τα ζευγάρια εγγραφών επανεξετάζονται από κάποιον ειδικό που αποφασίζει σε ποια κλάση ανήκουν. Θέτοντας όρια για σφάλμα υπό όρους, μπορεί να οριστεί η περιοχή reject καθώς και πιθανότητα reject , η οποία μετράει την πιθανότητα ένα ζευγάρι εγγραφών να επανεξεταστεί από κάποιον ειδικό.

- Supervised Learning: Τα supervised learning συστήματα βασίζονται στην ύπαρξη των δεδομένων εκπαίδευσης στην μορφή των ζευγαριών εγγραφών, τα οποία διαθέτουν ετικέτα διπλότυπων ή διαφορετικών. Οι supervised τεχνικές συμπεριφέρονται σε κάθε ζευγάρι (α, β) ανεξάρτητα, όπως ακριβώς και στις πιθανολογικές μεθόδους. Ο Cochinwala (Cochinwala, Kurienb, GailLalka, & DennisShashac, 2001) χρησιμοποιεί τον αλγόριθμο CART ο οποίος παράγει δένδρα ταξινόμησης και παλινδρόμησης, τον γραμμικό αλγόριθμο διάκρισης ο οποίος παράγει γραμμικό συνδυασμό των παραμέτρων για να ταξινομήσει τα δεδομένα ανάλογα με τις ομάδες ταξινόμησης και μια προσέγγιση διανυσματικής κβαντοποίησης ο οποίος αποτελεί την γενίκευση του αλγορίθμου του κοντινότερου γείτονα (nearest neighbor), με τον CART να αποτελεί την καλύτερη λύση. Ο Bilenko (Bilenko, Mooney, Cohen, & Fienberg, 2003) χρησιμοποιεί την προσέγγιση του SVM (Support Vector Machine) η οποία δείχνει ότι είναι καλύτερη από τις απλές προσεγγίσεις.
- *Active-Learning-Based Techniques:* Ένα από τα προβλήματα με τις supervised τεχνικές μάθησης είναι η απαίτηση για μεγάλο αριθμό παραδειγμάτων εκπαίδευσης (training data). Ενώ είναι εύκολο να δημιουργηθεί ένας μεγάλος αριθμός ζευγών εκπαίδευσης που είναι είτε σαφώς μη διπλότυπα είτε σαφώς διπλότυπα, είναι πολύ δύσκολο να δημιουργηθούν διφορούμενες περιπτώσεις που θα βοηθούσαν στη δημιουργία ενός ταξινομητή υψηλής ακρίβειας. Με βάση αυτή την παρατήρηση, ορισμένα συστήματα ανίχνευσης διπλότυπων χρησιμοποιούν τεχνικές Active Learning [για να εντοπίσουν αυτόματα τέτοια διφορούμενα ζεύγη. Σε αντίθεση με έναν "συνηθισμένο" εκπαιδευόμενο που εκπαιδεύεται χρησιμοποιώντας ένα στατικό σύνολο δεδομένων εκπαίδευσης, ένας active εκπαιδευόμενος επιλέγει ενεργά υποσύνολα περιπτώσεων από δεδομένα χωρίς ετικέτα, τα οποία, όταν τους τοποθετείται ετικέτα, θα παρέχουν το υψηλότερο κέρδος πληροφοριών στον εκπαιδευόμενο. Ο Sarawagi και ο Bhamidipaty (Sarawagi & Bhamidipaty, 2002) σχεδίασαν το ALIAS, ένα σύστημα

ανίχνευσης διπλότυπων που βασίζεται στη μάθηση, το οποίο χρησιμοποιεί την ιδέα μιας περιοχής απόρριψης (reject region), για να μειώσει σημαντικά το μέγεθος του σετ δεδομένων εκπαίδευσης. Η κύρια ιδέα πίσω από το ALIAS είναι ότι τα περισσότερα διπλότυπα και μη ζεύγη είναι σαφώς διακριτά. Για τέτοια ζεύγη, το σύστημα μπορεί να τα κατηγοριοποιήσει αυτόματα σε U και M χωρίς να χρειάζεται χειροκίνητη προσθήκη ετικετών. Αυτό το μοντέλο απαιτεί από τους ανθρώπους να προσθέτουν ετικέτες μόνο σε περιπτώσεις υψηλής αβεβαιότητας.

- *Distance-Based Techniques:* Ένας τρόπος για να αποφευχθεί η ανάγκη για δεδομένα εκπαίδευσης είναι ο καθορισμός μιας μετρικής απόστασης για τις εγγραφές, η οποία δεν χρειάζεται ρυθμιστεί σύμφωνα με τα δεδομένα εκπαίδευσης. Χρησιμοποιώντας τη μετρική απόστασης και ένα κατάλληλο όριο ομοιότητας, είναι δυνατό να αντιστοιχιστούν παρόμοιες εγγραφές. Μια προσέγγιση είναι να αντιμετωπίζετε μια εγγραφή ως μεγάλο πεδίο και να χρησιμοποιείται μια από τις μετρικές απόστασης που περιγράφονται στην αντιστοίχιση πεδίων για να προσδιοριστεί ποιες εγγραφές είναι παρόμοιες. Ο Ananthakrishna (Ananthakrishna, Chaudhuri, & Ganti, 2002) σε μια άλλη προσέγγιση, περιέγραψε μια μετρική ομοιότητας που χρησιμοποιεί εκτός από την ομοιότητα του κειμένου και την ομοιότητα συνεμφάνισης σε μια βάση δεδομένων. Όπως για παράδειγμα στο πεδίο της περιοχής μπορεί να βρεθεί το 'CA' και το 'California' τα οποία έχουν μικρή ομοιότητα κειμένου. Όπως και τα 'San Diego', 'Los Angeles', 'San Francisco' τα οποία έχουν foreign keys που δείχνουν στα 'CA' και 'California'. Επομένως, είναι δυνατόν να συμπεράνουμε ότι το 'CA' και το 'California' είναι ισοδύναμα. Επί της ουσίας δείχνουν ότι με τη χρήση πληροφοριών συνεμφάνισης foreign key, μπορεί να βελτιωθεί ουσιαστικά η ποιότητα της ανίχνευσης διπλότυπων σε βάσεις δεδομένων που χρησιμοποιούν πολλούς πίνακες για την αποθήκευση των εγγραφών μιας εγγραφής.

4.3.3 **Τύποι αποδιπλασιασμού πληροφοριών στο** Cloud Computing

Οι μεγάλες ποσότητες δεδομένων που αποθηκεύουν καθημερινά οι χρήστες είναι πολύ πιθανό να δημιουργήσουν επαναλαμβανόμενα δεδομένα τα οποία δεσμεύουν χώρο σε συστήματα δίχως να εξυπηρετούν κάποιο σκοπό. Αυτό συμβαίνει συχνότερα στα αποκαλούμενα «σύννεφα» (clouds), τα οποία αποτελούν τους χώρους αποθήκευσης

μεγάλου όγκου δεδομένων εφαρμογών από το διαδίκτυο, οπότε και η παρουσίαση των παρακάτω τύπων αποδιπλασιασμού αφορούν το cloud computing και λόγω της μεγάλης χρήσης τους στην επικαιρότητα. Η διαδικασία αποδιπλασιασμού των δεδομένων χωρίζεται σε κάποιους συγκριμένους τύπους, όπως είναι ο Source Deduplication, ο Target Deduplication και ο Global Deduplication.

a) *Source Deduplication*: Ορίζεται η απομάκρυνση των διπλότυπων από το σετ των δεδομένων πριν την μεταφορά αυτών προς τον τελικό χώρο αποθήκευσης τους. Λειτουργεί στο λογισμικό του πελάτη το οποίο επικοινωνεί με τον διακομιστή που είναι αποθηκευμένα τα υπόλοιπα δεδομένα, έτσι ώστε να συγκρίνει τα νέα μπλοκ δεδομένων με τα παλιά για να βρει τυχών διπλότυπα.
b) *Target Deduplication*: Αποβάλλει όλες τις πλεονάζουσες πληροφορίες μέσα στο σύστημα αναμονής, συχνά στην VTL[2]. Ακόμα μειώνει το όριο του αποθηκευτικού χώρου που απαιτείται για τις πληροφορίες αναμονής. Ο τύπος target deduplication χωρίζεται σε δυο επιμέρους κατηγορίες:
    i. *Inline Deduplication*: Είναι η αφαίρεση των διπλότυπων από τα δεδομένα πριν ή κατά την διάρκεια της εγγραφής τους στον χώρο αποθήκευσης. Επιπλέον μειώνει τον όγκο των δεδομένων μιας εφαρμογής και τη χωρητικότητα που χρειάζεται για αποθήκευση των δεδομένων, σε σύγκριση με την επόμενη υποκατηγορία. Βέβαια με αυτόν τον τρόπο η inline deduplication διαδικασία μπορεί να επιβραδύνει την αποθήκευση των δεδομένων, διότι οι συσκευές της λαμβάνουν χώρα στην διαδρομή των δεδομένων μεταξύ των διακομιστών και των συστημάτων αποθήκευσης του δίσκου.
    ii. *Post-Process Deduplication*: Γνωστή και ως ασύγχρονη μέθοδος αποδιπλασιασμού των δεδομένων, είναι η ανάλυση και αφαίρεση των επαναλαμβανόμενων δεδομένων αφότου έχει ολοκληρωθεί η διαδικασία αποθήκευσης. Η PPD κατηγορία αντιπαραβάλλεται της inline διαδικασίας κατά την οποία τα περιττά δεδομένα αναγνωρίζονται κατά την διάρκεια της αντιγραφής των δεδομένων.

[2] Μια βιβλιοθήκη εικονικής ταινίας (VTL), είναι μια τεχνολογία απεικόνισης της αποθήκευσης δεδομένων, και χρησιμοποιείται συνήθως για σκοπούς δημιουργίας αντιγράφων ασφαλείας και ανάκτησης.

c) *Global Deduplication*: Αποτελεί έναν τύπο αποδιπλασιασμού δεδομένων κατά την διάρκεια αντιγραφής αυτών σε πολλαπλές συσκευές. Όταν τα δεδομένα αποστέλλονται από τον ένα κόμβο στον άλλο, ο δεύτερος κόμβος αναγνωρίζει ότι ο πρώτος έχει ήδη κάποιο αντίγραφο των δεδομένων και δεν το προσθέτει επιπλέον. Επειδή τα μεγάλα κέντρα δεομένων (data centers) χρησιμοποιούν αρκετούς προορισμούς αποθήκευσης δεδομένων, αυτός ο τύπος προτιμάται για την κατάργηση των διπλότυπων εγγραφών, επειδή τα αφαιρεί σε όλους τους διαφορετικούς προορισμούς. Ένας οργανισμός που διαθέτει μεγάλο όγκο δεδομένων, στην περίπτωση που ενεργήσει σύμφωνα με τους άλλους τύπους αφαίρεσης είναι πιθανό να προκαλέσει συμφόρηση.

5. Python frameworks **κατάλληλα για** Data Cleaning

Η γλώσσα προγραμματισμού Python είναι μια γλώσσα ανοιχτού κώδικα και υψηλού επιπέδου η οποία παρέχει πολύ καλή προσέγγιση στον αντικειμενοστραφή προγραμματισμό. Αποτελεί μια από τις καλές γλώσσες προγραμματισμού για να χρησιμοποιήσει κάποιος επιστήμονας δεδομένων (data scientist) για πολλά διαφορετικά εγχειρήματα και εφαρμογές. Η Python παρέχει μεγάλη λειτουργικότητα όσον αφορά την αντιμετώπιση μαθηματικών, στατιστικών και επιστημονικών λειτουργειών. Ακόμα διαθέτει ποικιλία βιβλιοθηκών σχετικά με την επιστήμη των δεδομένων. Η ευκολία χρήσης και κατανόησης της σύνταξης της Python την έχουν φέρει σε περίοπτη θέση σε όλα τα πεδία της επιστημονικής κοινότητας, αφού και ένας επιστήμονας χωρίς προγραμματιστικό υπόβαθρο μπορεί να την χρησιμοποιήσει με ελάχιστη εκπαίδευση. Αρκετοί μηχανικοί δεδομένων αναφέρουν πως τα deep learning frameworks που διατίθενται μέσω των Python APIs, σε συνεργασία με τα επιστημονικά πακέτα, έχουν φτάσει την Python να είναι αρκετά παραγωγική και ευέλικτη. Υπάρχει μεγάλη εξέλιξη στα frameworks της Python που αφορούν το deep learning και μια συνεχής αναβάθμιση αυτών.

Στο πεδίο που μελετάται, στην επιστήμη των δεδομένων, υπάρχουν αρκετές αξιόλογες και εύχρηστες βιβλιοθήκες οι οποίες χρησιμοποιούνται στην διαδικασία «μεταμόρφωσης» των δεδομένων με σκοπό την ανάλυση της σε μεταγενέστερο χρόνο. Με τον όρο μορφοποίηση εννοείται οι διαδικασίες που γίνονται προκειμένου να υπάρχει υψηλός βαθμός ποιότητας στα δεδομένα, όπως είναι ο καθαρισμός (cleansing), η εφαρμογή ενιαίων κανόνων μορφοποίησης (standardization), αποδιπλασιασμός (deduplication), απομάκρυνση άχρηστων δεδομένων (verification) και ταξινόμηση (sorting). Ο καθαρισμός των δεδομένων ,που αφορά την παρούσα εργασία, διαθέτει κάποιες από τι πιο γνωστές βιβλιοθήκες όπως είναι η Pandas, η NumPy , η SciPy και η Matplotlib. Η βιβλιοθήκη pandas είναι κατάλληλη για τον καθαρισμό, ανάλυση και εξερεύνηση δεδομένων σε μορφή πινάκων. Η βιβλιοθήκη NumPy είναι και αυτή κατάλληλη για δεδομένα τοποθετημένα σε πίνακες και παρέχει αρκετές μαθηματικές συναρτήσεις στους χρήστες που την χρησιμοποιούν. Η βιβλιοθήκη SciPy η οποία χρησιμοποιεί NumPy από κάτω, έχει δημιουργηθεί με σκοπό την παροχή βοηθητικών λειτουργιών για βελτιστοποίηση, στατιστικά και επεξεργασία σημάτων. Τέλος η βιβλιοθήκη Matplotlib παρέχει χαμηλού επιπέδου γραφήματα που βοηθούν στην

οπτικοποίηση των δεδομένων. Αυτές οι βιβλιοθήκες αποτελούν συνήθως και την βάση για τις υπόλοιπες βιβλιοθήκες που έχουν δημιουργηθεί για τον καθαρισμό δεδομένων που θα παρουσιαστούν παρακάτω.

5.1 Missingno

(missingno, n.d.)

Η βιβλιοθήκη missingno παρέχει ένα μικρό σύνολο ευέλικτων και εύχρηστων εργαλείων οπτικοποίησης των δεδομένων που λείπουν, και βοηθητικών προγραμμάτων που επιτρέπουν την λήψη γρήγορης οπτικής περίληψης της πληρότητας (ή της έλλειψης) ενός συνόλου δεδομένων. Η πιο σημαντική συνάρτηση είναι η msno.matrix() η οποία προβάλει μια οθόνη που αφορά την πυκνότητα των δεδομένων. Με αυτόν τον τρόπο επιτρέπει γρήγορη οπτική εικόνα βοηθώντας στην επιλογή μοτίβων για τον καθαρισμό δεδομένων. Μια ακόμα σημαντική συνάρτηση είναι η heatmap() που μετρά και απεικονίζει πόσο έντονα επηρεάζει η παρουσία ή η απουσία μιας μεταβλητής την παρουσία μιας άλλης. Άλλες απεικονίσεις μπορεί να είναι με τη μορφή dendrogram και bar chart.

Λειτουργεί μετατρέποντας πίνακες δεδομένων σε «μάσκες» boolean με βάση το αν οι μεμονωμένες καταχωρίσεις περιέχουν δεδομένα (που αξιολογούνται σε αληθές) ή μένουν κενές (που αξιολογούνται σε ψευδείς). Αυτός ο «ελλιπής πίνακας» στη συνέχεια εκτίθεται στην αξιολόγηση των χρηστών μέσω μιας ποικιλίας οπτικοποιήσεων δεδομένων ειδικού σκοπού. Η missingno είναι συμβατή απευθείας με dataframes της pandas.

5.2 Datacleaner

(Datacleaner, n.d.)

Το datacleaner είναι μια βιβλιοθήκη ανοιχτού κώδικα η οποία χρησιμοποιείται για να αυτοματοποιήσει την διαδικασία του καθαρισμού των δεδομένων. Η συγκεκριμένη βιβλιοθήκη έχει χτιστεί χρησιμοποιώντας το Pandas Dataframe και το scikit-learn τα οποία βοηθούν στην προεπεξεργασία των δεδομένων. Βέβαια είναι μια βιβλιοθήκη η οποία συνεχώς προστίθενται νέα χαρακτηριστικά από διάφορους που προσπαθούν να συνεισφέρουν στην βελτίωση της συγκεκριμένης βιβλιοθήκης. Κάποιες από τις κύριες δουλειές καθαρισμού που γίνονται μέσω της datacleaner είναι:

- Αποβάλλει τις τιμές που είναι null
- Αντικαθιστά τις null τιμές με το μέσο όρο (mean) για τα αριθμητικά και με τον διάμεσο (median) για τον τα κατηγορικά δεδομένα.
- Κωδικοποιεί μη αριθμητικά δεδομένα με αριθμητικά ισοδύναμα.

Όπως σωστά αναφέρεται στο documentation της βιβλιοθήκης, το datacleaner δεν κάνει μαγικά μια διαδικασία καθαρισμού η οποία απαιτεί μεγάλη πολυπλοκότητα. Το datacleaner επί της ουσίας δέχεται δεδομένα τα οποία έχουν στηθεί σε ένα dataframe της Pandas και αφού έχει γίνει η διαδικασία της μορφοποίησης των δεδομένων ανά στήλη, μπορεί να κάνει πράξη κάποιες συγκεκριμένες τεχνικές αφαίρεσης ελλιπών τιμών με αυτοματοποιημένο τρόπο έτσι ώστε να μην επηρεαστούν αρνητικά ή θετικά τα στοιχεία των στηλών. Αρχικά μετατρέπει όλες τις στήλες που δεν είναι σε αριθμητική μορφή ως κατηγορικά, δηλαδή αντικαθιστά το περιεχόμενο τους με έναν συγκεκριμένο αριθμό, για παράδειγμα αν σε μια στήλη έχουμε την πόλη με όνομα “New York” θα αντικαταστήσει το όνομα με έναν αριθμό και αυτός ο αριθμός θα ισχύει για όλες τις εγγραφές που έχουν το όνομα αυτό. Έπειτα από αυτή την διαδικασία βρίσκει τις κενές θέσεις των δεδομένων (NaN) και τα αντικαθιστά ανάλογα με τον τύπο της στήλης, αν για παράδειγμα υπάρχει μια στήλη “Amount” που έχει κενά, τα αντικαθιστά με τον μέσο όρο τον υπόλοιπων τιμών της στήλης. Στην περίπτωση των κατηγορικών δεδομένων χρησιμοποιεί την τεχνική των διαμέσων, και αντικαθιστά τις κενές θέσεις των κατηγορικών δεδομένων μιας στήλης όπως είναι για παράδειγμα η στήλη “City Name” με τον διάμεσο αριθμό μεταξύ των αριθμών που είχε αντικαταστήσει στην προαναφερθείσα διαδικασία. Η π

5.3 Dora

(Dora, n.d.)

Η βιβλιοθήκη Dora έχει δημιουργηθεί με σκοπό να απλοποιήσει όσο το δυνατό περισσότερο την διερευνητική ανάλυση των δεδομένων αυτοματοποιώντας κάποιες επαναλαμβανόμενες εργασίες που καταναλώνουν αρκετό χρόνο. Χρησιμοποιεί κατά βάση τις βιβλιοθήκες scikit-learn, pandas και matplotlib. Οι συναρτήσεις που περιέχονται στην Dora είναι αρκετά χρήσιμες για καθαρισμό δεδομένων, εξαγωγή γνωρισμάτων από τα δεδομένα και οπτικοποίηση του σετ δεδομένων. Επιπροσθέτως χρησιμοποιείται για μετασχηματισμούς των δεδομένων. Η Dora επειδή χρησιμοποιεί κατά βάση pandas μπορεί

να αποθηκεύσει μέσω της συνάρτησης Dora() απευθείας ένα σετ δεδομένων ως dataframe. Ακόμα όπως προαναφέρθηκε έχει συναρτήσεις που θεωρούνται κατάλληλες για κάποιες εργασίες καθαρισμού δεδομένων, όπως είναι impute_missing_values() και scale_input_values() οι οποίες χρησιμοποιούνται πάνω σε κάποιο dataframe. Η πρώτη αντικαθιστά τις ελλιπείς αριθμητικές τιμές με τον μέσο όρο της στήλης που βρίσκεται ενώ η δεύτερη αντικαθιστά τις ελλιπείς αριθμητικές τιμές με τον διάμεσο της στήλης που βρίσκεται. Επίσης η Dora περιέχει δυο συναρτήσεις που δεν συμβάλλουν μόνο στον καθαρισμό δεδομένων, αλλά σίγουρα μπορούν να βοηθήσουν την διαδικασία αυτή, η remove_feature() η οποία ως παράμετρο παίρνει ένα γνώρισμα του πίνακα το οποίο δεν χρειάζεται και το διαγράφει, και η extract_ordinal_feature() η οποία παίρνει ως παράμετρο μια στήλη η οποία έχει κατηγορικά δεδομένα και τα μετατρέπει σε one-hot encoding. Η βιβλιοθήκη dora με τις λειτουργίες της καλύπτει τα πεδία εισαγωγής δεδομλενων, καθαρισμού δεδομένων , αφαίρεση/εξαγωγής γνωρίσματων, οπτικοποίησης δεδομένων και εγκυρότητα μοντέλων

5.4 Scrubadub

(scrubadub, n.d.)

Η βιβλιοθήκη ανοικτού κώδικα Scrubadub παρέχει μια πολύ σημαντική δυνατότητα στους επιστήμονες των δεδομένων κάποιων συγκεκριμένων πεδίων, όπως είναι της υγείας και των ελεγκτικών-λογιστικών. Αυτό διότι αυτά τα πεδία χρησιμοποιούν ευαίσθητα προσωπικά δεδομένα τα οποία δεν θα πρέπει να φαίνονται στις αναλύσεις και τις αναφορές που δημιουργούνται από τους επιστήμονες δεδομένων ή τους αναλυτές δεδομένων. Η Scrubadub δίνει την δυνατότητα στους προγραμματιστές που την επιλέγουν να μπορούν να αφαιρέσουν τα ευαίσθητα προσωπικά δεδομένα που υπάρχουν σε αυτά ακόμα και σε ελεύθερο κείμενο, διατηρώντας την ιδιωτικότητα και ασφάλεια των κατόχων τους. Τέτοιες πληροφορίες μπορεί να είναι διευθύνσεις ηλεκτρονικού ταχυδρομείου, ονοματεπώνυμα, usernames, τηλεφωνικοί αριθμοί, κωδικοί χρηστών και ΑΦΜ. Επί της ουσίας είναι δεδομένα τα οποία δεν χρειάζονται στην ανάλυση των δεδομένων οπότε θα πρέπει να περαστούν από διαδικασία καθαρισμού.

Η δομή λειτουργίας του Scrubadub έχει ως εξής:

- Filth («βρώμικα») αντικείμενα χρησιμοποιούνται για να αναγνωρίσουν «βρώμικα» δεδομένα τα οποία περιέχουν ευαίσθητες πληροφορίες.
- Detector («ανιχνευτικά») αντικείμενα χρησιμοποιούνται για να ανιχνεύσουν συγκεκριμένους τύπους filth
- PostProcessor («Μετεπεξεργασμένα») αντικείμενα χρησιμοποιούνται για να τροποποιήσουν τα ευρήματα filth. Αυτό σημαίνει ότι αντικαθιστούν το filth με δίεση ή κάποιο σύμβολο.
- Ο Scrubber («καθαριστής») είναι υπεύθυνος για την διαχείριση της διαδικασίας καθαρισμού. Καταγράφει τις ενέργειες των υπόλοιπων αντικειμένων και λύνει τυχών συγκρούσεις μεταξύ διαφορετικών αντικειμένων Detector.

Επιπλέον υπάρχουν αρκετές συναρτήσεις Detector που κάνουν διαφορετικές δουλειές όταν χρησιμοποιούνται, δηλαδή βρίσκουν διαφορετικά προσωπικά δεδομένα σε ένα κείμενο. Αυτές οι συναρτήσεις είναι: Credential Detector, Credit Card Detector, Drivers License Detector, Email Detector, National Insurance Number Detector, Phone Detector, Postal Code Detector, Social Security Number Detector, Tax Reference Detector, Twitter Detector, URL Detector, Vehicle License Plate Detector τα οποία αποτελούν τους προεπιλεγμένους ανιχνευτές και οι Date of Birth Detector, Skype Detector, Tagged Evaluation Filth Detector, Text Blob Name Detector, User Supplied Filth Detector, Address Detector, Spacy Name Detector, Spacy Entity Detector και Stanford Entity Detector οι οποίοι πρέπει εισαχθούν χειροκίνητα μετέπειτα για να λειτουργήσουν. Η συνάρτηση clean είναι εκείνη που πραγματοποιεί όλα αυτά που έχουν προαναφερθεί μέσω των αντικειμένων που «στήνει» ένας προγραμματιστής.

5.5 Arrow

(Arrow, n.d.)

Η βιβλιοθήκη της Python Arrow έχει δημιουργηθεί με στόχο την εύκολη προσέγγιση των προγραμματιστών για την δημιουργία, τον χειρισμό και την μορφοποίηση των πεδίων που περιέχουν ημερομηνίες, ώρες ή και χρονική σφραγίδα. Με την χρήση της δίνεται η δυνατότητα εφαρμογής και ενημέρωσης του τύπου μιας ημερομηνίας, καλύπτοντας κενά στην λειτουργικότητα ενώ παρέχει και ένα έξυπνο API που μπορεί να υποστηρίξει αρκετά

συνηθισμένα σενάρια δημιουργίας. Συνοπτικά βοηθά στην εργασία που αφορά τις ημερομηνίες και τις ώρες ενός σετ δεδομένων με λίγα imports και ακόμα λιγότερο κώδικα.

Κάνοντας import την arrow υπάρχουν κάποιες αξιόλογες συναρτήσεις που μπορούν να χρησιμοποιηθούν κατά την διαδικασία καθαρισμού και μορφοποίησης ενός σετ δεδομένων. Μια από αυτές είναι η get(), η οποία παίρνει ως παράμετρο μια μεταβλητή που περιέχει κάποιο int ή float timestamp και το μετατρέπει σε κανονική ημερομηνία, ενώ με την προσθήκη δεύτερης παραμέτρου δίνεται και η μορφή της ημερομηνίας (όπως π.χ. 'YYYY-MM-DD HH:mm:ss'). Ακόμα μια χρήσιμη συνάρτηση είναι η .to() σε μια μεταβλητή που περιέχει ημερομηνία και ώρα, καθώς μέσω της παραμέτρου μπορεί να αλλαχθεί η ζώνη ώρας της ημερομηνίας που περιέχεται. Η arrow είναι μια βιβλιοθήκη με αρκετά εξειδικευμένο σκοπό και χρήση στον καθαρισμό δεδομένων, αφού κατά κύριο λόγο οι λειτουργίες τις αφορούν κυρίως data formatting.

5.6 dataprep

(Dataprep, n.d.)

Το DataPrep είναι μια βιβλιοθήκη ανοιχτού κώδικα διαθέσιμη για python που επιτρέπει την προετοιμασία των δεδομένων χρησιμοποιώντας μια ενιαία βιβλιοθήκη με λίγες μόνο γραμμές κώδικα. Το DataPrep μπορεί να χρησιμοποιηθεί για την αντιμετώπιση πολλαπλών προβλημάτων που σχετίζονται με δεδομένα και η βιβλιοθήκη παρέχει πολυάριθμες δυνατότητες μέσω των οποίων κάθε πρόβλημα μπορεί να λυθεί και να επιλυθεί. Χρησιμοποιώντας τη Βιβλιοθήκη DataPrep, μπορεί κανείς να συλλέξει δεδομένα από πολλαπλές πηγές δεδομένων χρησιμοποιώντας τη λειτουργική μονάδα dataprep.connector, να εκτελέσει έντονη διερευνητική ανάλυση χρησιμοποιώντας τη μονάδα dataprep.eda και να καθαρίσει και να τυποποιήσει σύνολα δεδομένων χρησιμοποιώντας τη μονάδα dataprep.clean. Το DataPrep εντοπίζει αυτόματα και επισημαίνει τις πληροφορίες που υπάρχουν στα δεδομένα, όπως δεδομένα που λείπουν, διακριτό πλήθος και στατιστικά στοιχεία.

Exploratory Data Analysis (EDA) είναι η διαδικασία εξερεύνησης ενός συνόλου δεδομένων και κατανόησης των κύριων χαρακτηριστικών του. Το πακέτο dataprep.eda απλοποιεί αυτή τη διαδικασία επιτρέποντας στον χρήστη να εξερευνήσει σημαντικά

χαρακτηριστικά με απλά API. Κάθε API επιτρέπει στο χρήστη να αναλύσει το σύνολο δεδομένων από υψηλό επίπεδο σε χαμηλό επίπεδο και από διαφορετικές οπτικές γωνίες.

Το DataPrep.Clean περιέχει απλές και αποτελεσματικές λειτουργίες για τον καθαρισμό, την τυποποίηση και την επικύρωση δεδομένων σε ένα DataFrame. Οι λειτουργίες χρησιμοποιούν μια ενοποιημένη διεπαφή για την εκτέλεση κοινών λειτουργιών καθαρισμού δεδομένων που απαιτούνται για διάφορους τύπους δεδομένων.

5.7 Seaborn

(Seaborn, n.d.)

Η βιβλιοθήκη seaborn έχει γραφτεί κάτω από την Matplotlib με ενσωματωμένα τις δομές δεδομένων της pandas, με σκοπό την δημιουργία επιπλέον στατιστικών γραφημάτων από του Matplotlib. Η seaborn βοηθά προγραμματιστές και επιστήμονες δεδομένων να κατανοήσουν σε μεγαλύτερο βαθμό τα δεδομένα που θέλουν να καθαρίσουν και επεξεργαστούν. Οι σχεδιαστικές συναρτήσεις που παρέχει λειτουργούν σε dataframes και σε πίνακες που περιέχουν ολόκληρα σετ δεδομένων και εκτελούν απαραίτητη χαρτογράφηση και στατιστική συγκέντρωση για την παραγωγή ενημερωτικών γραφημάτων.

Μέσω της seaborn μπορούν να πραγματοποιηθούν στατιστικές εκτιμήσεις χρησιμοποιώντας τον μέσο όρο μίας μεταβλητής σε συνάρτηση με άλλες μεταβλητές όπως είναι η relplot() και μάλιστα μαζί με να υπολογίσει την πιθανότητα λάθους της εκτίμησης που έχει σχεδιάσει όπως πραγματοποιεί η συνάρτηση lmplot(). Επιπλέον, η seaborn μπορεί να δημιουργήσει πληροφοριακές περιλήψεις κατανομών των δεδομένων μέσω ιστογραμμάτων που δημιουργεί η συνάρτηση displot(). Ακόμα, διαθέτει ξεχωριστή κατηγορία που αφορά στην κατανομή των κατηγορικών δεδομένων με μεγάλη ποικιλία παρουσίασης των αναλύσεων από την συνάρτηση catplot(). Η seaborn διαθέτει ακόμα την συνάρτηση jointplot() η οποία συνθέτει γραφήματα σύγκρισης μεταξύ γνωρισμάτων ενός σετ δεδομένων. Γενικά, η seaborn αποτελεί ένα χρήσιμο εργαλείο οπτικοποίησης των δεδομένων με σκοπό την αξιολόγηση αυτών πριν την όποια επεξεργασία γίνει με αυτά.

5.8 Fuzzywuzzy

(fuzzywuzzy-python-library, n.d.)

Σε ένα σετ από δεδομένα είναι πολύ πιθανό να βρεθούν διπλότυπες εγγραφές. Στην προσπάθεια να βρούμε αυτές τις εγγραφές ,η βιβλιοθήκη της Python Fuzzywuzzy μας διευκολύνει με τις λειτουργίες της για τον σκοπό αυτό. Δεν αυτοματοποιεί κάποια διαδικασία καθαρισμού αυτή καθ' αυτή, αλλά συμβάλλει σημαντικά σε ένα μέρος του.. Η βιβλιοθήκη αυτή βαθμολογεί το ταίριασμα δυο string πεδίων με ανώτατο βαθμό 100. Αυτό μπορεί να εφαρμοστεί βέβαια και σε στήλες ενός dataframe. Η βιβλιοθήκη χρησιμοποιεί την τεχνική μέτρησης της απόστασης Levenshtein, η οποία βασίζεται στον αριθμό των μετατροπών (εισαγωγή, διαγραφή ή αντικατάσταση) για να μετατραπεί η μία περίπτωση στην άλλη. Οι λειτουργίες που μας παρέχει είναι σημαντικές όταν ψάχνουμε να βρούμε πληροφορίες που υπάρχουν δυο φορές ή και παραπάνω φορές σε ένα σετ δεδομένων, γεγονός που δημιουργεί λανθασμένα αποτελέσματα σε μελλοντικές αναλύσεις.

5.9 PyOD

(pyod, n.d.)

Το PyOD είναι μια ευέλικτη και επεκτάσιμη εργαλειοθήκη σχεδιασμένη για τον εντοπισμό ακραίων τιμών ή ανωμαλιών σε δεδομένα πολλαπλών μεταβλητών, εξ ου και το όνομα Pyod (Python Outlier Detection). Το PyOD είναι μια βιβλιοθήκη Python ανοιχτού κώδικα, που παρέχει μέχρι σήμερα πάνω από 20 αλγόριθμους ανίχνευσης ακραίων σημείων που κυμαίνονται από παραδοσιακές τεχνικές όπως ο τοπικός παράγοντας ακραίων στοιχείων έως καινοτόμες αρχιτεκτονικές νευρωνικών δικτύων, όπως μοντέλα αντιπάλου ή αυτόματη κωδικοποίηση. Είναι συμβατό με Python 2 και Python 3 σε λειτουργικά συστήματα Linux, MacOS και Windows. Η συμβατότητα επιτυγχάνεται χρησιμοποιώντας την βιβλιοθήκη six.

Το PyOD μπορεί να δώσει σωρευτικά αποτελέσματα συνδυάζοντας διάφορες μεθόδους ανίχνευσης ακραίων τιμών, ανιχνευτές και σύνολα. Περιλαμβάνει ένα εύχρηστο API και διαδραστικά παραδείγματα για τους υποστηριζόμενους αλγόριθμους. Τεχνικές βελτιστοποίησης, όπως η παραλληλοποίηση και η συλλογή Just-In-Time (JIT) μπορούν να χρησιμοποιηθούν για επιλεγμένα μοντέλα όποτε απαιτείται. Πρακτικές όπως δοκιμή

μονάδας, κάλυψη κωδικού, συνεχής ενοποίηση και έλεγχοι διατήρησης κώδικα λαμβάνονται υπόψη από τα υποστηριζόμενα μοντέλα.

5.10 Statsmodels

(statsmodels, n.d.)

Το Statsmodels είναι μια βιβλιοθήκη για στατιστική και οικονομετρική ανάλυση στην Python. Παρέχει κλάσεις και συναρτήσεις για την εκτίμηση πολλών διαφορετικών στατιστικών μοντέλων, καθώς και για τη διεξαγωγή στατιστικών δοκιμών και την εξερεύνηση στατιστικών δεδομένων. Επί του παρόντος, υπάρχουν πέντε ενότητες στην κύρια βάση κώδικα που περιέχουν στατιστικά μοντέλα, η regression (μοντέλα παλινδρόμησης ελαχίστων τετραγώνων), glm (γενικευμένα γραμμικά μοντέλα), rlm (σταθερά γραμμικά μοντέλα), discretemod (μοντέλα διακριτής επιλογής) και contrast (ανάλυση αντίθεσης). Η παλινδρόμηση περιέχει γενικευμένα ελάχιστα τετράγωνα (GLS), σταθμισμένα ελάχιστα τετράγωνα (WLS) και συνηθισμένα ελάχιστα τετράγωνα (OLS). Το Glm περιέχει γενικευμένα γραμμικά μοντέλα με υποστήριξη έξι κοινών εκθετικών οικογενειακών κατανομών και τουλάχιστον δέκα τυπικών λειτουργιών συνδέσμου. Το statsmodels περιλαμβάνει μια σειρά από classes και συναρτήσεις που βοηθούν σε εργασίες που σχετίζονται με τη στατιστική ανάλυση. Αυτές περιλαμβάνουν λειτουργίες για εύχρηστη προβολή περιγραφικών στατιστικών, μια class για τη δημιουργία πινάκων ποιότητας δημοσίευσης και λειτουργίες για τη μετάφραση «ξένων» συνόλων δεδομένων.

5.11 outlier-utils

(outlier-utils, n.d.)

Η Βιβλιοθήκη outlier-utils εξυπηρετεί τον εντοπισμό και την αφαίρεση ακραίων στοιχείων από κανονικά κατανεμημένα σύνολα δεδομένων χρησιμοποιώντας τη δοκιμή Smirnov-Grubbs. Υποστηρίζεται τόσο η έκδοση δύο όψεων(two-sided) όσο και η μονόπλευρη (one-sided version) έκδοση της δοκιμής. Το πρώτο επιτρέπει την εξαγωγή ακραίων τιμών και από τα δύο άκρα του συνόλου δεδομένων, ενώ το δεύτερο λαμβάνει υπόψη μόνο τα ελάχιστα/μέγιστα ακραία σημεία. Κατά την εκτέλεση μιας δοκιμής, κάθε ακραίο στοιχείο θα

αφαιρείται έως ότου δεν μπορεί να βρεθεί κανένα στο σύνολο δεδομένων. Η έξοδος της δοκιμής είναι αρκετά ευέλικτη ώστε να ταιριάζει με πολλές περιπτώσεις χρήσης. Από προεπιλογή, τα δεδομένα χωρίς ακραίες τιμές θα επιστραφούν, αλλά η δοκιμή μπορεί επίσης να επιστρέψει τα ίδια τα ακραία στοιχεία ή τους δείκτες τους στο αρχικό σύνολο δεδομένων. Τέλος χρησιμοποιείται για μια στήλη κάθε φορά, και δεν διαθέτει συνάρτηση ώστε να μπορεί να δεχθεί ολόκληρο dataframe

6. **Μελέτη περίπτωσης**

Σε αυτό το μέρος θα προσπαθήσουμε να εφαρμόσουμε όσο το δυνατόν καλύτερα τα modules που έχουν προαναφερθεί ανάλογα με τα χαρακτηριστικά τους, έτσι ώστε να καθαρίσουμε ένα σετ δεδομένων. Έχουμε ένα σετ δεδομένων το οποίο αφορά πληροφορίες που έχουν αντληθεί από τις Ηνωμένες Πολιτείες της Αμερικής σχετικά με τους θανάτους του κορονοϊού ανά κάποιες περιοχές αυτών. Το σετ δεδομένων έχει τις εξής στήλες Admin 2 FIPS Code, Province/State, Admin 2 Level (City/County/Borough/Region), Date, Total Death, Total Confirmed, location. Η διαδικασία το καθαρισμού του συγκεκριμένου αρχείου θα γίνει με αρκετά αναλυτικό τρόπο, καθότι έχουμε ορίσει τις βασικές εργασίες που πρέπει να εφαρμοστούν πάνω σε ένα αρχείο. Έτσι αφότου θα έχουν διεκπεραιωθεί όλες θα έχουμε ένα αρχείο με υψηλής ποιότητας πληροφορία. Για τις ανάγκες των εργασιών θα χρειαστεί να κάνουμε υποθέσεις απαιτήσεων, έτσι ώστε να προβληθούν αρκετές τεχνικές καθαρισμού δεδομένων. Η χρήση της βιβλιοθήκης Pandas είναι αναγκαία σε όλα τα κομμάτια του κώδικα μας, αφού είναι η μόνη βιβλιοθήκη της Python που διαχειρίζεται τα δεδομένα ενός αρχείου σαν πίνακα δεδομένων με στήλες (column) και αριθμό γραμμής (index), ο οποίος ονομάζεται dataframe. Η σειρά βημάτων που έχει οριστεί ως η κατάλληλη για τον βέλτιστο καθαρισμό είναι η εξής:

1. *Data Overview (Προεπισκόπηση Δεδομένων):* Η προβολή των τύπων των στηλών, της μορφής των δεδομένων, η απαρίθμηση των NaN τιμών και η επισκόπηση κάποιων βασικών στατιστικών στοιχείων είναι απαραίτητα για να μπορούμε να δούμε τις ενέργειες που θα πραγματοποιήσουμε μετέπειτα. Αυτό το μέρος αρκετές φορές θεωρείται προκαταρκτικό του καθαρισμού και είναι πολύ χρήσιμο, καθότι ο προγραμματιστής μπορεί μέσω αυτού να βγάλει πολύτιμα συμπεράσματα για τα δεδομένα του. Εν παραδείγματι, ενδέχεται να διαπιστώσει ότι μια στήλη δεν περιέχει καμία τιμή ή και το ότι στο σετ δεδομένων δεν έχει εφαρμοστεί σωστά το delimiter που έχει οριστεί, κατά την εισαγωγή των δεδομένων, ως dataframe.
2. *Fix data shape (Μορφοποίηση Δεδομένων):* Σε αυτό το μέρος θα γίνει η πρώτη διαδικασία που αφορά σε κομμάτι του data cleaning και είναι η μορφοποίηση των δεδομένων. Κατά την εισαγωγή των δεδομένων σε ένα dataframe, οι πληροφορίες που περιέχονται είναι αρκετά συχνό φαινόμενο να μη έχουν την επιθυμητή μορφή,

ώστε να τα επεξεργαστεί κάποιος αναλυτής. Η διαδικασία μορφοποίησης, δηλαδή, πραγματοποιεί τις απαραίτητες αλλαγές, έτσι ώστε να προχωρήσουμε την διαδικασία καθαρισμού χωρίς ο τύπος ή η μορφή μιας στήλης να αποτελεί εμπόδιο για τα επόμενα βήματα μας.

3. *Deduplicate Data (Αποδιπλασιασμός Δεδομένων):* Αφορά στην μείωση των δεδομένων με σκοπό την αύξηση της εγκυρότητάς τους. Αυτή η μείωση των πληροφοριών δεν αφορά δεδομένα που δεν είναι σχετικά με το αντικείμενο που θα μελετηθεί αργότερα, αλλά με δεδομένα που επαναλαμβάνονται στον πίνακα δεδομένων και αφορούν ένα αντικείμενο του πραγματικού κόσμου.
4. *Remove Irrelevant Data (Αφαίρεση μη σχετικών δεδομένων):* Αφορά στην απομάκρυνση δεδομένων που δεν θεωρούνται σχετικά με το αντικείμενο που θα μελετηθεί. Για παράδειγμα σε ένα σετ δεδομένων με αυτοκίνητα πρέπει σε ένα σενάριο καθαρισμού να κρατήσουμε μόνο αυτά που είναι μάρκας “Honda”, οπότε όλα τα άλλα δεδομένα που αφορούν άλλες μάρκες αυτοκινήτων θα πρέπει να διαγραφούν.
5. *Deal with missing data (Αντιμετώπιση Ελλιπών Τιμών):* Η διαχείριση περιπτώσεων με ελλιπείς τιμές στο σετ δεδομένων είναι πολύ σημαντικό κομμάτι του καθαρισμού, καθώς υπάρχουν διαφορετικές προσεγγίσεις διαχείρισής αυτών. Μπορεί να αντιμετωπιστούν με την διαγραφή των εγγραφών που περιέχουν ελλιπείς τιμές ή να αντικατασταθούν οι ελλιπείς τιμές με βάση τα υπόλοιπα στοιχεία που υπάρχουν στο σετ δεδομένων.
6. *Detection of data outliers (Ανίχνευση Έκτοπων Τιμών):* Αυτό το μέρος του καθαρισμού αφορά στα αριθμητικά δεδομένα που υπάρχουν σε ένα σετ δεδομένων. Ο εντοπισμός των έκτοπων τιμών είναι μια πολύ συγκεκριμένη διαδικασία εύρεσης εγγραφών, με βάση κάποιο αριθμητικό τους χαρακτηριστικό που έχει πολύ μεγάλη ή πολύ μικρή τιμή σε σχέση με τις υπόλοιπες που περιέχονται στα δεδομένα.
7. *Validate data (Έλεγχος Ορθότητας Δεδομένων):* Η επικύρωση δεδομένων είναι η διαδικασία διασφάλισης ότι τα δεδομένα έχουν δεχτεί καθαρισμό κι ως εκ τούτου διέπονται από αυξημένη ποιότητα δεδομένων, δηλαδή ότι είναι ταυτόχρονα σωστά και χρήσιμα.

6.1 **Υλοποίηση** Data Overview

Η υλοποίηση της προεπισκόπησης δεδομένων, αφότου έχουν εισαχθεί τα δεδομένα σε ένα dataframe με όνομα covid_df, γίνεται αρχικά με την βοήθεια της ίδιας βιβλιοθήκης Pandas έτσι ώστε να φανεί τι περιέχεται μέσα στον πίνακα.

6.1.1 Pandas,Matplotlib,Missingo

Στο πρώτο μέρος παρουσιάζεται το περιεχόμενο των δεδομένων σε συνδυασμό με τους τύπους των στηλών, έτσι όπως τους έχει αναθέσει η βιβλιοθήκη pandas.

1.1a) Προβολή dataset

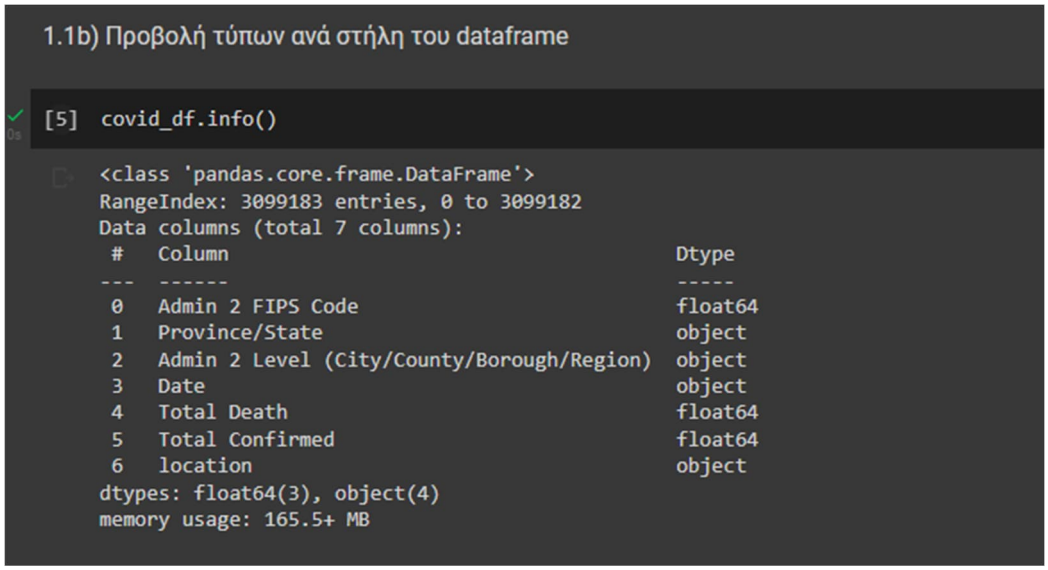
```
covid_df
```

	Admin 2 FIPS Code	Province/State	Admin 2 Level (City/County/Borough/Region)	Date	Total Death	Total Confirmed	location
0	12119.0	Florida	Sumter	2020-01-31	0.0	0.0	28.70181754,-82.0794267
1	13153.0	Georgia	Houston	2020-01-31	0.0	0.0	32.45802497,-83.66879087
2	13227.0	Georgia	Pickens	2020-01-31	0.0	0.0	34.46589159,-84.46406611
3	13303.0	Georgia	Washington	2020-01-31	0.0	0.0	32.96711864,-82.79357039
4	16003.0	Idaho	Adams	2020-01-31	0.0	0.0	44.89333571,-116.4545247
...	...	...	...	...	...	...	...
3099178	5083.0	Arkansas	Logan	2022-03-10	74.0	5076.0	35.21413234,-93.71951016
3099179	6009.0	California	Calaveras	2022-03-10	117.0	7595.0	38.20537103,-120.552913
3099180	6035.0	California	Lassen	2022-03-10	60.0	9813.0	40.67311306,-120.59350990000002
3099181	6037.0	California	Los Angeles	2022-03-10	31178.0	2808409.0	34.30828379,-118.2282411
3099182	12005.0	Florida	Bay	2022-03-10	778.0	46333.0	30.26548745,-85.62122584

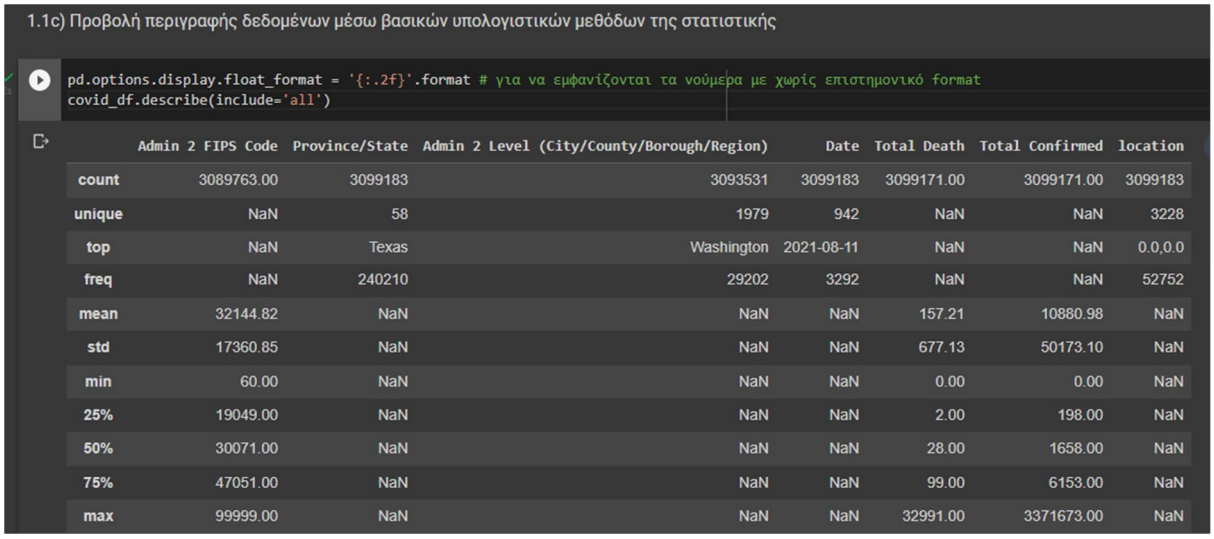
3099183 rows × 7 columns

Εικόνα 11: *Προεπισκόπηση δεδομένων του dataframe μας*

Με την εντολή covid_df παρουσιάζεται ένα μικρό δείγμα του dataset από τις πρώτες και τελευταίες στήλες. Με την info() εμφανίζεται ένας πίνακας με τα ονόματα των στηλών σε συνδυασμό με τον τύπο που έχουν τα δεδομένα μέσα σε αυτό. Επίσης, δίνει πληροφορίες για το σύνολο των γραμμών (εγγραφών) που έχει το αρχείο μας καθώς και για το μέγεθος του αρχείου.



Εικόνα 12: *εφαρμογή συνάρτησης info()*



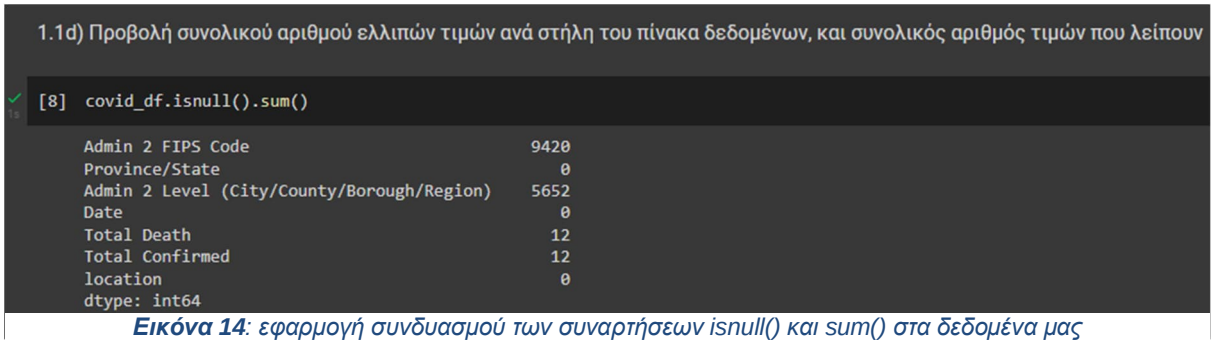
	Admin 2 FIPS Code	Province/State	Admin 2 Level (City/County/Borough/Region)	Date	Total Death	Total Confirmed	location
count	3089763.00	3099183	3093531	3099183	3099171.00	3099171.00	3099183
unique	NaN	58	1979	942	NaN	NaN	3228
top	NaN	Texas	Washington	2021-08-11	NaN	NaN	0.0,0.0
freq	NaN	240210	29202	3292	NaN	NaN	52752
mean	32144.82	NaN	NaN	NaN	157.21	10880.98	NaN
std	17360.85	NaN	NaN	NaN	677.13	50173.10	NaN
min	60.00	NaN	NaN	NaN	0.00	0.00	NaN
25%	19049.00	NaN	NaN	NaN	2.00	198.00	NaN
50%	30071.00	NaN	NaN	NaN	28.00	1658.00	NaN
75%	47051.00	NaN	NaN	NaN	99.00	6153.00	NaN
max	99999.00	NaN	NaN	NaN	32991.00	3371673.00	NaN

Εικόνα 13: *αλλαγή format προβολής float τιμών και εφαρμογή συνάρτησης describe() με τα αποτελέσματα της δεύτερης*

Επιπλέον, η συνάρτηση describe(include='all') παρουσιάζει κάποια βασικά στατιστικά στοιχεία ανάλογα με τον τύπο των δεδομένων, που μπορούν να ληφθούν υπόψη για τις μετέπειτα διαδικασίες του καθαρισμού. Στο σετ δεδομένων, ανάλογα με τα αποτελέσματα, υπάρχουν κάποια στοιχεία που ενδέχεται να προβληματίζουν. Παρατηρείται ότι στο σετ δεδομένων που αφορά στις Η.Π.Α. οι unique πολιτείες στην στήλη “Province/State” είναι 58, ενώ είναι γνωστό ότι όλες οι πολιτείες είναι 50. Άρα, αυτό μπορεί να κρατηθεί ως συμβάν υπό εξέτασή για μετέπειτα. Ακόμα, στα αποτελέσματα που παρουσιάζονται διαπιστώνεται ότι το count είναι διαφορετικό ανά στήλη, κάτι που σημαίνει ότι υπάρχουν ελλιπείς τιμές σε κάποιες στήλες. Επιπλέον, στις στήλες των θανάτων “Total Death” και “Total confirmed” βρέθηκε κατώτατη τιμή 0, οπότε είναι πολύ πιθανό αυτές οι εγγραφές να μην χρειάζονται, αφού δεν μας δίνουν κάποια επιπλέον πληροφορία.

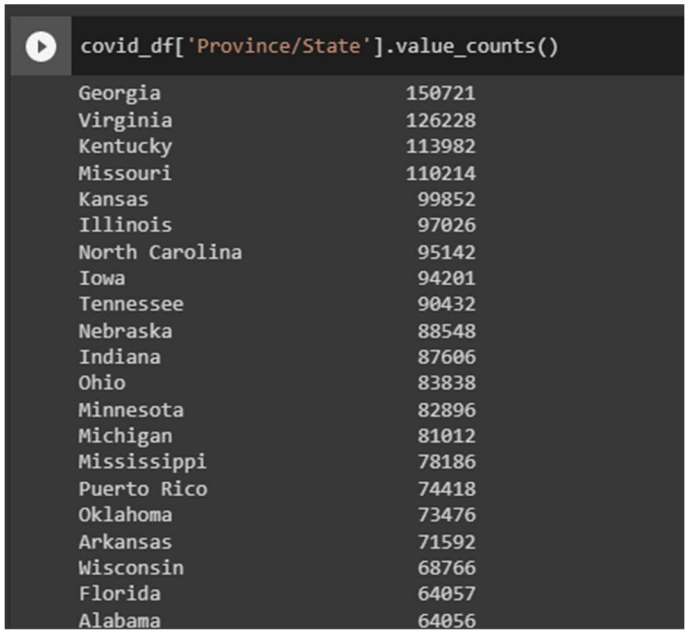
Για να ελεγχθεί το γεγονός που προαναφέρθηκε με τις ελλιπείς τιμές, η βιβλιοθήκη pandas περιέχει την συνάρτηση isnull() που σε συνδυασμό με την .sum() παρουσιάζουν τον αριθμό των NaN για κάθε στήλη.

Όπως προαναφέρθηκε είναι σημαντική η γνώση των τιμών μιας στήλης, για να μπορούν να



***Εικόνα 14**: εφαρμογή συνδυασμού των συναρτήσεων isnull() και sum() στα δεδομένα μας*

αξιολογηθούν, όπως στην περίπτωση που βρέθηκε ότι το dataset περιέχει παραπάνω αριθμό πολιτειών. Αυτό μπορεί να ελεγχθεί μέσω της συνάρτησης value_counts() της pandas που δείχνει τον αριθμό με τον οποίο επαναλαμβάνεται κάθε μια τιμή σε μια στήλη του dataframe.

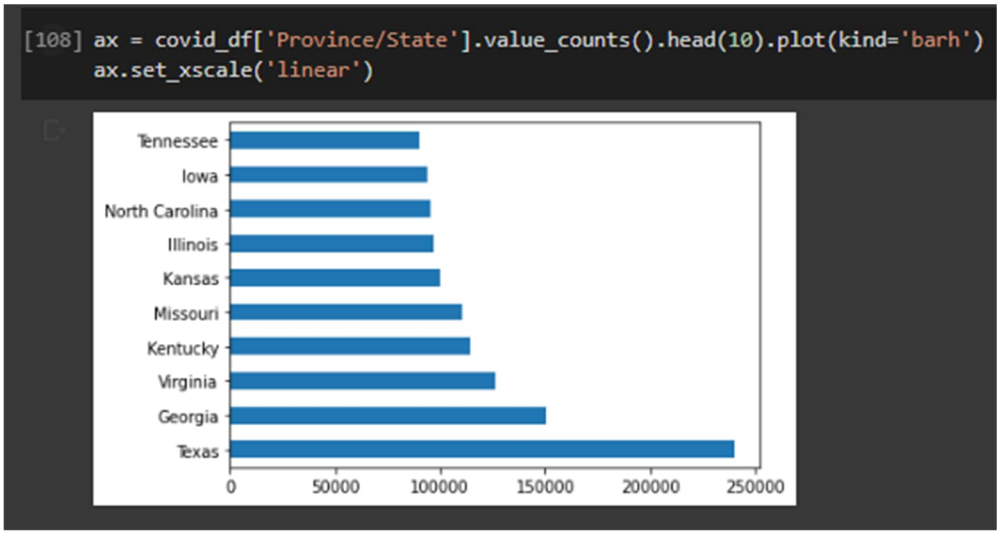


***Εικόνα 16**: εφαρμογή συνάρτησης value_counts() και το πρώτος μέρος των αποτελεσμάτων*



***Εικόνα 15:** Δεύτερο μέρος αποτελεσμάτων της συνάρτησης.*

Τρέχοντας την συνάρτηση για την στήλη παρουσιάζεται ότι το σετ δεδομένων δεν έχει κάποιο αλφαριθμητικό λάθος παρότι περιέχει περισσότερες πολιτείες. Το γεγονός ότι περιέχει περισσότερες πολιτείες από τις υπάρχουσες εξηγείται διότι περιέχει και ορισμένες περιοχές οι οποίες δεν είναι στις επίσημες πολιτείες, αλλά ανήκουν στις Η.Π.Α, όπως είναι η



Εικόνα 17: *Σχεδιάγραμμα της Pandas για την καταμέτρηση των 10 συχνότερων πολιτειών*

American Samoa. Επιπλέον, με την βοήθεια της plot() οπτικοποιούνται, μέσω σχεδιαγράμματος, τα αποτελέσματα αυτά, εμφανίζοντας τις 10 πολιτείες με τις περισσότερες πλειάδες.

Μια ακόμα χρήσιμη εντολή που τρέξαμε αφορά στην προβολή των duplicates εγγραφών στον πίνακα των δεδομένων μας. Με την χρήση της συνάρτηση duplicated(), η βιβλιοθήκη pandas δημιουργεί μια νέα στήλη στο πίνακα, τύπου Boolean, και αναθέτει τιμή True σε όποια εγγραφή έχει ακριβώς την ίδια πληροφορία σε κάθε στήλη άλλη μια φορά τουλάχιστον, ενώ αναθέτει False σε αυτές που είναι unique. Στα αποτελέσματα παρατηρείται ότι υπάρχουν τρεις διαφορετικές εγγραφές που είναι διπλότυπες. Αυτό το πρόβλημα θα αντιμετωπιστεί με τον καθαρισμό των δεδομένων μετέπειτα.

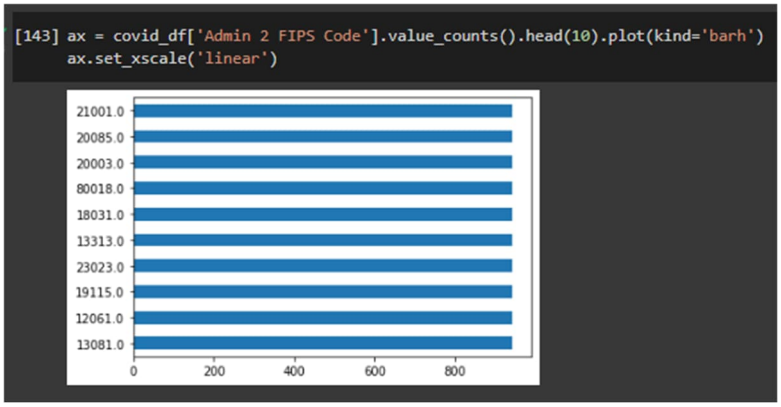


	Admin 2 FIPS Code	Province/State	Admin 2 Level (City/County/Borough/Region)	Date	Total Death	Total Confirmed	location	Duplicate
13	19115.00	Iowa	Louisa	2020-01-31	0.00	0.00	41.21842119,-91.26102611	True
2461	12061.00	Florida	Indian River	2021-08-11	306.00	15477.00	27.69308961,-80.60556721	True
2780	13081.00	Georgia	Crisp	2021-08-11	94.00	2270.00	31.92289565,-83.76811843	True

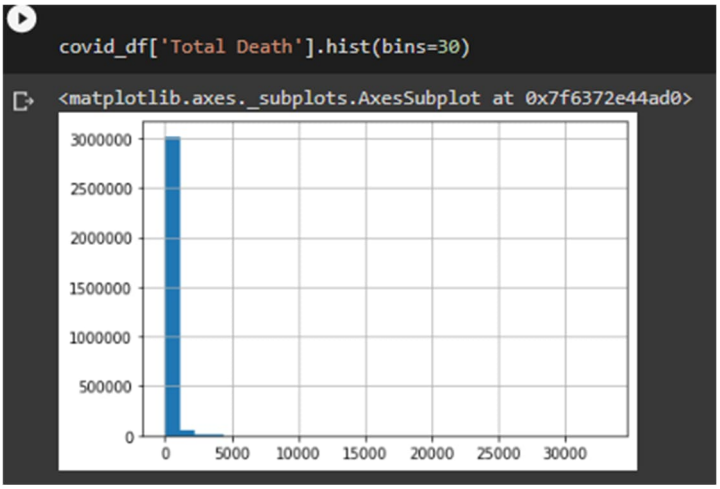
Εικόνα 18*: Εφαρμογή της συνάρτησης duplicated για την εύρεση διπλότυπων και προβολή αυτών*

Στην συνέχεια, με την βοήθεια των βιβλιοθηκών missingno,matplotlib και pandas έχουν εξαχθεί αρκετά χρήσιμα διαγράμματα για τα περιεχόμενα των στηλών. Για τις στήλες που αφορούν στην τοποθεσία προέλευσης κάθε εγγραφής έχουν χρησιμοποιηθεί σχεδιαγράμματα που δείχνουν τις 10 πρώτες τιμές σε συχνότητα ενώ για τις αριθμητικές έχουν χρησιμοποιηθεί ιστογράμματα με «κάδους» που χωρίζουν τις τιμές σε σύνολα και μετράνε την συχνότητα στο κάθε ένα σύνολο. Επιπλέον, με την missingno και την συνάρτηση

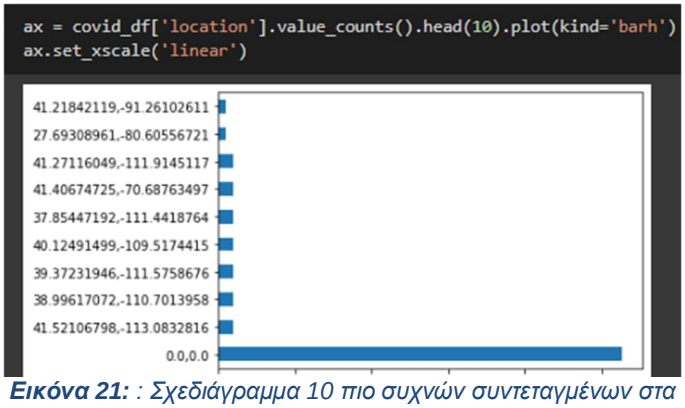
της matrix() γίνεται οπτικά αντιληπτό πόσο πλήρη είναι τα δεδομένα μας ανά στήλη . Τέλος, με την χρήση της συνάρτησης .corr() και της plot() της βιβλιοθήκης Pandas εξάγεται ένα διάγραμμα που δείχνει την συσχέτιση που έχουν οι αριθμητικές στήλες μεταξύ τους.



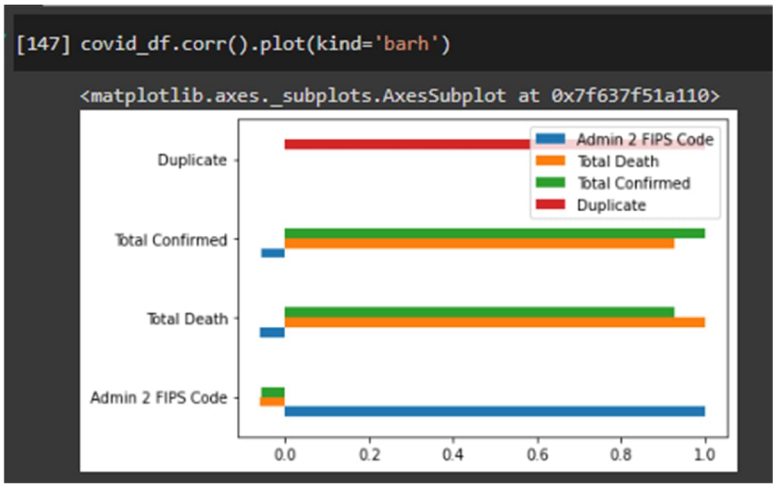
Εικόνα 20: *Σχεδιάγραμμα 10 πιο συχνών FIPS Code στα δεδομένα μας*



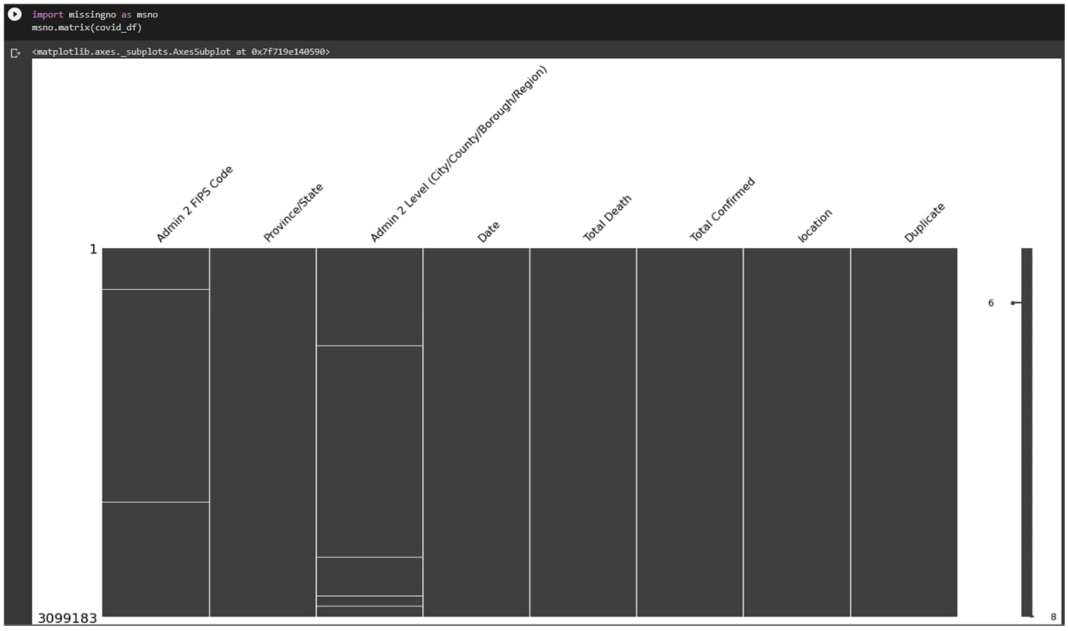
Εικόνα 19: *: Ιστόγραμμα που παρουσιάζει την κατανομή των αριθμών 'Total Death'*



Εικόνα 21: *: Σχεδιάγραμμα 10 πιο συχνών συντεταγμένων στα δεδομένα μας*



Εικόνα 23: *Εφαρμογή correlation σε συνδυασμό με την οπτικοποίηση των αποτελεσμάτων του σε σχεδιάγραμμα*

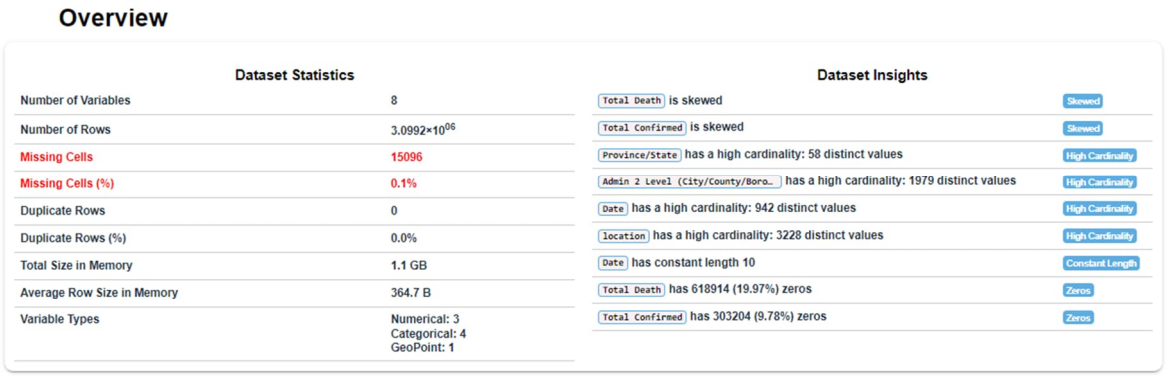


Εικόνα 22: *Εφαρμογή συνάρτηση matrix και αποτέλεσμα αυτής*

6.1.2 dataprep.eda

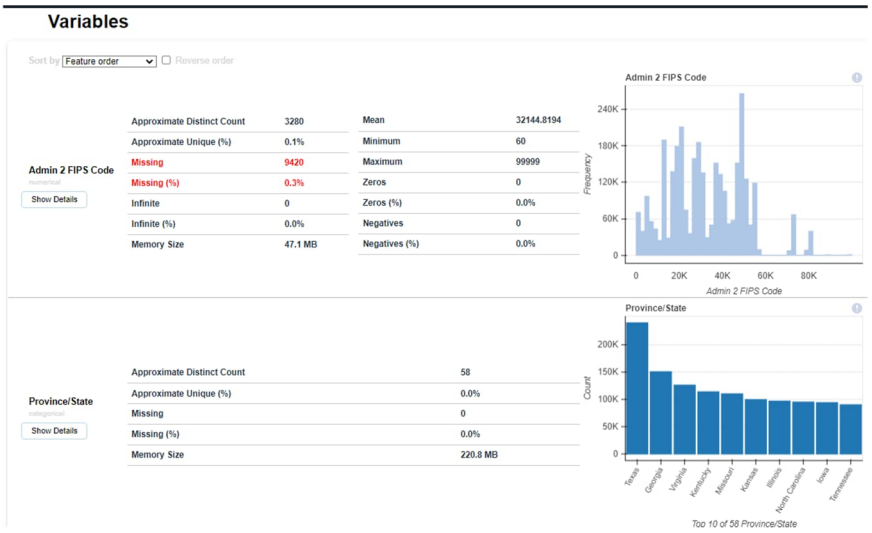
Με την βοήθεια της βιβλιοθήκης dataprep.eda δίνεται η δυνατότητα να γίνει η ανασκόπηση των δεδομένων με την συνάρτηση create_report(). Η συνάρτηση αυτή παρουσιάζει ως αποτελέσματα 5 ενότητες που περιέχουν κάποια από τα πιο κύρια στατιστικά των δεδομένων, σε αριθμητική ή και διαγραμματική μορφή. Στο πρώτο τμήμα,

που ονομάζεται overview, και συγκεκριμένα στους πίνακες Dataset Statistics και Dataset Insights εμφανίζονται τα απαραίτητα αριθμητικά δεδομένα που αφορούν στην ποιότητα του



Εικόνα 24: *Overview πίνακας αποτελεσμάτων συνάρτησης create_report, τμήμα 1*

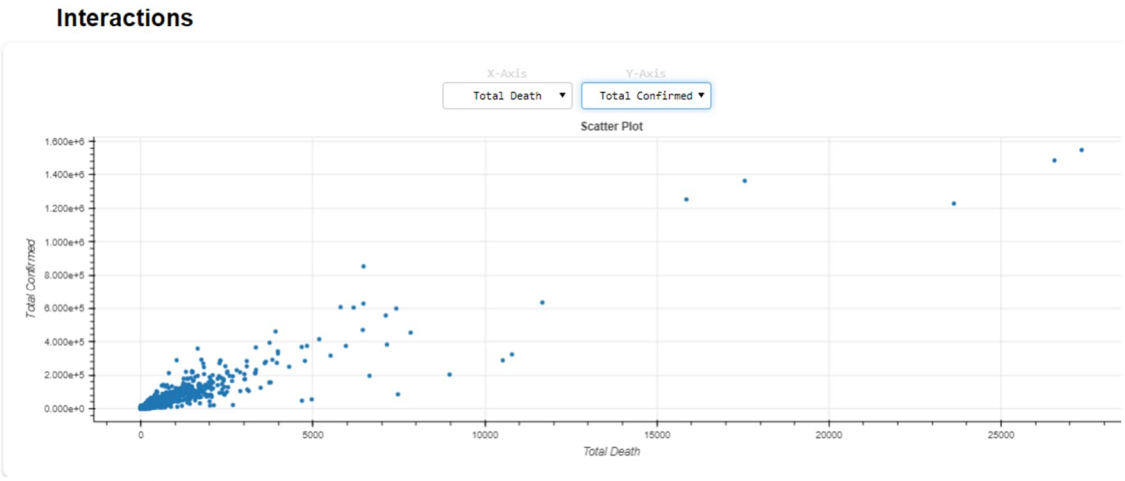
σετ δεδομένων και ταυτοχρόνως παρουσιάζονται ορισμένα χαρακτηριστικά της κάθε στήλης δεδομένων, όπως είναι το γεγονός ότι η μορφή του date πεδίου έχει σταθερό μήκος 10 χαρακτήρων. Στο δεύτερο τμήμα του report παρουσιάζονται ανά στήλη στατιστικά δεδομένα για την συχνότητα των τιμών, τα NaN και ο αριθμός των μοναδικών τιμών, ενώ εμφανίζονται



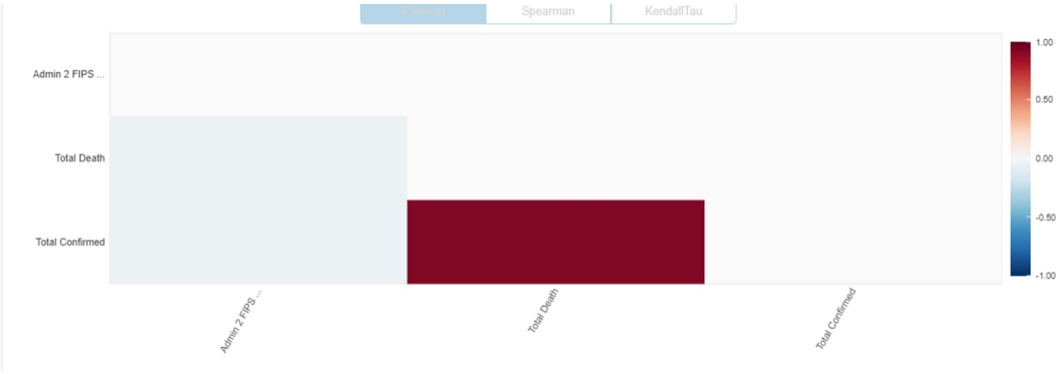
Εικόνα 25*:Variables πίνακας αποτελέσματα και γραφήματα συνάρτησης create_report, τμήμα 2*

σε διάγραμμα ο καταμερισμός των αριθμών ανά σύνολο τιμών ή οι 10 πιο συχνές τιμές των μη αριθμητικών δεδομένων. Στο τρίτο τμήμα του report, εμφανίζεται ένα γράφημα παλινδρόμησης μεταξύ των αριθμητικών δεδομένων του dataframe, με σκοπό την προβολή εξάρτησης μια στήλης x από μια y (μας δίνει την δυνατότητα να δούμε όλες τις σχέσεις μεταξύ των στηλών σε γράφημα παλινδρόμησης χωρίς να χρειαστεί δεύτερη εκτέλεση. Στο τέταρτο τμήμα, εμφανίζεται ένα διαγραμματικό correlation analysis μεταξύ των αριθμητικών

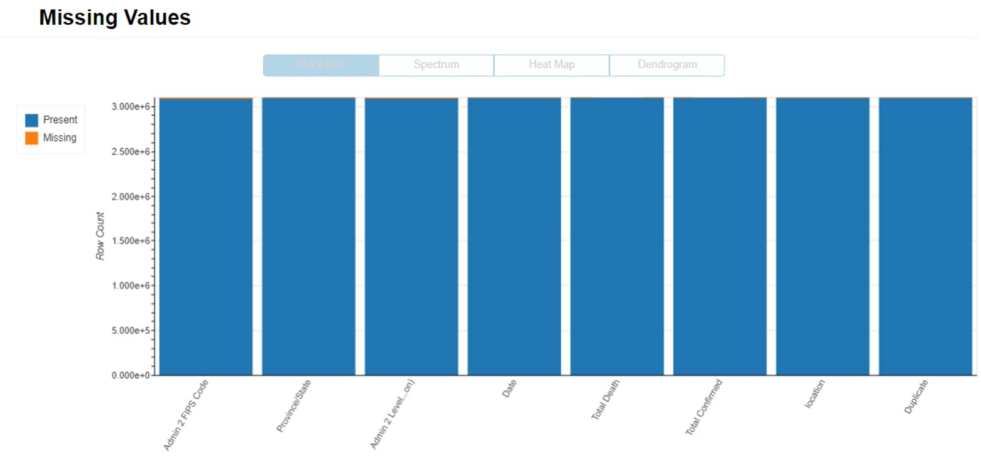
στηλών. Στο τελευταίο τμήμα, παρουσιάζεται ένα bar chart που δείχνει πόσες είναι οι υπαρκτές τιμές (μπλε) και πόσες είναι ελλιπείς (πορτοκαλί χρώμα).



***Εικόνα 26**: Interactions γράφημα συνάρτησης create_report, τμήμα 3*



***Εικόνα 27**: Correlations γράφημα συνάρτησης create_report, τμήμα 4*



***Εικόνα 28**::Variables πίνακας αποτελέσματα και γραφήματα συνάρτησης create_report, τμήμα 2*

6.1.3 **Σύγκριση**

Η βιβλιοθήκη Pandas προσφέρει πολλές ξεχωριστές συναρτήσεις για την προβολή και εξερεύνηση δεδομένων ενός dataframe, με αποτέλεσμα να είναι πολύ εύκολη η κατανόηση και η χρήση αυτών. Οι πολλές διαφορετικές συναρτήσεις την κάνουν ευέλικτη ως προς την εστίαση σε ένα συγκεκριμένο πρόβλημα που αντιμετωπίζουν τα δεδομένα. Η οπτικοποίηση των δεδομένων μέσω γραφημάτων είναι περιορισμένη, οπότε θα πρέπει να χρησιμοποιηθούν δυο ακόμα βιβλιοθήκες, η Matplotlib και η missingno, που επιλύουν το πρόβλημα αυτό με τις συναρτήσεις που περιέχουν για την γραφική απεικόνιση των δεδομένων.

Η βιβλιοθήκη dataprep.dataset είναι μια βιβλιοθήκη που αυτοματοποιεί τις διαδικασίες χωρίς μεγάλη ποικιλία συναρτήσεων. Απαντάει ,όμως, πολύ εύστοχα στα προβλήματα που προσπαθεί να επιλύσει και παρέχει στον χρήστη ίσως και παραπάνω από τα απαραίτητα στοιχεία για να ξεκινήσει την διαδικασία του καθαρισμού των δεδομένων. Δημιουργεί πίνακες με στατιστικά δεδομένα ανασκόπησης αλλά και σχεδιαγράμματα με αυτά.

Η επισκόπηση των δεδομένων γίνεται με πολύ περισσότερες γραμμές κώδικα, αλλά και πολύ πιο γρήγορα με την βοήθεια των βιβλιοθηκών Pandas, Matplotlib και missingno. Αντιθέτως, η dataprep βιβλιοθήκη δημιουργεί με μία εντολή ένα report με αρκετούς πίνακες και γραφήματα και για αυτό κάνει σχεδόν τον τετραπλάσιο χρόνο να εκτελεστεί.

***Πίνακας 1**: Data overview, σύγκριση των modules.*

Data Overview	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*
Pandas+Matplotlib+Missingno	22	23,8 seconds
dataprep.eda	1	87,78

6.2 **Υλοποίηση** Fix data shape

Αυτό το μέρος του καθαρισμού δεδομένων αφορά στην διόρθωση της μορφής δεδομένων, έτσι ώστε να λυθεί το πρόβλημα της σταθερότητας/συνέπειας τους. Θα πρέπει να επιδιορθωθούν ο τύπος και η μορφή τους, έτσι ώστε να αποκτήσουν συνέπεια ως προς την

δομή τους όλες οι εγγραφές. Παρακάτω υλοποιούνται, μέσω Pandas και Arrow, ορισμένες μετατροπές μορφής των δεδομένων σε τρεις διαφορετικές εργασίες, πάνω στο dataframe covid_df.

6.2.1 **Μορφοποίηση ονομάτων και τύπων γνωρισμάτων**

Στο πρώτο κομμάτι των εργασιών μορφοποίησης των δεδομένων πρέπει να αλλαχθούν τα ονόματα των στηλών, έτσι ώστε να μην είναι πολύ μεγάλα και να περιέχουν την πληροφορία που χρειάζεται και τέλος για να μετατραπούν οι τύποι δεδομένων σε int, float ή string, ανάλογα με το τι περιέχουν.

6.2.1.1 Pandas

pandas

```
[222] start=time.time()
      covid_df.rename(columns = {'Admin 2 FIPS Code':'FIPS Code',
                                 'Admin 2 Level (City/County/Borough/Region)':'City_County_Borough_Region',
                                 'Province/State':'Province_State'},
                      inplace = True)
      covid_df = covid_df.convert_dtypes()
      end=time.time()
      covid_df.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 3099183 entries, 0 to 3099182
      Data columns (total 7 columns):
       #   Column                   Dtype
      ---  ------                   -----
       0   FIPS Code                Int64
       1   Province_State           string
       2   City_County_Borough_Region  string
       3   Date                     string
       4   Total Death              Int64
       5   Total Confirmed          Int64
       6   location                 string
      dtypes: Int64(3), string(4)
      memory usage: 174.4 MB

      arrow.get(end)-arrow.get(start)

      datetime.timedelta(microseconds=782387)
```

Εικόνα 29: *Μετονομασία στηλων και μετατροπή τύπων στήλης με Pandas*

Με την βοήθεια των συναρτήσεων της pandas rename(columns={}) είναι δυνατή η αλλαγή ονόματος όποιας στήλης επιθυμεί ο προγραμματιστής, ενώ με την convert.dtypes() γίνεται αυτοματοποιημένα η μετατροπή του τύπου των δεδομένων σε κάθε στήλη . Οι αλλαγές που έχουν γίνει είναι οι εξής:

- Αλλαγή ονομάτων “Admin 2 FIPS Code” σε “FIPS Code” “Admin 2 Level (City / County / Borough / Region)” σε “City_County_Borough_Region” και “Province / State” σε “ 'Province_State”

- Αυτόματη μετατροπή τύπου κάθε στήλης, για να ταιριάζει περισσότερο με τα δεδομένα κάθε στήλης.

6.2.1.2 dataprep.clean

```
[225] from dataprep.clean import clean_headers
      from dataprep.clean import clean_df
      start=time.time()
      cleaned_covid_df=clean_headers(covid_df,replace= {'Admin 2 ':'','Level':''},case='const')
      inferred_dtypes,cleaned_covid_df  = clean_df(covid_df, data_type_detection = "semantic")
      end=time.time()
      cleaned_covid_df.info()

      Column Headers Cleaning Report:
              7 values cleaned (100.0%)
      Data Type Detection Report:
              These data types are supported by DataPrep to clean: ['URL']
      Column Headers Cleaning Report:
              6 values cleaned (85.71%)
      Downcast Memory Report:
              Memory reducted from 942163060 to 904972864. New size: (96.05%)
      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 3099183 entries, 0 to 3099182
      Data columns (total 7 columns):
       #   Column                     Dtype
      ---  ------                     -----
       0   fips_code                  Int32
       1   province_state             string
       2   city_county_borough_region string
       3   date                       string
       4   total_death                Int32
       5   total_confirmed            Int32
       6   location                   string
      dtypes: Int32(3), string(4)
      memory usage: 138.9 MB

      arrow.get(end)-arrow.get(start)

      datetime.timedelta(seconds=52, microseconds=554837)
```

Εικόνα 30: *Μετονομασία στηλων και μετατροπή τύπων στήλης με dataprep*

Με την χρήση της βιβλιοθήκης dataprep και συγκεκριμένα με την συνάρτηση clean_headers(). γίνεται αυτοματοποιημένος καθαρισμός των στηλών καθώς και αντικατάσταση λέξεων από τα ονόματα αυτών. Η παράμετρος replace χρησιμοποιείται για να αντικαταστήσουμε τις λέξεις ‘Admin 2’ και ‘Level’, που υπάρχουν στα ονόματα των στηλών, ενώ με την παράμετρο case=’const’ μετατρέπονται όλα τα κενά σε _ για να διαχωρίζονται μεταξύ τους οι λέξεις. Ακόμα, η συνάρτηση clean_df() σε συνδυασμό με την χρήση της παραμέτρου data_type_detection καταφέρνουν την αυτοματοποιημένη μετατροπή του τύπου κάθε στήλης, σύμφωνα με τα δεδομένα και με σκοπό την μικρότερη κατανάλωση μνήμης.

6.2.2 **Μορφοποίηση συντεταγμένων**

Στα παρόντα δεδομένα βρίσκεται μια στήλη με όνομα ‘location’ η οποία περιέχει, σε μη εύχρηστη μορφή, το γεωγραφικό μήκος και πλάτος της πόλης που αναφέρεται η συγκεκριμένη πλειάδα.

6.2.2.1 Geopandas

```
lat_long = covid_df['location'].str.split(",", expand = True)

start=time.time()
gdf = gpd.GeoDataFrame(covid_df, geometry=gpd.points_from_xy(lat_long[1], lat_long[0]))
#covid_df = covid_df.drop(columns='location')
end=time.time()
gdf
```

	fips_code	province_state	city_county_borough_region	date	total_death	total_confirmed	location	Full_Date	Year	Month	Day	geometry
0	12119	Florida	Sumter	2020-01-31	0	0	28.70181754,-82.0794267	2020-01-31	2020	1	31	POINT (-82.07943 28.70182)
1	13153	Georgia	Houston	2020-01-31	0	0	32.45802497,-83.66879087	2020-01-31	2020	1	31	POINT (-83.66879 32.45802)
2	13227	Georgia	Pickens	2020-01-31	0	0	34.46589159,-84.46406611	2020-01-31	2020	1	31	POINT (-84.46407 34.46589)
3	13303	Georgia	Washington	2020-01-31	0	0	32.96711864,-82.79357039	2020-01-31	2020	1	31	POINT (-82.79357 32.96712)
4	16003	Idaho	Adams	2020-01-31	0	0	44.89333571,-116.4545247	2020-01-31	2020	1	31	POINT (-116.45452 44.89334)
...	...	...	...	...	...	...	...	...	...	...	...	...
3099178	5083	Arkansas	Logan	2022-03-10	74	5076	35.21413234,-93.71951016	2022-03-10	2022	3	10	POINT (-93.71951 35.21413)
3099179	6009	California	Calaveras	2022-03-10	117	7595	38.20537103,-120.552913	2022-03-10	2022	3	10	POINT (-120.55291 38.20537)

Εικόνα 31: *Δημιουργία geodataframe*

Με την παραπάνω μορφοποίηση και με την συνάρτηση .str.split() της pandas, «σπάει» η στήλη location μέσω του διαχωριστικού ‘,’ και με την βιβλιοθήκη Geopandas, μέσω συνάρτησης GeodataFrame(), αποθηκεύονται σε μία στήλη geometry οι δυο στήλες ως geometry point, όπου περιέχεται το γεωγραφικό πλάτος και μήκος. Με αυτήν την μέθοδο δημιουργείται ένα geodataframe, όπου κάθε εγγραφή έχει ένα γεωγραφικό σημείο που είναι εύκολα διαχειρίσιμο.

6.2.2.2 dataprep.clean

Η βιβλιοθήκη dataprep.clean παρέχει, για το πρόβλημα των πεδίων που έχουν συντεταγμένες μέσα τους, την συνάρτηση clean_lat_long(). Η συνάρτηση αυτή χρησιμοποιείται στο πεδίο ‘location’, το οποίο περιέχει συντεταγμένες σε μορφή string, διαχωρίζοντας το γεωγραφικός μήκος και πλάτος με το διαχωριστικό ‘,’. Το διαχωριστικό πρέπει να οριστεί ως παράμετρος της συνάρτησης για να κατανοηθεί η δομή τού πεδίου. Η

συνάρτηση χωρίζει το πεδίο και δημιουργεί δυο νέες στήλες στο dataframe και σβήνει το πεδίο location, όπως φαίνεται και στο αποτέλεσμα του σχήματος .

```
start=time.time()
covid_df_s = clean_lat_long(covid_df,"location", split=True, inplace=True)
end=time.time()
covid_df_s
```

```
Latitude and Longitude Cleaning Report:
        3099183 values cleaned (100.0%)
Result contains 3099183 (100.0%) values in the correct format and 0 null values (0.0%)
```

	fips_code	province_state	city_county_borough_region	date	total_death	total_confirmed	Full_Date	Year	Month	Day	latitude	longitude
0	12119	Florida	Sumter	2020-01-31	0	0	2020-01-31	2020	1	31	28.7018	-82.0794
1	13153	Georgia	Houston	2020-01-31	0	0	2020-01-31	2020	1	31	32.4580	-83.6688
2	13227	Georgia	Pickens	2020-01-31	0	0	2020-01-31	2020	1	31	34.4659	-84.4641
3	13303	Georgia	Washington	2020-01-31	0	0	2020-01-31	2020	1	31	32.9671	-82.7936
4	16003	Idaho	Adams	2020-01-31	0	0	2020-01-31	2020	1	31	44.8933	-116.4545
...	...	...	...	...	...	...	...	...	...	...	...	...
3099178	5083	Arkansas	Logan	2022-03-10	74	5076	2022-03-10	2022	3	10	35.2141	-93.7195
3099179	6009	California	Calaveras	2022-03-10	117	7595	2022-03-10	2022	3	10	38.2054	-120.5529
3099180	6035	California	Lassen	2022-03-10	60	9813	2022-03-10	2022	3	10	40.6731	-120.5935
3099181	6037	California	Los Angeles	2022-03-10	31178	2808409	2022-03-10	2022	3	10	34.3083	-118.2282
3099182	12005	Florida	Bay	2022-03-10	778	46333	2022-03-10	2022	3	10	30.2655	-85.6212

3099183 rows × 12 columns

```
arrow.get(end)-arrow.get(start)
```

```
datetime.timedelta(seconds=59, microseconds=187114)
```

Εικόνα 32: *Διαχωρισμός συντεταγμένων μέσω dataprep*

6.2.3 **Μορφοποίηση Ημερομηνίας**

Για την μετατροπή της τελευταίας στήλης που απέμεινε χωρίς μορφοποίηση, δηλαδή αυτής με την ημερομηνία κάθε εγγραφής, θα χρησιμοποιηθούν οι βιβλιοθήκες Pandas, Arrow και dataprep.clean, με σκοπό να δημιουργηθεί μια ημερομηνία με σταθερό format καθώς και κάποιες άλλες βοηθητικές στήλες με πληροφορία ημερομηνίας.

6.2.3.1 Pandas

Η μορφοποίηση της στήλης ‘Date’ του dataframe covid_df γίνεται με την βιβλιοθήκη pandas και με την βοήθεια της συνάρτησης pd.to_datetime, στην οποία τοποθετείται ως παράμετρος η στήλη που περιέχει τις ημερομηνίες, οποιαδήποτε μορφή ημερομηνίας και να έχει, και τις μετατρέπει by default στην μορφή “YYYY:MM:DD”. Μετά την αποθήκευση της

σταθερής μορφής της ημερομηνίας σε νέα στήλη με όνομα “Full_Date” δημιουργούνται τρεις ξεχωριστές στήλες, οι οποίες περιέχουν χωριστά το έτος, τον μήνα και την ημέρα, με την βοήθεια των επεκτάσεων στην συνάρτηση pd.DatetimeIndex(). year/month/day//.

```
[18]
    covid_df['Full_Date'] = pd.to_datetime(covid_df.Date)
    covid_df['Year'] = pd.DatetimeIndex(covid_df['Full_Date']).year
    covid_df['Month'] = pd.DatetimeIndex(covid_df['Full_Date']).month
    covid_df['Day'] = pd.DatetimeIndex(covid_df['Full_Date']).day

    covid_df = covid_df.drop(columns='Date')
```

Εικόνα 33: *Μορφοποίηση ημερομηνίας με Pandas*

6.2.3.2 Arrow

Η μορφοποίηση με την βιβλιοθήκη Arrow γίνεται με την βοήθεια της native Python, καθώς δεν έχει γραφτεί με βάση την pandas, οπότε δεν περιέχει αυτοματοποιημένη συνάρτηση για dataframes. Έχουν δημιουργηθεί δυο λίστες με όνομα list_dateformat2 και list_dateHumanized, έτσι ώστε μέσω της επανάληψης for να αναθέτεται η μετατροπή της ημερομηνίας σε μια μεταβλητή, στο index που βρίσκεται και μετά να το αποθηκεύει στην λίστα. Στην πρώτη λίστα αποθηκεύεται η ημερομηνία μέσω της συνάρτησης *arrow.get.().format('MMMM D YYYY'),* σε μορφή “July 2 2019”, και στην δεύτερη μετατρέπεται με την βοήθεια της συνάρτησης *arrow.get.().humanize* σε μορφή λέξεων. Στο

```
#covid_df['Date'] = pd.to_datetime(covid_df.Date)

list_dateformat2 =[]
list_dateHumanized =[]
for ind in covid_df.index:
    date = arrow.get(covid_df['Full_Date'].iloc[ind]).format('MMMM D YYYY')
    list_dateformat2.append(date)
    date2 = arrow.get(covid_df['Full_Date'].iloc[ind]).humanize(locale='el')
    list_dateHumanized.append(date2)
    #print(ind)
    if ind == len(covid_df) :
      break


covid_df['Full_Date_format 2'] = pd.DataFrame(data=list_dateformat2)
covid_df['Full_Date_Humanized'] = pd.DataFrame(data=list_dateHumanized)
```

Εικόνα 34: *Μορφοποίηση ημερομηνιών με Arrow*

τελευταίο βήμα δημιουργούνται δυο στήλες στο dataframe, όπου και αποθηκεύονται οι δύο λίστες.

6.2.3.3 dataprep.clean

```
[240] from dataprep.clean import clean_date
      start=time.time()
      covid_df=clean_date(covid_df, 'date',output_format='yyyy/MM/dd')
      end=time.time()

      Dates Cleaning Report:
              3099183 values cleaned (100.0%)
      Result contains 3099183 (100.0%) values in the correct format and 0 null values (0.0%)
```

```
covid_df
```

	fips_code	province_state	city_county_borough_region	date	total_death	total_confirmed	location	date_clean
0	12119	Florida	Sumter	2020-01-31	0	0	28.70181754,-82.0794267	2020/01/31
1	13153	Georgia	Houston	2020-01-31	0	0	32.45802497,-83.66879087	2020/01/31
2	13227	Georgia	Pickens	2020-01-31	0	0	34.46589159,-84.46406611	2020/01/31
3	13303	Georgia	Washington	2020-01-31	0	0	32.96711864,-82.79357039	2020/01/31
4	16003	Idaho	Adams	2020-01-31	0	0	44.89333571,-116.4545247	2020/01/31
...	...	...	...	...	...	...	...	...
3099178	5083	Arkansas	Logan	2022-03-10	74	5076	35.21413234,-93.71951016	2022/10/03
3099179	6009	California	Calaveras	2022-03-10	117	7595	38.20537103,-120.552913	2022/10/03
3099180	6035	California	Lassen	2022-03-10	60	9813	40.67311306,-120.59350990000002	2022/10/03
3099181	6037	California	Los Angeles	2022-03-10	31178	2808409	34.30828379,-118.2282411	2022/10/03
3099182	12005	Florida	Bay	2022-03-10	778	46333	30.26548745,-85.62122584	2022/10/03

3099183 rows × 8 columns

```
arrow.get(end)-arrow.get(start)
```

```
datetime.timedelta(seconds=649, microseconds=724441)
```

Εικόνα 35: *Μορφοποίηση ημερομηνίας με την χρήση dataprep*

Η βιβλιοθήκη dataprep.clean περιέχει την συνάρτηση clean_date, η οποία πραγματοποιεί αυτοματοποιημένη μετατροπή ενός πεδίου που περιέχει ημερομηνίες σε οποιοδήποτε format επιθυμεί να μετατρέψει κάποιος προγραμματιστής το γνώρισμα της ημερομηνίας. Παρακάτω πραγματοποιείται αυτή η μετατροπή σε format 'yyyy/MM/dd'.

Η μορφή του dataframe έπειτα από τα βήματα που ακολουθήθηκαν είναι η παρακάτω:

	fips_code	province_state	city_county_borough_region	date	total_death	total_confirmed	date_clean	Full_Date	Year	Month	Day	Full_Date_format 2	Full_Date_Humanized	latitude	longitude
0	12119	Florida	Sumter	2020-01-31	0	0	2020/01/31	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	28.70	-82.08
1	13153	Georgia	Houston	2020-01-31	0	0	2020/01/31	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	32.46	-83.67
2	13227	Georgia	Pickens	2020-01-31	0	0	2020/01/31	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	34.47	-84.46
3	13303	Georgia	Washington	2020-01-31	0	0	2020/01/31	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	32.97	-82.79
4	16003	Idaho	Adams	2020-01-31	0	0	2020/01/31	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	44.89	-116.45
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3099178	5083	Arkansas	Logan	2022-03-10	74	5076	2022/10/03	2022-03-10	2022	3	10	March 10 2022	6 μήνες πριν	35.21	-93.72
3099179	6009	California	Calaveras	2022-03-10	117	7595	2022/10/03	2022-03-10	2022	3	10	March 10 2022	6 μήνες πριν	38.21	-120.55
3099180	6035	California	Lassen	2022-03-10	60	9813	2022/10/03	2022-03-10	2022	3	10	March 10 2022	6 μήνες πριν	40.67	-120.59
3099181	6037	California	Los Angeles	2022-03-10	31178	2808409	2022/10/03	2022-03-10	2022	3	10	March 10 2022	6 μήνες πριν	34.31	-118.23
3099182	12005	Florida	Bay	2022-03-10	778	46333	2022/10/03	2022-03-10	2022	3	10	March 10 2022	6 μήνες πριν	30.27	-85.62

3099183 rows × 15 columns

Εικόνα 36*: Τελική εικόνα του σετ δεδομένων μετά από το βήμα μορφοποίησης*

6.2.4 **Σύγκριση**

Στην πρώτη εργασία που εκτελέστηκε σχετικά με την «Μορφοποίηση ονομάτων και τύπων γνωρισμάτων», οι βιβλιοθήκες Pandas και dataprep.clean χρησιμοποιούν ίδιες ακριβώς γραμμές κώδικα, κάτι που δείχνει σαφή επάρκεια συναρτήσεων για τη ζητούμενη εργασία. Η dataprep, όπως φαίνεται από τα αποτελέσματα (πίνακας 2), λειτουργεί πιο γρηγορά με τις δικές της συναρτήσεις. Το σημαντικότερο ,όμως, πλεονέκτημα που έχει η dataprep.clean ενάντια στις συναρτήσεις της Pandas είναι ότι κάνει downcasting στα γνωρίσματα την στιγμή που τους αλλάζει τον τύπο, χωρίς να επηρεάζεται από NaN τιμές. Αντίθετα, η downcasting υποβιβλιοθήκη που έχει η Pandas δεν μπορεί να πραγματοποιηθεί με ελλιπείς τιμές.

```
import pdcast as pdc
start=time.time()
covid_df.rename(columns = {'Admin 2 FIPS Code':'FIPS Code',
                           'Admin 2 Level (City/County/Borough/Region)':'City_County_Borough_Region',
                           'Province/State':'Province_State'},
                inplace = True)
covid_df = covid_df.convert_dtypes()
pdc.downcast(covid_df)

end=time.time()
covid_df.info()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-33-685e9ec69221> in <module>
      6                 inplace = True)
      7 covid_df = covid_df.convert_dtypes()
----> 8 pdc.downcast(covid_df)
      9
     10 end=time.time()

                              10 frames
/usr/local/lib/python3.7/dist-packages/pandas/core/arrays/masked.py in to_numpy(self, dtype, copy, na_value)
    291             ):
    292                 raise ValueError(
--> 293                     f"cannot convert to '{dtype}'-dtype NumPy array "
    294                     "with missing values. Specify an appropriate 'na_value' "
    295                     "for this dtype."

ValueError: cannot convert to 'uint16'-dtype NumPy array with missing values. Specify an appropriate 'na_value' for this dtype.

SEARCH STACK OVERFLOW
```

Εικόνα 37: *Σφάλμα pandas με την downcasting*

Πίνακας 2*: Fix Data shape, σύγκριση των modules 1*

1.Μορφοποίηση ονομάτων και τύπων γνωρισμάτων	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*	*Χρηστικότητα*
Pandas	2	78 microseconds	
dataprep.clean	2	52.55 seconds	

Στην δεύτερη εργασία που αφορά στη μορφοποίηση συντεταγμένων που περιέχονται στο γνώρισμα ‘location’ έχουν χρησιμοποιηθεί οι βιβλιοθήκες Geopandas και

dataprep.clean. Η dataprep.clean κάνει αξιόλογη δουλειά χωρίζοντας, με μια μόνο συνάρτηση και γραμμή κώδικα (πίνακας 3), το γνώρισμα με τις συντεταγμένες σε δυο στήλες , μέσα στο ίδιο dataframe. Έτσι, καθιστά εύκολη την διαχείρισή τους για τους αναλυτές των δεδομένων. Με την βιβλιοθήκη Geopandas, αν και χρειαζόμαστε τον διπλάσιο χρόνο για την πραγματοποίηση της διαχώρισης του γνωρίσματος, λαμβάνεται πίσω ένα geodataframe το οποίο υπερέχει σε λειτουργικότητα και προοπτική μελλοντικής αξιοποίησης από άλλα modules, σε σχέση με την dataprep.clean.

***Πίνακας 3**: Fix Data shape, σύγκριση των modules 2*

2.Μορφοποίηση συντεταγμένων	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*	*Χρηστικότητα*
Geopandas	2	138.6 seconds	
dataprep.clean	1	59.18 seconds	

Όσον αφορά στην τρίτη εργασία για την μορφοποίηση της ημερομηνίας έχουν χρησιμοποιηθεί τρία διαφορετικά modules. Η πιο γρήγορη σύμφωνα με τα αποτελέσματα (πίνακας 4) είναι η pandas, με τέσσερις γραμμές κώδικα και αρκετά μεγάλη ποικιλία τόσο ως προς το input που δέχεται, για να το μετατρέψει σε ημερομηνία αλλά όσο και ως προς το output. Μεγάλη ποικιλομορφία διαθέτει και η βιβλιοθήκη dataprep.clean η οποία είναι πάρα πολύ αργή σχετικά με την Pandas. Τέλος, η Arrow θα μπορούσε να χαρακτηριστεί η πιο δύσχρηστη σχετικά με τις άλλες δυο αφού για την μετατροπή χρειάζεται έντεκα γραμμές κώδικα, εξαιτίας του ότι δεν είναι συμβατή με dataframe και θα πρέπει με την βοήθεια τη Python να επεξεργαστεί το πεδίο της ημερομηνίας. Η Arrow έχει το αρνητικό, ως προς την λειτουργικότητα, του ότι δεν μπορεί να μετατρέψει ημερομηνίες, πέρα από εκείνες που ξεκινούν με το έτος, όπως πχ '13/12/2016'.

***Πίνακας 4:** Fix Data shape, σύγκριση των modules 3*

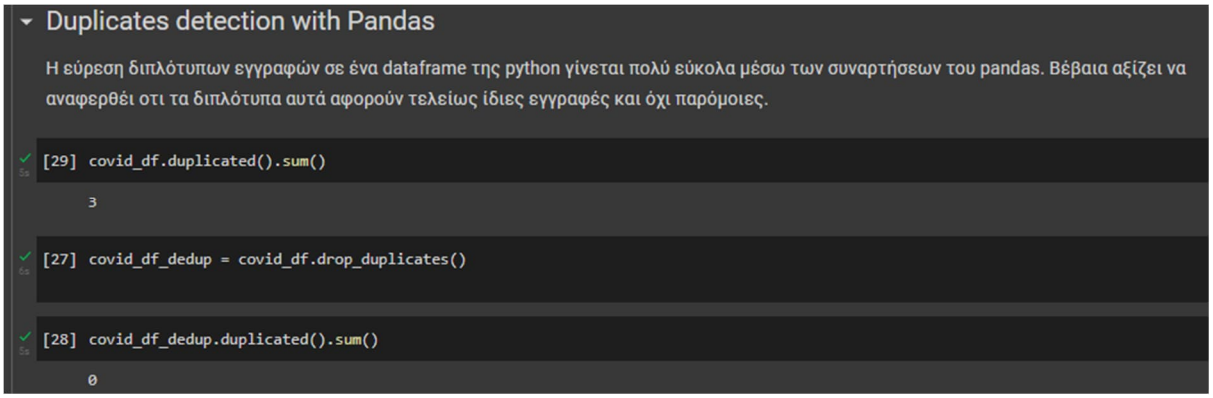
3 Μορφοποίηση Ημερομηνίας	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*
Pandas	4	1.79 seconds
dataprep.clean	2	649.72 seconds
Arrow	11	345.94 seconds

6.3 **Υλοποίηση** Deduplication of Data

Ο αποδιπλασιασμός των δεδομένων, που αφορά στην εύρεση εγγραφών που επαναλαμβάνονται για το ίδιο αντικείμενο του πραγματικού κόσμου, θα γίνει με την βιβλιοθήκη pandas και με τον cross chain αλγόριθμο, που χρησιμοποιεί την βιβλιοθήκη Fuzzywuzzy.

6.3.1 Pandas

Με την βοήθεια του βήματος Data overview γνωρίζουμε ότι το dataframe covid_df έχει τρεις εγγραφές που είναι επαναλαμβανόμενες, χωρίς να εξυπηρετούν κάποιο σκοπό. Αντιθέτως, υποβαθμίζουν την αξιοπιστία των δεδομένων. Η υλοποίηση μέσω της pandas είναι πολύ απλή με την χρήση της συνάρτηση .drop_duplicates, η οποία αφαιρεί τις εγγραφές που είχαν βρεθεί και νωρίτερα. Όπως φαίνεται παρακάτω από την υλοποίησή της, μετά από την εφαρμογή της συνάρτησης, υπάρχουν 0 διπλότυπες εγγραφές.



Εικόνα 38: *Εφαρμογή drop_duplicates() στο σετ δεδομένων covid_df*

6.3.2 Cross Chain Algorithm with Fuzzywuzzy

Ο αλγόριθμος cross chain είναι ένας επαναλαμβανόμενος αλγόριθμος ο οποίος ψάχνει για διπλότυπες εγγραφές οι οποίες δεν έχουν ακριβώς ίδια χαρακτηριστικά, αλλά μοιάζουν αρκετά. Ο πυρήνας του αλγορίθμου βρίσκεται στη συνάρτηση find_partition. Η ιδέα είναι να επεκταθεί αναδρομικά ένα σύνολο περιπτώσεων που ταιριάζουν. Το γεγονός ότι δύο περιπτώσεις ταιριάζουν αποφασίζεται από την παράμετρο match_func, που πρέπει να παρέχεται από τον χρήστη. Ο αλγόριθμος επιστρέφει ένα pandas.Series που περιέχει ακέραιους αριθμούς που συσχετίζουν κάθε τιμή ευρετηρίου με ένα αναγνωριστικό αριθμό

οντότητας. Στην εν λόγω περίπτωση, μέσω των συναρτήσεων που ξεκινάνε με “same_”, έχουν οριστεί οι κανόνες ισότητας, προκειμένου να θεωρούνται δυο οντότητες παρόμοιες και ο κεντρικός κανόνας να βρίσκεται στην same_records. Στις πρώτες τρεις εικόνες παρουσιάζεται ο κώδικας του αλγορίθμου. Στην επόμενη εικόνα διαπιστώνεται ότι με την χρήση της συνάρτησης find_partitions δημιουργείται μια νέα στήλη που περιλαμβάνει τα νέα indexes. Αφού πραγματοποιηθεί η διαδικασία αυτή, αναζητούνται μέσω της duplicated() τα διπλότυπα indexes και με την χρήση της drop_duplicates() διαγράφονται.

```python
# https://maxhalford.github.io/blog/transitive-duplicates/
import numpy as np
import pandas as pd


def find_partitions(df, match_func, max_size=None, block_by=None):
    """Recursive algorithm for finding duplicates in a DataFrame."""

    # If block_by is provided, then we apply the algorithm to each block and
    # stitch the results back together
    if block_by is not None:
        blocks = df.groupby(block_by).apply(lambda g: find_partitions(
            df=g,
            match_func=match_func,
            max_size=max_size
        ))

        keys = blocks.index.unique(block_by)
        for a, b in zip(keys[:-1], keys[1:]):
            blocks.loc[b, :] += blocks.loc[a].iloc[-1] + 1

        return blocks.reset_index(block_by, drop=True)

    def get_record_index(r):
        return r[df.index.name or 'index']

    # Records are easier to work with than a DataFrame
    records = df.to_records()

    # This is where we store each partition
    partitions = []
```

***Εικόνα 40**:πρώτο μέρος αλγόριθμου cross chain*

```python
def find_partition(at=0, partition=None, indexes=None):

    r1 = records[at]

    if partition is None:
        partition = {get_record_index(r1)}
        indexes = [at]

    # Stop if enough duplicates have been found
    if max_size is not None and len(partition) == max_size:
        return partition, indexes

    for i, r2 in enumerate(records):

        if get_record_index(r2) in partition or i == at:
            continue

        if match_func(r1, r2):
            partition.add(get_record_index(r2))
            indexes.append(i)
            find_partition(at=i, partition=partition, indexes=indexes)

    return partition, indexes

while len(records) > 0:
    partition, indexes = find_partition()
    partitions.append(partition)
    records = np.delete(records, indexes)

return pd.Series({
    idx: partition_id
    for partition_id, idxs in enumerate(partitions)
    for idx in idxs
})
```

***Εικόνα 39**: δεύτερο μέρος αλγόριθμου cross chain*

```
from fuzzywuzzy import fuzz


def same_FIPS_code(r1, r2):
    return r1['FIPS Code'] == r2['FIPS Code']


def similar_area(r1, r2):
    return (abs(r1['Latitude']- r2['Latitude'])<0.001 and
            abs(r1['Longitude']- r2['Longitude'])<0.001 and
            fuzz.partial_ratio(r1['City_County_Borough_Region'],r2['City_County_Borough_Region'])> 70
            )



def same_state(r1, r2):
    return fuzz.partial_ratio(r1['Province_State'], r2['Province_State']) > 70



def similar_numer_of_Deaths(r1, r2):
    return (r1['Total Death']==r2['Total Death'] and
            r1['Total Confirmed']==r2['Total Confirmed'])



def same_date(r1,r2):
  return(r1['Full_Date']==r2['Full_Date'])



def same_record(r1, r2):
    return (
        same_date(r1,r2) and
        same_FIPS_code(r1, r2) and
        similar_area(r1, r2) and
        same_state(r1, r2) and
        similar_numer_of_Deaths(r1, r2)
    )
```

Εικόνα 41: *τρίτο μέρος αλγόριθμου cross chain μαζί με Fuzzywuzzy βιβλιοθήκη*

```
[45] covid_df['real_id'] = find_partitions(df=covid_df,match_func=same_record)

[46] covid_df[covid_df.duplicated('real_id',keep=False)]
```

	FIPS Code	Province_State	City_County_Borough_Region	Total Death	Total Confirmed	Full_Date	Year	Month	Day	Full_Date_format 2	Full_Date_Humanized	Latitude	Longitude	real_id
12	19115	Iowa	Louisa	0.0	0.0	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	41.21842119	-91.26102611	12
13	19115	Iowa	Louisa	0.0	0.0	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	41.21842119	-91.26102611	12
26	22123	Louisiana	West Carroll LS	0.0	0.0	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	32.78550125	-91.461201	25
27	22123	Louisiana	West Carroll	0.0	0.0	2020-01-31	2020	1	31	January 31 2020	2 χρόνια πριν	32.78550125	-91.461201	25
2461	12061	Florida	Indian River	306.0	15477.0	2021-08-11	2021	8	11	August 11 2021	ένα χρόνο πριν	27.69308961	-80.60556721	2459
2462	12061	Florida	Indian River	306.0	15477.0	2021-08-11	2021	8	11	August 11 2021	ένα χρόνο πριν	27.69308961	-80.60556721	2459
2463	13081	Georgia	Crisp	94.0	2270.0	2021-08-11	2021	8	11	August 11 2021	ένα χρόνο πριν	31.92289565	-83.76811843	2460
2781	13081	Georgia	Crisp	94.0	2270.0	2021-08-11	2021	8	11	August 11 2021	ένα χρόνο πριν	31.92289565	-83.76811843	2460
9984	31117	Nebraska	McPherson	1.0	29.0	2021-06-08	2021	6	8	June 8 2021	ένα χρόνο πριν	41.56797682	-101.0599362	9980
9985	31117	Nebraska	McPherson	1.0	29.0	2021-06-08	2021	6	8	June 8 2021	ένα χρόνο πριν	41.56797682	-101.0599362	9980
9986	31117	Nebraska	McPherson	1.0	29.0	2021-06-08	2021	6	8	June 8 2021	ένα χρόνο πριν	41.56797682	-101.0599362	9980
10001	37179	North Carolina	Union	225.0	24740.0	2021-06-08	2021	6	8	June 8 2021	ένα χρόνο πριν	34.98960527	-80.52955873	9995
10002	37179	North Carolina	Union NC	225.0	24740.0	2021-06-08	2021	6	8	June 8 2021	ένα χρόνο πριν	34.98960520	-80.52955871	9995

Εικόνα 42: *Εφαρμογή αλγορίθμου στα dataset covid_df*

```
covid_df_dedu2['real_id_2'] = find_partitions(df=covid_df_dedu2,match_func=same_record)

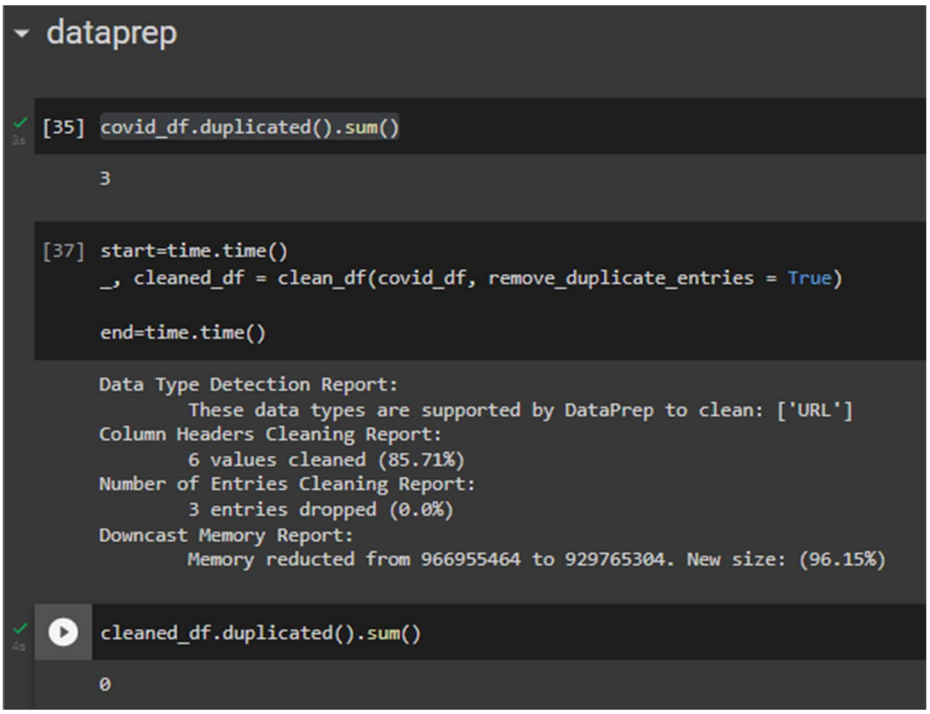
[57] covid_df_dedu2[covid_df_dedu2.duplicated('real_id_2',keep=False)]
```

FIPS Code	Province_State	City_County_Borough_Region	Total Death	Total Confirmed	Full_Date	Year

Εικόνα 43*: Έλεγχος αποδιπλασιασμού δεδομένων με την βοήθεια της pandas*

6.3.3 dataprep.clean

Η βιβλιοθήκη dataprep.clean περιέχει την συνάρτηση clean_df() στην οποία, όταν εισαχθεί η παράμετρος remove_duplicate_entries, εξυπηρετείται ο σκοπός εύρεσης διπλότυπων πλειάδων μέσα σε ένα dataframe. Η υλοποίηση επιτυγχάνεται, όπως φαίνεται και στο παρακάτω σχήμα που την απεικονίζει.



***Εικόνα 44**: Αποδιπλασιασμός dataset με την χρήση της dataprep.clean*

6.3.4 **Σύγκριση**

Οι βιβλιοθήκες Pandas και dataprep.clean επιτυγχάνουν τον ίδιο σκοπό με μία συνάρτηση η καθεμία. Η συνάρτηση αφαιρεί τις ίδιες ακριβώς εγγραφές που βρίσκει μέσα στο dataframe που του ορίζεται. Η συνάρτηση της pandas είναι 8 φορές πιο γρήγορη από εκείνη της dataprep.clean. Ο αλγόριθμος cross chain με την βοήθεια της βιβλιοθήκης Fuzzywuzzy επιτυγχάνει έναν πολύ περίπλοκο στόχο, να βρει πλειάδες που τα γνωρίσματα τους είναι

αρκετά παρόμοια για να θεωρούνται διπλότυπα. Επί της ουσίας κάνει την ανίχνευση των περίπου όμοιων πλειάδων και αφού ο χρήστης συμφωνήσει με τις εγγραφές που ο αλγόριθμος θεωρεί διπλότυπες, τις διαγράφει από το dataframe. Η πολυπλοκότητα του προβλήματος είναι έκδηλη και στο χρόνο που χρειάζεται να εκτελεστεί.

***Πίνακας 5**:Deduplication of data, σύγκριση των modules*

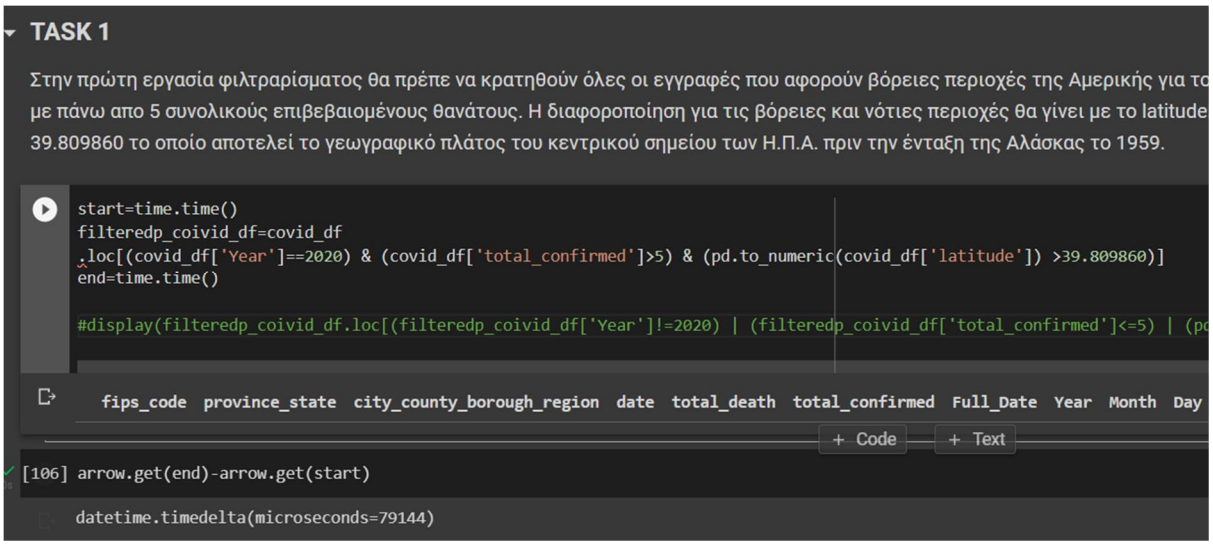
Deduplication of Data	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*	*Χρηστικότητα*
Pandas	1	7.39 seconds	Εντοπισμός όμοιων διπλότυπων
Dataprep.clean	1	57.94 seconds	Διαγραφή όμοιων διπλότυπων
Cross chain + Fuzzywuzzy	2 + (58)	3+ hours	Εύρεση περίπου όμοιων πλειάδων

6.4 Remove Irrelevant Data

Σε αυτό το σημείο έχει δημιουργηθεί ένα σενάριο κατά το οποίο ορισμένα δεδομένα από το υπάρχον σετ δεδομένων δεν χρησιμεύουν στους επόμενους κύκλους εργασιών. Από το σετ δεδομένων θα πρέπει να εξαιρεθούν όλες οι εγγραφές που αφορούν στις νότιες περιοχές της Αμερικής για το χρονολογικό έτος 2020 στις οποίες έχουν καταγραφεί περισσότεροι από 5 συνολικά θάνατοι. Η υλοποίηση γίνεται με τις βιβλιοθήκες Pandas και NumPy.

6.4.1 filtering **με** Pandas

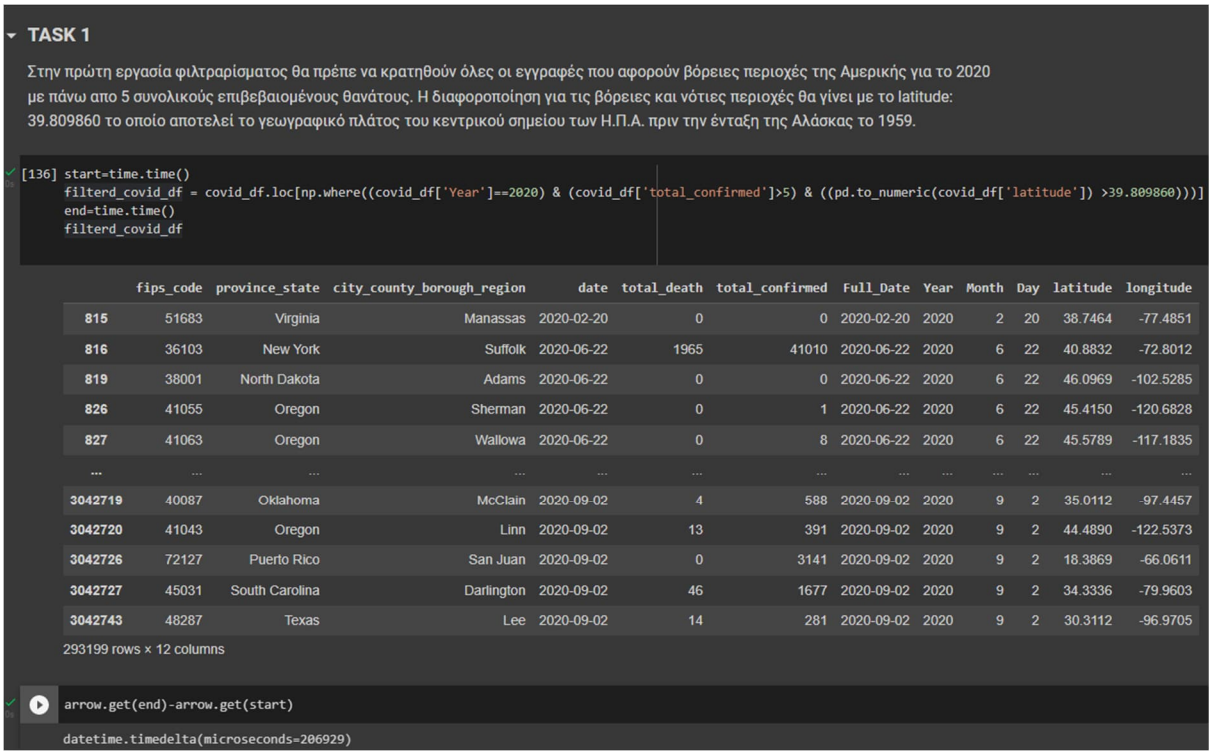
Στην υλοποίηση με την βιβλιοθήκη Pandas χρησιμοποιείται η συνάρτηση loc[]. Με την συνάρτηση loc() πραγματοποιείται η εύρεση του index, μέσω κάποιων Boolean συγκρίσεων. Όπως φαίνεται στην υλοποίηση πιο κάτω, έχουν δημιουργηθεί τρεις ανισότητες. Μόνο αν ισχύουν όλες, θα εμφανιστεί η πλειάδα μέσω της loc(). Για το κέντρο του γεωγραφικού πλάτους έχει ορισθεί ο αριθμός 39.809860, ο οποίος ήταν το γεωγραφικό πλάτος του κεντρικού σημείο των Η.Π.Α. πριν την ένταξη της Αλάσκας.



Εικόνα 45: *Φιλτράρισμα δεδομένων με την χρήση Pandas συναρτήσεων*

6.4.2 filtering με NumPy

Στην δεύτερη υλοποίηση έχει γίνει η ίδια ακριβώς διαδικασία, με την βοήθεια της βιβλιοθήκης NumPy και πιο συγκεκριμένα με την συνάρτηση np.where. Η συνάρτηση np.where χρησιμοποιήθηκε για να μας επιστρέψει έναν πίνακα από indexes, τα οποία προέκυψαν από τις ίδιες ανισότητες που εφαρμοστήκαν και πριν, στο dataframe covid_df.



Εικόνα 46: *Φιλτράρισμα δεδομένων με την χρήση NumPy συναρτήσεων*

6.4.3 **Σύγκριση**

Οι δύο βιβλιοθήκες χρησιμοποιούν τον ίδιο τρόπο για την εύρεση των ζητούμενων δεδομένων, δηλαδή προσπαθούν να βρουν το index κάθε πλειάδας που συμφωνεί με τις προϋποθέσεις που έχουν τεθεί. Η μόνη διαφορά τους είναι ότι μέσω της pandas το φιλτράρισμα μπορεί να γίνει γρήγορα, σε μια μόνο εντολή κώδικα. Με την απευθείας ανάθεση της συνάρτησης np.where μέσα στην loc() παρατηρείται ότι, παρότι χρησιμοποιείται μια γραμμή κώδικα, αυξάνεται η πολυπλοκότητα και ο χρόνος εκτέλεσης. Οπότε έχουμε δυο συναρτήσεις που κάνουν ακριβώς το ίδιο, με την διαφορά ότι η βιβλιοθήκη pandas είναι πιο γρήγορη.

***Πίνακας 6**: Remove irrelevant data, σύγκριση των modules*

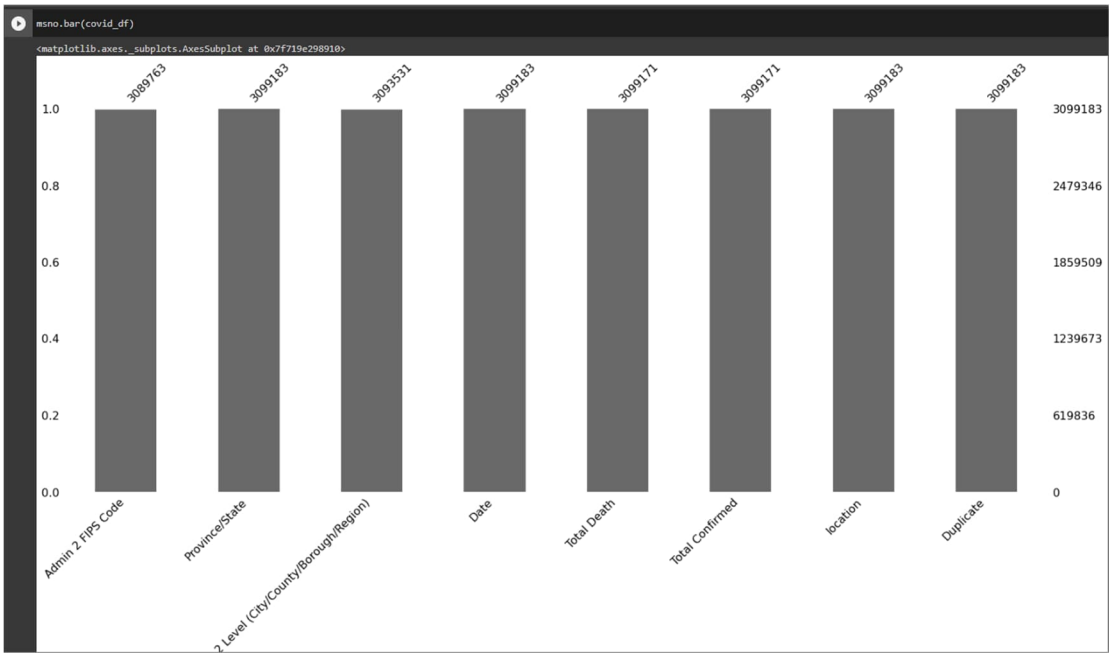
Remove irrelevant data	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*
Pandas	1	0.08 seconds
NumPy	1	0.20 seconds

6.5 Deal with missing values

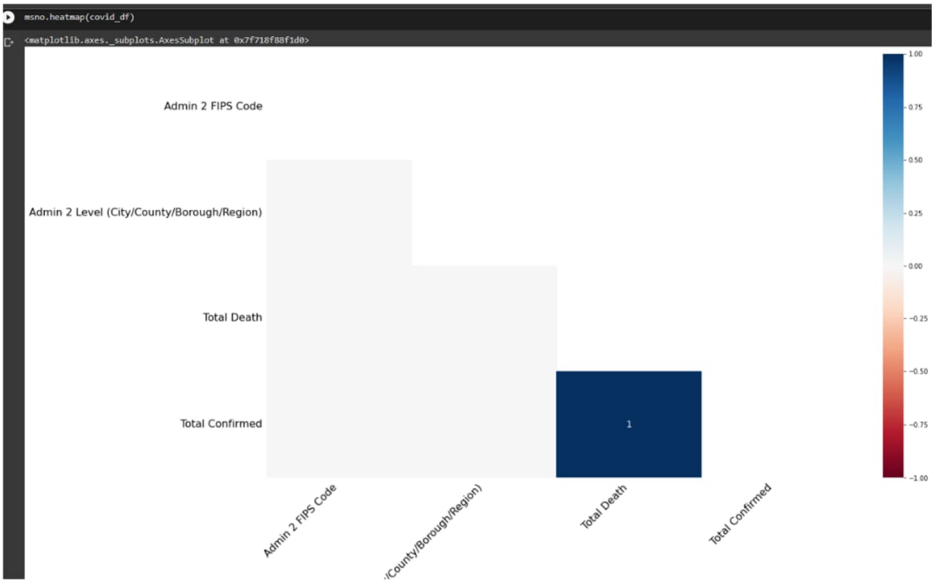
6.5.1 Missingno

Τα διαγράμματα που προσφέρει η βιβλιοθήκη missingno χρησιμοποιούνται, για να κατανοηθεί οπτικά το πόσο πλήρη είναι τα δεδομένα μας. Χρησιμοποιήθηκαν τρία διαφορετικά σχήματα οπτικοποίησης που διαθέτει η βιβλιοθήκη .matrix(), .bar() και .heatmap(). Στα πρώτα δυο φαίνεται ότι ο πίνακας δεδομένων έχει κάποιες αλλά όχι πολλές ελλιπείς τιμές, κάτι που σημαίνει ότι τα δεδομένα έχουν πυκνή πληροφορία και σε αυτό το κομμάτι δεν θα χάνονταν πολλές εγγραφές, αν αποφασιζόταν να αφαιρεθούν όσες εγγραφές έχουν έστω και μία ελλιπή τιμή.

Στο τρίτο διάγραμμα παρουσιάζονται οι συσχετίσεις που μπορεί να ισχύουν ανάμεσα στις στήλες των ελλιπών τιμών. Στα αποτελέσματα που παρουσιάζονται, οι δύο στήλες που έχουν άμεση συσχέτιση είναι οι στήλες που αφορούν στους θανάτους κάθε περιοχής.



Εικόνα 47: *Διάγραμμα matrix της missingno για ελλιπείς τιμές*



Εικόνα 48: *Διάγραμμα correlation της missingno*

.

6.5.2 Datacleaner

Στην περίπτωση των ελλιπών τιμών έχει χρησιμοποιηθεί το datacleaner, το οποίο διαθέτει την συνάρτηση autoclean, η οποία αυτοματοποιεί κάποιες διαδικασίες διαχείρισής τους. Η autoclean() βρίσκει τις κενές τιμές και τις αντικαθιστά σύμφωνα με τον τύπο της στήλης όπου βρέθηκαν. Στην περίπτωση του dataframe covid_df, τα κενά που βρίσκονται στις τύπου int στήλες (fips_code, total_death και total_confirmed), έχουν πάρει την τιμή του μέσου όρου της στήλης '30072','28 και '1658' αντίστοιχα. Στην στήλη string 'city_county_borough_region' που βρέθηκαν κενές τιμές ανατέθηκε η διάμεσος τιμή που θεωρείται ότι είναι η περιοχή 'Washington', η οποία είναι η δεύτερη πιο συχνή λέξη στην στήλη αυτή.

```
from datacleaner import autoclean
start=time.time()
cleaned_covid_df=covid_df.copy()
autoclean(cleaned_covid_df)
end=time.time()
cleaned_covid_df

[146] arrow.get(end)-arrow.get(start)

      datetime.timedelta(seconds=2, microseconds=906416)
```

***Εικόνα 49**: Καθαρισμός ελλιπών τιμών με datacleaner*

```
covid_df.isnull().isnull()

indexes = covid_df[covid_df['city_county_borough_region'].isnull()].index

covid_df.loc[indexes].head(5)
```

	fips_code	province_state	city_county_borough_region	date	to
175	69	Northern Mariana Islands	<NA>	2020-02-01	
675	99999	Grand Princess	<NA>	2020-02-21	
1150	66	Guam	<NA>	2020-06-13	
1563	88888	Diamond Princess	<NA>	2020-06-16	
1900	60	American Samoa	<NA>	2020-06-19	
2550	78	Virgin Islands	<NA>	2021-08-11	
3091	66	Guam	<NA>	2021-08-07	
4968	69	Northern Mariana Islands	<NA>	2021-08-04	

***Εικόνα 50:** Εμφάνιση των πλειάδων με null από το covid_df*

```
[140] cleaned_covid_df.loc[indexes].head(50)
```

	fips_code	province_state	city_county_borough_region
175	69	Northern Mariana Islands	Washington
675	99999	Grand Princess	Washington
1150	66	Guam	Washington
1563	88888	Diamond Princess	Washington
1900	60	American Samoa	Washington
2550	78	Virgin Islands	Washington
3091	66	Guam	Washington
4968	69	Northern Mariana Islands	Washington

***Εικόνα 51**: Προβολή των ίδιων πλειάδων με εικόνα 50, μετά την εφαρμογή datacleaner στην covid_df*

6.5.3 Pandas

Με την χρήση της βιβλιοθήκης pandas πραγματοποιείται η διαδικασία εύρεσης και διαχείρισης των ελλιπών τιμών. Με τις συναρτήσεις isnull(), fillna() και dropna() αντιμετωπίζεται η ύπαρξη των ελλιπών τιμών στο dataframe covid_df. Αρχικά, για την αντιμετώπιση των ελλιπών τιμών έχει ανατεθεί η τιμή 0 για όλες τις κενές τιμές του fips_code μέσω της συνάρτησης fillna(). Επιπλέον, με την χρήση της ίδιας συνάρτησης ανατίθεται η τιμή της στήλης ‘province_state’ σε κάθε ‘NaN’ τιμή της στήλης ‘city_county_borough_region’. Τέλος, σβήνονται όσες πλειάδες έχουν κενή τιμή, εν προκειμένω οι στήλες ‘total_death’ και ‘total_confirmed’.

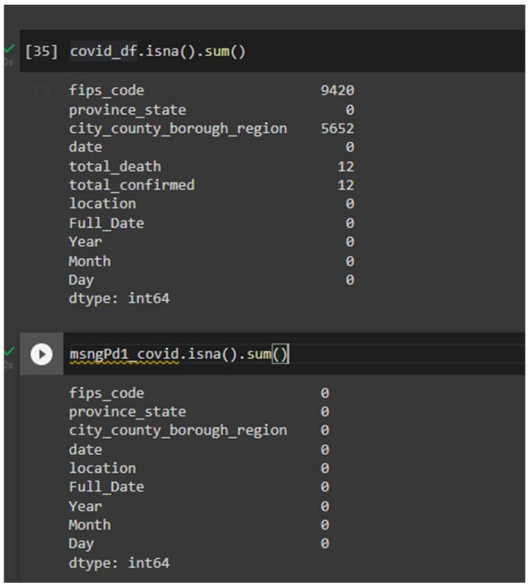
Cleaning with Pandas

```
[156] msngPd1_covid = covid_df.copy()
      start=time.time()
      msngPd1_covid['fips_code']=msngPd1_covid['fips_code'].fillna(0)
      msngPd1_covid['city_county_borough_region']=msngPd1_covid['city_county_borough_region'].fillna(covid_df['province_state'])
      msngPd1_covid=msngPd1_covid.dropna(axis='columns')
      end=time.time()

      arrow.get(end)-arrow.get(start)

      datetime.timedelta(microseconds=852385)
```

***Εικόνα 52**: Καθαρισμός σετ δεδομένων με Pandas*



Εικόνα 53: *Προβολή ελλιπών τιμών dataset πριν και μετά την χρήση pandas*

6.5.4 **Σύγκριση**

Η πρώτη βιβλιοθήκη missingno χρησιμοποιείται ειδικά για την οπτικοποίηση των ελλιπών τιμών, οπότε δεν θα ήταν δόκιμο να συγκριθεί με τα άλλα δυο modules που τις διαχειρίζονται. Η βιβλιοθήκη datacleaner διαθέτει την συνάρτηση autocleaner(), η οποία αντικαθιστά τις τιμές που είναι κενές με το μέσο όρο ή την διάμεσο. Στην περίπτωση του covid_df δίνει πολύ καλή λύση στις αριθμητικές στήλες 'total_death' και 'total_confirmed' που έχουν NaN, χρησιμοποιώντας τον μέσο όρο της κάθε μιας. Αντιθέτως, στις δύο άλλες περιπτώσεις στηλών βγάζει λάθος συμπεράσματα οπότε και δημιουργεί έλλειψη ακρίβειας στο dataframe. Με την βιβλιοθήκη pandas και τις συναρτήσεις της, οι ελλιπείς τιμές έχουν καλυφθεί ανά περίπτωση, κάτι που αυξάνει την ακρίβεια και την εγκυρότητα των δεδομένων του dataframe. Επίσης, αν και η προσέγγιση με pandas έχει περισσότερες γραμμές κώδικα εκτελείται πιο γρήγορα από την μια γραμμή του datacleaner.

Πίνακας 7: *Deal with missing data, σύγκριση των modules*

Deal with missing data	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*	*Χρηστικότητα*
Missingno	X	X	Απεικόνιση ελλιπών τιμών
Datacleaner	1	2.9 seconds	Κάλυψη ελλιπών τιμών
Pandas	3	0.85 seconds	Διαχείριση ελλιπών τιμών

6.6 Detection of data outliers

Η διαδικασία ανίχνευσης των έκτοπων τιμών έχει γίνει με την βοήθεια τεσσάρων βιβλιοθηκών, των statsmodels-seaborn, της pyod και της outlier_utils. Αρχικά έχει γίνει κανονικοποίηση στις αριθμητικές στήλες του covid_df, με στόχο την εύρεση των έκτοπων τιμών.

6.6.1 Statsmodels-Seaborn

Η statsmodels με την χρήση της απόστασης Cook εφαρμόζεται με σκοπό την γραφική απεικόνιση των τιμών των αριθμητικών δεδομένων των στηλών σε ζευγάρια (‘latitude-longitude’ και total death ‘total confirmed’) της covid_df, έτσι ώστε να ληφθεί η γνώση για ενδεχόμενες έκτοπες τιμές τους. Έχει εφαρμοστεί το διάγραμμα scatterplot() από την seaborn, που παρέχει την δυνατότητα προβολής του καταμερισμού των τιμών και διαχωρίζει τα inliers με τα outliers, αναπαριστώντας με κίτρινο χρώμα τα πιθανά inliers και μοβ τα πιθανά outliers.

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.outliers_influence import OLSInfluence

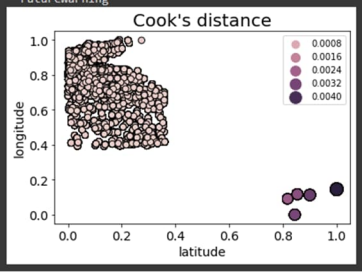
#regression model
f = 'latitude ~ longitude'
model = ols(formula=f, data=df1_out_norm).fit()

# calculate the cooks_distance - the OLSInfluence object contains multiple influence measurements
cook_distance = OLSInfluence(model).cooks_distance
(distance, p_value) = cook_distance
# scatter plot - x axis (independent variable height), y-axis (dependent variable weight)
sns.scatterplot(df1_out_norm.latitude , df1_out_norm.longitude, hue=distance, size=distance, sizes=(50, 200), edgecolor='black', linewidth=1)

plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

plt.xlabel('latitude', fontsize=14)
plt.ylabel('longitude', fontsize=14)
plt.title('Cook\'s distance', fontsize=20);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From vers
  FutureWarning
```



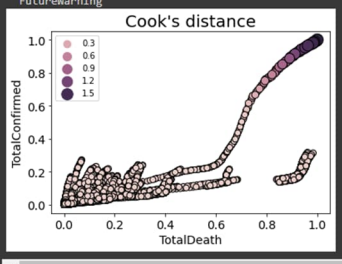
Εικόνα 54: *Οπτική ανίχνευση outliers συντεταγμένων με την χρήση statsmodels και seaborn βιβλιοθηκών*

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.outliers_influence import OLSInfluence

f = 'TotalDeath ~ TotalConfirmed'
model = ols(formula=f, data=df1_out_norm).fit()
# calculate the cooks_distance - the OLSInfluence object contains multiple influence measurements
cook_distance = OLSInfluence(model).cooks_distance
(distance, p_value) = cook_distanc
# scatter plot - x axis (independent variable height), y-axis (dependent variable weight)
sns.scatterplot(df1_out_norm.TotalDeath , df1_out_norm.TotalConfirmed, hue=distance, size=distance, sizes=(50, 200), edgecolor='black', linewidth=1)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

plt.xlabel('TotalDeath', fontsize=14)
plt.ylabel('TotalConfirmed', fontsize=14)
plt.title('Cook\'s distance', fontsize=20);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12,
  FutureWarning
```



Εικόνα 55: *Οπτική ανίχνευση outliers αριθμητικών στηλών ('TotalDeath' και 'TotalConfrimed) με την χρήση statsmodels και seaborn βιβλιοθηκών*

6.6.2 PyOD

Η βιβλιοθήκη PyOD χρησιμοποιείται για την ανίχνευση των έκτοπων τιμών του dataframe covid_df. Στην υλοποίηση χρησιμοποιείται το σετ δεδομένων αφού έχει δεχθεί filtering από το βήμα 4. Με την pyod εισάγουμε τον αλγόριθμο K-Nearest Neighbor για να κάνει την ανίχνευση των έκτοπων τιμών. Παρατηρείται ότι κατά την πρώτη υλοποίηση βρέθηκαν 13.013 outliers που είναι αρκετά μεγάλος αριθμός, σε σχέση με τον αρχικό αριθμό πλειάδων που ήταν 130.124. Στην δεύτερη υλοποίηση εφαρμόζεται στην KNN() συνάρτηση η παράμετρος 'contamination' για να ορισθεί το ποσοστό των outliers που μπορεί να βρει σε σχέση με το μέγεθος του σετ δεδομένων, και παρατηρείται από τα αποτελέσματα ότι μειώνονται οι πλειάδες των outliers

```
# Import the KNN

from pyod.models.knn import KNN
start=time.time()
KNN_model = KNN()
KNN_model.fit(df1_out)
outlier_labels = KNN_model.labels_
number_of_outliers = len(outlier_labels[outlier_labels == 1])
#print(number_of_outliers)
outliers = df1_out.iloc[outlier_labels == 1]
end=time.time()
outliers
```

	Total Confirmed	Total Death	latitude	longitude
816	41010	1965	40.8832	-72.8012
1142	16277	1345	41.2681	-73.3881
1176	10069	712	45.0076	-93.4769
1490	65680	7012	40.7109	-73.8168
1534	9249	601	47.4914	-121.8346
...	...	...	...	...
3042308	23306	1022	42.3280	-71.0785
3042341	25067	511	39.9700	-83.0112
3042445	67479	7311	40.6362	-73.9494
3042505	136993	5147	41.8414	-87.8166
3042707	66046	7290	40.6362	-73.9494

13013 rows × 4 columns

```
arrow.get(end)-arrow.get(start)
```

```
datetime.timedelta(microseconds=983209)
```

Εικόνα 57: *Εφαρμογή KNN αλγόριθμου για την εύρεση έκτοπων τιμών*

```
KNN_model = KNN(contamination=0.01)
KNN_model.fit(df1_out)
outlier_labels = KNN_model.labels_
number_of_outliers = len(outlier_labels[outlier_labels == 1])
print(number_of_outliers)
outliers = df1_out.iloc[outlier_labels == 1]
outliers
```

```
1302
```

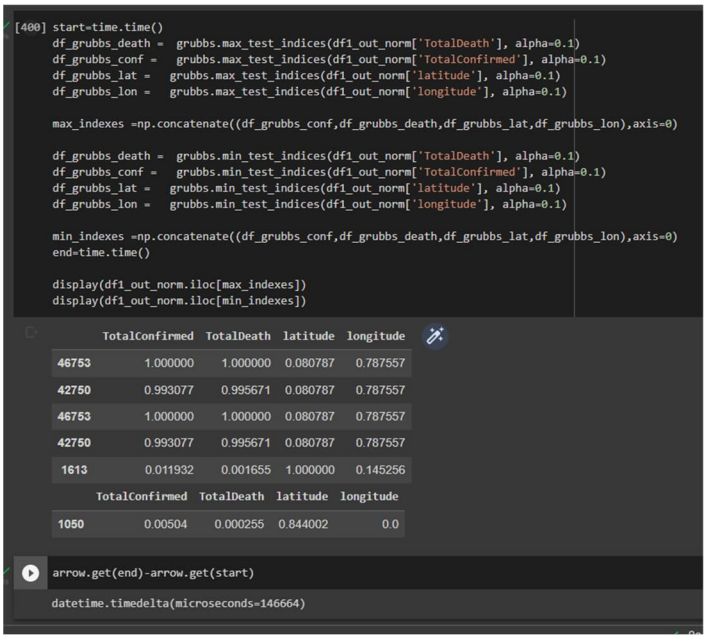
	Total Confirmed	Total Death	latitude	longitude
10767	41990	4008	40.6362	-73.9494
11372	31463	2452	40.8521	-73.8628
14894	32050	2337	40.7109	-73.8168
25649	61847	6711	40.7109	-73.8168
35222	101411	4822	41.8414	-87.8166
...	...	...	...	...
3031066	82427	3944	41.8414	-87.8166
3033949	62898	6798	40.7109	-73.8168
3034126	45988	4537	40.8521	-73.8628
3042183	133686	5112	41.8414	-87.8166
3042505	136993	5147	41.8414	-87.8166

1302 rows × 4 columns

Εικόνα 56*: Επανεκτέλεση με την χρήση contamination παραμέτρου*

6.6.3 Outlier-utils

Η βιβλιοθήκη Outlier-utils χρησιμοποιείται για να βρεθούν οι μέγιστες ή ελάχιστες έκτοπες τιμές που περιέχονται σε μία στήλη. Αυτό επιτυγχάνεται με τις συναρτήσεις min_test_indices() και max_test_outliers(). Στην υλοποίηση χρησιμοποιούνται και οι δύο συναρτήσεις μια φορά για κάθε στήλη. Οι συναρτήσεις αυτές έχουν ως έξοδο το index της dataframe στήλης που κλήθηκε. Οι τέσσερις έξοδοι ενώνονται για να σχηματίσουν ένα



Εικόνα 58: *Εφαρμογής βιβλιοθήκη Outlier-utils για την εύρεση έκτοπων τιμών ανά στήλη*

πίνακα από indexes με τιμές που θεωρεί μέγιστες έκτοπες τιμές, ενώ με παρόμοιο τρόπο πραγματοποιείται η εύρεση ελάχιστον έκτοπων τιμών κάθε στήλης.

6.6.4 **Σύγκριση**

Οι δυο πρώτες βιβλιοθήκες συνθέτουν το κομμάτι ανίχνευσης έκτοπων τιμών μέσω γραφημάτων, και σύμφωνα με τα αποτελέσματα που επέστρεψαν θεωρούνται πολύ χρήσιμες και εύχρηστες για αυτόν τον σκοπό. Η pyod με την χρήση του αλγορίθμου KNN έκανε εξαιρετική δουλειά από την στιγμή που εφαρμόστηκε και η παράμετρος ‘contamination’, με αξιόλογο χρόνο εκτέλεσης για την εργασία που πραγματοποίησε και με ικανοποιητικό αριθμό γραμμών. Αντιθέτως, αν και η βιβλιοθήκη outliers_utils «έτρεξε» πιο

γρήγορα, είναι αρκετά περίπλοκη ως προς την χρήση της με dataframes και δεν θα έπρεπε να προτιμηθεί σε για ανίχνευση έκτοπων τιμών σε πίνακα με πολλά αριθμητικά στοιχεία. Είναι όμως μια χρήσιμη βιβλιοθήκη, σε ένα σενάριο εύρεσης έκτοπων τιμών από μιας στήλης και εκτελείται πολύ γρήγορα.

***Πίνακας 8**: Detection of data outliers, σύγκριση των modules*

Detection of data outliers	*Γραμμές Κώδικα*	*Χρόνος Εκτέλεσης*	*Χρηστικότητα*
Statsmodels-Seaborn	X	X	Απεικόνιση πιθανών outliers
Pyod (KNN algorithm)	5	0.98 seconds	Εύρεση όλων των outliers
Outlier-utils (Grubb's approach)	10	0.14 seconds	Εύρεση index μέγιστου ή ελάχιστου outlier

Βιβλιογραφία

Acuña, E., & Rodriguez, C. (2004). The Treatment of Missing Values and its Effect on Classifier Accuracy. Στο D. Banks, F. McMorris, P. Arabie, & W. Gaul, *Classification, Clustering, and Data Mining Applications.* Springer, Berlin, Heidelberg.

Alexopoulos, E. C. (2010). Introduction to Multivariate Regression Analysis . *HIPPOKRATIA , 14 (Suppl 1):*, 23-28.

Allison, P. D. (2002). *Missing Data.* Thousand Oaks: SAGE Publications, Inc.

Amirteimoori, A., & Kordrostami, S. (2010). A Euclidean distance-based measure of efficiency in data envelopment analysis. *A Journal of Mathematical Programming and Operations Research*, 985-996.

Ananthakrishna, R., Chaudhuri, S., & Ganti, V. (2002). Eliminating fuzzy duplicates in data warehouses. *28th international conference on Very Large Data Bases* (σσ. 586–597). Hong Kong: VLDB Endowment.

Andridge, R. R., & Little, R. (2010). A Review of Hot Deck Imputation for Survey Non-response. *International Statistical Review, vol. 78*, 40-64.

Arrow. (χ.χ.). Ανάκτηση από Arrow: https://arrow.readthedocs.io/en/latest/

Babu, S. S., & Divya, D. (016). METHODS TO DETECT DIFFERENT TYPES OF OUTLIERS. *International Conference on Data Mining and Advanced Computing (SAPIENCE)*, (σσ. 23-24).

Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009, July). Methodologies for data quality assessment. *ACM Computing Surveys, Vol. 41, No. 3, Article 16*, σ. 52.

Beretta, L., & Santaniello, A. (2016). Nearest neighbor imputation algorithms: a critical evaluation. *BMC Medical Informatics and Decision Making volume*.

Bilenko, M., Mooney, R. J., Cohen, W. W., & Fienberg, S. E. (2003). Adaptive name matching in information integration. *IEEE Intelligent Systems, Volume: 18, Issue: 5,* , 16 - 23.

Brown, M. L., & Kros, J. F. (2003). Data mining and the impact of missing data. Στο *Industrial Management & Data Systems* (σσ. 611-621). Emerald Group Publishing.

Cheema, J. R. (2014). A Review of Missing Data Handling Methods in Education Research.

Chu, X. (2019). Data Cleaning. Στο S. Sakr, & A. Zomaya, *Encyclopedia of Big Data Technologies.* Springer.

Cochinwala, M., Kurienb, V., GailLalka, & DennisShashac. (2001). Efficient data reconciliation. *Information Sciences,Volume 137, Issues 1–4,*, 1-15.

Cukier, K. (2010). Data, Data Everywhere: A Special Report on Managing Information. *The Economist, 394*, 3-5.

Datacleaner. (χ.χ.). Ανάκτηση από GitHub: https://github.com/rhiever/datacleaner

Dataprep. (χ.χ.). Ανάκτηση από Dataprep: https://docs.dataprep.ai/user_guide/user_guide.html

deepblade. (2021, December 17). *Outlier Detection using Z-Score*. Ανάκτηση από deepblade.com: https://deepblade.com/outlier-detection-using-z-score/

Dora. (χ.χ.). Ανάκτηση από GitHub: https://github.com/NathanEpstein/Dora

Elmagarmid, A. K., Verykios, V. S., & Ipeirotis, P. G. (2007, January 30). Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering (Volume: 19, Issue: 1)*, σσ. 1-16.

English, L. P. (1999). *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits.* Wiley.

English, L. P. (2009). *Information Quality Applied: Best Practices for Improving Business Information, Processes and Systems.* Wiley.

Fellegi, I., & Sunter, A. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association Volume 64*, 1183-1210 .

Fisher, C. (2011). *Introduction to Information Quality.* AuthorHouse.

Furht, B., & Vilanustre, F. (2016). Big Data Technologies and Applications. Στο B. Furht, & F. Villanustre, *Big Data Technologies and Applications* (σσ. 3-11). Cham: Springer International Publishing.

fuzzywuzzy-python-library. (χ.χ.). Ανάκτηση από GeeksforGeeks: https://www.geeksforgeeks.org/fuzzywuzzy-python-library/

Gandomi, A., & Haider, M. (2014). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 138-139.

GARCÍA-LAENCINA, P. J., SANCHO-GÓMEZ, J.-L., & FIGUEIRAS-VIDAL, A. R. (2010). Classification with missing data. Στο *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques.* Madrid.

Grubbs, F. E. (1969, February). Procedures for Detecting Outlying Observations in Samples. *Technometrics, Vol. 11*, σσ. 1-21.

Hawkins, D. M. (1980). *Identification of Outliers.* Chapman and Hall.

JO, K., & J., C. (1977). The treatment of missing data in multivariate analysis. *Sociological methods & research*, 215-239.

Knorr, E. M., & Ng, R. T. (1998). Algorithms for Mining Distance-Based in Large Datasets. *24th VLDB Conference* , (σσ. 392-403). New York.

Lin, W. C., & Tsai, C. F. (2020). Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, 1487-1509.

Little, R., & Rubin, D. (2019). *Statistical Analysis with Missing Data.* Wiley.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining.* Pisa, Italy: IEEE.

Loshin, D. (2010). *The Practitioner's Guide to Data Quality Improvement.* Morgan Kaufmann.

Mahanti, R. (2019). Data Quality Dimensions. Στο R. Mahanti, *Data Quality: Dimensions, Measurement, Strategy, Management, and Governance.* american society for quality.

McGilvray, D. (2021). *Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information.* Academic Press.

Mishra, P. (2021, June 12). *5 Outlier Detection Techniques that every "Data Enthusiast" Must Know*. Ανάκτηση από Towards Data Science: https://towardsdatascience.com/5-outlier-detection-methods-that-every-data-enthusiast-must-know-f917bf439210

Mishra, P. (2022, 06). *outlier-detection-with-abod*. Ανάκτηση από paperspace.com: https://blog.paperspace.com/outlier-detection-with-abod/

missingno. (χ.χ.). Ανάκτηση από GitHub: https://github.com/ResidentMario/missingno/blob/master/paper.md

Moore, S. (2018, June 19). Ανάκτηση από www.gartner.com: https://www.gartner.com/smarterwithgartner/how-to-create-a-business-case-for-data-quality-improvement

Olson, J. E. (January 2003). *Data Quality: The accuracy Dimensions .* Morgan Kaufmann.

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2017). Big data technologies : A survey. *Journal of King Saud University - Computer and Information Science*, 433-434.

outlier-utils. (χ.χ.). Ανάκτηση από GtiHub: https://github.com/c-bata/outlier-utils

Punn, N. S., Agarwal, S., Adiyarta, K., & Syafrullah, M. (2019). Testing Big Data Application. *EECSI.* Bandug-Indonesia: IEEE.

pyod. (χ.χ.). Ανάκτηση από pyod: https://pyod.readthedocs.io/en/latest/

Redman, T. (1996). *Data Quality for the Information Age.* Artech House Publishers.

Redman, T. (2001). *Data Quality: The Field Guide.* Digital Press.

Redman, T. C. (2016, September 22). Ανάκτηση από Harvard Business Review: https://hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year

Rubin, D. B. (1978). MPUTATION OF MISSING DATA. *Survey Research Methods Section, American Statistical Association* (σσ. 20-28). ASA.

Sarawagi, S., & Bhamidipaty, A. (2002). Interactive deduplication using active learning. *ighth ACM SIGKDD international conference on Knowledge discovery and data mining* (σσ. 269–278). Edmonton Alberta Canada: Association for Computing Machinery.

Scannapieco, M., Missier, P., & Batin, C. (2005, January). Data Quality at a Glance. *Datenbank-Spektrum 14*, σσ. 6-14.

scrubadub. (χ.χ.). Ανάκτηση από scrubadub: https://scrubadub.readthedocs.io/en/stable/index.html

Seaborn. (χ.χ.). Ανάκτηση από Seaborn: https://seaborn.pydata.org/index.html

Sebastian-Coleman, L. (2013). *Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework.* Morgan Kaufmann.

Sidi, F., Panahy, P. H., Affendey, L. S., Jabar, M. A., Ibrahim, H., & Mustapha, A. (2012). Data quality: A survey of data quality dimensions. *International Conference on Information Retrieval & Knowledge Management* (σ. 300). Kuala Lumpur, Malaysia: IEEE.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2006). *Database System Concepts.* New York: McGraw-Hill Companies.

Soley-Bori, M. (2013). *Dealing with missing data: key assumptions and methods for applied analysis.* Boston.

Song, Q., & Shepperd, M. J. (2007). Missing Data Imputation Techniques. *nternational Journal of Business Intelligence and Data Mining*, 261-291.

statsmodels. (χ.χ.). Ανάκτηση από statsmodels: https://www.statsmodels.org/stable/index.html

Stefano, C. D., Sansone, C., & Vento, M. (2000, February). 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) (Volume: 30, Issue: 1)*, σσ. 84-94.

Stewart, T. G., Zeng, D., & Wu, M. C. (2018). Constructing support vector machines with missing data.

Sunagar, P., Hanumantharaju, R., Siddesh, G. M., Kanavalli, A., & Srinivasa, K. G. (2020). Chapter 2 - Influence of big data in smart tourism. Στο *Hybrid Computational Intelligence* (σσ. 25-47). Academic Press.

Tress, F. (2017, February 2). *What is Data Quality?* Ανάκτηση από norstat: https://norstat.de/blog/what-is-data-quality-part-2-2

Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence, 23(5)*, 373–405.

Wang, R. Y., & Strong, D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *: Journal of Management Information Systems, Vol. 12, No. 4*, 5-33.

Wang, R. Y., & Wand, Y. (1996, November). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, σσ. 86-95.

Wang, R. Y., Storey, V. C., & Firth, C. P. (1995). A framework for analysis of quality research. *IEEE Transactions on Knowledge and Data Engineering* , 623 - 640.
Williams, R. (2021). *Missing Data Part 1: Overview, Traditional Methods.*