گزارش تمرین شماره ۳

طبقه بندی کامنت های دیجی کالا

گردآوری : **علی شریفی**

شماره دانشجویی : **۹۵۲۲۲۰۴۶**

۱ مقدمه و مسئله :

این تمرین در راستای آشنایی با طبقه بندی و الگوریتم های مرتبط با آن طراحی شده است . داده های مورد استفاده در این تمرین ، کامنت های لیبل خورده و لیبل نخورده شرکت دیجی کالا میباشند . تعداد داده های لیبل خورده ، مورد استفاده قرار گرفته برابر ۱۶۲۰۰۰ میباشد . که هر داده دارای ۵ رکورد میباشد که این رکورد ها به ترتیب عبارتند از :

- id
- title
- comment
- rate
- verification-status

رکورد verification-status همان رکورد هدف ما می باشد یعنی ما میخواهیم با ساخت بهترین طبقه بندی ممکن با بررسی متن کامنت به این نتیجه برسیم که آیا کامنت مورد نظر باید منتشر شود و یا نه .
اهمیت ساخت طبقه بندی برای کامنت ها از جایی مورد اهمیت قرار میگیرد که در شرکت های بزرگ روزانه هزاران کامنت ارسال میشود و بررسی تک به تک این کامنت ها کاری بسیار دشوار و هزینه بر است و نیروی انسانی عظیمی را می طلبد به همین جهت با پیدا کردن مدل خوبی که این کار را انجام دهد تا حد بسیار زیادی انتشار کامنت ها به صورت اتوماتیک صورت میگیرد . البته لازمه بالا بودن عملکرد این روش آن است که مدل مورد استفاده به صورت مداوم مورد ارزیابی و اصلاح قرار گیرد .

۲ راه حل و ایده ها :

با توجه با ماهیت کامنت ها که داده های متنی میباشند و الگوریتم های طبقه بندی ما با اعداد سرکار دارند ، بخش اول کار باید پیدا کردن راه حلی برای تبدیل داده های متنی به داده های عددی باشد به گونه ای که دو کامنت قابل مقایسه با هم دیگر باشند .

برای به دست آوردن نتیجه مطلوب در زمینه داده های متنی ، نیاز است که ابتدا داده ها پاکسازی شوند برای این کار از کتابخانه Hazm در زبان پایتون استفاده میکنیم که براساس نسخه عربی کتابخانه NLTK دانشگاه استنفورد ، برای زبان فارسی ایجاد شده است بهره میبریم. پس طی مراحل Normalize برای استاندارد کردن فاصله کلمات و حذف نیم فاصله ها ، Stem' جهت تبدیل کلمات جمله به حالت مفرد خود همانند تبدیل کلمه کتاب ها به کلمه کتاب ، lemmatization یافتن بن مضارع و ماضی فعل ها که ما از بن ماضی بهره بردیم . حال در این لازم است که کلمات stopword از متن حذف شوند چرا که این کلمات نقش به سزایی در ماهیت جملات ندارد همانند حروف ربط که برای این کار از حدود ۴۰۰ کلمه موجود در صفحه گیت پروژه Hazm بهره بردم .

پس طی مراحل فوق داده پاکسازی شده آماده تبدیل به داده های عددی میباشد . در پروژه از دو رویکرد متفاوت استفاده شده است .

۱. استفاده فقط از بخش پاکسازی کامنت ها (comment) پس از مراحل گفته شده و وضعیت انتشار (verification-status)

۲. استفاده از بخش پاکسازی کامنت ها (comment) پس از مراحل گفته شده به همراه بخش پاکسازی نشده تیتر (title) که این یک متن نظر گرفته شده اند و وضعیت انتشار (verification-status)

برای تبدیل داده های متنی به داده های عددی از کتابخانه gensim و روش Doc2vec بهره میبریم . برای آموزش Doc2vec از پارامتر های زیر بهره میبریم .

- dm=0 , distributed bag of words (DBOW) is used.

- vector-size=300 , 300 vector dimensional feature vectors.
- negative=5 , specifies how many “noise words” should be drawn.
- min-count=1, ignores all words with total frequency lower than this.

این مدل را با ۳۰ epochs آموزش میدهم .
که مدل حاصل از موارد گفته شده از رویکرد ۱ را model-dbow و مدل حاصل از از موارد گفته شده از رویکرد ۲ را model-dbow1 مینامیم.
پس دستیابی به داده های عددی با تقسیم ۷۰ - ۳۰ داده ها شروع به پیاده سازی طبقه بندی میکنیم . برای طبقه بندی داده ها دو رویکرد در نظر گرفته میشود .

1. Logistic Regression
2. Naive Bayes

۳ ارزیابی نتیجه ها :

ابتدا به بررسی خروجی مدل model-dbow میپردازیم .

```
Testing accuracy:
 0.8341358024691358
```

و برای خروجی مدل model-dbow1 نیز داریم

```
Testing accuracy:
0.8341358024691358
```

برای خروجی مدل model-dbow برای طبقه بندی بیز داریم :

```
Accuracy NB: 0.78
```

برای خروجی مدل model-dbow1 برای طبقه بندی بیز داریم :

```
Accuracy NB: 0.73
```

۴ جمع بندی و نتیجه گیری

با توجه به خروجی ها به دست آمده انتظار میرود با تلاش بیشتر و صرف وقت بیشتر بر روی بخش پاکسازی داده ها مثلا افزایش تعداد stopwords ما قادر خواهیم بود که به مدل های عددی بهتری دست یابیم . هم چنین انتظار میرود که پیچیده کردن مدل ها بتواند نقش قابل توجه ای در افزایش کارآیی ما داشته باشد.