دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

درس تحلیل کلان داده ها

استاد درس جناب آقای دکتر چهرقانی

(تمرین اول)

محسن عبادپور | ۴۰۰۱۳۱۰۸۰ | m.ebadpour@aut.ac.ir

نیمسال دوم سال تحصیلی ۱۴۰۲-۱۴۰۱

فهرست پاسخ ها

فهرست مطالب

بخش اول: Map Reduce

در این بخش پاسخ‌های مربوط به مبحث Map Reduce آورده شده است که مشتمل بر دو سوال می‌باشد که سوال اول تشریحی/توصیفی بوده و سوال دوم بصورت پیاده می‌باشد.

سوال اول: ضرب ماتریس ها

الف) توضیح عملکرد توابع

انجام ضرب ماتریس با الگوی Map Reduce میتواند در دو حالت انجام بپذیرد که در یکی صرفا یک فاز(تکرار) از الگوی Map Reduce عملیاتی می‌شود و در دیگری در دو فاز(تکرار) از اجرای الگوی Map Reduce اجرا می شود. طبق متن سوال، حالتی را در نظر میگیریم که در آن دو فاز از اجرای الگوریتم مذکور اجرا می‌شود. فرض می کنیم که ضرب ماتریس بصورت M1xM2 انجام می‌شود و سطر های ماتریس اول در ستون‌های ماتریس دوم ضرب می‌شود.

میدانیم ضرب جبری دو ماتریس به این صورت انجام می‌شود که عناصر سطر اول از ماتریس اول در عناصر متناظر در ستون اول از ماتریس دوم ضرب شده و سپس با یک دیگر جمع شده و عنصر اول از سطر اول ماتریس جواب را حاصل می کند و این روند ادامه می‌یابد. در الگوی دو فازی از Map Reduce، در فاز اول فرآیند ضرب عناصر در یکدیگر اجرا می شود و در فاز دوم نیز جمعِ ضرب های انجام شده انجام می شود؛ حال در زیر بصورت کامل و با مثال توضیح داده می شود.

در تکرار اول تابع map هر عنصر از هر ماتریس را ورودی گرفته و آنان را بصورتی که ضرب می شوند در وجه(بُعد) مشترک خروجی می دهد. به عبارتی دیگر، کلید خروجی برابر با index مشترک بوده و مقدار خروجی یک tuple سه تایی می باشد که به ترتیب مقدار عنصر، اندیس بعد غیر مشترک و نام ماتریس(که مشخص کند که ماتریس چپ است یا راست) می باشد. برای مثال عنصر $M1_{ij}$ در $M2_{jk}$ ضرب می شود که j وجه مشترک می باشد پس آن کلید خروجی هر یک از عناصر بوده و برای اولی مقدار خروجی تابع $(M1,i,M1_{ij})$ و برای دومی $(M2,k,M2_{jk})$ را حاصل می‌کند.

در تکرار اول تابع reduce عمل ایجاد ارتباط و اتصال را بین عناصر ضرب شونده دو ماتریس انجام می‌دهد بدین صورت که key-value های map قبلی را ورودی میگیرد(به ازای کلید های یکسان که همان وجه مشترک j بود) و سپس ضرب مقداری دو به دو را انجام داده و حال key-value جدید را بطوری که j کلید بوده و یک tuple سه تایی حاوی اندیس های هدف(سطر و ستون در ماتریسِ جواب) و مقدار ضرب را باز می گرداند. در ادامه‌ی مثال قبل، کلید خروجی تابع reduce برابر با j بوده و مقادیر $(i,k,M1_{ij}xM2_{jk})$ تولید میشود.

در تکرار دوم تابع map تمامی خروجی‌های تابع reduce در تکرار قبل را ورودی گرفته و آن را بدین گونه تغییر می‌دهد که کلید خروجی تابع map اندیس های هدف و مقدار ضرب نیز مقدار هر کدام از کلیدها باشد. در ادامه‌ی مثال قبل، کلیدهای خروجی برابر با (i,j) بوده و مقدار نیز $M1_{ij}xM2_{jk}$ می‌باشد.

در تکرار دوم و تابع reduce نیز عملیات گروه‌بندی انجام شده و خروجی آن یک key-value میباشد که key یک tuple دو تایی می باشد که نشان‌دهنده‌ی اندیس عنصر در ماتریس جواب است و value نیز حاصل جمع تمام ضرب‌های انجام شده برای بدست آوردن آن عنصر می باشد. در ادامه‌ی مثال قبل، ورودی تابع به ازای تمامی (i,j) های یکسان و مقدار برابر با $M1_{ij}xM2_{jk}$ می باشد و key خروجی نیز (i,j) بوده و value خروجی نیز مجموع تمام $M1_{ij}xM2_{jk}$ معادل با آن اندیس می باشد.

ب) خروجی تابع map در تکرار اول

در این قسمت برای دو ماتریس داده شده mapping در تکرار اول انجام می شود؛ طبق توضیحات فوق و ضرب عنصر $M1_{ij}$ در $M2_{jk}$ فرآیند نگاشت را انجام می‌دهیم:

$M1: \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$ => j=1 (ستون اول) : (M1,1,2), (M1,2,1)

$M1: \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$ => j=2 (ستون دوم) : (M1,1,3), (M1,2,2)

$M2: \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$ => j=1 (ردیف اول) : (M2,1,1), (M2,2,4)

$M2$: $\begin{bmatrix}1 & 4\\2 & 3\end{bmatrix}$ => j=2 (ردیف دوم) : (M2,1,2), (M2,2,3)

ج) خروجی تابع reduce در تکرار اول

در این قسمت ابتدا فرآیند Shuffling را انجام میدهیم که نتیجه بصورت زیر حاصل می شود:

j=1: (M1,1,2), (M1,2,1), (M2,1,1), (M2,2,4)

j=2: (M1,1,3), (M1,2,2), (M2,1,2), (M2,2,3)

حال فرآیند کاهش را طبق توضیحات ارائه شده و رابطه‌ی $(i,k,M1_{ij}xM2_{jk})$ انجام می‌دهیم تا عملیات های ضرب بصورت زیر انجام شود:

j=1: (1,1,2*1), (1,2,2*4), (2,1,1*1), (2,2,1*4)

➔ (1,1,2), (1,2,8), (2,1,1), (2,2,4)

j=2: (1,1,3*2), (1,2,3*3), (2,1,2*2), (2,2,2*3)

➔ (1,1,6), (1,2,9), (2,1,4), (2,2,6)

د) خروجی تابع map در تکرار دوم(فعالیت بیشتر)

حال هر کدام از خروجی های مرحله قبل را که بصورت $j:(i,k,M1_{ij}xM2_{jk})$ می باشد ورودی گرفته و خروجی را بصورت $(i,k):(M1_{ij}xM2_{jk})$ و طبق توضیحات مشروح فوق حاصل می‌کنیم:

j=1: (1,1,2), (1,2,8), (2,1,1), (2,2,4) (ورودی)

⇨ (1,1):2, (1,2):8, (2,1):1, (2,2):4 (خروجی)

j=2: (1,1,6), (1,2,9), (2,1,4), (2,2,6) (ورودی)

⇨ (1,1):6, (1,2):9, (2,1):4, (2,2):6 (خروجی)

ه) خروجی تابع reduce در تکرار دوم(فعالیت بیشتر)

حال هر کدام از خروجی های مرحله قبل را که بصورت $(i,k):(M1_{ij}xM2_{jk})$ می‌باشد ورودی گرفته و خروجی را بصورت $(i,k):(\sum M1ijxM2jk)$ و طبق توضیحات مشروح فوق حاصل می‌کنیم. البته قبل از آن shuffling را انجام میدهیم که بصورت زیر حاصل می شود:

(1,1): 2, 6 (1,2): 8, 9

(2,1): 1, 4 (2,2): 4,6

حال عملیات تابع کاهش را انجام و با جمع کردن مقادیر حاصل ضرب ماتریس را انجام می‌دهیم:

(1,1): 2 + 6 = 8 (1,2): 8 + 9 = 17

(2,1): 1 + 4 = 5 (2,2): 4 + 6 = 10

ماتریس حاصل از ضرب ماتریس ها بصورت (1,1) = 8, (1,2) = 17, (2,1) = 5, (2,2) = 10 بدست می آید که فرم ماتریسی آن را بصورت زیر میتوان نشان داد:

$$\begin{bmatrix} 8 & 17 \\ 5 & 10 \end{bmatrix}$$

سوال دوم: تبادل کانال‌های شبکه‌های مجازی

پیاده سازی این سوال کاملا با الگوی Map Reduce و با کتابخانه‌ی pyspark انجام شده و در بستر google colab اجرا و فایل ژوپیتر از آن خروجی گرفته شده و در پیوست قابل ملاحظه است(کامنت گذاری انجام شده است). برای اطمینان از حصول صحیح پیاده‌سازی و نتایج حاصل، یک پیاده‌سازی دیگر با الگوی خطی (دستی و با استفاده از حلقه) انجام شده و نتایج با آن تطبیق داده شده است که حاکی از صحت پیاده سازی دارد.

تابع map بدین صورت کار می کند که هر سطر از دیتاست را ورودی گرفته و جفت کلیدهایی را تولید می کنید که در آن key خروجی ID کانال تبلیغ کننده بوده و مقدار آن نیز یک می‌باشد(یک تبلیغ انجام داده است). برای مثال اگر سطری از دیتاستی بصورت زیر باشد، خروجی متناظر در مقابل هر یک آورده شده است:

1: 2, 3 => 2:1, 3:1

2: 1, 3 => 1:1, 3:1

1: 4 => 4:1

3: 1, 2 => 1:1, 2:1

4: 1 => 1:1

حال اگر shuffling را انجام دهیم نتیجه بصورت زیر حاصل می‌شود:

1: 1,1,1

2: 1,1

3: 1,1

4: 1

تابع reduce نیز به این صورت کار می‌کند که تمامی pair های key-value را دریافت می‌کند که کلیدشان یکسان (ID یک کانال) بوده و مقادیرشان همگی یک است. این تابع تمامی یک ها را با هم جمع می‌کند که نشان دهد کانال متناظر چند بار تبلیغ انجام داده است تا در نتیجه تعداد تبادل آن کانال حاصل شود(طبق گفت و گوی طرح شده در گروه تلگرامی و بررسی دیتاست این فرض شده است که: تعداد تبلیغ شده = تعداد تبلیغ کرده = تعداد تبادل). در ادامه‌ی مثال فوق، تابع reduce به صورت زیر خروجی خواهد داد:

1: 1+1+1 = 3

2: 1+1 = 2

3: 1+1 = 2

4: 1 = 1

الف) پنج کانال با بیشترین تبادل

پس از اجرای الگوی نگاشت–کاهش با توضیحات فوق بروی دیتاست داده شده، تعداد تبادل هر یک از کانال‌ها بدست می‌آید که پس از مرتب‌سازی لیست حاصل بر اساس تعداد تبادل هر کانال، نتیجه خواسته شده حاصل می‌شود که بصورت زیر آورده شده است:

```
=> Top five TV Channels:
1. Channel with ID # 859 has 1933 exchanges
2. Channel with ID #5306 has 1741 exchanges
3. Channel with ID #2664 has 1528 exchanges
4. Channel with ID #5716 has 1426 exchanges
5. Channel with ID #6306 has 1394 exchanges
```

ب) تعداد تبادل کانال‌های خواسته شده

از خروجی مرحله قبل استفاده کرده و با اعمال فیلتر و شرط گذاری در مرحله مرتب‌سازی، تعداد تبادل سه کانال خواسته شده را بدست می‌آوریم که بصورت زیر می‌باشد:

```
-> Asked TV Channels:
Channel with ID #1748 has 130 exchanges
Channel with ID #3469 has 119 exchanges
Channel with ID #5633 has  30 exchanges
```

بخش دوم: Association Rule

در این بخش پاسخ‌های مربوط به مبحث Association Rule آورده شده است که مشتمل بر دو سوال می‌باشد که سوال اول تشریحی/توصیفی بوده و سوال دوم بصورت پیاده می‌باشد.

سوال اول: محاسبه احتمالاتی آیتم‌های پر تکرار

برای حل این سوال، امید ریاضی حضور هر یک از ۱۰ آیتم را در سبدها محاسبه می‌کنیم که بصورت زیر حاصل می‌شود؛ از جایی که N خیلی بزرگ است، میتواند تقریب ها را بصورت تساوی نوشت:

$E[\text{item 1}] = N * p(1/1) = N > 1\%*N,$ $E[\text{item 2}] = N * p(1/2) \sim N/2 = N2 > 1\%*N$

$E[\text{item 3}] = N * p(1/3) \sim N/3 = N/3 > 1\%*N,$ $E[\text{item 4}] = N * p(1/4) \sim N/4 = N/4 > 1\%*N$

$E[\text{item 5}] = N * p(1/5) \sim N/5 = N/5 > 1\%*N,$ $E[\text{item 6}] = N * p(1/6) \sim N/6 = N/6 > 1\%*N$

$E[\text{item 7}] = N * p(1/7) \sim N/7 = N/7 > 1\%*N,$ $E[\text{item 8}] = N * p(1/8) \sim N/8 = N/8 > 1\%*N$

$E[\text{item 9}] = N * p(1/9) \sim N/9 = N/9 > 1\%*N,$ $E[\text{item 10}] = N * p(1/10) \sim N/10 = N/10 > 1\%*N$

طبق محاسبات فوق، قابل مشاهده است همه‌ی آیتم‌ها، آیتم های پرتکرار بوده و support بالاتر از حد آستانه ۱٪ را دارند لذا از همه‌ی آنان برای نوشتن قوانین انجمنی بایستی استفاده نمود. حال برای هر آیتم دلخواهی مانند i میتوان نشان داد که قانون انجمنی i -> j به ازای هر j از ۹ آیتم باقیمانده معتبر است چرا که j نیز پر تکرار بوده و از حد آستانه بزرگتر است لذا میتوان تمام ترکیب های ممکن دوتایی را قانون انجمنی دانست که تعداد بسیار بالایی می باشد(طبق همین محاسبات میتوان برای قوانین ۳و۴و... تایی هم این مورد را نشان داد). طبق این نکته که تعداد قوانین انجمنی بالقوه بسیار بالا بوده و طبق حد آستانه تعیین شده، همه‌ی آیتم‌ها پرتکرار حاصل شده و باهم وابستگی بالایی دارند، میتوان گفت که حضور هر آیتم یا مجموعه آیتم‌های دلخواهی، حضور هر آیتم یا مجموعه آیتم های دیگری را نتیجه می‌دهد که این ارزشمند نبوده و بدردبخور نمی‌باشد لذا میتوان نتیجه گرفت که در همچین شرایطی قوانین انجمنی

استخراج شده ارزشمند و جالب نبوده و استفاده‌ی عملیاتی آن منطقی نمی‌باشد لذا نمیتوان از آن اطلاعات/رابطه مفید دریافت کرده و استفاده‌ی تجاری/صنعتی داشت چرا که کار بیهوده‌ای انجام شده و هیچ نکته‌ی آموزشی استخراج نشده است.

سوال دوم: تشخیص تراکنش‌های جعلی کارت‌های اعتباری

برای پیاده‌سازی این سوال از زبان برنامه نویسی پایتون استفاده شده و تمامی بخش ها بصورت کامل و دستی پیاده سازی شده است. اجرا در بستر گوگل کولب انجام شده و فایل ژوپیتر متناظر خروجی گرفته شده و در پیوست قابل ملاحظه است. ضمنا سلول های فایل ژوپیتر، کامنت گذاری شده است.

الف) پیش‌پردازش‌ها و گسسته‌سازی

همانطور که قابل انتظار بوده و از بررسی مجموعه داده مشخص است، خروجی مولفه‌های اصلی هم رنج نبوده و گستره متفاوتی دارند لذا بایستی آن ها را نرمال کرد. در گام اول تمام ستون های مجموعه داده را بارگزاری کرده و سپس ستون Time را حذف می‌کنیم. (زمان انجام تراکنش هیچگونه همبستگی/وابستگی با جعلی بودن یا نبودن آن نداشته و قابل حذف است) سپس ستون های باقیمانده را با متد min/max نرمال می کنیم.

در گام بعدی، ویژگی‌های موجود را گسسته می کنیم. یک روز گسسته سازی استفاده از هیستوگرام و bin های متناظر بصورت one-hot است که از آن نمیتوان در این مسئله استفاده نمود چرا که تراکم و فراوانی مدنظر قرار نگرفته و طول همه‌ی bin ها یکی می باشد. در نتیجه برای گسسته سازی ازquantile کردن استفاده میکنیم که در آن مکانیزم شبیه هیستوگرام می باشد با این تفاوت که طول هر bin متغیر بوده و بدین گونه تعیین می شود که تعداد نمونه‌های قرار گرفته در هر bin با یکدیگر برابر باشد. پس از انجام این گام، ویژگی را بصورت one-hot تبدیل می کنیم که آن نشان میدهد هر ویژگی در کدام bin قرار گرفته است و bin متناظر را 1 میکند.

مورد فوق را به ازای تمامی ویژگی ها انجام میدهیم. برای همه ویژگی‌ها تعداد bin ها برابر با ۱۵ بوده و برای ویژگی Amount نیز برابر با ۲۰ تعیین شده است(چرا که گستره آن در بین تراکنش های عادی بیشتر است و ما برخلاف سایر ویژگی‌ها در خصوص آن آگاهی داریم).

ب) پیاده‌سازی a-priori

روند و الگوریتم مورد بحث در کلاس برای این مورد پیاده سازی شده و trick برای افزایش سرعت نیز به کار گرفته شده است. در گام شروع و به ازای قوانین به طول دو (یک آیتم سمت چپ) support همه‌ی آیتم ها تقریبا برابر است چرا که توسط quantization حاصل شده اند. حال مستقیما قوانین به طول سه(دو آیتم سمت چپ) را محاسبه می‌کنیم. برای این منظور تمام حالات ممکن را بررسی می‌کنیم و قوانین برتر را استخراج می کنیم. در خصوص چگونگی تعیین حد آستانه support و confidence در بخش بعدی بحث خواهد شد.

وقتی که قوانین انجمنی به طول سه استخراج شد، برای استخراج قوانین انجمنی به طول چهار(سه آیتم سمت چپ) ترکیب تمام حالات ممکن قوانین استخراج شده به طول سه را با تمام ویژگی‌ها محاسبه کرده و قوانین برتر به طول چهار را استخراج می کنیم این شگرد باعث افزایش سرعت و کاهش محاسبات می‌شود. **یک شگرد** ابتکاری دیگری که برای کاهش محاسبات بیهوده و افزایش سرعت مورد استفاده قرار گرفته است این است که ترکیب ستون های گسسته شده‌ی حاصل از یک ویژگی با یکدیگر در نظر گرفته نمی شود چرا که میدانیم بصورت one-hot پر شده و امکان ندارد دو مورد با یک برابر باشند. در ادامه قوانین استخراج شده به همراه confidence و interest آورده شده است.

Association Rules for with len 3 for Non-Fruad Class
Potential Pattern: V5_14,V6_14 -> Non-Fruad | Confidence: 10790/10790=1.0 | Intrest:1.0-0.9983=0.0017
Potential Pattern: V6_14,V24_14 -> Non-Fruad | Confidence: 9229/9229=1.0 | Intrest:1.0-0.9983=0.0017
Potential Pattern: V6_14,V8_14 -> Non-Fruad | Confidence: 6836/6836=1.0 | Intrest:1.0-0.9983=0.0017
Potential Pattern: V1_14,V28_3 -> Non-Fruad | Confidence: 6381/6381=1.0 | Intrest:1.0-0.9983=0.0017
Potential Pattern: V1_14,V28_2 -> Non-Fruad | Confidence: 6045/6045=1.0 | Intrest:1.0-0.9983=0.0017

==========
Association Rules for with len 4 for Non-Fruad Class
Pattern: V6_14,V24_14,V5_14 -> Non-Fruad | Confidence: 5170/5170=1.0 | Intrest:1.0-0.9983=0.0017
Pattern: V5_14,V6_14,V24_14 -> Non-Fruad | Confidence: 5170/5170=1.0 | Intrest:1.0-0.9983=0.0017
Pattern: V5_14,V6_14,V8_13 -> Non-Fruad | Confidence: 4742/4742=1.0 | Intrest:1.0-0.9983=0.0017
Pattern: V8_13,V24_14,V6_14 -> Non-Fruad | Confidence: 3753/3753=1.0 | Intrest:1.0-0.9983=0.0017
Pattern: V6_14,V24_14,V8_13 -> Non-Fruad | Confidence: 3753/3753=1.0 | Intrest:1.0-0.9983=0.0017

==========
Association Rules for with len 3 for Fruad Class
Pattern: V12_0,V17_0 -> Fruad | Confidence: 372/587=0.6337 | Intrest:0.6337-0.0017=0.632
Pattern: V17_0,V18_0 -> Fruad | Confidence: 274/488=0.5615 | Intrest:0.5615-0.0017=0.5598
Pattern: V16_0,V17_0 -> Fruad | Confidence: 359/706=0.5085 | Intrest:0.5085-0.0017=0.5068
Pattern: V4_14,V10_0 -> Fruad | Confidence: 352/798=0.4411 | Intrest:0.4411-0.0017=0.4394
Pattern: V4_14,V16_0 -> Fruad | Confidence: 316/720=0.4389 | Intrest:0.4389-0.0017=0.4372

==========

Association Rules for with len 4 for Fruad Class
Pattern: V16_0,V17_0,V4_14 -> Fruad | Confidence: 313/369=0.8482 | Intrest:0.8482-0.0017=0.8465
Pattern: V4_14,V16_0,V17_0 -> Fruad | Confidence: 313/369=0.8482 | Intrest:0.8482-0.0017=0.8465
Pattern: V12_0,V17_0,V7_0 -> Fruad | Confidence: 335/400=0.8375 | Intrest:0.8375-0.0017=0.8358
Pattern: V12_0,V14_0,V7_0 -> Fruad | Confidence: 342/409=0.8362 | Intrest:0.8362-0.0017=0.8345
Pattern: V16_0,V17_0,V9_0 -> Fruad | Confidence: 255/306=0.8333 | Intrest:0.8333-0.0017=0.8316

ج) تعیین حد آستانه support

از جایی که هدف ما دسته‌بندی می‌باشد بایستی تعیین حد آستانه support برای انتخاب آیتم‌های پرتکرار بدرستی انجام شود؛ از طرفی چون مجموعه داده‌ی مورد نظر نامتعادل(imbalanced) می‌باشد تعیین حد آستانه دو چندان اهمیت پیدا می‌کند و بایستی برای آن چاره ای اندیشید. راه حل ارائه شده با راهنمایی تدریس‌یار محترم بدین گونه است که مرحله‌ای که میخواهیم قوانین انجمنی را به ازای کلاس fraud استخراج نماییم، علاوه بر حد پایین، حد بالا نیز تعیین می شود چرا که بدنبال آیتم هایی هستیم که بتوانند کلاس fraud را بدرستی تشخیص دهند و این زمانی اتفاق میافتاد که تکرار آن در مرتبه‌ی تعداد نمونه‌های کلاس باشد. تعداد نمونه‌های کلاس ۴۹۲ می باشد لذا حد آستانه‌ی را بین نصف تا دو برابر این عدد تعیین میکنیم[۹۸۴–۲۴۶]. در این صورت اعداد حاصل برای confidence بسیار کوچک نتیجه نمی شود.

همچنین در مرحله‌ای که میخواهیم قوانین انجمنی را به ازای کلاس non-fraud محاسبه کنیم صرفا تعیین حد پایین آستانه کافی است که برابر با ۱۵۰۰ تعیین شده است. (طبق گسسته سازی، حداکثر تکرار هر آیتم تقریبا برابر ۱۵۰۰۰ است که بنده ۱۰٪ آن را در نظر گرفته ام)

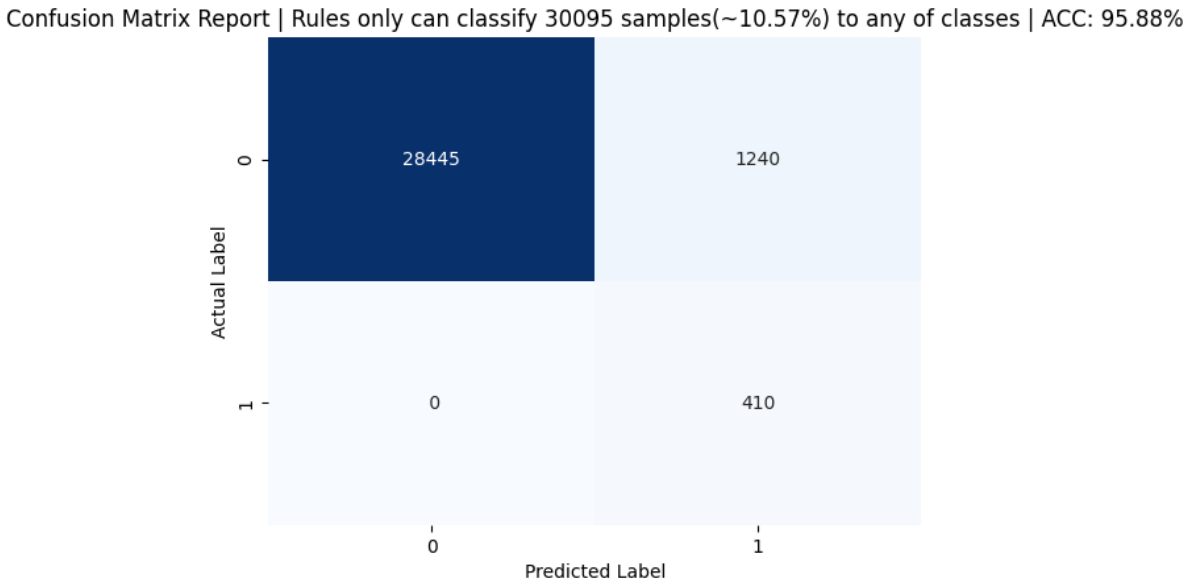
د) گزارش interest و confidence

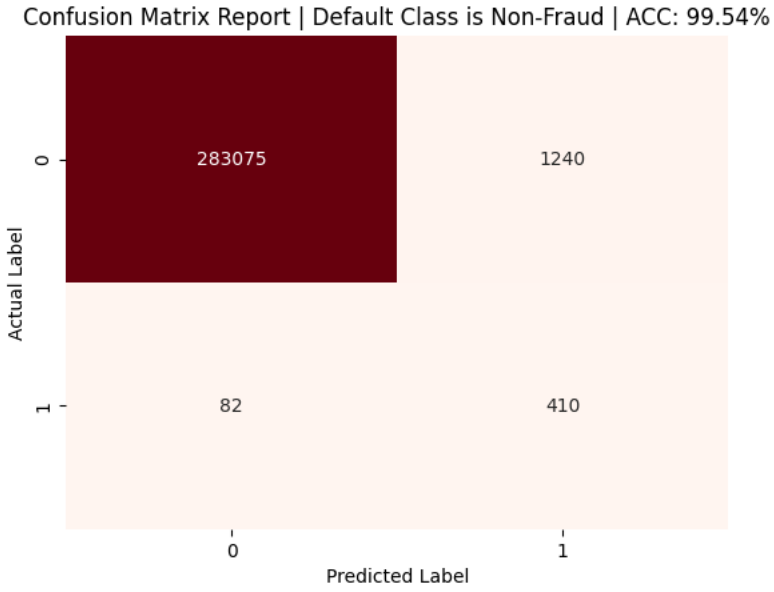
اعداد خواسته شده در قسمت (ب) آورده شده و قابل مشاهده است. نکته‌ای که قابل اهمیت است این است که به ازای قوانین انجمنی برای کلاس fraud مقدار interest بالایی حاصل شده است و نشان میدهد که از اهمیت بالایی برخوردار است و اعمال حد بالا برای support تاثیر مثبت داشته است. همچنین به ازای قوانین انجمنی برای کلاس non-fraud مقدار interest پایین حاصل شده است که نشان میدهد از اهمیت کمی برخوردار است و میتوان از آن چشم پوشی نمود چرا که در قسمت بعد نیز قابل مشاهده است که پیشفرض قراردادن کلاس non-fraud به عنوان برچسب، تاثیر چندانی در دقت دسته‌بندی مربوط به آن کلاس ندارد.

ه) دسته‌بندی

برای دسته بندی از ۵ قوانین برتر بدست آمده در قالب voting وزن دار استفاده می‌کنیم که در آن هر نمونه را با تمام قوانین بررسی کرده و شمارش میکنیم که آن نمونه به ازای چند قانون انجمنی برای کلاس fraud و به ازای چند قانون انجمنی برای کلاس non-fraud صدق میکند(وزن صدق کردن قانون های سه تایی برابر با ۱ و وزن صدق کردن قانون های چهارتایی برابر با ۲ تنظیم شده است چرا که اهمیت آن بیشتر است). سپس وزن‌های بدست آمده به ازای دو کلاس را باهم مقایسه میکنیم و کلاسی که وزن آن بیشتر بود به عنوان برچسب پیشبینی تعیین می شود.

نتایج دسته‌بندی در زیر آورده شده است که در دو صورت می باشد. صورت اول بدین گونه است که پیشفرض برچسب کلاس non-faud می باشد و توانسته است دقتی برابر ۹۹.۵۴ و TP برابر با ۴۱۰ را حاصل کند(ماتریس درهمریختگی قرمز رنگ)؛ در صورت دیگری از دسته‌بندی هیچ پیشفرضی وجود ندارد و ممکن است نمونه ای با هیچ قانون مطابقت نداشته باشد و کلا نتوانیم آن را دسته بندی کنیم. در این حالت ۱۰ درصد نمونه‌ها با قوانین میتوانند پیش‌بینی شوند و دقتی برابر با ۹۵.۸۸٪ و TP برابر با ۴۱۰ را نیز به همراه آورد(ماتریس درهم ریختگی آبی رنگ).




Confusion Matrix Report | Default Class is Non-Fraud | ACC: 99.54%
283075
1240
82
410
0
1
Actual Label
0
1
Predicted Label

بخش سوم: Locality Sensitive Hashing

در این بخش پاسخ‌های مربوط به مبحث Locality Sensitive Hashing آورده شده است که مشتمل بر سه سوال می‌باشد که سوال اول و دوم تشریحی/توصیفی بوده و سوال سوم بصورت پیاده می‌باشد.

سوال اول: محدودیت LSH

محدودیت‌های متعددی را میتوان برای LSH مطرح نمود که هر یک در جای خود اهمیت دارد. اولین مورد بحث تعیین پارامترهای LSH می باشد که تاثیر مستقیم بر کیفیت و عملکرد آن دارد؛ در این راستا، بایستی مقادیری برای تعداد توابع هش(Min-hashes)، تعداد bands و تعداد سطر(rows) مشخص کرد که اینان عملکرد LSH، نرخ و نوع خطا(FN/FP) را مستقیما تحت تاثیر قرار می‌دهد و آن عملکرد شباهت سنجی LSH محدود می‌کند. در همین راستا، عدم قطعیت LSH و تقریبی بودن آن مورد بحث بعدی می‌باشد که میتواند در مواردی با نتیجه‌ی قطعی تفاوت داشته باشد و دلیل این تقریبی بودن از بابت نگاشت سریع از فضای بسیار بزرگ(Sparse) به فضای کوچک(signature) است که توسط توابع hash انجام می‌شود که تصادفی بوده و امکان رخداد خطا و رفتار گوناگون در اجرا های مختلف وجود دارد. مورد بعدی بحث زمان اجرایی و سرعت می باشد که در مجموعه داده های بسیار بزرگ و گسترده هزینه بالایی در ایجاد ماتریس ها و مین–هش ها وجود دارد.

موردی بعدی در محدودیت LSH عدم توانایی در استخراج روابط پیچیده در شباهت سنجی می‌باشد. LSH بدنبال استخراج شباهت عینی بوده و نمیتواند شباهت های معنایی را استخراج کند برای مثال بین دو جمله‌ی "خودرو تصادف کرده است" و "ماشین با دیوار برخورد داشته است" نمیتواند شباهتی استخراج نماید.

سوال دوم: Similarity preserving در min–hashing

برای اینکه نشان دهیم در min-hashing قابلیت حفظ شباهت وجود دارد، کافی است نشان دهیم که طی آن عملکرد معیار شباهت jacard تقریبا ثابت مانده و min-hashing میتواند به خوبی معیار شباهت jacard را تخمین زند. این مورد

را می‌توانیم با نشان دادن اینکه احتمال برابر بودن و مطابقت داشتن مقادیر min-hash برابر با شباهت جاکارد است. این اثبات در اسلایدهای درس وجود داشته و در کلاس نیز انجام شده است(صفحات ۲۶ و ۲۷) و بنده بر اساس آن اثبات را پیش می‌برم:

اگر بخواهیم در قالب ریاضیاتی مسئله مورد نظر را بازنویسی کنیم، رابطه‌ی زیر بدست می‌آید که C ها دو ستون دلخواه می‌باشد:

$$Pr[h\_permutation(C_1) = h\_permutation(C_2)] = sim(C_1, C_2)$$

حال اگر X را یک داکیومنت در نظر بگیریم که مجموعه‌ای از shingle ها بوده و z یک shingle عضو X باشد میتوانیم بنویسیم احتمال اینکه هر shingle در permutation انتخاب شود، بصورت یکنواخت برابر با ۱ تقسیم بر تعداد shingleها در آن داکیومنت می باشد چرا که خود permutation ها بصورت تصادفی یکنواخت بوده است(احتمال انتخاب هر خانه از ستون با هم برابر و یکنواخت است).

$$Pr[permutation(z) = min(permutation(X))] = 1/|X|$$

حال یک ستون فرضی از اجتماع دو ستون C1 و C2 در نظر میگیریم و فرض میکنیم که خانه yام آن در permutation انتخاب می‌شود. حال این نتیجه حاصل می‌شود که خانه y حداقل در یکی از ستون های C1 و C2 برابر با یک بوده و بواسطه‌ی آن انتخاب شده است :

$$permutation(y) = min(permutation(C1 \cup C2))$$

⇨ permutation (y) = min(permutation (C1)) if y ∈ C1

or/and

⇨ permutation (y) = min(permutation (C2)) if y ∈ C2

حال احتمال اینکه خانه y در هر دو ستون ظاهر شده باشد برابر است با $C1 \cap C2$ و در نتیجه میتوان نوشت:

$$Pr[min(permutation(C1)) = min(permutation(C2))] = \left|\frac{C1 \cap C2}{C1 \cup C2}\right| = sim(C1, C2)$$

طبق روابط بالا اثبات میشود که برای یک permutation شباهت در min-hashing حفظ می شود و حال برای تعمیم آن برای چند permutation و حفظ شباهت در signature ها، امید ریاضی شباهت signatureها را مدنظر قرار میدهیم چرا که احتمالاتی می باشد؛ اگر تعداد min-hash ها را برابر با m در نظر بگیریم میتوانیم رابطه امیدریاضی را نوشته و ساده سازی کنیم:

$$E[sim(S1, S2)] = E[\#permutations\ \pi: h\pi\ (C1) = h\pi(C2)] / m$$

$$= m*Pr[h\pi(C1) = h\pi(C2)] / m = Pr[h\pi(C1) = h\pi(C2)] = sim(C1, C2)$$

در نتیجه امید ریاضی اینکه شباهت دو signature برابر است با شباهت دو ستون متناظر لذا در حالت کلی تر نیز میتوان نتیجه که min-hashing شباهت را حفظ می‌کند.

سوال سوم: پیاده‌سازی LSH

تمامی توابع مورد نظر پیاده‌سازی شده و نتایج حاصل شده است. در تابع create_shingle داکیومنت ها یک به یک در یک حلقه پیمایش می شود و در یک حلقه داخلی‌تر shingle ها استخراج می شود که طول shingle با گام حلقه کنترل می شود. همچنین برای بهبود نتایج حاصل، تمامی نقاط پایانی جملات حذف می شود.

در تابع create_sparse_vectors ابتدا یک ماتریس خالی از صفر با ابعاد تعداد shingle * تعداد vocabulary ساخته می‌شود و سپس حضور تک تک vocabulary در هر داکیومنت بررسی می شود و ماتریس ساخته شده با ۱ بروزرسانی شده و در نهایت ماتریس بازگردانده می شود. تابع create_hash_fucntion نیز یک لیست اعداد مرتب به طول عدد وارد شده تولید کرده و سپس آن لیست را shuffle کرده و باز میگرداند. تابع create_minhash_functions نیز از تابع قبلی استفاده کرده و لیستی از توابع هش را می‌سازد. تابع create_signature_matrix طبق تصویر موجود در فایل ژوپیتر پیاده‌سازی شده است که برای هر داکیومنت بر اساس توابع هش ها، signature معادل را تولید می‌کند.

تابع jaccard نیز شباهت جاکارد را محاسبه می‌کند که برابر است با تعداد عناصر مشترک تقسیم بر تعداد عناصر اجتماع. تابع get_topk_similar نیز شباهت بین داکیومنت هدف و داکیومنت های candidate را ارزیابی کرده و k تا از شبیه ترین ها را برمیگرداند. برای این منظور بنده سه حالت مختلف را ارزیابی کرده ام که در یکی سنجش شباهت بین کلمات،

در یکی سنجش شباهت بین shingle ها و در دیگری سنجش شباهت بین signature ها انجام پذیرفته و نتیجه چاپ شده است. در زیر نتیجه به ازای شباهت سنجی مابین shingle های داکیومنت هدف و داکیومنت‌های داوطلب آورده شده است:

Target sentence

The lazy dog is jumped over by a quick brown fox.

=== Similarity based on documents shingles ===

The lazy dog is jumped over by a quick and brown-colored fox.(0.5873)

The lazy dog is leaped over by a quick and brown fox.(0.5862)

The dog that is asleep is jumped over by a quick brown fox.(0.5077)

The quick and brown fox is jumped over by the lazy dog.(0.4921)

The lazy dog has a quick and brown fox that is jumping over it.(0.3421)

Index: [34, 19, 46, 41, 29]
