



عنوان : تمرین پنجم یادگیری ماشین (بیزین)

نگارنده : سحر داستانی اوغانی

شماره دانشجویی : ۹۹۱۱۲۱۰۸



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

# پاسخ قسمت تشریحی

(۱) مفهوم مدل‌های مولد (generative) و مدل‌های متمایزی (discriminative) را شرح داده و این دو مدل را در ۵

مورد مقایسه کنید.

شرح مفاهیم: برای درک بهتر مفاهیم ذکر شده در صورت سوال، از دو مثال زیر کمک می‌گیریم.

مثال اول: پدری دوفرزند به نام‌های A و B دارد. فرزند A مفاهیم را به صورت عمیق یاد می‌گیرد ولی فرزند B تنها تفاوت میان دو چیز را یاد می‌گیرد. روزی پدر، آن‌ها را به باغ وحش کوچکی می‌برد که تنها دو نوع حیوان از انواع فیل و شیر داشت. پس از تمام شدن بازدید باغ‌وحش، پدر حیوانی را به هر دوی آن‌ها نشان می‌دهد و از آن‌ها می‌خواهد که نوع حیوان مور نظر را تشخیص دهند. فرزند A بلافاصله دو تصویر کشید. یکی متعلق به تصویری بود که از فیل در باغ وحش دریافت کرده بود و دیگری برای شیر بود. این دو تصویر را با حیوانی که پدر به آن‌ها نشان داده بود، مقایسه کرد و جوابش را که "شیر" بود، بر حسب نزدیکی حیوان به هر کدام از تصاویر داد. فرزند B، پاسخ خود را که "شیر" بود، بر حسب تفاوت‌هایی که در باغ وحش دریافت کرده بود، داد. مثال دوم: فرض کنید باید متن یک سری سخنرانی را بر حسب زبان‌های مختلف آن‌ها دسته‌بندی کنید. راهکار ۱ برای این کار این است که تمامی زبان‌های مورد نیاز را یادگیرید و بر اساس دانشی که آموختین، دسته‌بندی را انجام دهید. راهکار ۲: بدون یادگیری زبان‌های مختلف و تنها با یادگیری تفاوت میان مدل‌های زبانی، دسته‌بندی را انجام دهیم.

**مدل مولد:** توزیع واقعی هر کلاس را مدل می‌کند. این مدل با استفاده از تئوری بیز، احتمال توزیع  $p(x, y)$  را یاد می‌گیرد و سعی

دارد این مقدار را با بدست آوردن مقدار مقابل جایگزین کند.  $\frac{p(x|y)p(y)}{p(x)}$  حال به دلیل این که در پی یافتن

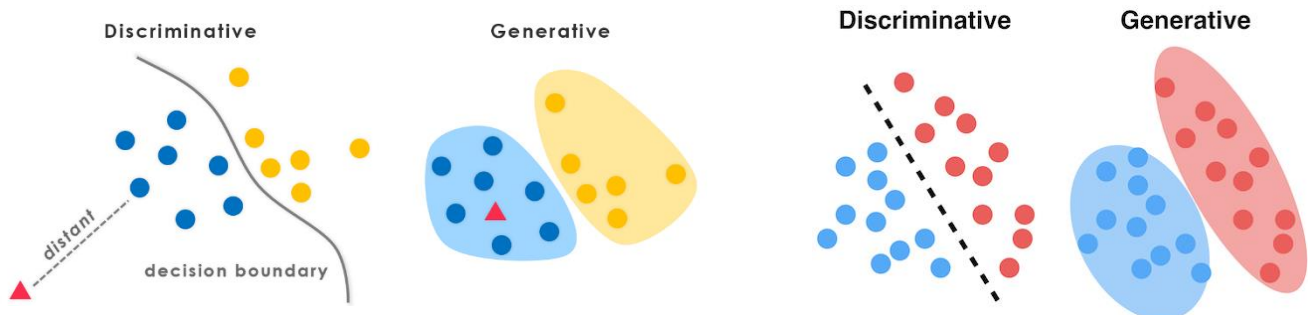
$\text{argmax } p(x|y)p(y)$  عبارت ذکر شده هستیم، می‌توان به طور مستقیم، تنها عبارت مقابل را تخمین زد.

در مثال‌های بالا، فرزند A و راهکار ۱ از نوع مدل مولد می‌باشند.

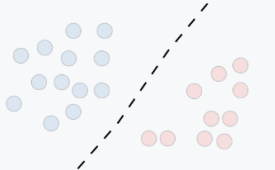
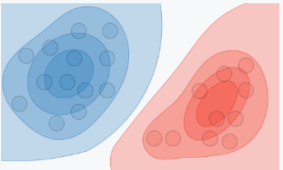
**مدل متمایزی:** مرز تصمیم‌گیری میان کلاس‌ها را مدل می‌کند. این مدل به صورت مستقیم عبارت  $p(y|x)$  را از مجموعه داده-

های train تخمین می‌زند. در مثال‌های بالا، فرزند B و راهکار ۲ از نوع مدل مولد می‌باشند.

برای درک کامل‌تر مطلب به شکل‌های زیر توجه کنید.



شکل بالا و شکل زیر مشخص می‌کنند که مدل مولد، توزیع احتمال داده‌ها و مدل متمایزی، مرز تصمیم‌گیری را مشخص می‌کند.

	Discriminative model	Generative model
Goal	Directly estimate $P(y x)$	Estimate $P(x y)$ to then deduce $P(y x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

مقایسه بر اساس تعاریف ذکر شده:

ردیف	شباهت/تفاوت	مدل مولد	مدل متمایزی
۱	تفاوت	سعی در یادگیری توزیع احتمال داده‌ها دارد.	سعی در یادگیری مرز تصمیم‌گیری دارد.
۲	تفاوت	تخمین $p(x y)p(y)$	تخمین مستقیم $p(y x)$
۳	تفاوت	محاسبات در مدل‌های مولد نسب به مدل‌های تمایزی بیشتر است. (در مدل تمایزی کمتر است)	
۴	تفاوت	زمانی که فرض استقلال مشروط رعایت نشود، مدل‌های مولد دقت کمتری نسبت به تمایزی دارند.	
۵	تفاوت	مدل‌های مولد می‌توانند با missing data کار کنند ولی در کل مدل‌های تمایزی نمی‌توانند.	
۶	تفاوت	مدل‌های مولد نسبت به مدل‌های تمایزی نیاز به داده‌ی train کمتری دارند. دلیل این امر این است که مدل‌های مولد با فرض‌های قویتر، بایاس ترند.	
۷	تفاوت	مدل‌های تمایزی به این دلیل تمایزی هستند زیرا برای تبعیض میان لیبل‌های Y کاربرد دارند، بنابراین تنها می‌توانند مسائل طبقه‌بندی را حل کنند. این درحالی است که مدل‌های مولد، کاربردها فراتری از طبقه‌بندی، مانند: samplings, bayes learning, MAP inference دارند.	
۸	شباهت	هر دو مدل احتمالی هستند، به این معنی که هر دو برای محاسبه کلاس‌های داده‌های ناشناخته از احتمال (به طور دقیق شرطی) استفاده می‌کنند.	

۲) با در نظر گرفتن دومدل مولد و تمایزی، صحت و عدم صحت موارد زیر را شرح دهید.

الف. مدل دسته‌بندی کننده **logistic regression** یک مدل تمایزی است.

صحیح است. Logistic regression بر اساس گفته‌های سوال قبل، یکمرز خطی میان دو کلاس ایجاد می‌کند. روی این خط احتمال تعلق داده به ۲ دسته را یکسان در نظر می‌گیرد. در یک طرف احتمال کلاس اول بیشتر و در طرف دیگر احتمال کلاس دوم بیشتر است. در واقع از خط، تعبیر احتمال می‌کند و آن را یاد می‌گیرد. (وزن‌های آن توسط ML تخمین زده می‌شوند). به همین دلیل می‌توان آن را یک مدل تمایزی دانست.

ب. در صورت داشتن **large dataset** بکارگیری مدل‌های تمایزی ارجح بر مدل‌های مولد است.

صحیح است. در مجموع، مدل‌های تمایزی قوی‌تر از مدل‌های مولد هستند و از این رو بر روی دیتاست‌های بزرگتر، نسبت به دیتاست‌های کوچکتر، بهتر عمل می‌کنند. از این رو ممکن است بر روی دیتاست‌های کوچک، دچار **overfit** شوند. از طرفی مدل‌های تمایزی، مانند **logistic regression** احتمال تعلق دسته‌ها را نیز مشخص می‌کند که خود می‌تواند دلیل خوبی برای استفاده در دیتاست‌های بزرگ باشد.

ج) در مورد **missing data** مدل مولد رفتار بهتری از خود نشان می‌دهد.

صحیح است. مدل‌های مولد عملکرد بهتری با داده‌های گم شده دارند ولی مدل‌های تمایزی خیر. در مدل‌های مولد، هنوز هم می‌توانیم **posterior**  $(p(y|x))$  را با **margin** کردن متغیرهای دیده نشده، تخمین بزنیم. این درحالی است که مدل‌های تمایزی، معمولاً تمام ویژگی‌های  $X$  را برای مشاهده، درخواست می‌کنند.

$$P(Y|X_{given}) = \sum_{X_{missing}} P(Y|X_{given}, X_{missing})$$

د. در مدل مولد وابستگی بین پارامترها تاثیر کمتری دارد.

در اینجا باید بررسی کنیم که کم یا زیاد شدن یکی از پارامترها بر دیگری چه اثری دارد. برای این کار، توزیع توام پارامترها را در نظر می‌گیریم؛ مدل مولد توزیع توام بین لیبل کلاس و پارامترها را به تصویر می‌کشد. بنابراین پارامترها روی یکدیگر اثر کمتری می‌گذارند.

این درحالی است که در مدل‌های تمایزی، احتمال شرطی را مدل می‌کنند.

ه. میزان بایاس در مدل‌های تمایزی بیشتر است.

غلط است. مدل‌های مولد، زمانی که فرض‌های قوی‌تری را می‌گیرند، بایاس‌تر هستند. ( assumption of

conditional independence)

و. ریسک **overfitting** در مدل‌های تمایزی بیشتر است.

صحیح است. مدل‌های تمایزی به دلیل قوی‌تر بودنشان نسبت به مدل‌های مولد، بر روی دیتاست‌های بزرگ بهتر عمل

می‌کنند و به همین دلیل بر روی دیتاست‌های کوچک دچار **overfitting** می‌شوند.

۳) در صورتی که  $x$  نشان دهنده مقدار پارامترها باشد و  $y$  کلاس دسته را نشان دهد، با داشتن مدل دسته‌بندی‌کننده باینری که  $y$  ویژگی دارد، مدل تمایزی و مدل مولد متناظر را رسم کرده و تعداد پارامترها را در هر یک محاسبه کنید.

متأسفانه این سوال را بلد نبودم.

۴) هر یک از مفاهیم **hypothesis space** ، **objective function** ، **knowledge discovery** ، **liklihood** ،

**hidden layer** و **latent variable** را توضیح دهید.

### **:hypothesis space**

یک مساله‌ی یادگیری ماشین را فرض کنید که ورودی آن با  $x$  و خروجی آن با  $y$  نمایش داده می‌شود که باید رابطه‌ای میان ورودی و خروجی مساله وجود داشته باشد. فرض کنید این رابطه، تابعی مانند  $y = f(x)$  باشد که به آن **target function** می‌گویند. در حالت معمول **target function** ناشناخته است، به همین دلیل الگوریتم‌های یادگیری ماشین تلاش می‌کنند که تابعی فرضی با نام **hypothesis function** را حدس بزنند که شامل فرضیه‌های  $h_1, h_2, \dots, h_n \rightarrow h \in H$  می‌باشد و به خوبی **target function** را تقریب می‌زند. مجموعه‌ای از **hypothesis**های ممکن را **hypothesis space** گویند. همچنین می‌توان آن را فضایی خواند که شامل فرضیه‌هایی است که ورودی‌ها را به خروجی‌ها، نگاشت می‌کنند. این فرضیه‌ها معمولاً توسط انتخاب قالب مساله، انتخاب مدل و انتخاب پیکربندی مدل، محدود می‌شوند.

### **:objective function**

تابعی است که می‌خواهیم آن را در مساله **minimise** یا **maximise** کنیم. برای مثال؛ فرض کنید مدلی مانند  $M$  تعریف کردیم. برای آموزش  $M$ ، معمولاً **loss function**ی مانند  $L$  را تعریف می‌کنیم که می‌خواهیم آن را **minimise** کنیم.  $L$  در اینجا **objective function** ایست که می‌خواهیم آن را **minimise** کنیم. یا برای مثال، مساله‌ی **TSP** را در نظر بگیرید. تابعی مانند

C تعریف می‌کنیم. این تابع هزینه‌ی تور یا دور همیلتونی را در بر دارد. در این مثال C، objective function ایست که ما در پی minimise کردن آن هستیم، زیرا در نهایت در پی آنیم که بهینه‌ترین و کم هزینه‌ترین مسیر را بیابیم.

### **:knowledge discovery**

فرایندی برای کشف الگوهایی است که منجر به دانش عملی (actionable knowledge) از طریق یک یا چند روش سنتی داده‌کاوی، مانند تجزیه و تحلیل سبد بازار و خوشه‌بندی می‌شود.

### **:Likelihood**

X را مجموعه‌ای از داده‌های مشاهده شده و  $\theta$  را مجموعه‌ای از پارامترها فرض کنید. از کجا مقدار  $\theta$  را بدانیم؟ اینجا جایی است که likelihood وارد عمل می‌شود. در واقع این تابع تراکم شرطی داده‌ها را نسبت به پارامتر آن‌ها می‌یابد.

$$\text{برای توزیع‌های پیوسته: } L_x(\theta) = f(x|\theta) \text{ و برای توزیع‌های گسسته: } L_x(\theta) = P(x|\theta)$$

### **:latent variable**

به طور شهودی، برخی از پدیده‌ها را مانند برخورد یا نوع دوستی، نمی‌توان اندازه گرفت ولی برخی دیگر، مانند سرعت یا قد را می‌توان به صورت کمی، عددی به آن‌ها نسبت داد. latent variable یا hidden variable یک متغیر رندوم است که قابل مشاهده در فازهای آموزش و تست، به صورت مستقیم، نمی‌باشد. این متغیرها، مانند متغیرهای ذکر شده، قابل اندازه‌گیری در مقیاس کمی نیستند.

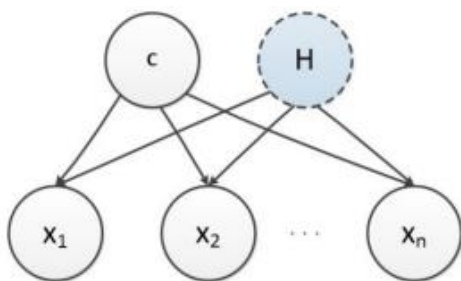
### **:hidden layer**

لایه یا لایه‌های مخفی در میان لایه‌ی ورودی و لایه‌ی خروجی قرار دارند و این دلیل اصلی نام‌گذاری آن به لایه‌ی مخفی است. این لایه‌(ها) برای سیستم‌های خارجی قابل رویت نیستند و برای شبکه عصبی، به صورت خصوصی کار می‌کنند. شبکه عصبی می‌تواند صفر یا بیشتر لایه‌ی مخفی داشته باشد. هرچه تعداد آن‌ها بالاتر می‌رود، زمان محاسبه خروجی و پیچیدگی شبکه نیز بالاتر می‌رود.

## (۵) شرح دهید در یک مدل بیز ساده، متغیر پنهان به چه معناست و چه کاربردی دارد؟

توضیحات از یک مقاله‌ی کنفرانسی با نام "ارائه‌ی مدل آمیخته طبقه‌بندی کننده بیز ساده برای پیش‌بینی خطای نرم‌افزار" نوشته شده است. (نویسندگان: نیما شیرینی - ساسان علی زاده)

طبقه‌بندی کننده بیز ساده، بر اساس تئوری بیز با فرض استقلال شرطی میان متغیرها، به شرط مقدار کلاس ساخته می‌شود. در داده‌هایی که ویژگی‌ها، وابستگی بالایی با یکدیگر دارند می‌توان با این فرض، آن‌ها را مستقل از یکدیگر در نظر گرفت. یعنی با استفاده از متغیرهای پنهان، فرض استقلال شرطی میان متغیرها به شرط کلاس استفاده شده را کمتر کنیم. این عمل، محاسبات را برای بدست آوردن بیز ساده، آسان‌تر می‌کند و درواقع متغیر پنهان باعث می‌شود که سایر متغیرها نسبت به یکدیگر مستقل شوند.



**نحوه‌ی اضافه کردن متغیر پنهان به دسته‌بند بیز ساده:**

ساختار طبقه‌بند بیز ساده با متغیر پنهان، مانند رویرو است.

متغیر  $C$  بیانگر کلاس بوده و متغیرهای  $x_1, x_2, \dots, x_n$  ویژگی‌های

مورد استفاده، جهت طبقه‌بندی می‌باشند.  $H$  نیز یک متغیر پنهان است.

نقش اصلی آن، مدلسازی وابستگی میان ویژگی‌ها به شرط کلاس می‌باشد. زیرا بر اساس قوانین شبکه‌های بیزین، ویژگی‌های

$x_1, x_2, \dots, x_n$  نسبت به یکدیگر، تشکیل یک رابطه همگرا یا **diverge** را می‌دهند که این رابطه، خود از ۲ مسیر  $C$  و مسیر  $H$

تشکیل شده است. درنتیجه زمانی ویژگی‌ها از یکدیگر مجزا خواهند شد که مقدار هر ۲ متغیر  $C$  و  $H$  مشخص باشند، در غیر این

صورت، این ویژگی‌ها به یکدیگر متصل‌اند.

**استفاده از متغیر پنهان  $H$  در روابط ریاضی:**

$$\arg \max_{c \in C} P(C | x_1, \dots, x_n) = \arg \max_{c \in C} \sum_H P(C, H | x_1, \dots, x_n)$$

$$P(C, H | x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n | C, H) P(C, H)}{\sum_{C, H} P(x_1, \dots, x_n | C, H)}$$

به دلیل اینکه مخرج برای تمام کلاس‌ها یکسان است، پس در محاسبه  $\max$  تاثیری نخواهد داشت و می‌توانیم آن را حذف کنیم.

$$\arg \max_{c \in C} P(C | x_1, \dots, x_n) \propto \arg \max_{c \in C} \sum_{C, H} P(x_1, \dots, x_n | C, H) P(C, H)$$

با توجه به قانون حاکم بر اتصالات و اگر در شبکه‌های بیزین داریم:

$$P(x_1, \dots, x_n | C, H) = \prod_{i=1}^n P(x_i | C, H)$$



همچنین با توجه به اینکه هیچ یک از متغیرهای  $x_1, x_2, \dots, x_n$  در احتمال توأم آن‌ها وجود ندارد این ۲ متغیر از یکدیگر مجزا

$$P(C, H) = P(C)P(H) \quad \text{می‌باشند:}$$

$$P(C, H | x_i) \neq P(C | x_i)P(H | x_i) \quad \text{توجه کنید با حضور } x_i \text{ ها، } C \text{ و } H \text{ دیگر مستقل نیستند:}$$

پس با استفاده از متغیرهای پنهان شرط استقلال بین ویژگی‌ها را کمتر گردیم و می‌توانیم در مسائلی که ویژگی‌ها از هم نسبت به

$$\arg \max_{c \in C} = \sum_H \prod_{i=1}^n P(x_i | C, H) P(C) P(H) \quad \text{هر کلاس مستقل نیستند از این روش استفاده کنیم.}$$

## ۶) هدف از آموزش در مدل بیز ساده چیست؟

مدل بیز در دسته‌بندی برای یک داده‌ی خاص، به دنبال دسته‌ای می‌گردد که محتمل‌ترین دسته برای آن داده باشد. برای اینکار،

فرض می‌کند مقادیر از لحاظ احتمالاتی نسبت به یکدیگر مستقل هستند.

این مدل، جستجوی خود را به ساده‌ترین روش ممکن، یعنی با شمارش تعداد ترکیبات داده‌ی ورودی، انجام می‌دهد.

## ۷) نقطه ضعف اصلی مدل‌هایی که بر اساس likelihood استنتاج می‌کنند، چیست؟

۱. Likelihood برای دیتاست‌هایی با تعداد کم، overfit می‌شود و ممکن است پاسخ اشتباه دهد.

برای مثال، فرض کنید می‌خواهیم احتمال شیر یا خط آمدن سکه‌ای را بررسی کنیم که سالم است، یعنی احتمال شیر یا خط

آمدن آن  $\frac{1}{2}$  است. حال فرض کنید ۳ پرتاب اول، شیر آمدند، در این صورت تخمین پارامتر توسط این روش، پاسخ احتمالی ۱ را

$$\frac{\partial \ln p(\mathcal{D} | \theta)}{\partial \theta} = 0 \Rightarrow \theta_{ML} = \frac{\sum_{i=1}^N x^{(i)}}{N} = \frac{m}{N} \quad \text{می‌دهد. یعنی تمامی پرتاب‌ها در این سکه، شیر می‌آیند.}$$

$$\mathcal{D} = \{1, 1, 1\}, \hat{\theta}_{ML} = \frac{3}{3} = 1$$

۲. likelihood نسبت به انتخاب نقطه‌ی شروع، حساس است.

۳. تخمین‌های عددی معمولاً پیش پا افتاده نیستند. به جز مواردی که فرمول ساده است.

۴. معادله‌ی likelihood باید به طور خاص برای هر توزیع و تخمینی کار کند. زیرا ریاضیات این فرمول، بدیهی و پیش پا افتاده

نیستند.

# گزارش پیاده‌سازی

الف. ابتدا از مجموعه داده معرفی شده، ویژگی‌های مورد نظر خود را استخراج کنید. چگونگی انتخاب ویژگی و پیش‌پردازش بر روی داده اولیه را توضیح دهید.

## فراخوانی دیتاست

دو راه برای فراخوانی دیتاست 20\_newsgroups وجود دارد:

راه اول ← استفاده از کتابخانه `scikitlearn`.

```
['alt.atheism',  
'comp.graphics',  
'comp.os.ms-windows.misc',  
'comp.sys.ibm.pc.hardware',  
'comp.sys.mac.hardware',  
'comp.windows.x',  
'misc.forsale',  
'rec.autos',  
'rec.motorcycles',  
'rec.sport.baseball',  
'rec.sport.hockey',  
'sci.crypt',  
'sci.electronics',  
'sci.med',  
'sci.space',  
'soc.religion.christian',  
'talk.politics.guns',  
'talk.politics.mideast',  
'talk.politics.misc',  
'talk.religion.misc']
```

اگر به این شیوه دیتاست را فراخوانی کنیم، موضوعات مختلف دیتاست به شرح روبرو می‌گردد.

طول متن‌ها و برچسب‌های آن‌ها نیز به شرح زیر است:

```
The length of texts are: 18846  
The length of lables are: 18846
```

راه دوم ← استفاده از سایت `uci` برای فراخوانی دیتاست.

برای شروع، ابتدا باید دیتاست مذکور را از سایت `uci` دانلود کنیم و آن را در فایل تمرین قرار

دهیم. سپس لازم است، آن را در کد، فراخوانی کنیم.

برای این کار انجام چندین مرحله لازم است. ابتدا کافی است نام هر فولدر را در لیستی (به نام `categories`) ذخیره کنیم. سپس

فایل‌های موجود در لیست قبل را به یک لیست (به نام `files`) اضافه کنیم. قدم بعدی اضافه کردن داده‌های هر `document` است

```
['alt.atheism',  
'comp.graphics',  
'comp.os.ms-windows.misc',  
'comp.sys.ibm.pc.hardware',  
'comp.sys.mac.hardware',  
'comp.windows.x',  
'misc.forsale',  
'rec.autos',  
'rec.motorcycles',  
'rec.sport.baseball',  
'rec.sport.hockey',  
'sci.crypt',  
'sci.electronics',  
'sci.med',  
'sci.space',  
'soc.religion.christian',  
'talk.politics.guns',  
'talk.politics.mideast',  
'talk.politics.misc',  
'talk.religion.misc']
```

و در نهایت کافی است، کلاس هر `document` را در متغیری ذخیره کنیم. به این ترتیب

توانستیم داده‌های موجود در دیتاست دانلود شده را فراخوانی کنیم.

این دیتاست شامل ۲۰,۰۰۰ پیام خبری از ۲۰ موضوع متفاوت است. موضوعات دیتاست به شرح

روبرو است.

طول متن‌ها و برچسب‌های آن‌ها نیز به شرح زیر است:

```
The length of texts are: 19997  
The length of lables are: 19997
```

موضوعات خبری در هر دو دیتاست یکسان ولی طول داده‌ها متفاوت است.

## پیش پردازش

دیتاست‌هایی که دارای متن هستند، شرایط ویژه‌ای را برای کار کردن ارائه می‌دهند. بنابراین نمی‌توان با آن‌ها مانند دیتاست‌های دیگر رفتار کرد. برای حل این مشکل، کافی است قبل از شروع کار با داده‌های متنی، آن‌ها را پیش‌پردازش کنیم.

پیش‌پردازش این داده‌ها می‌تواند شامل حذف، ایجاد تغییرات و یا حتی اضافه کردن مواردی به متون باشد. از موارد دیگری که می‌توان در پیش‌پردازش متون به آن اشاره کرد، وجود علامت‌های نگارشی مانند: نقطه، ویرگول یا کاما، علامت‌های دیگر مانند نقل قول تکی و دوتایی، وجود کلمات stopwords مانند a, about, above, after,...، علامت‌های سوال و تعجب، وجود فاصله‌ی میان خط‌ها یا پاراگراف‌ها، بزرگ و کوچک بودن حروف، وجود اعداد در متن و ... است.

اینگونه موارد گاهی باعث بوجود آمدن خطا در پردازش متن می‌گردند. برای مثال، اگر علامت ویرگول به یک کلمه چسبیده باشد، ممکن است پردازنده‌ی متن، ویرگول را به همراه کلمه‌ی چسبیده به آن به عنوان یک کلمه ببیند.

پیش‌پردازش در این کد در دو بخش انجام گرفته است. بخش اول را به تنهایی پیاده سازی کرده‌ام ولی بخش دوم را کاملاً از سایت [github](https://github.com) کمک گرفته‌ام و نکات بسیاری را از این تکه پیاده‌سازی یاد گرفتم.

پیش‌پردازش در بخش اول: در این بخش punctuationها، اعداد، علامت‌های موجود در آدرس‌های ایمیل، علامت‌های موجود در آدرس‌های IP به طور کامل حذف شدند. از طرفی تمامی حروف(فارق از بزرگ یا کوچک بودنشان) به حروف کوچک(lowercase) تبدیل شدند.

پیش‌پردازش در بخش دوم: در این بخش کار اساسی‌تری شکل گرفته است که در زیر به صورت مفصل به توضیح آن می‌پردازیم.

ابتدا stopwordsها به صورت جامع تعریف شدند، سپس توابعی تعریف شده است که هر کدام از آن‌ها وظیفه‌ی پردازش موردی خاص را در متن بر عهده دارند.

تابع preprocess: این تابع برای حذف punctuationها در لیست لغات به کار می‌رود. این تابع از تابع translate در پایتون استفاده کرده است تا بتواند مجموعه‌ای از کاراکترها را به مجموعه‌ای دیگر نگاشت کند.

در ابتدا فیلتری معرفی می‌کند که داده‌هایی را مانند tabها که غیر ضروری هستند را، پاک می‌کند.

سپس علامت نقل قول تکی، که در بسیاری از stopwordها ظاهر شده است را به همان صورت باقی می‌گذارد. زیرا در صورت حذف آن‌ها ممکن است معنای کلمات تغییر کند.

این تابع در ادامه white spaceها، رشته‌های عددی، کاراکترهای تکی، blankها و لغاتی که تنها ۲ کاراکتر دارند را حذف می‌کند و نقل قول‌ها را به جمله‌ی معمولی و حروف بزرگ را به حروف کوچک تبدیل می‌کند.

تابع `remove_stopwords`: انین تابع سعی دارد `stopword`هایی را که در ابتدا تعریف کرده بود را از متون حذف کند.

تابع `tokenize_sentence`: این تابع یک جمله را به لیستی از لغات تبدیل می‌کند.

تابع `tokenize`: این تابع یک `document` را به لیستی از لغات تبدیل می‌کند.

تابع `flatten`: این تابع بدون استفاده از `numpy`، یک آرایه‌ی ۲ بعدی را به یک آرایه‌ی یک بعدی تبدیل می‌کند.

## ویژگی‌ها

متون موجود در دیتاست به صورت مجموعه‌ای از کلمات ظاهر شدند. بدین صورت که ابتدا تمامی کلمات از متون استخراج شده و سپس بر حسب تکرارشان و تعداد حضورشان در متن، مرتب شدند. برای اختصاص ویژگی به مجموعه داده‌ها یک دسته `key_feature`، به عنوان اولین `n` کلمه از مجموعه مرتب شده انتخاب شده است. این `key_feature` برای نمایش متن‌ها به کار می‌رود. بدین صورت که به ازای هر کلمه در متن یک عدد نمایش داده می‌شود که این عدد نشان‌دهنده‌ی حضور آن کلمه در متن است. در این کد از دو تابع برای استخراج ویژگی‌ها استفاده کردیم که در ادامه به توضیح آن می‌پردازیم.

- `CountVectorizer`

- `TfidfTransformer`

`CountVectorizer` شامل پردازش متن، `tokenizing` و فیلتر کردن `stopword`ها می‌شود. این `vectorizer`، این عملیات را با ساختن یک `dictionary` از ویژگی‌ها و انتقال `document`ها به بردارهای ویژگی انجام می‌دهد.

مشکلی که هست این است که `document`های طولانی‌تر نسبت به `document`های کوتاه‌تر دارای میانگین `count value` بالاتری هستند. برای جلوگیری از این اختلافات بالقوه کافی است تعداد وقایع هر کلمه در یک `document` را بر کل کلمات موجود در `document` تقسیم کنید: این ویژگی‌های جدید را `tf` می‌نامند. یکی دیگر از اصلاحاتی که در `tf` وجود دارد، کوچک کردن وزن برای کلماتی است که در بسیاری از `document`ها در مجموعه وجود دارند و بنابراین از اطلاعات کمتری نسبت به کلماتی که فقط در قسمت کوچکتر از `document`ها وجود دارد، برخوردارند. این کوچک کردن، `tf-idf` نام دارد.

ب) سپس داده‌ها را به دو مجموعه داده آموزشی و آزمایشی یا نسبت ۷۰ و ۳۰، تقسیم کنید و مدل ساده بیز را با استفاده از داده آموزشی، آموزش دهید و ماتریس درهم ریختگی را برای داده‌های آزمایشی محاسبه کرده و ضمن ارائه آن به تحلیل نتایج بپردازید.

### آموزش مدل

Naïve bayes classifier یک خط پایه خوب ارائه می‌دهد. انواع مختلفی از این classifier وجود دارد که در این بخش تنها از MultinomialNB استفاده می‌کنیم. این طبقه‌بند، برای طبقه‌بندی داده‌هایی که ویژگی‌های گسسته دارند مناسب است. از جمله‌ی این داده‌ها می‌توان به لغات در یک دیتاست متنی اشاره کرد. به طور معمول این طبقه‌بند، تعدادی از ویژگی‌ها نیاز دارد که عدد صحیح باشند. با این حال، تعداد کسری مانند tf-idf نیز ممکن است موثر باشد. در پیاده‌سازی از دو طریق می‌توان مدل را fit کرد داده‌ها را predict کرد. در یکی از این طرق، می‌توان vectorize کردن داده‌ها را ابتدا به کمک countvectorizer و سپس به کمک tfidftransformet انجام داد. در نهایت مدل MultinomialNB را بر روی خروجی مرحله‌ی قبل پیاده کرد. راه دیگر استفاده از یک Pipeline (که مانند یک طبقه بندی کننده مرکب رفتار می‌کند) به جای انجام مراحل ذکر شده است.

### ماتریس درهم ریختگی

ماتریس درهم ریختگی مدل fit شده بر روی داده‌ها به شرح زیر است. این ماتریس دارای ۲۰ سطر و ۲۰ ستون است که هر کدام نشان‌دهنده‌ی یک کلاس از ۲۰ موضوع خبرهاست. این موضوعات به ترتیب بر روی ستون افقی و عمودی قرار گرفتند. خانه‌ی  $C_{i,j}$  تعداد دفعاتی را نشان می‌دهد که خبر  $i$  به اشتباه به عنوان خبر  $j$  دسته بندی شده است. خانه‌ی  $C_{i,i}$  تعداد دفعاتی را نشان می‌دهد که خبر  $i$  به درست به عنوان همان خبر دسته بندی شده است.

**True Positive:** You predicted positive and it's true.

**True Negative:** You predicted negative and it's true.

**False Positive:** You predicted positive and it's false.

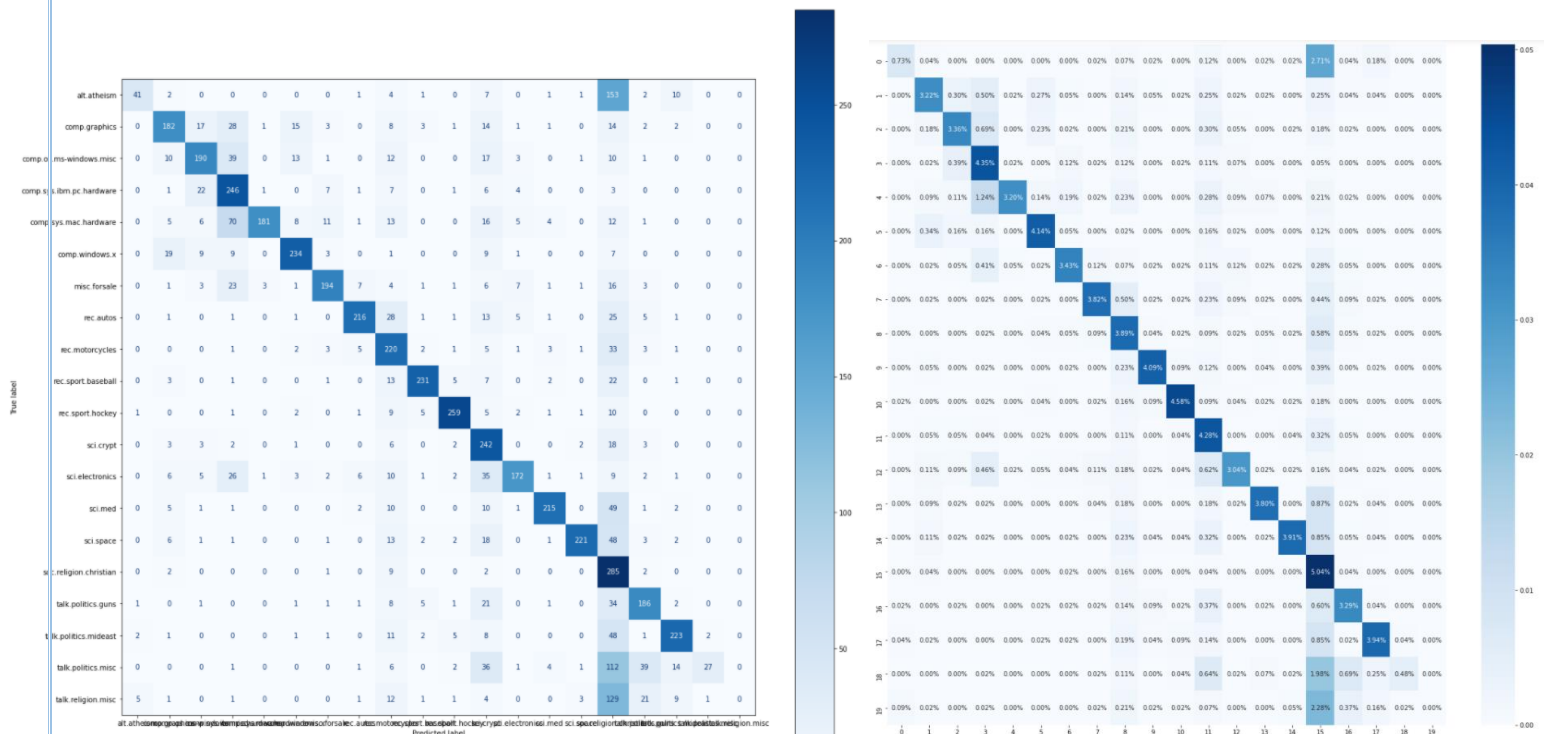
**False Negative:** You predicted negative and it's false.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

خبرهای زیر بسیار به یکدیگر نزدیک بوده و در طول دسته‌بندی، تعداد دفعات زیادی به اشتباه به جای یکدیگر دسته بندی شدند:

Atheism & Religion.Christian	153	Sport.hockey & Religion.Christian	10
Graphics & pc.hardware	28	Crypt & Religion.Christian	18
Windows.misc & pc.hardware	39	Electronics & Crypt	35
Pc.hardware & Windows.misc	22	Med & Religion.Christian	49
Mac.hardware & Pc.hardware	70	Space & Religion.Christian	48
Windows.x & Graphics	19	Christian & Motorcycles	9
Forsale & Pc.hardware	23	Politics.guns & Religion.Christian	34
Autos & Motorcycles	28	Politics.mideast & Religion.Christian	48
Motorcycles & Religion.Christian	33	Politics.misc & Religion.Christian	112
Sport.baseball & Religion.Christian	22	Religion.misc & Religion.Christian	129

در ادامه تصویری از نمودار heatmap خروجی ماتریس درهم ریختگی را می‌بینید. تصویر سمت چپ، خروجی خود ماتریس را نمایش می‌دهد و تصویر سمت راست، درصد داده‌هایی که در دسته‌بندی غلط یا درست قرار گرفتند را نمایش می‌دهد.



در گزارش زیر نیز می‌توانید دقت الگوریتم را مشاهده کنید. این الگوریتم دارای دقت ۶۶٪ می‌باشد.

	precision	recall	f1-score	support
alt.atheism	0.83	0.10	0.18	242
comp.graphics	0.66	0.69	0.67	283
comp.os.ms-windows.misc	0.74	0.62	0.68	298
comp.sys.ibm.pc.hardware	0.66	0.74	0.70	317
comp.sys.mac.hardware	0.87	0.64	0.73	310
comp.windows.x	0.80	0.81	0.80	298
misc.forsale	0.87	0.69	0.77	307
rec.autos	0.87	0.71	0.78	301
rec.motorcycles	0.92	0.63	0.75	308
rec.sport.baseball	0.96	0.77	0.86	309
rec.sport.hockey	0.56	0.93	0.70	279
sci.crypt	0.61	0.81	0.70	301
sci.electronics	0.75	0.65	0.70	271
sci.med	0.93	0.70	0.80	302
sci.space	0.90	0.68	0.78	297
soc.religion.christian	0.24	0.96	0.38	278
talk.politics.guns	0.74	0.65	0.69	285
talk.politics.mideast	0.77	0.81	0.79	270
talk.politics.misc	0.85	0.16	0.27	221
talk.religion.misc	1.00	0.01	0.01	177
accuracy			0.66	5654
macro avg	0.78	0.64	0.64	5654
weighted avg	0.78	0.66	0.66	5654

### ج) بررسی کنید که تعداد ویژگی‌های انتخاب شده چه تاثیری بر دقت دسته‌بندی می‌گذارد؟

هدف از به کار بردن ویژگی در این پیاده‌سازی، تشخیص بهتر، سریعتر و آسان‌تر کلاس مورد نظر می‌باشد. بنابراین هرچه تاثیر ویژگی‌ها و تعداد آن‌ها را بر روی دیتاست بیشتر کنیم، مشخصاً دقت تشخیص نیز بالاتر می‌رود. زیرا ویژگی‌ها، داده‌هایی را دسته‌بندی می‌کنند که دارای نزدیکی به یکدیگر هستند. بنابراین این داده‌ها از دیگر کلاس‌ها دور می‌شوند و همین امر موجب سریعتر دسته‌بندی شدن داده‌ها می‌گردد.

### د) با فرض مشخص بودن کلاس، برای زوج‌های مختلف از ویژگی‌های انتخابی در قسمت الف، شرط وابستگی یا استقلال ویژگی‌ها را مورد بررسی قرار دهید و تحلیل خود را در این زمینه بیان کنید.

لغات بسیاری هستند که معمولاً در کنار یک‌سری لغات دیگر ظاهر می‌شوند. برای مثال، فرض کنید می‌خواهیم در ایمیل‌های خود به دنبال این هستیم که ایمیل "یادگیری ماشین" را بیابیم. برای جستجوی آن بهتر است از واژه‌ی "یادگیری ماشین" به جای دو



واژه‌ی "یادگیری" و "ماشین" استفاده کنیم. زیرا ممکن است نتایج دیگری برای جستجوی "ماشین" وجود داشته باشد. بنابراین نتایج حاصل و دسته‌بندی، غلط انجام می‌شود.

برعکس آن نیز امکان دارد. یعنی اگر بخواهیم در ایمیل‌های خود به دنبال ایمیلی با موضوع "ماشین" بگردیم و به جای آن کلمه‌ی "یادگیری ماشین" را جستجو کنیم، نتایج اشتباه زیادی را مشاهده خواهیم کرد. این گونه موارد، استقلال و وابستگی ویژگی‌ها را نشان می‌دهد. طبق مثال انجام شده، با ایجاد شرط استقلال، دقت الگوریتم کاهش می‌یابد.

**ه) انتخاب چندین ویژگی (مثلا کلمه کلیدی) از یک دسته خاص از اسناد، بر روی نتیجه نهایی دسته‌بندی چه تاثیری خواهد گذاشت؟ تحلیل خود را در مورد آن بیان کنید.**

در این قسمت، تعداد ویژگی‌ها را مورد بحث قرار می‌دهیم و با کم و زیاد کردن آن، دقت الگوریتم را می‌سنجیم.

ابتدا، تنها لغاتی را مورد سنجش قرار می‌دهیم که حداقل  $n$  بار در متن ظاهر شده باشند. این کار را با به کار بردن  $\text{min\_df} = n$  در ورودی تابع `CountVectorizer` انجام می‌دهیم. نتایج را در جدول زیر بررسی می‌کنیم:

تعداد تکرار لغات	دقت الگوریتم
100	0.588256101874779
200	0.5042447824548992
300	0.4220021223912275
400	0.3638132295719844
500	0.32136540502299255
1000	0.2456667845772904
2000	0.19331446763353377

همانطور که مشاهده می‌کنید، هرچه تعداد ویژگی‌ها را زیادتر می‌کنیم دقت الگوریتم کمتر می‌گردد و اگر ویژگی‌ای روی آن قرار ندهیم، و تعداد خاصی برای بررسی لغات در نظر نگیریم، الگوریتم به بالاترین دقت خود دست می‌یابد. که این مقدار برابر با 0.6607711354793067 است.

(و) تعداد داده‌های آموزشی چه تاثیری بر دقت دسته‌بندی روی داده‌های آزمایشی خواهد داشت؟

تعداد داده‌های آموزشی را تغییر داده و نتایج را در جدولی آوردیم:

دقت الگوریتم	درصد داده‌های آموزشی
0.6742705570291777	%۹۰
0.6660477453580902	%۸۰
0.6563494870887867	%۷۰
0.6527390900649953	%۶۰

همانطور که مشاهده می‌کنید، با افزایش درصد داده‌های آموزشی، دقت الگوریتم بالا می‌رود. این امر نشان‌دهنده‌ی آن است که داده‌های آموزشی از توزیع مناسبی برخوردار هستند بنابراین با افزایش آن‌ها، تعمیم‌پذیری مدل افزایش می‌یابد.

1	<a href="https://medium.com/better-programming/generative-vs-discriminative-models-d26def8fd64a">https://medium.com/better-programming/generative-vs-discriminative-models-d26def8fd64a</a> - <a href="https://medium.com/@mlengineer/generative-and-discriminative-models-af5637a66a3">https://medium.com/@mlengineer/generative-and-discriminative-models-af5637a66a3</a> - <a href="https://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-a-discriminative-algorithm">https://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-a-discriminative-algorithm</a> - <a href="http://primo.ai/index.php?title=Discriminative_vs._Generative">http://primo.ai/index.php?title=Discriminative_vs._Generative</a>
2	<a href="https://deveshbatra.github.io/Generative-vs-Discriminative-models/">https://deveshbatra.github.io/Generative-vs-Discriminative-models/</a> - <a href="https://towardsdatascience.com/generative-vs-2528de43a836">https://towardsdatascience.com/generative-vs-2528de43a836</a> منبع کمکی دیگر: توضیحات سرکار خانم شیخی در گروه تلگرامی درس در رابطه با cnn
4	<a href="https://machinelearningmastery.com/what-is-a-hypothesis-in-machine-learning/">https://machinelearningmastery.com/what-is-a-hypothesis-in-machine-learning/</a> - <a href="https://www.quora.com/What-is-hypothesis-in-machine-learning">https://www.quora.com/What-is-hypothesis-in-machine-learning</a> - <a href="https://stats.stackexchange.com/questions/183989/what-exactly-is-a-hypothesis-space-in-machine-learning">https://stats.stackexchange.com/questions/183989/what-exactly-is-a-hypothesis-space-in-machine-learning</a> - <a href="https://ai.stackexchange.com/questions/9005/what-is-an-objective-function">https://ai.stackexchange.com/questions/9005/what-is-an-objective-function</a> - <a href="https://www.sciencedirect.com/topics/computer-science/knowledge-discovery">https://www.sciencedirect.com/topics/computer-science/knowledge-discovery</a> Fundamentals of Machine Learning (Part 2)   by William Fleshman   Towards Data Science <a href="https://medium.com/@manasmahanta10/latent-variable-models-demystified-7f1342698985">https://medium.com/@manasmahanta10/latent-variable-models-demystified-7f1342698985</a> - What Are Hidden Layers?. Important Topic To Understand When...   by Farhad Malik   FinTechExplained   Medium
5	<a href="https://www.researchgate.net/publication/324819388_arayh_mdl_amykhth_tbqh_bndy_knndh_byz_sadh_brav_pysh_byny_khtay_nrm_afzar">https://www.researchgate.net/publication/324819388_arayh_mdl_amykhth_tbqh_bndy_knndh_byz_sadh_brav_pysh_byny_khtay_nrm_afzar</a>
6	<a href="#">Mitchel book page 177 and 178</a>
7	<a href="https://www.itl.nist.gov/div898/handbook/eda/section3/eda3652.htm">https://www.itl.nist.gov/div898/handbook/eda/section3/eda3652.htm</a>

<https://towardsdatascience.com/implementing-a-naive-bayes-classifier-f206805a95fd>

<https://panjeh.medium.com/figure-size-plot-confusion-matrix-in-scikit-learn-2c66f3a69d81>

<https://github.com/jupyter/notebook/issues/2135>

[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\\_matrix.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)

<https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

<https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea>

[https://scikit-learn.org/stable/modules/generated/sklearn.naive\\_bayes.MultinomialNB.html](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html)

[https://scikit-learn.org/stable/tutorial/text\\_analytics/working\\_with\\_text\\_data.html](https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html)

[https://github.com/gokriznastic/20-newsgroups\\_text-classification](https://github.com/gokriznastic/20-newsgroups_text-classification)