دانشگاه صنعتی شریف

دانشکده مهندسی صنایع

گزارش پروژه کارشناسی

عنوان پروژه:

پیشبینی کالاهای سبد خرید بعدی مشتریان فروشگاه مواد تند مصرف به کمک یادگیری ماشین

استاد پروژه:

استاد حسن نایبی

استاد راهنما:

استاد هوشمند

تهیه و گردآوری:

سجاد عابد، تابستان و پاییز ۱۴۰۱

چکیده

نگهداری مشتری در سیستم فروشگاه و جلوگیری از ریزش[1] مشتریان یکی از مسائل مهم فروشگاه‌های آنلاین و حضوری است. روش‌های مختلفی مانند برگزاری کمپین[2]، ارسال کد تخفیف[3]، برگزاری حراج‌های هفتگی، ماهانه یا مناسبتی و... برای این کار وجود دارد. یکی از این روش‌ها پیشبینی نیاز مشتری و یادآوری آن به مشتری و مشتاق کردن او به خرید از طریق پیام تبلیغاتی شخصی یا ارسال کد تخفیف است. این روش بیشتر در فروشگاه‌هایی که کالای تند مصرف[4] (مانند مواد خوراکی، لوازم پخت و پز و لوازم بهداشتی که عموما قیمت پایینی دارند و مرتبا مصرف می‌شوند) به مشتریان خود عرضه می‌کند کاربرد دارد. هدف اصلی این کار بازگرداندن مشتری به سیستم و هدایت آن به سمت ثبت سفارشی جدید است گر چه این کار مزیت‌های دیگری مانند افزودن کالاهای بیشتر به سبد خرید مشتریان هنگام ثبت سفارش آنان نیز دارد. در این مقاله با بررسی داده‌های خرید مشتریان یک فروشگاه آنلاین کالای تند مصرف به کمک یادگیری ماشین[5] و تست الگوریتم‌های مختلف با زبان برنامه نویسی پایتون[6] برای رسیدن به دقت بیشتر برای شناخت رفتار مشتری، سعی داریم الگوریتمی برای پیشبینی کالاهای مورد نیاز مشتری در خرید بعدی بیابیم.

کلمه‌های کلیدی: فروشگاه آنلاین، کالای تندمصرف، ریزش مشتری، یادگیری ماشین، رفتار مشتری، پیشبینی سبد خرید بعدی، پایتون، برنامه نویسی

[1] churn
[2] campaign
[3] Voucher code
[4] Fast-moving consumer goods
[5] Machine learning
[6] Python

فهرست

۱. مقدمه ۱

۲. مقدمه‌ای بر یادگیری ماشین کلاسیک ۲

۲.۱. شاخه بندی یادگیری ماشین کلاسیک ۲

۲.۲. مفهوم گرادیان کاهشی در یادگیری ماشین ۳

۲.۳. روش‌های ترکیبی یادگیری ماشین کلاسیک ۶

۳. بررسی پایگاه داده ۷

۳.۱. آیتم‌ها ۷

۳.۲. کتگوری‌ها ۷

۳.۳. کلاس‌ها ۷

۳.۴. سفارشات ۸

۳.۵. مصورسازی داده‌ها ۸

۴. پیش پردازش اولیه‌ی داده‌ها ۱۷

۴.۱. پیش پردازش داده‌ها ۱۷

۵. رویه حل مسئله ۱۹

۵.۱. مشخص کردن نحوه‌ی حل مسئله ۱۹

۵.۲. ساخت ویژگی‌های مربوط به مشتری ۲۰

۵.۳. ساخت ویژگی‌های مربوط به محصول ۲۱

۵.۴. ساخت ویژگی‌های مربوط به کتگوری ۲۱

۵.۵. ساخت ویژگی‌های مشتری-محصول ۲۲

۵.۶. ساخت ویژگی‌های مشتری-کتگوری ۲۲

۵.۷. ساخت ویژگی‌های مربوط به زمان ۲۲

۶. آماده سازی داده‌ها برای آموزش مدل یادگیری ماشین ۲۴

۶.۱. مشخص کردن لیبل و جدا کردن دیتای تست و آموزش ۲۴

۶.۲. نرمال سازی داده‌ها ۲۵

۶.۳. متعادل سازی داده‌ها ۲۶

۷. نتایج گزارش شده توسط مدل‌های یادگیری ماشین و بررسی نتایج آن‌ها ۲۸
۷.۱. گزارش کلاس‌بندی ۲۸
۷.۲. اهمیت ویژگی ۳۰
۸. بررسی و آماده سازی داده‌ها برای آموزش مدل بر پایه‌ی کتگوری ۳۴
۸.۱. چرا بر اساس کتگوری ۳۴
۸.۲. تغییرات نسبت به حالت قبل ۳۵
۸.۳. فیچرهای جدید ۳۵
۹. بررسی نتایج مدل بر پایه‌ی کتگوری و مقایسه‌ی نتایج آن ۳۷
۹.۱. گزارش کلاس‌بندی ۳۷
۹.۲. اهمیت ویژگی ۳۷
۹.۳. ویژگی‌های نسبی ۴۲
۱۰. جمع بندی ۴۶
۱۱. منابع ۴۸

۱. مقدمه

در سال‌های اخیر دسترسی عمومی به اینترنت، سطح خدمت‌دهی فروشگاه‌های کالا و خدمات در بستر اینترنت و همچنین سهولت استفاده از آن از طریق هر دستگاهی افزایش چشمگیری داشته است. این عامل در کنار افزایش مشغله‌ی مردم و تمایل به انجام ساده‌تر کارهایی که ارزش افزوده‌ای ندارند (مانند حرکت به سمت فروشگاه و قدم زدن بین قفسه‌های فروشگاه و حمل کیسه‌های خریداری شده به سمت خانه) باعث شده است که بسیاری از مردم به جای انجام شخصی این کارها مایل باشند از پلتفرم‌هایی که این خدمات را انجام می‌دهند استفاده کنند. گرچه همچنان میل افراد برای خرید برخی اجناس گران قیمت و خرید کالاهایی که به ندرت خرید می‌کنند به این سمت است که به صورت حضوری خرید نمایند اما برای کالاهایی که به صورت روزانه استفاده می‌شوند بیشتر به این سمت مایلند که حتی‌الامکان فعالیت‌های مذکور که فاقد ارزش افزوده اند را انجام ندهند. از این رو بخش قابل توجهی از مردم خرید مواد تند مصرف مانند مواد غذایی، بهداشتی و... را به صورت منظم از فروشگاه‌های آنلاین تهیه می‌کنند. همچنین در سال‌های اخیر با افزایش سطح کیفیت سیستم‌های اطلاعاتی، فروشگاه‌های آنلاین می‌توانند با تحلیل بر روی داده‌های بسیاری که از مشتریان خود دارند، رفتارهای آنان را بررسی و شناسایی کنند.

در این بین با توجه به افزایش علاقه‌ی مردم به خرید آنلاین این نوع کالاها، پلتفرم‌ها و فروشگاه‌هایی که این نوع خدمت را برای مشتریان انجام می‌دهند افزایش می‌یابند. از طرفی فروشگاه‌هایی که فقط به صورت حضوری فروش دارند سعی می‌کنند با به کار بردن ترفندهایی مشتریان جدید جذب کنند. به این ترتیب نگه داشتن مشتری در سیستم فروشگاه و جلوگیری از منتقل شدن او به فروشگاه آنلاین یا حضوری دیگر، از مسائلی است که همواره باید مورد توجه صاحبان این نوع کسب و کار باشد، زیرا که جذب مشتری همواره با هزینه‌ی بسیار بالاتری نسبت به نگهداری مشتری همراه است و تا زمانی که اعتماد مشتری به فروشگاه جلب نشده باشد، سود زیادی از او عاید فروشگاه نخواهد شد. توجه به مشتری و ارسال پیام‌های شخصی سازی شده برای هر مشتری یکی از روش‌هایی است که در کنار آنچه در چکیده‌ی مقاله به آن اشاره شد به مشتری حس رضایت بخش، اطمینان و نزدیکی به فروشگاه می‌دهد و منجر به حفظ مشتری در طولانی مدت می‌شود. حال در این مقاله سعی می‌کنیم به کمک این روش امکان حفظ مشتری را بررسی کنیم.

همانطور که بالاتر اشاره شد به دلیل خرید دوره‌ای و منظم بخشی از مشریان این نوع فروشگاه‌ها، غالبا داده‌های زیادی از این مشتریان در سیستم اطلاعاتی فروشگاه موجود است. به کمک این داده‌ها و الگوریتم‌های یادگیری ماشین می‌توان رفتار مشتریان را پیشبینی و برای هر فرد به صورت شخصی سازی شده پیام‌های تبلیغاتی، یادآوری و یا تخفیف ارسال کرد. روشی که در این مقاله به آن می‌پردازیم، پیشبینی سبد خرید بعدی مشتری به کمک خریدهای قبلی مشتری است. رویه‌ی کار به این صورت است که در زمانی که انتظار داریم مشتری برای خرید مجدد اقدام کند، کالاهایی که بر اساس داده‌های قبلی به نظر می‌رسند که باید در سبد خرید جدید مشتری باشند را پیشبینی کنیم و با اعلام یادآوری به مشتری و یا اعمال تخفیف شخصی برای آن مشتری بر روی آن کالاها، کششی بر روی مشتری به سمت ثبت خرید مجدد آن ایجاد کنیم. در این مقاله از دیتاست فروشگاهی که به مشتریان خود کالاهای تند مصرف عرضه می‌کند استفاده می‌کنیم که در ادامه به توضیح آن دیتاست می‌پردازیم.

۲. مقدمه‌ای بر یادگیری ماشین کلاسیک

۲.۱. شاخه بندی یادگیری ماشین کلاسیک

برای یادگیری ماشین تعاریف مختلفی ارائه می‌شود. برای مثال از آن تحت عنوان "یک روش تحلیل داده که به صورت خودکار کار ساخت مدل را انجام می‌دهد "[1] یا روشی که "به نرم افزارها اجازه می‌دهد که دقت پیشبینی خود را افزایش دهند بدون آن که به صورت اختصاصی برای آن کار برنامه ریزی شده باشند"[2] یا "با تمرکز بر روی داده‌ها و الگوریتم‌ها، برای تقلید از مدلی که انسان یادمی‌گیرد قصد افزایش دقت آن را دارد"[3] . تمام تعاریف بالا با توجه به کاربرد مورد استفاده‌ی ما از یادگیری ماشین می‌تواند صحیح باشد؛ اما به طور کلی یادگیری ماشین زیرمجموعه‌ای از هوش مصنوعی[۷] است که در به خاطر سپاری و انجام محاسبات سخت و پیچیده‌ی ریاضی و آماری که انسان در به دست آوردن الگوهای مختلف با آن‌ها دست و پنجه نرم می‌کند، به او کمک می‌کند. زیرشاخه‌ای از یادگیری ماشین که امروزه بیش از ۵۰ درصد پروژه‌های یادگیری ماشین را شامل می‌شود یادگیری ماشین کلاسیک[۸] نام دارد. البته در این پروژه از روش‌های ترکیبی[۹] نیز استفاده می‌کنیم اما تمرکز بر روی روش‌های کلاسیک است. غالبا شرکت‌های بزرگ تکنولوژی از روش‌های یادگیری عمیق[۱۰] و شبکه‌های عصبی[۱۱] برای پروژه‌های خود استفاده می‌کنند زیرا یک افزایش کوچک در دقت مدل می‌تواند برای آن‌ها میلیون‌ها و حتی میلیاردها سوددهی داشته باشد اما امروزه با توجه به زمان اجرای آن و سخت افزاری که نیاز دارد، استفاده‌ی آن در صنایع و پروژه‌های کوچکتر منطقی نیست.

یادگیری ماشین کلاسیک به دو زیرشاخه‌ی با ناظر[۱۲] و بدون ناظر[۱۳] تقسیم می‌شود. در یادگیری ماشین با ناظر ما به ازای هر موجودیت[۱۴] یک مقدار پاسخ یا به اصطلاح لیبل[۱۵] داریم. هدف ما در این بخش پیشبینی کردن آن مقدار پاسخ برای موجودیت‌هایی است که لیبل آن را نمی‌دانیم. همین زیرشاخه نیز به دو بخش دیگر تقسیم می‌شود؛ رگرسیون[۱۶] و کلاس‌بندی[۱۷] که در رگرسیون لیبل یک مقدار پیوسته و عددی دارد در حالی که در کلاس‌بندی لیبل چند مقدار مشخص و محدود دارد و ما مشخص می‌کنیم که یک موجودیت با ویژگی[۱۸]هایی که دارد، متعلق به کدام کلاس است.

در یادگیری بدون ناظر موجودیت‌های ما لیبل مشخصی ندارند. این شاخه را می‌توان به سه دسته‌ی خوشه‌بندی[۱۹]، کاهش ابعاد[۲۰] و قوانین وابستگی[۲۱] تقسیم کرد. در خوشه‌بندی هدف ما این است که موجودیت‌هایی که ویژگی‌های مشابه با یکدیگر را دارند در یک دسته قرار دهیم و موجودیت‌های خود را گروه بندی کنیم.

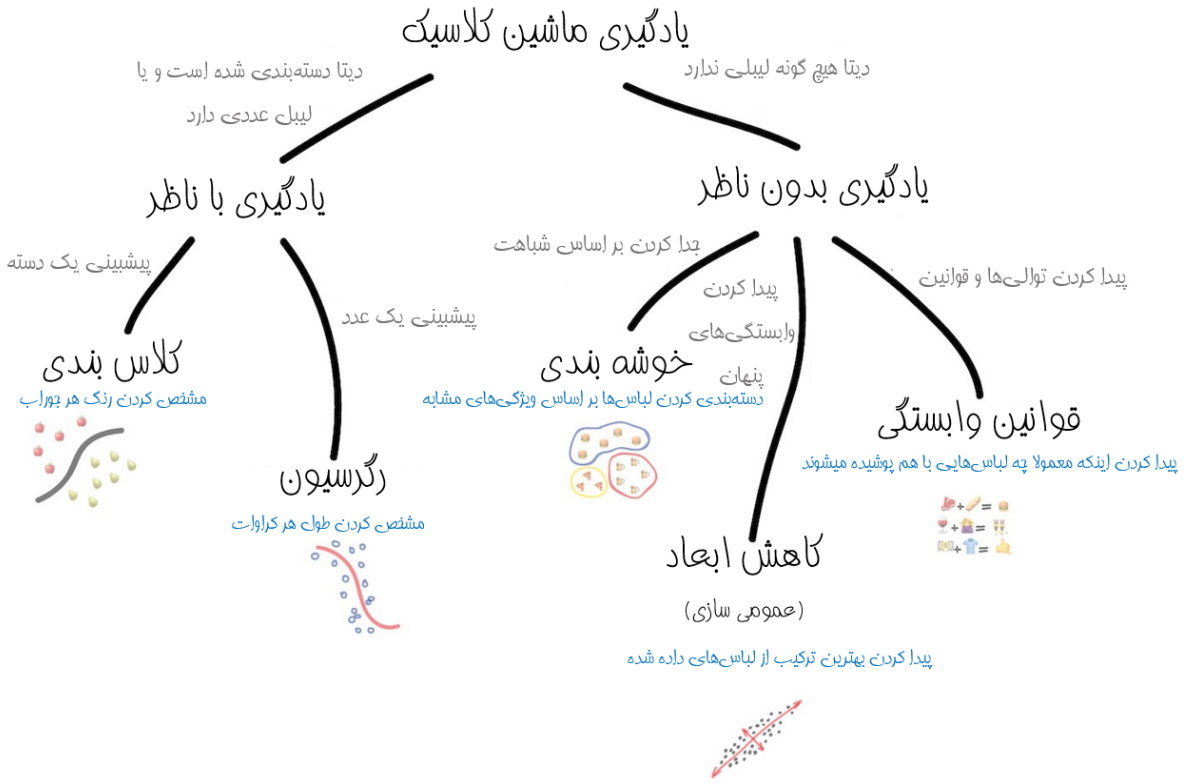
در کاهش ابعاد ما به دنبال این هستیم که بدون اینکه اطلاعات زیادی را از دست بدهیم، تعداد ویژگی‌ها را کاهش دهیم. هدف این بخش این است که برای یادگیری الگو روی دیتاست زمان کمتری صرف شود. همچنین می‌توان از آن برای کم کردن حجم دیتای

---

[7] Artificial Intelligence (AI)
[8] Classical Machine Learning
[9] Ensemble Methods
[10] Deep Learning
[11] Neural Networks
[12] Supervised
[13] Unsupervised
[14] Entity
[15] Label
[16] Regression
[17] Classification
[18] Attribute
[19] Clustering
[20] Dimension Reduction
[21] Association Rules

موجود در دیتابیس استفاده کرد. بهترین کاربرد این روش آن است که بتوان یک ویژگی را با کمترین خطای ممکن به کمک دو یا چند ستون دیگر به دست آورد. البته برای کاهش ابعاد می‌توان از لیبل داده‌های لیبل‌دار برای پاسخ بهتر استفاده کرد اما به طور کلی این بخش زیردسته‌ی یادگیری ماشین بدون ناظر است.

در قوانین وابستگی به دنبال کشف الگوها و قوانینی در موجودیت‌ها هستیم. به عنوان مثال اینکه غالبا در یک سبد خرید اگر محصول x باشد محصول y نیز هست. یا اگر در یک سبد محصول w و z باشند، در سبد بعدی محصول p خواهد بود.



۲.۲. مفهوم گرادیان کاهشی در یادگیری ماشین

همانطور که در بخش قبل گفته شد، دلیل استفاده از "ماشین" این است که کامپیوتر محاسبات دشوار و تکراری را برای حجم زیادی از داده‌ها انجام می‌دهد و همچنین برای ساختن مدل‌های پیچیده، نیاز به ذخیره سازی و استفاده‌ی سریع از داده‌های قبلی را دارد که این کار در دیتاست‌های بزرگ برای انسان بسیار سخت و تقریبا غیر ممکن است و در صورت امکان سرعت آن به مراتب کمتر از کامپیوتر است.

الگوریتمی که کامپیوتر غالبا برای یادگیری از آن استفاده می‌کند الگوریتم گرادیان کاهشی[۲۲] یا گرادیان نزولی است. گرادیان نزولی یک روش تکرار شونده[۲۳] برای یافتن کمینه‌ی محلی[۲۴] یک تابع است که در آن با حرکت به سمت سمت منفیِ شیب تابع، کمینه‌ی محلی آن را پیدا می‌کنیم. باید توجه داشته باشیم که این الگوریتم یک الگوریتم تکرار شونده است و مقدار بهینه‌ی متغیر موردنظر یک باره بدست نمی‌آید و به تدریج به سمت نقطه‌ی بهینه حرکت می‌کنیم.

[22] Gradient Descent
[23] Iterative
[24] Local Minimum

برای درک بهتر ابتدا از رگرسیون خطی تک متغیره برای درک شهودی این مفهوم استفاده می‌کنیم. در رگرسیون تابع هزینه‌ای که غالبا استفاده می‌شود تابع مجموع مربعات خطا[۲۵] است که تابعی درجه دو و محدب است.

$$J = \sum_{i=1}^{m} (y_i - f(x_i))^2$$

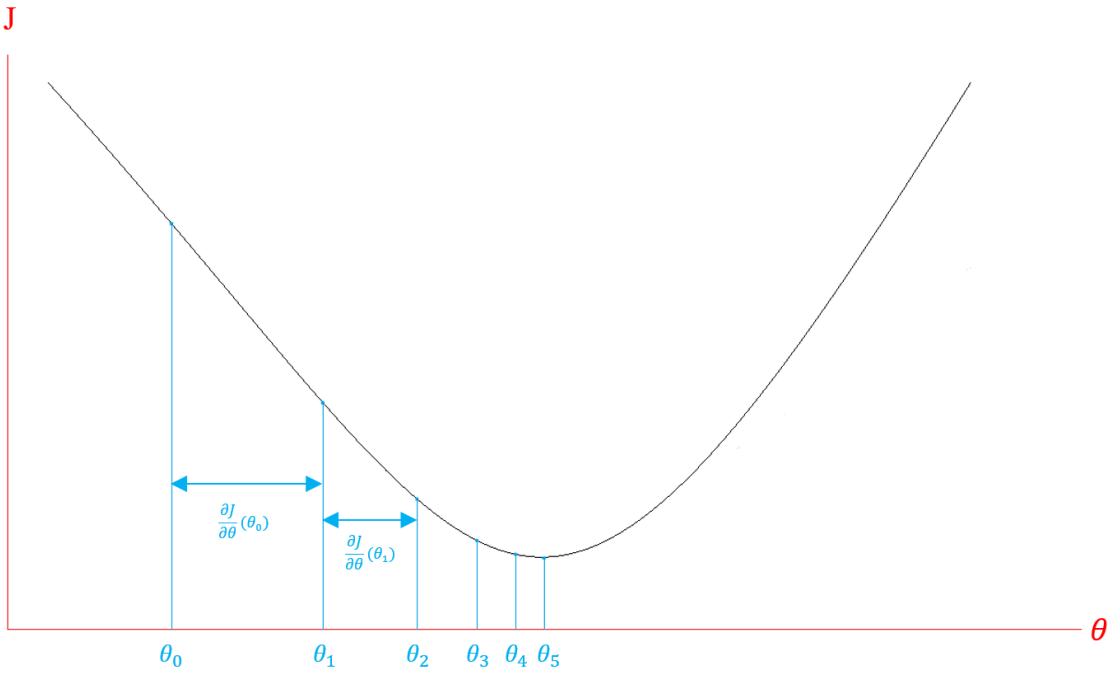
حال برای تعیین یک θ خاص که پارامتری از تابع f است، ابتدا یک مقدار تصادفی برای آن و یک مقدار ثابت برای α (نرخ آموزش) در نظر می‌گیریم. سپس θ مرحله‌ی بعدی را به کمک رابطه‌ی زیر به دست می‌آوریم:

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial J}{\partial \theta} (\theta_t)$$

الگوریتم تا جایی ادامه می‌یابد که مقدار $\frac{\partial J}{\partial \theta}$ برابر با صفر شود و یا از مقدار مشخصی کوچکتر شود.

در تعیین نرخ آموزش باید به این نکته توجه کنیم که اگر نرخ آموزش را بزرگ انتخاب کنیم، ممکن است هیچگاه الگوریتم ما همگرا نشود و هیچ وقت به شرط توقفی که بالاتر به آن اشاره شد نرسیم. اگر این نرخ را خیلی کوچک نیز انتخاب کنیم سرعت آموزش ما بسیار کند می‌شود. غالبا نرخ آموزش عددی در اردر یک هزارم یا یک صدم انتخاب می‌شود.

مشخص است با توجه به درجه دو بودن تابع هزینه، هر چه به نقطه‌ی بهینه‌ی تابع نزدیک‌تر می‌شویم، سرعت و مقدار پیشرفت ما در هر تکرار کمتر می‌شود. دلیل آن این است که اندازه‌ی شیب در نقاط نزدیک‌تر به نقطه‌ی بهینه کوچکتر است.



حال اگر تابعی که ما به عنوان تابع پیشبینی در نظر میگیریم یک چندجمله‌ای[۲۶] با درجه‌ای بیش از یک باشد، با توجه به اینکه ما می‌خواهیم مقدار θ ها را مشخص کنیم، تفاوتی در کار ما ایجاد نمی‌کند زیرا در هر حالت تابع هزینه نسبت به θ ها خطی است.

[25] Sum Squared Error (SSE)
[26] Polynomial

به عنوان مثال به تابع زیر نگاه کنید:

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2$$

تابع زیر یک تابع خطی بر حسب θ هاست. بنابراین اگر تابع هزینه (J) را به وسیله‌ی این تابع پیشبینی بنویسیم، همچنان یک منحنی درجه دو بر حسب هر θ خواهیم داشت که یک نقطه‌ی مینیموم موضعی دارد که نقطه‌ی مینیموم مطلق آن است.

اگر بخواهیم از راه معادله نرمال به پاسخ برسیم خواهیم داشت:

$$\theta X = Y$$

که در آن θ مجهول است و ماتریس X یک ماتریس $m \times n$ است که m بردار x ما را نشان می‌دهد. اگر X مربعی باشد می‌توانیم θ را به شکل زیر بدست بیاوریم:

$$\theta = YX^{-1}$$

اما در اغلب موارد X مربعی نیست و باید از روش زیر که گرفتن شبه معکوس[۲۷] ماتریس نام دارد استفاده کرد:

$$\theta X = Y$$
$$\theta X X^T = Y X^T$$
$$\theta X X^T (X X^T)^{-1} = Y X^T (X X^T)^{-1}$$
$$\theta = Y X^T (X X^T)^{-1}$$

تفاوت روش گرادیان کاهشی با معادله‌ی نرمال در این است که در گرادیان کاهشی پس از چندین تکرار به پاسخی با دقت بالا می‌رسد اما در معادله‌ی نرمال پاسخ نهایی و دقیق یک باره به دست می‌آید اما در طرف مقابل محاسبه‌ی معکوس یک ماتریس بزرگ بسیار زمان بر و پیچیده است و سرعت کار در گرادیان کاهشی بسیار کمتر خواهد بود. همچنین اگر ستون‌های ماتریس وابسته‌ی خطی باشند، در روش گرادیان کاهشی رسیدن به پاسخ سخت‌تر می‌شود اما در روش معادله‌ی نرمال، محاسبه‌ی معکوس ماتریس چون وابسته به محاسبه‌ی دترمینان است که تعریف نشده خواهد بود، امکان پذیر نمی‌باشد.

در این بخش برای توضیح روش گرادیان کاهشی، محاسبات مربوط به روش رگرسیون تک و چند متغیره مرور شد. هدف از این پروژه بررسی ریاضیات روش‌های یادگیری ماشین نیست و به موارد بیشتری پرداخته نشده است.

[27] Pseudo-Inverse

۲.۳. روش‌های ترکیبی یادگیری ماشین کلاسیک

هر روش ممکن است در رسیدن به پاسخ پایانی به دلیل وجود داده‌های پرت خطا داشته باشد و یا در قسمت‌هایی پیشبینی ضعیفی داشته باشد. اما اگر از تعداد زیادی روش برای رسیدن به پاسخ و تصمیم نهایی استفاده کنیم، هر خطای هر روش به کمک روش‌های دیگر که در آن قسمت عملکرد خوبی دارند پوشیده می‌شود. سه راه برای ترکیب روش‌های مختلف یادگیری ماشین وجود دارد که عبارت اند از:

- **پشته سازی**[۲۸]

خروجی‌های روش‌های مختلف در مرحله‌ی بعدی به یک الگوریتم دیگر داده می‌شوند تا آن الگوریتم با توجه به آن‌ها یک تصمیم نهایی را برای پیشبینی اعلام کند. توجه کنید که در این روش روش‌های مختلف، بر روی تمام دیتاست اجرا می‌شوند. بنابراین استفاده از روش‌های یکسان (با توجه به یکسان بودن دیتاست) منطقی نیست.

- **بگینگ**[۲۹]

بگینگ که از Bootstrap AGGregatING برداشته شده است، شامل این است که زیر مجموعه‌هایی از دیتاست به روش نمونه‌برداری تصادفی با جایگذاری از دیتاست را انتخاب کرده و با یک الگوریتم ثابت مورد آموزش قرار می‌دهد و سپس تصمیم نهایی به وسیله‌ی رای گیری ساده از الگوریتم‌های مختلف به دست می‌آید. اگر از روش درخت تصمیم[۳۰] در این رویکرد استفاده شود به آن جنگل تصادفی[۳۱] گفته می‌شود.

- **تقویت**[۳۲]

در این رویکرد ابتدا یک الگوریتم بر روی دیتاست آموزش می‌بینند. در تکرار اول وزن تمام دیتاها یکسان است. در تکرار بعد، دیتاهایی که اشتباه پیشبینی شده‌اند، وزن بیشتری می‌گیرند. در واقع وزن قبلی دیتاها در یک وزن جدید ضرب می‌شوند که برای دیتاهایی که در الگوریتم آخر اشتباه پیشبینی شده‌اند، این وزن ثابت جدید بزرگ‌تر از دیتاهایی است که به درستی پیشبینی شده‌اند. به همین ترتیب در تکرار سوم دیتاهایی که در الگوریتم دوم اشتباه پیشبینی شده اند وزنشان در عدد بزرگتری نسبت به باقی دیتاها ضرب می‌شود. این کار ادامه پیدا می‌کند تا زمانی که دقت الگوریتم دیگر افزایش چشم‌گیری نداشته باشد و یا به دقت مورد نظر رسیده باشد. هدف از نسبت دادن وزن بیشتر به داده‌های غلط این است که در تکرار جدید اشتباه پیشبینی کردن آن‌ها تابع هزینه را بیشتر از حالت معمولی افزایش دهد، بنابراین الگوریتم سعی می‌کند تا جلوی اشتباه مجدد در پیشبینی این داده‌ها را بگیرد. با این کار نقاط ضعف الگوریتم **تقویت** می‌شوند. روش‌های ADABoost و XGBoost از این رویکرد هستند.

---

[28] Stacking
[29] Bagging
[30] Decision Tree
[31] Random Forest
[32] Boosting

۳. بررسی پایگاه داده

دیتاستی که در اختیار داریم از ۴ جدول کلاس، کتگوری، آیتم و سفارشات تشکیل شده است. جدول اصلی که با آن مدل یادگیری ماشین خود را پیش می‌بریم جدول سفارشات است. باقی جداول برای شناخت بیشتر و بهتر نسبت به داده‌ها در اختیار ما قرار داده شده اند. ابتدا به بررسی جدول آیتم‌ها می‌پردازیم.

۳.۱. آیتم‌ها

در جدول آیتم‌ها هر ردیف مختص یک محصول خاص است. این جدول از ۶ ستون و حدود ۵۳ هزار ردیف تشکیل شده است که به ترتیب این ستون‌ها عبارتند از:

- iid: این ستون شناسه‌ی هر محصول (item ID) را نشان می‌دهد که کلید اصلی این جدول است.
- item_name: این ستون نام هر محصول را که به فارسی نوشته شده است بیان می‌کند.
- classid: این ستون که یک کلید خارجی برای جدول کلاس‌هاست، نشان می‌دهد که این محصول به کدام کلاس اختصاص دارد.
- catid: این ستون هم یک کلید خارجی برای جدول کتگوری است و شناسه‌ی کتگوری‌ای که محصول متعلق به آن است را نمایش می‌دهد.
- brandid: شناسه‌ی برند محصول را نشان می‌دهد که کلید خارجی جدول برندهاست اما در این دیتاست آن جدول در اختیار قرار نگرفته است.
- brand_name: نام برند هر محصول را نشان می‌دهد. (محصولاتی که برند ندارند نام برند آن‌ها "فله" ذکر شده است.)

توضیح مهم در مورد کلاس و کتگوری:

> آیتم مشخصا به یک محصول خاص با بارکد خاص که تولید کننده، مزه و وزن مخصوص به خود را دارد در صورتی که کتگوری نوع آن محصول را مشخص می‌کند و این مورد را می‌گوید که محصول در کدام دسته قرار میگیرد در حالی که کلاس یک دسته بندی کلی تر از یک محصول است و برای دسته بندی داخل سایت از آن استفاده می‌شود که به طور کلی ممکن است هر کلاس چندین کتگوری را در بربگیرد. به عنوان مثال "ماست میوه‌ای آلوئه‌ورا و میوه‌های جنگلی ۱۲۵ گرمی کاله" یک محصول خاص (آیتم) است که به کتگوری "ماست‌های طعم دار" تعلق دارد و کلاس آن "ماست" است. حال کلاس هر محصول نیز ممکن است به یک کلاس بزرگتر متعلق باشد که در بررسی جدول مربوطه به آن اشاره می‌کنیم.

۳.۲. کتگوری‌ها

در جدول کتگوری‌ها ۲ ستون و ۲۰۶۴ ردیف داریم که این دو ستون عبارت اند از:

- catid: این ستون شناسه‌ی هر کتگوری (category ID) را نشان می‌دهد که کلید اصلی این جدول است.
- cat_name: این ستون نام هر کتگوری را که به فارسی نوشته شده است بیان می‌کند.

۳.۳. کلاس‌ها

در جدول کلاس‌ها سه ستون و ۱۹۸ ردیف داریم که این ستون‌ها عبارت اند از:

- classid: این ستون شناسه‌ی هر کلاس (class ID) را نشان می‌دهد که کلید اصلی این جدول است.

- class_name: این ستون نام هر کلاس را که به فارسی نوشته شده است بیان می‌کند.
- primaryparentid: این ستون شناسه‌ی کلاس بزرگتری که هر کلاس به آن متعلق است را نشان می‌دهد. در مثالی که ذکر شد، ماست به کلاس بزرگتر "لبنیات و تخم مرغ" تعلق دارد. همچنین ستون‌هایی که در کلاس بزرگتری جای نمی‌گیرند، ستون primaryparentid آن‌ها برابر با ۱ (که مربوط به ردیفی با نام "root" است) قرار داده شده است.

۳.۴. سفارشات

در جدول سفارشات ۱۱ ستون و حدود ۲.۵ میلیون ردیف داریم که هر ردیف مربوط به خرید یک محصول خاص توسط یک مشتری در یک خرید خاص است. ستون‌های این جدول عبارت اند از:

- bid: شناسه‌ی سبد خرید را نشان می‌دهد. ممکن است در یک خرید چندین کالا باشد و به همین دلیل ممکن است این ستون در چند ردیف یکسان باشد اما برای هیچ دو نفر مجزا و هیچ دو خریدی از یک نفر که در دو زمان متفاوت انجام شده اند یکسان نیست.
- cid: شناسه‌ی مشتری‌ای که این خرید را ثبت کرده است.
- checkoutdate: تاریخ و ساعت خرید را نشان میدهد
- classid: مشخص می‌کند کالا خریداری شده مربوط به کدام کلاس است.
- catid: مشخص میکند محصول به کدام دسته بندی تعلق دارد.
- iid: شناسه‌ی محصول را نشان می‌دهد.
- quantity: این ستون تعداد محصول خریداری شده توسط مشتری در آن خرید را نشان می‌دهد.
- price: این ستون قیمت واحد آن محصول را در زمان خرید مشتری نشان می‌دهد.
- segmentationlabel: لیبل مشتریان را نشان می‌دهد. مشتریان بیش از ۳ خرید داشته باشند با توجه به رفتار و میزان خرید و فاصله‌ی بین خریدها به یکی از دسته‌های champion, gonechampion, loyal, goneloyal , potential, gonepo ,soso, goneso, zombie, goneZ تقسیم می‌شوند.
- days: این ستون فاصله‌ی تاریخ خرید تا روز ساخته شدن دیتاست که ۱۶ جولای ۲۰۲۱ است را نشان می‌دهد.
- marketid: با توجه به اینکه خریدها از دو مرکز فروشگاه‌های زنجیره‌ای و فروشگاه‌های میوه و تره‌بار صورت می‌گیرد، خریدهایی که از مرکز تره‌بار باشند عدد ۱ و خریدهایی که از هایپرمارکت‌های زنجیره‌ای باشند عدد ۰ دارند.

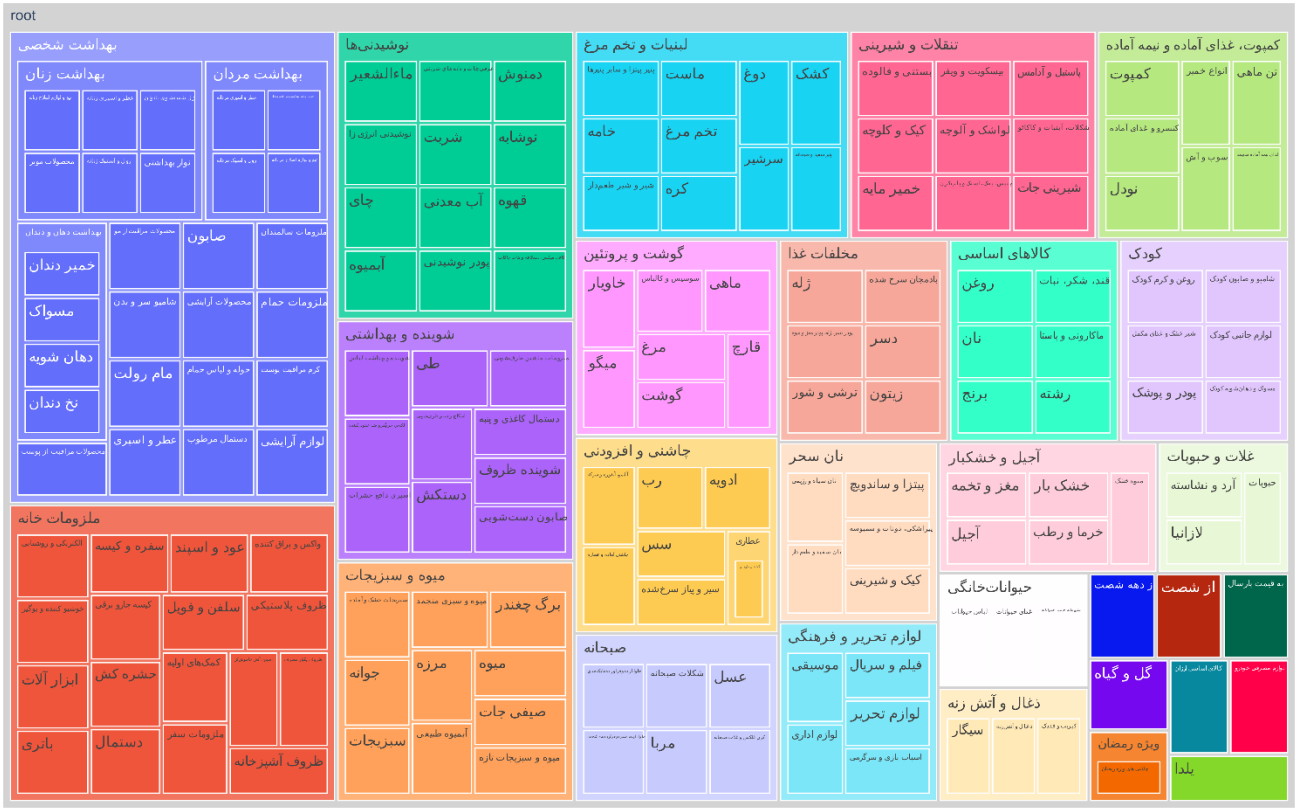
۳.۵. مصورسازی داده‌ها

حال به کمک نمودارها سعی می‌کنیم شناخت بهتری نسبت به داده‌ها پیدا کنیم.

نقشه‌ی درختی[۳۳] کلاس‌ها و زیرکلاس‌ها به شکل زیر است. در این نمودار اندازه‌ی هر زیر کلاس از قاعده‌ی خاصی پیروی نمی‌کند و اندازه‌ی کلاس‌ها به تعداد زیرکلاس‌های وابسته به آن بستگی دارد که طبق آنچه در نمودار دیده می‌شود، بیشترین زیرکلاس مربوط به کلاس بهداشت شخصی و ملزومات خانه است.

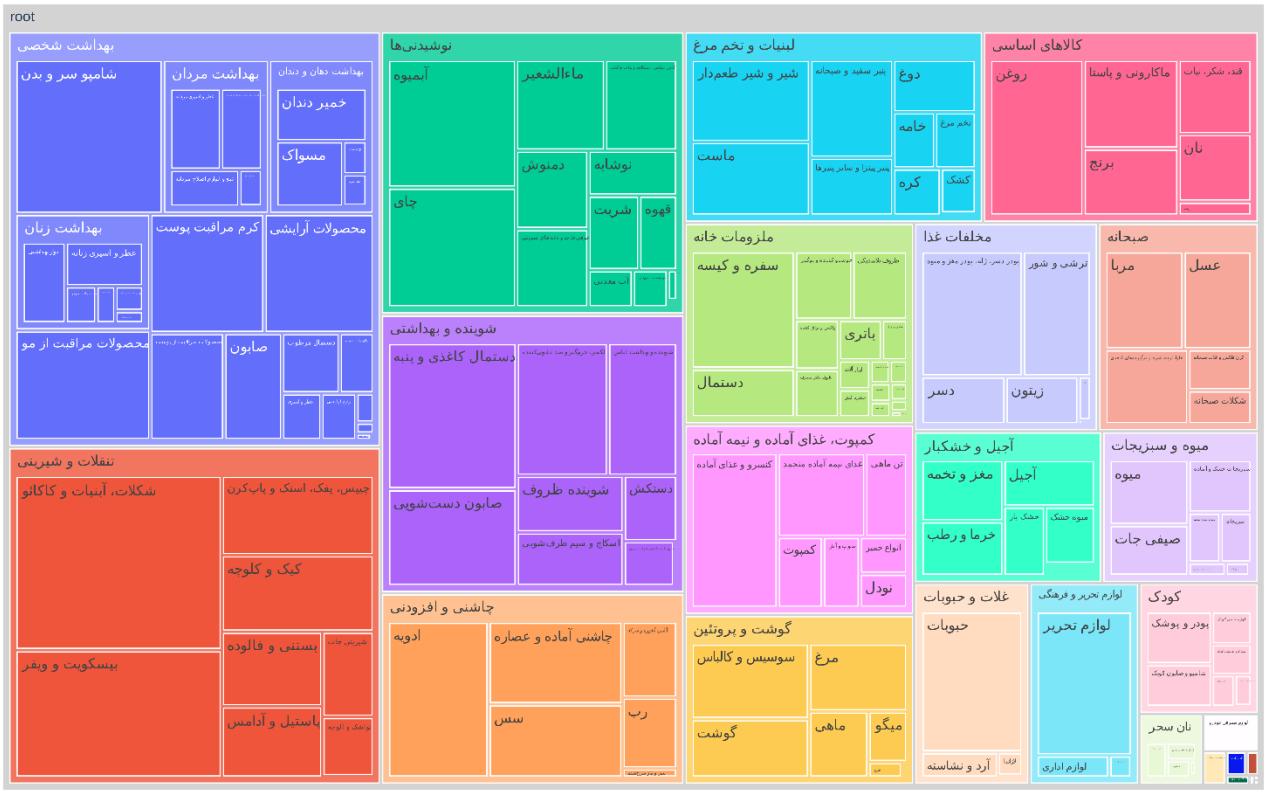
[33] Tree map

Classes and subclasses treemap



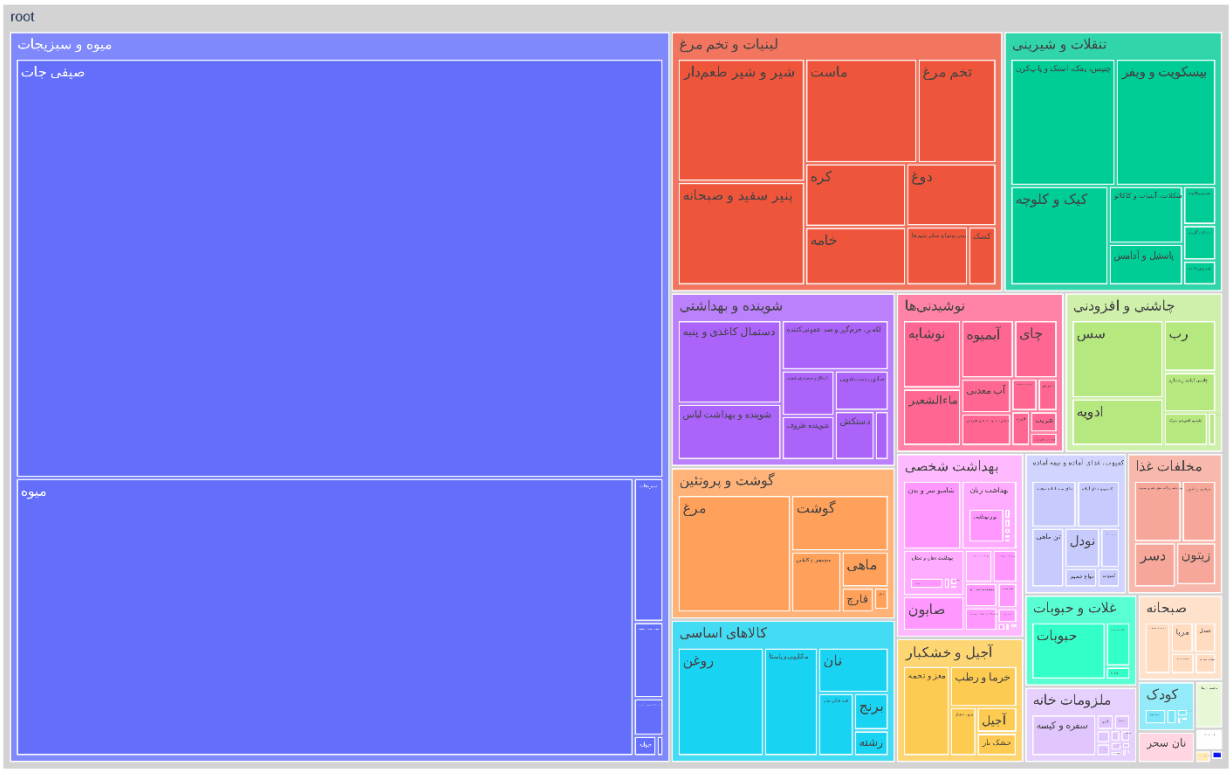
نمودار بعدی نیز یک نقشه‌ی درختی است که در آن اندازه‌ی هر زیرکلاس به تعداد آیتم‌های متنوع موجود در آن زیر کلاس بستگی دارد. طبق این نقشه می‌توان دریافت که متنوع ترین کالاها در کلاس بهداشت شخصی، تنقلات و شیرینی و نوشیدنی‌ها قرار دارند. همچنین بیشترین تنوع یک زیرکلاس مربوط به زیر کلاس شکلات،آبنبات و کاکائو، شامپو سر و بدن، بیسکوئیت و ویفر و دستمال کاغذی و ویفر است. مشخص است که کالاهایی که در این زیردسته‌ها قرار دارند در انواع برندها، مزه‌ها و اندازه‌ها تولید میشوند بنابراین آیتم‌های مختلف زیادی در هر زیر کلاس از آن‌ها وجود دارد.

Classes and subclasses treemap, sized by items' variety

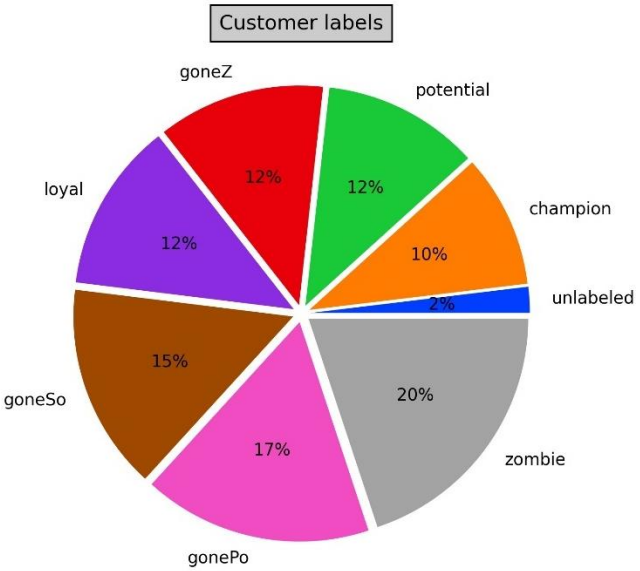


حال اگر نمودار بالا را با توجه به میزان خرید مشتریان رسم کنیم می‌بینیم که تقریبا نیمی از سفارش مشتریان را صیفی‌جات و میوه تشکیل می‌دهد. این نشان می‌دهد که تنوع زیاد در این دیتاست ارتباطی با علاقه‌ی مشتریان به خرید آن محصول ندارد. محصولاتی مانند میوه و سبزی‌جات که به طور مرتب مورد نیاز مردم است حجم زیادی از فروش را به خود اختصاص می‌دهد. پس از آن لبنیات و تخم و مرغ در رتبه‌ی بعدی هستند که با سرعت کمتری خرید می‌شوند و پس از آن تنقلات و شیرینی و شوینده‌های بهداشتی بیشترین میزان خرید را دارند.

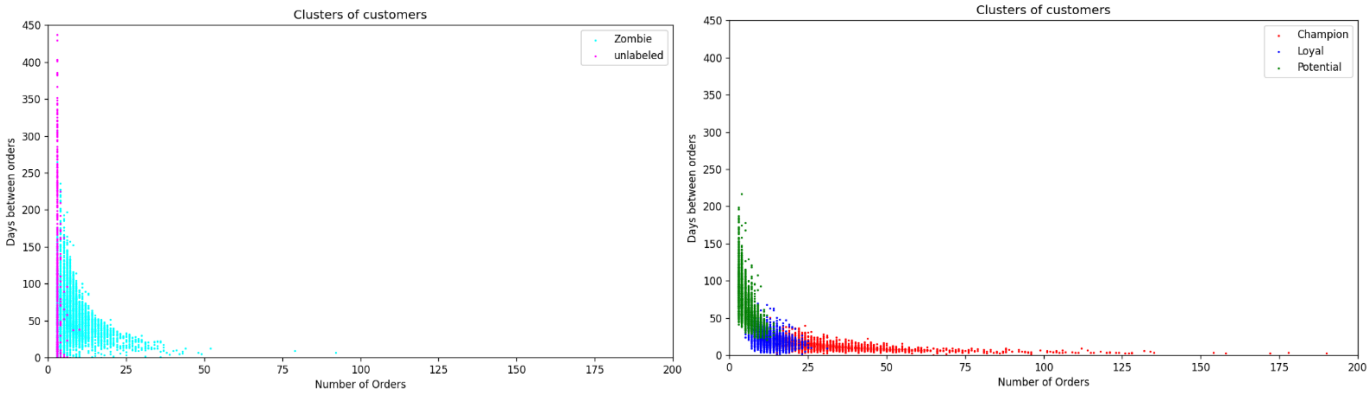
Classes and subclasses treemap,sized by number of orders



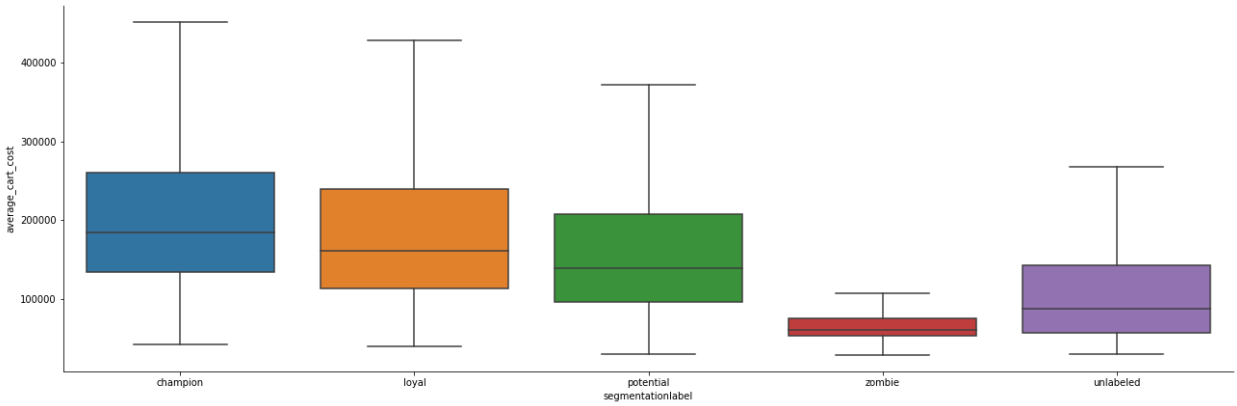
در نمودارهای بعدی سعی داریم رفتار و مشخصات مشتریان را شناسایی کنیم. در نمودار دایره‌ای زیر توزیع برچسب مشتریان را نشان می‌دهیم که بر حسب آن تعداد کمی از مشتریان بدون برچسب هستند. قسمت زیادی از مشتریان زامبی هستند که این برچسب به مشتریانی که سود زیادی برای فروشگاه ندارند و غالبا برای بهره بردن از تخفیف از فروشگاه خرید می‌کنند قرار دارند و قسمت کمتری از مشتریان برچسب "champion" یا "loyal" دارند که به این معنی است که آنان خرید زیاد و مرتب از فروشگاه دارند و سود زیادی نسیب فروشگاه می‌کنند اما به طور کلی نسبت برچسب مشتریان به طور زیادی با یکدیگر تفاوت ندارد.



برای بررسی بهتر لیبل مشتریان، نمودار توزیع میانگین فاصله‌ی بین دو خرید و همچنین تعداد خرید هر مشتری در نمودار نقطه‌ای[۳۴] زیر نمایش داده شدند. برای اینکه لیبل‌های مختلف با یکدیگر تداخل دارند و بتوان آن‌ها را به طور متمایز نشان داد، لیبل‌ها را در دو نمودار متفاوت نشان داده اند.



در ادامه نیز نمودار جعبه‌ای[۳۵] هزینه‌ای که هر لیبل به ازای هر سبد خرید پرداخت می‌کند را نشان می‌دهد. هر چه قیمت سبد بیشتر باشد، سودی که عاید فروشگاه می‌شود بیشتر است. دلیل آن این است که هر سبد هزینه‌ای تقریبا ثابت بسته بندی و ارسال دارد که اگر قیمت کل سبد بیشتر باشد، سود بیشتری به ازای مقدار کمی هزینه‌ی بسته بندی و ارسال به فروشگاه می‌رسد. همچنین اگر یک مشتری برای بهره بردن از تخفیف اقدام به خرید از فروشگاه بکند، معمولا خرید را با کمترین هزینه انجام می‌دهند تا از بیشترین درصد تخفیف بهره ببرند.

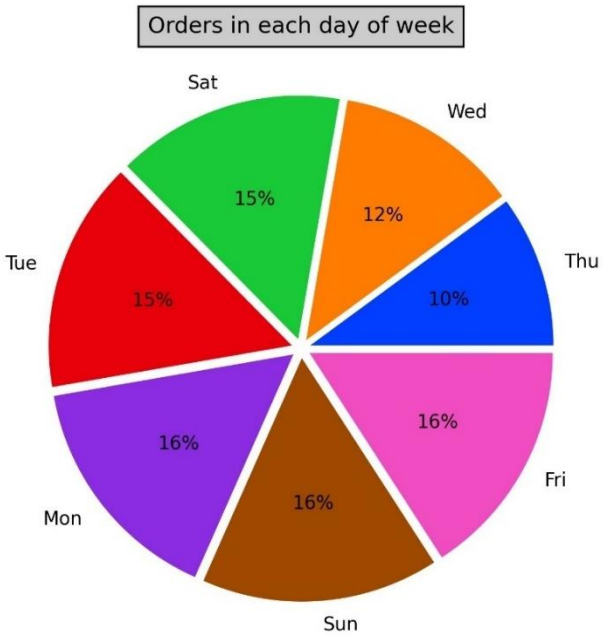


با توجه به نمودار بالا می‌توان به صورت بهتری تفاوت لیبل‌های متفاوت را درک کرد. Champion ها خریدهای بیشتری نسبت به باقی مشتریان انجام داده‌اند و فاصله‌ی بین خریدهای آن‌ها کم است. همچنین میانگین قیمت سبد آن‌ها بیشتر از سایر گروه‌هاست. Loyal ها به نسبت champion ها تعداد خرید کمتری داشته اند اما فاصله‌ی بین خریدهای آنان نیز کم است. مشتریان potential آنهایی اند که تعداد خرید کمی انجام داده‌اند و فاصله‌ی بین خریدهایشان زیاد است اما سبدهای گران قیمتی خریداری می‌کنند و به نسبت برای فروشگاه سود آورند. در ادامه مشتریان zombie بعضا تعداد خرید کم و بعضا تعداد خرید زیاد داشته اند و در مورد فاصله‌ی بین دو خرید نیز اینگونه است، برخی با فاصله‌ی کم و برخی با فاصله‌ی زیاد. آن چه باعث تفاوت این مشتریان با باقی

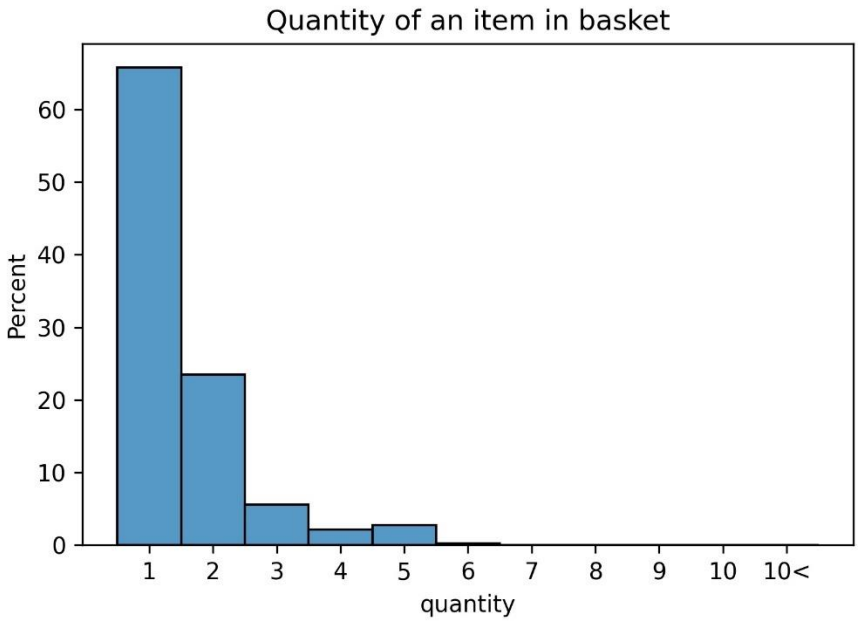
[34] Scatter plot
[35] Box plot

مشتریان است، قیمت سبد خرید آن‌هاست که با فاصله‌ی زیادی نسبت به باقی گروه‌ها، ارزان‌ترین سبدها را دارند. در انتها نیز مشتریانی که لیبل ندارد مشخص اند که لیبل نداشتن آن‌ها به دلیل تعداد کم خریدشان و همچنین فاصله‌ی زیاد از آخرین خریدشان است. همچنین این مشتریان چون تازه به فروشگاه پیوسته اند سبدهای نسبتا ارزان قیمتی دارند. به طور کلی عوامل دیگری مانند واریانس زمان بین دو خرید و استفاده از کد تخفیف برای لیبل بندی مشتریان استفاده شده است اما در این جا سعی شده است که به طور شهودی تفاوت رفتار گروه‌های مختلف مشتریان از یکدیگر نمایش داده شود.

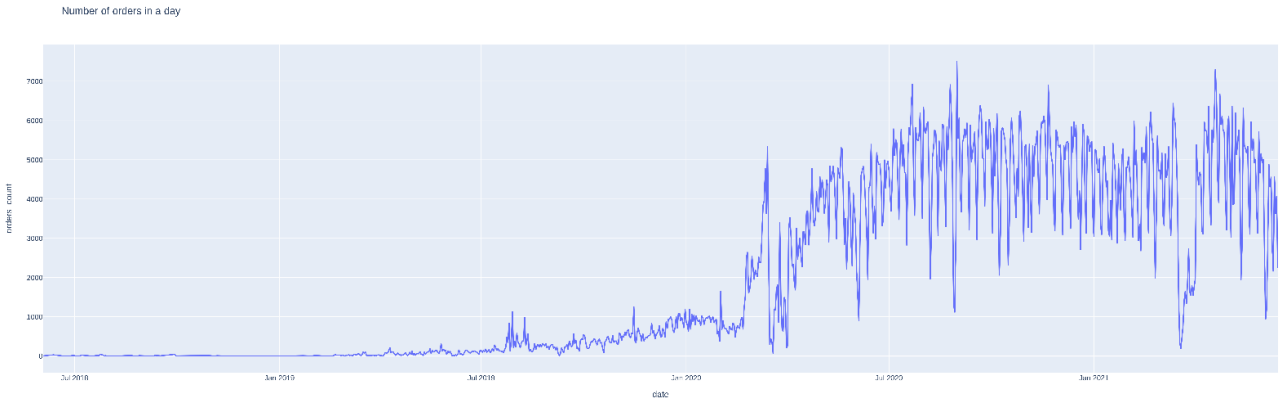
در نمودار دایره‌ای بعدی نسبت خرید مشتریان در روزهای مختلف به نمایش گذاشته می‌شود که نشان می‌دهد به جز روزهای چهارشنبه و پنجشنبه که نسبت کمتری از خریدها را به خود اختصاص می‌دهند، باقی روزها تقریبا نسبت یکسانی از خریدها را شامل می‌شوند.



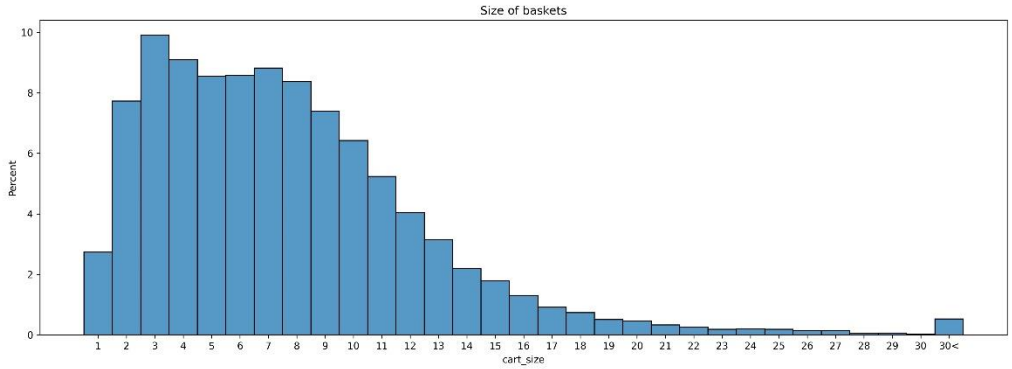
نمودار هیستوگرام زیر نشان می‌دهد که وقتی یک کالا در یک سبد قرار دارد، چه تعداد و یا چند کیلو از آن (در صورتی که محصول فله باشد) خریداری می‌شود. با توجه به نمودار زیر حدود ۶۵ درصد محصولات وقتی خریداری می‌شوند تنها یک واحد از آنها در سبد قرار می‌گیرد. حدود ۲۳ درصد موارد ۲ واحد/کیلویی هستند و تقریبا مواردی که بیش از ۵ کیلو یا ۵ واحد در یک سبد خریداری شده اند ناچیز هستند.



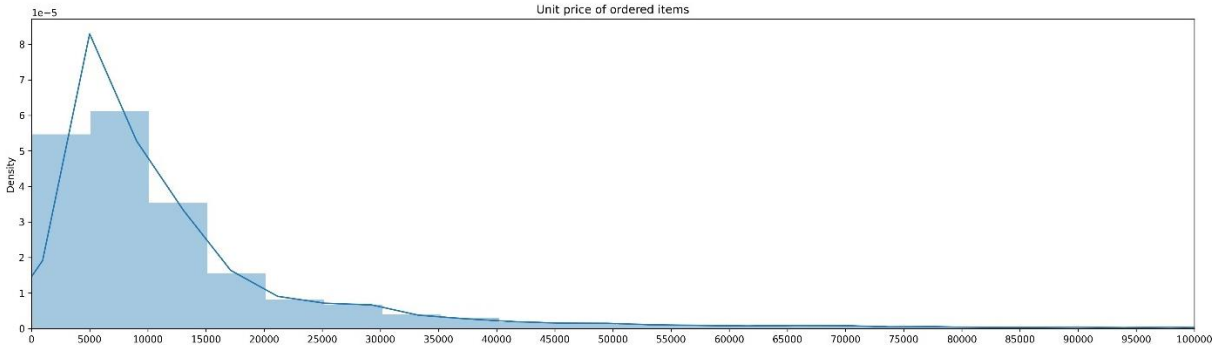
نمودار سری زمانی زیر تعداد خریدها در هر روز را نشان می‌دهد. طبق این نمودار تقریبا مشخص می‌شود که تعداد خریدها تقریبا پس از هر ۵ روز با کاهش مواجه می‌شود که به نظر مربوط به همان کاهش خرید در روز چهارشنبه و پنجشنبه است. همچنین با تغییراتی که در فروشگاه صورت گرفته است، تقریبا در ۱۶ ماه گذشته تعداد خریدها افزایش چشمگیری نسبت به قبل داشته است. همچنین در انتها تعداد خریدها به حدود ۳۰۰ برای روزهای چهارشنبه و پنجشنبه و حدود ۵۰۰ برای باقی روزها رسیده است.



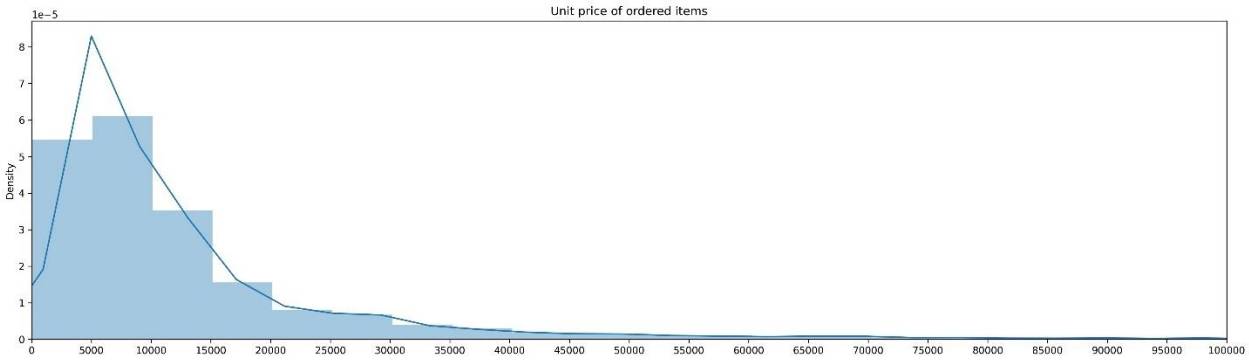
نمودار هیستوگرام زیر تعداد محصولات یکتای موجود در هر سبد را نشان می‌دهد. بر اساس آن در غالب سبدها ۲ تا ۹ محصول یکتا وجود دارد و بعد از آن فراوانی اندازه‌ی سبد با شیب قابل ملاحظه‌ای کاهش می‌یابد. اما این نکته نیز حائز اهمیت است که حدودا یک درصد از سبدها بیش از ۳۰ محصول یکتا در خود دارند.



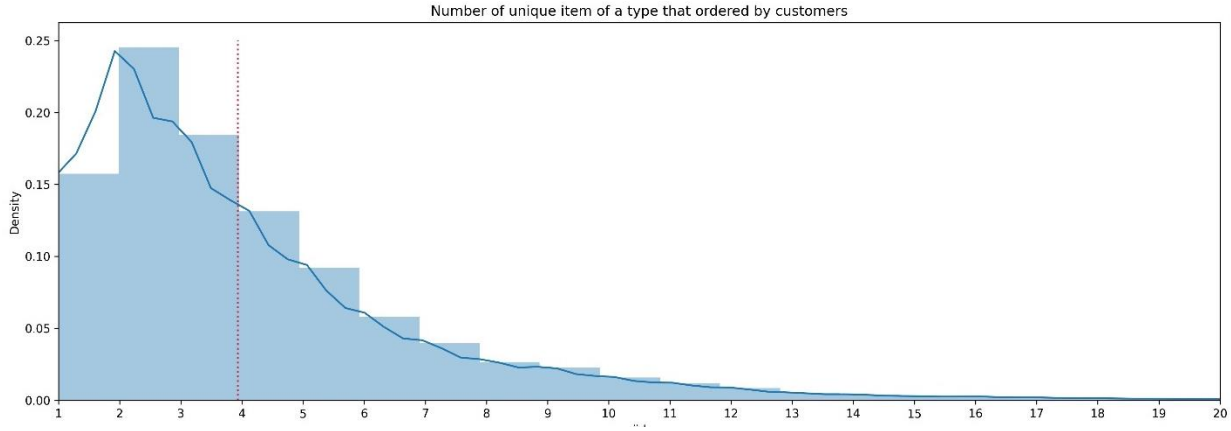
نمودار زیر فراوانی خرید محصولات بر اساس قیمت آن‌ها نشان می‌دهد. بیشتر کالاهایی که خریداری می‌شوند، زیر ده هزار تومان قیمت داشتند. (قیمت‌ها برای سال‌های ۹۷ تا ابتدای ۱۴۰۰ است) و به نسبت کالاهای بسیار کمی قیمت بالای ۲۰ هزارتومان (برای هر واحد یا هر کیلو) دارند.



نمودار بعدی نشان می‌دهد معمولا یک خرید توسط یک مشتری چند روز بعد از خرید قبلی انجام می‌شود. با توجه به این هیستوگرام، محتمل‌ترین حالت این است که تا یک هفته پس از خرید، مشتری مجددا برای خرید به فروشگاه مراجعه کند و پس از آن احتمال مراجعه‌ی مجدد مشتری به فروشگاه کاهش می‌یابد. تقریبا پس از هفته‌ی سوم احتمال کمی دارد که مجددا به فروشگاه باز گردد. بنابراین بهترین زمان برای بازگرداندن مشتری به فروشگاه در هفته‌ی اول است و اگر بیش از ۳ هفته از مراجعه‌ی قبلی مشتری به فروشگاه بگذرد، بازگرداندن مشتری به فروشگاه کار سختی خواهد بود. در واقع هدف از این پروژه این است که افرادی که فروشگاه را ترک می‌کنند و یا با فواصل بسیار زیاد خرید می‌کنند، به طور منظم و با فاصله‌ی کم اقدام به خرید کنند. حالت بهینه برای فروشگاه این است که در نمودار زیر تمام تجمع در دو ستون اول جمع شده باشد و هیچ کدام از مشتریان فروشگاه را ترک نکند.



در نمودار زیر که تنها داده‌ی خرید کالاهایی از یک مشتری در آن آمده است که آن مشتری حداقل ۱۰ مرتبه از کتگوری آن کالا خریده باشد، مشخص است که مشتریان، از یک کتگوری با شرایط ذکر شده، چند محصول متفاوت خریده‌اند. با توجه به این نمودار به طور میانگین هر مشتری از هر کتگوری حدودا ۴ محصول متفاوت خریده است.



۴. پیش پردازش اولیه‌ی داده‌ها

برای پیشبرد این مسئله ابتدا باید به این نکته توجه کنیم که برای تشخیص نیاز مشتری در این روش قصد پیشبینی خرید کالایی که مشتری در سبدهای قبلی سابقه‌ی خریدن آن‌ها را داشته است داریم. بنابراین تنها کالاهایی که مشتری سابقه‌ی سفارش دادن آن‌ها را دارد به عنوان کالاهای سبد خرید بعدی پیشنهاد می‌شوند و قصد دادن پیشنهاد جدید به مشتری نداریم. همچنین با توجه به اینکه هدف ما از این کار پیشبینی تمام کالاهای ممکن نیست و تنها قصد یادآوری خرید به او داریم، فقط بر روی کالاهایی که در خرید آخر مشتری خریداری شده اند را برای بررسی در نظر میگیریم.

برای این امر مراحل کار به شکل زیر خواهد بود که در ادامه به آن‌ها می‌پردازیم:

۱. پیش پردازش داده‌ها
۲. مشخص کردن نحوه‌ی حل مسئله
۳. ساخت ویژگی‌های مربوط به مشتری
۴. ساخت ویژگی‌های مربوط به محصول
۵. ساخت ویژگی‌های مربوط به کتگوری
۶. ساخت ویژگی‌های مشتری-محصول
۷. ساخت ویژگی‌های مشتری-کتگوری
۸. جداکردن خرید آخر هر مشتری از باقی داده‌ها جهت تست داده‌ها

در ادامه توضیحی مختصر در مورد هر بخش داده می‌شود و سپس در بخش‌های بعدی کدهای مربوط به هر بخش توضیح داده خواهد شد.

۴.۱. پیش پردازش داده‌ها

در ابتدا داده‌ها و نوع آنان و همچنین داده‌های از دست رفته‌ی هر جدول را بررسی می‌کنیم. این نکته حائز اهمیت است که داده‌های خالی به این دلیل در دیتاست وجود دارند که در ابتدای شروع به کار سیستم برخی اطلاعات به طور صحیحی ثبت نمی‌شدند بنابراین برخی از ردیف‌هایی که به نسبت قدیمی‌تر هستند ممکن است دیتای از دست رفته داشته باشند.

- در جدول کلاس‌ها ۱۰ ردیف از ستون primaryparentid سلول خالی دارند که از بین ردیف‌هایی که زیر دسته‌ی کلاس بزرگتری نیستند (primaryparentid آن‌ها ۱ است) کلاس مناسب را برای ردیف‌های خالی انتخاب می‌کنیم و به این ترتیب مشکل این ستون حل می‌شود.
- در جدول تایپ‌ها تنها یکی از ردیف‌ها یک ردیف خالی دارد که اهمیت زیادی ندارد.
- در جدول آیتم‌ها در ستون کتگوری و کلاس و برند دیتای خالی داریم که این نیز بیشتر مربوط به کالاهای قدیمی است که اکنون در سایت موجود نیستند و نمایش داده نمی‌شوند.
- جدول سفارشات مهم‌ترین جدول ماست و باید به طور کامل پیش پردازش داده را بر روی آن انجام دهیم و آن را برای مدل یادگیری ماشین خود آماده کنیم. ستون کلاس، کتگوری و لیبل مشتریان بیشترین دیتای از دست رفته را دارد. همانطور که گفته مشتریانی سه خرید یا کمتر داشته اند لیبل ندارند به همین دلیل دیتای خالی در ستون segmentationlabel زیاد است که برای پر کردن این بخش لیبل جدیدی تحت عنوان unlabeled برای مشتریانی که کمتر از ۴ خرید داشته‌اند تعریف می‌کنیم. با حذف کردن باقی ستون‌هایی که داده‌ی خالی دارند، تعداد ردیف‌ها از ۲

میلیون و ۵۱۱ هزار به ۲ میلیون و ۴۹۴ هزار کاهش می‌یابد که کاهشی قابل چشم‌پوشی است. بنابراین برای از بین بردن داده‌های خالی ردیف‌هایی که شامل آن‌ها هستند را حذف می‌کنیم. همچنین همانطور که گفته شد ما قصد پیشبینی برای مشتریانی که حداقل دو خرید داشته اند را داریم. بنابراین خرید مشتریانی کمتر از این تعداد خرید را انجام داده‌اند از دیتاست حذف می‌کنیم. در نهایت ۲.۱۵ میلیون ردیف که مربوط به ۲۴۵ هزار سبد خرید از ۲۶ هزار مشتری است برای ما باقی می‌ماند.

۵. رویه حل مسئله

۵.۱. مشخص کردن نحوه‌ی حل مسئله

برای حل این مسئله می‌خواهیم از یادگیری ماشین استفاده کنیم. برای این کار ابتدا باید نوع مسئله، نحوه‌ی حل، پاسخ نهایی که می‌خواهیم دریافت کنیم و ورودی‌هایی که می‌خواهیم از طریق آن‌ها قضاوت کنیم را انتخاب کنیم.

نحوه‌ی حل این مسئله بر خلاف آنچه ممکن است در نگاه اول برسد، نحوه‌ی حل به این صورت که پیشبینی کنیم یک مشتری خرید بعدی را در چه تاریخی انجام خواهد داد و چه کالاهایی که در سبد خرید او خواهد بود. بلکه روش حل ما اینچنین خواهد بود که میسنجیم اگر قرار باشد مشتری در این روز سفارشی ثبت کند، چقدر احتمال دارد که این کالای خاص در سبد خرید مشتری باشد. دلیل استفاده از این مدل این است که در صورتی که بخواهیم هر مشتری را جداگانه بررسی کنیم و رفتار، زمان خرید بعدی و کالاهای موجود در سبد خرید بعدی آن را پیشبینی کنیم، نحوه‌ی حل بسیار سخت و محاسبات بسیار پیچیده خواهند شد. بنابراین به این ترتیب عمل میکنیم که با توجه به تعداد روزی که از سبد خرید قبلی مشتری و دفعه‌ی آخری که مشتری کالای مدنظر را سفارش داده است گذشته است، چه میزان احتمال دارد که (با توجه به سابقه‌ی مشتری و نوع کالا) دوباره آن کالا توسط مشتری خریداری شود.

این که چه ساعتی به مشتری پیشنهاد خرید را اعلام کنیم بستگی به رفتار کلی مشتریان یا رفتار خاص مشتریان دارد که احتمال خرید در چه ساعتی بالاتر است و در روزی که احتمال خرید مشتری در وضعیت مناسب‌تری بود در این مورد تصمیم‌گیری می‌شود و برای جلوگیری از پیچیده کردن مدل، از اعدادی که مربوط به اعداد ساعت سفارش می‌شود صرف‌نظر می‌کنیم و تنها روز سفارش را در نظر می‌گیریم.

در اینکه در چه روزی به مشتری پیشنهاد خرید یا تخفیف را ارسال کنیم نیز می‌توان دو رویکرد را در دستور کار قرار داد. روش اول این است که بدون در نظر گرفتن این مدل و با توجه به رفتار مشتری روزی که با احتمال بیشتری مشتری خرید خواهد کرد را به عنوان روز ارسال پیشنهاد به مشتری انتخاب کنیم. (که در پیشنهاد ارسالی شامل کالاهایی که در این مدل انتخاب شده‌اند خواهد بود) در رویکرد دیگر می‌توانیم از مدل برای انتخاب این روز استفاده کنیم. در این حالت می‌توانیم به عنوان مثال یکی از این سه روش را انتخاب کنیم:

۱. یک حد برای بیشترین احتمال خرید یک کالا در نظر بگیریم. به عنوان مثال اگر در یک روز احتمال خرید حداقل یکی از کالاها بیشتر از ۰.۶ بود، آن روز پیشنهاد خود را به مشتری ارسال کنیم.
۲. اگر مجموع احتمالات خرید بیشتر از یک عدد خاصی بود، مثلا می‌توانیم مشخص کنیم اگر مجموع احتمالات ۵ کالای محتمل بیشتر از ۱ باشد، این روز را به عنوان روز پیشنهاد انتخاب کنیم.
۳. می‌توانیم یک تعداد حداقلی برای تعداد کالاهایی که احتمالشان از یک احتمال حداقلی بیشتر است قرار دهیم. به عنوان مثال هر گاه حداقل ۳ کالا احتمالی حداقل برابر با ۰.۳ یا بیشتر داشته باشند، آن روز به مشتری پیشنهاد خود را ارسال می‌کنیم.

بنابراین انتخاب روز ارسال پیشنهاد به مشتری نیز از مسائلی است که می‌توان بیرون از مدل و به کمک نتیجه‌های نهایی آن را حل کرد و تاثیری در مدلی که آموزش داده می‌شود و قصد آن تنها این است که نشان دهد در روز خواسته شده، احتمال خرید مجدد توسط مشتری به چه میزان است.

در نهایت برای حل، این مسئله را به چشم یک مسئله‌ی با ناظر[۳۶] کلاس‌بندی[۳۷] دو کلاسه نگاه می‌کنیم. به این صورت که کلاس ۱ به معنی محتمل بودن خرید در این روز و کلاس ۰ به معنی محتمل نبودن خرید در این روز خواهد بود. البته باید توجه کرد که هدف ما این است که کالایی که به نظر مورد نیاز مشتری است به او یادآوری کنیم تا او را ترغیب به ثبت سفارش کنیم و هدف ما پیشبینی اینکه مشتری خرید انجام خواهد داد یا خیر، نیست. بنابراین آستانه‌ی پایین پذیرش[۳۸] احتمال خرید برای اینکه این کالا رو در کلاس ۱ قرار دهیم احتمالا بهتر است عددی کمتر از ۰.۵ باشد. با این حال پس از آموزش مدل و پیشبینی داده‌های تست، می‌توانیم این آستانه را به طوری که نتایج مناسب‌تری دریافت کنیم تنظیم کنیم. اما برای بررسی دقت مدل‌ها، مسئله را به شکل یک مسئله‌ی کلاس بندی مطرح می‌کنیم.

در ادامه برای ساخت مدل یادگیری ماشین نیاز داریم به کمک داده‌هایی که از خریدها داریم، ویژگی[۳۹]هایی که در مدل به ما کمک می‌کنند ایجاد کنیم. برای شروع این کار ابتدا ویژگی‌هایی که طور منطقی به نظر می‌آیند در تصمیم گیری مدل برای تعیین کلاس نهایی تاثیر گذار باشند را انتخاب و ایجاد می‌کنیم. در این حین ممکن است با تحلیل داده‌ها و یا بررسی بیشتر به ویژگی‌های بیشتری که به آموزش مدل ما کمک می‌کنند بپردازیم. در این مسیر سه نوع ویژگی داریم که باید آن‌ها را به کمک اطلاعات خود بسازیم. دسته‌ی اول ویژگی‌های مربوط به مشتری است. دسته‌ی دوم ویژگی‌های مربوط به محصول و دسته‌ی سوم ویژگی‌های مربوط به مشتری-محصول است. در واقع دسته‌ی سوم ویژگی‌ها مربوط به رفتار مشتری در رابطه با آن محصول خاص می‌شود و به رفتار باقی افراد با آن کالا و رفتار مشتری مدنظر با کالاهای دیگر ارتباط ندارد.

در این مسئله ویژگی‌های در ارتباط با کتگوری محصولات را نیز به صورت جداگانه در ویژگی‌ها ایجاد می‌کنیم. البته در واقع این ویژگی‌ها یعنی ویژگی‌های مربوط به کتگوری و ویژگی‌های مربوط به مشتری-کتگوری به ترتیب در همان دسته‌های ویژگی‌های مربوط به محصول و مشتری-محصول قرار می‌گیرد. در ادامه در هر بخش اشاره می‌کنیم که با توجه به داده‌هایی که داریم، در هر دسته چه ویژگی‌هایی ایجاد خواهیم کرد.

۵.۲. ساخت ویژگی‌های مربوط به مشتری

ویژگی‌های مربوط به مشتری آن ویژگی‌هایی است که به رفتار کلی مشتری در قبال کالاها و یا ویژگی‌های شخصی خود مشتری می‌پردازد و ارتباطی به محصول یا کالای خاص ندارد و برای تصمیم گیری در مورد خرید هر کالا توسط آن شخص، این ویژگی‌ها تاثیرگذار خواهند بود. با توجه به داده‌هایی که در جدول سفارشات داریم، می‌توانیم ویژگی‌های زیر را ایجاد کنیم.

- **تعداد کل کالاها:** این ستون تعداد کل کالاهایی که مشتری از ابتدا تا کنون در سبدهای خرید خود داشته است را نشان می‌دهد.
- **تعداد کالاهای یکتا**: این ستون تعداد کالاهایی یکتایی که مشتری تا به حال از فروشگاه خریداری کرده است را نشان می‌دهد. در این ستون اگر از یک کالای خاص چندین بار خرید شده باشد، یک بار شمرده می‌شود.
- **نرخ بازخرید مشتری**: چند درصد از کالاهایی که مشتری خریده است بازخرید بوده اند. باز خرید در اینجا به این معنی است که آن مشتری قبلا سابقه‌ی خرید آن را داشته باشد و مجددا آن را خریداری کند.
- **میانگین اندازه‌ی سبد**: به طور میانگین هر باری که مشتری سفارشی ثبت می‌کند چند کالا در سبد خرید او وجود دارد.

[36] supervised
[37] classification
[38] Threshold
[39] feature

- **فاصله‌ی بین خریدها**: میانگین فاصله‌ی زمانی بین دو خرید مشتری (روز)
- **تعداد کالاهای بازخرید شده توسط مشتری**: تعداد کالاهای یکتایی که توسط مشتری بازخرید شده اند.
- **نسبت محصولات باز خرید شده توسط مشتری**: تعداد کالاهای یکتایی که توسط مشتری بازخرید شده اند به کل تعداد یکتای محصولات خریده شده توسط مشتری
- **تعداد کتگوری‌های یکتا**: همانطور که در مورد کالاها گفته شد در مورد کتگوری‌ها
- **تعداد کتگوری‌های بازخرید شده توسط مشتری**: همانطور که در مورد کالاها گفته شد در مورد کتگوری‌ها
- **نسبت کتگوری‌های باز خرید شده توسط مشتری**: همانطور که در مورد کالاها گفته شد در مورد کتگوری‌ها
- **تعداد کلاس‌های یکتا**: همانطور که در مورد کالاها گفته شد در مورد کلاس‌ها
- **تعداد کلاس‌های بازخرید شده توسط مشتری**: همانطور که در مورد کالاها گفته شد در مورد کلاس‌ها
- **نسبت کلاس‌های باز خرید شده توسط مشتری**: همانطور که در مورد کالاها گفته شد در مورد کلاس‌ها

۵.۳. ساخت ویژگی‌های مربوط به محصول

در ویژگی‌های مربوط به محصول تنها مواردی که مربوط به خود محصول است و ارتباطی به خریدار آن ندارد آورده می‌شود. این بخش ذات خود محصول را نشان می‌دهد. به عنوان مثال محصولی مانند کره یا شیر محصولی است که افراد به صورت منظم مصرف می‌کنند و با فاصله‌های مشخص به خرید این کالاها می‌پردازند. اما کالایی مانند یک ادویه احتمالا خیلی با نسبت کمی بازخرید می‌شود و فاصله‌ی بین دو خرید برای یک مشتری بسیار زیاد و با واریانس بالا باشد. این دسته ویژگی‌های یک محصول برای تمام افراد یکسان خواهد بود. در دیتاست داده شده می‌توانیم به کمک داده‌ها ویژگی‌های زیر را از این دسته ایجاد کنیم:

- **نرخ بازخرید شدن**: تعداد دفعاتی که این کالا بازخرید شده به کل تعداد دفعاتی که خریداری شده است.
- **p_reduced_features**: همانطور که اشاره شد کلاس برخی از اجناس می‌تواند به گونه‌ای باشد که مشتریان به آن‌ها نیاز روزانه دارند و با فاصله‌های زمانی مشخصی از آن‌ها خرید می‌کنند. چند دسته‌ی مهم این اجناس عبارتند از شیر، سبزی و صیفی‌جات، میوه، سایر لبنیات، پروتئین، تنقلات و کالاهای اساسی مانند روغن، تخم‌مرغ و رب کالاهایی هستند که مرتبا برای یک مشتری لازم هستند. البته این کالاها ممکن است با فاصله‌های متفاوتی نیاز شوند. به عنوان مثال کالایی مانند رب گوجه هر هفته خریداری نمی‌شود اما در طرف مقابل کالایی مانند شیر بسیار تند مصرف است و ممکن است هر هفته نیز چندین بار نیاز شود (با توجه به تاریخ مصرف کوتاه این محصول) به این دلیل این ستون‌ها که مشخص می‌کند هر کالا برای کدام کلاس است ساخته می‌شوند اما برای جلوگیری از زیاد شدن تعداد فیچرها که زمان آموزش دادن الگوریتم را افزایش می‌دهد، از روش NMF برای کاهش ابعاد این ستون‌ها به سه ستون استفاده می‌کنیم که این ستون‌ها را p_reduced_features نامگذاری می‌کنیم.

۵.۴. ساخت ویژگی‌های مربوط به کتگوری

در این بخش تنها **نرخ بازخرید شدن کتگوری** قرار دارد که نشان می‌دهد در تمام خریدهایی که از یک کتگوری صورت گرفته است، چند درصد مواقع باز خرید بوده است. این نکته حائز اهمیت است که بازخرید کتگوری به این معنا است که مشتری قبلا از آن کتگوری خرید داشته باشد و لزومی ندارد که حتما همان کالا را از آن کتگوری خریده باشد.

۵.۵. ساخت ویژگی‌های مشتری-محصول

این دسته از ویژگی‌ها که احتمالا مهم‌ترین بخش ویژگی‌های ما باشند، رفتار مشتری در قبال یک محصول خاص را نشان می‌دهد. به عنوان مثال نشان می‌دهد که فرد مذکور به طور مرتب هر هفته در سبد خرید خود آب آلبالو داشته است و از طرفی شیر که انتظار می‌رود به طور مرتب در سبد خرید مشتری باشد را با فاصله‌های زیاد و غیر منظم خریداری می‌کند. پس از این ویژگی‌ها می‌توان با دقت بیشتری کار پیشبینی سبد خرید آینده را انجام داد. با توجه به دیتاستی که داریم این ویژگی‌های مشتری-محصول را می‌توانیم ایجاد کنیم:

- **نرخ سفارش‌دهی**: نشان می‌دهد این کالا چند درصد از کالاهایی که مشتری خرید کرده است را شامل می‌شده است. هر چه تعداد دفعات خرید مشتری بیشتر باشد، این نرخ بیشتر خواهد بود. البته به تعداد کل محصولات خریده شده نیز بستگی دارد.
- **نرخ بازخرید**: در این قسمت نشان داده می‌شود چند بار از دفعاتی که مشتری این کالا را خریده است به شکل باز خرید بوده است. در واقع اگر کالا تنها یک بار خریده شده باشد این نرخ برابر با صفر است و در غیر این صورت هرچه تعداد خرید از این کالا بیشتر باشد این نرخ به یک نزدیک‌تر می‌شود.
- **فاصله‌ی u_p از خرید قبلی**: نشان می‌دهد آخرین باری که مشتری از این کالا در سبد خرید خود داشته است، چند خرید پیش بوده است.
- **فاصله‌ی u_t از خرید قبلی**: نشان می‌دهد آخرین باری که مشتری از این کتگوری در سبد خرید خود داشته است، چند خرید پیش بوده است.
- **فاصله‌ی u_c از خرید قبلی**: نشان می‌دهد آخرین باری که مشتری از این کلاس در سبد خرید خود داشته است، چند خرید پیش بوده است.
- **Max_streak**: نشان می‌دهد که حداکثر چند خرید پشت سر هم در تاریخ خریدهای مشتری وجود دارد که همگی شامل این کالا باشند.

۵.۶. ساخت ویژگی‌های مشتری-کتگوری

این دسته از ویژگی‌ها نیز مانند دسته‌ی قبل است با این تفاوت که رفتار مشتری نسبت به یک کتگوری خاص را بررسی می‌کنیم.

- **نرخ سفارش‌دهی**: نشان می‌دهد این کالا چند درصد از کتگوری‌هایی که مشتری خرید کرده است را شامل می‌شده است. هر چه تعداد دفعات خرید مشتری بیشتر باشد، این نرخ بیشتر خواهد بود. البته به تعداد کل کتگوری‌های خریده شده نیز بستگی دارد.
- **نرخ بازخرید**: در این قسمت نشان داده می‌شود چند بار از دفعاتی که مشتری این کتگوری را خریده است به شکل باز خرید بوده است. در واقع اگر کالا تنها یک بار خریده شده باشد این نرخ برابر با صفر است و در غیر این صورت هرچه تعداد خرید از این کالا بیشتر باشد این نرخ به یک نزدیک‌تر می‌شود.
- **Max_streak_cat**: نشان می‌دهد که حداکثر چند خرید پشت سر هم در تاریخ خریدهای مشتری وجود دارد که همگی شامل این کتگوری باشند.

۵.۷. ساخت ویژگی‌های مربوط به زمان

این دسته از ویژگی‌ها بیشتر مربوط به این هستند که چه احتمالی وجود دارد که مشتری در این زمان خرید خود را انجام دهد.

- **فاصله از آخرین سفارش**: در این حالت فاصله‌ی روزی که می‌خواهیم احتمال خرید را بررسی کنیم با آخرین خریدی که توسط مشتری انجام شده است محاسبه می‌کنیم.
- **روز هفته**: در این بخش روزی از هفته که قصد بررسی آن را داریم و عددی بین ۰ تا ۶ است نشان می‌دهد.
- **درصد خرید محصول در این روز از هفته**: نشان می‌دهد چند درصد از خریدهای مشتری از این محصول در این روز از هفته بوده است.
- **درصد خرید کتگوری در این روز از هفته**: نشان می‌دهد چند درصد از خریدهای مشتری از این کتگوری در این روز از هفته بوده است.
- **درصد خرید کلاس در این روز از هفته**: نشان می‌دهد چند درصد از خریدهای مشتری از این کلاس در این روز از هفته بوده است.
- **درصدی از بازخریدهای کل آن محصول که پس از t روز انجام شده است**: در اینجا t فاصله‌ی روزی است که می‌خواهیم احتمال خرید در آن را بسنجیم تا آخرین روزی که آن محصول خریده شده است.
- **درصدی از بازخریدهای کل آن کتگوری که پس از t روز انجام شده است**: در اینجا t فاصله‌ی روزی است که می‌خواهیم احتمال خرید در آن را بسنجیم تا آخرین روزی که آن کتگوری خریده شده است.
- **درصدی از بازخریدهای کل آن کلاس که پس از t روز انجام شده است**: در اینجا t فاصله‌ی روزی است که می‌خواهیم احتمال خرید در آن را بسنجیم تا آخرین روزی که آن کلاس خریده شده است.
- **درصدی از خریدهای یک مشتری که پس از t روز از خرید قبلی انجام شده است**: در اینجا t فاصله‌ی روزی است که می‌خواهیم احتمال خرید در آن را بسنجیم تا آخرین روزی که آن خریدی ثبت است.
- **درصدی از بازخریدهای مشتری مدنظر از آن محصول که پس از t روز انجام شده است**: در اینجا t فاصله‌ی روزی است که می‌خواهیم احتمال خرید در آن را بسنجیم تا آخرین روزی که آن محصول خریده شده است.
- **درصدی از بازخریدهای مشتری مدنظر از آن کتگوری که پس از t روز انجام شده است**: در اینجا t فاصله‌ی روزی است که می‌خواهیم احتمال خرید در آن را بسنجیم تا آخرین روزی که آن کتگوری خریده شده است.
- **درصدی از بازخریدهای مشتری مدنظر از آن کلاس که پس از t روز انجام شده است**: در اینجا t فاصله‌ی روزی است که می‌خواهیم احتمال خرید در آن را بسنجیم تا آخرین روزی که آن کلاس خریده شده است.

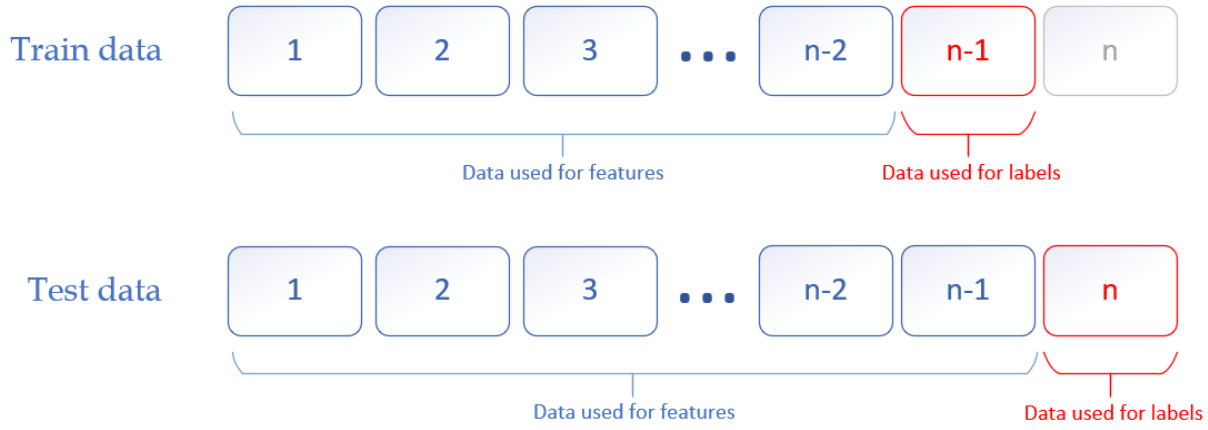
۶. آماده سازی داده‌ها برای آموزش مدل یادگیری ماشین

۶.۱. مشخص کردن لیبل و جدا کردن دیتای تست و آموزش

همانطور که گفته شد، هدف ما در این پروژه این است که خرید مجدد کالاهایی که قبلا توسط مشتری خریداری شده است را پیشبینی کنیم. بنابراین اگر بخواهیم پیشبینی را برای خرید n ام انجام دهیم، آن کالاهایی باید برای آن‌ها پیشبینی خرید یا عدم خرید انجام دهیم عبارت اند از تمام کالاهایی که مشتری مدنظر در n-1 سبد قبلی آن‌ها را خریده است. حال برای مشخص کردن کالاهایی که لیبل یک (خریداری شده) دارند، باید محصولاتی از سبد n ام که در سابقه‌ی خرید آن حداقل در یکی از n-1 خرید قبلی بودند را جدا کنیم. مجددا این نکته حائز اهمیت است که قصد ما پیشبینی خرید کالای جدید نیست. بنابراین با کالاهایی که برای اولین بار در خرید n ام خریده می‌شوند کاری نداریم و از آن‌ها را از میان دیتاهای خود حذف می‌کنیم.

برای مشخص کردن داده‌های آموزش[۴۰] و تست[۴۱] باید به این نکته توجه کنید که نباید دیتایی که قرار است به وسیله‌ی آن کارایی مدل تست شود، در آموزش مدل استفاده شود. این امر باعث بیش برازش[۴۲] مدل می‌شود. بنابراین باید توجه کنیم هیچگاه لیبل‌های دیتای تست نباید در دیتای آموزش بیاید.

بنابراین برای دیتای تست، کالاهای سبد n ام را به عنوان لیبل و کالاهای n-1 سبد قبلی را به عنوان دیتایی که از آن‌ها ویژگی‌های دیتاست را استخراج می‌کنیم. استفاده می‌کنیم. می‌توانیم با جداسازی دیتای آموزش و تست[۴۳] می‌توانیم آموزش دیتا را آغاز کنیم اما می‌توانیم با جداسازی دیتای سبد n ام و لیبل قرار دادن کالاهای خرید n-1 ام، دیتای آموزش جدیدی تولید کنیم. به این ترتیب می‌توانیم از این مورد نیز اطمینان خاطر پیدا کنیم که رفتار گذشته‌ی هر فرد نیز در مورد آموزش مدل قرار گرفته است. بنابراین n-1 سبد ابتدایی هر مشتری را به عنوان دیتای آموزشی جدا می‌کنیم که در آن n-2 سبد اول برای محاسبه‌ی ویژگی‌ها و سبد n-1 ام به عنوان دیتا لیبل استفاده می‌شود. سپس برای سنجیدن کارایی مدل از n-1 سبد اول برای ایجاد ویژگی‌ها و از سبد n ام به عنوان لیبل استفاده می‌کنیم. همچنین با توجه به این مورد که در این صورت حدودا تعداد داده‌های تست و آموزش با هم برابر می‌شود، می‌توانیم برای آموزش بهتر مدل، بخشی از داده‌هایی که سبد n ام مشتری لیبل آن است، برای آموزش نیز استفاده کنیم.



[40] Train data
[41] Test data
[42] Overfitting
[43] Train-Test split

۶.۲. نرمال سازی داده‌ها

داده‌هایی که در بخش‌ها ۲-۵ تا ۷-۵ مشخص شد، مقیاس یکسانی ندارند و این امر باعث اختلال در یادگیری برخی مدل‌های یادگیری ماشین می‌شود. به عنوان مثال تفاوت یک روزه در زمانی که از آخرین خرید یک کالا توسط مشتری گذشته است (به عنوان مثال ۱۴ روز و ۱۵ روز) در برخی مدل‌ها همان تفاوتی را ایجاد می‌کند که نرخ بازخرید ۱ و ۰ ایجاد میکند. این در حالی است که تفاوت یک روزه نباید تغییر زیادی در احتمال خرید مشتری در آن روز داشته باشد، اما نرخ بازخرید یک به آن معناست که مشتری به احتمال زیاد این کالا را بازخرید می‌کند و بالعکس اگر نرخ بازخرید صفر باشد به این معناست که مشتری تمایلی به خرید مجدد آن ندارد. در برخی مدل‌ها این تفاوت مقیاس ممکن است مشکلی به وجود نیاورد اما به عنوان مثال در مدلی مانند KNN (K-Nearest Neighbor) فاصله‌ای که به ازای یک روز تفاوت در مدت زمان گذشته از خرید قبلی ایجاد می‌شود، برابر خواهد بود با تفاوتی که به علت اختلاف یک واحدی نرخ بازخرید ایجاد می‌شود. حال برای این مشکل باید به این نکته توجه کنیم که بازه‌ی تغییرات و یا انحراف معیار هر ویژگی به چه اندازه است. در مثالی که گفته شد، روزهای گذشته از خرید قبلی یک کالا توسط مشتری می‌تواند بین صفر تا عددی مانند ۱۰۰ و حتی بیشتر هم باشد، در طرف مقابل نرخ بازخرید در هر حال عددی بین صفر و یک است. به کمک آنچه گفته شد مشخص می‌شود که انحراف معیار ویژگی اول نیز در مقایسه با ویژگی دوم به شدت بیشتر است. برای حل این مشکل می‌توانیم به کمک روابط آماری و کتابخانه‌هایی که بر اساس آن‌ها طراحی شده اند، مقیاس تمامی ستون‌ها را یک اندازه کنیم. یکی از روش‌های این کار استفاده از مقیاس بندی بیشینه-کمینه [۴۴] است که منطق آن این است که اگر فاصله‌ی بین بیشینه و کمینه‌ی یک ویژگی را به صورت خطی در نظر بگیریم، برای یک ردیف خاص، این عدد چه نسبتی از این خط را پوشش می‌دهد. در مثالی که گفته شد اگر کمینه‌ی روز گذشته از خرید آخر یک محصول برابر با یک و بیشینه‌ی آن برابر با ۵۱ باشد (منظور کمینه و بیشینه‌ی ستون فاصله‌ی خرید از روزهای سپری شده است.) و بخواهیم با مقیاس بندی مقدار جدید ردیفی که در آن مقدار متناظر برابر با ۸ است را معرفی کنیم باید توجه کنیم که عدد ۸ چه نسبتی از فاصله‌ی خطی بین ۱ و ۵۱ را پوشش می‌دهد که این عدد برابر با ۰.۱۴ خواهد بود. به طور کلی این روش بر اساس رابطه‌ی مقابل عمل می‌کند.

$$y = \frac{x - x_{min}}{x_{max} - x_{min}}$$

که در آن y مقدار جدید با مقیاس جدید، x داده‌ی اصلی آن ردیف با مقیاس اصلی و x_{min} و x_{max} به ترتیب کمینه و بیشینه مقدار آن ویژگی هستند. در این روش y بدست آمده همواره عددی بین صفر و ۱ خواهد بود و به این ترتیب تمام ویژگی‌ها مقیاس و در نتیجه تاثیری یکسان در تمامی مدل‌های یادگیری ماشین خواهند داشت.

مشکلی که به روش بالا وارد است این است که این روش به داده‌های پرت به شدت حساس است. در همان مثال مذکور، اگر از خرید کالایی ۲۵۰ روز گذشته باشد (در حالی که این عدد در میان باقی ردیف‌ها بسیار کوچک‌تر از این باشد) ، باعث می‌شود که مخرج کسر بیش از اندازه بزرگ شود و اکثر غریب به اتفاق اعداد ردیف‌ها مقداری کمتر از ۰.۵ به خود بگیرند و این مشکل ممکن است تنها به دلیل وجود یک داده‌ی پرت در میان میلیون‌ها ردیف به وجود بیاید. برای جلوگیری از بروز این مشکل از روش مقیاس بندی نرمال استاندارد[۴۵] استفاده می‌کنیم. در این روش فرض را بر نرمال بودن توزیع اعداد یک ویژگی در نظر گرفته و به جای استفاده از بیشینه و کمینه‌ی ستون برای یافتن دامنه‌ی تغییرات اعداد، از میانگین و انحراف معیار آن استفاده می‌کنیم. به طور کلی در این روش از معادله‌ی زیر برای یافتن عدد جدید استفاده می‌شود:

[44] Max-Min Scaler
[45] Standard Scaler

$$z = \frac{x - \mu}{\sigma}$$

که در آن μ میانگین داده‌های ستون و σ انحراف معیار آن‌هاست. در این روش با فرض نرمال بودن توزیع داده‌ها، اعدادی با توزیع نرمال استاندارد (با میانگین صفر و انحراف معیار ۱) ایجاد می‌کنیم و به این وسیله مقیاس تمام ستون‌ها یکسان می‌شود. مشخص است که در این روش ستون‌ها مانند روش قبل بازه‌ی مشخصی ندارند. ایراد این روش نسبت به روش قبلی آن است که به دلیل آنکه باید میانگین و انحراف معیار داده‌ها محاسبه شود، محاسبات زمان‌بر تر از حالت قبلی خواهد بود. با این حال با توجه به اینکه داده‌های ما از مشتریان زیادی در طول چندین ماه جمع آوری شده است، امکان وجود داده‌ی پرت در آن‌ها زیاد است، پس برای یکسان سازی مقیاس داده‌ها در این پروژه از روش نرمال استاندارد استفاده می‌کنیم.

۶.۳. متعادل سازی داده‌ها

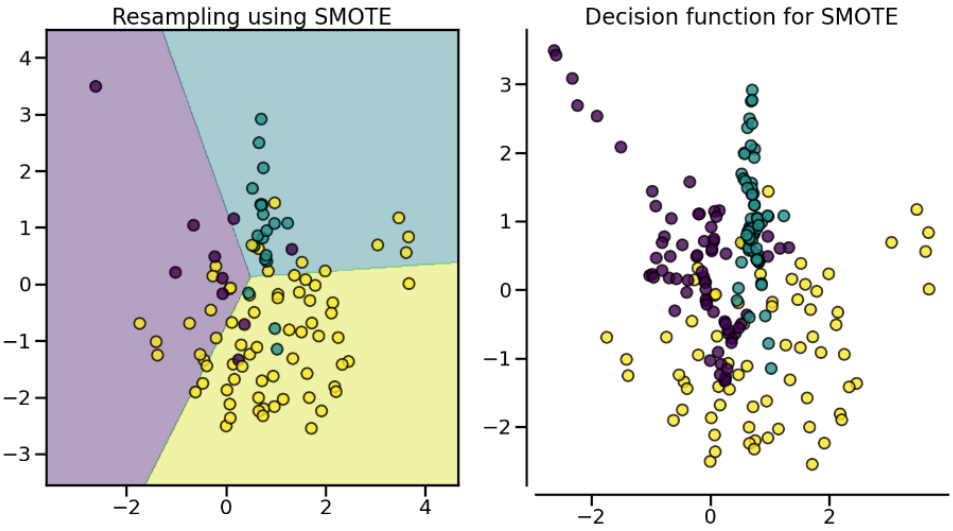
اگر به داده‌ها توجه کنیم به سادگی مشخص می‌شود که درصد کمی از داده‌ها لیبل ۱ (مجددا خریداری شده) دارند و درصد غالب داده‌ها لیبل صفر دارند. دلیل آن این است که تنها کالاهای بازخرید شده از سبد آخر هستند که لیبل یک می‌گیرند. این در حالی است که تمام کالاهایی که برای یک فرد بررسی می‌کنیم، شامل تمام کالاهایی است که آن فرد در n-1 خرید قبلی‌اش آن‌ها را در سبد خرید خودش داشته است. برای مثال در داده‌های آموزشی ۶.۵ درصد از داده‌ها لیبل ۱ و باقی آن‌ها لیبل صفر دارند.

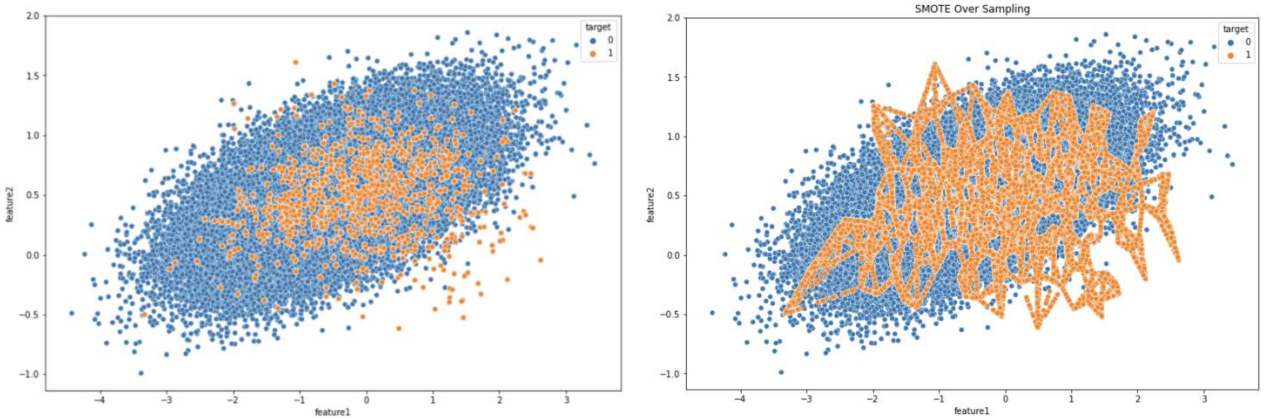
مشکل نامتعادل بودن داده‌ها در این بخش این است که مدل یادگیری ماشین برای آموزش دیدن سعی می‌کند دقت[۴۶] مدل را بیشینه کند و دقت یعنی چند درصد از داده‌ها را درست پیشبینی کرده است، بنابراین مدل در این بخش اگر تمام لیبل‌ها را صفر پیشبینی کند، با دقت حدود ۹۴ درصد پیشبینی‌ها را انجام داده‌است که در ظاهر دقت قابل قبولی است اما واقعیت به گونه‌ی دیگریست. اگر مدل تمام داده ها را صفر گزارش کند با اینکه با دقت بالایی کالاها پیشبینی شده‌اند اما اتفاقی که می‌افتد این است که هیچ کالایی به عنوان کالای سبد خرید بعدی مشتری معرفی نمی‌شود و این یعنی که ما هیچ کاری انجام نداده‌ایم! در برخی از مدل‌ها ممکن است تمام ردیف‌ها صفر پیشبینی نشوند اما به دلیل عدم تعادل زیاد داده‌ها، تمایل به سمت صفر گزارش کردن داده‌ها زیاد است. در واقع اگر داده‌ای یک گزارش شود به احتمال ۹۴ درصد اشتباه است پس مدل سعی می‌کند حتی‌الامکان لیبل را صفر گزارش کند. از طرفی در اینجا برای ما دیتایی که لیبل ۱ دارد اهمیت بیشتری دارد. زیرا ما به دنبال تشخیص کالاهایی هستیم که مشتری در خرید بعدی آن‌ها را خریداری می‌کند. اگر تعدادی از کالاهایی که مشتری با احتمال کمتری آن‌ها را خریداری می‌کند با لیبل ۱ پیشبینی شوند نتیجه‌ی کار آن است که در چند کالای بیشتر به مشتری تخفیف می‌دهیم و یا می‌توانیم موارد محتمل‌تر را در تخفیف لحاظ کنیم؛ اما اگر کالایی که محتمل است مشتری خریداری کند با لیبل صفر پیشبینی شود، ممکن است به قیمت از دست دادن یک مشتری برای فروشگاه تمام شود. به این دلیل تمایل ما به سمت این است که اگر خطایی در پیشبینی وجود دارد، این خطا از نوع مثبت کاذب[۴۷] باشد. حال برای برطرف کردن این مشکل از امکانی در کتابخانه‌ی imblearn پایتون به اسم نمونه برداری اضافه یا oversampling استفاده می‌کنیم. این امکان به کمک شناسایی الگوی داده‌های قبلی، داده‌های جدید مشابه با آن‌ها و با همان الگو تولید می‌کند تا عدم تعادل در داده‌ها را برطرف کند. در این کتابخانه ۸ روش تولید داده‌ی اضافی از جمله RandomOverSampler، SMOTE و ADASYN قرار دارد که برای این پروژه از روش SMOTE استفاده می‌کنیم.

روش ترکیبی نمونه برداری اضافی از اقلیت یا SMOTE[۴۸] به این طریق کار می‌کند که هر نقطه‌ی اقلیت (در اینجا نقاطی که لیبل ۱ دارند) را در نظر می‌گیرد و با ایجاد نقاط مصنوعی جدید، آن را به k تا از نزدیک‌ترین همسایگانش متصل می‌کند. به صورت

[46] Accuracy
[47] False Positive
[48] Synthetic Minority Oversampling Technique

پیشفرض k برابر با ۵ است. فاصله‌ی بین نقاط برای محاسبه‌ی همسایگان نزدیک با توجه به اینکه تمام ستون‌ها در بخش قبل نرمال سازی شده‌اند، به طور استانداردی محاسبه می‌شود. تعداد و فاصله‌ی نقاط مصنوعی با توجه به تعداد نقطه‌ی مصنوعی مورد نیاز برای متعادل کردن دیتاست و همچنین فاصله‌ی دو نقطه‌ی انتخاب شده تعیین می‌شوند. در نهایت همانطور که در دو مثال زیر که نمونه‌هایی دو بعدی را نشان می‌دهند مشخص است، تقریبا شکل داده‌ی اقلیت پس از متعادل سازی به گونه‌ای است که انگار نقاط اصلی را به کمک نقاط مصنوعی جدید به یکدیگر متصل کرده ایم.





پس داده‌ها را به کمک این روش متعادل می‌کنیم تا مشکلی که پیش از این در مورد آن بحث شد، مرتفع شود و مدل آماده‌ی آموزش شود.

۷. نتایج گزارش شده توسط مدل‌های یادگیری ماشین و بررسی نتایج آن‌ها

۷.۱. گزارش کلاس‌بندی[۴۹]

پس از نرمال‌سازی و متعادل کردن داده‌ها حالا داده‌ها تقریبا آماده برای یادگیری هستند. داده‌هایی که لیبل آن‌ها مربوط به خرید n ام مشتری است حدود یک میلیون و صد هزار رکورد و داده‌هایی که لیبل آن‌ها مربوط به خرید n-1 ام است حدود۹۶۰ هزار رکورد هستند. برای آموزش مدل هر چه از داده‌های بیشتری استفاده کنیم بهتر است. اما برای اینکه در صورت بیش برازش مدل بتوانیم آن را تشخیص دهیم، باید بخشی از داده‌ها را برای تست مدل کنار بگذاریم. داده‌هایی که برای تست مدل کنار گذاشته می‌شوند نباید از ۲۰ الی ۲۵ درصد کل مجموعه بیشتر باشند. هر چه دیتاست بزرگتر باشد (منظور این است که ردیف‌های بیشتری داشته باشد) این درصد می‌تواند کمتر و کمتر شود. برای این دیتاست حدود ۱۵ درصد از داده‌ها را برای تست کارایی مدل‌ها جدا می‌کنیم. باید توجه کنیم طبق آمچه در بخش ۱-۶ گفته شد، نباید دیتایی که قرار است در تست مدل پیشبینی شود، در آموزش آن استفاده شده باشد. به این دلیل تنها از داده‌هایی که خرید n ام را پیشبینی می‌کنند به اندازه‌ی حدود ۱۵ درصد کل داده‌ها در داده‌های تست نگه می‌داریم و از باقی ردیف‌ها برای آموزش مدل استفاده می‌کنیم. در نهایت ۳۳۰ هزار ردیف برای تست و یک میلیون و ۷۳۰ هزار ردیف برای آموزش خواهیم داشت. حال مدل‌های یادگیری ماشین را انتخاب کرده و شروع به آنان می‌کنیم.

در ابتدا باید به این نکته توجه کنیم که برخی از مدل‌ها برای دیتاست‌های بزرگ کارایی مناسبی ندارند و برخی نیز زمان اجرای بسیار زیادی برای این نوع دیتاست‌ها خواهند داشت. به عنوان مثال در روش KNN مدل باید فاصله‌ی ۳۹ بعدی بین یک میلیون و ۷۰۰ هزار ردیف را محاسبه کند تا در نهایت بر اساس K همسایه‌ی نزدیک کلاس آن ردیف را مشخص کند. پس این روش نمی‌تواند روش مناسبی برای پیشبینی باشد. در نهایت ۷ مدل که در جدول زیر آورده شده اند برای پیشبینی سبد خرید بعدی مشتری آموزش داده شده‌اند. برای ارزیابی عملکرد مدل‌ها باید توجه کنیم که کلاسی که مدل برای یک ردیف تعیین می‌کند می‌تواند صفر یا یک باشد و هر کدام از این پاسخ‌ها ممکن است صحیح یا غلط باشند بنابراین در ارزیابی عملکرد یک مدل یک کلاس پیشبینی شده می‌تواند ۴ حالت داشته باشد که با True Positive، False Positive، True Negative و False Negative مشخص می‌شوند. همانطور که در بخش ۳-۶ اشاره شد، داده‌هایی که لیبل ۱ دارند بسیار کمتر از لیبل صفر هستند اما بر خلاف آموزش، در تست مدل باید از داده‌های واقعی استفاده کنیم زیرا می‌خواهیم کارایی مدل در هنگامی که قرار است با داده‌های واقعی کار کند را ببینیم. بنابراین اگر از روش‌های متعادل سازی استفاده کنم به ارزیابی صحیحی از مدل دست پیدا نمی‌کنیم. به علاوه مطمئن نیستیم که داده‌های مصنوعی تولید شده توسط الگوریتم متعادل سازی واقعا لیبل نسبت داده شده را داشته باشند.

همانطور که پیش‌تر اشاره شد، پیشبینی درست کلاس ۱ برای ما اهمیت بیشتری دارد و معیار Accuracy با توجه به اینکه بیشتر داده‌ها کلاس صفر دارند، نمی‌تواند معیار خوبی برای ارزیابی ما باشد. بنابراین از معیارهایی که تمرکز آن بر روی پاسخگویی صحیح به کلاس ۱ است استفاده می‌کنیم. این معیارها عبارت اند از:

- Precision: این معیار مشخص می‌کند که چه میزان از ردیف‌هایی که مدل کلاس آن‌ها را ۱ پیشبینی کرده است، واقعا کلاس یک داشته‌اند. در واقع این معیار برای آن است که بدانیم مدل برای اطلاق کلاس ۱ به یک نمونه زیاده روی نمی‌کند و تا زمانی که به حد خاصی از اطمینان نرسیده باشد این کلاس را به نمونه نسبت نمی‌دهد. به طور کلی زیاد شدن پیشبینی‌های مثبت کاذب از عوامل افت این معیار است و از فرمول زیر به دست می‌آید:

$$Precision = \frac{TP}{TP + FP}$$

[49] Classification report

- Recall: این معیار نشان می‌دهد که مدل ما چند درصد از ردیف‌هایی که واقعا به کلاس ۱ تعلق داشته‌اند را به درستی پیشبینی کرده است. در واقع اگر اشتباه تشخیص دادن کلاس نمونه‌ای که کلاس واقعی آن برابر با ۱ است هزینه‌ی زیادی داشته باشد، بالا بردن این معیار از اهمیت زیادی برخوردار می‌شود. بنابراین اگر منفی‌های کاذب ما کمتر باشند این شاخص بیشتر و بیشتر خواهد بود. این شاخص از رابطه‌ی زیر به دست می‌آید.

$$Recall = \frac{TP}{TP + FN}$$

- F1-score: دو معیار بالا شاخص‌های خوبی برای حالتی هستند که کلاس ۱ اهمیت بالاتری برای ما داشته باشد. اما استفاده از آن‌ها به صورت جداگانه ایراداتی دارد. اگر مدل کلاس تمام نمونه‌ها را یک پیشبینی کند و یا بسیار سهل‌گیرانه به نمونه‌ها کلاس یک نسبت دهد، هیچ منفی کاذبی نداریم و در واقع معیار Recall برابر با یک خواهد شد. حال از طرفی اگر مدل بسیار سخت‌گیرانه در پیشبینی کلاس ۱ عمل کند و به عنوان مثال به تعداد انگشت شماری از نمونه‌ها که بسیار از آن‌ها مطمئن است کلاس ۱ نسبت دهد، معیار Precision مقدار یک به خود می‌گیرد. بنابراین نگاه جداگانه به این دو معیار به ما گزارش خوبی نمی‌دهد و همچنین بررسی همزمان دو عدد با دو عدد دیگر برای فهمیدن کارایی یک مدل بسیار دشوار و همراه با خطاهایی خواهد بود. پس بهتر است از یک معیار که نشانگر هر دو معیار قبلی باشد استفاده کنیم. F1-score معیاری است که از میانگین توافقی[۵۰] (یا همساز) دو معیار Recall و Precision به دست می‌آید و واحد آن نیز برابر با همان معیار هاست. علت استفاده از میانگین توافقی این است که بالارفتن بیش از اندازه‌ی یک معیار که دلایل آن سخت‌گیری یا سهل‌گیری زیاد مدل است، بیش از اندازه موجب بالارفتن این معیار نشود. میانگین توافقی دو عدد، عددی بین آن دو و نزدیک به عدد کوچکتر است و هر چه عدد کوچکتر مقدار کمتری داشته باشد، این میانگین بیشتر تحت تاثیر قرار می‌گیرد و بالابردن بهینه‌ی آن در گروی کم شدن فاصله‌ی بین دو عدد است و در واقع این میانگین نوعی سنجش گرایش به مرکز نیز هست. به عنوان مثال سه جفت عدد (۰.۵،۰.۵)، (۰.۹۹،۰.۰۱) و (۱،۰) را در نظر بگیرید. میانگین حسابی هر سه جفت برابر با ۰.۵ است حال آنکه میانگین توافقی آنها به ترتیب برابر با ۰.۵، ۰.۰۱۹۸ و صفر. پس متوجه می‌شویم که ضعف در یک معیار به طور مشخصی در F1-score مشخص شده و معیاری است که برای بالابردن آن باید مدل از هر نظر مناسب عمل کند. همچنین برای بهبود دادن، بالابردن این معیار نشانگر بهبود قابل توجهی در مدل است. فرمول ریاضی این روش به طریق زیر محاسبه می‌شود.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

حال که معیارهای کارایی مدل‌ها را نشان دادیم، با هفت مدلی که آموزش داده‌ایم، این معیارها را بر روی داده‌های تست بدست می‌آوریم که عبارت خواهند بود از:

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic regression	0.91	0.33	0.42	0.37
Gaussian Naïve Bayes	0.82	0.20	0.58	0.29
Ada Boost	0.83	0.22	0.64	0.33
Decision Tree	0.88	0.20	0.29	0.24
Random Forest	0.89	0.31	0.55	0.39
Extra Trees	0.90	0.32	0.47	0.38
XGBoost	0.91	0.35	0.45	0.39

[50] Harmonic mean

با توجه به گزارش بالا می‌بینیم که دقت مدل‌های آموزش دیده شده همگی در سطح بالا و تقریبا مناسبی قرار دارند اما علت آن این است که درصد قابل توجهی از داده‌ها لیبل صفر دارند و مدل‌ها در این نوع دیتاست‌ها کار سختی برای پیشبینی با دقت بالا ندارند. همانطور که اشاره شد در این نمونه‌ها برای ما بیشتر داده‌هایی که کلاس ۱ دارند اهمیت دارند. بنابراین از سه معیار داده شده استفاده کردیم. حال مجددا به مسئله‌ی اصلی از کمی عقب‌تر نگاه می‌کنیم. قصد ما این است که با شناخت نیاز مشتری در یک زمان خاص و ارائه‌ی پیشنهاداتی که احتمالا مورد نیاز اوست، او را به سمت خرید مجدد از فروشگاه هدایت کنیم. حال در داده‌های تست هر چه درصد بیشتری از کالاهایی که مشتری واقعا خریده است را پیشبینی کنیم، مدل بهتری داریم و پیشبینی و پیشنهاد دادن برخی کالاهایی که مشتری خریدی از آنان انجام نداده است می‌تواند تا حدی قابل چشم پوشی باشد. همچنین پیشنهاد کالایی دیگر و یا اعمال تخفیف بر روی آن ممکن است باعث ترغیب مشتری به خرید آن نیز بشود. چه بسا در میان داده‌های تست اگر قبل از خرید، برخی از کالاهایی که خریده نشده‌اند (کلاس صفر هستند) را به مشتریان پیشنهاد می‌دادیم، آن کالاها توسط مشتریان خریداری می‌ش اما به هر حال اگر بخش قابل توجهی از داده‌هایی که کلاس یک به آن‌ها نسبت داده شده است مثبت کاذب باشند، مشخصا مدل کارایی خوبی ندارد. بنابراین مبنای اصلی ما برای تعیین بهترین مدل همان F1-score است در حالی که می‌توانیم برای انتخاب مدل بهتر نگاهی نیز به معیار Recall که نشان می‌دهد چند درصد از کلاس‌های ۱ واقعی را پیشبینی کردیم، داشته باشیم.

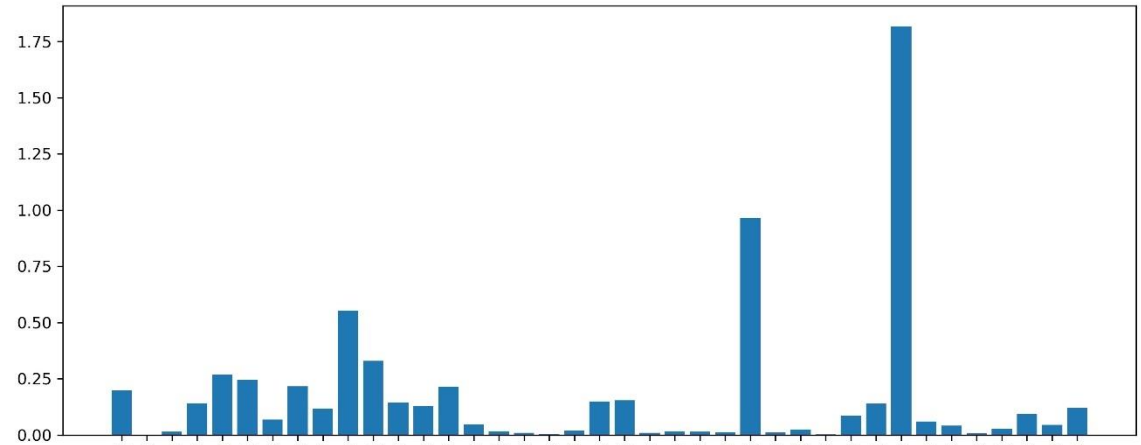
با توجه به آنچه گفته شد و جدول بالا، مدل‌های جنگل تصادفی و XGBoost بیشترین F1-score را دارند اما معیار Recall در روش جنگل تصادفی حدودا ۱۰ درصد بیشتر از روش XGBoost است. بنابراین این روش را به عنوان بهترین روش انتخاب می‌کنیم. همچنین روش ADABoost و رگرسیون لاجستیک Recall بالاتری از روش جنگل تصادفی دارند اما به دلیل آن که F1-score آن‌ها کمتر از روش جنگل تصادفی است، این روش‌ها را انتخاب نمی‌کنیم. این نکته حائز اهمیت است که چون F1-score یک میانگین توافقی میان دو عدد است، افزایش آن ساده نیست و باید هر دو معیار با یکدیگر افزایش پیدا کنند و افزایش معیار بزرگتر و کاهش معیار دیگر به همان اندازه باعث افت این میانگین می‌شود. بنابراین تفاوت اندک در این معیار می‌تواند نشانگر تفاوت کارایی چشمگیری باشد.

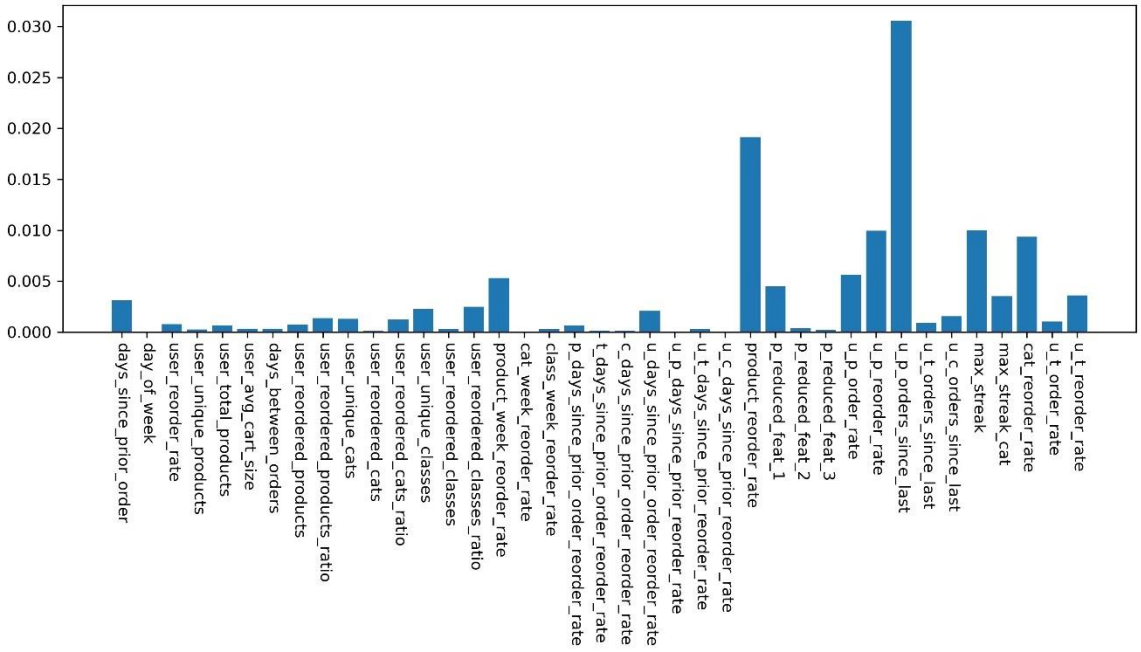
۷.۲. اهمیت ویژگی[51]

یکی از کارهایی که پس از آموزش هر مدل یادگیری ماشین مهم است، بررسی اهمیت هر ویژگی است. ویژگی‌هایی که در بخش ۵ به کمک داده‌های دیتاست آن‌ها را ایجاد کردیم هم اکنون باید مورد بررسی قرار بگیرند. هدف از این بررسی آن است که اولا ویژگی‌هایی که در اغلب مدل‌ها تاثیر ناچیزی دارند را شناسایی کنیم و با حذف آن‌ها زمان یادگیری و پیشبینی مدل را کاهش دهیم و دوما ویژگی‌هایی که اهمیت بالایی دارند را شناسایی کنیم و سعی کنیم به کمک آن‌ها و ویژگی‌های جدیدی که حس می‌کنیم به افزایش کارایی مدل کمک کند ایجاد می‌کنیم. به هر حال این بخش از پروژه‌ی یادگیری ماشین بخشی است که بخش زیادی از آن مربوط به بینش دانشمند داده[52] و شناخت آن از مسئله و کسب و کار بستگی دارد و باید توسط شخص مشخص شود.

در نمودارهای زیر اهمیت ویژگی‌ها در مدل‌های مختلف آورده شده است. با توجه به تفاوت رویه‌ی کار مدل‌ها، مقیاس نمودارها با یکدیگر متفاوت است اما هدف ما قیاس نسبی هر ویژگی با دیگر ویژگی‌های همان مدل است.

[51] Feature importance
[52] Data Scientist

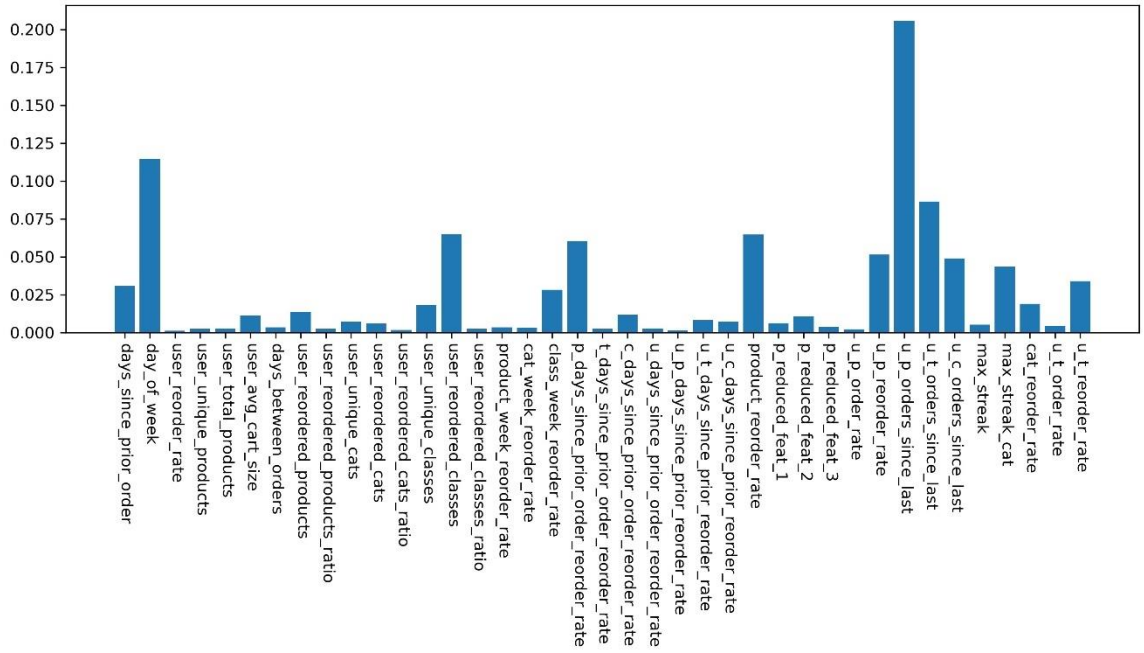

1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
days_since_prior_order
day_of_week
user_reorder_rate
user_unique_products
user_total_products
user_avg_cart_size
days_between_orders
user_reordered_products
user_reordered_products_ratio
user_unique_cats
user_reordered_cats
user_reordered_cats_ratio
user_unique_classes
user_reordered_classes
user_reordered_classes_ratio
product_week_reorder_rate
cat_week_reorder_rate
class_week_reorder_rate
p_days_since_prior_order_reorder_rate
t_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
u_p_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_c_days_since_prior_reorder_rate
product_reorder_rate
p_reduced_feat_1
p_reduced_feat_2
p_reduced_feat_3
u_p_order_rate
u_p_reorder_rate
u_p_orders_since_last
u_t_orders_since_last
u_c_orders_since_last
max_streak
max_streak_cat
cat_reorder_rate
u_t_order_rate
u_t_reorder_rate


0.030
0.025
0.020
0.015
0.010
0.005
0.000
days_since_prior_order
day_of_week
user_reorder_rate
user_unique_products
user_total_products
user_avg_cart_size
days_between_orders
user_reordered_products
user_reordered_products_ratio
user_unique_cats
user_reordered_cats
user_reordered_cats_ratio
user_unique_classes
user_reordered_classes
user_reordered_classes_ratio
product_week_reorder_rate
cat_week_reorder_rate
class_week_reorder_rate
p_days_since_prior_order_reorder_rate
t_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
u_p_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_c_days_since_prior_reorder_rate
product_reorder_rate
p_reduced_feat_1
p_reduced_feat_2
p_reduced_feat_3
u_p_order_rate
u_p_reorder_rate
u_p_orders_since_last
u_t_orders_since_last
u_c_orders_since_last
max_streak
max_streak_cat
cat_reorder_rate
u_t_order_rate
u_t_reorder_rate

0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
days_since_prior_order
day_of_week
user_reorder_rate
user_unique_products
user_total_products
user_avg_cart_size
days_between_orders
user_reordered_products
user_reordered_products_ratio
user_unique_cats
user_reordered_cats
user_reordered_cats_ratio
user_unique_classes
user_reordered_classes
user_reordered_classes_ratio
product_week_reorder_rate
cat_week_reorder_rate
class_week_reorder_rate
p_days_since_prior_order_reorder_rate
t_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
u_p_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_c_days_since_prior_reorder_rate
product_reorder_rate
p_reduced_feat_1
p_reduced_feat_2
p_reduced_feat_3
u_p_order_rate
u_p_reorder_rate
u_p_orders_since_last
u_t_orders_since_last
u_c_orders_since_last
max_streak
max_streak_cat
cat_reorder_rate
u_t_order_rate
u_t_reorder_rate

0.5
0.4
0.3
0.2
0.1
0.0
days_since_prior_order
day_of_week
user_reorder_rate
user_unique_products
user_total_products
user_avg_cart_size
days_between_orders
user_reordered_products
user_reordered_products_ratio
user_unique_cats
user_reordered_cats
user_reordered_cats_ratio
user_unique_classes
user_reordered_classes
user_reordered_classes_ratio
product_week_reorder_rate
cat_week_reorder_rate
class_week_reorder_rate
p_days_since_prior_order_reorder_rate
t_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
u_p_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_c_days_since_prior_reorder_rate
product_reorder_rate
p_reduced_feat_1
p_reduced_feat_2
p_reduced_feat_3
u_p_order_rate
u_p_reorder_rate
u_p_orders_since_last
u_t_orders_since_last
u_c_orders_since_last
max_streak
max_streak_cat
cat_reorder_rate
u_t_order_rate
u_t_reorder_rate

0.20
0.15
0.10
0.05
0.00
u_t_reorder_rate
u_t_order_rate
cat_reorder_rate
max_streak_cat
max_streak
u_c_orders_since_last
u_t_orders_since_last
u_p_orders_since_last
u_p_reorder_rate
u_p_order_rate
p_reduced_feat_3
p_reduced_feat_2
p_reduced_feat_1
product_reorder_rate
u_c_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_p_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
t_days_since_prior_order_reorder_rate
p_days_since_prior_order_reorder_rate
class_week_reorder_rate
cat_week_reorder_rate
product_week_reorder_rate
user_reordered_classes_ratio
user_reordered_classes
user_unique_classes
user_reordered_cats_ratio
user_reordered_cats
user_unique_cats
user_reordered_products_ratio
user_reordered_products
days_between_orders
user_avg_cart_size
user_total_products
user_unique_products
user_reorder_rate
day_of_week
days_since_prior_order
0.12
0.10
0.08
0.06
0.04
0.02
0.00
u_t_reorder_rate
u_t_order_rate
cat_reorder_rate
max_streak_cat
max_streak
u_c_orders_since_last
u_t_orders_since_last
u_p_orders_since_last
u_p_reorder_rate
u_p_order_rate
p_reduced_feat_3
p_reduced_feat_2
p_reduced_feat_1
product_reorder_rate
u_c_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_p_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
t_days_since_prior_order_reorder_rate
p_days_since_prior_order_reorder_rate
class_week_reorder_rate
cat_week_reorder_rate
product_week_reorder_rate
user_reordered_classes_ratio
user_reordered_classes
user_unique_classes
user_reordered_cats_ratio
user_reordered_cats
user_unique_cats
user_reordered_products_ratio
user_reordered_products
days_between_orders
user_avg_cart_size
user_total_products
user_unique_products
user_reorder_rate
day_of_week
days_since_prior_order



ویژگی‌های u_p_orders_since_last، u_p_orders_since_last، product_reorder_rate و day_of_week ویژگی‌هایی هستند که غالبا جز ویژگی‌های مهم و تاثیرگذار در مدل‌ها هستند و در طرف دیگر بسیاری از ویژگی‌ها در غالب مدل‌ها تاثیر چندانی ندارند و با حذف آن‌ها می‌توان زمان آموزش و پیشبینی مدل را افزایش داد. با توجه به اینکه پروژه به این صورت است که باید برای هر روز ویژگی‌ها مجددا با داده‌های جدید ساخته شوند و پیشبینی‌ها انجام شود بنابراین زمان اجرای کد مهم است و هر چه بتوانیم آن را کمتر کنیم، به ما در اجرای کد کمک می‌کند اما اگر ویژگی‌ها بیشتر باشند، ممکن است کمک کوچکی به پیشبینی بهتر مدل بکند و همچنین ممکن است باعث بیش برازش مدل شود. به هر حال این امری دیگر است که باید توسط فردی که پروژه را انجام می‌دهد مشخص شود و با تست حالات مختلفی می‌شود به پاسخ بهتر رسید. حال در این بخش به حذف ویژگی نمی‌پردازیم و در بخش آخر این کار را بررسی می‌کنیم.

۸. بررسی و آماده سازی داده‌ها برای آموزش مدل بر پایه‌ی کتگوری

۸.۱. چرا بر اساس کتگوری

برای توضیح بهتری کتگوری باید اشاره شود که محصولاتی که طعم و نوع آن‌ها با هم فرق دارد در کتگوری‌های مختلفی قرار می‌گیرند. به عنوان مثال کتگوری پنیر خامه‌ای با پنیر سفید یا انواع پنیرهای آشپزی متفاوت است یا در آبمیوه‌ها، طعم‌های مختلف در کتگوری‌های متفاوت قرار می‌گیرند، همچنین گاز دار بودن آب‌میوه نیز باعث جدا شدن کتگوری آن با آب میوه‌های دیگر می‌شود. با توجه به تصویر شماره‌ی ؟؟؟؟؟ (صفحه ۱۶) مشتریان به ازای هر کتگوری (که حداقل ۱۰ بار از آن خریده باشند) حدودا ۴ محصول مختلف خریداری می‌کنند. این در حالی است که برخی از کتگوری‌ها مانند انواع مختلف میوه‌ها تنها یک نوع محصول در هر کتگوری وجود دارد. با این حال میانگین ۴ محصول به ازای هر کتگوری عدد بزرگی است که نشان می‌دهد مشتریان غالبا تعصبی بر روی خرید از یک برند خاص از یک محصول ندارند.

از طرف دیگر این امر که مشتریان به یک کالای خاص متعهد نیستند و ممکن است در خریدهای متوالی از برندهای متفاوت خرید کنند روال کار مدل‌های یادگیری ماشین را با اشکال مواجه می‌کند. فرض کنید یک مشتری در تمام ۵ خرید اول آب پرتقال از یک

برند خریده است. این مشتری برای تنوع و یا هر دلیل دیگری ممکن است در خرید ششم از آن برند خریداری نکند و رو به برند دیگری بیاورد. مسلما ما شرایط استثنا را در نظر نمی‌گیریم و انتظار داریم فردی که در تمام خریدها آب پرتقال خریده است، در خرید بعدی نیز آن را در سبد خود داشته باشد. اما تغییر برند مخصوصا با توجه به موارد گفته شده در ابتدای این بخش، امری غیر عادی نیست. در این شرایط یادگیری مدل با این اختلال رو به رو خواهد شد که کالایی که در تمام خریدهای قبلی در سبد قرار داشته، در سبد خرید بعدی قرار ندارد. حال آنکه این طور نیست و مشتری تنها برند مدنظر خود را تعویض کرده است. در نمونه‌ای دیگر فرض کنید که یک مشتری در ۱۰ خرید اول آب پرتقال خریده است اما این خریدها از ۴ شرکت متفاوت بوده است. به این دلیل مدل یادگیری ماشین با وجود اینکه ویژگی‌های مربوط به کتگوری را نیز در خود دارد، درک مناسبی از این قضیه ندارد که مشتری در تمام ۱۰ خرید قبلی خود آب پرتقال را در سبد داشته و نمی‌تواند به درستی تصمیم بگیرد که کدام محصول خریداری خواهد شد، هر چند ممکن است به دلیل اینکه هر محصول تنها در ۲ یا ۳ خرید قرار داشته، هیچ کدام از محصولات را با لیبل ۱ در سبد بعدی پیشبینی نکند. اما اگر پیشبینی تنها بر اساس تایپ باشد این مشکلات برطرف خواهد شد و مدل می‌تواند بخشی از مشکلات خود را به راحتی حل کند.

حال ممکن است که این سوال پیش بیاید که کدام محصول قرار است به مشتری پیشنهاد شود؟ آیا محصول انتخاب شده از کتگوری مدنظر می‌تواند در سبد خرید مشتری قرار بگیرد یا خیر؟ در ابتدا باید به این نکته توجه کنیم که ما تشخیص دهیم که مشتری به کالایی در یک کتگوری خاص نیاز دارد، حال اگر برای یک کالا در این کتگوری تخفیفی خاص برای این مشتری در نظر بگیریم همین امر موجب آن خواهد شد که مشتری برای خرید آن محصول خاص ترغیب شود با این حال ما سعی می‌کنیم محصولی را به مشتری پیشنهاد دهیم که بیشترین تکرار را در سبد خرید او داشته است و یا آخرین بار از آن برند خرید کرده است. اما اگر مجددا کمی از عقب‌تر نگاه کنیم هدف ما این است که مشتری را به سمت خرید مجدد سوق دهیم. اگر مشتری به برند دیگر به نحوی علاقه داشته باشد که در هر صورت آن را ترجیح می‌دهد، با یادآوری ما برای آن کتگوری، کالای مورد نظر خود از آن کتگوری را خریداری می‌کند و مشکلی زیادی ممکن نیست از این طریق برای هدف ما ایجاد شود. البته به هر حال باید توجه کرد که محصولی که انتخاب می‌کنیم حتی‌الامکان از خریدهای قبلی مشتری و یا محصولی که به صورت کلی پرفروش است باشد.

۸.۲. تغییرات نسبت به حالت قبل

برای اینکه بتوانیم بر اساس کتگوری مدل را بسازیم باید توجه کنیم که باید تمام دیتاست را به نحوی تغییر دهیم که ردیف‌ها خرید یک کتگوری را نشان دهند، نه خرید یک محصول. بنابراین کاری که باید بکنیم این است که ابتدا داده‌های مربوط به خرید محصولات مختلف از یک کتگوری را در یک سبد با یکدیگر جمع کنیم، یعنی به عنوان مثال اگر در یک سبد خرید ۱ چیپس لیمویی از شرکت x و ۲ چیپس لیمویی از شرکت y خریداری شده است، در دیتاست جدید به اینگونه خواهد بود که ۳ چیپس لیمویی خریداری شده است. پس از آن باید داده‌های مربوط به آیتم‌ها را حذف کنیم زیرا در این دیتاست برای ما اهمیتی ندارند. به عنوان مثال شناسه‌ی محصول و یا ویژگی‌هایی که در مورد محصول است دیگر نیازی به ساختن آن‌ها نیست.

پس از انجام آنچه گفته شد، به ساختن ویژگی‌هایی که بدون ارتباط به خود محصول می‌باشند و به کمک زمان، کتگوری و یا کلاس به دست می‌آیند اقدام می‌کنیم. در این حالت تعداد ویژگی‌های ما از از ۳۹ به ۲۸ کاهش می‌یابد.

۸.۳. فیچرهای جدید

در حالتی که بر اساس کتگوری قضاوت می‌کنیم، این امکان وجود دارد که کالاهایی که با یکدیگر یکسان در نظر بگیریم که حجم آن‌ها با یکدیگر متفاوت باشد، به عنوان مثال مایع دستشویی با برند، نوع و رایحه‌ی یکسان ممکن است در دو وزن دو لیتری و چهار

لیتری فروخته شوند اما این دو محصول در مدل جدید یکسان در نظر گرفته می‌شوند. به کمک پردازش زبان طبیعی[۵۳] می‌توان از درون اسم کالا وزن و یا حجم آن‌ها را تشخیص داد اما این کار لازم به انجام یک پروژه‌ی دیگر دارد و مساله‌ی آن در این پروژه نمی‌گنجد. بنابراین برای تخمین حدودی حجم و یا وزن کالا از ستون قیمت آن که تا کنون از آن استفاده نمی‌کردیم استفاده کنیم. البته این کار دقیق نیست اما به عنوان مثال بسته‌های ۱۲ عددی تخم مرغ شرکت‌های مختلف معمولا قیمت‌های نزدیک به هم دارند و بسته‌های ۳۰ عددی قیمتی حدودا دو و نیم برابر آنان دارند و می‌توان به صورت حدودی برای پیشبینی بهتر آن‌ها را به ویژگی‌ها اضافه کرد. سه ویژگی برای افزوده شدن به ۲۹ ویژگی قبلی انتخاب شده اند که به شرح زیر می‌باشند:

- ave_price_day_ratio: این ویژگی نشان می‌دهد که به طور میانگین در بین تمامی مشتریان، این کتگوری روزانه چند هزار تومان مصرف می‌شود. روش محاسبه‌ی آن این گونه است که اگر یک مشتری به عنوان مثال در خرید k ام کتگوری x را به اندازه‌ی ۹۰ هزارتومان خریده است و مجددا ۱۸ روز بعد از آن کتگوری خرید کرده است، این مشتری در این دفعه از خرید روزانه ۵ هزارتومان از آن کالا را مصرف کرده است. حال اگر میان کل خریدها این نسبت را محاسبه کنیم، به ویژگی اشاره شده می‌رسیم.
- user_ave_price_day_ratio: این ویژگی همان ویژگی بالاست با این تفاوت که تنها برای همان مشتری محاسبه می‌شود. در واقع این ویژگی نشان می‌دهد که مشتری مدنظر به طور میانگین چند تومان از آن محصول مصرف می‌کند.
- user_days_price_ratio_since_prior: این ویژگی نشان می‌دهد که از آخرین باری که مشتری از آن کتگوری خرید کرده است، این نسبت چقدر است. یعنی این نسبت از تقسیم مقدار هزینه شده برای این کتگوری که مشتری در آخرین باری که از آن خرید کرده است به تعداد روزی که از آن زمان گذشته است به دست می‌آید. منطقی است که این نسبت هر چه به ویژگی قبلی نزدیک‌تر باشد، احتمال خرید افزایش می‌یابد. در روزهای ابتدایی که مشتری به تازگی از کتگوری مدنظر خرید کرده است، این ویژگی بزرگ است و پس از گذشت زمان کوچک و کوچک تر می‌شود و انتظار داریم وقتی عدد آن نزدیک به عدد قبلی شد، مشتری مجددا از آن محصول خرید کند.

[53] Natural Language Processing

۹. بررسی نتایج مدل بر پایه‌ی کتگوری و مقایسه‌ی نتایج آن

۹.۱. گزارش کلاس‌بندی

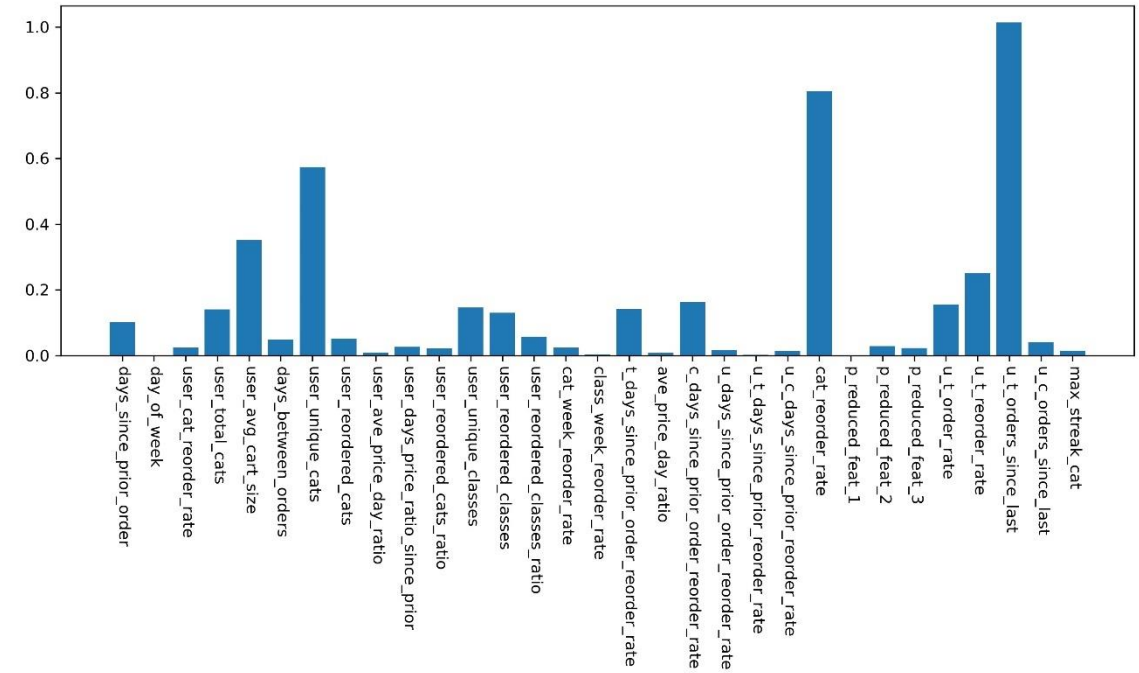
با توجه اینکه مدل جدید را بر اساس کتگوری ایجاد کردیم، همانطور که گفته شد لیبل ما این است که مشتری از آن کتگوری در سبد بعدی خود داشته است یا خیر و معیارها بر اساس آن گزارش می‌شوند. در این حالت تعداد ردیف‌ها از دو میلیون و شصت هزار به یک میلیون و پانصد هزار کاهش یافته و همچنین با وجود افزودن سه ویژگی جدید، تعداد ویژگی‌ها از ۳۹ به ۳۱ کاهش پیدا کرده است که موجب کم شدن زمان مورد نیاز برای آموزش و پیشبینی مدل و حجم مدل ذخیره شده می‌شود. همچنین این به کمک این روش زمان ساخت فیچرها که باید هر روزه انجام شود کاهش می‌یابد اما این بهبودها زمانی موثر خواهند بود که کارایی مدل افزایش پیدا کند و یا همان میزان بماند. البته در مواردی که زمان پیشبینی و حجم مدل و داده‌ها برای ذخیره سازی اهمیت زیادی دارد، ممکن است با وجود کم شدن اندک کارایی مدل، همچنان روش جدید برای انجام پروژه به کار گرفته شود. به کمک ۷ مدلی که در بخش قبل پیشبینی و ارزیابی کارایی آنان را انجام دادیم، در این قسمت نیز داریم:

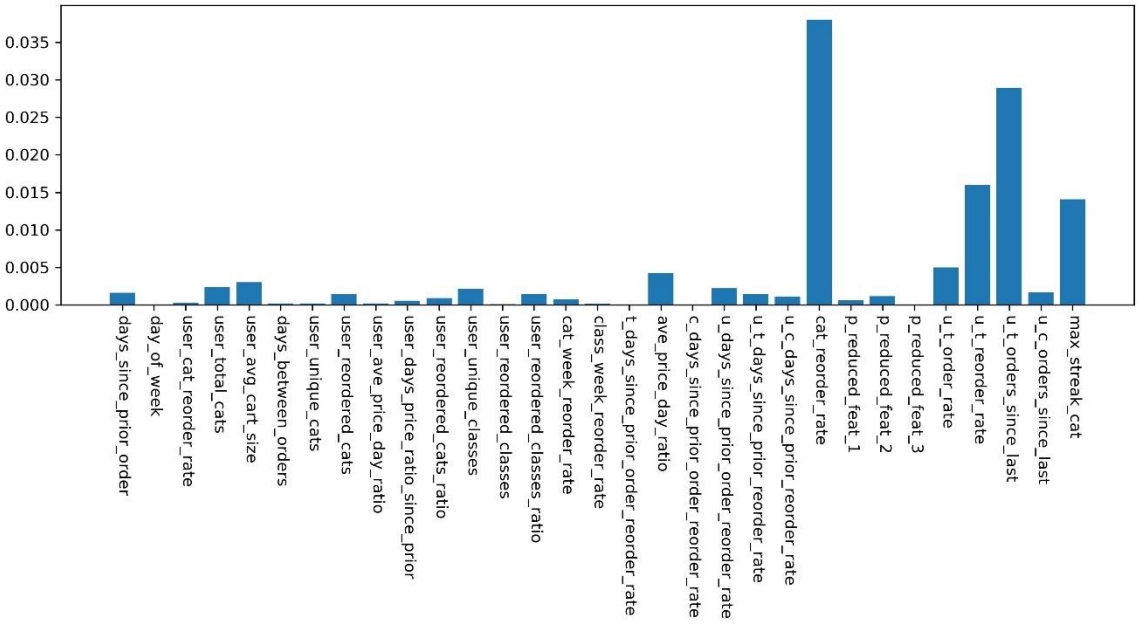
Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic regression	0.82	0.34	0.57	0.43
Gaussian Naïve Bayes	0.80	0.30	0.51	0.38
Ada Boost	0.79	0.31	0.64	0.41
Decision Tree	0.81	0.27	0.35	0.31
Random Forest	0.84	0.39	0.57	0.46
Extra Trees	0.85	0.40	0.49	0.44
XGBoost	0.85	0.40	0.53	0.46

حال همانطور که مشخص است بیشترین F1-score متعلق به مدل‌های جنگل تصادفی و XGBoost است. بنابر آنچه در بخش ۱-۷ گفته شد، با توجه به مدل مسئله بهتر است برای انتخاب مدل بهتر پس از F1-score به recall آن نگاه کنیم که در این صورت الگوریتم جنگل تصادفی به عنوان مدل بهتر انتخاب می‌شود. اگر به این جدول و جدول قبلی نگاه کنیم متوجه آن خواهیم شد که دقت مدل‌ها کاهش یافته است و دلیل آن این است که در دیتاست جدید درصد نمونه‌هایی که لیبل صفر دارند از ۹۴ درصد به ۸۸ درصد کاهش یافته است (با توجه به ادغام محصولات یک کتگوری با یکدیگر احتمال بازخرید آن افزایش پیدا کرده است.) و در نتیجه مدل که با توجه به کم بودن درصد واقعی کلاس‌های ۱ با احتمال بیشتری کلاس یک نمونه را برابر با صفر پیشبینی می‌کند، در حالت جدید به دلیل زیادتر بودن کلاس‌های ۱ معیار Accuracy کمتری خواهد داشت. اما با توجه به آنچه ما به عنوان معیار ارزیابی در نظر داشتیم، F1-score مدل از ۰.۳۹ به ۰.۴۶ افزایش یافته است که افزایش ۱۸ درصدی در این معیار را نشان می‌دهد. همانطور که بخش ۱-۷ گفته شد این معیار یک میانگین توافقی است که بالا بردن آن لازمه‌ی بهبود مدل در هر دو جهت مدنظر ماست و از نشان دادن بهبود کاذب در مدل جلوگیری می‌کند.

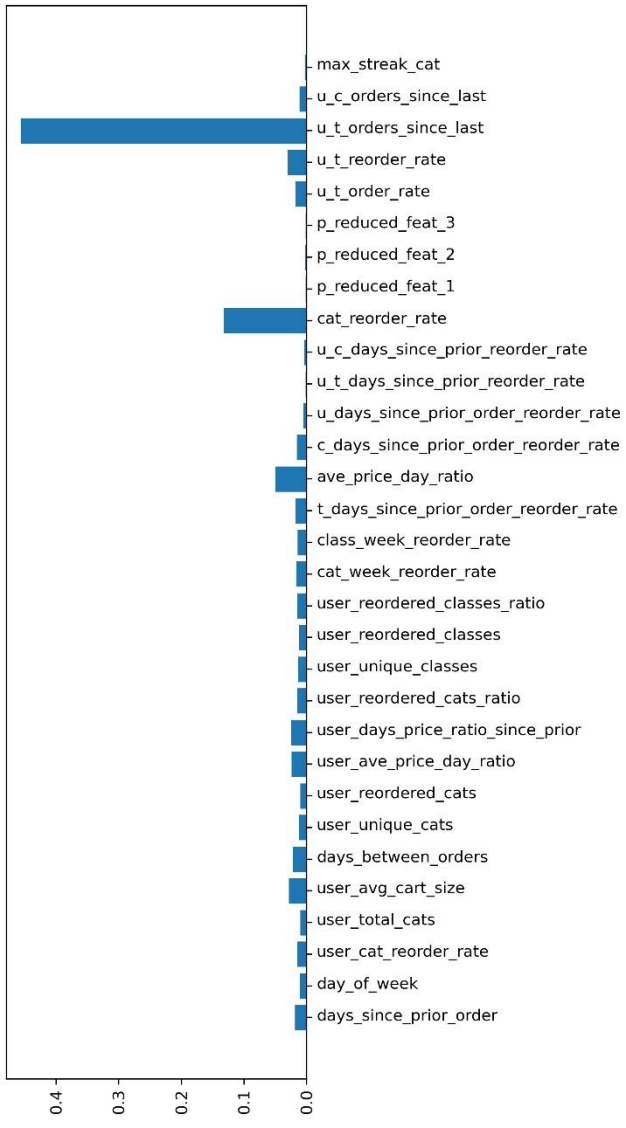
۹.۲. اهمیت ویژگی

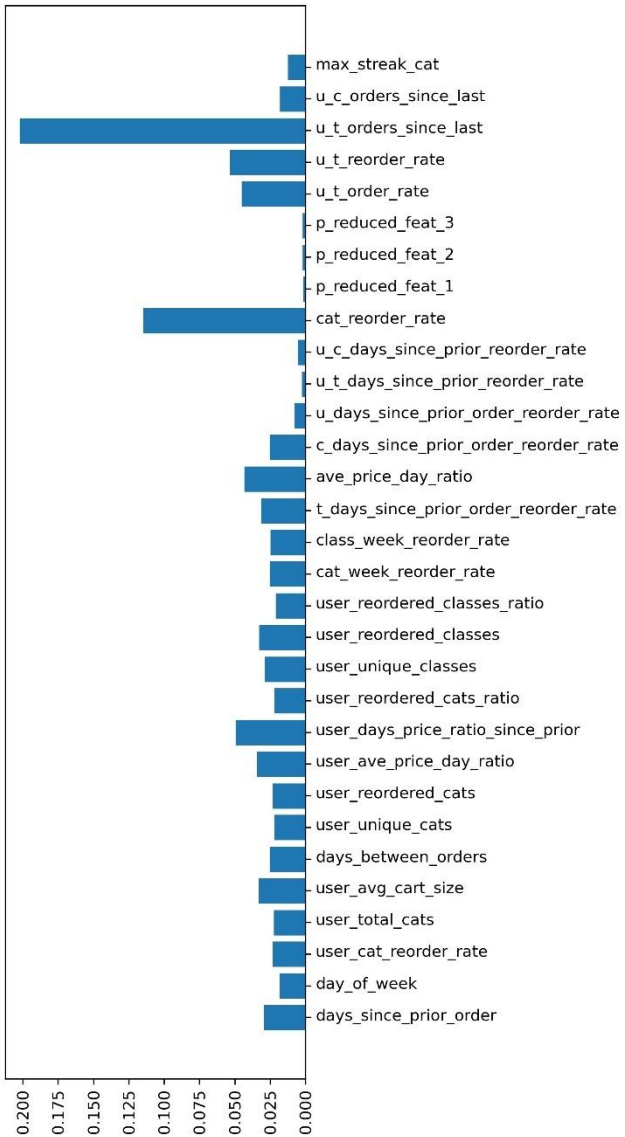
در این قسمت می‌توانیم ببینیم با توجه به اینکه ۱۱ فیچر حذف شده است و ۳ فیچر جدید نیز اضافه شده است، وضعیت اهمیت ویژگی‌ها برای هر مدل به چه گونه است. مانند بخش قبل نمودارهای در مدل‌های مختلف با مقیاس یکسان رسم نشده‌اند و هدف ما تنها مقایسه‌ی ویژگی‌های هر مدل با دیگر ویژگی‌های همان مدل است.

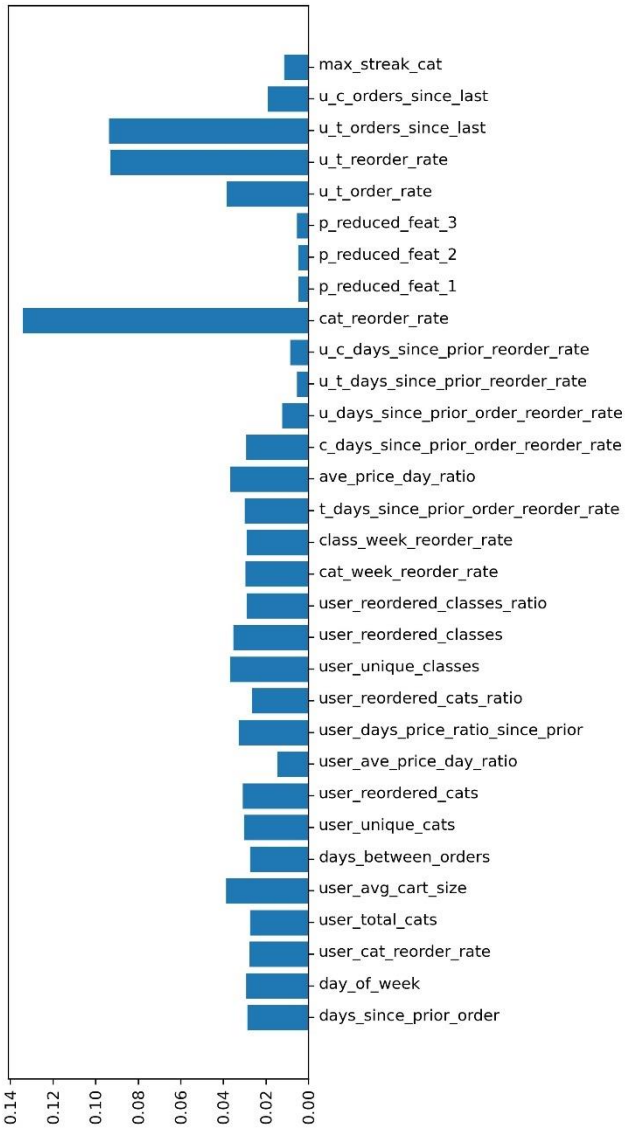

1.0
0.8
0.6
0.4
0.2
0.0
days_since_prior_order
day_of_week
user_cat_reorder_rate
user_total_cats
user_avg_cart_size
days_between_orders
user_unique_cats
user_reordered_cats
user_ave_price_day_ratio
user_days_price_ratio_since_prior
user_reordered_cats_ratio
user_unique_classes
user_reordered_classes
user_reordered_classes_ratio
cat_week_reorder_rate
class_week_reorder_rate
t_days_since_prior_order_reorder_rate
ave_price_day_ratio
c_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
u_t_days_since_prior_reorder_rate
u_c_days_since_prior_reorder_rate
cat_reorder_rate
p_reduced_feat_1
p_reduced_feat_2
p_reduced_feat_3
u_t_order_rate
u_t_reorder_rate
u_t_orders_since_last
u_c_orders_since_last
max_streak_cat

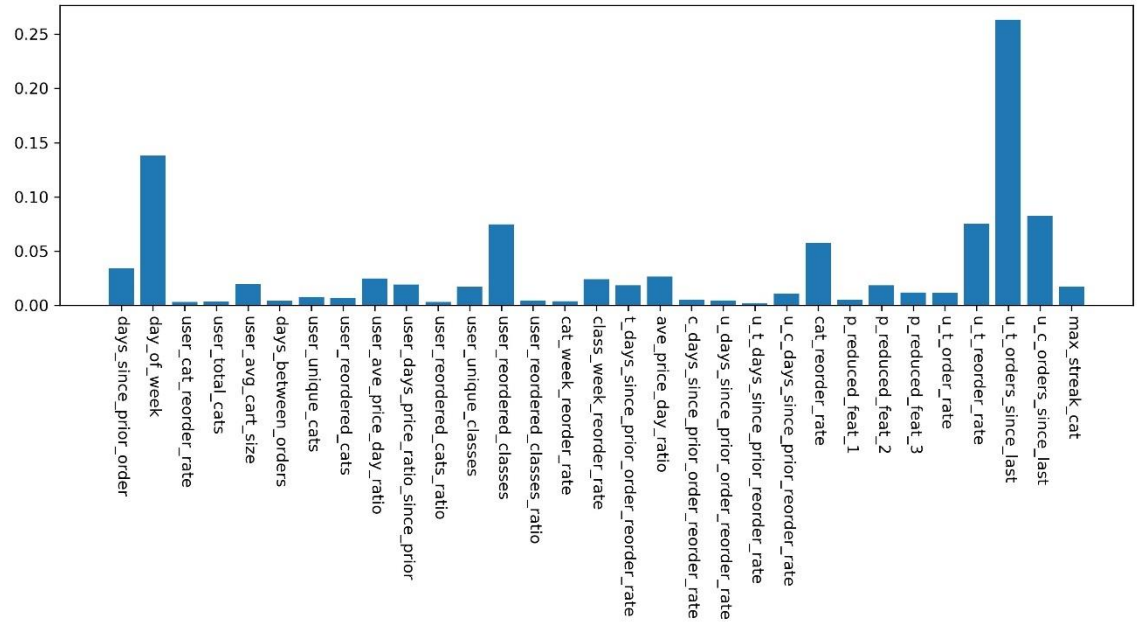

0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
days_since_prior_order
day_of_week
user_cat_reorder_rate
user_total_cats
user_avg_cart_size
days_between_orders
user_unique_cats
user_reordered_cats
user_ave_price_day_ratio
user_days_price_ratio_since_prior
user_reordered_cats_ratio
user_unique_classes
user_reordered_classes
user_reordered_classes_ratio
cat_week_reorder_rate
class_week_reorder_rate
t_days_since_prior_order_reorder_rate
ave_price_day_ratio
c_days_since_prior_order_reorder_rate
u_days_since_prior_order_reorder_rate
u_t_days_since_prior_reorder_rate
u_c_days_since_prior_reorder_rate
cat_reorder_rate
p_reduced_feat_1
p_reduced_feat_2
p_reduced_feat_3
u_t_order_rate
u_t_reorder_rate
u_t_orders_since_last
u_c_orders_since_last
max_streak_cat


max_streak_cat
u_c_orders_since_last
u_t_orders_since_last
u_t_reorder_rate
u_t_order_rate
p_reduced_feat_3
p_reduced_feat_2
p_reduced_feat_1
cat_reorder_rate
u_c_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
ave_price_day_ratio
t_days_since_prior_order_reorder_rate
class_week_reorder_rate
cat_week_reorder_rate
user_reordered_classes_ratio
user_reordered_classes
user_unique_classes
user_reordered_cats_ratio
user_days_price_ratio_since_prior
user_ave_price_day_ratio
user_reordered_cats
user_unique_cats
days_between_orders
user_avg_cart_size
user_total_cats
user_cat_reorder_rate
day_of_week
days_since_prior_order
0.25
0.20
0.15
0.10
0.05
0.00


max_streak_cat
u_c_orders_since_last
u_t_orders_since_last
u_t_reorder_rate
u_t_order_rate
p_reduced_feat_3
p_reduced_feat_2
p_reduced_feat_1
cat_reorder_rate
u_c_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
ave_price_day_ratio
t_days_since_prior_order_reorder_rate
class_week_reorder_rate
cat_week_reorder_rate
user_reordered_classes_ratio
user_reordered_classes
user_unique_classes
user_reordered_cats_ratio
user_days_price_ratio_since_prior
user_ave_price_day_ratio
user_reordered_cats
user_unique_cats
days_between_orders
user_avg_cart_size
user_total_cats
user_cat_reorder_rate
day_of_week
days_since_prior_order
0.4
0.3
0.2
0.1
0.0

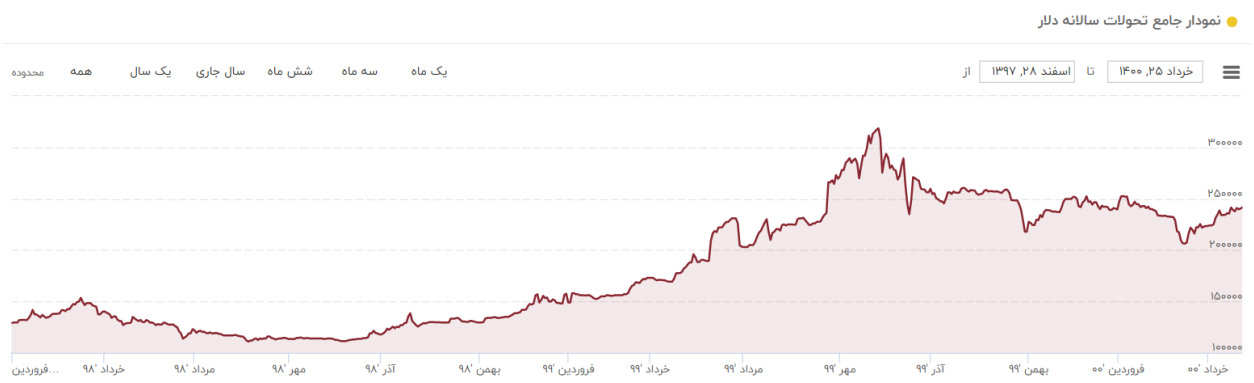

max_streak_cat
u_c_orders_since_last
u_t_orders_since_last
u_t_reorder_rate
u_t_order_rate
p_reduced_feat_3
p_reduced_feat_2
p_reduced_feat_1
cat_reorder_rate
u_c_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
ave_price_day_ratio
t_days_since_prior_order_reorder_rate
class_week_reorder_rate
cat_week_reorder_rate
user_reordered_classes_ratio
user_reordered_classes
user_unique_classes
user_reordered_cats_ratio
user_days_price_ratio_since_prior
user_ave_price_day_ratio
user_reordered_cats
user_unique_cats
days_between_orders
user_avg_cart_size
user_total_cats
user_cat_reorder_rate
day_of_week
days_since_prior_order
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000


max_streak_cat
u_c_orders_since_last
u_t_orders_since_last
u_t_reorder_rate
u_t_order_rate
p_reduced_feat_3
p_reduced_feat_2
p_reduced_feat_1
cat_reorder_rate
u_c_days_since_prior_reorder_rate
u_t_days_since_prior_reorder_rate
u_days_since_prior_order_reorder_rate
c_days_since_prior_order_reorder_rate
ave_price_day_ratio
t_days_since_prior_order_reorder_rate
class_week_reorder_rate
cat_week_reorder_rate
user_reordered_classes_ratio
user_reordered_classes
user_unique_classes
user_reordered_cats_ratio
user_days_price_ratio_since_prior
user_ave_price_day_ratio
user_reordered_cats
user_unique_cats
days_between_orders
user_avg_cart_size
user_total_cats
user_cat_reorder_rate
day_of_week
days_since_prior_order
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00



همانطور که در نمودارهای فوق مشخص است، می‌توان گفت پنج ویژگی p_reduced_feat_1، p_reduced_feat_2، p_reduced_feat_3، u_c_days_since_prior_reorder_rate و u_t_days_since_prior_reorder_rate تقریبا در تمام مدل‌ها اثر گذاری خیلی پایینی دارند و به نظر نمی‌رسد با حذف آن‌ها تغییری در مدل حاصل شود. در حالت قبل به جز day_of_week تمام ویژگی‌هایی که تاثیر بالایی داشتند مربوط به محصول خاص بودند که از مدل حذف شده‌اند اما به جای آن‌ها در مدل‌های جدید ویژگی‌های u_t_orders_since_last، u_t_reorder_rete و cat_reorder_rate اهمیت بالایی دارند. البته با توجه به اینکه در هر دو حالت مدل‌های جنگل تصادفی و XGBoost بهترین پاسخ را ارائه دادند، می‌توان تنها اهمیت ویژگی در این دو مدل را بررسی کرد و برای بهبودهای بعدی تنها بر روی آموزش این دو مدل تمرکز کرد.

در ادامه به ویژگی‌هایی که جدیدا به مدل اضافه کردیم می‌پردازیم. با توجه به نمودارها هیچ کدام از این ویژگی‌ها از ویژگی‌های مهم مدل‌ها نیستند. البته میزان تاثیرگذاری آنان به قدری کم نیست که بتوان به راحتی گفت حذف آن‌ها بدون تاثیر منفی بر روی مدل‌ها خواهد بود. دلیل کم بودن تاثیر این ویژگی می‌تواند این باشد، که در طول زمان قیمت‌ها دسخوش تغییرات زیادی شده اند. موارد جزئی تغییر قیمت که بر اساس سیاست‌های خاص یک شرکت و یا تخفیفات مناسبتی به وجود می‌آید تاثیر ناچیزی بر روی مدل خواهد داشت اما اگر این تغییر عظیم باشد باعث اختلال در کار مدل می‌شود. به طوری که به عنوان مثال مشتری در ابتدا روزانه ۵۰۰ تومان از یک کتگوری خاص مصرف می‌کند اما پس از مدتی همین مشتری برای همان مصرف روزانه مجبور می‌شود ۸۰۰ تومان از آن کالا به طور روزانه مصرف کند که این امر موجب ایجاد اشکال در محاسبه‌ی ویژگی‌ها و آموزش مدل می‌شود. اگر این اختلالات به صورت کلی اتفاق بیافتد و تمام محصولات را تحت تاثیر قرار دهد مسلما موجب آن خواهد شد که مدل نتواند آن طور که باید و شاید از ویژگی‌های جدید بهره ببرد. البته این ویژگی احتمالا در بیشتر کشورهای جهان و بیشتر صنعت‌ها بتواند موثر باشد اما در این داده‌ها مربوط به کسب و کاری در ایران است که قیمت‌ها در آن به شدت مورد توجه تورم است. ممکن است قیمت دلار شاخصی باشد که بتوان به کمک آن درک بهتری از نحوه‌ی افزایش قیمت‌ها داشت. در نمودار ؟؟؟؟(صفحه ۴۲) که تغییرات قیمت دلار به ریال در بازه‌ی زمانی داده‌های دیتاست است می‌توانیم ببینیم که قیمت دلار در این زمان در بازه‌ی ۱۲ تا ۳۲ هزارتومان متغیر بوده است که نشان می‌دهد انتظار اینکه قیمت کالاها در این بازه ثبات نسبی داشته باشد انتظار اشتباهی است.



۹.۳. ویژگی‌های نسبی

آنچه از مدل‌های ساده‌ی رگرسیون و یا کلاس‌بندی در خاطر داریم این است که فیچرهایی که افزایش یا کاهش آن‌ها مستقلاً موجب افزایش یا کاهش مقدار پاسخ و یا احتمال انتخاب یک کلاس می‌شود برای ما مهم خواهند بود. بنابراین ویژگی‌های مثل روزهای عبور کرده از خرید قبلی، تعداد کتگوری متمایز خریده شده توسط مشتری، میانگین فاصله‌ی زمانی بین خرید دوباره‌ی یک کتگوری و یا نسبت قیمت به روز یک کالا نمی‌تواند در آن مدل‌های یادگیری ماشین، ویژگی‌های مهمی باشد و باید به صورتی که گفته شد در بیایند. حالا در دو مدلی که بهتری پاسخ‌ها را تا به اینجا داده‌اند (جنگل تصادفی و XGBoost) ویژگی‌هایی با این مشخصه وارد می‌کنیم تا تاثیر آن بر نتایج را مشاهده کنیم. برخی از ویژگی‌ها مانند درصد دفعاتی که کالا بازخرید شده، درصد کالاهای بازخرید شده و یا حتی ویژگی‌هایی مانند max_streak که نشان می‌دهد بیشینه تعداد سبدهای متوالی یک فرد که آن کالا در آن‌ها قرار داشته است چه مقداری است، از ویژگی‌هایی هستند که افزایش و کاهش آنان می‌تواند تاثیر مستقیمی در احتمال یک گزینه داشته باشند. اما اگر به یک ویژگی مانند تعداد روزی که از خرید آخر گذشته است نگاه کنیم، تاثیر این ویژگی وابسته به آن است که به طور میانگین فاصله‌ی بین خریدهای آن فرد چند روز است، میانگین فاصله‌ی بین دو خرید از آن کتگوری چند روز است و میانگین فاصله‌ی بین دو خرید از آن کتگوری برای آن شخص چند روز است. یا اگر ویژگی نسبت قیمت به تعداد روزی که از آن گذشته زمانی معنا پیدا می‌کند که آن را نسبت به ویژگی‌های "میانگین نسبت قیمت به روز برای آن کتگوری"و " میانگین نسبت قیمت به روز آن کتگوری برای آن شخص" معنی پیدا می‌کند. حال اگر ویژگی‌هایی که نسبت این ویژگی‌ها را بیان کند داشته باشیم به نظر می‌توانند به بهبود مدل کمک کنند. به این منظور ویژگی‌های زیر را ایجاد می‌کنیم.

- days_between_cat_orders: میانگین فاصله‌ی بین دو سفارش را برای هر کتگوری نشان می‌دهد. (میان کل مشتریان)
- days_between_user_cat_orders: میانگین فاصله‌ی بین دو سفارش را برای هر کتگوری نشان می‌دهد. (برای همان مشتری)
- since_prior_days_cat_ratio: نسبت تعداد روز گذشته از سفارش قبلی به ویژگیِ days_between_cat_orders نشان می‌دهد.
- since_prior_days_ratio: نسبت تعداد روز گذشته از سفارش قبلی به میانگین فاصله‌ی بین دو سفارش آن شخص نشان می‌دهد.
- since_prior_days_user_cat_ratio: نسبت تعداد روز گذشته از سفارش قبلی به ویژگیِ days_between_cat_orders نشان می‌دهد.

- user_DPR_tot_ratio: نسبت ویژگی user_days_price_ratio_since_prior به ویژگی ave_price_day_ratio را نشان می‌دهد.
- user_DPR_user_ratio: نسبت ویژگی user_days_price_ratio_since_prior به ویژگی user_ave_price_day_ratio نشان می‌دهد.
- user_unique_cat_ratio: نشان می‌دهد چند درصد از کالاهای یک مشتری کالاهایی هستن که برای بار اول خریده شده اند.

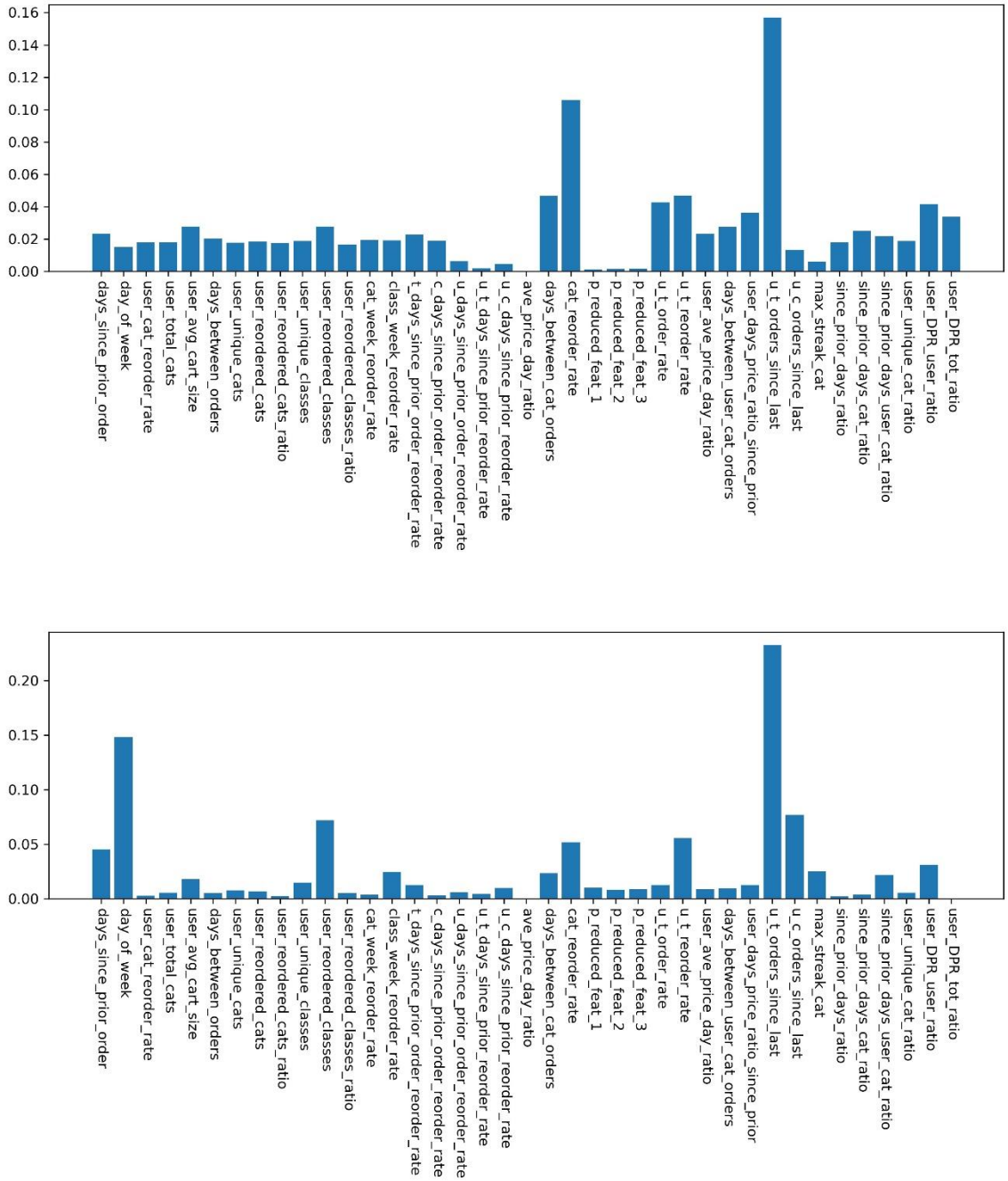
حال اگر فرض ما درست باشد با این ۸ ویژگی جدیدی که به مدل اضافه کرده‌ایم انتظار داریم که معیار مدنظر ما یعنی f1-score بهبود داشته باشد. با اجرا کردن مجدد مدل‌های جنگل تصادفی و XGBoost در حالت جدید داریم:

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Random Forest	0.86	0.41	0.52	0.46
XGBoost	0.85	0.41	0.53	0.46

این در حالی است که در بخش قبل داشتیم:

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Random Forest	0.84	0.39	0.57	0.46
XGBoost	0.85	0.40	0.53	0.46

همانطور که مشخص است معیار اصلی ما تغییری نکرده است. و افزودن ویژگی‌های نسبی جدید تاثیر زیادی در کارایی مدل نداشته است. دلیل اصلی این مورد، پیشرفته بودن مدل‌های جنگل تصادفی و XGBoost است که باعث می‌شود بخش زیادی از حالات ممکن در نظر گرفته شوند و اضافه کردن یک ویژگی که از دو ویژگی دیگر به دست آمده اند تاثیر زیادی در پاسخ‌های مدل نداشته باشد. ممکن بود اگر به یک مدل ساده‌ی رگرسیون یا کلاس‌بندی یک ویژگی اینچنینی اضافه کنیم موجب افزایش کارایی مدل شود اما در مدل‌های پیشرفته این کار تاثیر زیادی ندارد و اگر دو ویژگی قبلی اهمیت کمی داشته باشند، ویژگی جدید که به کمک آن‌ها ایجاد شده است نیز اهمیت زیادی نخواهد داشت. نمودار اهمیت ویژگی برای مدل‌های اجرا شده در تصاویر؟؟؟؟؟ (صفحه۴۴) آمده است.



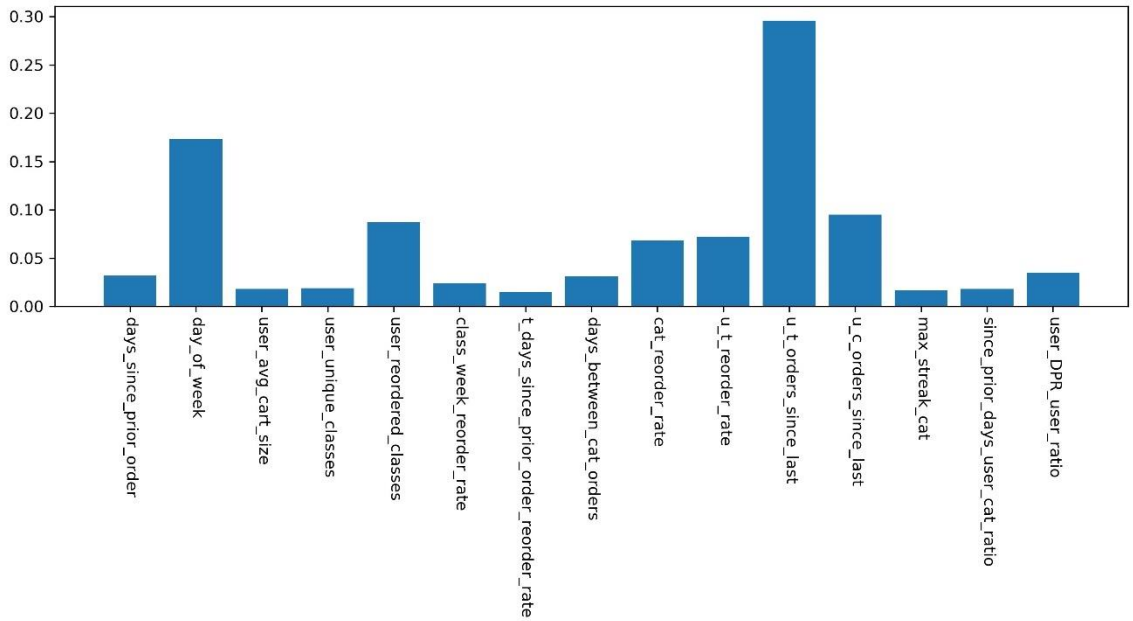
حال که تعداد ویژگی‌ها به ۳۹ رسیده است و بهبودی در کارایی مدل ایجاد نشد و همانطور که در تصویر مشخص است برخی از ویژگی‌ها به نسبت تاثیر بسیار کمی در مدل دارند، اقدام بعدی برای بهتر کردن مدل آن است که با همین دقت در پاسخگویی، مدت زمان آماده سازی دیتاست، آموزش مدل و پیشبینی آن کاهش یابد. در این حالت می‌توان به کمک کاهش ابعاد[۵۴] از طریق حذف ویژگی‌های کم اهمیت به این سمت حرکت کرد. اگر بخواهیم برای مدل XGBoost این کاهش ابعاد را انجام دهیم با توجه به

[54] Dimension Reduction

نمودار اهمیت ویژگی‌های آن (تصویر دوم صفحه ۴۴ ؟؟؟؟؟؟؟؟؟) ویژگی‌هایی که اهمیت آن‌ها تقریبا ناچیز است را حذف می‌کنیم. در حالت جدید معیارهای ارزیابی ما برای مدل XGBoost به شکل زیر خواهد بود.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
XGBoost	0.85	0.40	0.53	0.46

همچنین اهمیت ویژگی‌ها در حالت جدید به شکل زیر خواهد بود.



همانطور که مشاهده می‌شود با وجود کاهش ویژگی‌ها از ۳۹ به ۱۵ همچنان مدل تقریبا با همان دقت پاسخ می‌دهد. البته ممکن است در برخی موارد ویژگی‌های زیاد موجب بیش برازش مدل شوند و با کاهش دادن ویژگی‌ها بتوان دقت مدل را بر روی داده‌های تست را افزایش داد. ویژگی‌هایی که در نهایت تاثیر زیادی در به پاسخ رسیدن مدل داشتند را می‌توانید در تصویر؟؟؟؟؟ (بالا) مشاهده کنید.

۱۰. جمع بندی

در یک پروژه‌ی یادگیری ماشین که دیتاست خام آن در دست است، آنچه توسط دانشمند داده باید مشخص شود نحوه‌ی تعریف مسئله، ایجاد ویژگی‌ها، انتخاب مدل‌ها، انتخاب ابر پارامترها[55] و در نهایت اقداماتی مانند کاهش ابعاد برای کاهش زمان اجرای کد و عدم بیش برازش مدل است.

هدف اصلی این پروژه بیرون کشیدن و ساخت ویژگی‌هایی از دیتاست داده شده بود که به کمک آن‌ها بتوان به وسیله‌ی الگوریتم‌های یادگیری ماشین به آنچه هدف نهایی ما بود دست پیدا کرد. در کنار آن نحوه‌ی انتخاب لیبل که در واقع چارچوب کار یادگیری ماشین را مشخص می‌کند و روش‌هایی برای افزایش دقت مدل و کاهش زمان آن به کار بردیم.

در ساخت ویژگی‌ها به کمک مصورسازی و مرتبط کردن سبد خرید یک فرد به خریدهای پیشین همان مشتری و مشتریان دیگر سعی کردیم تا جای ممکن ویژگی‌هایی که موجب پیشبینی بهتر توسط مدل‌ها می‌شوند را شناسایی و ایجاد کنیم. این کار در حالتی که جمع آوری داده‌ها از قبل برای انجام این پروژه‌ی یادگیری ماشین نبوده باشد ساخت و انتخاب ویژگی‌ها از اهمیت بیشتری برخوردار خواهد بود.

در مرحله‌ی اول ۳۹ ویژگی برای مدل‌ها ایجاد کردیم و بر اساس هر محصول خاص پیشبینی را انجام دادیم.

در مرحله‌ی بعد به دلایل ذکر شده در بخش ۱-۸ پیشبینی را بر اساس کتگوری محصول انجام دادیم. با افزودن ۳ ویژگی دیگر و حذف ۱۱ ویژگی که مربوط به محصول بودند تعداد ویژگی‌ها را به ۳۱ رساندیم و توانستیم معیار F1-score را ۱۸ درصد افزایش دهیم. البته فرض را بر آن گذاشتیم که پاسخ داده شده در این بخش نیز می‌تواند در صورت یکسان بودن همان نتیجه‌ی حالت قبل را به ما بدهد. البته برای این کار می‌توان AB-تست مناسب انجام داد و پاسخ این سوال را گرفت که اگر محصول x متعلق به کتگوری y، نتیجه‌ی مثبتی در مدل "بر پایه‌ی محصول" بدهد، پیشنهاد آن به مشتری کارایی مشابهی با حالتی خواهد داشت که کتگوری y نتیجه‌ی مثبتی در مدل "بر پایه‌ی کتگوری" گرفته باشد و با توجه به سیاست‌های انتخاب محصول، یک محصول را به مشتری پیشنهاد شود یا خیر. در اهمیت ویژگی‌های این مدل دیدیم که سه فیچر اضافه شده تاثیر زیادی بر روی پیشبینی مدل نداشتند. در این قسمت هم بهترین F1-score متعلق به

در بخش بعد این مسئله را بررسی کردیم که برخی از ویژگی‌ها برای تعیین احتمال خرید یک کتگوری، در نسبت با ویژگی دیگری معنا پیدا می‌کنند بنابراین ویژگی‌های جدیدی به کمک ویژگی‌های قبلی ایجاد کردیم که در نهایت تعداد ویژگی‌ها مجددا به ۳۹ رسید اما در معیارهایی که برای سنجش کارایی مدل به کار می‌بردیم تغییر زیاد ایجاد نشد که این امر نشان از آن می‌داد که در مدل‌های پیشرفته افزودن ویژگی‌هایی که از ترکیب ویژگی‌های قبلی به دست آمده باشند تفاوت چندانی در پاسخگویی مدل ایجاد نمی‌کند و تاثیری که ممکن بوده است داشته باشند توسط مدل در حالت قبل محاسبه شده است.

در بخش انتهایی با توجه به اهمیت ویژگی‌های مدل بهینه‌ی بخش قبل (XGBoost)، ویژگی‌های کم اهمیت مدل را حذف کردیم و تعداد ویژگی‌ها از ۳۹ به ۱۵ رسید و دیدیم مدل همچنان با همان دقت پاسخگویی از خود نشان می‌دهد. در نهایت ۱۵ ویژگی که در تصویر ؟؟؟؟؟ (صفحه ۴۵) آمده است به عنوان ویژگی‌های نهایی، مدل XGBoost (با توجه به زمان کمتر آموزش نسبت به جنگل تصادفی) به عنوان مدل بهینه و روش "بر پایه‌ی کتگوری" به عنوان روش بهتر برای انجام این پروژه انتخاب می‌شوند. مشخص

[55] Hyperparameter

است که بر خلاف آن چه در طول انجام پروژه رخ داد، در مراحل بعد نیاز به محاسبه‌ی تمام ویژگی‌ها نداریم و تنها ۱۵ ویژگی انتخاب شده در بخش آخر را برای اجراهایی که در آینده انجام خواهیم داد محاسبه می‌کنیم.

۱۱. منابع