به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و

کامپیوتر

درس داده کاوی پیشرفته

تمرین دوم

نام و نام خانوادگی	عرفان شهابی
شماره دانشجویی	۸۱۰۱۰۳۱۶۶
تاریخ ارسال گزارش	۱۴۰۴.۰۱.۲۰

فهرست

نمودارها

جدول‌ها

سوال ۱

الف:

برای هر subset از ویژگی ها یک cuboid متناظر وجود دارد. اگر تعداد ویژگی ها برابر با ۱۰ باشد، تعداد کل ترکیب های ممکن از این ویژگی ها یعنی هر subset از ویژگی ها، با یا بدون بعضی از آن ها برابر است با:

$$2^n = 2^{10} = 1024$$

ب:

سلول aggregate یعنی ترکیب سلول‌هایی با داده‌های خاص (یعنی همون p و qها) که در یک یا چند بعد تعمیم (generalization) داده شدند؛ به عبارتی، جایگزینی بعضی ویژگی‌ها با *.

ما دو سلول پایه داریم که هر کدام دارای مقادیر مشخص ۲۲ و ۱۴ هستند. از هر سلول پایه، می توانیم با تعمیم ویژگی ها، ۱۰۲۴ سلول aggregate بسازیم. ولی از این ۱۰۲۴ سلول، ۱ مورد خودش است یعنی سلول خالی و ۱ مورد هم سلول all است درنتیجه تعداد سلول های aggeregate برابر است با:

1024-2=1022

و از آن جا که دو سلول داریم این مقدار دو برابر شده و ۲۰۴۴ سلول aggregate غیر بدیهی وجود دارد.

ج:

سلول بسته سلولی است که هیچ سلول تعمیم یافته دیگری با همان مقدار aggregate value نداشته باشد. در نتیجه دو سلول پایه بسته هستند چرا که سلول تعمیم یافته ای از این دو سلول وجود ندارد. همچنین سلولی که به جز دو بعد یکسانی که در این دو سلولی که هستند، نیز بسته می باشد. در نتیجه سلول های بسته برابر است با:

(p1, p2, q3, p4, q5, p6, q7, p8, p9, p10) 22

(q1, q2, q3, q4, q5, q6, q7, q8, q9, q10) 14

(*, *, q3, *, q5, *, *, *, *, *) 36

د:

Iceberg cube سلول هایی است که مقدار آن ها از minimum support بیشتر باشد. با توجه به این که minimum support برابر ۲۵ است،هیچ کدام از دو سلول پایه در iceberg cube قرار نمی گیرند. با توجه به این موضوع سلول های که هر دو سلول پایه را تشکیل می دهند می توانند در iceberg cube قرار گیرند که مقدار آن ها برابر با ۳۶ است.

پس سلول های aggregate غیر تهی در iceberg cube برابر هستند با:

(*, *, q3, *, q5, *, *, *, *, *) 36

(*, *, q3, *, *, *, *, *, *, *) 36

(*, *, *, *, q5, *, *, *, *, *) 36

(*, *, *, *, *, *, *, *, *, *) 36

سوال ۲

الف:

Cuboidهای ممکن در این data cube:

(Time, Hospital, Department) Base

(*, Hospital, Department)

(Time, *, Department)

(Time, Hospital, *)

(*, *, Department)

(*, Hospital, *)

(*, *, Department)

(*, *, *) All

ب:

۱)

ابتدا از Cuboid پایه شامل ابعاد

(Time, Hospital Branch, Department)

شروع می‌کنیم. سپس با انجام عملیات Roll-up روی ابعاد Time و Hospital Branch، به Cuboid

(Department, *, *)

می‌رسیم که تعداد کل بیماران درمان شده در هر بخش را نشان می‌دهد.

در این Cuboid، مجموع بیماران برای هر Department را محاسبه می‌کنیم. پس از تعیین بخش پرترافیک، با استفاده از Slice روی همان بخش (مثلاً Cardiology) به Cuboid زیر میرسیم:

(Time, Hospital Branch, Cardiology)

در ادامه، برای یافتن ماهی که در آن، این بخش بیشترین بیمار را درمان کرده، عملیات Drill-down روی بعد Time انجام می‌شود (در صورت وجود داده در سطح روز یا هفته). سپس با مشاهده و مقایسه مقادیر، متوجه می‌شویم که مثلاً در بخش Cardiology بیشترین بیماران در ماه February درمان شده‌اند.

۲)

ابتدا از Cuboid پایه استفاده می‌کنیم. سپس با اعمال عملیات Slice روی مقدار Hospital Branch برابر با New York و Los Angeles، و همچنین عملیات Dice برای انتخاب فقط بخش‌های Cardiology و Neurology، داده‌ها را محدود می‌کنیم.

سپس عملیات Drill-down روی بعد Time برای رسیدن به سطح ماه انجام می‌شود (در این مثال، ماه‌های January تا March). در نهایت با استفاده از Pivot Table یا Trend Analysis، روند تغییر تعداد بیماران در این دو بخش و دو بیمارستان طی سه ماه بررسی می‌شود. این تحلیل امکان شناسایی رشد، افت یا پایداری تعداد بیماران را فراهم می‌سازد.

۳)

در این بخش، هدف مقایسه میانگین تعداد بیماران در دو سلول مشخص از Cube است. برای این منظور:

ابتدا از Cuboid پایه شروع می‌کنیم. با استفاده از Slice ابتدا داده‌های مربوط به Time = March و Hospital Branch = New York را جدا می‌کنیم و از آن میانگین بیماران را استخراج می‌کنیم.

سپس با یک Slice دیگر، داده‌های Time = Januaryو Hospital Branch = Los Angeles

را جدا کرده و میانگین بیماران در آن را نیز محاسبه می‌کنیم.

در نهایت، با مقایسه دو مقدار Aggregated، تحلیل نهایی انجام می‌شود. (Aggregation در اینجا میانگین یا AVG است.)

سوال ۳

الف:

۱) برای انتخاب بهترین ترتیب در الگوریتم BUC، باید ابعادی که کاردینالیتی کمتری دارند زودتر پردازش شوند. چرا که اگر از ابعادی با تنوع زیاد شروع کنیم، تعداد مسیر های محاسبه شده به صورت نمایی زیاد می شود.

شمارش کاردینالیتی برای هر بعد به صورت زیر است:

Dimension	Cardinality	Num
Department	Cardiology, Neurology, Orthopedic, Oncology	4
Education Level	Bachelor, Master, PhD	4
Specialization	Cardiologist, Neurologist, Orthopedic, Oncologist	4

جدول 1

پس ترتیب مناسب برای اجرای الگوریتم BUC به صوزت زیر است:

Education Level => Specialization => Department

۲) در الگوریتم BUC بهتر است ابعادی که دارای مقدار کاردینالیتی کمتری هستند، زود تر بررسی شوند تا درخت محاسباتی BUC شاخه های کمتری داشته باشد و سریع تر prune شود.

۳) دلیل این انتخاب این است که چون در مراحل ابتدایی، تعداد گروه بندی ها کمتر است و الگوریتم زود تر می تواند مسیرهایی را که تعداد نمونه کافی ندارند را خذف کند. این باعث می شود که شاخه های زیادی زود تر prune شوند و الگوریتم ادامه پیدا نکند. در نتیجه اجرای آن بسیار سریع تر خواهد بود.

ب:

۱) تعدا کل داده ها برابر با ۱۲ رکورد است. همانطور که گفته شد اجرای الگوریتم BUC با ترتیب زیر بهینه است:

Education Level => Specialization => Department

به همین دلیل الگوریتم را با تقسیم داده ها بر اساس Education Level شروع میکنیم. در مرحله بعد برای هر پس از آن، برای هر مقدار این بعد، الگوریتم به‌صورت بازگشتی وارد بعد بعدی یعنی Specialization شده و در ادامه به بعد سوم یعنی Department می‌رسد. اگر در هر مرحله، تعداد رکوردهای موجود برای یک مسیر کمتر از حداقل minimum support (مثلاً ۲) باشد، آن مسیر قطع

می‌شود و ادامه پیدا نمی‌کند؛ مثل حالت Bachelor که تنها شامل یک رکورد بود و از مسیر حذف شد.

در نهایت سلول هایی که شرط minimum support را پاس میکنند و هرس هم نشده اند برابر است با:

Educatio Level	Specialization	Department	Num
Phd	*	*	4
Phd	Neurologist	*	2
Phd	Neurologist	Neurology	2
Phd	*	Neurology	2
Master	*	*	6
Master	Cardiologist	*	2
Master	Orthopedic	*	2
Master	*	Cardiology	2
Master	*	Orthopedic	2
Master	Cardiologist	Cardiology	2
Master	Orthopedic	Orthopedic	2
*	Cardiologist	*	3
*	Orthopedic	*	3
*	Neurologist	*	3
*	Oncologist	*	2
*	*	Cardiology	3
*	*	Orthopedic	3
*	*	Neurology	3
*	*	Oncology	2
*	Cardiologist	Cardiology	3
*	Orthopedic	Orthopedic	3
*	Neurologist	Neurology	3
*	Oncologist	Oncology	2
*	*	*	11

جدول 2

۲) در حالت Full Cube، تمام سلول‌هایی که در داده‌ها حداقل یک بار ظاهر شده‌اند در نظر گرفته می‌شوند. با توجه به جدول و ترکیب سه بعد (Department، Education Level و Specialization)، مجموع سلول‌های ممکن برابر است با:

$$4 \times 5 \times 5 = 100\ Cells$$

از طرفی بین Department و Specialization رابطه یک به یک وجود دارد و طبیعتا تعداد سلول های صحیح کمتر از ۱۰۰ است. با درنظر گرفتن سلول‌های واقعی (یعنی آن‌هایی که حداقل یک نمونه دارند)، تعداد کل سلول های غیر تهی درFull Cube برابر 40 سلول بوده است.

از این میان، با اعمال شرط min support = 2، تنها 24 سلول در Iceberg Cube باقی مانده‌اند.

در نتیجه:

$$42 - 24 = 16, \qquad \frac{16}{40} \times 100 = \ 40\%$$

این کاهش نشان می دهد که استفاده از Iceberg Cube به طور مؤثری باعث کاهش اندازه داده ها می‌شود و تمرکز تحلیل را روی الگوهای پرتکرارتر و معنادارتر قرار می دهد. این موضوع در پایگاه های داده بزرگ یا چندبعدی اهمیت بالایی دارد، زیرا رشد نمایی تعداد سلول ها در چنین شرایطی می‌تواند محاسبات را دشوار و منابع را درگیر کند.

سوال عملی

1 . طراحی اسکیماهای Snowflake و Star در Pandas

الف:

اسکیمای Star:

در این بخش هدف، ساخت star schema برای سیستم data warehouse بر اساس مجموعه داده فروش سوپر مارکت ایت. این مدل شامل یک Fact Table و چندین Dimension Table است. ابتدا Dimension Table برای مشتری ها تشکیل شده است به این صورت که این جدول شامل نوع مشتری به عنوان عضو یا عادی و جنسیت است و به هر ترکیب یکتا از این دو ویژگی یک Customer_IDاختصاص داده شده است. سپس جدول Dimension Table به همین ترتیب برای مابقی ویژگی ها هم تشکیل می شوند.
سپس در مرحله بعد پس از ایجاد جداول ابعاد کلید های اصلی از جداول بعدی به داده های اصلی متصل می شوند. نتیجه این کار این است که جدول نهایی آماده سازی شده که شامل شناسه ها یکتا برای هر بعد است.
سپس در مرحله بعدی به سراغ ساخت جدول Fact Table می رویم. این جدول شامل متغیر های کلیدی مربوط به هر تراکنش فروش است. ست.ن های عددی مانند Marginr, COGs, Quantity, Ratingدر این جدول ذخیره شده اند. کلید های خارجی برای اتصال به جداول dimension نیز وجود دارد. در نهایت هم چند سطر اول برای اطمینان از صحت ساختار داده چاپ می شوند.

اسکیمای Snowflake:

در ابتدا، Dimension Table مربوط به مشتریان ساخته شده است. این جدول شامل دو ویژگی جنسیت و نوع مشتری است و به هر ترتیب یکتای این دو ویژگی یک شناسه منحصر به فرد تحت عنوان Customer_ID اختصاص داده شده است. به همین ترتیب جداول ابعادی برای سایر ویژگی ها نیز ساخته می شود.

در مرحله بعد، Primary Key ها از جداول ابعاد به داده های اصلی متصل شده اند. برای این منظور، جدول اصلی به صورت گام به گام با هر جدول بعدی بر اساس ویژگی های مشترک marge شده است. در نتیجه این مرحله، تشکیل یک جدول تجمیعی است که در آن به هر رکورد، کلید های خارجی مرتبط با ابعاد اختصاص داده شده است. سپس در مرحله نهایی، جدول Fact Table ساخته می شود. این جدول شامل متغیر های کلیدی مرتبط با هر تراکنش فروش مانند Rating, gross income و ... و همچنین کلیدهای خارجی برای اتصال به جداول Dimension است. در مرحله نهایی جدول Fact Table تحت عنوان fact_table_snow آماده شده و در پایان چند سطر

اول آن نمایش داده می شود تا از صحت ساختار و اتصال ابعاد به این جدول اطمینان حاصل شود.

ب:

برای ارزیابی کارایی مدل سازی داده ها از منظر مصرف حافظه، میزان حافظه اشغال شده توسط دو مدل Star Schema و Snowflake Schema اندازه گیری شده است.

نتایج به دست آمده به شرح زیر است:

• Star Schema:

• مصرف حافظه: 399,502 بایت

• معادل با: 0.4 مگابایت

• Snowflake Schema:

• مصرف حافظه: 341,363 بایت

• معادل با: 0.34 مگابایت

با مقایسه این مقادیر مشخص می شود که Snowflake Schema حدود 58 کیلوبایت حافظه کمتری نسبت به Star Schema مصرف می‌کند.

این اختلاف به‌دلیل جدا کردن ویژگی های تکرا رشونده به جداول ابعادی نرمال شده در Snowflake است که موجب کاهش افزونگی داده و در نتیجه، کاهش مصرف حافظه می‌شود.

این مقایسه نشان می دهد که در شرایطی که کاهش مصرف حافظه یا فشرده‌سازی داده اهمیت دارد، استفاده از Snowflake Schem می‌تواند گزینه بهینه‌تری باشد.

2 . مقایسه سرعت اجرای عملیات گروه بندی در اسکیماهای Star و Snowflake

الف و ب:

هدف این بخش، مقایسه زمان اجرای عملیات aggregation در دو ساختار متفاوت Star Schema و Snowflake Schema است. برای این منظور،مجموع فروش Total برای هر Product line را محاسبه کردیم.

برای مدل star schema ابتدا جدول fact table آن مستقیما با جدول product_dim_star ادغام شده است. ستون product line به صورت مستقیم در جدول ابعاد وجود دارد و عملیات groupby بلافاصله روی آن انجام شده است.

برای مدل snowflake نیز ابتدا جدول fact table آن با جدول product_dim ادغام شده است. سپس برای دستیابی به نام محصول، یک join اضافی با جذول product_line_dim نیز انجام شده است. پس از آن عملیات groupby روی ستون product line انجام شده است. نتایج به شرح زیر است:

Star Schema: 0.003571 Seconds

Snowflake Schema: 0.005694 Seconds

همانطور که از نتایج مشخص است مدل star سریع تر از مدل snowflake عمل کرده است. این تفاوت به دلیل تعداد کمتر join در مدل star است. در star schema، ویژگی هایی مانند product_line مستقیما در جدول ابعاد ذخیره شده اند در حالی که در snowflake schem نیاز به join های زنجیره ای برای دستیابی به همان ویژگی ها است. اگر چه snowflake از نظر نرمال سازی و بهینه سازی فضای ذخیره سازی عملکرد بهتری دارد، اما در شرایطی که هدف انجام تحلیل سریع و مکرر است، مدل star کارایی بهتری از نظر زمان پاسخ دارد.

3 . محاسبه Roll-up: تحلیل فروش در سطوح مختلف

الف:

هدف این بخش تحلیل روند فروش در طول زمان با استفاده از Fact Table و جدول DateTime Dimension در ساختار star schema است.

برای این بخش ابتدا جدول fact_table_star با جدول datetime_dim_star بر اساس کلید DateTime_ID ادغام شد. سپس ستون Date به فرمت تاریخ میلادی (DateTime) تبدیل گردید تا امکان استخراج اجزای زمانی مانند سال و ماه فراهم شود.

سپس با استفاده از تابع dt در پانداس ستون های جدی Year و Month از تاریخ استخراج شدند. همچنین شماره ماه (Month_Num) نیز برای مرتب سازی منطقی رکورد ها استفاده شدند.

در مرحله بعد مجموع فروش برای هر ترکیب Year و Month را نحاسبه کردم و برای حفظ ترتیب ماه ها، جدول نهایی بر اساس Year و Month مرتب گردید.

در مرحله بعد برای هر سال موجود در داده ها، مجموع فروش به صورت جداگانه محاسبه شده و در جدول Yearly_sales ذخیره گردد.

در نهایت نیز نمودار ها را رسم کردم.

ب:

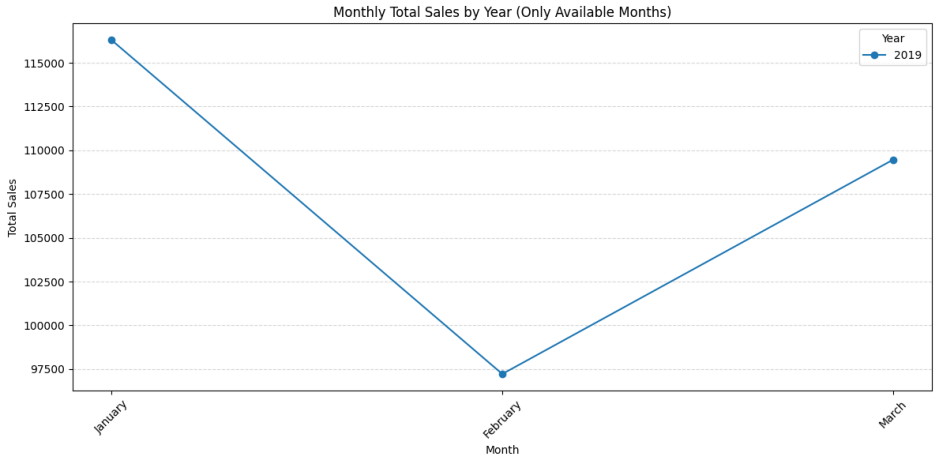
نتایج این تحلیل به شکل زیر است:

```
Monthly Sales (only for available months):
   Year     Month       Total
1  2019   January  116291.868
0  2019  February   97219.374
2  2019     March  109455.507
```

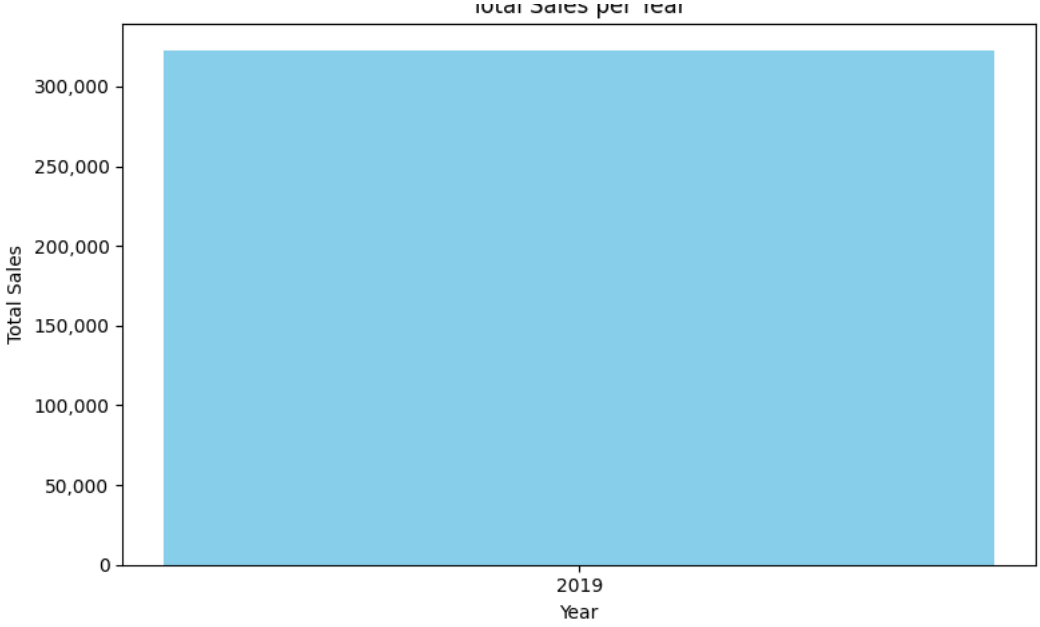
همانطور که از نتایج مشخص است از ماه ژانویه تا مارس به صورت کلی کاهش بوده است

ج:

طبق گفته سوال نمودار های مربوطه برای تغییرات در سطح ماه و سال به شرح زیر است:



نمودار 1



نمودار 2

برای تغییرات در سطح ماه میتوان مشاهده کرد که در ماه ژانویه این فروشگاه بیشترین میزان فروش برای ماه ژانویه بوده است که میتواند به دلیل شروع سال جدید میلادی و شروع تعطیلات باشد. همچنین مجموع فروش در ماه فوریه کاهش پیدا کرده و در ماه مارس افزایشی بوده اما نتوانسته از ماه ژانویه رد شود.

با توجه به اینکه تنها داده های یکسال در دیتاست موجود است، نمودار تغییرات سالانه صرفا مجموع فروش را در طی سال ۲۰۱۹ به ما می دهد.

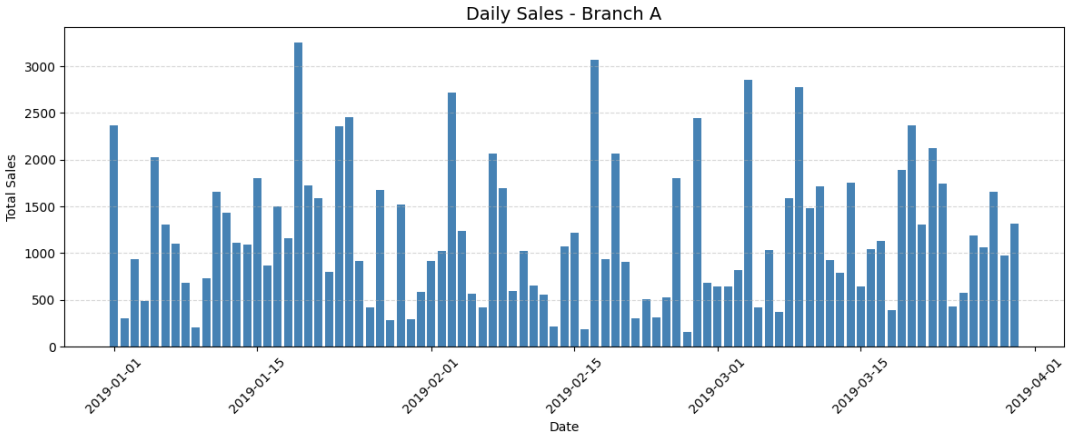
4 . محاسبه Drill-down: تحلیل فروش روزانه

الف:

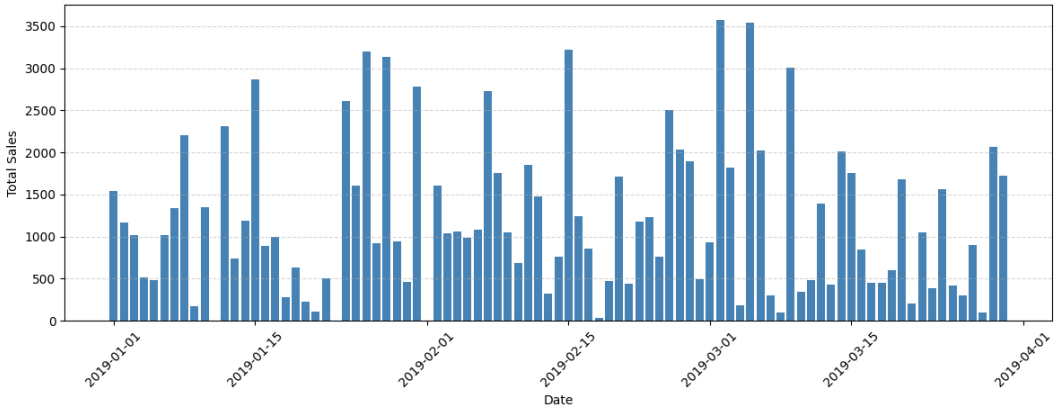
هدف این بخش بررسی و نمایش روند فروش روزانه برای هر شعبه فروشگاه با استفاده از داده های مدل شده در قالب Star Schema است. در این تحلیل، فروش هر شعبه به تفکیک روز محاسبه شده و هم به صورت متنی و هم نموداری نمایش داده می شود.

در ابتدا جدول واقعیت fact_table_star با جدول بعد شعبه branch_dim_star برای دریافت نام هر شعبه و با جدول بعد زمان datetime_dim_star برای استخراج تاریخ تراکنش ادغام شده اند.

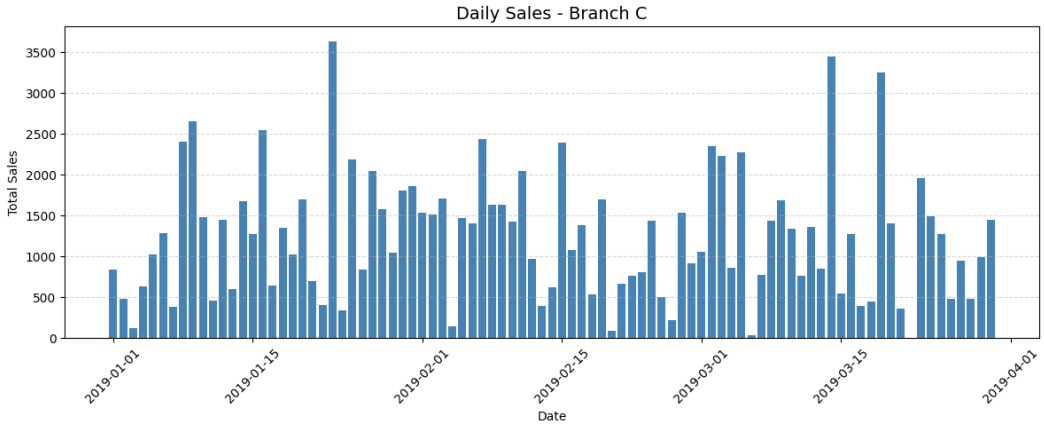
سپس برای امکان گروه بندی زمانی، ستون Date به فرمت تاریخ میلادی (datetime) تبدیل شده است. در مرحله بعد مجموع فروش برای هر ترکیب از Data و Branch محاسبه شده است و سپس نتایج بر اساس تاریخ و نام شعبه مرتب سازی شدند و خروجی اطلاعات برای هر روز و هر شعبه چاپ شدند. همچنین نمودار های مربوط به فروش روزانه هر شعبه نیز رسم شدند که در زیر قابل مشاهده اند.



نمودار 3



نمودار 4



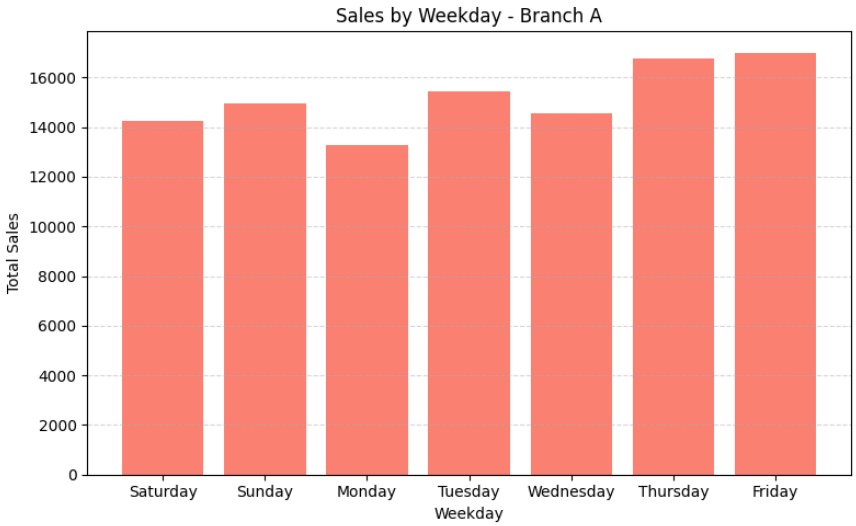
نمودار 5

ب:

برای این قسمت من شعبه A را در نظر گرفتم و بیشینه و کمینه فروش برای این شعبه به صورت زیر است:

```
Highest sale day in branch A: 2019-01-19 → 3254
Lowest sale day in branch A: 2019-02-26 → 156
```

همچنین مجموع فروش بر اساس روز های هفته بر این شعبه نیز نمایش داده شده است:



نمودار 6

به نظر من نتیجه ای که میتوان از نمودار مجموع فروش در روز های هفته گرفت میتواند این باشد که این فروشگاه در روز های منتهی به اخر هفته، یعنی روز های پنج شنبه و جمعه بیشترین فروش را در طی هفته داشته است که میتواند به دلیل تعطیلات آخر هفته باشد.

5 . تحلیل فروش در شهر ها و شعب با Data Cube

الف:

هدف از اجرای این کد، ساخت یک OLAP Cube دو بعدی ساده با استفاده از Pivot Table در پایتون است تا مجموع فروش را به تفکیک شهر و شعبه نمایش دهد.

از تابع pd.pivot_table کتابخانه Pandas استفاده شده است. ستون مقدار (value) برابر با Total که نشان دهنده مجموع فروش هر رکورد است. سطر ها بر اساس نام شهر و ستون ها بر اساس نام شعبه تعریف شده اند. سپس از تابع sum برای تجمیع فروش استفاده شده است و یک ردیف و یک ستون اضافی به جدول اضافه شده است که جمع کل را برای هر ردیف و ستون نمایش می دهد که حاصل به شکل زیر است:

Branch City	A	B	C	sum
Mandalay	NaN	106197.672	NaN	106197.6720
Naypyitaw	NaN	NaN	110568.7065	110568.7065
Yangon	106200.3705	NaN	NaN	106200.3705
sum	106200.3705	106197.672	110568.7065	322966.7490

جدول 3

ب:

در این قسمت نیز با استفاده از sum(axis=1) مجموع فروش در هر ردیف یعنی هر شهر محاسبه شده است. و شهری که بیشترین فروش را داشته با استفاده از idmax() شناسایی ش و در متغیر top_city ذخیره شده است.

برای محاسبه فروش کل به تفکیک شعبه از ردیف sum در pivot table مجموع فروش مربوط به هر شعبه استخراج شده است. سپس ستون sum حذف شد تا فقط شعبه ها باقی بمانند. شعبه با بیشترین فروش با استفاده از idmax() شناسایی شده و در متغیر top_branch ذخیره شده است که نتایج حاصله از این بخش به صورت زیر است:

```
City with maximum seles: Naypyitaw
Branch with maximum seles: C
```

ج:

در این بخش، با استفاده از مدل Star Schema، یک جدول pivot دو بعدی به‌منظور تحلیل مجموع فروش به تفکیک خط محصول (Product line) و شهر (City) ایجاد شده است. ابتدا جدول واقعیت fact_table_star با جدول های ابعاد branch_dim_star و product_dim_star ادغام شد تا ستون‌های مورد نیاز شامل City و Product line به داده ها افزوده شود. سپس با استفاده از pd.pivot_table، یک Data Cube ساخته شد که در آن سطرها نمایانگر خطوط محصول و ستون ها نمایان گر شهرها بودند و مقادیر داخل جدول مجموع فروش ها را نمایش می دادند. در ادامه، برای هر خط محصول، شهری که بیش ترین فروش را به خود اختصاص داده با استفاده از تابع idxmax() شناسایی شد. در نهایت، برای هر محصول، نام شهر و مقدار فروش آن به‌صورت متنی چاپ شده است که نشان می دهد کدام شهر بهترین عملکرد فروش را برای هر خط محصول داشته است که نتایج به شرح زیر است:

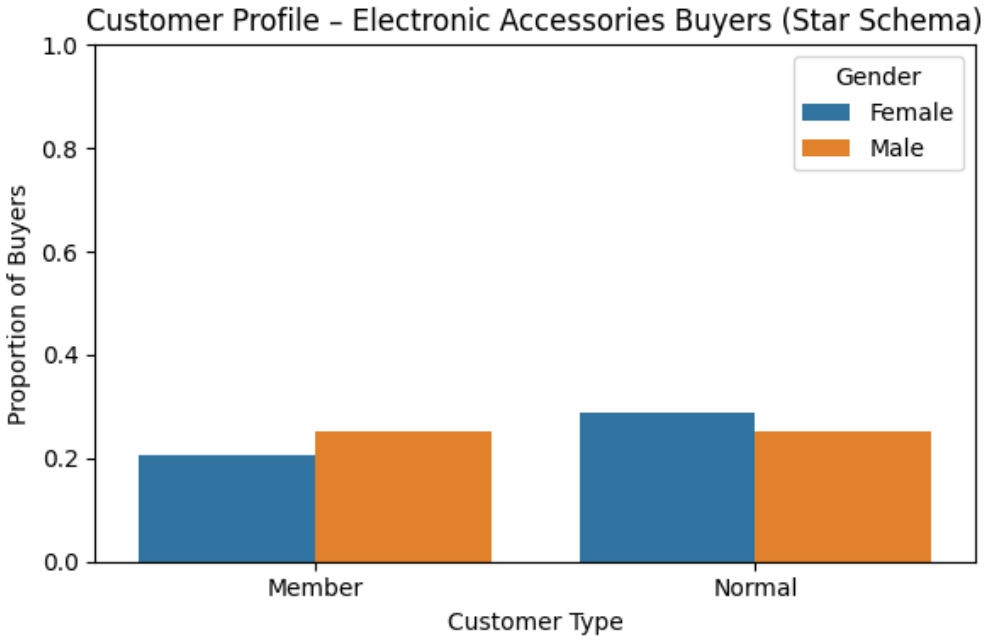
```
Total Sales by Product Line and City:
City                        Mandalay    Naypyitaw
Yangon        Total
Product line
Electronic accessories   17051.4435   18968.9745
18317.1135   54337.5315
Fashion accessories      16413.3165   21560.0700
16332.5085   54305.8950
Food and beverages       15214.8885   23766.8550
17163.1005   56144.8440
Health and beauty        19980.6600   16615.3260
12597.7530   49193.7390
Home and lifestyle       17549.1645   13895.5530
22417.1955   53861.9130
```

```
Sports and travel        19988.1990   15761.9280
19372.6995   55122.8265
Total                  106197.6720  110568.7065
106200.3705  322966.7490

Best-selling City per Product Line:
• Electronic accessories: Naypyitaw → 18968
• Fashion accessories: Naypyitaw → 21560
• Food and beverages: Naypyitaw → 23766
• Health and beauty: Mandalay → 19980
• Home and lifestyle: Yangon → 22417

• Sports and travel: Mandalay → 19988
```

6 . تحلیل رفتار مشتریان با Slice & Dice

الف:

در این بخش با استفاده از مدل Star Schema، پروفایل مشتریان محصولات مربوط به دسته "Electronic accessories" استخراج شده است. ابتدا جدول واقعیت با جدول ابعاد محصول ادغام شده تا نام خط محصول برای هر تراکنش مشخص شود. سپس تنها رکوردهایی که مربوط به محصولات "Electronic accessories" هستند فیلتر شده اند. در ادامه، اطلاعات مشتری شامل جنسیت و نوع مشتری (عضو یا عادی) از طریق اتصال به جدول مشتریان به رکوردها اضافه شده است. پس از آن، تعداد خریدها به تفکیک جنسیت و نوع مشتری محاسبه شده و نسبت هر گروه نسبت به کل خریدها نیز محاسبه گردیده است. در نهایت این توزیع با استفاده از نمودار میله ای نمایش داده شده تا مقایسه ای بصری از ترجیحات خریداران این دسته محصول ارائه شود.



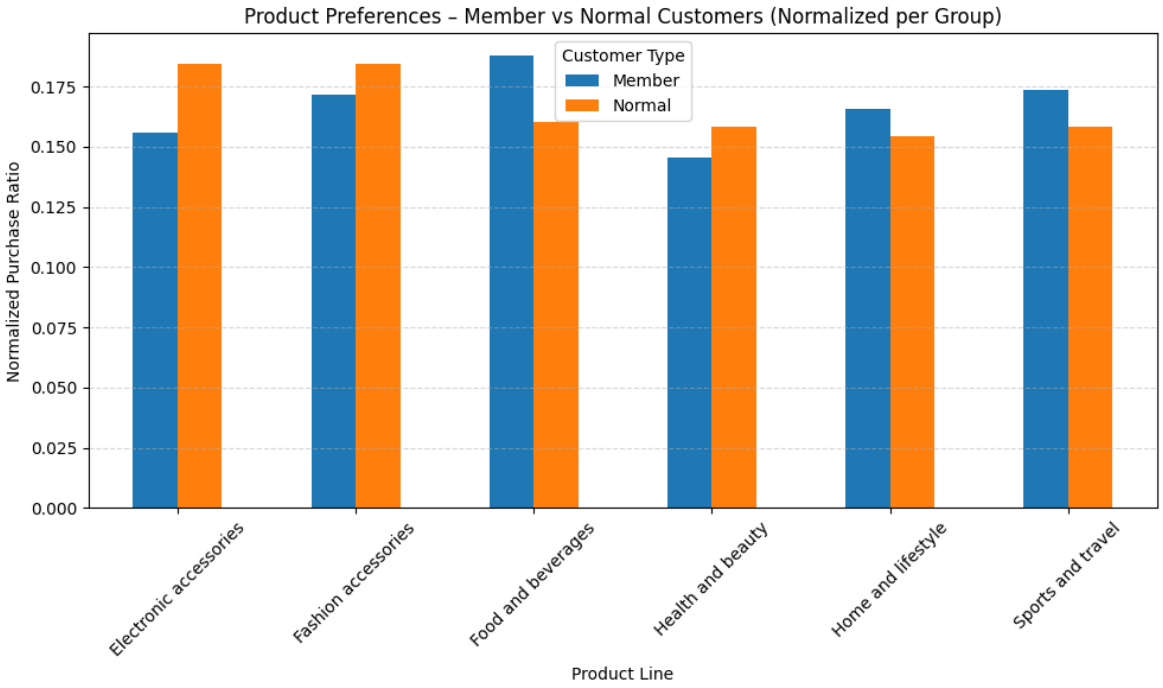
جدول 4

همانطور که از نمودار مشخص است بیشتر مشتریان این فروشگاه را افراد عادی تشکیل می دهد در صورتی که این اختلاف زیاد نیست. همچنین در بین مشتریان عضو جمعیت زنان کمی از مردان بیشتر است در حالی که این ترکیب برای مشتریان عادی بالعکس است. احتمالا فروشگاه بتواند با ارائه پیشنهاداتی مانند تخفیفات و یا پرداخت اقساطی و یا ذخیره اعتبارات به تعداد مشتریان عضو اضافه کند.

ب:

در این بخش با استفاده از مدل Star Schema، ترجیحات خرید مشتریان در دسته های مختلف محصول به تفکیک نوع مشتری (عضو یا عادی) بررسی شده است. ابتدا جدول واقعیت با جداول مشتری و محصول ادغام شده تا اطلاعات مربوط به نوع مشتری و خط محصول به هر رکورد فروش اضافه شود. سپس با گروه بندی بر اساس ترکیب "Customer type × Product line"، تعداد خریدها برای هر گروه محاسبه شده است. به منظور مقایسه نسبی، تعداد خریدها درون هر گروه مشتری نرمال سازی شده و نسبت سهم هر خط محصول از کل خریدهای آن گروه به دست آمده است. در ادامه، جدول نهایی به صورت نمودار میله ای ترسیم شده تا ترجیحات نسبی مشتریان عضو و غیرعضو نسبت به خطوط مختلف محصول به صورت بصری قابل مقایسه باشد که نتایج به شرح زیر است:

```
Product Preference Ratio (per Customer Type) – Star Schema:
   Customer type            Product line  Count     Ratio
0         Member  Electronic accessories     78  0.155689
1         Member     Fashion accessories     86  0.171657
2         Member      Food and beverages     94  0.187625
3         Member       Health and beauty     73  0.145709
4         Member      Home and lifestyle     83  0.165669
5         Member       Sports and travel     87  0.173653
6         Normal  Electronic accessories     92  0.184369
7         Normal     Fashion accessories     92  0.184369
8         Normal      Food and beverages     80  0.160321
9         Normal       Health and beauty     79  0.158317
10        Normal      Home and lifestyle     77  0.154309

11        Normal       Sports and travel     79  0.158317
```



جدول 5

همانطور که از نتایج مشخص است مشتریان عضو بیشتر به دسته های مواد غذایی، لوازم خانه و لایف استایل و لوازم سفر و ورزشی علاقه مند هستند و نسبت اعضا بیشتر از مشتریان عادی است. دلیل این نتیجه به نظر من میتواند مثلا تخفیفاتی باشد که فروشگاه در این دسته ها برای اعضا در نظر گرفته است. همچنین این دسته ها نسبت به محصولات دیگر مصرفی تر هستند و مشتریان باید زودتر آن ها را جایگزین کند. مثلا ممکن از یک مشتری عضو در سال یک بار از این فروشگاه لپتاپ بخرد اما هر هفته باید مواد غذایی خود را تامین کند.

ج:

به نظرم با توجه به این که به جز دسته های مواد مصرفی، در بقیه دسته ها تعداد افراد عادی بیشتر است، احتمالا فروشگاه بتواند با ارائه تخفیفات و امتیازاتی افراد عضو را به خرید از این محصولات ترغیب کند و به نظر من این نتایج نشان میدهد که فروشگاه امتیازات کافی را برای مشتریان عضو در این دسته ها قائل نشده. همچین در حالت کلی فروشگاه باید بتواند که امتیازات و مزایای عضویت مشتریان را تشریح کند تا بتواند اعضای بیشتری را برای فروشگاه جذب کند.

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.