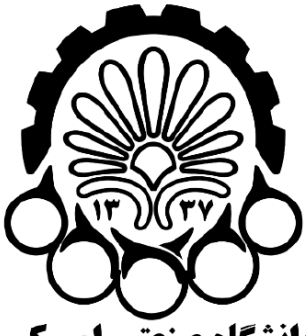
به نام او



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

روزبه بازرگانی، محمدرضا اردستانی

9513703، 9513004

مباحثی در علوم کامپیوتر

پروژه پایانی

استاد درس: دکتر اکبری

پاییز 1399

هدف:

هدف پروژه امتیاز دهی نظرات مردم در بین اعداد 1 تا 5 با توجه به نظر آن‌ها در مورد محصول خریداری شده در شرکت آمازون و به طور خاص قسمت فیلم و تلویزیون است. همچنین در گام دوم، هدف این است که ویژگی-هایی که هر نظر به آن اشاره می‌کند، به طور مثال در مورد فیلم برداری یا داستان فیلم، به همراه احساسات شخص در مورد آن ویژگی مشخص گردند.

کاربرد:

امتیاز هر محصول از این نظر که می‌تواند روی خرید مردم تاثیر بگذارد و همچنین آگاهی از نقاط ضعف و قوت هر محصول توسط خود سازنده، بسیار هائز اهمیت است. همچنین گاهی امتیازدهی افراد متناسب با متن آن‌ها نیست. لذا پیشبینی از روی دلایل آن‌ها می‌تواند نمره‌دهی بهتر و همچنین دقیق‌تری را پس از دسته بندی ویژگی‌ها و نمره‌دهی بر اساس آن‌ها فراهم آورد و سپس تولید کننده محصول ایراد محصولاتش را بیابد.

فایل‌ها:

این پروژه شامل دو قسمت می‌باشد. قسمت آموزش و بررسی نتایج الگوریتم‌ها که در قالب فایل Topics_in_cs_final_project.ipynb است و همچنین قسمت وبسایت که دارای فایل وبسایت (myweb.py) و فولدرهای Data، static و templates می‌باشد. در فولدر Data مدل‌های سیو شده، در static، فایل css و بکگراند سایت، و در نهایت در templates فایل‌های html وجود دارند.

شرح کار:

1. **خواندن دیتا از دیتاست:**

برای این کار 100,000 نظرات اولیه در مورد محصولات فیلم و تلویزیون سایت آمازون از این سایت دانلود شد.

پس از مشخص کردن تعداد داده‌هایی که برای آموزش و تست نیاز است، کد به صورت رندوم به تعداد مجموع داده‌های مورد نیاز از دیتاست داده انتخاب می‌کند به نحوی که تکراری نباشند. (توسط تابع sample در کتابخانه random). سپس از بین داده‌های انتخابی به طور رندوم داده‌های تست انتخاب می‌شوند. در نتیجه داده‌های آموزش و تست کاملا در سطح دیتاست پخش هستند که هدف اصلی ما در این قسمت بود.

از هر داده، متن نظر، تعداد دفعاتی که شخص رای داده، و همچنین اینکه اکانت شخص معتبر هست یا خیر استخراج شده و هر کدام جداگانه در یک لیست قرار گرفتند.

2. **پردازش اولیه:**

این پردازش شامل موارد زیر بود:

1) removing user names

2) removing numbers

3) removing URLs

4) removing punctuations

5) removing stopwords

6) transform to the lower case

7) stemming

3. **تشخیص احساسات توسط کتابخانه Textblob:**

در این قسمت، با ورودی دادن متن نظرات، احساسات رو توسط کتابخانه پیشبینی کردیم. در این قسمت از فقط از متن استفاده شد و از تعداد دفعاتی که شخص رای داده، و همچنین اینکه اکانت شخص معتبر هست یا خیر استفاده نشد. خروجی کتابخانه در بازه 1 و 1- بود. با انتقال خطی و همچنین با انتقال غیر خطی مبنی بر تعداد دفعات تکرار ریتینگ‌ها خروجی کتابخانه را به بازه 1 تا 5 بردیم. برای انتقال غیر خطی به دقت 58 درصد برای ریتینگ بین 1 تا 5 و 79.9 درصد برای دسته بندی نظرات به مثبت، منفی و خنثی رسیدیم.

4. **تبدیل جملات به بردار:**

در ابتدا کملات در متن‌های پردازش شده را جداسازی کردیم. سپس یک بردار تشکیل داده و همه‌ی کلمه را به یک المان نسبت دادیم. با توجه به بزرگی داده قابلیت تبدیل آن به ماتریس نبود و RAM اختصاص داده شده توسط Colab که 12 گیگا بایت بود پر می‌شد و کد کرش می‌کرد. در نتیجه کلمات به صورت tuple به شماره المان خود اختصاص داده شدند تا به خاطر حذف صفرها که اکثر المان‌های ماتریس را تشکیل می‌دادند، حجم بسیار کمتری اشغال گردد. نکته مهم این است که چون این تبدیل کننده به بردار خاص است، به صورت فایل json دخیره شده تا در وبسایت اطلاعات آن بازیابی گردد.
همچنین در هر مرحله داده‌ها در یک دیکشنری ذخیره شدند تا بتوان در پایان کار، آن‌ها را توسط کتابخانه pandas نمایش داد.
در انتها فایل تبدیل کننده جملات به بردار به اسم vectorizer ذخیره گردید تا در وب مورد استفاده قرار گیرد.

5. انتخاب تاثیرگزارترین کلمات و ایجاد داده‌ها برای آموزش:

برای انجام لرنینگ روی متن، پس از تبدیل کردن جملات به کلمات، حال با استفاده از روش X^2 تعداد 2000 داده که بیشترین تاثیرگذاری را داشتند، انتخاب گشتند. ماتریسی که هر سطر شامل یک متن و هر ستون شامل تکرار این کلمات است ایجاد شد. سپس دو ستون در آخر به عنوان تعداد دفعات رای دادن و معتبر بودن اکانت فرد، اضافه شد. قابل توجه است که از Ngram 1,2 استفاده شده است، لذا دارای تک کلمه و جفت کلمه هستیم. به عنوان مثال دارای could و couldn’t بودیم که بسیار در تشخیص احساسات کمک‌کننده بوده است. خروجی‌ها را نیز تنها به بردار numpy تبدیل کردیم تا در لرنینگ استفاده شوند.

6. تشخیص احساسات با لرنینگ:

در این قسمت از کتابخانه Keras که یکی از معروف‌ترین کتابخانه‌های یادگیری عمیق می‌باشد، استفاده شد. روش آموزش supervised بود، به نحوی که X_train داده آموزش، Y_train داده درست نهایی می‌باشد. داده‌ها تکرار زیادی روی نمره 5 داشتند و همانطور که در کد نیز خروجی گرفته شده است، معمولا در حدود 70 درصد ریتینگ‌ها 5 است. لذا برای آموزش کلاس‌ها وزن تعیین کردیم. به این صورت که رابطه‌ای معکوس با تعداد دفعات داشته باشد تا در تابع هزینه به عنوان ضریب، ارزش همه را یکسان کند. قابل توجه است که f_1-score نیز معیار بررسی می‌باشد و با فقط خروجی دادن 5 و دقت 70 درصد به نتیجه خوبی نرسیده‌ایم. لایه‌های مدل استفاده شده به شرح زیر است:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 2002, 16)          799216
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 2000, 128)         6272
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_4 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_5 (Dense)              (None, 6)                 390
=================================================================
Total params: 830,646
Trainable params: 830,646
Non-trainable params: 0
_________________________________________________________________
```

پس از پایان یادگیری، مدل بر داده آموزش عملکرد زیر را داشت:

```
781/781 [==============================] - 41s 52ms/step
              precision    recall  f1-score   support

           1       0.16      0.35      0.22      2503
           2       0.06      0.29      0.09      1540
           3       0.10      0.14      0.12      3002
           4       0.19      0.04      0.06      7568
           5       0.77      0.67      0.72     35338

    accuracy                           0.52     49951
   macro avg       0.26      0.30      0.24     49951
weighted avg       0.59      0.52      0.54     49951
```

و در نهایت پس از اجرا بر روی داده تست به نتایج زیر رسیدیم:

```
16/16 [==============================] - 1s 51ms/step
              precision    recall  f1-score   support

           1       0.14      0.31      0.19        49
           2       0.05      0.21      0.08        38
           3       0.12      0.20      0.15        60
           4       0.22      0.05      0.08       153
           5       0.76      0.66      0.70       698

    accuracy                           0.50       998
   macro avg       0.26      0.28      0.24       998
weighted avg       0.58      0.50      0.53       998
```

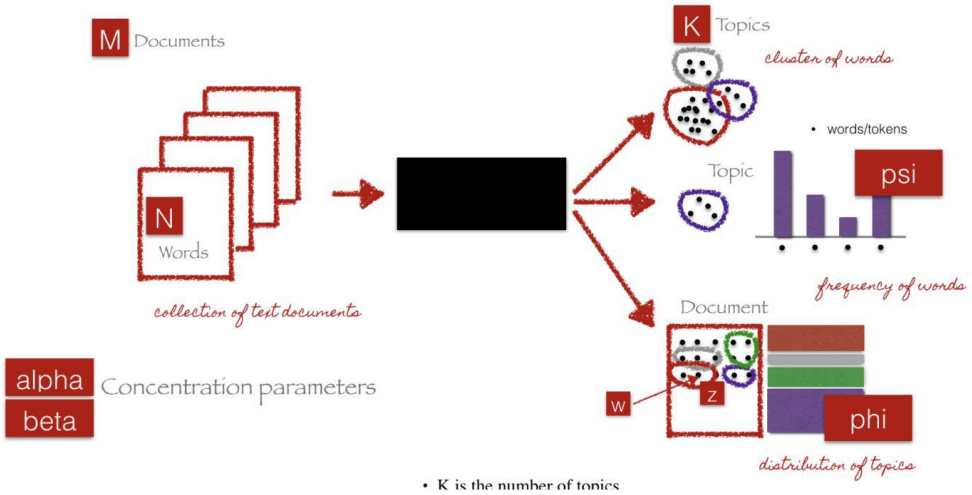
در مورد نتایج باید به این موضوع دقت نمود که دسته‌بندی نظرات بین 1 تا 5 بسیار پیچیده‌تر از دسته-بندی بین مثبت، منفی یا خنثی، همانطور که در قسمت Textblob هم دیدیم، می‌باشد زیرا تعیین مرز بین دسته‌ها با افزایش آن‌ها بسیار سخت‌تر می‌گردد.

برای مقایسه‌ی نتایج، از نتایج گروه‌های شرکت کننده در سایت kaggle استفاده کردیم که می‌توان دید نتایج آن‌ها در حدود 50 درصد بوده است.

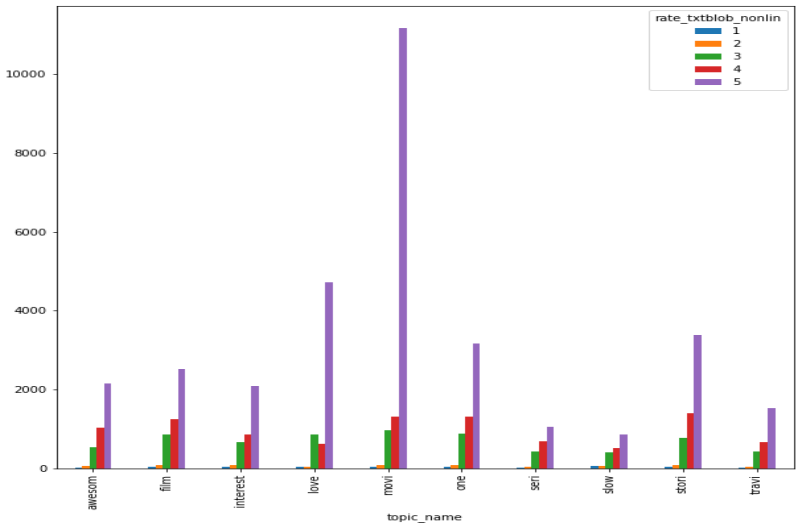
برای استفاده این روش در وب، مدل‌های chi-square و keras ذخیره شده‌اند تا در آنجا بازیابی گردند.

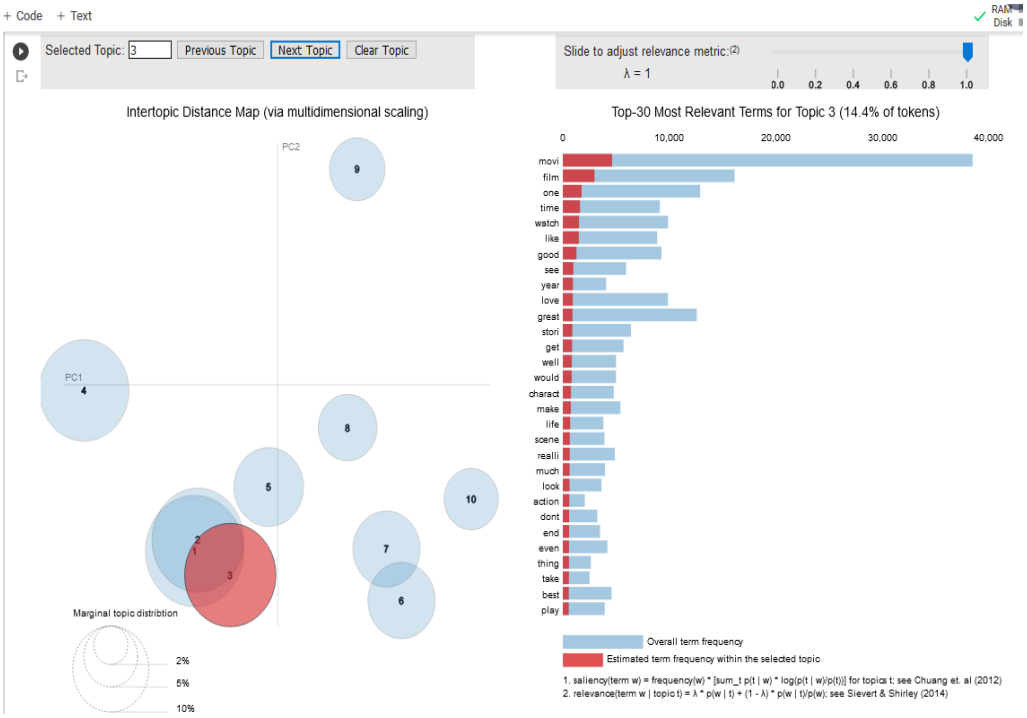
7. LDA

در این روش ما یک مجموعه از دایکومنت ها داریم و یک سری پارامتر های از پیش تعیین شده (مثلا تعداد تاپیک هایی که میخواهیم داشته باشیم ، پارامتر های آلفا و بتا و تعداد آیتریشن هایی که میخواهیم داشته باشیم) و بعد این پارامتر ها را به تابع (LDA) پاس میدهیم و این تابع سه خروجی با توجه به عکس پایین بر میگرداند.



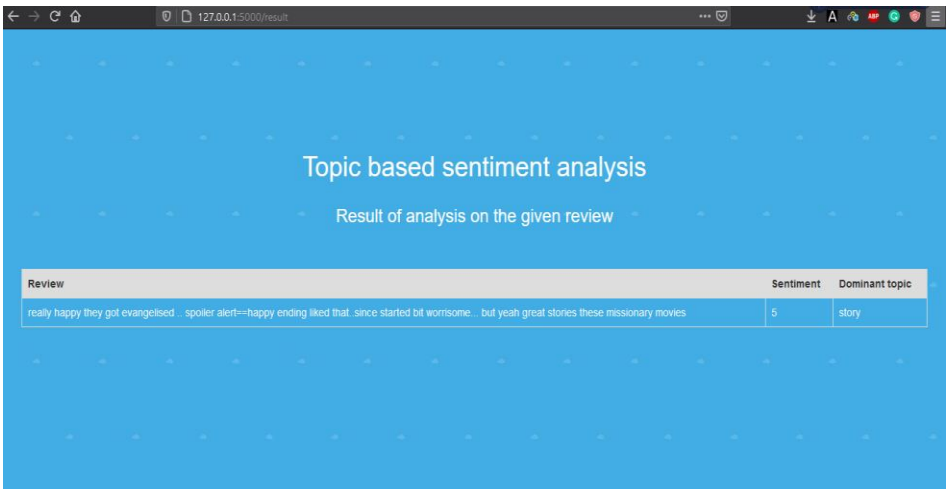
نتایج نیز به شکل زیر بود.





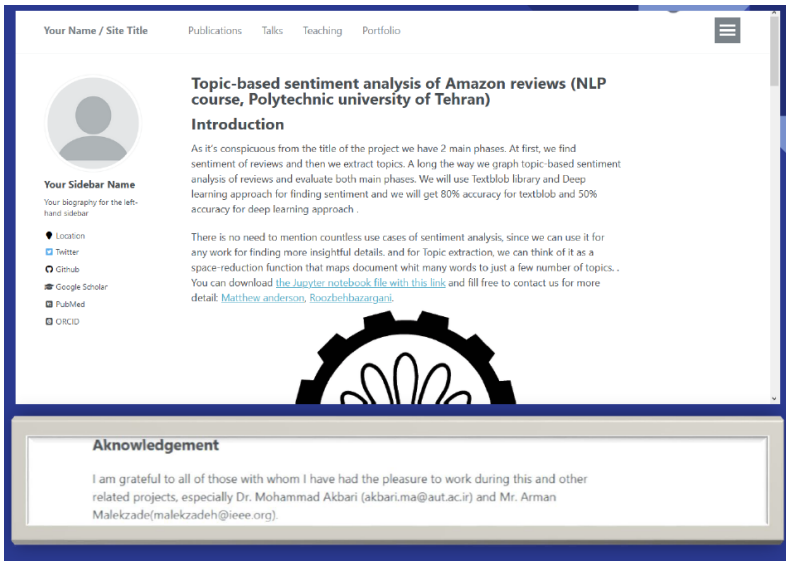
8. وبسایت:

برای وبسایت از کتابخانه Flask استفاده شد تا درخواست‌های داده شده به سایت را پاسخ گوید. به این منظور در سایت اصلی یک قسمت متن داریم که شخص متن مورد نظرش را وارد می‌نماید و سپس بعد از کلیک بر دکمه Submit، با لود کردن اطلاعات مدل‌ها، متن را پردازش می‌کنیم. پس از کلیک بر دکمه متن به صفحه /results انتقال می‌یابیم که نتایج در آن به صورت نتیجه تحلیل نظر و مرتبط‌ترین موضوع نمایش پیدا می‌کند. در صورت نیاز به وارد کردن متن بعدی کافیست به لینک صفحه اصلی بروید.



+ همین طور برای این که افراد بتوانند به پروژه ای که ما انجام داده ایم دسترسی داشته باشند یک

GitHub repository هم تدارک دیده شده و از این لینک میتوانید آن را مشاهده کنید. در این وب سایت ما به شرح کامل پروژه پرداخته ایم.



اجرای کد:

برای اجرای کد نوتبوک، کافی است که آن را در محیط Colab اجرا نمایید یا اینکه کتابخانه‌ها را نصب داشته باشید که در اولین سل‌ها کتابخانه‌ها همگی موجود می‌باشند، به جز کتابخانه Keras که در بخش DL موجود است. برای وب سایتی که پردازش انجام می‌دهد، کافیست آن را اجرا کنید و به لینک داده شده رفته و سایت را تست نمایید. همچنین به دلیل وجود مشکل در نصب Keras نتوانستم تابع آن را اجرا نمایم، اما به صورت کامنت شده قرار می‌دهم تا با از کامنت درآوردن تابع و قسمت flask که تابع را در وب صدا می‌زند، بتوانید بررسی نمایید. همان تابع در Colab موجود است و برنامه را اجرا می‌نماید.

با سپاس