Loanliness: Predicting Loan Repayment Ability

Yiyun Liang,[1] Xiaomeng JIn,[1] Zihan Wang[1]

Department of Computer Science, Stanford University

Introduction

Due to insufficient credit histories, many people are struggling to get loans from trustworthy sources. The untrustworthy lenders can take advantage of these borrowers by taking high interest rates or including hidden terms in the contract. Instead of evaluating the borrower based on their credit score, there are other alternative ways to measure or predict the loaners repayment. In our project, we use machine learning algorithms to study the correlations between borrower status and repayment ability.

Data Processing

The data pre-processing step has three main parts: feature

- **Feature concatenation:** By using the unique ID numbers of the data entries, we concatenate all the features together. After feature concatenation, each data point has 217 features in total.
- **Label Encoding and One-hot Encoding:** We use label encoding and one-hot encoding to encode features with string-type categorical values.
- **Invalid/Empty Entry Replacement:** If the percentage of invalid values in the column is greater than the threshold, 30%, we mark the feature as an invalid feature and remove the column from the dataset. Otherwise, we just remove the row which contains the invalid value.

Models

K-means clustering for loaner categorization

We used K-means algorithm to first cluster the loaner into different groups before performing classifications on each group to get the best performance.

Supervised machine learning classifiers

We used the following supervised machine learning algorithms to tackle the classification problem:

- **Logistic regression, Random forest, Naïve Bayes, Multi-layer perceptron (MLP), LightGBM** (a tree based classifier with gradient boosting)

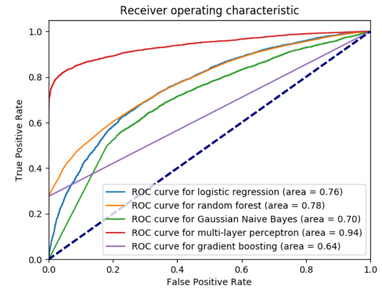
Results

Method 1: Running classification using different algorithms

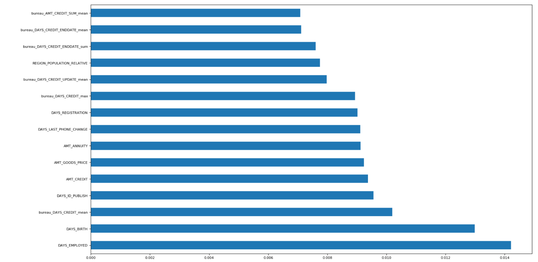
Model	Accuracy	Precision	Recall	F1
LogReg	69.34%	0.66/0.75	0.81/0.58	0.72/0.65
RandForest	63.51%	0.58/1.00	1.00/0.27	0.73/0.43
NB	52.11%	0.51/0.71	0.97/0.07	0.67/0.13
MLP	62.59%	0.57/1.00	1.00/0.25	0.73/0.40
LightGBM	57.47%	0.54/1.00	1.00/0.15	0.70/0.26

Method 2: K-means clustering + classifications

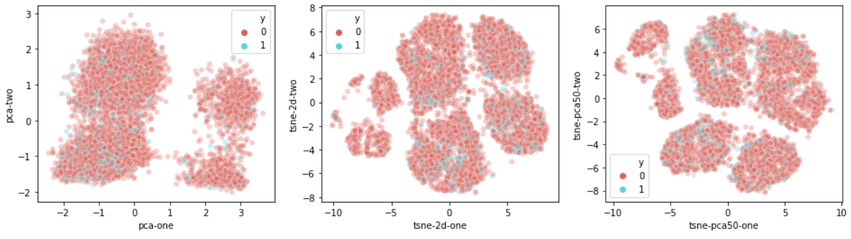
Model	Accuracy	Precision	Recall	F1
Cluster 1	72.24%	0.63/1.00	1.00/0.47	0.77/0.64
Cluster 2	67.01%	0.62/1.00	1.00/0.28	0.77/0.43
Cluster 3	82.34%	0.77/1.00	1.00/0.56	0.87/0.72
Cluster 4	70.78%	0.56/1.00	1.00/0.53	0.72/0.69
Overall	71.57%	0.63/1.00	1.00/0.43	0.77/0.59



ROC curves for the different learning algorithms



Feature Importance extracted from Random Forest Classifier



Visualization of high-dimensional data with PCA and t-SNE

Conclusion

Based on our results and observations, random forest seems to be the least prone to the problem of imbalanced dataset. We also observed that performing k-means clustering first, and then running classification on different clusters separately helps in improving model performance.

References:

- https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b
- Ke, Guolin, et al. ”Lightgbm: A highly efficient gradient boosting decision tree.” Advances in Neural Information Processing Systems. 2017.