پروژه درس داده‌کاوی – ترم پاییز 1401 – فاز 4 – دسته شماره 2

سید محمد طاها طباطبایی – ملیکا ذبیحی نیشابوری

**سوال اول : نسبت تعداد کالاهای خریداری شده به بازگشت به انبار به تفکیک سازمان**

**سوال دوم: تحلیل ورود دارایی به خروج دارایی، به تفکیک زمان و معاونت**

**سوال سوم: وضعیت کالاهای مشابه، و پیش بینی قیمت هر کالا متناسب با خوشه آن**

برای پاسخ به این سوال، باید یک مدل خوشه‌بندی پیشنهاد دهیم که دیتای ترین را خوشه‌بندی کند، تا بتوانیم، در گام بعدی، دیتای تست را به وسیله خوشه های موجود، پیش‌بینی کنیم(لیبل بزنیم). در این بخش، چند مدل خوشه بندی، از جمله DBSCAN, K-means, OPTICS, Brich و HDBSCAN تست و نتایج آن ها بررسی شد، و در نهایت سه مدل K-means و DBSCAN و HDBSCAN که به نظر برای داده های ما مناسب تر بود، انتخاب شدند تا نتیجه خوشه بندی روی آن ها به طور جدی تر بررسی شود. در این گزارش، برای هر کدام از این مدل ها یک اسکریپت پایتون نوشته شد و نتایج را بر اساس پیاده‌سازی انجام شده بررسی کردیم. در میانه بررسی اسکریپت، چالش های پیش‌آمده نیز بررسی می‌شوند. با توجه به اینکه بهترین پیاده‌سازی، روی مدل مبتنی بر خوشه‌بندی K-means به‌دست آمد، تنها این مدل با جزئیات شرح داده خواهد شد، اما اسکریپت های دو مدل دیگر نیز در فایل های ارسالی قرار داده شده اند.

**توضیح فنی پیاده‌سازی، متمرکز بر مدل K-means[[1]](#footnote-1)**

برای خوشه‌بندی، در اولین گام نیاز به تشخیص ویژگی های مناسب برای پردازش داده ها بود. برای این کار، تمام تیبل های داده شده، به صورت دستی بررسی شدند. ایده اولیه انتخاب ویژگی ها، خوشه‌بندی بر اساس نوع و قیمت بود. دلیل این انتخاب، این بود که بتوان در مرحله پیش‌بینی نیز، با اتکا به اینکه قیمت در خوشه‌بندی نیز لحاظ شده، و همچنین این اصل بدیهی که کالاهای هم نوع را می‌توان تا حد خوبی، مشابه هم فرض کرد، به یک خوشه بندی با دقت مناسب برسیم. بررسی تیبل ها با این ایده اولیه، منجر به انتخاب سه ویژگی شد. ویژگی BOOKVALUE از تیبل productInstance ، و ویژگی های NAME و VALUE از تیبل products. ویژگی اول، طبق تعریف ارائه شده، ارزش روز یک کالا را بیان می‌کند. ویژگی دوم، یک رشته به زبان فارسی است که اسم یک کالا را نمایش می‌دهد. در مورد این ویژگی لازم به ذکر است که رشته های فارسی ارائه شده در دیتا، بعضا شامل الگوی خاصی در نام گذاری بودند که در بخش های بعدی توضیح بیشتری داده خواهد شد. ویژگی سوم نیز، شامل اعدادی است که متناظر با نوع کالا، بر اساس شباهت کالا ها، مقدار دارند. شاید بتوانید برای تقریب ذهنی، عملکرد این ستون را به عنوان یک بازنمایی(مشابه embedding vector) از رشته فارسی به عدد در نظر بگیرید، به این صورت که کالاهایی با اسم مشابه، اعداد نزدیک به هم در ستون VALUE دارند. ویژگی M\_PRODUCT\_ID نیز از هر دو تیبل، به این دلیل انتخاب شد تا بتوان به عنوان کلید مرحله مرج، برای ادغام دو سری دیتا استفاده شود. با این توضیحات برای انتخاب ویژگی، وارد مرحله پیش پردازش می‌شویم.

با بررسی ویژگی ها NAME و VALUE، یک نکته جلب توجه کرد. مقادیری در ستون NAME وجود داشتند، که به فرمت:

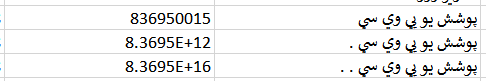
اسم کالا + " . ."

یا به فرمت:

اسم کالا + " ."

بودند.

پس در دیتاست، یک حالت رخ می‌داد، مشابه شکل زیر:



که به ازای یک اسم مشابه، بدون تفاوت در دیگر ستون های رکورد مورد نظر، و فقط وجود یک پسوند شامل نقاط در اسم، تعدادی صفر به انتهای عدد بازنمایی در ستون VALUE، اضافه می‌شد. در اولین گام پیش پردازش دیتا، این صفر های اضافی از مقادیر رکورد های ستون VALUE حذف می‌شوند.

در خطوط 30 تا 66، این تبدیل روی رکورد هایی که به این فرمت هستند، اعمال می‌شود. با توجه به اینکه وجود دو نقطه، تعداد صفر بیشتری اضافه می‌کرد، برای پاکسازی، دو رشته رجیکس متفاوت استفاده کردیم تا بتوانیم براساس تعداد صفر های اضافه شده، تخمین مورد نظر را اعمال کنیم. بر اساس رجیکس مچ شده با رشته و دیتاتایپ تشخیص داشته شده(برخی مقادیر عددی، رشته تشخیص داده می‌شدند که باید به شکل رشته با آنها برخورد می‌شد) تبدیل روی مقادیر انجام می‌شود.

همچنین در حلقه یاد شده، در انتهای حلقه، یکبار دیتاها را به مقدار عددی تبدیل می‌کنیم( با کمک متد to\_numeric). علت این کار، این است که همچنان برخی مقادیر در دیتاست، طول عددی زیادی دارند که باعث می‌شد، این اعداد به عنوان رشته در لایبرری های مورد استفاده پردازش شوند. با این تغییر فرمت، خواستیم تا این اشکال نیز برطرف شود(خطوط 58 تا 63). در خط 66 نیز، یکبار کل ستون به فرمت float64 تبدیل می‌شود، تا از این بابت، دیگر هیچ مشکلی در پردازش مقادیر عددی نداشته باشیم.

گام بعد، ادغام دیتاهای دو تیبل است. با کمک ویژگی ID انتخاب شده، ستون های موجود را با یکدیگر مرج می‌کنیم. متد ادغام، inner در نظر گرفته شده تا فقط اشتراکات با یکدیگر ادغام شوند، و دیتای بلااستفاده جدید تولید نشود.(خط69)

سپس، مقادیر null از دیتا حذف شده.(خط 72)

در خط 74، دو فیچر مورد نیاز برای پردازش در مدل خوشه‌بندی، VALUE و BOOKVALUE به عنوان دیتای نهایی انتخاب شده اند. این دیتاها را برای استفاده بهتر در لایبرری ها، به فرمت numpy تبدیل می‌کنیم.

در خط 80، تابع train\_test\_split از کتابخانه sci-kit برای دسته بندی دیتا به دو بخش train و test استفاده می‌شود. 70 درصد دیتای موجود، به عنوان train و 30 درصد به عنوان test انتخاب می‌شود.

گام بعدی، اسکیل دیتا به رنج [0,1] است(خطوط 84 تا 86). تبدیل MaxAbsScaler نیز امتحان شد، که در اجرا تفاوت خاصی با روش فعلی(MinMaxScaler) نداشت. پس از اسکیل کردن، مرحله پیش‌پردازش داده ها به اتمام می‌رسد.

در این بخش، به توضیح مدل ساخته شده می‌پردازیم. برای خوشه بندی، در مقایسه بین 3 مدل DBSCAN، HDBSCAN و K-means، مدل K-means خروجی بهتری داشت، به همین جهت انتخاب نهایی این مدل بود. چالش اصلی مدل K-means برای خوشه‌بندی، انتخاب تعداد بهینه خوشه هاست، به شکلی که خوشه های با کیفیت مناسبی تولید شود. برای حل این موضوع، با اجرای الگوریتم elbow، تعداد خوشه های مناسب را پیدا کردیم.

الگوریتم elbow، با اجرای k های مختلف و شبیه سازی خوشه‌بندی، سعی می‌کند تا بهترین k را انتخاب کند. معیار انتخاب، بهترین دقت کسب شده، به نسبت زمان اجراست. الگوریتم، خوشه‌بندی با k های مختلف را شبیه سازی می‌کند. سپس دقت خوشه‌بندی های به‌دست آمده را محاسبه می‌کند. نقطه elbow که در واقع، نقطه ای است که تغییرات بهبود دقت خوشه‌بندی، به نسبت زمان صرف شده بسیار کم می‌شود، نقطه مورد نظر ما است. مقدار k این نقطه، به عنوان تعداد خوشه های بهینه انتخاب می‌شود. به طور بدیهی، به نظر می‌رسد که شاید در نمودار، فقط یک نقطه به این شکل وجود داشته باشد، اما به طور تجربی مشخص می‌شود که ممکن است متناسب با توزیع دیتا، چند نقطه به این شکل داشته باشیم. به همین دلیل، باید متد elbow در چند رنج مختلف k اجرا شود، تا احتمال انتخاب k بهینه به طور گلوبال را افزایش دهیم. با توجه به این نکته، ما نیز در ساخت مدل خود، چند بار در رنج های مختلف k، الگوریتم را اجرا کردیم، و در انتها مقدار k=48 انتخاب شد. جدول زیر، نمایش حاصل چندبار اجرای متد elbow و مقادیر k به‌دست آمده، زمان تقریبی اجرا و امتیاز کسب شده است.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | K = 101 | K = 79 | **K = 48** | K = 34 | K = 11 | K = 7 |
| Score | 0.002 | 0.003 | **0.008** | 0.018 | 0.173 | 0.576 |
| Run time | 7.1 | 5.0 | **2.8** | 2.2 | 0.7 | 0.5 |

متد محاسبه امتیاز در این مقایسه، distortion است. محاسبات به این شکل انجام می‌شود که مجموع مربعات فاصله هر نقطه از مرکز انتساب داده شده اش، محاسبه می‌شود. طبعا هرچقدر این مقدار کمتر باشد، یعنی خوشه‌ های متمرکز تری ایجاد شده، پس عدد پایین تر مطلوب تر است.

در مرحله بعد، مدل K-means را با پارامتر k=48 ترین می‌کنیم.

گام نهایی، مرحله تست است. در این بخش به طور مرسوم، دیتای تست را به مدل دادیم و با استفاده از متد predict، لیبل های اختصاص یافته را محاسبه کردیم.

**نتایج:**

برای بررسی دقت مدل، از متد امتیاز دهی Silhouette استفاده کردیم، که یک متد مرسوم در روش های یادگیری بدون ناظر است. این متد برای هر نمونه از دیتا، مطابق فرمول زیر، ضریب امتیاز آن خوشه را محاسبه می‌کند.

که α، فاصله درون خوشه ای برای هر نمونه دیتا و β، فاصله از نزدیک ترین خوشه به جز خوشه ای که نمونه در آن قرار دارد است. عدد نهایی، میانگین ضرایب تمامی نمونه ها است. این عدد، در رنج [-1,1] است. عدد 1 نشان دهنده بهترین حالت خوشه‌بندی، و عدد 1- ، نشان دهنده انتساب نمونه به خوشه اشتباه است. اعداد در محدوده 0 نیز، بیانگر هم پوشانی بین خوشه ها است، که مطلوب نیست.

ضریب Silhouette را برای دیتای train و test محاسبه کردیم. نتایج به شکل زیر است.

|  |  |  |
| --- | --- | --- |
|  | Train | Test |
| Silhouette score | 0.841 | 0.838 |

با توجه به امتیازات، مدل ما به دقت بسیار خوب 84 درصد روی نمونه train و 83 درصد روی نمونه test دست یافت. همچنین به طور رندوم، 10 اندیس از دیتای test تولید می‌شود که در خروجی مقدار لیبل پیش‌بینی شده برای آن‌ها را چاپ می‌کنیم.

همچنین، نتایج ویژال خوشه‌بندی، در 3 فولدر جداگانه، با نام های Train Result، Test Predictions و Selected Points که به ترتیب، نتایج اجرای مدل برای دیتای train، دیتای test و لیبل نقاط منتخب است.

به این ترتیب موفق شدیم تا یک خوشه‌بندی بر اساس نوع کالا و قیمت کالا بسازیم که در ادامه موفق به پیش‌بینی لیبل روی دیتای تست با دقت مناسبی است.

1. پیاده‌سازی در فایل Kmeans.py انجام شده است. [↑](#footnote-ref-1)