# تعریف مساله و شناخت مجموعه داده:

در این پروژه از یک مجموعه داده­ی نامتوازن استاندارد به نام german credit dataset[[1]](#footnote-1) استفاده می­کنیم. این مجموعه داده بخشی از پروژه­یStatlog بوده است که یک پروژه ی اروپایی در سال 1990 برای ارزیابی و مقایسه­ی تعداد زیادی از الگوریتم های یادگیری ماشین بر روی مسائل مختلف طبقه­بندی بوده است.

این مجموعه داده جزئیات مالی و بانکی مشتریان را توصیف می­کند. هدف مساله پیش­بینی مشتری خوب و مشتری بد برای اعطای وام یا اعتبار به مشتری است. این مجموعه داده شامل 1000 رکورد و 20 متغیر ورودی است که 7 ستون عددی و 13 ستون کتگوریکال است.

 Status of existing checking account

 Duration in month

 Credit history

 Purpose

 Credit amount

 Savings account

 Present employment since

 Installment rate in percentage of disposable income

 Personal status and sex

 Other debtors

 Present residence since

 Property

 Age in years

 Other installment plans

 Housing

 Number of existing credits at this bank

 Job

 Number of dependents

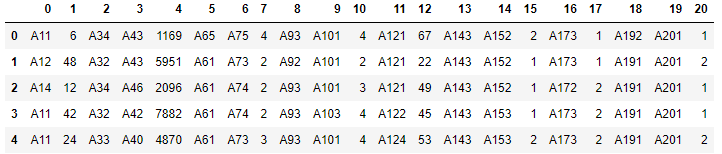
 Telephone

 Foreign worker

برخی از ستون های غیر عددی رابطه ی ترتیبی دارند مانند ستون saving account. دو کلاس خروجی وجود دارد : 1 برای مشتری خوب و 2 برای مشتری بد. مشتری خوب کلاس اکثریت یا منفی است که 70 درصد از داده ها را تشکیل داده و مشتری بد کلاس اقلیت یا مثبت است که سی درصد از داده ها را به خود اختصاص داده است.

در توضیحات مجموعه داده، یک ماتریس هزینه [[2]](#footnote-2) وجود دارد که به هر خطای طبقه بندی کلاس مثبت، پنالتی متفاوتی احتصاص داده است. هزینه ی 5 برای یک حطای false negative (پیش بینی یک مشتری بد به عنوان مشتری خوب) و هزینه ی 1 برای خطای false positive (پیش بینی یک مشتری خوب به عنوان بد) اعمال می شود. این نشان می­دهد که کلاس مثبت اهمیت زیادی در این مساله دارد و برای بانک یا موسسه مالی هزینه برتر است که به یک مشتری بد وام/اعتبار بدهد تا اینکه به یک مشتری خوب وام ندهد. این مورد هنگام انتخاب معیار ارزیابی بایستی لحاظ شود.

با خواندن مجموعه داده، سطرها به این صورت دیده می شوند:



ستون های categorical به صورت Axxx کدگذاری شده اند که xxxx اعداد صحیح نامنفی هستند. به عنوان پیش پردازش نیاز به one hot encoding برای این ستونها داریم. همچنین برای ستونهای عددی مقیاس های مختلفی داریم؛ به عنوان مثال در ستون دوم 6 و 48 و 12 داریم و در ستون پنجم 1169 و 5951 داریم. هم مقیاس کردن این ستونهای عددی نیز به عنوان پیش پردازش لازم است. با مصورسازی ستونهای مختلف نیز می بینیم که هرستون توزیع متفاوتی دارد که اهمیت هم مقیاس کردن ستونها را قبل از مدلسازی نشان میدهد.

# انتخاب معیار و روش ارزیابی:

برای ارزیابی مدلها از استراتژی repeated stratified k-fold cross-validation استفاده می کنیم. خود روش kfold cross validation برای مجموعه داده ی نامتوازن مناسب نیست اما در حالت stratified تضمین می شود که هر fold توزیع یکسانی از کلاسها با مجموعه داده ی اصلی داشته باشد. Repeated نیز به این معناست که پروسه ی ارزیابی جند بار تکرار می شود تا از نتایج شانسی جلوگیری کند. ما از 3 تکرار و k=10 استفاده کردیم که به این معناست که هر مدل 30 بار fit و ارزیابی می شودو میانگین و انحراف معیار این اجراها گزارش می شود.

هدف ما پیش بینی خوب/بد بودن یک مشتری است بنابراین نیاز به معیاری داریم که برای ارزیابی برچسب پیش بینی شده مناسب باشد. طبق توضیحات دیتاست تمرکز بر روی کلاس مثبت(مشتریان بد) است. معیارهای precision و recall به نظر مناسب میرسند چرا که ماکزیمم کردن precision به معنای مینیمم کردن false positive و ماکزیمم کردن recall به معنای مینیمم کردن false negative است. معیار f-score میانگین هارمونیک بین این دو متریک را محاسبه میکند اما در این مساله false negative آسیب بیشتری از false positive دارد وطبق ماتریس هزینه هزینه ی 5 برابر دارد. برای مدل کردن این موضوع در معیار ارزیابی از معیار fbeta با beta=2 استفاده می کنیم :

برای پیاده سازی این معیار از fbeta\_score() در scikit-learn استفاده می کنیم.

# مدلسازی بدون استفاده از Sampling:

الگوریتم های انتخاب شده برای این مساله عبارتند از :

* Logistic Regression (LR)
* Linear Discriminant Analysis (LDA)
* Naive Bayes (NB)
* Gaussian Process Classier (GPC)
* Support Vector Machine (SVM )
* Decision Tree Classifier(DT)
* Multilayer Perceptron (MLP)
* XGBoost(xgb)

تابع classification\_models() را برای تعریف لیستی از مدلها برای ارزیابی استفاده میکنیم. پیش پردازش ستونهای عددی برای هم مقیاس شدن با minmaxscaler و تبدیل ستونهای categorical به عددی را با onehotencoder انجام می دهیم.

بدون استفاده از تکنیک های sampling بهترین عملکرد مربوط به مدل NB با میزان f2 برابر با 0.639است .

# مدلسازی با استفاده از تکنیک های undersampling:

مبنای روش های undersampling بر حذف یا انتخاب زیرمجموعه ای از کلاس اکثریت است. در اینجا از روشهای زیر برای انجام undersampling استفاده میکنیم :

Tomek Links (TL)

 Edited Nearest Neighbors (ENN)

 Repeated Edited Nearest Neighbors (RENN)

 One Sided Selection (OSS)

 Neighborhood Cleaning Rule (NCR)

روش های tomek links و ENN نمونه هایی از کلاس اکثریت را برای حذف انتخاب میکنند درحال که روش های OSS و NCR هردو نمونه هایی را برای نگه داشتن و نمونه هایی را برای حذف انتخاب میکنند. در ابتدا الگوریتم svm را با روش های مختلف فوق تست و سپس این روشها را برروی همه الگوریتم ها تست میکنیم.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TL | ENN | RENN | OSS | NCR | No sampling |
| LR | 0.525 | 0.694 | 0.716 | 0.529 | 0.685 | 0.498 |
| LDA | 0.548 | 0.696 | 0.717 | 0.553 | 0.688 | 0.519 |
| NB | 0.653 | 0.660 | 0.571 | 0.651 | 0.692 | 0.639 |
| GPC | 0.312 | 0.641 | 0.690 | 0.314 | 0.668 | 0.219 |
| SVM | 0.506 | 0.675 | 0.712 | 0.507 | 0.672 | 0.436 |
| DT | 0.477 | 0.622 | 0.673 | 0.486 | 0.603 | 0.438 |
| MLP | 0.549 | 0.679 | 0.714 | 0.547 | 0.666 | 0.544 |
| XGBoost | 0.504 | 0.670 | 0.706 | 0.517 | 0.674 | 0.467 |

# مدلسازی با استفاده از تکنیک های oversampling:

تکنیک های oversampling نمونه های کلاس اقلیت را تکرار میکنند یا نمونه های جدید از کلاس اقلیت می سازند. در اینجا برای ارزیابی عملکرد تکنیک های oversampling از دو روش زیر استفاده کردیم :

* SMOTE
* ADASYN

هردو الگوریتم بر مبنای ایجاد نمونه های جدید از کلاس اقلیت کار میکنند. با مشاهده ی نتایج به دست آمده مشاهده می شود که تکنیکهای undersampling عملکرد بهتری دارند.

|  |  |  |  |
| --- | --- | --- | --- |
|  | SMOTE | ADASYN | No sampling |
| LR | 0.656 | 0.650 | 0.498 |
| LDA | 0.659 | 0.657 | 0.519 |
| NB | 0.671 | 0.667 | 0.639 |
| GPC | 0.654 | 0.657 | 0.219 |
| SVM | 0.583 | 0.586 | 0.436 |
| DT | 0.489 | 0.485 | 0.438 |
| MLP | 0.562 | 0.559 | 0.544 |
| XGBoost | 0.553 | 0.552 | 0.467 |

1. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data) [↑](#footnote-ref-1)
2. Cost matrix3 [↑](#footnote-ref-2)