

دانشگاه صنعتی خواجه نصیرالدین طوسی  
دانشکده مهندسی برق - کروه مهندسی کنترل

# درس یادگیری ماشین

## پیش بینی ریزش مشتریان یک فروشگاه خرد

### فروشی

نام و نام خانوادگی	علی مهرابی
شماره دانشجویی	۴۰۲۲۳۸۵۴
تاریخ	۱۴۰۳ تیر
استاد درس	دکتر علیاری

## فهرست مطالب

۴	۱ لینک های مربوطه
۴	۱.۱ لینک مربوط به گیت هاب
۴	۲.۱ لینک مربوط به گوگل کلب
۵	
۵	۲ خلاصه
۵	
۵	۳ مقدمه
۵	
۸	۴ مرور ادبیات
۸	
۹	۵ آشنایی با دیتاست
۹	۶ تجزیه و تحلیل داده
۹	۱.۶ پاکسازی دیتاست
۹	۲.۶ گروه بندی و نمودار های مربوط به دیتاست
۱۳	
۱۳	۷ پیش پردازش
۱۳	۱.۷ فیلتر کردن دیتاست
۱۴	۲.۷ تشکیل ۳ زیر-دیتاست
۱۵	
۱۵	۸ استخراج ویژگی ها
۱۶	۱.۸ استخراج ویژگی های مربوط به زمان خرید
۱۷	۲.۸ استخراج ویژگی های مربوط به ارزش خرید
۱۸	۳.۸ اضافه کردن برچسب به دیتاست ها
۱۹	
۱۹	۹ آماده سازی دیتاست برای ساخت مدل
۲۰	
۲۰	۱۰ کاهش بعد به روش PCA
۲۱	
۲۱	۱۱ خوشه بندی (clustering) داده ها
۲۴	
۲۴	۱۲ تولید داده و بالانس کردن دیتاست
۲۵	۱.۱۲ تولید داده به کمک GAN
۲۷	۲.۱۲ تولید داده به کمک Variational Auto Encoder
۲۷	۳.۱۲ تولید داده به کمک SMOTE

## فهرست تصاویر

۷	.....	فوچارت شبکه GAN در [۷]	۱
۸	.....	سطرهای ابتدایی دیتاست خرده فروشی آنلайн	۲
۱۰	.....	میانگین خریدها در هر کشور	۳
۱۰	.....	ارزش کل خریدها در هر کشور	۴
۱۱	.....	ارزش کل فروش به صورت ماهانه	۵
۱۱	.....	بیشترین کالا به فروش رفته	۶
۱۱	.....	بیشترین ارزش تولید شده توسط هر کالا	۷
۱۲	.....	میزان خرید در هر سال	۸
۱۲	.....	میزان خرید در هر ماه	۹
۱۲	.....	میزان خرید در ماه به طور میانگین	۱۰
۱۳	.....	میزان خرید در ساعت روز به طور میانگین	۱۱
۱۴	.....	تاریخ در دیتاست	۱۲
۱۴	.....	تاریخ شروع و پایان هر دیتاست	۱۳
۱۵	.....	شماتیک سه دیتا فریم	۱۴
۱۵	.....	ابعاد هر دیتاست در پایان مرحله فیلتر کردن	۱۵
۱۸	.....	ابعاد هر دیتاست در پایان مرحله استخراج ویژگی ها	۱۶
۱۹	.....	دیتاست متغیر های مستقل X	۱۷
۱۹	.....	توزیع متغیر وابسته u در دیتاست	۱۸
۲۰	.....	فراآنی داده های آموزش و آزمون	۱۹
۲۰	.....	متغیر های categorical	۲۰
۲۰	.....	تمامی ویژگی ها	۲۱
۲۱	.....	واریانس توضیح داده شده توسط principle components	۲۲
۲۲	.....	معیار بازو برای تعیین تعداد خوش بھینه	۲۳
۲۲	.....	معیار silhouette score برای تعیین تعداد خوش بھینه	۲۴
۲۴	.....	توزیع داده ها در هر خوش	۲۵
۲۴	.....	گروه بندی خوش ها و نمایش میانگین و واریانس هر ویژگی	۲۶
۲۴	.....	نمایش تفاوت های خوش ها به کمک web-chart	۲۷
۲۵	.....	نمای کلی الگوریتم GAN	۲۸
۲۶	.....	برچسب ها پس از اضافه کردن خروجی GAN	۲۹



## فهرست برنامه‌ها

۱۳	.....	customer for transactions aggregating	۱
۱۴	.....	dataset the in names column the renaming	۲
۱۴	.....	..... data the cutting	۳
۲۳	.....	clustering k-means	۴

## ۱ لینک های مربوطه

### ۱.۱ لینک مربوط به گیت هاب

از این لینک [گیت هاب \(Github\)](#) می توانید برای دسترسی به صفحه Github مربوط به این پروژه استفاده کنید.

### ۲.۱ لینک مربوط به گوگل کلب

از این لینک [گوگل کلب \(Google Colab\)](#) می توانید برای دسترسی به notebook نوشته شده دسترسی پیدا کنید.

## ۲ خلاصه

## ۳ مقدمه

ریزش مشتریان یک مسئله بحرانی برای همه کسب و کارها محسوب می‌شود، زیرا از دست دادن مشتریان منجر به کاهش سود آینده می‌شود. هرچند جذب مشتریان جدید می‌تواند این خسارات را جبران کند، اما این فرآیند معمولاً هزینه بیشتری نسبت به حفظ مشتریان فعلی دارد. بنابراین، شناسایی مشتریان بالقوه‌ای که ممکن است ریزش کنند و سپس حفظ آن‌ها بسیار مهم است. [۱۲]

## ۴ مرور ادبیات

در مطالعه‌ای که در سال ۲۰۲۰ منتشر شده است، تکنیک‌های پیش‌بینی ریزش که تاکنون معرفی شده‌اند، بررسی شده‌اند [۱]. تحلیل ریزش به معنای شناسایی و پیش‌بینی احتمال قطع ارتباط مشتریان با یک کسب و کار است. این مقاله به بررسی تعاریف مختلف ریزش در حوزه‌های مدیریت کسب و کار، بازاریابی، فناوری اطلاعات، ارتباطات، روزنامه‌نگاری، بیمه و روانشناسی پرداخته و تفاوت‌های آن‌ها را تشریح کرده است. بر اساس این تعاریف، زیان‌های ناشی از ریزش، مهندسی ویژگی‌ها و مدل‌های پیش‌بینی ریزش طبقه‌بندی و توضیح داده شده‌اند. مطالعه نشان می‌دهد که تعریف ریزش و مدل‌های مرتبط با آن باید به تناسب حوزه خدماتی که پژوهشگران به آن علاقه‌مند هستند، انتخاب شوند. همچنین، این مقاله به معرفی مدل‌های پیش‌بینی ریزش مبتنی بر یادگیری عمیق پرداخته و مزایای آن‌ها را نسبت به مدل‌های سنتی تر تشریح کرده است. از دیگر نتایج تحقیق می‌توان به شناسایی معیارهای مختلف برای تعریف ریزش و زمان‌بندی آن اشاره کرد که بسته به ویژگی‌های هر خدمت متفاوت است. همچنین، مقاله به تحلیل ویژگی‌های مختلف و تکنیک‌های مهندسی ویژگی‌ها پرداخته که در پیش‌بینی ریزش مؤثر هستند.

مطالعه انجام شده در [۲] بر روی تحلیل ریزش مشتریان و پیش‌بینی آن با استفاده از روش‌های یادگیری ماشین تمکز دارد. در این پژوهش، یازده روش یادگیری نظارت شده supervised و نیمه نظارت شده semi-supervised و همچنین هفت روش نمونه‌برداری بر روی شانزده مجموعه داده مختلف و عمومی مرتبط با ریزش مورد بررسی قرار گرفته‌اند. هدف این تحقیق، ارائه راهنمایی‌های کلی از ارزیابی تکنیک‌های مختلف یادگیری ماشین در ارتباط با روش‌های نمونه‌برداری داده‌ها در زمینه پیش‌بینی ریزش است. انتخاب روش مناسب نمونه‌برداری و مدل طبقه‌بندی بستگی زیادی به ویژگی‌های ذاتی داده‌ها دارد. نتایج این تحقیق با استفاده از معیار مساحت زیر منحنی (AUC) گزارش شده است و تأثیر روش‌های نمونه‌برداری ویژگی‌های داده‌ها بر عملکرد روش‌های یادگیری مورد مطالعه قرار گرفته است. علاوه بر این، از آزمون نمینی و تحلیل تطابق به عنوان ابزارهای مقایسه و تجسم ارتباط بین الگوریتم‌های طبقه‌بندی، روش‌های نمونه‌برداری و مجموعه داده‌ها استفاده شده است. نتایج تجربی نشان می‌دهد که یک رویکرد ترکیبی می‌تواند به موفقیت پیش‌بینی ریزش در مجموعه داده‌های مختلف منجر شود.

در مطالعه انجام شده در سال ۲۰۲۳ پیش‌بینی ریزش مشتریان در بخش بانکی با استفاده از مدل‌های طبقه‌بندی مبتنی بر یادگیری ماشین بررسی شده است [۳]. این مطالعه به بررسی تأثیر تقسیم‌بندی مشتریان بر دقت پیش‌بینی ریزش مشتریان و ارزیابی مدل‌های یادگیری ماشین نظری KNN، رگرسیون لجستیک، درخت تصمیم، جنگل تصادفی

و ماشین بردار پشتیبانی پرداخته است. نتایج نشان می‌دهد که مدل جنگل تصادفی با دقت حدود ۹۷ درصد بهترین عملکرد را دارد. همچنین، تقسیم‌بندی مشتریان تأثیر زیادی بر دقت پیش‌بینی ندارد و این مسئله به مجموعه داده‌ها و مدل‌های انتخابی بستگی دارد.

مطالعه بعدی بر روی بهینه‌سازی ابرپارامترها و تکنیک‌های ترکیبی نمونه‌برداری داده‌ها در یادگیری ماشین برای پیش‌بینی ریزش مشتریان تمرکز دارد [۴]. در این پژوهش، مدل‌های مختلف یادگیری ماشین از جمله شبکه‌های عصبی مصنوعی، درخت‌های تصمیم‌گیری، ماشین‌های بردار پشتیبانی، جنگل‌های تصادفی<sup>۱</sup> و رگرسیون لجستیک مورد بررسی قرار گرفته‌اند. برای مقابله با چالش‌های داده‌های نامتعادل از استراتژی‌های مختلف نمونه‌برداری داده‌ها نظیر SMOTE استفاده شده است. نتایج این تحقیق نشان می‌دهد که مدل CatBoost با استفاده از بهینه‌سازی ابرپارامترها با، Optuna به عملکرد برتری دست یافته و F1-score ۹۳ درصد را کسب کرده است. همچنین XGBoost و CatBoost در معیار AUC ROC نیز عملکردی بی‌نظیر داشته و امتیازات ۹۱ درصد را به دست آورده‌اند. این عملکرد برای XGBoost پس از اعمال SMOTE پس از بهینه‌سازی ابرپارامترها با Optuna به دست آمده است.

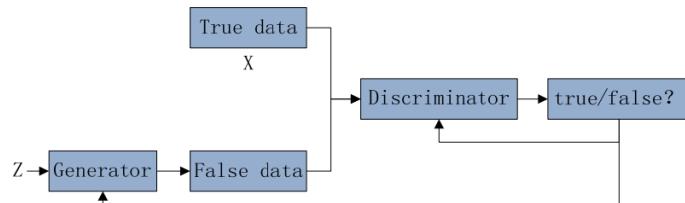
در مطالعه بعدی، یک چارچوب یادگیری ماشین برای پیش‌بینی خرید مشتریان در محیط‌های غیرقراردادی ارائه شده است [۵]. در این پژوهش، ویژگی‌های پویا و مبتنی بر داده از خریدهای گذشته مشتریان استخراج شده و هر ماه به روزرسانی می‌شوند. سپس الگوریتم‌های پیشرفته یادگیری ماشین نظیر رگرسیون لجستیک و XGBoost Lasso برای XGBoost دارد و با استفاده از یک مجموعه داده حاوی بیش از ۱۰،۰۰۰ مشتری و تعداد کل ۲۰۰،۰۰۰ خرید، به دقت ۸۹ درصد و مقدار AUC برابر با ۹۵٪ برای پیش‌بینی خریدهای ماه آینده دست یافته‌اند.

مطالعه انجام شده در [۶] با هدف بهبود دقت پیش‌بینی نیت ترک کار فارغ‌التحصیلان جدید دانشگاهی با حل مشکل داده‌های نامتوازن انجام شده است. برای این منظور، از داده‌های نظرسنجی جابجایی شغلی فارغ‌التحصیلان (GOMS) استفاده شده است. این داده‌ها شامل ویژگی‌های مختلفی از جمله نیت ترک کار، ویژگی‌های شخصی و شغلی فارغ‌التحصیلان جدید است و نسبت کلاس نیت ترک کار نامتوازن است. برای حل مشکل داده‌های نامتوازن، از تکنیک‌های SMOTE و شبکه‌های مولد متخصص (GAN) استفاده شده است. نتایج نشان می‌دهد که بالاترین دقت پیش‌بینی در داده‌های متوازن شده با استفاده از GAN به دست آمده است و دقت پیش‌بینی داده‌های متوازن شده با SMOTE نیز بهتر از داده‌های نامتوازن اصلی بوده است. این مطالعه نشان می‌دهد که GAN با گسترش و کاربرد آن در داده‌های ساختاریافته، بهبود چشم‌گیری در پیش‌بینی نیت ترک کار فارغ‌التحصیلان جدید دانشگاهی دارد.

در مطالعه [۷] نویسنده‌گان یک روش مبتنی بر شبکه‌های مولد متخصص (GAN) را برای حل مشکل داده‌های نامتوازن ارائه داده‌اند. داده‌های مشتریان بانک معمولاً نامتوازن هستند، به این معنی که تعداد مشتریان راضی بسیار بیشتر از مشتریان ناراضی است. روش پیشنهادی این مقاله از GAN برای تولید نمونه‌های گروه اقلیت (مشتریان ناراضی) استفاده می‌کند تا با متوازن‌سازی داده‌ها، عملکرد پیش‌بینی بهبود یابد. نتایج این تحقیق نشان می‌دهد که روش GAN در مقایسه با روش‌های سنتی نمونه‌برداری مانند BSSMOTE و SMOTE عملکرد بهتری دارد. با استفاده از معیارهایی مانند F1 و دقت، این روش توانست نتایج بهتری در شناسایی مشتریان ناراضی ارائه دهد و کاربرد عملی بالایی در مسائل طبقه‌بندی داده‌های نامتوازن بانکی داشته باشد.

در مطالعه بعدی نویسنده‌گان یک مدل مبتنی بر شبکه‌های مولد متخصص (GAN) را برای افزایش داده‌ها و حل مشکل نامتوازن بودن داده‌ها ارائه دادند [۸]. این مطالعه عملکرد طبقه‌بندی را بر روی سه مجموعه داده مختلف ارزیابی کرد و تکنیک‌های افزایش داده را برای تولید داده‌های مصنوعی برای کلاس‌های اقلیت به کار برد. نتایج نشان داد که مدل

<sup>1</sup>Random Forest



شکل ۱: فوچارت شبکه GAN در [۷]

پیشنهادی دقت طبقه‌بندی را بهبود بخشدید است، به طوریکه دقت مدل در مجموعه داده‌های نامتوازن به ترتیب ۸۷.۲٪، ۹۵.۷٪ و ۷۶٪ بود. این تحقیق نشان می‌دهد که استفاده از تکنیک‌های افزایش داده مبتنی بر GAN می‌تواند عملکرد طبقه‌بندی در مجموعه داده‌های نامتوازن را بهبود بخشد و به ایجاد پیش‌بینی‌های دقیق‌تر و منصفانه‌تر کمک کند.

در مطالعه بعدی، نویسنده‌گان یک مدل مبتنی بر خوش‌بندی و پیش‌بینی به نام ClusPred را برای پیش‌بینی ریزش مشتریان پیشنهاد دادند [۹]. این مدل، داده‌های دموگرافیک و تراکنش‌های کاربران را برای خوش‌بندی مشتریان بر اساس رفتار و ویژگی‌های دموگرافیک ترکیب می‌کند. سپس از یک فرایند پواسون غیرهمگن (IHPP) برای مدل‌سازی رفتار مشتریان و پیش‌بینی ریزش استفاده می‌شود. نتایج تجربی نشان داد که ClusPred با IHPP عملکرد پیش‌بینی را بهبود بخشدید و توانسته است دقت و نرخ مثبت حقیقی (TPR) را در مقایسه با روش‌های دیگر افزایش دهد. این مدل به دلیل کارآیی بالا، برای کاربردهای داده‌های بزرگ نیز مناسب است.

در مطالعه بعدی؛ نویسنده‌گان از مدل‌های هیبریدی متشکل از تکنیک‌های خوش‌بندی بدون نظارت<sup>۲</sup> و درخت‌های تصمیم استفاده کردند [۱۰]. هدف این مطالعه بررسی دو روش مختلف برای هیبریدی کردن مدل‌ها برای استفاده از نتایج خوش‌بندی بر اساس ویژگی‌های مختلف مرتبط با استفاده از خدمات و سهم درآمدی مشتریان است. نتایج نشان داد که استفاده از خوش‌بندی منجر به بهبود عملکرد مدل‌های هیبریدی در مقایسه با حالتی که از خوش‌بندی استفاده نمی‌شود. همچنین نشان داده شد که استفاده از برچسب‌های خوش به عنوان ورودی به درخت‌های تصمیم روشی مطلوب برای هیبریدی کردن است. این تحقیق با استفاده از دو مجموعه داده مختلف انجام شد و نتایج آن نشان داد که تکنیک‌های خوش‌بندی مختلف می‌توانند الگوهای مفیدی را برای شناسایی مشتریان ریزشی ارائه دهند.

در یک مطالعه دیگر در این زمینه، نویسنده‌گان یک مدل هیبریدی مبتنی بر ترکیب الگوریتم‌های خوش‌بندی و طبقه‌بندی با استفاده از مجموعه‌ای از تکنیک‌های ترکیبی ارائه دادند [۱۱]. این مدل با ارزیابی الگوریتم‌های مختلف خوش‌بندی (مانند K-means، K-medoids، X-means و خوش‌بندی تصادفی) و سپس ترکیب این خوش‌های با هفت الگوریتم طبقه‌بندی متفاوت، بر روی دو مجموعه داده مختلف مربوط به پیش‌بینی ریزش مشتریان آزمایش شد. نتایج نشان داد که مدل پیشنهادی بالاترین دقت پیش‌بینی را به ترتیب ۹۴.۷٪ در مجموعه داده GitHub و ۹۲.۴٪ در مجموعه داده Bigml کسب کرده است. همچنین مقایسه با مدل‌های پیشرفته موجود نشان داد که مدل پیشنهادی عملکرد بهتری در پیش‌بینی ریزش مشتریان دارد.

<sup>2</sup>unsupervised clustering



## ۵ آشنایی با دیتاست

در این بخش دیتاستی که در طی پروژه با آن برای پیش بینی ریزش مشتریان کار شده است معرفی می شود. این مجموعه داده که در [این لینک](#) به صورت عمومی در دسترس است، مربوط به اطلاعات مشتریان یک فروشگاه اینترنتی خرده فروشی واقع در کشور انگلستان است که مربوط به سال ۲۰۱۰ تا ۲۰۱۱ است.

این دیتاست ذات ترتیبی و time series دارد و اطلاعات خرید مشتری را با ۸ ویژگی ثبت کرده است. این فروشگاه به طور عمدۀ هدایای منحصر بفرد را برای همه مناسبت‌ها می فروشد. بسیاری از مشتریان این فروشگاه عمدۀ فروشان هستند.

در [شکل ۲](#) چند سطر اول این دیتاست آمده است.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

شکل ۲: سطرهای ابتدایی دیتاست خرده فروشی آنلاین

در این دیتاست factor ID InvoiceNo برابر با همان object داده است. اگر اعداد در این ستون C قرار بگیرد یعنی این سفارش با این فاکتو کنسل شده است. تایپ این ویژگی است.

StockCode مربوط به آیدی به خصوص برای هر کالا است، برای مثال آویز سفید قلبی شکل در نمونه‌ها یک آیدی مخصوص دارند که نمایش داده است. تایپ این ویژگی object است.

Description مربوط به توضیحات یک آیتم است. شامل اسم کالایی که خریداری شده است. تایپ این ویژگی object است.

Quantity مربوط به تعداد کالا در سبد خرید است. برای مثال در سطر اول تعداد ۶ آویز آسفید قلبی شکل وجود دارد. تایپ این ویژگی int64 است.

InvoiceDate تاریخ میلادی و ساعت دقیق ثبت فاکتور را نشان می دهد. تایپ این ویژگی datetime64 و از نوع زمان است.

UnitPrice قیمت یک کالا با واحد پوند استرلینگ<sup>۳</sup> را نشان می دهد. برای مثال در سطر اول قیمت هر آویز سفید قلبی شکل £ 2.55 است. تایپ این ویژگی float64 است.

CustomerID آیدی مخصوص هر مشتری را نشان می دهد. یک مشتری می تواند با یک آیدی منحصر به فرد چندین خرید را انجام بدهد و چندین InvoiceNo را در دیتاست ثبت کند. این فاکتورها خود می توانند شامل چندین StockCode باشد. تایپ این ویژگی float64 است.

Country کشوری است که خرید در آنجا انجام شده و معرف محل زندگی مشتری است. تایپ این ویژگی object است.

دیتاست شامل ۵۴۱۹۰۹ نمونه است که هر کدام به صورت بالا اطلاعات خرید های مشتریان فروشگاه را نشان می دهند.

<sup>3</sup>Pound sterling

علاوه بر ویژگی های معرفی شده در این بخش یک ویژگی دیگری که با آن زیاد کار می شود و مورد نیاز است، ویژگی TotalPrice است که بیانگر حاصل ضرب تعداد هر واحد در تعداد کالا است و به ویژگی ها اضافه شده است. تا پنجمین ویژگی float64 است.

## ۶ تجزیه و تحلیل داده

### ۱.۶ پاکسازی دیتاست

ابتدا به کمک دستور `df.describe()` می توان متوجه شد که میانگین قیمت هر کالا در این فروشگاه ۴.۶£ و میانگین هر فاکتور برای مشتریان ۱۷.۹۸£ است. به کمک دستور `df.isnull().sum()` و یا `df.info()` می توان متوجه شد که تعداد missing value در این دیتاست وجود دارد که تعداد ۱۴۵۴ از آنها مربوط به ویژگی Description و ۱۳۵۰۸۰ از آنها مربوط به CustomerID است. برای مواجه با این موارد چون دسترسی به اطلاعات بیشتری نداریم و به فروشگاه نیز دسترسی نداریم، مجبور به حذف این موارد هستیم.

در مرحله بعد به سراغ فاکتور های کنسل شده می رویم، گفته شد که ابتدا اعداد در InvoiceNo برای فاکتور های کنسلی یک C قرار می گیرد. با توجه به این مشخصه، دیتا هایی که این ویژگی را دارند به عنوان `canceled_orders` جدا می کنیم. سپس داده هایی که `Quantity > 0` دارند را به عنوان `df_clean` قرار می دهیم.  
پس از اعمال این تغییرات تعداد سفارش های منحصر به فرد در دیتا است به ۱۸۵۳۶ و تعداد مشتریان با شناسه<sup>۴</sup> منحصر به فرد به ۴۳۳۹ کاهش می یابد.

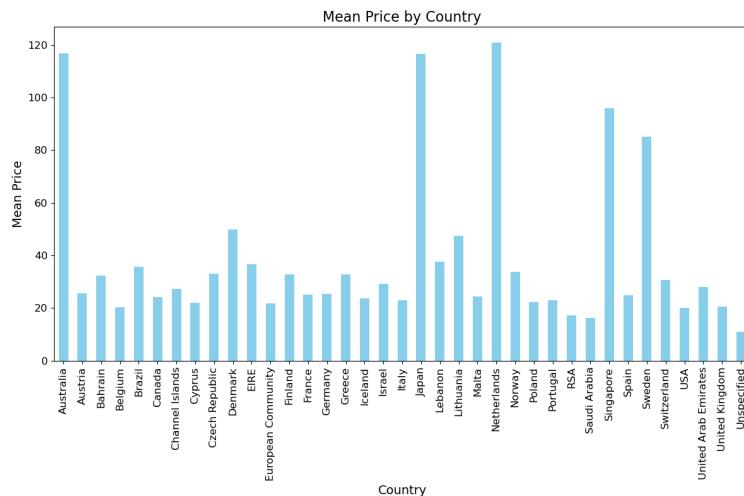
### ۲.۶ گروه بندی و نمودار های مربوط به دیتاست

در این بخش ابتدا به سراغ کشورهایی که مشتریان در آن هستند می رویم. می توان میانگین خرید در هر کشور به به صورت **شکل ۳** نشان داد. مطابق این نمودار بهترین کشورهای مشتریان از نظر سودآوری مربوط به استرالیا، ژاپن، هلند، سنگاپور و سوئد هستند. نکته جالب اینجاست که با وجود اینکه این فروشگاه در انگلستان قرار دارد اما میانگین ارزش خرید مشتریان در این کشور نسبت به سایر کشور ها کمتر است.

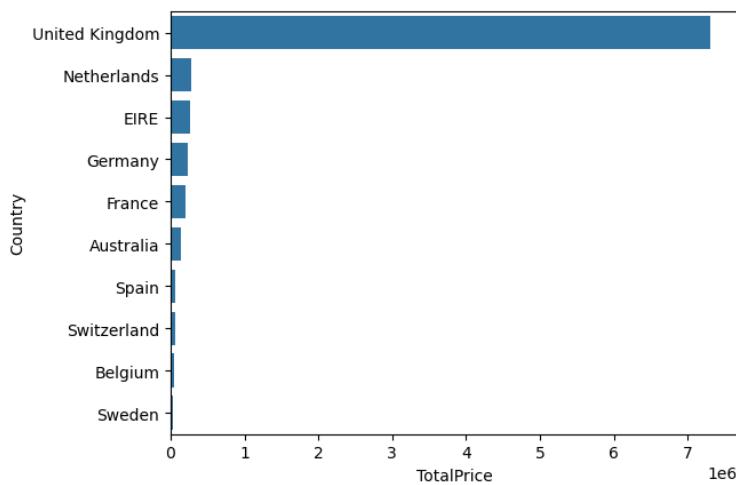
در مرحله بعد مشخص می کنیم کدام کشور ها TotalPrice بیشتری نسبت به بقیه دارند. این آمار در **شکل ۴** آمده است. مطابق این آمار میانگین خرید ها در انگلستان اگرچه پایین است اما میزان خرید در این کشور بسیار بالاست که منجر به ارزش کل بالا با اختلاف بسیار بیشتری شده است.

آمار بعدی مشخص می کند که فروش کل به صورت ماهانه چگونه تغییر می کند. مطابق **شکل ۵** دیده می شود که این آمار خطی است و روبه افزایش است. یعنی احتمالاً سود این کسب و کار به صورت خطی افزایش می یابد. می توان همچنان مشخص کرد که کدام کالا ها بیشترین فروش (از نظر تعداد) را در بین محصولات دارد. همچنین گفت که این محصولات در کدام کشور بیشترین فروش را دارد. مطابق **شکل ۶** بیشترین کالایی که به فروش رفته (از نظر

<sup>4</sup>CustomerID



شکل ۳: میانگین خریدها در هر کشور



شکل ۴: ارزش کل خریدها در هر کشور

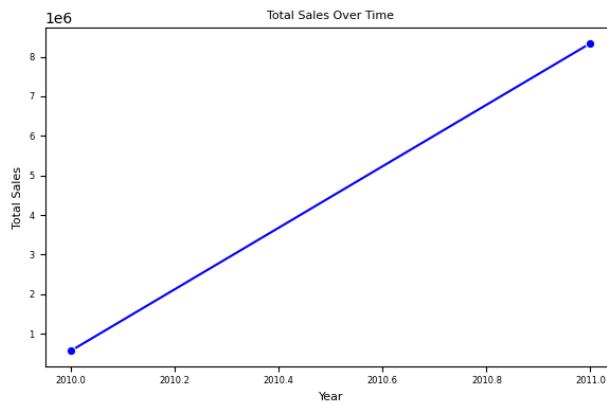
تعداد کالا) یک محصول کاردستی کاغذی است که احتمالاً طرفدار زیادی دارد. در رتبه دوم پایه کیک سلطنتی است که آن هم فروش زیاده داشته و کالای محبوبی به شمار می‌آید. در رتبه سوم آویز سفید قلبی شکل است.

همچنین در ادامه [شکل ۶](#) می‌توان مشخص کرد که کدام کالا ها بیشترین ارزش آوری را برای فروشگاه داشته‌اند.

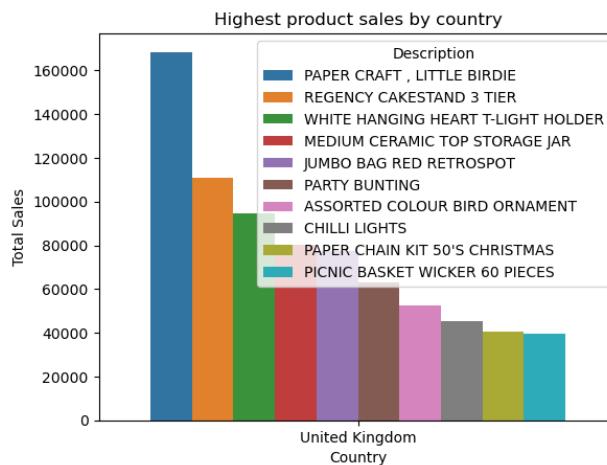
[شکل ۷](#) ارزش کل نصیب شده فروشگاه برای پر فروش ترین محصولات را نشان می‌دهد.

در مراحل بعد می‌توانیم میزان کل فروش را به صورت نمودار میله‌ای و نقطه‌ای برای ماه و سال رسم کنیم که مطابق [شکل ۸](#) و [شکل ۹](#) است. اختلاف در [شکل ۸](#) بین دو سال به این علت است که دیتابست شامل ماه آخر سال ۲۰۱۰ و تمامی ماه‌های سال ۲۰۱۱ است. در [شکل ۹](#) دیده می‌شود که در ماه‌ها منتهی به سال جدید مخصوصاً در ماه ۱۱ میزان فروش به شدت زیاد می‌شود که احتمالاً مربوط به تعطیلات کریسمس است.

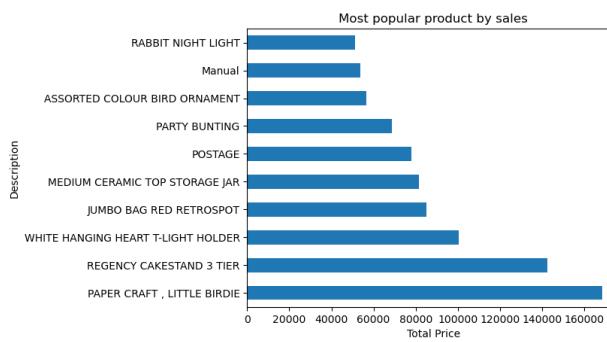
نمودار در [شکل ۱۰](#) تعداد تراکنش‌ها در یک روز را به طور میانگین نشان می‌دهد. در روزهای اول ماه به علت دریافت حقوق و توانایی بیشتر مردم در خرید، تعداد تراکنش‌ها بالاتر است، اما در روزهای آخر ماه قدرت خرید مردم



شکل ۵: ارزش کل فروش به صورت ماهانه



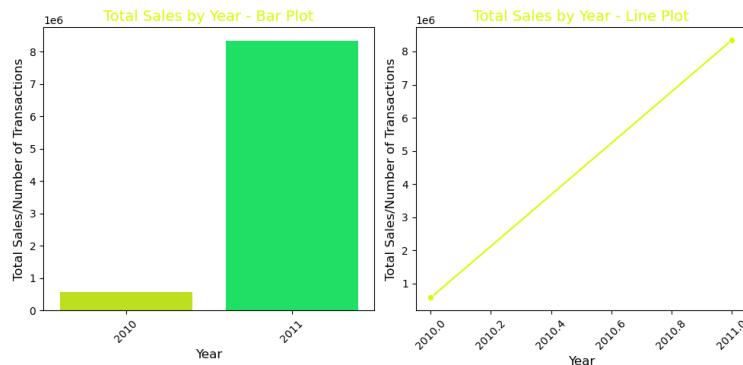
شکل ۶: بیشترین کالا به فروش رفته



شکل ۷: بیشترین ارزش تولید شده توسط هر کالا

کمتر است و در روز آخر به کمترین میزان خود می‌رسد.

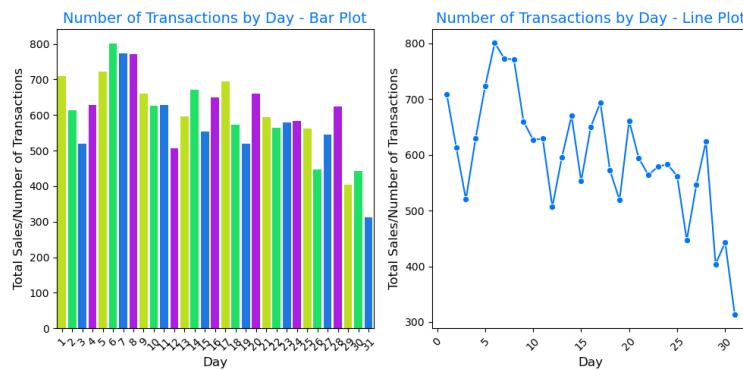
در شکل ۱۱ نمودار میزان تراکنش بر اساس ساعت در روز آورده شده که نشان می‌دهد تراکنش‌ها از طرف مشتریان در ساعت ۱۲ ظهر قله می‌زند و سپس رو به کاهش می‌رود.



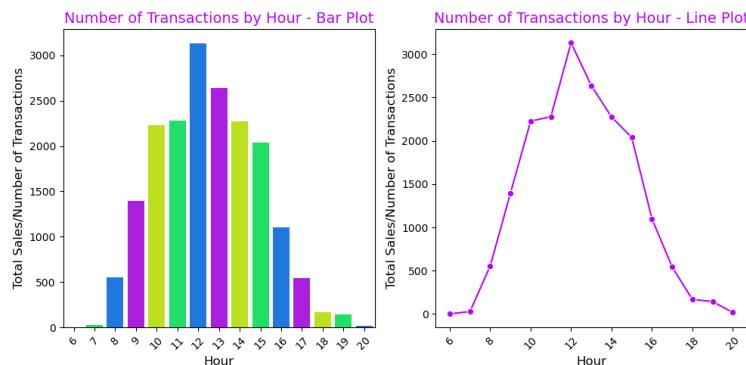
شكل ۸: میزان خرید در هر سال



شكل ۹: میزان خرید در هر ماه



شكل ۱۰: میزان خرید در ماه به طور میانگین



شکل ۱۱: میزان خرید در ساعت‌های روز به طور میانگین

## ۷ پیش‌پردازش

### ۱.۷ فیلترکردن دیتاست

در این بخش با یک دیتاست پاکسازی شده طرف هستیم که مقادیر missing value ندارد و فاکتورهای کنسلی از آن حذف شده‌اند. سعی می‌کنیم تا دیتارا برای ادامه کار به ساختار منظم تر که درک بهتری از آن داریم تقسیم کنیم. دیتاست به طور دقیق‌تر از تاریخ '۰۱-۱۲-۰۹' تا '۱۲-۱۲-۲۰۱۱' در اختیار ما قرار دارد.

در ادامه یک 'CustomerID', 'InvoiceDate', 'TotalPrice' درست شده که شامل ستون‌های 'Country' است. در ادامه تنها با این ستون‌ها کار می‌کنیم و سایر ویژگی‌ها حذف می‌شوند.

تغییر دیگری که باید در دیتاست ایجاد شود، یکی کردن سبد خرید برای هر مشتری است. در واقع باید مشتری‌هارا بر اساس تاریخ خرید و شناسه آن‌ها دسته‌بندی کنیم، سپس ارزش تمامی خرید‌ها در یک روز و با یک فاکتور را با هم جمع کنیم. در واقع با این کار یک سبد خرید برای مشتری درست کرده‌ایم و این سبد خرید شامل مجموع ارزش تمامی کالا‌ها با تاریخ خرید یکسان و فاکتور یکسان هستند. دقیقاً مانند فاکتوری که شما هنگام خرید از یک فروشگاه زنجیره‌ای دریافت می‌کنید. با انجام اینکار یک df\_merged بوجود می‌آید. اینکار به صورت برنامه ۱ انجام شده.

```

1 df_merged = df_filtered.groupby(['CustomerID', 'InvoiceDate', 'Country']).agg({
2     'TotalPrice': 'sum'
3 }).reset_index()

```

Code 1: aggregating transactions for customer

در مرحله بعد اسامی ویژگی‌ها به صورت استاندارد تغییر می‌کند تا در ادامه کار با آن راحت‌تر باشد. مطابق برنامه ۲ انجام شده.

در این پروژه کمترین واحد زمانی ماه در نظر گرفته شده است. با این استاندارد نیاز است تا ماه آخر که تنها ۹ روز از آن در دسترس است حذف شود. به صورت دقیق‌تر اینکار در برنامه ۳ انجام شده است.

بعد از انجام اینکار، تاریخ در دیتاست به صورت شکل ۱۲ خواهد بود.



```

1 df_merged = df_merged.rename(columns={
2     'CustomerID': 'customers_id',
3     'InvoiceDate': 'date',
4     'TotalPrice': 'price_purchase',
5     'Country': 'country'
6 })

```

Code 2: renaming the column names in the dataset

```

1 start_date = '2010-12-01'
2 end_date = '2011-11-30'
3 df = df[(df['date'] >= start_date) & (df['date'] <= end_date)]

```

Code 3: cutting the data

```

1 df['date'].min(), df['date'].max()
Python
Timestamp('2010-12-01 00:00:00'), Timestamp('2011-11-30 00:00:00'))

```

شکل ۱۲: تاریخ در دیتاست

## ۲.۷ تشکیل ۳ زیر-دیتاست

مطابق [شکل ۱۲](#) ۱۲ ماه از داده در اختیار ماست که شامل بازه ابتدای ماه ۱۲ سال ۲۰۱۰ تا انتهای ماه ۱۱ سال ۲۰۱۱ می‌شود. آنچه که ما قصد انجامش را داریم تقسیم دیتاست به سه دیتاست مجزا است که هریک اطلاعات ۴ ماه مشتریان را دارند.

در هر دیتاست که شامل ۴ ماه است، ویژگی‌های سه ماه اول آن به مدل داده می‌شود و اینکه مشتری در ماه چهارم آن خرید داشته یا نه به عنوان معیار ریزش برگزیده می‌شود. اگر مشتری در ماه چهارم خرید داشت، یعنی ریزش نکرده و در غیر این صورت ریزش انجام شده است. ماه‌های شروع و پایان هر دیتاست به صورت [شکل ۱۳](#) است.

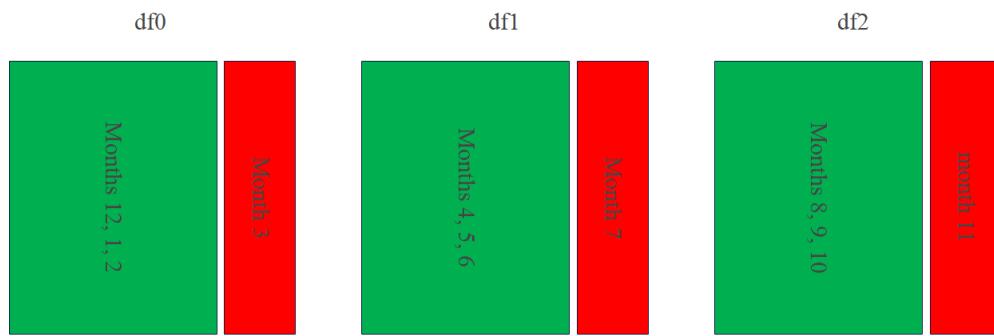
```

1 start_cutoff_dates, break_cutoff_dates
Python
([Timestamp('2010-12-01 00:00:00'),
 Timestamp('2011-04-01 00:00:00'),
 Timestamp('2011-08-01 00:00:00')],
 [Timestamp('2011-03-01 00:00:00'),
 Timestamp('2011-07-01 00:00:00'),
 Timestamp('2011-11-01 00:00:00')])

```

شکل ۱۳: تاریخ شروع و پایان هر دیتاست

به طور دقیق‌تر می‌توان دیاگرام [شکل ۱۴](#) را به عنوان ساختار تاریخ‌بندی سه دیتاست درنظر گرفت. در این سه شکل قسمت‌های آبی شامل ویژگی و نمودار قرمز شامل برچسب‌های مربوط به سه ماه گذشته آن است. به عنوان مثال df0 از روز اول ماه ۱۲ سال ۲۰۱۰ شروع شده و تا آخر ماه سوم اطلاعات فروش را دارد. از این ۴ ماه



شکل ۱۴: شماتیک سه دیتا فریم

سه ماه اول به عنوان ویژگی به مدل داده می شود و ماه آخر دارای برچسب است. فراموش نشود که در این پروژه باید اطلاعات خریدهای گذشته مشتریان به مدل داده شود و پیش بینی شود که آیا در ماه آینده خرید خواهد کرد یا ریزس خواهد کرد.

در مرحله بعد اگر هر مشتری بیشتر از یک خرید در هر روز داشته باشد، یعنی از آنها را نگه می داریم. همچنین مشتریانی که کمتر از ۲ خرید در کل دیتاست دارند را حذف می کنیم چون این مشتریان باعث انحراف مدل می شوند و اطلاعات خاصی در اختیار آن قرار نمی دهند. در نهایت این اعمال، ابعاد هر دیتاست به صورت شکل ۱۵ است. همانطور که دیده می شود بسیاری از نمونه ها فیلتر شده است و ابعاد زیادی مواجه نیستیم.

```
1 df0.shape, df1.shape, df2.shape
((1445, 5), (1625, 5), (1918, 5))
```

شکل ۱۵: ابعاد هر دیتاست در پایان مرحله فیلتر کردن

## ۸ استخراج ویژگی ها

ویژگی ها به طور کلی به دو قسمت ویژگی های مربوط به زمان خرید مشتری<sup>۵</sup> و ویژگی های مربوط به ارزش خرید مشتری<sup>۶</sup> تقسیم بندی می شوند که در بخش های بعدی به توضیح هر ویژگی پرداخته می شود.

<sup>5</sup>Time of purchase

<sup>6</sup>Value of purchase

## ۱.۸ استخراج ویژگی های مربوط به زمان خرید

:purchase\_month

این ویژگی ماه و سال را از ستون 'date' در DataFrame استخراج می کند. این ویژگی تاریخ های تراکنش را به دوره های ماهانه گروه بندی می کند و ویژگی مبتنی بر زمان برای تحلیل فراهم می آورد.

:most\_common\_day

این ویژگی پر تکرارترین روز هفته را که یک مشتری خرید انجام می دهد، شناسایی می کند. تراکنش ها را بر اساس مشتری گروه بندی کرده و مدت ستون 'day\_of\_week' را تعیین می کند.

:number\_of\_purchases

این ویژگی تعداد کل خرید های انجام شده توسط هر مشتری را شمارش می کند. این ویژگی یک معیار از فعالیت مشتری را با شمارش تراکنش ها برای هر مشتری فراهم می کند.

:weighted\_mean\_time\_between\_purchases

این ویژگی میانگین وزنی زمان بین خریدها را برای هر مشتری محاسبه می کند. ابتدا تراکنش ها را بر اساس تاریخ و مشتری مرتب کرده و تفاوت های زمانی بین خرید های متوالی را محاسبه می کند و سپس یک فرمول وزنی برای تأکید بر تراکنش های اخیر اعمال می کند. مقدار حاصل نمایانگر میانگین فاصله زمانی بین خریدها با وزن بیشتر برای فعالیت های اخیر است.

:std\_between\_purchase

این ویژگی انحراف معیار زمان بین خریدها برای هر مشتری را محاسبه می کند که توسط تازگی تراکنش ها وزن دهنده شده است. این ویژگی نشانی از تغییر پذیری در فاصله زمانی بین خرید های یک مشتری را می دهد.

:max\_time\_without\_purchase

این ویژگی حداقل فاصله زمانی بین خریدها برای هر مشتری را محاسبه می کند که نشان دهنده طولانی ترین دوره ای است که یک مشتری بدون خرید سپری کرده است.

:time\_since\_last\_purchase

این ویژگی زمان از آخرین خرید هر مشتری را از یک تاریخ قطع خاص محاسبه می کند که بر حسب ماه اندازه گیری شده است. این ویژگی معیاری از تازگی مشتری را ارائه می دهد.

:transaction\_recency

این ویژگی تازگی تراکنش ها برای هر مشتری را که با تعداد کل مشتریان منحصر به فرد نرمال سازی شده است، محاسبه می کند. این ویژگی به فهمیدن تازگی نسبی آخرین خرید هر مشتری کمک می کند.

#### :last\_between\_{threshold}\_and\_{upper\_limit}

این ویژگی ستون‌های کدگذاری شده یک بار داغ ایجاد می‌کند که نشان می‌دهد آیا زمان از آخرین خرید در محدوده‌های مشخصی قرار می‌گیرد (مثلاً ۳۰-۶۰ روز). این ویژگی نمایشی دسته‌بندی از تازگی خرید ارائه می‌دهد.

#### :diff\_last\_and\_penultimate

این ویژگی تفاوت در روزها بین آخرین و ماقبل آخرین خرید هر مشتری را محاسبه می‌کند که بینشی در رفتار خرید اخیر ارائه می‌دهد.

#### :diff\_penultimate\_and\_previous

این ویژگی تفاوت در روزها بین ماقبل آخرین خرید و خرید قبل از آن برای هر مشتری را محاسبه می‌کند که زمینه‌ی بیشتری در مورد یکنواختی فواصل خرید ارائه می‌دهد.

#### :tresh\_۳ and tresh\_۲ and tresh\_۱

این ویژگی‌ها آستانه‌هایی را بر اساس میانگین وزنی زمان بین خریدها و انحراف معیار، که با عوامل مختلف ( $h_1 h_2 h_3$ ) مقیاس‌بندی شده‌اند، محاسبه می‌کنند. این ویژگی‌ها برای طبقه‌بندی تازگی خریدهای مشتری به دسته‌های مختلف استفاده می‌شوند.

#### :freq\_class

این ویژگی مشتریان را بر اساس تازگی خرید آن‌ها نسبت به آستانه‌های محاسبه‌شده به دسته‌های "عادی"، "ترک"، "در معرض خطر"، و "از دست رفته" طبقه‌بندی می‌کند. این ویژگی به شناسایی بخش‌های مشتری بر اساس رفتار خرید آن‌ها کمک می‌کند.

#### :rolling\_avg\_۳

این ویژگی میانگین متحرک سه خرید آخر هر مشتری را محاسبه می‌کند که روند کوتاه‌مدت در رفتار هزینه‌کرد مشتری را نشان می‌دهد.

#### :rolling\_avg\_۶

این ویژگی میانگین متحرک شش خرید آخر هر مشتری را محاسبه می‌کند که روند میان‌مدت در رفتار هزینه‌کرد مشتری را نشان می‌دهد.

## ۲.۸ استخراج ویژگی‌های مربوط به ارزش خرید

#### :binned\_purchase

این ویژگی مجموع مبلغ خریدهای هر مشتری را محاسبه کرده و به DataFrame اصلی اضافه می‌کند.



#### :relative\_change

این ویژگی تغییر نسبی در ارزش خرید را با مقایسه آخرین خرید با میانگین متحرک شش خرید آخر برای هر مشتری محاسبه می‌کند.

#### :purchase\_trend

این ویژگی روند خرید مشتریان را بر اساس تغییر نسبی در ارزش خریدهای آن‌ها به دسته‌هایی مانند 'کاهشی'، 'ثبت' و 'افزایشی' طبقه‌بندی می‌کند.

#### :max\_purchase

این ویژگی حداقل مبلغ خرید برای هر مشتری را شناسایی کرده و به DataFrame اصلی اضافه می‌کند.

#### :mean\_purchase

این ویژگی میانگین مبلغ خرید برای هر مشتری را محاسبه کرده و به DataFrame اصلی اضافه می‌کند.

#### :median\_purchase

این ویژگی میانه مبلغ خرید برای هر مشتری را محاسبه کرده و به DataFrame اصلی اضافه می‌کند.

#### :relative\_purchase\_value

این ویژگی تغییر نسبی در ارزش خرید برای هر مشتری را با مقایسه آخرین خرید با یا اولین خرید یا پنجمین خرید از آخر، بسته به تعداد خریدها، محاسبه می‌کند.

#### :time\_frame\_char

این ویژگی تغییر نسبی ارزش خرید را به دسته‌های 'ثبت'، 'در حد' و 'تغییر' بر اساس آستانه مشخص شده طبقه‌بندی می‌کند.

در نهایت و بعد از اضافه کردن این ویژگی‌ها به هر سه دیتاست، ابعاد آن‌ها به صورت **شکل ۱۶** خواهد بود.

```
> v
1 df0.shape, df1.shape, df2.shape
[51]
.. ((1445, 32), (1625, 32), (1918, 32))
```

شکل ۱۶: ابعاد هر دیتاست در پایان مرحله استخراج ویژگی‌ها

### ۳.۸ اضافه کردن برچسب به دیتاست‌ها

مطابق آنچه در **شکل ۱۴** توضیح داده شد، برچسب را به دیتا است اضافه می‌کنیم. برای یادآوری اگر در سه ماه گذشته، مشتری خرید داشته باشد برچسب آن ۱ و در غیر این صورت برچسب صفر خواهد داشت.



## ۹ آماده سازی دیتاست برای ساخت مدل

در ابتدای این قسمت زمان آن است که دیتافریم های ساخته شده با هم ترکیب شوند. در این حالت از دستور  $final\_df = pd.concat(dfs, ignore_index = True)$  استفاده می کنیم و یک  $final\_df$  می سازیم که تمامی رکورد هارا در خود دارد. این دیتاست به طور کلی ۴۹۸۸ سطر دارد و شامل برچسب نیز هست. بعد از این مرحله سطر های دیتاست را جابجا یا اصطلاحا shuffle می کنیم. با تعریف متغیر وابسته و مستقل، X به صورت [شکل ۱۷](#) است.

```

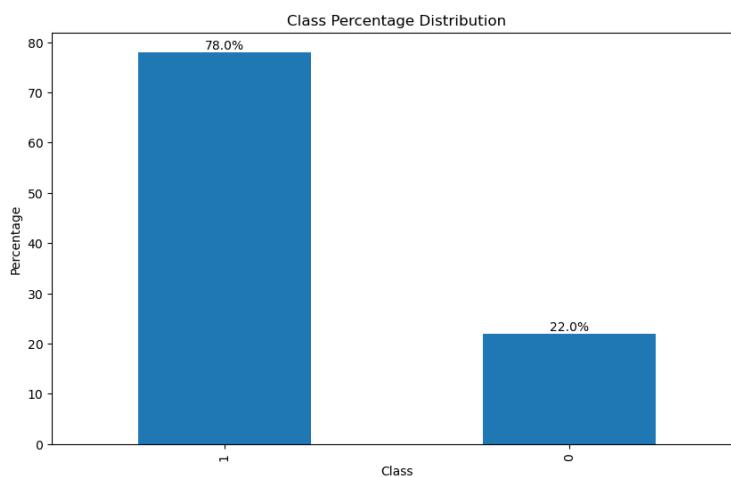
1 X.columns

Index(['customers_id', 'date', 'country', 'price_purchase', 'day_of_week',
       'purchase_month', 'most_common_day', 'number_of_purchases',
       'weighted_mean_time_between_purchases', 'std_between_purchase',
       'max_time_without_purchase', 'time_since_last_purchase',
       'transaction_recency', 'last_between_0_and_30',
       'last_between_30_and_60', 'last_between_60_and_90',
       'diff_last_and_penultimate', 'diff_penultimate_and_previous', 'tresh_1',
       'tresh_2', 'tresh_3', 'freq_class', 'rolling_avg_3', 'rolling_avg_6',
       'binned_purchase', 'd', 'purchase_trend', 'max_purchase',
       'mean_purchase', 'median_purchase', 'rel_purchase_value',
       'time_frame_char', 'year', 'month', 'day'],
      dtype='object')

```

شکل ۱۷: دیتاست متغیر های مستقل X

علاوه بر این می توان توزیع متغیر وابسته را نیز به صورت [شکل ۱۸](#) نشان داد. همانطور که دیده می شود توزیع این متغیر یا برچسب های ما یک توزیع غیر متعادل [۱۸](#) است. ۷۸ درصد از برچسب ها ۱ و ۲۲ درصد از برچسب ها ۰ هستند.



شکل ۱۸: توزیع متغیر وابسته y در دیتاست

سپس داده را به دو بخش آموزش و آزمون تقسیم می کنیم. ۲۰ درصد از داده را به بخش آزمون اختصاص می دهیم. فراوانی داده ها در این بخش به صورت [شکل ۱۹](#) است.

<sup>7</sup>Imbalanced



```
X_train shape: (3990, 35)
X_test shape: (998, 35)
y_train shape: (3990,)
y_test shape: (998,)
```

شکل ۱۹: فراوانی داده های آموزش و آزمون

همچنین در داده ها تعدادی متغیر categorical داریم که به کمک Label encoder تبدیل به داده عددی می کنیم. این متغیر ها در [شکل ۲۰](#) آمده است.

```
1 object_columns = X.select_dtypes(include=['object'])
2 object_column_names = object_columns.columns.tolist()
3 print(object_column_names)
4
5 ['country', 'day_of_week', 'most_common_day', 'freq_class', 'purchase_trend', 'time_frame_char']
```

شکل ۲۰: متغیر های categorical

بعد از این باید متغیر هایی که در ساخت مدل به آن ها نیازی نداریم را drop کنیم. در نهایت تمامی ویژگی های باقی مانده به صورت [شکل ۲۱](#) خواهد بود.

```
1 X_train.columns
2
3 Index(['customers_id', 'country', 'price_purchase', 'day_of_week',
4        'most_common_day', 'number_of_purchases',
5        'weighted_mean_time_between_purchases', 'std_between_purchase',
6        'max_time_without_purchase', 'time_since_last_purchase',
7        'transaction_recency', 'last_between_0_and_30',
8        'last_between_30_and_60', 'last_between_60_and_90',
9        'diff_last_and_penultimate', 'diff_penultimate_and_previous',
10       'freq_class', 'rolling_avg_3', 'rolling_avg_6', 'binned_purchase',
11       'purchase_trend', 'max_purchase', 'mean_purchase', 'median_purchase',
12       'rel_purchase_value', 'time_frame_char', 'year', 'month', 'day'],
13      dtype='object')
```

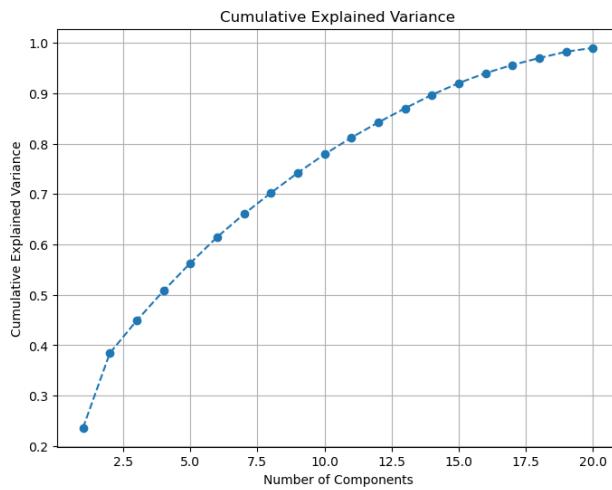
شکل ۲۱: تمامی ویژگی ها

سپس باید متغیر هارا scale کنیم که برای داده های آموزش و آزمون به صورت *fit\_transform* و *transform* انجام خواهد شد.

## ۱۰ کاهش بعد به روش PCA

در حال حاضر ۲۹ متغیر مستقل وجود دارد و ابعاد مسئله می تواند کم شود تا از correlation در بین ویژگی ها جلوگیری شود (principle components نسبت به هم عمود هستند). علاوه بر این برای جلوگیری از آثار منفی نویز در دیتاست و جلوگیری از بیش برآذش یا overfitting از کاهش بعد به روش PCA استفاده می شود. همچنین با استفاده از کاهش بعد به روش PCA اجرای کد سریع تر و حافظه کمتری نیز مصرف می شود. [منبع](#)

برای تشخیص اینکه از چند principle component استفاده کنیم باید از میزان واریانس توصیف شده توسط principle component های مختلف اطلاع داشته باشیم. این جدول به صورت [شکل ۲۲](#) است.



شکل ۲۲: واریانس توضیح داده شده توسط principle components

مطابق شکل ۲۲ با ۱۰ principle component کار را ادامه می دهیم چون مقدار مناسبی به نظر می رسد.

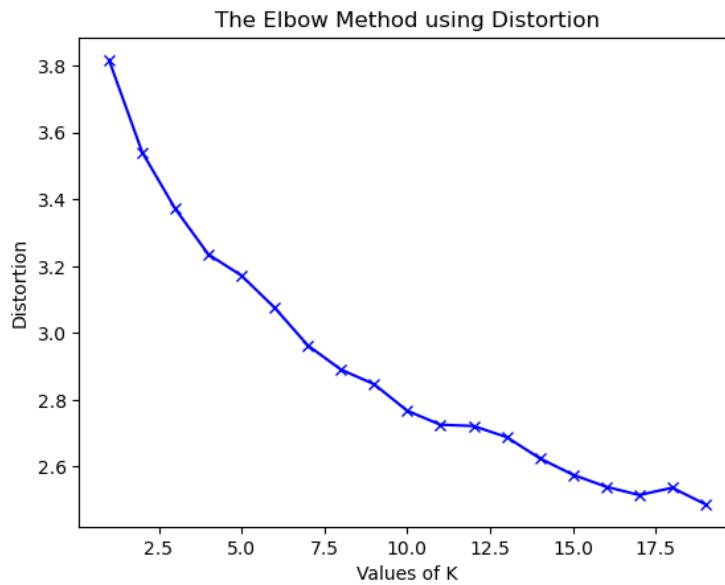
## ۱۱ خوشبندی (clustering) داده ها

همانطور که در بخش مرور ادبیات به آن اشاره شد، دو روش کلی برای دخیل کردن کار خوشبندی برای پیش بینی ریزش مشتریان در مطالعات وجود دارد و به آن اشاره شده است. مطابق یافته های [۱۰] روش اضافه کردن نتایج خوشبندی به عنوان ویژگی به دیتا است در ۱۳ مورد از ۱۶ مورد عملکرد کلی بهتری نسبت به حالت عدم استفاده از خوشبندی دارد. در نقطه مقابل، توسعه مدل های مختلف برای هر cluster تها در ۳ مورد از ۱۶ مورد نسبت به حالت بدون استفاده از خوشبندی بهتر جواب می دهد. در نتیجه در این پژوهه نتایج خوشبندی به عنوان یک ورودی ویژگی اضافه به مدل افزوده می شود.

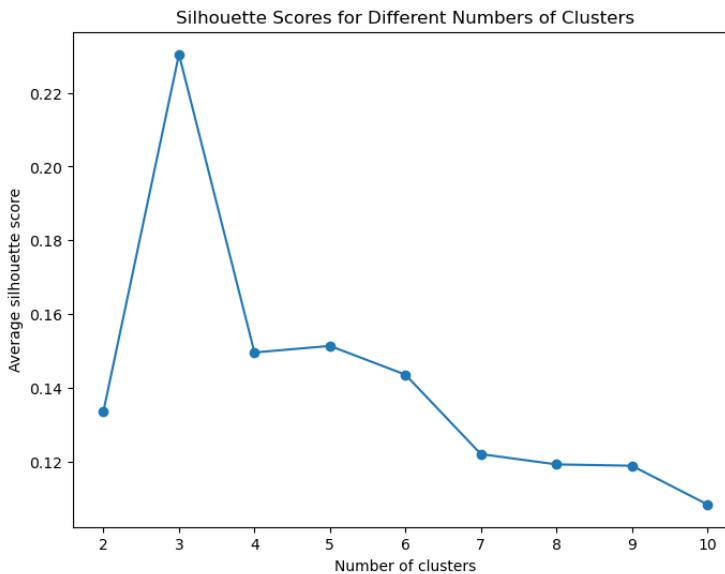
ابتدا باید با استفاده از استاندارد های شناخته شده تعداد خوشبندی که میخواهیم دیتابست را به آن تعداد خوشبندی کنیم مشخص کنیم. به طور کلی خوشبندی یک روش unsupervised روشنی است که نمی توان با دقت کامل و اطمینان تعداد خوشبندی هارا مشخص کرد و سپس بعد از آن، interpretability مربوط به نتایج خود یک بخش جدا است که در این پژوهه جزو اهداف نیست.

به کمک دو معیار Elbow و silhouette\_score تعداد خوشبندی را بررسی می کنیم. با بررسی inertia و distortion برای تعداد خوشبندی های ۱ تا ۲۰ که در شکل ۲۳ است شروع می کنیم. در این نمودار تعدادی خوشبندی معرفی می شود که یک نقطه شکست باشد. در این حالت نقطه خاصی با این مشخصات دیده نمی شود.

به عنوان معیار بعدی از silhouette\_score استفاده می شود. این معیار یکی از متريک های ميزان کارآمد و موثر بودن خوشبندی است و بسيار معروف تلقی می شود. اين معیار اندازه می گيرد هر نمونه به چه اندازه به سایر نمونه ها در خوشبندی نزدیک است. اين معیار بین -۱ تا ۱ است و هرچه بزرگتر باشد، عملکرد الگوريتم خوشبندی بهتر است. اين نمودار در شکل ۲۴ آمده است. مطابق اين نمودار بهينه تعداد خوشبندی ها ۳ معرفی می شود.



شکل ۲۳: معیار بازو برای تعیین تعداد خوشه بهینه



شکل ۲۴: معیار silhouette\_score برای تعیین تعداد خوشه بهینه

در نتیجه با توجه به شکل ۲۴ تعداد خوشه ها ۳ درنظر گرفته می شود. از الگوریتم K-means برای خوشه بندی استفاده می شود. الگوریتم K-means یکی از ساده‌ترین و پرکاربردترین روش‌های خوشه‌بندی است. هدف آن تقسیم  $n$  داده به  $k$  خوشه است به گونه‌ای که هر داده به نزدیک‌ترین مرکز خوشه تعلق داشته باشد. مراحل اجرای الگوریتم به شرح زیر است:

۱. ابتدایی سازی: انتخاب تصادفی  $k$  مرکز اولیه از داده‌ها.

۲. اختصاص: اختصاص هر داده به نزدیک‌ترین مرکز.

۳. بروزرسانی: محاسبه مجدد مرکز به عنوان میانگین داده‌های هر خوشه.

۴. تکرار: تکرار مراحل اختصاص و بروزرسانی تا زمانی که مرکز تغییر نکند.

این الگوریتم سعی می‌کند مجموع فواصل مربعی بین هر نقطه و مرکز خوشه‌اش را به حداقل برساند:

$$\min \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

الگوریتم K-means دارای مزایای متعددی است:

۱. سادگی: الگوریتم K-means ساده و قابل فهم است.

۲. کارایی: این الگوریتم از نظر محاسباتی کارا است.

۳. مقیاس‌پذیری: قابلیت کار با داده‌های بزرگ و پرابعاد را دارد.

با کمک [برنامه ۴](#) خوشه بندی انجام می‌شود. همانطور که دیده می‌شود برای داده آموزش از متدهای `fit_predict` و برای داده آزمون از متدهای `predict` استفاده شده است.

```
1 n_clusters = 3 # Number of clusters
2 kmeans = KMeans(n_clusters=n_clusters, random_state=54)
3 train_clusters = kmeans.fit_predict(X_train_scaled)
4
5 test_clusters = kmeans.predict(X_test_scaled)
6
7 X_train_with_clusters = pd.DataFrame(X_train_scaled)
8 X_train_with_clusters['Cluster'] = train_clusters
9 X_test_with_clusters = pd.DataFrame(X_test_scaled)
10 X_test_with_clusters['Cluster'] = test_clusters
```

Code 4: k-means clustering

پس از انجام خوشه بندی، توزیع داده‌ها در هر خوشه به صورت [شکل ۲۵](#) است.

می‌توان میانگین و میانه هر ویژگی را بر اساس دسته بندی هر خوشه انجام داد و به صورت فایل جدا با فرمت CSV ذخیره کرد. بخشی از این فایل به صورت [شکل ۲۶](#) است و می‌توان تفاوت هارا به صورت چشمی متوجه شد.

همچنین می‌توان به کمک Web-chart تقawat های هر خوشه را بیشتر به نمایش کشید.. این نمودار در [شکل ۲۷](#) آمده است. دیده می‌شود که همه ویژگی‌ها به طور یکسان در عمل خوشه بندی دخیل نیستند.



```

1 X_train_with_clusters['Cluster'].value_counts()
[199]    0   0s
...   Cluster
0    2217
2    1722
1     51
Name: count, dtype: int64

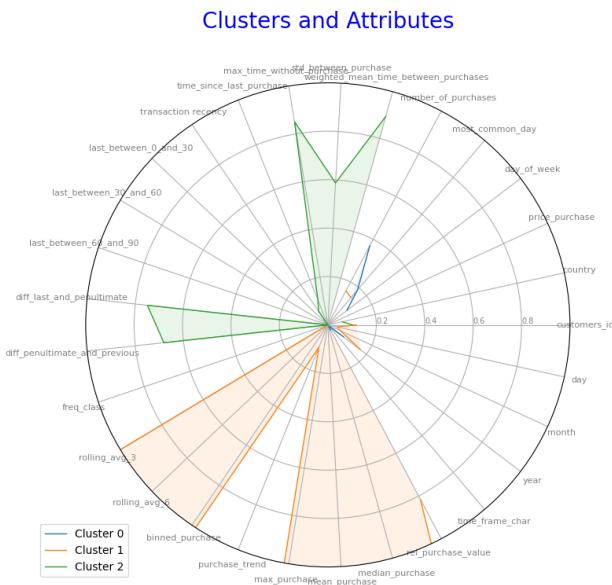
1 X_test_with_clusters['Cluster'].value_counts()
[199]    0s
...   Cluster
0     555
2     423
1     20
Name: count, dtype: int64

```

شکل ۲۵: توزیع داده ها در هر خوشه

	0	0	1	1	2	2	3
Cluster	mean	std	mean	std	mean	std	mean
0	0.281684	1.480944	-1.30522	1.168582	0.056214	1.155738	0.130678
1	16.9376	5.59788	3.948067	3.105112	-0.41079	1.138069	0.983644
2	-0.86429	0.771604	1.563483	1.20757	-0.06021	1.401031	-0.19737

شکل ۲۶: گروه بندی خوشه ها و نمایش میانگین و واریانس هر ویژگی



شکل ۲۷: نمایش تفاوت های خوشه ها به کمک web-chart

## ۱۲ تولید داده و بالانس کردن دیتاست

همانطور که در شکل ۱۸ مشاهده شد دیتاست این پروژه imbalanced است به طوری که تعداد برچسب های ۰ یا کسانی که خرید نداشته اند خیلی کمتر از برچسب های ۱ است.

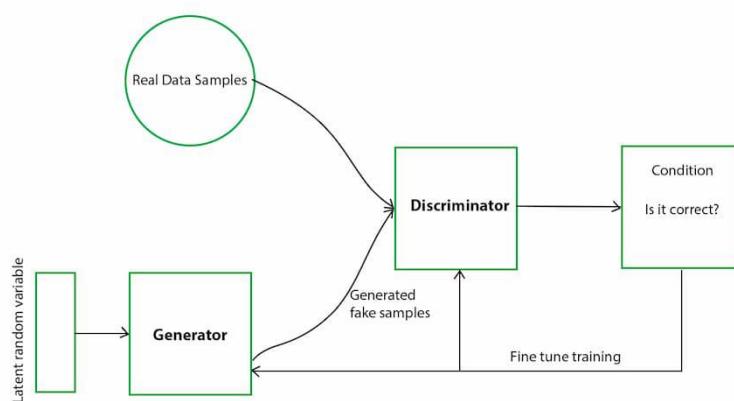


## ۱.۱۲ تولید داده به کمک GAN

شبکه‌های مولد تخاصمی (Generative Adversarial Networks or GANs) یک نوع معماری شبکه عصبی هستند که توسط ایان گودفلو و همکارانش در سال ۲۰۱۴ معرفی شدند [۱۳]. GANs از دو شبکه عصبی اصلی تشکیل شده‌اند: یک مولد (Generator) و یک تمیزدهنده (Discriminator) که به صورت همزمان آموزش می‌بینند و با یکدیگر رقابت می‌کنند. مولد از یک بردار نویز تصادفی (معمولًاً از توزیع گاووسی) به عنوان ورودی استفاده می‌کند و سعی می‌کند داده‌های مصنوعی تولید کند که شبیه به داده‌های واقعی باشند. از سوی دیگر، تمیزدهنده داده‌های ورودی را که می‌تواند داده‌های واقعی یا داده‌های تولید شده توسط مولد باشند دریافت می‌کند و سعی می‌کند بین داده‌های واقعی و مصنوعی تمایز قائل شود. در طی فرآیند آموزش، مولد تلاش می‌کند تا تمیزدهنده را فریب دهد و داده‌هایی تولید کند که تمیزدهنده نتواند آنها را از داده‌های واقعی تشخیص دهد. تمیزدهنده نیز سعی می‌کند توانایی خود در تشخیص داده‌های واقعی و مصنوعی را بهبود بخشد. این رقابت به تدریج باعث بهبود کیفیت داده‌های تولید شده توسط مولد می‌شود. GANs از طریق فرآیند آموزش تخاصمی به دست می‌آیند که شامل دو مرحله اصلی است. در مرحله اول، تمیزدهنده بر اساس داده‌های واقعی و داده‌های تولید شده توسط مولد آموزش می‌بیند و سعی می‌کند داده‌های واقعی را از داده‌های مصنوعی تمایز دهد. در مرحله دوم، مولد بر اساس بازخوردی که از تمیزدهنده دریافت می‌کند، سعی می‌کند داده‌هایی تولید کند که تمیزدهنده را فریب دهد. این فرایند تازه‌مانی که تعادل بین مولد و تمیزدهنده برقرار شود ادامه می‌یابد.

استفاده از GANs برای تولید داده‌های مصنوعی مزایای متعددی دارد. این شبکه‌ها قادر به تولید داده‌هایی هستند که بسیار شبیه به داده‌های واقعی هستند و از لحاظ آماری ویژگی‌های مشابهی دارند. همچنین، GANs در زمینه‌های مختلفی مانند تولید تصاویر، داده‌های پزشکی، بهبود کیفیت تصاویر، تولید موسیقی و ویدئو وغیره کاربرد دارند. یکی از ویژگی‌های برجسته GANs توانایی آنها در یادگیری توزیع داده‌های پیچیده است که به آنها امکان می‌دهد داده‌هایی با توزیع‌های چندمتغیره تولید کنند.

نمای کلی این الگوریتم در [شکل ۲۸](#) آمده است. منبع



شکل ۲۸: نمای کلی الگوریتم GAN

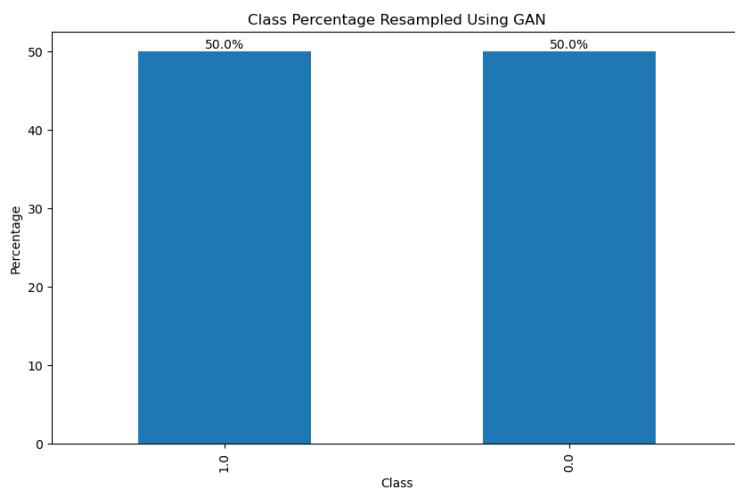
معماری شبکه مولد متخصص به صورت زیر است:

معماری مولد (Generator): این شبکه از یک بردار نویز تصادفی به عنوان ورودی استفاده کرده و سعی می کند داده مصنوعی تولید کند به طرزی که شبکه Discriminator تفاوت آن با داده واقعی را تشخیص ندهد. این شبکه دارای یک لایه ورودی با ۱۲۸ نرون و تابع فعال ساز Leaky ReLU است. لایه میانی دارای ۲۵۶ نرون و تابع فعال ساز ReLU است و لایه خروجی برابر با تعداد نرون یکسان با تعدا ویژگی ها است و تابع فعال سازی تانژانت هیپربولیک دارد. در میان این لایه ها نیز برای لایه های نرمال ساز برای بهبود پایداری وجود دارد.

معماری تمییزدهنده (Discriminator): این شبکه داده های ورودی را دریافت می کند و سعی می کند بین داده های واقعی و داده های تولید شده تمییز قائل شود. این شبکه دارای لایه ورودی با ۱۲۸ نرون و تابع فعال ساز Leaky ReLU است. دو لایه میانی هریک با ۶۴ نرون و ۳۲ نرون و توابع فعال ساز Leaky ReLU هستند. لایه خروجی به تعداد ویژگی ها نرون دارد و تابع فعال ساز آن تانژانت هیپربولیک است.

در طول فرآیند آموزش ابتدا تمییز دهنده با استفاده از داده های واقعی و داده های تولید شده توسط مولد آموزش می بیند سپس مولد با استفاده از نویز تصادفی آموزش میبیند تا داده هایی تولید کند که تمییز دهنده نتواند آن را از داده واقعی تمییز دهد.

برای تولید داده مصنوعی از مولد استفاده می شود که با ورودی نویز تصادفی، داده هایی مشابه با داده واقعی تولید می کند. این تولید تا جایی ادامه می یابد که داده balanced شود. به نوعی که در [شکل ۲۹](#) دیتاست پس از اضافه کردن داده های غیرواقعی ساخته شده توسط GAN را نشان می دهد.



شکل ۲۹: برچسب ها پس از اضافه کردن خروجی GAN

۲.۱۲ تولید داده به کمک **Variational Auto Encoder**۳.۱۲ تولید داده به کمک **SMOTE**

## منابع

- [1] Ahn, Jaehyun, et al. "A survey on churn analysis in various business domains." *IEEE Access* 8 (2020)
- [2] Geiler, Louis, Séverine Affeldt, and Mohamed Nadif. "A survey on machine learning methods for churn prediction." *International Journal of Data Science and Analytics* 14.3 (2022)
- [3] Tran, Hoang, Ngoc Le, and Van-Ho Nguyen. "CUSTOMER CHURN PREDICTION IN THE BANKING SECTOR USING MACHINE LEARNING-BASED CLASSIFICATION MODELS." *Interdisciplinary Journal of Information, Knowledge & Management* 18 (2023).
- [4] Imani, Mehdi, and Hamid Reza Arabnia. "Hyperparameter optimization and combined data sampling techniques in machine learning for customer churn prediction: a comparative analysis." *Technologies* 11.6 (2023)
- [5] Martínez, Andrés, et al. "A machine learning framework for customer purchase prediction in the non-contractual setting." *European Journal of Operational Research* 281.3 (2020)
- [6] Park, Jungryeol, Sundong Kwon, and Seon-Phil Jeong. "A study on improving turnover intention forecasting by solving imbalanced data problems: focusing on SMOTE and generative adversarial networks." *Journal of Big Data* 10.1 (2023)
- [7] Li, Bo, and Jiuzuo Xie. "Study on the Prediction of Imbalanced Bank Customer Churn Based on Generative Adversarial Network." *Journal of Physics: Conference Series*. Vol. 1624. No. 3. IOP Publishing, 2020.
- [8] Ayoub, Shahnawaz, et al. "Adversarial approaches to tackle imbalanced data in machine learning." *Sustainability* 15.9 (2023)
- [9] Zheng, Hanming, Ling Luo, and Goce Ristanoski. "A clustering-prediction pipeline for customer churn analysis." *Knowledge Science, Engineering and Management: 14th International Conference, KSEM 2021, Tokyo, Japan, August 14–16, 2021, Proceedings, Part III* 14. Springer International Publishing, 2021.
- [10] Bose, Indranil, and Xi Chen. "Hybrid models using unsupervised clustering for prediction of customer churn." *Journal of Organizational Computing and Electronic Commerce* 19.2 (2009)



- [11] Bilal, Syed Fakhar, et al. "An ensemble based approach using a combination of clustering and classification algorithms to enhance customer churn prediction in telecom industry." PeerJ Computer Science 8 (2022)
- [12] Hasumoto, K., Goto, M. Predicting customer churn for platform businesses: using latent variables of variational autoencoder as consumers' purchasing behavior. Neural Comput & Applic 34, 18525–18541 (2022)
- [13] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).