Loan Default Prediction for the Financial Risk Control of an Online Lending Company

Zhehao Guo(zhg26) Xiaoqian Xu (xix64) TongKa(tok31)

Introduction

Peer to peer (P2P) lending is an innovative option for borrowing from individuals without using a traditional bank or credit union. The problem we want to explore is to predict whether a user of online P2P lending website will default in 6 months based on their lending data on the lending website. This is a binary classification.

We are interested in this topic because:

- **Huge dataset**: 3 datasets with 60000 user data with 400 features, including chaotic text data, time-series data, categorial and numerical data, and geographic data
- **Challenges from masking dataset**: meanings of most of features are unknown or vague due to privacy protection
- **Practical**: all data comes from real-world business - FinVolution Group (PPDAI Group Inc).

Data

The dataset discloses the credit risk of loan data from a real online Peer to Peer lending company in China. In order to protect the privacy of users, the dataset provider use data masking techniques to filter sensitive information and hide the meaning of features.

- Source: https://ai.ppdai.com/resource/pdf/PPD-First-Round-Data-Updated.zip
- Description:

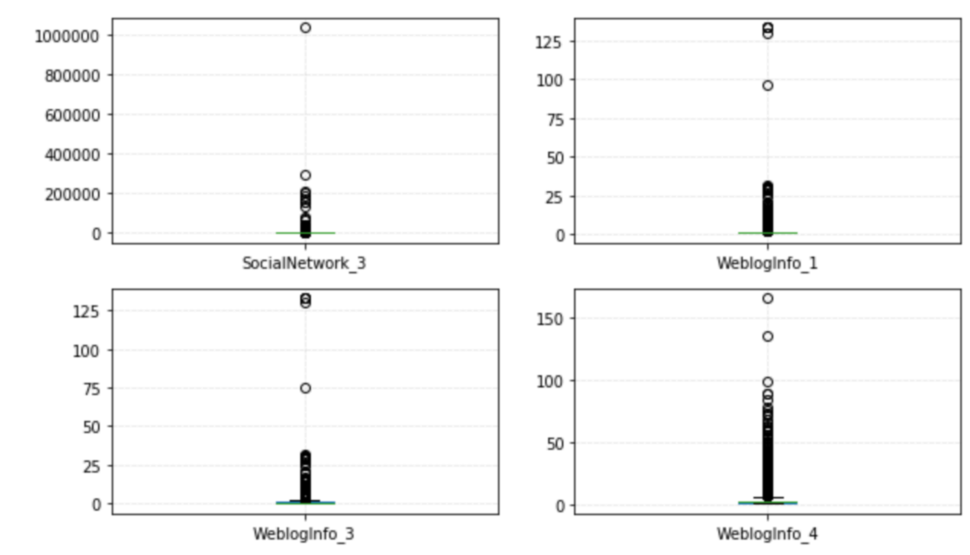
Dataset 1- Master	
idx	Unique key from each loan
target	Target value. Default or not (1= default 0 = not default)
ListingInfo	Date of each loan
UserInfo_*(1-24)	Features describe the users, vague meaning
Education_Info_*(1-8)	Education background, vague meaning
WeblogInfo_*(1-58)	Features describe behaviors of users, vague meaning
ThirdParty_Info_PeriodN_* N(1-7) *(1-17)	Data from third party which contains 17 different features in 7 different periods, vague meaning
SocialNetwork_*(1-17)	Social network related information, vague meaning

Dataset 2 - Log_Info: login records	
idx	Unique key from each loan
ListingInfo	Date of each loan
LogInfo1	Users' operations on the website, vague categories
LogInfo2	Users' operation type
LogInfo3	Users' login time

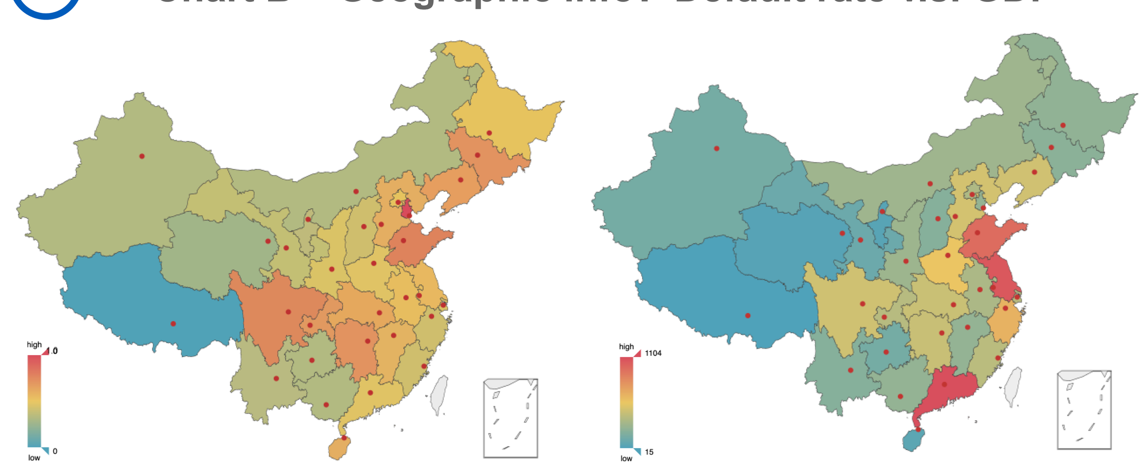
Dataset 3 - Userupdate_Info: update records	
ListingInfo1	Date of each loan
UserupdateInfo1	The field updated by a user
UserupdateInfo2	The date which a user updated his/her information
idx	Unique key from each loan

- Data exploration:

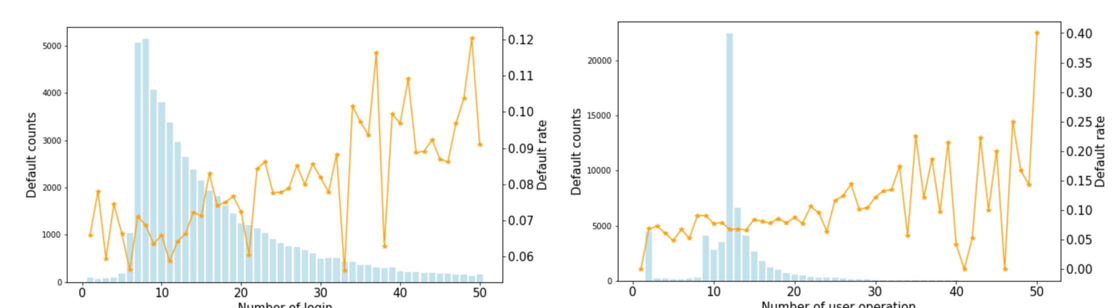
A Chart A - Outliers



B Chart B – Geographic info：Default rate v.s. GDP



C Chart C – Log_Info / UserUpdate_Info



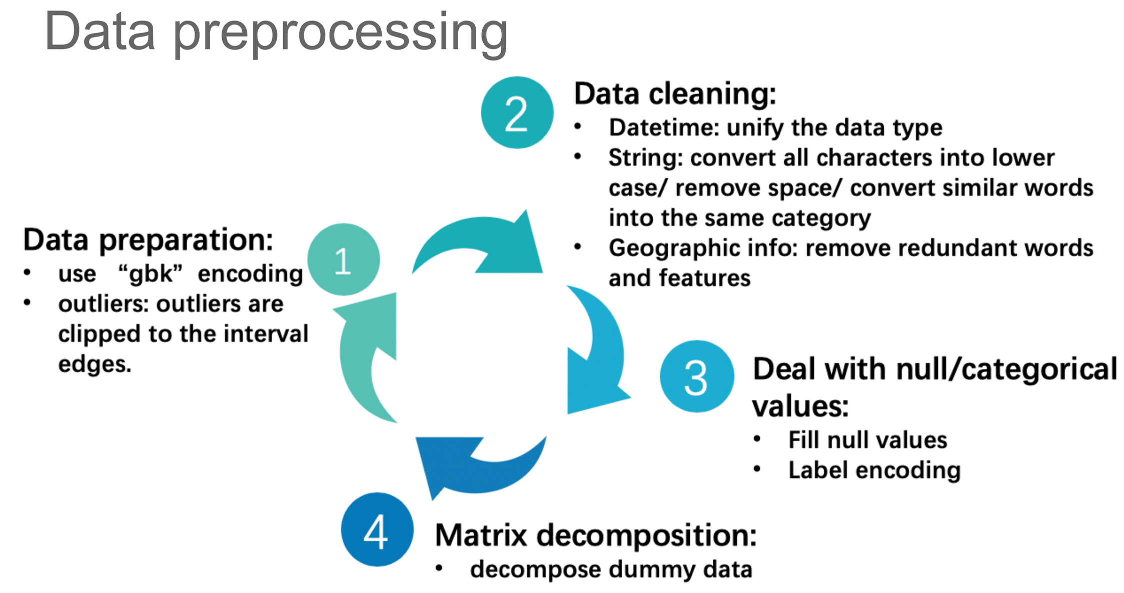
Hypothesis

1. The modified contents and frequency will affect default rate: frequency – positively related/ contents – negatively related
2. The geographic info may affect default risk: Users who live in some specific provinces have higher risk to default

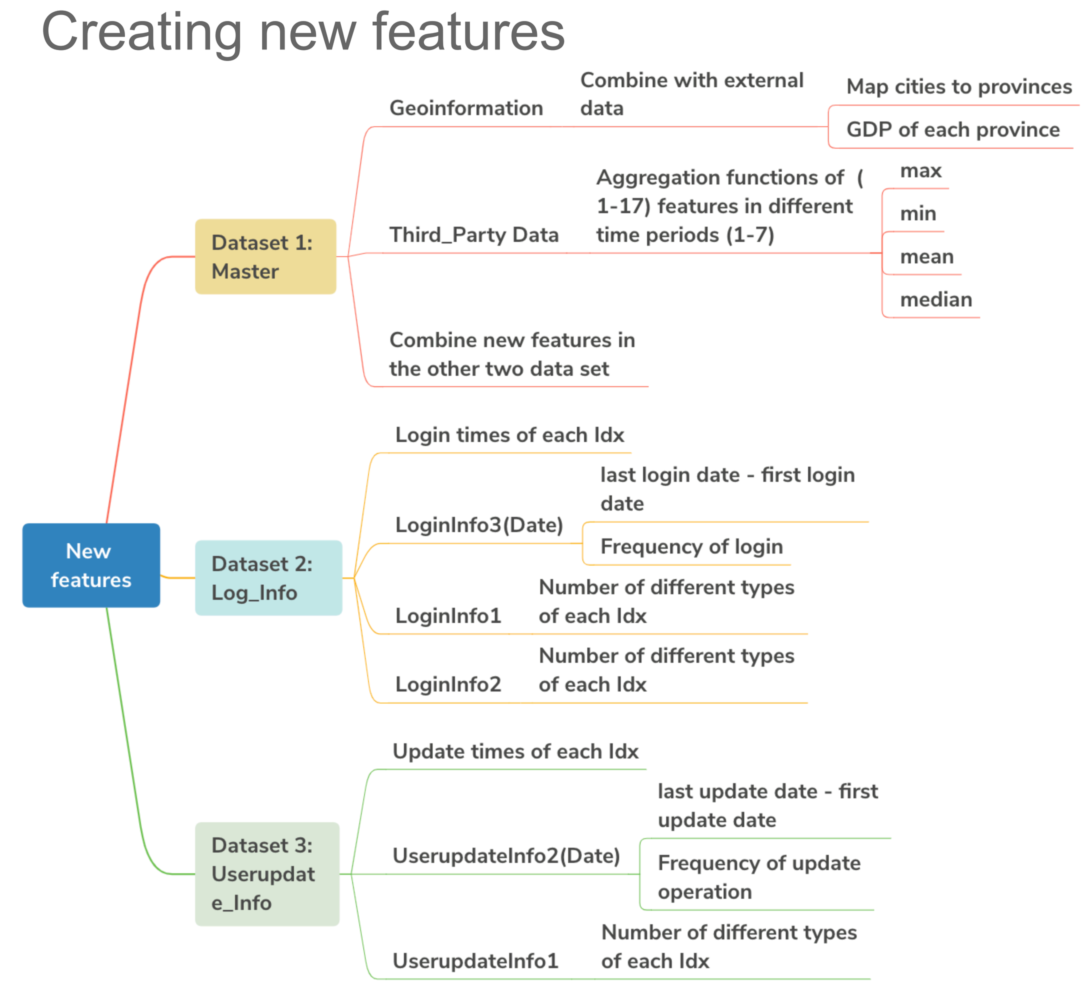
Solution

1. Feature engineering

- Data preprocessing



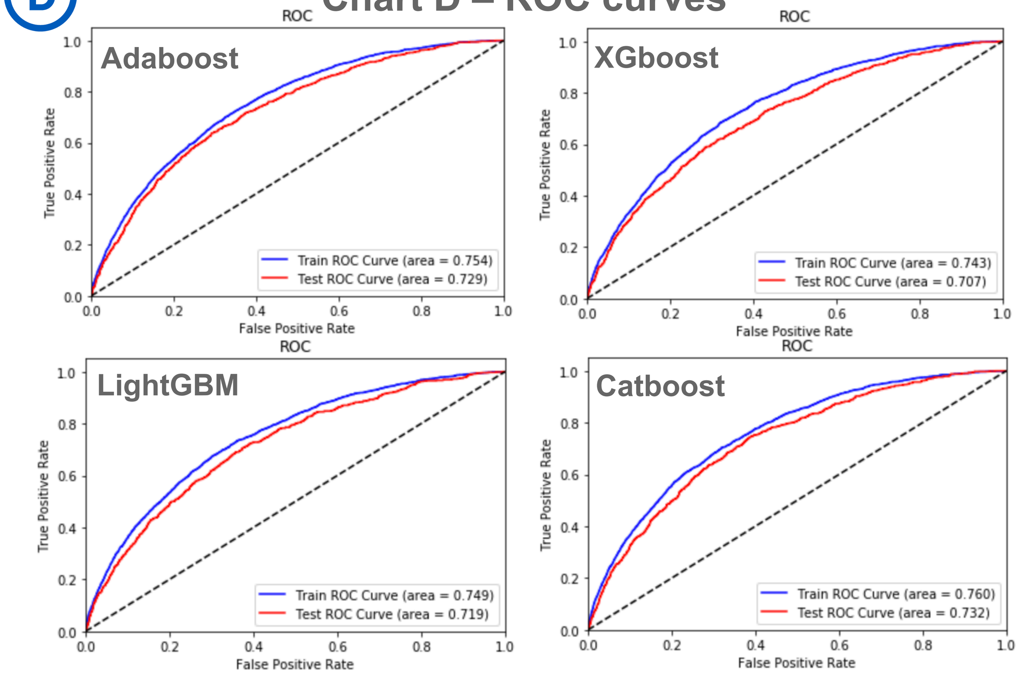
- Creating new features



2. Modeling

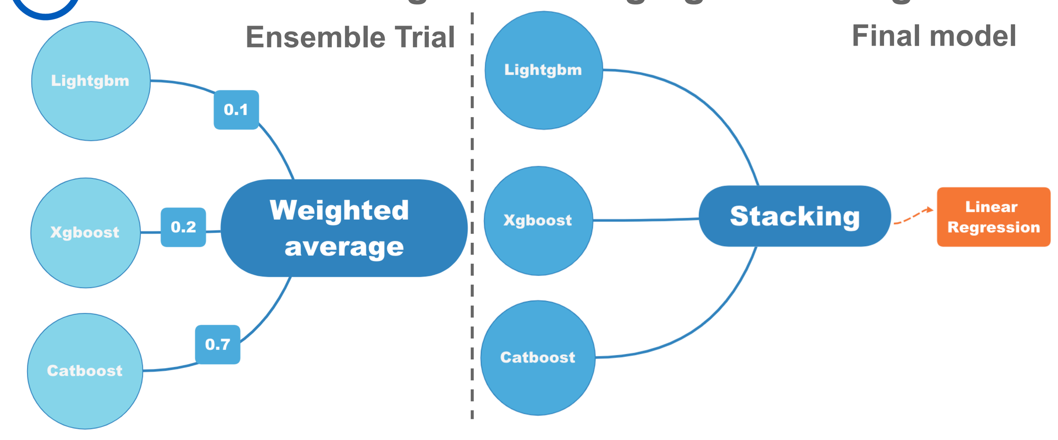
- Gradient boosting trees:

D Chart D – ROC curves



- Ensemble methods:

E Chart E – weighted averaging & stacking

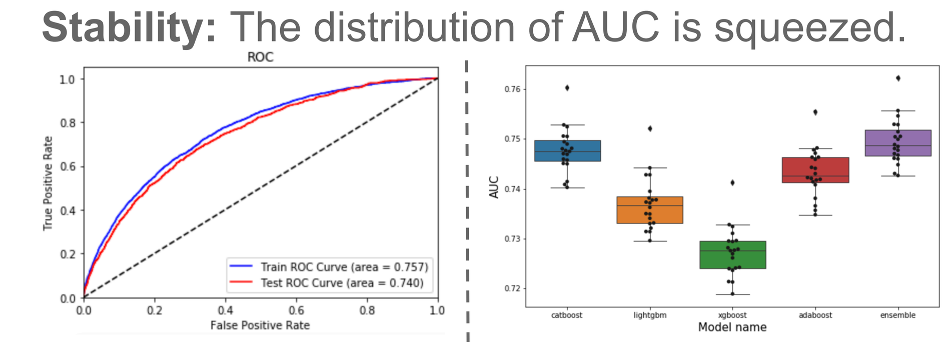


Deployment

To measure performance of our final model and decide the deployment strategy, we compared our model based on following 3 attributes by using bagging. We think our model can be deployed into practice.

- **Good performance:** Test AUC (Stacking) = 0.757

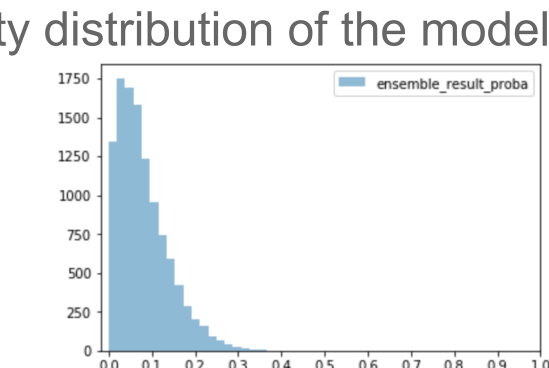
  Train AUC – Test AUC = 0.017
- **Recall:** 0.68 (highest among 5 models)
- **Stability:** The distribution of AUC is squeezed.



Findings

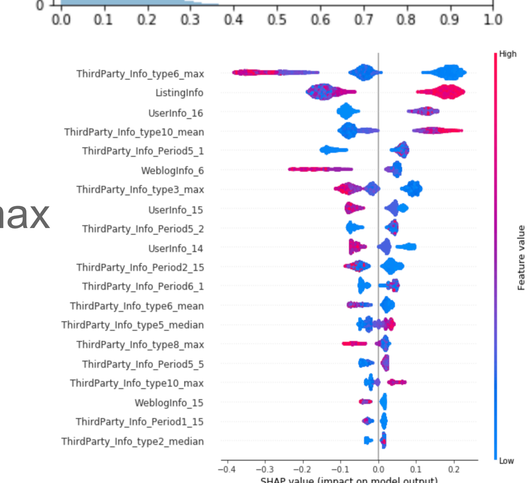
1. We will identify defaulter in terms of risk based on the prediction probability distribution of the model.

- **Low risk**: < 0.03
- **Medium risk:** >=0.03, < 0.1
- **High risk**: >= 0.1



2. Feature importance:

- **Most important feature**: ThirdParty_Info_type6_max
- **Features included in Third Party information are very significant.**



Conclusion

1. Technical strength: Ensemble methods;
2. Innovation: Geoinformation, aggregation function, matrix decomposition, deployment strategy;
3. Challenge: Masking data, data cleaning, model hyperparameters tuning, ensemble technique.

References

1. Pyecharts: github.com/pyecharts/pyecharts
2. SVD: sklearn.decomposition.TruncatedSVD
3. NMF: sklearn.decomposition.NMF
4. Ensemble methods: zhangruochi.com/Ensemble-Methods/2019/07/17/