باسمه تعالی

درس: مبانی داده‌کاوی

نام و نام خانوادگی: تینا توکلی و هیوا شاهرخ

شماره دانشجویی: 9922762220 , 9912762121

< فاز دوم پروژه داده کاوی>

تسک اول:

در این فاز می بایست حداقل 7 مورد از الگو های مکرر موجود در دیتاست clean حاصل از فاز قبلی را استخراج کنیم.

برای اینکار طبق آموخته های درس از الگوریتم Apriori استفاده کردیم.

برای تسهیل فرایند از تابع get_dummiesدر پانداس استفاده می‌کنیم تا داده‌های کاتگوریکال (طبقه‌ای) را به روش one-hot در DataFrame را به داده‌های عددی تبدیل کند .برای داده های عددی مقادیر ستون فعلی را با میانگین مقایسه می‌کنیم اگر مقدار بزرگتر از میانگین باشد، به ۱ تبدیل می‌شود، در غیر این صورت به ۰ تبدیل می‌شود .سپس، مقادیر جدید در همان ستون ذخیره می‌شوند.

سپس با استفاده از تابع اماده apriori داده های frequentرا بدست میاوریم (برای اینکه خروجی قابل فهم تر شود انها ستون تعداد را اضافه کردیم و بر اساس طول و پشتیبانی مرتب می‌کند .آیتم ست‌های کوتاه‌تر با support بالاتر در اولویت قرار می‌گیرند)

	support	itemsets	length
0	1.0	(Ad Supported)	1
3	0.995536	(Currency_USD)	1
4	0.846429	(Rating Category_Excellent)	1
2	0.598214	(Avarage Rating)	1
1	0.5375	(Rating)	1
7	0.995536	(Ad Supported, Currency_USD)	2
8	0.846429	(Rating Category_Excellent, Ad Supported)	2
12	0.841964	(Rating Category_Excellent, Currency_USD)	2
6	0.598214	(Avarage Rating, Ad Supported)	2
11	0.598214	(Avarage Rating, Currency_USD)	2
5	0.5375	(Ad Supported, Rating)	2
10	0.5375	(Rating Category_Excellent, Rating)	2
9	0.533036	(Rating, Currency_USD)	2
16	0.841964	(Rating Category_Excellent, Ad Supported, Curr...	3
15	0.598214	(Avarage Rating, Ad Supported, Currency_USD)	3
14	0.5375	(Rating Category_Excellent, Ad Supported, Rating)	3
13	0.533036	(Ad Supported, Rating, Currency_USD)	3
17	0.533036	(Rating Category_Excellent, Rating, Currency_USD)	3
18	0.533036	(Rating Category_Excellent, Ad Supported, Rati...	4

تحلیل:

ما پس از بررسی min_supprtهای مختلف عدد 0.5 را انتخاب کردیم زیرا که هم تعداد frequent patterns منطقی ای بدست میاورد هم خیلی سختگیرانه نیست.

حال rule ها را با استفاده از تابع اماده association_rules بدست میاوریم.(این بخش اضافه بر پروژه انجام شده)

چند تا از rule های منطقی:
{Ad Supported} -> {Currency_USD}

- Support: 0.9955
- Confidence: 0.9955

این قانون نشان می‌دهد که تقریباً همه‌ی برنامه‌هایی که از تبلیغات پشتیبانی می‌کنند، از دلار آمریکا به عنوان واحد پولی خود استفاده می‌کنند که از منظر اقتصادی منطقی است

{Avarage Rating} -> {Ad Supported, Currency_USD}

- Support: 0.5982
- Confidence:1.0

برنامه‌هایی که امتیاز متوسط خاصی دارند، معمولاً هم از تبلیغات پشتیبانی می‌کنند و هم از دلار آمریکا به عنوان واحد پولی استفاده می‌کنند که به نشانه ارتباط بین رضایت کاربر و استراتژی پولی می‌باشد.

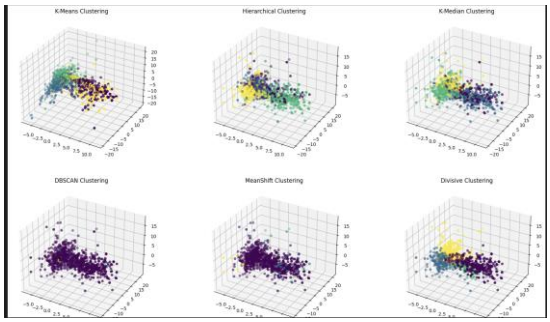
{Rating Category_Excellent, Ad Supported} -> {Currency_USD}

- Support: 0.8420
- Confidence: 0.9955

برنامه‌هایی با امتیاز بالا (امتیاز عالی) که از تبلیغات پشتیبانی می‌کنند، بسیار احتمال دارد از دلار آمریکا استفاده کنند که با انتظار مطابقت دارد که برنامه‌های موفق از طریق تبلیغات به دلار پولی خود را کسب کنند.

تسک دوم:

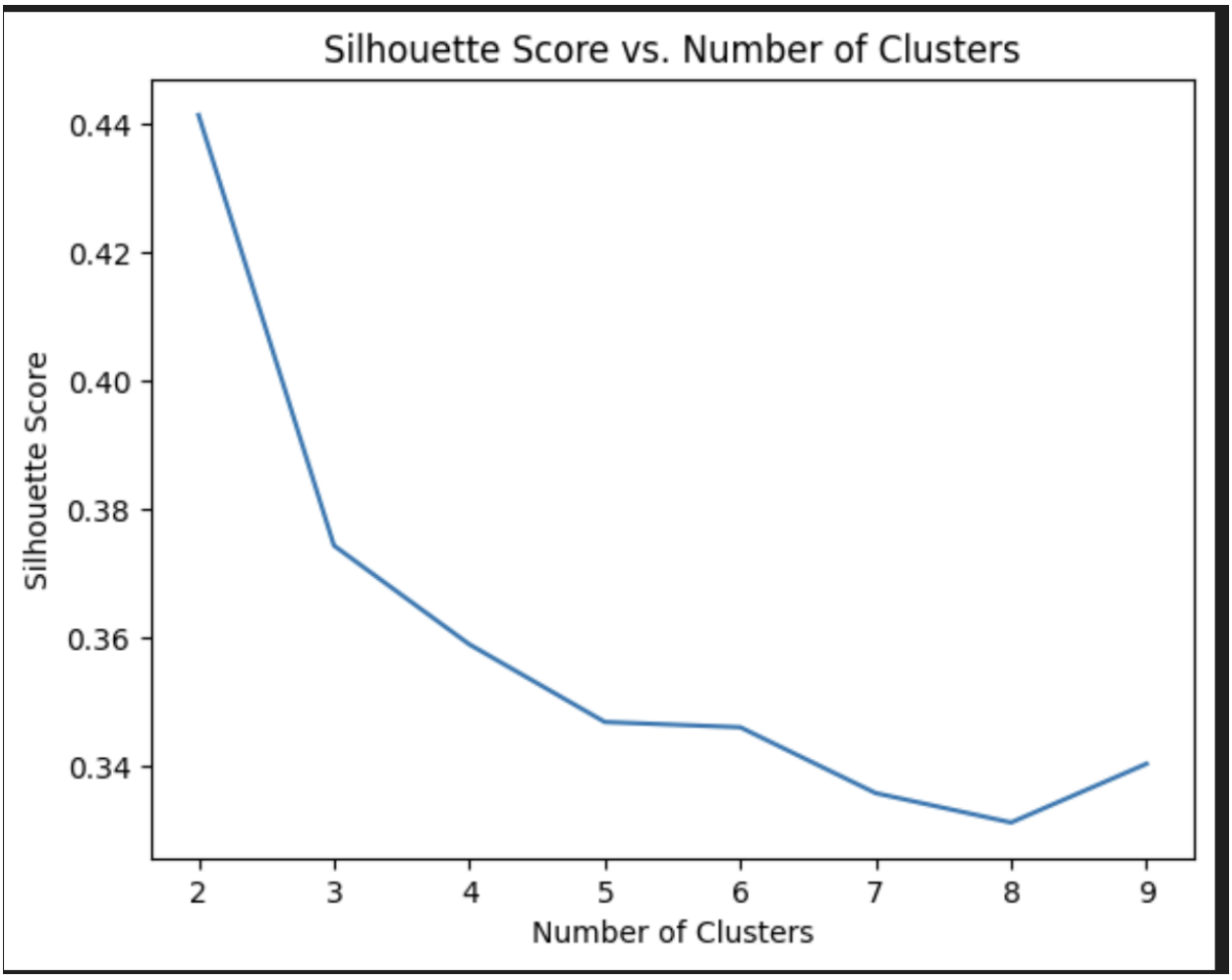
1-خوشه بندی و مصور سازی:
در این بخش از ما خواسته شده بود تا الگوریتم های مختلف خوشه بندی را روی دیتاست انجام داده و الگوهای موجود در خوشه هارا شناسایی کنیم؛ ما از 6 الگوریتم kmeans ، Agglomerative ، k_median ، DBSCAN ، MeanShift ، Divisive را به صورت رندوم انجام دادیم و از DBSCAN و MeanShift خروجی های مناسبی نگرفتیم که انتظار میرفت زیرا دیتاست ما ساختار مشخصی ندارد که بخواهیم ساختار انرا حفظ کنیم.



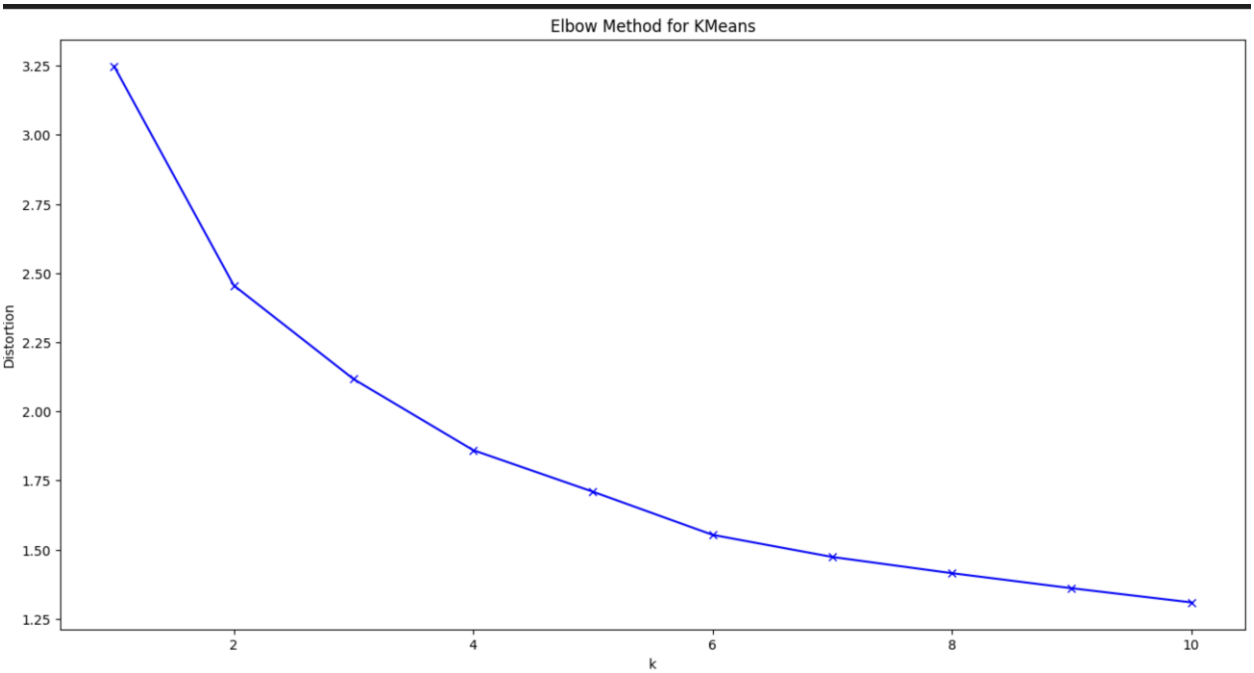
حال روی 3 الگوریتم kmeans ، Hierarchical Clustering و K-Median تحلیل انجام میدهیم تا به خوشه بندی بهتری برسیم.
کد بر روی تعداد مختلف خوشه ها (2 تا 9) تکرار می شود و امتیاز silhouette را برای هر کدام محاسبه می کند.

امتیاز silhouette یک معیار برای ارزیابی کیفیت خوشه‌بندی است و نشان می‌دهد که هر نمونه چقدر به خوشه خود نزدیک و از خوشه‌های دیگر دور است. امتیاز سایلوئت بین -1 تا 1 متغیر است، که مقدار بالاتر نشان‌دهنده خوشه‌بندی بهتر است.

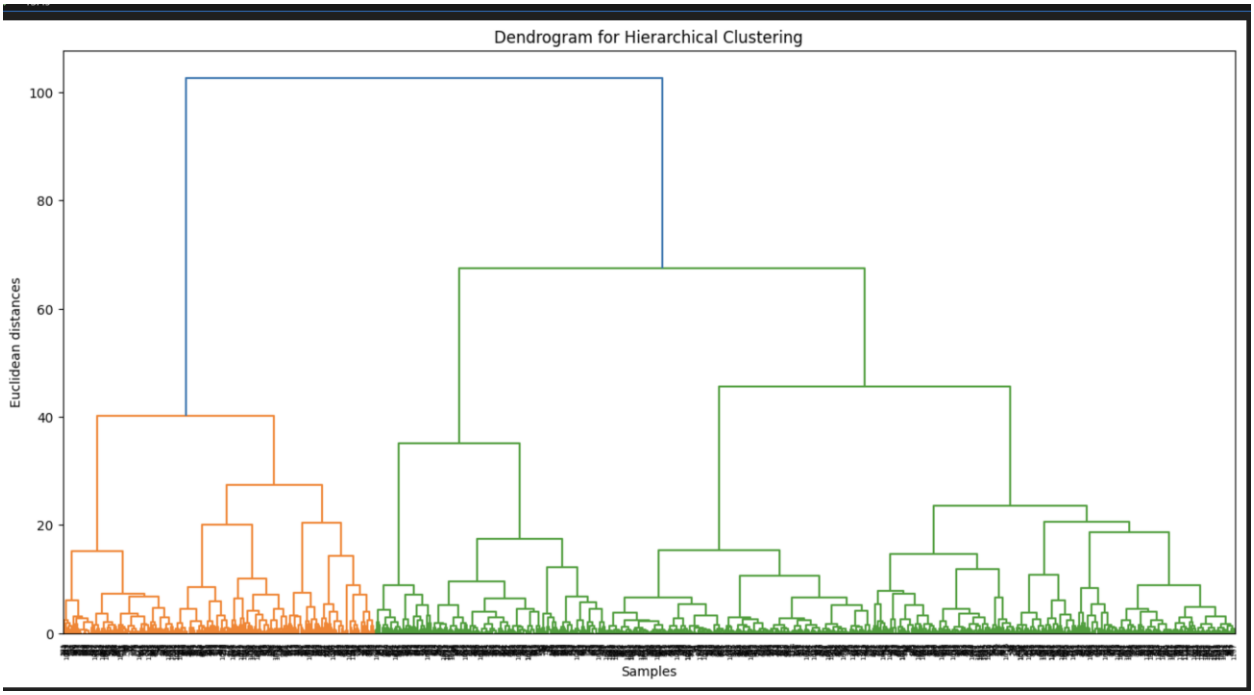
نمودار امتیازهای silhouette را برای فهم بهتر رسم کردیم بهترین امتیاز برای تعداد کلاستر 2 بود که با توجه به اینکه عدد کوچکی است ما عدد 4 را برای تعداد کلاستر ها با توجه به اینکه امتیاز نسبتا خوبی دارد و در elbowنیز خوب بود انتخاب کردیم.



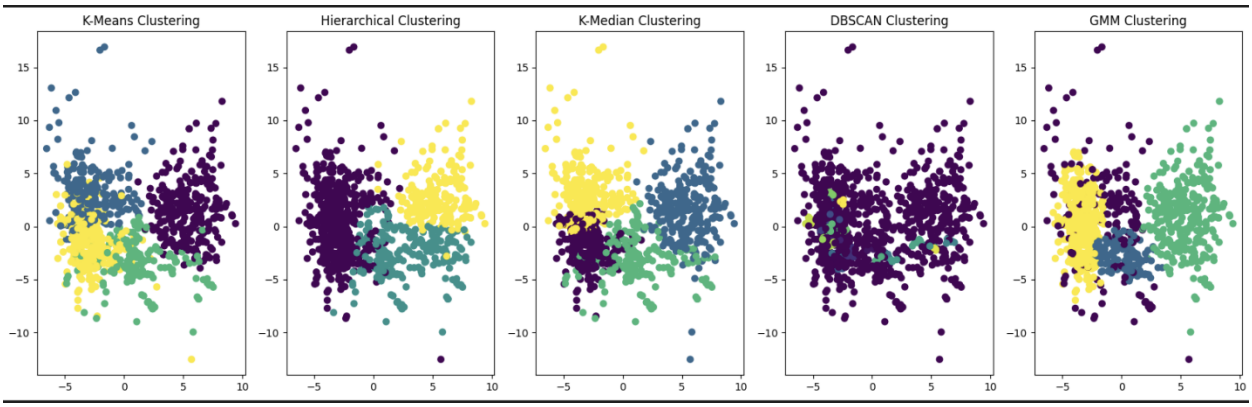
از Elbow Method برای تعیین تعداد بهینه خوشه‌ها نیز استفاده شده (نقطه آرنج نشان‌دهنده تعداد بهینه خوشه‌ها است).



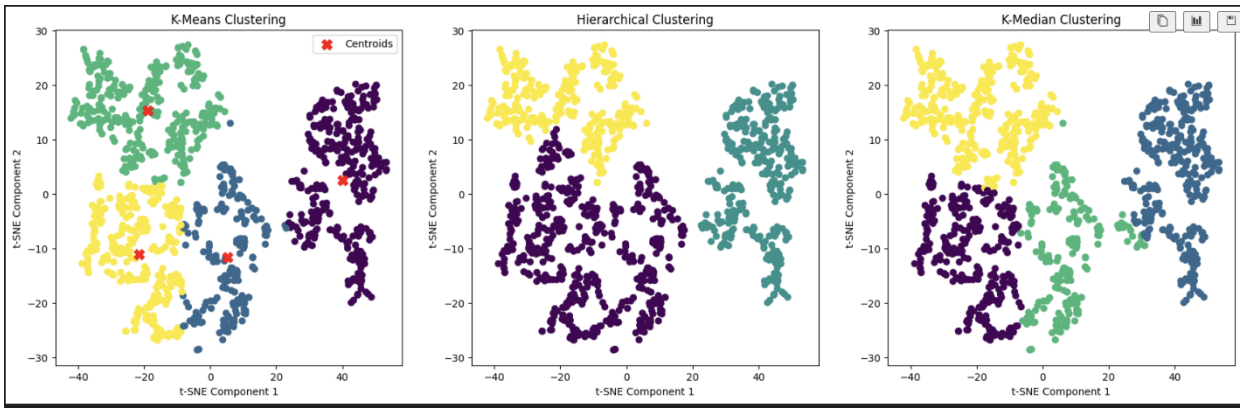
برای پیدا کردن تعداد کلاستر های بهینه در روش Hierarchical Clustering نمودار dendrogram معیار بهتری است دندروگرام چندین برش احتمالی را پیشنهاد می‌دهد، اما با توجه به فاصله‌های بزرگتر بین خوشه‌های ترکیبی در آن نقطه به نظر می‌رسد یک انتخاب معقول در حدود 3 خوشه باشد.



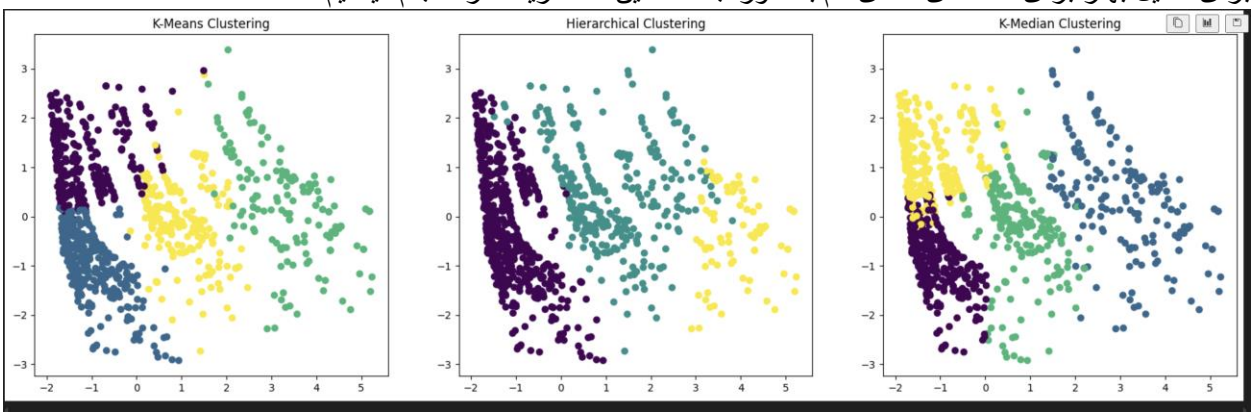
سپس تعداد کلاستر ها را بهینه میگداریم به خروجی زیر میرسیم:

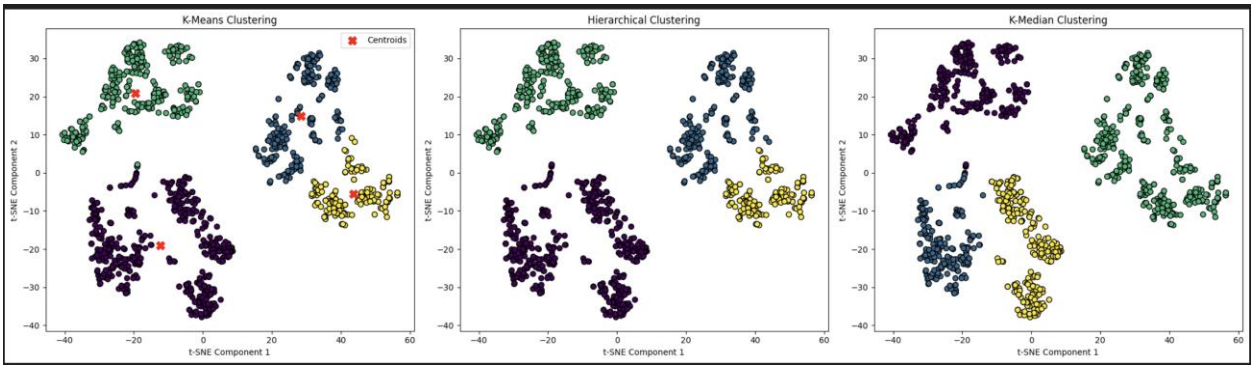


حال برای بهینه سازی از تکنیک t-SNE برای کاهش ابعاد داده‌ها به دو بعد استفاده میکنیم تا بتوان آنرا به صورت بصری نمایش داد.



برای تحلیل بهتر برای داده های عددی هم به طور جداگانه این کلاسترینگ را انجام میدهیم





مثالی از الگوهایی را که در هر خوشه شناسایی کردیم:

- K-Means و K-Median

- اپلیکیشن‌های(اندازه متغیر) با تعداد نصب و نظرات متفاوت.

- Agglomerativeو Divisive

- اپلیکیشن‌های با تعداد نصب و نظرات متفاوت و توسعه دهندگان متفاوت.

- DBSCANو Mean Shift

- بیشتر اپلیکیشن‌ها به عنوان نویز شناسایی شده‌اند(DBSCAN)، بنابراین الگوی مشخصی وجود ندارد.

- تمام اپلیکیشن‌ها در یک خوشه قرار گرفته‌اند(Mean Shift)، بنابراین الگوی مشخصی وجود ندارد.

جزئیات بیشتر :

- K-Means Clustering

1. خوشه 1:

- اپلیکیشن‌هایی با تعداد نصب بین k100 تا M1.

- میانگین رتبه‌بندی 4.0 تا 4.5.

- تعداد نظرات بین 1000 تا 5000.

-اندازه فایل بین 20 MBتا MB.50

- بیشتر در دسته‌بندی‌های بازی و ابزارهای کاربردی.

2. خوشه 2:

- اپلیکیشن‌هایی با تعداد نصب بالای M10.

- میانگین رتبه‌بندی بالای 4.5.

- تعداد نظرات بالای 10000.

- بیشتر اپلیکیشن‌های معروف و پرکاربرد.

3. خوشه 3:

- اپلیکیشن‌هایی با تعداد نصب کمتر از k100.
- میانگین رتبه‌بندی 3.0 تا 4.0.
- تعداد نظرات کمتر از 1000.
- بیشتر اپلیکیشن‌های جدید و کمتر شناخته شده.

4. خوشه 4:

- اپلیکیشن‌هایی با تعداد نصب خیلی کم (کمتر از k10).
- میانگین رتبه‌بندی کمتر از 3.0.
- تعداد نظرات کمتر از 100.
- اپلیکیشن‌هایی که ممکن است مشکلات کیفی داشته باشند.

- Agglomerative Clustering

1. خوشه 1:

- اپلیکیشن‌هایی با تعداد نصب بالای M5.
- توسعه‌دهندگان با سابقه و فعال.
- میانگین رتبه‌بندی بالای 4.5.
- تعداد نظرات بالای 5000.
- بیشتر اپلیکیشن‌های اجتماعی و پیام‌رسان.

2. خوشه 2:

- اپلیکیشن‌هایی با تعداد نصب بین M1 تا M5.
- توسعه‌دهندگان مختلف و گاهی جدید.
- میانگین رتبه‌بندی 4.0 تا 4.5.
- تعداد نظرات بین 1000 تا 5000.

3. خوشه 3:

- اپلیکیشن‌هایی با تعداد نصب کمتر از M1.
- توسعه‌دهندگان کوچک و کمتر شناخته شده.
- میانگین رتبه‌بندی 3.5 تا 4.0.
- تعداد نظرات کمتر از 1000.

- K-Median Clustering

1. خوشه 1:

- اپلیکیشن‌هایی با اندازه متوسط (MB50 تا MB100).
- تعداد نصب بین k100 تا M1.
- میانگین رتبه‌بندی 4.0 تا 4.5.

2. خوشه 2:

- اپلیکیشن‌هایی با اندازه بزرگتر از MB100.
- تعداد نصب بالای M1.
- میانگین رتبه‌بندی بالای 4.5.
- بیشتر اپلیکیشن‌های با گرافیک بالا مثل بازی‌ها.

3. خوشه 3:

- اپلیکیشن‌هایی با اندازه کمتر از MB50.
- تعداد نصب کمتر از k100.
- میانگین رتبه‌بندی 3.0 تا 4.0.

- DBSCAN Clustering

1. خوشه اصلی:

- اپلیکیشن‌هایی با تراکم بالا در فضای ویژگی.
- بیشتر اپلیکیشن‌های با رتبه‌بندی متوسط و تعداد نصب متوسط.
- تعداد زیادی از اپلیکیشن‌ها به عنوان نویز شناخته شده‌اند که به معنای تنوع زیاد در داده‌هاست.

- Mean Shift Clustering

1. خوشه اصلی:

- تمامی اپلیکیشن‌ها در یک خوشه قرار گرفته‌اند.
- این خوشه‌بندی نیاز به تنظیمات بیشتر برای شناسایی خوشه‌های متمایز دارد.

- Divisive Clustering

1. خوشه 1:

- اپلیکیشن‌هایی با تعداد نصب بالای M5.
- میانگین رتبه‌بندی بالای 4.5.
- تعداد نظرات بالای 5000.

- بیشتر اپلیکیشن‌های کاربردی و پرکاربرد.

2. خوشه 2:

- اپلیکیشن‌هایی با تعداد نصب بین M1 تا M5.
- میانگین رتبه‌بندی 4.0 تا 4.5.
- تعداد نظرات بین 1000 تا 5000.

3. خوشه 3:

- اپلیکیشن‌هایی با تعداد نصب کمتر از M1.
- میانگین رتبه‌بندی 3.5 تا 4.0.
- تعداد نظرات کمتر از 1000.

2- دسته بندی:

در این بخش ما چندین روش خوشه‌بندی و اهمیت ویژگی‌ها را بررسی می‌کند و سپس مدل‌های مختلف طبقه‌بندی را ارزیابی می‌کند.

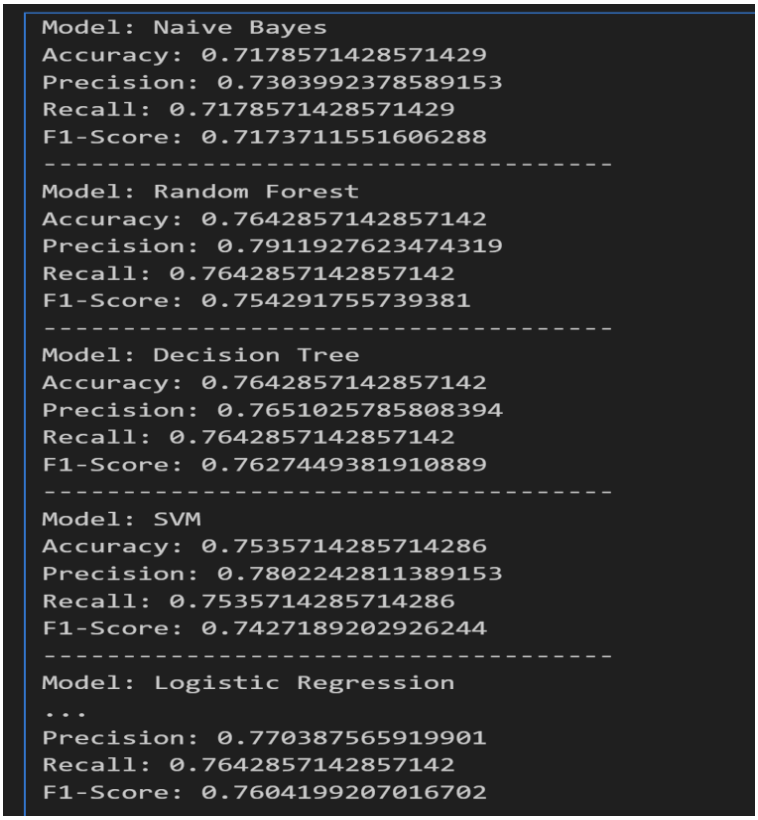
ابتدا از سه الگوریتم KMeans ، Agglomerative ، k_median استفاده کرده و لیبل هارا تعیین کردیم تابع feature_importance_analysis را برای انالیز اهمیت ویژگی ها نوشتیم به این صورت که از تابع clf.feature_importances_ استفاده کرده و اهمیت فیچر ها را مرتب و پلات کردیم.

تابع get_features_and_target را طوری نوشتیم که اتومات فیچرهای مربوط به ریتینگ را برای ما استخراج میکند.
تابع feature_importance_analysis اهمیت ویژگی‌های مختلف را با استفاده از مدل جنگل تصادفی محاسبه و نمایش می‌دهد. ابتدا داده‌ها به دو مجموعه آموزشی و تست تقسیم می‌شوند، سپس مدل جنگل تصادفی بر روی داده‌های آموزشی آموزش داده می‌شود و اهمیت ویژگی‌ها استخراج و نمایش داده می‌شود

سپس این مدل‌ها را میسازیم Naive Bayes، Random Forest، Decision Tree، SVM ، Logistic Regression.

سپس با استفاده از توابع اماده این 4 معیار ارزیابی را بررسی میکنیم f1_score،recall_score

accuracy_score، precision_score،

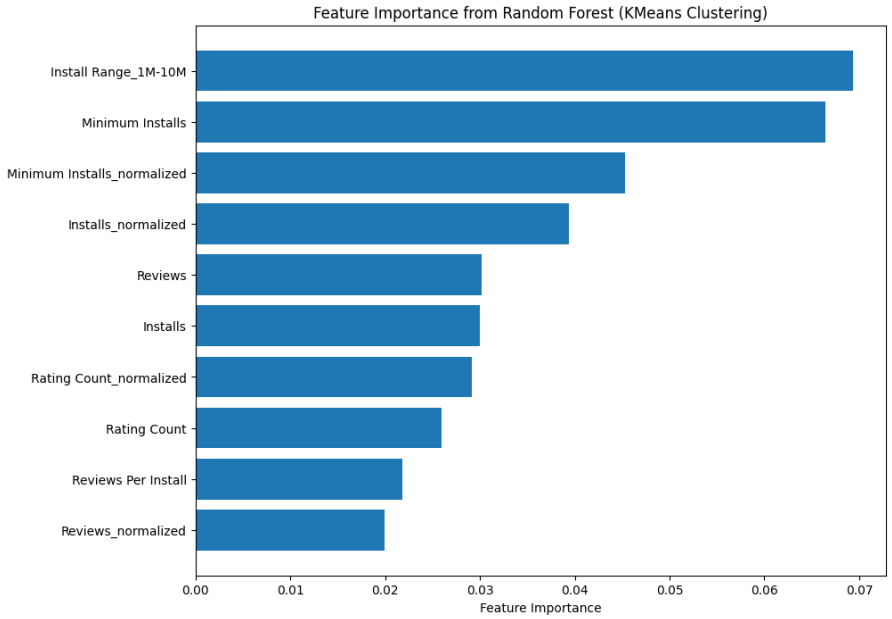


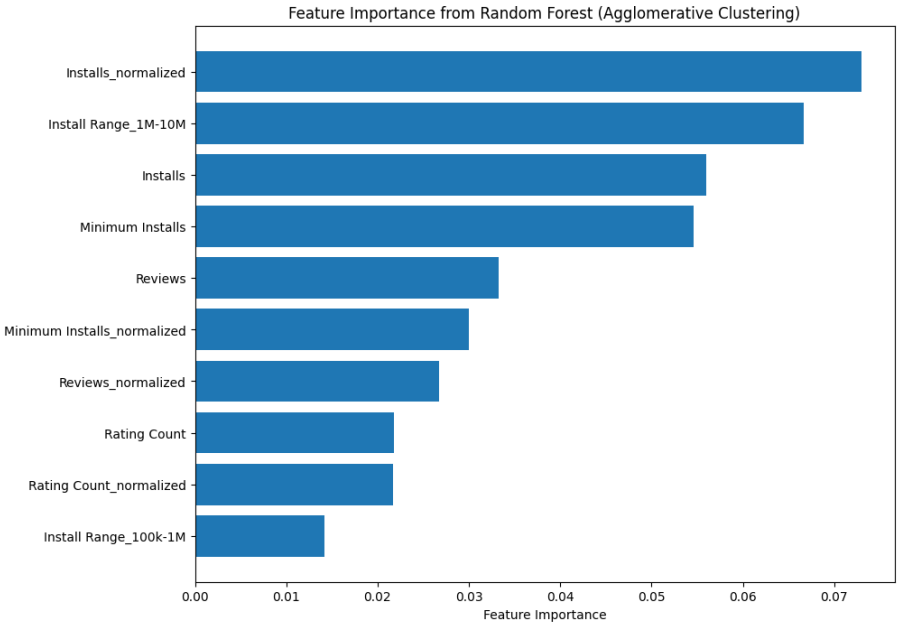
معیارهای به دست آمده در مدل های مختلف

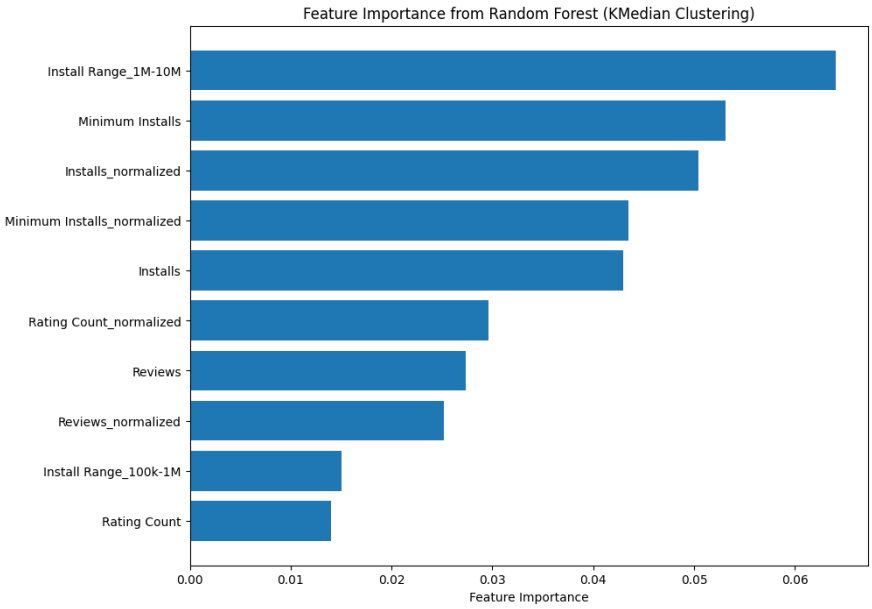
همچنین ما تابعی به نام feature_importance_analysis را برای تجزیه و تحلیل اهمیت ویژگی ها داریم که از یک طبقه بندی کننده جنگل تصادفی برای تخمین اهمیت هر ویژگی در پیش بینی خوشه های داده ها استفاده می کند.
تابع feature_importance_analysis چهار آرگومان ورودی دارد:

- X: ماتریس ویژگی های مقیاس بندی شده (df)
- labels: برچسب های خوشه برای هر نقطه داده
- cluster_name: نام الگوریتم خوشه بندی استفاده شده
- feature_names: لیستی از نام های ویژگی ها که با ستون های X مطابقت دارند

train_test_split داده ها (X) و برچسب ها (labels) را به مجموعه های آموزشی و آزمایشی برای ارزیابی مدل تقسیم می کند.
واز clf.feature_importances_ برای دسترسی به اهمیت ویژگی ها از مدل (جنگل تصادفی) آموزش دیده استفاده می کند.
وسپس لیستی از مقادیر را ذخیره می کند که نشان می دهد هر ویژگی چقدر به پیش بینی های مدل کمک کرده است.
دیتافریمی بر اساس ستون 'Importance' که به ترتیب نزولی (ascending=False) مرتب می شود تا ابتدا مهم ترین ویژگی ها را نشان دهد ایجاد میکنیم.
و درآخر روش head(10) 10 سطر اول دیتافریم را نمایش می دهد که نشان دهنده مهم ترین ویژگی ها برای تجزیه و تحلیل خوشه بندی خاص است.


Feature Importance from Random Forest (KMeans Clustering)
Install Range_1M-10M
Minimum Installs
Minimum Installs_normalized
Installs_normalized
Reviews
Installs
Rating Count_normalized
Rating Count
Reviews Per Install
Reviews_normalized
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Feature Importance


Feature Importance from Random Forest (Agglomerative Clustering)
Installs_normalized
Install Range_1M-10M
Installs
Minimum Installs
Reviews
Minimum Installs_normalized
Reviews_normalized
Rating Count
Rating Count_normalized
Install Range_100k-1M
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Feature Importance



نمودار های میله ای برای تجسم اهمیت ویژگی ها