Kaggle submission file: submission.csv
Catboost Classifier Trained model: fit_model.pkl
Feature importance file: featimp.csv
Notebook source code: catboost_BNP.ipynb
Screen Shoot for score: score.jpg

Load data
Preprocess data
Distinguish cat and num cols
Distinguish univariate and drop them
Convert low cardinality num to cat
Distinguish ultra high variance feature and drop #didn’t do this
Calc relation to target,drop ultra low rel feat
Reduce redundancy on highly related features #didn’t do this
Fill NA for cat
Fill mean or extreme val (-999) for num
Make features
Cat 2-way comb
Cat 3-way comb
Concat v22 with 2-way combi
Concat v22 with 3-way combi
cat num cat2 cat3 v22 cat2 v22 cat3
Drop specific features by some reason
Train small dataset on CPU
Train on CPU
Save model and feature importance
~~Filter feature on CPU according to model’s feature importance~~
Save filtered feature importance
Save submission
Save model

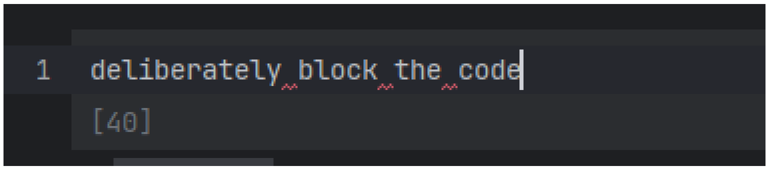
Preprocess again
Filter columns by model’s feature importance
Sort feat importance
Keep cols list accord to importance percentage
Prepare the gridsearchcv parameters
Search for Iteration and depth size #didn’t do this, CPU is too slow

GPU Training
GPU train on small data set
GPU grid search on small data set
Grid search on GPU
Enough money deposition
Search for Iteration and depth size

Often disconnected, segment the hyper parameters, search block by block
Save model
Save feature importance
Save best estimate
Save search path
Save submission

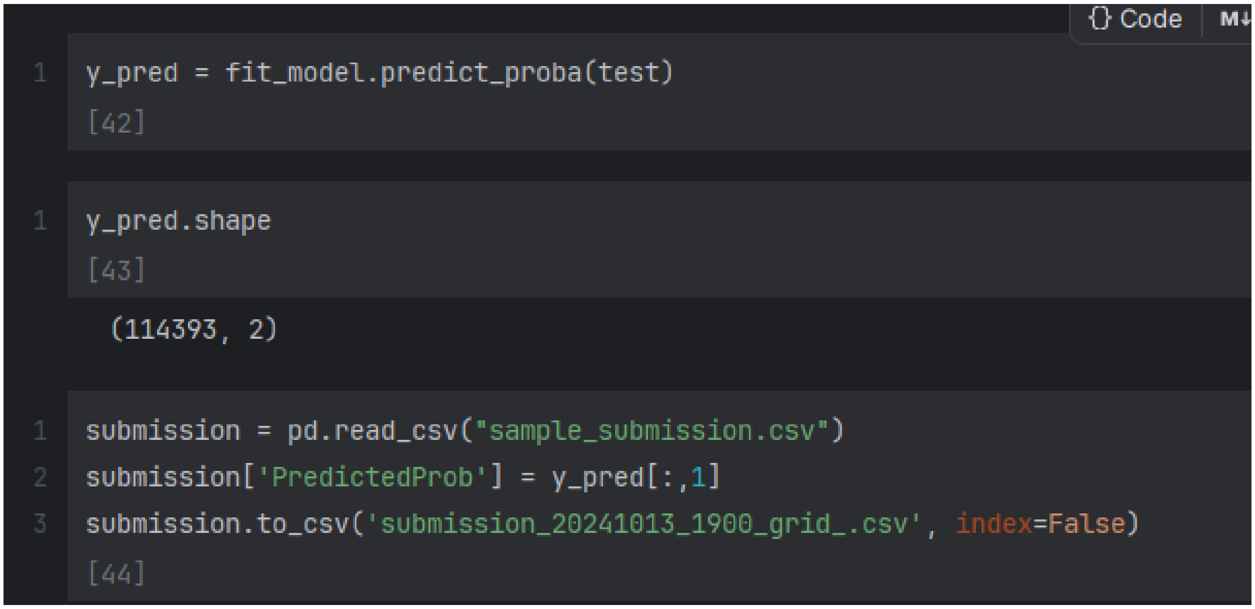
Preprocess data third time for training on CPU #always better score than GPU
Choose feature
Train on CPU
Predict on CPU
Save submission
Save model for future
Save feature importance

Comment on the code



```
deliberately block the code
[40]
```

After model training, I block the code from execution gridsearchcv. Jump over the grid search. Need to manually continue execution after grid search.



```
y_pred = fit_model.predict_proba(test)
[42]

y_pred.shape
[43]

  (114393, 2)

submission = pd.read_csv("sample_submission.csv")
submission['PredictedProb'] = y_pred[:,1]
submission.to_csv('submission_20241013_1900_grid_.csv', index=False)
[44]
```

I also implement the pipeline and preprocessor class. But currently, they do not make a help.

For grid search, I searched iterations from 2000~3000. For depth, I searched 4~10.
Currently the best parameters are iteration:2800, depth:6.
I use joblib to dump the model.
Currently, the model uses more than 2k features.
I save the feature importance to file. We can delete some to speed up training and predict, but it will result in a lower score.