به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و

کامپیوتر

درس داده کاوی پیشرفته

تمرین اول

نام و نام خانوادگی	عرفان شهابی
شماره دانشجویی	۸۱۰۱۰۳۱۶۶
تاریخ ارسال گزارش	۱۴۰۴.۲.۱۱

فهرست

نمودارها

جدول‌ها

سوال ۱

. 1

Item	Frequency	Item	Frequency
تن ماهی شیلا	۱	نان تست سحر	۲
شیر پگاه	۱	اولیه نامی نو	۱
پنیر هزار	۱	شیر چوپان	۱
نان تست سه نان	۱	کیک معینی پور	۱
پنیر کاله	۱	نان تست نان آوران	۱
شیر کاله	۲	پنیر میهن	۱
کیک سه نان	۱		

1 جدول

در این سوال، از آنجایی که ۴ تراکنش داریم و minimum support برابر ۷۰٪ است:

$$4 \times 0.7 = 2.8$$

که به این معنی است که برای پاس کردن این minimum support آیتم مورد نظر باید حداقل در ۳ تراکنش ظاهر شده باشد.

تعداد حضور آیتم در جدول بالا نشان داده شده است که همانطور که می بینیم، بیشترین تکرار یک آیتم برابر ۲ بوده است که شرط minimum support را پاس نمی کند.

که نتیجه می گیریم frequent 1 itemset نداریم چرا که ساپورت همه آیتم ها کمتر از ۷۰٪ است.

همچنین از آن جایی که frequent 2 itemset فقط از frequent 1 itemset ساخته می شود، frequent 2 itemset هم طبیعتا وجود نخواهد داشت.

. 2 بعد از نگه داشتن دسته کالا ها جدول تراکنش ها به شکل زیر تغییر می کند:

لیست خرید ها پس از ساده سازی	لیست خرید ها	شماره تراکنش
تن ماهی | شیر | پنیر | نان تست	تن ماهی شیلا | شیر پگاه | پنیر هزار | نان تست سه نان	۱
پنیر | شیر | کیک | نان تست	پنیر کاله | شیر کاله | کیک سه نان | نان تست سحر	۲
الویه | شیر | نان تست | کیک	الویه نامی نو | شیر چوپان | نان تست سحر | کیک معینی پور	۳
نان تست | شیر | پنیر	نان تست نان آوران | شیر کاله | پنیر میهن	۴

2 جدول

که در این صورت جدول فراوانی آیتم ها برابرست با:

Item	Frequency
تن ماهی	۱
شیر	۴
پنیر	۳
نان تست	۴
الویه	۱
کیک	۱

جدول 3

که طبق این جدول فراوانی:

شیر در ۴ تراکنش ظاهر شده: support = 100%

پنیر در ۳ تراکنش ظاهر شده: support =75%

نان تست در ۴ تراکنش ظاهر شده: support = 100%

بنابراین آیتم های شیر، پنیر و نان تست frequent 1-itemset هستند.

حالا ترکیب های دو تایی را بررسی میکنیم:

Item	Frequency	support
(شیر، پنیر)	۳	50%
(شیر، نان تست)	۴	100%
(پنیر، نان تست)	۳	75%

جدول 4

که با توجه به minimum support = 70%، هر سه آیتم ست frequent هستند.

حالا در این مرحله به سراغ ساخت frequent 3-itemsets با استفاده frequent 2-itemset می رویم که بر این اساس تنها ترکیب سه تایی ممکن، (شیر، نان تست، پنیر) است که حضور آن به شکل زیر می باشد:

- تراکنش ۱: (شیر، نان تست، پنیر) حضور دارد.
- تراکنش 2: (شیر، نان تست، پنیر) حضور دارد.
- تراکنش 4: (شیر، نان تست، پنیر) حضور دارد.

که بدین ترتیب ساپورت برای این ۳ تایی برابر با 75% است و شرط minimum support را پاس میکند

در مرحله بعد به سراغ استخراج association rules می رویم:

از یک frequent 3-itemset می توان ۳ قانون 1<=2 استخراج کرد:

قانون ۱:

پنیر <= شیر ، نان تست

Support (پنیر <= شیر ، نان تست): ¾ = 75%

Confidence (پنیر <= شیر ، نان تست): ¾ = 75%

قانون ۲:

نان تست <= شیر، پنیر

Support (نان تست <= شیر، پنیر): ¾ = 75%

Confidence (نان تست <= شیر، پنیر): 3/3 =100%

قانون ۳:

شیر <= پنیر، نان تست

Support(شیر <= پنیر، نان تست): ¾ = 75%

Confidence(شیر <= پنیر، نان تست): 3/3 = 100%

که با توجه به مقادیر به دست آمده، همانطور که مشخص است، قانون اول شرط Confidence را پاس نمی کند ولی قوانین دوم و سوم بر قرارند.

سوال ۲

1 . در گام اول فراوانی آیتم ها را می شماریم:

Item	support
A	3
B	5
C	2
D	3
E	2
F	1

جدول 5

که از آنجایی که minimum support برابر با ۲ است، F حذف می شود.

در نتیجه Frequent 1-itemsetها به شرح زیر می باشند:

{A}, {B}, {C}, {D}, {E}

در مرحله بعد 2-itemsetها را محاسبه می کنیم:

Item	support
{A, B}	3
{A, C}	1
{A, D}	1
{A, E}	0
{B, C}	2
{B, D}	2
{B, E}	2
{C, D}	1
{C, E}	1
{D, E}	2

جدول 6

که در نتیجه Frequent 2-itemsetها به شرح زیر می باشند:

{A, B}, {B, C}, {B, D}, {B, E}, {D, E}

در گام سوم باید Frequent 3-itemsetها را محاسبه کنیم که تنها آیم ست {B, D, E} با فراوانی ۲ شرط minimum support را پاس می‌کند.

در نتیجه پاسخ این بخش به شرح زیر است:

{A}, {B}, {C}, {D}, {E}, {A, B}, {B, C}, {B, D}, {B, E}, {D, E}, {B, D, E}

2. طبق تعریف داریم:

Closed Pattern: اگر هیچ superset با فرکانس مشابه نداشته باشد.

Maximal Pattern: اگر هیچ superset وجود نداشته باشد که frequent باشد.

با استفاده از این تعاریف داریم:

Itemset	Superset Frequent	Same Support?	Closed?	Maximal?
A	{A, B} 3	Yes	No	No
B	{B, C} 2, {B, D} 2, {B, E}, {B, D, E} 2	Yes	Yes	No
C	{B, C} 2	Yes	No	No
D	{B, D} 2, {D, E} 2, {B, D, E} 2	Yes	No	No
E	{D, E} 2	Yes	No	No
{A, B}	-	-	Yes	Yes
{B, C}	-	-	Yes	Yes
{B, D}	{B, D, E} 2	No	Yes	No
{B, E}	{B, D, E} 2	Yes	No	No
{D, E}	{B, D, E} 2	Yes	No	No
{B, D, E}	-	-	Yes	Yes

جدول 7

در نتیجه جواب نهایی این قسمت به شرح زیر است:

Closed Patterns: {C}, {A, B}, {B, C}, {B, D, E} => 4

Maximal Patterns: {A, B}, {B, C}, {B, D, E} => 3

3 . آیا هر maximal pattern یک closed pattern هم هست؟

بله، چون هیچ supersetای frequent ندارد. پس قطعا superset با همان support هم ندارد.
بنابراین هر maximal pattern الزاما closed هم هست ولی برعکس این گزاره درست نیست.

برای مثال گفته شده به صورت زیر عمل می کنیم:

برای فهم از این موضوع مثال ساده زیر را در نظر بگیریم:

T1 = {A, B, C}

T2 = {B, C, D}

Frequent itemset ها برای این مثال به صورت زیر می باشند:

1-itemsets:

A, B, C, D

2-itemsets:

{A, B}, {A, C}, {B, C}, {B, D}, {C, D}

3-itemsets:

{A, B, C}, {B, C, D}

همانطور که در بالاتر گفته شد، یک itemset زمانی closed است که هیچ superset با همان support نداشته باشد. در نتیجه closed patternها در این مثال به شرح زیرند:

{A, B, C}, {B, C}, {B, C, D}

و طبق تعریف یک itemset زمانی maximal است که هیچ superset frequent نداشته باشد در نتیجه maximal patternهای ما عبارتند از:

{A, B, C}, {B, C, D}

به همین ترتیب برای مثال گفته شده maximal patterns و closed patterns به شرح زیرند:

Maximal patterns: {a1, ….., an}, {a2, ……, an+1}

Closed patterns: {a1, ….., an}, {a2, ……, an+1}, {a2, ……, an}

. 4

در اینجا ما علاوه بر شرط support، میخواهیم فقط آن itemsetهایی را نگه داریم که حداقل یکی از آیتم هایشان کوچکتر و یا مساوی ۱۰۰ باشد.

این قید دارای خاصیت anti monotonic است. به این معنی که اگر یک itemset شرط را نقض کند (یعنی min(price) > 100)، آنگاه تمام supersetهای آن نیز حتما شرط را نقض می کنند.

نحوه اعمال در Apriori:

در هنگام گسترش کاندیداها و یا بعد از تولید هر frequent itemset:

1 . برای هر itemset، قیمت هر آیتم را از جدول استخراج می کنیم.
2 . اگر min(price) > 100 بود آن مجموعه و همه supersetهایش prune می شوند.

. 5

در شرط این قسمت میخواهیم فقط itemsetهایی را نگه داریم که میانگین قیمت آیتم هایشان بزرگتر و یا برابر با ۳۰۱ باشند.

این قید خاصیت monotonic دارد. یعنی اگر یک itemset این شرط را ارضا کند (یعنی $(avg(price \geq 301)$)، آنگاه تمام supersetهای آن نیز شرط را حتما ارضا خواهند کرد. چرا که با توجه به این که قیمت عددی نامنفی است، میانگین آن ممکن است افزایش یابد و یا ثابت بماند.

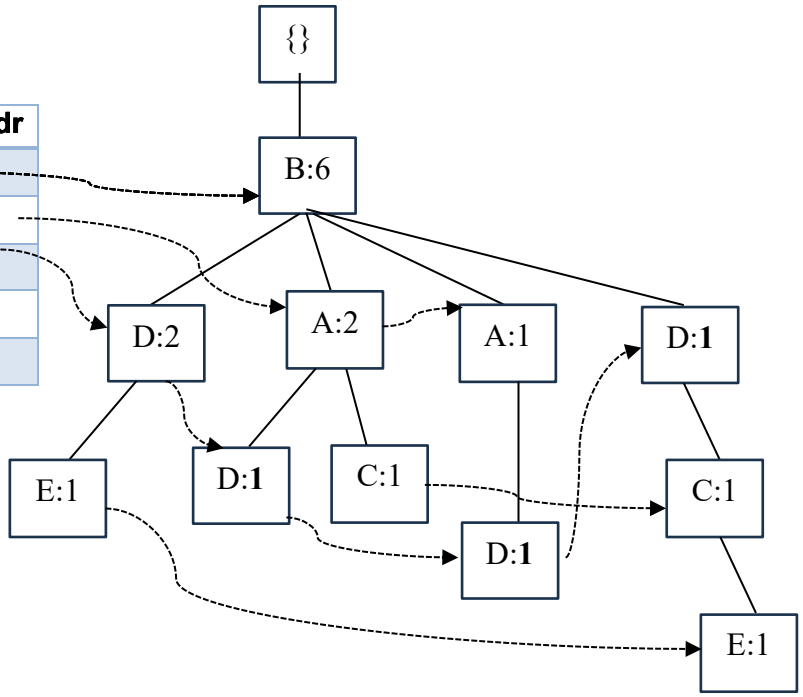
نحوه اعمال در FP-Growth:

در الگوریتم FP-Growth:

1 . پس از استخراج itemsetهای frequent از conditional DBها،
2 . از پایین ترین آیتم شروع میکنیم و برای هر itemset:
- قیمت آیتم ها را نگاه می کنیم.
- اگر avg(price) < 301 بود آن مجموعه prune می شود.
3 . چون شرط monotonic است، نیازی نیست subtree های آن مجموعه بررسی شوند

برای حل این قسمت ابتدا تراکنش ها را با ترتیب نزولی مرتب می کنیم و FP-tree را تشکیل می دهیم.

Item	support	hdr
B	6	
A	3	
D	3	
C	2	
E	2	



در مرحله بعد از پایین ترین آیتم شروع میکنیم:

Item E:

Conditional pattern base:

path	support
B=>D	1
B=>D=>C	1

جدول 8

که از این ها:

E(2), {D, E}(2), {B, D, E}(2), {C, D, E}(1)=>حذف می شود

Item C:

Paths:

path	support
B=>A	1
B=>D	1

جدول 9

C(2), {B, C}(2)

Item D:

Paths:

path	support
B=>A	1
B=>A	1
B	1

جدول 10

D(3), {B, D}(3)

{A, D}(1), {B, A, D}(1)=>حذف می شوند

Item A:

Paths:

path	support
B	3

جدول 11

A(3), {B, A}(3)

Item B:

B(6)

در مرحله بعد به سراغ محاسبه میانگین و فیلتر کردن الگوها می رویم:

Itemset	Support	Avg (price)	Valid
B	6	200	No
A	3	100	No
D	3	400	Yes
E	2	500	Yes
C	2	300	No
A, B	3	150	No
B, D	3	300	No
B, E	2	350	Yes
D, E	2	450	Yes
B, D, E	2	366.66	Yes
B, C	2	250	No

جدول 12

که در نهایت الگوهایی که شرط مسئله را ارضا می کنند به شرح زیرند:

Itemset	Support	Avg (price)	Valid
D	3	400	Yes
E	2	500	Yes
B, E	2	350	Yes
D, E	2	450	Yes
B, D, E	2	366.66	Yes

جدول 13

سوال ۳

. 1

برای محاسبه Chi-Square داریم:

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

که مقادیر مورد انتظار به شرح زیر می باشند:

E(American, SUV): $\frac{200\times250}{1800} \approx 27.78$

E(American, NOT_SUV): $\frac{200\times1550}{1800} \approx 172.22$

E(NOT American, SUV): $\frac{1600\times250}{1800} \approx 222.22$

E(NOT American, NOT_SUV): $\frac{1600\times1550}{1800} \approx 1377.78$

بنابراین برای Chi-Square داریم:

$$\chi^2 = \frac{(150 - 27.78)^2}{27.78} + \frac{(50 - 172.22)^2}{172.22} + \frac{(100 - 222.22)^2}{222.22} + \frac{(1500 - 1377.78)^2}{1377.78} \approx 537.03$$

نتیجه نشان میدهد که مقدار chi-square بسیار بزرگ است که نشان دهنده ارتباط قوی بین ملیت آمریکایی و خرید SUV است (فرض صفر استقلال رد می شود).

محاسبه معیار kulczynski

برای محاسبه این معیار از فرمول زیر استفاده می کنیم:

$$\mathbf{kulczynski} = \frac{1}{2}\left(\frac{P(A \cap B)}{P(A)} + \frac{P(A \cap B)}{P(B)}\right)$$

برای قانون American => SUV:

$$P(A \cap B) = \frac{150}{1800}$$

$$\boldsymbol{P(A) = \frac{200}{1800}, P(B) = \frac{250}{1800}}$$

در نتیجه:

$$\textbf{kulczynski} = \frac{1}{2}\left(\frac{\frac{150}{1800}}{\frac{200}{1800}} + \frac{\frac{150}{1800}}{\frac{250}{1800}}\right) = \mathbf{0.675}$$

مقدار kulaczynski بین ۰ و ۱ است و مقدار 0.675 نشان دهنده ارتباط نسبتا قوی بین ملیت آمریکایی و خرید SUV می باشد.

محاسبه معیار IR

معیار Imbalance Ratio برای سنجش عدم تعادل در توزیع داده ها استفاده می شود که فرمول آن به صورت زیر است:

$$IR = \frac{|P - N|}{P + N}$$

که در آن:

P = تعداد موارد مثبت = 150 (American & SUV)

N = تعداد موارد منفی = 100 (Not American & SUV)

بنابراین

$$IR = \frac{|150 - 100|}{150 + 100} = \frac{50}{250} = 0.2$$

این معیار بین ۰ و ۱ است و هر چه به ۱ نزدیک تر باشد به این معناست که عدم تعادل بیشتر است. مقدار 0.2 نشان دهنده عدم تعادل نسبتا کم در این داده هاست.

. 2

Null Invariance برای معیاری بر قرار است که به مقادیر منفی (عدم وقوع رویداد) حساس نباشد. در این سوال مقادیر منفی Not American و NOT_SUV می باشند.

برای معیارهای محاسبه شده داریم:

:Chi-Square

- به تمامی مقادیر جدول حساس است.

- ممکن است نتیجه را در داده های نامتوازن تحریف کند.

Kulczyynski:

- فقط به موارد مثبت توجه می کنند.

IR:

- عدم تعادل را اندازه گیری می کند اما مستقیما ارتباط را نمی سنجد.

نتیجه گیری:

با توجه به تعریف گفته شده از Null Invariance، در اینجا معیار kulczynski دقیق تر است و Chi-Square ممکن است تحت تاثیر حجم بالای داده های منفی قرار گیرد.

سوال ۴

. 1

BFS :Apriori

- این الگوریتم از رویکرد سطح به سطح استفاده می کند (Level-wise)
- ابتدا تمام itemsetهای با طول ۱ را بررسی میکند و سپس به itemsetهای به طول ۲ را بررسی می کند و به همین ترتیب ادامه می دهد.

DFS :FP-Growth

- این الگوریتم از درخت FP-Tree استفاده می کند و به صورت شاخه به شاخه itemsetهای مکرر را استخراج می کند (مثلا از ریشه به برگ).

BFS :GSP

- این الگوریتم نسخه توسعه یافته الگوریتم Apriori است و به صورت سطح به سطح عمل می کند.
- این الگوریتم به این صورت عمل می کند که دنباله ها با بر اساس طولشان گسترش می دهد. برای مثال ابتدا دنباله ها با طول ۱ را استخراج می کند و سپس دنباله ها به طول ۲ و به همین ترتیب ادامه می دهد.

DFS :PrefixSpan

- این الگوریتم از پروجکشن (برش دنباله بر اساس پیشوند) استفاده می کند.
- این الگوریتم دنباله ها را به صورت عمقی بررسی می کند و برای هر پیشوند، پایگاه داده پروجکشن شده جدیدی ایجاد می کند.

. 2

برای حل این سوال در مرحله اول به دنبال <a> سپس <b> هستیم (نه لزوما در یک itemset، بلکه پشت سر هم در توالی).

ID 1: a<bc><ac><cf>

شامل a در آیتم اول و b در آیتم دوم است. => <ab> دوم هست.

ID 2: a<bc>a

شامل a در آیتم اول و b در آیتم دوم است. => <ab> دوم هست.

ID 3: abc

در این بخش a و b در یک آیتم هستند ولی چون هم زمان رخ می دهند و نه به صورت sequence رد می شود.

در نتیجه فقط ID های ۱ و ۲ وارد <ab>-Projected DB می شوند.

مرحله بعد ساخت <ab>-projected DB است.

ID 1: a<bc><ac><cf>

بعد از <a> و بعد از <b> در آیتم بعدی <ac><cf> هستند پس projected suffix = <ac><cf>

ID 2:

a<bc>a

بعد از <a> و بعد از <b> فقط a باقی می ماند پس projected suffix = a

مرحله بعد استخراج frequent itemها از projected DB با minimum support = 2 است.

بر اساس مرحله قبل ما دو suffix داریم:

- <ac><cf>
- a

حالا occurrence آیتم ها را حساب می کنیم:

- a: ۲ بار
- c: فقط در <ac> و <cf> => ۱ بار (مهم این است که در sequence چند بار تکرار شده)
- f: فقط در <cf> => ۱ بار

پس فقط آیتم a frequent است.

سوال عملی

1.

در این قسمت ابتدا محیط colab را به google drive متصل کردم تا دیتا ست را لود کنم.

بعد از لود کردن دیتا ست نمونه ای از دیتا ست را برای آشنایی چاپ کردم.

در مرحله بعد، همانطور که سوال خواسته بود، ستون UnitPrice را به فرمت عددی تبدیل کردم و فرمت ستون های موجود را هم چاپ کردم.

برای پیش پردازش داده ها، در مرحله بعد ابتدا ستون Country و Unnamed: 0 که بلا استفاده بودند را از دیتافریم حذف کردم.

برای حذف کردن داده های تکراری، در مرحله بعد، داده هایی که در ستون های InvoiceNo، Description، StockCode و Quantity دارای مقادیر یکسان بودند را حذف کردم. تعداد ردیف داده در دیتا فریم خام، ابتدا ۴۶۴۳۱ داده بود و بعد از حذف مقادیر تکراری، تعداد ردیف داده ها به ۴۶۳۲۵ رسید که یعنی ۱۰۶ داده به عنوان داده تکراری شناسایی و حذف شدند.

سپس در مرحله بعد، تعداد تراکنش ها برای هر مشتری با CustomerID یکسان را محاسبه کردم که مشتری با شماره ۵۸۹۸ بیشترین تعداد تراکنش را در این فروشگاه داشته است.

در مرحله بعدی پیش پردازش متوجه شدم که کالاهایی که با پست ارسال شدند، دارای StockCode و Description مشابه بودند و برای مرحله پیدا کردن الگوها، با اینکه ممکن است این کالاها متفاوت باشند، دچار اشتباه می شویم. به همین دلیل داده های کالاهایی که با پست ارسال شده بودند را حذف کردم که تعداد ردیف این داده ها ۱۱۱۲ عدد بود.

2.

برای قسمت دوم سوال، ابتدا کتابخانه mlxtend را نصب کردم تا بتوانم از ابزارهایی مانند apriori و fpgrowth و ... استفاده کنم.

در مرحله بعد، سبد خرید را از داده های اصلی ساختم. ابتدا داده ها را بر اساس InvoiceNo گروه بندی کردم و برای هر فاکتور، تمام StockCodeها رو در قالب یک لیست جمع کردم. خروجی این مرحله دیتافریمی است که هر ردیفش یک transaction است. سپس لیست ها را به one hot encoding تبدیل کردم بدین صورت که هر لیست از آیتم ها در ستون StockCode رو به یک سطر از ۰ و ۱ ها تبدیل می کند. اگر سلول ۱ باشد یعنی آیتم مورد نظر در آن تراکنش هست و اگر ۰ باشد به معنی این است که آیتم در تراکنش نیست.

	10002	10120	10125	10133	10135	11001	15034	15036	15039	15044A
0	True	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False

5 rows × 2806 columns

سپس در مرحله بعد سعی کردم دو الگوریتم Apriori و FP-Growth را روی داده ها اجرا کنم. به این صورت که ابتدا الگوریتم Apriori را روی ماتریس تراکنش ها اعمال کردم و طبق گفته سوال minimum support را برابر با 0.06 قرار دادم که به این معناست که آیتم هایی frequent هستند که حداقل در 6% از کل تراکنش ها ظاهر شده باشند. Use_colnames را هم برابر با True قرار دادم که باعث می شود تا به جای اندیس ها، نام ستون ها (یعنی کد محصول) در خروجی نمایش داده شود.

در مرحله بعد نیز مشابه با مرحله قبل منتها این بار الگوریتم FP-Growth را روی داده ها اجرا کردم. مزیت این الگوریتم نسبت به الگوریتم Apriori سریع تر و بهینه تر بودن آن است. چرا که بدون تولید مرحله به مرحله کاندیداها کار می کند.

در نهایت top frequent itemها را چاپ کردم.

```
Apriori Results:
     support itemsets
11  0.144079  (22423)
9   0.134673  (22326)
13  0.094912  (22554)
14  0.089782  (22556)
15  0.085934  (22629)
```

```
FP-Growth Results:
     support itemsets
13  0.144079  (22423)
0   0.134673  (22326)
3   0.094912  (22554)
4   0.089782  (22556)
1   0.085934  (22629)
```

که همانطور که در نتایج مشخص است نتایج هر دو الگوریتم یکسان است و نشان دهنده این است که هر دو الگوریتم صحیح اجرا شده اند. بیشترین تکرار برای آیتم 22423 بوده است که نشان می دهد این آیتم در 14.4% تراکنش ها تکرار شده است.

در مرحله بعد نام این محصولات پر تکرار را چاپ کردم که به شرح زیر است:

```
22423: ['REGENCY CAKESTAND 3 TIER']
22326: ['ROUND SNACK BOXES SET OF4 WOODLAND ']
22554: ['PLASTERS IN TIN WOODLAND ANIMALS']
22556: ['PLASTERS IN TIN CIRCUS PARADE ']
22629: ['SPACEBOY LUNCH BOX ']
```

نتایج نشان دهنده این است که پر تکرار ترین محصول در تراکنش ها، پایه کیک ۳ طبقه بوده است که احتمالا در دسته بندی ظروف آشپرخانه دسته بندی می شود و در رتبه دوم هم ست ۴ عددی ظرف گرد برای تنقلات قرار دارد که آن هم دسته بندی مشابه با آیتم قبلی دارد. آیتم پر تکرار پنجم هم مربوط به دسته ظروف است و دو آیتم سوم و چهارم هم مربوط به لوازم بهداشتی می باشند. میتوان تصور کرد که در بازه ای که داده ها ثبت شده اند، فروشگاه طرح فروش ویژه و یا تخفیف روی این دو دسته بندی داشته است.

در مرحله بعدی این سوال، زمان اجرای الگوریتم و تعداد الگوهای پیدا شده توسط این دو الگوریتم را محاسبه کردم که نتایج به شرح زیر می باشد:

```
Apriori time: 0.024270057678222656
FP-Growth time: 0.26194119453430176
```

```
Apriori patterns found: 24
FP-Growth patterns found: 24
```

طبق نتایج به دست آمده، هر دو الگوریتم توانستند ۲۴ الگوی پر تکرار را استخراج کنند که نشان می دهد هر دو الگوریتم نتایج یکسانی داشته و به درستی اجرا شده اند.

اما مسئله ای که قابل تامل است زمان اجرای این دو الگوریتم است. ما انتظار داشتیم که الگوریتم FP-Growth سریع تر اجرا شود چرا که این الگوریتم بدون تولید مرحله به مرحله کاندیداها کار میکند. اما احتمالا دلیل نتیجه به دست آمده و سریع تر بودن الگوریتم Apriori، تعداد کم داده هاست.

.3

در مرحله بعد طبق خواسته سوال سعی کردم کدی را بنویسم که association ruleها رو از frequent itemsetها استخراج کند. برای این کار از متد association_rules از کتابخانه mlxtend استفاده کردم. با توجه به این که ابتدا confidence را برابر با 0.1 قرار دادم، تعداد قوانین استخراج شده برابر با ۲ قانون بود. این مقدار confidence به این معناست که صرفا قوانینی نگه داشته می شوند که حداقل در 10% موارد، وقتی سمت چپ قانون رخ داده، سمت راست قانون هم رخ دهد. طبیعی است که با افزایش مقدار confidence، تعداد قوانین استخراج شده کاهش پیدا میکند.

برای مثال با مقادیر confidence مختلف، تعداد قوانین استخراج شده به شرح زیر است:

```
Min Confidence: 0.1 → 2 rules
Min Confidence: 0.3 → 2 rules
Min Confidence: 0.5 → 1 rules
Min Confidence: 0.7 → 1 rules
Min Confidence: 0.9 → 0 rules
```

که همانطور که مشخص است تعداد قوانین با افزایش confidence کاهش پیدا میکند.

طبق گفته سوال من مقدار confidence را برابر با 0.7 قرار دادم که تنها یک قانون استخراج شد که احتمالا دلیل تعداد کم قوانین استخراج شده، تعداد محدود داده هاست.

قانونی که با این مقدار confidence به دست آمده قانون زیر است:

(22328) => (22326)

آیتم ۲۲۳۲۶، ست ۴ تایی ظرف خوراکی با طرح جنگل است.

آیتم ۲۲۳۲۸، ست ۴ تایی ظرف خوراکی با طرح میوه ای است.

و مقادیر محاسبه شده برای این قانون به شرح زیر است:

Support: 6.1%

که به این معناست که این قانون در ۶.۱ درصد از کل تراکنش ها صدق میکند

Confidence: 72.95%

که به این معنی است که دز ۷۲.۹۵ درصد از مواقعی که آیتم ۲۲۳۲۸ خریداری شده، ۲۲۳۲۶ هم خریداری شده است.

Lift: 5.42

که به این معنی است که احتمال خرید ۲۲۳۲۶ وقتی ۲۲۳۲۸ خریده شده، بیش از ۵ برابر است.

Kulczynski: 0.59

که به این منی است میانگین confidence دو طرفه برای این دو آیتم برابر با ۰.۵۹ است. این عدد به این معناست که نمی توان انتظار داشت برعکس این قانون هم اتفاق بیفتد، یعنی حدودا در ۴۱ درصد مواقع نمیتوان انتظار داشت که مشتر با خرید آیتم ۲۲۳۲۶، آیتم ۲۲۳۲۸ را هم خریداری کند.

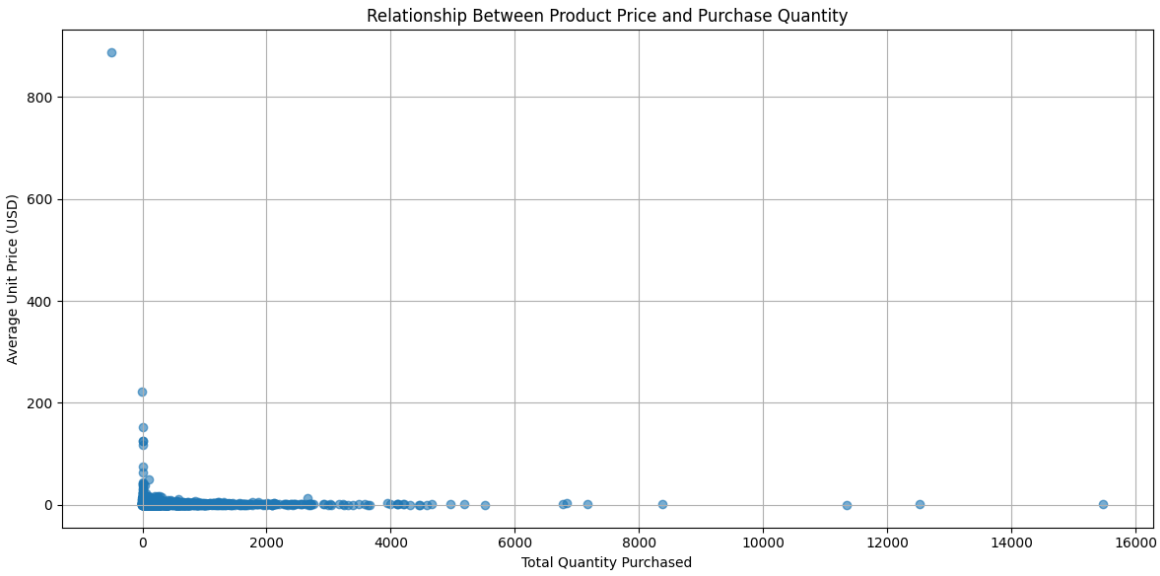
اما در حالت کلی، از آنجایی که confidence برای این قانون درصد قابل توجهی است، به طور کلی میتوان انتظار داشت که این قانون معتبر بوده و برقرار باشد.

.4

در این مرحله نیز برای تحلیل ارتباط بین میزان فروش و قیمت، ابتدا پروفروش ترین محصولات را انتخاب کردم و جدول تعداد فروش ۱۰ محصول اول با قیمت هایشان را چاپ کردم:

	Description	Quantity	UnitPrice
1956	RABBIT NIGHT LIGHT	15478	1.923436
1522	MINI PAINT SET VINTAGE	12517	0.642727
1687	PACK OF 72 RETROSPOT CAKE CASES	11337	0.534000
2496	SPACEBOY LUNCH BOX	8373	1.893284
726	DOLLY GIRL LUNCH BOX	7161	1.884194
2144	ROUND SNACK BOXES SET OF4 WOODLAND	6829	2.895397
2048	RED TOADSTOOL LED NIGHT LIGHT	6774	1.631633
2853	WORLD WAR 2 GLIDERS ASSTD DESIGNS	5521	0.286923
1903	PLASTERS IN TIN WOODLAND ANIMALS	5178	1.635650
1900	PLASTERS IN TIN SPACEBOY	4954	1.629348

همچنین در مرحله بعد نمودار مربوط به تعداد فروش و قیمت محصولات را در قالب یک scatter plot چاپ کردم:



نمودار 1

در این نمودار، محور افقی نشان دهنده تعداد کل خرید از هر محصول است و محور عمودی، نشان دهنده میانگین قیمت هر محصول می باشد.

الگوی اصلی نمودار نشان دهنده رابطه معکوس بسیار قوی بین قیمت و تعداد فروش کالا است. در این نمودار بیشترین نقاط فروش در ناحیه ای هستند که تعداد خرید بالاست ولی قیمت پایین است. یعنی محصولات ارزان زیاد خریداری شده اند)

محصولات با قیمت بالا مثلا بین ۱۰۰ تا بالای ۸۰۰ دلار، فقط در تعداد بسیار پایین فروخته شده‌اند. (معمولا زیر ۱۰ تا ۲۰ عدد)

نتیجه گیری نهایی: در کل میتوان به این نتیجه گرفت که در این فروشگاه، محصولاتی با قیمت پایین تر فروش بسیار بالاتری داشته اند. در این فروشگاه، رفتار مشتریان به وضوح تحت تاثیر قیمت قرار دارد و ارزان بودن کالا نقش کلیدی در افزایش فروش دارد. همچنین کالاهای گران قیمت تقریبا فروش بسیار پایینی داشته اند و بعضا ممکن است فقط یک بار خریداری شده باشند.

اظهارنامه استفاده از هوش مصنوعی

تأیید می‌کنم که از ابزارهای هوش مصنوعی مطابق با دستورالعمل‌های بارگذاری شده در سامانه Elearn درس به طور مسئولانه استفاده کرده‌ام. تمام اجزای کار خود را درک می‌کنم و آماده بحث شفاهی درباره آنها هستم.