استاد: رامتین خسروی طراحان: امیرمحمد رنجبرپازکی، ژیوار حسن‌زاده، محسن فیاض، آرمان رستمی

## ۱ مقدمه

در این فاز از پروژه شما قرار است با استفاده از الگوریتم naive bayes مدلی را طراحی کنید تا بتواند تشخیص دهد comment ورودی حاوی مفاهیم بد است یا خیر. به بیان بهتر، اگر بخواهیم آن را طبقه‌بندی کنیم comment خوب تلقی می‌شود یا comment بد. علاوه بر ساخت مدل شما باید با استفاده از معیارهایی که در ادامه بیان می‌شوند، دقت مدل خود را بدست‌آورید. برای طراحی این مدل از bayes theorem و همینطور کمی از مفاهیم احتمالاتی استفاده خواهیم کرد. کاری که شما در این پروژه انجام خواهید داد شامل چندین مرحله است:

۱. خواندن و استفاده از اطلاعات به دست آمده از comment ها.

۲. ساخت مدل با استفاده از اطلاعات داده‌شده از متن comment ها که چگونگی آن را در ادامه شرح خواهیم‌داد.

۳. ارزیابی مدل بدست‌آمده برای اطمینان از صحت عملکرد مدل، به بیان بهتر بدست‌آوردن متر و معیاری برای تشخیص اینکه مدل شما در کجا خوب و در کجا بد عمل کرده‌است.

۴. تشخیص خوب یا بد بودن comment ورودی جهت تصمیم آن که اعمال بشود یا خیر.(در این مرحله شما باید به جز تشخیص آن که comment ورودی خوب است یا بد، در صورت خوب بودن، comment را به comment های قبلی کالای مربوطه اضافه کنید و یا در صورت بد بودن، آن را درنظر نگیرید)

## ۲ معرفی الگوریتم

قبل از معرفی الگوریتم ،لازم است تا بعضی از اصطلاحاتی که در ادامه برای توضیح الگوریتم استفاده می‌شوند، توضیح داده‌شود:

- ویژگی : در مباحث هوش مصنوعی و یادگیری ماشین،ویژگی ، یک مشخصه قابل اندازه‌گیری یا به طور کلی تر هرگونه مشخصه مربوط به یک پدیده است. به عنوان مثال اگر داده ورودی، تعدادی ماشین باشند، ویژگی های آن ها می‌تواند سال ساخت، رنگ، تعداد سیلندر و یا هر مشخصه دیگری باشد که مربوط به این ماشین ها است.
- خروجی (Label) : کلاس نهایی و به بیان بهتر خروجی یک مدل، ویژگی یا مشخصه‌ای است که برای شناختن بهتر و جامع‌تر آن استفاده می‌شود. به بیان بهتر اگر بخواهید مدلی با استفاده از ویژگی های مختلف بسازید که بتواند طبقه‌بندی خاصی را انجام دهد، تنها با استفاده از ویژگی گفته شده (Label) می‌توانید با دقت ۱۰۰ درصد طبقه‌بندی را انجام دهید.
- مستقل بودن ویژگی‌ها : به این معنا است که وجود و یا احتمال وجود ویژگی های مختلف تاثیری روی وجود و احتمال وجود ویژگی‌های دیگر نگذارد. به بیان کاربردی تر می‌توان گفت اگر شما بخواهید احتمال وجود دو ویژگی با هم را بررسی کنید و مقدار آن را بدست‌آورید می‌توانید این کار را با ضرب احتمال دو ویژگی به تنهایی(با استقلال)در نظر بگیرید و مقدار آن را محاسبه کنید.

naive bayes یک تکنیک طبقه بندی است که بر پایه bayes theorem انجام می‌گیرد و با این فرض عمل می‌کند که ویژگی‌های مختلف یک داده با فرض دانستن آن که چه خروجی‌ای(Label) دارد از هم مستقل هستند. به بیان بهتر فرض می‌کنیم ویژگی‌های یک داده که در این پروژه کلمات تشکیل دهنده comment ها هستند، از هم مستقل هستند و برای محاسبه احتمال کلی ویژگی‌ها همان‌طور که پیش‌تر گفته‌شد، می‌توانید از ضرب ویژگی‌ها استفاده کنید. به عنوان مثال اگر تکه شعری داشته باشیم و بدانیم که شاعر آن سعدی است یا حافظ، احتمال وجود کلمات مختلف که همان ویژگی‌های ما در این مسئله هستند، در این تکه شعر از هم مستقل خواهندبود.

naive bayes روشی ساده برای مدل کردن یک طبقه‌بند خوب با اطلاعات ورودی با حجم زیاد است و در عین سادگی می‌تواند مدل‌های بسیار پیچیده را نیز تولید کند.

با استفاده از آن می‌توان احتمال خروجی (Label) به شرط دیدن ویژگی را محاسبه کرد که در آن از احتمال دیدن ویژگی به شرط دانستن خروجی (Label) استفاده می‌شود که در ادامه توضیحات نحوه عمل‌کردن آن آمده‌است.

$$P(c|X) = \frac{P(X|c)P(c)}{P(X)}$$

$$P(X|c) = P(x_1|c) \times P(x_2|c) \times ... \times P(x_n|c)$$

- P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

$P(c|X)$ همان احتمال دانستن کلاس خروجی به شرط دانستن ویژگی‌های موجود می‌باشد که در مسئله ما به شکل احتمال خوب یا بد بودن comment به شرط دانستن کلمات آن می‌شود.

$P(c)$ احتمال کلاس خروجی است که در مسئله ما همان احتمال خوب یا بد بودن comment است.

$P(x|c)$ احتمال داشتن ویژگی x به شرط دانستن کلاس خروجی است که در مسئله ما، همان احتمال وجود داشتن کلمه x در comment به شرط دانستن اینکه comment خوب یا بد است،می باشد.

$P(x)$ احتمال وجود داشتن ویژگی x به کلی است که در مسئله ما همان احتمال وجود داشتن کلمه‌ای خاص داخل متن com-ment است که با توجه به اینکه برای هم comment خوب و بد احتمال یکسانی دارد پس می‌توانید آن را درنظر نگیرید.

## ۳ توضیح مسئله و اطلاعات ورودی

همان طور که قبل‌تر گفته‌شد، شما برای ساختن مدل خود به احتمالات ویژگی به شرط دانستن کلاس خروجی نیاز دارید که در فایلی به نام train.csv در اختیار شما قرار می‌گیرد.با توجه به اینکه مدل گفته‌شده با این احتمالات کار می‌کند، می‌توانید درنظر داشته باشید که مدل به شما داده شده است و شما باید از آن استفاده کنید. پس تنها کاری که شما باید انجام دهید این است که با گرفتن هر comment ویژگی ها یا همان کلمات آن را درنظر بگیرید. پس از آن احتمال داشتن کلاس خروجی خوب یا بد به شرط دیدن ویژگی‌ها(کلمات) را برای آن comment محاسبه کنید و با توجه به آن که کدام بیشتر باشد، کلاس خروجی را تعیین می‌کنید. محاسبه این احتمال برای هر کدام از کلاس‌های خروجی نیز از ضرب احتمال کلاس خروجی P(c) و احتمال داشتن ویژگی‌ها(کلمات) به شرط دانستن کلاس خروجی بدست می‌آید. P(X | c) برای محاسبه احتمال داشتن ویژگی‌ها به شرط دانستن کلاس خروجی نیز با توجه به مباحثی که گفته‌شد، از ضرب احتمال تمامی ویژگی ها به شرط دانستن کلاس خروجی استفاده می‌کنیم که این این احتمالات به شما داده شده‌است.

نکته : توجه داشته باشید در انجام این محاسبات به دلیل آن که ممکن است کلمه‌ای در اطلاعاتی که شما داشته‌باشید نباشد و در نتیجه ضرب احتمالاتی که انجام می‌دهید خروجی صفر داشته‌باشد،پس بهتر است تا به جای ضرب این احتمالات از جمع

لگاریتم این احتمالات استفاده کنید و در انتها خروجی آن ها را مورد بررسی قرار دهید. همان طور که پیش‌تر توضیح داده شد شما قرار است با احتمالات شرطی کار کنید، به این شکل که احتمالات ویژگی ها را به شرط دانستن کلاس خروجی در نظر بگیرید.
فایل train.csv که در اختیار شما قرار داده شده است، حاوی ۵۰۰۰ نمونه است و شامل احتمالات ویژگی‌ها(کلمات) به شرط دانستن کلاس خروجی آن‌ها(Label) می‌باشد. به بیان بهتر در هر خط از فایلی که به شما داده می‌شود سه ستون وجود دارد. ستون اول یک کلمه است که شما از آن به عوان ویژگی استفاده می‌کنید و پیش‌تر درباره آن توضیح داده‌شده. در ستون دوم احتمال وجود این کلمه در comment به شرطی است که comment، کلاس خروجی خوب داشته باشد. و ستون سوم نیز احتمال وجود این کلمه در comment به شرطی است که comment، کلاس خروجی بد داشته باشد. دقت داشته باشید این احتمالات همان $P(x|c)$ هستند.
فایل دیگری به نام class probeblities به شما داده‌شده است که شامل یک خط است که در تنها ستون آن احتمال کلی خوب بودن comment به شما داده شده‌است.
به عنوان راهنمایی بهتر است این اطلاعات را در قالب unordered map ذخیره سازی کنید تا در کار کردن با این اطلاعات راحت‌تر باشید.

در ادامه با ارائه یک مثال، می‌توانید طرز کار این الگوریتم را بهتر درک کنید.

## ۴ مثال

برای درک بهتر الگوریتم و شیوه کار کردن آن به مثال زیر توجه کنید

دسته‌ای از ورودی‌ها به شما داده شده است که حاوی یک شعر و همینطور شاعر آن شعر است. خواسته مسئله نیز مدلی است که بتواند با دیدن یک شعر جدید شاعر آن را با احتمال خوبی تشخیص دهد. ویژگی‌های زیادی را می‌توان در این مسئله درنظر گرفت که برای مثال و نزدیکی بیشتر به مسئله خودمان ،می‌توان کلمات موجود در هر شعر را انتخاب کرد. برای تمرین دادن و یا همان ساختن مدل خود نیز به این شکل عمل می‌کنیم که احتمال وجود داشتن هر کلمه در شعرهای یک شاعر را درنظر می‌گیریم. همینطور احتمال اینکه هر شعر برای چه شاعری باشد را نیز درنظر می‌گیریم. حال برای به دست‌آوردن خروجی برای یک شعر جدید به این شکل عمل می‌کنیم که برای هر شاعر احتمال وجود کلمات آن شعر با فرض دانستن اینکه شاعر آن کیست را از هم مستقل گرفته و در هم ضرب می‌کنیم، همین طور احتمال کلی آن که شعر برای چه شاعری باشد را نیز در نتیجه ضرب می‌کنیم. در انتها عدد حاصل برای هر کدام از شاعرها که بیشتر به دست آمد، شعر ورودی را به آن نسبت‌می‌دهیم.

## ۵ معیار ارزیابی

برای اطمینان از صحت عملکرد مدل خود باید آن را ارزیابی کنید و میزان دقت آن را با توجه به معیار های مختلف بررسی کنید. برای این کار فایل test.csv به شما داده شده است که شامل دو ستون است. ستون اول شامل متن comment هایی است که به شما داده شده.و ستون دوم حاوی کلاس خروجی آن comment ها است.در زمانی که دستور زیر در ترمینال وارد میشود شما باید مدل خود را evaluate یا ارزیابی کنید که شرح چگونگی این ارزیابی در ادامه داده شده است. شما باید با مدل خود کلاس خروجی این comment ها را محاسبه کرده و با کلاس خروجی‌ای که در فایل test.csv وجود دارد مقایسه کنید و با توجه به مقایسه این خروجی ها خطاهای زیر را محاسبه کرده و در فایل output.txt وارد کنید. فرمت خروجی شما نیز باید به این شکل باشد که به ترتیب خطاهای زیر در هر خط،در ابتدا نام خطا و پس از آن درصد خطا را جلوی آن گزارش دهید. به عنوان مثال :

POST evaluateModel

Recall: 70
Precision: 72
Accuracy: 75

$$Recall = \frac{\text{Correct Detected Appropriate Comments}}{\text{All Appropriate Comments}}$$

$$Precision = \frac{\text{Correct Detected Appropriate Comments}}{\text{Detected Appropriate Commments (This also includes wrong detections)}}$$

$$Accuracy = \frac{\text{Correct Detected}}{\text{All Comments}}$$

Correct Detected Appropriate Comments شامل comment هایی می‌شود که شما به درستی آن‌ها را comment خوب تشخیص داده‌اید.
All Appropriate Comments شامل همه comment های خوبی که به شما داده شده‌است می‌شود.
Detected Appropriate Commments شامل همه comment هایی است که توسط مدل شما، comment خوب تشخیص داده شده‌است.
Correct Detected شامل همه تشخیص های درست مدل شما چه comment خوب و چه comment بد می‌شود.

دقت داشته باشید برای accuracy دقت بالای ۶۵ درصد مطلوب است و همینطوری برای precision و recall دقت بالای ۶۰ درصد مطلوب است

# ۶ نحوه‌ی تحویل و نکات پایانی

- پرونده‌های برنامه‌ی خود را با نام A7-2-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال، اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۷۹۹۹ باشد، نام پرونده‌ی شما باید A7-2-810197999.zip باشد.
- این پروژه حتما باید به روش شی‌گرایی و به صورت Multi File پیاده‌سازی شود. همچنین استفاده از makefile اجباری است.
- برنامه‌ی شما باید در سیستم‌عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود. **دقت کنید** که باید در multifile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.