ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Τμήμα Μηχ. Η/Υ & Πληροφορικής

ΘΕΩΡΙΑ ΑΠΟΦΑΣΕΩΝ

Αναφορά Project 2020-2021

Εργαστηριακή Άσκηση 6

Ζούλφος Γεώργιος, 1047141

Καψουλάκης Ευάγγελος, 1047062

Ρήγας Στέφανος, 1047065

Περιεχόμενα

Σκοπός άσκησης

Ο σκοπός αυτής της άσκησης είναι η δημιουργία ενός συστήματος που να προτείνει ταινίες στους χρήστες σύμφωνα με τις ταινίες που έχουν παρακολουθήσει και τις αξιολογήσεις που έχουν κάνει. Τα υπόλοιπα δεδομένα που βρίσκονται στο dataset χρησιμοποιούνται για ανάλυση και οπτικοποίηση.

Στις παρενθέσεις των κεφαλίδων βρίσκονται τα ονόματα των αρχείων που ανταποκρίνονται στην εκάστοτε ενότητα.

Github: https://github.com/vg442/Decision-Theory

Σύστημα Πρότασης Ταινιών (user_based_filtering)

Περιγραφή Συστήματος

Για τη δημιουργία του συστήματος χρησιμοποιήθηκε η τεχνική του user-based collaborative filtering. Σύμφωνα με την τεχνική αυτή, προτείνονται νέες ταινίες σε ένα χρήστη σύμφωνα με τις προτιμήσεις που δείχνουν οι όμοιοι με αυτόν χρήστες. Στο σύστημα διακρίνονται δύο περιπτώσεις. Στην πρώτη περίπτωση ο χρήστης στον οποίο προτείνονται νέες ταινίες έχει ήδη παρακολουθήσει και αξιολογήσει κάποιο αριθμό ταινιών. Τότε η ομοιότητα προκύπτει από τις βαθμολογίες που έχουν δοθεί. Στη δεύτερη περίπτωση, που πρόκειται για κάποιο νέο χρήστη ο οποίος δεν έχει αξιολογήσει κάποια ταινία, του προτείνονται ταινίες με βάση το φύλο και την ηλικία του. Και στις δύο περιπτώσεις για την εύρεση των ομοίων χρηστών χρησιμοποιείται ο αλγόριθμος kNN.

Αλγόριθμος kNN

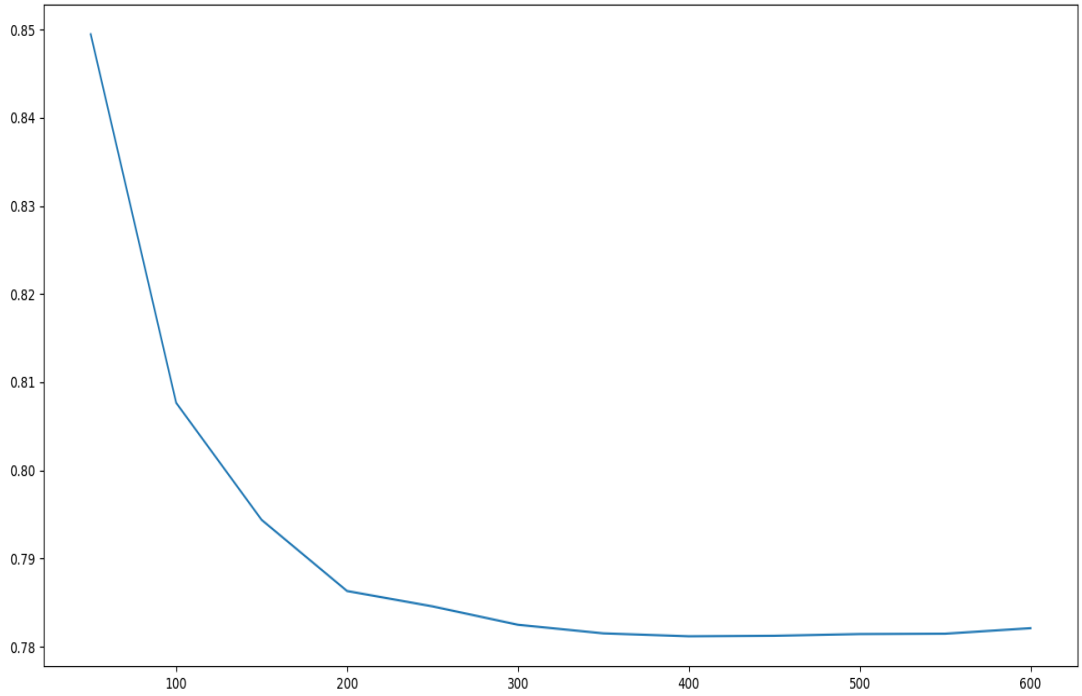
Για την εύρεση όμοιων χρηστών χρησιμοποιήθηκε ο αλγόριθμος kNN. Ο αλγόριθμος αυτός βρίσκει τους k πιο κοντινούς γείτονες για ένα διάνυσμα δεδομένων, βάσει κάποιας μετρικής απόστασης. Στην περίπτωση του χρήστη που έχει ήδη δει και αξιολογήσει ταινίες, κάθε χρήστης αναπαρίσταται ως ένα διάνυσμα 1xm, όπου m είναι το πλήθος των ταινιών του dataset που έχουν έστω και μια αξιολόγηση. Στην περίπτωση νέου χρήστη το διάνυσμα είναι μεγέθους 1x2, όπου στην πρώτη θέση βρίσκεται το φύλλο του χρήστη και στη δεύτερη η ηλικία του.

Αξιολόγηση Συστήματος και Επιλογή Κατάλληλου k

Τόσο για την αξιολόγηση του συστήματος όσο και για την επιλογή του βέλτιστου k στον αλγόριθμο kNN χρησιμοποιήθηκε το Μέσο Απόλυτο Σφάλμα (MAE). Το MAE δίνεται από τον τύπο:

$$MAE = \frac{\sum_{i=1}^{n} \left| y_i - yp_i \right|}{n}$$

όπου y_i είναι η βαθμολογία που έχει βάλει ένας χρήστης στην ταινία i, ενω yp_i η βαθμολογία που προβλέπει το σύστημα. Η διαδικασία που θα περιγραφεί στη συνέχεια αφορά μόνο την περίπτωση που ο χρήστης στον οποίο θα προταθούν ταινίες έχει ήδη παρακολουθήσει και αξιολογήσει κάποιες ταινίες. Αρχικά χωρίζεται το dataset σε σύνολα train και test με την αναλογία να είναι 1/9. Το σύνολο train δίνεται ως είσοδος στον αλγόριθμο kNN και αφου εκπαιδευτεί το μοντέλο, για κάθε χρήστη του συνόλου test προβλέπονται οι βαθμολογίες που θα βάλει στις ταινίες. Η διαδικασία αυτή έγινε για διάφορες τιμές του k έτσι ώστε να εντοπιστεί η κατάλληλη τιμή του, η οποία ελαχιστοποιεί το MAE.



Εικόνα 1

Στην *Εικόνα 1* φαίνεται η τιμή του MAE (άξονας y) σε σχέση με το πλήθος των κοντινότερων γειτόνων, το k (άξονας x). Σύμφωνα και με το παραπάνω σχήμα επιλέχθηκε k=300, καθώς από εκεί και μετά δεν υπάρχει αισθητή βελτίωση στην τιμή του MAE. Μάλιστα για k=600 η τιμή του σφάλματος δείχνει να έχει μια ανοδική τάση, πράγμα λογικό, καθώς στην διαδικασία της πρόβλεψης συμμετέχουν και χρήστες που τελικά δεν είναι και τόσο όμοιοι με το χρήστη του στον οποίο θα προταθούν οι ταινίες.

Περιγραφή Κώδικα

Ο κώδικας του συστήματος που περιγράφηκε παραπάνω αποτελείται από τρεις βασικές συναρτήσεις και τη main. Οι συναρτήσεις είναι οι εξής:

- `def trainModel(kValue, train, test):`

  Η συνάρτηση αυτή παίρνει ως είσοδο τον αριθμό των κοντινότερων γειτόνων που θα έχει το μοντέλο, το σύνολο χρηστών train και το σύνολο χρηστών test. Επιστρέφει το σφάλμα πρόβλεψης και το μοντέλο που εκπαιδεύτηκε.

- `def recommendMovies(user, knnModel, N):`

  Αυτή η συνάρτηση παίρνει ως είσοδο τον χρήστη στον οποίο θα προταθούν ταινίες, το μοντέλο που εκπαιδεύτηκε και τον αριθμό των ταινιών που θα προταθούν. Επιστρέφει τις N υψηλότερα βαθμολογημένες ταινίες σύμφωνα με το μοντέλο.

- `def noRatings(newUser, kValue, N):`

  Η συνάρτηση αυτή χρησιμοποιείται στην περίπτωση του χρήστη που δεν έχει αξιολογήσει κάποια ταινία. Παίρνει ως είσοδο το χρήστη στον οποίο θα προταθούν ταινίες, τον αριθμό των κοντινότερων γειτόνων του μοντέλου και τον αριθμό των ταινιών που θα προταθούν. Επιστρέφει τις N υψηλότερα βαθμολογημένες ταινίες σύμφωνα με το μοντέλο. Πιο αναλυτικός σχολιασμός για τα παραπάνω υπάρχει εντός του κώδικα.

Αποτελέσματα

Για την προβολή των αποτελεσμάτων στην περίπτωση που ο χρήστης έχει ήδη παρακολουθήσει και αξιολογήσει κάποιες ταινίες, επιλέχθηκε ένας τυχαίος χρήστης από το σύνολο train. Για αυτό το χρήστη εφαρμόζεται το μοντέλο που έχει εκπαιδευτεί και βάσει των κοντινότερων γειτόνων του προτείνονται ταινίες, τις οποίες προβλέπεται οτι θα αξιολογούσε με πολύ καλό βαθμό. Παρακάτω φαίνονται τα αποτελέσματα για δέκα ταινίες.

```
Nixon (1995)
Show, The (1995)
Cure, The (1995)
I.Q. (1994)
Ready to Wear (Pret-A-Porter) (1994)
National Lampoon's Senior Trip (1995)
Getting Even with Dad (1994)
North (1994)
Short Cuts (1993)
Great Day in Harlem, A (1994)
```

Στην δεύτερη περίπτωση δημιουργήθηκε ένας νέος χρήστης, ο οποίος δεν έχει αξιολογήσει κάποια ταινία από το dataset. Πρόκειται για γυναίκα είκοσι τριών ετών, στην οποία προτάθηκαν πάλι δέκα ταινίες, οι οποίες φαίνονται παρακάτω.

```
Brief Encounter (1946)
Pretty Woman (1990)
Pallbearer, The (1996)
Deadly Friend (1986)
Secret Garden, The (1993)
Army of Darkness (1993)
Dog of Flanders, A (1999)
Strawberry and Chocolate (Fresa y chocolate) (1993)
Raging Bull (1980)
Stripes (1981)
```

Εξαγωγή Πληροφοριών (demographics)

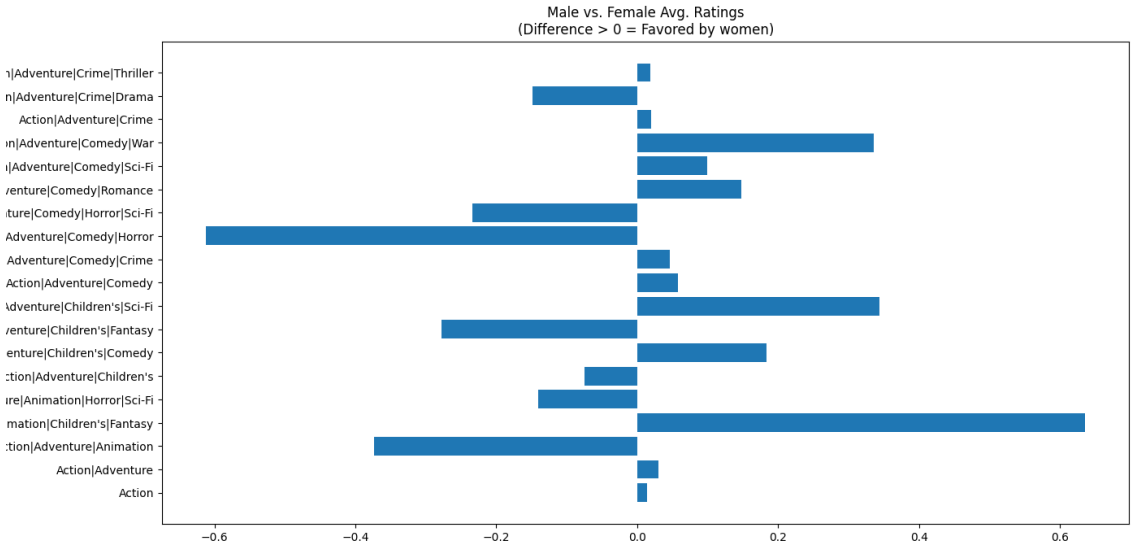
Περιγραφή

Το παρών κομμάτι της εργασίας αφοσιώνεται στη οπτικοποίηση κάποιων δεδομένων και στην πρόταση ταινίας σε συγκεκριμένες κατηγορίες users του dataset βασισμένη πάνω σε δημογραφικά στοιχεία (πχ. Πρόταση ταινιών με βάση την κατηγορία ηλικίας/επαγγέλματος κτλπ)

Προτιμήσεις με βάση το φύλο

Το πρώτο κομμάτι του κώδικα παρουσιάζει τις διαφορές στις προτιμήσεις που έχουν τα δύο φύλα ανάμεσα στους διαφορετικούς συνδυασμούς ειδών ταινιών(πχ.Δραμα,Δραμα/Κωμωδια κτλ).

Οι παρακάτω γραφικές παραστάσεις απεικονίζουν αυτά τα δεδομένα. Στον άξονα y είναι ο μοναδικός συνδυασμός ειδών των ταινιών και στο x οι διαφορές των μέσων όρων των βαθμολογιών των δύο φύλων. (μέσο της βιβλιοθήκης matplotlib). Λόγο του ότι τα dataset είναι πολύ μεγάλα και τα είδη ταινιών πολλά χρειάστηκε να εκτυπώσουμε 6 figures. Κάθε figure είναι ανά 50 είδη ταινιών (το maximum που μας άφηνε η matplotlib να έχουμε χωρίς να πέφτει το ένα πάνω στο άλλο).Συνολικά υπάρχουν 301 μοναδικοί συνδυασμοί ειδών ταινιών.



Η παραπάνω γραφική παράσταση είναι ενδεικτικό screenshot. Βάλαμε να φαίνονται μόνο οι 20 πρώτοι συνδυασμοί ειδών ταινιών για σε μεγαλύτερο αριθμό δε φαίνεται καλά.Τα 6 figure βρίσκονται στο git.

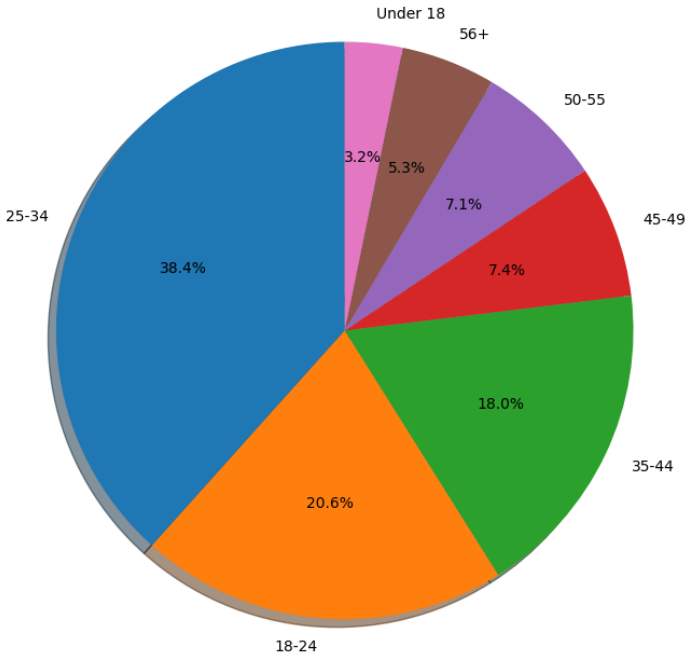
Όταν η μπάρα είναι > 0 σημαίνει ότι το συγκεκριμένο είδος ταινίας το προτιμούν οι γυναίκες και όταν είναι < 0 το προτιμούν οι άντρες. Για την υλοποίηση αυτού του κομματιού για αρχή φτιάξαμε ένα dataframe που έχει ώς index τα είδη ταινιών, ως columns τα φύλα και για τιμές τη συνολική βαθμολογία που έχει βάλει η κάθε κατηγορία στο συγκεκριμένο είδος.

Στη συνέχεια φτιάξαμε 2 dictionary (ένα για κάθε φύλο) και 2 counters, στα dictionary κρατάμε το **μοναδικό** συνδυασμό των ειδών και την αθροιστική βαθμολογία για τον κάθε συνδυασμό. Ο counter κάθε φορά που συναντάμε ένα νέο είδος απλά τον αυξάνουμε. Στη συνέχεια διαιρούμε την αθροιστική βαθμολογία του κάθε μοναδικού είδος ταινίας με τον counter και παίρνουμε το μέσο όρο. Τέλος φτιάχνουμε ένα τρίτο dictionary στο οποίο έχουμε τα είδη ταινιών και αφαιρούμε το μέσο όρο βαθμολογίας των γυναικών από αυτή των αγοριών και κρατάμε τη διαφορά. Έτσι άμα αυτή η διαφορά είναι θετική σημαίνει ότι τη συγκεκριμένη κατηγορια ταινιών την προτιμούν οι γυναίκες, αντίθετα άμα είναι αρνητική οι άντρες.

Πρόταση ταινίας με βάση την κατηγορία ηλικίας

Για αυτό το κομμάτι λαμβάνουμε υπόψιν μας το age_desc του dataset.

Αρχικά βλέπουμε ποιές ηλικίες είναι οι περισσότεροι users. (Μέσο της βιβλιοθήκης matplotlib). Παρουσιάζεται στο παρακάτω pie chart.



Φαίνεται ότι οι περισσότεροι users είναι μεταξύ των ηλικιών 25-34.

Για τη πρόταση ταινιών ανά συγκεκριμένο age_desc παίρνουμε τα παρακάτω αποτελέσματα:

	age_desc	title	counts
0	25-34	American Beauty (1999)	1334
39	18-24	American Beauty (1999)	715
65	35-44	Star Wars: Episode IV - A New Hope (1977)	626
677	45-49	American Beauty (1999)	258
743	50-55	American Beauty (1999)	248
1214	56+	American Beauty (1999)	184
2323	Under 18	Toy Story (1995)	112

Για την υλοποίηση αυτού του κομματιού δεν λαμβάνουμε υπόψη μας τις βαθμολογίες που έχουν βάλει οι χρήστες αλλά πόσοι χρήστες της τάδε age_desc έχουν βαθμολογήσει (άρα δει αυτή τη ταινία).

Πρόταση ταινίας με βάση το occ_desc

Πάλι εδώ για την υλοποίηση δεν λαμβάνουμε υπόψη μας τις βαθμολογίες αλλά το ποσό των χρηστών του συγκεκριμένου επαγγέλματος που έχουν αξιολογήσει τη ταινία.

Τα αποτελέσματα φαίνονται στα παρακάτω screenshot :

	occ_desc	title	counts
0	college/grad student	American Beauty (1999)	519
6	other or not specified	American Beauty (1999)	393
12	executive/managerial	American Beauty (1999)	352
46	academic/educator	American Beauty (1999)	289
65	technician/engineer	Terminator 2: Judgment Day (1991)	271
137	programmer	Star Wars: Episode V - The Empire Strikes Back...	227
270	sales/marketing	American Beauty (1999)	187
288	writer	American Beauty (1999)	181
498	artist	American Beauty (1999)	151
517	self-employed	American Beauty (1999)	149
685	doctor/health care	American Beauty (1999)	136
1339	clerical/admin	American Beauty (1999)	101
1486	K-12 student	Star Wars: Episode VI - Return of the Jedi (1983)	97
1788	scientist	Star Wars: Episode V - The Empire Strikes Back...	88
2581	lawyer	American Beauty (1999)	73
2683	retired	American Beauty (1999)	71
3068	customer service	American Beauty (1999)	66
4943	unemployed	American Beauty (1999)	48
5383	homemaker	Shakespeare in Love (1998)	45
6791	tradesman/craftsman	American Beauty (1999)	38
22909	farmer	Star Wars: Episode I - The Phantom Menace (1999)	10

1) Πρόταση ταινίας με βάση τον τόπο κατοικίας (zipcode)

	zipcode	title	counts
0	22903	American Beauty (1999)	15
1	94110	Pulp Fiction (1994)	14
5	48104	American Beauty (1999)	13
10	55455	American Beauty (1999)	12
12	94114	American Beauty (1999)	11
...	...	...	...
871458	75005	Shawshank Redemption, The (1994)	1
871657	75001	Hang 'em High (1967)	1
871707	74601	Where the Heart Is (2000)	1
872254	74467	Alien Nation (1988)	1
872264	74403	Wes Craven's New Nightmare (1994)	1

2) Δημοφιλέστερη ταινία ανάλογα τη χρονιά που έγινε η αξιολόγηση

	timestamp	title	counts
0	2000	American Beauty (1999)	3291
1051	2001	Almost Famous (2000)	266
2374	2002	Almost Famous (2000)	68
5058	2003	Back to the Future Part II (1989)	14

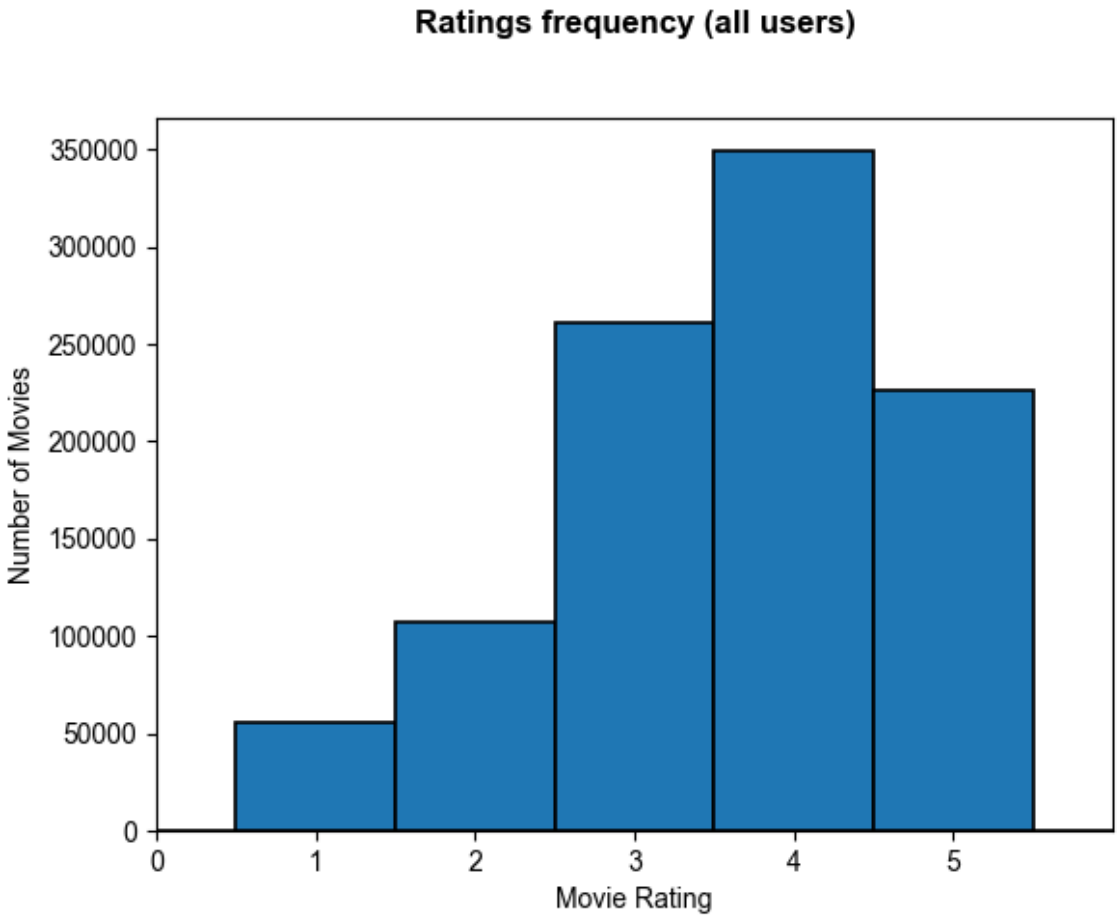
Μετατρέπουμε το timestamp σε μορφή Date : '%d-%m-%Y' και κρατάμε μόνο το %Y γιατί μας νοιάζει μόνο η χρονιά, και βρίσκουμε σε κάθε έτος ποια ταινία είχε τις περισσότερες αξιολογήσεις.

Ανάλυση Δεδομένων (showtime)

Περιγραφή

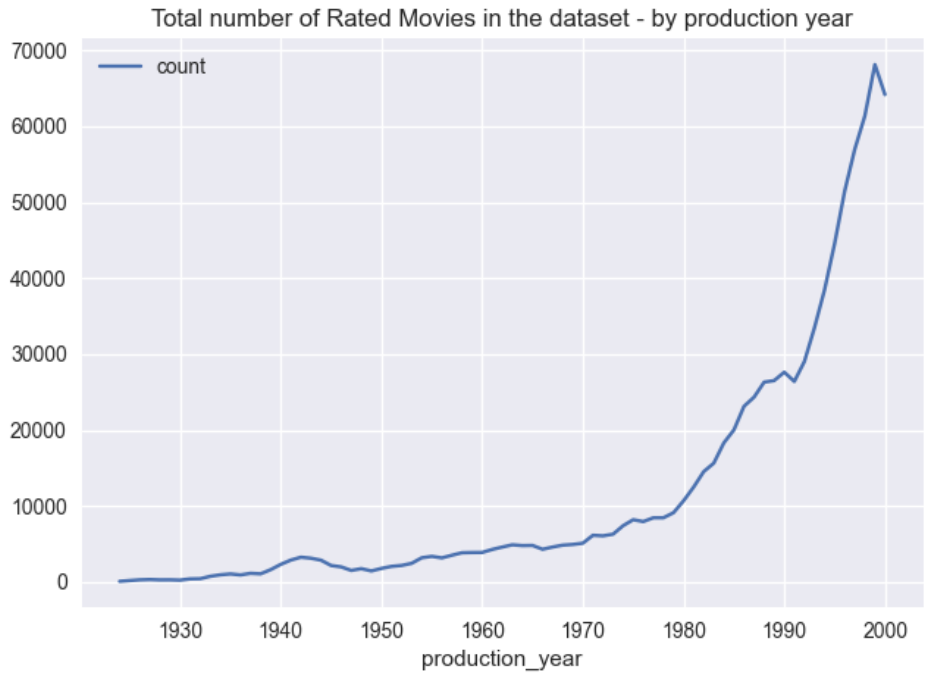
Στο πρόγραμμα αυτό αναλύουμε τα δεδομένα των datasets. Μερικές από τις τεχνικές που χρησιμοποιούνται είναι ο καθαρισμός/οργάνωση των datasets, ενωποίηση συνόλων δεδομένων, εξαγωγή και φιλτράρισμα αποτελεσμάτων, καθώς και οπτικοποίηση δεδομένων/στατιστικών.

Αρχικά δημιουργούμε τα users και ratings dataframes από τα αντίστοιχα αρχεία και αφού αφαιρέσουμε την 1η στήλη που είναι περιττή, εισάγουμε ξανά τα dataframes στα νέα αρχεία fixed_users & fixed_ratings. Μετά φτιάχνουμε το 1ο μας plot με histogram της matplotlib, το οποίο αναδεικνύει τη συχνότητα των βαθμών που βάζουνε οι χρήστες (1 εώς 5) στις ταινίες του dataset. Για να εξάγει τα στατιστικά αυτά χρησιμοποιεί όλα τα ratings που έχουνε γίνει μέσα στο dataset.



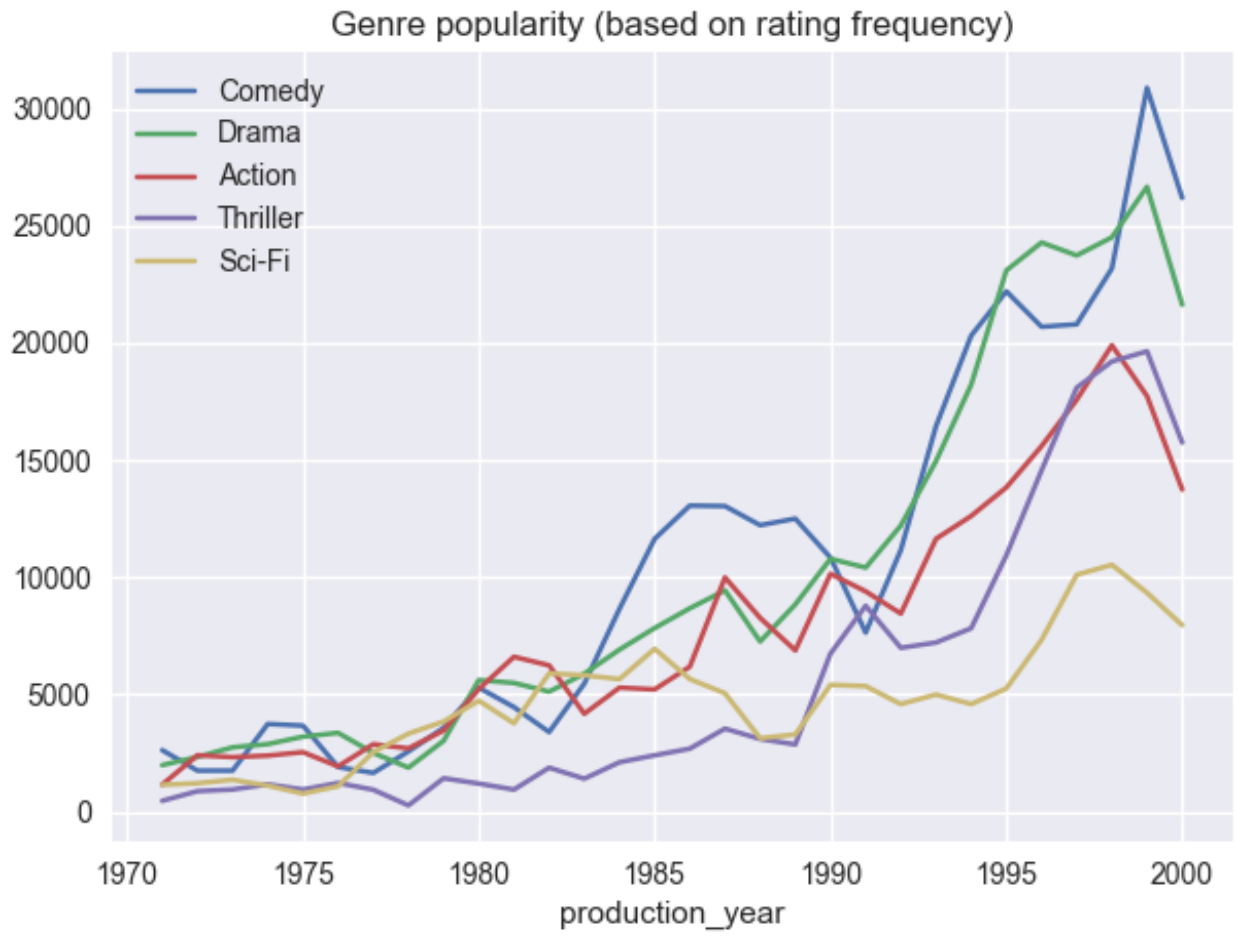
Τα συμπεράσματα που βγάζουμε από το 1ο plot είναι ότι οι βαθμοί που βάζουν οι χρήστες είναι σχετικά καλοί. Τα 4ρια κυριαρχούν με τους βαθμούς 3 και 5 να ακολουθούν στενά από πίσω. Άρα υπάρχει ένας ξεκάθαρα θετικός μέσος όρος. Αυτό σημαίνει ότι οι χρήστες του dataset γενικά απολαμβάνουν τις ταινίες που βλέπουνε.

Για το plot 2, φτιάχνουμε το movies dataframe το οποίο θα κάνουμε inner join με το ratings df, αυτό σημαίνει ότι οι ταινίες θα μεταφερθούν στο ενοποιημένο dataframe μόνο εάν έχουνε βαθμολογία. Στο νέο df όμως προκύπτει το πρόβλημα υπολογισμού των ειδών. Κάνουμε μετρήσιμα λοιπόν τα είδη καλώντας την dummies, η οποία βάζει τις τιμές 1 και 0 όταν μια ταινία βρίσκεται στο συγκεκριμένο είδος ή οχι. Προσθέττουμε την πληροφορία αυτή στο νέο tidy_movie_ratings dataframe με τη χρήση concat. Στη συνέχεια για να οργανώσουμε ακόμα καλύτερα τα δεδομένα μας εξάγουμε την πληροφορία της ημερομηνίας παραγωγής όλων των ταινιών και την τοποθετούμε σε νέα στήλη. Παρατηρούμε ότι η πληροφορία αυτή βρίσκεται μαζί με τους τίτλους των ταινιών μέσα σε παρενθέσεις. Τώρα είμαστε έτοιμοι για να μετρήσουμε τον αριθμό των παραγωγών για κάθε χρόνο (που έχουνε έστω και κάποιο rating προφανώς), για να δούμε την πορεία της παραγωγής ταινιών στο πέρασμα του χρόνου. Αυτό το κάνουμε με την prodcount στην οποία βάζουμε συνθήκη να πάρει από το tidy_movie_ratings μόνο τα στοιχεία movie_id & production_year, για τα οποία θα μετρήσει τα movie ids και θα τα κατανείμει με groupby ανα χρονιά παραγωγής (πιο συγκεκριμένα ανα δεκαετία 1930-2000).



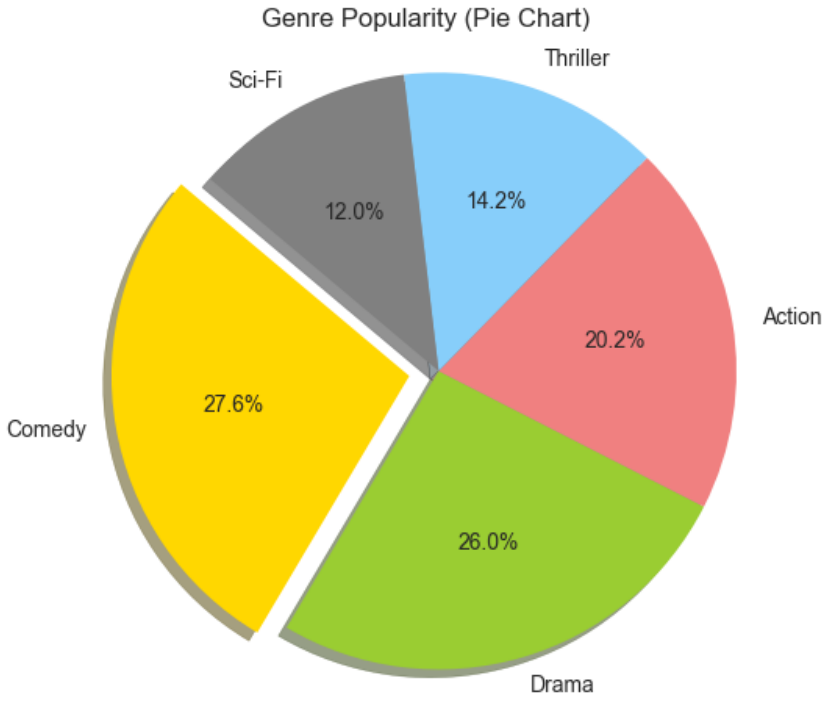
Εδώ παρατηρούμε ότι με την πάροδο του χρόνου έχουμε πολλές περισσότερες βαθμολογημένες παραγωγές ταινιών. Αυτό συμβαίνει λόγω την εξέλιξης του τομέα αλλά και λόγω της εξέλιξης της τεχνολογίας που επιτρέπει πρόσβαση στους χρήστες να αξιολογούνε ταινίες συχνότερα και ευκολότερα μέσο του διαδικτύου ή άλλων μέσων. Επίσης είναι γεγονός ότι ένας χρήστης τείνει να βλέπει ταινίες κοντά στην εποχή του και όχι πολύ παλιότερα. Επομένως λαμβάνοντας υπόψη και τις πληθυσμιακές διαφορές είναι λογικό που υπάρχουν πολύ λίγες βαθμολογημένες ταινίες από παλιότερες δεκαετίες. Στο τέλος του διαγράμματος προς το 2000 παρατηρούμε μια μικρή πτώση των rated movies. Αυτό συμβαίνει επειδή όχι επειδή παράγονται λιγότερες ταινίες αλλά γιατί οι τελευταίες κυκλοφορίες δεν είχανε ακόμα τον απαραίτητο χρόνο για να αξιολογηθούν από το κοινό.

Στη συνέχεια για το plot 3, επιχειρούμε να βρούμε τα πιο δημοφιλή είδη με κριτήριο το πόσο συχνά βαθμολογούνται. Αυτό γιατί αν είχαμε κριτήριο τις αξιολογήσεις θα αδικούσαμε τα πιο διάσημα είδη ταινιών που είναι και το ερώτημα μας. Η δουλειά μας τώρα είναι εύξολη αφού έχουμε το tidy_movie_ratings, οπότε φτιάχνουμε κατευθείαν ένα dataframe top5_genre στο οποίο : Παίρνουμε όλες τις στήλες με τα είδη, αθροίζουμε για κάθε είδος τις εμφανίσεις του, ταξινομούμε κατά φθίνουσα σειρά, και εκτυπώνουμε με head τις πρώτες 5 γραμμές (βάζοντας inxed.values θα πάρουμε σκέτο τα ονόματα των ειδών). Για να δείξουμε όμως και τη δημοτικότητα των ειδών ανα τα χρόνια πρέπει να πάρουμε και τα ratings των ειδών αυτών για να οπτικοποιήσουμε την ποσότητα που ανέδειξε αυτά τα είδη ως πιο δημοφιλή. Φτιάχνουμε λοιπόν το df genre_groups στο οποίο παίρνουμε τη στήλη των ratings με groupby τα production years και αθροίζουμε τα ratings με συνθήκη όμως τις δεκαετίες μεταξύ 1970 και 2000 θεωρώντας ότι αυτές μας ενδιαφέρουν περισσότερο. Τα αποτελέσματα του plot είναι τα εξής:



Παρατηρούμε ότι οι κατηγορίες Comedy και Drama είναι διαχρονικά οι πιο δημοφιλείς, και ότι οι ταινίες που βρίσκονται πιο κοντά στη δημιουργία του dataset έχουνε τις περισσότερες αξιολογήσεις. Με την αναμενόμενη μείωση στις καινούργιες ταινίες που ακόμα δεν είχανε την ευκαιρία να διαδοθούν και να πάρουνε πολλά ratings.

Στο plot 4 εκμεταλλευόμαστε την προηγούμενη δουλειά που βρήκαμε τα κορυφαία 5 είδη και αθροίζουμε όλα τα occurencies του κάθε είδους όλων των ετών παραγωγής προκειμένου να πάρουμε το συνολικό αριθμό τους. Τώρα μπορούμε να κατασκευάσουμε ένα pie chart που θα φαίνεται ξεκάθαρα η all time δημοτικότητα των κορυφαίων ειδών σε ποσοστά.



Τέλος ένας υπολογισμός που κάνουμε είναι η εύρεση των 5 καλύτερων ταινιών δράσης της κάθε δεκαετίας. Μπορούμε να κάνουμε το ίδιο για οποιοδήποτε είδος αλλάζοντας το “Action”. Έχουμε την επιλογή επίσης να φιλτράρουμε τα αποτελέσματα μας για όποιες δεκαετίες θέλουμε. Τα βήματα που ακολουθούμε είναι τα εξής:

- Αρχικά φτιάχνουμε ένα action table που θα περιέχει τις επιθυμητές στήλες
- Προσθέτουμε μια νέα στήλη που θα περιέχει τις δεκαετίες
- Ελέγχουμε πόσες φορές μια ταινία βαθμολογήθηκε (γιατί θέλουμε να αποφύγουμε άγνωστες ταινίες που έχουν καλή βαθμολογία)
- Χτίζουμε τις μετρικές μας με τα παραπάνω δεδομένα στο top_rate_by_decade
- Εκτυπώνουμε στην κονσόλα τα αποτελέσματα μαζί με τα ratings των ταινιών

		rating
decade	title	
1930	Adventures of Robin Hood, The	3.97
1940	Fighting Seabees, The	3.44
1950	Seven Samurai (The Magnificent Seven) (Shichini...	4.56
	African Queen, The	4.25
	Ben-Hur	4.11
	Night to Remember, A	3.88
	War of the Worlds, The	3.86
1960	Butch Cassidy and the Sundance Kid	4.22
	Good, The Bad and The Ugly, The	4.13
	Dirty Dozen, The	4.00
	Fistful of Dollars, A	3.99
	Longest Day, The	3.97
1970	Godfather, The	4.52
	Star Wars: Episode IV - A New Hope	4.45
	Godfather: Part II, The	4.36
	Alien	4.16
	French Connection, The	4.10
1980	Raiders of the Lost Ark	4.48
	Princess Bride, The	4.30
	Boat, The (Das Boot)	4.30
	Star Wars: Episode V - The Empire Strikes Back	4.29
	Killer, The (Die xue shuang xiong)	4.20
1990	Saving Private Ryan	4.34
	Matrix, The	4.32
	Braveheart	4.23
	Run Lola Run (Lola rennt)	4.22
	Princess Mononoke, The (Mononoke Hime)	4.15

Content-based filtering

Περιγραφή

Στον φάκελο Content Based Filtering έγινε μια προσπάθεια για την δημιουργία συστήματος που θα προτείνει ταινίες στους χρήστες με βάση το περιεχόμενο των ταινιών που παρακολουθούν. Δηλαδή τα είδη που παρακολουθούν (καθώς δεν έχουμε άλλα στοιχεία για το περιεχόμενο των ταινιών). Τα βήματα που σχεδιάστηκαν για την επίτευξη αυτού είναι τα εξής:

1. Καθαρισμός του movies dataset αφαιρώντας την 1η περιττή στήλη και αποθήκευση στο 1_titles.csv. Στη συνέχει κρατάμε μόνο τα ids με τα είδη στο αρχείο 2_ids.csv που θα μας χρησιμεύσει αργότερα. Τέλος φτιάχνουμε και ένα csv (3_genres.csv) μόνο με τα είδη των ταινιών το οποίο θα μας χρειαστεί στο βήμα 3.

2. Απομόνωση μοναδικών ειδών (πχ. Drama) και μοναδικών συνδυασμών ειδών (π.χ. Comedy:Romance) που παρουσιάζονται μέσα στο dataset για τις ταινίες, σε ξεχωριστά csv files (5_unique_genres & 4_sorted_genres). Αυτό θα μας χρειαστεί για να γνωρίζουμε τους μοναδικούς συνδυασμούς που θέλουμε να μετρήσουμε και τα μοναδικά είδη που υπάρχουν στο σύνολο δεδομένων μας.

3. Μέτρηση κάθε φοράς που 2 είδη συνυπάρχουν σε κάθε σειρά συνδυασμών για όλους τους συνδυασμούς στο dataset (3_genres) που περιλαμβάνει όλα τα instances των ειδών των ταινιών του dataset. Αφού αντιστοιχίσουμε τις μετρήσεις στους συνδυασμούς που βρήκαμε πριν (4_sorted_genres), αποθηκεύουμε τα αποτελέσματα σε νέο csv file (to 6_combinations), ταξινόμηση των αποτελεσμάτων σε φθίνουσα σειρά (7_sorted).

   Το αποτέλεσμα είναι να γνωρίζουμε τις συσχετίσεις μεταξύ των ειδών και τον βαθμό συγγενικότητάς τους, αφού όταν βλέπουμε ένα είδος να παρουσιάζεται πολλές φορές μαζί με ένα άλλο μπορούμε να συμπεράνουμε ότι ταιριάζουνε. Έτσι έχουμε σχέσεις μεταξύ όλων των ειδών τις οποίες μπορούμε να κατηγοριοποιήσουμε σαν αδύναμες ή ως δυνατές, για να μας βοηθήσουν να προτείνουμε ταινίες στο χρήστη. Για παράδειγμα: Έστω χρήστης που βλέπει πολλές ταινίες Action, βλέπουμε ότι πρόκειται για ένα είδος που έχει δυνατή σχέση με τα είδη Thriller, Adventure, Scifi. Μπορούμε λοιπόν να προτείνουμε συνδυασμούς Action ταινιών που <u>δεν</u> έχει δει ο χρήστης με τα παραπάνω είδη.

Η προεπεξεργασία για το σύστημα πρότασης ταινιών με βάση το περιεχόμενο έχει ολοκληρωθεί με τα παραπάνω. Τα αποτελέσματα της φαίνονται στα τελικά αρχεία csv. Ωστόσο δεν έχει υλοποιηθεί το knn για την εύρεση των όμοιων ταινιών που θα υπολογίζαμε με βάση τις σχέσεις μεταξύ των ειδών. Τέλος χρειάζεται και ένα query από εμάς για τον έλεγχο του συστήματος.