Kaggle

American Express – Default Prediction

Daniel_WJ

Discussion

목표

Discussions 의 내용을 이해하는 것을 통해 데이터와 모델 개선에 도움을 얻는 정보를 얻는다.

Integer columns in the data - here you go!

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/328514

- 모든 float 유형 열에는 각 열에 [0, 0.01]의 무작위 균일 노이즈가 추가된다.

=> 정수 유형의 열을 찾을 수 있다.

=> https://www.kaggle.com/code/raddar/the-data-has-random-uniform-noise-added

- 188개의 실수/범주 자료형이 있다.

95개의 np.int8/np.int16 자료형.

93개의 np.float32 유형

Integer columns in the data - here you go!

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/328514

- 정리된 데이터 셋을 다음 데이터 셋에서 찾을 수 있다.

  => **https://www.kaggle.com/datasets/raddar/amex-data-integer-dtypes-parquet-format**

- 데이터 변환에 사용되는 노트북

  => **https://www.kaggle.com/code/raddar/amex-data-int-types-train**

  => **https://www.kaggle.com/code/raddar/amex-data-int-types-test**

Integer columns in the data - here you go!

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/328514

- S_13, S_11은 이미 정수로 변환되었다.

  => train['S_11']은 15,16,17 값이 존재한다.

- 해당 데이터를 이용하여 0.792 -> 0.794로 향상시켰다.

  => https://www.kaggle.com/code/cdeotte/xgboost-starter-0-793

  => R_26은 0.2이하의 유일한 값이 548160이 있다. 그러나 실제로 확인 결과 노이즈가 추가된 6개의 구분된 값이었다. 그래서 int8로 변환이 가능하였다.

Integer columns in the data - here you go!

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/328514

- B_19, S_8, S_13은 정수로 변환하였다.

  => train['S_11']은 15,16,17 값이 존재한다.

- B_4 도 임의의 랜덤 값이 추가된 것 같다.

- 작성자의 EDA 링크

  => https://www.kaggle.com/code/raddar/the-data-has-random-uniform-noise-added

Integer columns in the data - here you go!

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/328514

- 데이터 전처리에 np.nanmean을 사용하였다.

I don't use cudf - useless for this competition.
as for how to assign NA is actually the problem itself. I am using np.nanmean and other similar numpy aggregate funcs

- 코드에서 cudf의 NA를 위해 -1을 모든 것을 변환.

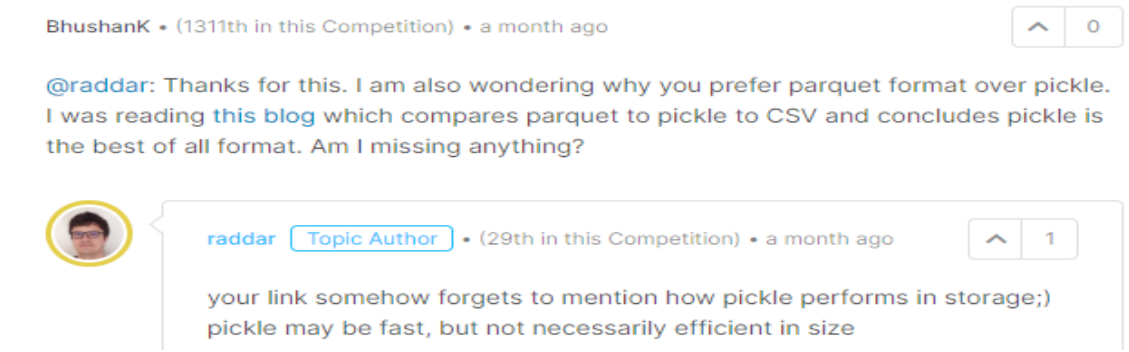
The following code will convert all -1 to NA in cudf

```
for c in df.columns:
    df.loc[df[c]==-1,c] = cudf.NA
```

Integer columns in the data - here you go!

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/328514

- Parquet vs pickle



=> Parquet가 사이즈 측면에서 효율적이다.

Integer columns in the data - here you go!

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/328514

- B_8, D_112, B_18는 정수형 변수

Carlos A. S. de Souza • (2612th in this Competition) • 2 months ago 0

Hello raddar! I think that another 4 variables could be squeezed into the **int** type as well: **D104,D_112,B_8 and B_18 ** (look at the histograms of Version 16 of this notebook.

B_18 seems like a counting variable (Versions 14 and 15 of the same notebook above), while D_104,D_112 and B_8 seems like binary variables.

For D_104 and D_112,the noise seems to be random, although not in an uniform or normal way (distribution). For B_8, the noise seems to be pretty uniform around 1.

raddar Topic Author • (29th in this Competition) • 2 months ago 2

Nice analysis. However, all your listed variables are not 100% certainly integer types. For example D_104 over 0.9 shows "normal" like distribution, which indicates that the values are definetly not integers. same goes for other variables - there are ranges where non-uniform distribution can be seen.

B_18 is most likely integer type - however for me it was impossible to reverse engineer it.

The data has uniform random noise injected

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327649

데이터는 균일한 랜덤 노이즈가 주입되었다.

https://www.kaggle.com/raddar/the-data-has-random-uniform-noise-added

The data has uniform random noise injected

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327651

데이터는 노이즈가 실수형에 [0, 0.01] 랜덤한 노이즈가 추가되어 있다.

데이터 노이즈를 제거한 데이터 셋

https://www.kaggle.com/datasets/raddar/amex-data-integer-dtypes-parquet-format

The data has uniform random noise injected

https://www.kaggle.com/competitions/amex-default-prediction/discussion/332574

8/1일 작성

D_39의 변수에 대한 분석

https://www.kaggle.com/code/raddar/deanonymized-days-overdue-feat-amex

Understanding NA in the dataset

https://www.kaggle.com/competitions/amex-default-prediction/discussion/331725

2022/06 작성

NA에 대해 분석한 내용에 대한 작성

https://www.kaggle.com/code/raddar/understanding-na-values-in-amex-competition/notebook

Summary: Basic pipeline for tabular competition for beginner!

https://www.kaggle.com/competitions/amex-default-prediction/discussion/331840

2022/06 작성

초보자가 정형 대회를 위한 여러가지 단계별 작업 내용 정리

01. 데이터 보기 + 데이터 전처리: EDA, 데이터 정리, 정보 손실 없이 데이터 크기 줄이기

02. 피처 엔지니어링

LB기능을 향상 시키기 위한 피처 엔지니어링

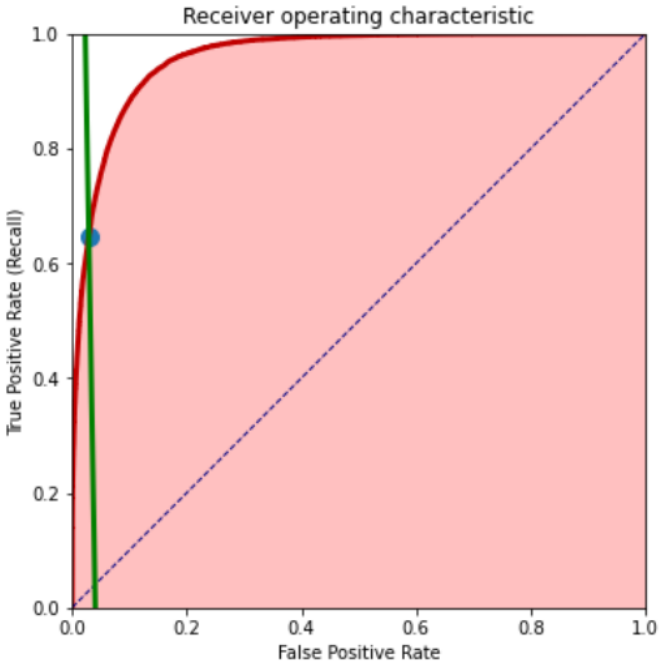
03. 피처 선택(Feature Selection)

04. 베이스 라인 모델링 및 하이퍼 파라미터 최적화

05. 앙상블 모델 – 가중치 및 평균 앙상블, 스태킹 모델

Graphical explanation of the competition metric

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327464



ROC 곡선

진한 빨간색 곡선

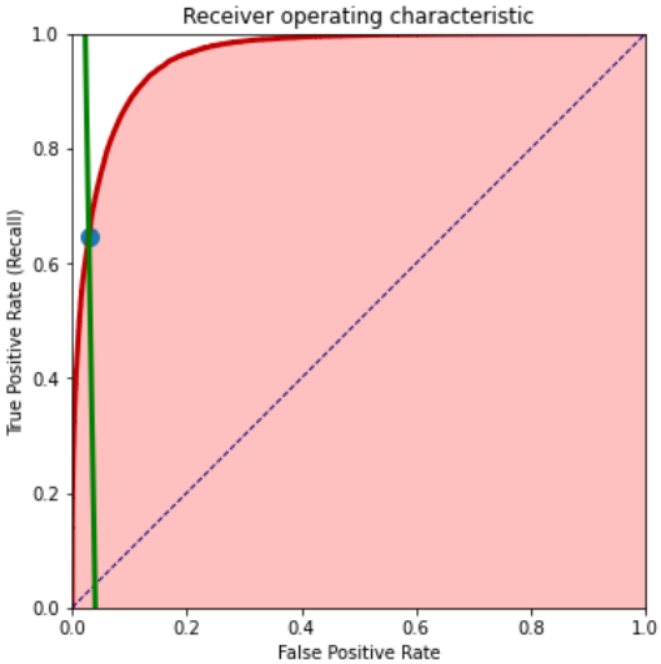
AUC

곡선 아래 영역 – 밝은 빨간색 부분

녹색선은 4%의 긍정적인 예측.

Graphical explanation of the competition metric

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327464



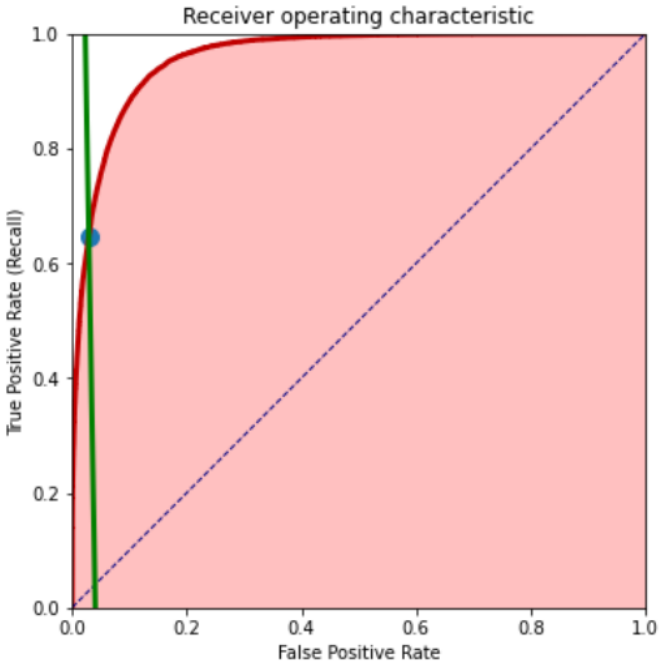
대회의 평가지표

(1) 정규화된 지니계수 – 단순히 확장된 AUC이다.

(2) 4%에 캡쳐된 기본 비율

Graphical explanation of the competition metric

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327464



대회의 평가지표

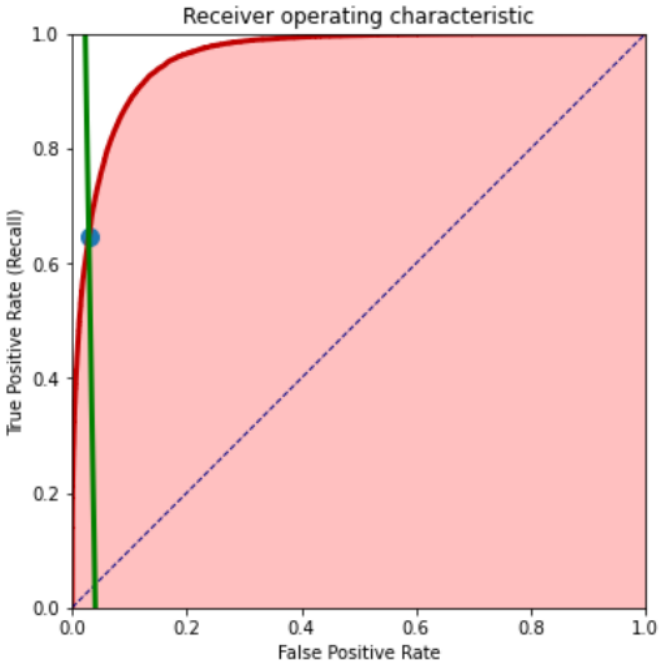
(1) 정규화된 지니계수 – 단순히 확장된 AUC이다.

값의 범위는 -1~1 사이. 2 * AUC - 1

(2) 4%에 캡쳐된 기본 비율

Graphical explanation of the competition metric

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327464



대회의 평가지표

(1) 정규화된 지니계수 – 단순히 확장된 AUC이다.

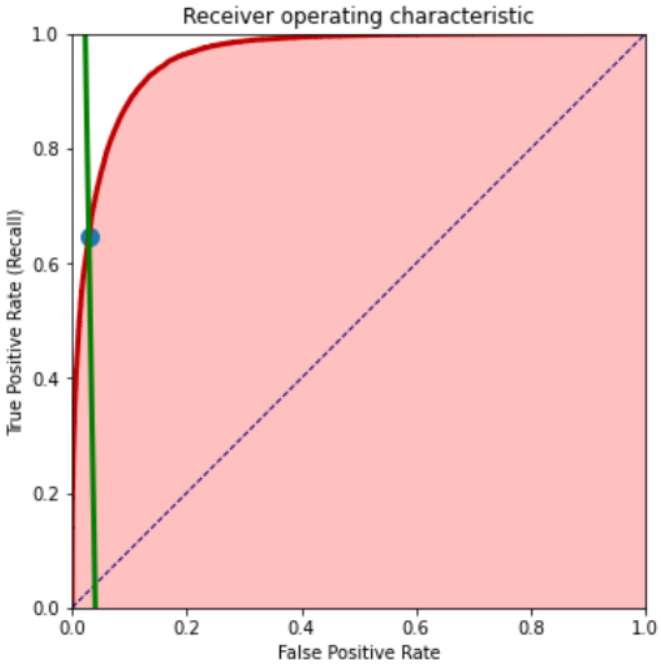
값의 범위는 -1~1 사이. 2 * AUC - 1

(2) 4%에 캡쳐된 기본 비율

총 샘플 수의 4%로 설정된 임계 값에 대한 참 양성 비율. 녹색선과 빨간색 ROC 곡선 사이의 교차점의 y좌표에 해당. 값의 범위는 0~1 사이.

Graphical explanation of the competition metric

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327464



대회의 평가지표

(1) 정규화된 지니계수(G)

(2) 4%에 캡쳐된 기본 비율(D)

$$M = 0.5 * (G + D)$$

두 평가지표의 평균. 곡선 아래의 큰 빨간색 영역과 녹색 선과의 높은 교차점을 동시에 최적화하기

Understanding competition metric step by step

https://www.kaggle.com/competitions/amex-default-prediction/discussion/333338

평가 지표에 대한 자세한 설명.

Let's catchup with all the learnings so far

https://www.kaggle.com/competitions/amex-default-prediction/discussion/328565

여러가지 노트북의 유용한 링크 정리

DART algorithm explained

https://www.kaggle.com/competitions/amex-default-prediction/discussion/334670

DART 알고리즘의 설명.

Speed Up XGB, CatBoost, and LGBM by 20x

https://www.kaggle.com/competitions/amex-default-prediction/discussion/328606

▶ CatBoost GPU

Kaggle jupyter 노트북에서 가속기를 GPU로 선택하고, 다음으로 분류기를 생성할 때, task_type='GPU'를 추가.

```
clf = CatBoostClassifier(iterations=5000, random_state=22,
                         task_type = 'GPU')
```

8시간 아닌 25분으로 속도 향상.

Speed Up XGB, CatBoost, and LGBM by 20x

https://www.kaggle.com/competitions/amex-default-prediction/discussion/328606

▶ 메모리 줄이기

Kaggle 노트북에서 실행 시, 노트북 메모리 오류가 발생하지 않도록, 메모리 사용을 최적화 해야 한다. 메모리 오류를 방지하기 위해 다음의 각 K-fold의 for문 끝에 다음의 코드 추가.

```
import gc
del clf, tr_x, val_x, tr_y, val_y, preds, preds_test
gc.collect()
```

Speed Up XGB, CatBoost, and LGBM by 20x

https://www.kaggle.com/competitions/amex-default-prediction/discussion/328606

▶ XGB를 GPU에서 사용하기

GPU에서 XGB를 사용하여 속도 향상된 모델을 사용할 수 있다.

```
xgb_parms = {
    'tree_method':'gpu_hist',
    'predictor':'gpu_predictor',
}
model = xgb.train(xgb_parms)
```

참조 : https://www.kaggle.com/code/cdeotte/xgboost-starter-0-793

Parquet Format Dataset for Low Memory Use

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327138

▶ parquet 데이터 포맷으로 메모리 줄이기

데이터 셋 링크 : https://www.kaggle.com/datasets/odins0n/amex-parquet

데이터 셋 변환 결과
전체 데이터 셋 : 50.31 GB -> 10.41 GB
train 데이터 셋 : 16.39 GB -> 3.35GB
test 데이터 셋 ： 33.82 GB -> 6.96GB

Parquet 파일 로드를 위한 예제 노트북

https://www.kaggle.com/code/odins0n/load-parquet-files-with-low-memory/

9x Data Compression achieved with Feather

https://www.kaggle.com/competitions/amex-default-prediction/discussion/327143

▶ feather를 이용한 데이터 용량 줄이기

데이터 셋 링크 : https://www.kaggle.com/datasets/ruchi798/parquet-files-amexdefault-prediction

Data	Size of csv file	Size of parquet file	**Size of feather file**
train_data	16.4 GB	6.7 GB	**1.8 GB**
test_data	33.8 GB	13.7 GB	**3.6 GB**

수치형 데이터를 float16으로 변경, 범주형 데이터를 category로 변경.

Tabular Classification - Tips and Tricks

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/335892

01 Faster data loading with pandas.

02 Data compression techniques to reduce the size of data by 70%.

03 Optimize the memory by reducing the size of some attributes.

04 Use open-source libraries such as Dask to read and manipulate the data, it performs parallel computing and saves up memory space.

05 Use cudf.

06 Convert data to parquet format.

07 Converting data to feather format.

08 Reducing memory usage for optimizing RAM.

Data exploration

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/335892

01 EDA for microsoft malware detection.

https://www.kaggle.com/code/youhanlee/my-eda-i-want-to-see-all/notebook

02 Time Series EDA for malware detection.

https://www.kaggle.com/code/cdeotte/time-split-validation-malware-0-68/notebook

03 Complete EDA for home credit loan prediction.

https://www.kaggle.com/code/codename007/home-credit-complete-eda-feature-importance/notebook

04 Complete EDA for Santader prediction.

https://www.kaggle.com/code/gpreda/santander-eda-and-prediction/notebook

05 EDA for VSB Power Line Fault Detection.

https://www.kaggle.com/code/go1dfish/basic-eda/notebook

Modeling

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/335892

XGBoost

LightGBM

CatBoost

Ensemble

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/335892

Ensemble

- Ensembling Techniques
- Blending
- Stacking
- Bagging
- Boosting

Hyperparameters Tuning

URL : https://www.kaggle.com/competitions/amex-default-prediction/discussion/335892

01 LGBM hyperparameter tuning methods.

02 Automated model tuning methods.

03 Parameter tuning with hyper plot.

04 Bayesian optimization for hyperparameter tuning.

05 Gpyopt Hyperparameter Optimisation.