Πανεπιστήμιο Ιωαννίνων,
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Εξόρυξη Δεδομένων – ΜΥΕ012/ΠΛΕ 059
Χειμερινό Εξάμηνο, 2020

Τρίτη Σειρά Ασκήσεων

Η προθεσμία για την Τρίτη σειρά ασκήσεων είναι Σάββατο 13 Φεβρουαρίου μέχρι το τέλος της ημέρας. Παραδώσετε το notebook με τον κώδικα και τα αποτελέσματα σας, και την αναφορά με τον σχολιασμό. Παραδώστε το notebook και σε ipynb και σε html μορφή. Για καθυστερημένες υποβολές ισχύει η πολιτική στην σελίδα του μαθήματος. Αν χρησιμοποιήσετε free passes θα πρέπει να το αναφέρετε στην αναφορά. Θα αφαιρεθούν από όλα τα μέλη της ομάδας. Αν κάποιο μέλος της ομάδας έχει χρησιμοποιήσει όλα τα free passes του θα χάσει το ποσοστό που αναλογεί. Λεπτομέρειες για το turn-in, και για το πώς να γράφετε αναφορές είναι στη σελίδα Ασκήσεις του μαθήματος. Οι ασκήσεις μπορούν να γίνουν σε ομάδες μέχρι δύο ατόμων.

Ερώτηση 1

Στην άσκηση αυτή θα εξασκηθείτε στην εφαρμογή αλγορίθμων κατηγοριοποίησης. Θα χρησιμοποιήσετε τα δεδομένα για τις επιχειρήσεις και τις κατηγορίες που δημιουργήσατε στην Δεύτερη Άσκηση στην Ερώτηση 3. Θα πάρετε τις επιχειρήσεις του Τορόντο που έχουν τα “Beauty & Spas”, “Shopping” και “Bars” στις κατηγορίες τους, και θα αναθέσετε την κάθε επιχείρηση σε μια κατηγορία όπως και στην Δεύτερη Άσκηση. Θα κρατήστε τις επιχειρήσεις που έχουν τουλάχιστον 10 reviews. Για κάθε επιχείρηση στην λίστα σας, πάρετε όλα τα reviews για την επιχείρηση από το αρχείο yelp_academic_dataset_review.json και ενώστε τα σε ένα μεγάλο κείμενο για την επιχείρηση. Ο στόχος είναι να φτιάξετε ένα classifier που ξεχωρίζει τις κατηγορίες.

Η Ερώτηση έχει δύο βήματα:

1. Στο πρώτο βήμα θα πάρετε την tf-idf αναπαράσταση των επιχειρήσεων, όπως και στην Δεύτερη Άσκηση. Θα πειραματιστείτε με πέντε classifiers: Logistic Regression, SVM, Decision Trees, K-NN, και Naïve Bayes. Για την αξιολόγηση θα χρησιμοποιήσετε 5-fold cross validation. Αναφέρετε το μέσο confusion matrix και τις μέσες τιμές για τις μετρικές accuracy, precision, recall και F1-measure. Μπορείτε να πειραματιστείτε με διάφορες παραλλαγές του tf-idf vectorizer (π.χ., συγκεκριμένο αριθμό από features, κλπ). Πόσο επιτυχής είναι η κατηγοριοποίηση σε σχέση με το clustering στα ίδια δεδομένα?
2. Στο δεύτερο βήμα θα χρησιμοποιήσετε τα ίδια δεδομένα όπως και στο Βήμα 1, αλλά θα εξάγετε τα features χρησιμοποιώντας τα word embeddings του Google. Η αναπαράσταση του κειμένου θα είναι η μέση τιμή των embeddings των λέξεων, όπως στο φροντιστήριο. Κάνετε την ίδια αξιολόγηση όπως στο Βήμα 1, και εξετάστε αν βελτιώνονται ή χειροτερεύουν τα αποτελέσματα.
   Υπόδειξη: Ο στόχος της άσκησης είναι να χρησιμοποιήσετε τα word embeddings του Google. Αν όμως έχετε πρόβλημα να τα φορτώσετε (λόγω περιορισμών στη μνήμη) μπορείτε να εκπαιδεύσετε το δικό σας word embedding μοντέλο.

Ερώτηση 2

Σε αυτή στην άσκηση θα χρησιμοποιήσετε embeddings για συστήματα συστάσεων. Θα δημιουργήσετε embeddings για τις επιχειρήσεις και τους χρήστες και θα χρησιμοποιήσετε αυτά τα embeddings για τον υπολογισμό της ομοιότητας στους αλγορίθμους UCF και ICF που υλοποιήσατε στην Δεύτερη Άσκηση.

Συγκεκριμένα, θα πάρετε ακριβώς το ίδιο σύνολο δεδομένων όπως και στην Ερώτηση 2 της Δεύτερης Άσκησης, και θα αφαιρέσετε και πάλι το 5% το οποίο θα προσπαθήσετε να προβλέψετε. Χρησιμοποιώντας το 95% θα δημιουργήσετε embeddings για τις επιχειρήσεις και τους χρήστες, και θα τα χρησιμοποιήσετε στον ICF και UCF αλγόριθμο αντίστοιχα.

1. Για το embedding των επιχειρήσεων, για κάθε χρήστη, θα δημιουργήσετε ένα «κείμενο» με λέξεις τα business ids των επιχειρήσεων που έχει βαθμολογήσει ο χρήστης ταξινομημένες με βάση την ημερομηνία που έγινε η βαθμολόγηση. Χρησιμοποιώντας αυτή την συλλογή των κειμένων θα χρησιμοποιήσετε το skipgram μοντέλο για να παράγετε ένα embedding για τις επιχειρήσεις. Υπολογίστε την ομοιότητα μεταξύ των επιχειρήσεων ως το cosine similarity των embeddings τους. Χρησιμοποιείστε αυτή την ομοιότητα στον ICF αλγόριθμο για συστάσεις.

Υπάρχουν τρεις παράμετροι που επηρεάζουν τον αλγόριθμο σας: η διάσταση του embedding d, το μέγεθος του παραθύρου w που χρησιμοποιεί το embedding και ο αριθμός των όμοιων επιχειρήσεων k. Για το embedding θα κρατήσετε την διάσταση σταθερή στο $d = 100$, και θα πειραματιστείτε με μέγεθος παραθύρου $w = [10,50,100,1000]$. Για το k θα δοκιμάστε τις τιμές $k = [1,5,10,20,40,50,60,70,80,100]$ όπως και στην Δεύτερη Άσκηση. Για τις τέσσερεις διαφορετικές τιμές του w δημιουργείστε μια καμπύλη για τις διαφορετικές τιμές του k. Βάλτε σε ένα plot όλες τις καμπύλες καθώς και την καμπύλη για τον ICF αλγόριθμο που υλοποιήσατε στην Δεύτερη Άσκηση (τον οποίο θα τρέξετε πάνω στα ίδια δεδομένα). Συγκρίνετε τους αλγορίθμους και γράψτε τις παρατηρήσεις σας.

2. Αντίστοιχα, θα κάνετε το embedding των χρηστών και θα υπολογίσετε την ομοιότητα μεταξύ των χρηστών για τον UCF αλγόριθμο. Για κάθε επιχείρηση, θα δημιουργήσετε ένα «κείμενο» με τα user ids των χρηστών που βαθμολόγησαν αυτή την επιχείρηση ταξινομημένα χρονολογικά και θα υπολογίσετε ένα embedding για τα user ids. Θα υπολογίσετε την ομοιότητα μεταξύ δύο χρηστών ως το cosine similarity των embeddings τους, και θα χρησιμοποιήσετε την ομοιότητα στον UCF αλγόριθμο. Στην αξιολόγηση θα χρησιμοποιήσετε ως διάσταση του embedding $d = 100$, και για το μέγεθος παραθύρου τις τιμές $w = [10,50,100,1000]$. Για τον αριθμό k των πιο όμοιων χρηστών θα δοκιμάστε τις τιμές $k = [1,5,10,20,50,100,200,500,1000]$ όπως και στην Δεύτερη Άσκηση. Όπως και για τον ICF βάλτε σε ένα plot όλες τις καμπύλες καθώς και την καμπύλη για τον UCF αλγόριθμο που υλοποιήσατε στην Δεύτερη Άσκηση (τον οποίο θα τρέξετε πάνω στα ίδια δεδομένα). Συγκρίνετε τους αλγορίθμους και γράψτε τις παρατηρήσεις σας.

Υπόδειξη: Θα σας βοηθήσει η βιβλιοθήκη datetime που έχει μεθόδους για την μετατροπή string σε datetime αντικείμενο και επιτρέπει σύγκριση μεταξύ ημερομηνιών. Επίσης θα σας βοηθήσει να χρησιμοποιήσετε την παράμετρο key στις μεθόδους ταξινόμησης της Python.

Ερώτηση 3

Για την άσκηση αυτή θα δείξετε την σχέση που υπάρχει μεταξύ του Pagerank διανύσματος με ομοιόμορφο jump vector, και των personalized Pagerank διανυσμάτων. Υπενθυμίζω ότι ο Pagerank αλγόριθμος έχει σαν παράμετρο ένα διάνυσμα $\boldsymbol{v}$ (το jump vector) το οποίο ορίζει μια κατανομή πιθανότητας πάνω στους κόμβους του γραφήματος και η τιμή $\boldsymbol{v}(i)$ καθορίζει την πιθανότητα να επιλέξουμε τον κόμβο i για επανεκκίνηση. Έστω $\boldsymbol{p_u}$ το Pagerank διάνυσμα με ομοιόμορφο jump vector (δηλαδή $\boldsymbol{v^T} = \left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$), και $\boldsymbol{p_i}$ το personalized Pagerank διάνυσμα όπου το jump vector δίνει όλη την πιθανότητα στον κόμβο i (δηλαδή, $\boldsymbol{v^T} = (0,0, \dots, 0,1,0, \dots 0)$ με το 1 στην i θέση).

Αποδείξετε ότι το διάνυσμα $\boldsymbol{p_u}$ είναι ο μέσος όρος των διανυσμάτων $\boldsymbol{p_1}, \boldsymbol{p_2}, \dots, \boldsymbol{p_n}$, δηλαδή, $\boldsymbol{p_u} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p_i}$. Για την απόδειξη θα χρησιμοποιήσετε το γεγονός ότι το Pagerank vector $\boldsymbol{p_v}$ (το Pagerank διάνυσμα με jump vector $\boldsymbol{v}$) μπορεί να γραφτεί σαν γραμμική συνάρτηση του jump vector, δηλαδή $\boldsymbol{p_v^T} = \boldsymbol{v^T Q}$, για κάποιο πίνακα $\boldsymbol{Q}$.

(Υπενθύμιση: Όταν αναφερόμαστε σε διανύσματα υποθέτουμε ότι είναι στήλες. Δηλαδή ένα n-διάστατο διάνυσμα $\boldsymbol{v}$ είναι ένας $n \times 1$ πίνακας. Αν θέλουμε να χρησιμοποιήσουμε το διάνυσμα σαν γραμμή, δηλαδή σαν ένα $1 \times n$ πίνακα θα το συμβολίζουμε ως $\boldsymbol{v^T}$)

Απαντήστε στα εξής ερωτήματα:

1. Χρησιμοποιώντας την σχέση $\boldsymbol{p_v^T} = (1-a)\boldsymbol{p_v^T P} + a\boldsymbol{v^T}$, δώστε την φόρμουλα για τον πίνακα $\boldsymbol{Q}$.
2. Δοθείσας της σχέσης $\boldsymbol{p_v^T} = \boldsymbol{v^T Q}$, τι μπορείτε να δείξετε για τις στήλες του πίνακα $\boldsymbol{Q}$?
3. Αποδείξτε ότι $\boldsymbol{p_u} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p_i}$
4. Στην γενική περίπτωση ενός οποιουδήποτε jump vector $\boldsymbol{v}$ (όχι απαραίτητα το ομοιόμορφο διάνυσμα), πως μπορούμε να εκφράσουμε το $\boldsymbol{p_v}$ σαν συνάρτηση των $\boldsymbol{p_i}$?

Ερώτηση 4

Σε αυτή την άσκηση θα χρησιμοποιήσετε το κοινωνικό δίκτυο μεταξύ των χρηστών του Yelp για να προβλέψετε τα ratings τους για νέες επιχειρήσεις. Για την πρόβλεψη θα υλοποιήσετε τον αλγόριθμο για value propagation τον οποίο περιγράψαμε στην τάξη.

Θα χρησιμοποιήσετε παρόμοια δεδομένα με αυτά που δημιουργήσατε για την Δεύτερη Σειρά Ασκήσεων, για τα συστήματα συστάσεων. Η διαφορά είναι ότι θα κάνετε πιο επιθετικό pruning. Συγκεκριμένα, στα τελικά σας δεδομένα θα έχετε ένα σύνολο από χρήστες U, και ένα σύνολο από επιχειρήσεις B, όπου ο κάθε χρήστης στο U θα έχει **τουλάχιστον 20 reviews** σε επιχειρήσεις στο B, και η κάθε επιχείρηση στο B θα έχει **τουλάχιστον 50 reviews** από χρήστες στο U.

Δημιουργείστε ένα γράφημα με κορυφές τους χρήστες στο σύνολο U και ακμές τις φιλίες μεταξύ των χρηστών, τις οποίες θα πάρετε από το αρχείο yelp_academic_dataset_user.json . Από αυτό το γράφημα κρατήστε τη μεγαλύτερη συνεκτική συνιστώσα (θα πρέπει να είναι ίδια με όλο το γράφημα). Αυτή θα ορίσει το γράφημα G με το οποίο θα δουλέψετε.

Αφαιρέστε τυχαία το 5% των ratings για τους χρήστες του γραφήματος που πήρατε. Αυτό είναι το test set D_{test} και το υπόλοιπο 95% είναι το training set D_{train}. Ο στόχος είναι να προβλέψουμε τα ratings στο D_{test} πραγματοποιώντας διάχυση τιμών (value propagation) στο γράφημα. Έστω B_{test} το σύνολο των διακριτών επιχειρήσεων που εμφανίζονται στο D_{test}. Θα πραγματοποιήσετε την διαδικασία του value propagation για κάθε επιχείρηση $b \in B_{test}$. Για κάθε τριάδα χρήστη-επιχείρηση-rating (u, b, r) που εμφανίζεται στο D_{train} ο κόμβος u στο γράφημα θα έχει σταθερή τιμή r (θα γίνει απορροφητικός με τιμή r). Για κάθε άλλο (μη απορροφητικό) κόμβο v, θα υπολογίσετε το rating $R(v, b)$ χρησιμοποιώντας τη διαδικασία του value propagation που περιγράψαμε στην τάξη. Για ένα ζευγάρι χρήστη-επιχείρηση (x, b) η πρόβλεψη σας θα είναι η τιμή $R(x, b)$.

Υπολογίσετε το Root Mean Square Error (RMSE) για αυτή τη μέθοδο. Στη συνέχεια, τρέξετε τους αλγορίθμους UCF, ICF, UA, IA που υλοποιήσατε στην Δεύτερη Σειρά για αυτό το dataset και συγκρίνετε το Root Mean Square Error (RMSE). Παρουσιάστε τα αποτελέσματα σας και γράψετε τις παρατηρήσεις σας.

Bonus: Προτείνετε, υλοποιήστε και τεστάρετε μια διαφορετική μέθοδο που να προβλέπει τα ratings των χρηστών χρησιμοποιώντας το γράφημα των φιλιών μεταξύ των χρηστών. Περιγράψετε την μέθοδο σας και τα αποτελέσματα της.

Ερώτηση 4 (bonus)

Ο στόχος της άσκησης αυτής είναι να εξασκηθείτε με κατηγοριοποίηση σε ένα πιο ανοιχτό πρόβλημα. Ο στόχος είναι να προβλέψετε αν σε κάποιο χρήστη άρεσε μια επιχείρηση που επισκέφτηκε.

Για την άσκηση αυτή δημιουργήθηκε ένας διαγωνισμός στο Kaggle για το μάθημα (εδώ είναι ο σύνδεσμος για τον διαγωνισμό). Δημιουργήστε ένα account με το email του πανεπιστημίου. Θα σας δοθεί πρόσβαση στον διαγωνισμό μέσω του link και θα μπορέστε να καταθέσετε μια λύση για τον διαγωνισμό. Εκεί σας δίνονται τα training data και τα test data.

Τα training data, και test data αποτελούνται από μια συλλογή από ζευγάρια από user και business ids. Για τα training data σας δίνεται και το class label του κάθε ζευγαριού το οποίο είναι 0 (δεν αρέσει) ή 1 (αρέσει). Το class label προκύπτει από το σκορ που έδωσε ο χρήστης για την επιχείρηση: Σκορ μικρότερα από 4 μπαίνουν στην αρνητική κλάση (κλάση 0) ενώ σκορ μεγαλύτερα ή ίσα του 4 μπαίνουν στην κλάση 1. Το αρχείο της λύσης έχει δύο πεδία: Το PairId το οποίο προκύπτει συνενώνοντας το user id και το business id με μια παύλα (‘-‘) και το class label που προβλέπετε. Υπάρχει ένα παράδειγμα στο tab Data.

Ο στόχος σας είναι να εκπαιδεύσετε ένα μοντέλο κατηγοριοποίησης που θα προβλέπει την κλάση του ζευγαριού. Για τον σκοπό αυτό θα πρέπει να εξάγετε χαρακτηριστικά (features) για τον χρήστη, την επιχείρηση και τον συνδυασμό τους και να εκπαιδεύσετε κάποιο μοντέλο κατηγοριοποίησης. Τα χαρακτηριστικά θα τα εξάγετε από τα δεδομένα που έχετε από το Yelp για τους χρήστες και τις επιχειρήσεις. Θα εξάγετε τα ίδια χαρακτηριστικά και για τα test data και θα υποβάλετε τα αποτελέσματα στον διαγωνισμό. Θα αξιολογηθείτε με βάση το accuracy της λύσης σας. Υπάρχει μία κατάταξη στην οποία μπορείτε να δείτε την θέση σας σε σχέση με άλλες λύσεις. Ο βαθμός σας εξαρτάται από την θέση που θα πάρετε.

Προφανώς μπορείτε να βρείτε το σκορ για τα test data και να υποβάλετε την τέλεια λύση. Για να έχει νόημα ο διαγωνισμός δεν θα πρέπει να χρησιμοποιήσετε καθόλου τα δεδομένα σχετικά με τα test data είτε κατά την εκπαίδευση του μοντέλου, είτε κατά την εφαρμογή του. Όταν κοιτάτε ένα ζευγάρι και εξάγετε χαρακτηριστικά, είτε στην διαδικασία της εκπαίδευσης (training), είτε στην διαδικασία της αξιολόγησης (test), θα θεωρείτε ότι δεν έχετε καμία πληροφορία γι αυτό το ζευγάρι. Για παράδειγμα, δεν μπορείτε να χρησιμοποιήσετε το review που έγραψε ο χρήστης για την συγκεκριμένη επιχείρηση για να εξάγετε χαρακτηριστικά (αλλά μπορείτε να χρησιμοποιήσετε τα reviews που έχει γράψει για άλλες επιχειρήσεις). Αν για παράδειγμα δημιουργήσετε κάποιο feature όπως το average score που έχει δώσει ο χρήστης σε επιχειρήσεις τις ίδιας κατηγορίας με την επιχείρηση, δεν μπορείτε σε αυτό το average να συμπεριλάβετε και το σκορ για την συγκεκριμένη επιχείρηση. Αν χρησιμοποιήσετε δεδομένα από τα test data για την εξαγωγή χαρακτηριστικών θα θεωρηθεί λάθος η λύση σας.

Στην αναφορά περιγράψετε τα χαρακτηριστικά που δημιουργήσατε και το μοντέλο που χρησιμοποιήσατε, και αναφέρετε το όνομα σας στο Kaggle, και το σκορ σας στον διαγωνισμό. Επίσης, αναφέρετε και πειράματα που κάνατε και δεν δούλεψαν, ή πως βελτιώσατε μια λύση που δεν απέδιδε καλά (περιλάβετε και αποτελέσματα από τα πειράματα). Παραδώστε όλο τον κώδικα σας.