Ροή Ο :: Διοίκηση και Απόφαση, ΕΜΠ

Βικέντιος Βιτάλης el18803

Διοίκηση Ψηφιακής Επιχείρησης

8ο Εξάμηνο, 4η Σειρά – Search Recommendation

ΕΡΓΑΣΙΑ 4: ΑΝΑΖΗΤΗΣΗ ΚΑΙ RECOMMENDER SYSTEMS

Η παρούσα άσκηση αφορά στην εφαρμογή διαφόρων μεθόδων recommenders πάνω σε δεδομένα που μας δίνονται προκειμένου να γίνουν οι αντίστοιχες «προτάσεις» στους πελάτες αλλά κι η αξιολόγηση των αποτελεσμάτων των αναζητήσεων.

1. ΑΝΑΖΗΤΗΣΗ – TFIDF:

Στο σενάριο το οποίο καλούμαστε να αναλύσουμε , υποθέτουμε πως πραγματοποιείται αναζήτηση σε search engine με τους όρους “Sci-Fi”, “space”, “astro”. Στην εκφώνηση δίνονται τα «μοναδικά» αποτελέσματα.

Αρχικά, ποσοτικοποιούμε την συχνότητα των keywords μέσα στις παρακάτω περιγραφές ταινιών ή αλλιώς TF, δηλαδή Term Frequency

	A Space Odyssey	Solaris	Gravity	Moon	Interstellar
Sci-Fi	1	0	1	2	1
space	3	2	2	1	0
astro	3	0	1	2	0

Τώρα, υπολογίζουμε τη μετρική Inverse Document Frequency, ή IDF, με τη βοήθεια του πίνακα για κάθε λέξη.

$$IDF(word) = \log(\frac{total\ number\ of documents}{number\ of\ documents\ containing\ the\ word})$$

Με αυτή τη λογική έχουμε:

$$IDF(Sci-Fi) = \log\left(\frac{5}{4}\right) = 0.0969$$

$$IDF(space) = \log\left(\frac{5}{4}\right) = 0.0969$$

$$IDF(astro) = \log\left(\frac{5}{3}\right) = 0.2218$$

Σε αυτό το στάδιο, υπολογίζουμε το γινόμενο IF*IDF, το οποίο υπολογίζεται ως τον εκάστοτε αριθμό εμφανίσεων της επιθυμητής λέξης/αποτέλεσμα με τον αντίστοιχο λογάριθμο που υπολογίσαμε παραπάνω:

	A Space Odyssey	Solaris	Gravity	Moon	Interstellar
Sci-Fi	0.0969	0	0.0969	0.1938	0.0969
space	0.2907	0.1938	0.1938	0.0969	0
astro	0.6654	0	0.2218	0.4436	0

Ο τύπος σύμφωνα με τον οποίο γίνονται οι υπολογισμοί, είναι ο εξής:

$$CosineSimilarity = \frac{query * doc}{||query|| + ||doc||}$$

Έτσι, θα βρούμε ποιο κείμενο (doc, περιγραφή της ταινίας) είναι πιο κοντά στο ζητούμενο query, υπολογίζοντας για κάθε κείμενο τη γωνία μεταξύ των διανυσμάτων query και κειμένου (περιγραφή ταινίας).

Το query διαμορφώνεται ως το διάνυσμα των λογαρίθμων που υπολογίσαμε παραπάνω: query = [0.0969, 0.0969, 0.2218]

Για τις ταινίες παίρνουμε την εκάστοτε κάθετη στήλη του παραπάνω πίνακα:

- A Space Odyssey =[0.0969, 0.2907, 0.6654]
- Solaris =[0, 0.1938, 0]
- Gravity =[0.0969, 0.1938, 0.2218]
- Moon =[0.1938, 0.0969, 0.4436]
- Interstellar = [0.0969, 0, 0]

Τώρα, κρατάμε ακρίβεια 4 δεκαδικών στοιχείων. Προκειμένου να γίνει αυτό τρέχουμε το παρακάτω python script σε ένα οποιοδήποτε IDE, στο τοπικό μηχάνημά μας έγινε στο περιβάλλον του Visual Studio Code:

cosine_similarity.py 1

cosine_similarity.py > ...

```python
from scipy import spatial

query = [ 0.0969, 0.0969, 0.2218 ]

# A Space Odyssey
movie1 =[ 0.0969, 0.2907, 0.6654]
result = 1 - spatial.distance.cosine(query, movie1)
print(round(result,4))

# Solaris
movie2 =[ 0, 0.1938, 0 ]
result = 1 - spatial.distance.cosine(query, movie2)
print(round(result,4))

# Gravity
movie3 =[ 0.0969, 0.1938, 0.2218 ]
result = 1 - spatial.distance.cosine(query, movie3)
print(round(result,4))

# Moon
movie4 =[ 0.1938, 0.0969, 0.4436 ]
result = 1 - spatial.distance.cosine(query, movie4)
print(round(result,4))

# Interstellar
movie5 = [ 0.0969, 0, 0 ]
result = 1 - spatial.distance.cosine(query, movie5)
print(round(result,4))
```

Τελικά προκύπτει:

```
PS C:\Users\viken\Program Files\VS Code Projects\mode_ex4> python cosine_similarity.py
0.9694
0.3717
0.957
0.9833
0.3717
PS C:\Users\viken\Program Files\VS Code Projects\mode_ex4>
```

- Cosine similarity(query, A Space Odyssey) = 0.9694
- Cosine similarity(query, Solaris) = 0.3717
- Cosine similarity(query, Gravity) = 0.957
- Cosine similarity(query, Moon) = 0.9833
- Cosine similarity(query, Interstellar) = 0.3717

Ταξινομούμε τα αποτελέσματα σε φθίνουσα σειρά και καταλήγουμε στην κατάταξη με το ποια περιγραφή είναι πιο κοντά στο ζητούμενο query:

1. Moon
2. A Space Odyssey
3. Gravity
4. Solaris & Interstellar

2. ΑΝΑΖΗΤΗΣΗ – PRECISION/RECALL

Ψάχνουμε στοιχεία σχετικά με τη λέξη Gravity στη μηχανή αναζήτησης της Google κι επιστρέφονται κάποια αποτελέσματα, που αφορούν λεξικογραφικούς όρους, διάφορες ιστοσελίδες εταιριών, μαγνήτες κινητού αυτοκινήτων κι άλλα. Έχουμε αποτελέσματα και για την ταινία που μας ενδιαφέρει.

α. Συνολικά επιστρέφονται από τη μηχανή αναζήτησης 27 αποτελέσματα. Όπως είδαμε κι από τα αποτελέσματα της αναζήτησης, ο όρος Gravity μπορεί επιπλέον να αντιστοιχεί σε όρο της φυσικής, μπορεί να έχει μεταφορική έννοια, να είναι κάποιο όνομα που έχει αποδοθεί σε φόρμα μιας σελίδας στο διαδίκτυο κι άλλα. Από αυτά τα 8 αναφέρονται στην ταινία Gravity, είτε ευθέως, είτε έμμεσα. Αν θέλουμε να πάρουμε πληροφορίες για την ταινία Gravity κι όχι γενικά για τη λέξη Gravity, θα μπορούσαμε να προσθέσουμε επιπλέον όρους στην αναζήτησή μας στη μηχανή, όπως movie ή film.

Επομένως, TP = 8 (true positive) και FP = 19 (false positive)

β. FN = 450 (false negative). Τα αποτελέσματα έχουν ακρίβεια τεσσάρων δεκαδικών ψηφιών.

i. Precision: Ακρίβεια ή πιστότητα των αποτελεσμάτων:

$$Precision = \frac{tp}{tp+fp} = \frac{8}{8+19} = 0.2962$$

Η τιμή του Precision ταυτίζεται με την τιμή της πιθανότητας εμφάνισης σωστής απάντησης. Είναι ο αριθμός των ορθών απαντήσεων προς το πλήθος των συνολικών απαντήσεων που λαμβάνουμε. Η τιμή του είναι χαμηλή, επομένως η ακρίβεια των αποτελεσμάτων είναι μικρή.

ii. Recall: Πληρότητα

$$Recall = \frac{tp}{tp+fn} = \frac{8}{8+450} = 0.0174$$

Η τιμή του Recall είναι το αποτέλεσμα του λόγου των σωστών αποτελεσμάτων που λαμβάνουμε (true positive) από την αναζήτηση προς το πλήθος των συνολικών σωστών αποτελεσμάτων που υπάρχουν είτε εμφανίζονται είτε, στα αποτελέσματα της αναζήτησης. Σχετικά με την τιμή του Recall, εφόσον πήραμε μόνο 8 σωστά αποτελέσματα από τα 458, τα αποτελέσματα που λάβαμε υστερούν ως προς την πληρότητα (δεν είναι πλήρη).

iii. F-Measure: Σταθμισμένος Αρμονικός Μέσος

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} = 2 * \frac{0.2962 * 0.0174}{0.2962 + 0.0174} = 0.0031$$

Το F-Measure είναι ο σταθμισμένος αρμονικός μέσος της ανάκλησης και της ακρίβειας. Έχει χαμηλή τιμή, καθώς και το Precision και το Recall είχαν χαμηλές τιμές. Η αναζήτησή μας στη Google δεν ήταν επιτυχής, αφού δε λάβαμε όλα τα σωστά αποτελέσματα, κι από όσα αποτελέσματα λάβαμε μερικά από αυτά ήταν άσχετα με το στόχο της αναζήτησής μας τίτλος, δηλαδή της ταινίας Gravity.

3. RECOMMENDER SYSTEMS

Στο τελευταίο μέρος της άσκησης, 6 άνθρωποι αξιολόγησαν τις ταινίες του ερωτήματος 1 με βαθμολογία από 1 έως 10. Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα. Έχουμε ακόμη X = 3 + 1 = 4

	A Space Odyssey	Solaris	Gravity	Moon	Interstellar
Χρήστης 1	4	4		6	7
Χρήστης 2	3	2		5	4
Χρήστης 3	5	1	4	3	4
Χρήστης 4	7	6	7	9	6
Χρήστης 5		2	4	3	4
Χρήστης 6	9	7	8	7	10

1) Similarity με Ευκλείδεια Απόσταση

Αρχικά περνάμε τον παραπάνω πίνακα σε ένα .csv αρχείο

40		A Space Odyssey	Solaris	Gravity	Moon	Interstellar
41	Χρήστης 1	4.000	4.000	0.000	6.000	7.000
42	Χρήστης 2	3.000	2.000	0.000	5.000	4.000
43	Χρήστης 3	5.000	1.000	4.000	3.000	4.000
44	Χρήστης 4	7.000	6.000	7.000	9.000	6.000
45	Χρήστης 5	0.000	2.000	4.000	3.000	4.000
46	Χρήστης 6	9.000	7.000	8.000	7.000	10.000

Εν συνεχεία, υπολογίζουμε την εκάστοτε ευκλείδεια όπου υπάρχει βαθμολογία και στις i,j γραμμές και στήλες αντίστοιχα. Ο υπολογισμός γίνεται με βάση το τύπο:

$$d(p,q) = \sqrt{(p1-q1)^2 + (p2-q2)^2 + \cdots + (pi-qi)^2 + \cdots + (pn-qn)^2}$$

58	Ευκλείδεια Απόσταση	Χρήστης 1	Χρήστης 2	Χρήστης 3	Χρήστης 4	Χρήστης 5	Χρήστης 6
59	Χρήστης 1	0.000	-	-	-	-	-
60	Χρήστης 2	3.873	0.000	-	-	-	-
61	Χρήστης 3	5.292	3.000	0.000	-	-	-
62	Χρήστης 4	4.796	7.211	8.832	0.000	-	-
63	Χρήστης 5	4.690	2.000	1.000	8.062	0.000	-
64	Χρήστης 6	6.633	10.050	10.954	10.488	15.427	0.000

Με βάση το τύπο $Similarity(i,j) = \frac{1}{1+sum(i,j)^{\frac{1}{2}}}$ προκύπτει:

49	Ευκλείδεια Similarity	Χρήστης 1	Χρήστης 2	Χρήστης 3	Χρήστης 4	Χρήστης 5	Χρήστης 6
50	Χρήστης 1	1.000	-	-	-	-	-
51	Χρήστης 2	0.205	1.000	-	-	-	-
52	Χρήστης 3	0.159	0.250	1.000	-	-	-
53	Χρήστης 4	0.173	0.122	0.102	1.000	-	-
54	Χρήστης 5	0.176	0.333	0.500	0.110	1.000	-
55	Χρήστης 6	0.131	0.090	0.084	0.087	0.061	1.000

Δεδομένου ότι οι γραμμές με τις στήλες ταυτίζονται κι ο πίνακας είναι τριγωνικός μπορούμε να τον αντιστρέψουμε για οπτικούς λόγους ώστε η τελική μορφή να είναι όμοια με αυτή του πίνακα παρακάτω:

Ευκλείδεια Similarity	Χρήστης 1	Χρήστης 2	Χρήστης 3	Χρήστης 4	Χρήστης 5	Χρήστης 6
Χρήστης 1	1	0.205	0.159	0.173	0.176	0.131
Χρήστης 2		1	0.250	0.122	0.333	0.090
Χρήστης 3			1	0.102	0.500	0.084
Χρήστης 4				1	0.110	0.087
Χρήστης 5					1	0.061
Χρήστης 6						1

2) Similarity με Pearson Correlation

Χρησιμοποιούμε το παρακάτω python script (στιγμιότυπο μετά την τελευταία επανάληψη):

cosine_similarity.py | pearson_similarity.py | pearson_similarity2.py

pearson_similarity2.py > ...

```python
from scipy import stats
from scipy.stats import pearsonr

#x = [4, 4, 6, 7]                  # User1
#x = [3, 2, 5, 4]                  # User2
#x = [5, 1, 4, 3, 4]               # User3
#x = [6, 7, 9, 6]                  # User4
x = [2, 4, 3, 4]                   # User5
y = [7, 8, 7, 10]                  # User6

corr=stats.pearsonr(x, y)[0]        # Calculate Pearson's correlation
sim=(corr+1)/2                      # Calculate Similarity
print(round(sim,4))                 # Print similarity
```

Παρατίθεται η λογική εκτέλεσης:

```
# Line 1:
# For the cell 1,1 -> 1
# For the cell 1,2 -> Execute with user1,2
# For the cell 1,3 -> Execute with user1,3 w/o the values that do not exist in both
# For the cell 1,4 -> Execute with user1,4 w/o the values that do not exist in both
# For the cell 1,5 -> Execute with user1,5 w/o the values that do not exist in both
# For the cell 1,6 -> Execute with user1,6 w/o the values that do not exist in both

# Line 2:
# For the cell 2,1 -> -
# For the cell 2,2 -> 1
# For the cell 2,3 -> Execute with user2,3 w/o the values that do not exist in both
# For the cell 2,4 -> Execute with user2,4 w/o the values that do not exist in both
# For the cell 2,5 -> Execute with user2,5 w/o the values that do not exist in both
# For the cell 2,6 -> Execute with user2,6 w/o the values that do not exist in both

# Line 3:
# For the cell 3,1 -> -
# For the cell 3,2 -> -
# For the cell 3,3 -> 1
# For the cell 3,4 -> Execute with user3,4 w/o the values that do not exist in both
# For the cell 3,5 -> Execute with user3,5 w/o the values that do not exist in both
# For the cell 3,6 -> Execute with user3,6 w/o the values that do not exist in both

# Line 4:
# For the cell 4,1 -> -
# For the cell 4,2 -> -
# For the cell 4,3 -> -
# For the cell 4,4 -> 1
# For the cell 4,5 -> Execute with user4,5 w/o the values that do not exist in both
# For the cell 4,6 -> Execute with user4,6 w/o the values that do not exist in both

# Line 5:
# For the cell 5,1 -> -
# For the cell 5,2 -> -
# For the cell 5,3 -> -
# For the cell 5,4 -> -
# For the cell 5,5 -> 1
```

```
# For the cell 5,6 -> Execute with user5,6 w/o the values that do not exist in both

# Line 6:
# For the cell 6,1 -> -
# For the cell 6,2 -> -
# For the cell 6,3 -> -
# For the cell 6,4 -> -
# For the cell 6,5 -> -
# For the cell 6,6 -> 1
```

Σύμφωνα με την οποία συμπληρώνεται πλήρως ο πίνακας Similarity with Pearson Correlation:

Pearson Correlation	Χρήστης 1	Χρήστης 2	Χρήστης 3	Χρήστης 4	Χρήστης 5	Χρήστης 6
Χρήστης 1	1	0.8873	0.6139	0.5786	0.991	0.7037
Χρήστης 2		1	0.689	0.8651	0.8273	0.543
Χρήστης 3			1	0.569	0.9924	0.854
Χρήστης 4				1	0.5	0.25
Χρήστης 5					1	0.8693
Χρήστης 6						1

a. Αφού έχουμε k = 2 και σταθμισμένο μέσο θα υπολογίσουμε πως οι χρήστες 3 και 4 (όντας οι πιο κοντινοί γείτονες στο χρήστη 2) βαθμολόγησαν την ταινία Gravity. Προκειμένου να το κάνουμε αυτό θα χρησιμοποιήσουμε και τις δύο μετρικές, δηλαδή την Ευκλείδεια Απόσταση και των Pearson Συσχέτιση.

- Ευκλείδεια Απόσταση:

$$PredictedRatingU2 = \frac{sim(U2,U3) * ratingU3 + sim(U2,U4) * ratingU4}{sim(U2,U3) + sim(U2,U4)}$$

$$PredictedRatingU2 = \frac{0.250 * 4 + 0.122 * 7}{0.250 + 0.122} = \frac{1.854}{0.372} = 4.9838$$

- Pearson Συσχέτιση:

$$PredictedRatingU2 = \frac{sim(U2,U3) * ratingU3 + sim(U2,U4) * ratingU4}{sim(U2,U3) + sim(U2,U4)}$$

$$PredictedRatingU2 = \frac{0.689 * 4 + 0.8651 * 7}{0.689 + 0.8651} = \frac{8.8117}{1.5541} = 5.6699$$

Αρχικά, παρατηρούμε ότι και στις δύο περιπτώσεις οι τιμές του Predicted Rating για τον Χρήστη 2 είναι σχετικά κοντά. Επιπλέον, μπορούμε να πούμε σχετικά με το αποτέλεσμα που λάβαμε στην περίπτωση της Ευκλείδιας Απόστασης, ότι η τιμή που τελικά υπολογίσαμε είναι πιο κοντά στη βαθμολογία (4) του χρήστη 3, με το μικρότερο similarity. Το ίδιο ισχύει αντίστοιχα και στην περίπτωση της μεθόδου Pearson Correlation. Η τιμή που τελικά υπολογίσαμε είναι σχετικά πιο κοντά στην τιμή 4 απ'ότι στην τιμή βαθμολογίας 7 που έδωσε ο Χρήστης 5.

b. Γενικά, για να προτείνουμε φίλους, με βάση τις προτιμήσεις τους, πρέπει να δούμε ποια ζεύγη χρηστών έχουν μεγαλύτερο similarity. Θα μπορούσαμε να σχηματίσουμε όλα τα πιθανά ζευγάρια και να τα κατατάξουμε σε φθίνουσα σειρά ως προς το similarity. Ωστόσο, δε θα καταγράψουμε όλες τις περιπτώσεις, αλλά τις πιο πιθανές.
    - Με την Ευκλείδια απόσταση, τις 4 μεγαλύτερες similarities σημειώνουν τα ζεύγη χρηστών 3 - 5 με 0.500, 2 - 5 με 0.333, 2 - 3 με 0.250 και των χρηστών 1 – 2 με 0.205.
    - Με βάση τα αποτελέσματα της τιμής similarity, όπου λάβαμε υπόψη τη μέθοδο Pearson Correlation, τις 4 μεγαλύτερες τιμές εμφανίζουν τα ζεύγη χρηστών 3 - 5 με 0.9924, 1 - 5 με τιμή similarity 0.991, 1 - 2 με 0.8873 και των 5 – 6 με 0.8693.
    - Παρατηρούμε ότι ανάμεσα σε αυτά τα 4 ζευγάρια των 2 μεθόδων, τα ζευγάρια 3 - 5 και 1 - 2 δίνουν σταθερά καλές (υψηλές) τιμές similarity, οπότε και θα τα προτείναμε.


VIKENTIOS VITALIS