

دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

**تکلیف پنجم (اختیاری و امتیازی) درس بازیابی اطلاعات**

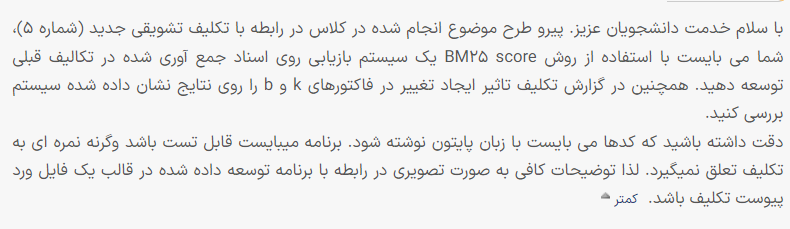
استاد: دکترمحمد مهدی رضاپور

مهروالسادات نوحی

993613061

بهار 1403

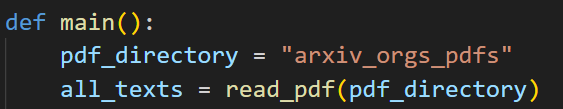
صورت تمرین:

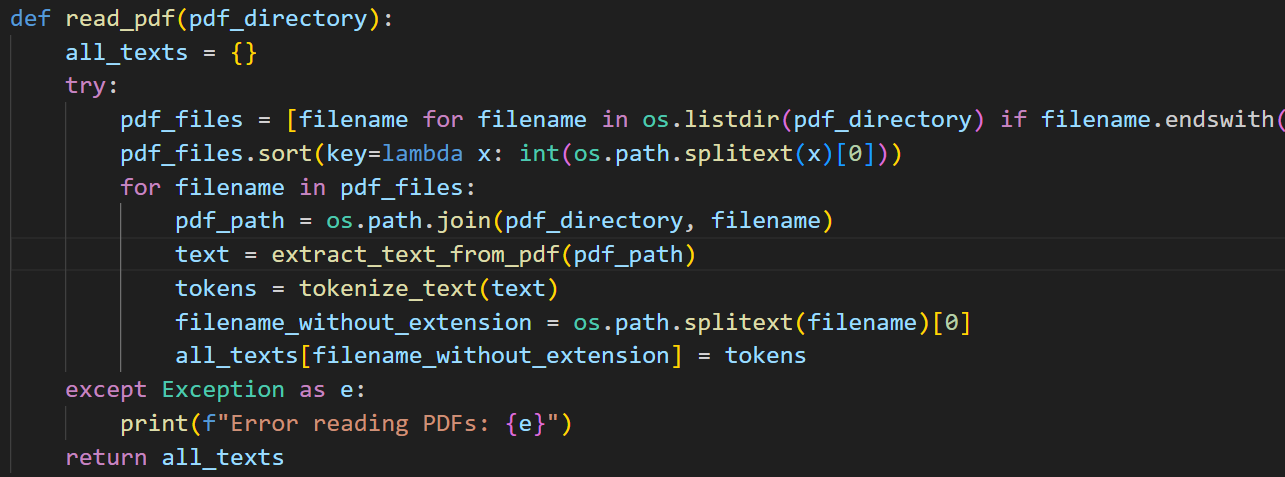


مانند گذشته تمرین را در چند گام تقسیم بندی کرده و جداگانه توضیح خواهم داد. ترجیحا از همان 20 مقاله دانلود شده در تمرین‌‌های قبل برای این تمرین و تمرین بعدی استفاده خواهد شدکه نتایج به صورت واضح‌تر نشان داده شود.

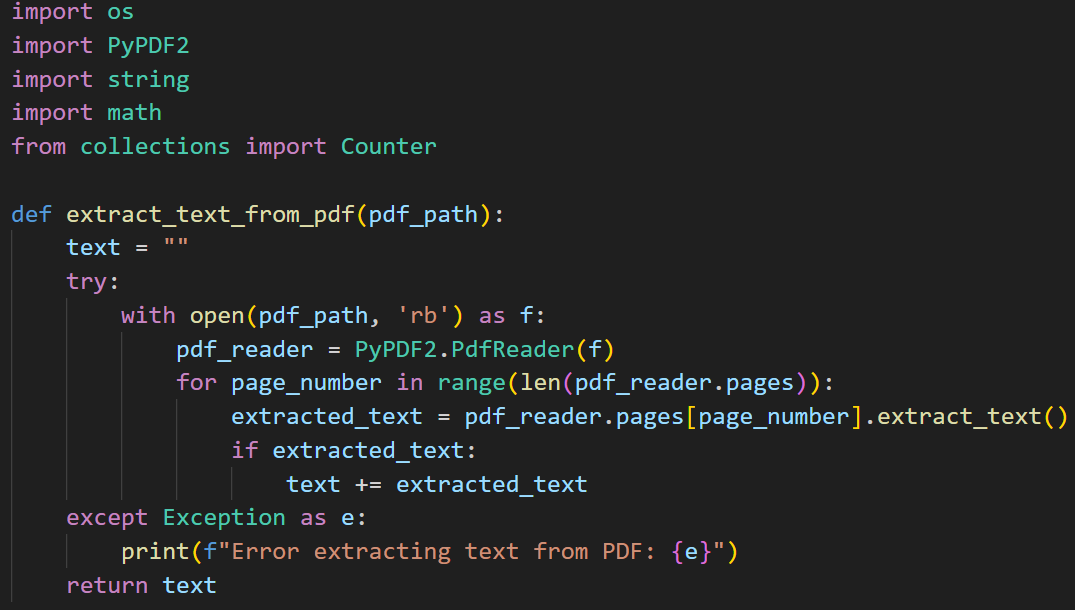
گام اول:

در گام اول از پوشه‌ای که 20 مقاله مورد نظر در آن است باید از هر 20 مقاله شروع به خواندن و استخراج متن کنیم.





این تابع read\_pdf، فایل‌های PDF را از یک دایرکتوری می‌خواند و متن هر فایل را استخراج و توکنیزه می‌کند. سپس نتیجه را به صورت یک دیکشنری ذخیره می‌کند که در آن نام فایل به عنوان کلید و لیست توکن‌ها به عنوان مقدار است.



این تابع extract\_text\_from\_pdf، متن یک فایل PDF را استخراج می‌کند. ابتدا فایل PDF باز می‌شود و سپس متن هر صفحه استخراج و به یک رشته اضافه می‌شود.

با تعریف یک دیکشنری که کلید آن شماره داکیومنت و مقدار آن متن داخل فایل است .

تذکر: من اسم فایل‌ها را برای راحتی کار به مثلا 1.pdf و.. تغییر دادم وبرای استخراج فقط شماره فایل داریم.

در واقع به شکل زیرشده است:

{

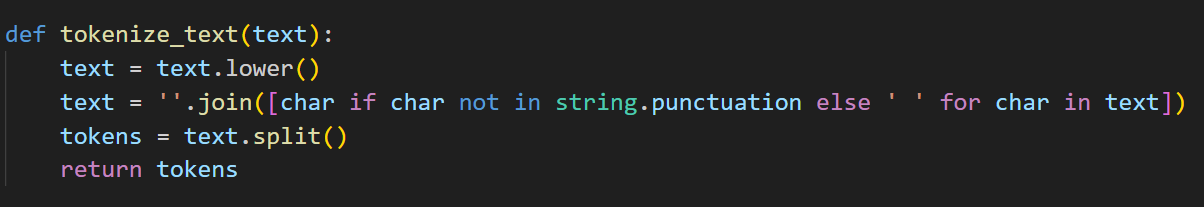
“Filename1”: text1,

“Filename2”:text2,

…{

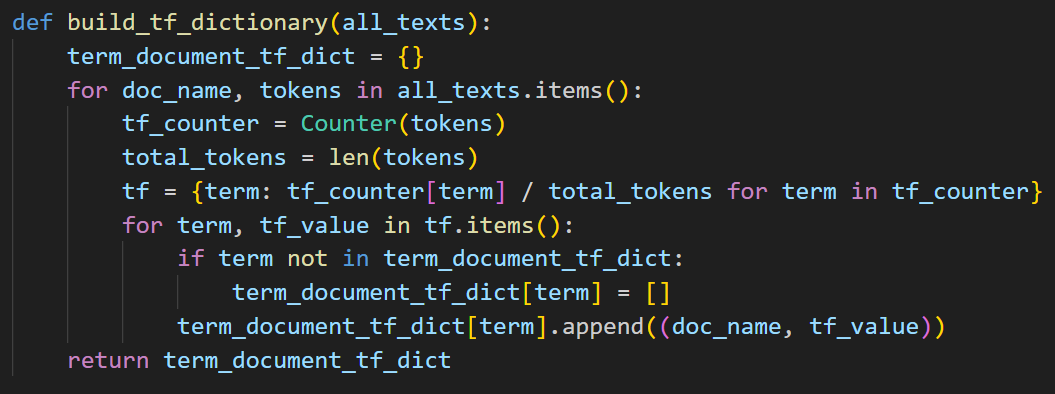
گام دوم:

بعد از اینکه متن‌ها استخراج شده است حال باید توکنایز شود و به اصطلاح هر توکن یک کلمه در نظر بگیریم برای درخواست‌های کاربر.



این تابع tokenize\_text، متن ورودی را به لیستی از توکن‌ها تبدیل می‌کند. ابتدا متن به حروف کوچک تبدیل می‌شود، سپس علائم نگارشی با فاصله جایگزین می‌شوند و در نهایت متن به لیستی از کلمات تقسیم می‌شود.

گام سوم:



این تابع build\_tf\_dictionary، دیکشنری TF را می‌سازد. برای هر داکیومنت، فراوانی نسبی هر توکن (TF) محاسبه می‌شود و در دیکشنری ذخیره می‌شود.به ازای هر داکیومنت، برای همه توکن‌ها در تمام متن استخراج شده یک دیکشنری ساخته و به عنوان مثال کلید آن ترم یا توکن و مقدار آن مقدارtf آن به ازای داکیومنت، و شماره داکیومنت است.

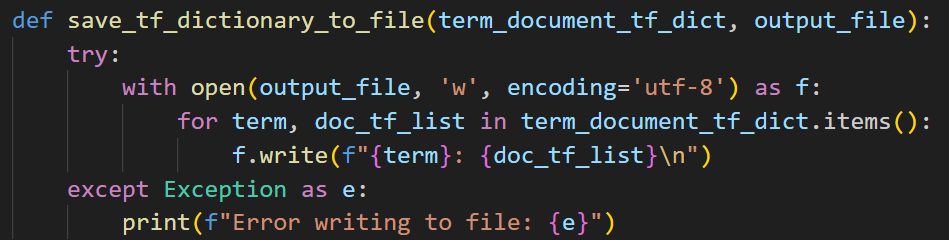
Dict {

Term1🡪Document\_ID🡪tf #for that document find tf

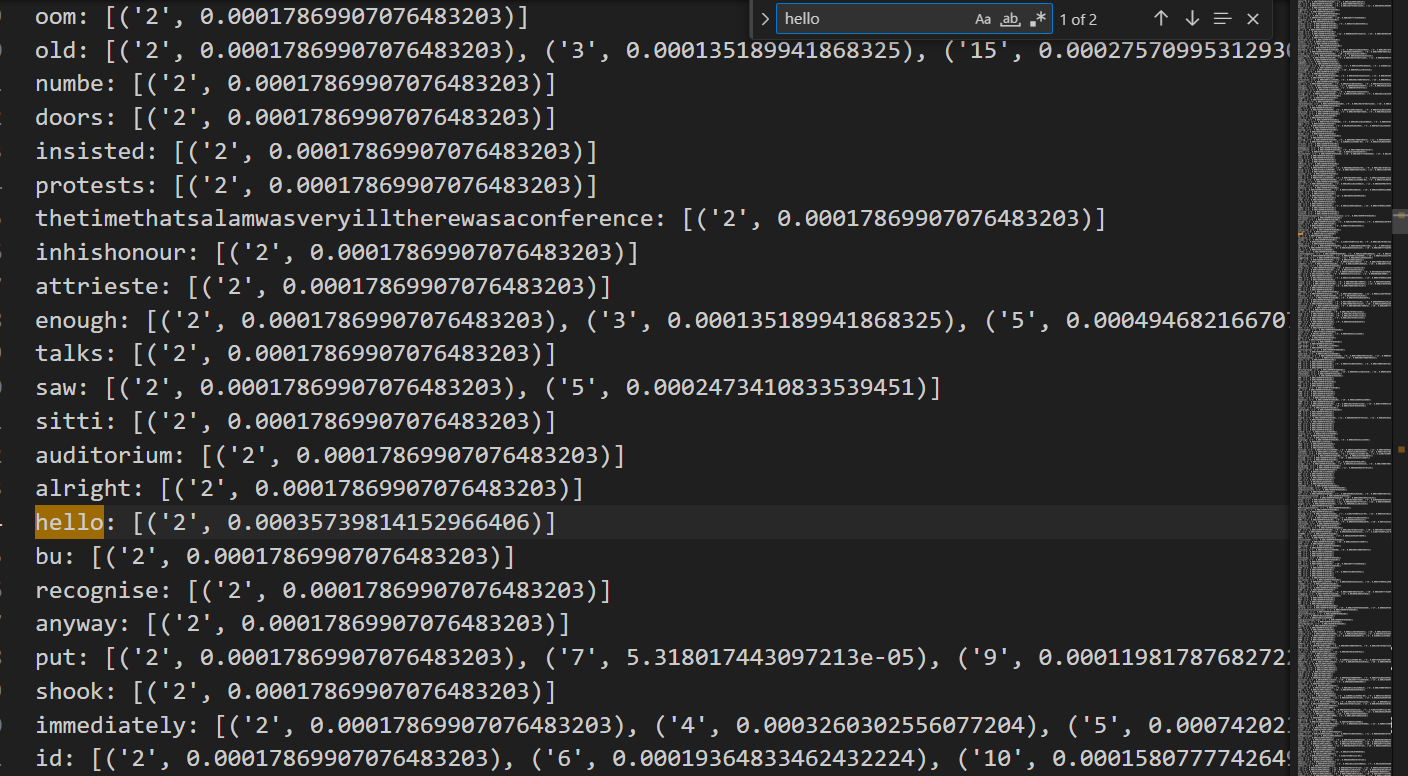
Term2🡪Document\_ID🡪tf #for that document find tf

…}

میتوانیم خروجی این گام را در فایل ذخیره کنیم تابع به صورت زیر است:

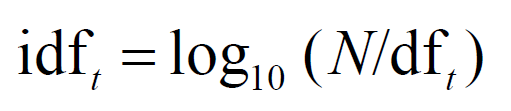


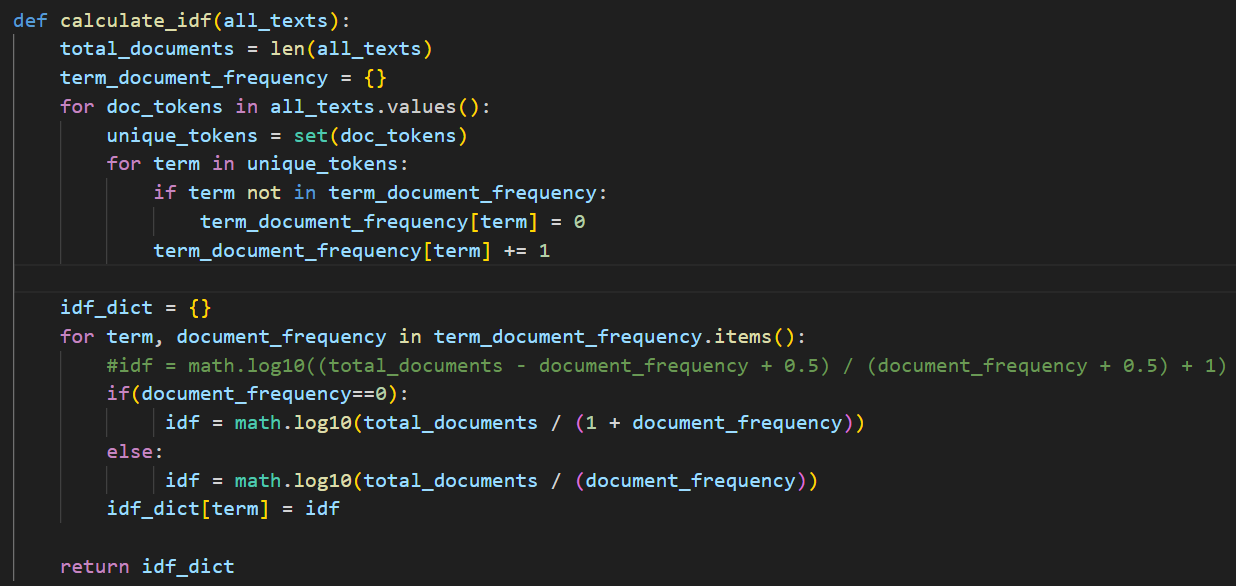
خروجی:



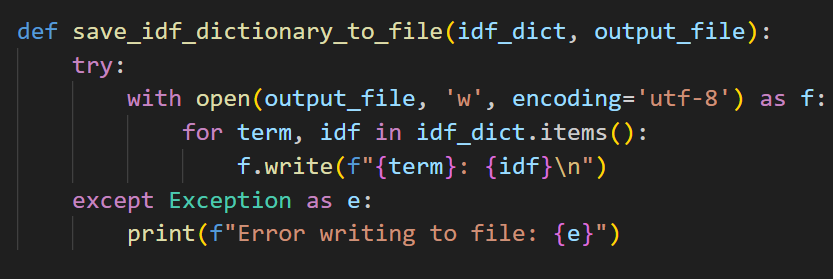
گام چهارم:

در برنامه یک دیکشنری دیگری تعریف کردم که در آن به ازای هر ترم ازهر داکیومنت، idf آن محاسبه شده است که در آینده از آن استفاده کنیم. در واقع این تابع calculate\_idf، مقادیر IDF را برای هر توکن محاسبه می‌کند. فراوانی مستندات برای هر توکن محاسبه می‌شود و سپس IDF محاسبه و در دیکشنری ذخیره می‌شود.

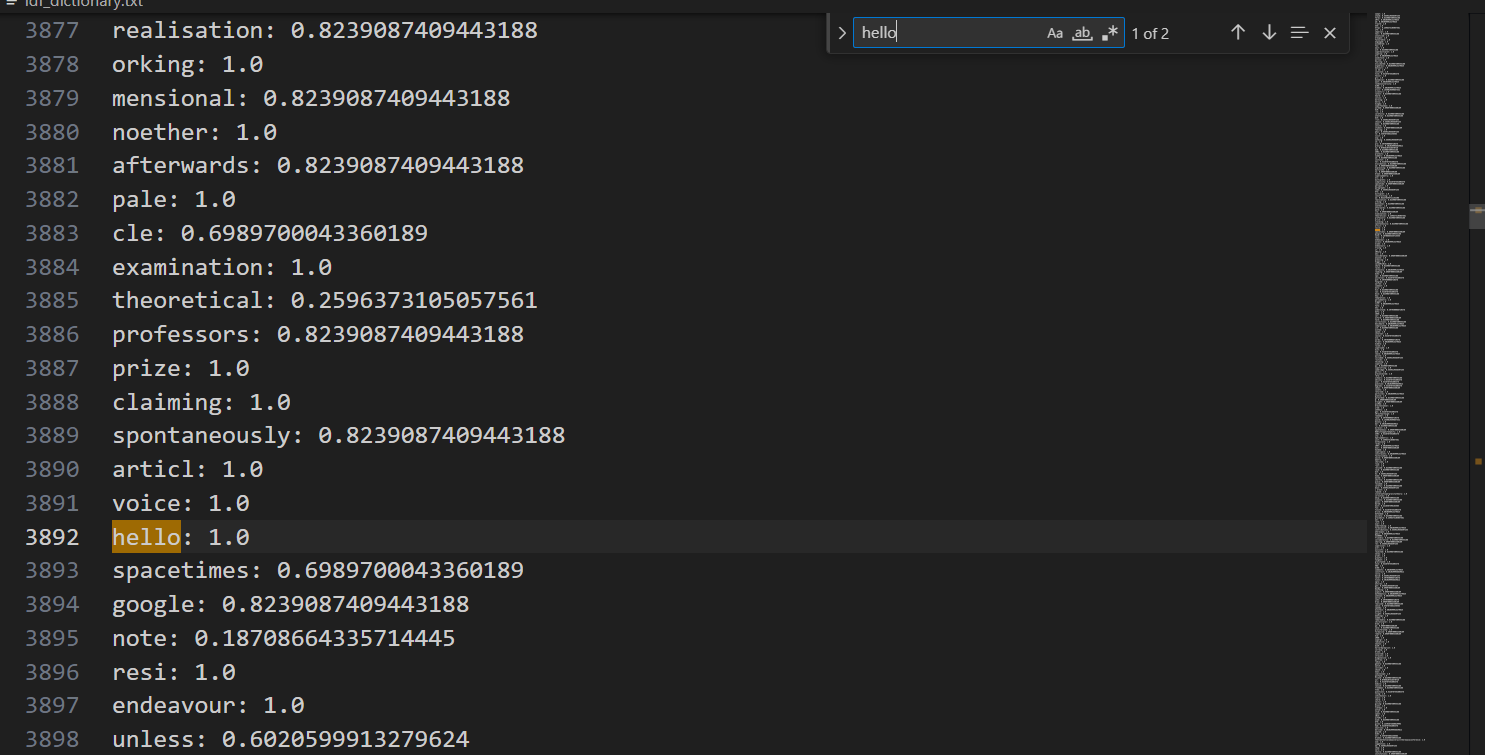




میتوانیم خروجی این گام را در فایل ذخیره کنیم تابع به صورت زیر است:

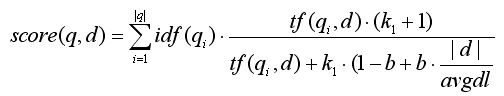


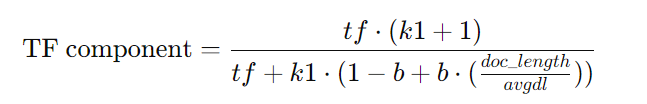
خروجی:

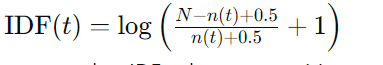


گام پنجم:

در این گام بپردازیم به BM25 چطور کار می‌کند:



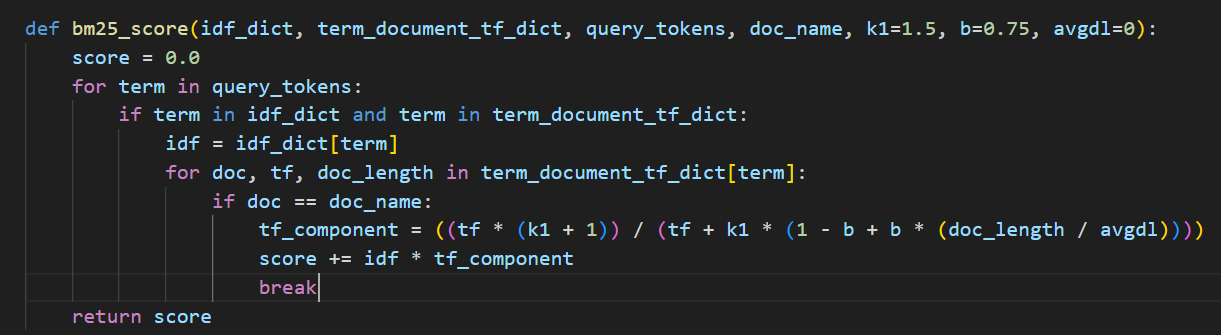


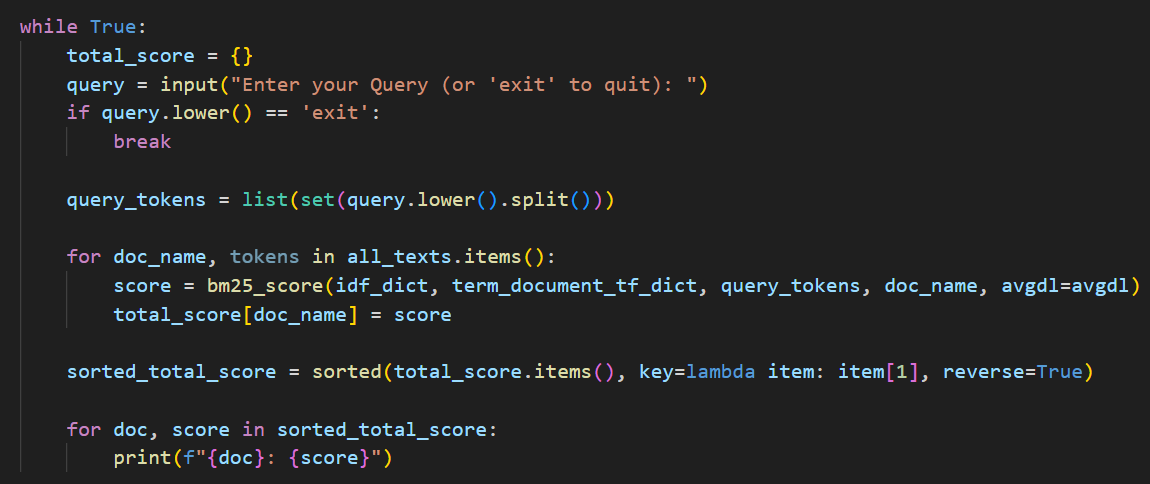


بریم برنامه رو ببینیم چطوری کار می‌کند:

**پارامترها**

1. **idf\_dict** : یک دیکشنری که کلمات (ترم‌ها) را به عنوان کلیدها دارد و مقادیرشان فراوانی معکوس سند است. فراوانی معکوس سند مهمیت یک کلمه را در کل مجموعه اسناد اندازه‌گیری می‌کند.
2. **term\_document\_tf\_dict**: یک دیکشنری که کلمات (ترم‌ها) را به عنوان کلیدها دارد و مقادیرش لیستی از تاپل‌ها هستند. هر تاپل شامل:
   * نام سند
   * فراوانی کلمه (TF) در آن سند
   * طول سند
3. **query\_tokens**: یک لیست از کلمات (ترم‌ها) از پرسمان.
4. **doc\_name**: نام سندی که امتیاز BM25 برای آن محاسبه می‌شود.
5. **k**: یک پارامتر تنظیم کننده که مقیاس فراوانی کلمات را تنظیم می‌کند. پیش‌فرض آن 1.5 است.
6. **b**: یک پارامتر تنظیم کننده که تأثیر نرمال‌سازی طول سند را کنترل می‌کند. پیش‌فرض آن 0.75 است.
7. **avgdl**: میانگین طول سند در کل مجموعه اسناد. این مقدار برای نرمال‌سازی طول استفاده می‌شود.

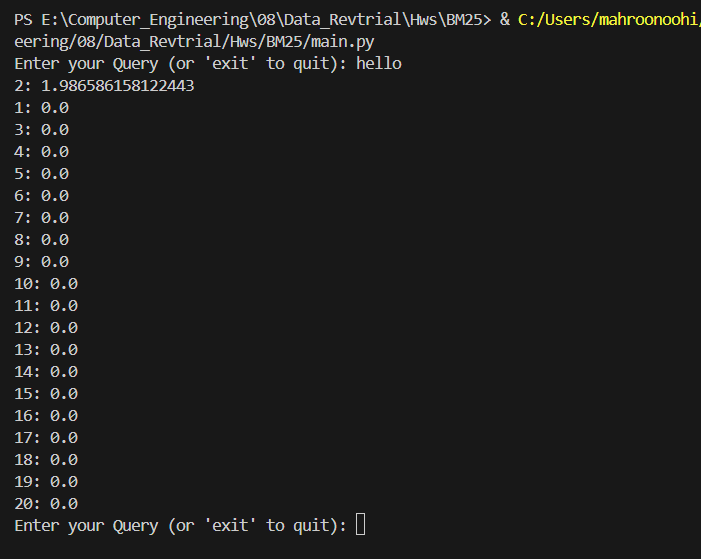




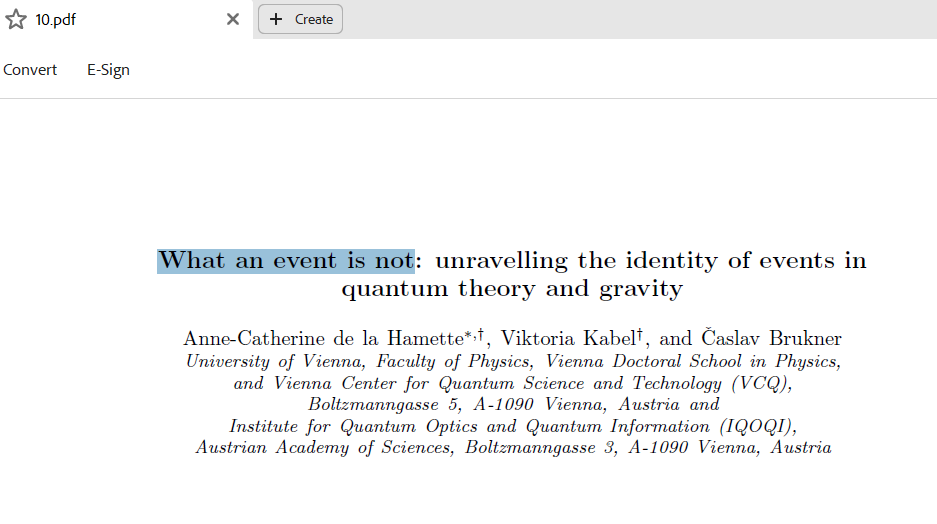
گام ششم:

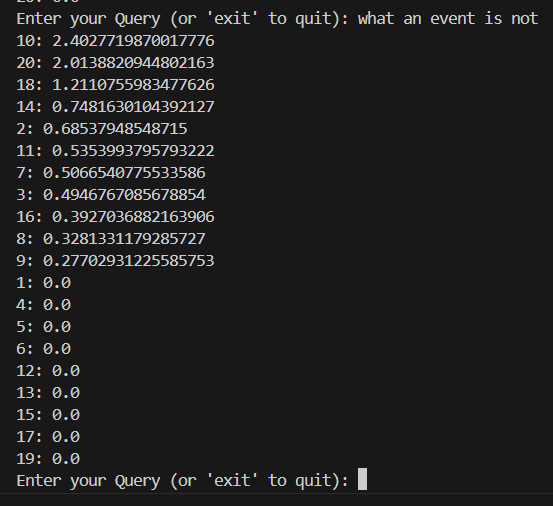
بریم چند تا تست کنیم والبته مقدار k,b هم تغییر بدیم تا تغییرات را بهتر ببینیم.

با k=1.5 و b=0.75

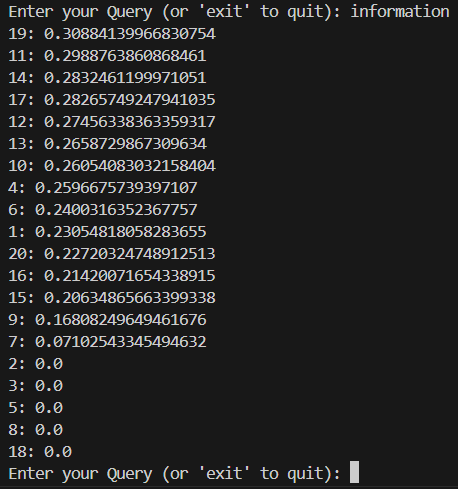


در بخش‌های قبل نیز داکیومنت 2 برگردانده شد.



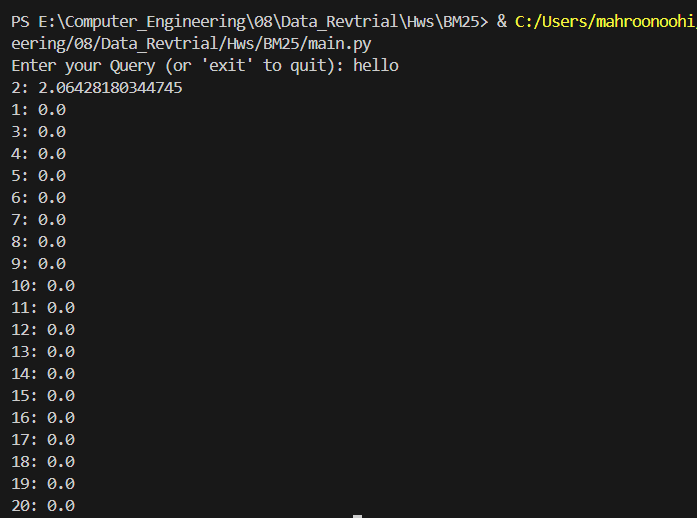


داکیومنت 10 به عنوان اولین داکیونت آورده شده است.



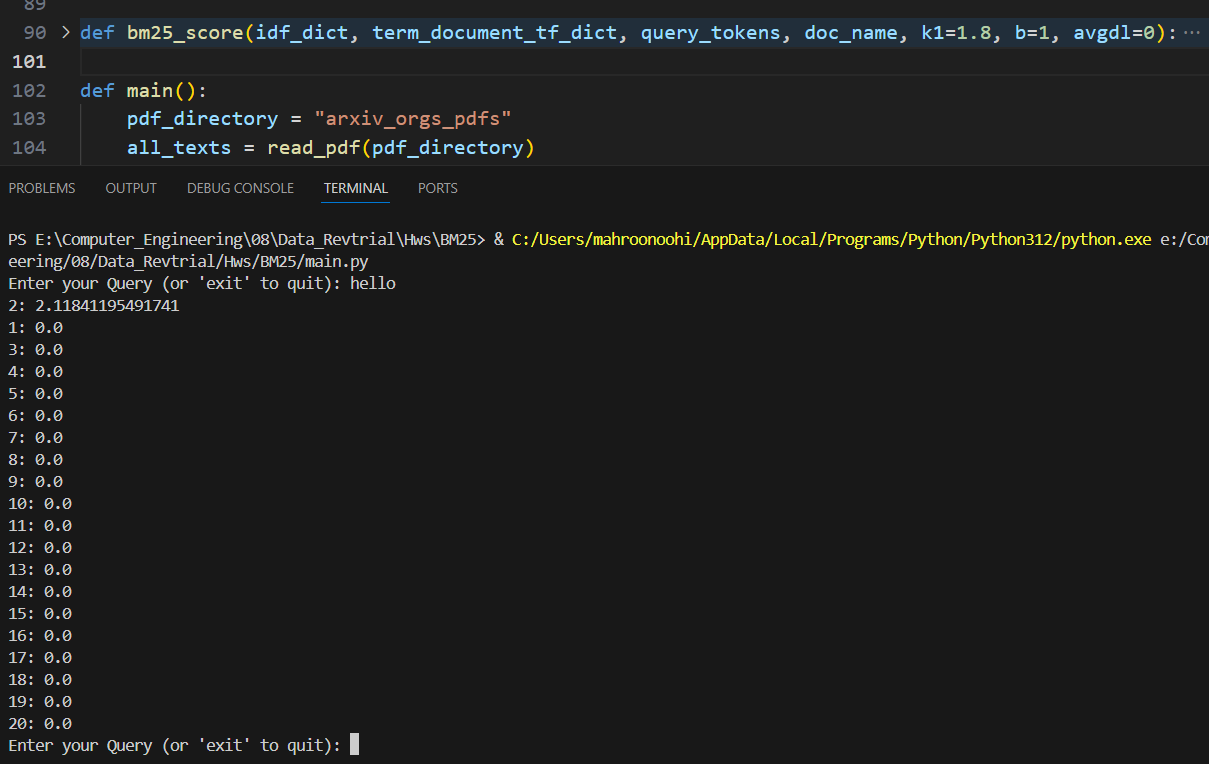
بریم با همین مثال فاکتورهای k,b تغییر بدهیم.

این سری k=1.8 و b=0.75 است:

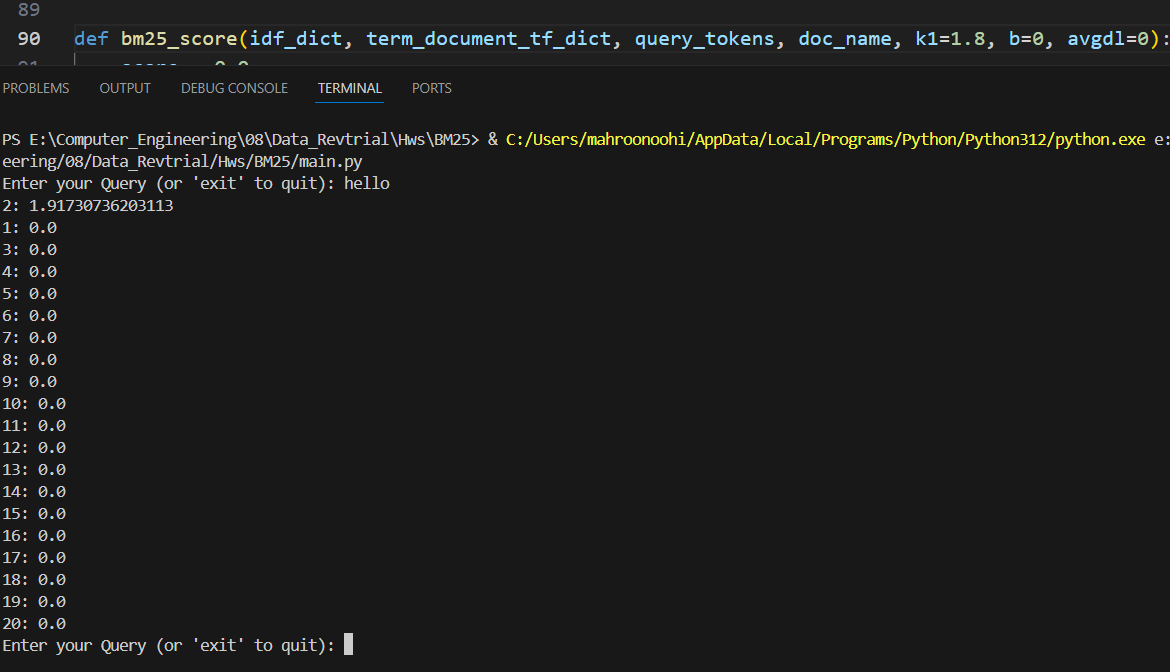


مقدار عددی عوض شده ولی همچنان داکیومنت 2 که درست است برگرداننده شده است.

این سری k=1.8 و b=1 است: مقدار عددی عوض شده ولی همچنان داکیومنت 2 که درست است برگرداننده شده است.



این سری k=1.8 و b=0 است:



با عوض شدن مقدار پارامترها عدد خروجی هم تغییر کرده و داکیومنت 2 همچنان برگردانده شده است. من برای جمله هم تست کردم و اوکی بوده است.

