پردیس دانشکده های فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

گزارش پروژه سوم درس هوش مصنوعی

عنوان پروژه : پردازش متن و شبکه های بیزین

استاد : سرکار خانم دکتر فدایی

رضوان بهمنی

810197473

مقدمه

در این پروژه یادگیری ماشین که در دسته پردازش زبان طبیعی قرار می گیرد(NLP)، درصدد آن هستیم که مدلی را آموزش دهیم که توانایی دسته بندی اخبار را داشته باشد. مدل باید با داشتن توضیح کوتاه برای هر خبر، دسته ای که آن خبر در آن قرار می گیرد را تشخیص دهد. در فاز اول این پروژه دسته های خبری شامل دسته های Travel و Business می باشد و در فاز دوم به دو دسته مذکور، دسته Style & Beauty نیز اضافه می گردد. در این پروژه از روش Bag of Words و قائده احتمالاتی بیز برای حل مساله استفاده شده است. در ادامه مفصلا مراحل کار برای ساخت و ارزیابی این مدل تشریح شده است.

راهبرد حل مساله

روند اجرایی این پروژه به این صورت می باشد که در ابتدا باید داده ورودی مدل آماده شده و برای آموزش مدل آماده گردد. این داده خام پیش از ورود به روند مدل سازی، باید از مرحله پیش پردازش(Pre-Processing) عبور کند. در این مرحله از روش هایی برای مرتب سازی و پاک سازی داده استفاده می شود. در این مرحله باید دیتا بصورت بسته هایی از کلمات جایگزین گردد. بعد از آماده سازی داده در مرحله پیش پردازش، آن را به دو بخش تقسیم میکنیم. بخش اول براش آموزش مدل(Train) و بخش دوم برای ارزیابی مدل(Evaluation) استفاده می گردد. پس از آموزش مدل و ارزیابی آن، شاخص های ارزیابی محاسبه شده و ارائه می شوند. در نهایت داده تست(Test) به مدل داده شده و نتایج طبقه بندی مدل ارائه می شوند.

پیش پردازه داده

همانطور که پیشتر توضیح داده شد، در این مرحله در صدد آن هستیم که به کمک استفاده از روش هایی بتوانیم داده خام را به بهترین نحو به شکل کلمات درآورده تا بتوانیم مدل خود را آموزش دهیم. این فرآیند نرمال سازی در این پروژه شامل مراحل زیر بوده است:

- حذف علائم نگارشی و اعداد

تمامی کارکتر ها بجز حروف الفبا اعم از علائم نگارشی و اعداد از داده خام حذف شده اند. این کاراکتر ها ارزشی در این پروژه نداشته و حذف آن ها باعث افزایش دقت آموزش مدل می گردد.

- حذف کلمات پرتکرار (stop words)

تمام متون دارای کلمات پرتکراری هستند که ارزشی در آموزش مدل ندارند. این کلمات شامل ضرایب فاعلی و مفعولی، ضرایب اشاره، کلمات تعریف و موارد ازین دست هستند. برای حذف این موارد از داده از کتابخانه nltk که ماژولی برای پردازش زبان های طبیعی است استفاده شده است. شایان ذکر است که کلمات پرتکرار در این ماژول با حروف کوچک تعریف شده اند لذا انجام مرحله پیشین یعنی تبدیل حروف بزرگ به کوچک ضروری بنظر می رسد.

پاسخ سوال 1

- جایگزین کردن کلمات با ریشه آن ها به کمک روش های stemming یا lemmatization

از آنجا که کلمات یک زبان دارای اشکال مختلفی از جمله فعل، اسم، صفت و قید هستند، بهتراست کلمات را به ریشه آن ها تبدیل کنیم. روش stemming با حذف پسوند ها و پیشوند ها از کلمات آن ها به ریشه خود تبدیل می کند. روش lemmatization علاوه بر حذف پسوند ها و پیشوند ها، چک کردن کلمه نهایی و معتبر بودن آن را چک می کند. از میان این دو روش، lemmatization عملکرد بهتری از خود نشان داد. استفاده از این دو روش باعث افزایش دقت مدل گردید.

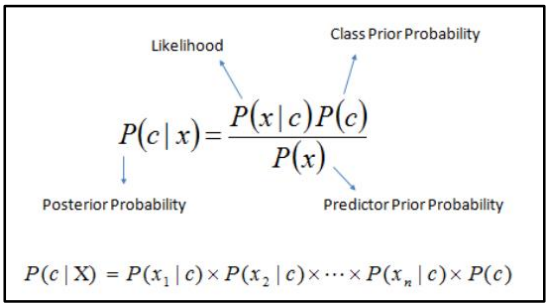
آموزش داده

داده ای که در مرحله پیش پردازش آماده شده است هم اکنون در این مرحله مورد استفاده قرار می گیرد.

در این پروژه از هر دو ستون title و description برای آموزش و ارزیابی مدل استفاده شده است و با دادن وزن بیشتر به description. گرچه این امکان فراهم بود که فقط با استفاده از داده یکی از ستون ها بتوان مدل را آموزش داد اما برای افزایش دقت مدل از هردو ستون استفاده شده است.

پس از آماده سازی بسته کلمات برای هر ردیف از دیتافریم، Bag of Words را کامل میکنیم. به این صورت که برای هر کتگوری، هر کلمه و تعداد تکرار آن کلمه را ذخیره میکنیم.

سپس از قائده بیزین برای محاسبه احتمال تعلق کلمات به دسته های مختلف استفاده شده است. فرض اساسی در استفاده از این قائده در این پروژه آن است که احتمال تعلق هر یک از کلمات به هر دسته از این احتمال برای کلمات دیگر مستقل است.



پاسخ سوال 2

- Posterior

احتمال آنکه واژه x در دسته c قرار گیرد. این احتمال برای تمام واژه های یک متن حساب شده و برای هر دسته امتیاز محاسبه می گردد و دسته با امتیاز بیشتر بعنوان دسته این متن پیش بینی می شود.

- Likelihood

احتمال آنکه بادانستن دسته c، واژه x در این دسته قرار گیرد.

- Prior

احتمال وجود دسته c با توجه به تعداد کل ردیف های دیتای ورودی

- Predictor

احتمال وجود واژه x که در واقع یک نرمالیزیشن است و به علت برابر بودن این مخرج در محاسبه احتمال هر 6 کتگوری، میتوان از Predictor چشم پوشی کرد

پاسخ سوال 3

واژه "شیر" در جملات زیر معانی متفاوتی دارد. در جمله اول یه معنی مایع خوراکی، در جمله دوم یه معنی یک حیوان است.

1. شیر ویتامین های بسیاری دارد.
2. شیر حیوانی وحشی است.

در مثال های بالا، اگر از bigram استفاده کنیم، با توجه به فقط یک واژه ی بعد از شیر میتوان به تفاوت معنی پی برد. اما در 2 جمله بعدی، با استفاده از bigram تفاوت معنی مشخص نمیشود و باید از n-gram های طولانی ترین استفاده کرد.

1. شیر و گوشت برای بدن مفید هستند.
2. شیر گوشت خوار است و انسان را میخورد.

Addmitive Smoothing

پاسخ سوال 4

در بررسی یک دیتای جدید، اگر دیتا حاوی واژه ای باشد که در هیچ کدام از sample های بررسی شده‌ی یک کتگوری خاص نبوده باشد، احتمال وجود آن واژه در آن کتگوری 0 میشود، در نتیجه، likelihood برای آن کتگوری برابر 0 میشود. و در این صورت احتمال تعلق آن sample به آن کتگوری صفر می‌شود.

پاسخ سوال 5

برای حل این مشکل از روش Addictive Smoothing استفاده میکنیم و در نتیجه با دیدن یک واژه کاملا جدید برای یک کتگوری، دیگر احتمال وجود آن واژه در آن کتگوری دقیقا برابر 0 در نظر گرفته نمیشود بلکه کاری میکنیم که آن واژه جدید با احتمال کمی (نه 0) متعلق به آن کتگوری باشد. بنابراین احتمال تعلق آن واژه به آن کتگوری برابر 0 نیست اما این احتمال، از واژه های خود آن کتگوری کمتر است. برای این کار از فرمول زیر استفاده میکنیم:

In order to avoid the problem of *zero* probabilities, an additional smoothing term can be added to the *multinomial Bayes* model. The most common variants of additive smoothing are the so-called *Lidstone smoothing* ($\alpha < 1$) and *Laplace smoothing* ($\alpha = 1$).

$$\hat{P}(x_i \mid \omega_j) = \frac{N_{x_i,\omega_j} + \alpha}{N_{\omega_j} + \alpha\, d} \quad (i = (1, \ldots, d))$$

where

- N_{x_i,ω_j}: Number of times feature x_i appears in samples from class ω_j.
- N_{ω_j}: Total count of all features in class ω_j.
- α: Parameter for additive smoothing.
- d: Dimensionality of the feature vector $\mathbf{x} = [x_1, \ldots, x_d]$.

پاسخ سوال 6

نمودارها در فایل ژوپیتر نشان داده شده‌اند.

میتوان پس از به دست آوردن bag of words ها، واژه هایی مثل کتاب، جلد، نسخه و ... را از دایره واژگانِ تست و ترین حذف کرد تا دقت مدل بالاتر برود.

ارزیابی مدل

در این مرحله درصدد ارزیابی مدل هستیم. داده ای که برای ارزیابی مدل کنار گذاشته شده بود، به مدل داده می شود و پیش بینی کتگوری برای هر ردیف از اطلاعات کتابها صورت می گیرد. کتگوری پیش بینی شده با کتگوری واقعی برای هر ردیف مقایسه می شود. با تعیین تعداد پیش بینی ها و تعداد حقیقی در هر دسته و مقایسه آن ها و استفاده از پارامتر های ارزیابی می توانیم مدل خود را در نهایت ارزیابی نماییم. این پارامتر های ارزیابی شامل موارد زیر هستند:

- Accuracy

نشان دهنده دقت کلی مدل می باشد. مقدار آن نسبتی بین تعداد کل ردیف های داده ارزیابی و تعداد ردیف هایی که درست پیش بینی شده اند می باشد.

- Recall

این پارامتر برای هر دسته بطور جدا محاسبه می گردد. مقدار آن برای هر دسته برابر با نسبت بین تعداد کل ردیف هایی که مربوط به یک دسته هستند و ردیف هایی که بطور صحیح مربوط به این دسته تشخیص داده شده است.

- Precision

این پارامتر نیز بصورت جدا برای هر دسته محاسبه می گردد. مقدار آن برابر با نسبت بین تعداد کل ردیف هایی که مربوط به یک دسته هستند و تعداد ردیف ها که برای این دسته تشخیص داده شده است (شامل پیش بینی های صحیح و غلط) می باشد.

- F1

معیار F1 که در واقع ترکیب متعادلی بین معیارهای دقت و صحت است. در مثالی که در پاسخ سوال 7 آورده ام، مشخص است که F1، معیار قابل اطمینان تری است نسبت به 3 معیار ارزیابی که دیگر و در نتیجه مقدار F1 یک مدل بالاتر باشد، مدل عملکرد بهتری دارد.

همچنین بجز پارامتر های مذکور از ماتریس اغتشاش (Confusion Matrix) نیز برای ارزیابی مدل می توان استفاده نمود. ستون ها و ردیف های این ماتریس دسته های موجود در مسئله می باشند. هر درایه این ماتریس نشان دهنده تعداد مواردی است که دارای دسته حقیقی ستون آن درایه بوده اند که دسته پیش بینی آن ها برابر با ردیف درایه بوده است.

برای رفع مشکل فاصله زیاد بین مقادیر Recall و Precision در دسته های مختلف از تکنیک Oversampling بهره گرفته شده است. با استفاده از این تکنیک، دسته هایی که حجم داده آموزش آن ها از حجم داده آموزش دسته های دیگر کمتر بوده است، انتخاب شده و با انتخاب رندوم از میان ردیف های آن ها و افزودن این ردیف ها به داده آموزش دسته نامبرده، حجم آن هم اندازه با داده دسته های دگیر می گردد. این تکنیک تاثیر بسازایی در یکنواختی مقادیر Recall و Precision برای دسته های مختلف داشت.

مثال ما سیستمی است که میخواهد با توجه به ویژگیها تشخیص دهد تومور خوشخیم یا بدخیم است.

Accuracy

نسبت تعداد مثال هایی که درست کلاس بندی شده اند به تعداد کل مثال ها را میگویند؛ در مثال ما درصد درستی پیش بینی های سیستم برابر با Accuracyاست.

Precision

نسبت تعداد مثال هایی که بدخیم بوده اند و سیستم بدخیم تشخیص داده (True Positive) به تمامی مثال هایی که سیستم بدخیم تشخیص داده است؛ یعنی چه نسبتی از تومورهای بدخیمی که تشخیص داده شده، تومور بدخیم هستند.

Recall

درصد تشخیص تومورهای بدخیم را می گویند؛ یعنی چه نسبتی از تومورهای بدخیم را سیستم درست تشخیص داده است.

فرض کنید که سیستم ما همه ی تومورها را بدخیم تشخیص میدهد. آنگاه Recallبالایی خواهیم داشت؛ در صورتی که می دانیم سیستم مان خوب نیست. مثال دیگر میتواند این باشد که سیستم ما تنها در صورتی که خیلی مطمئن است که تومور بدخیم است، بدخیم تشخیص دهد. آنگاه Precisionبالایی خواهیم داشت؛ اما تعداد زیادی از بیماران توسط اشتباه سیستم ما دچار مشکل شده اند. برای Accuracyنیز میتوان این مثال را عنوان کرد که فرض کنید ۹۹درصد بیماران تومور خوش خیم داشته باشند. در این صورت سیستمی که همه ی تومور ها را خوش خیم تشخیص میدهد، سیستم خوبی نیست اما Accuracyبالایی دارد. به همین خاطر است که از ترکیب موارد بالا استفاده می شود.

اگر هر دو عدد های بزرگی باشند، با Precisionمی توان فهمید که سیستم ما با گفتن کلاس مثبت، به احتمال زیادی درست گفته است و با Recallبالا می توان به این نتیجه رسید که سیستم ما کلاس منفی های کمی را مثبت پیش بینی کرده است. پس در نتیجه ی این دو، سیستم ما به خوبی کلاس بندی را انجام داده است. اگر هر دوی آن ها کم باشند نیز به همین ترتیب نشان دهنده ی بد بودن کلاس بندی سیستم است. اما فرض کنید Precisionبالا و Recallپایین داشته باشیم. سیستم ما در این صورت کلاس مثبت را به خوبی کلاس بندی کرده و تعداد زیادی از کلاس منفی را نیز مثبت پیش بینی کرده است. خلاف این حالت نیزPrecision پایین و Recallبالاست که معادل این است که کلاس منفی با درصد خوبی درست پیش بینی شده و درصد زیادی از کلاس مثبت نیز منفی پیش بینی شده است

مثال دیگر برای حالتی که که مدل یادگیری ماشین دارای مقدار بالای precision باشد اما خوب کار نمی کند.
در محاسبه مقدار precision، تمام پیش بینی های صحیح و غلط یکسان در نظر گرفته می شوند. در مدل های یادگیری ماشین که پیش بینی غلط مدل باعث ضرر به سیستم می شود، درنظرگرفتن تنها این پارامتر برای بررسی ارزیابی مدل اشتباه است. برای مثال مدل یادگیری ماشینی را در نظربگیرید که وظیفه آن تشخیص موشک های کروز و هواپیماهای مسافربری از یکدگیر است. فرض کنید از میان 10 شی پرنده موجود در هوا، 9 مورد آن موشک باشد. اگر مدل 5 مورد آن را از میان 9 موردی که موشک هستند، موشک تشخیص دهد دارای precision برابر با 100 درصد است اما مدل خوب کار نمی کند چرا که درصد بالایی از موشک ها را نتوانسته پیش بینی کند.

پاسخ سوال 9

- حالت میانگین گیری macro: یک حالت میانگین گیری ساده است.

در اینجا macroF1 خواسته شده بود پس بین همه‌ی 6 تا مقدار بدست امده برای F1 هر 6 کلاس، یک میانگین گیری ساده انجام شد تا macroF1 بدست آید.

- حالت میانگین گیری Weighted: به هر کدام از مقادیر، بر حسب تعداد آن پارامتر، یک وزن اختصاص میدهیم.

برای مثال در همین مساله اگر میخواستیم از weighted F1 استفاده کنیم، باید تعداد هرکدام از کتگوری ها را حساب میکردیم و F1 آن کتگوری را در تعدادش ضرب میکردیم. سپس حاصل جمع را بر تعداد همه‌ی sample ها تقسیم میکردیم.

- در روش microF1: در این روش تمام داده ها را با هم در نظر میگیریم

یعنی دیگر true positive و false negative را جدا نمیکنیم در نتیجه به این میرسیم که

micro-F1 = micro-precision = micro-recall = accuracy

پاسخ سوال 10

a:

نتایج با درنظرگرفتن Additive Smoothing

[61, 2, 1, 1, 0, 10]
[0, 29, 1, 36, 3, 6]
[13, 0, 58, 2, 2, 0]
[2, 4, 1, 62, 2, 4]
[1, 1, 2, 10, 60, 1]
[2, 0, 0, 0, 0, 73]

Accuracy: %76.22

داستان کوتاه
Precision: %80.56
Recall : %38.67
F1: %52.25

کلیات اسلام
Precision: %92.06
Recall : %77.33
F1: %84.06

رمان
Precision: %55.86
Recall : %82.67
F1: %66.67

داستان کودک و نوجوانان
Precision: %89.55
Recall : %80.00
F1: %84.51

جامعه‌شناسی
Precision: %77.22
Recall : %81.33
F1: %79.22

مدیریت و کسب و کار
Precision: %77.66
Recall : %97.33
F1: %86.39

.macroF1: %75.52

b:

نتایج بدون درنظرگرفتن Additive Smoothing

[23, 47, 2, 0, 2, 1]
[3, 60, 3, 6, 2, 1]
[3, 41, 29, 2, 0, 0]
[0, 58, 4, 10, 3, 0]
[1, 39, 1, 1, 33, 0]
[3, 45, 0, 0, 0, 27]

Accuracy: %40.44

داستان کوتاه
Precision: %20.69
Recall : %80.00
F1: %32.88

کلیات اسلام
Precision: %74.36
Recall : %38.67
F1: %50.88

رمان
Precision: %52.63
Recall : %13.33
F1: %21.28

داستان کودک و نوجوانان
Precision: %82.50
Recall : %44.00
F1: %57.39

جامعه‌شناسی
Precision: %69.70
Recall : %30.67
F1: %42.59

مدیریت و کسب و کار
Precision: %93.10
Recall : %36.00
F1: %51.92

.macroF1: %42.82

پاسخ سوال 11

دقت در حالت بدون در نظر گرفتن additive smoothing بسیار کمتر است چرا که در این مساله‌ی پردازش زبان، هرچقدر دیتای train ما وسیع باشد، اما باز هم نمیتوان تمام واژگانی را که ممکن است در یک sample جدید در آن دسته قرار بگیرد را شامل شود. یه این علت، بدون درنظر گرفتن additive smoothin احتمال ها به اشتباه برابر با 0 میشوند و دقت مدل بسیار پایین می‌آید.

پاسخ سوال 12

1. مورد شماره "162"، "داستان کوتاه" است اما مدل آن را "رمان" پیش بینی کرده است.
2. مورد شماره "193"، "داستان کودک و نوجوان" است اما مدل آن را "رمان" پیش بینی کرده است.
3. مورد شماره "245"، "داستان کوتاه" است اما مدل آن را "رمان" پیش بینی کرده است.
4. مورد شماره "243"، "داستان کوتاه" است اما مدل آن را "رمان" پیش بینی کرده است.
5. مورد شماره "269"، "کلیات اسلام" است اما مدل آن را "جامعه شناسی" پیش بینی کرده است.

طبق cell اخر ژوپیتر، اشتباه ها اکثرا یا بین "داستان کوتاه" و "داستان کودک و نوجوان" و "رمان" اتفاق افتاده است یا بین "کلیات اسلام" و "جامعه شناسی".

دلیل این اشتباه هم نزدیکی دایره واژگان این دسته ها به یکدیگر است.

با عملیات های مختلف (n-gram، حذف بیشتر واژه های تکراری...) میتوان دقت را بالاتر برد اما لازم است نکته ای را در نظر گرفت؛

چیزی که ما داریم دیتای train است. ما با استفاده از train میخواهیم به اطلاعاتی برسیم که با این اطلاعات داده های test را طبقه بندی کنیم.

dicision boundry را نمیتوان از یک حدی دقیقتر انتخاب کرد چرا که اگر داده‌های train را با دقت زیادی مدل کنیم، تمامی نویز های train را هم به مدل آموزش داده ایم و در نتیجه overfitting اتفاق می افتد و دقت کلسیفیکیشن داده های تست ما پایین می‌آید.