دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده علوم کامپیوتر

تمرین هفتم

شناسایی داده پرت و ناهنجاری، مقایسه روشهای متوازن‌سازی داده و ارائه ارزیابی

نگارش
ثمین مهدی پور
۹۸۳۹۰۳۹

استاد راهنما
دکتر قطعی

استادمشاور
دکتر یوسفی مهر

دی ۱۴۰۲

چکیده

این پروژه شامل چندین بخش در حوزه تحلیل داده و یادگیری ماشین است. در ابتدا، داده‌های زمانی تصادفی از چند دسته از داده‌ها به دست آمدند. سپس، تحلیل و تصحیح داده‌ها صورت گرفت و روش‌های کاهش ابعاد مثل PCA و t-SNE برای مشاهده توزیع داده‌ها در فضای کم‌ابعاد استفاده شدند. در مرحله بعدی، تشخیص انومالی و تخریب داده با استفاده از روش‌های متنوعی از جمله One-Class SVM و Local Outlier Factor انجام شد.

سپس، رویکردهای نوین برای افزایش داده‌های اقلیمی به کار گرفته شدند. از تکنیک‌های OverSampling مانند Random OverSampler و SMOTE برای تقویت داده‌های اقلیمی کمی و از تکنیک‌های UnderSampling مانند Random UnderSampler برای تعادل کلاس‌ها استفاده شد. همچنین، مدل کلاسیفیکیشن جنگل تصادفی بهینه‌سازی شد و عملکرد آن بر روی داده‌های تقویت شده و ناتوازه مورد بررسی قرار گرفت.

نهایتاً، برای مدل‌های زمانی از یک شبکه LSTM برای شناسایی الگوهای زمانی در داده‌ها استفاده شد. این شبکه LSTM با ورودی‌ها و برچسب‌های ساخته شده از توابع زمانی داده‌ها آموزش داده شد تا بتواند الگوهای مهم زمانی را در داده‌ها استخراج کند.

این تحلیل‌ها و مدل‌ها در جمع‌بندی نشان می‌دهند که چگونه از تکنیک‌های متنوع در تحلیل داده‌ها، تقویت کلاس‌های کم‌تعداد، و شناسایی الگوهای زمانی می‌توان بهبود عملکرد و اطمینان در حوزه تحلیل داده‌های پیچیده مرتبط با اقلیم به دست آورد.

واژه‌های کلیدی:

شناسایی ناهنجاری، متوازن سازی داده، داده سری زمانی، کاهش بعد، داده خارج از محدوده

فهرست مطالب

صفحه

صفحه

فهرست اشکال

No table of figures entries found.

فصل اول

مقدمه

مقدمه

شناسایی ناهنجاری، یکی از حوزه‌های کلان یادگیری ماشین، در واقعیت‌های مختلف از اهمیت بسیاری برخوردار است. این مسئله زمانی اهمیت پیدا می‌کند که در یک مجموعه داده، اطلاعات ناهنجار یا ایستا از نظر توزیع متفاوت با سایر داده‌ها باشند. به عبارت دیگر، هنگامی که ویژگی‌های یک نمونه یا یک گروه از نمونه‌ها از حد معمول خود کم یا زیاد باشند.

در اولین بخش از این پروژه به شناسایی این داده ها پرداختیم:

۱-۱-شناسایی ناهنجاری

۱-۱-۱- Boxplots

شناسایی داده‌های نوعی (Outlier) با استفاده از نمودار جعبه (Boxplot):

نمودار جعبه یا "باکس‌پلات" یک ابزار تصویری قدرتمند در شناسایی داده‌های نوعی و اطلاعات آماری مربوط به توزیع داده‌ها است. این نمودار به ویژه برای تشخیص داده‌های ایستا و اطلاعات پرتی در مجموعه‌های داده بسیار مفید است.

- ساختار نمودار جعبه:

یک نمودار جعبه از اجزای مختلفی تشکیل شده است که هرکدام اطلاعاتی خاص را ارائه می‌دهند:

۱. باکس (Box):

- باکس در وسط نمودار قرار دارد و حاوی ۵۰ درصد داده‌ها (کوچکترین تا بزرگترین مقادیر) است.

- انتهای پایین باکس نشان‌دهنده کوچکترین مقدار و انتهای بالایی نشان‌دهنده بزرگترین مقدار در مجموعه داده است.

۲. ویسکرز (Whiskers):

- ویسکرز خطوطی هستند که از هر سمت باکس خارج می‌شوند و محدوده داده‌ها را نشان می‌دهند.

- اغلب با استفاده از قوانین محاسبه خاص، محدوده داده‌ها ویسکرز مشخص می‌شود.

۳. نقاط پرت (Outliers):

- نقاطی که خارج از ویسکرز هستند و به عنوان داده‌های پرت شناخته می‌شوند.

- این نقاط می‌توانند به شناسایی داده‌های نوعی و اطلاعات غیرعادی کمک کنند.

استفاده از نمودار جعبه در شناسایی Outlier:

- اگر داده‌های پرت وجود داشته باشند، نمودار جعبه به سرعت تشخیص آنها را ممکن می‌سازد.

- افزایش فاصله بین ویسکرز یا افت فاصله باکس از ویسکرز می‌تواند نشان‌دهنده وجود داده‌های پرت باشد.

- نقاط پرت به عنوان نقاطی که خارج از محدوده ویسکرز هستند، قابل مشاهده هستند.

- نکات مهم:

- توجه به اندازه و مقیاس:

برای مقایسه درست داده‌ها و شناسایی پرتی‌ها، توجه به اندازه و مقیاس داده‌ها بسیار حائز اهمیت است.

- تحلیل همراه با معیارهای دیگر:

همیشه نباید تنها به نمودار جعبه اکتفا کرد، بلکه از معیارهای آماری دیگر همچون انحراف معیار و میانگین نیز برای تحلیل دقیق‌تر استفاده کرد.

- استفاده در زمینه‌های مختلف:

نمودار جعبه در زمینه‌های مختلف از جمله علوم اقتصاد، زمینه پزشکی، مهندسی و غیره کاربرد دارد و به راحتی قابل فهم برای افراد غیرتخصصی نیز است.

با استفاده از این نمودار، می‌توان به سادگی داده‌های پرت و ناهنجور در مجموعه‌های داده را تشخیص داد و در تحلیل‌ها و تصمیم‌گیری‌ها از دقت بیشتری برخوردار شد.

IQR -۲-۱-۱

استفاده از بردار بین‌کوارتیل (IQR) در شناسایی داده‌های ناهنجار:

بردار بین‌کوارتیل یا IQR یک معیار آماری است که در تحلیل داده‌ها برای شناسایی داده‌های پرت و ناهنجار به کار می‌رود. این روش مبتنی بر توزیع فاصله داده‌ها در یک مجموعه است و میزان پراکندگی آنها را ارزیابی می‌کند.

- استفاده از بردار بین‌کوارتیل در شناسایی ناهنجار:

۱. تعیین بازه IQR:

- بازه IQR به عنوان تفاوت بین کوارتیل بالا (Q3) و کوارتیل پایین (Q1) تعریف می‌شود: IQR = Q3 - Q1.

۲. حدود ناهنجاری:

- اغلب یک ضریب ضرب‌شده در IQR به عنوان معیاری برای تشخیص داده‌های ناهنجار در نظر گرفته می‌شود.

- حدودی مشخص می‌شوند که داده‌هایی خارج از این حدود به عنوان ناهنجار در نظر گرفته می‌شوند.

۳. تشخیص ناهنجارها:

- داده‌هایی که خارج از حدود تعیین شده در مرحله قبل هستند، به عنوان داده‌های ناهنجار شناخته می‌شوند.

نکات مهم:

- مقیاس‌پذیری:

- IQR بر مبنای توزیع داده‌ها است، بنابراین مقیاس‌پذیر بوده و مستقل از مقیاس داده‌ها عمل می‌کند.

- مقاومت در برابر داده‌های پرت:

- به دلیل استفاده از کوارتیل‌ها که مقاومت خوبی در برابر داده‌های پرت دارند، این روش به‌طور کلی مقاومت زیادی در برابر نوسانات ناشی از داده‌های پرت دارد.

- محدوده‌های مشخص:

- این روش نیازمند تعیین حدودی برای شناسایی داده‌های ناهنجار است که ممکن است نیاز به تنظیم دقیق داشته باشد.

استفاده از IQR به عنوان یک معیار اندازه‌گیری فاصله داده‌ها و شناسایی داده‌های ناهنجار به صورت ساده و قابل فهمی امکان‌پذیر است و این روش معمولاً در کنار روش‌های دیگر مورد استفاده قرار می‌گیرد تا تصمیم‌گیری‌ها دقیق‌تر و قابل اعتمادتر باشند.

Z-score -۳-۱-۱

استفاده از امتیاز Z یکی از روش‌های معمول برای شناسایی داده‌های ناهنجار است که بر مبنای انحراف از میانگین استاندارد داده‌ها ارزیابی می‌شود. این روش مبتنی بر توزیع نرمال داده‌ها است و برای تشخیص داده‌هایی که از الگوی متناسب با اکثر داده‌ها خارج شده‌اند، به‌طور گسترده‌ای مورد استفاده قرار می‌گیرد.

۱. محاسبه میانگین و انحراف معیار:

- میانگین و انحراف معیار داده‌ها محاسبه می‌شوند.

۲. محاسبه امتیاز Z:

- برای هر داده، امتیاز Z به عنوان فاصله انحراف معیاری از میانگین محاسبه می‌شود:

$$Z = \frac{(X - \text{میانگین})}{\text{انحراف معیار}}$$

۳. تعیین آستانه ناهنجاری:

- یک آستانه تعیین می‌شود که به‌عنوان حد مرز برای شناسایی داده‌های ناهنجار استفاده می‌شود. داده‌هایی که امتیاز Z آنها از این آستانه بیشتر یا کمتر باشد، به عنوان داده‌های ناهنجار در نظر گرفته می‌شوند.

۴. تشخیص ناهنجارها:

- داده‌هایی که امتیاز Z آنها از آستانه تعیین شده خارج شوند، به عنوان داده‌های ناهنجار شناخته می‌شوند.

- نکات مهم:

- توزیع نرمال:

- استفاده از امتیاز Z منوط به توزیع نرمال داده‌ها است، بنابراین در صورتی که توزیع داده‌ها نرمال نباشد، ممکن است این روش به دقت کمتری برخوردار باشد.

- آستانه تعیین آزاد:

- تعیین آستانه برای شناسایی داده‌های ناهنجار یک چالش است و معمولاً به تجربه و تصمیم‌گیری متخصص بستگی دارد.

- حساسیت به داده‌های پرت:

- داده‌های پرت می‌توانند تأثیر قابل توجهی بر محاسبات امتیاز Z داشته باشند و در نتیجه، حساسیت به داده‌های پرت وجود دارد.

- استفاده به همراه روش‌های دیگر:

- استفاده از امتیاز Z معمولاً به‌طور همزمان با روش‌های دیگر شناسایی ناهنجاری مورد استفاده قرار می‌گیرد تا نتایج دقیق‌تر و کارآمدتری حاصل شود.

بر اساس این روش، داده‌هایی که امتیاز Z آنها از آستانه تعیین شده خارج شوند، به عنوان داده‌های ناهنجار در نظر گرفته می‌شوند و ممکن است در تحلیل دقیق‌تر داده‌ها و شناسایی الگوهای ناهنجار مورد استفاده قرار گیرد.

۱-۲-شناسایی Anomalies

۱-۲-۱ One-class SVM

تشخیص ناهنجاری یکی از روش‌های مهم در حوزه یادگیری ماشین است که به وسیله ماشین‌های بردار پشتیبان (SVM) انجام می‌شود. این روش بر اساس ایده اصلی SVM برای مسائل طبقه‌بندی توسعه یافته و برای شناسایی داده‌های ناهنجار یا پرت از نمونه‌های معمولی استفاده می‌شود.

مراحل کلی تشخیص ناهنجاری با استفاده از SVM:

۱. آموزش مدل SVM:

- یک مدل SVM بر روی داده‌های معمولی آموزش داده می‌شود. این داده‌های معمولی باید نمونه‌های عادی یا نمونه‌هایی با ویژگی‌های نرمال را نمایند.

۲. تعیین حد مرز:

- مدل آموزش دیده، یک حد مرز (threshold) را برای تفکیک داده‌های عادی از داده‌های ناهنجار تعیین می‌کند. این حد مرز معمولاً بر اساس فاصله داده‌ها از هایپرصفحه جداکننده SVM تعیین می‌شود.

۳. تست و شناسایی ناهنجاری:

- با استفاده از داده‌های تست، مدل SVM بررسی می‌کند که هر داده چقدر از هایپرصفحه جداکننده فاصله دارد. داده‌هایی که از حد مرز بیشتر انحراف دارند، به عنوان داده‌های ناهنجار شناسایی می‌شوند.

نکات مهم:

- تعادل میان نمونه‌ها:

- اهمیت تعادل میان نمونه‌های عادی و ناهنجار در آموزش مدل بسیار حائز اهمیت است تا مدل به‌درستی بتواند داده‌های ناهنجار را تشخیص دهد.

- انتخاب هایپرپارامترها:

- تنظیم صحیح هایپرپارامترهای مدل SVM، از جمله پارامترهای مربوط به هسته و حاشیه، می‌تواند به بهبود عملکرد مدل در شناسایی داده‌های ناهنجار کمک کند.

- آستانه تصمیم:

- تعیین آستانه تصمیم برای انتخاب دقیق ناهنجارها نیز از اهمیت بالایی برخوردار است و نیاز به تجربه و ارزیابی دقیق دارد.

استفاده از SVM برای تشخیص ناهنجاری می‌تواند به‌عنوان یک روش قوی و کارآمد در حوزه تشخیص ناهنجاری مورد استفاده قرار گیرد.

۱-۲-۲- Local Outlier Factor (LOF) algorithm

Local Outlier Factor (LOF) یک الگوریتم مفهوم و قوی در زمینه شناسایی ناهنجاری (Anomaly Detection) است که توسط Markus M. Breunig و همکارانش در سال ۲۰۰۰ معرفی شد. این الگوریتم به طور خاص برای شناسایی ناهنجاری‌های محلی در دیتاست‌ها کاربرد دارد.

مراحل کلی الگوریتم LOF:

۱. محاسبه فاصله‌ها:

- برای هر نقطه در دیتاست، محاسبه فاصله‌های آن با سایر نقاط انجام می‌شود. فاصله معمولاً با استفاده از معیار فاصله اقلیدسی یا معیارهای دیگر اندازه‌گیری می‌شود.

۲. محاسبه میزان ناهنجاری محلی (Local Outlier Factor - LOF):

- برای هر نقطه، LOF محلی محاسبه می‌شود. LOF بر اساس نسبت فاصله نقطه مورد نظر با همسایگان آن به فاصله میان همسایگان هر همسایه بنا شده است. اگر نقطه‌ای فاصله بیشتری نسبت به همسایگان داشته باشد، LOF بزرگتر خواهد بود و نشان‌دهنده احتمال ناهنجاری محلی آن نقطه است.

۳. تصمیم گیری درباره ناهنجاری:

- با استفاده از مقدار LOF محلی برای هر نقطه، می‌توان تصمیم گرفت که نقاطی که LOF آن‌ها بیشتر از یک آستانه مشخص باشد به عنوان ناهنجار در نظر گرفته شوند.

ویژگی‌ها و نکات:

- حساسیت به نواحی محلی:

- LOF به خوبی با داده‌هایی که ناهنجاری در نواحی محلی دارند، عمل می‌کند و نه تنها به نقاطی که در تمام دیتاست ناهنجار هستند تمرکز دارد.

- تطبیق به توزیع نواحی:

- LOF قادر است به شناسایی ناهنجاری‌هایی که به توزیع داده در نواحی خاص وابسته‌اند، بپردازد.

- آستانه تصمیم:

- انتخاب آستانه LOF یکی از چالش‌های این الگوریتم است و باید با دقت تنظیم شود.

- سرعت اجرا:

- LOF به دلیل نیاز به محاسبات فاصله با تمام نقاط دیتاست، در مقیاس داده‌های بزرگ ممکن است زمان‌بر باشد.

استفاده از LOF به عنوان یک روش ارزشمند در تشخیص ناهنجاری، به ویژه در داده‌هایی با نواحی محلی ناهنجار موثر است.

۱-۳- متوازن سازی

در این پروژه، به منظور بهبود عملکرد مدل‌های کلاسیفیکیشن برای داده‌های اقلیمی کمی، از رویکردهای نوین در فرآیند متوازن‌سازی و افزایش حجم داده‌ها استفاده شد. این رویکردها به صورت زیر توضیح داده می‌شوند:

۱. متوازن‌سازی با سپس:

- استفاده از تکنیک‌های متوازن‌سازی با سپس، که شامل ترکیب تکنیک‌های OverSampling و UnderSampling می‌شود. این رویکرد به منظور تقویت داده‌های اقلیمی کمی و تعادل کلاس‌ها مورد استفاده قرار گرفت.

۲. OverSampling با استفاده از Random OverSampler و SMOTE:

- از تکنیک‌های OverSampling مانند Random OverSampler و Synthetic Minority Over-sampling Technique (SMOTE) برای افزایش تعداد نمونه‌های کلاس‌های اقلیمی کمی استفاده شد. این تکنیک‌ها با افزایش تنوع داده‌ها و جلوگیری از ناتوازه بهبود عملکرد مدل را هدفمندتر می‌کنند.

۳. UnderSampling با استفاده از Random UnderSampler:

- از تکنیک‌های UnderSampling مانند Random UnderSampler برای حذف تعدادی از نمونه‌های کلاس اکثریت (Overrepresented) استفاده شد. این کار با کاهش تعداد نمونه‌های کلاس اکثریت، موجب تعادل بیشتری در توزیع کلاس‌ها می‌شود.

۴. بهینه‌سازی مدل جنگل تصادفی:

- مدل کلاسیفیکیشن جنگل تصادفی بهینه‌سازی شد تا بر روی داده‌های تقویت شده و تعادل‌یافته عمل کند. این بهینه‌سازی با هدف بهبود دقت و عملکرد مدل در شناسایی کلاس‌های مختلف در مواجهه با داده‌های ناهنجار و ناتوازه انجام شد.

این رویکردها باعث افزایش کارایی مدل‌های کلاسیفیکیشن در حوزه‌ی داده‌های اقلیمی شده و بهبود قابل‌توجهی در شناسایی ناهنجاری‌ها و دسته‌بندی کلاس‌ها داشته است.

۱-۴-بررسی ویژگی داده های سری زمانی

نهایتاً، برای مدل‌های زمانی از یک شبکه LSTM برای شناسایی الگوهای زمانی در داده‌ها استفاده شد. این شبکه LSTM با ورودی‌ها و برچسب‌های ساخته شده از توابع زمانی داده‌ها آموزش داده شد تا بتواند الگوهای مهم زمانی را در داده‌ها استخراج کند.

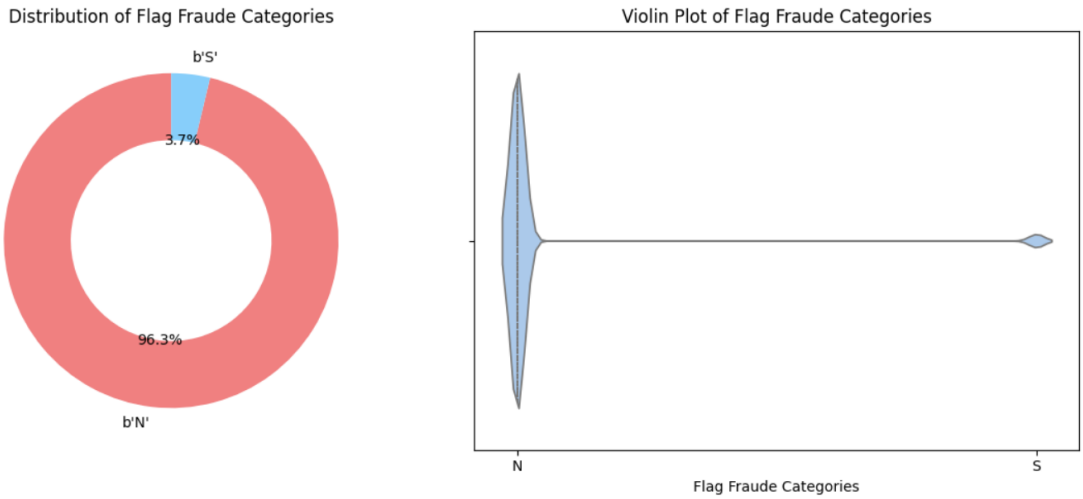
با بهره‌گیری از مدل LSTM، توانستیم بهبودی در شناسایی الگوهای زمانی از داده‌ها حاصل کنیم. این مدل توانایی خود را در تشخیص و تعیین الگوهای پیچیده و وابستگی‌های زمانی را در داده‌ها به نمایش گذاشته و از اهمیت ویژگی‌های زمانی در دسته‌بندی به خوبی بهره‌مند شده است. این رویکرد به ما امکان می‌دهد تا بهبودهای قابل‌توجهی در دقت و کارایی در شناسایی ناهنجاری‌ها و الگوهای زمانی داشته باشیم و بهترین عملکرد را از مدل‌های زمانی به دست آوریم.

فصل دوم

پیاده‌سازی

۲-۱- بررسی و پیش پردازشو مصورسازی مجموعه داده

پس از لود کردن مجموعه داده و بررسی کلی دیتاست شروع به مصور سازی آن کردیم.



این نمودارها دو نمایش مختلف از توزیع داده‌ها در مورد دسته‌بندی "Flag Fraude" را ارائه می‌دهند:

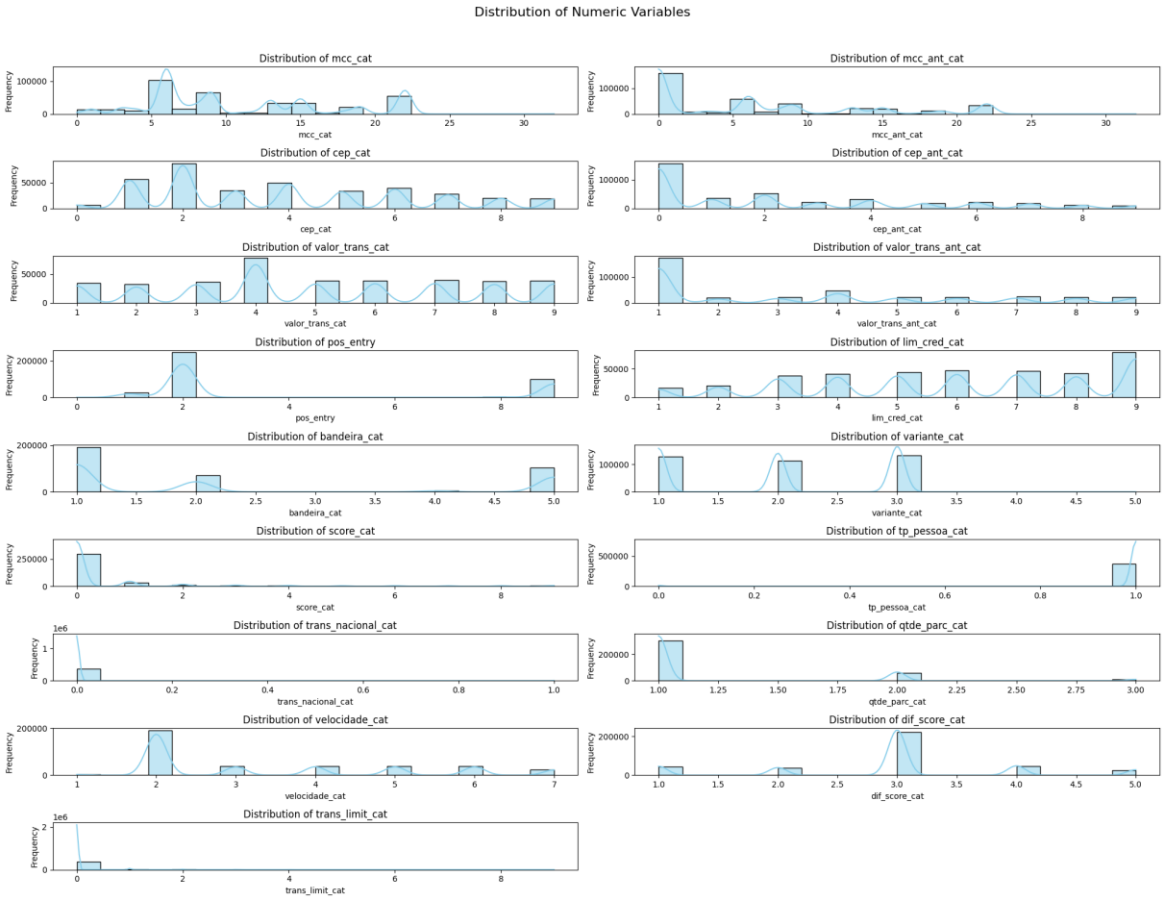
۱. نمودار دایره‌ای (Pie Chart):

در این نمودار، بخش‌های مختلف دایره نشان‌دهنده تعداد نمونه‌ها در هر دسته از "Flag Fraude" هستند. برای هر دسته، درصد تعداد نمونه‌های آن دسته به کل نمونه‌ها نمایش داده شده است. در اینجا از دو رنگ مختلف (قرمز و آبی) برای نمایش دو دسته مختلف استفاده شده است. دایره سفید در وسط نمودار به عنوان یک حلقه میانی اضافه شده است تا نمودار به شکل یک دونات نشان داده شود.

۲. نمودار ویولین (Violin Plot):

در این نمودار، برای هر دسته از "Flag Fraude" یک نمودار ویولین نمایش داده شده است. این نمودارها نشان‌دهنده توزیع احتمالی داده‌ها در هر دسته هستند. قسمت‌های متمرکز نمودار ویولین (quartiles) میانگین و پراکندگی داده را نمایش می‌دهند. رنگهای مختلف برای نمایش دسته‌های مختلف استفاده شده‌اند.

هر دو نمودار به صورت کلی به تحلیل توزیع داده‌ها در مورد دسته‌بندی "Flag Fraude" کمک می‌کنند.



این نمودارها به شما یک نگاه کلی از توزیع متغیرهای عددی در دیتافریم فراهم می‌کنند. موارد زیر نشان‌دهنده جوانب مختلف این کد هستند:

۱. تعداد ستون‌ها و ردیف‌ها:

- تعداد ستون‌ها: حداکثر ۲ ستون (num_cols) برای هر سطر در نمودارها.

- تعداد ردیف‌ها: به تعداد لازم برای جاگیری تمام متغیرهای عددی در دیتافریم (num_rows).

۲. ساختار نمودارها:

- برای هر متغیر عددی یک هیستوگرام رسم شده است که توزیع مقادیر آن متغیر را نمایش می‌دهد.

- تعداد باکس‌ها (bins) برای هر هیستوگرام ۲۰ در نظر گرفته شده است.

- همچنین، در هر نمودار، منحنی تخمین چگالی احتمال (KDE) نیز رسم شده است.

۳. آرایش عنوان و محورها:

- هر نمودار عنوانی دارد که نام متغیر مرتبط با آن را نشان می‌دهد.

- محور افقی (X) نام متغیر و محور عمودی (Y) تعداد تکرارها (Frequency) را نمایش می‌دهد.

۴. آرایش کلی:

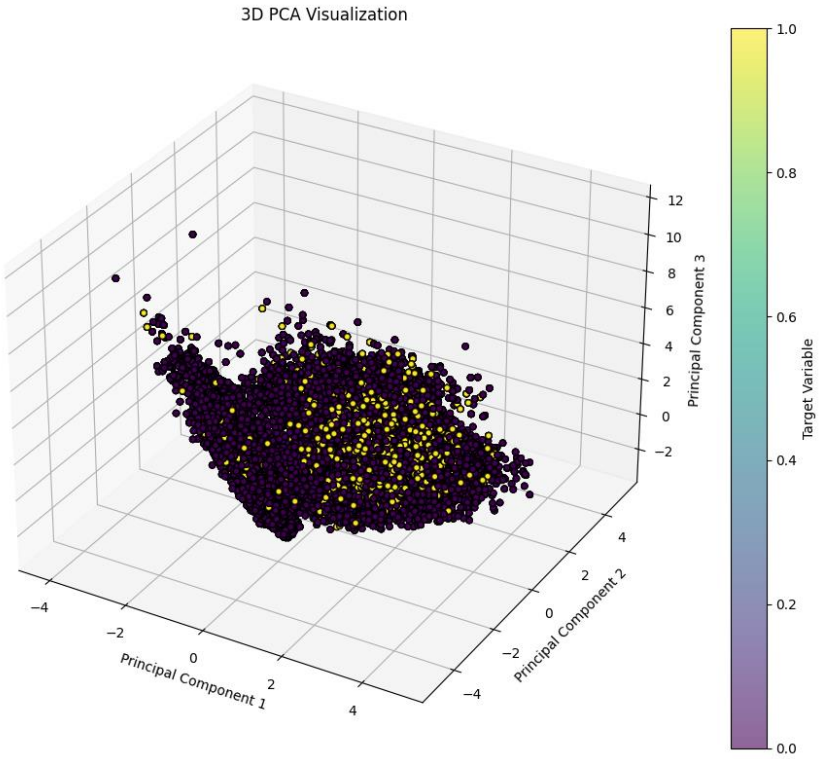
- اگر تعداد متغیرها فرد باشد، زیرنمودار اضافی حذف شده و آرایه نمودارها به شکل مناسبی تنظیم شده است.

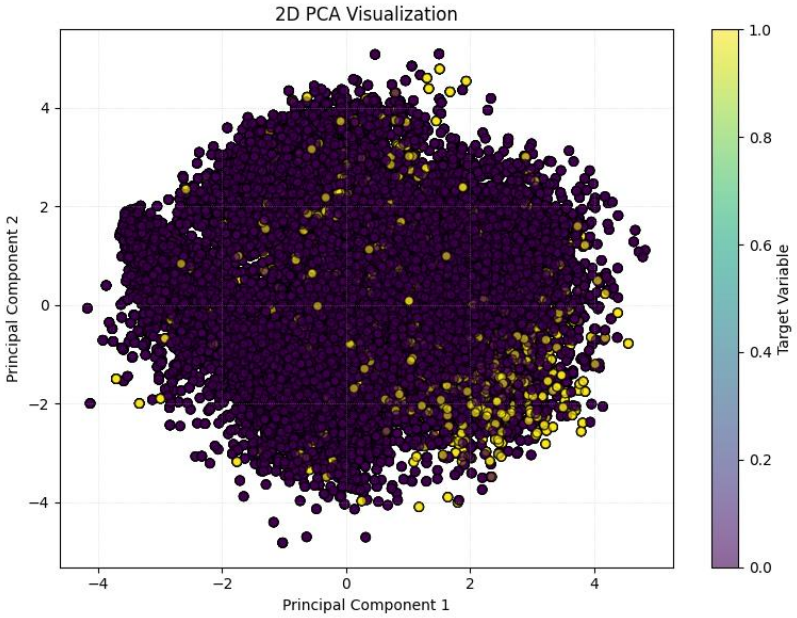
۵. آپدیت لی‌اوت:

- توابع `tight_layout` و `delaxes` برای بهبود لی‌اوت و حذف نمودارهای اضافی به کار گرفته شده‌اند.

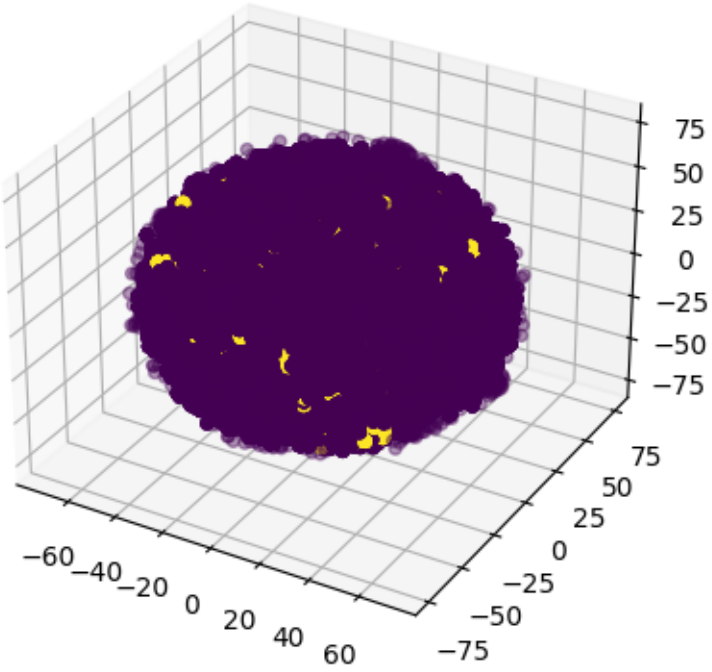
۲-۲-مصورسازی بعد از کاهش بعد

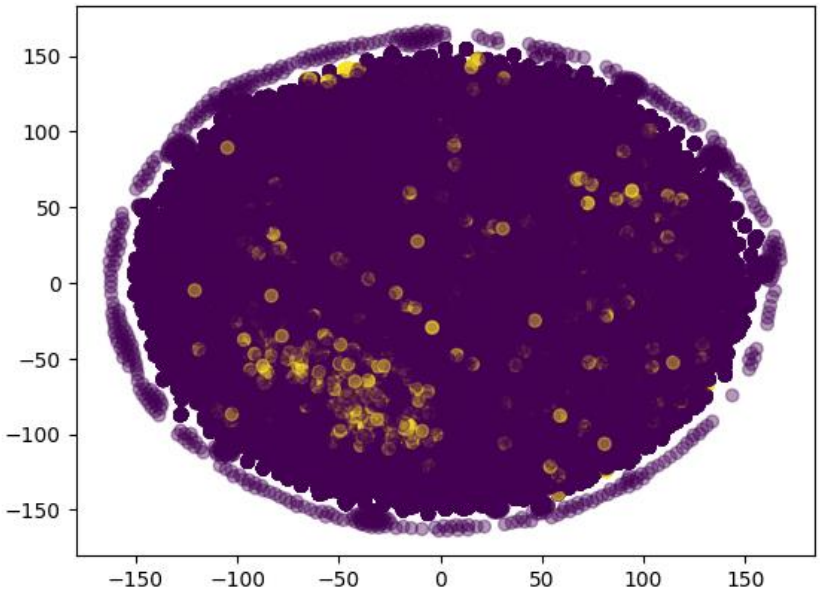
PCA-۱-۲-۲





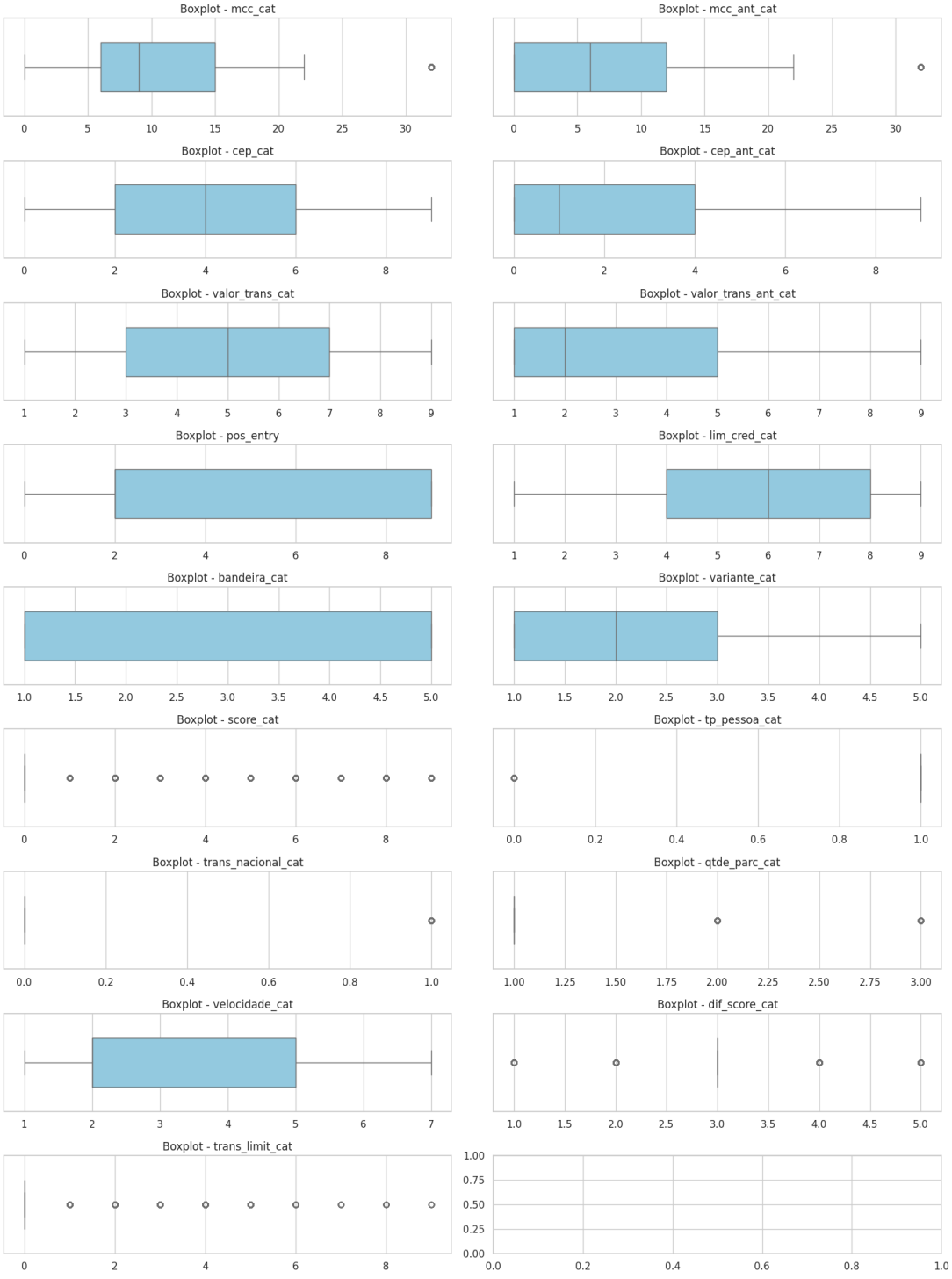
t-SNE-۲-۲-۲


75
50
25
0
−25
−50
−75
−60
−40
−20
0
20
40
60
75
50
25
0
−25
−50
−75

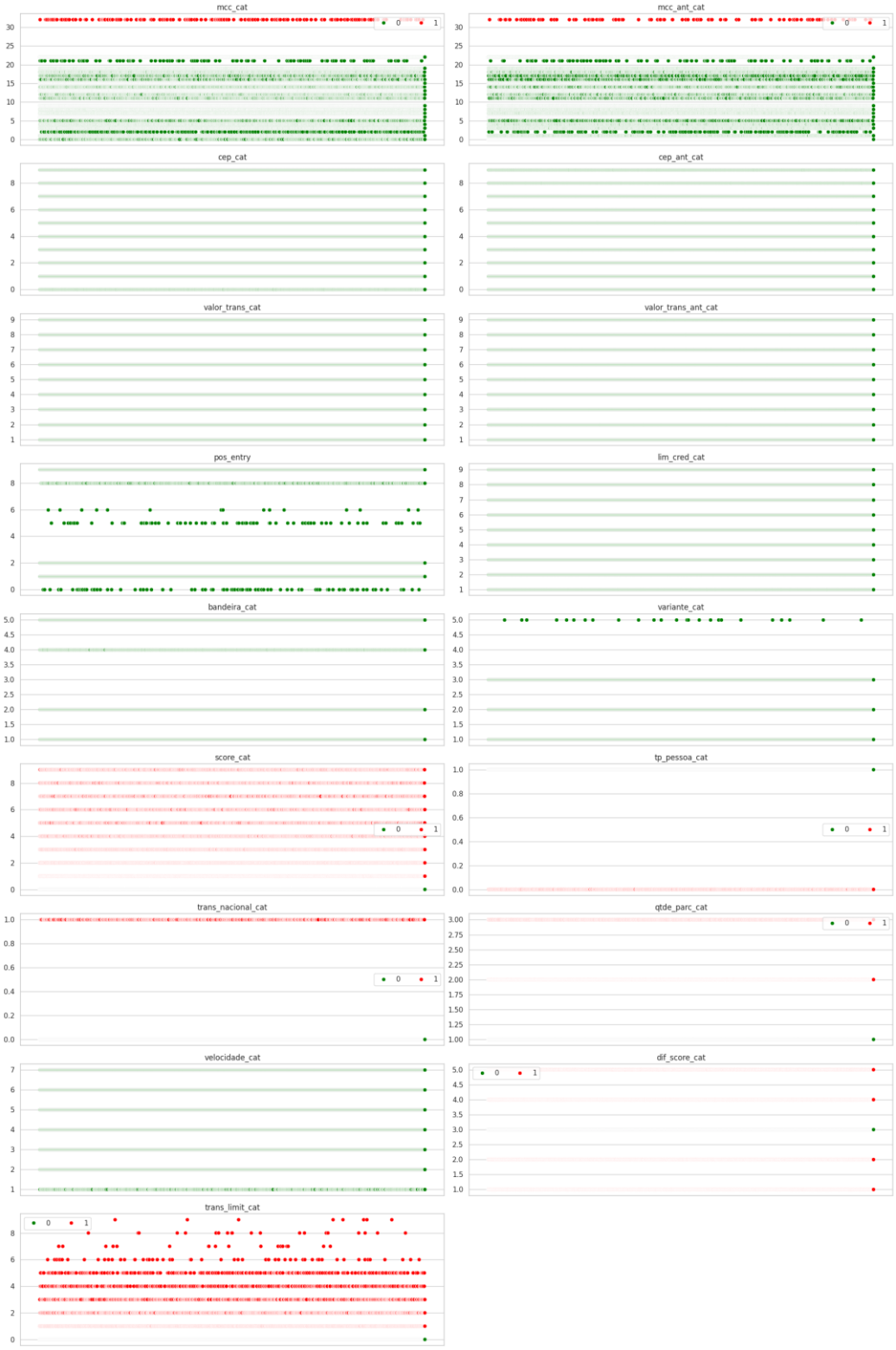

150
100
50
0
−50
−100
−150
−150
−100
−50
0
50
100
150

۲-۳-شناسایی ناهنجاری

۲-۳-۱- Boxplots



IQR -۲-۳-۲


mcc_cat
mcc_ant_cat
cep_cat
cep_ant_cat
valor_trans_cat
valor_trans_ant_cat
pos_entry
lim_cred_cat
bandeira_cat
variante_cat
score_cat
tp_pessoa_cat
trans_nacional_cat
qtde_parc_cat
velocidade_cat
dif_score_cat
trans_limit_cat
0
1

این نمودارها یک نگاه کلی از توزیع متغیرهای مختلف در دیتافریم داده شده و در عین حال نقاطی که به عنوان نوع خارج از محدوده (Outlier) شناخته شده‌اند را نیز مشخص می‌کنند. موارد زیـر نشـان‌دهنده جوانب مختلف این کد هستند:

۱. تعداد ستون‌ها و ردیف‌ها:

- تعداد ستون‌ها: ۲ ستون برای هر ردیف.

- تعداد ردیف‌ها: ۹ ردیف برای نمایش متغیرها.

۲. ساختار نمودارها:

- برای هر متغیر، یک Scatter Plot رسم شده است که نمایانگر توزیع مقادیر آن متغیر می‌باشد.

- نقاطی که به عنوان Outlier تشخیص داده شده‌اند، با رنگ قرمز مشخص شده‌اند.

- برای تشخیص Outlier از روش IQR (محدوده بین‌کارگیری) استفاده شده است.

۳. تشخیص نوع Outlier:

- از IQR برای تعیین محدوده معقول برای مقادیر هر متغیر استفاده شده است.

- نقاطی که خارج از این محدوده‌ها قرار دارند، به عنوان Outlier شناخته شده‌اند.

۴. آرایش کلی:

- هر نمودار عنوانی دارد که نام متغیر مرتبط با آن را نشان می‌دهد.

- محور افقی (X) شماره نمونه‌ها و محور عمودی (Y) مقادیر متغیر را نمایش می‌دهد.

- اگر Outlier وجود داشته باشد، یک لژاند (Legend) با رنگ‌هـای متفـاوت بـرای نمـایش Outlier اضافه می‌شود.

۵. آپدیت لی‌اوت:

- توابع `tight_layout` برای بهبود لی‌اوت به کار گرفته شده‌اند.


mcc_cat
mcc_ant_cat
cep_cat
cep_ant_cat
valor_trans_cat
valor_trans_ant_cat
pos_entry
lim_cred_cat
bandeira_cat
variante_cat
score_cat
tp_pessoa_cat
trans_nacional_cat
qtde_parc_cat
velocidade_cat
dif_score_cat
trans_limit_cat

این نمودارها نسبت به نمودارهای قبلی یک تفاوت اساسی دارند:

۱. تفاوت در نوع Outlier:

- در نمودارهای جدید، از روش IQR برای تشخیص نقاط خارج از محدوده (Outlier) استفاده شده است.

- در نمودارهای قبلی، از نقاط خارج از محدوده تعیین شده توسط Boxplot (با استفاده از ۱.۵ برابر IQR) به عنوان Outlier استفاده شده است.

۲. نحوه نمایش Outlier:

- در نمودارهای جدید، Outlierها با رنگ‌های متفاوت برای هر Scatter Plot نمایش داده شده‌اند.

- در نمودارهای قبلی، هر نمودار Scatter Plot دو رنگ دارد که نقاط Outlier با رنگ قرمز و نقاط معمولی با رنگ سبز نمایش داده شده‌اند

۳. ساختار لی‌اوت:

- در هر دو نمودار، از یک لی‌اوت با ستون‌ها و ردیف‌های مشابه برای نمایش اطلاعات استفاده شده است.

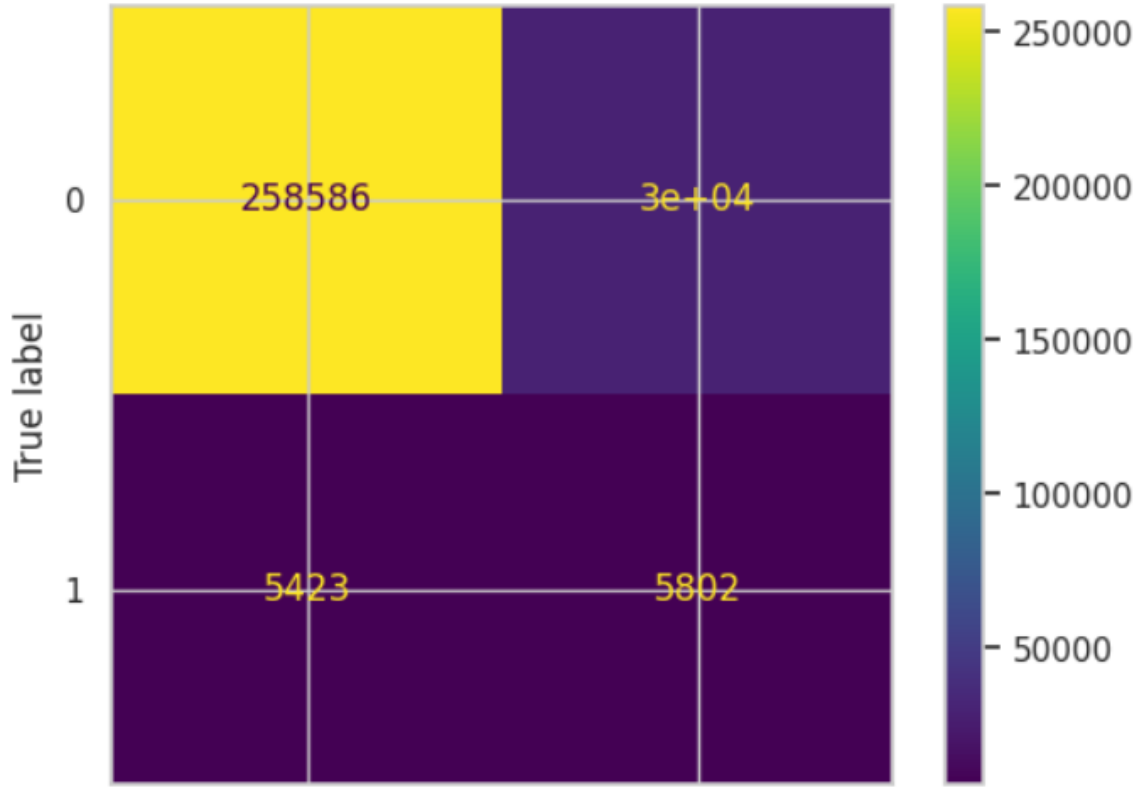
۴. محتوای اطلاعات:

- هر دو نمودار به شما اطلاعاتی ارائه می‌دهند در مورد توزیع مقادیر متغیرها و نقاطی که به عنوان Outlier شناخته شده‌اند.

به طور کلی، تفاوت‌ها به نحوه تشخیص و نمایش Outlierها در Scatter Plotها برمی‌گردد. در نمودارهای اولیه، از نمودار Boxplot برای تشخیص Outlierها استفاده شده بود، در حالی که در نمودارهای دومی، Outlierها به صورت مستقیم با استفاده از Scatter Plot نمایش داده شده‌اند.

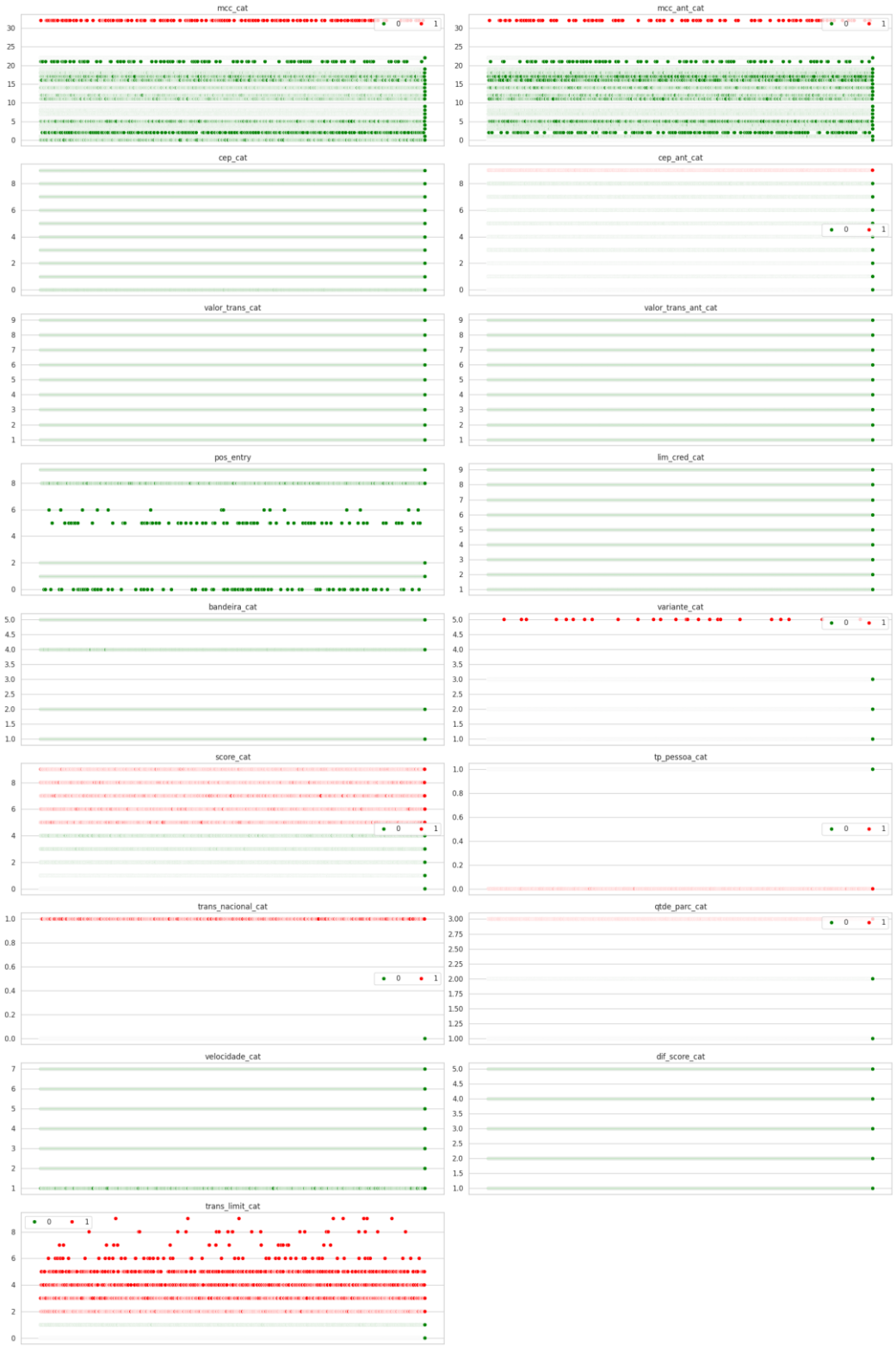
در نهایت به بررسی عملکرد این شیوه پرداختیم:

```
Precision: 0.5706522761485011
Recall: 0.7063904521225337
F1 Score: 0.591161451376123
Accuracy: 0.8817106763868231
```



Z-score -۳-۳-۲

نمودار ۱)

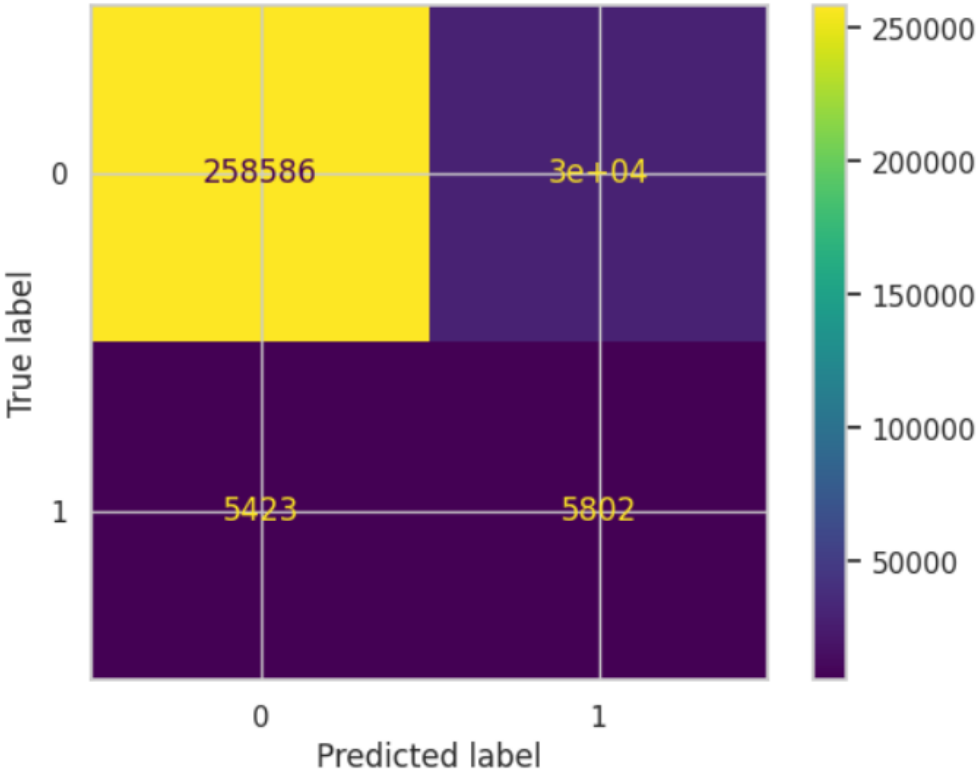


نمودار۲)


mcc_cat
mcc_ant_cat
cep_cat
cep_ant_cat
valor_trans_cat
valor_trans_ant_cat
pos_entry
lim_cred_cat
bandeira_cat
variante_cat
score_cat
tp_pessoa_cat
trans_nacional_cat
qtde_parc_cat
velocidade_cat
dif_score_cat
trans_limit_cat
0
1

بررسی عملکرد:

```
Precision: 0.5706522761485011
Recall: 0.7063904521225337
F1 Score: 0.591161451376123
Accuracy: 0.8817106763868231
```



۲-٤-شناسایی Anomaly

۲-۴-۱- One-class SVM

در این کد، از مدل One-Class SVM برای تشخیص ناهنجاری‌ها در داده‌های آموزش و آزمون استفاده شده است. ابتدا یک نمونه از این مدل با پارامتر nu=0.01 ایجاد شده و سپس روی داده‌های آموزش (X_train) آموزش دیده می‌شود. سپس، پیش‌بینی‌های مربوط به ناهنجاری‌ها بر روی داده‌های آموزش و آزمون انجام می‌شود. نتایج پیش‌بینی با استفاده از این مدل در دو مجموعه داده با دسته‌بندی ۱- به عنوان ناهنجار و ۱ به عنوان داده معمولی ذخیره می‌شوند. در نهایت، تعداد ناهنجاری‌های شناسایی شده در هر یک از مجموعه‌های آموزش و آزمون به چاپ درآورده می‌شود.

OneClassSVM anomalies in the training set: 3026

OneClassSVM anomalies in the test set: 745

۲-۴-۲-Local Outlier Factor (LOF) algorithm

در این کد، از مدل Local Outlier Factor (LOF) برای تشخیص ناهنجاری‌ها در داده‌های آموزش و آزمون استفاده شده است. ابتدا یک نمونه از مدل LOF با پارامتر contamination=0.01 و novelty=Trueایجاد شده و سپس روی داده‌های آموزش (X_train) آموزش دیده می‌شود. سپس، پیش‌بینی‌های مربوط به ناهنجاری‌ها بر روی داده‌های آموزش و آزمون انجام می‌شود. نتایج پیش‌بینی با استفاده از این مدل در هر یک از مجموعه‌های داده با دسته‌بندی ۱- به عنوان ناهنجار و ۱ به عنوان داده معمولی ذخیره می‌شوند. در نهایت، تعداد ناهنجاری‌های شناسایی شده در هر یک از مجموعه‌های آموزش و آزمون به چاپ درآورده می‌شود.

LOF detected 2022 anomalies in the training set.

LOF detected 678 anomalies in the test set.

۲-۵-شیوه های متوازن سازی

این کد از ماژول‌های `imblearn` و `sklearn` برای توازن کلاس‌های داده‌ها و سپس ارزیابی عملکرد یک مدل دسته‌بندی (در اینجا از `RandomForestClassifier` استفاده شده) استفاده می‌کند. تابع `balance_and_evaluate` با گرفتن داده‌های آموزش و آزمون، برچسب‌های کلاس، و یک `sampler`، داده‌های آموزش را با استفاده از `fit_resample` از `imblearn` توازن می‌دهد و سپس مدل را با داده‌های توازن‌یافته آموزش می‌دهد. نهایتاً پیش‌بینی‌های مدل بر روی داده‌های آزمون انجام شده و دقت و گزارش دقیقی از عملکرد مدل (precision، recall، f1-score و ...) به دست می‌آید. این عملیات برای داده‌های اصلی و همچنین برای داده‌های توازن‌یافته با استفاده از سه نوع توازن‌دهنده (oversampler، smote، undersampler) اجرا می‌شود و نتایج به چاپ درآمده و مقایسه می‌شوند.

```
Original Data:
Accuracy: 0.9751945136935866
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.99    108527
           1       0.60      1.00      0.75      4190

    accuracy                           0.98    112717
   macro avg       0.80      0.99      0.87    112717
weighted avg       0.99      0.98      0.98    112717


RandomOverSampler Sampling:
Accuracy: 0.9997338467134504
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    108527
           1       0.99      1.00      1.00      4190

    accuracy                           1.00    112717
   macro avg       1.00      1.00      1.00    112717
weighted avg       1.00      1.00      1.00    112717
```

```
SMOTE Sampling:
Accuracy: 0.9997338467134504
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    108527
           1       0.99      1.00      1.00      4190

    accuracy                           1.00    112717
   macro avg       1.00      1.00      1.00    112717
weighted avg       1.00      1.00      1.00    112717


RandomUnderSampler Sampling:
Accuracy: 0.9751945136935866
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.99    108527
           1       0.60      1.00      0.75      4190

    accuracy                           0.98    112717
   macro avg       0.80      0.99      0.87    112717
weighted avg       0.99      0.98      0.98    112717
```

در بخش بعدی کد یک تابع با نام `evaluate_method` ایجاد می‌کند که با گرفتن داده‌های آموزش و آزمون، برچسب‌های کلاس، نام روش (مثل 'Original' یا 'Random Oversampling') و یک `sampler` (که در صورت موجود بودن، داده‌ها را توازن می‌دهد)، یک مدل `RandomForestClassifier` را با داده‌های مورد نظر آموزش داده و سپس پیش‌بینی‌ها را بر روی داده‌های آزمون انجام می‌دهد. سپس، معیارهای ارزیابی از جمله دقت (`accuracy`)، دقت (`precision`)، حساسیت (`recall`)، و اف۱-اسکور (`f1`) برای مدل محاسبه شده و نتایج به صورت یک دیکشنری خروجی داده می‌شود.در ادامه، این تابع بر روی داده‌های اصلی و داده‌های توازن‌یافته با سه نوع توازن‌دهنده (oversampler، smote، undersampler) اجرا می‌شود و نتایج به یک جدول (`table`) افزوده می‌شود. این جدول شامل اطلاعاتی نظیر نام روش، دقت، دقت، حساسیت و اف۱-اسکور برای هر متد است و به صورت خروجی نمایش داده می‌شود.

	Method	Accuracy	Precision	Recall	F1
0	Original	0.999787	0.999520	0.994749	0.997129
1	Random Oversampling	0.999734	0.992891	1.000000	0.996433
2	SMOTE	0.999734	0.992891	1.000000	0.996433
3	Random Undersampling	0.975195	0.599771	1.000000	0.749821

۲-۶-بررسی ویژگی سری زمانی

کد ارائه شده یک مثال از استفاده از یک شبکه LSTM برای تشخیص الگوهای زمانی در داده‌های سری زمانی است. در ابتدا، از یک تابع به نام `WindowGenerator` برای ایجاد پنجره‌های زمانی از داده‌های ورودی استفاده شده است. سپس، داده‌های ورودی و خروجی این پنجره‌ها به عنوان ورودی برای شبکه LSTM قرار می‌گیرند.

در اینجا، مراحل اصلی کد:

۱. آماده‌سازی داده:

- ابتدا، داده‌های جدول `entire_data` و `entire_test` با استفاده از `pd.concat` به یکدیگر ادغام می‌شوند.

- برچسب‌های دودویی 'N' و 'S' به عنوان نرخ تقلب (`flag_fraude_cat`) با استفاده از `map` تبدیل به اعداد ۰ و ۱ می‌شوند.

- سپس، داده‌ها به فیچرها (`X_train` و `X_test`) و برچسب‌ها (`y_train` و `y_test`) جدا می‌شوند.

۲. استانداردسازی فیچرها:

- از یک `StandardScaler` برای استانداردسازی داده‌های آموزش استفاده می‌شود.

۳. ساخت پنجره‌های زمانی:

- با استفاده از تابع `WindowGenerator`، داده‌های آموزش (`X_train_time_seri` و `y_train_time_seri`) و داده‌های آزمون (`X_test_time_seri` و `y_test_time_seri`) به صورت پنجره‌های زمانی با طول `window_size` ساخته می‌شوند.

۴. تعریف مدل LSTM:

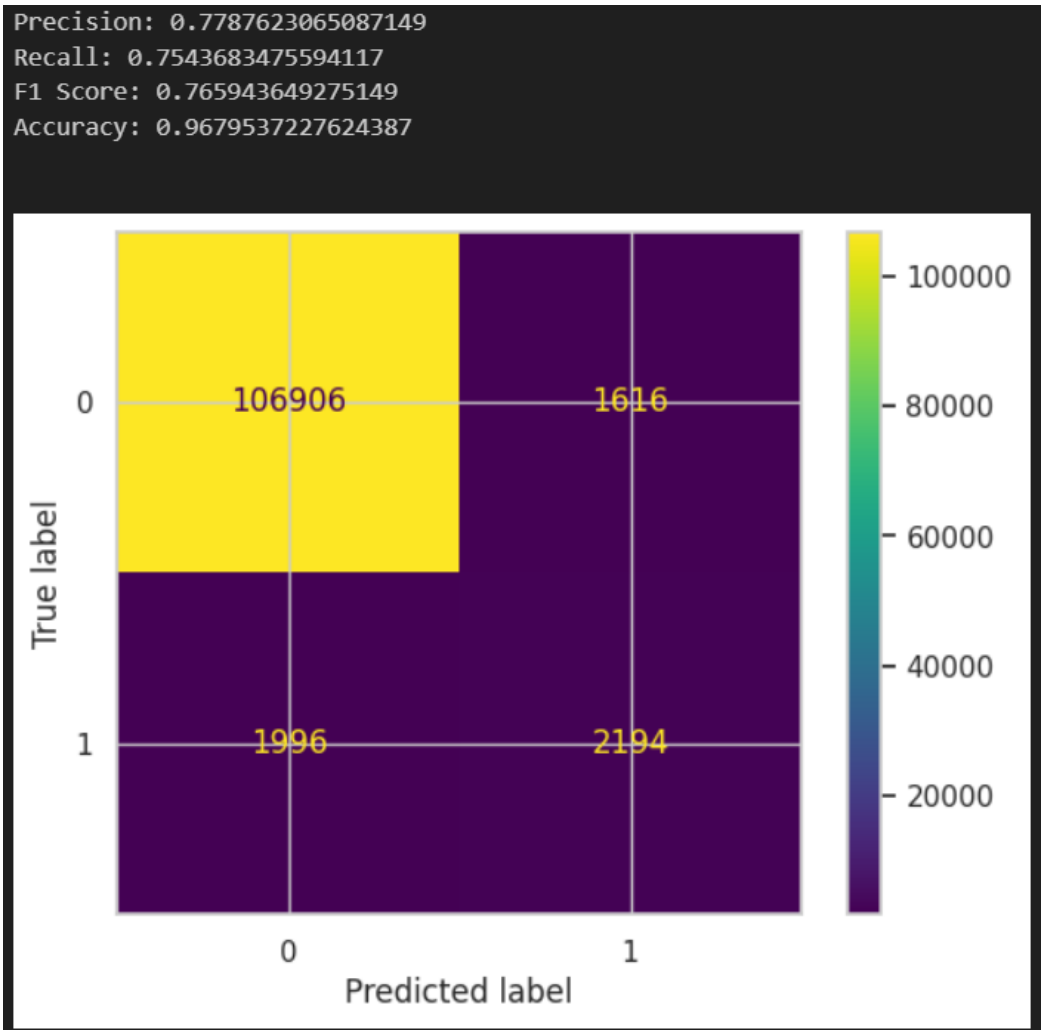
- یک مدل LSTM با استفاده از کتابخانه TensorFlow/Keras تعریف شده است. این مدل شامل یک لایه Bidirectional LSTM با تعداد واحدها ویژگی‌های مخفی ۱۰ و یک لایه Fully Connected با تعداد واحدها ۱ است.

۵. آموزش مدل:

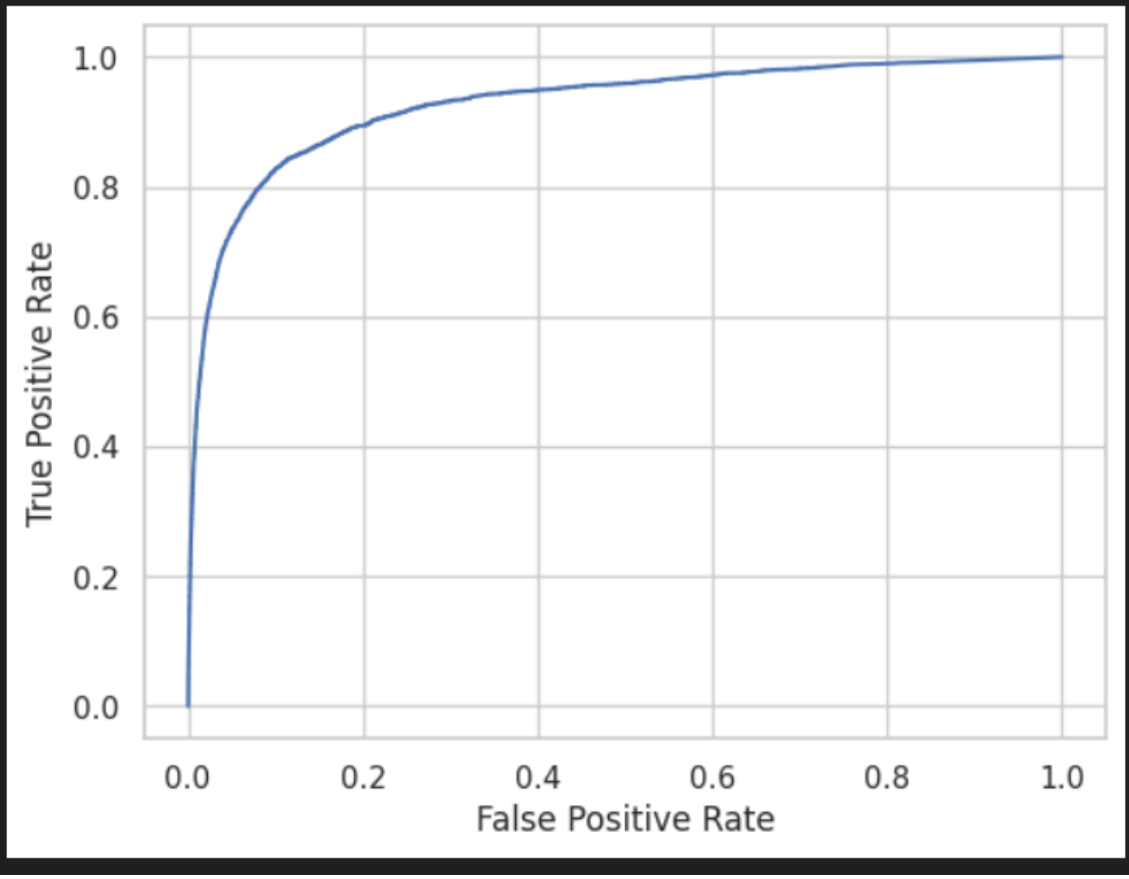
- مدل بر روی داده‌های آموزش (`X_train_time_seri` و `y_train_time_seri`) با استفاده از تابع `fit` و با انتخاب پارامترهای مختلف مانند ``batch_size، epochs`` و ... آموزش داده می‌شود.

```
Total params: 49321 (192.66 KB)
Trainable params: 49321 (192.66 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/10
3686/3686 [==============================] - 35s 8ms/step - loss: 0.0951 - accuracy: 0.9665 - val_loss: 0.0801 - val_accuracy: 0.9701
Epoch 2/10
3686/3686 [==============================] - 27s 7ms/step - loss: 0.0717 - accuracy: 0.9744 - val_loss: 0.0713 - val_accuracy: 0.9728
Epoch 3/10
3686/3686 [==============================] - 27s 7ms/step - loss: 0.0633 - accuracy: 0.9775 - val_loss: 0.0640 - val_accuracy: 0.9780
Epoch 4/10
3686/3686 [==============================] - 27s 7ms/step - loss: 0.0581 - accuracy: 0.9793 - val_loss: 0.0586 - val_accuracy: 0.9797
Epoch 5/10
3686/3686 [==============================] - 26s 7ms/step - loss: 0.0545 - accuracy: 0.9805 - val_loss: 0.0577 - val_accuracy: 0.9802
Epoch 6/10
3686/3686 [==============================] - 26s 7ms/step - loss: 0.0516 - accuracy: 0.9816 - val_loss: 0.0612 - val_accuracy: 0.9779
Epoch 7/10
3686/3686 [==============================] - 26s 7ms/step - loss: 0.0500 - accuracy: 0.9823 - val_loss: 0.0567 - val_accuracy: 0.9793
Epoch 8/10
3686/3686 [==============================] - 26s 7ms/step - loss: 0.0495 - accuracy: 0.9823 - val_loss: 0.0562 - val_accuracy: 0.9807
Epoch 9/10
3686/3686 [==============================] - 27s 7ms/step - loss: 0.0483 - accuracy: 0.9831 - val_loss: 0.0568 - val_accuracy: 0.9802
Epoch 10/10
3686/3686 [==============================] - 26s 7ms/step - loss: 0.0469 - accuracy: 0.9836 - val_loss: 0.0595 - val_accuracy: 0.9803
```

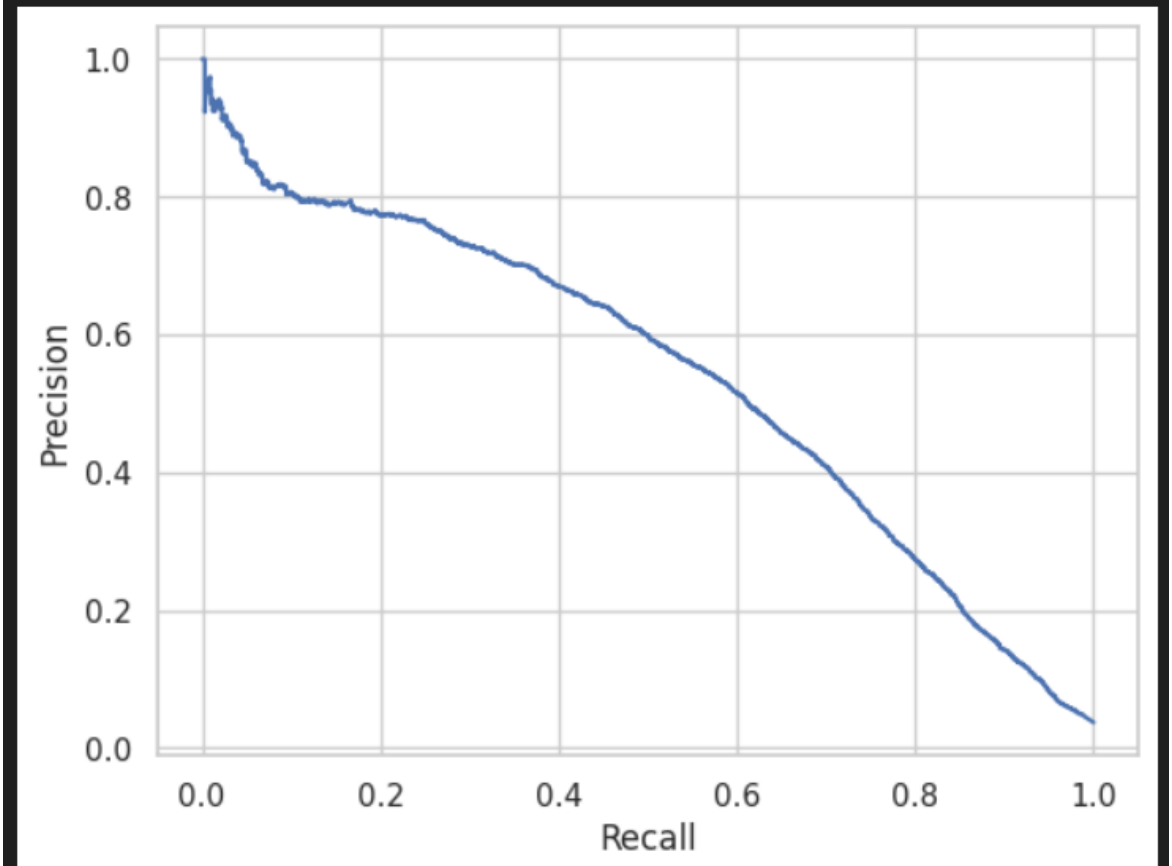
```
Precision: 0.7787623065087149
Recall: 0.7543683475594117
F1 Score: 0.765943649275149
Accuracy: 0.9679537227624387
```



```
AUC score: 0.9274881078411825
```




```
AUPRC score: 0.5407336488736196
```



فصل سوم

جمع‌بندی

جمع‌بندی و نتیجه‌گیری

در این پروژه، سعی کردیم ابزارهایی را برای شناسایی الگوهای ناهنجار و تقلب در این داده‌ها ارائه دهیم. مراحل اصلی پروژه عبارت بودند از:

۱. آشنایی با داده:

- ابتدا، با بارگیری و ادغام داده‌های مالی از چند فایل، یک داده ترکیبی بزرگ تشکیل دادیم.

۲. تحلیل و آمارگیری اولیه:

- از نمودارها مانند نمودار boxplot و histogram برای بررسی توزیع داده‌ها و تشخیص الگوهای ناهنجاری استفاده کردیم.

۳. استفاده از الگوریتم‌های مختلف برای شناسایی ناهنجاری:

- از الگوریتم‌های مختلفی مانند One-Class SVM و Local Outlier Factor برای شناسایی ناهنجاری در داده‌ها استفاده کردیم.

۴. مدل‌های ماشین لرنینگ:

- از مدل‌های ماشین لرنینگ مانند Random Forest برای تحلیل و پیش‌بینی تقلب در داده‌ها استفاده کردیم.

۵. استفاده از شبکه‌های عصبی برای تحلیل سری زمانی:

- با استفاده از یک شبکه LSTM برای تشخیص الگوهای زمانی در داده‌های سری زمانی تراکنش‌ها، موفق به ساخت و آموزش یک مدل بر روی داده‌ها شدیم.

۶. متوازن سازی کلاس‌ها:

- با استفاده از تکنیک‌های متوازن‌سازی مانند Random Oversampling و Random Undersampling، سعی کردیم تا تعادل بین کلاس‌های مختلف را در مدل‌های ماشین لرنینگ حفظ کنیم.

منابع و مراجع

[1] https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset

پیوست‌ها

- https://colab.research.google.com/drive/1KL9mp8M5mLmXfTyMl2XL16oDaCDH_T0k?usp=sharing

شایان ذکر است که در این تمرین از راهنمایی های خانم شرافتمندجو استفاده شده است

از طرفی در ذخیره سازی نوت بوک به مشکلات متعددی خوردم و فایل اصلی بصورت مجزا ارسال شده است.

Abstract

This project includes several sections in the field of data analysis and machine learning. Initially, random time data was obtained from several data sets. Then, the data was analyzed and corrected, and dimensionality reduction methods such as PCA and t-SNE were used to observe the data distribution in the low-dimensional space. In the next step, anomaly detection and data destruction were performed using various methods including One-Class SVM and Local Outlier Factor.

Then, new approaches were used to increase climate data. OverSampling techniques such as Random OverSampler and SMOTE were used to enhance quantitative climate data and UnderSampling techniques such as Random UnderSampler were used to balance the classes. Also, the random forest classification model was optimized and its performance was evaluated on the augmented and de-emphasized data.

Finally, for temporal models, an LSTM network was used to identify temporal patterns in the data. This LSTM network was trained with inputs and labels made from temporal functions of the data to extract important temporal patterns in the data.These analyzes and models in summary show how various techniques in data analysis, strengthening of sparse classes, and identification of temporal patterns can be used to improve performance and reliability in the field of climate-related complex data analysis.

Keywords:

Anomaly detection, data balancing, time series data, dimensionality reduction, data out of range

Abstract

Amirkabir University of Technology
(Tehran Polytechnic)

Department of Computer Science

Project 7

Identifying data outliers and anomalies, comparing data balancing methods and providing evaluation

By
Samin Mahdipour

Supervisor
Dr.Ghatee

Advisor
Dr.Yousofi Mehr

December 2023