

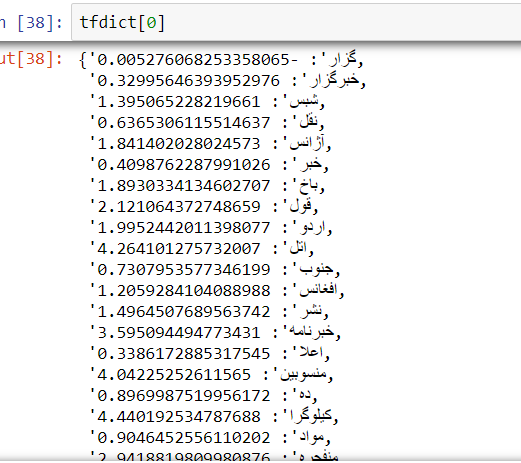
**نام دانشجو: سپهر عسگریان ابیانه**

**شماره دانشجو: 9531901**

**مرداد 1399**

**گزارش پروژه**

در ادامه ی پروژه ی قبلی در ابتدا posting list مرحلهی قبل را به صورت فایل pkl ذخیره کرده برای اینکه بتوانیم بدون نیاز به اجرای دوباره از آن استفاده کنیم برای بدست آوردن tfidf تابع word\_count استفاده شد که برای هر جمله تعداد آن کلمه ی خاص را مشخص میکند که این جمله در واقع از preprocess فاز 1 گرفته شده است . سپس در تابع tffunc که در فایل jupyter میباشد با توجه به فرمول ارائه شده در فایل پروژه tfidf برای هر سند محاسبه شده و سچس به صورت فایل pkl بدست آمده شده است برای مثال tfidf جمله ی صفر به صورت زیر میباشد

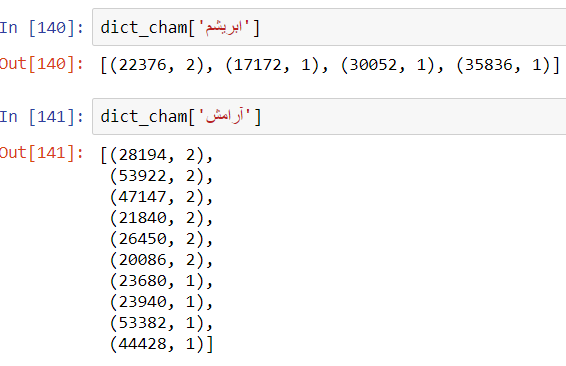


در همین فایل jupyter در تابع زیر با استفاده از posting list ای از قبل داشتیم برای هر کلمه به صورت پیش فرض 10 سند معتبر نگه داری شده است و لازم به ذکر است طبق توضیحات استاد قبل از گرفان پرسمان این مرحله انجام شده است سپس فایل pkl آن ساخته شده است که بتوان از آن استفاده کرد



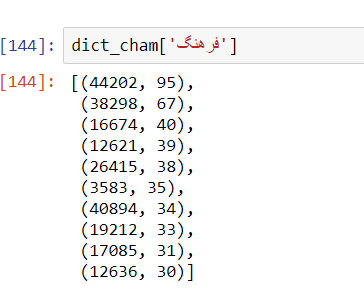
که در این جا sort orders همان 10 برترین سند برای هر کلمه را انتخاب کرده و سپس توسط دیکشنری dict­cham ، save میشود.

برای مثال برای سه مثال آورده شده



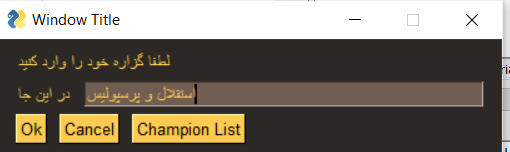
که در اینجا علاوه بر سند frequency آن در آن سند نیز قابل مشاهده است

برای فرهنگیان چون پیش پردازش ان و ی را حذف کرده و کلمه به صورت فرهنگ ذخیره شده است به صورت زیر میباشد



حال به فایل دوم که به صورت .py است میرویم

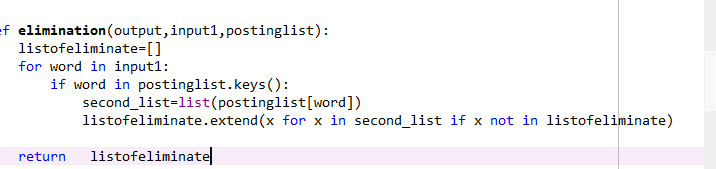
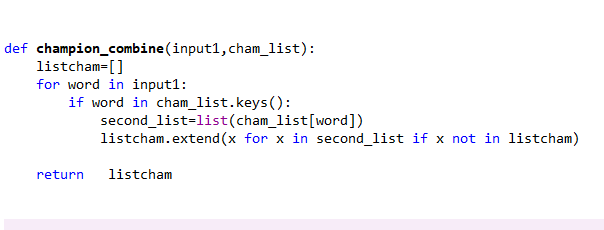
در این فایل تمامی preprocess های انجام شده در فاز 1 بر روی پرسمان ورودی انجام میشود ابتدا پرسمان به صورت واسط زیر از کاربر دزیافت شده و از او پرسیده میشود که به صورت champion باشد یا معمولی که مانند شکل زیر انتخاب میکند



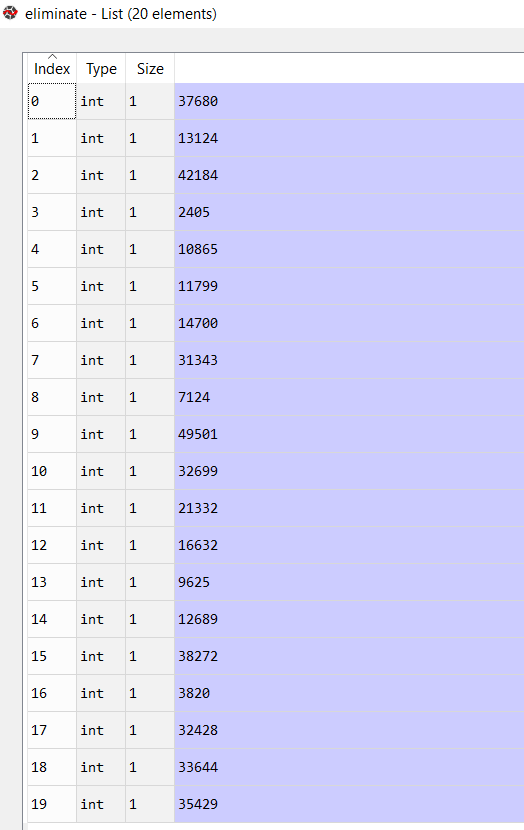
سپس query بعد از preprocess به صورت vector و tf idf آن بدست میآید مانند تابع tfidf در فایل jupyter با این تفاوت که در اینجا idf برابر با 1 در نظر گرفته شده است

برای eliminate کردن در این پروژه در تابع زیر ه این صورت در نظر گرفته شده است برای query که به صورت token شده است تمامی لیست هایی که حداقل شامل یکی از token ها باشد شده است و سپس فایل ها با هم مرج های یکتا مرج شده است چون ممکن است یک سند در چند کلمه باشد که این راه حل تا حد زیادی به ما میتواند کمکم کند.

همچنین فایل خروجی champion نیز مانند eliminate استفاده شده است با این تفاوت که در champion باید فقط مرج شود اما در eliminate پیدا هم باید بشود اسناد



برای استقلال و پرسپولیس champion list



برای یک سند به طور خاص که summury آن به صورت

رئیس سازمان سنجش آموزش کشور گفت: رأی هیأت عمومی دیوان عدالت اداری مبنی بر بلامانع شدن شرکت در کنکور سراسری پس از دو بار پذیرفته شدن در این آزمون ها، در حکم قانون است، اما هنوز به این سازمان ابلاغ نشده است. رئیس سازمان سنجش آموزش کشور گفت: رأی هیأت عمومی دیوان عدالت اداری مبنی بر بلامانع شدن شرکت در کنکور سراسری پس از دو بار پذیرفته شدن در این آزمون ها، در حکم قانون است، اما هنوز به این سازمان ابلاغ نشده است.

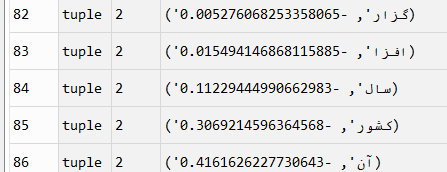
است

و content آن نیز قابل مشاهد ه است که بدلیل زیاد بودن در این جا نیورده شده است

اما در اکسل اول 5 ای است

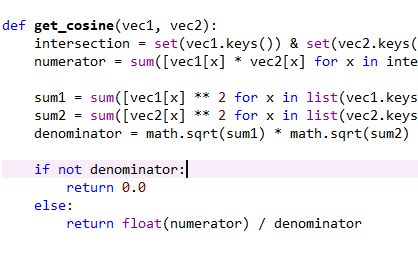
5 تا با بیشترین وزن و کم ترین وزن به صورت زیر میباشد





در حالت کلی خیر نمیتوان گفت وزن بیشتر بهتر است زیرا کلماتی مانند به برای و از این قبیل بیشتر تکرار شده اند در صورتی که در کوئری ما جایی ندارند و بدرد نمیخورند.

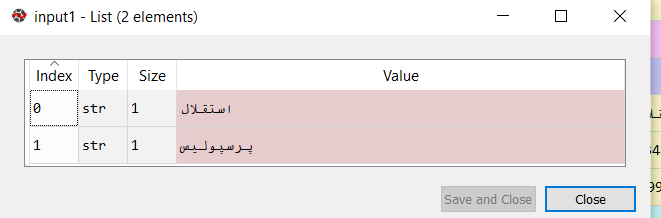
در تابع زیر طبق پروژه coise similarity محاسبه شده است



که بسار واضح است

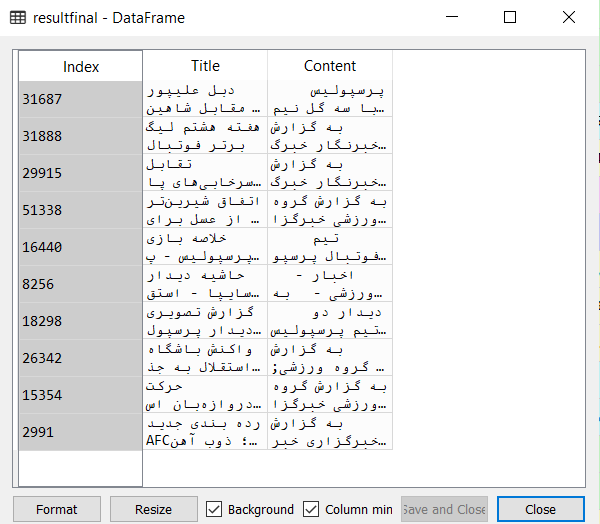
سپس از درختی بنام maxheap استفاده شده است که به تعداد elimination های مان سایز درخت است و در نهایت با insert کردن و maxheapify بهترین را مشخص کرده و k بهترین ها را خروجی میدهد

برای ورودی استقلال و پرسپولیس که این روزها بحث آن داغ است



که و بخاطر preprocess حذف شده است

و نتیجه برای حالت عادی که به نظر درست است



زمان آن 12.838354587554932time

و در حالت Champion زمان آن به صورت

10.241499423980713time

و نتیجه آن



که به نظر هگی درست میباشند اما شاید اختلاف در اوردن نتیجه در حالت کلی برای champion list بهتر میباشد و همینطور سریعتر

و خیلی خوب است که قبل از اجرای برنامه آن را داشته باشیم