Τεχνικές Εξόρυξης Δεδομένων

Εαρινό Εξάμηνο 2022-2023

2η Άσκηση - Ατομική ή ομαδική εργασία (2 Ατόμων)

Σκοπός της εργασίας

Σκοπός της εργασίας είναι η εξοικείωσή σας με τα βασικά στάδια της διαδικασίας που ακολουθούνται για την εφαρμογή τεχνικών εξόρυξης δεδομένων, ήτοι: συλλογή, προ-επεξεργασία / καθαρισμός, μετατροπή, εφαρμογή τεχνικών εξόρυξης δεδομένων και αξιολόγηση. Η υλοποίηση θα γίνει στην γλώσσα προγραμματισμού Python με την χρήση των εργαλείων/βιβλιοθηκών: jupyter notebook, pandas και SciKit Learn.

Περιγραφή

Το **Goodreads** είναι μια κοινωνική πλατφόρμα σε συνδυασμό με μια βάση δεδομένων βιβλίων που μοιάζει αρκετά με το IMDb για τις ταινίες: οι χρήστες μπορούν να αναζητούν βιβλία, να βάζουν ετικέτες τα, να μοιράζονται τις σκέψεις τους και να συζητούν. Το πιο σημαντικό είναι ότι μπορούν να βαθμολογούν τα βιβλία που έχουν διαβάσει σε μια κλίμακα από το 1 έως το 5 και να ανακαλύπτουν νέα βιβλία προς ανάγνωση. Στόχος μας είναι να μελετήσουμε ένα υποσύνολο δεδομένων από αυτή την πλατφόρμα και να δημιουργήσουμε έναν πρόγραμμα που συστήνει βιβλία. Ένας άλλος στόχος της εργασίας είναι να κατηγοριοποιήσετε τα βιβλία αυτόματα στο είδος στο οποίο ανήκουν βασιζόμενοι στις περιγραφές τους. Τέλος όσοι θέλουν μπορούν να αξιοποιήσουν τα εξώφυλλα των βιβλίων για να φτιάξουν ένα απλό σύστημα ανάκτησης εικόνας με βάση το περιεχόμενο της (***μπόνους***).

Το αρχείο που θα χρησιμοποιήσετε είναι το Best Books Ever Dataset. Το αρχειο μπορείτε να το δείτε στο αποθετήριο zenodo στον παρακάτω σύνδεσμο (εκεί θα βρείτε και το κουμπί για να το κατεβάσετε) https://zenodo.org/record/4265096#.Y-N2DnbP1jE.

Ζητούμενα

Αρχικά θα εισάγετε όλο το αρχείο σε ένα dataframe και θα μελετήσετε αν υπάρχουν τιμές Nan στις στήλες ώστε να αφαιρέσετε αυτές τις γραμμες από το dataframe. Στην συνέχεια θα γράψετε την κατάλληλες εντολές σε python για να απαντήσετε στα παρακάτω ζητούμενα. Τι περισσότερες

φορές οι απαντήσει θα δίνονται με ένα γράφημα και μπορείτε να χρησιμοποιήσετε όποια python βιβλιοθήκη θέλετε για το σκοπό αυτό.

Προεπεξεργασία (10%)
Παρατηρήστε την στήλη ratingsByStars, περιέχει 5 τιμές χωρισμένες με κόμματα , χωρίστε τις τιμές αυτές και προσθέστε στο dataframe ξεχωριστά τα ratings, δηλαδή ratingStar5, ratingStar4, ratingStar3 κτλ.

Επίσης η στήλη genres περιέχει για κάθε βιβλίο περισσότερα από ένα genre (είδος). Δημιουργήστε μία νέα στήλη (ονομάστε την genreSingle) και βάλτε μόνο το πρώτο genre από όλα τα genres που συναντάμε σε κάθε γραμμή. (πχ ['Fantasy', 'Young Adult', 'Fiction', 'Magic', 'Childrens', 'Adventure', 'Audiobook', 'Middle Grade', 'Classics', 'Science Fiction Fantasy'] -> η νέα στήλη θα έχει το 'Fantasy') deleting books without any genre information.

Χρησιμοποιήστε την στήλη publishDate και δημιουργήστε μία νέα στήλη με το έτος έκδοσης κάθε βιβλίου (μπορείτε να χρησιμοποιήστε την μέθοδο to_datetime() που παρέχει το pandas ή ότι άλλο θέλετε) .

Ερωτήματα για μελέτη των δεδομένων - απαντήστε σε 5 από τα παρακάτω (20%)

1. Κατασκευάστε το ιστόγραμμα των ratings στο σύνολο δεδομένων (χρησιμοποιήστε την στήλη rating)
2. Ποιά είναι τα 10 βιβλία με τις περισσότερες σελίδες.
3. Ποιά είναι τα 10 βιβλία με τα περισσότερα 5-αστέρια (χρησιμοποιήστε μόνο τα βιβλία που έχουν λάβει πάνω από 10.000 5-star ratings από τη στήλη ratingStar5) .
4. Ποιές είναι οι πιο συχνές λέξεις στους τίτλους των βιβλίων (αφού αφαιρεθούν τα stop words)
5. Ποιοι είναι οι 10 συγγραφεις με τα περισσότερα βιβλία
6. Ποιοι είναι οι 10 συγγραφείς με τις περισσότερες κριτικές (χρησιμοποιήστε την στήλη numRatings).
7. Κατατάξτε του συγγραφείς με βάση τα βιβλία τους ανά έτος.
8. Ποιές είναι οι πιο συχνές γλώσσες που έχουν γραφτεί τα βιβλία στα δεδομένα σας
9. Ποιοί είναι οι 10 εκδότες που έχουν εκδώσει τα περισσότερα βιβλία.
10. Έχουν τα βιβλία με τις περισσότερες σελίδες (πχ περισσότερες από 1000 pages) υψηλότερα ratings ?
11. Συγκεντρώστε σε ένα γράφημα η σε ένα πίνακα όλα τα μοναδικά είδη βιβλίων (genres). Ποιά είναι τα πιο συχνά genres; Το ίδιο και για τα awards.
12. Φτιάξτε τα wordclouds για τη στήλη description. Σε αυτό το ερώτημα αφαιρέστε τα stop words, πειραματιστείτε με τις παραμέτρους του wordcloud και εντοπίστε τις πιο χαρακτηριστικές λέξεις που χρησιμοποιούνται στα βιβλία του συνόλου των δεδομένων σας.

13. Πόσα βιβλία εκδίδονται ανά έτος ;

Υλοποίηση Recommendation system (35%)

Ο στόχος ενός τέτοιου συστήματος είναι (1) να προβλέψει τις αξιολογήσεις ενός χρήστη για τα βιβλία που δεν έχει διαβάσει ακόμα, και (2) να εμφανίσει ένα ταξινομημένη λίστα με τα κορυφαία **N** βιβλία για τα οποία πιστεύουμε ότι θα ήθελαν να μάθουν περισσότερα. 'Ενας άλλος στόχος ενός Recommender είναι (3) να βοηθήσει τους χρήστες να ανακαλύψουν σχετικά βιβλία που δεν θα είχαν βρει διαφορετικά.

Σε αυτό το ερώτημα θα χρειαστείτε τις στήλες
BookId
Description
Και μόνο όσες γραμμές έχουν γλώσσα "English".

Δημιουργήστε τον **TF-IDF** (Term Frequency - Inverse Document Frequency) πίνακα των unigrams και των bigrams για τη στήλη description (χρησιμοποιήστε την παράμετρο stop_word του TfidfVectorizer).

Cosine Similarity: Η μετρική αυτή υπολογίζει την ομοιότητα μεταξύ δύο διανυσμάτων x,y, χρησιμοποιώντας τη γωνία μεταξύ τους (όταν η γωνία είναι 0 σημαίνει ότι τα x και y είναι ίσα , αν εξαιρέσουμε το μήκος τους). Διατρέξτε τον TF-IDF πίνακα και υπολογίστε το similarity καθενός βιβλίου με τα υπόλοιπα. Αποθηκεύστε σε ένα python dictionary τα 100 πιο όμοια βιβλία.
Πρόβλεψη: Φτιάξτε μία συνάρτηση η οποία παίρνει σαν είσοδο ένα id και ένα ακέραιο αριθμό N, και επιστρέφει τα N πιο όμοια βιβλία.

recommend(item_id = 4085439, num = 5)

Η έξοδος της συνάρτησης να είναι της παρακάτω μορφής μορφής

Recommending 5 books similar to: The Hunger Games

Recommended: NAME
Description: DESCRIPTION
(score:0.12235188993161432)

Recommended: NAME
Description: DESCRIPTION
(score:0.12235188993161432)
……..

Υλοποίηση Κατηγοριοποίησης (Classification) (35%)

Χρησιμοποιήστε την στήλη genreSingle, βρείτε τα 10 πιο συχνά genres και κρατήστε σε ένα νέο dataframe τα βιβλία εκείνα που ανήκουν σε αυτες τις 10 πιο συχνές κατηγορίες. Θα χρειαστείτε το bookId το description και το genreSingle. Στην συνέχεια καθαρίστε την στήλη description χρησιμοποιώντας τις μεθόδους που είδαμε στα φροντιστήρια (πχ αφαίρεση σημείων στίξης, μετατροπή όλων των χαρακτήρων σε πεζά, κ.α.). Εφαρμόστε την μέθοδο word2vec για τα descriptions και στην συνέχεια με την χρήση των embeddings να υπολογίσετε για κάθε description ένα διάνυσμα με 200-300 τιμές (features) - αυτό θα είναι ο μέσος όρος των embeddings των λέξεων από τις οποίες αποτελείται το description.

*Χρησιμοποιήστε τη βιβλιοθήκη pickle της Python για να αποθηκεύσετε τα χαρακτηριστικά σε αρχεία *.pkl . Με αυτό τον τρόπο δεν χρειάζεται να υπολογίζονται από την αρχή τα χαρακτηριστικά κάθε φορά που τρέχετε το πρόγραμμά σας, αλλά μπορείτε μόνο να τα φορτωνεται στην μνήμη χρησιμοποιώντας την αντίστοιχη μέθοδο* ***load****.*

Χωρίστε το σύνολο των δεδομένων σε train (80%) και test (20%) χρησιμοποιώντας την μέθοδο ***train_test_split()*** της βιβλιοθήκης sklearn.

Σε αυτό το ερώτημα θα πρέπει το πρόγραμμα σας να μπορεί να βρει τις κατηγορίες (genre) του συνόλου δοκιμής (test) χρησιμοποιώντας τις παρακάτω μεθόδους Classification:
- Naive Bayes
- Support Vector Machines (SVM, να πειραματιστείτε με τις παραμέτρους kernel (rbf, linear), c και gamma. Η επιλογή των παραμέτρων μπορεί να γίνει και με GridSearchCV)
- Random Forests

Ολα τα παραπάνω μοντέλα θα εκπαιδευτούν **ΜΟΝΟ** στο σύνολο train και θα αξιολογηθούν στο σύνολο test. Επίσης θα πρέπει να αξιολογήσετε και να καταγράψετε την απόδοση κάθε μεθόδου χρησιμοποιώντας 10-fold Cross Validation χρησιμοποιώντας τις παρακάτω μετρικές:
- Precision / Recall / F-Measure
- Accuracy

Στο τέλος να ετοιμάσετε ένα πίνακα με τα αποτελέσματα των πειραμάτων σας για κάθε μετρική/παράμετρο που χρησιμοποιήσατε.

BONUS - Judging a book by its cover..!

Η ανάκτηση εικόνας με βάση το περιεχόμενο (*Content Based Image Retrieval*) είναι μια τεχνική που χρησιμοποιεί οπτικά χαρακτηριστικά (όπως χρώμα, υφή, σχήμα) για την αναζήτηση

εικόνων. Τα χρωματικά χαρακτηριστικά θεωρούνται από τα πλέον χρησιμοποιούμενα χαρακτηριστικά χαμηλού επιπέδου για την αναζήτηση εικόνων σε μεγάλες βάσεις δεδομένων εικόνων. Το ιστόγραμμα χρώματος είναι απλώς ένα ιστόγραμμα που δείχνει το επίπεδο χρώματος για κάθε μεμονωμένο κανάλι χρώματος RGB (όπου οι τιμές των εικονοστοιχείων είναι στο εύρος 0-255).

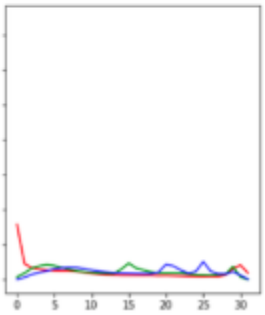
Στο eclass θα βρείτε ένα python notebook το οποίο μπορείτε να χρησιμοποιήσετε για να κατεβάσετε όλα τα εξώφυλλα του Best Books Ever Dataset σε μορφή jpg. Οδηγίες για το κατέβασμα των εικόνων θα σας δοθούν και στα φροντιστήρια.

Βήμα πρώτο: Χρησιμοποιήστε τον κώδικα που σας δίνουμε για να κατεβάσετε τοπικά ένα σύνολο από τις εικόνες του dataset (μπορεί να κατεβάσετε πχ 400-500. Όσες περισσότερες εικόνες έχετε τόσο καλύτερα αποτελέσματα θα έχει η αναζήτηση σας στο τελευταίο βήμα). Κρατήστε μερικές εικόνες για δοκιμές αργότερα (πχ 5%) .

Βήμα δεύτερο: Υπολογισμός ιστογράμματος για κάθε εικόνα. Για το σκοπό αυτο θα χρησιμοποιήσετε την βιβλιοθήκη OpenCV και συγκεκριμένα την μέθοδο calcHist.

cv2.calcHist([images], [channels], [mask], [bins], [hist_range])

images είναι η εικόνα σε μορφή BGR. Αυτό το όρισμα αναμένει μια λίστα εικόνων, γι' αυτό και έχουμε τοποθετήσει μια εικόνα μέσα σε αγκύλες [] αν πρόκειται για μία εικόνα.
channels είναι το χρωματικό κανάλι (BGR) για το οποίο θέλουμε να δημιουργήσουμε ένα ιστόγραμμα- το κάνουμε αυτό για ένα μόνο κανάλι κάθε φορά.
mask είναι ένας άλλος πίνακας εικόνων που αποτελείται από τιμές 0 και 1 που μας επιτρέπουν να καλύψουμε (π.χ. να κρύψουμε) μέρος των εικόνων αν θέλουμε. Δεν θα το χρησιμοποιήσουμε αυτό, οπότε το θέτουμε στην τιμή **None**.
bins είναι ο αριθμός των ράβδων του ιστογράμματος στις οποίες τοποθετούμε τις τιμές μας. Μπορούμε να το ορίσουμε σε 256 αν θέλουμε να διατηρήσουμε όλες τις αρχικές τιμές, αλλά συστήνουμε 32 ή 8 bins για λόγους απόδοσης.
hist_range είναι το εύρος των χρωματικών τιμών που αναμένουμε. Καθώς χρησιμοποιούμε RGB/BGR, αναμένουμε ελάχιστη τιμή 0 και μέγιστη τιμή 255, οπότε γράφουμε [0, 256].



Μπορείτε να δημιουργήσουμε ένα ιστόγραμμα για κάθε κανάλι χρώματος (BGR) - το κάνουμε αυτό για ένα μόνο κανάλι κάθε φορά και μετά θα ενώσετε τα αποτελέσματα.

Αφού γίνει η μετατροπή των εικόνων μας σε τρία διανύσματα που αντιπροσωπεύουν τα τρία κανάλια χρώματος, πρέπει να συνθέσετε αυτά τα τρία διανύσματα σε ένα ενιαίο διάνυσμα. Για κάθε εικόνα θα γίνει η διαδικασία αυτή και αυτό που θα προκύψει ειναι ενα dataframe με τα ιστογράμματα για κάθε εικόνα και ένα αναγνωριστικό id για να μπορείτε να τις ξεχωρίζετε.

Βήμα τρίτο: Επιλέξτε μία εικόνα από το σύνολο δοκιμής και υπολογιστε και για αυτή το ιστογραμμα της.

Βήμα τέταρτο: Αναζήτηση για τις πιο κοντινές εικόνες στο σύνολο των εικόνων συγκρίνοντας τα ιστογράμματα είτε με ευκλείδεια απόσταση είτε με cosine similarity. Για κάθε ερώτημα να εμφανίζονται οι 4 πιο κοντινές εικόνες. Χρησιμοποιήστε matplotlib για την απεικόνιση των αποτελεσμάτων. Αυτό που πρέπει να ξέρετε για αυτή την βιβλιοθήκη είναι ότι φορτώνει εικόνες σε μορφή **B**lue **G**reen **R**ed (BGR) και όχι στο κλασικό RGB που αναφέρθηκε νωρίτερα. Θα χρειαστεί να κάνετε μετατροπή σε κάθε εικόνα (κυρίως flipping - αναστροφή με την εντολή flip() της βιβλιοθήκης numpy) για να απεικονιστεί σωστά στην έξοδο.