MSBA 6420 Predictive Analytics
Home Credit Default Risk Project

Laura Cattaneo, Callie Page, Max Traub,
Cathy Zheng

Project Definition & Introduction

Home Credit, an international consumer finance provider, helps people with limited credit history get loans. Since the company tailors to a non traditional market, they use alternative data to predict repayment ability. This includes telco and transactional information, as well as previous application data.
While they already use machine learning models to make predictions, they hope to improve their algorithms. We believe the best way to do so is to implement a light gradient boosting model, after conducting feature selection and engineering on the 220 features originally provided. Our model will help Home Credit meet their goal to provide a positive loan experience to typically underserved customers who are often abused by other, untrustworthy, lenders.

Assumption

- **Data consistency:** We assume that the data provided is an accurate subset of all available and future data. If this assumption does not hold, our results will not be generalizable.

Technical and Dataset Details

Software and Tools

Python 3.7.9	numpy 1.19.2
	pandas 0.25.3
	sklearn 0.22.2
	lightgbm 3.2.0

Dataset Introduction

To create our model, we used eight datasets introduced below.
We first clean each dataset and then merge them using the loan ID.

1. The **application** datasets, which are split into a training and a testing dataset, include customer details. This includes gender, income, family size and status, age, whether they own a car, details about their work, details about the current building they inhabit, contact information, and normalized credit scores from external sources.
2. The **bureau** dataset connects to the applications datasets through the loan ID. The dataset includes information from the credit bureau, including the status of the credit bureau credit, the currency it is in, the duration of the credit, how often the credit was prolonged, the amount overdue, credit limit, and annuity of the credit, amongst others.

3. The **bureau balance** dataset contains 3 columns: the loan ID, the month of balance relative to the application data, and the status of the credit bureau loan during the month.
4. The **POS cash balance** dataset contains columns regarding previous credit, including instalments left to pay, days past due during the previous month of credit, and the contract status during the month.
5. The **credit card balance** dataset has many columns that describe the applicant's previous credit history by month. This includes how much the client paid, various columns regarding how many drawings they had, including at an ATM, and other columns regarding the days past due on a payment.
6. We also use a **previous application** dataset that includes information about prior loan contracts. This dataset has 38 columns detailing previous annuity, down payment, interest rate, purpose, contract status, industry of the seller, payment method, and term.
7. The final dataset we use is the **instalments payments** table which includes information on when the instalment was paid (if so), when it was supposed to be paid, the prescribed instalment amount, and what the client actually paid.

Data Examination

Before beginning the pre-processing steps discussed below, we examine the available data more closely. We find that while there are no direct ‘nan’ missing values, they are identified with intentional outlier numbers that are consistent throughout the dataset (i.e. 365243). We replace these values, or drop them if a reasonable replacement is not available.

We also consider the balance of the data. We want to understand how the available data is split by class. We find that the majority of the data has a label of 0, or that a loan was given. We find that only 8 percent of our data has a label of 1, where a loan is not given. This level of imbalance is classified as ‘moderate’. To combat it, we will use evaluation metrics like recall, and AUC, instead of accuracy to evaluate our results. We also tune the hyperparameters associated with class weights in each model, presented below.

Data Preparation

In order to create the best predicting model, we conducted pre-processing steps discussed below.

Feature Engineering and Data Cleaning

We transformed a variety of given columns into features that better represent the process in order to increase the accuracy of our predictive model. We present these features below and, for convenience and reproducibility, split them by their respective datasets (in the same order as they are presented in the dataset introduction section). Unless specified, the variables were grouped by the current sk id and aggregated by calculating the mean.

Application Dataset:
For feature engineering purposes, the train and test datasets were temporarily combined.

- Age Int: Days birth variable divided by -365.

The following features relate to credit and its relation with customer financial standings.

- Credit Annuity Ratio: Credit amount divided by annuity amount.
- Credit Goods Price Ratio: Credit amount divided by goods price amount.
- Credit Down Payment: Goods price amount minus credit amount.

This feature is based on the assumption that older people and those who have a solid work history are more secure.

- Current Employment to Age: Days employed divided by days birth.

A lower annuity and a higher income would indicate a more secure lending situation.

- Annuity to Income Ratio: Annuity amount divided by total income amount plus one (to avoid errors where income is not present).

The following features rely on the assumption that large purchases can reduce financial security but age and length of employment might reduce that risk.

- New Car to Birth Ratio: Own car age divided by days birth for all non zero days birth values.
- New Car to Days Employed Ratio: Own car age divided by days employed for all non zero days employed.
- New Phone to Birth Ratio: Days last phone change divided by days birth for all non zero days birth values.
- New Phone to Days Employed Ratio: Days last phone change divided by days employed for all non zero days employed.

The following feature explores the relationship between the newly adopted credit and current income.

- New Credit to Income Ratio: Amount credit divided by amount income total for all non zero values of amount income total.

Unnamed external data sources are also provided by Home Credit. They are aggregated and included in our model.

- External Mean: The mean value of external sources one through three.
- External Standard Deviation: The standard deviation of external sources one through three.
- External Product: The product of external sources one through three.

The Following are simple numeric adjustments to meet the scikit learn requirements.

- Gender Code: Replace ‘M’ with 0 and ‘F’ with 1.
- Flag Own Car: Replace ‘Y’ with 1 and ‘N’ with 0.
- Flag Own Realty: Replace ‘Y’ with 1 and ‘N’ with 0.
- Emergency State Mode: ‘Yes’ with 1 and ‘No’ with 0.
- Name Education Type: Replace 'Lower secondary' with 0, 'Secondary / secondary special' with 1, 'Incomplete higher' with 2, 'Higher education' with 3, and 'Academic degree' with 4.
- One hot encode the following columns: fondkapremont_mode, housetype_mode, name_contract_type, name_family_status, name_housing_type, name_income_type,

name_type_suite, occupation_type, organization_type, wallsmaterial_mode, weekday_appr_process_start.

Bureau Dataset:
Due to the historic nature of the data set, and our interest in a client's previous credit history, each of the following variables were grouped by id and then summed:

- Credit Days Overdue, Days Credit Enddate, Credit Max Overdue Amount, Credit Sum Amount, Credit Sum Debt Amount, Credit Sum Limit Amount, Credit Sum Overdue Amount, Credit Type, Days Credit Update, Annuity Amount.

The following features follow the assumption that long existing debt and recent debt both play a role in current financial stability.

- Days Credit Oldest Account: Grouped by current sk id then took the minimum of days credit column.
- Days Credit Newest Account: Grouped by current sk id then took the maximum days credit column.

The following features look at the quantity of outstanding loans.

- Number Credit Accounts: Grouped by current sk id then counted the sk id bureau column.
- Number of Credit Accounts in Each Status: Grouped by current sk id and status of account then counted the sk id bureau column and pivoted so each status is its own column.

We use one hot encoding for model preparation on the following feature.

- Credit Type: One hot encoded the credit type column, aggregated by current sk id and then calculated the column sums.

Bureau Balance Dataset:
Home Credit also has access to monthly data on clients' payment trends. This is important to include in order to get a better understanding of each client over time. We therefore adapt the following columns in the described way.

- Months Balance: Grouped by bureau sk id, and took the count. Then grouped by the current sk id and calculated the mean, minimum, and maximum.
- Status: One hot encoded status column, then grouped by bureau sk id and took the sum of each and divided by months balance. Finally, took these columns, grouped by current sk id and calculated the mean, minimum and maximum.

POS Cash Balance Dataset:
The following features are based on the assumption that history and cash loan amount play a role in a client's ability to repay on time.

- Months Balance: We calculated the size, maximum, and mean for each grouped current sk id.
- POS Count: We calculated the size for each grouped current sk id.

The following features look into the average and worst late payment values per customer.

- Days Past Due: We calculated the maximum and mean for each grouped current sk id.
- Days Past Due with Tolerance: We calculated the maximum and mean for each grouped current sk id.

Credit Card Balance Dataset:
The following feature was created under the assumption that credit usage history is important to financial stability.

- Credit Card Count: We calculated the size for each grouped sk id.

The remaining features had various aggregations performed as we believed them important but did not have a specific assumption in mind.

- All other variables were aggregated by current sk id. Each had their respective mean, minimum, maximum, sum, and variance calculated.

Previous Application Dataset:
The features in this data set were aggregated as a whole as well as subset based on application approval. The feature below was created based on the assumption that the ratio between the application and approved credit amount can help indicate the clients financial health.

- Application Credit Percentage: For all values with a credit greater than zero, we divided the application amount by the credit amount and then calculated the minimum, maximum, mean, and variance for each grouped current sk id.

The following features had basic aggregations performed as we believed them important but had no specific assumptions in mind.

- For the following variables, the minimum, maximum, and mean amount for each grouped current id were calculated: Annuity Amount, Application Amount, Credit Amount, Down Payment Amount, Goods Price Amount, Hour Approval Process Start, Down Payment Rate, Decision time in days.

This feature is separate since we did not see much reason in finding the minimum or maximum amount. However, we did think the sum and mean of previous credit terms could help indicate financial stability.

- Payment Count: We calculated the mean and summed amount for each grouped current sk id.

Instalments Payments Dataset:
The following two features relate to how extended the client is on their current payments.

- Payment Percentage: We calculated the payment amount divided by instalment amount for all instalments greater than zero and then calculated the minimum, maximum, mean, and variance for each grouped current sk id.
- Payment Difference: We calculated the instalment amount minus payment amount and then the minimum, maximum, mean, and variance for each grouped current sk id.

The following two features relate to previous instalment payment punctuality.

- Days Past Due: We calculated the days entry payment minus instalment days, kept only positive values, and then calculated the maximum, mean, and sum for each grouped current sk id.
- Days Before Due: We calculated the instalment days minus days entry payment, kept only positive values, and then calculated the maximum, mean, and sum for each grouped current sk id.

This feature is based on the assumption that the amount of previous instalment records with Home Credit can help indicate financial health.

- Instalment Count: We calculated the size for each grouped current sk id.

The remaining features had basic aggregations performed as we believed them important but had no specific assumptions in mind.

- For the following variables we calculated the maximum, mean, and sum for each grouped current sk id: Instalment Amount, Payment Amount, Days Entry Payment

Feature Selection

Initially, Principal Component Analysis (PCA) reduction was performed, selecting the top 70% of explained variation, in order to speed up the model selection process. However after comparison, we found that our star model, Light GBM, performed better with the full set of features. For consistency, we present the evaluation results of all models selected on the full set of features.

Modeling

Model Selection

To select a model, we explored five different algorithms, namely Decision Tree, K-NN, Logistic Regression, SVM, and LightGBM. We use a small sample (5%) of the original training dataset and conduct a nested cross validation in order to select the optimal hyperparameters that tune each model. The hyperparameters that differ from the default are shown below.

	Best Parameters
Decision Tree	{'max_depth': 11, 'criterion': 'gini'}
k-NN	{'weights': 'distance', 'n_neighbors': 51}
Logistic Regression	{'C': 0.1}
SVM	{'kernel': 'rbf', 'gamma': 0.001, 'C': 10}
LightGBM	{'scale_pos_weight': 1, 'num_leaves': 100, 'n_estimators': 100, 'min_child_samples': 850, 'boosting_type': gbdt}

After selecting the optimal hyperparameters, we evaluate each of the five algorithm's average performance on the subset using randomized search. The metrics explored are AUC (area under the ROC curve) and recall.

	Area under ROC Curve (AUC)	**Recall**
Decision Tree	0.559	0.125
k-NN	0.617	0.062
Logistic Regression	0.682	0.002
SVM	0.554	0.052
LightGBM	0.664	0.452

While logistic regression has a marginally higher AUC score, we ultimately elect to pursue the light GBM model due its significantly better recall performance. We focus on recall since false negatives would be much more costly in our case than false positives, as discussed in the cost analysis section below.

Additionally, we target the light GBM model, released by Microsoft in early 2017, because it is one of the most powerful boosting models currently available. It is an improved boosting model with faster training speeds, and higher efficiency that is more suitable for large datasets. Light GBM uses the gradient boosting framework, with a decision tree base. It uses lower memory than other boosting models because it replaces values to discrete bins. Further, since it produces more complex trees it has higher predictive accuracy than other models.

Cost Analysis

It's imperative that the model performs well on the test data, but model performance also needs to translate to business value. In the case of mortgage contracts, approving a mortgage for someone who will be unable to pay has a high cost to the business when the mortgage defaults. Based on a study by Laurie Goodman and Jun Zhu from The Urban Institute[1], the average loss due to mortgage default in 2011-2013 was $26,895. On the other hand, the lending company will make $4,548 on each mortgage that is successfully paid off according to 2020 data published by the Mortgage Bankers Association[2]. Of course, there is no loss or gain if an applicant who can not pay is denied, but denying an applicant who is able to pay can hurt the company reputation if they complain online or to their friends and family. We estimate this loss at $1,000 if the unhappy customer dissuades 1 in 4 people they tell.

[1] Goodman, L., & Zhu, J. (2015, February). *Loss Severity on Residential Mortgages* [PDF]. Washington, DC: Urban Institute.

[2] Olick, D. (2020, September 10). Mortgage lenders just saw record profit, and expect to do better in the next quarter. Retrieved from https://www.cnbc.com/2020/09/10/mortgage-lenders-just-saw-record-profit-and-expect-to-do-better-in-the-next-quarter.html

In addition to estimating the cost and benefit of each potential outcome when using the model, we must also consider the likelihood of each outcome based on the normalized confusion matrix calculated on our validation dataset. The likelihood of approving a mortgage that defaults in only 7.7%, while the likelihood of approving a mortgage that is paid back is 91.6%. The likelihood of denying a customer who can't pay is 0.4% and the likelihood of denying someone who can pay is 0.3%.

We can now determine the total expected value per customer of using the model by multiplying the cost or benefit of the outcome with the likelihood and totaling them up.

Outcome	Cost/Benefit	Likelihood	Expected Value
Approved mortgage to applicant unable to pay (default)	-$26,895	7.7%	$2,078.82
Approved mortgage to applicant able to pay	$4,548	91.6%	
Denied mortgage to applicant unable to pay	$0	0.4%	
Denied mortgage to applicant able to pay	-$1,00	0.3%	

We can compare the results of our model to the outcome of approving every applicant without considering their ability to pay. The expected value in this scenario is show below:

Outcome	Cost/Benefit	Likelihood	Expected Value
Approved mortgage to applicant unable to pay (default)	-$26,895	8.1%	$1,990.75
Approved mortgage to applicant able to pay	$4,548	91.9%	

Our model is associated with a 4.4% increase in profit. If Home Credit works with 100,000 applicants per year, the additional profit would amount to $8,807,497.78 in that year.

Final Model Results

As previously mentioned, we choose the light GBM model for two reasons. First, it performs best on our training data when considering both AUC and recall, and second, it is one of the most powerful algorithms currently available.

Hyperparameters

To identify the hyperparameters that best tune the model, we conduct a randomized search with one hundred iterations on the following parameter grid:

```
lgbm_grid = {'boosting_type': ['gbdt', 'dart', 'goss'],
             'num_leaves': range(20, 200, 20),
             'min_child_samples': range(100, 1000, 100),
             'min_child_weight': [0.001,0.01, 0.1, 1, 10],
             'scale_pos_weight': range(1, 15, 3),
             'learning_rate': [0.001, 0.01, 0.1],
             'n_estimators': [1000]}
```

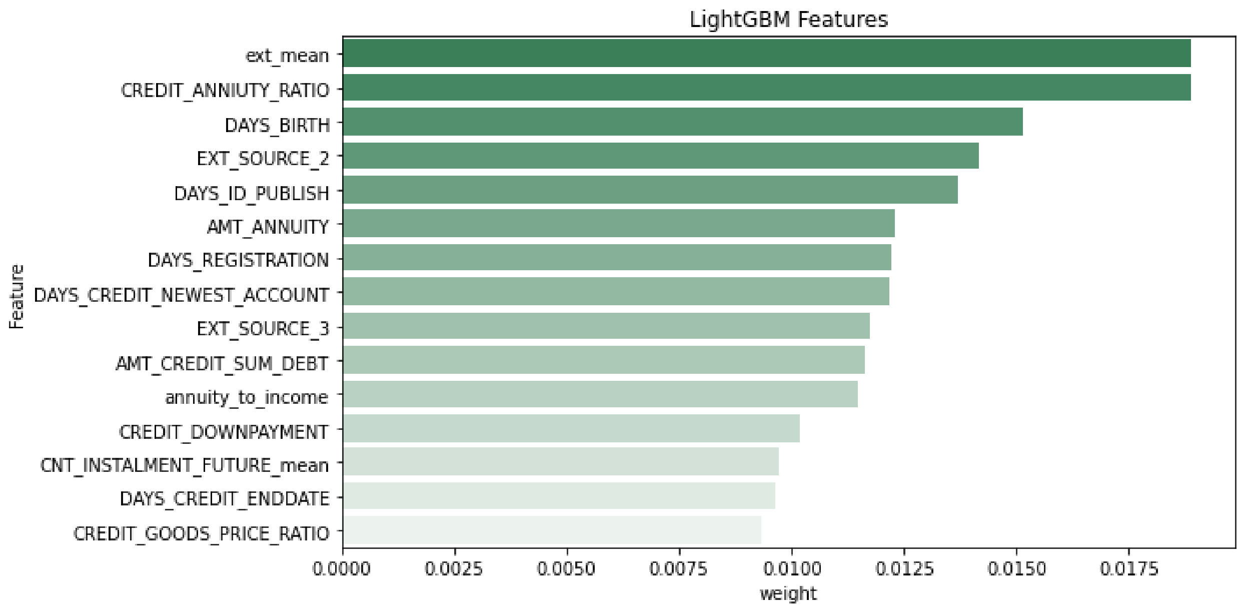
The search yields the model with the highest area under the ROC curve. We then manually explored different values for the number of estimators to include, as we were limited by our computational resources to explore it in our randomized search. After trying different numbers of estimators, we found that including 1500 estimators gives the best performance for both training and testing data.
The parameters for our final model are as follows:

- ➔ boosting_type='goss'
    - ◆ We use the Gradient based One-Side Sampling type because the nested cross validation results indicate it was the best for our problem.
- ➔ Colsample_bytree = 1.0
    - ◆ We use the default. This hyperparameter selects the subsample ratio of columns when constructing each tree.
- ➔ importance_type='split'
    - ◆ We use the default. This hyperparameter is for feature importance. It tells the algorithm to keep the result that contains the numbers of times the feature is used in the model.
- ➔ learning_rate=0.01
    - ◆ We change the default (0.1) by a factor of 10. We lower the learning rate for a more accurate result.
- ➔ Max_depth = -1
    - ◆ We use the default. A depth for base learners of less than 0 means there is no depth limit.
- ➔ min_child_samples=500
    - ◆ This hyperparameter sets the data points that each leaf must have.
- ➔ Min_child_weight = 10
    - ◆ We set the minimum sum of instance weight needed in a child (left) to 10.
- ➔ min_split_gain=0.0
    - ◆ We use the default. This hyperparameter sets the minimum loss reduction required to make a further partition on a leaf node of the tree.
- ➔ n_estimators=1500
    - ◆ We set the number of boosted trees to fit to 1500.
- ➔ n_jobs=-1
    - ◆ We use the default. This hyperparameter sets the number of parallel threads.
- ➔ num_leaves=140
    - ◆ We set the maximum tree leaves for base learners to 140.
- ➔ objective=None
    - ◆ We use the default. This parameter allows the developer to specify the learning task.
- ➔ random_state=None
    - ◆ We use the default. We do not set a random number seed.

- ➔ reg_alpha=0.0
    - ◆ We use the default. We do not set an L1 regularization term on weights.
- ➔ reg_lambda=0.0
    - ◆ We use the default. We do not set an L2 regularization term on weights.
- ➔ silent=True
    - ◆ We use the default. This does not print the messages while running boosting.
- ➔ subsample=1.0
    - ◆ We use the default. This hyperparameter sets the subsample ratio of the training instance.
- ➔ subsample_for_bin=200000
    - ◆ We use the default. This hyperparameter sets the number of samples for constructing bins.
- ➔ subsample_freq=0
    - ◆ We use the default. A value of 0 means the frequency is not enabled.

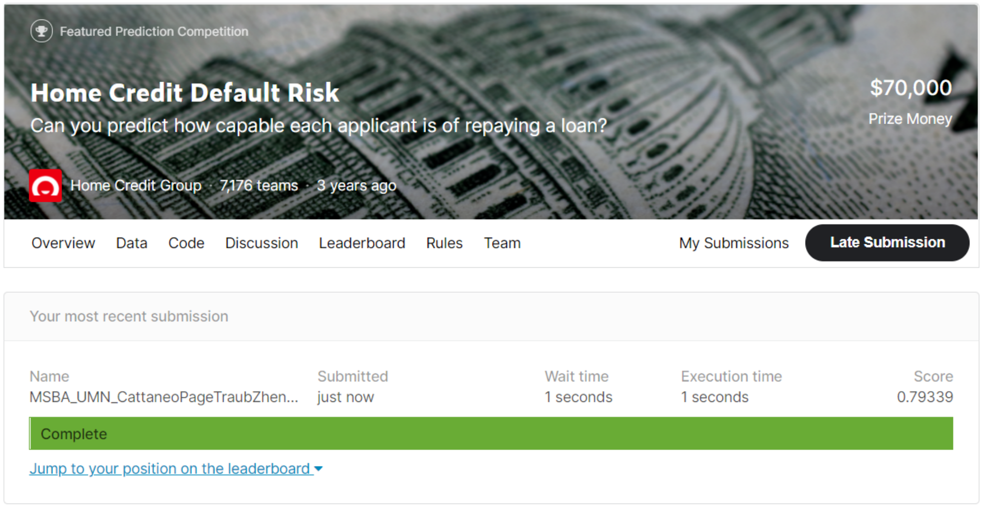
Feature Importance

The model uses all 1,057 features in the dataset, but some are more valuable than others in predicting the correct outcome. If all of the features were weighted equally, they would have a weight value of .001 or .1%. The top 15 features shown here account for more than 20% of the weight, which is significantly more than the 1.9% that would be expected This indicates that the presented features are very valuable to the model.



Conclusion

Home Credit, an international consumer finance provider, helps people with limited credit history get loans. Since the company tailors to a non-traditional market, they use alternative data to predict repayment ability. While Home Credit already uses machine learning models to make predictions, we created a new predictive model that can increase their profit by 4.4 percent, which translates to an $8.8 million gain per year for the company. We used the 8 provided datasets with 220 features, and engineered new ones to be included in 5 models we tested. The model that gave us the best performance is a boosting model, light GBM. Although we tried to train the model after implementing a Principal Component Analysis, we found that the best performance (a new MSBA record) was with all 1057 features. In addition to greatly increasing Home Credit's profit, we helped them meet their goal to provide a positive loan experience to typically underserved customers who are often abused by other, untrustworthy, lenders.


Featured Prediction Competition
Home Credit Default Risk
Can you predict how capable each applicant is of repaying a loan?
$70,000
Prize Money
Home Credit Group · 7,176 teams · 3 years ago
Overview
Data
Code
Discussion
Leaderboard
Rules
Team
My Submissions
Late Submission
Your most recent submission
Complete
Jump to your position on the leaderboard

Name	Submitted	Wait time	Execution time	Score
MSBA_UMN_CattaneoPageTraubZhen...	just now	1 seconds	1 seconds	0.79339

Appendix

See code in GitHub and submitted on Canvas.