

**گونه­ی محاوره ای در مقابل کونه­ی رسمی**

تهیه کنندگان:

معین سلیمی

سحر زال

حسین پرتو

تمرين چهارم

درس پردازش زبان­هاي طبيعي

استاد درس :

آقاي دکتر احسا ن الدين عسگري

زمستان 1401

|  |  |
| --- | --- |
| ***فهرست مطالب:*** |  |

[*فهرست مطالب:* ‌ب](#_Toc126906219)

[1 مقدمه و تعریف مسئله 3](#_Toc126906220)

[2 طبقه بندی متن 3](#_Toc126906221)

[2‌.1‌ مدل رگرسیون لجستیک 4](#_Toc126906222)

[2‌.2‌ استفاده از fasttext 7](#_Toc126906223)

# مقدمه و تعریف مسئله

امروزه در سطوح و کاربردهاي متعددي از پردازش متن استفاده ميگردد. یکی از موارد مورد بحث در این حوزه بررسی متن­های با زبان محاوره­ای است. که کاربرد های مختلفی از جمله نظارت بر شبکه­های اجتماعی، ارتباط با مشتریان، تحلیل نظرات افراد، دسته بندی مدارک متنی و ... می­تواند داشته باشد.

در این تمرین متن های محاوره ای و رسمی در سه بخش طبقه بندی متن، طبقه بندی توکن و تولید متن مورد بررسی قرار گرفته اند و هر بخش با استفاده از روش­های گفته شده پیاده سازی شده است که در ادامه به بررسی هر کدام از آن­ها پرداخته می­شود.

# طبقه بندی متن

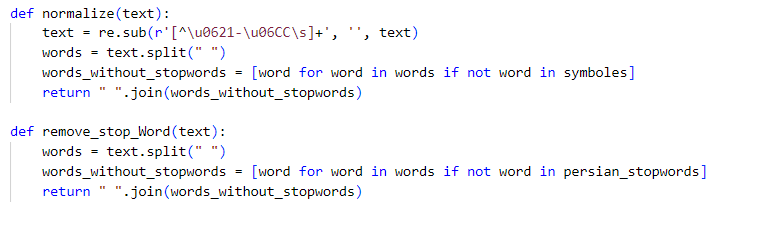
در این بخش قرار بود که جملات مختلف به دسته­های رسمی و محاوره­ای تقسیم بندی شوند. به طور کلی تفاوت جملات رسمی و محاوره­ای در چند مورد است. شکل بعضی کلمات در حالت محاوره ای و رسمی متفاوت است، بعضی کلمات صرفا در یکی از حالات مورد استفاده قرار می­گیرند، جملات رسمی ساختارمند هستند و از قواعد زبانی تبعیت می­کنند. که دو مورد اول در بخش بعدی (طبقه بندی توکن ) مورد بررسی قرار گرفته است.

دیتای مورد نیاز برای استفاده در این بخش در اختیار ما قرار گرفته بود که حاوی جملاتی بود که هر کدام به دسته های رسمی یا محاوره­ای تقسیم شده بودند. در ادامه روش های پیاده شده برای این بخش به تفکبک آمده است.

## مدل رگرسیون لجستیک

این روش یکی از دسته بندهای ساده است که به صورت از پیش تعریف شده در دسترس است. همچنین طبق صورت سوال تعیین شده است که باید از Tf-Idf استفاده نماییم. در واقع متن ورودی نیاز است تحلیل و پردازش شود تا ویژگی­های خاص آن استخراج گردند و تبدیل به ورودی مدل regression-Logistic شوند.

برای آماده سازی داده­ها ابتدا نیاز است که داده­های مربوطه نرمال سازی شوند به این منظور دو تابع به صورت زیر به کار برده شده اند.



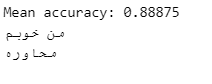
دلیل جدا کردن این توابع به این دلیل بود کهstopword ها اگرچه اطلاعات زیادی از متن در اختیار ما قرار نمی­دهند ولی شکل متفاوت ادا شدن آن­ها در حالت رسمی و محاوره­ای می­تواند بسیار به طبقه بندی ما کمک کند.

به همین منظور یک بار با استفاده از آن­ها و یک بار بدون آن­ها این بخش انجام شد و نتایج آن مورد بررسی قرار گرفت که مطابق انتظار در حالتی که این کلمات حذف نشده بودند نتایج بهتری به دست آمد که در ادامه به آن خواهیم پرداخت.

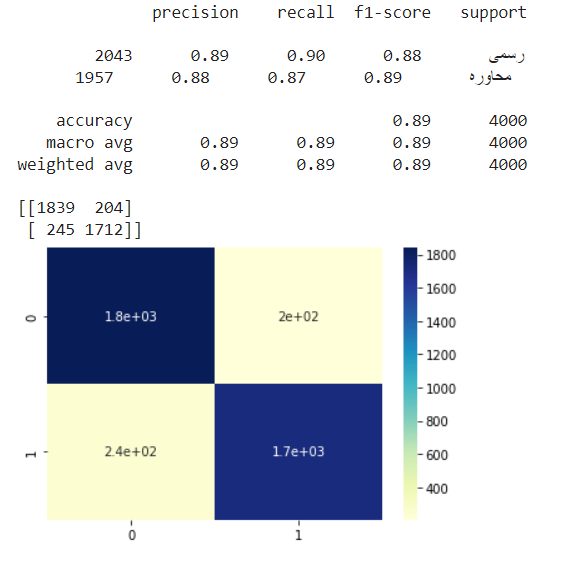
در مرحله­ی بعد با استفاده از TfidfVectorizer بردار ویژگی­های tf-idf استخراج شده­اند.

از آن­جایی که در این بخش از ما خواسته شده بود که cross validation را محاسبه کنیم به همین منظور از kFold استفاده کردیم در مرحله­ی آخر داده­های train و test هر دسته رو به logestic-regration دادیم .

نتایج به دست آمده در این بخش برای حالت های بدون حذف stop word برابر 0.88 و برای حالت حذف برابر 0.82 شد.



همچنین برای بررسی بیشتر یک بار نیز داده­ی تست را جدا کردیم و نتایج آن را اندازه­گیری نمودیم که نتایج به صورت زیر به دست آمد.



همان گونه که مشخص است تقریبا به صورت متعادلی خطا در تعیین برچسب ها داشته­ایم.

مدل مربوطه همچنین tfidf ذخیره شده­اند و برای استفاده از آن­ها نیز می­توان به فایل use\_saved\_model.py مراجعه کرد و با دادن جمله­ی دلخواه برچسب متناسب با آن را بر اساس مدل به دست آورد . در این فایل مدل ذخیره شده خوانده می­شود و بر چس از نرمال سازی جمله به مدل ذخیره شده داده می­شود .

یک نوت بوک نیز در این فایل دیده می­شود که برای تحلیل بخش جدا شده­ی داده های تست قرار گرفته است.

## استفاده از fasttext

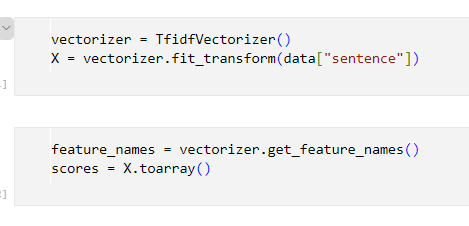
مراحل کلی این بخش به این صورت بود که بعد از نرمال سازی، بردار­های ویژگی fasttext استخراج شد . همچنین tf-idf کلمات نیز محاسبه شد . با استفاده از tfidf میانگین وزن دار بردار­ها محاسبه شده و در آخر یک شبکه­ی LSTM با این بردار ها آموزش دید. که در ادامه به هر مرجله می­پردازیم.

در ابتدا ذکر شود که به دلیل محدودیت­های موجود در استفاده از fasttext به اجبار این بخش روی colab انجام شد و نوت بوک و مدل ذخیره شده در اختیارتان قرار گرفته است.

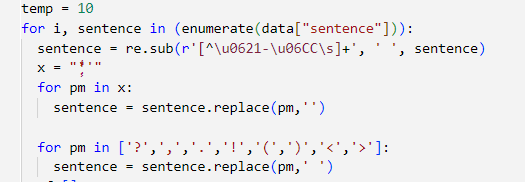
در ابتدا مدل fasttext برای استخراج embedding کلمات به صورت زیر استفاده شد.

fasttext.load\_model("cc.fa.300.bin")

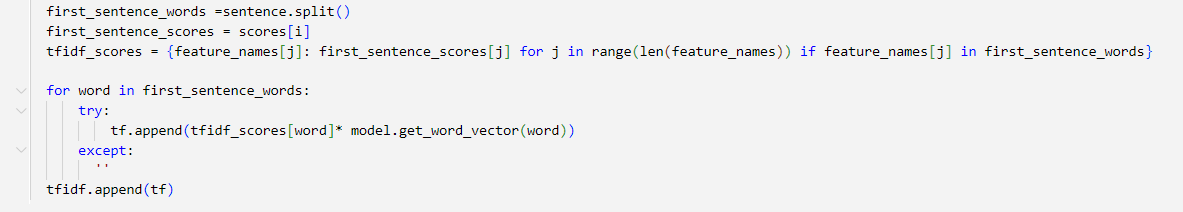
همچنین tfidf کلمات نیز به صورت زیر به دست آمد.



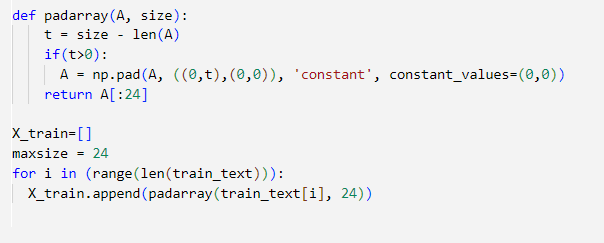
مانند بخش قبل جملات نرمالیزه شدند.



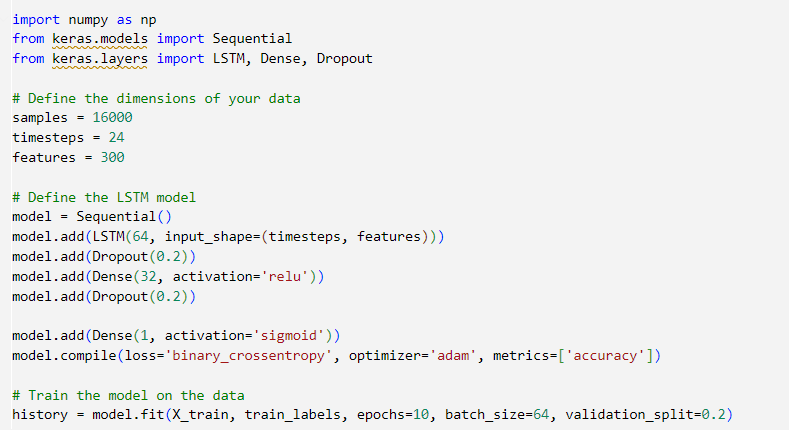
حال جمله به جمله پیش میرویم. در هر جمله کلمات را جدا میکنیم و بررسی میکنیم که tfidf و بردار embedding آن به چه صورت است.



حال که بردار­های وزن­دار استخراج شد موقع آموزش شبکه است ولی مشکلی که در اینجا وجود داشت این بود که طول همه­ی جملات یکسان نبود . برای همین به صورت زیر یک zero padding اضافه شد و با تبدیل بردار­ها آن­ها را آماده­ی آموزش کردیم.



در مرحله­ی بعد شبکه­ی Lstm به صورت زیر پیاده سازی شد.



مدل آموزش دیده شده در این مرحله ذخیره و در اختیارتان قرار گرفته است .

در آخر با استفاده از این شبکه به پیش بینی داده های تست پرداختیم که نتایج زیر حاصل شد:

