بسمه تعالی

مقدمه‌ای بر یادگیری ماشین

نیمسال اول ۹۷-۹۸

تمرین عملی سری دوم

مدرس: صابر صالح

- داده‌های این تمرین را می‌توانید از آدرس زیر دریافت کنید. از آن جایی که این آدرس از پروتکل انتقال فایل یا همان FTP استفاده می‌کند، باید آن را در حال اتصال به شبکه‌ی داخلی دانشگاه باز کرده و دانلود کنید. اگر داخل محیط دانشگاه یا خوابگاه‌ها هستید، با اتصال به نقاط دسترسی بی‌سیم یا اتصال کابل شبکه‌ی Ethernet می‌توانید به محض اتصال به شبکه، دانلود را انجام دهید. اگر خارج از دانشگاه هستید، می‌توانید با اتصال VPN مربوط به شناسه‌ی شریف به آدرس `access2.sharif.edu` به شبکه‌ی داخلی دانشگاه متصل شوید و سپس دانلود را انجام دهید. توجه کنید که در هر دو حالت، نباید سرویس VPN دیگری فعال باشد و یا نرم‌افزاری که با آن دانلود را انجام می‌دهید به Proxy متصل باشد.

```
ftp://ftp.sharif.edu/OpenData-Dataset/EE/Dr.Saleh/data.tar.xz
```

- می‌توانید برای این که مطمئن شوید فایل فشرده بدون نقص دانلود شده است، از ترمینال سیستم دستور زیر را اجرا کنید. خروجی باید به شکل خط زیر آن باشد:

```
md5sum data.tar.xz

2c48abf7b24c0e8c9a64c5e496d8ce95  data.tar.xz
```

- این تمرین از تاریخ پنج‌شنبه ۲۶ مهر ۱۳۹۷ بارگذاری شده و به مدت ۲۰ روز برای آن فرصت دارید. موعد تحویل تمرین تا **ساعت ۱۱:۵۹ روز سه‌شنبه ۱۵ آبان ۱۳۹۷** خواهد بود.

- نمره‌ی تمرین بر اساس گزارش تحویلی، کد زده شده و میزان دقت مدل شما بر اساس معیاری که در تمرین مشخص شده است محاسبه می‌شود. بنابراین در مورد تهیه‌ی گزارش هم دقت لازم را به عمل بیاورید که بخش مهمی از نمره‌ی تمرین را شامل می‌شود.

- برای راحتی تست مدل خود روی داده‌های تست، در سامانه‌ی کوئرا، بخش **تست خروجی تمرین عملی سری دوم** قرار گرفته است. با فرستادن فایل خروجی خود روی داده‌ی تست به فرمت **output.csv** که در یک فایل zip ذخیره شده است، معیارهای گفته شده در تمرین برای خروجی مدل شما محاسبه می‌شود و به شما گزارش می‌گردد. توجه کنید که تغییراتی که در مدل خود بر اساس این فیدبک ایجاد می‌کنید نباید مدل شما را به صورت آماری تحت تأثیر قرار دهد. به زبان ساده‌تر شما نباید با استفاده از این خروجی مستقیماً پارامترهای مدل را دست‌کاری کنید تا به حد مطلوب‌تری برسید، بلکه باید نحوه‌ی ساخت مدل یا پارامترهای آن را تغییر داده و دوباره روی داده‌ی آموزش، آن را آموزش دهید. این کار این تضمین را به شما می‌دهد که اگر داده‌ی دیگری هم در اختیارتان قرار بگیرد، مدل شما هم‌چنان درست عمل می‌کند و به اصطلاح به داده‌ی تست، overfit نشده است.

- در مورد تقلب و کپی‌برداری از کار دیگران در هر بخش از تمرین مطابق سیاست‌های درس رفتار خواهد شد. می‌توانید از منابع اینترنتی برای حل سؤال کمک بگیرید ولی ابداً به کپی کردن از آن‌ها اکتفا نکنید و حتماً آن‌ها را قبل از استفاده درک کنید و به صورت مناسب برای تمرین در بیاورید.

۱ مقدمه

دنیای امروز، دنیای داده‌هاست و روز به روز به ارزش داده‌ها افزوده می‌شود. بسیاری از فن‌آوری‌های جدید روی بستر داده‌ها شکل می‌گیرد و در ضمن داده‌ها شکل فن‌آوری‌های قدیمی را هم دگرگون کرده است. در این میان، شناخت درست داده و استفاده‌ی صحیح و مطلوب از آن، امری بسیار حیاتی است. اگر شناخت کافی از یک داده نداشته باشیم و به مفاهیم پایه‌ای در مورد آن مسلط نباشیم، هیچ‌گاه نخواهیم توانست که همه‌ی آن ارزشی که درون داده وجود دارد را بهره‌برداری کنیم. **تحلیل اکتشافی داده**[۱] ، یکی از ابزارهای مهم برای شناخت داده است که هر چند در نگاه اول شاید روش مهم یا پیچیده‌ای به نظر نیاید ولی بخش مهمی از هر فرآیند یادگیری یا تحلیل داده را تشکیل می‌دهد. در این تمرین این مهارت مهم را تجربه خواهیم کرد و علاوه بر آن یک یادگیری ساده با مدل‌های خطی را خواهیم دید.

داده‌های ارزشمند معمولاً حاصل ساعت‌ها تحقیق و جمع‌آوری اطلاعات دقیق و حساب شده است. برخی از این داده‌ها برای عموم عرضه می‌شوند تا همه فرصت کار کردن و اکتشاف روی آن‌ها را داشته باشند و برخی از چنان اهمیتی برخوردارند که منتشر نمی‌شوند. با این وجود منبع عظیمی از داده‌های متنوع آماده‌ی کار هستند. برای مثال می‌توانید به سایت Kaggle مراجعه کنید. Kaggle که با ارائه‌ی مسابقات یادگیری ماشین شروع به کار کرده است، در حال حاضر نیز ارائه دهنده‌ی یک پلتفرم داده‌های عمومی، به صورت یک میز کار ابری[۲] برای علوم داده ها است. این وب‌سایت در ۸ مارس ۲۰۱۷ توسط گوگل خریداری شده است. می‌توانید دنیای بزرگی از مسائل را در آن تجربه کنید و به صورت هم‌زمان از تجربیات دیگران هم استفاده کنید.

۲ شرح مسئله

پشت سر هر مسئله‌ی یادگیری ماشین، قصه‌ای وجود دارد که انگیزه‌ی آن و هدف از مسئله را به خوبی نشان می‌دهد. این داستان پیش‌زمینه، می‌تواند در معرفی بهتر مسئله کمک کند. مسئله‌ی این تمرین از این قرار است:

---

[۱] Exploratory Data Analysis (EDA)

[۲] Cloud Platform

یکی از قوی‌ترین ابزارهای موجود در دنیای برنامه‌های کاربردی و سرویس‌های موبایل برای جذب کاربران، **پوش نوتیفیکیشن** است. این ابزار امکان ارتباط خارج از نرم‌افزار را با کاربر فراهم می‌کند و می‌تواند او را به داخل برنامه‌ی کاربردی بکشاند. اما همیشه ابزارهای قوی خطرناک هم هستند؛ ارسال اعلان به کاربری که علاقه‌ای به آن ندارد، او را آزار خواهد داد و احتمال حذف برنامه‌ی کاربردی و از دست دادن کاربر یا ساکت کردن پوش نوتیفیکیشن[۳] از طرف آن برنامه را بالا خواهد برد.

در چنین موقعیتی، هوشمندی در ارسال و تشخیص مخاطب درست هر اعلان اهمیت ویژه‌ای پیدا می‌کند. با شناسایی افراد علاقمند می‌توان بدون ایجاد مزاحمت برای کسانی که به محتوای یک اعلان علاقه ندارند، از مزایای ارسال آن برای افرادی که آن محتوا برایشان مهم است، استفاده کرد و آن‌ها را به استفاده از برنامه ترغیب نمود. در این مسئله از شما خواسته می‌شود که با کمک تاریخچه کلیک کردن یا نکردن کاربران روی اعلان‌های مختلف، احتمال کلیک آن‌ها روی اعلان‌های جدید را پیش‌بینی کنید.

۳ پیش‌پردازش داده‌ها

قبل از این که بتوان از یک داده در یک مسئله‌ی یادگیری ماشین یا تحلیل داده استفاده کرد، باید عملیات‌های متعددی برای مرتب‌سازی آن داده انجام داد که به **پیش‌پردازش داده**[۴] شناخته می‌شوند. این عملیات با خواندن صحیح فایل‌ها آغاز می‌شود و با حذف سطرها و ستون‌های اضافی و حل مشکل داده‌های از دست‌رفته ادامه پیدا می‌کند.
توجه کنید که داده‌ها به صورت دو مرحله‌ای فشرده شده‌اند. برای خارج کردن فایل‌ها از حالت فشرده ابتدا باید با اجرای دستور زیر در خط فرمان تحت یونیکس[۵] فایل‌ها را از یک‌دیگر جدا کنید:

```
tar -xzf <file-name>
```

سپس با مراجعه به محلی که داده‌ها از حالت فشرده خارج شده‌اند، برای هر فایل جداگانه این دستور را اجرا کنید.

```
gzip -k -d <file-name>
```

هم‌چنین می‌توانید از رابط گرافیکی سیستم‌عامل‌ها مانند Archive Manager یا WinRar استفاده کنید.

۱.۳ خواندن درست داده‌ها

داده‌ها را می‌توان به روش‌های متنوعی ذخیره کرد و انتقال داد که در این تمرین با یک روش استاندارد و رایج آن آشنا خواهیم شد. داده‌ها به فرمت **بردارهای جداشده با کاما**[۶] یا CSV به صورت خطوط جدا شده است که نماینده‌ی سطرهای داده هستند. ستون‌های داده هم با کاما از یک‌دیگر جدا شده‌اند. این یکی از ساده‌ترین و فشرده‌ترین نمایش هایی است که می‌توان از یک داده ارائه کرد. برای اطلاعات بیش‌تر در مورد این شیوه‌ی ذخیره‌سازی می‌توانید به توضیحات کامل و مفصل صفحه‌ی ویکی‌پدیای انگلیسی آن مراجعه کنید. با این که بیش‌تر نرم‌افزارهای صفحه‌گستر[۷] مانند اکسل[۸] می‌توانند این فایل‌ها را باز کنند ولی اگر این کار را با داده‌های این تمرین انجام دهید، می‌فهمید که استفاده و باز کردن این داده‌ها با این روش‌ها ممکن نیست. این نرم‌افزارها سعی می‌کنند همه‌ی فایل را وارد حافظه‌ی اصلی سیستم کنند که در مورد داده‌های بزرگ ممکن نیست. در ضمن برخی از این نرم‌افزارها محدودیت تعداد سطرها یا ستون‌ها دارند که در CSV وجود ندارد. پس چگونه باید برای بررسی مختصر محتوای این فایل‌ها اقدام کرد؟
در سیستم‌عامل‌های تحت یونیکس، دستورات متنوعی برای کار با فایل‌های بزرگ یا گسترده وجود دارد. در این جا به چند مورد مفید آن اشاره می‌کنیم.

دستور less با توجه به عملکرد دستور less که فقط قسمتی از فایل متنی را برای نشان دادن به کاربر باز می‌کند، باز کردن سنگین‌ترین فایل‌ها با آن هم بدون دردسر است. با کلیدهای جهت می‌توانید در فایل حرکت کنید و با q نیز فایل بسته می‌شود.

```
less <file-name>
```

دستور grep این دستور برای جستجو در فایل‌های متنی کاربرد دارد. در این تمرین تعدادی فایل داده در اختیار شما قرار دارد که ستون‌های مشترک دارند. برای پیدا کردن ویژگی‌های یک اعلان که در فایل train داده شده است، شما باید سطر مربوط به آن را در فایل مربوط به ویژگی‌های اعلان پیدا کنید. تصور کنید که قصد دارید ویژگی‌های یک اعلان خاص را بررسی کنید. مسلماً انجام این کار به صورت دستی یا حتی با استفاده از کد هم بسیار وقت‌گیر است. دستور grep می‌تواند بسیار مفید باشد. یک حالت کلی از این دستور در ادامه آمده است.

```
grep -rnw <file-name> -e "<key-word>"
```

در صورت حذف w اگر عبارت مذکور جزیی از یک کلمه دیگر باشد نیز در نتیجه نمایش داده می‌شود.

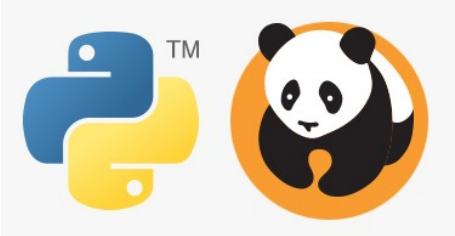
---

[۳] Push Notification
[۴] Data Pre-processing
[۵] Unix-like Command Line
[۶] Comma Seperated Vector (CSV)
[۷] Spreadsheet
[۸] Microsoft Excel

دستورهای head و tail این دو دستور نیز برای نمایش ۱۰ خط ابتدایی یا ۱۰ خط انتهایی از یک فایل کاربرد دارند. با آپشن n می‌توان تعداد خطوط قابل نمایش را مشخص کرد.

```
head -n <line-count> <file-name>
tail -n <line-count> <file-name>
```

برای برخی از کاربردها، تحلیل و یادگیری داده‌های خیلی بزرگ، مثل داده‌های ژنومیک[۹]، حتی کار از این هم پیچیده‌تر و سنگین‌تر می‌شود. پیش‌رفت علمی در این حوزه‌ها سال‌ها پشت سد بلند پردازش‌های کامپیوتری گیر کرده بود که اخیراً با کمک پردازش‌های توزیع شده[۱۰] از آن رهایی یافته است. پردازش‌های توزیع شده مبتنی بر تکه‌تکه کردن داده‌هاست و هر بخش از داده به صورت جداگانه توسط پردازنده‌ای جدا پردازش می‌شود. در ابتدا ممکن است این کار بدیهی به نظر برسد اما همه‌ی داده‌ها و الگوریتم‌ها برای توزیع‌شده بودن آماده نیستند و حتی ممکن است نتایج هم‌گرا نشوند. برای استفاده‌ی درست و بهینه از این روش‌ها باید داده‌ها به صورت مناسب در بیایند. بعضی از الگوریتم‌ها نیز در این محیط مطالعه شده‌اند و بهینگی آن‌ها بررسی شده است. در حال حاضر این حوزه از حوزه‌های جدید و داغ علوم کامپیوتر است. از روش‌های تازه توسعه‌یافته و بسیار کاربردی می‌توان به **آپاچی اسپارک**[۱۱] اشاره کرد. می‌توانید برای اطلاعات بیش‌تر به صفحه‌ی ویکی‌پدیای انگلیسی مراجعه کنید.



یکی از بخش‌های زمان‌گیر تحلیل داده عملیات خواندن و نوشتن روی دیسک سخت است. میلیون‌ها خط داده که به صورت خام روی دیسک سخت نوشته شده‌اند باید برای تحلیل و استفاده وارد حافظه‌ی اصلی کامپیوتر شوند که این عملیات بسیار وقت‌گیری است و بنابراین لازم است درست و بهینه انجام شود. برای این کار، کتابخانه Pandas در پایتون پیشنهاد می‌شود که در ادامه درباره‌ی آن توضیحاتی داده خواهد شد. برای مثال، برای خواندن یک فایل csv در این کتابخانه و لود کردن آن در یک فریم داده DataFrame تکه‌کد زیر پیشنهاد می‌شود.

```
train_df = pandas.read_csv('../data/train.csv',
                   usecols=[0, 1],
                   skiprows=4,
                   nrows=10,
                   )
```

با اجرای تکه کد فوق، از مسیر مشخص شده، فقط از ستون‌های اول و دوم، با شروع از سطر چهارم، به تعداد ده سطر از فایل csv خوانده می‌شود و در یک شی از جنس DataFrame ریخته می‌شود. برای مطالعه بیشتر درباره این تابع و بقیه توابع، به مستندات کتابخانه Pandas مراجعه کنید.

۲.۳ ساختار داده‌ها

اطلاعات موجود در فایل‌ها به شرح زیر طبقه‌بندی می‌شود. باید فایل‌های لازم را بنا به نیاز سؤال بارگذاری کرده و استفاده کنید. بعضی فایل‌ها با ستون‌هایی به برخی دیگر از فایل‌ها ارتباط پیدا می‌کنند. مهارت شما در فراخوانی داده، باید ارتباط این فایل‌ها را با یک‌دیگر آسان کند. در مورد انواع داده و همچنین مفهوم index در دیکشنری در بخش بعدی توضیحات مختصری آمده است.

فایل train.csv این فایل شامل ۹ ستون زیر است:

- user_id - شناسه‌ی کاربر
- notif_id - شناسه‌ی اعلان[۱۲]
- interaction - آیا کاربر روی آن کلیک کرده است یا خیر که به صورت باینری (صفر، یک) است.
- interaction_min - interaction_hour - interaction_dow - به ترتیب روزهفته، ساعت و دقیقه‌ای که کلیک یا رد اعلان در آن صورت گرفته است.
- delivery_min - delivery_hour - delivery_dow - به ترتیب روزهفته، ساعت و دقیقه‌ای که اعلان به دستگاه کاربر رسیده است.

فایل test.csv این فایل شامل ۲ ستون زیر است:

- user_id - شناسه‌ی کاربر
- notif_id - شناسه‌ی اعلان

---

[۹] Genomic Data
[۱۰] Distributed Computing
[۱۱] Apache Spark
[۱۲] Notification

فایل users.csv این فایل شامل ویژگی‌های کاربران بوده و دارای ستون‌های زیر است:

- user_id - شناسه کاربر
- N1 - N2 - N3 - سه ویژگی از نوع عددی
- C1 - C6 - شش ویژگی نوع دسته‌ای
- I1 - I96 - شامل ۹۶ ویژگی تنک از اعداد صحیح

فایل notifs.txt هر خط از این فایل ویژگی‌های یک اعلان را نشان می‌دهد و فرمت آن به ترتیب به این صورت است (موارد زیر با فاصله از هم جدا شده‌اند)

- شناسه‌ی اعلان (notif_id)
- شماره‌ی روز هفته‌ی ارسال اعلان
- ساعت ارسال اعلان
- دقیقه‌ی ارسال اعلان
- یک ویژگی از نوع دسته‌ای
- تعداد نامشخصی کلمات آن اعلان (که هر کلمه باعددی که index آن در یک دیشکنری است جایگزین شده است)

به عنوان تمرین، ابتدا همه‌ی سطرهای مربوط به یک کاربر را جدا کنید، سپس سطرهای متناظر اعلان‌های این کاربر را در فایل خود پیدا کرده و نمایش دهید. خروجی‌ها را در یک فایل متنی با نام search-result.txt ذخیره کنید.

۳.۳ انواع داده‌ها

داده‌ها به صورت متنوعی گزارش می‌شوند. قبل از کار با یک داده باید نوع آن را فهمید و روش درست را به کار گرفت. در این جا فهرست مختصری از این داده‌ها آمده است:

۱. **عددی**[۱۳] - این داده‌ها در واقعیت عدد هستند و مقادیر احتمالاً پیوسته‌ای را نشان می‌دهند. چیزی که این نوع داده را مشخص می‌کند امکان عملیات‌های ساده‌ی ریاضی روی آن است به نحوی که هم‌چنان معنا داشته باشند. از این عملیات‌ها می‌توان به جمع و ضرب اسکالر اشاره کرد. برای مثال شدت رنگ یک کمیت عددی است چرا که می‌توان دو شدت رنگ را با یک‌دیگر جمع کرد و باز شدت رنگ به دست آورد و یا ده برابر شدت بیش‌تر یک رنگ معنی دارد. اما خود رنگ کمیت عددی نیست و مثلاً ده زرد معنی رنگ را با خود حمل نمی‌کند.

۲. **دسته‌ای**[۱۴] - به داده‌هایی گفته می‌شود که برچسب آن‌ها به جای یک عدد، یک طبقه یا دسته است مانند جنسیت یا حالت مو. بر خلاف قد که با یک عدد بیان می‌شود، حالت مو با دسته‌هایی مانند فر، صاف و غیره مشخص می‌شود. این داده‌ها به صورت مجموعه‌ای متناهی از حالت‌ها ذخیره شده‌اند. برخی از الگوریتم‌های یادگیری ماشین مانند درخت تصمیم‌گیری می‌توانند مستقیم با داده‌های دسته‌ای کار کنند ولی بیشتر الگوریتم‌ها فقط با اعداد کار می‌کنند. بنابراین باید این دسته‌ها را به نحوی به اعداد تبدیل کرد.

   اگر اعداد نسبت داده شده معنای ترتیبی خود را حفظ کرده باشند، این اعداد به صورت مناسب نمایان‌گر داده هستند. برای مثال اگر مدرک تحصیلی افراد را به ترتیب از دبستان با صفر شماره‌دهی کنیم، شماره‌های این داده‌ی فهرستی، معنی ترتیبی خود را حفظ کرده‌اند. یعنی وقتی ۳ بزرگ‌تر از یک است یعنی کسی که مدرک کاردانی دارد از کسی که مدرک سیکل دارد دانش بیش‌تری دارد که خب این صحیح است. ولی در مورد مثال مو این گونه نیست. شما نمی‌توانید ترتیبی ذاتی بین حالت‌های فر، صاف و غیره پیدا کنید. در این موارد باید احتیاط بیش‌تری در تبدیل داده به خرج داد که یک روش ساده **one-hot-encoding** است. در این روش هر حالت به صورت یک ستون جدا معرفی می‌شود و مقادیر صفر و یکی داده می‌شود. این کار دو مزیت دارد. اول این که فرض ترتیبی که در حالت اعداد صحیح وجود دارد را از بین می‌برد و دوم داده شکل شبه‌پراکنده پیدا می‌کند که ذخیره و پردازش آن به مراتب ساده‌تر است. اما توجه کنید که این روش در برخی حالات ممکن است تعداد ستون‌ها را خیلی زیاد کند! در واقع برخی از ستون‌های این داده‌ها ارزش محتوایی زیادی ندارند. اگر به صورت نادقیق صحبت کنیم، پراکندگی مقادیر آن‌ها به قدری زیاد است که پیچیدگی مدل [۱۵] بالایی برای یادگیری احتیاج دارند که ممکن است کار را مشکل کند یا نتیجه‌ی نادرستی ایجاد کند.

۳. **تنک**[۱۶] - این نوع داده به صورت ذاتی به طور متفاوتی ذخیره شده است. در واقع مجموعه‌ای از مشخصه‌های وابسته به یک‌دیگر، به صورت یک ماتریس نمایش داده شده‌اند که تعداد زیادی از خانه‌های این ماتریس مقدار صفر دارد. برخی از این خانه‌ها فقط مقدار ناصفر دارند که تعداد آن‌ها ناچیز است.

منظور از index در دیکشنری به این صورت است که برای ذخیره‌سازی کلمات به جای این که خود کلمات که حجم زیادی دارند ذخیره شود، شماره‌ی آن‌ها در یک دیکشنری، مانند دفترچه‌ی تلفن ذخیره می‌شود. در این روش به طور ضمنی فرض می‌شود که بار معنایی کلمات اهمیتی ندارند و تنها تکرار کلمات و تعداد آن‌ها مهم است.

---

[۱۳] Numerical
[۱۴] Categorical
[۱۵] Model Complexity
[۱۶] Sparse

۴.۳ کار با داده‌های گم‌شده

اگر به خوبی در داده‌ها عمیق شوید، می‌بینید که برخی از داده‌ها گم شده[۱۷] هستند. این به معنی است که این بخش از داده‌ها جمع‌آوری نشده، نامعتبر بوده یا از دست رفته است. در هر صورت شما هیچ اطلاعی از مقدار واقعی آن ندارید و بنابراین ارزش اطلاعاتی خود را از دست داده است. سطرها یا ستون‌هایی که دارای تعداد زیادی از N/A هستند، فرآیندهای یادگیری را دچار اختلال می‌کنند و بنابراین لازم است حذف شوند. ابتدا باید ستون‌ها یا سطرهای با این ویژگی را تشخیص داده و بعد از داده حذف کنید. توجه کنید که کیفیت یادگیری شما می‌تواند به این پارامتر هم وابسته باشد.
برای سایر نمونه‌های نامعلوم می‌توانید روش‌های مختلفی برای پر کردن آن‌ها به کار بگیرید. مثلاً از میانگین نمونه‌ها دیگر، مد نمونه‌ها یا میانه‌ی آن‌ها استفاده کنید. فقط حتماً توجه کنید که برای این کار باید توجیه داشته باشید. انجام هر عملیات بدون توجیه بر روی داده می‌تواند اثرات مخربی روی دقت یادگیری داشته باشد.

۵.۳ نشتی اطلاعات

یکی از اشتباهات بسیار رایج که بیش‌تر توسط کسانی انجام می‌گیرد که به تحلیل اکتشافی مسلط نیستند، نشتی اطلاعات[۱۸] است. منظور از نشتی اطلاعات، استفاده از اطلاعاتی برای ساخت مدل یادگیری ماشین است که از آینده به دست آمده‌اند. شما قرار است از یک مدل یادگیری ماشین برای پیش‌بینی استفاده کنید و در واقع می‌خواهید در مورد رخدادی در آینده صحبت کنید. پس باید حتماً متوجه باشیم که کدام یک از ویژگی‌ها قبل از ثبت نتیجه به دست آمده است و فقط و اکیداً فقط این اطلاعات ارزش آماری دارند.
در این داده‌ها تعدادی از داده‌ها دچار نشتی اطلاعات هستند و شما باید آن‌ها را پیدا کنید و ستون‌های مربوطه را حذف کنید. البته که ممکن است این کار در این تمرین ساده به نظر برسد ولی در حالت کلی و در مسائل پیچیده‌تر تشخیص آن به مراتب مشکل‌تر است و به همان مراتب مهم‌تر.
بعد از خواندن این بخش شما باید بتوانید بر اساس استدلال خود، انواع داده‌ی ستون‌های این فایل‌ها را تشخیص دهید و تغییرات لازم را روی آن‌ها اعمال کنید. ما شما را ملزم به اجرای تغییرات خاصی نمی‌کنیم و شما باید بتوانید بر اساس نیاز آن‌ها را انجام دهید. دقت کنید که بخشی از نمره‌ی این تمرین برای صحت مدل در نظر گرفته شده است و اگر نتوانید دقت کافی را در خروجی کسب کنید، مطمئناً نمره‌ی آن را دریافت نخواهید کرد. البته در مورد این نمره خیلی نگران نباشید زیرا حداقل‌های انتخابی خیلی سخت نیستند.

۴ تحلیل اکتشافی داده

در این بخش قرار است ابتدا با برخی آماره‌ها[۱۹] آشنا شویم و با رسم تعدادی نمودار با ماهیت داده بیش‌تر آشنا شویم. عملیاتی که در این بخش یاد می‌گیرید در مورد هر داده‌ای می‌تواند مفید باشد.

۱.۴ آماره‌های مهم

آماره‌ها در واقع توابعی هستند از فضای آماری داده‌ها به اعداد حقیقی که پارامتری از داده را محاسبه می‌کنند. در اولین برخورد با یک داده باید آماره‌های متنوع ولی ساده‌ای را در مورد آن به دست آوریم و تحلیل کنیم. مهم‌ترین آماره‌ها میانگین، واریانس، مد و میانه هستند. امید است شما برای دانستن معانی این آماره‌ها نیاز به یادآوری نداشته باشید ولی اگر این گونه نیست، گوگل پاسخ‌گوی سؤالات شماست!
به عنوان تمرین، آماره‌ها فوق را برای برای دقیقه، ساعت و روز دریافت اعلان‌ها محاسبه کرده و نتایج حاصله را تحلیل کنید. منظور از تحلیل این است که برای مثال بیان کنید وقتی میانگین به این مقدار است یعنی احتمالاً چه گزاره‌هایی در مورد داده درست است؟ آماره‌های گفته شده را برای هر کاربر به صورت جداگانه هم محاسبه کنید.

۲.۴ نمودارهای مهم

با وجود این که آماره‌ها، اعداد بسیار مهمی در کار با داده هستند اما آن‌ها روح زنده‌ی نمودارها را با خود ندارند. در این بخش قرار است با رسم نمودارهای مختلف داده را ارزیابی کنیم. نمودارهای نام‌برده در هر بند را برای داده‌ی خواسته شده رسم کنید و نتایج آن‌را تحلیل کنید. برای راهنمایی در مورد نحوه‌ی رسم نمودارها به بخش بعدی که توضیحاتی در مورد چند کتاب‌خانه‌ی مفید پایتون است مراجعه کنید. تا زمانی که نمودارهای رسم شده صحیح باشد شما مقید به استفاده از کتاب‌خانه‌ی خاصی نیستید ولی بهترین پیش‌نهادات ما در زیر آورده شده است.

---

[۱۷] Missing Value
[۱۸] Information Leakage
[۱۹] Statistics

نمودار پراکندگی نمودار پراکندگی[۲۰] در هر بعد خود یک متغیر را در نظر می‌گیرد و نقاط نمودار را از روی داده بر اساس مقادیر آن متغیرها پیدا می‌کند. مثلاً فرض کنید داده‌ای در مورد وزن، تاریخ ساخت و قیمت محصولات یک خط تولید دارید. اگر نمودار پراکندگی وزن و تاریخ را رسم کنیم، هر نقطه معرف یک کالای تولید شده است و مختصات آن از روی وزن و تاریخ آن پیدا شده است. در این تمرین شما باید نمودار پراکندگی دوبعدی N1 و N2 را برای کاربران رسم کنید و آن را تحلیل کنید. اگر تعداد داده‌ها، نمایش را مشکل می‌کند از آن به صورت تصادفی نمونه‌برداری کرده و نمایش دهید.

نمودار جعبه‌ای برای درک نمودار جعبه‌ای[۲۱] لازم است مفاهیم میانه و چارک را از آمار مرور کنید. در نمودار جعبه‌ای چارک‌های اول و سوم و میانه به همراه داده‌ی کمینه و بیشینه رسم می‌شود. این نمودار شهود بسیار خوبی در مورد پراکندگی مقادیر داده می‌دهد. لطفاً نمودار جعبه‌ای ویژگی N3 را رسم کنید و آن را تحلیل کنید.

نمودار توزیع نمودار توزیع[۲۲]، کاربردی مشابه نمودار جعبه‌ای دارد ولی اطلاعات بیش‌تری را در خود دارد. این نمودار می‌تواند شما را در انتخاب مدل درست هم یاری کند. برای این تمرین شما باید نمودار توزیع کلمات به کار رفته در اعلان‌ها را رسم کنید و با استفاده از آن وضعیت این کلمات را بررسی کنید و توضیح دهید.

نمودار میله‌ای نمودارهای میله‌ای[۲۳]، تقریبی از نمودار توزیع یا فراوانی هستند. برای داده‌های پیوسته نمودار توزیع بهترین گزینه است ولی برای داده‌های عددی گسسته نمودار میله‌ای بهتر است. نمودار میله‌ای مربوط به دقیقه‌ی دریافت اعلان را رسم کرده و تحلیل کنید.

نمودار دایره‌ای نمودار دایره‌ای[۲۴] سهم هر دسته را نشان می‌دهد و بنابراین برای مشخصه‌های دسته‌ای مناسب است. برای این تمرین شما باید نمودار دایره‌ای متغیر دسته‌ای فایل notif.txt را رسم کرده و تحلیل کنید.

۳.۴ برخی کتاب‌خانه‌های کاربردی

دانش کافی از کتاب‌خانه‌ها، قدرت یک برنامه‌نویس در استفاده‌ی درست از یک زبان برنامه‌نویسی را مشخص می‌کند. در ضمن می‌تواند شما را در رسیدن به یک کد بهینه و خوب یاری کند چرا که بیش‌تر کدهای این کتاب‌خانه‌ها در بهترین حالت نوشته شده‌اند. در این جا لیست کوتاهی از کتاب‌خانه‌هایی که با آن‌ها کار می کنید، آورده شده است. یکی از اهداف تمرین این است که شما خودتان بتوانید توابع مورد نظر را از این کتاب‌خانه‌ها استخراج کرده و استفاده کنید. به همین دلیل توضیحی در مورد توابع آن‌ها داده نشده است. پیشنهاد همیشه‌ی ما در این موارد گوگل است.

کتاب‌خانه‌ی NumPy برای محاسبات علمی کاربرد بسیار زیادی دارد. عنصر اصلی آن ماتریس اعداد است و امکانات بسیار زیادی برای کار با آن‌ها و اعمال توابع روی آن‌ها فراهم می‌کند. این کتاب‌خانه پایه‌ی خیلی از عملیات‌های ریاضی نه تنها در حوزه‌ی یادگیری ماشین بلکه به طور کلی در کار با اعداد است.

کتاب‌خانه‌ی Pandas برای آنالیز و اعمال تغییرات روی داده‌های بزرگ کاربرد بسیار دارد. بسیاری از عملیات پیش‌پردازش توسط این کتاب‌خانه انجام می‌شود.

کتاب‌خانه‌ی scikit-learn این کتاب‌خانه نیز امکانات بسیار زیادی برای اعمال تغییرات روی داده‌ها و همچنین مهم‌تر از همه، اعمال الگوریتم‌های یادگیری ماشین روی آن‌ها دارد. بسیاری از الگوریتم‌های یادگیری ماشین پیاده‌سازی شده و آماده برای اعمال روی داده‌ها هستند.

کتاب‌خانه‌ی Matplotlib کاربرد اصلی این کتاب‌خانه برای رسم نمودار است. اکثر انواع نمودار توسط این کتاب‌خانه پشتیبانی می‌شوند. انتخاب و طراحی یک الگوریتم یادگیری مناسب، نیازمند داشتن درک درست از داده است. در بسیاری از موارد، مخصوصاً وقتی حجم بالایی از دیتا موجود باشد به راحتی این درک حاصل نمی‌شود، اما استفاده از نمودارها می‌تواند به شناخت بهتر داده‌ها کمک کند.

کتاب‌خانه‌ی SciPy این کتاب‌خانه، توابع فراوانی در مورد کارهای ریاضی و مهندسی و هم‌چنین کار با انواع داده‌ی تنک دارد. پیشنهاد می‌کنیم به این کتاب‌خانه هم سری بزنید.
در ادامه یک کتاب‌خانه دیگر را معرفی می‌کنیم که احتمالاً نام آن را تا به حال نشنیده‌اید اما هنگام اجرای الگوریتم‌های سنگین که حلقه‌های طولانی دارند، به شدت کاربرد دارد.

کتاب‌خانه‌ی tqdm نام این کتابخانه از واژه تقدم گرفته شده است که به معنای پیش‌روی یا پیش‌رفت است. می‌توانید میزان پیش‌رفت حلقه‌های کد خود را با استفاده از این کتابخانه به صورت بهینه مشاهده کنید و از این بابت از اجرای درست و میزان زمان اجرای کد خود آگاه شوید. استفاده از این کتابخانه اجباری نیست ولی توصیه می‌شود. برای اطلاعات بیش‌تر به صفحه‌ی این کتاب‌خانه رجوع کنید.

۵ مدل‌های رگرسیونی

ساده‌ترین مدل در کار با داده‌ها مدل‌های خطی[۲۵] هستند که البته با وجود سادگی آن‌ها نباید دست کم گرفته شوند. اگر یک مدل خطی برای داده‌ای کار بکند، یک جواب مناسب است چرا که در زمان کم خروجی را خواهید داشت و همه‌ی این مدل‌ها از رابطه‌های ساده‌ی ریاضی تبعیت می‌کنند. از مدل‌های خطی مهم مدل **رگرسیون خطی**[۲۶] است. در این مدل یک خروجی داریم که قرار است به صورت ترکیب خطی توابعی از مشخصه‌های ورودی تعیین شود و مدل روی ضرایب این ترکیب خطی به دست می‌آید.
این مدل در کلاس درس به خوبی بسط داده شده است و در این جا فقط کمی در مورد ضرورت آن صحبت می‌کنیم. مدل‌های رگرسیون خطی که در درس آمده‌اند، به راحتی می‌توانند داده‌های خطی را یاد بگیرند و پیش‌بینی کنند اما روش رگرسیون به صورت مدل ساده‌ی خطی یک فرض و گاهی ایراد بزرگ دارد: این روش تصور می‌کند که داده به صورت پیوسته است یعنی مقادیر خروجی می‌تواند هر عددی حقیقی‌ای باشد که در این جا ابداً فرض درستی نیست!

---

[۲۰] Scatter Plot
[۲۱] Box Plot
[۲۲] Distribution Plot
[۲۳] Bar Plot
[۲۴] Circular Plot
[۲۵] Linear Models
[۲۶] Linear Regression

داده‌های ما تنها دو حالت خروجی صفر و یک دارند اما با یک تبدیل ساده می‌توان این داده‌های صفر و یکی را به محور اعداد حقیقی برد. تبدیلی مشهور به **تبدیل لاجستیک**[۲۷] این کار را انجام می‌دهد و حالا خروجی جدید به دست آمده را می‌توان با مدل ساده یاد گرفت. در درس با این تبدیل آشنا شدید و می‌توانید در مورد خواص آماری آن در اینترنت بیش‌تر مطالعه کنید. در ضمن، با فرض نرمال بودن داده، مدل انتخاب شده در رگرسیون خطی بهینه خواهد بود. تصور اولیه ما این است که بیش‌تر داده‌ها، نزدیک به نرمال[۲۸] هستند اما این هم جزو اشتباهات متداول است.
در کار با مدل‌های خطی می‌توانید به صورت دقیق‌تر از نمودار چارک به چارک یا Q-Q استفاده کنید. این نمودار می‌تواند یک توزیع را با توزیع نرمال مقایسه کند. این کار ممکن است برخی مقادیر پرت[۲۹] را مشخص کند که کار یادگیری را دچار خطای اضافه می‌کنند. از معیارهای مناسب دیگر برای تست این فرضیه می‌توان به رابطه‌ی کوک[۳۰] و آزمون-فرض‌های نرمالیتی[۳۱] اشاره کرد که خارج از بحث هستند. با این که در این تمرین از این ابزارها استفاده نمی‌کنیم ولی حتماً باید آن‌ها را مد نظر داشته باشید چرا که بخش مهمی از دانش شما در مورد مدل‌های خطی در آینده خواهند بود.

۱.۵ معیارهای ارزیابی یک مدل

در ارزیابی کارآیی یک الگوریتم یادگیری، می‌توان به دو روش کلی اشاره کرد. روش اول استفاده از داده‌های جدید و برچسب‌دار است که با مقایسه‌ی برچسب واقعی با نتیجه‌ی الگوریتم می‌توان معیارهایی مانند precision, recall, f1 score را اندازه گرفت. این روش نیاز به دسترسی به داده‌های تست و برچسب متناظر آن‌ها دارد. توجه کنید که همیشه روش به دست آوردن داده‌های جدید ساده نیست و خیلی از اوقات، تنها داده‌هایی که در دست‌رس شما هستند همین داده‌های آموزش خواهند بود. بنابر این در روشی دیگر استفاده از یک تکنیک در آمار به نام resampling استفاده می‌شود که از همان اطلاعات آموزش برای اهداف تست استفاده می‌شود.
همان طور که در درس هم با آن رو به رو شدید باید داده‌های تست دو ویژگی مهم داشته باشند. اول این که به لحاظ آماری با داده‌های آموزش از یک منشأ و مدل باشند که این با جمع‌آوری درست اطلاعات در همان ابتدا، به راحتی ارضا می‌شود. ویژگی مهم‌تر اما استقلال آماری با فرآیند آموزش است که در نگاه اول با ویژگی اول متناقض است اما می‌توانید این گونه آن را تعبیر کنید که شما نباید جواب سؤالات امتحان را قبل از آن بدانید تا دانش شما در درس درست محک زده شود!
یک روش resampling شکستن ساده‌ی داده‌های در دست‌رس به دو قسمت train و validation است که داده‌ی validation نقش تست را برای ارزیابی مدل ایفا می‌کند. روش دیگربه k-fold cross validation معروف است. در این روش داده‌های در دست‌رس را به k قسمت تقسیم می‌کنند، سپس از k-1 قسمت به عنوان داده train و از یک قسمت به عنوان داده test استفاده می‌شود. این کار آن قدر تکرار می‌شود تا هر کدام از k قسمت یک‌بار به عنوان داده test استفاده شوند. در نهایت از میانگین یا واریانس برای تجمیع نتایج به دست آمده از هربار اجرای الگوریتم استفاده می‌شود. البته که این روش دارای توجیه ریاضی مشخصی نیست و استدلالی برای انتخاب پارامترهای آن وجود ندارد. با این وجود آن، کاربرد آن چنان گسترده شده که کم‌تر کسی به ریشه‌ی ریاضی آن فکر می‌کند.

۲.۵ معیارهای اندازه‌گیری دقت الگوریتم‌های یادگیری

برای اندازه‌گیری دقت الگوریتم‌های یادگیری ماشین، مشخصه‌های مختلفی وجود دارد. در این جا به صورت خاص در مورد مشخصه‌های الگوریتم‌های برچسب‌گذاری باینری[۳۲] صحبت می‌کنیم که دقیقاً خروجی الگوریتم شما خواهد بود. در این مسئله خروجی الگوریتم به صورت صفر و یک و مقدار واقعی هم به صورت صفر و یک است و بنابراین بر اساس تساوی یا عدم تساوی آن‌ها چهار حالت زیر متصور است:

- **True Positive** - تعداد نمونه‌های پیش‌بینی‌شده‌ی درست که مقدار یک دارند. بدین معنا که برچسب واقعی یک است و برچسب پیش‌بینی شده توسط الگوریتم نیز یک است.
- **True Negative** - تعداد نمونه‌های پیش‌بینی شده‌ی درست که مقدار صفر دارند. بدین معنا که برچسبواقعی صفر است و برچسب پیش‌بینی شده توسط الگوریتم نیز صفر است.
- **False Positive** - تعداد نمونه‌های پیش‌بینی شده‌ی اشتباه که مقدار واقعی صفر دارند ولی یک پیش‌بینی شده‌اند. بدین معنا که برچسب واقعی صفر است ولی الگوریتم یادگیری آن را یک پیش‌بینی کرده است.
- **False Negative** - تعداد نمونه‌های پیش‌بینی شده‌ی اشتباه که مقدار واقعی یک دارند ولی صفر پیش‌بینی شده‌اند. بدین معنا که برچسب واقعی یک است ولی الگوریتم یادگیری آن را صفر پیش‌بینی کرده است.

به ماتریسی که درایه‌های آن مقادیر گفته شده را نشان می‌دهد، **ماتریس درهم‌ریختگی**[۳۳] می‌گویند. شما می بایست این ماتریس را هم تشکیل داده و نمایش دهید.
بر اساس تعاریف فوق می‌توان معیارهای زیر را تعریف کرد. از شما انتظار می‌رود برای این تمرین همه‌ی این موارد را محاسبه کرده و گزارش کنید. در ضمن باید حدود حداقلی ذکر شده در انتهای تمرین را نیز به دست آورید تا نمره‌ی صحت مدل را کسب کنید.

Accuracy ساده‌ترین و کلی‌ترین معیار برای اندازه‌گیری، دقت مدل یادگیری است که از تقسیم تعداد نمونه‌هایی که درست پیش‌بینی شده‌اند به کل تعداد نمونه‌ها محاسبه می‌شود. این مقدار باید همیشه بیش‌تر از ۵۰ درصد باشد زیرا در غیر این صورت الگوریتم برچسب‌گذاری معکوس شما بهتر عمل می‌کند. در ضمن باید مقدار خوبی از ۵۰ درصد فاصله داشته باشد، چرا که این درصد به یک معنا این است که شما همه‌ی خروجی‌ها را تصادفی و با انداختن سکه انتخاب کرده‌اید. یک معیار مهم دیگر برای سنجیدن دقت این است که شما باید از نسبت بیشترین بین مقادیر صفر یا یک عمل‌کرد بهتری داشته باشید. مثلاً در نظر بگیرید که خروجی واقعی داده‌ها ۸۰ درصد مقدار یک دارد و ۲۰ درصد مقدار صفر. در این صورت هر الگوریتمی که دقت زیر ۸۰ درصد کسب کند بی‌ارزش است چرا که اگر همه‌ی مقادیر خروجی را ثابت یک می‌کردیم حداقل این عملکرد را می‌گرفتیم!

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (۱)$$

[۲۷] Logistic Function
[۲۸] Gaussian (Normal) Distribution
[۲۹] Outliers
[۳۰] Cook's Distance
[۳۱] Normality Hypotesis Testing
[۳۲] Binary Classification
[۳۳] Confusion Matrix

Precision نسبت تعداد نمونه‌هایی که برچسب یک خورده‌اند و در واقعیت هم برچسب یک دارند به کل تعداد نمونه‌ها که یک پیش‌بینی شده‌اند. این معیار نشان می‌دهد وقتی که مدل یادگیری مقدار یک گزارش می‌کند چقدر دقت دارد. این معیار زمانی خیلی مهم می‌شود که یک اعلام شدن یک متغیر حیاتی باشد. مثل تشخیص جرم در یک سیستم جرم‌شناسی که بسیار مهم است که اگر جرمی تشخیص داده شد، واقعا آن جرم وجود داشته باشد و اگر همه‌ی جرم‌ها شناسایی نشد چندان مشکلی نیست.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{۲}$$

Recall نسبت تعداد نمونه‌های که برچسب یک خورده‌اند و در واقعیت هم برچسب یک دارند به کل تعداد نمونه‌هاست که در واقعیت مقدار یک دارند. این معیار نشان می‌دهد که مدل یادگیری چه قدر در پیدا کردن همه‌ی نمونه‌ها یک کارایی دارد. مشابه معیار قبل این معیار هم در برخی مواقع حیاتی است. در حالاتی که تشخیص نمونه‌های یک خیلی مهم است. می‌توانید مثال تشخیص بیماری را در نظر بگیرید که بسیار مهم است بیماران (نمونه‌های یک) تشخیص داده شوند. نام‌های دیگر این معیار حساسیت[۳۴] ، نرخ ضربه[۳۵] و نرخ True Positive[۳۶] است.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{۳}$$

Specificity این معیار که به نرخ True Negative[۳۷] و انتخاب‌کنندگی[۳۸] هم مشهور است نسبت نمونه های صفر واقعی را که به صورت درست بچسب صفر خورده‌اند را نشان می‌دهد. برای مثال میزان افراد سالمی که به درستی در نداشتن بیماری شناسایی شده‌اند.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{۴}$$

F1 Score این معیار میانگین هم‌ساز[۳۹] Precision و Recall است. به همین دلیل این معیار هر دو مقدار FP و FN را در نظر می‌گیرد. بنابراین اگر هزینه‌ی FP و FN در مسئله‌ی ما یکسان نباشد، این معیار به کار خواهد آمد. در این معیار به این دلیل از میانگین هم‌ساز استفاده می‌شود که اثر نرخ این مقادیر بیش‌تر از اثر مقدارشان مهم است که در این میانگین به خوبی دیده می‌شود.

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} \tag{۵}$$

۳.۵ منحنی مشخصه‌ی اپراتور دریافت کننده یا ROC

این منحنی[۴۰] مشهور ریشه در وقایع جنگ جهانی دوم دارد که برای ارزیابی عمل‌کرد اپراتورهای رادار استفاده می گشت و بعداً وارد روان‌شناسی و در نهایت یادگیری ماشین شده است. همان‌گونه که دیدید خروجی مدل رگرسیون لاجستینک یک عدد بین صفر و یک است و شما باید با انتخاب یک آستانه[۴۱]، برچسب صفر و یک را از یک‌دیگر جدا کنید. بر اساس این که این آستانه کجا قرار بگیرد معیارهای معرفی شده در بخش قبل مقادیر متفاوتی پیدا خواهند کرد. این نمودار رابطه‌ی بین نرخ True Positive و True Negative را برای مقادیر مختلف آستانه نشان می‌دهد.
شما برای رسم این نمودار باید برای هر مقدار صفر غلط پنج برابر جریمه‌ی بیش‌تر نسبت به هر مقدار یک غلط در نظر بگیرید. سپس مقادیر مختلف آستانه را تغییر دهید و مقدار مساحت زیر نمودار[۴۲] یا AUC را اندازه بگیرید. بهترین AUC بهترین آستانه را معرفی می‌کند. در مورد این که چرا این معیار مناسبی است می‌توانید خودتان کمی فکر کنید و اگر نیاز به کمک داشتید از صفحه‌ی ویکی‌پدیا یا استاد درس کمک بگیرید. این لینک هم یک نمودار تعاملی جالب در این موضوع ارائه می‌کند.
از شما انتظار می‌رود بتوانید آستانه‌ی درست را پیدا کرده و در ضمن نمودار مشخصه‌ی درست را هم رسم کنید. مدل نهایی شما باید بر اساس خروجی این قسمت تنظیم شده باشد.

۶ قسمت امتیازی

بعد از نسخه‌ی ۵ سیستم‌عامل اندروید، مدیریت اعلان‌ها برای هر برنامه‌ی کاربردی فعال شد و با رشد این سیستم‌عامل، قابلیت‌های بیشتری مانند Notification Category به آن اضافه شد که کاربر می‌تواند انتخاب کند چه نوع اعلانی از طرف هر برنامه‌ی کاربردی نمایش داده بشود. ساکت کردن اعلان، بدترین اتفاق ممکن برای سیستم اعلان یک برنامه‌ی کاربردی است زیرا توسعه‌دهندگان برنامه، دیگر نمی‌توانند اعلان‌های خود را به کاربر نمایش دهند. فرض کنید در این تمرین علاوه بر حالت کلیک یا رد اعلان، حالت ساکت کردن اعلان برای برنامه‌ی کاربردی را هم داشته باشیم. در انتهای گزارش خود، در قسمتی جداگانه بررسی کنید که آیا مدلی که برای حالت باینری (رد یا کلیک) توسعه دادید، در حالت سه‌تایی (رد، کلیک یا ساکت کردن) هم کار می‌کند؟ اگر نه، چگونه می‌توانید با کمترین تغییر، مدل خود را برای این حالت نیز آماده کنید؟ دقت کنید که **به هیچ عنوان** از ابتدا مدل خود را بر مبنای اینکه برای حالت سه‌تایی هم کار کند طراحی نکنید چرا که هدف این قسمت، بررسی تحلیل شما برای حالتی است که از ابتدا بر اساس فرض خروجی باینری، مدل خود را طراحی کرده‌اید.

---

[۳۴] Sensitivity
[۳۵] Hit Rate
[۳۶] True Positive Rate (TPR)
[۳۷] True Negative Rate (TNR)
[۳۸] Selectivity
[۳۹] Harmonic Mean
[۴۰] Receiver Operating Characteristic
[۴۱] Threshold
[۴۲] Area Under Curve

۷ راهنمای تحویل تمرین

برای تحویل این تمرین، اول لازم است گزارشی از کارهای انجام گرفته در مورد تمرین و پاسخ سؤالات آمده در متن را آماده کنید به طوری که نشان دهد در مورد موارد مختلف خوب آن‌ها را یاد گرفته‌اید و توانسته‌اید محاسبات لازم را انجام دهید. سؤالات به صورت واضح در متن آمده‌اند و پاسخ به هر کدام از آن‌ها نمره‌ای جداگانه دارد. در گزارش خود توضیح مختصری هم در مورد الگوریتم خود اضافه کنید ولی لازم نیست قطعه کدها را در آن قرار دهید. اگر توضیحات تکمیلی‌تری برای هر قسمت دارید از آن‌ها استقبال می‌کنیم! گزارش در قالب یک فایل PDF[۴۳] آماده شود و می‌تواند به زبان فارسی یا انگلیسی و در هر تعداد صفحه باشد. بخش مهمی از نمره‌ی تمرین از بررسی گزارش شما، صحت مطالب آن، تمیزی و کامل بودن آن داده می‌شود. تهیه‌ی گزارش در قالب LaTeX امتیاز بیش‌تری دارد اما حداکثر می‌تواند امتیاز گزارش شما را کامل کند و نمره‌ی کد و مدل مستقلاً محاسبه می‌شود. نام فایل report.pdf باشد.

در ضمن باید کد پایتونی که برای این تمرین نوشته اید را در قالب یک فایل با فرمت main.py ارائه کنید. تمام عملیات در این فایل باید با اجرای آن خودکار باشد. کد شما باید فایل‌ها را خوانده و موارد خواسته شده را محاسبه یا رسم نماید و در نهایت برچسب‌های خروجی را برای تست دل‌خواه به همان صورت نمونه، خروجی دهد.

می‌توانید به جای گزارش در قالب متنی از گزارش در قالب Jupyter (IPython) Notebook هم استفاده کنید. نام فایل را به صورت main.ipynb قرار دهید. در این صورت ارائه‌ی فایل کد به صورت جداگانه لازم نیست. توجه کنید که باید تمام موارد خواسته شده در گزارش در همین فایل موجود باشد. اگر لازم می‌دانید می‌توانید به صورت ترکیبی هم گزارش را تهیه کنید. فقط حتماً توجه کنید که فایل PDF معیار اصلی است و باید در آن ذکر کنید چه مواردی را در نوت‌بوک آورده‌اید.

یک فایل دیگر نیز در صورت تمرین خواسته شده است که آن را نیز جداگانه با همان دستور باید آماده کنید و به نام گفته شده ذخیره کنید. مدل شما روی یک داده‌ی تست دیگر بررسی خواهد شد. پس حتماً سعی کنید کد را درست بزنید که وابسته به داده نباشد. شکل داده‌ی تست جدید دقیقاً با داده‌ی تست داده شده هم‌خوانی دارد. این تست فقط برای سنجش کد شما در معیار واقعی به کار می‌رود و نمره‌ای نخواهد داشت، جز در حالتی که کد شما تفاوت قابل ملاحظه‌ای بین دو نمونه‌ی تست (در اختیار شما و در اختیار ما) داشته باشد. در ضمن خروجی داده‌ی تست را هم به صورت گفته شده با نام output.csv ذخیره کنید. مدل شما باید از حالت‌های بدیهی گفته شده در تمرین عمل‌کرد بهتری داشته باشد. یعنی از حالتی که خروجی‌ها تصادفی انتخاب شده‌اند و یا این که همه‌ی آن‌ها به شکل یکسانی مقدار صفر یا یک دارند بهتر عمل کند. در غیر این صورت نمره‌ی صحت مدل را دریافت نخواهید کرد.

همه‌ی فایل‌های گفته شده در بالا را در قالب یک فایل فشرده‌ی زیپ[۴۴] با نام شماره‌ی دانشجویی خود در سامانه‌ی کوئرا، در بخش **تمرین عملی سری دوم** بارگذاری کنید.

سؤالات مربوط به این تمرین را می‌توانید در Piazza بپرسید. سؤالات شما در اسرع وقت پاسخ داده خواهد شد.

---

[۴۳] Portable Document Format

[۴۴] Zip Compressed Format