در این بخش، هدف ما شناسایی ایمیل‌های اسپم از ایمیل‌های عادی است. مجموعه داده‌ای که استفاده می‌کنیم، شامل دو دسته از ایمیل‌ها است که با ۰ (غیر اسپم) و ۱ (اسپم) مشخص شده‌اند. شما باید از قانون بیز برای تعیین اینکه آیا یک ایمیل جدید اسپم است یا خیر، استفاده کنید. برای این کار از طبقه‌بند بیز ساده استفاده می‌کنیم.

در این مسئله یک فایل emails.csv در اختیار شما قرار داده شده است که شامل محتوای ایمیل‌ها و برچسب آن‌ها است. همان طور که گفته شد، ما باید به ازای هر داده (هر ایمیل) بردار ویژگی را بدست آوریم. یک راه ساده این است که هر کلمه موجود در دیکشنری، که شامل تمام کلمات است، را یک ویژگی در نظر بگیریم که هر ایمیل یا شامل آن کلمه هست یا نیست.

**گام اول: پیش‌پردازش داده‌ها (۱۰ نمره)**

در گام اول باید متن ایمیل‌های فایل را پیش‌پردازش کنید. برای این کار می توانید از کتابخانه nltk استفاده کنید یا خودتان موارد مورد نیازتان را پیاده‌سازی کنید. در این مرحله باید سعی کنید اطلاعات پیام‌ها را به نحوی مدیریت کنید که به بهترین حالت در پروژه استفاده کنید. به طور مثال، یکی از پیشنهادهای اولیه در این مرحله می‌تواند lowercase کردن و حذف علائم نگارشی و اعداد از هر پیام باشد؛ زیرا این علائم اطلاعات خاصی در مورد نوع پیام به ما نخواهند داد و قابل حذف هستند.

**گام دوم: تقسیم به داده آموزش و آزمایش (۵ نمره)**

این فرآیند به طور کلی شامل تقسیم داده‌های موجود به دو بخش جداگانه است:

مجموعه آموزش: این بخش از داده‌ها برای پیدا کردن $P(y)$ و $P(x_i|y)$ استفاده می‌شود.

مجموعه آزمایش: این بخش از داده‌ها برای ارزیابی عملکرد مدل استفاده می‌شود. مدل با استفاده از این داده‌ها آزمایش می‌شود تا ببیند چقدر خوب می‌تواند داده‌های جدید را پیش‌بینی کند.

هدف از تقسیم داده‌ها به این دو مجموعه، ارزیابی صحیح عملکرد مدل است. با استفاده از داده‌های آزمایش، می‌توان فهمید که مدل در پیش‌بینی داده‌های جدید چقدر دقیق و معتبر است.

برای اینکار می‌توانید از تابع `train_test_split` از کتابخانه scikit-learn استفاده کنید.

**گام سوم: ساخت مدل BoW (۲۰ نمره)**

همانطور که گفتیم برای سادگی در اینجا هر کلمه را یک ویژگی در نظر می‌گیریم. به این روش **B**ag **o**f **W**ords می‌گویند . همانطور که از نام این روش مشخص است، فرض می‌کنیم مجموعه‌ای از کلمات داریم که بدون توجه به دستور زبان کنار هم قرار گرفته‌اند. به عنوان مثال به دو جمله زیر دقت کنید:

- جمله‌ی اول: من از غذای این رستوران خوشم آمد.
- جمله‌ی دوم: غذای رستوران خیلی خوب بود ولی رفتار پرسنل خوب نبود.

| | من | از | غذای | این | رستوران | خوشم | آمد | خیلی | خوب | بود | ولی | رفتار | پرسنل | نبود |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **جمله ۱** | ۱ | ۱ | ۱ | ۱ | ۱ | ۱ | ۱ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| **جمله ۲** | ۰ | ۰ | ۱ | ۰ | ۱ | ۰ | ۰ | ۱ | ۲ | ۱ | ۱ | ۱ | ۱ | ۱ |

همانطور که در بالا مشاهده می‌شود یک $BoW$ تشکیل شد که نشان می‌دهد هر واژه در جمله وجود دارد یا خیر. اگر تعداد زیادی نمونه از این جملات متعلق به دسته‌بندی "اسپم" و "غیر اسپم" را داشته باشیم، می‌توانیم ماتریس $BoW$ را طوری تشکیل دهیم که بعداً بتوانیم از آن برای پیش‌بینی کلاس یا برچسب پیام‌های جدید استفاده کنیم.

در این پروژه $BoW$ بر اساس تعداد تکرار کلمات و بر اساس دسته‌بندی پیام مشخص می‌شود. یعنی در نهایت ابعاد ماتریس $BoW$ حاصل به صورت تعداد کلمات یکتا × ۲ خواهد بود، که تعداد تکرار هر کلمه در هر دسته را به صورت مجزا نشان می‌دهد.

با توجه به این توضیحات ماتریس $BoW$ را برای **مجموعه آموزش** بدست آورید و با استفاده از آن احتمال رخ دادن هر کلمه به شرط اسپم بودن و نبودن ($P(x_i|y)$) و احتمال رخداد ایمیل اسپم و غیر اسپم ($P(y)$) را بدست آورید.

### گام چهارم: پیش‌بینی با استفاده از قانون بیز (۲۰ نمره)

حال برای هر داده موجود در **مجموعه آزمایش** با استفاده از احتمالاتی که در مرحله قبل بدست آوردید، کلمات آن ایمیل را بررسی کنید و رابطه زیر را محاسبه کنید و کلاس پیش‌بینی شده را بدست آورید.

$$\prod_{i=۱}^{D}[P(x_i|y=۱)]P(y=۱) \overset{?}{>} \prod_{i=۱}^{D}[P(x_i|y=۰)]P(y=۰) \Rightarrow \begin{cases} Yes: & label = ۱ \\ No: & label = ۰ \end{cases}$$

اینکار را برای تمام ایمیل های موجود در مجموعه آزمایش انجام دهید و نتیجه پیش بینی خود را با برچسب‌های اصلی مقایسه کرده و دقت مدل خود را به دست آورید.

### پرسش‌ها

**۱.** اگر در متن ایمیل کلمه‌ای باشد که در ماتریس BoW وجود ندارد باید چکار کرد؟ صفر در نظر گرفتن احتمال آن یا در نظر نگرفتن آن کلمه چه پیامدی دارد؟ در مورد روش Laplace Smoothing تحقیق کنید و آن را در پروژه خود استفاده کنید. (۱۰ نمره)

**۲.** اگر متن پیام طولانی باشد، با ضرب شدن احتمال کلمات در هم چه اتفاقی می افتد؟ راه حل شما برای این مشکل چیست؟ نتایج استفاده از این روش را در پروژه‌ی خود گزارش کنید. (۱۰ نمره)

- راهنمایی:

$$\log(P(c \mid X)) \propto \log(P(c)) + \sum_{i=۱}^{n} \log(P(x_i \mid c))$$

**۳.** یکی از مشکلاتی که باعث می‌شود دقت پیش‌بینی ما کاهش پیدا کند، وجود کلماتی است که در هر متنی ممکن است وجود داشته باشند. کلماتی نظیر حروف اضافه، ضمایر ملکی و ... که به آنها stop words می گویند. به عبارت دیگر، برخی از کلمات به طور مکرر در تمام جملات از دو کلاس و برچسب مختلف تکرار می‌شوند. یعنی با اینکه احتمال وقوع آنها بالاست، اطلاعاتی در مورد برچسب آن جمله به ما اضافه نمی‌کنند. در نتیجه برای افزایش دقت پیش‌بینی، یکی از راه حل ها می تواند حذف این کلمات از BoW باشد. این راه را پیاده‌سازی کرده و نتیجه را با قسمت‌های قبل مقایسه کنید. (۱۰ نمره)