

گزارش تسک شماره {۱}

عنوان تسک:

{تسک آنالیز داده های بانکی }

تهیه کننده/تهیه کنندگان:

{پیمان دایی رضایی}

تاریخ: {۱۴۰۳/۱۱/۷}

۰. درباره ما

من یکی از علايقم کار کردن از بچگی حدس زدن کار دیگران بوده و تحلیل آينده بعد از آشنایی با هوش مصنوعی در طرح شهید بابایی علاقه زیادی به اين پیدا کردم و وارد این زمینه شدم

۱. مقدمه

در اين گزارش قصد دارم داده های مربوط به تراکنش های خريد مشتریان را بررسی کنم تا ببینم چه الگوهایی در خريد آنها وجود دارد. اطلاعات موجود شامل شناسه مشتری، تعداد تراکنش ها، مبلغ کل خريد، قیمت واحد، تاريخ و زمان خريد، و فروشگاهی که خريد از آن انجام شده است. هدف اصلی اين تحلیل، شناسایی مشتریان پرخرج و کم خرج، بررسی میزان تأثیر تخفیف ها بر خريد، و پیدا کردن روندهای زمانی خريد است. مثلاً می خواهم ببینم آیا خريدها در ساعت خاصی از روز یا روزهای خاصی از هفته بیشتر هستند؟ آیا برخی فروشگاهها عملکرد بهتری نسبت به بقیه دارند؟

در ادامه، ابتدا يك نمای کلی از داده ها ارائه می کنم، سپس دسته بندی مشتریان بر اساس میزان خريد را انجام می دهم. همچنین بررسی می کنم که کدام مشتریان وفادارتر هستند و چگونه می توان رفتار آنها را بهتر درک کرد. در پایان هم نتایج تحلیل ها را خلاصه کرده و پیشنهاداتی برای بهبود فروش ارائه خواهم داد.

۲. فعالیت های انجام شده

۱، ۱ یافتن راه حل

{ ۱. بررسی و درک داده ها }

- مشاهده ساختار کلی دیتابست و شناسایی ستون های کلیدی مانند `transaction_id`, `user_id`, `store` و `transaction_date`, `unit_price`, `total_price`
- بررسی وجود مقادیر گمشده (**null values**) در دیتابست و تصمیم گیری درباره حذف یا جایگزینی آنها
- بررسی توزیع آماری داده ها از جمله حداقل، حداکثر، میانگین، میانه و صدک های کلیدی برای ستون های عددی

۲. پیش پردازش داده ها

- تبديل داده های تاریخی به فرمت استاندارد برای تحلیل های زمانی
- محاسبه فاکتورهای جدید مانند مقدار تخفیف (تفاوت بین `total_price` و `unit_price`)

۳. محاسبه شاخص‌های کلیدی و تحلیل رفتار خرید مشتریان

- محاسبه تعداد خریدهای هر مشتری با استفاده از `transaction_id.nunique()` برای هر `user_id`.
- محاسبه مجموع مبلغ پرداخت شده توسط هر مشتری برای هر `user_id` جمع شده با `total_price`.
- شناسایی مشتریان پرخرج و کم خرج براساس صدکهای آماری خریدها.
- بررسی توزیع تعداد خریدها و مقایسه رفتار خرید مشتریان مختلف.

۴. دسته‌بندی مشتریان براساس رفتار خرید

۵. تحلیل روندهای زمانی خرید

- بررسی الگوی خرید در روزهای هفته (آیا مشتریان بیشتر در آخر هفته خرید می‌کنند؟)
- تحلیل پیک‌های زمانی خرید (بررسی ساعات پر تکرار خریدها)
- مقایسه میزان خرید در فصول مختلف سال (آیا در زمان‌های خاصی خرید افزایش می‌یابد؟).

۶. تحلیل عملکرد فروشگاه‌ها

- بررسی میزان فروش هر فروشگاه و مقایسه آن‌ها
- شناسایی فروشگاه‌هایی که بیشترین مشتریان را جذب کرده‌اند
- بررسی رفتار خرید مشتریان در هر فروشگاه (آیا مشتریان وفاداری وجود دارند که مرتباً از یک فروشگاه خاص خرید کنند؟)

۲. داده‌ها

۱) شرح داده‌ها:

داده‌های موجود شامل ۸۱۹۲ ردیف و ۱۱ ستون هستند که هر ردیف، نشان‌دهنده‌ی یک تراکنش خرید است. این مجموعه داده اطلاعات مختلفی مانند شناسه مشتری، شناسه تراکنش، زمان و مکان خرید، مشخصات محصول، مقدار خریداری شده و هزینه‌های پرداختی را در بر می‌گیرد.

۲) ساختار داده‌ها و نوع داده‌ها:

ستون	توضیحات	نوع داده
user_id	شناسه‌ی یکتای مشتری	int64
transaction_id	شناسه‌ی یکتای تراکنش	int64
transaction_date	تاریخ انجام تراکنش	object
transaction_time	ساعت انجام تراکنش	object
store	نام فروشگاهی که خرید از آن انجام شده است	object
product_name	نام محصول خریداری شده	object
quantity	تعداد خریداری شده از یک محصول	int64
unit_price	قیمت هر واحد محصول	float64
total_price	مجموع مبلغ پرداختی قبل از تخفیف	float64
total_amount	مبلغ نهایی پرداخت شده پس از تخفیف	float64
total_items	تعداد کل اقلام خریداری شده در تراکنش	int64

۳) مرتب سازی و پیش پردازش داده ها :

- برای تحلیل بهتر، فرمت تاریخ و زمان اصلاح شده است و ستون هایی برای ماه و روز و ساعت ایجاد شد
- ستون هایی مثل دسته بندی مشتریان بر اساس میزان خریدشان و با استفاده از الگوریتم FRM و ستون ها تخفیف ها انجام شد
- حدود ۴۰۷ ردیف کشف شد که داده های پرت دارند
- داده های Missing Value وجود نداشت

۴) توضیحات داده ها :

۱) بررسی وجود داده های پرت در دیتاست :

ستون	توضیحات	داده پرت
user_id	شناسه منحصر به فرد هر کاربر.	ندارد
transaction_id	شناسه منحصر به فرد هر تراکنش.	ندارد
quantity	تعداد کالاهای خریداری شده در هر تراکنش	بله، برای مقادیر زیاد ممکن است داده های پرت وجود داشته باشد

ستون	توضیحات	داده پرت
unit_price	قیمت هر واحد کالا	بله، برای قیمت‌های غیرمعمول می‌تواند داده‌های پرت وجود داشته باشد
total_price	مجموع قیمت هر تراکنش (تعداد کالا × قیمت واحد)	بله، داده‌های پرت برای تراکنش‌های با مبلغ بالا قابل مشاهده است
total_amount	مجموع مبلغ خرید (با احتساب تخفیف و مالیات)	بله، در صورتی که تراکنش‌های غیرعادی وجود داشته باشد
total_items	تعداد کل اقلام خریداری شده در یک تراکنش	بله، تعداد اقلام غیرمعمول می‌تواند نشان‌دهنده داده‌های پرت باشد

۲) توضیحات بیشتر در مورد داده‌ها :

ستون	پر تکرارترین مقدار	تعداد یکتا	تعداد تکرار پر تکرارترین مقدار	توضیح
store	15	Supermarket	973	به دلیل دسترسی آسان و رشد و تنوع محصولات و رشد فروش آنلاین مخصوصاً اسنپ شاپ و نکته مهم تر خرید از سوپرمارکت بیشتر به خرید روزانه و نیازهای فوری مربوط است. به عنوان مثال، مردم برای خرید مواد غذایی روزانه، نوشیدنی‌ها، و کالاهای مصرفی دیگر به سوپرمارکتها مراجعه می‌کنند پس ما با توجه به علاقه مردم به خرید از سوپرمارکت‌ها ما باید مورد توجه قرار بگیرد.
product_name	1446	Pet Store Product 10	24	در سال‌های اخیر، به ویژه در شهرهای بزرگ، نگهداری حیوانات خانگی (مانند سگ، گربه، پرندگان و غیره) به یک روند محبوب تبدیل شده است. این ممکن است باعث شود که محصولات مرتبط با حیوانات خانگی، مانند غذا، لوازم جانبی، و مراقبت‌های بهداشتی، تقاضای بالایی پیدا کنند یکی از مواردی که باید بهش توجه کنیم و با توجه به خرید بالای زیاد این

	توضیح	تعداد تکرار پر تکرارترین مقدار	تعداد یکتا پر تکرارترین مقدار	ستون
weekday	محصول که تقاضای زیادی دارد باید مورد توجه قرار بگیرد روز شنبه بیشترین تعداد تراکنش را دارد که میتواند به دلیل شروع هفته کاری و برنامه‌ریزی خریدهای هفتگی، تکمیل خریدهای عقب افتاده از تعطیلات جمعه، خرید عمده کسب وکارها، افزایش خریدهای سازمانی، دریافت حقوق و افزایش نقدینگی، تخفیف‌های ویژه فروشگاه‌ها، تهیه مایحتاج مدارس و دانشگاه‌ها باشد	1320	7	Saturday
product_name_re	یکی از نقاط هدفی که باید به آن توجه کرد عمده خرید های ایرانی ها از سوپر مارکت ها بوده است و با توجه به تعطیلات و ماه رمضان احتمال اینکه خرید ها از سوپر مارکت ها بیشتر شود هست احتمال دارد به خاطر تعطیلی هفته در شهریور ماه بوده که این تاریخ چندین بار تکرار شده است	973	15	Supermarket Product
transaction_date_shamsi	با توجه به اینکه ۸ ماه از سال در داده‌ها ثبت شده‌اند و اینکه به دلیل ماه مصادف بودن با ماه رمضان مردم خرید های زیادی در این ماه کرده‌اند و پس نتیجه می‌گیریم ماه رمضان یکی مهمترین نقاط هدفی هست که باید مد نظر ما باشد	70	215	1403-06-06
month_of_year	۲, ۳ پیاده‌سازی	1251	8	اردیبهشت

شرح کتابخانه‌ها و تکنولوژی‌های به کار رفته

- زبان برنامه‌نویسی : Python
- محیط توسعه : Jupyter Notebook / PyCharm

- کتابخانه‌های مورد استفاده :

- برای خواندن، پردازش و تحلیل داده‌ها → pandas
- برای عملیات عددی و مدیریت آرایه‌ها → numpy
- برای تجسم داده‌ها و رسم نمودارها → seaborn و matplotlib
- برای مدیریت و پردازش تاریخ و زمان → datetime
- برای پیاده‌سازی الگوریتم RFM و سایر تحلیل‌های داده → mlxtend

محیط توسعه :

- Jupyter Notebook : برای اجرای گام‌به‌گام کدها، مشاهده خروجی‌ها و مستندسازی تحلیل‌ها

ماژول‌ها و توابع پیاده‌سازی شده و نمودار‌ها :

کد ۱:

این دو خط کد وظیفه تبدیل تاریخ‌های شمسی به میلادی و سپس تبدیل به فرمت Pandas DateTime در DataFrame را بر عهده دارند :

```
:  
df["transaction_date_shamsi"] = df["transaction_date"].apply(lambda x: jdatetime.date.fromgregorian(year=x.year,  
month=x.month, day=x.day))
```

- .۱ df["transaction_date"] : این ستون شامل تاریخ‌های شمسی است.
- .۲ apply(lambda x: ...) : برای هر مقدار در این ستون، یک تابع اعمال می‌شود.
- .۳ jdatetime.datetime.strptime(x, "%Y-%m-%d") : تبدیل می‌کند.
- رشته‌ی تاریخ شمسی را به یک شیء تاریخ شمسی (jdatetime) تبدیل می‌کند.
- فرمت %Y-%m-%d یعنی داده‌های ورودی به شکل سال-ماه-روز (مثلًا 1403-06-10).
- .۴ .togagregorian() : تاریخ شمسی را به تاریخ میلادی تبدیل می‌کند.

کد ۲:

ستون product_name_re برای پیدا کردن تعداد کل محصولات بدون کد محصول :

```
df.groupby('product_name_re')['total_amount'].sum().sort_values(ascending=False).plot.barh()
```

۱. یک ستون جدید به نام "product_name_re" در دیتافریم ایجاد می‌کند
۲. از ستون "product_name" اعداد را حذف می‌کند و \d+ (یعنی تمام اعداد (یک یا چند رقم) را پیدا و حذف کند) : df["product_name"].str.replace(r"\d+", "", regex=True)
۳. فاصله‌های اضافی در ابتدا و انتهای متن باقیمانده را حذف می‌کند : .str.strip()

کد ۳:

مقدار هر سطر در این ستون نام ماه میلادی معادل تاریخ موجود در "transaction_date" خواهد بود:

```
df["month_of_year"] = df["transaction_date"].dt.month_name()
```

۱. df["transaction_date"] : این ستون قبلً به datetime تبدیل شده و شامل تاریخ‌های میلادی است.
۲. : .dt.month_name()
- ویژگی dt : مختص داده‌های تاریخی (datetime) در Pandas است.
- "January", "February", ...) نام کامل ماه را از تاریخ استخراج می‌کند (مثالاً).

کد ۴:

مقدار هر سطر در این ستون نام روز‌های هفته معادل تاریخ موجود در "transaction_date" خواهد بود:

```
df["day"] = df["transaction_date"].dt.day
```

۱. df["transaction_date"] : این ستون قبلً به datetime تبدیل شده و شامل تاریخ‌های میلادی است.
۲. : .dt.month_name()
- ویژگی dt : مختص داده‌های تاریخی (datetime) در Pandas است.
- نام کامل روز‌های هفته را از تاریخ استخراج می‌کند

کد ۵:

تعداد تراکنش‌های مربوط به هر ماه را محاسبه کرده و در قالب نمودار دایره‌ای نمایش می‌دهند

```
df["month_of_year"].value_counts().plot.pie(autopct='%1.1f%%')
```

۱. ستون مربوط به نام ماهها را از دیتافریم می‌گیرد `df["month_of_year"]`
۲. تعداد وقوع هر ماه را می‌شمارد (یعنی چند تراکنش در هر ماه ثبت شده است) `.value_counts()`
۳. از خروجی `.value_counts()` یک نمودار دایره‌ای رسم می‌کند `.plot.pie()`
۴. درصد هر دسته روی نمودار نوشته می‌شود `autopct='%.1f%%'`
۵. نمایش درصد با یک رقم اعشار `%1.1f`
۶. نشان دادن علامت درصد `%%` روی نمودار

کد ۶:

نکته: از این برای ستون دیگه هم استفاده شده است

```
df["product_name_re"] = df["product_name"].str.replace(r'\d+', '', regex=True).str.strip()
```

این کد برای حذف کد محصولات در فروشگاه هست با استفاده از REGEX

این کد یک نمودار میله‌ای از تعداد خریدهای انجام شده در ساعات مختلف روز را ترسیم می‌کند:

کد ۷:

```
hourly_purchases = df.groupby("hour")["transaction_id"].nunique().reset_index()
hourly_purchases.rename(columns={"transaction_id": "purchase_count"}, inplace=True)

plt.figure(figsize=(10, 5))
sns.barplot(data=hourly_purchases, x="hour", y="purchase_count", palette="coolwarm")
plt.title("Purchases by Hour of the Day")
plt.xlabel("Hour of Day")
plt.ylabel("Number of Purchases")
plt.show()
```

```
hourly_purchases =
df.groupby("hour") ["transaction_id"].nunique().reset_index()
```

این خط داده‌ها را بر اساس ساعت گروهبندی می‌کند و تعداد تراکنش‌های یکتا را برای هر ساعت محاسبه می‌کند و یعنی مشخص می‌کند که در هر ساعت چند خرید انجام شده است

```
hourly_purchases.rename(columns={"transaction_id": "purchase_count"} ,  
inplace=True)
```

نام ستون "transaction_id" را به "purchase_count" تغییر می‌دهد تا مشخص باشد که این ستون تعداد خریدها را نشان می‌دهد

```
plt.figure(figsize=(10, 5))
```

اندازه نمودار را ۱۰ در ۵ تنظیم می‌کند تا نمایش بهتری داشته باشد

```
sns.barplot(data=hourly_purchases, x="hour", y="purchase_count",  
palette="coolwarm")
```

یک نمودار میله‌ای رسم می‌کند که در آن محور افقی (x) ساعت‌های روز و محور عمودی (y) تعداد خریدهای ثبت شده در هر ساعت است و همچنین از پالت رنگی "coolwarm" برای زیبایی بیشتر نمودار استفاده شده است

```
plt.title("Purchases by Hour of the Day")
```

یک عنوان برای نمودار قرار می‌دهد: "تعداد خریدها در ساعات مختلف روز"

```
plt.xlabel("Hour of Day")  
plt.ylabel("Number of Purchases")
```

برچسب‌های محور را تنظیم می‌کند. محور افقی (x) به عنوان "ساعت‌های روز" و محور عمودی (y) به عنوان "تعداد خریدها" مشخص می‌شود

```
plt.show()
```

نمودار را نمایش می‌دهد

: کد ۸

این کد نشان می‌دهد که در بیشترین فروش برای چه ماه‌هایی بوده است :

```
df.groupby('month_of_year')[['total_amount']].sum().sort_values(ascending=False).plot.barh()
```

- داده‌ها را بر اساس ماه گروه‌بندی می‌کند : •
- مجموع مقدار خریدها را برای هر ماه محاسبه می‌کند : •
- مقادیر محاسبه شده را به صورت نزولی مرتب می‌کند، یعنی ماهی که بیشترین فروش را دارد در بالاترین رتبه قرار می‌گیرد : •
- یک نمودار میله‌ای افقی رسم می‌کند تا مقدار فروش هر ماه را نشان دهد : •

: کد ۹

```

def skew_df(dff, name):
    dff[f'{name}_reciprocal'] = (1 / (dff[name] + 1))
    dff[f'{name}_sqrt'] = np.sqrt(dff[name])
    dff[f'{name}_log'] = np.log1p(dff[name])
    name_df = skew(dff[name]) * 10
    name_reciprocal = skew(dff[f'{name}_reciprocal'])
    name_sqrt = skew(dff[f'{name}_sqrt'])
    name_log = skew(dff[f'{name}_log'])

    return name_df, name_reciprocal, name_sqrt, name_log

```

خط کد	توضیحات
def skew_df(dff, name):	تعريف تابع skew_df که دو ورودی می‌گیرد dff که دیتا فریم است و name که نام ستونی است که باید روی آن عملیات انجام شود.
dff[f'{name}_reciprocal'] = (1 / (dff[name] + 1))	ایجاد یک ستون جدید به نام {name}_reciprocal که معکوس مقادیر ستون name به علاوه ۱ است. این عمل برای جلوگیری از تقسیم بر صفر انجام می‌شود.
dff[f'{name}_sqrt'] = np.sqrt(dff[name])	ایجاد یک ستون جدید به نام {name}_sqrt که ریشه دوم مقادیر ستون name است.
dff[f'{name}_log'] = np.log1p(dff[name])	ایجاد یک ستون جدید به نام {name}_log که لگاریتم طبیعی مقادیر ستون name به اضافه ۱ است. برای جلوگیری از محاسبه لگاریتم صفر است.
name_df = skew(dff[name]) * 10	محاسبه انحراف از توزیع نرمال (skewness) برای ستون اصلی name و ضرب آن در ۱۰.
name_reciprocal = skew(dff[f'{name}_reciprocal'])	محاسبه انحراف از توزیع نرمال برای ستون معکوس {name}_reciprocal.
name_sqrt = skew(dff[f'{name}_sqrt'])	محاسبه انحراف از توزیع نرمال برای ستون ریشه دوم {name}_sqrt.
name_log = skew(dff[f'{name}_log'])	محاسبه انحراف از توزیع نرمال برای ستون لگاریتم {name}_log.
return name_df, name_reciprocal, name_sqrt, name_log	بازگشت مقادیر انحراف از توزیع نرمال (skewness) برای ستون اصلی و تغییرات مختلف آن (معکوس، ریشه دوم، لگاریتم).

```

top_products = df.groupby("product_name")["quantity"].sum().reset_index()
top_products = top_products.sort_values(by="quantity", ascending=False).head(30)

plt.figure(figsize=(12, 5))
sns.barplot(data=top_products, x="quantity", y="product_name", palette="magma")
plt.title("Top 10 Best-Selling Products")
plt.xlabel("Quantity Sold")
plt.ylabel("Product Name")
plt.show()

```

خط کد	توضیحات
گروهبندی دیتافریم (df) بر اساس نام محصولات (product_name) و محاسبه مجموع تعداد فروخته شده برای هر محصول با استفاده از sum() سپس نتیجه به دیتافریم جدید منتقل می شود.	
مرتبسازی محصولات بر اساس تعداد فروخته شده به (quantity)	
صورت نزولی (ascending=False) و انتخاب ۳۰ محصول پرفروش اول با استفاده از head(30).	

خط کد	توضیحات
plt.figure(figsize=(12, 5))	تعیین اندازه نمودار 5×12 اینچ.
sns.barplot(data=top_products, x="quantity", y="product_name", palette="magma")	رسم نمودار میله‌ای با استفاده از Seaborn:
کد ۱۱:	
کد های گروه بندی :	
<pre>customer_spending = df.groupby('user_id')['total_amount'].sum().reset_index() customer_spending["log_total_amount"] = np.log1p(customer_spending["total_amount"]) q1 = customer_spending["log_total_amount"].quantile(0.33) q2 = customer_spending["log_total_amount"].quantile(0.66) q3 = customer_spending["log_total_amount"].max() bins = [0, q1, q2, q3] labels = ["Low Spender", "Regular Spender", "VIP"] customer_spending["category"] = pd.cut(customer_spending["log_total_amount"], bins=bins, labels=labels) customer_spending</pre>	

خط کد	توضیحات
محاسبه لگاریتم طبیعی (مقدار کل خرید) total_amount با استفاده از تابع np.log1p	با استفاده از تابع np.log1p مقدار کل خرید total_amount محاسبه لگاریتم طبیعی می‌شود.
به طور خودکار ۱ را به مقدار داده اضافه np.log1p(customer_spending["total_amount"]) می‌کند تا از مشکلات مقدار صفر log_total_amount جلوگیری شود. نتیجه در ستون جدید ذخیره می‌شود.	می‌کند تا از مشکلات مقدار صفر log_total_amount جلوگیری شود. نتیجه در ستون جدید ذخیره می‌شود.
محاسبه صدک اول (Q1) از داده‌های log_total_amount که معادل ۳۳٪ پایین‌ترین مقادیر است.	محاسبه صدک اول (Q1) از داده‌های log_total_amount که معادل ۳۳٪ پایین‌ترین مقادیر است.

خط کد	توضیحات
<pre>q2 = customer_spending["log_total_amount"].quantile(0.66)</pre>	محاسبه صدک دوم (Q2) که معادل ۶۶٪ پایین‌ترین مقادیر است (نیمه بالا).
<pre>q3 = customer_spending["log_total_amount"].max()</pre>	پیدا کردن بیشترین مقدار در ستون log_total_amount. معادل ۱۰۰٪ بالاترین مقادیر است.
<pre>bins = [0, q1, q2, q3]</pre>	تعریف محدوده‌های تقسیم‌بندی برای دسته‌بندی مشتری‌ها (bins)

```
from itertools import combinations
from collections import Counter

user1=df[df['user_id']==1]

transactions = user1.groupby("transaction_id")["product_name"].apply(list)

product_combinations = Counter()
for items in transactions:
    product_combinations.update(combinations(sorted(items), 2))

product_combinations.most_common(10)
```

خط کد	توضیحات
	گروهبندی داده‌ها بر اساس شناسه تراکشن
	و (transaction_id)
	جمع‌آوری لیستی از نام محصولات
transactions = user1.groupby("transaction_id") ["product_name"].apply(list) t = (product_name) که در هر (t) تراکشن خریداری شده‌اند. نتیجه در متغیر transactions ذخیره می‌شود.	

خط کد	توضیحات
product_combinations = Counter()	ایجاد یک شیء Counter برای شمارش ترکیب‌های مختلف محصولات که در
for items in transactions:	تراکنش‌های مختلف خریداری شده‌اند از کتابخانه collections برای شمارش اشیاء مختلف استفاده می‌شود.
product_combinations.update(combinations(sorted(items), 2))	حلقه‌ای برای پیمایش از طریق تمامی تراکنش‌ها. در هر دور حلقة، لیستی از items محصولات خریداری شده در یک تراکنش است.
product_combinations.most_common(10)	برای هر تراکنش، تمام ترکیب‌های دو تایی از محصولات خریداری شده (با استفاده از combinations کتابخانه itertools) محاسبه می‌شود. ابتدا محصولات مرتب شوند (sorted(items))، سپس ترکیب‌ها شمارش می‌شوند و به product_combination اضافه می‌شوند.
	نمایش ۱۰ جفت محصولی که بیشترین تکرار را در ترکیب‌های خرید همزمان داشته‌اند. این جفت محصول با استفاده از متند most_common استخراج می‌شود.

```

from scipy.spatial.distance import cdist

scaler = StandardScaler()
product_scaled = scaler.fit_transform(customer_product.T)

distortions = []
for k in range(1, 10):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(product_scaled)
    distortions.append(sum(np.min(cdist(product_scaled, kmeans.cluster_centers_, 'euclidean'), axis=1)) / product_scaled.shape[0])

plt.figure(figsize=(8, 4))
plt.plot(range(1, 10), distortions, marker='o')
plt.xlabel("Number of clusters (k)")
plt.ylabel("Intra-cluster variation")
plt.title("Elbow diagram for determining the number of clusters")
plt.show()

```

خط کد

توضیحات

ایجاد یک شیء

برای StandardScaler

استانداردسازی داده‌ها (یعنی مقادیر ویژگی‌ها را به مقیاس یکسان تبدیل می‌کند تا از تأثیر تفاوت مقیاس‌ها جلوگیری شود).

استانداردسازی داده‌ها با استفاده از

بر روی fit_transform

ترانهاده (transpose) داده‌های customer_product به این

صورت که هر ویژگی مشتری در ابعاد مناسب تبدیل می‌شود.

ایجاد یک لیست خالی برای ذخیره

مقادیر "اختلالات"

که در هر تکرار (distortions) تعداد خوشها محاسبه می‌شود.

شروع حلقه‌ای برای پیدا کردن بهترین تعداد خوشها از ۱ تا ۹.

ایجاد یک مدل KMeans با تعداد

خوشاهای k و استفاده از

random_state=42 برای ایجاد

نتیجه تکرارپذیر.

خط کد	توضیحات
<pre>kmeans.fit(product_scaled)</pre>	آموزش مدل KMeans بر روی داده‌های استاندارد شده product_scaled پیرایی پیدا کردن خوشه‌ها.
<pre>distortions.append(sum(np.min(cdist(product_scaled, kmeans.cluster_centers_, 'euclidean'), axis=1)) / product_scaled.shape[0])</pre>	محاسبه و اضافه کردن "اختلال" به لیست:
	کد ۱۴:

الگوریتم RFM

```

: features = df.groupby("user_id").agg(
    total_spent=("total_amount", "sum"),
    transaction_count=("transaction_id", "count"),
    product_variety=("product_name", "nunique")
).reset_index()

scaler = StandardScaler()
features_scaled = scaler.fit_transform(features[["total_spent", "transaction_count", "product_variety"]])

kmeans = KMeans(n_clusters=4, random_state=42)
features[["Cluster"]] = kmeans.fit_predict(features_scaled)

features

```

شماره خط	کد	تحلیل
1	features = df.groupby("user_id").agg(داده‌ها بر اساس شناسه‌ی کاربر گروه‌بندی می‌شوند تا اطلاعات خرید هر کاربر به طور جداگانه محاسبه شود.
2	total_spent=("total_amount", "sum"),	مجموع کل مبلغ خرچ شده توسط هر کاربر محاسبه می‌شود.
3	transaction_count=("transaction_id", "count"),	تعداد کل تراکنش‌های انجام شده توسط هر کاربر محاسبه می‌شود.
4	product_variety=("product_name", "nunique")	تعداد محصولات منحصر به فردی که هر کاربر خریداری کرده، محاسبه می‌شود.
5).reset_index()	داده‌های گروه‌بندی شده به یک DataFrame جدید تبدیل و شاخص‌ها مجدداً تنظیم می‌شوند.

شماره خط

کد

تحلیل

6 scaler = StandardScaler()

شیء استانداردسازی داده‌ها از کتابخانه‌ی `sklearn.preprocessing` ایجاد می‌شود تا مقیاس ویژگی‌ها یکسان شود.

7 features_scaled =
 scaler.fit_transform(features[["total_spent",
 "transaction_count", "product_variety"]])

مقدارهای عددی ستون‌های کل مبلغ، تعداد تراکنش و تنوع محصولات نرمال‌سازی می‌شوند تا تأثیر مقادیر بزرگ‌تر در خوشبندی کاهش یابد. با ۴ خوشه و K-Means الگوریتم مقدار اولیه‌ی تصادفی ثابت تعریف می‌شود.

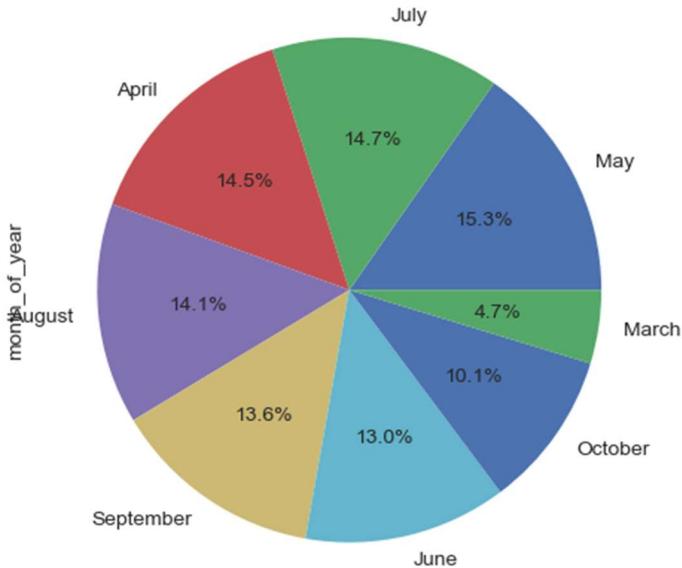
8 kmeans = KMeans(n_clusters=4, random_state=42)

روی داده‌های K-Means مدل نرمال‌شده اعمال شده و هر کاربر به یکی از ۴ خوشه تخصیص داده می‌شود. نمایش داده‌های پردازش شده شامل کل مبلغ خرج شده، تعداد تراکنش‌ها، تنوع محصولات و خوشبندی کاربران.

9 features["cluster"] =
 kmeans.fit_predict(features_scaled)

10 features

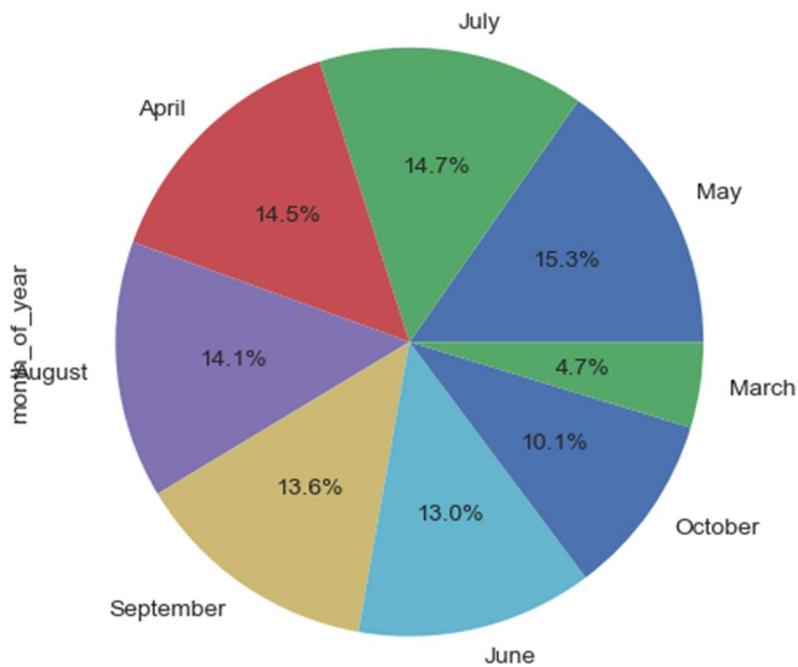
نتایج :



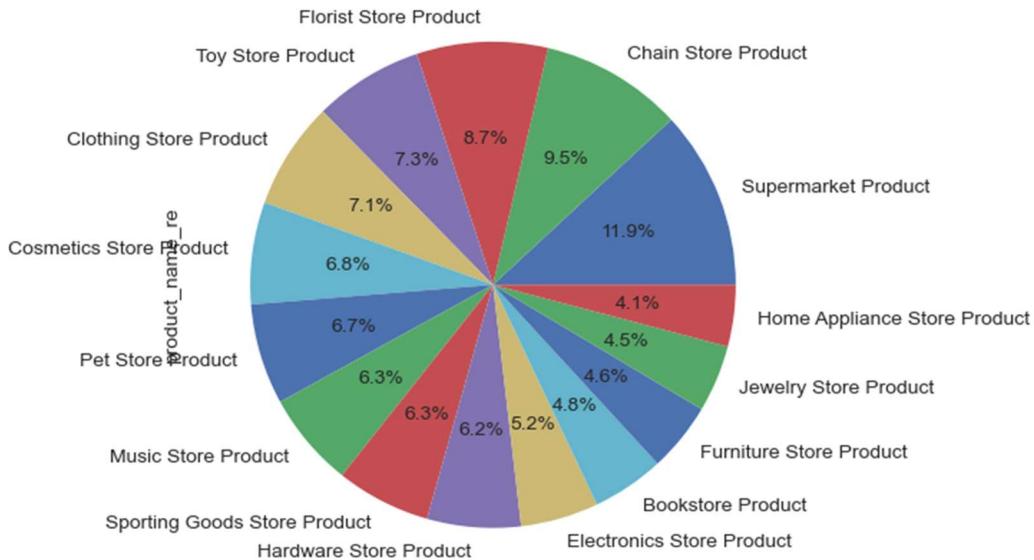
ماه ها	تحلیل	نتیجه گیری
May بیشترین خرید در (اردیبهشت)	این ماه بیشترین تعداد خرید را دارد، احتمالاً به دلیل مناسبتهای خاص، تخفیفها یا افزایش تقاضای فصلی برخلاف انتظار، خریدها در اسفند کم بوده‌اند. ممکن است به دلیل تعطیلات عید نوروز کم شده است در این ماه و این مورد باید در نظر گرفته شود که فروش کاهش موجودی، یا افزایش قیمت‌ها باشد	برنامه‌ریزی برای افزایش تبلیغات و تخفیف‌های ویژه در این ماه به دلیل ماه رمضان و خرید‌های خانواده‌ها برای سفره‌های افطار و سحری باید مورد توجه باشد
March کمترین خرید در (اسفند)	این ماهها پس از May بالاترین میزان خرید را دارند و نشان‌دهنده افزایش تقاضا در این دوره است	به دلیل تعطیلات عید نوروز کم شده است در این ماه و این مورد باید در نظر گرفته شود که فروش برای این ماه کم هست
July, April و August	این ماه‌ها نسبت به بقیه کاهش خرید را نشان می‌دهند، ممکن است تقاضا فصلی باشد یا تأثیر عوامل اقتصادی باشد	در ماه فروردین به دلیل عید نوروز و ماه رمضان و در ماه‌های تیر و مرداد به دلیل مسافرت خانواده‌ها و افزایش تقاضا برای کالاهای برای فصل گرما مثل کولر و بطری آب معدنی افزایش خرید داشته‌اند
October و June خریدهای کمتری دارند	در ماه خرداد احتمال دارد به خاطر شروع امتحانات دانش آموزان و دانشجویان و در ماه مهر به دلیل اینکه خانواده‌ها در ماه شهریور هزینه‌های سنجین برای لوازم التحریر کرده‌اند بودجه آن‌ها کاهش پیدا کرده	

ماه ها	تحلیل	نتیجه گیری
--------	-------	------------

است که البته فروشگاه ها تخفیف های ویژه برای مشتریان خودشون بزارند

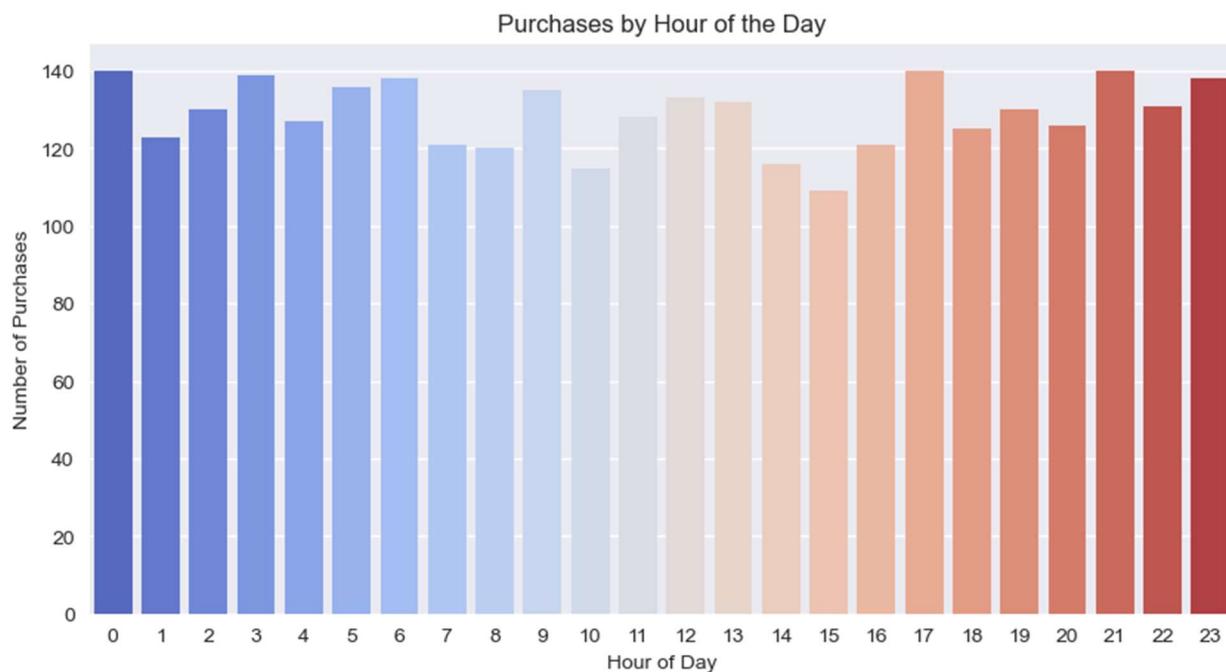


تحلیل و توضیحات	درصد تراکنشها	روز هفتہ
شروع هفته و اولین روز کاری؛ افراد پس از تعطیلات، خریدهای ضروری خود را انجام می‌دهند	16.1%	شنبه
نزدیک به آخر هفته، زمان مناسبی برای خریدهای هفتگی و آماده شدن برای تعطیلات	15.1%	چهارشنبه
روزی نسبتاً پر تراکنش، احتمالاً مردم خریدهای میان‌هفته را انجام می‌دهند	14.2%	سه‌شنبه
روز تعطیل، برخی مشاغل بسته‌اند ولی خانواده‌ها برای خریدهای تفریحی بیرون می‌روند	14.0%	جمعه
روزی با تراکنش‌های کمتر، معمولاً مردم در حال تطبیق با برنامه کاری خود هستند	13.7%	دوشنبه
کم تراکنش، مشابه دوشنبه، خریدهای ضروری انجام نمی‌شود	13.6%	یکشنبه
کمترین میزان تراکنش؛ برخی ادارات و شرکت‌ها تعطیل می‌شوند، مردم کمتر به خرید می‌روند نکته مهم: برای روزهای شنبه و چهارشنبه و سه‌شنبه حتماً باید طرح‌هایی در نظر گرفته بشه چون در این روزها خرید مردم بیشتر از روزهای دیگه هست	13.2%	پنج‌شنبه



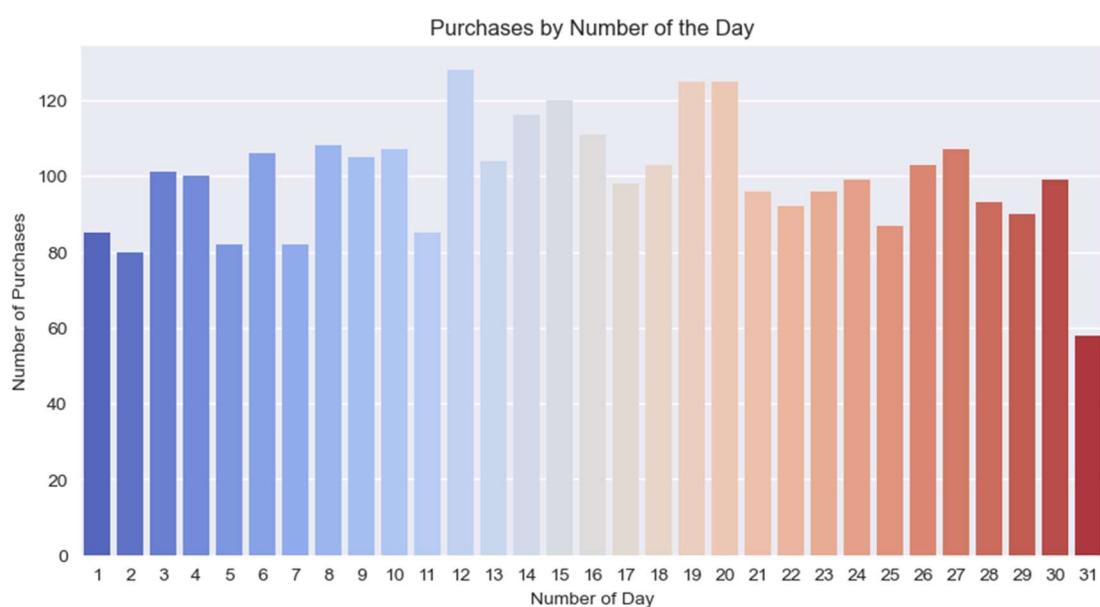
دسته‌بندی محصول	سهم از کل فروش (%)	تحلیل و پیشنهادها
محصولات الکترونیکی Product (Electronics Store Product)	4.8%	قیمت بالا باعث کاهش خریدهای مکرر شده؛ ارائه فروش اقساطی و گارانتی پیشنهاد می‌شود.
کتاب‌فروشی (Bookstore Product)	5.2%	فروش متوسط؛ افزایش تخفیف‌های نمایشگاهی و ارسال رایگان می‌تواند مؤثر باشد.
لوازم منزلي (Furniture Store Product)	4.6%	بازار محدود؛ امکان ارائه تخفیف در خریدهای عمده یا طرح‌های اقساطی مفید است.
جواهرات (Jewelry Store Product)	4.5%	خرید کمتر به دلیل قیمت بالا؛ پیشنهاد می‌شود کمپین‌های تبلیغاتی خاص و اقساطی ارائه شود.
لوازم خانگي (Home Appliance Store Product)	4.1%	خرید گران‌قیمت و کمتر تکرارشونده؛ ارائه طرح‌های اقساطی توصیه می‌شود.

تحلیل خرید بر اساس ساعت شبانه‌روز



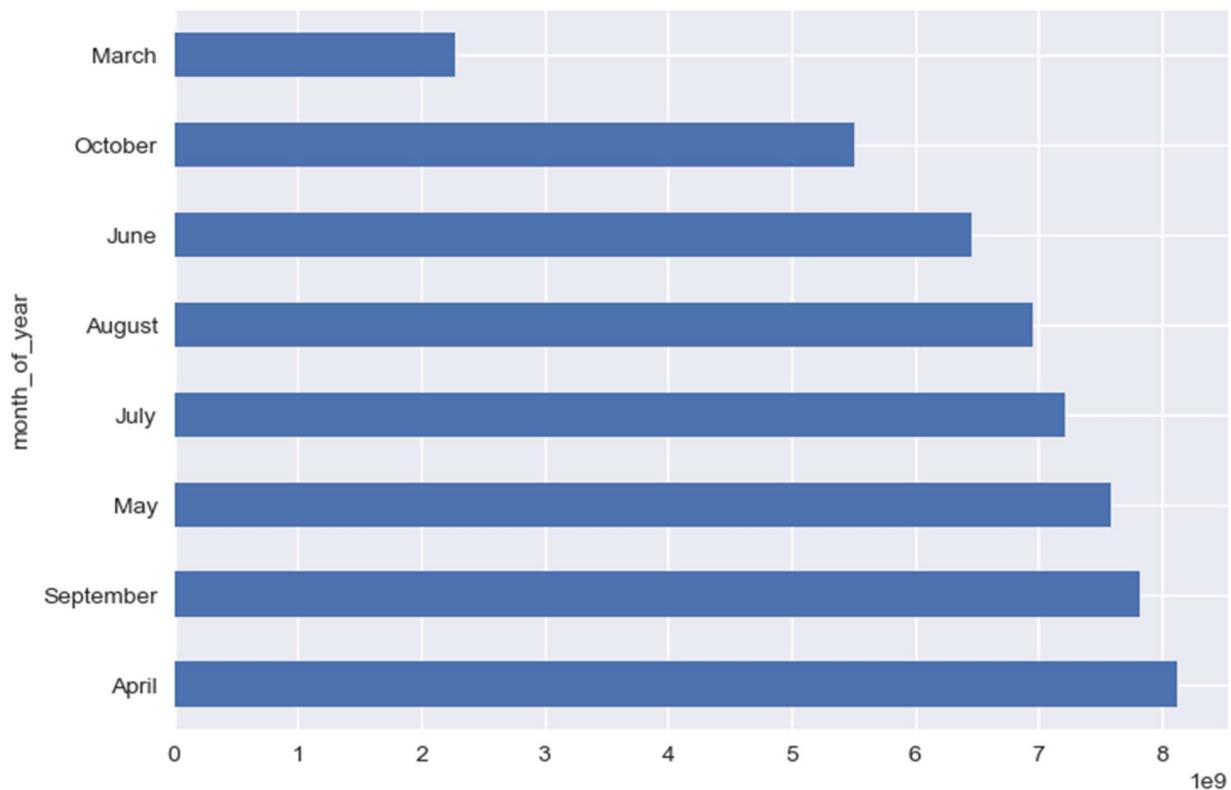
ساعت	میزان خرید	تحلیل بر اساس رفتار خرید ایرانی‌ها
۳ - ۰	بالا	خریدهای شبانه به دلیل تخفیف‌های آنلاین و فعالیت کاربران شب‌بیدار
۶ - ۴	متوسط	کاهش خرید به دلیل ساعات خواب بیشتر مردم
۹ - ۷	متوسط	خریدهای صبحگاهی افراد شاغل و خانه‌دارها
۱۲ - ۱۰	بالا	افزایش خرید به دلیل فعالیت روزانه و دسترسی بیشتر به اینترنت
۱۵ - ۱۳	متوسط رو به کم	کاهش خرید به دلیل استراحت بعد از ناهار
۱۸ - ۱۶	اوج خرید	ساعات پیک خرید به دلیل پایان کار ادارات و فراغت کاربران
۲۱ - ۱۹	بالا	خریدهای خانوادگی و برنامه‌ریزی برای روز بعد
۲۳ - ۲۲	بالا	افزایش خریدهای لحظه‌آخري و استفاده از پیشنهادهای ویژه پایان روز

تحلیل خرید بر اساس روزهای ماه



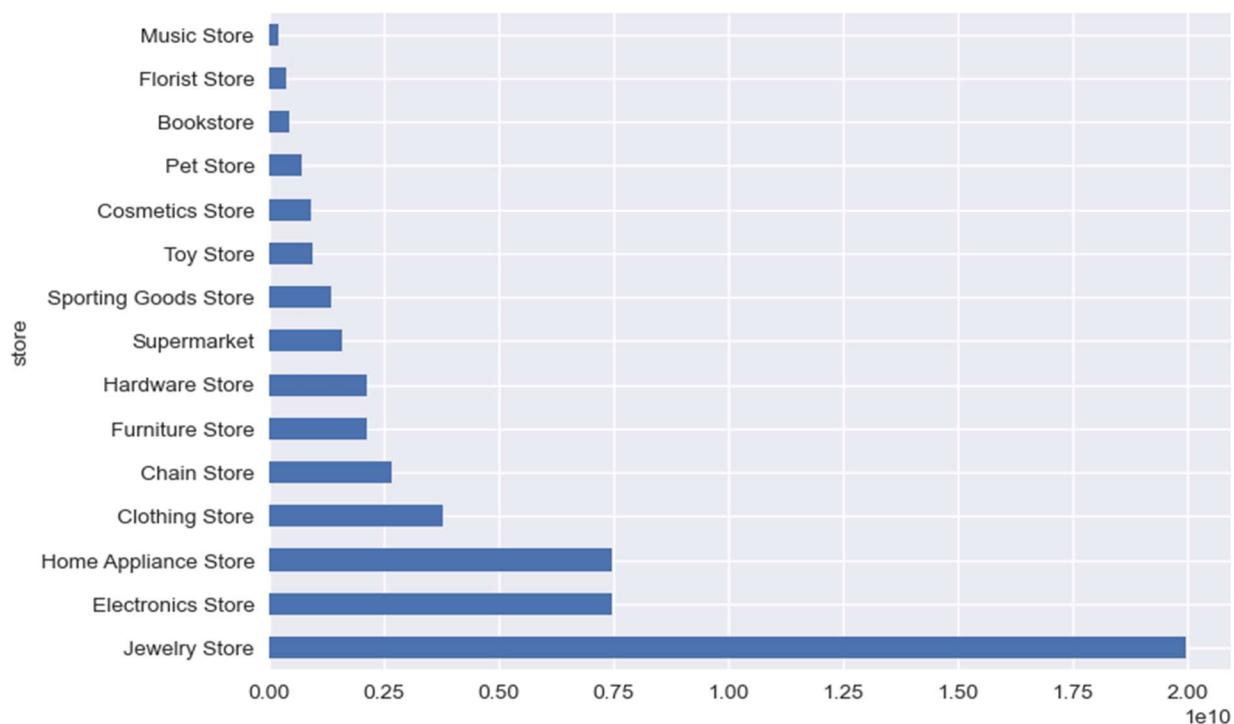
تحلیل بر اساس رفتار خرید ایرانی‌ها	روز ماه	میزان خرید
خریدهای محدود به دلیل شروع ماه و پرداخت هزینه‌های ضروری	۱ - ۵	متوسط رو به پایین
افزایش خرید با گذشت چند روز از دریافت حقوق و تسویه بدهی‌ها	۶ - ۱۰	متوسط رو به بالا
بیشترین میزان خرید، معمولاً پس از دریافت حقوق کارمندان و کارگران	۱۱ - ۱۳	اوج خرید
کاهش خرید به دلیل انجام خریدهای اصلی در روزهای قبل	۱۴ - ۱۷	متوسط
افزایش دوباره خریدها، احتمالاً به دلیل تخفیف‌های میانی ماه	۱۸ - ۲۱	اوج خرید
خریدهای برنامه‌ریزی شده برای پایان ماه	۲۲ - ۲۶	متوسط رو به بالا
خریدهای ضروری قبل از اتمام بودجه خانوار	۲۷ - ۳۰	متوسط
کمترین خرید به دلیل پایان بودجه ماهانه و انتظار برای ماه بعد	۳۱	کاهش شدید

تحلیل فروش ماهانه



همانطور که قبلن هم گفته در ماه فرودین و اردیبهشت به دلیل عید نوروز و ماه رمضان خرید های مشتریان خیلی زیاد هست و در ماه شهریور به دلیل بازگشایی مدارس در مهر ماه خرید های خانواده ها زیاد شده است و در ماه هایی مثل تیر و مرداد در پله های بعدی توجه باید باشند به دلیل شروع مسافرت خانواده ها و لوازم مورد نیاز در فصل تابستان مثل خرید آب معدنی و کولر و لباس هایی مثل تی شرت خرید های خانواده های ایرانی افزایش میابد

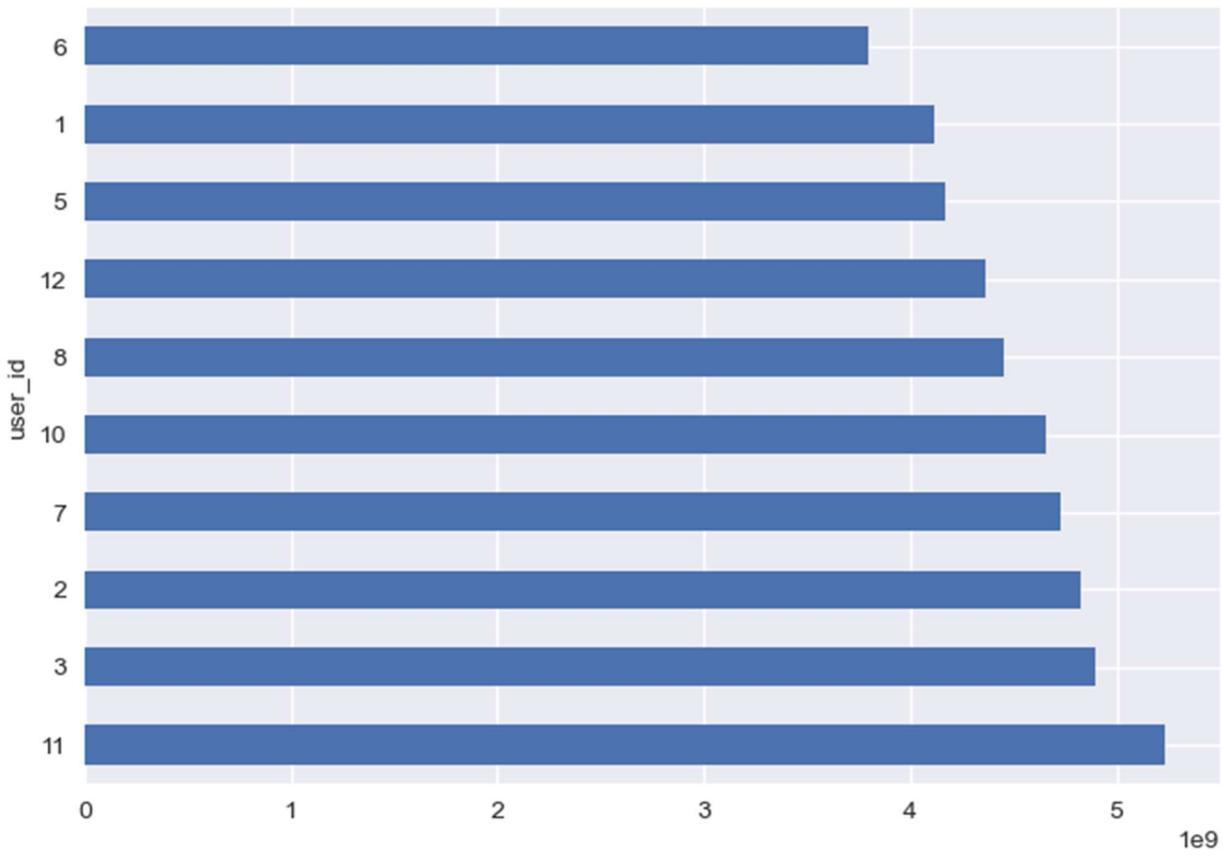
تحلیل فروش بر اساس دسته‌بندی محصولات



دسته‌بندی محصول	میزان فروش	تحلیل بر اساس رفتار خرید ایرانی‌ها
جواهرات	بسیار بالا	خریدهای لوکس، تقاضای بالا برای سرمایه‌گذاری و هدايا
لوازم الکترونیکی	بالا	پرطرفدار به دلیل نیازهای روزمره و تکنولوژی
لوازم خانگی	بالا	خرید ضروری برای خانه‌ها، مخصوصاً در مناسبت‌ها
پوشاک	متوسط رو به بالا	خرید دوره‌ای، وابسته به فصل و تخفیف‌ها
فروشگاه‌های زنجیره‌ای	متوسط	خریدهای ترکیبی با تخفیف‌های دوره‌ای
مبلمان و دکوراسیون	متوسط	وابسته به تغییرات فصلی و نیازهای خانوار
فروشگاه‌های سخت‌افزار	متوسط	تقاضای خاص برای تعمیرات و نوسازی
سوپرمارکت	متوسط	خریدهای ضروری روزمره، با رقابت بالا

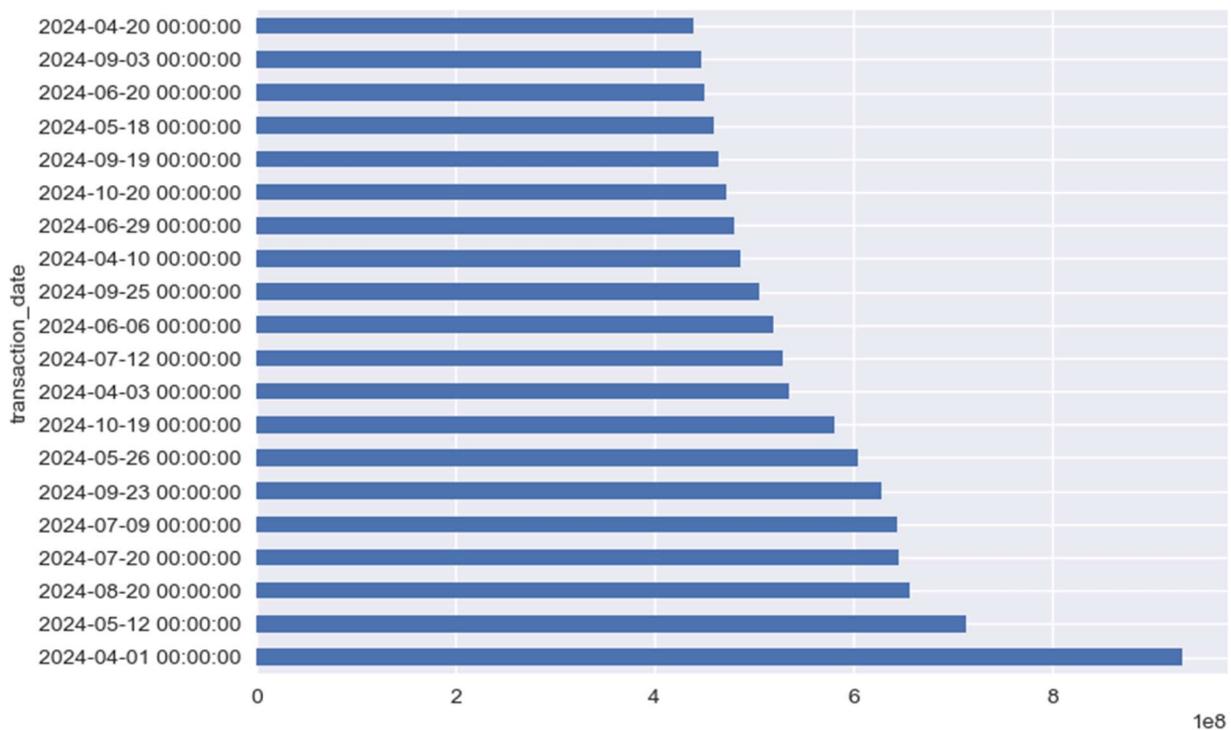
لوازم ورزشی	کم	بازار محدود، تقاضای خاص در زمان تخفیفها
اسباب‌بازی	کم	وابسته به فصل‌های خاص مثل عید و بازگشایی مدارس
لوازم آرایشی	کم	رقابت بالا، نیاز به تبلیغات قوی
فروشگاه‌های حیوانات خانگی	بسیار کم	بازار خاص و محدود به افراد علاقه‌مند
کتاب‌فروشی	بسیار کم	نیاز به تبلیغات گسترده و تخفیف‌های جذاب
گل‌فروشی	بسیار کم	خرید مناسبتی، نیاز به کمپین‌های خاص
فروشگاه موسیقی	بسیار کم	کاهش تقاضا به دلیل دیجیتالی شدن موسیقی

تحلیل رفتار کاربران ایرانی بر اساس میزان تراکنش



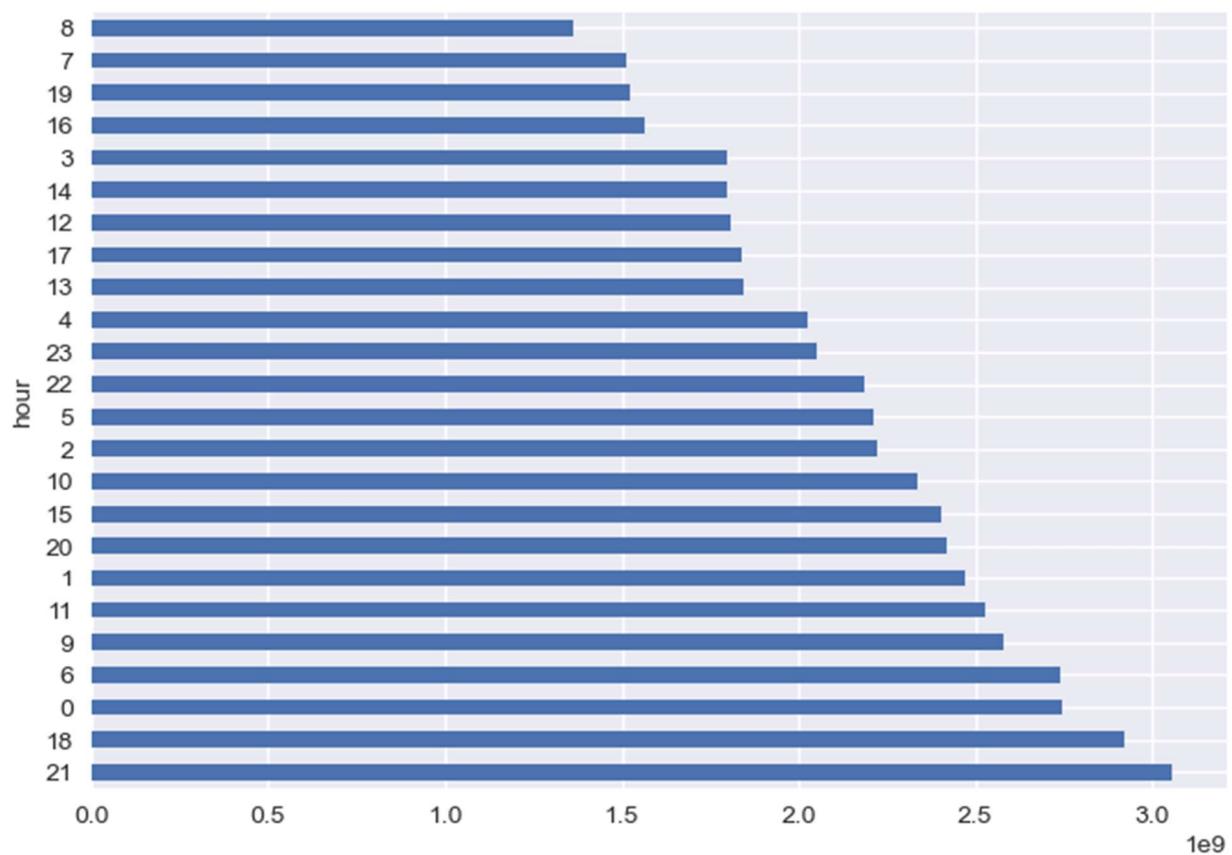
کاربران شماره ۱۱ و ۳ و ۲ با تراکنش بالا و کاربران ۷ و ۱۰ و ۸ و ۱۲ با تراکنش متوسط و کاربرانی کمتر خرچ داشتن ۵ و ۱ و ۶ هستند که باید برای هر کدام از این گروه‌ها برنامه‌ای داشت مثل ارائه خدمات پس از فروش قوی برای ایجاد حس اعتماد بیشتر، خصوصاً در خریدهای گران قیمت مانند لوازم خانگی و الکترونیکی

تحلیل تاریخ‌های پر تراکنش بر اساس تقویم شمسی و رویدادهای مهم ایران



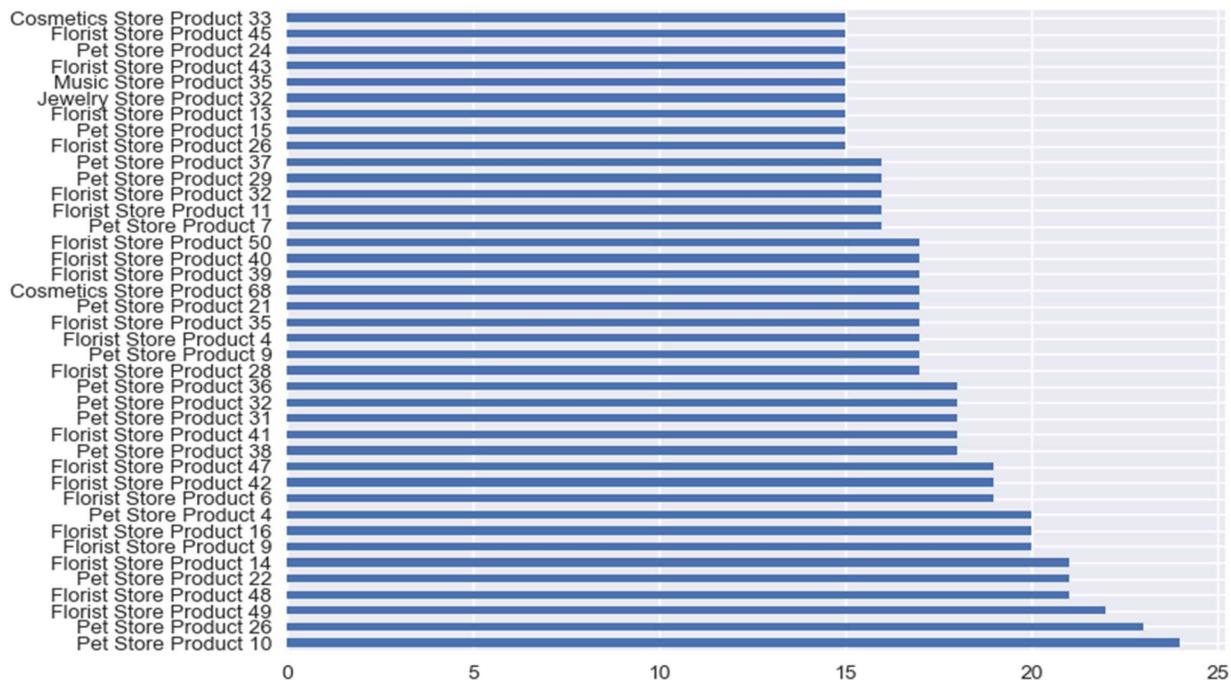
با توجه به تحلیل داده‌ها، سه دوره‌ی زمانی اعیاد ملی و مذهبی، فصل تابستان و بازگشایی مدارس بیشترین حجم خرید را در ایران دارند. اعیاد مذهبی مانند نوروز، عید قربان و غدیر باعث افزایش خرید پوشاک، هدایا و مواد غذایی خاص می‌شوند، در حالی که تابستان به دلیل سفرها و اوقات فراغت، تقاضای بالایی برای لوازم تفریحی، دیجیتال و پوشاک دارد. همچنین، شهریور و مهر با شروع مدارس و دانشگاه‌ها، اوج خرید لوازم تحریر، کیف، کفش و وسایل الکترونیکی را به همراه دارد. بنابراین، تمرکز بر تخفیف‌های مناسبتی، تبلیغات هدفمند و افزایش موجودی کالاهای پر تقاضا در این بازه‌ها، می‌تواند منجر به رشد فروش و جذب مشتریان بیشتر شود.

تحلیل ساعت‌های پر تراکنش در ایران



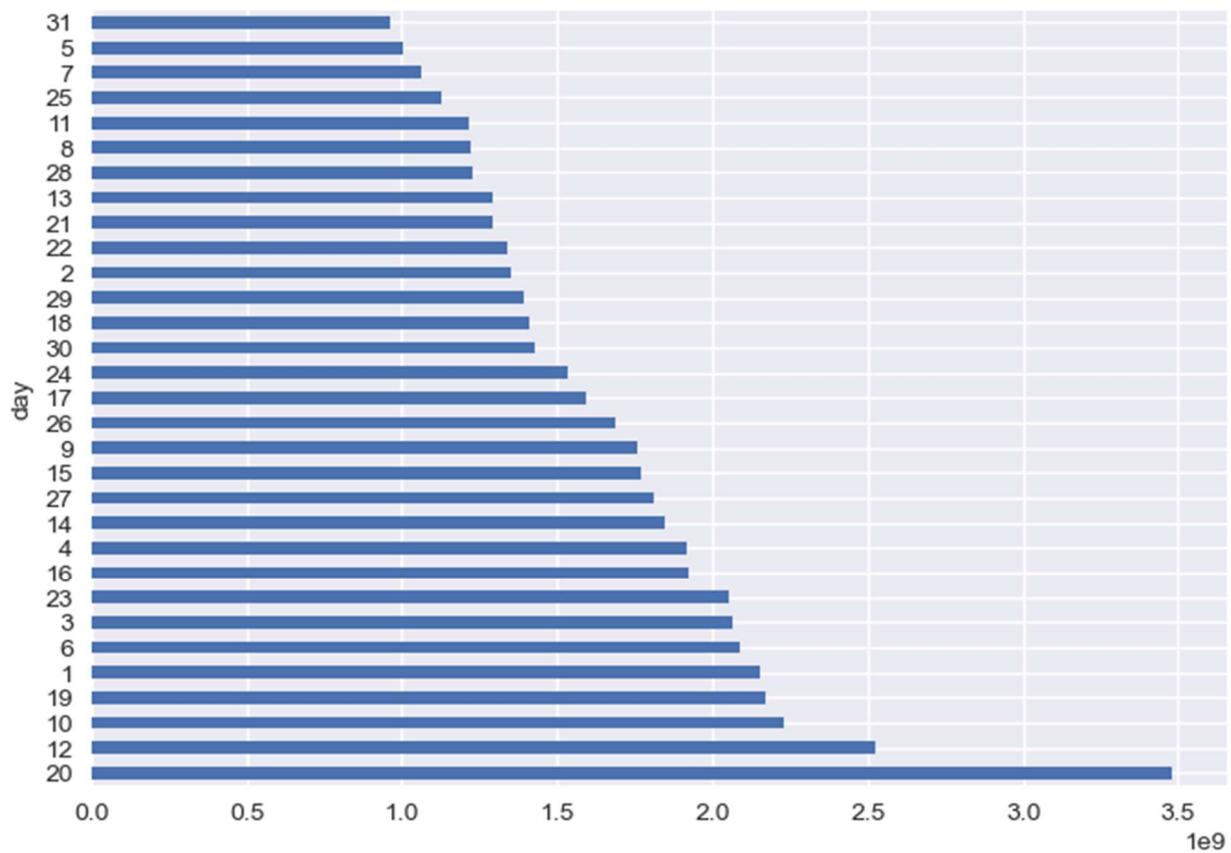
بر اساس داده‌های ارائه شده، بیشترین میزان تراکنش در ساعت ۲۱، ۰۰ (نیمه شب) انجام شده است. این نشان می‌دهد که کاربران ایرانی بیشتر در ساعت پایانی شب و اوایل صبح خریدهای خود را انجام می‌دهند. ساعت ۶ و ۹ صبح نیز جزو بازه‌های پُر تراکنش هستند که می‌تواند نشان‌دهنده خریدهای روزانه و کسب وکارهای زودهنگام باشد. ساعت ۱۴ تا ۱۹ کمترین میزان تراکنش را دارند که احتمالاً به دلیل ساعت کاری و استراحت بعد از ظهر است.

تحلیل بیشترین محصولات خریداری شده



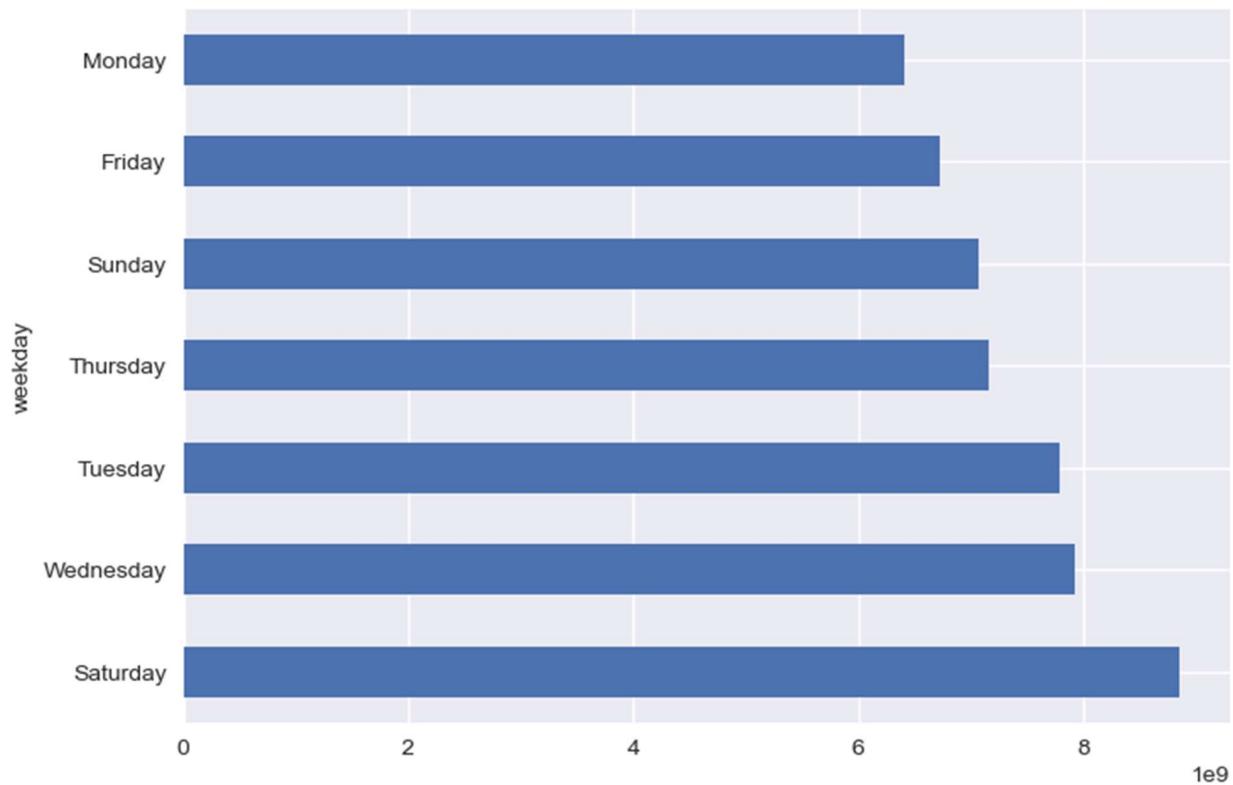
بر اساس داده‌های ارائه شده، محصولات فروشگاه‌های حیوانات خانگی (Pet Store)، گل فروشی (Florist Store) و لوازم آرایشی (Cosmetics Store) بیشترین میزان خرید را داشته‌اند. در این میان، محصولات مرتبط با حیوانات خانگی مانند محصول شماره ۱۰ و ۲۶ و محصولات گل فروشی مانند محصولات شماره ۴۸، ۴۹ و ۱۴ در صدر لیست هستند و الگوی رفتاری ایرانی‌ها را باید بهش توجه کرد با توجه به جدول باید به رشد بازار محصولات حیوانات خانگی که نشان از افزایش تمایل ایرانیان به نگهداری از حیوانات خانگی دارد و خرید بالا از گل فروشی‌ها می‌تواند به دلیل مناسبت‌های خاص مانند اعیاد، تولدات‌ها و رویدادهای عاشقانه باشد و محصولات آرایشی نیز با تقاضای بالا مواجه‌اند که نشان‌دهنده‌ی اهمیت زیبایی و مراقبت شخصی در جامعه است.

تحلیل روزهای پر معامله بر اساس الگوی خرید ایرانیان



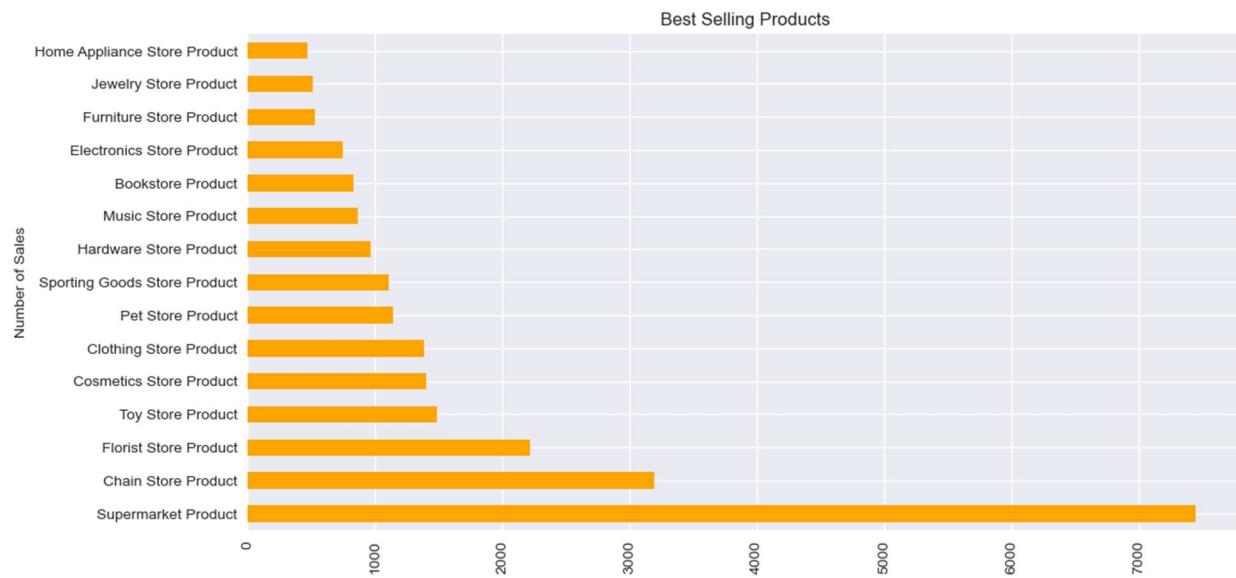
بررسی الگوی خرید ایرانیان نشان می‌دهد که روزهای ۱۰ تا ۲۰ ماه به دلیل دریافت حقوق و افزایش قدرت خرید، بیشترین میزان تراکنش را دارند، در حالی که روزهای پایانی ماه (۲۰ تا ۳۰) نیز به دلیل خریدهای برنامه‌بیزی شده و تکمیل نیازهای ضروری، سهم قابل توجهی از معاملات را به خود اختصاص داده‌اند. همچنین، اوایل ماه (۱ تا ۶) به دلیل خریدهای اساسی، میزان خرید نسبی بالایی دارد. بر این اساس، کسب‌وکارها می‌توانند با ارائه تخفیف‌های ویژه در روزهای اوج خرید، افزایش تبلیغات در بازه‌های پر تقاضا و بهینه‌سازی موجودی کالاهای فروش خود را به حداقل برسانند و سهم بیشتری از بازار را تصاحب کنند.

تحلیل روزهای پر معامله بر اساس الگوی خرید ایرانیان



بررسی میزان تراکنش‌ها در روزهای هفته نشان می‌دهد که شنبه‌ها بیشترین حجم خرید را دارند که می‌تواند ناشی از آغاز هفته کاری و تأمین نیازهای جدید باشد. سه‌شنبه و چهارشنبه نیز به دلیل قرار گرفتن در میانه هفته، حجم بالایی از تراکنش‌ها را به خود اختصاص داده‌اند. در مقابل، دوشنبه‌ها کمترین میزان خرید را دارند که احتمالاً به دلیل کاهش نقدینگی پس از خریدهای ابتدایی هفته است. همچنین، پنجشنبه و جمعه با وجود تعطیلی، همچنان میزان تراکنش بالایی دارند که نشان‌دهنده افزایش خریدهای خانوادگی و تفریحی در پایان هفته است.

تحلیل روند خرید مشتریان و فرصت‌های بهینه‌سازی فروش

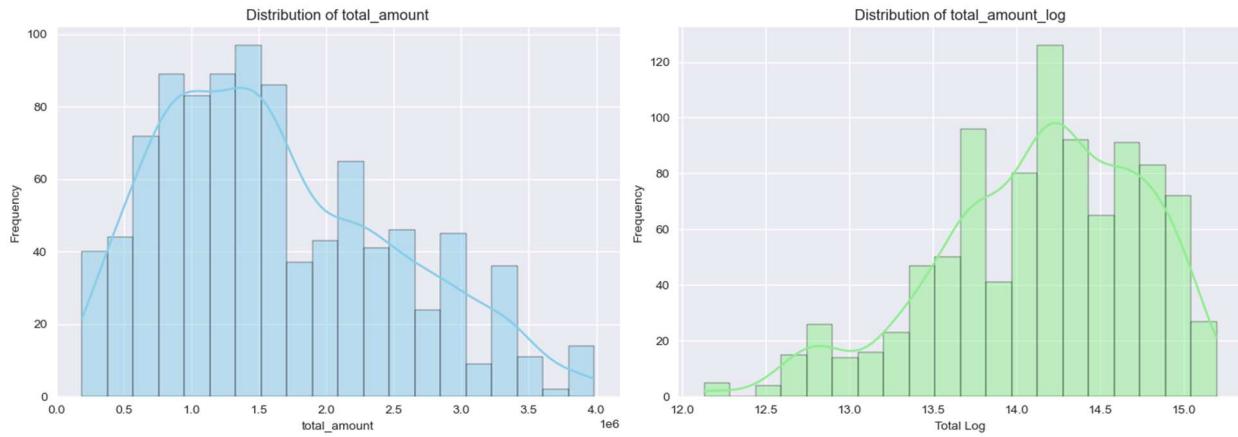


داده‌های مربوط به میزان خرید در دسته‌بندی‌های مختلف نشان می‌دهد که محصولات سوپرمارکتی بیشترین میزان خرید را داشته‌اند که به دلیل نیاز روزمره و دسترسی آسان به این محصولات قابل توجیه است. پس از آن، محصولات فروشگاهی زنجیره‌ای در رتبه دوم قرار دارند که نشان‌دهنده استقبال مشتریان از خریدهای متنوع در یک مکان واحد است. محصولات گل‌فروشی و اسباب‌بازی نیز در جایگاه‌های بعدی قرار دارند که می‌تواند به دلیل هدایا و مناسبت‌های خاص باشد. محصولات الکترونیکی، لوازم منزل و جواهرات با کمترین میزان خرید مواجه بوده‌اند که احتمالاً به دلیل قیمت بالاتر و خریدهای دوره‌ای آن‌ها است. بر این اساس، فروشگاه‌های زنجیره‌ای و سوپرمارکتها می‌توانند با پیشنهادات ویژه و طرح‌های تخفیفی، میزان فروش خود را بیشتر کنند و فروشگاه‌های با میزان خرید کمتر، می‌توانند بر روی ارائه شرایط اقساطی و تبلیغات هدفمند تمرکز کنند.

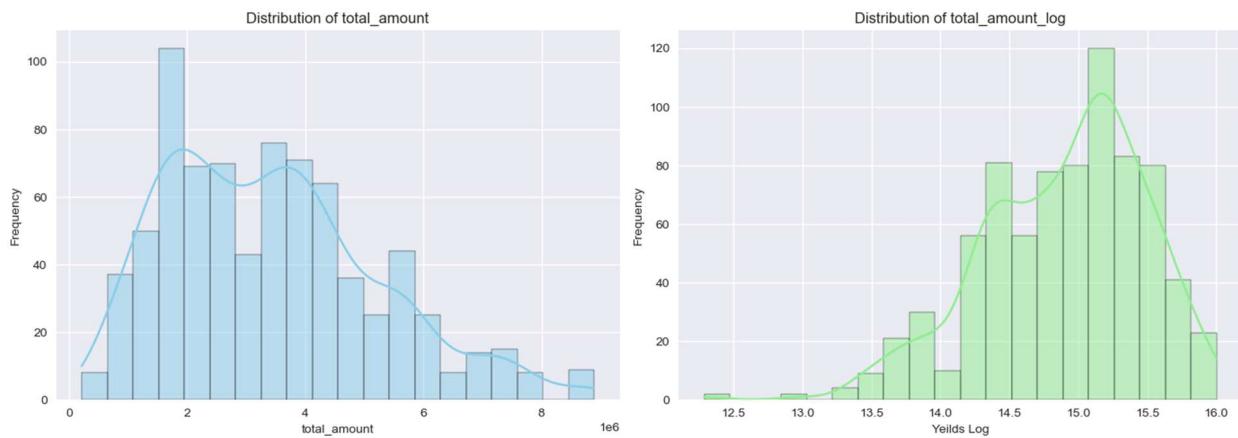
پیشرفت داده‌ها برای سود بیشتر :

فعلاً من به دلیل تقاضاً زیاد مردم ایران برای خرید از سوپر مارکت‌ها و فروشگاه‌های زنجیره‌ای این دو را پیشرفت خواهم داد

سوپرمارکت‌ها :



فروشگاه زنجیره ای :



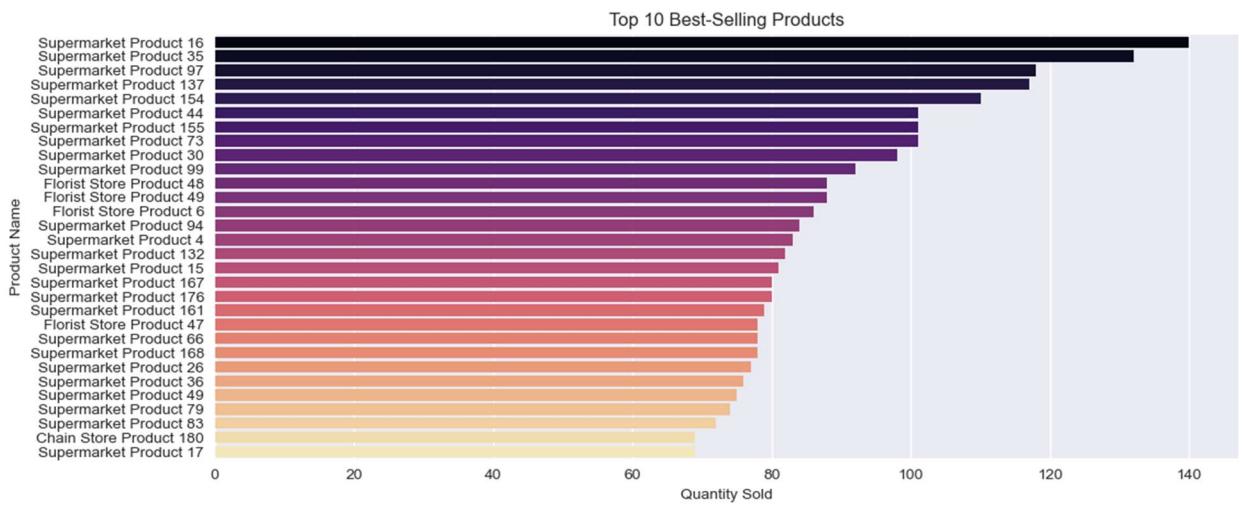
با پیشرفت دادن داده ها میتوانیم متوجه بشیم اگر جشنواره و یا تخفیفی بزاریم ممکنه چقد سود کنیم در پلات های بالا نشانم داده شده است و به طور مثال اگر از یک سوپر مارکت ۱۲۰ خرید داشته باشیم میتوانیم حدود ۱,۵۴ تومان سود میکند

بررسی قیمت واحد هر محصول :



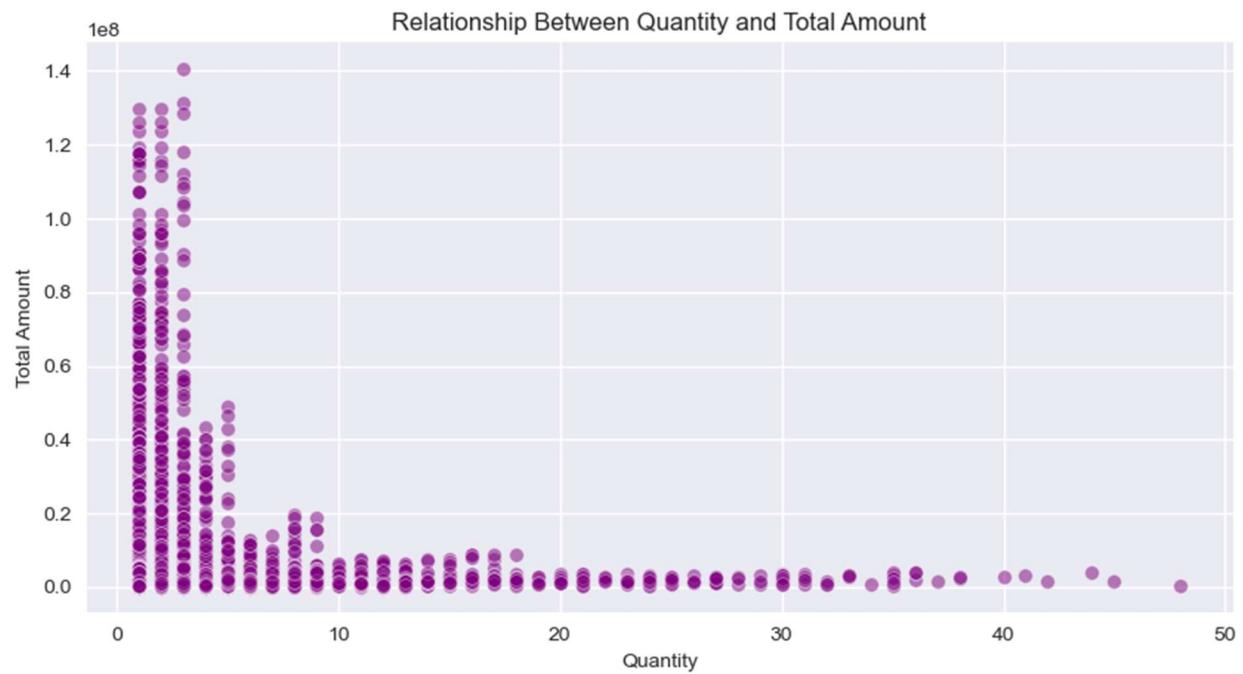
محصولاتی مثل جواهرات و لوازم الکترونیکی و الپلیکشن های خانگی و گل فروشی ها قیمت بالایی دارند

بررسی محصولات خریداری شده با ستون تعداد خرید ها :



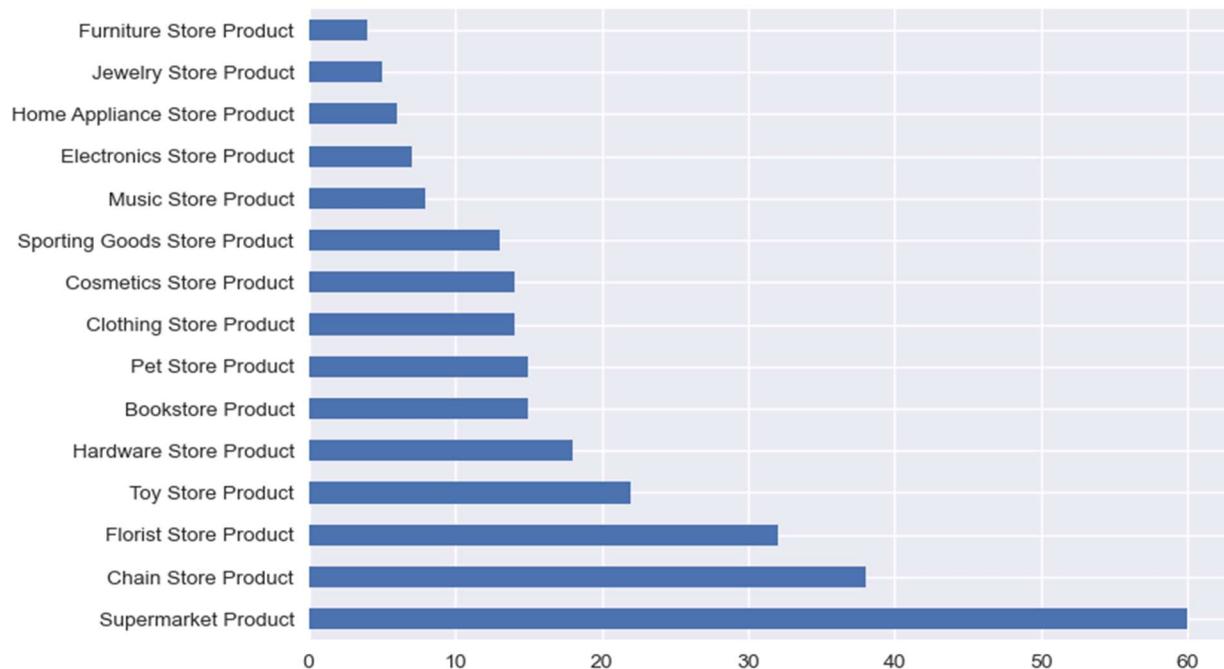
سوپر مارکت ها و گل فروشی ها و فروشگاه های زنجیره ای بیشتر سهم از تراکنش های ایرانی ها را دارند ولی بیشتر تراکنش ها با تعداد بالا برای سوپر مارکت ها هست

بررسی تعداد تراکنش برای هر خرید :



بیشتر تراکنش ها بین ۱ تا ۵ عدد بوده هست و تراکنش های بالا تر از ۵ خیلی کم هست

بررسی محصولات تخفیف دار :

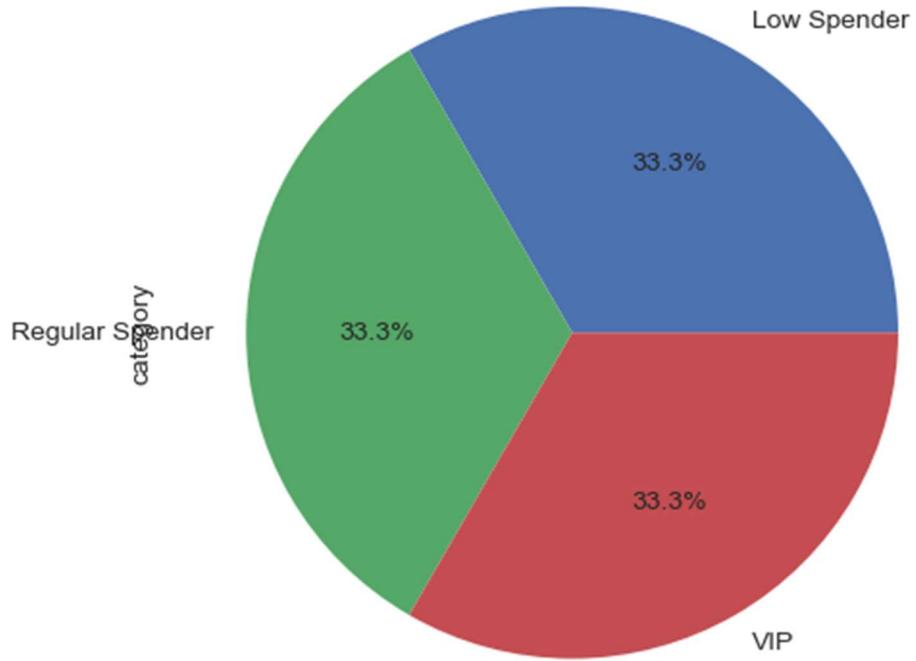


سوپر مارکت ها و فروشگاه های زنجیره ای و گل فروشی ها بیشترین تعداد محصولات دارای تخفیف را دارا هستند

گروه های مشتریان :

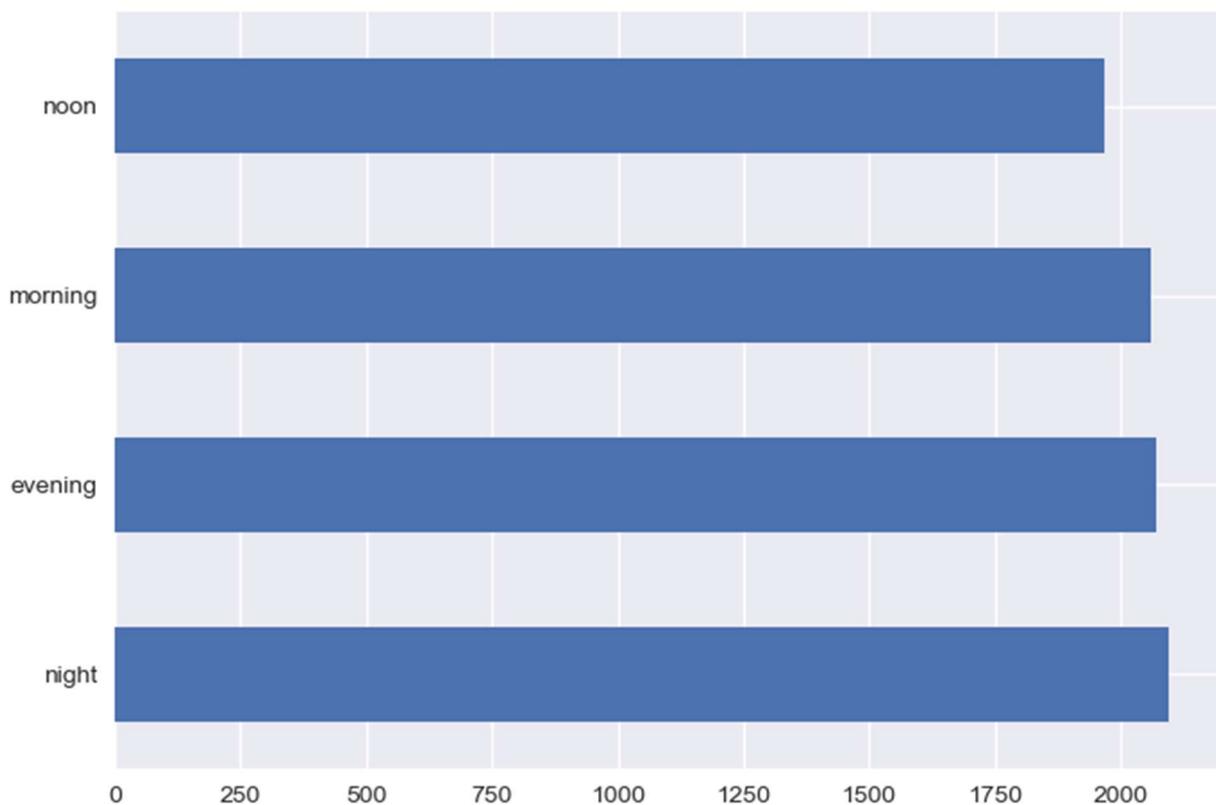
(۱) گروه مشتریان براساس یوزر ایدی کاربران و مبلغ تراکنش ها :

	user_id	total_amount	log_total_amount	category
0	1	4.114019e+09	22.137666	Low Spender
1	2	4.824619e+09	22.296998	VIP
2	3	4.897524e+09	22.311996	VIP
3	4	3.333319e+09	21.927234	Low Spender
4	5	4.167616e+09	22.150610	Regular Spender
5	6	3.797353e+09	22.057570	Low Spender
6	7	4.731971e+09	22.277608	VIP
7	8	4.452082e+09	22.216638	Regular Spender
8	9	3.356738e+09	21.934235	Low Spender
9	10	4.654176e+09	22.261031	Regular Spender
10	11	5.236716e+09	22.378960	VIP
11	12	4.368383e+09	22.197659	Regular Spender



به خاطر چپ چولگی در این ستون میلغ تراکنش های کاربران بقا استفاده از لگاریتم و صدک ها گروه بندی که انجام شد با عدالت بود و طبق پلات بالا تعداد افراد پر خرج و کم خرج و متوسط برابر است

۲) گروه بندی براساس خرید در زمان مختلف روز:



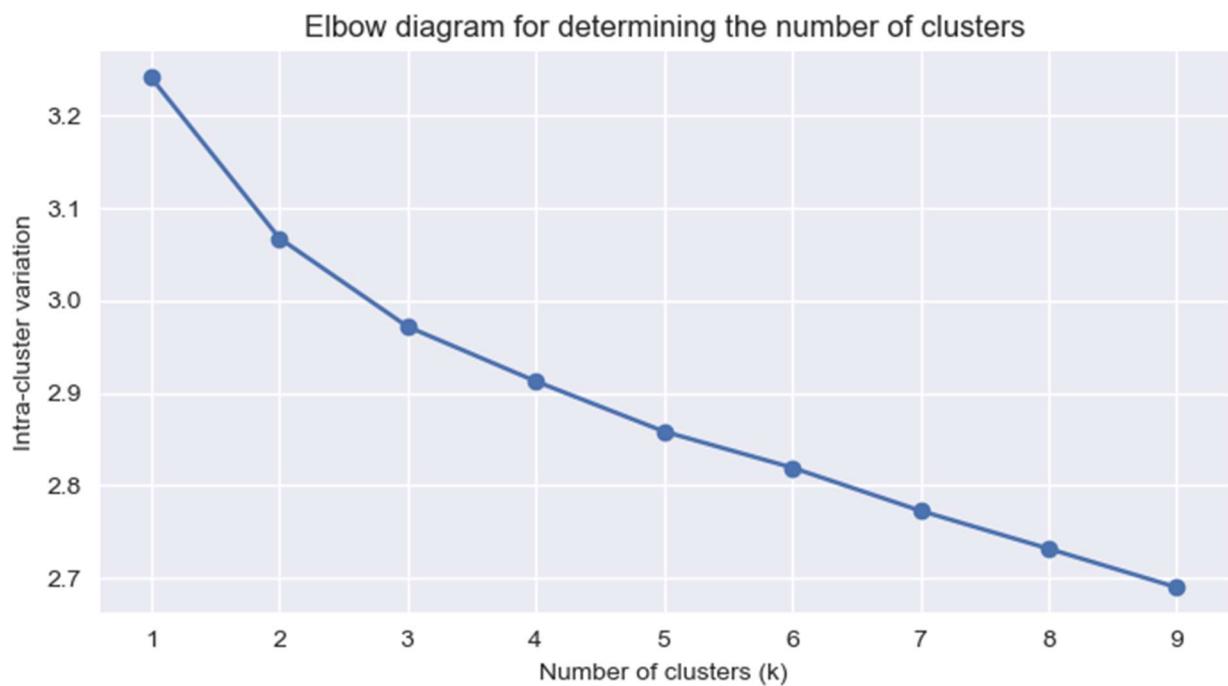
طبق گروه بندی بالا بیشتر ایرانی ها در شب و عصر و صبح خرید های خودشان را انجام میدهند:

۳) گروه بندی کاربرانی که محصولاتی به صورت جفتی خریداری شدند:

```
[('Florist Store Product 40', 'Florist Store Product 41'), 2),
 ('Florist Store Product 16', 'Florist Store Product 47'), 2),
 ('Jewelry Store Product 2', 'Jewelry Store Product 48'), 2),
 ('Supermarket Product 140', 'Supermarket Product 55'), 2),
 ('Pet Store Product 22', 'Pet Store Product 36'), 2),
 ('Sporting Goods Store Product 100', 'Sporting Goods Store Product 106'), 2),
 ('Sporting Goods Store Product 100', 'Sporting Goods Store Product 74'), 2),
 ('Pet Store Product 20', 'Pet Store Product 7'), 2),
 ('Sporting Goods Store Product 44', 'Sporting Goods Store Product 76'), 2),
 ('Pet Store Product 21', 'Pet Store Product 25'), 2)]
```

براساس اطلاعات بالا میشود. اگر مشتری از یک محصول خریداری کرده دفعه بعد همین محصول را بهش پیشنهاد بدیم اطلاعات بالا که برای کاربر شماره یک داده بوده هست

۴) گروه بندی با استفاده از الگوریتم KMeans با استفاده از نمودار Elbow



نمودار Elbow نشان می‌دهد که با افزایش تعداد خوشه‌ها، میزان تغییرات درون‌خوشه‌ای کاهش می‌یابد، اما از یک نقطه به بعد این کاهش کمتر شده و نمودار تقریباً روند ثابتی پیدا می‌کند. در اینجا، به نظر می‌رسد که بهترین نقطه برای انتخاب تعداد

خوشها حدود ۳ یا ۴ است، زیرا پس از آن، کاهش تغییرات چندان چشمگیر نیست. انتخاب این مقدار باعث می‌شود که خوشبندی بهینه انجام شده و داده‌ها بدون افزایش بیش از حد پیچیدگی، به درستی گروه‌بندی شوند.

۵) گروه بندی با استفاده از کاربران و تعداد محصولات خریداری شده :

	user_id	product_name	category
0	1	506	low-diversity
1	2	528	medium-diversity
2	3	556	high-diversity
3	4	550	high-diversity
4	5	477	low-diversity
5	6	507	medium-diversity
6	7	503	low-diversity
7	8	505	low-diversity
8	9	534	medium-diversity
9	10	537	high-diversity
10	11	584	high-diversity
11	12	509	medium-diversity

۶) گروه بندی براساس کاربران و تعداد تراکنش‌ها و روز‌های هفته :

	user_id	weekday	transaction_id	category
0	1	Wednesday	1	Midweek
1	1	Thursday	2	Weekend
2	1	Friday	3	Weekend
3	1	Friday	3	Weekend
4	1	Saturday	4	Start of Week
...
8187	12	Wednesday	306	Midweek
8188	12	Friday	307	Weekend
8189	12	Friday	307	Weekend
8190	12	Friday	308	Weekend
8191	12	Sunday	309	Midweek

8192 rows × 4 columns

۷) گروه بندی با استفاده از ساعت و تعداد تراکنش های کاربران :

	user_id	transaction_time	transaction_id	category
0	1	23	1	Off-Peak Hours
1	1	16	2	Off-Peak Hours
2	1	7	3	Off-Peak Hours
3	1	7	3	Off-Peak Hours
4	1	15	4	Off-Peak Hours
...
8187	12	13	306	Off-Peak Hours
8188	12	8	307	Off-Peak Hours
8189	12	8	307	Off-Peak Hours
8190	12	22	308	Off-Peak Hours
8191	12	6	309	Off-Peak Hours

۸) گروه بندی با استفاده الگوریتم RFM :

	user_id	total_spent	transaction_count	product_variety	Cluster
0	1	4.114019e+09	661	506	2
1	2	4.824619e+09	673	528	2
2	3	4.897524e+09	708	556	1
3	4	3.333319e+09	717	550	3
4	5	4.167616e+09	624	477	0
5	6	3.797353e+09	636	507	2
6	7	4.731971e+09	660	503	2
7	8	4.452082e+09	678	505	2
8	9	3.356738e+09	689	534	3
9	10	4.654176e+09	725	537	1
10	11	5.236716e+09	762	584	1
11	12	4.368383e+09	659	509	2

براساس خوش بندی توسط الگوریتم RFM :

- 50% مشتریان در خوشه ۲ هستند که نشان می دهد این دسته بخش بزرگی از مشتریان را تشکیل داده اند
- 25% در خوشه ۱ قرار دارند که مشتریان با ارزش بالاتر هستند و باید حفظ شوند
- تنها 8% از مشتریان در خوشه ۰ هستند، که نشان می دهد تعداد کمی از مشتریان کم خرید وجود دارد
- 16.67% مشتریان در خوشه ۳ هستند که دارای تنوع خرید بالا اما تراکنش های متوسط هستند

۳. نتیجه‌گیری

۴.

- (۱) مردم ایران بیشتر خرید های خود را از سوپر مارکت ها انجام میدهند و میشود گفت بیشتر مواد غذایی برای مردم ایران مهم است
- (۲) بیشتر در شب و عصر خرید میکنند
- (۳) باید به اعیاد و ایام ملی و مذهبی و فصل هایی مثل تابستان توجه کرد به دلیل مسافرت ها و شهریور هم بهع دلیل نزدیک شدن به خرید برای مدرسه خرید های والدین زیاد میشود
- (۴) جواهرات و لفوارم الکترونیکی و اپلیکیشن های خانگی هزینه زیادی دارند

- (۵) بیشتر خرید های آن ها از ۳ عدد همزمان بیشتر نمیشود
- (۶) در الگوریتم RFM برای گروه بندی بیشتر از ۳ گروه نمیشوند
- (۷) به دلیل داشتن چولگی چپ در مبلغ هر تراکنش باید از الگوریتم و صدک ها استفاده کرد تا گروه بندی با عدالت باشد
- (۸) حدود ۴۰۷ عدد ردیف دارای داده پرت وجود دارد

۵. منابع

- (۱) گوگل
CHATGPT (۲)
medium (۳)