ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ. ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ

Διπλωματική Εργασία: Πειραματική Μελέτη Συσταδοποίησης Δεδομένων με τον Αλγόριθμο K-Means στο Spark

Νικόλαος Πεφάνης **Επιβλέπων :Χρήστος Δουλκερίδης**

Αθήνα Ιούνιος 2020

Περιεχόμενα

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Δουλκερίδη Χρήστο για την αμέριστη συμπαράστασή του και την πολύτιμη καθοδήγηση που μου παρείχε καθ’ όλη τη διάρκεια της εκπόνησης της μεταπτυχιακής μου διατριβής.

Επίσης, θα ήθελα να ευχαριστήσω τους καθηγητές του μεταπτυχιακού προγράμματος «Ψηφιακά Συστήματα και Υπηρεσίες» για τις πολύτιμες γνώσεις που μου παρείχαν καθ’ όλη τη διάρκεια των σπουδών μου.

Τέλος, ένα τεράστιο ευχαριστώ στη μητέρα μου για τη στήριξη και την αγάπη της όλα αυτά τα χρόνια, στον αδερφό μου, στον Μήτσο που είναι σαν πατέρας μου, στην οικογένειά μου και στους φίλους μου για την πολύτιμη συμπαράστασή τους καθ’ όλη τη διάρκεια των σπουδών μου.

Περίληψη

Στα πλαίσια της παρούσας διπλωματικής γίνεται εισαγωγή στις έννοιες Επιστήμη Δεδομένων και τη Μηχανική Μάθηση. Σκοπός της μελέτης που έγινε είναι μέσα από διαφορετικά σύνολα δεδομένων να παρατηρήσουμε τη συμπεριφορά του δημοφιλούς αλγορίθμου K-Means χρησιμοποιώντας ταυτόχρονα διάφορες τεχνολογίες όπως η γλώσσα προγραμματισμού Python και το Apache Spark.

Abstract

An introduction to Data Science and Machine Learning is attempted in the context of this diploma thesis. The purpose of the study made was to observe, through different data sets, the behavior of the popular K-Means algorithm by using various technologies, such as the Python and Apache Spark.

ΜΕΡΟΣ 1: ΕΙΣΑΓΩΓΗ

1.1. Εισαγωγή στην Επιστήμη Δεδομένων (Data Science)

Η εξέλιξη στην υπολογιστική δυνατότητα των συστημάτων Η/Υ οδήγησε τις τελευταίες δεκαετίες στην ραγδαία αύξηση του όγκου των δεδομένων που παράγονται, αποθηκεύονται και διακινούνται. Αυτό με τη σειρά του δημιούργησε την ανάγκη ανάλυσης και επεξεργασίας δεδομένων μεγάλου όγκου και διαφορετικών τύπων, κάτι το οποίο δεν ήταν εφικτό με τα παραδοσιακά μοντέλα και εργαλεία. Δημιουργήθηκε έτσι ο τομέας της Επιστήμης Δεδομένων (Data Science) ο οποίος έχει ως αντικείμενο αυτήν τη νέα πραγματικότητα στα δεδομένα. Παράλληλα γεννήθηκε η ζήτηση για επαγγελματίες επιστήμονες δεδομένων (Data Scientists) οι οποίοι θα είχαν επιπλέον δεξιότητες από τον απλό αναλυτή δεδομένων (data analyst) ή στατιστικό (statistician).

Ο όρος Data Science σε συνδυασμό με τον όρο Δεδομένα μεγάλης κλίμακας (Big Data) έχει αποκτήσει αυξανόμενη δυναμικότητα τα τελευταία χρόνια και μνημονεύεται πλέον πολύ συχνά τόσο σε ακαδημαϊκό όσο και επαγγελματικό επίπεδο. Αν και δεν υπάρχει σήμερα ένας καθολικά αποδεκτός ορισμός, μπορούμε να χρησιμοποιήσουμε κάποια στοιχεία που είναι ευρέως αποδεκτά για να τον ορίσουμε. Πολύ γενικά, θα λέγαμε ότι Data Science θεωρείται το πεδίο το οποίο έχει ως αντικείμενο την εξαγωγή γνώσης από δεδομένα με τη βοήθεια, χρήση και συνεργασία θεωριών και τεχνικών από αρκετά επιστημονικά πεδία όπως: μαθηματικά, στατιστική, θεωρία πληροφορίας, τεχνολογίες πληροφορικής και επιστήμη υπολογιστών.

Από τις απαρχές της επιστήμης υπολογιστών μπορεί να βρει κανείς αναφορές στον επιστημονικό όρο Data Science ή σε συναφείς σε αυτόν όρους. Τις δεκαετίες του 1970 και 1980 οι αναφορές αυτές λειτουργούσαν περισσότερο ως υποκατάστατα της επιστήμης υπολογιστών και της επεξεργασίας δεδομένων και δεν είχαν κάποια σχέση με τη σημερινή έννοια του όρου. Αργότερα, τη δεκαετία του 1990, όπου υπήρξε η άνθηση της στατιστικής ως της κατεξοχήν επιστήμης ανάλυσης δεδομένων, υπήρχαν επίσης αναφορές στον όρο αυτόν. Όμως, λειτουργούσε κυρίως ως υποκατάστατο του όρου της στατιστικής μέσω Η/Υ, καθώς ήταν μια περίοδος όπου η ανάλυση των δεδομένων αναπτυσσόταν ταχέως με τη βοήθεια των υπολογιστών.

Η πρώτη εμφάνιση του όρου Data Science ως ένα νέο διακριτό πεδίο γίνεται το 2001 σε ένα άρθρο του William Cleveland ο οποίος το παρουσίασε ως μια προσπάθεια για την επέκταση του πεδίου της στατιστικής. Από τότε ο όρος άρχισε να αποκτά τη δική του δυναμική σχετικά σύντομα και στην πορεία άρχισαν να εμφανίζονται τα πρώτα εξειδικευμένα περιοδικά στο χώρο. Επίσης το 2008 εμφανίστηκαν και οι πρώτες αναφορές σε επιστημονικές ομάδες ως επιστήμονες δεδομένων και έκτοτε άρχισε να διαμορφώνεται αντίστοιχη ζήτηση στην αγορά εργασίας.

Όπως αναφέρθηκε προηγουμένως η δυναμική της Επιστήμης Δεδομένων δεν θα ήταν η ίδια αν δεν προχωρούσε ταυτόχρονα η εξέλιξη των υπολογιστικών συστημάτων, που προκάλεσε κατά τη δεκαετία 2000-2010 τεράστια ώθηση στην παραγωγή δεδομένων, τη φθηνή αποθήκευση και γρήγορη επεξεργασία τους. Η συσσώρευση δεδομένων μεγάλης κλίμακας (Big Data) οδήγησε σύντομα σε

δυσκολία διαχείρισης και επεξεργασίας τους με τα υφιστάμενα μοντέλα και αρχιτεκτονικές. Οι μεγάλοι οργανισμοί προώθησαν νέες τεχνολογίες με κυριότερη αυτή του υπολογιστικού νέφους (Cloud Computing) όπου κατευθύνονται πλέον αρκετά από τα δεδομένα μεγάλης κλίμακας.

Στην τρέχουσα περίοδο από το 2010 και έπειτα, το πεδίο έγινε περισσότερο δημοφιλές καθώς πραγματοποιήθηκαν αρκετές εκδόσεις νέων βιβλίων και περιοδικών, δημιουργήθηκαν φορείς και οργανώθηκαν συνέδρια που ώθησαν την διαμόρφωση των ορίων του νέου πεδίου αλλά και συνέβαλαν στην εκμάθηση τεχνικών και μεθόδων σε συνδυασμό με γλώσσες προγραμματισμού όπως η Python και η R.

Σε συνδυασμό με την προβλεπόμενη αύξηση των αναγκών σε θέσεις εργασίας στην Επιστήμη Δεδομένων αρκετά πανεπιστήμια ξεκίνησαν να προσφέρουν προγράμματα τόσο σε προπτυχιακό όσο και μεταπτυχιακό επίπεδο. Επιπλέον δημιουργήθηκαν επαγγελματικοί οργανισμοί και κοινότητες όπως ο Data Science Central και ο Kaggle, όπου ο καθένας συμβάλλει με το δικό του τρόπο στην προώθηση του αντικειμένου.

1.2. Το πρόβλημα της Διπλωματικής

Η έννοια της χωρικής ανάλυσης (spatial analysis) εσφαλμένα συσχετίζεται στη χώρα μας με τις έννοιες των Συστημάτων Γεωγραφικών Πληροφοριών (Geographical Information Systems – GIS) και της Γεωπληροφορικής, εξαιτίας του γεγονότος ότι δεν αποτελεί ένα διαδεδομένο γνωστικό αντικείμενο στην Ελλάδα. Αν και είναι άμεσα συνδεδεμένη με τα Συστήματα Γεωγραφικών Πληροφοριών και τη Γεωπληροφορική, πρόκειται στην πραγματικότητα για έναν ξεχωριστό τομέα έρευνας ο οποίος βασίζεται στην ανάλυση της χωρικής πληροφορίας των δεδομένων. Συναντάται πολύ συχνά να εφαρμόζεται κάποια μέθοδος χωρικής ανάλυσης για τη δημιουργία ενός θεματικού χάρτη ή ενός πίνακα στατιστικών δεδομένων, χωρίς να αποδεικνύεται αυτό επαρκώς από τον υπεύθυνο της ανάλυσης. Σε διεθνή κλίμακα, η χωρική ανάλυση αφορά στην ποσοτική γεωγραφία και γενικότερα στην ποσοτική ανάλυση δεδομένων, το οποίο την καθιστά ιδανική για εφαρμογές σε πολλά επιστημονικά πεδία που απαιτούν ανάλυση στατιστικών δεδομένων με γεωγραφική αναφορά. Η έννοια «ανάλυση χώρου» δεν είναι ταυτόσημη με την χωρική ανάλυση και σε καμία περίπτωση δεν πρέπει να χρησιμοποιείται αντί αυτής.

Με βάση έναν άλλον ορισμό (Unwin 1981) η χωρική ανάλυση είναι η επεξεργασία των σημείων, γραμμών, περιοχών κα επιφανειών ενός χάρτη, ή απλούστερα η επιστήμη που εξάγει συμπεράσματα από τα χωρικά δεδομένα ενός Συστήματος Γεωγραφικών Πληροφοριών. Εν αντιθέσει με αυτή την μάλλον γενική περιγραφή για το τι είναι χωρική ανάλυση, οι Bailey and Gatrell (1995) θεωρούν τη χωρική ανάλυση ως μια ποσοτική ανάλυση των χωρικών φαινομένων που βρίσκονται στον γεωγραφικό χώρο, οι οποίοι, στην προσπάθειά τους να αποφύγουν την γενικότητα του όρου χωρική ανάλυση, εστιάζουν στην ανάλυση χωρικών δεδομένων. Αυτή λαμβάνει υπόψη δεδομένα παρατήρησης για διάφορα φαινόμενα του γεωγραφικού χώρου, εξετάζοντας μοντέλα, μεθόδους και τεχνικές για να εξακριβώσει μια πιθανή ύπαρξη σχέσης ανάμεσα σε διαφορετικά χωρικά φαινόμενα.

Η ολοένα αυξανόμενη διαθεσιμότητα μεγάλου όγκου δεδομένων, από τα οποία εξάγονται διάφορα συμπεράσματα για τις ανθρώπινες συμπεριφορές (τόσο ατομικά όσο και συλλογικά), έχει οδηγήσει στην περαιτέρω διείσδυση της έννοιας της χωρικής ανάλυσης στους κλάδους της πληροφορικής και της στατιστικής. Πυρήνα για την εφαρμογή πολλών μεθόδων και τεχνικών χωρικής ανάλυσης αποτελούν τα μαθηματικά και η στατιστική, συνοδευόμενα από κατάλληλα λογισμικά και κώδικες που είναι γραμμένοι σε κάποια γλώσσα προγραμματισμού. Καθοριστική είναι η συμβολή της εξόρυξης δεδομένων (data mining) και της ανάλυσης μεγάλων δεδομένων (big data analysis) στο να γίνει η χωρική ανάλυση πιο επίκαιρη και σύγχρονη.

Στην παρούσα διπλωματική εργασία γίνεται προσπάθεια ομαδοποίησης διαφορετικών συνόλου δεδομένου τα οποία περιέχουν γεωγραφικές συντεταγμένες (longitude ,latitude) . Χρησιμοποιείται ο δημοφιλής αλγόριθμος KMeans ,για τον οποίο και θα μιλήσουμε αναλυτικά στο Κεφάλαιο 2, με στόχο μέσα από μια σειρά επαναλήψεων να βρούμε τον μέγιστο αριθμό ομάδων(k) που σχηματίζονται για τα δεδομένα μας ξεχωριστά και στη συνέχεια θα προσπαθήσουμε να βγάλουμε κάποια ασφαλή συμπεράσματα σχολιάζοντας τα αποτελέσματα.

1.3. Διάρθρωση των Κεφαλαίων

Η παρούσα Διπλωματική εργασία χωρίζεται σε τρία κεφάλαια. Στο πρώτο κεφάλαιο γίνεται μία σύντομη εισαγωγή στο Data Science, όπου δίνεται ο ορισμός της και κάποια βασικά χαρακτηριστικά που την καθιστούν σημαντική τεχνολογία τη σημερινή εποχή. Έπειτα γίνεται μία αναφορά στη χωρική ανάλυση και στα χωρικά δεδομένα τα οποία απασχολούν την ανθρωπότητα χιλιάδες χρόνια στο παρελθόν και τέλος το πρώτο κεφάλαιο κλείνει με μία πολύ σύντομη περιγραφή του προβλήματος με το οποίο θα ασχοληθεί η τρέχουσα διπλωματική.

Στη συνέχεια στο Κεφάλαιο 2 γίνεται εισαγωγή στη μηχανική μάθηση, το οποίο συνδέεται άμεσα με το Data Science. Παρουσιάζονται τα κυριότερα χαρακτηριστικά της καθώς επίσης γίνεται εκτενής περιγραφή ενός προβλήματος της μηχανικής μάθησης, από την αρχή μέχρι το τέλος που είναι η λύση και η ανάλυση των αποτελεσμάτων. Παράλληλα γίνεται αναφορά στους κυριότερους αλγόριθμους ομαδοποίησης.

Στο τρίτο και τελευταίο Κεφάλαιο θα αναφερθούμε στις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας εργασίας και θα πούμε μερικά πράγματα για τον K-Means όπως επίσης θα τονίσουμε τα βασικά του χαρακτηριστικά. Στη συνέχεια θα περιγράψουμε τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμός μας και θα αρχίσουμε να αναλύουμε τη λύση του προβλήματος. Ταυτόχρονα θα γίνει παρουσίαση των αποτελεσμάτων και σχολιασμός τους.

ΜΕΡΟΣ 2 : ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

2.1. Εισαγωγή στη Μηχανική Μάθηση

Η μηχανική μάθηση (machine learning) αποτελεί βασικό εργαλείο της Επιστήμης Δεδομένων και επειδή αρκετή από την εργασία ενός επιστήμονα δεδομένων περιλαμβάνει εργασίες αυτού του τύπου απαιτείται εξοικείωση με το συγκεκριμένο πεδίο. Ο βαθμός στον οποίο θα εμβαθύνει κανείς εξαρτάται από τον προσανατολισμό του. Δηλαδή αν έχει υπόβαθρο επιστήμης υπολογιστών θα είναι σε θέση να ασχοληθεί περισσότερο με τον σχεδιασμό, τον προγραμματισμό και υλοποίηση αλγόριθμων εν γένει. Ενώ αν έχει υπόβαθρο μαθηματικών ή στατιστικής με τα θεωρητικά μοντέλα και την ανάλυση δεδομένων. Η μηχανική μάθηση είναι ένα σύγχρονο επιστημονικό πεδίο το οποίο βρίσκεται στην τομή των πεδίων της επιστήμης υπολογιστών, της μηχανικής και της στατιστικής. Η εφαρμογή της είναι ευρεία καθώς είναι ένα εργαλείο το οποίο μπορεί να εφαρμοστεί σε πληθώρα προβλημάτων τα οποία έχουν σχέση με εργασίες σε δεδομένα και την ερμηνεία τους.

Ειδικότερα, η μηχανική μάθηση είναι μια περιοχή της τεχνητής νοημοσύνης που έχει ως αντικείμενο την κατασκευή αλγόριθμων και μεθόδων που επιτρέπουν στους υπολογιστές να «μαθαίνουν». Λίγο αυστηρότερα, μηχανική μάθηση ονομάζεται η δημιουργία μοντέλων ή προτύπων από ένα σύνολο δεδομένων, από ένα υπολογιστικό σύστημα. Ο στόχος της είναι η κατασκευή προσαρμόσιμων και ευέλικτων προγραμμάτων υπολογιστών τα οποία λειτουργούν με βάση αυτοματοποιημένη ανάλυση συνόλων δεδομένων και όχι τη διαίσθηση των μηχανικών που τα προγραμμάτισαν. Η μηχανική μάθηση χρησιμοποιεί σε σημαντικό βαθμό τη στατιστική, καθώς και τα δύο πεδία μελετούν την ανάλυση δεδομένων, το καθένα από τη δική του οπτική.

Ένα γνωστό παράδειγμα της χρήσης της μηχανικής μάθησης αποτελεί η αναγνώριση ανεπιθύμητων μηνυμάτων email. Η αναζήτηση εμφάνισης μιας λέξης δεν είναι επαρκής για την ταξινόμηση του μηνύματος. Αλλά η εμφάνιση συγκεκριμένων λέξεων σε συνδυασμό με το μέγεθος του μηνύματος και άλλους παράγοντες οδηγεί στην ταξινόμηση με μεγάλη ακρίβεια, κάτι που επιτυγχάνεται με τη βοήθεια αλγόριθμων μηχανικής μάθησης.

Η σημασία της είναι ήδη μεγάλη και αυξάνεται με σημαντικό ρυθμό. Ο λόγος δεν είναι άλλος από την μεγάλη αλλαγή που επιφέρει ο όγκος δεδομένων που συσσωρεύεται συνεχώς. Εκτός των δεδομένων τα οποία παράγονται από τους ανθρώπους χρήστες του διαδικτύου, και περιέχουν κείμενα, εικόνες, βίντεο και οτιδήποτε μπορεί να μετατραπεί σε ψηφιακή μορφή, τα τελευταία χρόνια έχει αυξηθεί κατακόρυφα η τεχνολογία ανιχνευτών και συσκευών καταγραφής που συνδέονται στο διαδίκτυο και δημιουργούν μεγάλο πλέον ποσοστό της κίνησης σε αυτό. Αυτό έχει ως αποτέλεσμα να είναι αδύνατη η εξαγωγή πληροφορίας και γνώσης δίχως τη βοήθεια αυτόματων αλγόριθμων και εργαλείων όπως αυτά που παρέχει η μηχανική μάθηση.

Παρακάτω παρουσιάζονται μια σειρά όρων που συναντιούνται συχνά στη μηχανική μάθηση και αποτελούν βασικά στοιχεία της.

- **Έμπειρα συστήματα:** Είναι συστήματα τα οποία είναι σε θέση να λειτουργούν όπως ένας εμπειρογνώμονας σε κάποιο πεδίο και να αποφασίζουν όπως αυτός. Περιέχουν κανόνες και φίλτρα που έχουν αποκτηθεί από εμπειρογνώμονες και έχουν ενσωματωθεί σε αυτά. Η λειτουργία τους μπορεί να είναι ενσωματωμένη σε αρκετά συστήματα τα οποία αυτοματοποιούν τις λειτουργίες τους με βάση κανόνες.

- **Ιδιότητες ή χαρακτηριστικά:** Στη γλώσσα της μηχανικής μάθησης οι ιδιότητες (attributes) ή τα χαρακτηριστικά (features) αποτελούν σημαντικό στοιχείο ενός αντικειμένου. Συνήθως ένα αντικείμενο χαρακτηρίζεται από ένα σύνολο ιδιοτήτων, από τις οποίες κάποιες είναι μετρήσιμες και κάποιες άλλες όχι. Το σύνηθες είναι να μετρούνται οι ιδιότητες που έχουν σχέση με το πρόβλημα που είναι προς επίλυση κάθε φορά. Το αποτέλεσμα της μέτρησης μιας ιδιότητας καταλήγει σε κάποια τιμή που αποτελεί δεδομένο το οποίο μπορεί να εκφραστεί σε οποιαδήποτε μορφή. Οι τιμές μπορεί να είναι ποσοτικές ή ποιοτικές, όμως σε κάθε περίπτωση πρέπει να είναι σύμφωνες με το μοντέλο δεδομένων που χρησιμοποιεί ο αλγόριθμος κάθε φορά.

- **Ταξινόμηση:** Μια από τις βασικές λειτουργίες της μηχανικής μάθησης είναι η ταξινόμηση. Πολύ συνοπτικά, στη διαδικασία της ταξινόμησης ένα αντικείμενο εντάσσεται σε μια κατηγορία με βάση τις ιδιότητές του. Η απόφαση ένταξής του ή όχι εξαρτάται από τα κριτήρια με βάση τα οποία καθορίζεται μια κατηγορία. Στη μηχανική μάθηση η διαδικασία εκτελείται από έναν αλγόριθμο ο οποίος έχει εκπαιδευτεί κατάλληλα προηγουμένως.

- **Σύνολο εκπαίδευσης:** Για να είναι ένας αλγόριθμος ταξινόμησης σε θέση να εκτελεί ταξινόμηση ή άλλη εργασία, πρέπει σε ορισμένες περιπτώσεις, να έχει περάσει από μια διαδικασία εκπαίδευσης. Για το σκοπό αυτό χρησιμοποιείται ένα σύνολο δεδομένων το οποίο ονομάζεται σύνολο εκπαίδευσης το οποίο αποτελείται από ένα αριθμό παραδειγμάτων τα οποία χαρακτηρίζονται από ορισμένες ιδιότητες και μια ή περισσότερες μεταβλητές στόχο. Η μεταβλητή στόχος είναι αυτή που επιχειρεί ο αλγόριθμος να προβλέψει και στα παραδείγματα είναι γνωστή στον αλγόριθμο. Ο αλγόριθμος αναγνωρίζει στα παραδείγματα κάποια συσχέτιση μεταξύ των τιμών των ιδιοτήτων και της μεταβλητής στόχου και στη συνέχεια δημιουργεί κανόνες για την ταξινόμηση.

- **Σύνολο δοκιμών:** Όταν ολοκληρωθεί η εκπαίδευση του αλγόριθμου ο αλγόριθμος τροφοδοτείται με ένα σύνολο δεδομένων στο οποίο οι τιμές των μεταβλητών στόχου δεν περιλαμβάνονται. Το σύνολο αυτό αποτελεί το σύνολο δοκιμών ή ελέγχου και ο αλγόριθμος ταξινομεί τα δεδομένα του με βάση την εκπαίδευσή του. Στο τέλος τα αποτελέσματα συγκρίνονται με τις σωστές τιμές και ελέγχεται ο αλγόριθμος για την ακρίβειά του.

- **Αναπαράσταση γνώσης:** Όταν ο αλγόριθμος έχει ελεγχθεί για την ακρίβειά του μπορεί να χρησιμοποιηθεί ως τμήμα ενός έμπειρου συστήματος. Η γνώση που έχει αποκτήσει μπορεί να παρασταθεί με διάφορους τρόπους, είτε ως κανόνες ενός έμπειρου συστήματος, ή ως μια κατανομή πιθανότητας, ή άλλη μορφή.

Η μηχανική μάθηση χρησιμοποιείται για αρκετές διαφορετικές εργασίες και συνήθως οι αλγόριθμοί της καταλήγουν σε κάποιο προϊόν το οποίο έχει τη μορφή εφαρμογής λογισμικού. Ανάλογα με το πρόβλημα που βρίσκεται υπό μελέτη οι εργασίες μπορούν να διαχωριστούν σε δύο βασικές κατηγορίες.

- **Μάθηση με επίβλεψη (supervised learning):** Σε αυτή την περίπτωση η μάθηση πραγματοποιείται με παραδείγματα και ο αλγόριθμος καλείται να μάθει μια έννοια ή μια συσχέτιση (συνάρτηση) από ένα σύνολο παραδειγμάτων.

- **Μάθηση χωρίς επίβλεψη (unsupervised learning):** Σε αυτή την περίπτωση η μάθηση γίνεται χωρίς παραδείγματα, και ο αλγόριθμος καλείται να μάθει μια έννοια ή μια συσχέτιση (συνάρτηση) από ένα σύνολο δεδομένων δημιουργώντας ο ίδιος τα πρότυπα που υπάρχουν, αν υπάρχουν.

Στη μάθηση με επίβλεψη εντάσσονται δύο είδη προβλημάτων, η ταξινόμηση και η παλινδρόμηση.

1. **Ταξινόμηση (classification):** Στην ταξινόμηση ο στόχος είναι η πρόβλεψη της κατηγορίας ή κλάσης στην οποία ανήκει ένα αντικείμενο με βάση τις τιμές των ιδιοτήτων του. Το βασικό σημείο είναι ότι οι κατηγορίες είναι διακριτές.
2. **Παλινδρόμηση (regression):** Στην παλινδρόμηση το ζητούμενο είναι η πρόβλεψη μιας αριθμητικής τιμής. Ένα συνηθισμένο παράδειγμα είναι η εύρεση της βέλτιστης γραμμής που διατρέχει ένα σύνολο σημείων και παριστά τη συνάρτηση.

Στη μάθηση χωρίς επίβλεψη εντάσσονται αρκετά είδη προβλημάτων όπως η ομαδοποίηση, η εκτίμηση πυκνότητας πιθανότητας, η μείωση των διαστάσεων.

1. **Ομαδοποίηση (clustering):** Στην ομαδοποίηση ο αλγόριθμος έχει ως στόχο να ανακαλύψει συσχετίσεις και ομάδες από τα δεδομένα με βάση μόνο τις τιμές των ιδιοτήτων τους. Έτσι, αντικείμενα που έχουν παραπλήσια στοιχεία ομαδοποιούνται στην ίδια ομάδα.

2. **Στην εκτίμηση πυκνότητας πιθανότητας**, ο στόχος είναι η εύρεση των τιμών ορισμένων στατιστικών παραμέτρων των δεδομένων.

3. Η **μείωση των διαστάσεων** αφορά στα δεδομένα τα οποία περιγράφονται από ιδιότητες πολλών διαστάσεων και είναι δύσκολη η οπτική τους αναπαράσταση. Σε αυτή την περίπτωση ο στόχος είναι να περιοριστούν οι διαστάσεις χωρίς όμως αυτό να έχει επίπτωση στα δεδομένα.

Για την δημιουργία μια εφαρμογής μηχανικής μάθησης πρέπει να καθορίσουμε την πηγή των δεδομένων, τον τρόπο συλλογής, καθαρισμού, προετοιμασίας και ανάλυσης. Τα δεδομένα είναι συνυφασμένα με το πρόβλημα και τον αλγόριθμο που θα επιλεγεί. Στην συνέχεια καθορίζουμε τα

δεδομένα εισόδου, την εκπαίδευση του αλγόριθμου και τον έλεγχό του. Οι εργασίες αυτές αποτελούν μια διαδικασία που είναι ανατροφοδοτούμενη και εξαρτάται από το πρόβλημα. Αναλυτικότερα, όπως αναφέρεται και στο βιβλίο του Marsland, S., (2014). Machine learning: an algorithmic perspective. CRC press η διαδικασία έχει ως εξής:

- Το αρχικό στάδιο είναι ο προσδιορισμός του προβλήματος και η επιλογή του κατάλληλου αλγόριθμου. Συνήθως το πρόβλημα εντάσσεται σε κάποια κατηγορία προβλημάτων και αυτό μας καθοδηγεί την επιλογή του αλγόριθμου και την οργάνωση των δεδομένων.

- Εφόσον καθοριστεί η προβληματική και ο τρόπος προσέγγισης, προσδιορίζεται το πεδίο συλλογής δεδομένων. Το στάδιο της συλλογής είναι πολύ εκτεταμένο και εξαρτάται κάθε φορά από το αντικείμενο μελέτης. Η προέλευση των δεδομένων μπορεί να είναι οποιαδήποτε, από ιστοσελίδες έως συσκευές που συλλέγουν μετρήσεις ενώ και τα ίδια τα δεδομένα μπορεί να περιέχουν οτιδήποτε. Ένα πολύ σημαντικό και σύγχρονο πεδίο έρευνας αποτελούν τα δημόσια δεδομένα ελεύθερης πρόσβασης (public open data).

- Τα δεδομένα από μια πηγή όπως το διαδίκτυο συχνά δεν είναι αξιοποιήσιμα στην αρχική τους μορφή. Επίσης, υπάρχει περίπτωση να απαιτηθεί ειδική μορφοποίηση για ορισμένους αλγόριθμους καθώς αρκετοί χειρίζονται μόνο συγκεκριμένους τύπους δεδομένων. Η προετοιμασία των δεδομένων αποτελεί το επόμενο βήμα στο οποίο τα δεδομένα αποκτούν μορφή που να είναι αξιοποιήσιμη. Το πλεονέκτημα αυτής της τυποποίησης είναι ότι μπορεί να γίνει μίξη αλγόριθμων και πηγών. Αυτό το στάδιο είναι πιο απλό σε σχέση με τη συλλογή.

- Στη συνέχεια προχωρά η ανάλυση των δεδομένων η οποία περιλαμβάνει τον έλεγχο και επιβεβαίωση ότι τα δεδομένα είναι σε καλή κατάσταση, δεν υπάρχουν ελλείψεις σε τιμές. Ο καθαρισμός αποτελεί το επόμενο βήμα το οποίο, αν απαιτηθεί, προχωρά με ανθρώπινη παρέμβαση για την κάλυψη κενών, τον καθαρισμό των δεδομένων και γενικά εργασίες όπου απομακρύνουν τα ‘σκουπίδια’. Το βασικό σημείο είναι ότι η εργασία δεν μπορεί να προχωρήσει δίχως ανθρώπινη παρέμβαση καθώς πρόκειται για λήψη αποφάσεων ως προς παραμέτρους των δεδομένων.

- Όταν ολοκληρωθούν τα ανωτέρω, επόμενο βήμα είναι εκπαίδευση του αλγορίθμου μας. Εισάγουμε τα δεδομένα μας μέσα στο πρόγραμμα και τα αποτελέσματα που θα παραχθούν τα χρησιμοποιούμε στα μετέπειτα βήματα. Στην περίπτωση της μάθησης χωρίς επίβλεψη (unsupervised learning) δεν υπάρχει βήμα εκμάθησης του αλγόριθμου καθώς δεν υπάρχουν τιμές αναφοράς.

- Όταν ολοκληρωθεί η διαδικασία της εκπαίδευσης του αλγορίθμου, σειρά έχει να ελέγξουμε αν λειτουργεί σωστά. Σε αυτό το σημείο η πληροφορία που αποκτήθηκε από τον αλγόριθμο τίθεται σε εφαρμογή και ελέγχεται ο αλγόριθμος ως προς την επάρκειά του. Στην περίπτωση της μάθησης με επίβλεψη (supervised learning) υπάρχουν ορισμένα σημεία κλειδιά τα οποία

χρησιμοποιούνται για την αξιολόγηση του αλγόριθμου. Αντίθετα, στην περίπτωση της μάθησης χωρίς επίβλεψη δεν υπάρχουν τέτοιες τιμές και απαιτούνται άλλες μετρικές για την αξιολόγηση. Σε κάθε περίπτωση, αν το αποτέλεσμα δεν είναι αποδεκτό, επαναλαμβάνεται το προηγούμενο βήμα και πραγματοποιείται επανέλεγχος. Αν το πρόβλημα εντοπίζεται στα δεδομένα μας τότε ίσως χρειαστεί να επιστρέψουμε στο προηγούμενο βήμα της συλλογής δεδομένων.

- Τελικό στάδιο είναι να ελέγξουμε αν τα αποτελέσματα που προέκυψαν με βάση ολόκληρη την διαδικασία είναι αξιόπιστα έτσι ώστε να μπορέσουμε να τα χρησιμοποιήσουμε σε αληθινά προβλήματα. Ωστόσο αν εμφανιστεί κάποιο πρόβλημα ή σφάλμα, ενδεχομένως να χρειαστεί να ξανά επαναλάβουμε από την αρχή ένα ένα τα βήματα. Σε αυτό το σημείο μπορούμε να εξετάσουμε την δημιουργία ενός προϊόντος που θα καλύπτει ανάλογα προβλήματα.

Αρκετά από τα προβλήματα που συναντάμε στον πραγματικό κόσμο και μπορούν να επιλυθούν με επεξεργασία δεδομένων μπορούν να προδιαγραφούν είτε ως προβλήματα ταξινόμησης ή ως προβλήματα πρόβλεψης. Για τις κατηγορίες αυτές υπάρχει πλήθος αλγόριθμων και μοντέλων που μπορεί να αξιοποιηθεί. Το ζητούμενο από τον επιστήμονα δεδομένων είναι κάθε φορά να είναι σε θέση να αναζητήσει τον κατάλληλο αλγόριθμο ή ενδεχομένως να αναπτύξει ένα νέο. Ένα πρόβλημα που καλείται να διαχειριστεί είναι πώς θα επιλέξει τον καταλληλότερο για το πρόβλημά του. Αυτό θα καθορίσει τα δεδομένα και τον τρόπο συλλογής τους. Ενδεικτικά, ένας τρόπος προσέγγισης είναι ο εξής:

1. Αν το πρόβλημα είναι η πρόβλεψη μιας τιμής στόχου, τότε απαιτείται προσέγγιση μάθησης με επίβλεψη. Σε αυτή την περίπτωση εξετάζεται τι τύπο τιμής αναζητούμε. Αν μπορεί να λάβει διακριτές τιμές τότε αναζητούμε αλγόριθμο ταξινόμησης ενώ αν μπορεί να λάβει συνεχείς τότε χρησιμοποιούμε αλγόριθμο παλινδρόμησης.

2. Αν το πρόβλημα δεν αφορά σε πρόβλεψη κάποιας τιμής στόχου τότε απαιτείται προσέγγιση μάθησης χωρίς επίβλεψη. Αν ο στόχος είναι ο διαχωρισμός των δεδομένων σε διακριτές ομάδες τότε εξυπηρετεί αλγόριθμος ομαδοποίησης. Ενώ, αν ο στόχος είναι κάποια αριθμητική εκτίμηση για το πόσο ισχυρά ανήκει μια τιμή σε μια ομάδα τότε ταιριάζει κάποιος αλγόριθμος πυκνότητας πιθανότητας.

Τα παραπάνω είναι απλοί ενδεικτικοί κανόνες και δεν εξαντλούν σε καμία περίπτωση τις περιπτώσεις που πολλές φορές δεν είναι τόσο ευδιάκριτες. Ένα βασικό σημείο που θα αναδείξει την επιλογή αλγόριθμου είναι τα ίδια τα δεδομένα, το κατά πόσο είναι συνεχή ή διακριτά, αν υπάρχουν κάποιες ιδιαιτερότητες και γενικά αν τα απαιτούν μια συγκεκριμένη προσέγγιση. Με αυτό τον τρόπο επιλέγεται κάποιος αλγόριθμος ή ομάδα αλγόριθμων και στη συνέχεια αναζητείται αυτός που θα είναι αποτελεσματικότερος και ταχύτερος για το πρόβλημα και τους διαθέσιμους πόρους.

2.2. Αλγόριθμοι Ομαδοποίησης (Clustering Algorithms)

Η συσταδοποίηση (cluster analysis) είναι μια τεχνική μηχανικής μάθησης η οποία περιλαμβάνει την ομαδοποίηση σημείων από ολόκληρο το dataset. Παίρνοντας ένα σετ από σημεία δεδομένων, μπορούμε να κατηγοριοποιήσουμε κάθε σημείο σε ένα συγκεκριμένο γκρουπ. Θεωρητικά τα σημεία των δεδομένων που ανήκουν στο ίδιο γκρουπ έχουν παρόμοιες ιδιότητες ή χαρακτηριστικά σε σχέση με άλλα σημεία από το υπόλοιπο dataset που δεν εντάσσονται στην ίδια κατηγορία. Η συσταδοποίηση-ομαδοποίηση είναι μία μέθοδος μάθησης χωρίς επίβλεψη (unsupervised learning) και συνήθως εντάσσεται σε μία από τις συνηθισμένες τεχνικές που χρησιμοποιούνται για στατιστική ανάλυση σε πολλά επιστημονικά πεδία. Στο επόμενο κεφάλαιο θα αναφερθούμε σε ορισμένους αλγόριθμους συσταδοποίησης.

2.2.1. Αλγόριθμος DBSCAN

Ο αλγόριθμος DBSCAN θεωρείται από τους πιο δημοφιλής αλγορίθμους συσταδοποίησης που χρησιμοποιούνται στη μηχανική μάθηση χωρίς επίβλεψη. Κύρια μέθοδος του είναι να ξεχωρίζει συστάδες με υψηλή πυκνότητα από συστάδες με χαμηλή πυκνότητα. Δεδομένου ότι ο DBSCAN είναι ένας αλγόριθμος ομαδοποίησης που βασίζεται στην πυκνότητα, κάνει εξαιρετική δουλειά στο να βρίσκει περιοχές μέσα στα δεδομένα μας όπου παρατηρείται υψηλή πυκνότητα παρατηρήσεων σε σχέση με σημεία όπου δεν είναι πολύ πυκνά μεταξύ τους. Επίσης μία άλλη λειτουργία του DBSCAN είναι ότι μπορεί να ταξινομήσει τα δεδομένα σε διαφορετικές ομάδες σχημάτων.

Σε αυτό σημείο θα περιγράψουμε πως λειτουργεί ο αλγόριθμος. Αρχικά χωρίζουμε το σύνολο δεδομένων σε n διαστάσεις . Στη συνέχεια παίρνουμε τυχαία ως κεντρικό σημείο ένα οποιοδήποτε σημείο του dataset το οποίο το ονομάζουμε πυρήνα (core point) και για κάθε παρατήρηση που βρίσκεται σε ακτίνα που ισούται με ε, σχηματίζει σχήμα γύρω από την περιοχή του. Μετέπειτα υπολογίζει πόσα σημεία βρίσκονται μέσα σε αυτή την περιοχή και όλα μαζί φτιάχνουν ένα cluster. Η διαδικασία ολοκληρώνεται μέχρις ότου δεν υπάρχει κάποιο σημείο στην περιοχή και συνεχίζει με την ίδια φιλοσοφία με τα επόμενα σημεία για να σχηματίσει τα υπόλοιπα clusters. Σε περίπτωση που κάποιο σημείο δεν «συνορεύει» με κάποιο άλλο τότε αυτό λογίζεται ως ακραίο σημείο και δεν συμπεριλαμβάνεται μέσα στο cluster. Παρακάτω παρουσιάζεται ο ψευδοκώδικας του DBSCAN όπως αναφέρεται στο κεφάλαιο 6 Συσταδοποίηση του βιβλίου(repository.kallipos.gr/bitstream/11419/2972/1/02_chapter_06.pdf) :

```
DBSCAN(D, eps, MinPts) {
    C = 0
    για κάθε σημείο P στη βάση D {
        αν το P είναι μαρκαρισμένο
```

```
                συνέχισε με το επόμενο σημείο
            μαρκάρισε το P
            NeighborPts = ερώτημαΠεριοχής(P, eps)
            αν πλήθος(NeighborPts) < MinPts
                μαρκάρισε το P ως θόρυβο
            αλλιώς {
                C = επόμενη συστάδα
                επέκτασηΣυστάδας(P, NeighborPts, C, eps, MinPts)
            }
        }
}



επέκτασηΣυστάδας(P, NeighborPts, C, eps, MinPts) {
        πρόσθεσε το P στη συστάδα C
        για κάθε σημείο P' στο σύνολο NeighborPts {
            αν το P δεν είναι μαρκαρισμένο {
                μαρκάρισε το P'
                NeighborPts' = ερώτημαΠεριοχής(P', eps)
                αν πλήθος(NeighborPts') >= MinPts
                    NeighborPts = NeighborPts U NeighborPts'
            }
            αν το P' δεν ανήκει ήδη σε κάποια συστάδα
                πρόσθεσε το P' στη συστάδα C
        }
}
ερώτημαΠεριοχής(P, eps)
        επέστρεψε όλα τα σημεία στην ε-γειτονιά του P (συμπεριλαμβανομένου και του P)
```

Τα βασικότερα πλεονεκτήματα του αλγορίθμου DBSCAN είναι τα ακόλουθα:

- Δεν απαιτείται εκ των προτέρων ορισμός του αριθμού συστάδων
- Καταλήγει σε αυθαίρετα σχήματα συστάδων
- Δέχεται ως είσοδο μόνο δύο παραμέτρους
- Πολύ καλή ευαισθησία στο θόρυβο και τις ακραίες τιμές

Αντίστοιχα τα μειονεκτήματα του DBSCAN είναι τα εξής:

- Τα αποτελέσματα εξαρτώνται από την μετρική απόστασης που θα χρησιμοποιηθεί
- Δεν ομαδοποιεί καλά δεδομένα με διαφορές στην πυκνότητα

2.2.2 K-Means

Ο αλγόριθμος k-means είναι ένας βασικός και αρκετά διαδεδομένος αλγόριθμος μη επιβλεπόμενης μάθησης ο οποίος έχει ως στόχο την εύρεση ομάδων δεδομένων χωρίς να υπάρχουν προκαθορισμένες κατηγορίες δεδομένων. Ο k-means αναζητά k ομάδες (clusters) για ένα σύνολο δεδομένων. Ο αριθμός k ορίζεται από τον χρήστη του αλγόριθμου και κάθε ομάδα περιγράφεται από ένα σημείο το οποίο ονομάζεται κεντροειδές (centroid). Το σημείο αυτό θεωρείται το κέντρο της ομάδας των σημείων.

Ο αλγόριθμος λειτουργεί ως εξής. Αρχικά, καθορίζεται ο αριθμός των κέντρων k, ο αλγόριθμός επιλέγει k τυχαία σημεία από το σύνολο δεδομένων ως κεντροειδή και ορίζει k ομάδες. Στη συνέχεια, κάθε σημείο του συνόλου δεδομένων εντάσσεται σε μια ομάδα. Η ένταξη πραγματοποιείται με την εύρεση του πλησιέστερου κεντροειδούς και την ένταξη του σημείου σε αυτή την ομάδα. Αυτή η διαδικασία επαναλαμβάνεται για όλα τα σημεία του συνόλου δεδομένων και για κάθε κεντροειδές. Όταν το βήμα ολοκληρωθεί, τα κεντροειδή ενημερώνονται με τον υπολογισμό των μέσων τιμών από όλα τα σημεία της κάθε ομάδας. Στη συνέχεια τα σημεία των μέσων τιμών γίνονται τα νέα κεντροειδή και η διαδικασία αρχίζει από την αρχή. Ο αλγόριθμος τερματίζει όταν κανένα από τα σημεία δεν αλλάζει την ομάδα του. Ο τερματισμός μπορεί να οριστεί με μια συνθήκη τερματισμού έπειτα από καθορισμένο αριθμό επαναλήψεων. Ένα βασικό σημείο του αλγόριθμου είναι η μετρική της απόστασης που χρησιμοποιείται για τον υπολογισμό της απόστασης των σημείων από το κεντροειδές. Μια συνηθισμένη μετρική είναι η Ευκλείδεια απόσταση. Παρακάτω παρουσιάζεται ο ψευδοκώδικας του k-means από το βιβλίο (repository.kallipos.gr/bitstream/11419/2972/1/02_chapter_06.pdf):

Αρχικοποίησε τυχαία τα k κεντροειδή των συστάδων $\mu_1, \mu_2, \ldots, \mu_k$.

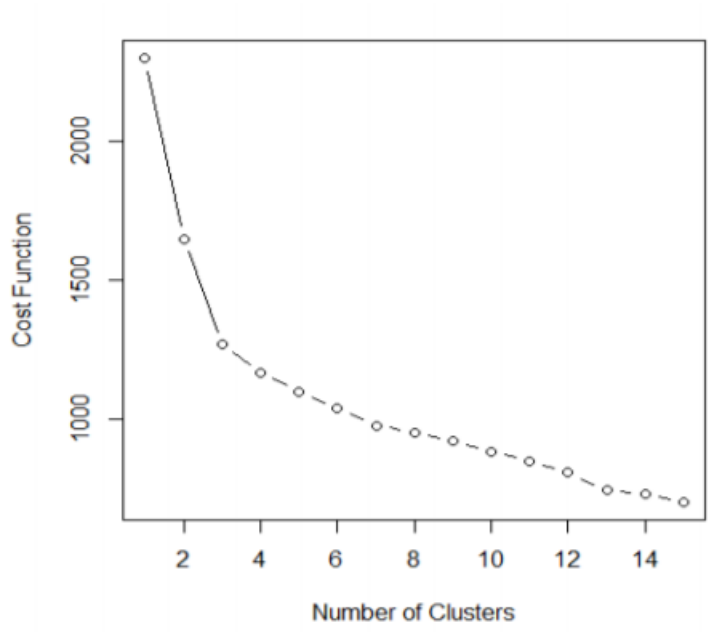
Επανέλαβε{

Εξέτασε κάθε δείγμα και ανέθεσε το στη συστάδα με το πλησιέστερο κεντροειδές ($\min \|x^{(i)} - \mu_k\|^2$)

Επανυπολόγισε τα κεντροειδή υπολογίζοντας το μέσο όρο των δειγμάτων της συστάδας

}

Δυστυχώς, για την επιλογή του αριθμού των συστάδων δεν υπάρχει κάποιος γενικός κανόνας, ο οποίος να λειτουργεί εγγυημένα και για όλες τις περιπτώσεις. Ένα απλό και πρακτικό τέχνασμα, το οποίο μπορεί να βοηθήσει σε ορισμένες περιπτώσεις, είναι «ο κανόνας του αγκώνα» (the elbow rule). Στην παρακάτω ο κανόνας του αγκώνα υποδεικνύει ότι η επιλογή k=3 είναι αρκετά καλή. Ωστόσο, υπάρχουν περιπτώσεις, όπου η γραφική είναι πιο ομαλή και δεν έχει τον τύπο σχήματος του αγκώνα, με αποτέλεσμα η επιλογή και πάλι να μην είναι ξεκάθαρη.



Τα βασικότερα πλεονεκτήματα του K-Means είναι τα εξής:

- Είναι απλός και εύχρηστος
- Αυτόματη κατανομή των παρατηρήσεων σε συστάδες
- Ταχύτητα σύγκλισης

Τέλος τα μειονεκτήματα του K-Means είναι τα ακόλουθα:

- Ορισμός αριθμού συστάδων
- Υποχρεωτική κατανομή των παρατηρήσεων σε συστάδες
- Δουλεύει μόνο για αριθμητικά δεδομένα

ΜΕΡΟΣ 3: ΠΕΙΡΑΜΑΤΙΚΟ ΣΚΕΛΟΣ

3.1. Κεντρική Ιδέα

Ο κύριος στόχος του τρίτου μέρους είναι να εξετάσουμε τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος του K-Means και να αναλύσουμε τα αποτελέσματα που παράγει όταν εκτελείται. Αρχικά θα γίνει μία σύντομη περιγραφή των τεχνολογιών που θα χρησιμοποιήσουμε. Το βασικό μας εργαλείο είναι η γλώσσα προγραμματισμού Python και το Apache Spark, το οποίο είναι ένα πλαίσιο λογισμικών (framework) κατάλληλο για υποστήριξη υπολογισμών σε clusters υπολογιστικών συστημάτων. Στη συνέχεια θα περιγράψουμε τον τρόπο υλοποίησης του K-Means με διαφορετικά datasets σε python και pyspark και τέλος θα γίνει σύγκριση των αποτελεσμάτων.

3.1.1. Python

Η Python είναι μια διερμηνευόμενη γλώσσα προγραμματισμού γενικής χρήσης, η οποία έχει συχνά το ρόλο γλώσσας εκτέλεσης σεναρίων (scripting language). Επίσης, προσδιορίζεται ως αντικειμενοστρεφής γλώσσα εκτέλεσης σεναρίων (object-oriented scripting language), κάτι που συνδυάζει το γεγονός ότι υποστηρίζει αντικειμενοστρεφή προγραμματισμό και ταυτόχρονα έχει και τον ρόλο γλώσσας εκτέλεσης σεναρίων. Η Python είναι μια σύγχρονη γλώσσα προγραμματισμού, η οποία αναπτύχθηκε από τον Guido van Rossum τη δεκαετία του 1990 και ονομάστηκε έτσι από τη διάσημη κωμική ομάδα των Monty Pythons.

Τα βασικότερα χαρακτηριστικά της Python είναι τα ακόλουθα:

- **Είναι γλώσσα υψηλού επιπέδου:** Η Python είναι γλώσσα υψηλού επιπέδου και προσφέρει ένα υψηλό επίπεδο αφαίρεσης, ώστε ο προγραμματιστής δεν απαιτείται να έρθει σε επαφή με χαμηλού επιπέδου λειτουργίες, όπως διαχείριση μνήμης, καθαρισμός αντικειμένων και θέματα του λειτουργικού περιβάλλοντος γενικότερα.
- **Είναι διερμηνευόμενη:** Ένα πρόγραμμα σε Python δεν μεταγλωττίζεται σε δυαδικό αρχείο αλλά εκτελείται απευθείας από τον πηγαίο του κώδικα.
- **Υποστηρίζει διαδικαστικό, αντικειμενοστρεφή και λειτουργικό προγραμματισμό:** Η Python είναι αντικειμενοστρεφής γλώσσα και υποστηρίζει τόσο διαδικασιακό προγραμματισμό (procedure-oriented) όσο και αντικειμενοστρεφή προγραμματισμό (object-oriented).
- **Είναι λογισμικό ανοικτού κώδικα:** Η Python είναι Λογισμικό Ανοικτού Κώδικα. Δηλαδή επιτρέπεται η εγκατάσταση, διανομή, αντιγραφή πηγαίου κώδικα, μεταβολές και χρήση της σε νέα προγράμματα ελεύθερα.
- **Είναι φορητή**. Η τυπική διανομή της Python στηρίζεται στη γλώσσα ANSI C και μπορεί να εκτελεστεί σε σχεδόν όλες τις υφιστάμενες πλατφόρμες.
- **Είναι εξαιρετικά ισχυρή:** Η Python με βάση τα χαρακτηριστικά της έχει υβριδικό χαρακτήρα. Από τη μια πλευρά, τοποθετείται κοντά στις παραδοσιακές γλώσσες εκτέλεσης σεναρίων (όπως Tcl, Scheme, and Perl) και, από την άλλη, κοντά σε παραδοσιακές γλώσσες ανάπτυξης εφαρμογών (όπως C, C++, και Java). Αυτός ο συνδυασμός τής δίνει το πλεονέκτημα ότι μπορεί να χρησιμοποιηθεί σε προγραμματισμό εφαρμογών μεγάλης κλίμακας και ταυτόχρονα να διατηρεί την αμεσότητα και ευκολία χρήσης μιας γλώσσας σεναρίων.
- **Έχει την δυνατότητα σύνδεσης και ανάμιξης με διαφορετικές γλώσσες:** Τα προγράμματα σε Python είναι εύκολο να συνδεθούν με αντικείμενα που έχουν δημιουργηθεί με διαφορετικές γλώσσες προγραμματισμού, όπως για παράδειγμα η γλώσσα C και η C++. Η ευελιξία αυτή είναι δυνατή με τη χρήση διεπαφών (για παράδειγμα το Python C API), με τη βοήθεια των οποίων ένα πρόγραμμα Python είναι δυνατό να καλέσει και να κληθεί από προγράμματα σε διαφορετική γλώσσα.
- **Είναι απλή και εύκολη στην εκμάθηση:** Η Python είναι πολύ απλή και σχεδόν μινιμαλιστική γλώσσα. Η ανάγνωση ενός καλοδομημένου προγράμματος σε Python είναι σαν ανάγνωση κειμένου καθώς σε μεγάλο βαθμό μοιάζει με ψευδοκώδικα.

Όταν πρόκειται για εφαρμογές στον κλάδο του data science, η Python είναι ένα πολύ ισχυρό εργαλείο. Όπως είπαμε και προηγουμένως πολύ βασικό χαρακτηριστικό της το οποίο την κάνει πολύ δημοφιλή είναι ότι πρόκειται για γλώσσα ανοιχτού κώδικα. Διαθέτει εξαιρετικές βιβλιοθήκες για την επεξεργασία δεδομένων και είναι σχετικά εύκολη στην εκμάθησή της.

Η python έχει μεγάλη ποικιλία από βιβλιοθήκες, όπου οι πιο διαδεδομένες είναι οι εξής:

- NumPy, για αριθμητικούς υπολογισμούς και κυρίως για τις πράξεις μεταξύ πινάκων
- Pandas, όπου με τη βοήθεια των πλαισίων δεδομένων (Dataframes) μπορούμε να επεξεργαστούμε τα δεδομένα μας
- Matplotlib, όπου μπορούμε να αναπαραστήσουμε τα δεδομένα μας με τη βοήθεια γραφημάτων

- Scikit-learn, για εφαρμογή αλγορίθμων στη μηχανική μάθηση

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

Όπως όλες οι γλώσσες προγραμματισμού, έτσι και η Python δεν είναι το εργαλείο που ταιριάζει σε όλα τα προβλήματα. Τα βασικά της μειονεκτήματα είναι τα εξής:

- **Ταχύτητα:** Η Python δεν είναι πλήρως μεταγλωττίσιμη γλώσσα. Αλλά, αρχικά μεταγλωττίζεται μερικώς σε μια μορφή δυαδικού κώδικα, ο οποίος στη συνέχεια εκτελείται από τον διερμηνευτή Python. Αυτό δημιουργεί μια υστέρηση σε ταχύτητα και γενικά μπορούμε να πούμε ότι σε σχέση με μια γλώσσα όπως η C τα αντίστοιχα προγράμματα σε Python εκτελούνται σχετικά βραδύτερα.
- **Βιβλιοθήκες:** Αν και η Python περιλαμβάνει, όπως αναφέραμε, ευρεία συλλογή βιβλιοθηκών, υπάρχουν γλώσσες όπως η Java ή η Perl που περιλαμβάνουν ακόμη εκτενέστερες συλλογές βιβλιοθηκών και σε μερικές εξειδικευμένες περιπτώσεις η Python δεν έχει αντίστοιχη βιβλιοθήκη.
- **Έλεγχος τύπου μεταβλητών κατά την εκτέλεση:** Στην Python οι μεταβλητές λειτουργούν περισσότερο ως ετικέτες οι οποίες αναφέρονται σε διάφορα αντικείμενα, όπως ακέραιοι, συμβολοσειρές, κλάσεις, ή οτιδήποτε άλλο. Αυτό σημαίνει ότι δεν συσχετίζονται με το συγκεκριμένο τύπο αλλά ότι απλά αναφέρονται σε αυτό, παρόλο που τα αντικείμενα έχουν κάποιο τύπο. Η λειτουργία αυτή, η συσχέτιση δηλαδή των τύπων με τα αντικείμενα και όχι με τις μεταβλητές, έχει ως αποτέλεσμα να μην είναι εφικτή η εύρεση σφαλμάτων του τύπου μιας μεταβλητής από το διερμηνευτή. Παρόλο που θεωρείται μειονέκτημα, συχνά εκλαμβάνεται και ως ευελιξία που είναι αποδεκτή και χρήσιμη.

3.1.2. Apache Spark

Το Apache Spark αποτελεί μία ισχυρή ,ανοιχτού κώδικα μηχανή επεξεργασίας δεδομένων που βασίζεται στην ταχύτητα , την ευκολία χρήσης και τις εξελιγμένες δυνατότητες ανάλυσης δεδομένων μεγάλης κλίμακας(Big Data). Αναπτύχθηκε το 2009 από το πανεπιστήμιο Berkeley της California. Αναλυτικότερα το Spark είναι ένα γρήγορο ,γενικού σκοπού και ανεκτικό σε σφάλματα cluster computing σύστημα. Παρέχει APIs υψηλού επιπέδου σε γλώσσα Java,Python,Scala και R ,καθώς και ένα βελτιστοποιημένο μηχανισμό που υποστηρίζει ως επί το πλείστον την εκτέλεση-απεικόνιση γράφων. Επιπροσθέτως υποστηρίζει ένα πλούσιο σύνολο βιβλιοθηκών(libraries) υψηλότερου επιπέδου, όπως το Spark SQL για SQL και επεξεργασία δομημένων δεδομένων, την βιβλιοθήκη MLlib που προσφέρεται για μηχανική μάθηση , το πακέτο GraphX για επεξεργασία γράφων και το Spark Streaming.

3.1.2.1. Spark Core

Το Spark Core είναι ο πυρήνας του Spark και ουσιαστικά είναι η μηχανή που στεγάζει όλες τις λειτουργίες του Spark συμπεριλαμβανομένων των στοιχείων για χρονοπρογραμματισμό εργασιών ,διαχείριση μνήμης ,αλληλεπίδραση με τα συστήματα αποθήκευσης και πολλά άλλα. Επίσης υποστηρίζει το API που ορίζει το RDD(Resilient Distributed Dataset) ,το οποίο αποτελεί την κύρια οντότητα δεδομένων του Spark.Ένα RDD είναι μία συλλογή στοιχείων που έχουν ανοχή σε σφάλματα και τα οποία έχουν την ικανότητα να επεξεργάζονται παράλληλα τα δεδομένα. Τα χαρακτηριστικά των RDDs όπως αναγράφονται στο βιβλίο MLlib: Scalable Machine Learning on Spark είναι τα εξής:

- **Resilient:** Ανεκτικά σε σφάλματα με τη βοήθεια ενός γράφου ζωής σε κάθε RDD. Είναι κατά αυτόν τον τρόπο ικανά να υπολογίσουν εκ νέου κομμάτια των δεδομένων που λείπουν ή έχουν υποστεί ζημιά εξαιτίας αποτυχιών ενός κόμβου.
- **Distributed:** Τα δεδομένα βρίσκονται διαμοιρασμένα σε πολλούς κόμβους ενός cluster.
- **Dataset:** Μια συλλογή δεδομένων που είναι χωρισμένη σε κομμάτια.

3.1.2.2. Ενσωματωμένες Βιβλιοθήκες

Spark SQL

Το Spark SQL αποτελεί ένα κομμάτι του Spark για την επεξεργασία δομημένων δεδομένων. Παρέχει ένα αφαιρετικό επίπεδο προγραμματισμού ,τα Dataframes και μπορεί επίσης να λειτουργήσει ως ένα κατανεμημένο σύστημα υποβολής ερωτημάτων SQL.Επιπλέον το Spark προσφέρει τη δυνατότητα συνδυασμού των προγραμματιστικών δυνατοτήτων του RDD με τη δυνατότητα υποβολής ερωτημάτων SQL.

MLib

Το Spark περιλαμβάνει μία βιβλιοθήκη η οποία ονομάζεται MLlib και παρέχει αλγορίθμους μηχανικής μάθησης συμπεριλαμβανομένων της ταξινόμησης, της ομαδοποίησης, της παλινδρόμησης και του συνδυαστικού φιλτραρίσματος. Επίσης η βιβλιοθήκη μπορεί να χρησιμοποιηθεί ως μέρος εφαρμογών του Spark ,οι οποίες είναι γραμμένες σε Java,Python ή Scala.

Spark Streaming

Το Spark Streaming αποτελεί μια βιβλιοθήκη του Spark που επιτρέπει την επεξεργασία ροών δεδομένων. Είναι ένα πολύ σημαντικό πακέτο δεδομένου ότι πολλές χρειάζονται την ικανότητα επεξεργασίας και ανάλυσης ροών δεδομένων σε πραγματικό χρόνο. Οι ροές δεδομένων θα μπορούσαν να περιλαμβάνουν αρχεία καταγραφής ενός διακομιστή, ουρές μηνυμάτων με ενημερώσεις κατάστασης που δημοσιεύονται από χρήστες μιας υπηρεσίας ιστού. Πιο συγκεκριμένα το Spark Streaming παρέχει ένα API για το χειρισμό ροών δεδομένων που ταιριάζει απόλυτα με το RDD API του Spark Core και συνδυάζεται εύκολα με μια μεγάλη ποικιλία δημοφιλών πηγών δεδομένων όπως το Twitter.

GraphX

Το GraphX αποτελεί μία βιβλιοθήκη του Spark η οποία δίνει τη δυνατότητα επεξεργασίας και χειρισμού γράφων ,καθώς επίσης και την εκτέλεση παράλληλων υπολογισμών σε αυτούς. Επιπροσθέτως επεκτείνει το Spark RDD API παρέχοντας έτσι τη δυνατότητα δημιουργίας ενός κατευθυνόμενου γράφου με αυθαίρετες ιδιότητες προσαρτημένες σε κάθε κορυφή και ακμή αυτού.

3.2. Μοντελοποίηση K-Means

Σε αυτό το σημείο θα περάσουμε στην παρουσίαση της υλοποίησης του K-Means που έγινε στην python και το pyspark ξεχωριστά. Η έκδοση της python που χρησιμοποιήθηκε είναι η 3.7 και το περιβάλλον ανάπτυξης του κώδικα είναι το Spyder. Αντίστοιχα για την υλοποίηση σε pyspark χρησιμοποιήθηκε η έκδοση 2.4.6 του spark και η έκδοση 3.5 της python ενώ το IDE περιβάλλον είναι το Jupyter Notebook.

Τα δεδομένα που θα χρησιμοποιήσουμε βρίσκονται ανεβασμένα στον ιστότοπο https://support.spatialkey.com/spatialkey-sample-csv-data/ . Η περιγραφή της υλοποίησης θα γίνει για το dataset Crime Records (SacramentocrimeJanuary2006.csv) και μόλις τελειώσουμε θα παρουσιάσουμε τα αποτελέσματα και για τα υπόλοιπα dataset. Το συγκεκριμένο σύνολο δεδομένων περιέχει πληροφορίες για εγκλήματα που έγιναν τον Γενάρη του 2006 στην πόλη Sacramento της Αμερικής. Είναι της μορφής csv , έχει μέγεθος 775 Kilobytes και έχει συνολικά 7584 εγγραφές. Οι στήλες έχουν τα ονόματα cdatetime,address,district,beat,grid,crimedescr,ucr_ncic_code,latitude και longitude. Εμείς θα ασχοληθούμε με τις στήλες latitude και longitude που είναι οι συντεταγμένες του τόπου που διαπράχθηκε το κάθε έγκλημα ξεχωριστά.

3.2.1. Υλοποίηση σε Python

Αρχικά θα ξεκινήσουμε με την εισαγωγή των βιβλιοθηκών που θα χρειαστούμε για την υλοποίηση σε python.

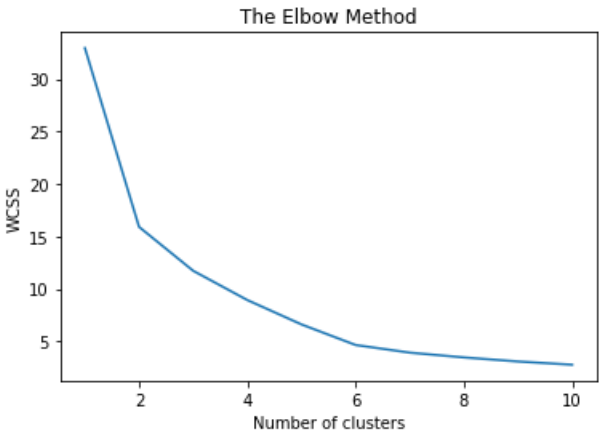
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Στη συνέχεια εισάγουμε το dataset και το διαμορφώνουμε καταλλήλως έτσι ώστε να είναι επιλεγμένες οι στήλες που μας ενδιαφέρουν, στην προκειμένη περίπτωση οι latitude και longitude.

```
dataset = pd.read_csv('SacramentocrimeJanuary2006.csv')
X = dataset.iloc[:, [7, 8]].values
```

Σειρά έχει η εισαγωγή του πακέτου KMeans της βιβλιοθήκης sklearn.cluster , ο υπολογισμός του wcss (άθροισμα τετραγώνων της συστάδας) καθώς επίσης και η οπτικοποίση του ,προκειμένου να μπορέσουμε μέσω της elbow method (ο κανόνας του αγκώνα) να βρούμε το k εκείνο που ομαδοποιεί καλύτερα τα δεδομένα μας.

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

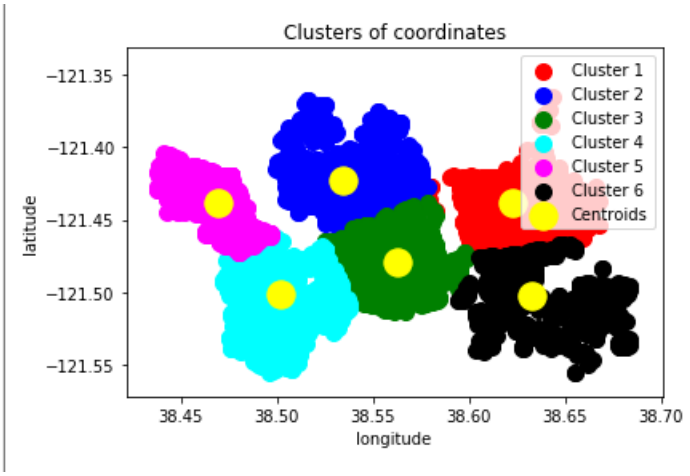


Παρατηρούμε ότι για k=6 το wcss μειώνεται πολύ λίγο συνεπώς αυτό θα θεωρήσουμε ως δεδομένο και θα προχωρήσουμε με το να κάνουμε train τα δεδομένα μας στον KMeans.

```
kmeans = KMeans(n_clusters = 6, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)
```

Τέλος ήρθε η ώρα να οπτικοποιήσουμε τα αποτελέσματά μας και να δούμε πως έχουν ομαδοποιηθεί τα δεδομένα μας.

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(X[y_kmeans == 5, 0], X[y_kmeans == 5, 1], s = 100, c = 'black', label = 'Cluster 6')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroids')
plt.title('Clusters of coordinates')
plt.xlabel('longitude')
plt.ylabel('latitude')
plt.legend()
plt.show()
```



3.2.2 Υλοποίηση στο Spark (Pyspark)

Αφού λοιπόν ολοκληρώθηκε η περιγραφή της υλοποίησης στην Python σειρά έχει να δούμε ένα ένα τα βήματα που ακολουθήσαμε και στο Spark.

Αρχίζουμε με την εισαγωγή της βιβλιοθήκης findspark και πληκτρολογούμε την εντολή import findspark ,η οποία είναι απαραίτητη για να μπορούμε να ξεκινήσουμε να γράφουμε σε spark απο το home του υπολογιστή μας. Σε αντίθετη περίπτωση θα έπρεπε να μπούμε μέσα από το cmd στο φάκελο που βρίσκεται εγκατεστημένο το spark και από εκεί να αρχίσουμε να γράφουμε κανονικά και να κάνουμε import βιβλιοθήκες όπως αυτή του SparkSession στην τέταρτη γραμμή της εικόνας μας. Έπειτα πληκτρολογούμε την εντολή findspark.init('home/nick/spark-2.4.6-bin-hadoop2.7') για να ξεκινήσουμε το spark. Στη συνέχεια εισάγουμε τις βιβλιοθήκες SparkSession και pyspark.H SparkSession είναι απαραίτητη στη δημιουργία ενός καινούριου application ,όπου στη δική μας περίπτωση την ονομάσαμε cluster.

```
In [1]: import findspark
        findspark.init('/home/nick/spark-2.4.6-bin-hadoop2.7')
        import pyspark
        from pyspark.sql import SparkSession
        spark = SparkSession.builder.appName('cluster').getOrCreate()
```

Το spark έχει πολλές επιλογές για να μπορέσει κάποιος να διαβάσει τα δεδομένα του σε όλα τα format. Συγκεκριμένα για το δικό μας format(csv) θα χρησιμοποιήσουμε την εντολή df=spark.read.csv η οποία μας επιτρέπει να διαβάσουμε το csv αρχείο μας. Οι παράμετροι inferSchema=True,header=True χρησιμοποιούνται για να διατηρηθεί το σχήμα των δεδομένων μας. Το df υποδηλώνει την ονομασία του dataframe μας. Ταυτόχρονα εισάγουμε τη βιβλιοθήκη KMeans η οποία είναι απαραίτητη για να τρέξουμε τον αλγόριθμο.

```
from pyspark.ml.clustering import KMeans
df = spark.read.csv('SacramentocrimeJanuary2006.csv',header=True,inferSchema=True)
```

```
df_feat = df.select(*(df[c].cast("float").alias(c) for c in df.columns[1:]))
df_feat.show()
```

```
+-------+--------+----+------+----------+-------------+---------+-----------+
|address|district|beat|  grid|crimedescr|ucr_ncic_code| latitude|  longitude|
+-------+--------+----+------+----------+-------------+---------+-----------+
|   null|     3.0|null|1115.0|      null|       2404.0| 38.55042| -121.39142|
|   null|     5.0|null|1512.0|      null|       2204.0|  38.4735| -121.49019|
|   null|     2.0|null| 212.0|      null|       2404.0|38.657845|  -121.4621|
|   null|     6.0|null|1443.0|      null|       2501.0|38.506775| -121.42695|
|   null|     2.0|null| 508.0|      null|       2299.0|38.637447| -121.38461|
|   null|     6.0|null|1084.0|      null|       2604.0|38.526978| -121.45134|
|   null|     4.0|null| 957.0|      null|       2299.0|38.537174| -121.48758|
|   null|     3.0|null| 853.0|      null|       2308.0|38.564335| -121.46188|
|   null|     2.0|null| 508.0|      null|       2203.0|38.637447| -121.38461|
|   null|     1.0|null| 444.0|      null|       2310.0|38.609604| -121.49184|
|   null|     6.0|null|1005.0|      null|       7000.0|38.554264|-121.454605|
|   null|     6.0|null|1088.0|      null|       2604.0|38.528164| -121.43145|
|   null|     4.0|null|1261.0|      null|       7000.0| 38.51092| -121.54882|
|   null|     3.0|null| 888.0|      null|       2604.0|38.556114|-121.414276|
|   null|     6.0|null|1447.0|      null|       2605.0| 38.50398|-121.392395|
|   null|     6.0|null|1054.0|      null|       2303.0| 38.54153| -121.44951|
|   null|     6.0|null|1403.0|      null|       7000.0|38.516575| -121.42348|
|   null|     3.0|null| 742.0|      null|       2308.0|38.581844| -121.50117|
|   null|     6.0|null|1034.0|      null|       2604.0| 38.54271| -121.45721|
|   null|     4.0|null|1225.0|      null|       2605.0|  38.5246| -121.52036|
+-------+--------+----+------+----------+-------------+---------+-----------+
```

Στην παραπάνω εικόνα μετονομάσαμε την ονομασία του dataset μας σε df_feat και με την εντολή df.select επιλέγουμε όλες της στήλες του dataset πλην της στήλης cdatetime. Έπειτα με την εντολή df_feat.show() έχουμε μία πρώτη εικόνα με τις πρώτες είκοσι παρατηρήσεις.

Συνεχίζουμε με τη δημιουργίας μιας νέας μεταβλητής FEATURES_COL, όπου εισάγουμε ως παραμέτρους τις στήλες latitude και longitude και στη συνέχεια μετατρέπουμε σε πραγματικούς αριθμούς τις τιμές των παρατηρήσεων του αρχικού dataset df.

```
FEATURES_COL = ['latitude', 'longitude']
for col in df.columns:
    if col in FEATURES_COL:
        df = df.withColumn(col,df[col].cast('float'))
df.show()
```

```
+-----------+--------------------+--------+----------+----+--------------------+-------------+---------+----------
-+
|  cdatetime|             address|district|      beat|grid|          crimedescr|ucr_ncic_code| latitude|  longitud
e|
+-----------+--------------------+--------+----------+----+--------------------+-------------+---------+----------
-+
|1/1/06 0:00|  3108 OCCIDENTAL DR|       3|3C        |1115|10851(A)VC TAKE V...|         2404| 38.55042| -121.3914
2|
|1/1/06 0:00| 2082 EXPEDITION WAY|       5|5A        |1512|459 PC  BURGLARY ...|         2204|  38.4735| -121.4901
9|
|1/1/06 0:00|          4 PALEN CT|       2|2A        | 212|10851(A)VC TAKE V...|         2404|38.657845|  -121.462
1|
|1/1/06 0:00|      22 BECKFORD CT|       6|6C        |1443|476 PC PASS FICTI...|         2501|38.506775| -121.4269
5|
|1/1/06 0:00|    3421 AUBURN BLVD|       2|2A        | 508|459 PC  BURGLARY-...|         2299|38.637447| -121.3846
1|
|1/1/06 0:00|  5301 BONNIEMAE WAY|       6|6B        |1084|530.5 PC USE PERS...|         2604|38.526978| -121.4513
4|
|1/1/06 0:00|       2217 16TH AVE|       4|4A        | 957|459 PC  BURGLARY ...|         2299|38.537174| -121.4875
8|
|1/1/06 0:00|           3547 P ST|       3|3C        | 853|484 PC   PETTY TH...|         2308|38.564335| -121.4618
8|
|1/1/06 0:00|    3421 AUBURN BLVD|       2|2A        | 508|459 PC  BURGLARY ...|         2203|38.637447| -121.3846
1|
|1/1/06 0:00|   1326 HELMSMAN WAY|       1|1B        | 444|1708 US   THEFT O...|         2310|38.609604| -121.4918
```

Επόμενο βήμα είναι η εισαγωγή της βιβλιοθήκης VectorAssembler.Δημιουργούμε ένα διάνυσμα (vecAssembler) το οποίο θα δέχεται(inputCols) ως ετικέτες τη μεταβλητή που ορίσαμε πριν την FEATURES_COL και θα τις μετασχηματίζει(outputCol) σε μία ετικέτα με το όνομα features. Τώρα που φτιάξαμε το διάνυσμα σειρά έχει η δημιουργία μιας νέας μεταβλητής με το όνομα df_kmeans η οποία δέχεται το διάνυσμα και σε συνδυασμό με τη μέθοδο transform() μας επιτρέπει να διαμορφώσουμε τα τελικά δεδομένα.

```
from pyspark.ml.feature import VectorAssembler
vecAssembler = VectorAssembler(inputCols=FEATURES_COL, outputCol="features")
df_kmeans = vecAssembler.transform(df).select('ucr_ncic_code', 'features')
df_kmeans.show()
```

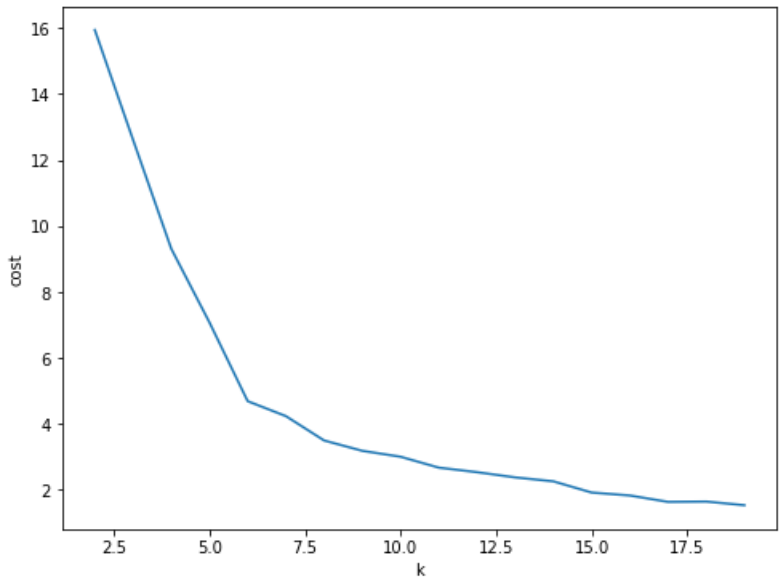
```
+-------------+--------------------+
|ucr_ncic_code|            features|
+-------------+--------------------+
|         2404|[38.5504188537597...|
|         2204|[38.4734992980957...|
|         2404|[38.6578445434570...|
|         2501|[38.5067749023437...|
|         2299|[38.6374473571777...|
|         2604|[38.5269775390625...|
|         2299|[38.5371742248535...|
|         2308|[38.5643348693847...|
|         2203|[38.6374473571777...|
|         2310|[38.6096038818359...|
|         7000|[38.5542640686035...|
|         2604|[38.5281639099121...|
|         7000|[38.5109214782714...|
|         2604|[38.5561141967773...|
|         2605|[38.5039787292480...|
|         2303|[38.5415306091308...|
|         7000|[38.5165748596191...|
|         2308|[38.5818443298339...|
|         2604|[38.5427093505859...|
|         2605|[38.5246009826660...|
+-------------+--------------------+
only showing top 20 rows
```

Έχουμε φτάσει στο σημείο όπου έχουμε την τελική μορφή των δεδομένων μας και θέλουμε να εφαρμόσουμε τον k-means πάνω σε αυτά. Για να γίνει αυτό θα πρέπει πρώτα να υπολογίσουμε το wcss όπως ακριβώς κάναμε και προηγουμένως με την Python.

```
import numpy as np
cost = np.zeros(20)
for k in range(2,20):
    kmeans = KMeans().setK(k).setSeed(1).setFeaturesCol("features")
    model = kmeans.fit(df_kmeans.sample(False,0.1, seed=42))
    cost[k] = model.computeCost(df_kmeans)
```

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(1,1, figsize =(8,6))
ax.plot(range(2,20),cost[2:20])
ax.set_xlabel('k')
ax.set_ylabel('cost')
```

```
Text(0, 0.5, 'cost')
```



Στην παραπάνω εικόνα φαίνεται ξεκάθαρα ότι το k=6. Ακριβώς το ίδιο αποτέλεσμα που είχαμε και στην υλοποίηση της Python. Η διαδικασία από αυτό το σημείο είναι σχεδόν παρόμοια με πριν . Κάνουμε train τα δεδομένα μας στον KMeans για k=6, εμφανίζουμε τα αποτελέσματα με τα centroids και δημιουργούμε μια νέα μεταβλητή με το όνομα rows και παραμέτρους τη στήλη ucr_ncir_code και το prediction , όπου στην ουσία πρόκειται για την πρόβλεψη του cluster στο οποίο ανήκει η κάθε παρατήρηση.

```
k = 6
kmeans = KMeans().setK(k).setSeed(1).setFeaturesCol("features")
model = kmeans.fit(df_kmeans)
centers = model.clusterCenters()

print("Cluster Centers: ")
for center in centers:
    print(center)
```

```
Cluster Centers:
[  38.5563585  -121.48491383]
[  38.47857722 -121.46330213]
[  38.63459171 -121.50865316]
[  38.62083106 -121.42442483]
[  38.62426996 -121.4583217 ]
[  38.53541359 -121.42391539]
```

```
transformed = model.transform(df_kmeans).select('ucr_ncic_code', 'prediction')
rows = transformed.collect()
print(rows[:3])
```

```
[Row(ucr_ncic_code=2404, prediction=5), Row(ucr_ncic_code=2204, prediction=1), Row(ucr_ncic_code=2404, prediction=
4)]
```

Τελικό στάδιο είναι η εισαγωγή των βιβλιοθηκών SQLContext και SparkContext με σκοπό τη δημιουργία της μεταβλητής df_pred η οποία μας επιτρέπει να βλέπουμε σε ποια ομάδα (cluster) ανήκει η κάθε παρατήρηση.

```
from pyspark.sql import SQLContext
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
sqlContext = SQLContext(sc)
df_pred = sqlContext.createDataFrame(rows)
df_pred.show()
```

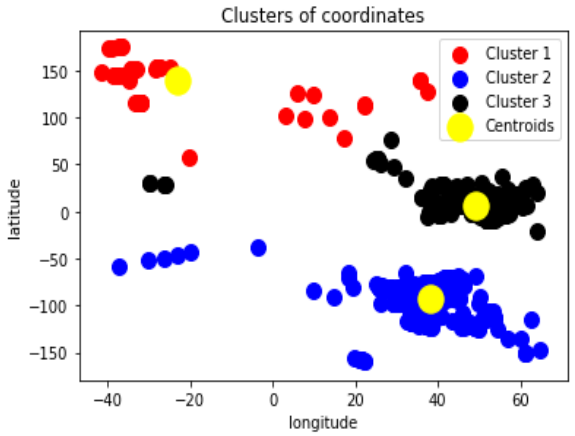
```
+-------------+----------+
|ucr_ncic_code|prediction|
+-------------+----------+
|         2404|         5|
|         2204|         1|
|         2404|         4|
|         2501|         5|
|         2299|         3|
|         2604|         5|
|         2299|         0|
|         2308|         0|
|         2203|         3|
|         2310|         2|
|         7000|         0|
|         2604|         5|
|         7000|         0|
|         2604|         5|
|         2605|         5|
|         2303|         5|
|         7000|         5|
|         2308|         0|
|         2604|         0|
|         2605|         0|
+-------------+----------+
```

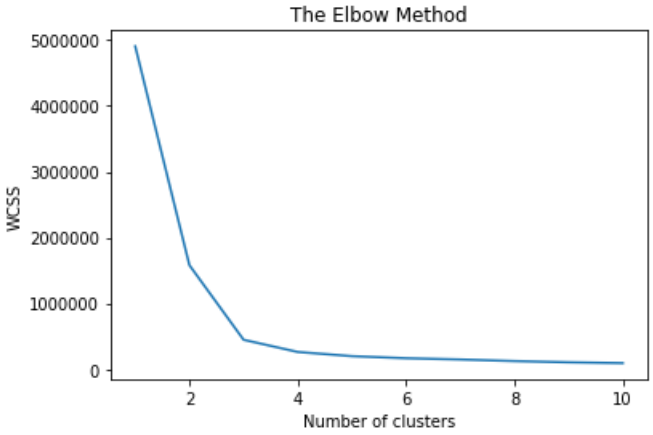
3.3. Αποτελέσματα πειραμάτων

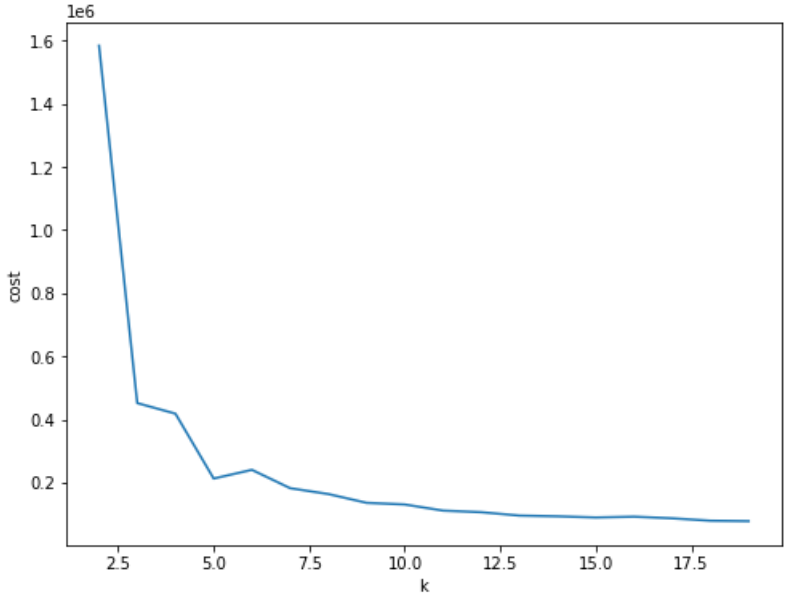
Σε αυτό το σημείο έχουμε τελειώσει την εκτέλεση των πειραμάτων μας για τα datasets μας και ακολουθεί η παρουσίαση των αποτελεσμάτων.

Sales transactions (SalesJan2009.csv)

Το συγκεκριμένο dataset αναφέρεται στις πωλήσεις που έγιναν το Γενάρη του 2009 σε διάφορες πόλεις του πλανήτη. Είναι σε μορφή csv, έχει μέγεθος 121 kilobytes και έχει συνολικά 998 εγγραφές. Οι στήλες έχουν τα ονόματα Transaction_date, Product, Price, Payment_Type, Name, City, State, Country, Account_Created, Last_Login, Latitude και Longitude. Από τα διαγράμματα προκύπτει ότι το k=3.

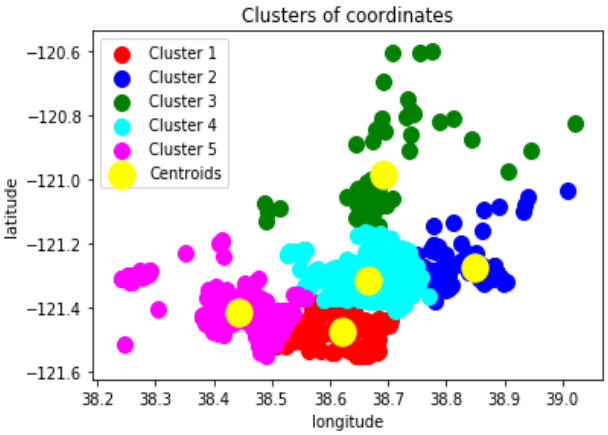


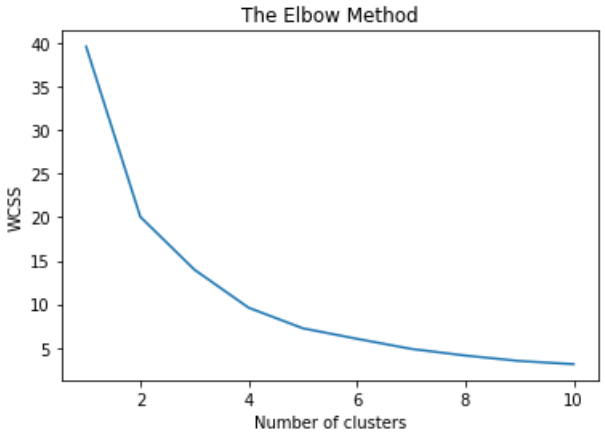


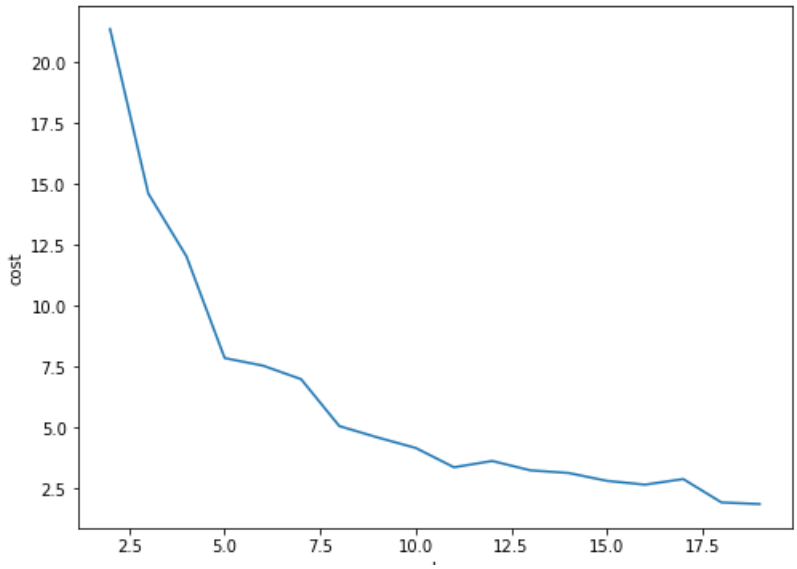


Real estate transactions (Sacramentorealestatetransactions.csv)

Το συγκεκριμένο dataset αναφέρεται σε συναλλαγές στα ακίνητα της πόλης του Sacramento σε διάστημα πέντε ημερών το μήνα Μάιο του 2008. Είναι και αυτό της μορφής csv, έχει μέγεθος 111 kilobytes και συνολικά 985 εγγραφές. Οι στήλες έχουν τα ονόματα street, zip, state, beds, baths, sq_ft, type, sale_date, price, latitude και longitude. Από τα διαγράμματα προκύπτει ότι k=5.



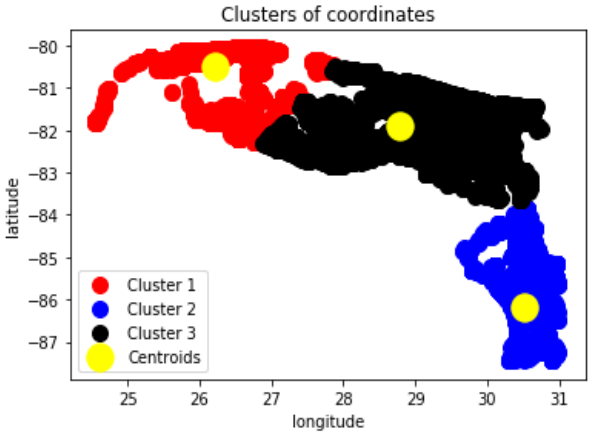


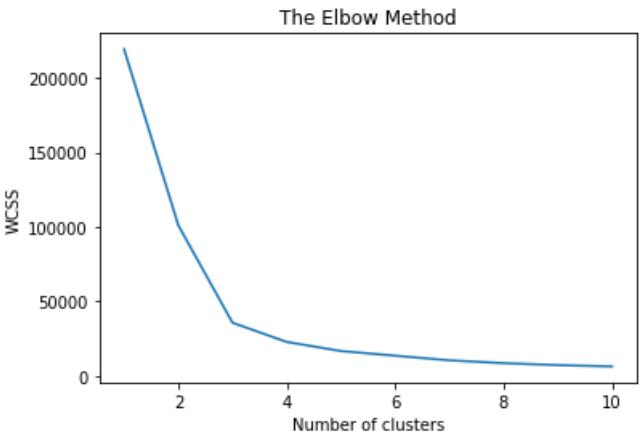


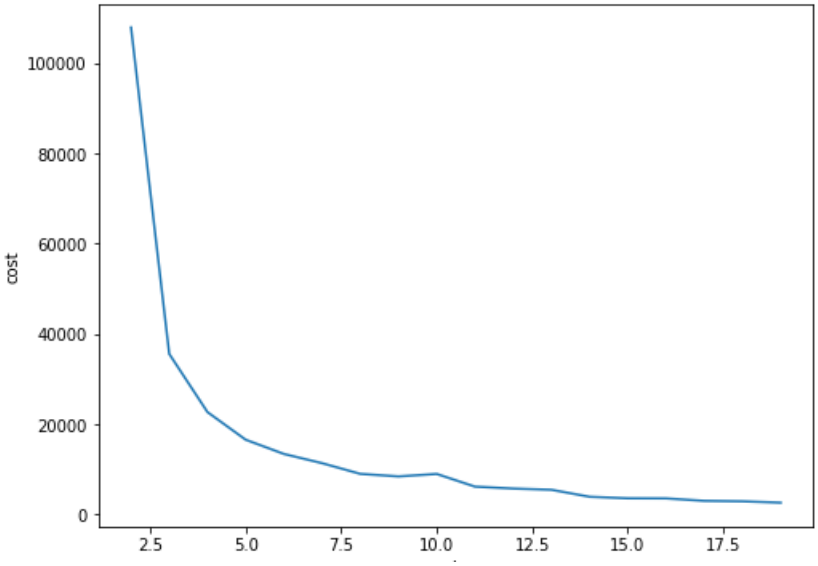
Sample insurance portofolio (FL_insurance_sample.csv)

Το τελευταίο dataset που θα χρησιμοποιήσουμε , περιέχει δείγματα από μια ασφαλιστική εταιρία της Florida που σημείωσε απότομη ανάπτυξη το 2012. Είναι σε μορφή csv, έχει μέγεθος 3,93 megabytes και συνολικά 36634 εγγραφές. Οι στήλες έχουν τα ονόματα policyID, statecode, country, eq_site_limit, hu_site_limit, fl_site_limit, fr_site_limit, tiv_2011, tiv_2012, eq_site_deductible, hu_site_deductible,

fl_site_deductible, fr_site_deductible, point_latitude, point_longitude, line, construction και point_granularity. Από τα διαγράμματα προκύπτει ότι k=3.







3.4.Συμπεράσματα και μελλοντικοί στόχοι

Αξιολογώντας τα αποτελέσματα της πειραματικής διαδικασίας οδηγούμαστε στο ξεκάθαρο συμπέρασμα ότι ο αλγόριθμος K-Means που έτρεξε σε python και spark (pyspark) ξεχωριστά φέρνει το ίδιο αποτέλεσμα όσον αφορά το clustering. Αυτό βέβαια οφείλεται στο γεγονός ότι και στις δύο περιπτώσεις ο αλγόριθμος ήταν μια βιβλιοθήκη έτοιμη. Ενδεχομένως αν φτιάχναμε εξ ολοκλήρου από την αρχή τον

κώδικα του K-Means και χρησιμοποιούσαμε διαφορετική μετρική απόσταση, τότε τα αποτελέσματα θα ήταν διαφορετικά.

Αν και ο γνωστικός στόχος της συγκεκριμένης διπλωματικής εργασίας σε γενικές γραμμές επετεύχθη, σίγουρα θα μπορούσαν να γίνουν ορισμένα παραπάνω πράγματα σε μελλοντικό χρόνο. Αρχικά όπως αναφέρθηκε και προηγουμένως θα μπορούσαμε να τρέξουμε τον αλγόριθμο σε python με μία custom μετρική απόσταση και να συγκρίνουμε τα αποτελέσματα με αυτά του spark. Επίσης θα μπορούσε να γίνει clustering συνδυάζοντας κάποιο text κείμενο και τη γεωγραφική τοποθεσία της κάθε εγγραφής (spatio-textual clustering).

Βιβλιογραφία

Bailey, T.C., & Gatrell, A.C. (1995). Interactive spatial data analysis. Essex: Addison-Wesley Longman.

Cleveland, W. S. (2001). “Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics”. International Statistical Review / Revue Internationale de Statistique 69 (1).

Data Mining Algorithms

DBSCAN

Dunham, M. H. (2003). Data Mining: Introductory and Advanced Topics. Pearson Education, Upper Saddle River, N. J. Prentice Hall.

Fischer, M.M., & Wang, J. (2011). Spatial Data Analysis: Models, Methods and Techniques. Berlin: Springer.

Johnston, R.J., Gregory, D., Pratt, G., & Watts, M. (2000). The Dictionary of Human Geography, 4th Edition. Oxford: Blackwell Publishers Ltd

Marsland, S., (2014). Machine learning: an algorithmic perspective. CRC press.

McKinney, W., (2012), Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc.

MLlib: Scalable Machine Learning on Spark

O’Neil, C., Schutt, R., (2013), Doing Data Science, O’Reilly Media, United States of America

Provost, F., & Fawcett, T., (2013), Data Science for Business: What you need to know about data mining and data-analytic thinking. O'Reilly Media, Inc.

repository.kallipos.gr/bitstream/11419/2972/1/02_chapter_06.pdf

Shaw, Z., (2014), Learn Python the hard way : a very simple introduction to the terrifyingly beautiful world of computers and code, Third edition, .Addison Wesley, United States of America.

Tan, P. N., Steinbach, M. & Kumar, V. (2006). Introduction to Data Mining. Boston, MA: Pearson/AddisonWesley.

Unwin, D. (1981). Introductory Spatial Analysis. New York: Methuen

Σταμάτης Καλογήρου (2015). Χωρική Ανάλυση: Μεθοδολογία και εφαρμογές με τη γλώσσα R