دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش ۶ درس هوش مصنوعی

ارائه یک سیستم توصیه گر برای یک مسئله ی کاربردی

به قلم:
امیر بابامحمودی

استاد
دکتر مهدی قطعی

اردیبهشت ۱۴۰۰

مقدمه:

امروزه از سیستم های توصیه گر در امور بسیاری مانند پیشبینی امتیاز یک محصول یا عملکرد آن استفاده میشود. استفاده ی مهم دیگر این سیستم ها در پیشنهاد دادن موارد مورد علاقه ی کاربران نسبت به چیز هایی که قبلا مورد پسند آن ها قرار گرفته میباشد. برای مثال آمازون از این سیستم ها برای پیشنهاد کالا به کاربران و یا نتفلیکس برای پیشنهاد فیلم و سریال و اسپاتیفای برای موسیقی استفاده میکند. سیستم های توصیه گری که امروزه در کمپانی های تکنولوژی پیشرفته مانند آنهایی که نام برده شد استفاده میشود بسیار پیچیده و دقیق میباشد. در این گزارش قصد داریم دو سیستم توصیه گر متکی به جدول زده و خروجی آن ها را بررسی کنیم.

سیستم های توصیه گر ساده:

این نوع توصیه گر ها به صورت عمومی و پیشنهادی باری عموم کاربران میکنند. برای مثال جدول ۲۵۰ فیلم برتر IMDB که بر اساس معیاری فیلم ها به ترتیب بیشترین امتیاز را نمایش میدهد. به طور کلی این سیستم ها بر اساس نمره ای که بقیه کاربران به یک محصول میدهند به دست میاید و احتمال اینکه یک کاربر از محصول با بالاترین امتیاز ها خوشش بیاید را بالا میبرد اما در کل این توصیه گر ها اصلا دقیق نبوده و جای خطای بسیاری دارد.

در این قسمت یک دیتاست مربوط به کتاب را تحلیل کرده و یک سیستم توصیه گر ساده را بروی آن اجرا میکنیم.

لینک دریافت دیتاست:

https://www.kaggle.com/bahramjannesarr/goodreads-book-datasets-10m

```
good_reads.head(5)
#good_reads2 = good_reads[['Name' , 'RatingDistTotal' , 'Publisher' , 'Authors' , 'Rating' ]]
```

	Id	Name	RatingDist1	pagesNumber	RatingDist4	RatingDistTotal	PublishMonth	PublishDay	Publisher	CountsOfReview	PublishYear	Language	A
0	1	Harry Potter and the Half-Blood Prince (Harry ...	1:9896	652	4:556485	total:2298124	16	9	Scholastic Inc.	28062	2006	eng	R
1	2	Harry Potter and the Order of the Phoenix (Har...	1:12455	870	4:604283	total:2358637	1	9	Scholastic Inc.	29770	2004	eng	R
2	3	Harry Potter and the Sorcerer's Stone (Harry P...	1:108202	309	4:1513191	total:6587388	1	11	Scholastic Inc	75911	2003	eng	R
3	4	Harry Potter and the Chamber of Secrets (Harry	1:11896	352	4:706082	total:2560657	1	11	Scholastic	244	2003	eng	R

همانگونه که میبینید دیتاست شامل عنوان کتاب و ۱۱ ستون دیگر شامل میانگین نمره ی کتاب و تعداد رای میباشد.حال ستون های مورد نیاز را از جدول استخراج کرده تا دیتا های ناکارامد را حذف کنیم.

```
good_reads2 = good_reads[['Name' , 'RatingDistTotal' , 'Publisher' , 'Authors' , 'Rating' ]]
good_reads2.head(5)
```

	Name	RatingDistTotal	Publisher	Authors	Rating
0	Harry Potter and the Half-Blood Prince (Harry ...	total:2298124	Scholastic Inc.	J.K. Rowling	4.57
1	Harry Potter and the Order of the Phoenix (Har...	total:2358637	Scholastic Inc.	J.K. Rowling	4.50
2	Harry Potter and the Sorcerer's Stone (Harry P...	total:6587388	Scholastic Inc	J.K. Rowling	4.47
3	Harry Potter and the Chamber of Secrets (Harry...	total:2560657	Scholastic	J.K. Rowling	4.42
4	Harry Potter and the Prisoner of Azkaban (Harr...	total:2610317	Scholastic Inc.	J.K. Rowling	4.57

اسم کتاب, نویسنده, مجموع تعداد رای دهندگان و میانگین امتیاز کتاب هارا به عنوان جدولی جدید در نظر میگیریم.حال برای اینکه بتوانیم از مجموع رای دهندگان به عنوان متغیری عددی استفاده کنیم با استفاده از regex آن را از حالت رشته در میاوریم.

```
total_vots["total_vote"] = good_reads2.RatingDistTotal.str.extract('(\d+)')
good_reads2['total_votes'] = total_vots['total_vote']
good_reads2.head(5)

```

	Name	RatingDistTotal	Publisher	Authors	Rating	total_votes
0	Harry Potter and the Half-Blood Prince (Harry ...	total:2298124	Scholastic Inc.	J.K. Rowling	4.57	2298124
1	Harry Potter and the Order of the Phoenix (Har...	total:2358637	Scholastic Inc.	J.K. Rowling	4.50	2358637
2	Harry Potter and the Sorcerer's Stone (Harry P...	total:6587388	Scholastic Inc	J.K. Rowling	4.47	6587388
3	Harry Potter and the Chamber of Secrets (Harry...	total:2560657	Scholastic	J.K. Rowling	4.42	2560657
4	Harry Potter and the Prisoner of Azkaban (Harr...	total:2610317	Scholastic Inc.	J.K. Rowling	4.57	2610317

همانگونه که میبینید ستون total_votes رو به دیتا ست اضافه میکنیم.
در دیتاست مذکور کتاب های مشابه به زبان های متفاوت نیز رویت شد که از میان آن ها با دستور زیر کتاب های انگلیسی تنها انتخاب شدند.

```
good_reads = good_reads[good_reads['Language'] == 'eng']
```

همانگونه که میدانید ممکن است کتابی ریتینگ 4.5 داشته باشد و کتاب دیگری ریتینگش 4 باشد. ظاهر ماجرا این است که کتاب با ریتینگ4.5 باید اولویت داشته باشد برای توصیه شدن اما این را در نظر بگیرید که شاید آن کتاب با 1000 رای به این نمره رسیده و کتاب با ریتینگ 4 با 2 ملیون به این امتیاز رسیده است. که در این صورت کتاب با امتیاز 4 اولویت پیدا میکند.حال با توجه به این توضیحات فرمولی ارائه داده و به کمک آن امتیاز بندی جدید ارائه میدهیم.

$$WeightedRating(\mathbf{WR}) = \left(\frac{\mathbf{v}}{\mathbf{v}+\mathbf{m}} \cdot \mathbf{R}\right) + \left(\frac{\mathbf{m}}{\mathbf{v}+\mathbf{m}} \cdot \mathbf{C}\right)$$

v: مجموع رای دهندگان به یک کتاب

m:معیاری برای حداقل تعداد رای داده شده به یک کتاب که از عدد خاصی کمتر نباشد.

R: ریتینگ یک کتاب

C:میانگین امتیاز تمام کتاب ها

مقادیر m , C را نداشته و باید آن هارا حساب کنیم. m به عنوان یک هایپرپارامتر میتواند مقدارش را سلیقه ای اختیار کند که در اینجا برای مثال به گونه ای انتخابش میکنیم که تعداد رای کتاب مورد نظر باید در ۹۰ درصد پر رای ترین کتاب ها باشد.

```
good_reads2['total_votes'] = good_reads2['total_votes'].astype(float)
C = good_reads2['Rating'].mean()
m = good_reads2['total_votes'].quantile(0.90)
q_books = good_reads2.copy().loc[good_reads2['total_votes'] >= m]
m
```

```
42348.60000000002
```

کل تعداد کتاب های این دیتا ست 15922 میباشد که با در نظر گرفتن ۹۰ درصد پر رای دهنده ترین آن ها یعنی تمام کتاب هایی را بررسی میکنیم که تعداد رای دهندگانش بیش از 42348 باشد.

با توجه به کد میتوانید ببینید که C چگونه بدست آمده است.در نهایت تمامی کتاب هایی که تعداد مورد نظر رای دارند را در جدولی دیگر بنام q_books میریزیم.

```
def weighted_rating(x, m=m, C=C):
    v = x['total_votes']
    R = x['Rating']
    # Calculation based on the IMDB formula
    return (v/(v+m) * R) + (m/(m+v) * C)
q_books['score'] = q_books.apply(weighted_rating, axis=1)
q_books = q_books.sort_values('score', ascending=False)
q_books.head(5)
q_books.drop('RatingDistTotal' , axis = 1 , inplace = True)
q_books.drop('Publisher' , axis = 1 , inplace = True)
q_books.head(20)
```

	Name	Authors	Rating	total_votes	score
41331	Harry Potter and the Deathly Hallows (Harry Po...	J.K. Rowling	4.62	2661573.0	4.609173
4	Harry Potter and the Prisoner of Azkaban (Harr...	J.K. Rowling	4.57	2610317.0	4.559762
54174	Harry Potter and the Half-Blood Prince (Harry ...	J.K. Rowling	4.57	2306655.0	4.558439
40728	Harry Potter and the Half-blood Prince (Harry ...	J.K. Rowling	4.57	2300327.0	4.558407
0	Harry Potter and the Half-Blood Prince (Harry ...	J.K. Rowling	4.57	2298124.0	4.558397
5	Harry Potter and the Goblet of Fire (Harry Pot...	J.K. Rowling	4.56	2431085.0	4.549191
9245	The Return of the King (The Lord of the Rings,...	J.R.R. Tolkien	4.53	642807.0	4.492835
44758	Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling	4.50	2367353.0	4.489960

و در نهایت در تابع weighted_rating فرمول محاسبه امتیاز جدید پیاده سازی شده و ستون امتیاز را با اعمال این تابع به روی تمامی کتاب ها اعمال کرده و سپس به صورت صعودی بر اساس این امتیاز ها کتاب ها رو مرتب میکنیم. خروجی ای که میبینید ۸ کتاب اول این جدول میباشد.

سیستم های توصیه گر content-based:

حال قصد داریم سیستم توصیه گری پیاده سازی کنیم که بتواند محصولاتی شبیه به محصول دیگر از نظر ویژگی را ارائه بدهد. برای مثال فرض کنید شما از فیلمی خوشتان میاید و به دنبال فیلمی در همان سبک و موضوع و ساخت هستید. در این نوع سیستم های توصیه گر ویژگی های خاصی که مدنظر از یک محصول میباشد رو استخراج کرده و با تحلیل آن ها و محاسبه ی درصد شباهت این ویژگی ها به محصولات دیگر امتیاز داده میشود و نزدیک ترین به محصول مورد نظر یافت میشود. برای پیاده سازی این بخش از دیتاست دیگری از کتاب ها استفاده شده است که دارای خلاصه ای از کتاب نیز باشد که در ادامه دلیل آن توضیح داده میشود.

لینک دیتاست:

https://www.kaggle.com/jdobrow/57000-books-with-metadata-and-blurbs

```
import pandas as pd
metadata = pd.read_csv('books_with_blurbs.csv' , low_memory=False)
metadata.head(3)
```

	ISBN	Title	Author	Year	Publisher	Blurb
0	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	Here, for the first time in paperback, is an o...
1	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	The fascinating, true story of the world's dea...
2	0399135782	The Kitchen God's Wife	Amy Tan	1991	Putnam Pub Group	Winnie and Helen have kept each others worst s...

همانگونه که میبینید ستونی تحت عنوان Blurb باری ارائه ی خلاصه ای از هر کتاب علاوه بر ستون هایی نظیر نام نویسنده ی کتاب و کمپانی منتشر کننده و کتاب و غیره وجود دارد.

برای پیاده سازی چیزی که توضیح داده شد مشخص میباشد که نیاز است پردازش زبان طبیعی صورت بگیرد تا بتوان با درصد تشابهی که میتوان از کلمات موجود در خلاصه هر کتاب و همینطور نویسنده ی کتاب و انتشاراتی آن استخراج کرد کتاب های نزدیک به کتابی خاص را پیدا کرد. این نکته لازم به ذکر میباشد که احتمال خوبی وجود دارد وقتی کسی کتابی از یک نویسنده را خیلی دوست داشته باشد کتاب های دیگر آن نویسنده را هم دوست بدارد و همچنین میدانیم هر انتشاراتی تخصص در یک ژانر خاص کتاب دارد که این عامل هم موثر هست و به همین دلایل ما از دیتای موجود در این دو ستون هم استفاده میکنیم.

در ابتدا سعی در تمیز کردن دیتا با کوچک کردن حروف سه ستون مذکور و همچنین ترکیب آن ها در یک ستون میکنیم.

```
metadata2 = metadata[0:20000]
metadata2['Blurb'] = metadata['Blurb'].str.lower()
metadata2['Author'] = metadata['Author'].str.lower()
metadata2['Publisher'] = metadata['Publisher'].str.lower()
```

```
def join_apb(x):
    return ''.join(x['Author']) + ' ' + ''.join(x['Publisher']) + ' ' + x['Blurb'] + ' '
metadata2['apb'] = metadata.apply(join_apb, axis=1)
```

در هنگام فراخوانی تابع join_apb سه ستون مورد نظر را تحت عنوان ستونی به نام apb به دیتا ست خود اضافه میکنیم. ستون apb در شکل زیر قابل مشاهده میباشد.

	ISBN	Title	Author	Year	Publisher	Blurb	apb
0	0060973129	Decision in Normandy	carlo d'este	1991	harperperennial	here, for the first time in paperback, is an o...	carlo d'este harperperennial here, for the fir...
1	0374157065	Flu: The Story of the Great Influenza Pandemic...	gina bari kolata	1999	farrar straus giroux	the fascinating, true story of the world's dea...	gina bari kolata farrar straus giroux the fasc...
2	0399135782	The Kitchen God's Wife	amy tan	1991	putnam pub group	winnie and helen have kept each others worst s...	amy tan putnam pub group winnie and helen have...

```
from sklearn.feature_extraction.text import CountVectorizer
count = CountVectorizer(stop_words='english')
count_matrix = count.fit_transform(metadata2['apb'])
```

حال با استفاده از فانکشن CountVectorizer ازکتابخانه ساکیت لرن میتوان تکست موجود در ستون apb رو تبدیل به ماتریسی از شمارش توکن ها یا همان کلمات کرد. آبجکت count را به صورتی میسازیم که کلمات پر تکرار مانند the , is , are و امثال آن ها را در تکست در نظر نگرفته و به طور کلی از کلمات کم اهمیت صرف نظر کند. حال میتوان ستون apb را به وسیله ی این متد پردازش کرده و به ماتریس مورد نیاز خود یعنی count_matrix که شمارش هر کلمه را دارد رسید. حال برای رسید به درصد تشابه کلمات موجود در ستون apb هر کتاب از فرمول تشابه کسینوسی استفاده میکنیم تا به یک معیار تشابه عددی بین هر دو کتاب برسیم. ساز کار این فرمول به صورت زیر میباشد.

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}^\intercal}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \cdot \mathbf{y}^\intercal{}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{y}_i)^2}}$$

در واقع با محاسبه ی

cos(count_matrix , count_matrix) به یک ماتریس n*n که n در واقع تعداد کتاب ها میباشد میرسیم که در هر درایه ij آن که i!=j باشددرصد تشابه کتاب با ردیف i و کتاب با ردیف j را به عنوان عددی بین ۰ و ۱ بیان میکند. حال به کد پیاده سازی شده میپردازیم.

```
from sklearn.metrics.pairwise import cosine_similarity
cosine_sim = cosine_similarity(count_matrix, count_matrix)
```

در واقع محاسبه ی تشابه کسینوسی را میتوان با تابع built_in موجود در خود سایکیت لرن انجام داد.
توجه داشته باشید که انجام این عملیات به دلیل زیاد بودن تعداد کلمات(در این پیاده سازی ۸۵۰۶۷ کلمه) بسیار وقت گیر و سنیگن میباشد برای همین از تمامی ۵۷۰۰۰ کتاب موجود در دیتاست استفاده نشده و تنها ۲۰۰۰۰ تای اول آن را انتخاب کردی.
حال که ماتریس n*n تحت عنوان cosine_sim را داریم کافی است تابعی را پیاده سازی کرده که در آن به عنوان آرگومان ورودی نام فیلم را گرفته و با استفاده از ماتریس cosine_sim کتاب ها با بیشترین میزان تشابه به کتاب مورد نظر را خروجی بدهد.

```
def get_recommendations(Title):
  idx = indices[Title]
  sim_scores = list(enumerate(cosine_sim[idx]))
  sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
  sim_scores = sim_scores[1:11]
  book_indices = [i[0] for i in sim_scores]
  return metadata2['Title'].iloc[book_indices]
```

حال خروجی کد برای کتاب A History of pi را میبینیم.

```
get_recommendations('A History of Pi')
```

```
1023                              The Years of Rice and Salt
9035     The Rise and Fall of the Third Reich : A Histo...
16732                               Loyalties a Son's Memoir
4478                   The History of the Siege of Lisbon
9608                                        House of Spirits
13470    Uncle John's Bathroom Reader Plunges into Hist...
1506     Lies My Teacher Told Me : Everything Your Amer...
15603    Lies My Teacher Told Me: Everything Your Histo...
10656                         Uppity Women of Medieval Times
3        What If?: The World's Foremost Military Histor...
Name: Title, dtype: object
```

منابع:

1) https://www.kaggle.com/gspmoreira/recommender-systems-in-python-101

2) https://www.datacamp.com/community/tutorials/recommender-systems-python

لینک گیتهاب جهت دسترسی به کد :
https://github.com/amirbabamahmoudi/AI-projects/tree/main/recommender_system