第２回コンペ　Home Credit Default Risk で意識したこと

１　先人の知恵を借りる（機械学習やデータ分析、Coding に慣れていないので）

私は開始から現在まで、Chat-GPT や Github Copilot を用いながら取り組んできました。このコンペで最初に行ったことは、下記の丸パクリです。

https://www.kaggle.com/code/osciiart/homecreditrisk-extensive-eda-baseline-model-jp/notebook

これをそのまま一度走らせた後、徐々に思いついたことを付け足しながら、コンペに取り組んでいきました。もちろん思いついたことをそのまま書いても動かないことや、そもそも書き方が分からないことも多かったので、GPT や Copilot を活用しました。

使用したモデルは Lightgbm のみです。最初は上のリンクのベースモデルをそのまま使っていましたが、途中で CV を使いたくなったので、元のモデルのコードを GPT に投げて、CV 化してもらいました。

２　前処理

前処理は、train と test を合わせたデータフレームで行いました。前処理と特徴量エンジニアリングを終えてから、モデルに入れる直前にデータを分割しました。

欠損値や異常値の処理は、下記の特徴量エンジニアリングとリンクします。特徴量エンジニアリングの際に使うカラムの欠損値や異常値のみに焦点を当てて処理したので、何度も行ったり来たりしながら、前処理を行いました。

EXT_SOURCE や AMT_GOODS_PRICE などの重要な特徴量の欠損は、Lightgbm で予測して埋めました。その他は中央値で埋めたり、似たようなカラムの数字で埋めたりなどしました。手が回らなかったカラムは-999 で埋めました。ゼロで埋めると精度が下がることがあったので、この形になりました。

その他の前処理としては、DAYS__などの日数が入っているカラムは年単位に直したり、カテゴリカル変数は One-Hot-Encoding したりと、基本的なことをしました。

また、一応ですが、データ全体のスケーリングも行いました。しかし、スケーリングしていないデータも学習に用いました。スケーリング前のデータでの CV が良くなることもあったので、アンサンブルと考えればいいんじゃないかな？と思い。。。間違った考えかもしれませんが。

３　特徴量エンジニアリング

主な特徴量エンジニアリングは、AMT_INCOME_TOTAL や AMT_CREDIT などのお金に関するカラムを、お互いで割ったり、DAYS_EMPLOYED や DAYS_IN_PUBLISH のような「信用」に関連しそうなカラムで割ったりしました。また EXT_SOURCE の３つも重要だったので、掛け算や足し算で新しく特徴量を作りました。

新しい特徴量を加えた後は、すぐに学習と予測を行い、精度が落ちる特徴量はコメントアウトしながら、特徴量の取捨選択を行っていきました。Lightgbm を使っていたので、Importance も表示して重要そうな特徴量の四則演算していきました。でも、AMT_ANNUITY を DAYS_EMPLOYED や DAYS_IN_PUBLISH などで割ると精度が下がったのは、未だに疑問です。

後は、教師なし学習と組わせてみようと思いたち、クラスタリングを行い、特徴量として使いました。これも若干ですが精度が上がりました。

４　モデルの構築

モデルは先程も書いたように Lightgbm のみを用いましたが、アンサンブルのため Lightgbm で３つ用意しました。１つ目は、標準化していないデータでの学習モデル、２つ目は標準化したデータで学習したモデル、３つ目は"boosting_type"に"dart"を用いたモデルです。最初の２つのモデルの"boosting_type"は"goss"を用いました。本来なら、異なるアルゴリズムを用いたほうが良かったのかもしれませんが、データ量が多いのと、先にも書いたように前処理と特徴量エンジニアリングを行ったり来たりしていたので、早く処理が終わる Lightgbm が使いやすかったので、この形になりました。このやり方でも多少の汎化性能は向上しました。

optuna のようなハイパーパラメータを調整するものは用いずに、手動で調整しました。できるだけ汎化性能があがるような調整を心がけました。もちろん使ったほうが良いモデルを作ることができるとは思うのですが、初心者の私は、まず前処理や特徴量エンジニアリングに力を入れた方が良いと思い、モデルの作成には力をいれませんでした。

５　まとめ

上のように、前処理や特徴量エンジニアリングに力を入れた理由はもう一つあります。第１回の Titanic コンペでは、モデルの作成に力を入れたのですが、どうしても精度が上がらず、スコアが改善しませんでした。データ分析は、前処理にほとんどの時間を割いていると言われているので、今回はモデルよりも前処理に力を入れてみようと考え、このような取り組み方をしてみました。抽象的な書き方が多いですが、以上です。ありがとうございました。