SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Computing and Information Systems

AY 2022/2023 Term 2

ISSS610: Applied Machine Learning

Final Report

Section G1 – Group 3

Home Credit Default Risk Analysis

Professor: Dai Bing Tian

Group Members: *Ding Yanmu, Huang Anni, Rao Ningzhen, Ren Xuezhe, Yu Di*

Table of Contents

1. Introduction

Default risk is the risk that a lender takes on in the chance that a borrower will be unable to make the required payments on their debt obligation. Accurately predicting the default risk will help financial institutions cut the loss greatly and hence is a heated research topic nowadays. Before granting a loan to an applicant, it is a common practice for financial institutions to collect the background information of the applicant and try to analyse the ability to repay the loan. It is crucial that the financial institution can distinguish important information from the ones that are less informative and develop smart algorithms to analyze the default risk given the large and messy historical data they collected.

To solve this real-world problem, our group participated in a Kaggle challenge posted by Home Credit, an international financial institution operating in 9 countries focusing on instalment lending. Home Credit makes use of a variety of alternative data--including telco and transactional information--to predict their clients' repayment abilities. Our group decided to use various statistical and machine learning methods to unlock the full potential of their data. As a result, we can successfully predict over 80% of the default cases and achieve overall 98% accuracy.

2. Related Work

We explored some previous work in this field. literature on predicting default risk can be categorized into two classes. The first category uses traditional mathematical models, such as ordinary least squares (OLS), Logistic Regression, to explore the possible factors affecting the probability of repayment failure. The second category adopts machine learning methods to predict the probability of credit default. Chen et al. (2013) used logistic regression to predict the impact of all the information on a Chinese P2P lending platform. They argue that the borrower's credit status, living conditions, region of residence, personal income, number of successful borrowing attempts and number of on-time repayments negatively influence a borrower's default probability, whereas the number of overdue payments, years of education, loan interest rate, and number of ahead-of-schedule payments positively influence a borrower's overdue rate. Xu.et.al (2021) employed XGBT to examine the various factors that affect loan repayment and found that a higher credit score, onsite certification and guarantees improve the probability of repayment. In contrast, verification of a borrower's job, income, mobile phone, residence, or marital status has a negative effect on repayment success.

3. Dataset

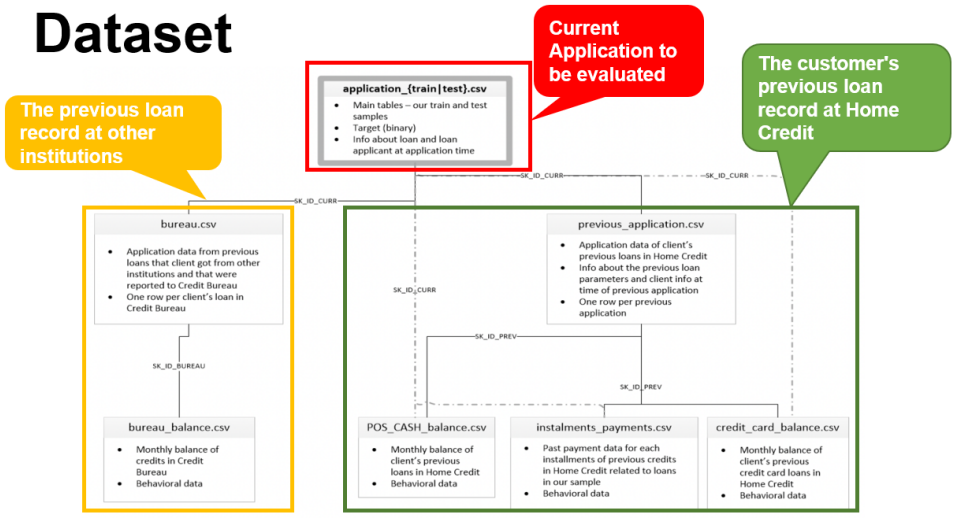

Fig1: Overview of our dataset

The whole dataset contains 8 tables as shown in Fig 1, application train and test data are the main tables that contain the target, the information about the loan and information about the loan applicant at the application time. The other 6 tables contain information about the credit history of the applicants. In practice, the occurrence of default is far less than normal repayment. We found that among 300 thousand training data, where label '1' indicates default, only 8% of it has positive labels. We also found that some tables, such as the bureau balance

table, have too many rows, because one previous credit record may have many associated rows in those tables, and one applicant may have multiple credit records. Hence, we need to do some pre-processing to aggregate the rows in these tables.

4. Evaluation Metrics

Dealing with highly imbalanced data (1:10) as shown in Fig 2, we need to choose our evaluation metrics carefully. We select recall rate and AUC score as our evaluation metrics for our models. We chose to prioritise recall over precision mainly based on our domain knowledge: The problem with a low recall is that the company would incur a lot of loss associated with bad debts and that agents will spend much time and effort trying to get the payment from the applicant. While the problem of low precision is that the company will lose some customers and revenues. We believe that misclassifying a default application will lead to greater loss to the company. We also chose AUC score as we want to evaluate the overall ability of classifying positive instances from negative instances of the model. We can change the threshold to make recall score higher, but the AUC score will remain the same and by referring to it, we make sure our model isn't going too far on the way of sacrificing precision for recall.

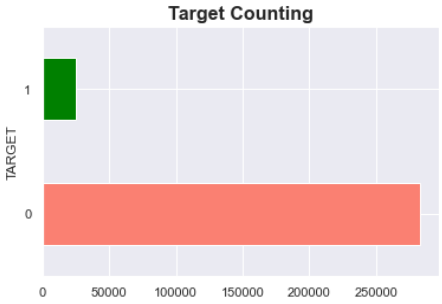

Fig 2: Highly unbalanced data

4.1 Missing value

We plot the percentage of missing values for each feature in the main table as shown in Fig 3. There are a lot of missing values in the dataset. Features related to the current accommodation of the applicant tend to have a larger percentage of missing values

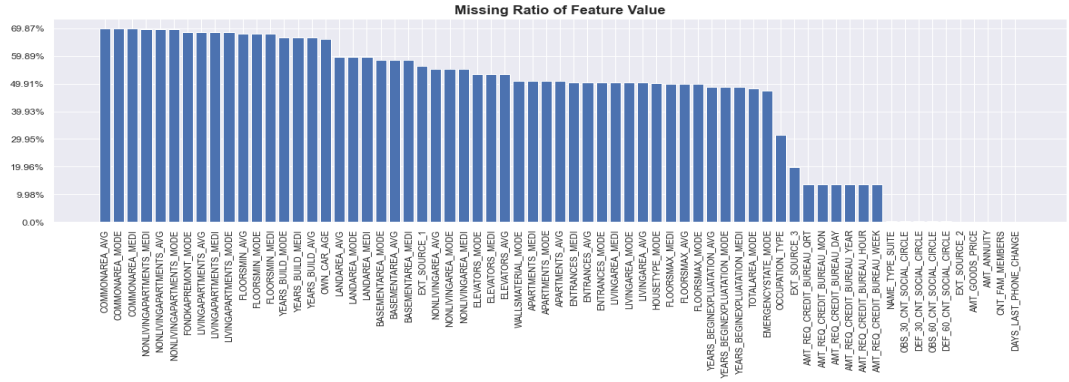

Fig 3: Missing ratio for some features

4.2 Feature distribution

According to the histogram and boxplot in Fig 4 and 5, many numerical features are skewed, such as AMT annuity and living area. To make our model converge faster, we may consider normalization of the data.

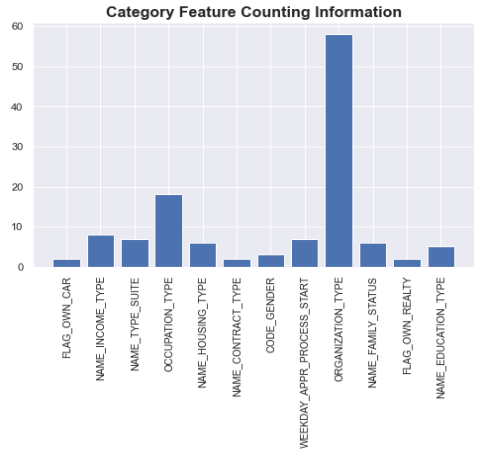

Fig 4: Histogram for numerical features

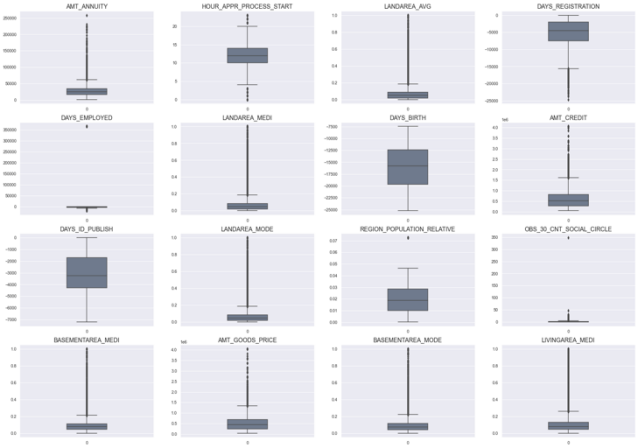

Fig 5: Boxplot for numerical features

4.3 Feature correlation

From the phi-k correlation of the features demonstrated in Fig 6, we noticed that the occurrence of correlated feature pairs is very common in our dataset, such as the number of children and the number of family members, which we may consider removing.

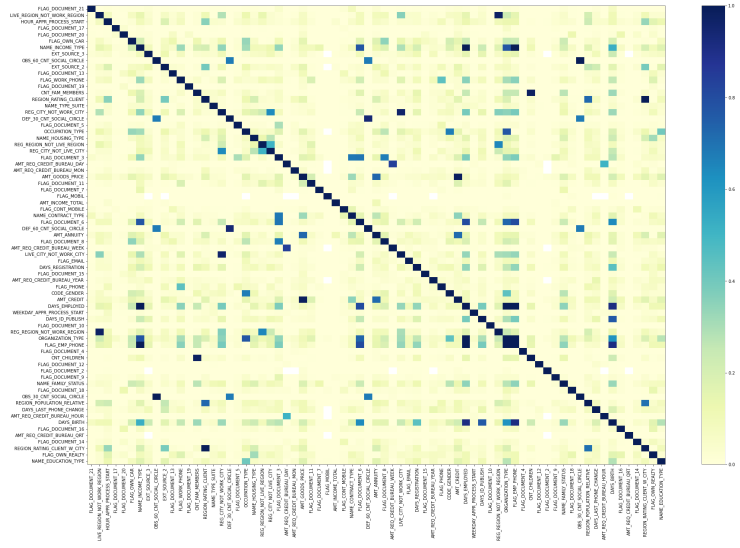

Fig 6: Correlation plot for all numerical features

5. Data Pre-processing

5.1 Joining tables

After getting the data and EDA of this project, the first step we do is using the pandas library provided in Python to read all the data in provided csv files and merge them together based on the SK_ID_CURR column, which records all the ID of loans. And we used the left outer join method to merge all the other tables with the application table, because the main training data for our final model are in application.csv file.

5.2 Data cleaning

After joining all the tables, we found that there were a lot of null values in this client data table. Among the four models (LightGBM, logistic regression, random forest, DeepFM) we selected to predict whether the client has potential repayment difficulties, only the LightGBM model can directly use the category features to train our final model. All the rest models need to perform one-hot encoding on the data first. In order to finally compare the prediction accuracy of these four models, we must use the same dataset to train all of them, so we have to remove some clients' data. Our three criteria to dropout are shown as followed:

1. Deleting all the rows without target values. If we do not even know whether the client can repay the money or not, it's almost impossible to let the computer give us an accurate prediction.
2. Deleting all the columns without detailed description. It will be very difficult for us to tune the weight of these features later.
3. Deleting the clients' data which has a lot of null values. Here, as long as the ratio of null values in a client's information is more than 10 percent, we just delete this row. Such procession will also improve the accuracy of the trained model.

5.3 One-hot encoding for categorical features

The third we do is to perform one-hot encoding on all the category data. Here, we use the get dummies() function in the pandas library to finish this step.

This process can transfer all the non-numerical data to numerical ones. Then we also standardized and normalized all the numerical data to make sure all of them have the same value range. This is the preparation for later features tuning.

After preprocessing the data, we used the agg() function in pandas library to aggregate all the rest columns, and calculated the mean value of these features. The reason for us to choose mean value here is that it can make sure all of these training data can have certain commonalities. In the following image, the code in the red box uses the agg() function to aggregate mean value for each feature in the *credit_card_balance* table. The code screenshot is shown in Fig 9.

5.4 Standardization and normalization for numerical features

We standardized and normalized our numerical data to make the convergence faster. For standardization, we use the robust scaler which is resistant to outliers. For normalization, we use l2 normalization.

5.5 Stratified train test splitting

In our project, the target value will equal to 1 when the clients have difficulties to repay back the money. And target value equals 0 means all the other situations. The ratio for positive and negative samples is around 2:8 which is highly unbalanced. Therefore, we use the stratify method to split our test dataset from our whole dataset in order to make the ratio of the test dataset close to 20%.

6. Models

In general, we have tried three machine learning models and one deep learning model. The machine learning models include logistic regression and two tree-based models, namely random forest and LightGBM. We select them as people in credit risk assessment usually use those models. For deep learning, we use DeepFM, which is a popular model in the recommendation field. We want to try if this works well in risk assessment as well. The performance of those models is shown in Table 1. As we can see, LightGBM has the second highest recall rate, and it runs much faster than DeepFM. Thus, we select LightGBM as the best model.

Table 1: The performance for all the models

KPI	Logistic Regression	Random Forest	LightGBM	DeepFM
roc_auc	0.85	0.9836	0.9448	0.8492
recall	0.3116	0.7907	0.8296	0.8326

Now, we'll go through the model architecture, results and hyper-parameter tuning for each model individually.

6.1 Logistic Regression

Logistic Regression is a simple classification model which uses sigmoid function to do binary classification.

The most important hyperparameter for logistic regression is C. A smaller C will lead to stronger regularization. We tuned the C from 0.01 to 100 and found that the best C is 100. Which means our model is under-fitting and does not need regularization.

As expected, it's not performing well. It has the worst recall rate on our dataset. The recall rate is only 0.3. But the AUC score is ok, it's 0.85.

6.2 Random Forest

The random forest is a classification algorithm consisting of many decision trees. It uses bagging and features randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

We tuned four hyper params for random forest, namely *n_estimators, max_depth, max_features, min_sample_split*. And find that for our problem, the max_depth's optimal value is 40, min_sample_split's optimal value is 0.78. *n_estimator*'s best value is 20.

Random forest has the highest AUC score, around 0.98. But its recall rate is the second worst. Why is that? High AUC means your algorithm does a good job at ranking the test data, with most negative cases at one end of a scale and positive cases at the other. And low recall means we are not very good at identify the ones can't pay the loan. But in our scenario, we care recall more than AUC. In credit risk detection, the AUC score is not accurate since positive samples don't occur as many times as the negatives (the negatives are too high). In such cases, the AUC might go high not because of large true positives but because of large false positives. It is just a side effect of having too many negative samples.

Besides, we plot the 5 most important features for random forest as shown in Fig 7. These features can be used to help the staff in banks to determine whether their clients are capable of paying the loan.

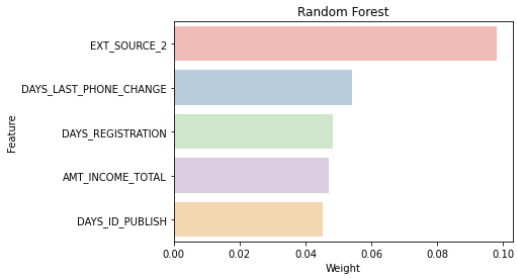

Fig 7: The 5 most important features selected by Random Forest

EXT_SOURCE_2: Normalized score from external data source, a higher document fill-in rate means the client is more serious about the loan and thus more capable to pay the loan. DAYS_REGISTRATION: How many days before the application did the client change his registration. If the client changes his application just before the registration. He or she may have not thought it carefully. AMT_INCOME_TOTAL: Income of the client, which largely determines how capable this client is for paying his or her loan. DAYS_ID_PUBLISH: How many days before the application did the client change the identity document with which he applied for the loan. A sudden change in the identity document implies that the client is sloppy, and we should be careful to give loan to this kind of customer.

6.3 LGBM

Light Gradient Boosting Model is an implementation of gradient boosted decision trees designed for speed and performance. In general, gradient boosting tree algorithms consider the gradient of the loss function and build the next subtrees as such the gradient of the loss function is maximised. For each subtree, the previous GBDT only considers first-order gradient to simulate the residuals, leading to a level wise growth of the subtrees, while LGBM includes second-order gradient and histogram algorithms to accelerate the calculation and adopts leaf-wise tree growth. LGBM is faster than other GBDT algorithms and works better with big datasets. We Implemented LGBM with K-fold cross-validation. We set an early stopping Since the LGBM itself is very fast to train, we managed to do its hyperparameter tuning automatically by using Bayesian optimization. We also computed and tried to maximise the recall of our model by changing the classifying threshold. Finally, we obtained the feature importance denoted by the frequency of splitting with certain features.

We trained the model multiple times with randomly generated hyperparameters, such as the maximum depth of subtrees, subsample, learning rate. And implemented Bayesian optimization to find the optimal set of hyperparameters.

The result of LGBM is to our satisfaction, the AUC is 0.94 and recall can be as high as 0.83 with a threshold of 0.1. The accuracy is as high as 0.93 because of sacrificing many false-positive cases to achieve a higher true positive rate. To make our model more explainable to the audience, we plot the most important features in Fig 8. Features such as total income and living region population topped the list, aligning with common sense.

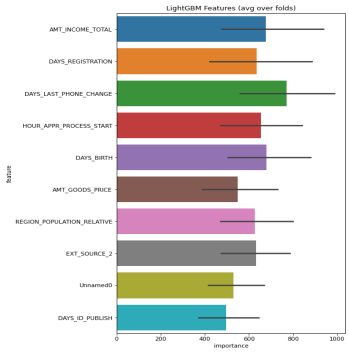

Fig 8: The 5 most important features selected by XGBoost

6.4 DeepFM

DeepFM is first used by Guo et.al (2017) to predict CTR(Clickthrough rate) tasks, and it could achieve some impressive results. This model whose architecture is shown in Fig 9, is based on FM and linear regression and aims at both low-order and high-order feature interactions.

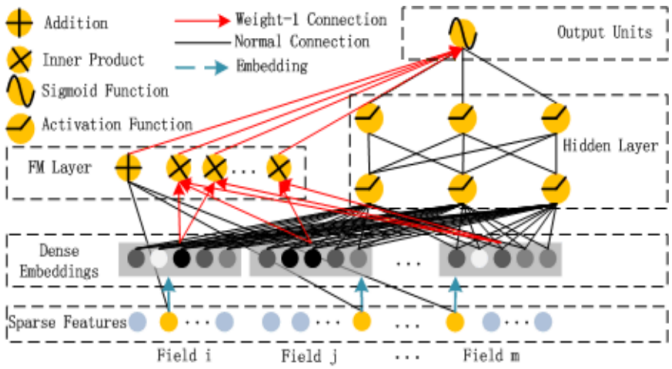

Fig 9: DeepFM model architecture

DeepFM consists of 2 components, FM part, and deep part, which are shown in Fig 10 and 11 respectively. They share the same inputs.

- FM part

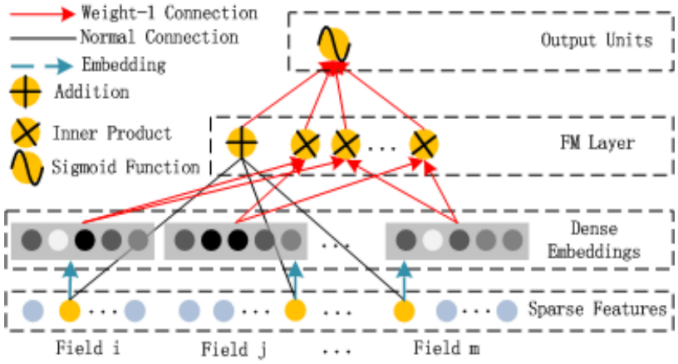

Fig 10: DeepFM FM part architecture

This part is a factorization machine to capture 2-order feature interaction more efficiently even when the dataset is sparse.

- Deep part

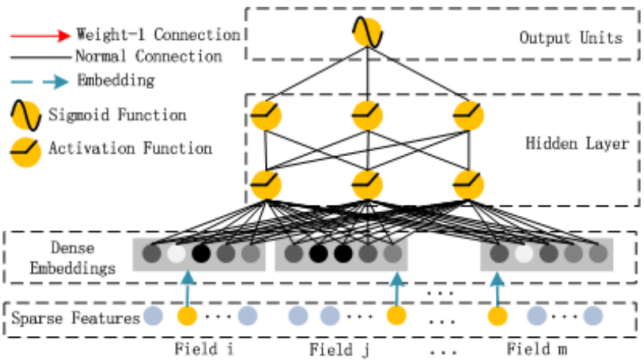

Fig 11: DeepFM FM part architecture

The deep component is a feed-forward neural network, which is used to learn high-order feature interactions. Consider our raw input data contains a large amount of high-dimensional categorical features mixed with continuous features. We include an embedding layer to compress the input vector to a low dimensional, dense real-value vector before further feeding into the first hidden layer, otherwise the network can be overwhelming to train.

The task is to do binary classification, so the output layer of DeeppFM must be processed by sigmoid function and using cross entropy as loss function to do back propagation. After getting the output probability, we use 0.5 as the threshold to determine whether a record has potential to make fraud.

The result shows that this model could achieve credible results as the other models, but the AUC cannot reach the same level as LGBM's, Thus, this model might not be used as the final model to deal with this problem.

7. Conclusion

Credit risk-related research is vital for guiding new researchers and practitioners who want to improve their credit risk management practices. For this reason, algorithms integrating various criteria and models have been developed to predict risk-based credit scores.

In our project, we used three machine learning models, namely Logistic Regression, Random Forest, LightGBM, and one deep learning model named DeepFM to do the binary classification task for credit risk assessment and achieved 0.82's recall rate on the test data, which is good considering the highly unbalanced data and high dimension challenges we faced. And the features we selected can be used as new determinants of creditworthiness. From the resulting feature importance of our models, we found that among the most important features, the more the applicant earned, the longer the applicant have registered, the more document the applicant provides, the lesser likely will applicant default the repayment.

However, there're some limitation for our work. Firstly, we didn't utilise the ability of model fully. Now we conducted the same feature engineering for all the models, such as standardization and normalization for numerical data and one-hot encoding for categorical data. But, some models like DeepFM, LightGBM can handle categorical data. Handling categorical data using the natural support of those model may lead to better performance.

There're other topics in credit risk assessment which we can explore in the future. According to the findings of the most studied topics in credit risk-related research are credit risk score. Moreover, we discovered that the most common objective of the papers related to credit scoring was the suggestion of new techniques, like utilising models in other area in this field. Further, the regulations renewed to overcome the risk management difficulties in the digital era have increased the relevant research output in the field.

8. Reference

[1] Yıldız, İ. (2021). Credit risk estimation with machine learning and artificial neural networks algorithms.
[2] Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247.
[3] Chen, X., Ding, X. & Wang, B. Research on overdue behavior of folk board: An empirical analysis based on P2P network borrowing. Financ. Forum China 65–72 (2013).
[4] Xu, Lu, Z., & Xie, Y. (2021). Loan default prediction of Chinese P2P market: a machine learning methodology. Scientific Reports, 11(1), 18759–18759. https://doi.org/10.1038/s41598-021-98361-6
[5] Çallı, B. A., & Coşkun, E. (2021). A Longitudinal Systematic Review of Credit Risk Assessment and Credit Default Predictors. SAGE Open. https://doi.org/10.1177/21582440211061333