گزارش پروژه 3 هوش مصنوعی
شبکه بیزی

استاد فدایی

محمدرضا شش پری
810198417

بهار1401

هدف پروژه

هدف از انجام پروژه ساخت یک classifier با الگوریتم Naive Bayes Classifier برای دسته بندی کردن مقاله های سایت دیجی کالا مگ به یکی از 4 دسته هنروسینما – علم و تکنولوژی – بازی ویدئوبی - سلامت و زیبایی با استفاده از متن مقاله می باشد.

تعریف مساله

در این مساله دو دیتاست از سایت دیجی کالا مگ شامل مقاله و موضوع آن برای train و test به ما داده شده و از ما خواسته شده تا با الگوریتم گفته شده و این داده ها ، یک مدل طبقه بندی train کنیم و با استفاده از آن دسته بندی مقاله های test را بدست بیاوریم.

همچنین دو مجموعه تمرین و تست به ترتیب شامل 5200 و 802 سطر داده هستند و از آنجایی که تعداد آگهی با موضوعات متفاوت با هم یکسان است، نیازی به resampling نیست.

پیش پردازش داده

در این مرحله ابتدا با استفاده از کتابخانه های pandas ، دیتاست های داده شده را به دیتافریم pandas تبدیل کردیم .

سپس برای سهولت در کار کردن ، ابتدا با استفاده از LabelEncoder از کتابخانه sklearn ، به هر یک از label های دیتاست های داده شده، یک عدد را نسبت می دهیم.

در ادامه اول برای همه سطر های ستون content دیتاست های داده شده از قابلیت های tokenize و normalize و stemmer کتابخانه هضم استفاده کردیم .

Tokenize : یک رشته از کلمات را بعنوان ورودی گرفته و آن را به لیست کلمات تشکیل دهنده آن که با space از هم جدا شده اند تبدیل می کند.(تبدیل جمله به کوچکترین واحد یا همان token)

Normalize : یک رشته از کلمات را بعنوان ورودی گرفته و با تبدیل half space های موجود در آن به space و بالعکس، متن فارسی را مرتب می کند.

Stemming: با حذف چند کاراکتر آخر یک کلمه، صورت مفرد یا ساده تر کلمه (یا میتوان گفت ریشه کلمه) را خروجی می دهد، هر چند یک سری ایراد نگارشی در کلمه خروجی وجود داشته باشد.

Lemmatize: این عمل نیز ریشه کلمه را خروجی می دهد ولی این ریشه خروجی شکل صحیح کلمه ورودی است و الزامی ندارد که از حذف چند کاراکتر آخر کلمه ورودی به وجود آمده باشد. (بصورت کلی lemma کلمه را خروجی می دهد.)

علاوه بر تفاوت گفته شده دو عمل lemmatize و stemming (حذف چند کاراکتر آخر) می توان به تفاوت در زمان اجرای دو عمل هم اشاره کرد که به وضوح عمل lemmatize بیشتر طول می کشد.

هدف همه این کار ها ساده کردن پردازش متن و همچنین پیدا کردن کلمات مشابه (برای پردازش راحت تر) متن با صورت های متفاوت است.

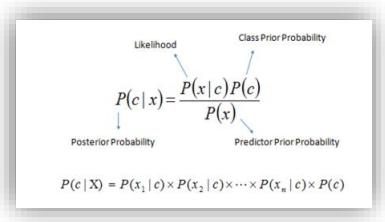
همانطور که گفته شد ، ابتدا از سه عمل tokenize و normalize و stemming استفاده شد ولی خروجی آن درست نبود (خروجی ها با توضیحات داده شده کتابخانه همخوانی نداشت.) بنابراین روند پیش پردازش را تغییر دادم و در ابتدا تمام half-space های متن را به space تبدیل و سپس متن را tokenize کردم.در ادامه یک مجموعه ای از حروف اضافی که نشان دهنده دسته خاصی نیستند (مثل از – به – با – برای و ...) ساختم و آن ها از متن حذف کردم.

بنظرم علت کارنکردن کتابخانه هضم، وجود کاراکترهای ناخواسته در متن ورودی است.

فرآیند مساله

در این مرحله ، ابتدا دو dictionary می سازیم. رابطه بین این دو دیکشنری به این صورت است که یکی از این دیکشنری ها ، همه کلمات دیده شده در متن های موجود در train dataset را به دیکشنری دیگری که نشان دهنده تعداد کلمات موجود از آن کلمه در هر دسته بندی است، map می کند.

علت انجام این کار (bag of words) محاسبه راحت تر احتمال های خواسته شده است .در قانون احتمال شرطی ما داریم:



که در آن prior probability نشان دهنده احتمال اولیه یک رخداد است که قبل از آشکار شدن و مشاهده مدارک آن محاسبه شده و posterior probability نشان دهنده احتمال یک رخداد بعد از مشاهده مدرک برای رخداد است. Likelihood نیز بیانگر احتمال وجود یک متغیر در یک دسته (تعلق یک مقاله به دسته بندی) است. این احتمال ها براساس قضیه بیز و همچنین با استفاده از موارد دیده شده در دیتاست train است.

Bigrams

کره زمین در کهکشان راه **شیری** قرار دارد.

مهدی **شیری** بازیکن استقلال است.

اگر فقط بخواهیم کلمه شیری را درنظر بگیریم، میبینیم که دسته بندی درستی اعمال نخواهد شد ولی اگر از هر جمله حداقل 2 کلمه برداریم، آنوقت میتوانیم معنی دقیق تری از کلمه شیری داشته باشیم و آن را در دسته بندی درست تری قرار دهیم.

از آنجایی که نتایج بدست آمده مطلوب بود و همچنین این بخش اختیاری است، در پروژه اعمال نشده.

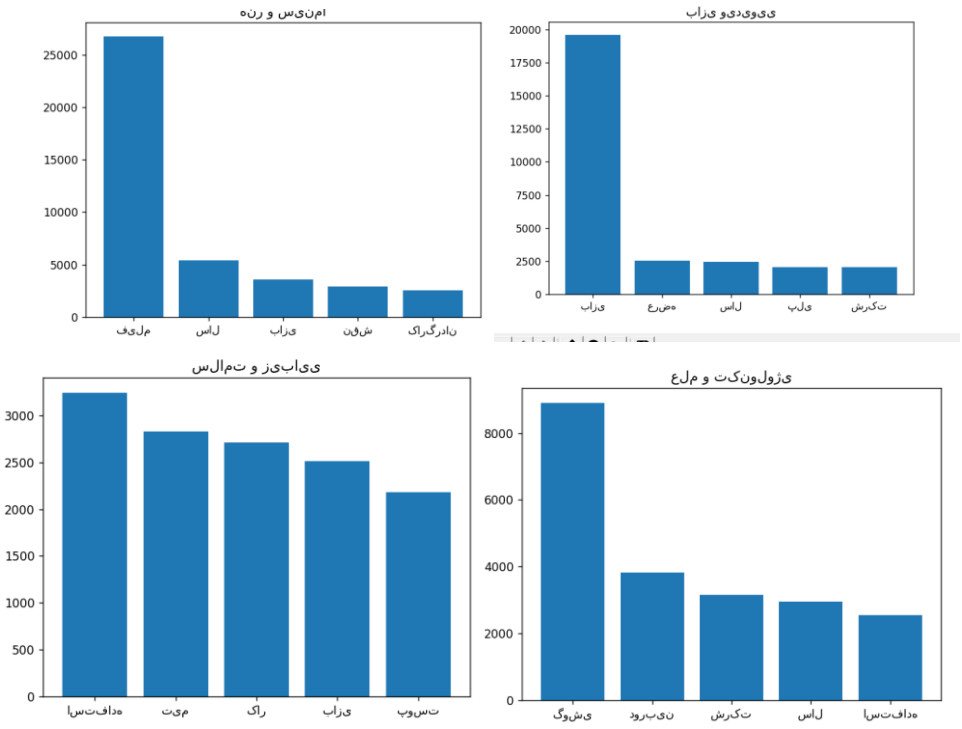
Additive smoothing

در فاز تست و هنگام پیشبینی کلاس یک کلمه، اگر کلمه ای وجود داشته باشد که در هیچ کلاس از پیش تعریف شده ای وجود نداشته باشد، آنگاه احتمال آن صفر خواهد شد و ضرب شدن صفر در بقیه احتمال ها موجب صفر شدن احتمال کل آن دسته از کلمات می شود که خب این موضوع نتیجه گیری درستی نیست.

برای رفع این مورد از روش additive smoothing استفاده می کنیم که در این روش احتمال یک کلمه را با یک عدد کوچکی جمع می کنیم تا از صفر شدن عبارت جلوگیری کنیم. این عدد کوچک را 0.5 درنظر گرفتیم.

بررسی صحت

خروجی بیشترین 5 کلمه پرتکرار در هر دسته بندی بصورت زیر شد:



ارزیابی

Precision به تنهایی خوب نیست، چون به ما اطلاعاتی درباره تعداد تمام مقاله های آن دسته نمی دهد. یعنی ما اطلاعاتی درباره همه مقاله های موجود در آن دسته که جزو دسته درست ، دسته بندی نشده اند نداریم.

Recall نیز به این دلیل به تنهایی خوب نیست که تعداد پیش بینی های غلط برای دسته مورد نظر را درنظر نمی گیرد و ignore می کند.

متغیر F1 ، میانگین هارمونیک (یعنی علاوه بر میانگین دو متغیر به نزدیکی دو متغیر نیز توجه می کند) دو متغیر recall و precision را حساب می کند.علت اهمیت این موضوع در این بخش، بخاطر اهمیت هر یک از متغیر ها به یک بخش از نتیجه بدست آمده (توضیحات بالا) است.

Macro : میانگین بین مقادیر بدست آمده.

Weighted : محاسبه میانگین وزن دار که وزن هر دسته ، تعداد اعضای آن است.

	Science Technology	Art Cinema	Video Games	Health Beauty	All Classes
Precision	98	88	30	92	-
Recall	26	22	98	30	-
F1-score	42	36	46	46	-
Accuracy	-	-	-	-	44.2
Macro Avg	-	-	-	-	42.5
Micro Avg	-	-	-	-	42.3
Weighted Avg	-	-	-	-	42

non-additive smoothing

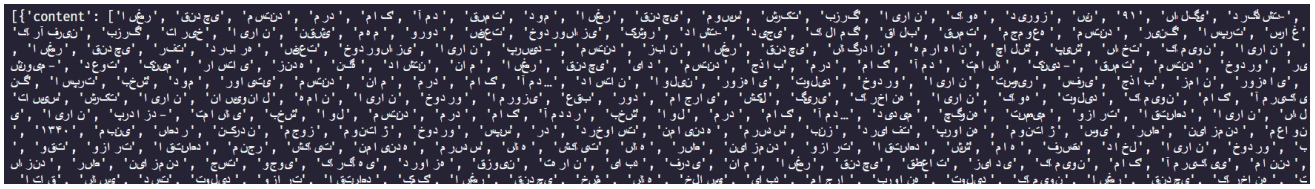
	Science Technology	Art Cinema	Video Games	Health Beauty	All Classes
Precision	94	93.6	70	90	-
Recall	80	79	93	89	-
F1-score	87.3	85.7	79	89.6	-
Accuracy	-	-	-	-	85.03
Macro Avg	-	-	-	-	85.4
Micro Avg	-	-	-	-	85.2
Weighted Avg	-	-	-	-	85.5

additive smoothing

ارزیابی

همانطور که در جدول میبینیم با استفاده از additive smoothing ، نتایج و دقت ما پیشرفت چشمگیری داشت .همچنان دقت قسمت بازی های ویدئویی نسبت به سایر بخش ها کمتر است که علت اصلی آن مشکل در قسمت pre-process متن است و کار نکردن کتابخانه ها و همچنین خیلی کامل نبودن مجموعه stop words ها است.

نمونه اشتباه :



علت اصلی این اشتباه ها بخاطر پیش پردازش نه چندان درست متن است . برای مثال اعداد فارسی از متن حذف نشدند. یا یک سری حروف اضافی و بی معنی در متن ها باقی ماندند. همچنین چون از bigrams استفاده نشده، بعضی کلمات در دسته بندی اشتباهی قرار گرفته اند.