Πανεπιστήμιο Πατρών
Τμήμα Οικονομικών Επιστημών
Πρόγραμμα Μεταπτυχιακών Σπουδών «Εφαρμοσμένη Οικονομική και Ανάλυση Δεδομένων»

Ακαδημαϊκό έτος 2023- 2024

3η Εργασία μαθήματος «Διαχείριση Μεγάλων Δεδομένων»

Γιατί η λογοκρισία είναι κακό πράγμα:
Original: *https://www.youtube.com/watch?v=3e7yYBDHOgg*
Λογοκριμένο: *https://www.youtube.com/watch?v=B-Wd-Q3F8KM*

Εισαγωγή

Σκοπός της εργασίας είναι να αποκτήσετε μία εξοικείωση με τη χρήση αλγορίθμων στην περιοχή των μεθόδων εκτίμησης συντελεστών μοντέλων παλινδρόμησης, συσταδοποίησης και ανάλυσης συσχετίσεων. Η υλοποίηση των αλγορίθμων θα πρέπει να γίνει με εργαλείο R ή/και την Python, όπου αυτό ζητείται.

Θέμα 1: Παλινδρόμηση

Κατεβάστε από τον ιστότοπο "UCI Machine Learning Repository" και ειδικότερα από την ιστοσελίδα https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime το σύνολο δεδομένων που περιέχει παρατηρήσεις σχετικά με την εγκληματικότητα ανά 100000 κατοίκους σε περιοχές των ΗΠΑ (μεταβλητή ViolentCrimesPerPop) μαζί με κοινωνικοοικονομικά στοιχεία για την κάθε περιοχή (Communities and Crime Data Set). Το σύνολο δεδομένων βρίσκεται στο αρχείο communities.data, που θα το βρείτε εάν στην παραπάνω σελίδα ακολουθήσετε τον σύνδεσμο "Data Folder". Η ιστοσελίδα παρέχει και πληροφορίες για τις μεταβλητές του αρχείου δεδομένων. Τις ίδιες πληροφορίες μπορείτε να τις βρείτε και στο αρχείο "Data Set Description" από την παραπάνω ιστοσελίδα για να δείτε σε ποια σειρά και τί συλλαμβάνει κάθε μεταβλητή του συνόλου δεδομένων.
Αφού εξοικειωθείτε με το σύνολο δεδομένων, τα γνωρίσματα και τη σημασία τους, εκτιμήστε τους συντελεστές του παρακάτω μοντέλου παλινδρόμησης

$$ViolentCrimesPerPop = \beta_1 medIncome + \beta_2 whitePerCap + \beta_3 blackPerCap + \beta_4 HispPerCap + \beta_5 NumUnderPov + \beta_6 PctUnemployed + \beta_7 HousVacant + \beta_8 MedRent + \beta_9 NumStreet + \beta_0$$

με τους τρόπους που ζητούνται παρακάτω:

i. Συγγράψτε πρόγραμμα στην **R και στην Python** το οποίο εκτιμά τους συντελεστές του παραπάνω γραμμικού μοντέλου παλινδρόμησης με τη μέθοδο των ελαχίστων τετραγώνων (OLS) χρησιμοποιώντας το σύνολο δεδομένων Communities and Crime που έχετε κατεβάσει. Τα προγράμματά σας θα πρέπει να εμφανίζουν στην οθόνη τις τιμές των συντελεστών που έχουν εκτιμηθεί. Τα προγράμματά σας θα πρέπει επίσης να αφαιρούν όσες παρατηρήσεις έχουν τουλάχιστον μία τιμή (σε οποιαδήποτε μεταβλητή) που λείπει (missing value) κατά τη διαδικασία προεπεξεργασίας. Ακολουθήστε τέτοια προεπεξεργασία των δεδομένων για όλα τα θέματα της εργασίας αυτής.

ii. Συγγράψτε πρόγραμμα **μόνο σε Python** που εκτιμά τους συντελεστές του παραπάνω γραμμικού μοντέλου παλινδρόμησης με τη μέθοδο της Σταδιακής Καθόδου Δέσμης (Batch Gradient Descent) και χρησιμοποιώντας το σύνολο δεδομένων Communities and Crime Data Set.
Προς τούτο, στο πρόγραμμά σας Python, δημιουργείστε μία συνάρτηση με όνομα batchGradientDescent που δέχεται τις ακόλουθες παραμέτρους και η οποία θα υπολογίζει τους συντελεστές ενός πολλαπλού μοντέλου γραμμικής παλινδρόμησης με μέθοδο της Σταδιακής Καθόδου Δέσμης:

def batchGradientDescent(independentVars, dependentVar, thetas, alpha=0.01, numIters=100):

όπου *independentVars* η μήτρα των τιμών των ανεξάρτητων μεταβλητών, *dependentVar* η μήτρα των τιμών της εξαρτημένης μεταβλητής, *thetas* ένα διάνυσμα με τις αρχικές τιμές των συντελεστών θ, *alpha* η τιμή της παραμέτρου μάθησης α και *numIters* το πλήθος των επαναλήψεων που θα πρέπει να κάνει η μέθοδος. Η υλοποίηση της μεθόδου της Σταδιακής Καθόδου Δέσμης (Batch Gradient Descent) θα πρέπει να γίνει από την αρχή (“from scratch”) χρησιμοποιώντας τον τύπο ενημέρωσης των συντελεστών για την εκτίμηση των συντελεστών γραμμικών μοντέλων παλινδρόμησης. Η συνάρτηση δεν θα πρέπει να κάνει χρήση υπάρχουσας βιβλιοθήκης Python, που να παρέχει έτοιμη τη μέθοδο της Σταδιακής Καθόδου Δέσμης. Το κριτήριο τερματισμού της μεθόδου Σταδιακής Καθόδου Δέσμης, είναι το πλήθος των επαναλήψεων.
Η διαδικασία της εκτίμησης των συντελεστών θα πρέπει να τερματίζει αφού έχει εκτελεστεί το πλήθος των επαναλήψεων που προσδιορίζεται από το όρισμα numIters. Η συνάρτηση batchGradientDescent που θα δημιουργήσετε θα πρέπει να επιστρέφει τόσο ένα διάνυσμα με τους συντελεστές που εκτιμήθηκαν όσο και τις τιμές της συνάρτησης κόστους σε κάθε επανάληψη της μεθόδου της Σταδιακής Καθόδου Δέσμης. Για την διευκόλυνσή σας, σας δίνεται στην ιστοσελίδα του μαθήματος στο eclass (https://eclass.upatras.gr/modules/units/?course=ECON1332&id=9946) έναν σκελετό κώδικα σε Python (αρχείο Python-BatchGradientDescent-Template.rar), προκειμένου να ξεκινήσετε από εκεί την υλοποίηση της μεθόδου Σταδιακής Καθόδου Δέσμης συμπληρώνοντας όσα τμήματα λείπουν.

Αφού έχετε υλοποιήσει τη συνάρτηση batchGradientDescent, χρησιμοποιείστε την για να εκτιμήσετε τους συντελεστές του γραμμικού μοντέλου παλινδρόμησης που αναφέρεται παραπάνω για τα ίδιο σύνολο δεδομένων (Communities and Crime) . Θέστε τις κατάλληλες τιμές για την παράμετρο μάθησης α και το πλήθος επαναλήψεων numIters για την εκτίμηση των συντελεστών. Το πρόγραμμά σας θα πρέπει, αφού έχουν εκτιμηθεί οι συντελεστές, να εμφανίζει:

a) τις τιμές των συντελεστών που εκτιμήθηκαν και

b) να απεικονίζει με γραφική παράσταση τις τιμές της συνάρτησης κόστους ως συνάρτηση του πλήθους επαναλήψεων ώστε να τεκμηριωθεί ότι επιλέξατε σωστά τις τιμές για την παράμετρο μάθησης α και το πλήθος επαναλήψεων.

Επιπλέον, συγκρίνετε τους συντελεστές που εκτιμήθηκαν με τη μέθοδο της Σταδιακής Καθόδου Δέσμης με τους συντελεστές που εκτιμήθηκαν από τη μέθοδο των ελαχίστων τετραγώνων (OLS) στο υποερώτημα i). Τί παρατηρήσεις/σχόλια μπορείτε να κάνετε;

Θέμα 2: Παλινδρόμηση

Συγγράψτε πρόγραμμα **σε R και Python** που εκτιμά τους συντελεστές ενός γραμμικού μοντέλου παλινδρόμησης με τη μέθοδο της **Στοχαστικής Σταδιακής Καθόδου (Stochastic Gradient Descent)**.

Χρησιμοποιείστε τα προγράμματα σε R και Python που έχετε συγγράψει, θέτοντας τις κατάλληλες τιμές για την παράμετρο μάθησης α και το πλήθος επαναλήψεων, και εκτιμήστε τους συντελεστές του γραμμικού μοντέλου παλινδρόμησης που αναφέρεται στο θέμα 1 της τρέχουσας εργασίας και για το ίδιο σύνολο δεδομένων (Communities and Crime Data Set). Τα προγράμματά σας σε R και Python θα πρέπει να εμφανίζουν στην οθόνη:

a) τους συντελεστές που έχουν εκτιμηθεί και

b) τη γραφική παράσταση των τιμών της συνάρτησης κόστους όπως αυτές έχουν προκύψει από την εκτέλεση της μεθόδου Στοχαστικής Σταδιακής Καθόδου που έχετε υλοποιήσει.

Τί παρατηρήσεις μπορείτε να κάνετε τόσο για τους συντελεστές όσο και για τη γραφική παράσταση των τιμών της συνάρτησης κόστους που έχουν προκύψει με τη μέθοδο της Στοχαστικής Σταδιακής Καθόδου εάν τους συγκρίνετε με τους συντελεστές και τις τιμές της συνάρτησης κόστους που προέκυψαν στο υποερώτημα ii) του θέματος 1 της τρέχουσας εργασίας;

Θέμα 3: Παλινδρόμηση

Από το αρχείο με όνομα «Κεφάλαιο-06-Ασκήσεις.pdf» που μπορείτε να το βρείτε στην ενότητα “Lecture 3: Regression analysis” στον ιστότοπο του μαθήματος (https://eclass.upatras.gr/modules/units/?course=ECON1332&id=9946), απαντήστε στα **ζητούμενα της άσκησης 19**. Απαντήστε **μόνο στα υποερωτήματα i) και ii) της άσκησης 19**.

ΣΗΜΕΙΩΣΗ: Η απάντηση στο υποερώτημα iii) της άσκησης 19 είναι προαιρετική.

Θέμα 4: Παλινδρόμηση

Από το αρχείο με όνομα «Κεφάλαιο-06-Ασκήσεις.pdf» που μπορείτε να το βρείτε στην ενότητα "Lecture 3: Regression analysis" στον ιστότοπο του μαθήματος (https://eclass.upatras.gr/modules/units/?course=ECON1332&id=9946), απαντήστε στα **ζητούμενα της άσκησης 58**.

ΣΗΜ: Μπορείτε εσείς να καθορίσετε το πλήθος των μεταβλητών στα πλασματικά δεδομένα καθώς και το γραμμικό μοντέλο παλινδρόμησης. Για την εκτίμηση των συντελεστών του γραμμικού μοντέλου θα πρέπει να γίνεται χρήση των συνθετικών δεδομένων που έχετε παράξει. Για την εκτίμηση των συντελεστών, μπορείτε να κάνετε χρήση της μεθόδου OLS.

Θέμα 5: Παλινδρόμηση

Κατεβάστε από την ιστοσελίδα https://archive.ics.uci.edu/ml/datasets/Forest+Fires σύνολο δεδομένων για πυρκαγιές από περιοχές της Πορτογαλίας. Τα δεδομένα περιέχουν γεωγραφικά και μετεωρολογικά στοιχεία όταν εκδηλώθηκαν πυρκαγιές καθώς επίσης και την επιφάνεια που κάηκε που μετριέται σε εκτάρια[1] (hectars). Έχοντας ως στόχο την πρόβλεψη της επιφάνειας που θα καεί βάσει των μετεωρολογικών συνθηκών που επικρατούν, συγγράψτε κώδικα **σε R και Python** που εκτιμά τους συντελεστές του παρακάτω μοντέλου παλινδρόμησης και κάνει μια εκτίμηση της ακρίβειας πρόβλεψης του μοντέλου

$$area = \beta_1 temp + \beta_2 wind + \beta_3 rain + \beta_0$$

Ειδικότερα ζητούνται τα εξής:

i. Χρησιμοποιώντας όλες τις παρατηρήσεις στο αρχείο που έχετε κατεβάσει, κάντε χρήση διασταυρωτικής επικύρωση 10-πτυχών (10-Fold Cross Validation) κατά την οποία θα εκτιμώνται οι συντελεστές του παραπάνω μοντέλου παλινδρόμησης με τη μέθοδο των Ελαχίστων Τετραγώνων (OLS) και επιπλέον θα υπολογίζει το μέσο τετραγωνικό σφάλμα (Root Mean Squared Error – RMSE) της πρόβλεψης. Το πρόγραμμά σας θα πρέπει να εμφανίζει το μέσο τετραγωνικό σφάλμα (RMSE).

ii. Εκτιμήστε πάλι τους συντελεστές του ίδιου μοντέλου παλινδρόμησης με τη μέθοδο των Ελαχίστων Τετραγώνων και διασταυρωτικής επικύρωσης 10-πτυχών (10-Fold Cross Validation), αλλά αυτή τη φορά *μην χρησιμοποιείστε ολόκληρο το σύνολο δεδομένων για εκπαίδευση και αξιολόγηση* αλλά μόνο εκείνες τις παρατηρήσεις όπου η τιμή της εξαρτημένης μεταβλητής (μεταβλητή area) είναι μικρότερη από 3.2 εκτάρια (δηλαδή area < 3.2) και χαρακτηρίζει μικρές πυρκαγιές. Εμφανίστε το μέσο τετραγωνικό σφάλμα (Root Mean Squared Error – RMSE) της πρόβλεψης.

Τί συμπέρασμα μπορείτε να βγάλετε σχετικά με την ακρίβεια της πρόβλεψης, αν συγκρίνετε τα μέσα τετραγωνικά σφάλματα που εκτιμήσατε στις περιπτώσεις i) και ii) παραπάνω;

Για την υλοποίηση της διασταυρωτικής επικύρωσης k-πτυχών στο περιβάλλον της R, μελετήστε από το κεφάλαιο 06Regression.pdf που υπάρχει στον χώρο του μαθήματος στο eclass, την ενότητα *«6.7.2 Γραμμικά μοντέλα γραμμικής παλινδρόμησης με στόχο την πρόβλεψη»*. Επιπλέον, ο κώδικας R που υλοποιεί διασταυρωτική επικύρωση k-πτυχών και αναφέρεται στην

[1] 1 εκτάριο = 10 στρέμματα

ενότητα 6.7.2 του κεφαλαίου 06Regression.pdf, υπάρχει διαθέσιμος και στον χώρο του μαθήματος στο eclass στο αρχείο R-k-FoldCrossValidation.rar στην ιστοσελίδα της ενότητας (https://eclass.upatras.gr/modules/units/?course=ECON1332&id=9946). Στην ίδια ενότητα υπάρχει και κώδικας που υλοποιεί τη μέθοδο της διασταυρωτικής επικύρωσης k-πτυχών στην Python (αρχείο Python-k-foldCrossValidation.rar). Μπορείτε να χρησιμοποιήσετε αυτόν τον πηγαίο κώδικα για την δημιουργία των προγραμμάτων σας, αφού κάνετε (προφανώς) τις απαραίτητες αλλαγές.

Θέμα 6: Παλινδρόμηση

Αναζητήστε πληροφορίες και παρουσιάστε τη μέθοδο μηχανικής μάθησης που είναι γνωστή με όνομα **Gradient Boosted Regression Trees**. Η παρουσίαση της μεθόδου θα πρέπει να έρθει με την μορφή κειμένου με τουλάχιστον 1000 λέξεις.
Ειδικότερα, η παρουσίαση που θα κάνετε θα πρέπει να απαντά στα εξής ερωτήματα:

1) Σε τί είδους δεδομένα μπορεί να χρησιμοποιηθεί η μέθοδος αυτή;
2) Ποιο πρόβλημα επιχειρεί να επιλύσει/αντιμετωπίσει η μέθοδος;
3) Περιγράψτε συνοπτικά τη λειτουργία της μεθόδου αυτής
4) Αναφέρετε σε ποιες περιπτώσεις χρησιμοποιείται η μέθοδος. Προς τούτο, αναζητήστε στο διαδίκτυο και παραθέστε τουλάχιστον 3 δημοσιεύσεις από την περιοχή της οικονομικής επιστήμης που κάνει χρήση της μεθόδου αυτής και σχολιάστε αυτές σύντομα.

Τέλος, μπορείτε να αναφέρετε οποιοδήποτε άλλο στοιχείο κρίνετε εσείς σκόπιμο για την παρουσίαση της μεθόδου[2].

Θέμα 7: Συσταδοποίηση

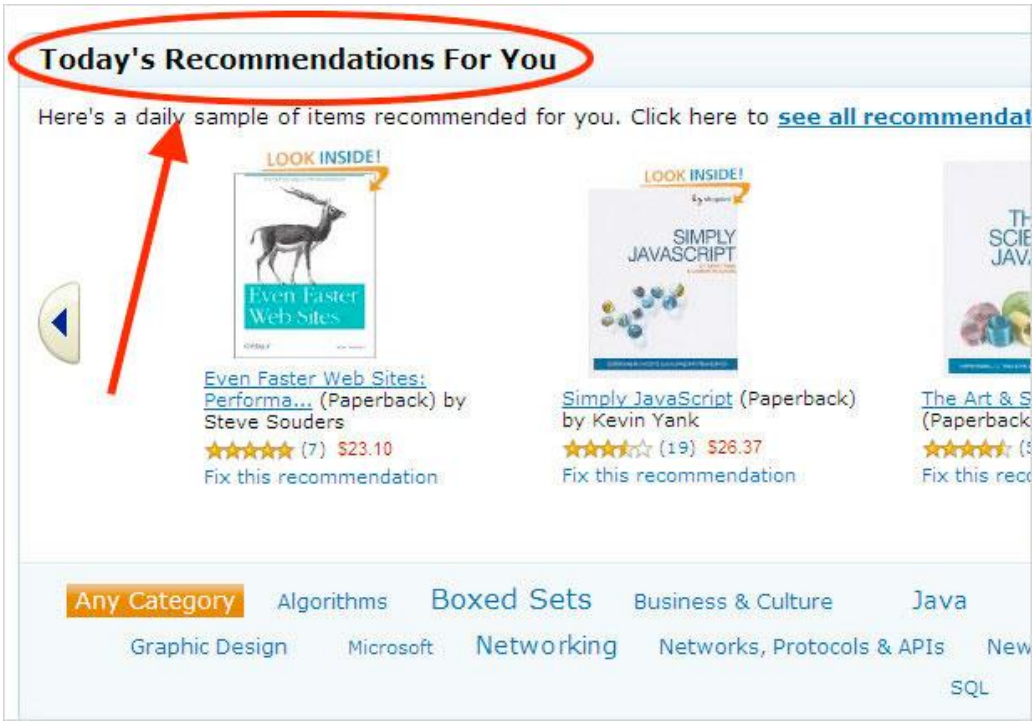
Στόχος του θέματος αυτού είναι η εξοικείωση με θέματα συσταδοποίησης (clustering). Ειδικότερα θα κάνετε χρήση των αλγορίθμων K-means (και ειδικότερα μία εκδοχή του για κατηγορικά δεδομένα) και Hierarchical clustering για την αντιμετώπιση των ζητημάτων που αναφέρονται παρακάτω.

Ζητούνται τα εξής:

I. **<u>CAVEAT: Μην τρομάξετε με την έκταση της εκφώνησης του ερωτήματος αυτού. Αναφέρει αναλυτικά τα βήματα που θα πρέπει να εκτελέσετε .</u>**
Στο ερώτημα αυτό σας ζητείται να φτιάξετε ένα σύστημα που ανήκει στην κατηγορία των προτάσεων/συστάσεων για ταινίες (movie recommender system[3]) με χρήση της γλώσσας R και Python.
Τα συστήματα προτάσεων ή συστάσεων (recommender systems - https://en.wikipedia.org/wiki/Recommender_system) είναι συστήματα, τα οποία προτείνουν ή συστήνουν σε χρήστες νέα προϊόντα βάσει είτε των προτιμήσεών τους είτε βάσει προϊόντων που έχουν αγοράσει στο παρελθόν και τους άρεσαν. Ο στόχος τέτοιων συστημάτων προτάσεων/συστάσεων είναι να παρέχουν εξατομικευμένες υπηρεσίες στους χρήστες. Όλα τα σύγχρονα συστήματα ηλεκτρονικού εμπορίου όπως Amazon και eBay παρέχουν τέτοιου είδους συστήματα προτάσεων/συστάσεων.

[2] Μπορείτε, εάν το κρίνετε σκόπιμο, να κάνετε χρήση εικόνων και δεδομένων που θα βρείτε στο διαδίκτυο υπό την προϋπόθεση ότι 1) ο δημιουργός έχει δώσει την άδεια χρήσης του υλικού αυτού σε έργα τρίτων και 2) κάνετε αναφορά στην πηγή προέλευσης του υλικού.

[3] Τα σύγχρονα συστήματα συστάσεων σήμερα βασίζονται σε διαφορετικές και πιο εξελιγμένες μεθόδους μηχανικής μάθησης και πρόβλεψης αξιολόγησης. Η μέθοδος που χρησιμοποιείται εδώ είμαι μία από τις διαθέσιμες που βολεύει για στατικά κυρίως δεδομένα.



Εικόνα 1: Παράδειγμα υπηρεσίας προτάσεων/συστάσεων βιβλίων στο Amazon. Τα αποτελέσματα που βλέπετε στην παραπάνω εικόνα, έχουν προκύψει από ένα σύστημα προτάσεων (recommender system).

Στα πλαίσια του ερωτήματος αυτού, σας ζητείται να συγγράψετε πρόγραμμα σε R και σε python, το οποίο προτείνει νέες ταινίες σε έναν χρήστη βάσει των προτιμήσεων ταινιών του συγκεκριμένου χρήστη. Για τον σκοπό αυτόν, σας δίνονται μαζί με την εκφώνηση δύο αρχεία δεδομένων: *movies.csv* και *ratings.csv*.

Το αρχείο *movies.csv* είναι αρχείο τύπου csv (comma separated values) το οποίο περιέχει τους τίτλους 9125 ταινιών μαζί με τις κατηγορίες στην οποία ανήκει κάθε ταινία (αν είναι περιπέτεια, δράμα, τρόμου, Film-Noir κλπ). Κάθε γραμμή του αρχείου movies.csv είναι της μορφής

<movieId>,<title>,Action,Adventure,Animation,Children,Comedy,Crime,Documentary,
Drama,Fantasy,Film-Noir,Horror,IMAX,Musical,Mystery,Romance,Sci-
Fi,Thriller,War,Western, (no genres listed)

η οποία ερμηνεύεται ως εξής: η ταινία με κωδικό <movieId> έχει τίτλο <title>[4] και η οποία ανήκει σε μία ή παραπάνω από τις ακόλουθες κατηγορίες: Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, IMAX, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western, (no genres listed) . Οι μεταβλητές που δηλώνουν κατηγορίες καλούνται μεταβλητές κατηγοριών και είναι δυαδικές μεταβλητές (dummy variables) λαμβάνοντας μόνο τις τιμές 0 ή 1 προσδιορίζοντας συνολικά τις κατηγορίες στις οποίες η ταινία ανήκει. Η ύπαρξη άσσου (1) σε μία μεταβλητή κατηγορίας σημαίνει ότι ανήκει στην κατηγορία αυτή. Η τιμή 0 σημαίνει ότι δεν ανήκει στην κατηγορία αυτή. Μία ταινία μπορεί να ανήκει σε περισσότερες από μία κατηγορία. Για παράδειγμα η γραμμή του αρχείου movies.csv

2,Jumanji (1995),0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0

[4] Μαζί με τον τίτλο της κάθε ταινίας εμφανίζεται –εντός παρενθέσεων- και το έτος πρώτης προβολής της.

σημαίνει ότι η ταίνια με κωδικό 2 έχει τίτλο “Jumanji (1995)” και ανήκει στις κατηγορίες Adventure, Children και Fantasy αφού οι αντίστοιχες μεταβλητές έχουν τιμή 1 ενώ π.χ. δεν ανήκει στην κατηγορία Western αφού στην αντίστοιχη μεταβλητή υπάρχει η τιμή 0. Αν υπάρχει τιμή 1 στη μεταβλητή “(no genres listed)” αυτό σημαίνει ότι η κατηγορία της αντίστοιχης ταινίας είναι είτε άγνωστη είτε δεν μπορεί να προσδιοριστεί από τις υπόλοιπες μεταβλητές κατηγοριών. Μία ταινία μπορεί να ανήκει σε πολλές κατηγορίες όπως για παράδειγμα η ταινία Jumanji (1995) παραπάνω. Κάθε ταινία έχει μοναδικό κωδικό (movieId) που φαίνεται στο αρχείο (movieId).

Το αρχείο *ratings.csv* περιέχει αξιολογήσεις ταινιών από 671 διακριτούς χρήστες. Συνολικά περιέχει 100004 αξιολογήσεις από τους 671 χρήστες. Κάθε γραμμή του αρχείου έχει τη μορφή

<userId>, <movieId>, <rating>,<timestamp>

η οποία ερμηνεύεται ως εξής: *ο χρήστης με κωδικό <userId> αξιολόγησε με βαθμό <rating> την ταινία με κωδικό <movieId> και η αξιολόγηση έγινε στις <timestamp>*. Η τιμή <timestamp> καθορίζει την χρονική στιγμή που έγινε η αξιολόγηση και αναφέρεται στην ημερομηνία και ώρα. Ειδικότερα, η τιμή <timestamp> είναι ένας ακέραιος αριθμός που εκφράζει πόσα δευτερόλεπτα πέρασαν από τα μεσάνυχτα της 1ης Ιανουαρίου 1970 (η οποία είναι γνωστή και με το όνομα “the Epoch”) μέχρι τη στιγμή που έγινε η αξιλόγηση. Για παράδειγμα η παρακάτω γραμμή του αρχείου ratings.csv:

3, 296, 4.5, 1298862418

ερμηνεύεται ως εξής: ο χρήστης με κωδικό 3 αξιολόγησε την ταινία με κωδικό 296 με βαθμολογία 4.5 και ότι η αξιολόγηση έγινε τη χρονική στιγμή 1298862418 (δηλαδή η αξιλόγηση έγινε 1298862418 δευτερόλεπτα μετά τα μεσάνυχτα της 1ης Ιανουαρίου 1970). Οι κωδικοί ταινιών του αρχείου ratings.csv αναφέρονται στο αρχείο movies.csv και έτσι μπορούμε να δούμε ότι η ταινία με κωδικό 296 της παραπάνω γραμμής του αρχείου ratings.csv αναφέρεται στην ταινία “Pulp Fiction (1994)” όπως προκύπτει από το αρχείο movies.csv. Δηλαδή ο χρήστης με κωδικό 3 βαθμολόγησε με 4.5 την ταινία “Pulp Fiction”. Η αξιολόγηση (rating) γίνεται σε κλίμακα από 1 έως και 5, με 1 να είναι η χαμηλότερη βαθμολογία και 5 η υψηλότερη. Μπορούν επίσης να δοθούν ως βαθμολογία «μισά» δηλαδή 2.5, 3.5 κλπ. Ένας χρήστης μπορεί να βαθμολογήσει παραπάνω από μία ταινία, αλλά ένας συγκεκριμένος χρήστης μπορεί να βαθμολογήσει μία συγκεκριμένη ταινία μία μόνο φορά. Στα πλαίσια της εργασία αυτής, μπορείτε να αγνοήσετε την χρονική στιγμή της βαθμολόγησης <timestamp> του αρχείου ratings.csv.

Ο κώδικάς σας που θα συγγράψετε σε R και σε python θα πρέπει να επεξεργάζεται καταλλήλως τα αρχεία movies.csv και ratings.csv και για έναν συγκεκριμένο κωδικό χρήστη (που θα τον δίνετε εσείς στο πρόγραμμά σας), να εμφανίζει στην οθόνη ταινίες, που ταιριάζουν στις προτιμήσεις του συγκεκριμένου χρήστη και δεν τις έχει δει.

Αν και υπάρχουν διάφορες προσεγγίσεις για να φτιαχτεί ένα τέτοιο σύστημα προτάσεων/συστάσεων για το συγκεκριμένο πρόβλημα ταινιών, το σύστημα προτάσεων ταινιών που θα φτιάξετε θα βασιστεί στην εξής προσέγγιση:

Προτείνει στον χρήστη με κωδικό X να δει εκείνες τις ταινίες T που δεν έχει δει και οι οποίες μοιάζουν αρκετά με ταινίες τις οποίες τις έχει δει και του άρεσαν πολύ[5].

[5] Το 2009 η Netflix έτρεξε διαγωνισμό για την πρόβλεψη του κατά πόσο ένας χρήστης θα του άρεσε μία ταινία, γνωστό ώς Netflix Prize. Ειδικότερα, ο στόχος ήταν να βρεθεί αλγόριθμος ο οποίος μπορεί να βελτιώσει την πρόβλεψη αξιολόγησης άγνωστων ταινιών βάσει προηγούμενων αξιολογήσεων χρηστών κατά τουλάχιστον 10% σε σχέση με τον αλγόριθμο που ήδη είχε η Netflix. Το έπαθλο ήταν 1.000.000 Ευρώ . Για περισσότερες πληροφορίες δείτε https://en.wikipedia.org/wiki/Netflix_Prize και http://www.netflixprize.com/

Το σύστημα προτάσεων/συστάσεων που θα υλοποιήσετε σε R και python θα κάνει χρήστη του αλγορίθμου συσταδοποίησης K-means.

Παρακάτω περιγράφεται με λόγια ένας τέτοιος αλγόριθμος και τον οποίο θα πρέπει να υλοποιήσετε τόσο με τη γλώσσα προγραμματισμού R όσο και με τη γλώσσα προγραμματισμού python:

1) Διαβάστε το αρχείο ταινιών movies.csv
2) Κάντε συσταδοποίηση των ταινιών που διαβάσατε από το αρχείο movies.csv βάσει των κατηγοριών στις οποίες αυτές ανήκουν με τον αλγόριθμο K-means. Ο στόχος της συσταδοποίησης με τον αλγόριθμο K-means είναι, ταινίες που ανήκουν στις ίδιες κατηγορίες (και κατά συνέπεια μοιάζουν μεταξύ τους) να μπουν στην ίδια συστάδα. Θα κάνετε τη συσταδοποίηση με τον αλγόριθμο K-means λαμβάβοντας μόνο υπόψιν τις μεταβλητές που υποδηλώνουν τις κατηγορίες της τανίας από τα δεδομένα του αρχείου movies.csv (δηλαδή τις ψευδομεταβλητές Action, Adventure, Animation, Children κλπ). Ωστόσο, επειδή οι μεταβλητές κατηγοριών των ταινιών δεν λαμβάνουν συνεχείς τιμές (λαμβάνουν δυαδικές τιμές 0 και 1 και κατά συνέπεια είναι αυτό που καλούμε εικονικές μεταβλητές ή ψευδομεταβλητές - dummy variables) δεν μπορεί να γίνει χρήση συναρτήσεων συσταδοποίησης που βασίσζονται στην Ευκλείδεια απόσταση. Γι'αυτόν τον λόγο θα πρέπει τόσο στο περιβάλλον της R όσο και στο περιβάλλον python να βρείτε και να εγκαταστήσετε τις κατάλληλες βιβλιοθήκες, οι οποίες θα σας επιτρέψουν να τρέξετε τον αλγόριθμο K-means με εκείνη τη μέθοδο υπολογισμού αποστάσεων των δεδομένων κατάλληλη για τα δεδομένα του αρχείου ratings.csv. Σας παραπέμπουμε στις βιβλιοθήκες amap της R και Kmodes της python τις οποίες θα πρέπει να εγκαταστήσετε στο δικό σας υπολογιστή και σας παρέχουν τις κατάλληλες εκδοχές του αλγορίθμου K-means για κατηγορικά δεδομένα. Δώστε ιδιαίτερη έμφαση στα εγχειρίδια χρήσης των βιβλιοθηκών αυτών για το ποια συνάρτηση να χρηστιμοποιήσετε και με ποια ορίσματα να την εκτελέσετε.
   Για τον προσδιορισμό του ακριβούς πλήθους των συστάδων K τόσο στο πρόγραμμα R όσο και στο πρόγραμμα python, κάντε χρήση της μεθόδου του αγκώνα (Elbow method). Ειδικότερα, τρέξτε τον αλγόριθμο συσταδοποίησης K-means με όλες τις τιμές K (κέντρων) από 2 έως και 100. Για κάθε τιμή K που θα εκτελέσετε τον αλγόριθμο K-means (K=2, 3,4, 5, … ,100) κρατήστε την τιμή της αντικειμενικής συνάρτησης που σας λέει πόσο καλή ήταν η συσταδοποίηση για την συγκεκριμένη τιμή K (όπως Sum of Squared Error, Average dispersion κλπ – μπορείτε εσείς να επιλέξετε την αντικειμενική συνάρτηση). **Απεικονίστε γραφικά τις τιμές K μαζί με την τιμή της αντικειμενικής συνάρτησης που επιλέξατε και επιλέξτε εκείνη την τιμή K η όποια παρουσιάζει τη μεγαλύτερη μείωση της αντικειμενικής συνάρτησης και συνεχίζει με μη-σημαντικές μεταβολές**. Στο γράφημα αυτό θα εφαρμόσετε τη μέθοδο του αγκώνα για τον προσδιορισμό της τιμής K (κέντρων).
3) Μόλις καταλήξετε στην κατάλληλη τιμή K (κέντρων) με την οποία προκύπτει η καλύτερη τιμή της αντικειμενικής συνάρτησης, ξανατρέξτε τον αλγόριθμο K-means() με την επιλεγείσα τιμή K για να πάρετε τις τελικές συστάδες των δεδομένων.
4) Διαβάστε το αρχείο αξιολογήσεων ratings.csv και βρείτε για κάθε ταινία του αρχείου movies.csv, τον μέσο όρο αξιολόγησης που έδωσαν σε αυτήν οι χρήστες. Θεωρείστε ότι οι τιμές αξιολόγησης ταινιών είναι ποσοτικό μέγεθος και ότι μπορεί να υπολογιστεί ο αριθμητικός μέσος όρος[6]. Σε περίπτωση που δεν θέλετε να κάνετε χρήση του μέσου όρου, μπορείτε να υπολογίσετε την επικρατούσα τιμή (mode) για κάθε ταινία.
5) Επιλέξτε έναν συγκεκριμένο χρήστη π.χ. τον χρήστη με κωδικό 198 (ή οποινδήποτε άλλο). Βρείτε ποιες ταινίες έχει αξιολογήσει και βρείτε για κάθε ταινία που ο χρήστης με κωδικό 198 έχει αξιολογήσει, σε ποια συστάδα ανήκει, όπως

[6] Αν και η μεταβλητή ratings φαίνεται να είναι κατηγορική, μπορείτε να υπολογίσετε τον μέσο όρο, μιας και λαμβάνει και «μισές» τιμές π.χ. 2.5, 4.5 κλπ και κατά συνέπεια μπορεί να θεωρηθεί ότι συμπεριφέρεται σαν ποσοτική μεταβλητή.

αυτή προέκυψε από το βήμα 3) παραπάνω. Δηλαδή, από τα δεδομένα του αρχείου ratings.csv, απομονώστε τις αξιολογήσεις του χρήστη 198 και εισάγετε μία νέα μεταβλητή με όνομα *clusterId*. Για κάθε ταινία που έχει αξιολογήσει ο χρήστης 198, θέστε ως τιμή στη μεταβλητή *clusterId* τη συστάδα στην οποία αυτή η ταινία ανήκει και όπως αυτή προέκυψε από το βήμα 3).

6) Βρείτε τον μέσο όρο βαθμολογίας (ή την επικρατούσα τιμή αν θέλετε) που έχει δώσει ο χρήστης με κωδικό 198 στις συστάδες στις οποίες ανήκουν οι ταινίες που έχει αξιολογήσει. Ειδικότερα, ομαδοποιείστε[7] τις ταινίες που έχει αξιολογήσει ο χρήστης 198 βάσει της συστάδας στην οποία ανήκει κάθε ταινία, και για κάθε ομάδα βρείτε τον μέσο όρο βαθμολογίας των ταινιών που αξιολόγησε ο χρήστης 198 και που ανήκουν στην ομάδα αυτή. Με τον τρόπο αυτό μπορείτε να πάρετε μία άποψη του χρήστη για κάθε συστάδα. Για παράδειγμα αν ο χρήστης 198 έχει αξιολογήσει τις εξής ταινίες, οι οποίες ανήκουν στις παρακάτω συστάδες (βλέπε clusterId), όπως αυτές προέκυψαν από το βήμα 3):

userId	**movieId**	**rating**	**clusterId**
198	345	2.5	45
198	153	1.5	16
198	76	4.5	45
198	236	4.5	16
198	58	3.5	16

στο βήμα αυτό θα πρέπει να προκύψει το εξής αποτέλεσμα:

clusterId	**Μ.Ο. βαθμολογίας**
16	(1.5+4.5+3.5) / 3 = 3.16
45	(2.5 + 4.5) / 2 = 3.5

Το παραπάνω αποτέλεσμα ερμηνεύεται ως εξής: ο μέσος όρος βαθμολογίας ταινιών του χρήστη 198 για τη συστάδα 16 είναι 3.16 ενώ για τη συστάδα 45 είναι 3.5.

7) Ακολούθως, αφαιρέστε εκείνες τις συστάδες του χρήστη 198, οι οποίες έχουν μέσο όρο αξιολόγησης ταινιών χαμηλό. Επειδή το «χαμηλό» είναι σχετική έννοια, ορίστε ως χαμηλή αξιολόγηση μία τιμή αξιολόγησης μικρότερη από 3.5. Στο παραπάνω παράδειγμα του χρήστη 198, θα πρέπει να αφαιρεθεί η συστάδα 16, η οποία έχει μέσο όρο αξιολόγησης ταινιών μικρότερη από 3.5 από τον χρήστη. Η συστάδα 45 δεν θα πρέπει να αφαιρεθεί, μιας και έχει αξιολόγηση μεγαλύτερη ή ίση από 3.5.
8) Αν για τον χρήστη δεν υπάρχουν συστάδες με μέσο όρο αξιολόγησης ταινιών ίση ή μεγαλύτερη από 3.5 (γιατί π.χ. όλοι οι μέσοι όροι συστάδων είναι μικρότεροι από 3.5), τότε δεν μπορούν να γίνουν προτάσεις/συστάσεις ταινιών για τον χρήστη και θα πρέπει να εμφανίζεται το μήνυμα: “Sorry, no recommendations for you!”.
9) Αν υπάρχουν συστάδες με μέσο όρο αξιολόγησης μεγαλύτερη ή ίση με 3.5 για τον χρήστη 198, τότε για κάθε τέτοια συστάδα βρείτε τις 2 ταινίες με την υψηλότερη βαθμολογία εντός της συστάδας αυτής και τις οποίες δεν έχει δει ο χρήστης 198. Εμφανίστε τον τίτλο των ταινιών αυτών και αυτές οι ταινίες είναι οι συστάσεις/προτάσεις ταινιών για τον χρήστη 198 που μοιάζουν με τις προτιμήσεις του. Ειδικότερα, εμφανίστε το μήνυμα “You may also like the following movies” και από κάτω εμφανίστε τους τίτλους των ταινιών που προτείνονται στον χρήστη.

ΣΗΜΕΙΩΣΗ: Για το πρόγραμμα που θα συγγράψετε σε R, ενδεχομένως να σας φανούν χρήσιμες οι εξής συναρτήσεις της R: subset(), match(), aggregate() και order(). Ανατρέξτε στα σχετικά εγχειρίδια για να δείτε τι ακριβώς κάνουν και πως λειτουργούν οι συναρτήσεις αυτές.

Συνοψίζοντας τα ζητούμενα του θέματος αυτού, θα πρέπει να παραδώσετε τα εξής:

[7] Προσοχή! “Ομαδοποιείστε” όχι “συσταδοποιείστε”! Με τον όρο ομαδοποίηση εννοούμε να βάλετε στην ίδια ομάδα τις ταινίες που ανήκουν στην ίδια συστάδα.

a) **Κώδικα γραμμένο σε R**, ο οποίος υλοποιεί τον παραπάνω αλγόριθμο συστάσεων ταινιών σε χρήστη. Ο κώδικας σε R θα πρέπει να εμφανίζει τη γραφική παράσταση που αναφέρετε στο σημείο 2) και να εμφανίζει προτάσεις/συστάσεις ταινιών για έναν συγκεκριμένο χρήστη, που θα πρέπει να μπορεί να τον δίνει ο χρήστης του προγράμματος.
b) **Κώδικα γραμμένος σε Python**, ο οποίος υλοποιεί τον παραπάνω αλγόριθμο συστάσεων ταινιών σε χρήστη. Ο κώδικας σε python θα πρέπει να εμφανίζει τη γραφική παράσταση που αναφέρετε στο σημείο 2) και να εμφανίζει προτάσεις/συστάσεις ταινιών για έναν συγκεκριμένο χρήστη, που θα πρέπει να μπορεί να τον δίνει ο χρήστης του προγράμματος.

II. Συγγράψτε κώδικα σε **R και Python**, ο οποίος διαβάζει τα δεδομένα του αρχείου *europe.csv*, που δίνεται μαζί με την εκφώνηση της εργασίας και εκτελεί ιεραρχική συσταδοποίηση πάνω στο σύνολο δεδομένων του αρχείου europe.csv. Στο περιβάλλον της R κάντε χρήση των συναρτήσεων *dist()* και *hclust()* της R, ενώ στην python κάντε χρήση των κατάλληλων συναρτήσεων της βιβλιοθήκης scikit-learn που μπορείτε να βρείτε εδώ: http://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering και θα πρέπει να έχετε εγκαταστήσει στο περιβάλλον εργασίας σας.
Τόσο το πρόγραμμα σε R όσο και το πρόγραμμα σε python που θα φτιάξετε θα πρέπει να εμφανίζει ως έξοδος, το δενδρόγραμμα κλάσεων με ετικέτες το όνομα της κάθε χώρας.

III. Μελετήστε τη δημοσίευση που υπάρχει στο αρχείο **paper-8iii.pdf** και δώστε μία συνοπτική περίληψη. Η περίληψη θα πρέπει οπωσδήποτε να αναφέρει τις τεχνικές ανάλυσης δεδομένων που παρουσιάζονται καθώς και τον λόγο που έχουν χρησιμοποιηθεί αυτές. Μπορείτε να αναφέρετε οποιοδήποτε άλλο στοιχείο κρίνετε εσείς σκόπιμο.

Θέμα 8: Ανάλυση Συσχετίσεων

Στόχος του θέματος αυτού είναι η εξοικείωση με θέματα ανάλυσης συσχετίσεων (association rules) με τη χρήση του πακέτου *arules* του εργαλείου R[8]. Επειδή το πακέτο *arules* δεν είναι προεγκατεστημένο στο περιβάλλον της R, θα πρέπει να εγκατασταθεί και να χρησιμοποιηθεί στο σύστημά σας. Το εγχειρίδιο βοήθειας του πακέτου arules μπορείτε να το βρείτε εδώ: https://cran.r-project.org/web/packages/arules/arules.pdf
Για τις ανάγκες του θέματος αυτού, κατεβάστε το σύνολο δεδομένων “Fertility Data Set” από το UCI Machine Learning Repository, ακολουθώντας τον εξής σύνδεσμο: https://archive.ics.uci.edu/ml/datasets/Fertility . Η σελίδα περιέχει πληροφορίες σχετικά με την ερμηνεία των τιμών που υπάρχουν στο σύνολο δεδομένων. Επιπλέον στοιχεία για την ερμηνεία του συνόλου δεδομένων μπορούν να βρεθούν στην εξής δημοσίευση: http://cs229.stanford.edu/proj2014/Axel%20Guyon,Florence%20Koskas,Yoann%20Buratti,SemenFertilityPrediction.pdf

Το σύνολο δεδομένων “Fertility Data Set” περιέχει στοιχεία για την αξιολόγηση της γονιμότητας ανδρών μαζί με στοιχεία του τρόπου ζωής τους (lifestyle). Τα ερωτήματα που επιχειρεί μελετήσει το θέμα αυτό είναι να ανακαλύψει παράγοντες του τρόπου ζωής που συνεμφανίζονται με την αλλαγή της γονιμότητας ανδρών.

Ειδικότερα, ζητούνται τα εξής:

[8] Στο θέμα αυτό, δεν θα χρειαστεί να υλοποιήσετε τον αλγόριθμο σε Python. Θα συγγράψετε το πρόγραμμά σας μόνο στη γλώσσα R.

I. Αφαιρέστε από το σύνολο δεδομένων “Fertility Data Set” τις στήλες που σχετίζονται με τις μεταβλητές “Age at the time of analysis” και “Number of hours spent sitting per day ene-16”. Το σύνολο δεδομένων “Fertility Data Set” χωρίς τα δύο αυτά γνωρίσματα θα αναφέρεται ως τροποποιημένο σύνολο δεδομένων “Fertility Data Set”. Ακολούθως, συγγράψτε κώδικα σε R, ο οποίος εφαρμόζει τον αλγόριθμο Apriori πάνω σε όλα τα γνωρίσματα που απομένουν του τροποποποιημένου συνόλου δεδομένων με τις προκαθορισμένες (default) τιμές. Ποιο είναι το πλήθος των κανόνων που επιστρέφονται και τί δηλώνουν οι κανόνες που επιστρέφονται ως αποτέλεσμα από τον συγκεκριμένο αλγόριθμο;

II. Για το τροποποποιημένο σύνολο δεδομένων “Fertility Data Set”, συγγράψτε κώδικα σε R ο οποίος εφαρμόζει τον αλγόριθμο Apriori για ελάχιστη υποστήριξη (support threshold) ίση με 0.02, εμπιστοσύνη (confidence) ίση με 1. Επιπλέον δώστε ως περιορισμό στο δεξί μέλος των κανόνων να υπάρχει μόνο το Diagnosis=altered. Πόσοι και ποιοι κανόνες επιστρέφονται;

III. Ένας κανόνας Υ θεωρείται περιττός, όταν υπάρχει κανόνας Χ, ο οποίος έχει μεγαλύτερο ή ίσο lift από τον Υ και επιπλέον ο Υ είναι υπερκανόνας του Χ. Ένας κανόνας έχει την γενική μορφή LHS => RHS. Ο κανόνας Υ είναι υπερκανόνας του Χ αν LHS(Υ) ⊃ LHS(Χ) και RHS(Υ) == RHS(Χ). Για παράδειγμα, ο Α,Β => Γ είναι υπερκανόνας του Α=>Γ. Το lift για κάθε κανόνα δίνεται από τον αλγόριθμο Apriori.
Ταξινομήστε τους κανόνες που προέκυψαν από το ερώτημα ΙΙ. ως προς το lift και στη συνέχεια αφαιρέστε τους περιττούς κανόνες. Δώστε το σύνολο των κανόνων που απομένουν, μετά την αφαίρεση των περιττών κανόνων.

Ομάδες εργασίας

Η εργασία θα εκπονηθεί ομαδικά και θα είναι οι ίδιες ομάδες που εκπόνησαν τις εργασίες 1 και 2.

Χρόνος και Τρόπος Παράδοση της εργασίας

Κάθε ομάδα θα πρέπει να παραδώσει μία αναφορά σε αρχείο μορφής .pdf, γραμμένη σε LaTeχ, η οποία περιέχει τον κώδικα σε R και σε python, τις γραφικές παραστάσεις και τις απαντήσεις σας στα θέματα της εργασίας. Επιπλέον, ο κώδικας R και python που θα δημιουργήσετε για όλα τα θέματα, θα πρέπει να σταλεί και σε μορφή κειμένου, ώστε να μπορεί να εκτελείται από την R και το περιβάλλον της Python. **Τα προγράμματα σε R και python που θα συγγράψετε για να απαντήσετε στα ερωτήματα της εργασίας, θα πρέπει οπωσδήποτε να περιέχουν και σχόλια που θα βοηθούν στην κατανόησή του.**

Η καταληκτική ημερομηνία παράδοσης της 3ης Εργασίας είναι η 31η Ιανουαρίου 2024.

Ερωτήσεις/Απορίες

Για οποιαδήποτε ερώτηση ή απορία σχετικά με την εργασία μπορείτε να στείλετε email στη διεύθυνση tzagara@upatras.gr . Απορίες μπορούν επίσης (**και συστήνεται!**) να συζητηθούν κατά τη διάρκεια του μαθήματος.

Βαρύτητα της εργασίας

Η εργασία είναι υποχρεωτική και συνεισφέρει το 10% του τελικού βαθμού.

Καλή επιτυχία!