ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΕΡΓΑΣΤΗΡΙΑΚΩΝ ΕΦΑΡΜΟΓΩΝ ΣΕ ΡΥΤΗΟΝ ΓΙΑ ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ

ΖΟΥΛΟΣ ΑΛΚΙΒΙΑΔΗΣ ΝΙΚΟΛΑΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:ΔΡΟΣΟΣ ΧΡΗΣΤΟΣ

ΙΟΥΛΙΟΣ 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ (1η ή 2η σελίδα)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η Αλκιβιάδης-Νικόλαος Ζούλας, του Ανδρέα, φοιτητής του Τμήματος Σχολής Μηχανικών Μεταλλείων-Μεταλλουργών του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (ΠΕ) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε, ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα, σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση Π.Ε με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της ενλόγω Π.Ε πρέπει να ολοκληρωθεί εντός του λάχιστον ενός ημερολογιακού έτους από την ημερομηνία ανάθεσής της.

Ο Δηλών

Ημερομηνία

21/7/2020

Περίληψη:

Με την τεράστια ανάπτυξη της τεχνολογίας και την όλο και μεγαλύτερη διάδοση του παγκοσμίου ιστού και των Social media έχει δημιουργηθεί η ανάγκη επεξεργασίας αλλά και ανάλυσης όλο και μεγαλύτερου όγκου δεδομένων και πληροφοριών τα οποία έχουν ονομαστεί BIG DATA .Ο όγκος και ο ρυθμός με τον οποίο παράγονται αυτά με την σειρά τους δημιουργεί την ανάγκη για ανάπτυξη καινούργιων πιο αποτελεσματικών τεχνικών και μεθόδων επεξεργασίας τους ώστε να μπορούν να αξιοποιηθούν καλυτέρα.

Ένα από τα αποτελεσματικότερα εργαλεία για την επεξεργασία των μεγάλων δεδομένων είναι η γλώσσα προγραμματισμού Python .

Ο προγραμματισμός των σύγχρονων συστημάτων μηχανικής μάθησης γίνεται συνήθως με Python. Ακόμη και όταν η Python δεν είναι η κύρια γλώσσα προγραμματισμού ενός τέτοιου συστήματος.

Η Python είναι επίσης η γλώσσα που χρησιμοποιούν πλέον οι περισσότεροι data scientists για desktop computing λόγω των εκτεταμένων βιβλιοθηκών έτοιμων αλγορίθμων μηχανικής μάθησης και επεξεργασίας δεδομένων διαφορετικών μορφών και μεγεθών.

Στόχος αυτής της εργασίας είναι η δημιουργία ασκήσεων και εφαρμογών που θα αξιοποιούνται σε εργαστηριακό περιβάλλον για την εκμάθηση των απαραίτητων βιβλιοθηκών και εντολών της γλώσσας προγραμματισμού Python με σκοπό την επεξεργασία των μεγάλων αυτών δεδομένων.

Ορισμοί:

Internet of Things: ΤοΔιαδίκτυο των πραγμάτων ή Ίντερνετ των πραγμάτων (αγγλικά: Internet of things) αποτελεί το δίκτυο επικοινωνίας πληθώρας συσκευών, οικιακών συσκευών, αυτοκινήτων καθώς και κάθε αντικειμένου που ενσωματώνει ηλεκτρονικά μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα σε δίκτυο ώστε να επιτρέπεται η σύνδεση και η ανταλλαγή δεδομένων. Απλούστερα, η φιλοσοφία του IoT είναι η σύνδεση όλων των ηλεκτρονικών συσκευών μεταξύ τους (τοπικό δίκτυο) ή με δυνατότητα σύνδεσης στο διαδίκτυο (παγκόσμιο ιστό).(30)

Entity Relationship Model: Το μοντέλο οντοτήτων-συσχετίσεων (*μοντέλο Ο/Σ - ER model*) είναι ένα αφαιρετικό ιδεατό μοντέλο δεδομένων, τα οποία έχουν καθορισμένη δομή. Στη μηχανική λογισμικού χρησιμοποιείται για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίασηβάσεων δεδομένων.(31)

SQL:(Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων ,που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών

βάσεων δεδομένων (Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα.(32)

Data Warehouse: Στην επιστήμη των υπολογιστών, μια αποθήκη δεδομένων (DW) είναι μια βάση δεδομένων που χρησιμοποιείται για την αναφορά και ανάλυση. Τα δεδομένα που είναι αποθηκευμένα στην αποθήκη φορτώνονται από το λειτουργικό σύστημα. Τα δεδομένα μπορεί να περάσουν μέσα από ένα λειτουργικό χώρο αποθήκευσης δεδομένων για τις πρόσθετες εργασίες πριν χρησιμοποιηθούν στις αποθήκες δεδομένων για την υποβολή εκθέσεων.(33)

Cloud Computing: Ορίζεται ως το βασιζόμενο στο διαδίκτυο computing μοντέλο που προσφέρει άμεσα(on-demand) πρόσβαση σε πηγές και εργαλεία. Αυτά μπορεί να είναι πολλά πράγματα, όπως εφαρμογές λογισμικού, servers και κέντρα δεδομένα κλπ. Συνήθως το κόστος για τέτοιες υπηρεσίες είναι ανάλογο της χρήσης, πράγμα που βοηθάει τις επιχειρήσεις να υπολογίσουν το κόστος για τις δικές τους ανάγκες. Επίσης, να ξεπεράσουν το κόστος της εγκατάστασης δικών της υποδομών, κάτι που πριν το cloud computing ήταν αναπόφευκτο.(34)

Machine Learning: Ασχολείται με το ερώτημα του πώς να κατασκευαστούν προγράμματα που βελτιώνονται αυτόματα με την εμπειρία. Εμπεριέχει τεχνικές από τη πληροφορική, τη στατιστική και την τεχνητή νοημοσύνη, ανάμεσα σε άλλα. Το κύριο χαρακτηριστικό της είναι οι αλγόριθμοι που εφαρμόζουν αυτή την αυτόματη βελτίωση μέσω της εμπειρία, με εφαρμογές σε πολλά διαφορετικά πεδία, όπως την εξόρυξη δεδομένων, την τεχνητή νοημοσύνη κα.(34)

Περιεχόμενα

BIG DATA.

Ο όρος μεγάλα δεδομένα χρησιμοποιείται από το 1990 και αποδίδεται στον John R.Mashley. Αρχικά, ο όρος αυτός είχε την έννοια των πολύ μεγάλων συνόλων, δομημένων ή αδόμητων δεδομένων, όπως αυτά παράγονται εντός του Παγκόσμιου Ιστού από ιστοσελίδες, ηλεκτρονικό ταχυδρομείο και μέσα κοινωνικής δικτύωσης[10].Γενικότερα ως big data ορίζουμε τα δεδομένα των οποίων το μέγεθος γίνεται μέρος του προβλήματος δηλαδή ως Big Data (Μεγάλα Δεδομένα) νοούνται «σύνολα δεδομένων τα οποία δεν μπορεί να αντιληφθεί και να διαχειριστεί η παραδοσιακή πληροφορική και τα εργαλεία λογισμικού μέσα σε ένα ανεκτό επίπεδο χρόνου» [1]

Ιστορική αναδρομή.

Προς τα τέλη της δεκαετίας του 1960, όταν οι υπολογιστές εμφανίστηκαν για πρώτη φορά στον εμπορικό κόσμο οι επιχειρήσεις, χρησιμοποιούσαν εξειδικευμένα προγράμματα υπολογιστή ,προκειμένου να αξιοποιήσουν τα δεδομένα τους , αντλώντας με δυσκολία κάποια αξία από τα στοιχεία τους.

Την δεκαετία του 1970 η κατάσταση βελτιώθηκε , με τη χρήση συστημάτων διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), η εξαγωγή αξιοποιήσιμων πληροφοριών έγινε πολύ ευκολότερη για τους προγραμματιστές με τη χρήση δομημένων γλωσσών προγραμματισμού όπως η SQL (Structured Query Language) όπως και με άλλα εργαλεία διαχείρισης.

Η ευκολία με την οποία γινόταν πια η αναζήτηση αλλά και η διαχείριση της πληροφορίας , είχε ως αποτέλεσμα τη δημιουργία ενός αυξανόμενου όγκου δεδομένων, του οποίου η επεξεργασία γινόταν ολοένα δυσκολότερη και η αποθήκευση συνεχώς πιο δαπανηρή . Ως λύση εμφανίσθηκε το μοντέλο Οντοτήτων -Συσχετίσεων (Entity-RelationshipModel, ER) που επέτρεψε τη δημιουργία σύνθετων μοντέλων συσχέτισης μεταξύ των πηγών δεδομένων, δίχως την ανάγκη περίπλοκου προγραμματισμού.

Ως λύση επίσης στο πρόβλημα του συνεχώς αυξανόμενου όγκου των δεδομένων , ήταν η δημιουργία των Αποθηκών Δεδομένων (Data Warehouses). Αυτός ο συνδυασμός υλικού και λογισμικού, επέτρεψε στις επιχειρήσεις την αποτελεσματική αποθήκευση και διαχείριση του τεράστιου όγκου δομημένων δεδομένων, δημιουργώντας μικρότερα και περισσότερο εξειδικευμένα σύνολα δεδομένων.

Πρόκειται για βάσεις δεδομένων οι οποίες είναι θεματικά προσανατολισμένες και οι οποίες επέτρεψαν στις επιχειρήσεις να διαχειριστούν και να αναλύσουν διαχρονικά μεγάλα

σύνολα από δεδομένα τα οποία προέρχονται από διαφορετικές πηγές, τόσο εσωτερικές όσο και εξωτερικές .Τελικά η προσπάθεια συσχετισμού αντικειμένων και βάσεων δεδομένων, οδήγησε στις αντικειμενοστραφείς βάσεις δεδομένων (Object Database Management System, ODBMS), βοηθώντας στην καλύτερη και απλούστερη διαχείριση τόσο των δομημένων όσο και των αδόμητων δεδομένων[2].

Παράλληλα με την ανάπτυξη της τεχνολογίας αποθήκευσης και διαχείρισης εταιρικών δεδομένων, εμφανίστηκε το 1977 ο προσωπικός υπολογιστής, ο οποίος έδωσε τη δυνατότητα πρόσβασης στο Internet, σε εκατομμύρια ανθρώπους σε όλο τον κόσμο. Η πρόσβαση στο Internet, έδωσε με τη σειρά της τη δυνατότητα πρόσβασης στο World Wide Web και μέσω αυτού σε αντικείμενα κειμένου ,εικόνων, ήχου και βίντεο συνδεδεμένα ως περιεχόμενο σε ιστοσελίδες. Ακολούθησε μια θεαματική τεχνολογική ανάπτυξη σε διάφορα συστήματα, δίνοντας τη δυνατότητα πρόσβασης στο Internet σε συσκευές και τεχνολογικά συστήματα, τα οποία μεταδίδουν πλέον έναν τεράστιο αριθμό δεδομένων μέσω των αισθητήρων τους, δημιουργώντας έτσι το λεγόμενο Internet of Things[3].

Αυτή η "έκρηξη" των δεδομένων, είτε αυτά είναι δομημένα, είτε είναι μη δομημένα ή ημιδομημένα, άνοιξε νέους δρόμους και φανέρωσε νέες δυνατότητες για δημιουργία αξιοποιήσιμης γνώσης μέσα από την ανάλυση και επεξεργασία τους, οδηγώντας σε αυτό που σήμερα είναι γνωστό ως Μεγάλα Δεδομένα (BigData).[10]

Είδη και πηγές Μεγάλων Δεδομένων.

Περίπου το 80% των δεδομένων παγκοσμίως δεν έχουν συγκεκριμένη δομή. Συνήθως εμφανίζονται ως κείμενα, εικόνες και βίντεο και συνεπώς οι συνηθισμένες μέθοδοι που χρησιμοποιούνται για τα δομημένα δεδομένα, δεν είναι κατάλληλες. Στο πρόβλημα λοιπόν του τεράστιου όγκου των διαθέσιμων δεδομένων, προστίθεται και το πρόβλημα της ιδιαίτερης μορφής τους. Αυτό οδήγησε στην ανάπτυξη νέων αλγορίθμων, ικανών να χειρίζονται δεδομένα που έχουν ποικίλες μορφές και προέρχονται από πολλές και διαφορετικές πηγές [2].

Με τον όρο δομημένα δεδομένα (structured data), συνήθως περιγράφουμε δεδομένα που είναι οργανωμένα σε καθορισμένη δομή, όπως είναι ένα υπολογιστικό φύλλο του MicrosoftExcel ή CSV αρχεία [4].

Τα δεδομένα αυτά μπορεί να παράγονται από υπολογιστές ή με άλλο μηχανικό μέσο, όπως τα δεδομένα αισθητήρων ταυτοποίησης μέσω ραδιοσυχνοτήτων τύπου RFID(Radio Frequency Identification), δεδομένα του Παγκόσμιου Συστήματος Θεσηθεσίας GPS(Global Positioning System), δεδομένα που παράγονται κατά την λειτουργία εξυπηρετητών

(servers) ή εφαρμογών λογισμικού καταγράφουν τη δραστηριότητά τους, δεδομένα που παράγονται στα σημεία πώλησης των επιχειρήσεων κλπ. Επίσης μπορεί να παράγονται από ανθρώπους οι οποίοι κάνοντας χρήση υπολογιστών εισάγουν δεδομένα σε βάσεις δεδομένων, επισκέπτονται ιστοσελίδες ή κάνουν αναζητήσεις οι οποίες καταγράφονται και αναλύονται, ενώ ακόμα και οι κινήσεις ενός παίκτη σε ένα ηλεκτρονικό παιχνίδι μπορεί να καταγραφούν και να αποτελέσουν αντικείμενο ανάλυσης και μελέτης [2].

Ως μη δομημένα (unstructured), χαρακτηρίζονται τα δεδομένα που στερούνται προκαθορισμένης δομής και συγκεκριμένης μορφής, όπως για παράδειγμα ο προφορικός λόγος, η μουσική, οι ταινίες ,μηνύματα στο Twitter ή στο Facebook, φωτογραφίες στο Instagram κλπ [4].

Τα δεδομένα αυτά όπως και στην κατηγορία των δομημένων, μπορεί να δημιουργούνται από υπολογιστές ή άλλα μηχανικά μέσα όπως είναι οι δορυφόροι, επιστημονικά όργανα και μηχανήματα, φωτογραφικές μηχανές, ραντάρ κλπ. Στην περίπτωση που δημιουργούνται από ανθρώπους, αυτά μπορεί να αποτελούν δεδομένα των κοινωνικών μέσων δικτύωσης, κείμενα, δεδομένα που δημιουργούνται σε φορητές συσκευές, περιεχόμενο ιστοσελίδων κλπ [2].

Τα μεγάλα δεδομένα, έχουν χαρακτηριστικά τα οποία μπορεί να περιγραφούν με επτά V:

Volume (Όγκος), Velocity (Ταχύτητα), Variety (Ποικιλία), Veracity(Ποιότητα),Value(Αξία),Visualization(Οπτικοποίηση) και Variability(Μεταβλητότητα).

1. Big Data Όγκος (Volume).

Ο όγκος αναφέρεται στην ποσότητα δεδομένων που παράγονται και αποθηκεύονται. Το μέγεθος των δεδομένων ορίζει την αξία και την πιθανή γνώση που μπορούν να δώσουν, αλλά ορίζει και αν πρόκειται για big data ή όχι.[5]

2. Big Data Ταχύτητα (Velocity).

Είναι η ταχύτητα με την οποία παράγονται και επεξεργάζονται τα δεδομένα για να ανταπεξέλθουν στις απαιτήσεις και προκλήσεις που σχετίζονται με τη χρήση τους και την αξιοποίησή τους.[5]

3. Big Data Ποικιλία (Variety).

Αναφέρεται στον τύπο και στην φύση των δεδομένων. Βοηθάει τους αναλυτές να χρησιμοποιήσουν επαρκώς την γνώση που αντλούν από αυτά.[5]

4.Big Data Ποιότητα (Veracity).

Αφορά την ποιότητα των δεδομένων. Είναι κάτι που ποικίλει πολύ ανάλογα με την πηγή των δεδομένων και επηρεάζει πολύ την ανάλυση που ακολουθεί.[5]

5.Big Data Αξία (Value).

Θεωρείται από τα βασικότερα χαρακτηριστικά, ιδιαίτερα από την οπτική γωνία μιας επιχείρησης. Αναφέρεται στην γνώση που μπορεί κανείς να αντλήσει από την επεξεργασία, ανάλυση και άλλες τεχνικές των δεδομένων.[5]

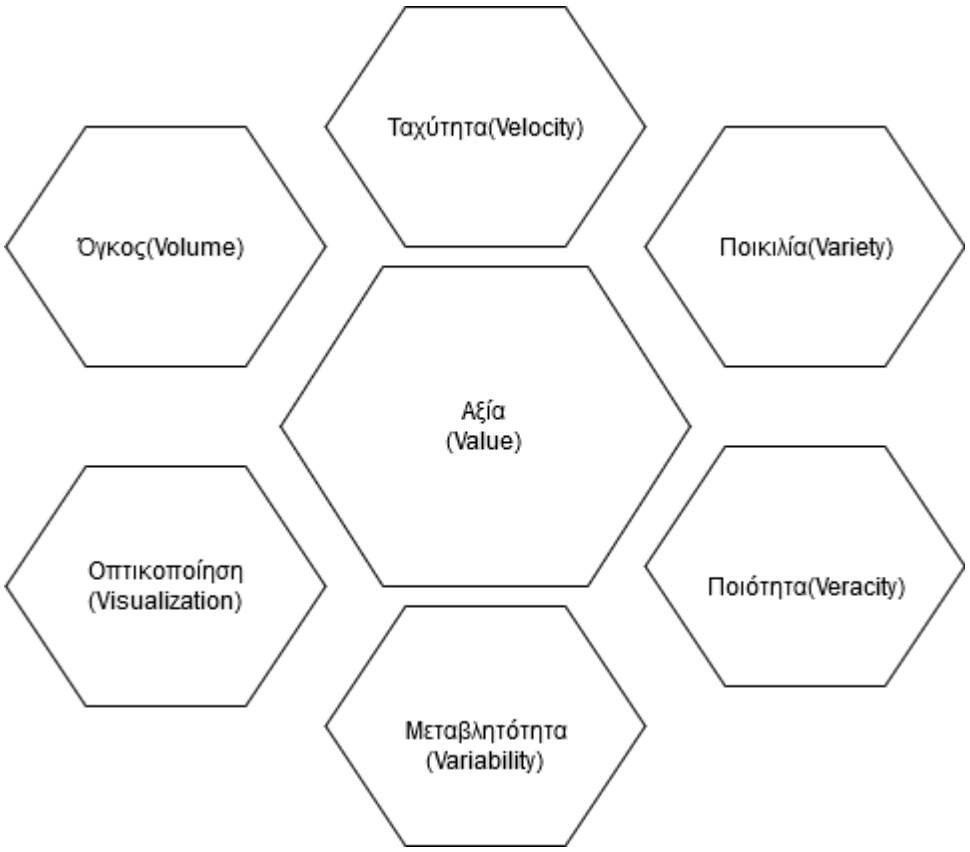
6.Big Data Οπτικοποίηση(Visualization).

Εδώ ο όρος έχει την έννοια της μετατροπής του χαοτικού όγκου των δεδομένων, σε μορφές απεικόνισης που να μπορούν να αναγνωρισθούν και να αναλυθούν, ώστε να εξαχθούν χρήσιμα συμπεράσματα.[6]

7.Big Data Μεταβλητότητα(Variability).

Τα δεδομένα μεταβάλλονται και αλλάζουν μορφές συνεχώς, απαιτώντας αντίστοιχες ενέργειες προσαρμογής και ορθής ερμηνείας τους. Ο αποτελεσματικός χειρισμός αυτήστης μεταβλητότητας, αποτελεί μια από τις

μεγαλύτερες προκλήσεις που αντιμετωπίζουν τα λογισμικά ανάλυσης των μεγάλων δεδομένων.[6]



Η τεράστια ανάπτυξη στον όγκο των δεδομένων έχει δημιουργήσει πολύ μεγάλες ανάγκες, όχι μόνο για την αποθήκευσή τους, αλλά και για την επεξεργασία τους. Στόχος της ανάλυσης των δεδομένων είναι η μετατροπή τους σε πολύτιμες πληροφορίες ή αλλιώς γνώση για τις επιχειρήσεις και τους οργανισμούς. Η ανάγκη αυτή οδήγησε στην δημιουργίας μία τεράστιας ποικιλίας από τεχνικές και εργαλεία ανάλυσης δεδομένων. Τα περισσότερα από αυτά έχουν τις βάσεις τους σε επιστήμες όπως τα μαθηματικά, η στατιστική, η οικονομία και η πληροφορική .Επιγραμματικά αυτές είναι :

- Βελτιστοποίηση (optimization).
- Στατιστική.
- Εξόρυξη δεδομένων(data mining).
- Τεχνικές οπτικοποίησης.
- Ανάλυση δικτύων(network analysis).
- Σημασιολογική ανάλυση(semantic analysis).
- Πληθοπορισμός(crowdsourcing).
- Μηχανική μάθηση(machine learning).

Εφαρμογές BigData.

Με την συσσώρευση αυτού του τεράστιου όγκου δεδομένων, διάφοροι αναλυτές αντιλήφθηκαν ότι τα Μεγάλα Δεδομένα είναι πολύ σημαντικά, για οποιονδήποτε έχει τα μέσα και την τεχνογνωσία να τα επεξεργαστεί. Απόδειξη τα δισεκατομμύρια δολάρια που δαπανούνται από κολοσσούς της πληροφορικής όπως η Microsoft, ηIBM, η Oracle και η SAP, ετησίως για την ανάπτυξη λογισμικού ανάλυσης και επεξεργασίας δεδομένων. Το 2010 μόνο, η αξία του συγκεκριμένου κλάδου παραγωγής λογισμικού ήταν περισσότερο από 100 δις. δολάρια, με ετήσιους ρυθμούς ανάπτυξης της τάξης του 10%[7].

Αυτό που κάνει τα Μεγάλα Δεδομένα τόσο σημαντικά είναι το γεγονός ότι παρέχουν απαντήσεις σε ερωτήματα που οι ερευνητές δεν γνώριζαν καν ότι υπάρχουν. Σημείο κλειδί βέβαια είναι η διεξοδική και έγκαιρη ανάλυση του κατάλληλου συνόλου δεδομένων ανάμεσα στο χάος της διαθέσιμης πληροφορίας, ώστε να δίνεται η δυνατότητα απόκτησης αξιόπιστης και αξιοποιήσιμης γνώσης, μέσα από αναλύσεις καταγραφής και πρόγνωσης. Σύμφωνα με μελέτη του MIT, οι επιχειρήσεις που χρησιμοποιούν τεχνικές ανάλυσης δεδομένων ,είναι 5% έως 6% περισσότερο παραγωγικές από τους ανταγωνιστές τους [8].

BigData και υπηρεσίες Υγείας.

Τα BigData χρησιμοποιούνται για την διαχείριση του μεγάλου όγκου των ιατρικών δεδομένων των ασθενών, της χορήγησης φαρμάκων κλπ. Η ικανότητα των BigData να συνδυάζουν μεγάλο όγκο δομημένων και αδόμητων δεδομένων είναι καταλυτική για τις διαγνώσεις και την περίθαλψη.

Η διαχείριση και ο έλεγχος των πληροφοριών στην υγεία και την ιατρική περίθαλψη (σε συνθήκες αυξημένης πολυπλοκότητας) έχουν μείζονα σημασία, δεδομένου ότι από αυτές προσδιορίζεται η (ιατρική) επίδοση και η (οικονομική) απόδοση του τομέα της υγειονομικής φροντίδας. Σχετικές αναφορές καταγράφουν ότι στις Ηνωμένες Πολιτείες το 2011, το σύστημα υγείας είχε «πληροφοριακό μέγεθος» 150 exabytes, το οποίο σύντομα πρόκειται να προσεγγίσει την τάξη zetabyte (10^21 gigabytes).(26)

Στη χώρα μας, από τα διαθέσιμα στοιχεία διαπιστώνεται ότι ένας σημαντικός αριθμός οργανισμών και φορέων –κυρίως του δημοσίου τομέα– εμπλέκεται στη συλλογή και την επεξεργασία ιατρικών και νοσηλευτικών δεδομένων.

Τα δεδομένα αυτά αναφέρονται κυρίως: (α) στη συνταγογράφηση των φαρμακευτικών σκευασμάτων και των παρακλινικών βιολογικών και απεικονιστικών εξετάσεων, (β) στη

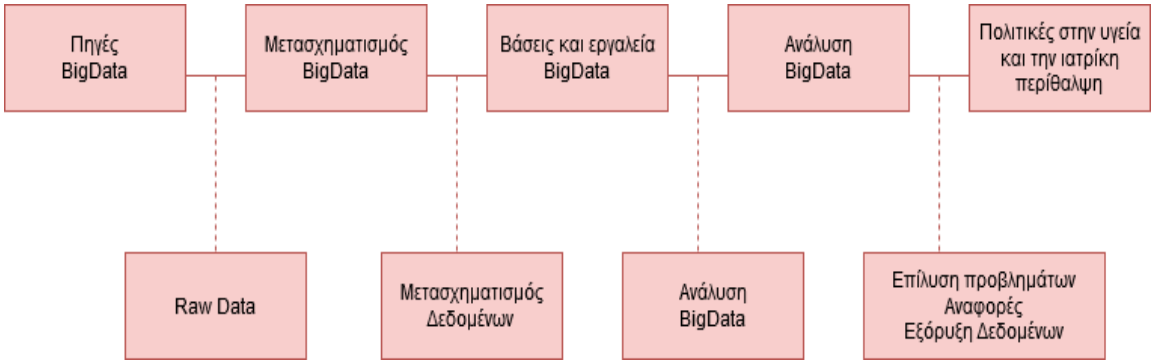
χρήση αγαθών και υπηρεσιών υγείας (φαρμακευτικά και τεχνολογικά προϊόντα, επισκέψεις στην πρωτοβάθμια φροντίδα, εισαγωγή στα νοσηλευτικά ιδρύματα), (γ) στην καταγραφή της κατανάλωσης φαρμάκων και ιατροτεχνολογικών προϊόντων, (δ) στη συλλογή δεδομένων από ιατρικά και νοσηλευτικά δεδομένα που συνδέονται με τη νοσηρότητα, (ε) στην τήρηση αρχείων από οικονομικά και διαχειριστικά δεδομένα των μονάδων υγείας και των προϋπολογιστικών επιλογών και (στ) στο μεγάλο αριθμό άλλων δεδομένων, τα οποία δεν δύνανται να ταξινομηθούν ή ταξινομούνται δυνητικά σε πολλές μικρότερου βάρους κατηγορίες.

Με κύριες πηγές ιατρικών δεδομένων τις εξής.

- Συστήματα κλινικών πληροφοριών.
- Δεδομένα ασφαλιστικών οργανισμών.
- Δημογραφικές και επιδημιολογικές εγγραφές.
- Δεδομένα βιοϊατρικών μετρήσεων.
- Δεδομένα ιατρικών συσκευών και φαρμάκων.
- Χρηματοδοτικές και οικονομικές συναλλαγές.
- Αναζητήσεις διαδικτύου.
- Βάσεις γενετικών δεδομένων.[18]

Οι δυνατότητες των ερευνητών διευκολύνονται σε μεγάλο βαθμό από την ελεύθερη πρόσβαση επιδημιολογικών, διαχειριστικών και κλινικών δεδομένων του υγειονομικού τομέα, γεγονός που αναμένεται να συμβάλλει στην αύξηση του όγκου των δεδομένων και τη βελτίωση της ποιότητας της επιστημονικής έρευνας, αλλά και στην επιστημονική εμβέλεια των ιδρυμάτων και της ερευνητικής και ακαδημαϊκής κοινότητας.

Μάλιστα, η συμβολή στη μεταφορά των αποτελεσμάτων της έρευνας στην άσκηση κλινικής πρακτικής και στην εγκαθίδρυση μεθόδων τεκμηριωμένης Ιατρικής (evidence-basedmedicine), είναι επίσης μια σημαντική συμβολή στην προστιθέμενη αξία της Ιατρικής περίθαλψης.(25)



BigData και επιχειρήσεις.

Ένας από τους τομείς που έχει ωφεληθεί από την ανάπτυξη των BigData είναι ο επιχειρησιακός τεράστιος όγκος πληροφοριών που λαμβάνουν καθημερινά οι επιχειρήσεις ,σχετικά με συνήθειες και τις τάσεις της αγοράς και η ανάλυση αυτών των πληροφοριών ,αποτελούν ένα μεγάλο κομμάτι της δημιουργίας ενός καινούργιου πιο αποτελεσματικού και ανταγωνιστικού επιχειρησιακού μοντέλου.

Οι επιχειρήσεις ανέκαθεν βασίζονταν στις προβλέψεις. Χάρη στα big data οι marketer μπορούν να αντλήσουν πληροφορίες από πηγές δεδομένων, να επιταχύνουν τη δημιουργία αναφορών, να κάνουν προβλέψεις σε πραγματικό χρόνο και να πάρουν αποφάσεις βάσει πληροφόρησης.

Για παράδειγμα, οι marketers μπορούν πλέον να αντιληφθούν πως αλλάζει το αποτέλεσμα του τζίρου αυξάνοντας τον προϋπολογισμό και να εντοπίσουν τη δυναμική μεμονωμένων προσεγγίσεων marketing.

Τα μοντέλα πρόγνωσης βοηθούν τις επιχειρήσεις να προβλέψουν με ακρίβεια τα αποτελέσματα στις ενέργειές τους χάρη σε αλγορίθμους BigData οι οποίοι ανταποκρίνονται στις δυναμικές αλλαγές και στην ομαδοποίηση των πληροφοριών.(27)

BigData και Τραπεζικό σύστημα.

Με τις μεγάλες ποσότητες πληροφοριών που ρέουν από αμέτρητες πηγές, οι τράπεζες αναζητούν την εξεύρεση νέων και καινοτόμων τρόπων για την διαχείριση των big data. Παρόλο που είναι σημαντικό να κατανοήσουν τις ανάγκες των πελατών και να αυξήσουν την ικανοποίησή τους, είναι εξίσου σημαντικό να ελαχιστοποιήσουν τους κινδύνους απάτης, διατηρώντας παράλληλα ένα ασφαλές περιβάλλον εντός ορίων.(20)

Στο χρηματοπιστωτικό τομέα, πέρα από τις παραδοσιακές πηγές δεδομένων όπως τα δεδομένα που καταχωρούνται στα συστήματα, πρόσθετες πηγές δεδομένων μπορεί να είναι δεδομένα συναλλαγών, δεδομένα αρχείων log που παράγονται από τα συστήματα, η ηλεκτρονική αλληλογραφία, τα κοινωνικά δίκτυα, αισθητήρες, εξωτερικές πηγές δεδομένων, δεδομένα RFID και POS, κείμενα, εικόνες και βίντεο .(28)

BigData και Εκπαίδευση.

Τα big data έχουν φέρει σημαντικές αλλαγές σε διάφορες πτυχές της εκπαίδευσης.

Οι εκπαιδευτικοί που έχουν εξειδίκευση στον τομέα αυτό και έχουν τη διορατικότητα , μπορεί να έχουν σημαντική επίδραση στα εκπαιδευτικά συστήματα, στους φοιτητές και στα προγράμματα σπουδών. Με την ανάλυση των big data, μπορούν να εντοπίσουν τους μαθητές που βρίσκονται σε κίνδυνο, να σιγουρευτούν ότι οι μαθητές κάνουν επαρκή πρόοδο και μπορούν να εφαρμόσουν ένα καλύτερο σύστημα για την αξιολόγηση και την υποστήριξη των εκπαιδευτικών καθώς επίσης και των διευθυντών.(20)

Ειδικότερα ένας από τους τομείς που συνυπάρχουν όγκος ,ποικιλία και ταχύτητα στα δεδομένα είναι η τριτοβάθμια εκπαίδευση .Μεγάλες ποσότητες εκπαιδευτικών δεδομένων δημιουργούνται και συλλέγονται σε καθημερινή βάση από διάφορες πηγές και σε διάφορες μορφές στο εκπαιδευτικό οικοσύστημα.

Τα εκπαιδευτικά δεδομένα κυμαίνονται από αυτά που παράγονται από την χρήση και την αλληλεπίδραση των μαθητών με τα συστήματα διαχείρισης μάθησης(LMS) και τις πλατφόρμες , έως τις μαθησιακές δραστηριότητες και τις πληροφορίες μαθήματων που περιλαμβάνουν ένα πρόγραμμα σπουδών όπως μαθησιακούς στόχους ,εκπαιδευτικό υλικό και δραστηριότητες ,αποτελέσματα εξετάσεων και αξιολόγηση μαθήματων.

Η περιορισμένη εκμετάλλευση των εκπαιδευτικών Big data και το μέγεθος καθώς και ο τύπος αυτών των δεδομένων στο πλαίσιο της τριτοβάθμιας εκπαίδευσης υποδηλώνει την ανάγκη εφαρμογής ειδικών τεχνικών προκειμένου να "εξορυγχθούν" νέες ευεργετικές πληροφορίες που επί του παρόντος βρίσκονται "κρυμμένες" στα δεδομένα.(29)

BigData και Μέσα Κοινωνικής Δικτύωσης.

Τα Μέσα Κοινωνικής Δικτύωσης δεν χρησιμοποιούνται αποκλείστηκα για την σύνδεση ατόμων μεταξύ τους, πλέον έχουν γίνει μια αποτελεσματική πλατφόρμα ώστε οι επιχειρήσεις να "πλησιάζουν" το κατάλληλο για αυτούς κοινό.

Με την άνοδο των bigdata το social media marketing έχει ανέλθει επίσης.

Όλες οι φωτογραφίες ή αναρτήσεις που κοινοποιούν οι χρήστες των ΜΚΔ παρέχουν σημαντικές πληροφορίες στις επιχειρήσεις για τα δημογραφικά στοιχεία, τις αρέσκεις και μη των χρηστών.

Οι επιχειρήσεις χρησιμοποιούν αυτές τις πληροφορίες με διάφορους τρόπους με σκοπό την επιχειρηματική ανταγωνιστικότητα.

Κυρίως τα BigData χρησιμοποιούνται από τους marketeers για να δημιουργήσουν στρατηγικές για μελλοντικές social media καμπάνιες με γνώμονα τις πληροφορίες αυτές για τους πιθανούς πελάτες τους.

BigData και Κυβερνήσεις.

Η χρήση και αξιοποίηση των Μεγάλων Δεδομένων από τις κυβερνήσεις, μπορεί να έχει ως αποτέλεσμα την μείωση του κόστους, την προώθηση της καινοτομίας και την αύξηση της παραγωγικότητας. Δεδομένου ότι η κυβέρνηση εποπτεύει και διοικεί πολλούς τομείς, δυναται να έχουμε οφέλη στον τομέα της ασφάλειας και των πληροφοριών, στην αποτελεσματική αξιολόγηση των φαρμάκων, στην επιστημονική έρευνα, πρόγνωση του καιρού, στην αντιμετώπιση της φοροδιαφυγής και στον σωστό συντονισμό των συγκοινωνιών.[19]

Όταν οι κρατικές υπηρεσίες είναι σε θέση να αξιοποιήσουν και να εφαρμόσουν τα big data analytics που έχουν στη διάθεσή τους, κερδίζουν σημαντικό έδαφος όταν πρόκειται για τη διαχείριση των υπηρεσιών κοινής ωφέλειας, υπηρεσίες που ασχολούνται με την κυκλοφοριακή συμφόρηση ή στην αντιμετώπιση και πρόληψη του εγκλήματος,. Όμως, ενώ υπάρχουν πολλά πλεονεκτήματα στα bigdata, οι κυβερνήσεις θα πρέπει επίσης να αντιμετωπίσουν τα ζητήματα της διαφάνειας και της ιδιωτικής ζωής.(20)

Η Ευρωπαϊκή Επιτροπή ξεκίνησε το 2010 το "Ψηφιακό θεματολόγιο για την Ευρώπη" με σκοπό την ανάπτυξη οικονομικά βιώσιμών λύσεων για τους πολίτες της Ευρωπαϊκής Ένωσης μέσω διαδικτυακών εφαρμογών (EC, 2010).Το 2012 συνέχισε το έργο της με το "Ψηφιακό θεματολόγιο για την Ευρώπη και τις προκλήσεις για το 2012".Σε αυτή της την πρωτοβουλία η Ευρωπαϊκή Επιτροπή κατασκεύασε πλανά για την ασφαλή και κερδοφόρα χρήση των big data, με κύρια θέματα τα οικονομικά δημόσια δεδομένα ,το διαδίκτυο των πράγματων , την συνδεσιμότητα των συσκευών αλλά και την ασφάλεια των διαδικτυακών συναλλαγών .Διασφαλίζοντας έτσι την προστασία των δεδομένων των πολιτών-χρηστών της Ευρωπαϊκής Ένωσης.

Python.

Εισαγωγή στην Python.

Ένα πρόγραμμα Python χωρίζεται σε έναν αριθμό λογικών γραμμών και κάθε λογική γραμμή τερματίζεται από το λογότυπο NEWLINE. Μια λογική γραμμή δημιουργείται από μία ή περισσότερες φυσικές γραμμές.

Μια γραμμή περιέχει μόνο κενά, καρτέλες, φόρμες, ενδεχομένως ένα σχόλιο, είναι γνωστό ως κενή γραμμή και ο διερμηνέας της Python το αγνοεί.

Μια φυσική γραμμή είναι μια ακολουθία χαρακτήρων που τερματίζεται με μια ακολουθία τέλους γραμμής . Δείτε το ακόλουθο παράδειγμα.

```
x=1
if x>0:
    print("Αυτές οι 3 γραμμές είναι Φυσικές/Λογικές γραμμές")
```

Σχόλια στην Python .

Ένα σχόλιο ξεκινά με έναν χαρακτήρα hash (#) ο οποίος δεν είναι μέρος της γραμματοσειράς και τελειώνει στο τέλος της φυσικής γραμμής. Όλοι οι χαρακτήρες μετά τον χαρακτήρα # μέχρι το τέλος της γραμμής είναι μέρος του σχολίου και ο διερμηνέας της Python τους αγνοεί. Δείτε το ακόλουθο παράδειγμα. Θα πρέπει να σημειωθεί ότι η Python δεν διαθέτει εγκατάσταση πολλαπλών γραμμών ή μπλοκ σχόλια.

```
χ=1
#Η αρχική τιμή του χ είναι 1.
if x>0:
    print("Αυτά τα δυο είναι σχόλια")#Print a string.
```

Ενώνοντας δυο γραμμές.

Όταν θέλετε να γράψετε έναν μακρύ κομμάτι κώδικα σε μία γραμμή, μπορείτε να σπάσετε τη λογική γραμμή σε δύο ή περισσότερες φυσικές γραμμές χρησιμοποιώντας το χαρακτήρα πίσω στροφής (\). Επομένως, όταν μια φυσική γραμμή τελειώνει με χαρακτήρες αντίστροφης

κάθετος (\) και όχι ένα μέρος μιας γραμμικής συμβολοσειράς ή σχόλιο τότε μπορεί να ενταχθεί σε μια άλλη φυσική γραμμή. Δείτε το ακόλουθο παράδειγμα.

```
u=0

v=1

w=2

x=3

y=4

z=5

if u==0 and v>0 \

and w>1 and x>2 \

print ("Αυτό είναι ένα παράδειγμα ένωσης γραμμών.")
```

Πολλαπλές εντολές σε μια γραμμή .

Μπορείτε να γράψετε 2 ξεχωριστές εντολές σε μια γραμμή χρησιμοποιώντας το (;) χαρακτήρα ανάμεσα στις 2 γραμμές.

```
print("statement 1")

print("statement 2")


#Οι εντολές μπορούν να γραφτούν αλλιώς.


print("statement 1");print("statement 2")
```

Συντακτικοί κανόνες.

1)Χρησιμοποιήστε 4 κενά ανά εσοχή και καμία καρτέλα.

2)Μην αναμιγνύετε καρτέλες και κενά. Οι καρτέλες δημιουργούν σύγχυση και συνιστάται να χρησιμοποιείτε μόνο κενά.

3)Μέγιστο μήκος γραμμής: 79 χαρακτήρες που βοηθούν τους χρήστες με μικρή οθόνη.

4)Χρησιμοποιήστε κενές γραμμές για να διαχωρίσετε τους ορισμούς των λειτουργιών και των κλάσεων ανώτερου επιπέδου και για την απλή κενή γραμμή για να διαχωρίσετε ορισμούς μεθόδων μέσα σε μια κλάση και μεγαλύτερα μπλοκ κωδικών μέσα στις λειτουργίες.

5)Όταν είναι εφικτό, βάλτε τα σχόλια μέσα (πρέπει να είναι πλήρεις προτάσεις).

6)Χρησιμοποιήστε κενά μεταξύ εκφράσεων και δηλώσεων.

Python Κατοχυρωμένες Λέξεις.

Οι ακόλουθες λέξεις χρησιμοποιούνται ως κατοχυρωμένες λέξεις της γλώσσας και δεν μπορούν χρησιμοποιηθούν αλλιώς.

False	True	None	class	Continue	finally	for
is	lambda	return	try	def	from	nonlocal
while	anddel	global	Not	with	as	el
or	if	yield	assert	Else	Import	pass
break	except	In	raise			

Βιβλιοθήκες της Python.

Για την διερευνητική ανάλυση δεδομένων και για την εφαρμογή του αλγορίθμου μηχανικής μάθησης χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python. Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου για γενικό προγραμματισμό. Δημιουργήθηκε από τον Guido van Rossumκαι κυκλοφόρησε για πρώτη φορά το 1991. Η Python έχει μια φιλοσοφία σχεδίασης που δίνει έμφαση στην αναγνωσιμότητα του κώδικα, χρησιμοποιώντας υποχρεωτικά indentation [Python, (n.d)].

Όταν πρόκειται για εφαρμογές στον κλάδο του data science, η Python είναι ένα πολύ ισχυρό εργαλείο. Πολύ βασικό χαρακτηριστικό της το οποίο την κάνει πολύ δημοφιλή είναι ότι

πρόκειται για γλώσσα ανοιχτού κώδικα .Η Python έχει ανελιχθεί τα τελευταία χρόνια σε "αρχηγό" του data science programming.Ενώ υπάρχουν αυτοί που ακόμα χρησιμοποιούν R,SPSS,Jula ή άλλες διάσημες γλώσσες , η τεράστια ανάπτυξη της Python είναι ορατή από την συνεχόμενα αυξανόμενη συλλογή βιβλιοθηκών της για data science και για την επεξεργασία δεδομένων μερικές απο τις πιο δημοφιλείς είναι.(12)

Pandas.

Μία από τις πιο διάσημες data science βιβλιοθήκες είναι η Pandas .Έχει ενσωματωμένα χαρακτηριστικά όπως την ικανότητα να διαβάζει δεδομένα από πολλαπλές πηγές , να δημιουργεί μεγάλα dataframes ή πίνακες από αυτές τις πήγες και να υπολογίζει συνολικές αναλυτικές βασισμένες στα ερωτήματα που τίθενται .Έχει επίσης ενσωματωμένα ορισμένα μέσα οπτικοποίησης έτσι ώστε να δημιουργούνται διαγράμματα των αποτελεσμάτων καθώς και μια πληθώρα από μεθόδους εξαγωγής ώστε να μεταφέρονται οι ολοκληρωμένες αναλύσεις σε αρχεία μορφής excel spreadsheet. Δημιουργήθηκε από τον Wes McKinney το 2008.(13)(16)

NumPy.

Η "βιβλιοθήκη " NumPy αποτελεί ουσιαστικά ένα πακέτο μικρότερων βασικών βιβλιοθηκών για επιστημονική ανάλυση στην python .Περιέχει πολυδιάστατους πίνακες αντικειμένων ,εργαλεία για ενσωμάτωση κώδικα c ,c++ και fortran καθώς και χρήσιμες εφαρμογές γραμμικής άλγεβρας μετασχηματισμών fourier και τυχαίων αριθμών .Δημιουργήθηκε από τον Travis Oliphant το 2005.(14)

Matplotlib.

Η Matplotlib είναι μια δισδιάστατη βιβλιοθήκη κατασκευής διαγραμμάτων της python με αποτελέσματα ποιότητας δημοσίευσης σε διάφορες έντυπες μορφές και δια δραστικά περιβάλλοντα .Μπορεί να χρησιμοποιηθεί σε κώδικες python στα python και ipython shells στο σημειωματάριο jupyter καθώς και σε διαδικτυακές εφαρμογές .Έχει την δυνατότητα να δημιουργεί διαγράμματα ,ιστογράμματα ,φάσματα ισχύος ,πίνακες σφαλμάτων και διαγράμματα διασποράς με μερικές μόνο γραμμές κώδικα .Δημιουργήθηκε από τον John D. Hunter το 2003.(15)

Agate.

Μια σχετικά καινούργια βιβλιοθήκη που σκοπό έχει να λύσει ορισμένα προβλήματα στην ανάλυση δεδομένων είναι η Agate .Δημιουργήθηκε με άξονα την δημοσιογραφία και έχει μια πληθώρα εφαρμογών για ανάλυση συνόλου δεδομένων .Η Agate είναι πιο εύκολη στην εκμάθηση από τις υπόλοιπες βιβλιοθήκες και χρησιμεύει στην σύγκριση και στις στατιστικές βάσεων δεδομένων καθώς έχει απλά και ταχύτατα μέσα οπτικοποίησης .Δημιουργήθηκε από τον Christopher Groskopf το 2017.(13)

Scikit-learn.

Η Scikit-learn αποτελεί μια συμπληρωματική βιβλιοθήκη της python που παρέχει αλγορίθμους μηχανικής μάθησης με η χωρίς επίβλεψη .Είναι χτισμένη πάνω στις πιο γνωστές NumPy,pandas και Matplotlib .Προσφέρει εφαρμογές γραμμικής και λογικής παλινδρόμησης τεχνικές ομαδοποίησης και επιλογής μοντέλων καθώς και προεξεργασία συμπεριλαμβανόμενης της κανονικοποίησης Min-Max .Δημιουργήθηκε απο τον David Cournapeau το 2007.(16)

Τέλος, για εργασίες ανάλυσης δεδομένων συνίσταται η χρήση ως IDE (Integrated Development Enviroment) του IPython Notebook (γνωστό και ως Jupyter Notebook).(12)

Μια βασική εισαγωγή εντολών της Python .

Integers.

Οι ακέραιοι αριθμοί στην python θεωρούνται τύπου integer .

Με την εντολή type μαθαίνουμε τον τύπο ενός στοιχείου .

πχ

```
  type (10)
  class int #αποτέλεσμα
  type(1982932)
class int
type (657*474)
  class int
```

Οι αριθμοί με δεκαδικά θεωρούνται τύπου float .

πχ

```
type (4.2)
class float
type (3/2)
class float
```

Γράψτε ένα πρόγραμμα Python για να μετατρέψετε τις μοιρες(deg) σε ακτινια(rad).

Σημείωση: Το ακτίνιο είναι η τυπική μονάδα μέτρου γωνίας, που χρησιμοποιείται σε πολλούς τομείς των μαθηματικών. Η μέτρηση μιας γωνίας σε ακτίνια είναι αριθμητικά ίση με το μήκος ενός αντίστοιχου τόξου ενός μοναδιαίου κύκλου.Ενα ακτίνιο είναι λίγο κάτω από 57,3 μοίρες (όταν το μήκος του τόξου είναι ίσο με την ακτίνα).

Λύση:

```
pi=22/7
degree = float(input("Input degrees: "))
radian = degree*(pi/180)
print(radian)
```

Strings.

Strings θεωρούνται σειρές από δεδομένα αλφαβητικών χαρακτήρων ο τύπος string στην python είναι str.Η σύνταξη τους είναι της μορφής a="this is a string"με μονά η διπλά εισαγωγικά.

πχ

```
var1="Hello World"
var2="Python Programming"
```

Μπορούμε να δούμε απομονωμένα στοιχεία με τον εξής τρόπο.

print ("var1[0]:",var1[0])#θα τυπώσει το πρώτο στοιχείο του string ξεκινώντας από τα αριστερά.

print ("var2[1:5]:",var2[1:5])#θα τυπώσει από το δεύτερο έως και το πέμπτο στοιχειο .

var1[0]:H#αποτέλεσμα 1

var2[1:5]:ytho#αποτέλεσμα 2

Με την εντολή len() μπορούμε να μάθουμε το μήκος ενός στοιχείου.

πχ

```
var1="Hello World"
len(var1)
```

10

Lists.

Οι λίστες lists περιέχουν αντικείμενα που χωρίζονται από κόμματα και είναι μέσα σε αγκύλές ([]).Μπορούμε να έχουμε πρόσβαση στις αποθηκευμένες τιμές όπως προηγουμένως στα strings με εκθέτες ξεκινώντας από το 0 μηδέν από τα αριστερά μέσα σε αγκύλες .Τα δεδομένα των lists μπορούν να είναι όλων των διαφορετικών τύπων.

Ο τύπος δεδομένων list έχει κάποιες επιπλέων μεθόδους.

list.append(x)

Προσθέτει ένα αντικείμενο στο τέλος της λίστας.

list.extend(iterable)

Μεγαλώνει την λίστα προσθέτοντας όλα τα στοιχεία μιας άλλης λίστας .

list.insert(i, x)

Εισάγει ένα αντικείμενο σε μια θέση .Το πρώτο μέλος είναι η θέση του στοιχείου πριν από το οποίο θα γίνει η εισαγωγή .

list.remove(x)

Αφαιρεί το πρώτο στοιχείο της λίστας με αξία ίση με x .Αν δεν υπάρχει τέτοιο στοιχείο εμφανίζει ValueError.

list.pop([i])

Αφαιρεί το στοιχείο στην θέση i της λίστας και το επιστρέφει .Αν δεν καθοριστεί θέση i η εντολή a.pop() αφαιρεί και επιστρέφει το τελευταίο στοιχείο της λίστας.

list.clear()

Αφαιρεί όλα τα στοιχεία της λίστας.

list.count(x)

Εμφανίζει πόσες φορές υπάρχει το στοιχείο x μέσα στην λίστα.

list.reverse()

Αντιστρέφει τα στοιχεία της λίστας .

list.copy()

Επιστρέφει ένα αντίγραφο της λίστας.

πχ

```
list=['abcd',578,2.23,'John',70.2]
tinylist=[123,'John']

print (list)#τυπώνειτηνλίστα1
print (tinylist)#τυπώνειτηνλίστα2
print (list[1:3])# τυπώνει από το δεύτερο έως και το τρίτο στοιχείο
print (list[2:])# τυπώνει από το τρίτο στοιχείο
print (tinylist*2)#τυπώνει την λίστα 2 φορές
print (list + tinylist)# τυπώνει τις 2 λίστες μαζί
```

```
['abcd',578,2.23,'John',70.2]
[123,'John']
[578,2.23]
```

[2.23,'John',70.2]

[123,'John',123,'John',70.2,123,'John']

Tuples(πλειάδες).

Τα tuples είναι μια άλλη μορφή δεδομένων ακολουθίας που μοιάζει με τις lists πέρα από μερικές βασικές διαφορές.

Οι κύριες διαφορές ανάμεσα στις lists και στα tuples είναι ότι στις λίστες τα δεδομένα εισάγονται μέσα σε αγκύλες ενώ στα tuples μέσα σε παρενθέσεις.

Χρηστικά η βασική διαφορά τους είναι πως τα δεδομένα και το μέγεθος μιας λίστας είναι μεταβλητά ενώ στα tuples τα δεδομένα και το μέγεθος είναι αμετάβλητα .

Μπορούμε να δούμε τα tuples σαν read only lists.

πχ

```
tuple=('abcd',786,2.34,'Jim',43.2)
tinytuple=(456,'Jim')
print (tuple)
print tuple[0]
print tuple[1:3]
print tuple[2:]
print tinytuple
print tuple+tinytuple

list=['abcd',786,2.34,'Jim',43.2]

tuple[2]=1000#InvalidSyntaxταστοιχείαστατupleσείναιαμετάβλητα
list[2]=1000#Valid Syntax
```

Dictionaries.

Τα Dictionaries είναι τύποι πινάκων κατακερματισμού και αποτελούνται από ζεύγη κλειδιών-τιμών .Στα Dictionaries τα κλειδιά μπορεί να είναι σχεδόν όλων των τύπων δεδομένων της Python ,αλλά είναι συνήθως intergers η strings .Οι τιμές μπορεί να είναι οποιοδήποτε αυθαίρετο αντικείμενο Python .Τα Dictionaries βρίσκονται μέσα σε {} και οι τιμές τους μέσα σε [].

πχ

```
dict={}
dict['one']="This is one"
dict[2]="This is two"
tinydict={'name':'John','code':6734,'dept':'sales'}


print(dict['one'])#τυπώνει τιμή για το κλειδί 'one'
print(dict[2])#τυπώνει τιμή για το κλειδί '2'
print(tinydict)#τυπώνει ολόκληρο το dictionarie
print(tinydict.keys())#τυπώνει όλα τα κλειδιά του tinydict
print(tinydict.values())#τυπώνει όλες τις τιμές του tinydict
```

If statement.

Η εντολή if στην Python είναι ίδια με τις άλλες γλώσσες προγραμματισμού. Εκτελεί μια σειρά δηλώσεων υπό όρους, με βάση την αξία μιας λογικής έκφρασης.

Εδώ είναι η γενική μορφή μιας δήλωσης ενός τρόπου.

```
if expression:
   statement_1
   statement_2
   .....
```

Στην παραπάνω περίπτωση, η έκφραση καθορίζει τις συνθήκες που βασίζονται στην Boolean μορφή έκφρασης . Όταν αξιολογείται μια Boolean έκφραση παράγει είτε μια τιμή αληθής είτε ψευδής. Όσες φορές η έκφραση αξιολογηθεί αληθινή τόσες δηλώσεις θα εκτελεστούν. Αυτή η ομάδα των δηλώσεων ονομάζεται μπλοκ.

If Else statement.

Στην if...else εντολή της Python υπάρχουν 2 μπλοκ μια ακολουθώντας την έκφραση και μια ακολουθώντας την ρήτρα else .Ακολουθεί η σύνταξη .

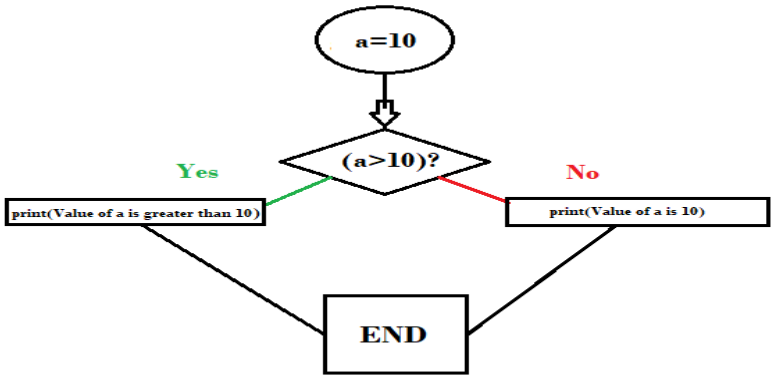
```
if expression:
        statement_1
        statement_2
        .....
          else :
       statement_3
       statement_4
        .....
```

πχ

```
a=10
if(a>10):
   print("Value of a is greater than 10")
else :
   print("Value of a is 10")
```

Αποτέλεσμα

```
   Value of a is 10
```



If..elif...else statement.

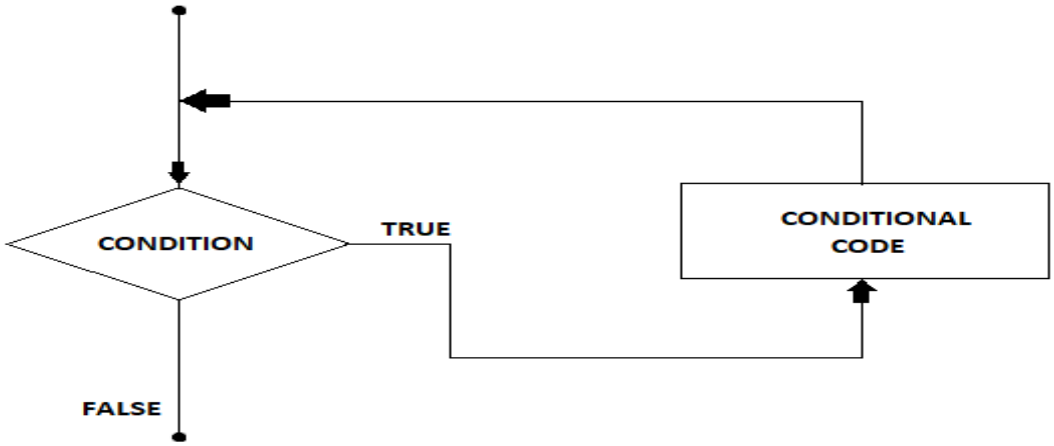
Μερικές φορές προκύπτει μια κατάσταση όπου υπάρχουν παραπάνω συνθήκες. Για να χειριστεί την κατάσταση, η Python επιτρέπει την προσθήκη οποιουδήποτε αριθμού ρητρών elif μετά από μια ρήτρα if και πριν από την ρήτρα else. Εδώείναιησύνταξη.

```
if expression:
    statement_1
    statement_2
    .....

elif expression2:
    statement_3
    statement_4
    .....
elif expression3:
```

```
    statement_5
    statement_6
    .....
else:
    statement_7
    statement_8
```

Loops.

Γενικα οι εντολές εκτελούνται ακολουθιακά .Οι γλώσσες προγραμματισμού έχουν διάφορους τρόπους που επιτρέπουν για πιο σύνθετες διαδρομές εκτέλεσης εντολών εντολή Loop μας επιτρέπει να εκτελούνται μια ή περισσότερες εντολές πολλές φορές.



Μερικές μορφές της loop

While loop.

Η while loop τρέχει όσο η έκφραση (condition) αξιολογείται ως True και εκτελείται το block του προγράμματος. Η συνθήκη ελέγχεται κάθε φορά στην αρχή του βρόχου και η πρώτη φορά που η έκφραση αξιολογείται ως False, ο βρόχος σταματά χωρίς να εκτελέσει οποιαδήποτε υπόλοιπη δήλωση.

```
while (expression) :
   statement_1
   statement_2
   ....
```

Το ακόλουθο παράδειγμα τυπώνει τα ψηφία 0 έως 4 καθώς ρυθμίζουμε την συνθήκη x <5.

```
x = 0;
while (x < 5):
    print(x)
x += 1
```

For loop.

Εκτελεί μια σειρά από εντολές πολλαπλές φορές και συμπτύσσει το κώδικα που ελέγχει την μεταβλητή βρόγχου.

Στην Python η for loop βρόχο χρησιμοποιείται για την επανάληψη των στοιχείων οποιασδήποτε ακολουθίας, συμπεριλαμβανομένης της λίστας Python, της συμβολοσειράς, της πλειάδας κτλ. Ο βρόχος for χρησιμοποιείται επίσης για την πρόσβαση σε στοιχεία από ένα σύνολο (για παράδειγμα λίστας, συμβολοσειρά, πλειάδα).

```
for variable_name in sequence :
statement_1
   statement_2
   ....
```

Στο παραπάνω παράδειγμα το color_list είναι μια ακολουθία που περιέχει μια λίστα με διάφορα ονόματα χρωμάτων. Όταν εκτελείται ο βρόχος για το πρώτο στοιχείο (δηλ. Κόκκινο) εκχωρείται στη μεταβλητή c. Μετά από αυτό, η εντολή εκτύπωσης θα εκτελεστεί και η διαδικασία θα συνεχιστεί μέχρι να φτάσουμε στο τέλος της λίστας.

```
#The list has four elements, indices start at 0 and end at 3
color_list = ["Red", "Blue", "Green", "Black"]
   for c in color_list:
      print(c)
```

Αποτέλεσμα

```
Red
Blue
Green
Black
```

Nested loops .

Μπορεί να χρησιμοποιηθούν μια ή περισσότερες loop μέσα σε άλλες while ή for loop.

Functions.

Οι συναρτήσεις functions είναι ένα κομμάτι οργανωμένου κώδικα πολλαπλών χρήσεων που χρησιμοποιείτε για να πραγματοποιηθεί μια συγκεκριμένη αποστολή.

Οι μεταβλητές σε μια function ορίζονται ως εξής.

Ξεκινάμε με την εντολή def ακολουθούμενη από το όνομα της συνάρτησης και παρενθέσεις .Μέσα στις παρενθέσεις ορίζονται οι παράμετροι τις συνάρτησης.

Η γραμμή κώδικα μέσα σε μια συνάρτηση ξεκινά με : .

Τέλος με την εντολή return κλείνουμε την συνάρτηση.

Καλούμε μια συνάρτηση απλά με το όνομα που της έχουμε δώσει ακολουθούμενο από τις τιμές που θέλουμε στην θέση των μεταβλητών.

πχ

```
def my_function(fname)#ορίζουμε την συνάρτηση
  print(fname+"Smith")#ορίζουμε μεταβλητή(fname) και σταθερά (Smith)
  return


 my_function("John")#καλούμε την συνάρτηση θέτοντας τιμές στην μεταβλητή
my_function("Will")
 my_function("Mike")
 John Smith#αποτελέσματα
 Will Smith
Mike Smith
```

NumPy.

Η Numpy είναι ένα πακέτο (σύνολο) συναρτήσεων ειδικά σχεδιασμένες για μαθηματικούς υπολογισμούς Κατασκευάζεται μια νέα δομή - κλάση αντικειμένων, η array, η οποία δίνει τη δυνατότητα να ορισθούν διανύσματα και πίνακες. Το array είναι μια δομή που μοιάζει με τη δομή list της Python. Σε αντίθεση με τη list η διάσταση ενός array είναι ορισμένη από την αρχή και αποθηκεύει αντικείμενα που είναι μόνο του ίδου τύπου (δηλ. int, float, boolean, str, ...).

Ξεκινάμε εισάγοντας την NumPy ως πακέτο με την εξής εντολή: import numpy as np

πχ

Γράψτε ένα πρόγραμμα στην python χρησιμοποιώντας την NumPy που να δημιουργεί ένα array με όλους τους αριθμούς από το 0 εως το 9 .

```
import numpy as np
 arr=np.arange(10)
arr
array([0,1,2,3,4,5,6,7,8,9])


 #Επειτα να εμφανίζει μόνο τους μονούς αριθμούς.
```

```
arr[arr%2==1]
```

```
array([1,3,5,7,9])
```

#Μετά να αλλάζει τους μονούς αριθμούς με -1 χωρίς να επηρεάζει το array.

```
arr[arr%2==1]=-1
```

```
array=([0,-1,2,-1,4,-1,6,-1,8,-1])
```

Μπορούμε να αλλάξουμε τη μορφή ενός array, δηλαδή ένα 1-διάστατο array μπορεί να γίνει 2-διάστατο. Χρησιμοποιούμε την εντολή reshape και ο μόνος περιορισμός που έχουμε είναι ότι το νέο array πρεπει να έχει το ίδιο πλήθος στοιχείων με το παλιό.

```
import numpy np

array = np.arange(8)
print("Original array : \n", array)

# Φτιάξτε array με 2 γραμμές και 4 στήλες.
array = np.arange(8).reshape(2, 4)
print("\narray reshaped with 2 rows and 4 columns : \n", array)

# Φτιάξτε array με 4 γραμμές και 2 στήλες.
array = np.arange(8).reshape(4 ,2)
print("\narray reshaped with 2 rows and 4 columns : \n", array)

# Φτιάξτε 3D array.
array = np.arange(8).reshape(2, 2, 2)
```

```
print("\nOriginal array reshaped to 3D : \n", array)
```

Αποτέλεσμα:

```
Original array :
[0 1 2 3 4 5 6 7]

array reshaped with 2 rows and 4 columns :
[[0 1 2 3]
[4 5 6 7]]

array reshaped with 4 rows and 2 columns :
[[0 1]
[2 3]
[4 5]
[6 7]]

Original array reshaped to 3D :
[[[0 1] [2 3]]

[[4 5] [6 7]]]
```

Γράψτε ένα πρόγραμμα στην python όπου να συγκρίνει τα στοιχεια 2 arrays και να τυπώνει με "μεγαλύτερο","μεγαλύτερο_ίσο","μικρότερο","μικρότερο_ίσο".

```
import numpy as np
x = np.array([3, 5])
y = np.array([2, 5])
print("Αρχικοί Αριθμοί:")
print(x)
print(y)
```

```
print("Σύγκριση-Μεγαλύτερο")
print(np.greater(x, y))
print("Σύγκριση - Μεγαλύτερο_Ισο")
print(np.greater_equal(x, y))
print("Σύγκριση - Μικρότερο")
print(np.less(x, y))
print("Σύγκριση - Μικρότερο_Ισο")
print(np.less_equal(x, y))
```

Αποτέλεσμα.

```
Αρχικοί Αριθμοί:
[3 5]
[2 5]
Σύγκριση-Μεγαλύτερο
[ True False]
Σύγκριση-Μεγαλύτερο_Ισο
[ True  True]
Σύγκριση-Μικρότερο
[False False]
Σύγκριση-Μεγαλύτερο_Ισο
[False  True]
```

Γράψτε ένα πρόγραμμα NumPy που θα δημιουργεί έναν 3X4 πίνακα και θα εμφανίζει όλα τα στοιχεία μαζί.

```
import numpy as np
a = np.arange(10,22).reshape((3, 4))
print("Original array:")
```

```
print(a)
print("Each element of the array is:")
for x in np.nditer(a):
  print(x,end=" ")
```

Αποτέλεσμα.

```
 Original array:
[[10 11 12 13]
[14 15 16 17]
[18 19 20 21]]
Each element of the array is:
10 11 12 13 14 15 16 17 18 19 20 21
```

Γράψτε ένα πρόγραμμα NumPy που θα δημιουργεί ένα διάνυσμα με τιμές από το 0 μέχρι και το 19 και θα αργότερα θα αλλάζει το πρόσημο των στοιχείων από το 9 μέχρι και το 15 .

```
import numpy as np
x = np.arange(20)
print("ΑρχικόΔιάνυσμα:")
print(x)
print("Το διάνυσμα μετά την αλλαγή των πρόσημών:")
x[(x >= 9) & (x <= 15)] *= -1
print(x)
```

Αποτέλεσμα.

Αρχικό Διάνυσμα:

[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]

Το διάνυσμα μετα την αλλαγή των πρόσημών:

[0 1 2 3 4 5 6 7 8 -9 -10 -11 -12 -13 -14 -15 16 17 18 19]

Γράψτε ένα πρόγραμμα με NumPy που να δημιουργεί εναν 3χ3χ3 πίνακα και να τον γεμίζει με τυχαίους αριθμούς.

```
import numpy as np
x = np.random.random((3, 3, 3))
print(x)
```

Αποτέλεσμα

```
[[[0.89206729 0.00180581 0.09577541]
  [0.34849376 0.23888253 0.01650311]
  [0.20927968 0.64180191 0.1695297 ]]

[[0.29354233 0.27323626 0.87722478]
  [0.61975391 0.97572226 0.31635234]
  [0.75580267 0.0413768  0.6441258 ]]

[[0.23648272 0.48477901 0.95840333]
  [0.65680114 0.61959274 0.29261268]
  [0.89747936 0.71915232 0.64876149]]]
```

Pandas.

Το pandas είναι ένα πακέτο Python που παρέχει γρήγορες, ευέλικτες και εκφραστικές δομές δεδομένων σχεδιασμένες για να κάνουν την εργασία με τα 'συγγενικά' ή 'taged' δεδομένα τόσο εύκολη όσο και διαισθητική. Σκοπός του είναι να αποτελέσει μια θεμελιώδη βάση υψηλού επιπέδου για την πραγματοποίηση πρακτικών αναλύσεων δεδομένων πραγματικού κόσμου στην Python.

πχ

Γράψτε ένα πρόγραμμα στην Python που θα δημιουργεί και θα εμφανίζει ένα μονοδιάστατο πίνακα αντικειμένων που περιέχει μια σειρά δεδομένων χρησιμοποιώντας το πακέτο Pandas.

```
import pandas as pd
ds=pd.Series9[2,4,6,8,10])
print(ds)
```

Αποτέλεσμα

```
0   2
1   4
2   6
3   8
4   10
dtype:int64
```

Μετατρέψτε τον Pandas πίνακα που δημιουργήσατε προηγουμένως σε λίστα Python και τύπο list

```
print(type(ds))
print("Convert Pandas Series to Python list")
print(ds.tolist())
print(type(ds.tolist()))
```

Αποτέλεσμα.

```
<class 'pandas.core.series.Series'>
Convert Pandas Series to Python list
[2, 4, 6, 8, 10]
<class 'list'>
```

Γράψτε ένα πρόγραμμα στην Python όπου θα μετατρέπει έναν πίνακα NumPy σε μια σειρά Pandas.

Sample Pandas Array d1=[10,20,30,40,50]

```
import numpy as np
import pandas as pd
np_array = np.array([10, 20, 30, 40, 50])
print("NumPy array:")
print(np_array)
new_series = pd.Series(np_array)
print("Converted Pandas series:")
print(new_series)
```

Αποτέλεσμα

NumPy array:

[10 20 30 40 50]

Converted Pandas series:

0 10

1 20

2 30

3 40

4 50

dtype: int64.

Γράψτε ένα πρόγραμμα στην Python για να αλλάξετε τον τύπο δεδομένων που δίνεται σε μια στήλη ή μια σειρά.

```
import pandas as pd
s1 = pd.Series(['100', '200', 'python', '300.12', '400'])  # Δημιουργίατηςαρχικήςστήλης
print("Original Data Series:")
print(s1)
print("Change the said data type to numeric:")
s2 = pd.to_numeric(s1, errors='coerce') # Μετάφρασησεαριθμητικότύπο .Μετηνεντολή errors ="coerce" μετατρέπειταπιθανά errors σε NaN(Not a Number)
print(s2)
```

Αποτέλεσμα

Original Data Series:

0 100

1 200

2 python

3 300.12

4 400

dtype: object

Change the said data type to numeric:

0 100.00

1 200.00

2 NaN # Δεν μπορεί να μεταφραστεί σε αριθμητικό

3 300.12

4 400.00

dtype: float64

Γράψτε ένα πρόγραμμα στην Python που θα δημιουργεί μια σειρά δεδομένων με Pandas και αργότερα προσθέστε επιπρόσθετα δεδομένα.

```
import pandas as pd
s = pd.Series(['400', '600', 'pandas', '500.56', '900'])
print("Original Data Series:")
print(s)
print("\nData Series after adding some data:")
new_s = s.append(pd.Series(['1000', 'php']))
print(new_s)
```

Αποτέλεσμα

Original Series:

```
0      400
1      600
2    pandas
3    500.56
4      900
dtype: object

Data Series after adding extra data:
0      400
1      600
2    pandas
3    500.56
4      900
0     1000
1      php
dtype: object
```

Γράψτε ένα πρόγραμμα στην Python χρησιμοποιώντας την Pandas που θα υπολογίζει τον μέσο όρο και την τυπική απόκλιση των δεδομένων μιας σειράς.

```
import pandas as pd
s = pd.Series(data = [1,2,3,4,5,6,7,8,9,5,3])
print("Original Data Series:")
print(s)
print("Mean of the said Data Series:")
print(s.mean()) #Εντολή μέσου όρου
print("Standard deviation of the said Data Series:")
print(s.std()) #Εντολή τυπικής απόκλισης
```

Αποτέλεσμα

Original Data Series:

0 1

1 2

2 3

3 4

4 5

5 6

6 7

7 8

8 9

9 5

10 3

dtype: int64

Mean of the said Data Series:

4.81818181818

Standard deviation of the said Data Series:

2.52262489555

Python File Input and Output.

Υπάρχουν διάφοροι τρόποι για να παρουσιάσετε την έξοδο ενός προγράμματος. Τα δεδομένα μπορούν να εκτυπωθούν σε μορφή αναγνώσιμη από τον άνθρωπο ή να αποθηκευτούν σε ένα αρχείο για μελλοντική χρήση.

Ο απλούστερος τρόπος παραγωγής εξόδου είναι η χρήση της εκτύπωσης (print) όπου μπορείτε να περάσετε καμία ή περισσότερες εκφράσεις διαχωρισμένες με κόμματα. Αυτή η συνάρτηση μετατρέπει τις εκφράσεις που περνάτε σε string.

open()

Για να μπορέσετε να διαβάσετε ή να γράψετε σε ένα αρχείο, πρέπει να το ανοίξετε χρησιμοποιώντας την ενσωματωμένη εντολή open() της Python. Αυτή η συνάρτηση δημιουργεί ένα αντικείμενο file, το οποίο θα χρησιμοποιηθεί για την κλήση άλλων μεθόδων υποστήριξης που σχετίζονται με αυτό.

Σύνταξη .

file object = open(file_name [, access_mode][, buffering])

Ακολουθεί εξήγηση των παραμέτρων.

file_name: Είναι μια τιμή μορφης string που περιέχει το όνομα του αρχείου στο οποίο θέλετε να αποκτήσετε πρόσβαση.

access_mode: Η εντολή αυτή καθορίζει τον τρόπο με τον οποίο πρέπει να ανοίξει το αρχείο, δηλαδή, να διαβάσετε(read), να γράψετε(write), να προσθέσετε(append), κ.λπ. Μια πλήρης λίστα πιθανών τιμών δίνεται παρακάτω στον πίνακα. Αυτή είναι προαιρετική παράμετρος και η προεπιλεγμένη λειτουργία πρόσβασης αρχείων είναι η read.

buffering: Εάν η τιμή προσωρινής αποθήκευσης έχει οριστεί σε 0, δεν γίνεται buffering. Εάν η τιμή προσωρινής αποθήκευσης είναι 1, η προσωρινή αποθήκευση κάθε γραμμής εκτελείται κατά την πρόσβαση σε ένα αρχείο. Εάν καθορίσετε την τιμή προσωρινής αποθήκευσης ως ακέραιο αριθμό μεγαλύτερο από 1, τότε εκτελείται ενέργεια προσωρινής αποθήκευσης με το υποδεικνυόμενο μέγεθος.

Λίστα μορφών πρόσβασης σε ένα αρχείο.

1:(r)

Ανοίγει ένα αρχείο μόνο για ανάγνωση. Ο δείκτης αρχείου τοποθετείται στην αρχή του αρχείου. Αυτή είναι η προεπιλεγμένη λειτουργία.

2:(rb)

Ανοίγει ένα αρχείο μόνο για ανάγνωση σε δυαδική μορφή. Ο δείκτης αρχείου τοποθετείται στην αρχή του αρχείου. Αυτή είναι η προεπιλεγμένη λειτουργία.

3:(r+)

Ανοίγει ένα αρχείο για ανάγνωση και γραφή. Ο δείκτης του αρχείου τοποθετείται στην αρχή του αρχείου.

4:(rb+)

Ανοίγει ένα αρχείο για ανάγνωση και εγγραφή σε δυαδική μορφή. Ο δείκτης του αρχείου τοποθετείται στην αρχή του αρχείου.

5:(w)

Ανοίγει ένα αρχείο μόνο για γραφή. Αντικαθιστά το αρχείο εάν υπάρχει. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για γραφή.

6:(wb)

Ανοίγει ένα αρχείο για εγγραφή μόνο σε δυαδική μορφή. Αντικαθιστά το αρχείο εάν υπάρχει. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για γραφή.

7:(w+)

Ανοίγει ένα αρχείο για γραφή και ανάγνωση. Αντικαθιστά το υπάρχον αρχείο εάν υπάρχει. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για ανάγνωση και γραφή.

8:(wb+)

Ανοίγει ένα αρχείο για γραφή και ανάγνωση σε δυαδική μορφή. Αντικαθιστά το υπάρχον αρχείο εάν υπάρχει. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για ανάγνωση και γραφή.

9:(a)

Ανοίγει ένα αρχείο για προσθήκη. Ο δείκτης αρχείου βρίσκεται στο τέλος του. Δηλαδή, το αρχείο βρίσκεται σε λειτουργία προσάρτησης. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για γραφή.

10:(ab)

Ανοίγει ένα αρχείο για προσθήκη σε δυαδική μορφή. Ο δείκτης αρχείου βρίσκεται στο τέλος του. Δηλαδή, το αρχείο βρίσκεται σε λειτουργία προσάρτησης. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για γραφή.

11:(a+)

Ανοίγει ένα αρχείο για προσθήκη και ανάγνωση. Ο δείκτης αρχείου βρίσκεται στο τέλος του. Δηλαδή, το αρχείο βρίσκεται σε λειτουργία προσάρτησης. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για γραφή.

12:(ab+)

Ανοίγει ένα αρχείο για προσθήκη και ανάγνωση σε δυαδική μορφή. Ο δείκτης αρχείου βρίσκεται στο τέλος του. Δηλαδή, το αρχείο βρίσκεται σε λειτουργία προσάρτησης. Εάν το αρχείο δεν υπάρχει, δημιουργεί ένα νέο αρχείο για γραφή.

Μόλις ανοίξει ένα αρχείο και έχετε ένα αντικείμενο file, μπορείτε να λάβετε διάφορες πληροφορίες που σχετίζονται με αυτό το αρχείο.

1:(file.closed)

Επιστρέφει αληθής(true) αν το αρχείο είναι κλειστό αλλιώς ψευδής(false).

2:(file.mode)

Επιστρέφει την μορφή με την οποία είχε ανοιχτεί το αρχείο.

3:(file.name)

Επιστρέφει το όνομα του αρχείου.

4:(file.softspace)

Επιστρέφει ψευδής(false) εάν για την εκτύπωση print απαιτείται κενό ,αληθής(true) διαφορετικά.

Close()

Η μέθοδος close() ενός file αντικειμένου ξεσκαρτάρει τυχόν άγραφες πληροφορίες και κλείνει το αρχείο, μετά την εντολή close()δεν μπορεί να γίνει πλέον εγγραφή.

Η Python κλείνει αυτόματα ένα αρχείο file όταν το αντικείμενο αναφοράς του αρχειου εκχωρείται σε άλλο αρχείο. Είναι βέλτιστο να χρησιμοποιείτε τη μέθοδο close () για να κλείσετε ένα αρχείο.

Write()

Η μέθοδος write() γράφει οποιαδήποτε συμβολοσειρά string σε ένα ανοιχτό αρχείο. Είναι σημαντικό να σημειωθεί ότι οι συμβολοσειρές στην Python μπορούν να έχουν δυαδικά δεδομένα και όχι μόνο κείμενο.

Σύνταξη

fileObject.write(string)

Read()

Η μέθοδος read () διαβάζει μια συμβολοσειρά string από ένα ανοιχτό αρχείο .Να σημειωθεί ότι οι συμβολοσειρές στην Python μπορούν να έχουν και δυαδικά δεδομένα εκτός από τα αλφαριθμητικά δεδομένα .

Σύνταξη

fileObject.read([count])

Αυτή η μέθοδος ξεκινά την ανάγνωση από την αρχή του αρχείου και αν λείπει το στοιχείο count, τότε προσπαθεί να διαβάσει όσο το δυνατόν περισσότερο, ίσως μέχρι το τέλος του αρχείου.

Παρακάτω θα χρειαστούμε την μέθοδο os της Python η οποία μας παρέχει μεθόδους που μας βοηθούν να εκτελούμε εργασίες επεξεργασίας αρχείων, όπως μετονομασία και κατάργηση αρχείων.

Για να χρησιμοποιήσετε αυτήν την ενότητα πρέπει πρώτα να την εισαγάγετε και στη συνέχεια μπορείτε να καλέσετε τυχόν σχετικές λειτουργίες.

rename()

Η μέθοδος της μετονομασίας δέχεται δυο παραμέτρους ,το τρέχον όνομα του αρχείου και το νέο όνομα.

Σύνταξη

os.rename(current_file_name, new_file_name)

remove()

Με την μέθοδο της κατάργησης μπορούμε να διαγράψουμε ένα αρχείο με παράμετρο το όνομα του αρχείου.

Σύνταξη

os.remove(file_name)

Συνδυαστικά Projects

Η παγκόσμια πανδημία με κωδική ονομασία COVID-19 η οποία είναι μια μολυσματική ασθένεια που προκαλείται από σοβαρό οξύ αναπνευστικό σύνδρομο coronavirus 2 (SARS-CoV-2) , εμφανίστηκε για πρώτη φορά τον Δεκέμβριο του 2019 στο Γουχάν της Κίνας και είχε ως αποτέλεσμα μια συνεχιζόμενη πανδημία. Η πρώτη υπόθεση μπορεί να εντοπιστεί στις 17 Νοεμβρίου 2019.Μέχρι και τις 10 Ιουνίου 2020, έχουν αναφερθεί περισσότερα από 7,24 εκατομμύρια περιπτώσεις σε 188 χώρες και εδάφη, με αποτέλεσμα περισσότερους από 411.000 θανάτους. Περισσότερα από 3,37 εκατομμύρια άνθρωποι έχουν ανακάμψει. [21}

Ένα από τα πλεονεκτήματα που έχουμε σήμερα στον αγώνα κατά του κορονοϊού που δεν ήταν τόσο εξελιγμένο στο ξέσπασμα SARS του 2003 είναι τα μεγάλα δεδομένα και το υψηλό επίπεδο τεχνολογίας που είναι διαθέσιμο. Η Κίνα αξιοποίησε τα μεγάλα δεδομένα(big data), την μηχανική εκμάθηση(machine learning) και άλλα ψηφιακά εργαλεία καθώς ο ιός εξαπλώθηκε σε ολόκληρο το έθνος, προκειμένου να εντοπίσει και να περιορίσει την επιδημία. Τα διδάγματα που αντλήθηκαν από την Κίνα εξακολούθησαν να εξαπλώνονται σε όλο τον κόσμο καθώς άλλες χώρες καταπολεμούν την εξάπλωση του ιού και χρησιμοποιούν ψηφιακή τεχνολογία για να αναπτύξουν προβλέψεις σε πραγματικό χρόνο και να οπλίσουν επαγγελματίες υγείας και κυβερνητικούς υπεύθυνους λήψης αποφάσεων με πληροφορίες που μπορούν να χρησιμοποιηθούν για να προβλέψουν το αντίκτυπο του κορονοϊου.(22)

Μεταξύ της αναγνώρισης συμπτωμάτων, της παρακολούθησης του ιού και της παρακολούθησης της διαθεσιμότητας των νοσοκομειακών πόρων, οι ερευνητές έρχονται αντιμέτωποι με τεράστιες ποσότητες πληροφοριών - ποσότητες τις οποίες είναι αδύνατον να κατανοήσουν και να αναλύσουν οι ίδιοι οι άνθρωποι. Είναι μια κατάσταση που φαίνεται να είναι ειδικά προσαρμοσμένη για προηγμένες τεχνολογίες ανάλυσης δεδομένων.(23)

Παρακάτω ακολουθεί ένα σύνολο ασκήσεων που αντλεί δεδομένα από το CSSE(Center for Systems Science and Engineering) του Johns Hopkins University πάνω στον κορονοΪό τα οποία ανανεώνονται καθημερινά .

Πηγή δεδομένων :https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports

Ονομασίεςαρχείων: MM-DD-YY.csv σε UTC(Coordinated Universal Time)

Περιγραφή περιοχών: Province(Επαρχία) / State(πολιτεία)/City name(όνομα πόλης).

Τελευταία ανανέωση: MM/DD/YY HH:mm (24 μορφή σε UTC).

Confirmed: ο αριθμός των επιβεβαιωμένων κρουσμάτων.

Deaths: ο αριθμός των θανάτων.

Recovered: αριθμός περιπτώσεων που ανέκαμψαν.

Update Frequency(ρυθμός ανανέωσης).Αρχεία μετα τη 1 Φεβρουάριου μια φορά την ημέρα στις 23:59 UTC.

lat/lon: Γεωγραφικό πλάτος/Γεωγραφικό μήκος.

Χρησιμοποιώντας το παραπάνω σύνολο δεδομένων έχουν δημιουργηθεί κάποιες ασκήσεις πάνω στο COVID-19 (Διάδοση του ιού, ανάλυση, οπτικοποίηση, πρόβλεψη και συγκρίσεις).

Άσκηση 1

Να γραφτεί ένα πρόγραμμα στην Python που θα εμφανίζει τις 5 πρώτες γραμμές από το σύνολο δεδομένων του COVID-19.Επισης να εκτυπωθούν οι πληροφορίες του συνόλου δεδομένων και να ελεγχθούν οι τις τιμές που λείπουν.

Λύση:

```
import pandas as pd

covid_data= pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/03-17-2020.csv').

pd.set_option('display.max_rows',None)#για να εμφανίσει όλες τις γραμμές με την μια καθώς η pandas έχει προεγκατεστημένο όριο 5 γραμμών.

pd.set_option('display.max_columns',12)#αντιστοιχα για τις στήλες

print(covid_data)

print("\nDataset information:")

print(covid_data.info())

print("\nMissing data information:")

print(covid_data.isna().sum())
```

Άσκηση 2

Να γραφτεί ένα πρόγραμμα στην Python που θα λαμβάνει τον τελευταίο αριθμό επιβεβαιωμένων, θανάτων, αναρρωμένων και ενεργών περιπτώσεων από COVID-19.

```
import pandas as pd

covid_data= pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/03-17-2020.csv')

pd.set_option('display.max_rows',None)

covid_data['Active'] = covid_data['Confirmed'] - covid_data['Deaths'] - covid_data['Recovered']

result = covid_data.groupby('Country/Region')['Confirmed', 'Deaths', 'Recovered', 'Active'].sum().reset_index()

print(result)
```

Άσκηση 3

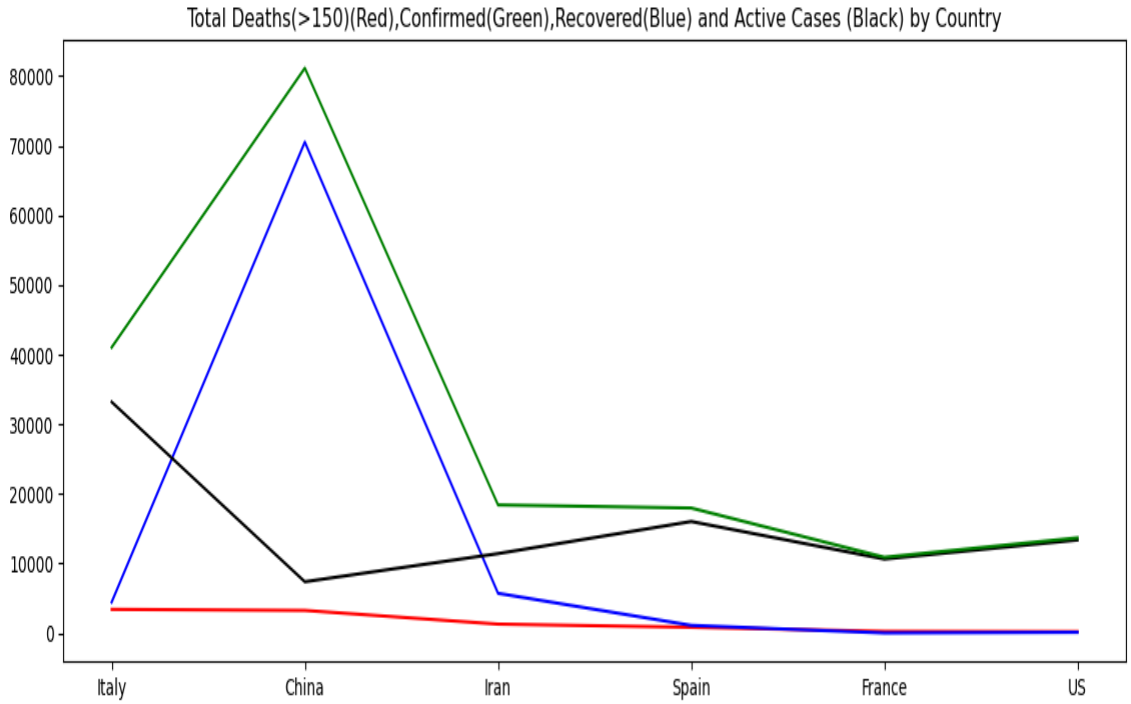
Να γραφτεί ένα πρόγραμμα στην Python όπου θα εμφανίζει τους θανάτους από COVID-19 ανά χώρα.

```
import pandas as pd

covid_data= pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/03-17-2020.csv')

pd.set_option('display.max_rows',None)

data = covid_data.groupby('Country/Region')['Confirmed', 'Deaths', 'Recovered'].sum().reset_index()

result = data[data['Deaths']>0][['Country/Region', 'Deaths']]

print(result)
```

4)Να γραφτεί ένα πρόγραμμα στην Python για να δημιουργήσετε ένα διάγραμμα των θανάτων(Deaths),των επιβεβαιωμένων κρουσμάτων(Confirmed), των περιπτώσεων που ανέκαμψαν(Recovered) και των ενεργών περιπτώσεων(Active) ανά χώρα όπου οι θάνατοι είναι περισσότεροι από 150.

```
#Αρχικά πρέπει να εγκαταστήσουμε την βιβλιοθήκη matplotlib

pip3 install matplotlib
```

```
import pandas as pd

import matplotlib.pyplot as plt

covid_data= pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/03-19-2020.csv', usecols = ['Last Update', 'Country/Region', 'Confirmed', 'Deaths', 'Recovered'])

covid_data['Active'] = covid_data['Confirmed'] - covid_data['Deaths'] - covid_data['Recovered']


r_data = covid_data.groupby(["Country/Region"])["Deaths", "Confirmed", "Recovered", "Active"].sum().reset_index()

r_data = r_data.sort_values(by='Deaths', ascending=False)

r_data = r_data[r_data['Deaths']>150]

plt.figure(figsize=(15, 5))

plt.plot(r_data['Country/Region'], r_data['Deaths'],color='red')

plt.plot(r_data['Country/Region'], r_data['Confirmed'],color='green')

plt.plot(r_data['Country/Region'], r_data['Recovered'], color='blue')

plt.plot(r_data['Country/Region'], r_data['Active'], color='black')


plt.title('Total Deaths(>150), Confirmed, Recovered and Active Cases by Country')

plt.show()
```



Άσκηση 5

Να γραφτεί ένα πρόγραμμα στην Python που θα απεικονίζει τα επιβεβαιωμένα κρούσματα κορονοϊού (COVID-19) σε παγκόσμιο επίπεδο με την πάροδο του χρόνου.

$ pip install plotly==4.8.1#Αρχικα εγκαθιστούμε την βιβλιοθήκη plotly ,είναι μια δια δραστική βιβλιοθήκη ανοιχτής πηγής κατασκευής στατιστικών, οικονομικών, γεωγραφικών, επιστημονικών και τρισδιάστατων γραφημάτων.

import pandas as pd

import plotly.express as px #Μας επιτρέπει ταχεία εξερεύνηση δεδομένων και δημιουργία διαγραμμάτων

import plotly.io as pio# Μας επιτρέπει να αποθηκεύουμε στατικές εικόνες των γραφικών παραστάσεων .

pio.templates.default = "plotly_dark"

covid_data= pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/03-19-2020.csv')

```
grouped = covid_data.groupby('Last Update')['Last Update', 'Confirmed', 'Deaths'].sum().reset_index()
fig = px.line(grouped, x="Last Update", y="Confirmed",
        title="Worldwide Confirmed Novel Coronavirus(COVID-19) Cases Over Time")
fig.show()
```

Επιπλέων συνδυαστικές ασκήσεις.

Άσκηση 6

Επιλογή τυχαίων λαχειοφόρων. Δημιουργήστε 100 τυχαίους αριθμούς λαχείων και διαλέξτε δύο τυχαίους αριθμούς από αυτούς ως νικητές.

```
import random

lottery_tickets_list = []
print("Δημιουργία 100 τυχαίων αριθμών λαχείων")#για να δημιουργηθούν 100 λαχεία

for i in range(100):

   lottery_tickets_list.append(random.randrange(1000000000, 9999999999))#οιαριθμοίλαχείωνπρέπειναείναι 10 ψηφίων (1000000000, 9999999999)

winners = random.sample(lottery_tickets_list, 2)#επιλογη 2 νικητήριωναριθμών
print("Τα 2 τυχερά λαχεία είναι", winners)
```

Άσκηση 7

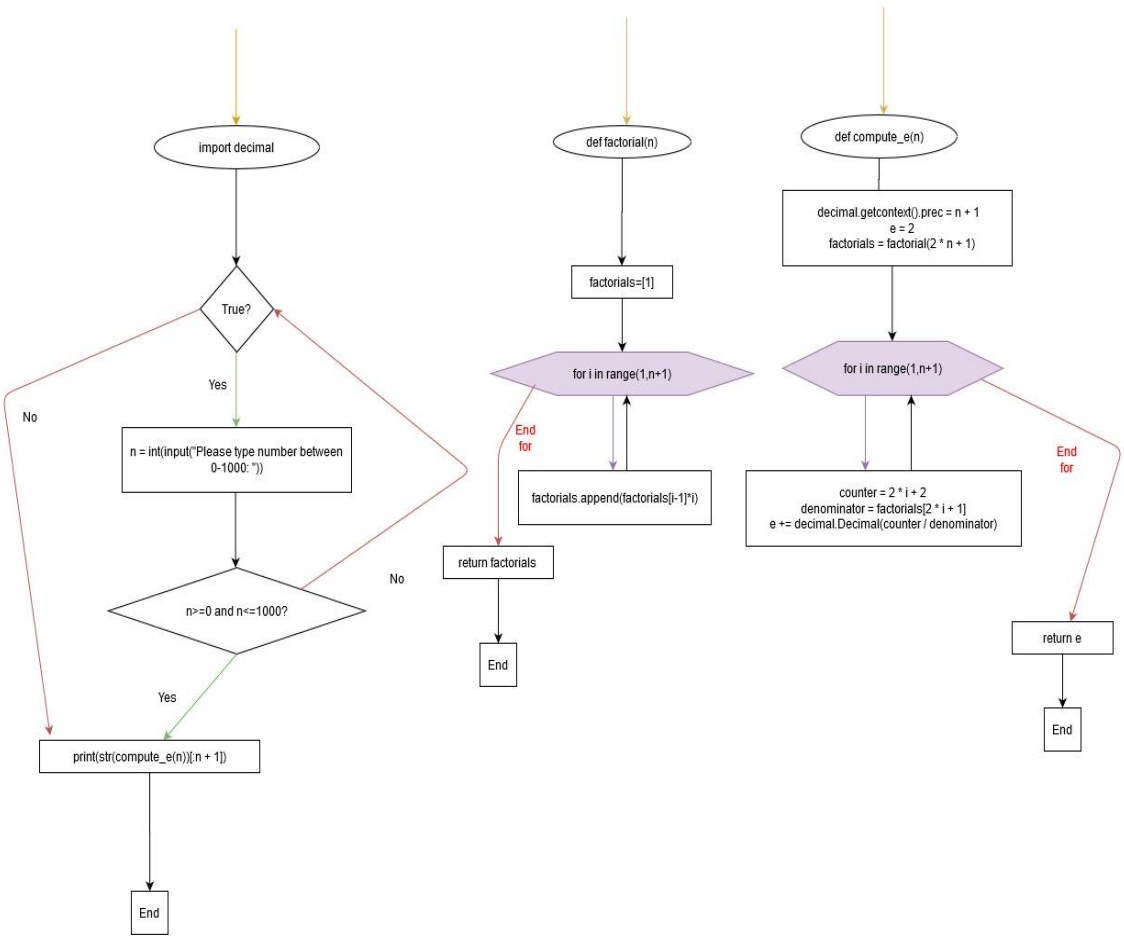
Να γραφτεί ένα πρόγραμμα στην Python όπου ο χρήστης θα προσπαθεί να μαντέψει έναν τυχαία επιλεγμένο αριθμό από το 0 έως το 100, αν απαντήσει λάθος ο χρήστης, το πρόγραμμα θα εκτυπώνει "ο αριθμός αυτός είναι μικρότερος από τον ζητούμενο ,προσπαθήστε ξανά" ή "ο αριθμός αυτός είναι μεγαλύτερος από τον ζητούμενο ,προσπαθήστε ξανά" αντίστοιχα. Αν ο χρήστης βάλει στοιχείο που δεν είναι ακέραιος αριθμός θα εμφανίζει "αυτό που εισήγατε δεν είναι ακέραιος αριθμός ,προσπαθήστε ξανά "Αν βρει τον αριθμό "συγχαρητήρια βρήκατε τον αριθμό".

```
import random
number=random.randrange(0,100)
guessCheck='wrong'
print('Καλωσήρθατε στο μαντέψτε τον αριθμό.')
while guessCheck=='wrong':
   response=int(input('Εισάγετε έναν αριθμό από το 0 έως το 100.'))
try:
      val=int(response)
      except ValueError:
print('Αυτό που εισαγάγατε δεν είναι ακέραιος αριθμός, προσπαθήστε ξανά.')
continue
      val=int (response)
      if val<number:
print('Ο αριθμός αυτός είναι μικρότερος από τον ζητούμενο αριθμό ,προσπαθήστε ξανά.')
      elif val>number:
         print('Ο αριθμός αυτός είναι μεγαλύτερος από τον ζητούμενο αριθμό ,προσπαθήστε ξανά.)
      else:
         print('Συγχαρητήρια βρήκατε τον αριθμό')
      guessCheck='correct'
print('Ευχαριστώ που παίξατε μαντέψτε τον αριθμό')
```

Άσκηση 8

Να γραφτεί ένα πρόγραμμα στην Python που να υπολογίζει τα n δεκαδικά ψηφία της σταθεράς του euler (e).

```
import decimal


def factorial(n):
    factorials = [1]
    for i in range(1, n + 1):
        factorials.append(factorials[i - 1] * i)
    return factorials


def compute_e(n):
    decimal.getcontext().prec = n + 1
    e = 2
    factorials = factorial(2 * n + 1)
    for i in range(1, n + 1):
        counter = 2 * i + 2
        denominator = factorials[2 * i + 1]
        e += decimal.Decimal(counter / denominator)
    return e

while True:
        n = int(input("Please type number between 0-1000: "))
        if n >= 0 and n <= 1000:
            break
print(str(compute_e(n))[:n + 1])
```


import decimal
True?
Yes
No
n = int(input("Please type number between 0-1000: "))
n>=0 and n<=1000?
No
Yes
print(str(compute_e(n))[:n + 1])
End
def factorial(n)
factorials=[1]
for i in range(1,n+1)
End for
factorials.append(factorials[i-1]*i)
return factorials
End
def compute_e(n)
decimal.getcontext().prec = n + 1
e = 2
factorials = factorial(2 * n + 1)
for i in range(1,n+1)
End for
counter = 2 * i + 2
denominator = factorials[2 * i + 1]
e += decimal.Decimal(counter / denominator)
return e
End

Βιβλιογραφία.

1)Διπλωματική εργασία του ΞΩΝΙΚΗ ΜΙΧΑΗΛ ΓΕΩΡΓΙΟΥ 2018

(2)Hurwitz, J. S., Nugent, A., Halper, F., & Kaufman, M. (2013). Big Data For Dummies.John Wiley & Sons.

(3)Foote, K. D. (14 Δεκεμβρίου2017). Dataversity.A Brief History of Big Data: http://www.dataversity.net/brief-history-big-data/

(4)Dasgupta, N. (2018). Practical Big Data Analytics.Packt Publishing Ltd.

[5](http://business-analytics.gr/news/1147-big-data)

[6]Reznor, E. P. (2017). A Beginner's Guide To Using Data Science For Business.

(7)Wikipedia. (2019). Big Data. https://en.m.wikipedia.org/wiki/Big_data

(8)HR's Secret Weapon: The Power of Big Data.(2017)Ultimate Softare:

https://www.ultimatesoftware.com/Contact/hcm-whitepapers-hr-secret-weapon-the-power-of-big-data

(9)[Πολυμένης Ιορδάνης,Διπλωματική εργασία--Πανεπιστήμιο Μακεδονίας, Θεσσαλονίκη. (2017)]

(10)Διπλωματική Εργασία του Τσαουσίδη Φώτιου (ΑΕΜ: 612) 2019

(11)[Hooja S.,(2017)].

(12)[Theuwissen M., (2015)]

(13)Introduction to Data Science: How to "Big Data" with Python https://dataconomy.com/2016/10/big-data-python

(14)NumPy official site https://numpy.org/

(15)matplotlib official site https://matplotlib.org/

(16)https://www.codecademy.com/articles/scikit-learn scikit-learn

(17)https://pandas.pydata.org/ pandas

(18)http://chronosmag.eu/index.php/big-data-ygeia.html big-data υγεια

(19) Wikipedia.(2020). Big Data. https://en.m.wikipedia.org/wiki/Big_data

(20)Διπλωματική εργασία του Θεόδωρου Αθανασόπουλου 2018

(21)https://en.wikipedia.org/wiki/Coronavirus_disease_2019

(22)https://www.linkedin.com/pulse/vital-role-big-data-fight-against-covid-19-coronavirus-bernard-marr

(23)https://healthitanalytics.com/news/understanding-the-covid-19-pandemic-as-a-big-data-analytics-issue

(24)https://www.intmath.com/exponential-logarithmic-functions/calculating-e.php

(25)Διπλωματική Εργασία του Τσώλα Λεωνίδα 'Η χρήση των Big Data Analytics για τη βελτίωση των ψηφιακών υπηρεσιών υγείας'ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ (2017)

(26)Raghupathi N., Raghupathi V., «Big Data analytics in healthcare: promise and potential», Health Information Science and Systems, 2014

(27)https://www.greekinternetmarketing.com/blog/proothisi-istoselidon/big-data-marketing

(28)Διπλωματικη εργασία "Εφαρμογή τεχνολογιών Big Data στον τραπεζικό τομέα" Χρυσοστομου Μαρια 2019.

(29)Christos Vaitsis, Vasilis Hervatis and Nabil Zary (July 20th 2016). Introduction to Big Data in Education and Its Contribution to the Quality Improvement Processes, Big Data on Real-World Applications, Sebastian Ventura Soto, José M. Luna and Alberto Cano, IntechOpen, DOI: 10.5772/63896.

(30)https://el.wikipedia.org/wiki/Internet of Things

(31)https://el.wikipedia.org/wiki/Entity Relationship Model

(32)https://el.wikipedia.org/wiki/SQL

(33)https://el.wikipedia.org/wiki/Data Warehouse

(34)(http://business-analytics.gr/news/1147-big-data)

Ηλεκτρονικές πηγές.

www.geeksforgeeks.org

www.greekinternetmarketing.com

fisica.cab.cnea.gov.ar

hypatia.teiath.gr

Ikee.lib.auth.gr

www.chronosmag.eu

Introduction to Data Science: How to “Big Data” with Python https://dataconomy.com/2016/10/big-data-python

https://numpy.org/

https://matplotlib.org/

https://www.codecademy.com/articles/scikit-learn

https://pandas.pydata.org/

http://chronosmag.eu/index.php/big-data-ygeia.html

https://www.w3schools.com/python/python_functions.asp

https://www.tutorialspoint.com/python/python_basic_syntax.htm

http://users.math.uoc.gr/~p.chatzipa/index_files/mem107/2019a/labs2019/labs/Numpy7/python24.html

https://www.practicaldatascience.org/html/exercises/Exercise_numpy.html

https://www.tutorialspoint.com/python/python_files_io.html/exercises/Exercise_numpy

https://apothesis.eap.gr/handle/repo/44241

https://www.intechopen.com/books/big-data-on-real-world-applications/introduction-to-big-data-in-education-and-its-contribution-to-the-quality-improvement-processes

www.physics4u.gr