Πρώτη Σειρά Ασκήσεων

Αυτό είναι το δεύτερο μέρος της πρώτης σειράς ασκήσεων. Η προθεσμία για την παράδοση αυτού του κομματιού είναι στις 12 Νοεμβρίου, 1:00 μ.μ., στο ξεκίνημα του μαθήματος. Κάνετε turn-in το Iron Python notebook, τον κώδικα σας, και ένα pdf με την αναφορά σας. Για καθυστερημένες υποβολές ισχύει η πολιτική στην σελίδα του μαθήματος. Λεπτομέρειες για το turn-in, και για το πώς να γράφετε αναφορές είναι στη σελίδα Ασκήσεις του μαθήματος.

Ερώτηση 1

Στη σελίδα Ασκήσεις του μαθήματος σας δίνονται δύο αρχεία “data1.csv” και “data2.csv”. Το κάθε αρχείο έχει τρεις στήλες χωρισμένες με κόμμα, με ονόματα A, B, και C, και 100 γραμμές. Ο στόχος σας είναι να βρείτε την σχέση μεταξύ της στήλης A και των στηλών B και C για καθένα από τα αρχεία. Δημιουργείστε ένα Iron Python Notebook το οποίο θα πρέπει να περιέχει τον κώδικα για την επεξεργασία των δεδομένων, τα γραφήματα και τους υπολογισμούς που κάνατε καθώς και μία αναφορά με τα συμπεράσματα σας.

Ερώτηση 2

Για την άσκηση αυτή θα χρησιμοποιήσουμε το Yelp Academic Dataset. Κατεβάσετε τα δεδομένα από το link που υπάρχει την σελίδα Υλικό του μαθήματος. Τα δεδομένα είναι σε JSON format (ένα αντικείμενο ανά γραμμή) οπότε θα χρειαστείτε και ένα JSON parser για να τα πάρετε σε μια βολική μορφή για επεξεργασία. Μπορείτε να χρησιμοποιήσετε κάποιον υπάρχοντα JSON parser.

Το Yelp έχει πληροφορία για χρήστες και τοποθεσίες/μαγαζιά. Οι χρήστες γράφουν reviews ή tips για τις τοποθεσίες. Επίσης υπάρχει ένα κοινωνικό δίκτυο μεταξύ τους. Εμείς θα κάνουμε χρήση κατά κύριο λόγω του αρχείου με τα tips. Συνήθως ένας χρήστης αφήνει ένα tip όταν κάνει check-in σε μια τοποθεσία οπότε θα υποθέσουμε ότι κάθε tip αντιστοιχεί σε ένα check-in την ίδια ημερομηνία.

Θα χρησιμοποιήσετε τα δεδομένα για να λύσετε τα εξής δύο προβλήματα:

1. Θέλουμε να βρούμε ομάδες από τοποθεσίες (μεγέθους τουλάχιστον 2) τις οποίες συχνά οι χρήστες επισκέπτονται μέσα στην ίδια ημέρα. (Αυτή η πληροφορία είναι χρήσιμη ώστε π.χ., αν ο χρήστης πάει σε ένα από τα μαγαζιά να του προτείνουμε να πάει και σε ένα από τα άλλα στην ομάδα.) Σχεδιάστε και υλοποιήστε ένα αλγόριθμο που χρησιμοποιεί την εξόρυξη συχνών στοιχειοσυνόλων για να βρει αυτές τις ομάδες. Χρησιμοποιείστε ένα αρκετά μεγάλο κατώφλι υποστήριξης (π.χ., 20) για να έχετε μικρό αριθμό από ζεύγη, και μετά χαμηλώσετε το ώστε να πάρετε και μερικές τριάδες. Επιστρέψετε τα ζεύγη και τις τριάδες με τα ονόματα από τα μαγαζιά. Σχολιάστε τα αποτελέσματα και όποιες ενδιαφέρουσες συσχετίσεις παρατηρείτε.
2. Αναστρέφοντας τα δεδομένα, τώρα θέλουμε να βρούμε ομάδες από χρήστες (μεγέθους τουλάχιστον 2) οι οποίοι βγαίνουν συχνά μαζί. (Αυτή η πληροφορία είναι χρήσιμη γιατί μπορούμε π.χ. να τους κάνουμε προσφορά σαν γκρουπ.) Μια ομάδα από χρήστες θεωρούμε ότι έχουν βγει μαζί, αν όλοι έχουν

αφήσει tip στο ίδιο μαγαζί την ίδια ημέρα και είναι όλοι φίλοι μεταξύ τους στο κοινωνικό δίκτυο. Σχεδιάστε και υλοποιήστε ένα αλγόριθμο που χρησιμοποιεί την εξόρυξη συχνών στοιχειοσυνόλων για να βρει αυτές τις ομάδες. Στην περίπτωση αυτή μας ενδιαφέρει πόσα τέτοια γκρουπ μπορούμε να βρούμε οπότε αναφέρετε τα μεγέθη για διαφορετικά κατώφλια υποστήριξης.

Μπορείτε να χρησιμοποιήσετε όποια γλώσσα προγραμματισμού θέλετε για την υλοποίηση σας. Συνίσταται η χρήση της Python λόγω της ευελιξίας που προσφέρει με τις δομές και την χρήση της βιβλιοθήκης Pandas αλλά δεν είναι απαραίτητη. Για την εύρεση των συχνών στοιχειοσυνόλων μπορείτε επίσης να χρησιμοποιήσετε μια έτοιμη υλοποίηση. Υπάρχουν πολλές υλοποιήσεις στη σελίδα του FIMI (υπάρχει link στην σελίδα Υλικό), καθώς και στο WEKA (το link για το WEKA είναι επίσης στην σελίδα Υλικό). Για κάποιες υλοποιήσεις θα χρειαστεί να κάνετε μετατροπή των δεδομένων.

Εκτός από τον κώδικα θα παραδώσετε και μια αναφορά στην οποία θα περιγράφετε εν συντομία τον σχεδιασμό του αλγόριθμου σας, τα βήματα για να τρέξει ο κώδικας σας, και ένα σχολιασμό για τα αποτελέσματα. Σχολιάστε αν βλέπετε κάτι ενδιαφέρον στις τοποθεσίες οι οποίες συσχετίζονται, και αν είναι εφικτό να ανακαλύψουμε φίλους που βγαίνουν μαζί.

Ερώτηση 3

Στην Ερώτηση 2, εκτός από τα δεδομένα στα οποία ψάχνουμε συχνά στοιχειοσύνολα, μας δίνεται και ένα γράφημα με διμερείς σχέσεις μεταξύ των αντικειμένων. Μας ενδιαφέρει να βρούμε k-στοιχειοσύνολα τα οποία να είναι συχνά, και ταυτόχρονα να είναι όλα συνδεδεμένα μεταξύ τους στο γράφημα. Περιγράψετε πως μπορούμε να χρησιμοποιήσουμε αυτή την πληροφορία για να επιταχύνουμε τον APriori αλγόριθμο περιορίζοντας την δημιουργία των υποψήφιων στοιχειοσυνόλων. Η απάντηση σας δεν χρειάζεται να είναι πολύ μακροσκελής: περιγράψετε τις αλλαγές που θα χρειαστούν στον αλγόριθμο και στην υλοποίηση του APriori. Παραδώστε ένα pdf με την αναφορά σας.