گزارش کار تمرین سوم سوال 8

توسط: کتایون کبرائی
استاد مربوطه: دکتر فراهانی

چکیده

در این سوال از سری سوم تمرینات، به بررسی دیتاست Supermarket dataset for predictive marketing می‌پردازیم. در این تمرین به طور کلی از تکنیک های مختلف clustering استفاده شده تا بتوان مشتریان این بخش را به بهترین روش دسته‌بندی کرد .

- ابتدا مانند تمرین‌های گذشته نیاز است روی دیتا preprocessing مرتبط با آن انجام شود تا اطلاعات اشکار و پنهان داده ها باهم بدست آید و بتوان از آن‌ها بدست آید. این preprocessingشامل هندل کردن دیتای null وNaN ، encodeکردن دیتای categoricalو feature engeneering های مرتبط خواهد بود.
- در گام بندی مدل k-means clustering را روی دیتاست اعمال میکنیم تا به بهترین مقدار k برای دسته بندی دست یابیم. برای این کار از مقادیری مثل silhouette score و ... نیز استفاده خواهیم کرد.
- در مرحله بعدی دسته بندی های مختلف را با نمودار های 3بعدی و 2بعدی رسم میکنم. برای این کار از مدل PCA استفاده خوهیم کرد.
- در مراحل بعدی به ترتیب الگوریتم های دیگر مثل DBSCAN را با مدل خود مقایسه می‌کنیم، سعی میکنیم ابعاد دیتا را با استفاده از PCA کاهش دهیم و در نهایت تمام نتایج را در راستای بینش کامل روی دیتاست بررسی کنیم.

مقدمه

دیتاستی که روی آن کا ر می‌کنیم Supermarket dataset for predictive marketingمی‌باشد. این دیتاست در سال 2023 معرفی شد که شامل بیش از 2 میلیون رکورد از سوپرمارکت Hunter است و در نهایت از ما خواسته شده با استفاده از این رکوردها خریداران را به دسته های مناسبب تقسیم بندی کنیم و مشخص کنیم هر شخص در چه دسته ای قرار می‌گیرد. از انجایی که با یک مسئله clustering برخورد کرده‌ایم نیاز است مدل معروف k-means را روی آن اعمال کنیم. برای تشخیص بهتر این دسته ها هم میتوان قبل از train کردن مدل تغییراتی را روی دیتاست انجام دهیم. سپس میتوانیم انها را با مدل های جدیدتری از روش clustering نیز train کنیم که بهترین دسته بندی را پیدا کنیم. در ادامه به بررسی این روش‌ها با جزئیات بیشتررری می‌پردازیم.

قسمت اول: PREPROCESSING DATA

ابتدا ساختار کلی دیتاست یعنی تمامی فیچرهای مسئله، تعداد کل نمونه ها، ساختار هر فیچر و چند نمونه تصادفی بررسی میشود .برای این کار نیاز است با استفاده از تابع csv_read()فایل مورد نظر بارگذاری میشود و در یک dataframe ذخیره میشود .دیتاست موردنظر 12 ستون دارد. با استفاده از columns. لیست این ستون را بدست می‌اوریم که شامل order_id, user_id, order_number, order_dow, order_hour_of_day, days_since_prior_order, product_id, add_to cart_order, reordered, department_id, department, product_name می‌شود. با استفاده از تابع info() می‌توانیم اطلاعاتی راجب این این فیچرها بدست اوریم:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2019501 entries, 0 to 2019500
Data columns (total 12 columns):
 #   Column                  Dtype
---  ------                  -----
 0   order_id                int64
 1   user_id                 int64
 2   order_number            int64
 3   order_dow               int64
 4   order_hour_of_day       int64
 5   days_since_prior_order  float64
 6   product_id              int64
 7   add_to_cart_order       int64
 8   reordered               int64
 9   department_id           int64
 10  department              object
 11  product_name            object
dtypes: float64(1), int64(9), object(2)
memory usage: 184.9+ MB
```

باتوجه به اطلاعات بالا درمی‌یابیم تنها دو فیچر department و product_name عددی نیست و نیاز به دیکود شدن دارد. میتوانیم با استفاده از تابع ()describe اطلاعات بیشتری برای داده های عددی داشته باشیم:

	order_id	user_id	order_number	order_dow	order_hour_of_day	days_since
count	2.019501e+06	2.019501e+06	2.019501e+06	2.019501e+06	2.019501e+06	1
mean	1.707013e+06	1.030673e+05	1.715138e+01	2.735367e+00	1.343948e+01	1
std	9.859832e+05	5.949117e+04	1.752576e+01	2.093882e+00	4.241008e+00	8
min	1.000000e+01	2.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0
25%	8.526490e+05	5.158400e+04	5.000000e+00	1.000000e+00	1.000000e+01	5
50%	1.705004e+06	1.026900e+05	1.100000e+01	3.000000e+00	1.300000e+01	8
75%	2.559031e+06	1.546000e+05	2.400000e+01	5.000000e+00	1.600000e+01	1
max	3.421080e+06	2.062090e+05	1.000000e+02	6.000000e+00	2.300000e+01	3

برای اشنای بیشتر برای انواع داده هایی که در هر ستون میبینیم میتوانیم از تابع ()unique. استفاده کنیم:

```
order_id                200000
user_id                 105273
order_number               100
order_dow                    7
order_hour_of_day           24
days_since_prior_order      31
product_id                 134
add_to_cart_order          137
reordered                    2
department_id               21
department                  21
product_name               134
dtype: int64
```

میبینم که به طور کلی 200000 رکورد وجود دارد. برای هر فیچر تعداد معنا دار و مشخصی از انواع دیتا وجود دارد که میتوانیم روی آن‌ها کار کنیم.

در قسمت بعدی به هندل کردن مقادیر NaN در دیتاست میپردازیم. روش های زیادی برای هندل کردن این مقادیر وجود دارد. برای مثال در ساده ترین فرم ها میتوان مقدار مشخصی برای تمامی ان‌ها قرار داد. ابتدا باید فیچر هایی که این مقادیر در ان‌ها وجود دارند را با استفاده از تابع ()isnull. و سپس فراخوانی تابع ()sum. روی ان تعداد این مقادیر را برای هر فیچر بدست اوریم:

```
order_id                      0
user_id                       0
order_number                  0
order_dow                     0
order_hour_of_day             0
days_since_prior_order    124342
product_id                    0
add_to_cart_order             0
reordered                     0
department_id                 0
department                    0
product_name                  0
dtype: int64
```

همان طور که دیده میشود فیچر days_since_prior_order مقادیر NaN زیادی دارد. این فیچر به ما نشان میدهد که چند روز از اخرین خرید مشتری گذشته است پس هرکدام از این مفادیر NaN ارزشمند بوده‌اند. راه های زیادی برای هندل کردن مقادیر NaN وجود دارد. اولین و ساده ترین راه گذاشتن مقادیر ثابتی برای آن یا میانگین بقیه سطرها است. از انجایی که این مقادیر بین 0 تا 30 هستند باید مقدار ما ارزشی معنادار به این فیچر ندهد پس متوانیم تنها عدد 1- را به ازای این مقادیر بگذاریم. راه های بهتر استفاده از knn یا رگرسیون است که این مقادیر NaN را پیش بینی میکند. ما برای این مقادیر از رگرسیون استفاده کردیم. برای این کار کافیست یک شیئ از کلاس IterativeImputer بسازیم که random forest estimator را برای ان انتخاب میکنیم. سپس یک کپی از فیچر موردنظر خود به تابع ()fit این شیئ میدهیم تا برای ما مقادیر را هندل کند. در نهایت ستون جدید را به جای ستون قبلی در دیتافریم اصلی قرار می‌دهیم. اگر این ستون جدید را با مقدار قبلی مقایسه کنیم میبینیم مقادیر به خوبی پیش بینی شده اند:

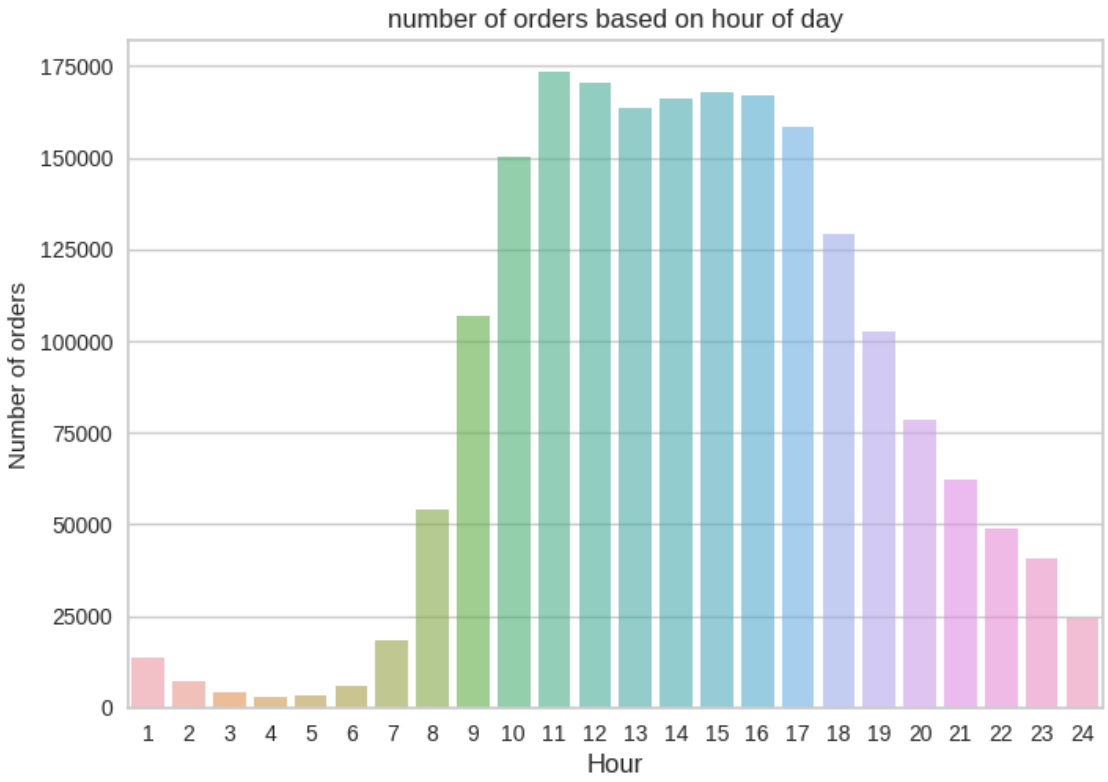
ستون قبلی شده

50	6.0
51	6.0
52	6.0
53	6.0
54	6.0
55	6.0
56	6.0
57	6.0
58	6.0
59	6.0
60	6.0
61	6.0
62	6.0
63	7.0
64	7.0
65	7.0
66	7.0
67	7.0
68	7.0
69	NaN

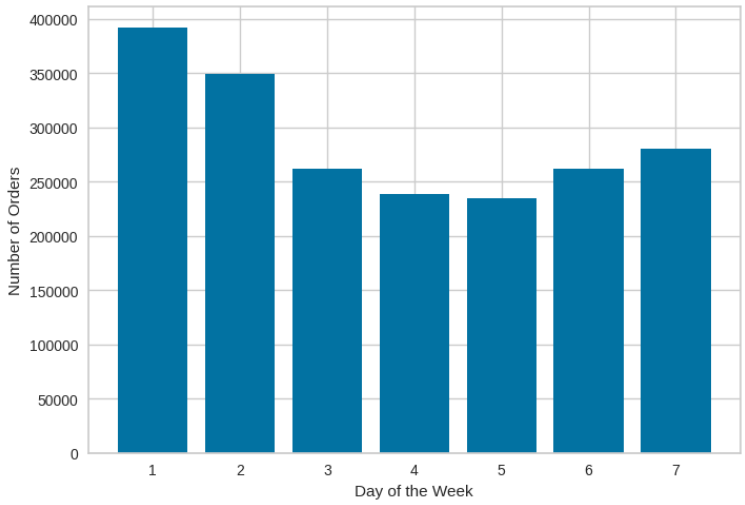
ستون جدید با مقادیر پیش‌بینی

50	6
51	6
52	6
53	6
54	6
55	6
56	6
57	6
58	6
59	6
60	6
61	6
62	6
63	7
64	7
65	7
66	7
67	7
68	7
69	11

در مرحله بعدی این قسمت با استفاده از نمودارهای مختلف وضعیت دیتای هر فیچر را میسنجیم. ابتدا با توجه به معنای هر فیچر میبینیم مثلا در کدام ساعت از روز تعداد خرید بیشتری انجام میشود:

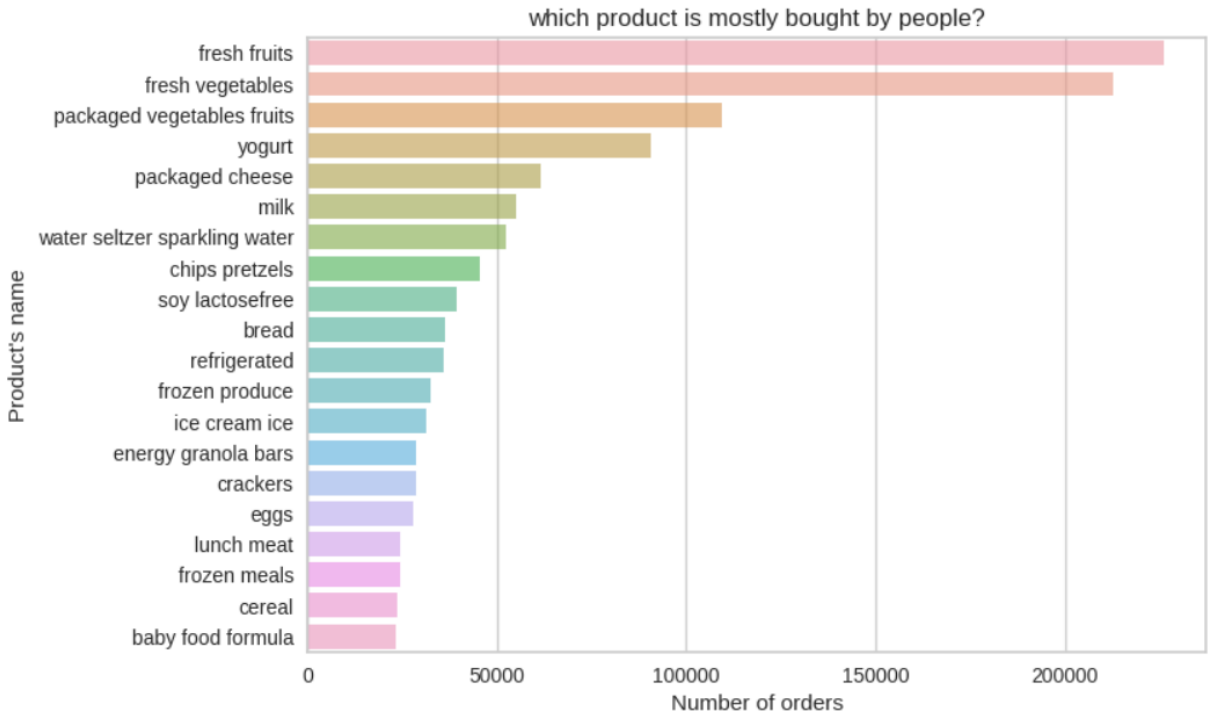


با استفاده از این نمودار میبینیم در ساعت 11 صبح بیشترین مقدار خرید در روز و به طور کلی از ساعت 10 تا 17 میزان خرید بسیار بیشتر است. اگر این بار بخواهیم ببینیم در کدام روز از هفته بیشتر مردم خرید میکنند نیز خواهیم دید:



با توجه به نمودار در دو روز اول هفته مقدار خرید بیشتری از فروشگاه میشود و این مقدار در روز های دیگر تا روز پنجم هفته کاهش و برای دو روز اخر هفته نیس کمی افزایش خواهد داشت.

خوب است معروف ترین محصولات را میان مردم نیز بشناسیم:

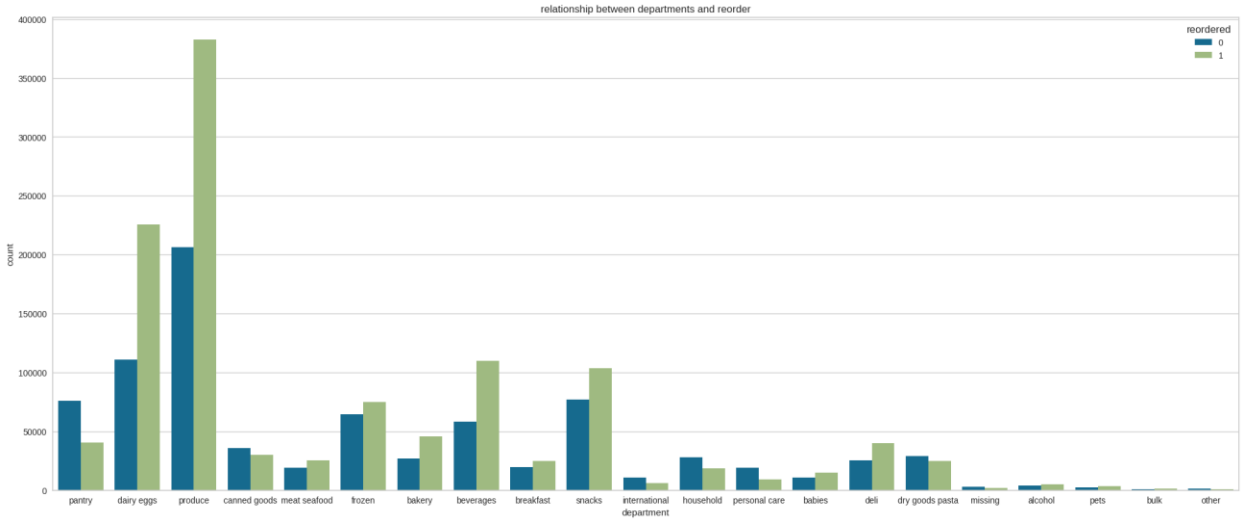


همان طور که میبینیم بیشتری محصولی که از این فروشگاه خریده میشود میوه تازه و بعد از به ترتیب سبزیجات تازه و سبزی و میوه بسته بندی شده است. بعد از ان ماست و پنیر بسته بندی شده قرار میگیرند.

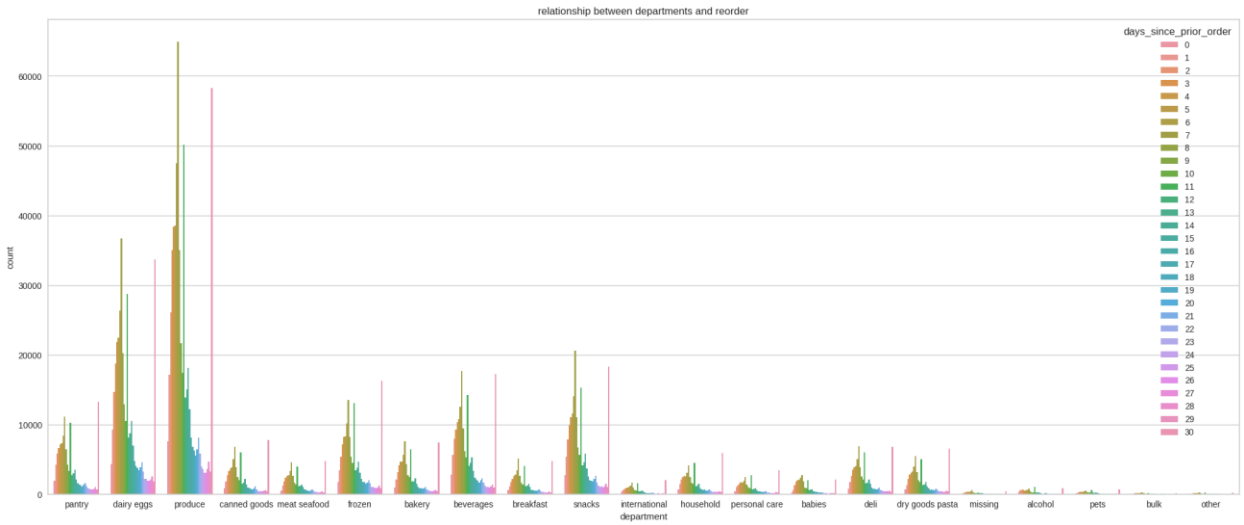
حال اگر بخواهیم چک کنیم در هر category پرفروش ترین محصول چیست مقادیر زیر را خواهیم داشت:

department	
alcohol	beers coolers
babies	baby food formula
bakery	bread
beverages	water seltzer sparkling water
breakfast	cereal
bulk	bulk grains rice dried goods
canned goods	soup broth bouillon
dairy eggs	yogurt
deli	lunch meat
dry goods pasta	dry pasta
frozen	frozen produce
household	paper goods
international	asian foods
meat seafood	hot dogs bacon sausage
missing	missing
other	other
pantry	baking ingredients
personal care	oral hygiene
pets	cat food care
produce	fresh fruits
snacks	chips pretzels

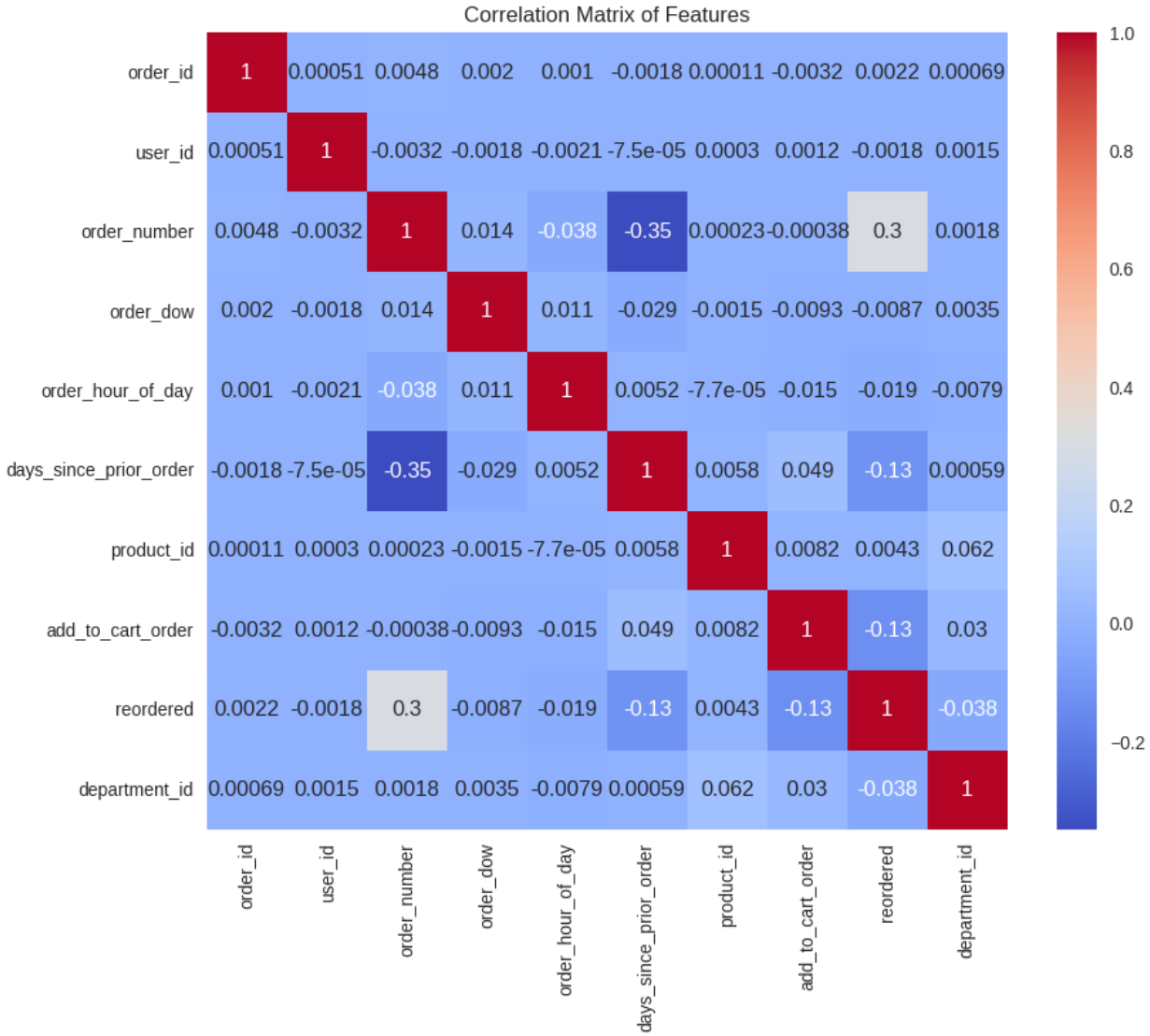
در گام بعدی نیاز است به روابط اشکار و نهان فیچرها باهم بپردازیم. اولین چیزی که بررسی میکنیم این است که ایا پیش خرید کردن یک محصول توسط شخصی ارتباطی با categoryها دارد یا نه:



رابطه دیگر که میتوانیم بررسی کنیم ارتباط تعداد روزهایی که از اخرین خرید گذشته و نوع categoryای است که محصول در ان قرار دارد.



در نهایت هم correlation matrix را برای تک‌تک این فیچرها رسم میکنیم:



در مرحله بعدی preprocessing دیتا باید فیچرهای categorical را اینکود کنیم پس مثلا اگر فیچر product را چک کنیم خواهیم داشت:

قبل از اینکود کردن

product_name
baking ingredients
soy lactosefree
butter
fresh vegetables
fresh vegetables

بعد از اینکود کردن

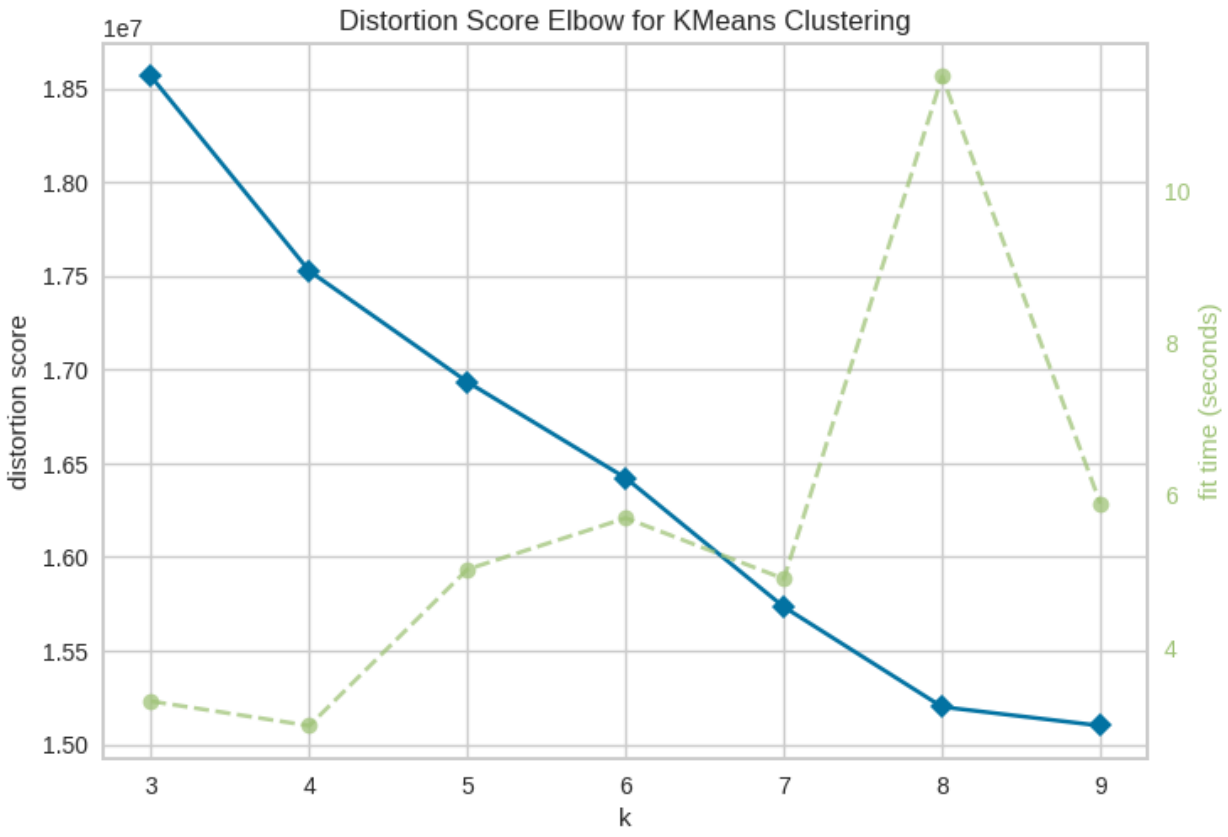
0	6
1	119
2	17
3	53
4	53

از انجایی که دیتا در هر فیچر رنج متفاوتی دارد نیاز است دیتا نیز scale شود. از scaler های متفاوتی میتوان استفاده کرد مثل minMaxScaler و یا standardScaler که در اینجا از standardScaler استفاده میکنیم. برای این از کلاس StandardScaler شیئ میسازیم. سپس ان شیئ را روی دیتافریم با تابع ()fit_transform صدا میکنیم تا تمامی مقادیر دیتافریم scale شود.

قسمت دوم:

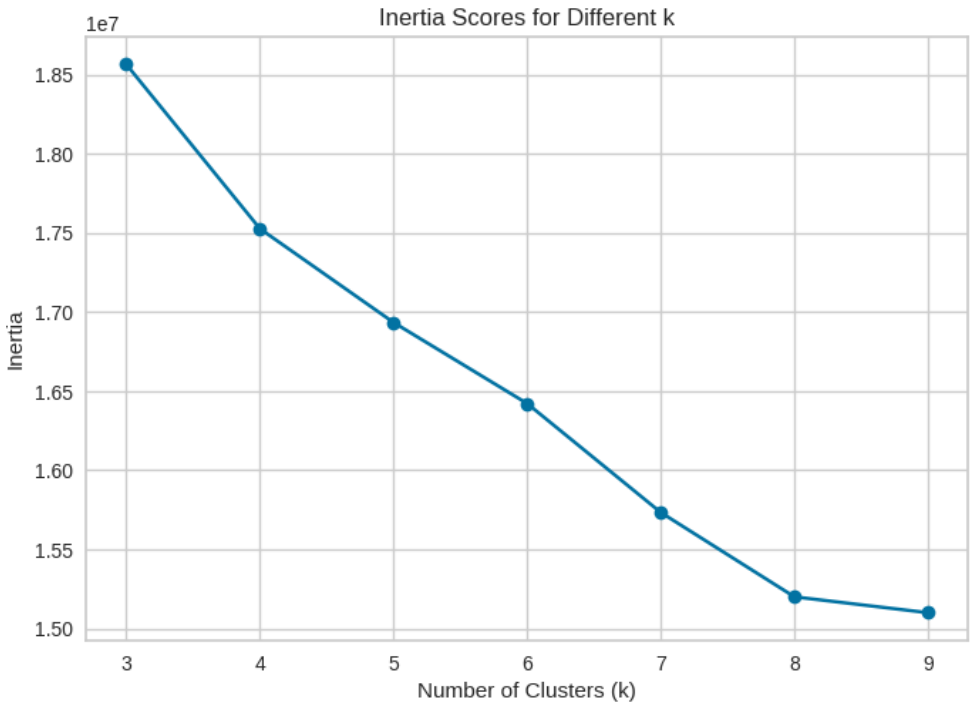
IMPLEMENTING KNN MODEL

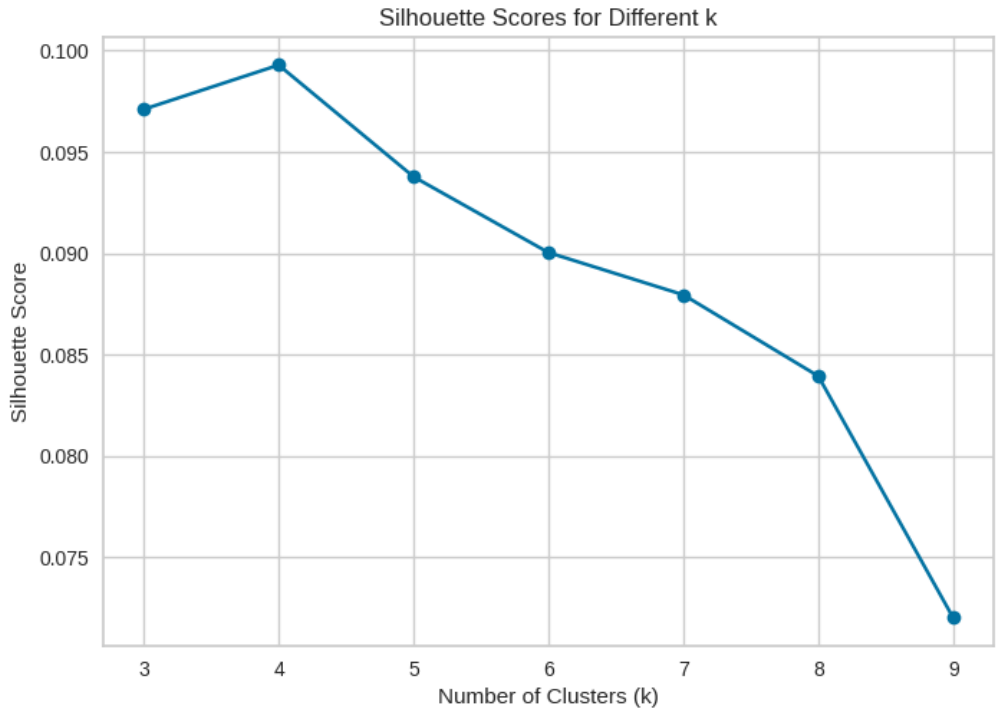
در قسمت اول نیاز است از کلاس Kmeans شئی بسازیم. سپس برای اینکه بهترین مقدار k برای این دیتاست پیدا شود از کلاس KElbowVisualizer استفاده میکنیم. برای اینکار ورودی کلاس به ان مدل K-means را و رنج kای که میخواهیم برای ما بسنجد، به ان میدهیم. سپس تابع fit این شئی را برای دیتای scaled شده مرحله قبل صدا میکنیم. نمودار ان خواهد شد:



میبنیم که این کلاس تشخیص داده بهترین میزان برای ما تقریبا 6 خواهد بود؛ زیرا به دنبال نقطه ای در نمودار هستیم که در آن کاهش inertia شروع به یکسان شدن می کند، که نشان دهنده کاهش بازدهی افزودن cluster های بیشتر است. این نقطه اغلب به عنوان "elbow" نامیده می شود و نشان دهنده تعادل خوبی بین گرفتن cluster های معنی دار و اجتناب از overfitting است. اما در ادامه خواهیم دید بهترین نتیجه برای ما k=5 خواهد بود.

حال نیاز است مقادیر مختلف k را خود امتحان کنیم تا با توجه به inertia score و silhouette score بهترین k را پیدا کنیم.





از نمودارها میبینیم که بهترین k، برابر است با k=5. پس مدل را بار دیگر با این k حل میکنیم. زمانی که clustering پایان می‌یابذ میتوانیم ستون شماره دسته برای هر رکورد را چک کنیم تا مطمئن شویم مدل به درستی کار کرده است:

_order	product_id	add_to_cart_order	reordered	department_id	department	product_name	cluster
11	17	1	0	13	16	6	2
11	91	2	0	16	7	119	4
11	36	3	0	16	7	17	4
11	83	4	0	4	19	53	2
11	83	5	0	4	19	53	2
11	91	6	0	16	7	119	4
11	120	7	0	16	7	133	4
11	59	8	0	15	6	21	4
11	35	9	0	12	13	104	2
11	37	1	0	1	10	71	2
11	24	2	0	4	19	50	2
11	83	3	0	4	19	53	2
11	84	4	0	16	7	83	4
11	91	5	0	16	7	119	4
11	24	6	0	4	19	50	2
11	24	7	0	4	19	50	2
11	24	8	0	4	19	50	2
11	21	9	0	16	7	93	4
11	112	10	0	3	2	11	4

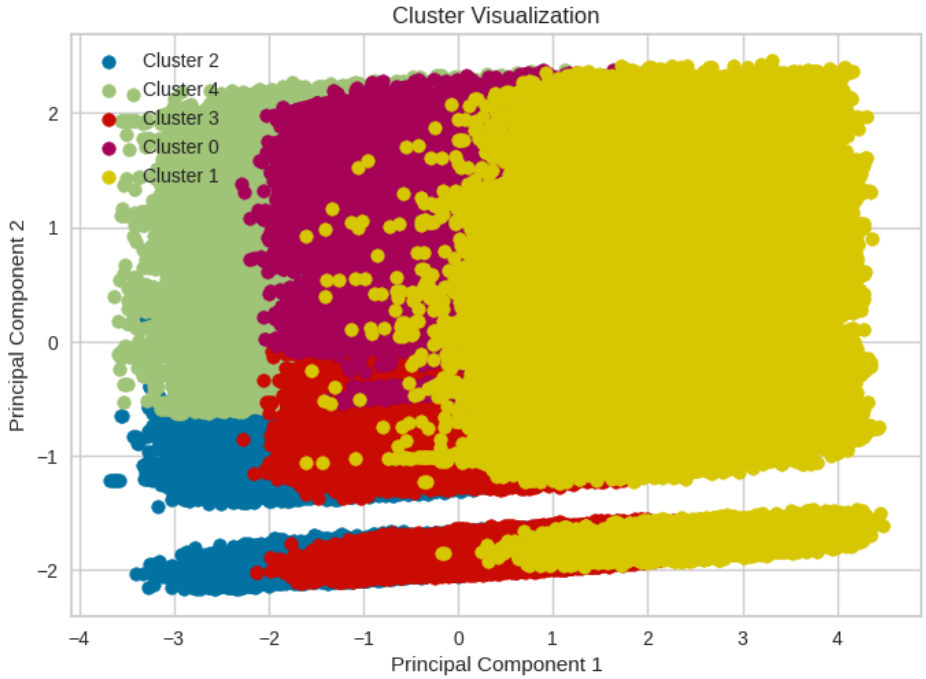
سپس میتوانیم چک کنیم در هر cluster چه تعداد از دیتای ما وجود دارد:

```
0    516607
3    475786
4    435354
2    359056
1    232698
Name: cluster, dtype: int64
```

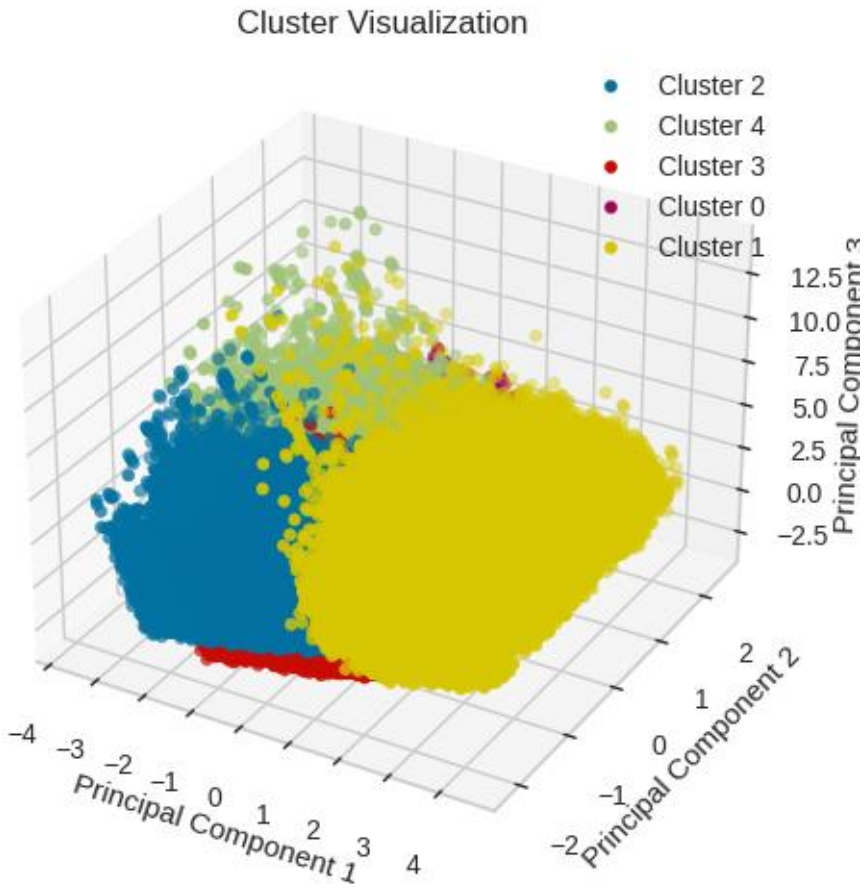
قسمت سوم: VISUALIZATION

در این قسمت از مسئله با استفاده از مدل هایی مثل PCA به نمایش دو بعدی و سه بعدی دیتاست دسته بندی شده میپردازیم.

ابتدا، یک شیء PCA ایجاد میکنیم و تعداد مولفه‌ها (n_components) را ۲ تنظیم میکنیم. سپس scaled_data به تابع fit_transform ارسال میکنیم تا تجزیه ترکیب خطی را اعمال شود و داده‌ها را به فضای دو بعدی تبدیل کند. سپس، داده‌های تبدیل شده را در یک دیتافریم جدید با نام df_pca ذخیره میکنیم و دو ستون به نام‌های 'PC1' و 'PC2' به آن اضافه میکنیم.. با استفاده از مقادیر unique_clusters، یک حلقه روی این مقادیر ایجاد میکنیم. در نهایت نتیجه را به شکل زیر نمایش میدهیم:



و برای نمایش سه بعدی ان به طریق مشابه و تنها با سه مولفه خواهیم داشت:



قسمت چهارم:

EXPERIMENTING WITH OTHER CLUSTERING ALGORITHMS

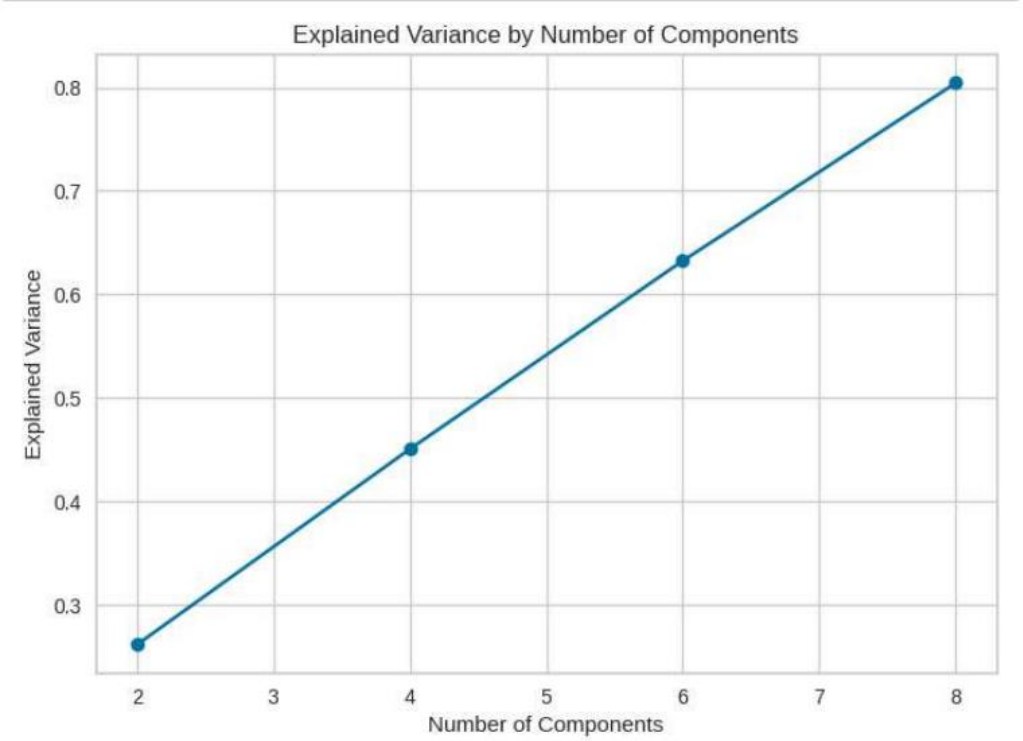
در این قسمت از تمرین الگوریتم های دیگر clustering را روی دیتاست تست میکنیم.
الگوریتمی که امتحان میکنیم DBSCAN و hierarchical clustering خواهد بود و در اخر پیاده سازی با استفاده از silhouette_score با مدل k-means مقایسه میکنیم.

```
K-means: 0.09236717828273076
DBSCAN: DBSCAN identified only one cluster.
Hierarchical Clustering: 0.07368430608775553
```

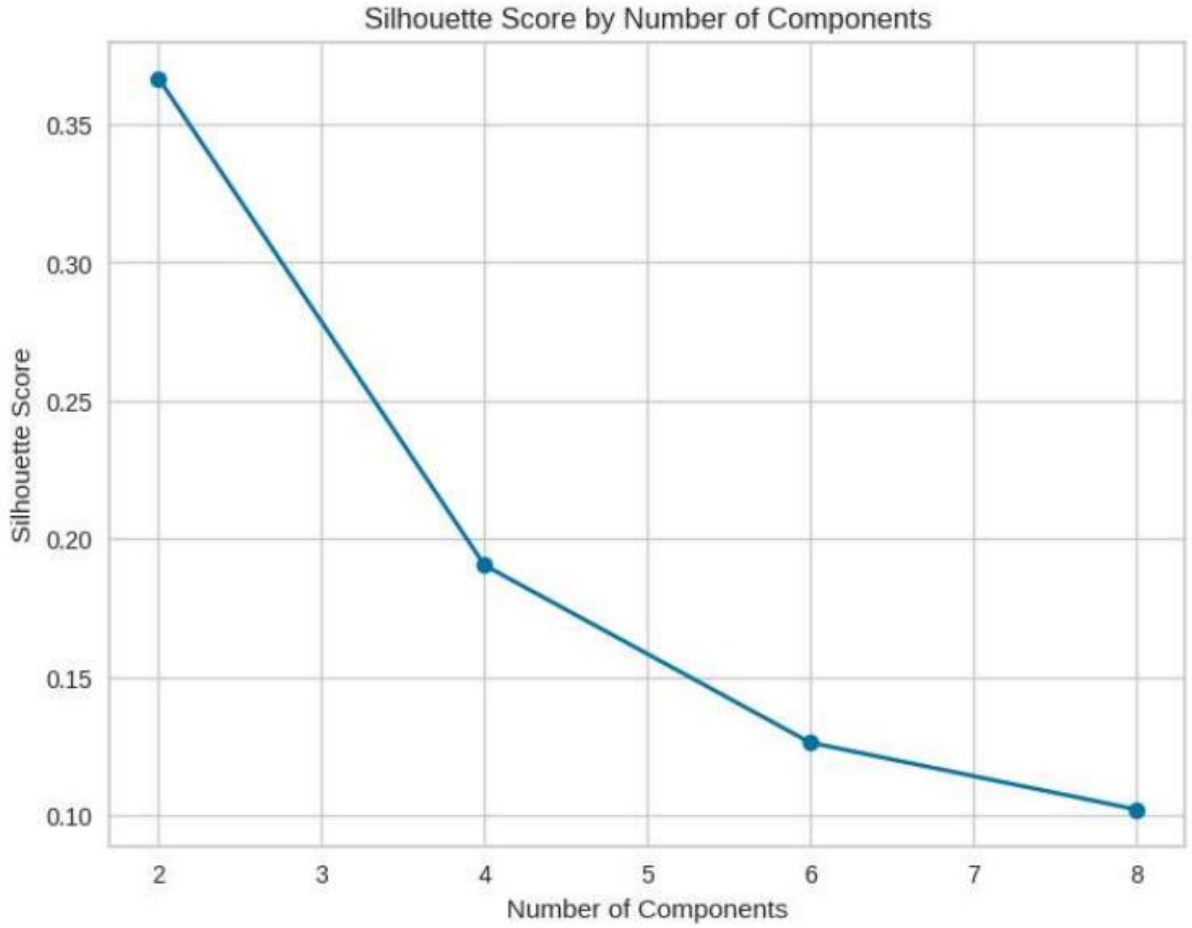
قسمت پنجم:

DIFFERENT NUMBER OF COMPONENTS

در این قسمت برای کاهش ابعاد از تعداد مولفه های محتلفی استفاده میکنیم تا بهترین تعداد بدست اید. برای این کار explained variance را به ازای هر تعداد مولفه بدست می‌اوریم و نهایتا ان را چاپ میکنیم.



اگر مقدار silhouette score را نیز به ازای هر مولفه چاپ کنیم خواهیم داشت:



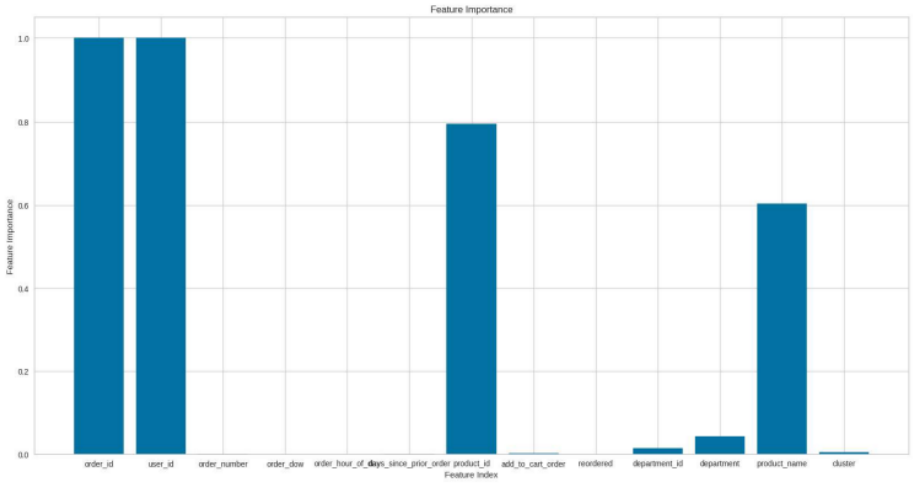
از انجایی که هم باید این مقدار پایین باشد و هم واریانس پایین و با توجه به نمودارهایشان این دو عکس هم هستند، به نظر می‌اید مولفه 5 مناسب تر خواهد بود.

قسمت ششم:

FEATURE ENGENEERING

در این قسمت ابتدا نیاز است مهم ترین فیچر های دیتاست را پیدا کنیم. برای این کار با استفاده از PCA به ترتیب تاثیرگذار ترین ویژگی‌ها را پیدا میکنیم که خواهند بود:

```
Feature 0 -> order_id : 1.0000311749976503%
Feature 1 -> user_id : 1.0000315609391568%
Feature 2 -> order_number : 0.000471133592124569%
Feature 3 -> order_dow : 6.573607663956962e-05%
Feature 4 -> order_hour_of_day : 8.935691911234986e-05%
Feature 5 -> days_since_prior_order : 0.0012356737713178404%
Feature 6 -> product_id : 0.7954423061179605%
Feature 7 -> add_to_cart_order : 0.0026929381934968142%
Feature 8 -> reordered : 0.00023313130728591064%
Feature 9 -> department_id : 0.014241212976986349%
Feature 10 -> department : 0.042582343030184426%
Feature 11 -> product_name : 0.604333346844525%
Feature 12 -> cluster : 0.0053344609281428 16%
```



حال نیاز است فیچر های بدون کاربرد را حذف و تنها مهم‌ترین ها را نگه داریم. پس برای این کار تنها فیچر های ' ,'order_number', 'order_dow
'order_hour_of_day', 'days_since_prior_order',
'add_to_cart_order', 'reordered', 'department_id’ را حذف میکنیم و برای فیچر های باقی مانده دوباره مدل را اجرا و میسنجیم:

```
Original K-means Silhouette Score: 0.093380473977278 99
Original K-means Inertia: 16934521.355000548
```

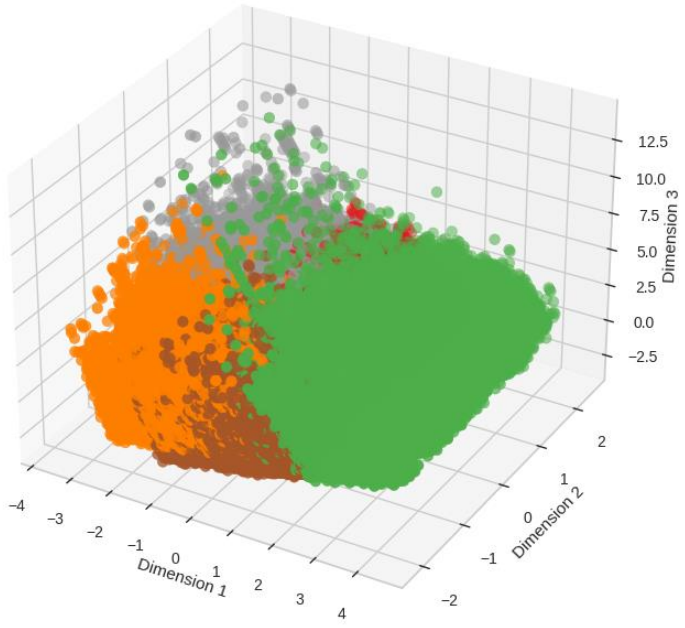
```
Filtered K-means Silhouette Score: 0.5298672062884408
Filtered K-means Inertia: 2130411261927395.5
```

همان‌طور که مشاهده میشود عملکرد مدل بهبود یافته است.

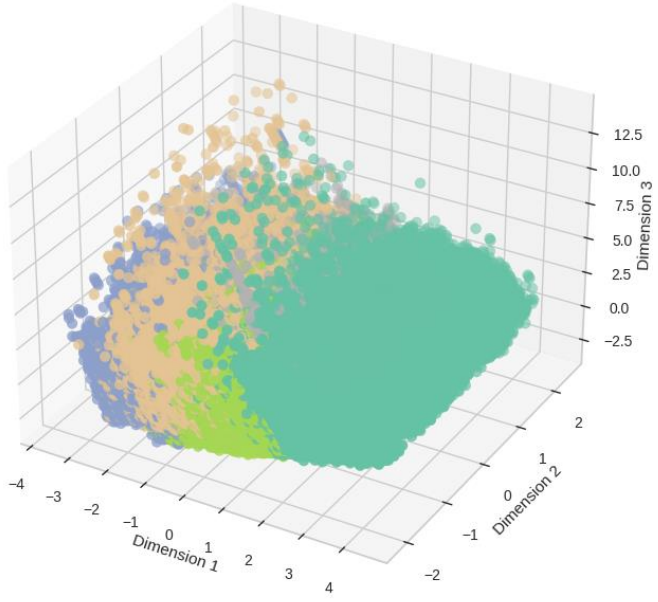
در گام بعدی میتوان فیچر جدید اضافه کرد. میتوانیم فیچر های جدیدی به اسم های avg_days_since_prior_order و order_dow_category بسازیم و عملکرد مدل را با این دو فیچر جدید بسنجیم:

```
Original K-means Silhouette Score: 0.093380473977278 99
Original K-means Inertia: 16934521.355000548
New K-means Silhouette Score: 0.15871459493003096
New K-means Inertia: 8905079.397437796
```

Original K-means Clustering

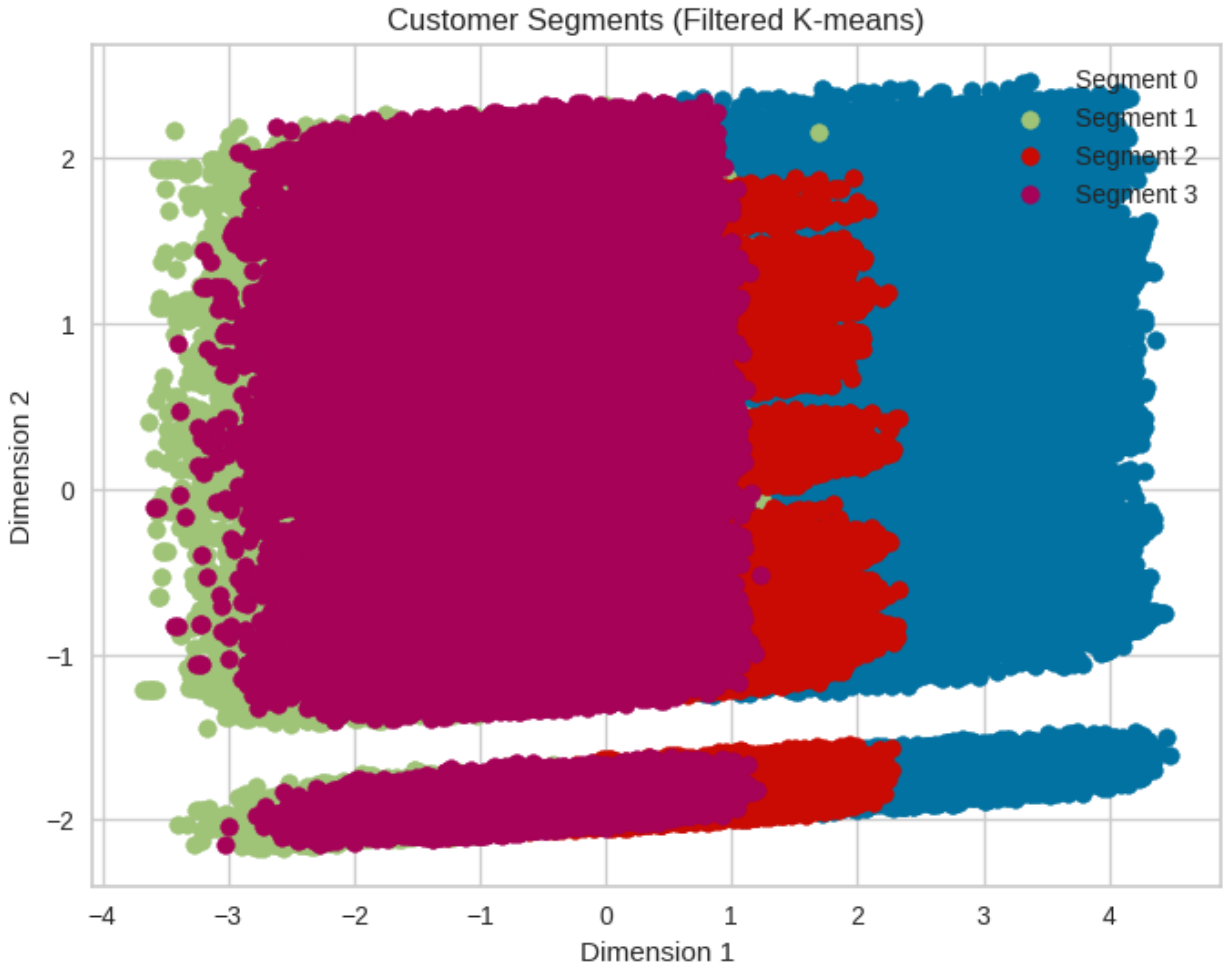


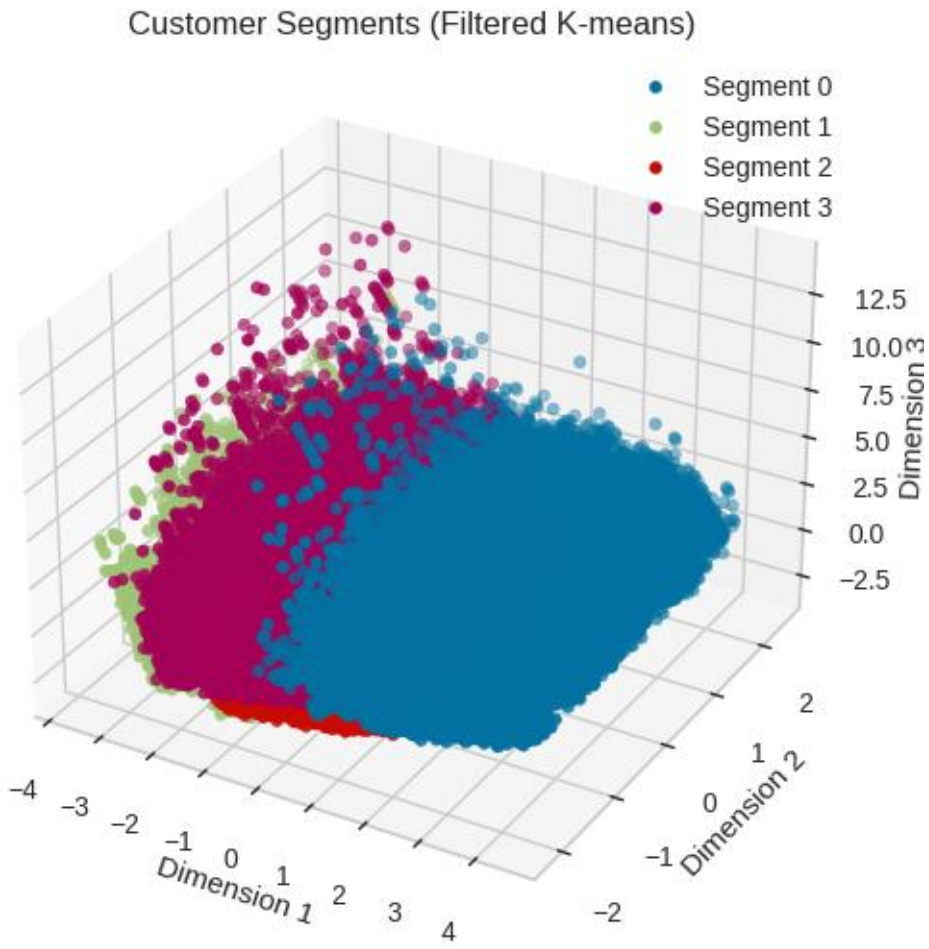
Filtered K-means Clustering



قسمت هفتم: INSIGHTS

حال در قسمت نهایی تمرین میتوانیم نهایتا به درک و بینش بهتری از کل دیتاست موردنظر برسیم. بهترین مدل خود را انتخاب کرده و نهایتا دیتاست را با cluster 5 به صورت دوبعدی و سه بعدی رسم میکنیم:




Customer Segments (Filtered K-means)
Segment 0
Segment 1
Segment 2
Segment 3
Dimension 1
Dimension 2
Dimension 3
−4
−3
−2
−1
0
1
2
3
4
−2
−1
0
1
2
12.5
10.0
7.5
5.0
2.5
0.0
−2.5