# High-dimensional analysis on transactions for potential customers

SUN Maojun[1*]

[1*]Department of AMA, PolyU, Kowloon, Hong Kong, China.

Corresponding author(s). E-mail(s): maojun.sun@connect.polyu.hk;

**Abstract**

In this report, I tackled a regression prediction task for potential customer identification with a high-dimensional dataset. To address the issue of dimensionality, I applied the PCA (Principal Component Analysis) technique for dimensionality reduction. Subsequently, I utilized various regression models, including linear regression, lasso regression, ridge regression, support vector machine, Random forests, Neural network, Gradient boosting, and lightGBM, to develop predictive models. Through experimentation, I achieved a promising RMSE score of 1.1 by LightGBM, indicating the successful completion of the prediction task.

**Keywords:** High-dimension, PCA, LightGBM

## 1 Introduction

The task of predicting the value of transactions for potential customers, also known as transaction value prediction or transaction prediction modelling, is a common problem in data science and predictive analytics. It involves using historical transaction data and other relevant features or variables to build a predictive model that can estimate the transaction value for potential customers or future transactions.

Transaction value prediction is an important business application in various industries, such as e-commerce, retail, finance, and telecommunications. It can help businesses make informed decisions about resource allocation, marketing strategies, pricing, and revenue forecasting. By accurately predicting transaction values, businesses can optimize their operations, tailor their marketing efforts, and maximize their profits.

# 2 methodology

## 2.1 DataSets

Datasets link: https://www.kaggle.com/competitions/santander-value-prediction-challenge/data

The datasets of this study comes from the Kaggle event: “Santander Value Prediction Challenge”. This datasets provide an anonymized dataset containing numeric feature variables, the numeric target column, and a string ID column. (anonymized means I can not get any information from the name of feature). I only use train.csv for experiment because only the file contain label.

The dimension of data is 4459 × 4993, which is quite high dimensional datasets. And the variables are all numeric except ID. Luckily, there is not any missing value in observation. This will extremely facilitate data processing task.

## 2.2 Data pre-processing

Firstly, I observed the distribution of targets. I found the distribution is quite skewed, as Fig1. Besides, the variance is quite large. The min value is 30000, the max value is 40000000, and the average is 5944923.

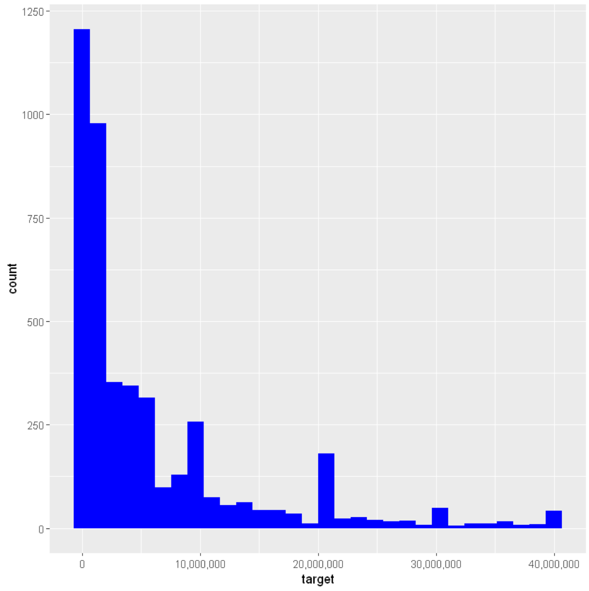


**Fig. 1** Distribution of target

So, I addressed the target by a log function as Eq1:

$$Y = \log(1 + y). \tag{1}$$

After processing, the distribution of target comes to Fig2, and the min value is 10.31, the max value is 17.50, and the average is 14.49.

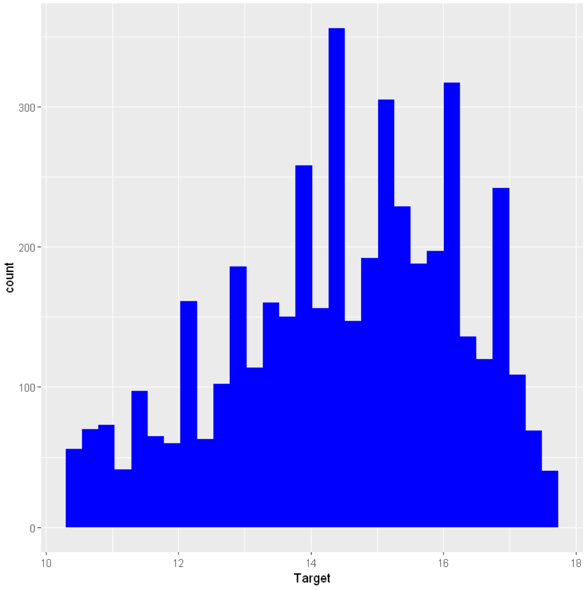


**Fig. 2** Distribution of target after log function

Besides, I observed the correlation of the first 20 variables, as Fig3. There are not many high-correlated variables.

## 2.3 Feature engineering

I deleted the column "ID" because it is meaningless. Delete some variables with 0 variances. Because this usually means all observations are the same. Then, the dimension comes to 4459 × 4735. After that, standardized the variables and made a principal component analysis so that I can reduce the dimension. The PC1 is 0.02392, and PC589 is 0.7001, as Fig3. This means I can describe 70% of the data using only 589 variables. So, I reduced the dimension to 4459 × 589, which is extremely reduced the dimension.

I divided the data with a proportion of 8:2. 8 for train set, 2 for test set. Used 5-fold cross-validation to make the result more credible.

## 2.4 Model

I Built some predictive models using machine learning algorithms. Including linear regression, lasso regression, ridge regression, SVM, Random forests, Neural network, Gradient boosting, lightGBM, etc. The model is trained using the historical transaction data and the selected features.

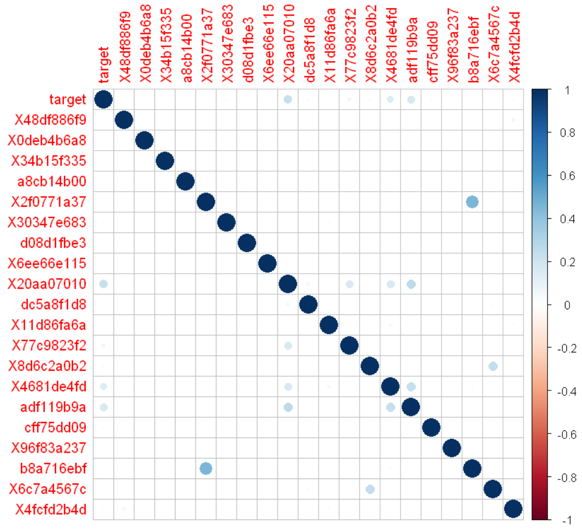
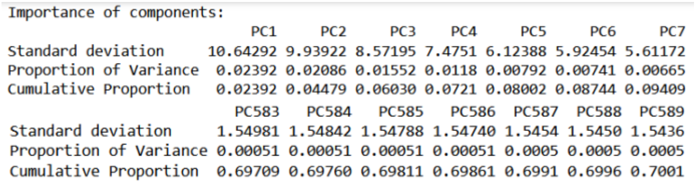


**Fig. 3** The correlation of the first 20 variables



```
Importance of components:
                            PC1     PC2     PC3    PC4     PC5     PC6     PC7
Standard deviation     10.64292 9.93922 8.57195 7.4751 6.12388 5.92454 5.61172
Proportion of Variance  0.02392 0.02086 0.01552 0.0118 0.00792 0.00741 0.00665
Cumulative Proportion   0.02392 0.04479 0.06030 0.0721 0.08002 0.08744 0.09409
                          PC583   PC584   PC585   PC586  PC587  PC588  PC589
Standard deviation      1.54981 1.54842 1.54788 1.54740 1.5454 1.5450 1.5436
Proportion of Variance 0.00051 0.00051 0.00051 0.00051 0.0005 0.0005 0.0005
Cumulative Proportion  0.69709 0.69760 0.69811 0.69861 0.6991 0.6996 0.7001
```

**Fig. 4** Result of principal component analysis

For linear models. Ridge regression controls the complexity of the model by limiting the size of the model parameters by adding a penalty term of L2 parity to the objective function, the object function is Eq2.

$$\text{minimize } J(\theta) = \text{MSE} + \alpha \sum_{j=1}^{p} \theta_j^2 \tag{2}$$

Lasso regression limits the size of the model parameters by adding a penalty term of L1 parity to the objective function to achieve sparsity of the model, i.e. automatic selection of the important characteristic variables. Its mathematical formulation is as Eq3.

$$\text{minimize } J(\theta) = \text{MSE} + \alpha \sum_{j=1}^{p} |\theta_j| \tag{3}$$

Random forests and LightGBM are both tree-based ensemble learning algorithm. There are some differences like feature selection. Random forests randomly select a

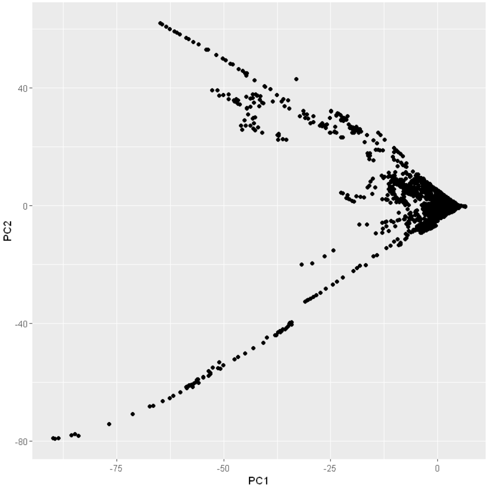


**Fig. 5** PC1 and PC2

subset of features at each split, while LightGBM considers all features for each split. This can make LightGBM more sensitive to noisy or irrelevant features, but it also allows it to potentially capture more complex interactions between features.

### 2.5 Evaluation

I mainly evaluate the model performance on RMSE. MAE was used as an auxiliary judgment.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{4}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{5}$$

## 3 Results

The evaluated results on the test set e are as Table below.

| Model/Score | RMSE | MAE |
|---|---|---|
| Linear | 2.928906 | 1.824384 |
| Lasso | 1.926282 | 1.501868 |
| Ridge | 1.67311 | 1.362062 |
| Elastic | 1.680414 | 1.369406 |
| SVM | 1.697674 | 1.380873 |
| RF | 1.514563 | 1.357896 |
| MLP | 1.954889 | 1.55686 |
| XGB | 1.679852 | 1.359144 |
| LightGBM | **1.170869** | **0.9386955** |

Feature importance output by LightGBM can be seen in Fig6

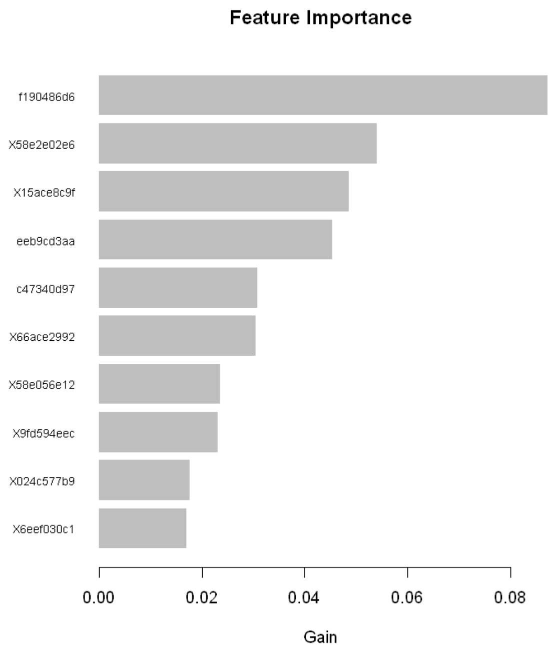


**Fig. 6** Feature importance outputed by LightGBM

# 4 Disscussion

## 4.1 PCA

The use of PCA for dimensionality reduction in my regression prediction task has both advantages and limitations. One major advantage is that PCA can effectively reduce the dimensionality of the high-dimensional dataset, which can help mitigate the curse of dimensionality and improve model performance and accelerate the training speed.

However, it's important to note that PCA is a linear technique that may not capture all the nonlinear relationships present in the data. This could potentially result in loss of information and reduced predictive performance, especially if the data has complex nonlinear interactions. In my study, although PCA was able to achieve a satisfactory RMSE score of 1.1, it's possible that other nonlinear dimensionality reduction techniques, such as t-SNE or UMAP, could have yielded even better results, which can be done in a future experiment.

Another limitation of PCA is the potential loss of interpretability, as the principal components obtained from PCA may not have direct physical or meaningful interpretations. This can make it challenging to interpret the contribution of each feature to the prediction task, which may limit the ability to gain insights and make informed decisions based on the model results.

Additionally, the optimal number of principal components to retain in PCA can be subjective and may require tuning (70% is my subjective). In my study, I may have made certain assumptions and choices in selecting the number of principal components to retain, which could have impacted the performance of the models. Further experimentation and sensitivity analysis on the choice of the number of principal components could be considered in future work to understand its impact on a model's performance better.

### 4.2 Model

From the feature importance of LightGBM, we can conduct further fine-tuning of hyper-parameters and redo feature engineering to give different weights to features.

From the train and evaluated results, we can see Lasso and Ridge have a better predictive for unknown samples. This is probably due to the regularization of L1 and 12, which slows down the over-fitting problem.

The test performance is better than the train/val performance in some ensemble models like LightGBM, which means we can run more epochs in the training process.

From Fig6, we can see f190486d6 is the most essential feature in this data. However, I can not know what it actually means.

## 5 Conclusion

In this study, I successfully addressed the challenges of a high-dimensional datasets by PCA for dimension reduction. I then applied multiple regression models and tree-based models to achieve a favourable RMSE score of 1.1, indicating the effectiveness of my predictive models for the task of potential customer identification. My findings highlight the significance of dimension reduction and appropriate feature selection techniques and regression models in achieving accurate predictions for high-dimensional datasets. The results of this study may have practical implications for businesses and marketers in identifying potential customers and optimizing their strategies for customer acquisition. Further research could be conducted to explore additional techniques for improving predictive performance and exploring other machine learning models for similar tasks.

## 6 Reference

[1]https://www.kaggle.com/code/couyang/santander-value-prediction-lightgbmpreprocess
[2]https://www.kaggle.com/code/ankur310794/eda-hack-pca-correlation
[3]https://www.kaggle.com/competitions/santander-value-prediction-challenge/data