

به نام خدا



درس هوش مصنوعی و سیستم‌های خبره

پروژه درسی

مدرس درس:
جناب آقای دکتر محمدی

طراحان:
سینا زمانی
بکتاش انصاری

مقدمه:

فرض کنید اپلیکیشن توییتر را باز کرده و در حال اسکرول کردن در تایم لاین خود هستید. پس از خواندن هر توییت احتمالاً حس خاصی خواهید داشت. به طور مثال ممکن است از خواندن داستان موفقیت یک فرد حس مثبتی بگیرید یا از مطلع شدن در مورد تجربه تلخ فردی دیگر حس منفی بگیرید. یا حتی حس خاصی نگیرید!

اما داستان به اینجا ختم نمی‌شود. احتمالاً تجربه ثبت نظر در مورد یک محصول که از فروشگاه آنلاین خریداری کرده‌اید را داشته‌اید. یکی از مهم‌ترین دغدغه‌های هر کسب و کاری این است که بتواند نظرات مشتریان را درک کرده و به درستی از آن‌ها در جهت بهبود خدمت‌رسانی استفاده کند. پس تشخیص حس و حال نظر شما برای آن فروشگاه آنلاین مقوله‌ای حیاتی‌ست.

احتمالاً با این توضیحات، تا این لحظه عبارات مثبت و منفی زیادی در ذهن شما نقش بسته است. مغز انسان در دوران کودکی یاد می‌گیرد تا ادراک خود از محرک‌ها را دسته‌بندی کند. این کار وابسته به عوامل پیچیده‌ی شناختی و علوم اعصاب است که در انسان تکامل یافته. حال فرض کنید ما می‌خواهیم این روند را با استفاده از یک سیستم کامپیوتری به صورت خودکار انجام دهیم تا بتوانیم به فروشگاه آنلاین کمک کنیم مشتریان خود و بازخوردشان به محصولاتش را بهتر بشناسد.

بدین منظور می‌توانیم از مدل‌های هوش مصنوعی بهره ببریم و قابلیت‌های شناختی انسان را شبیه‌سازی کنیم. روش‌های متفاوتی برای این کار وجود دارد که رویکرد ما در اینجا استفاده از یک مدل آماری برای طبقه‌بندی داده‌ها است. در ادامه بخش‌های مختلف این پروژه را شرح می‌دهیم.

۱ مدل طبقه‌بندی:

می‌دانیم که دو دسته کلی از مدل‌ها وجود دارند. اگر در حین آموزش مدل از داده‌های برچسب‌گذاری شده استفاده کنیم، با یادگیری تحت نظارت سروکار داریم. در این مسئله ما تعداد زیادی توییت جمع‌آوری کرده‌ایم و به آن‌ها برچسب مثبت، منفی یا خنثی نسبت داده‌ایم. از شما می‌خواهیم با طراحی و آموزش یک مدل Naive Bayes که بر اساس قانون بیز در مورد احتمالات شرطی کار می‌کند، سعی کنید توییت‌های تست را برچسب‌گذاری کنید. در حوزه دسته‌بندی داده‌های متنی با مدل‌های آماری، باید از پارامتری به نام میزان تکرار کلمات استفاده کنیم. به طور ساده به این معنی است که اگر کلمه **خوش‌حال** در توییت‌های مثبت بیش‌تر از توییت‌های منفی به کار رفته پس این کلمه یک کلمه مثبت قلمداد می‌شود و اگر در یک توییت جدید بدون برچسب این کلمه را ببینیم وزن مثبت بودن آن توییت بالا می‌رود. در ادامه مراحل آموزش یک مدل برای طبقه‌بندی دو دسته‌ای را شرح می‌دهیم.

آ از جایی که هر توییت ممکن است شامل توکن‌های اضافی باشد که دقت مدل ما را پایین بیاورد پس باید در قدم اول داده‌های خود را تمیز کنیم. به طور مثال توییت زیر را با یک سری از قواعد مرسوم تمیز کرده‌ایم. شما هم سعی کنید با جست‌وجو در اینترنت تعدادی روش بهینه برای این کار پیدا کرده و روی داده‌های خود اعمال کنید.

I am happy because I am learning #NLP → ['happy', 'because', 'learn', 'nlp']

ب جدولی از تعداد دفعات تکرار هر کلمه در توییت‌های مثبت و منفی تشکیل دهید. با این کار می‌خواهیم به کلمات وزن‌دهی کنیم تا میزان مثبت یا منفی بودن آن‌ها مشخص شود.

Positive tweets
[happy, because, learn, NLP]
[happy, not, sad]
Negative tweets
[sad, not, learn, NLP]
[sad, not, happy]

Step 2:
Word
count

word	Pos	Neg
happy	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

ج حال باید با استفاده از مقادیر ذخیره شده در جدول مرحله قبل، احتمال وقوع هر کلمه در هر کلاس را محاسبه کنیم. فرمول آن در عکس زیر مشخص شده است. در این فرمول مقادیری به صورت و مخرج اضافه شده تا از احتمال صفر برای کلماتی که در یک کلاس مشاهده نشده اند پیشگیری شود. به این روش Laplacian Smoothing می گویند.

Training Naïve Bayes

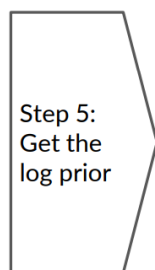
freq(w, class)			$\lambda(w) = \log \frac{P(w pos)}{P(w neg)}$			
word	Pos	Neg				
happi	2	1	<div>Step 3:</div> <div>$P(w class)$</div> <div>$V_{class} = 6$</div> <div>$\frac{\text{freq}(w, class) + 1}{N_{class} + V_{class}}$</div>			
because	1	0				
learn	1	1				
NLP	1	1				
sad	1	2				
not	1	2				
N_{class}	7	7				

word	Pos	Neg	λ
happy	0.23	0.15	0.43
because	0.15	0.07	0.6
learning	0.08	0.08	0
NLP	0.08	0.08	0
sad	0.08	0.17	-0.75
not	0.08	0.17	-0.75

د متغیر لامبدا را به عنوان پارامتر تصمیم گیری محاسبه می کنیم. در عکس بالا می بینیم که اگر این مقدار مثبت شود یعنی کلمه مورد نظر مثبت است و اگر منفی شود یعنی کلمه منفی است.

ه یک عامل مهم که باید در محاسبات خود در نظر بگیریم تعداد توییت های مثبت و منفی است. ممکن است تعداد توییت های این دو کلاس با هم برابر نباشد که در این صورت یک پارامتر پیشینی مطابق تصویر زیر به محاسبات خود اضافه می کنیم.

Training Naïve Bayes



D_{pos} = Number of positive tweets
 D_{neg} = Number of negative tweets

$$\text{logprior} = \log \frac{D_{pos}}{D_{neg}}$$

If dataset is balanced, $D_{pos} = D_{neg}$ and $\text{logprior} = 0$.

تا به اینجا با مراحل آموزش یک مدل طبقه بندی آشنا شدید. مراحل بعدی به منظور تست کردن این مدل و استفاده از آن برای برچسب گذاری توییت های جدید مورد استفاده قرار می گیرد.

آ ابتدا هر توییت را مشابه مرحله قبل توکنایز کرده و تمیز می‌کنیم.

ب حال برای هر توییت مجموع لامبدهای کلمات آن و پارامتر لگاریتمی پیشینی را محاسبه می‌کنیم.

ج در مرحله تصمیم‌گیری، باید امتیاز محاسبه شده را برای کلاس‌های مختلف بازه‌بندی کنیم. به این صورت که اگر امتیاز مثبت شد یعنی توییت مثبت بوده و اگر منفی شد یعنی توییت منفی بوده.

Predict using Naïve Bayes

- log-likelihood dictionary $\lambda(w) = \log \frac{P(w|pos)}{P(w|neg)}$

$$\logprior = \log \frac{D_{pos}}{D_{neg}} = 0$$

- Tweet: [I, pass, the, NLP, interview] 🍌

$$score = -0.01 + 0.5 - 0.01 + 0 + \logprior = 0.48$$

$$pred = score > 0$$

word	λ
I	-0.01
the	-0.01
happi	0.63
because	0.01
pass	0.5
NLP	0
sad	-0.75
not	-0.75

در مرحله آموزش از داده‌های train و در مرحله تست از داده‌های test استفاده می‌کنیم. در مرحله تست باید خروجی مدلی که طراحی کرده‌ایم را با لیبل‌های داده‌شده مقایسه کنیم تا دقت مدل به دست آید. به بیان دیگر در این مرحله ما برچسب‌ها را به مدل نمی‌دهیم. صرفاً از مدل برای پیش‌بینی برچسب‌ها استفاده کرده و آن را با برچسب درست مقایسه می‌کنیم. تا اینجا با یک مدل طبقه‌بندی دوکلاسه آشنا شدید. از شما می‌خواهیم مدل مشابهی را برای طبقه‌بندی سه‌کلاسه به صورت مثبت، منفی و خنثی پیاده‌سازی کنید.

۲ مراحل پروژه:

۱.۲ مرحله اول

- در این مرحله ابتدا به شما مجموعه داده‌های train در قالب یک فایل csv داده می‌شود و شما باید مدل پیاده‌سازی شده خودتان را بر روی این مجموعه داده آموزش بدهید.
- مدل شما باید در قالب یک کلاس با مجموعه‌ای از متدها باشد که خروجی و ورودی هر کدام از متدها در فایل template.py مشخص شده‌اند.
- پس از پیاده‌سازی، با استفاده از توابع موجود در فایل run.py مدل خود را آموزش دهید.
- فرآیند پیاده‌سازی مدل سه کلاس باید در فایل گزارش شما توضیح داده شود و دقت مدل خود را گزارش کنید.
- در فایل گزارش مدت زمان فرآیند آموزش باید ذکر شود.
- پس از آموزش مدل بر روی داده‌های train مدل خود را بر روی داده eval ارزیابی کنید و دقت مدل را گزارش دهید.
- سعی کنید مدل خود و پیش‌پردازش داده‌های آموزش را با توجه به دقت مدل بهبود ببخشید.
- در نهایت با استفاده از مدل خود برای داده‌های test فرآیند لیبیل‌گذاری را انجام دهید و نتیجه را در قالب یک فایل txt ارائه دهید.
- فایل txt خود را با نام result.txt ذخیره کنید و فرمت آن باید به این شکل باشد که در خط i لیبیل مربوط به دیتای i قرار گرفته باشد.

۳ قوانین:

۱. این تمرین را در گروه‌های دو نفره انجام دهید.
۲. استفاده از کتابخانه‌ها برای مدل‌های آماده مجاز نمی‌باشد.
۳. کدها و گزارش خود را در قالب یک فایل زیپ با فرمت StudentNumber.zip ارسال کنید.
۴. در صورت مشاهده شباهت غیرعادی بین پیاده‌سازی‌ها نمره طرفین طبق قوانین درس محاسبه خواهد شد.