Capstone Project Proposal

Udacity Machine Learning Nanodegree

James Dellinger

June 22, 2018

1 Domain Background

For my capstone project I will participate in the Home Credit Default Risk competition on Kaggle[Kagb]. The goal of the competition is to be able to predict the likelihood that an applicant will experience difficulty in repaying their loan. The competition is sponsored by Home Credit[Gro], whose mission is to provide a positive and safe borrowing experience to groups of people that traditional, mainstream banks and financial institutions typically refuse to serve. The competition's dataset is available for download directly from 'Data' section of its Kaggle webpage[Kaga]. There has been extensive research in using machine learning algorithms to predict loan repayment outcomes, such as a 2017 paper published by Xiaojiao Yu that explores using an XGBoost model to predict online lending risk: Machine learning application in online lending risk prediction[Yu17].

Home Credit targets a demographic that typically has no recourse but to deal with shady characters such as loan sharks when borrowing money. Many of these unbanked individuals are hard-working, well-intentioned folks who, either due to circumstances beyond their control or past mistakes, have fallen through the financial system's cracks.

I want to live in a world where a second chance is available to anyone who would make a good faith effort to make the most of it. This is why Home Credit's goal of providing a safe lending alternative to otherwise financially down-and-out folks resonates with me. I want to help Home Credit have the ability to expand its services to as many deserving applicants as possible.

2 Problem Statement

Home Credit needs an algorithm that will take as inputs various personal and alternative financial information originally taken from a loan applicant's profile, and then determine a probability of the applicant having at least one late payment when repaying their loan. This probability will be in the range [0.0, 1.0], where 1.0 represents a 100% certainty that the applicant will have at least one delinquent repayment and 0.0 indicates that there is

zero chance that the applicant will ever be delinquent. The algorithm will be tested on a set of 48,744 individuals who previously borrowed from Home Credit. A CSV file must be produced that contains one header row, and 48,744 prediction rows, where each prediction row contains both a user ID, the `SKI_ID_CURR` column, and the probability, the `TARGET` column, of that user being delinquent. The file must be formatted as follows:

```
SK_ID_CURR,TARGET
100001,0.1
100005,0.9
100013,0.2
etc.
```

Home Credit knows which borrowers ultimately paid off their loans, and which ones had one or more late payments. A good algorithm will need to predict a high probability of delinquency for the majority of borrowers who did actually make one or more late payments. This algorithm will also need to predict a low probability of delinquency for the majority of borrowers who eventually did successfully repay their loans with no late payments.

3 Datasets and Inputs

This competition's dataset was provided by Home Credit Group's data scientists. It contains a wide variety of personal and financial information belonging to 356,255 individuals who had previously been recipients of loans from Home Credit. These individuals are divided into training and testing sets. The training group contains 307,511 individuals' records. The test group contains 48,744 records. The dataset is anonymized, with each individual represented by their loan ID. Any personally indentifying infomation, such as name, phone number, or address, has been omitted.

Because Home Credit targets the unbanked population, it is unable to rely on traditional measures, such as a credit score, that mainstream financial institutions use when making lending decisions. Home Credit works around this obstacle by looking at an extensive and diverse array of personal and financial information for each of its applicants. These features range from common characteristics, such as marital status, age, type of housing, region of residence, job type, and education level, to some incredibly niche characteristics, such as how many elevators are in an applicant's apartment building. Home Credit also looks at nitty gritty aspects of applicants' financial backgrounds, including month-by-month payment performance on any loans or credit card balances that the applicant has previously had with Home Credit, as well as the amount and monthly repayment balances of any loans that the applicant may have received from other lenders.

All of these features are spread across seven data tables. The main data table (application_{train | test}.csv) contains 120 features that comprise applicants' personal background information. The other six data tables contain applicants' previous loan and credit card balance payment histories. Detailed descriptions of each data table can be found in Appendix A. Explanations of all features in each data table are in Appendix B. The following diagram

provides a brief summary:

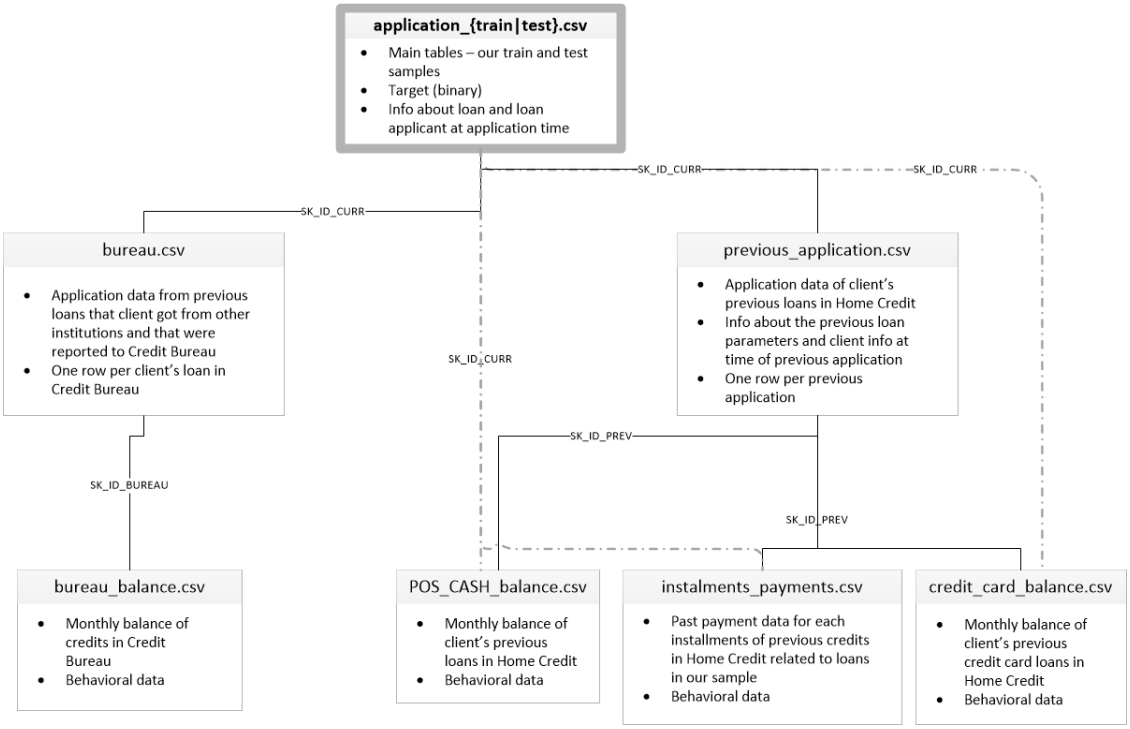

Figure 1: Home Credit Data Table Relationships[Kaga]

4 Solution Statement

The solution to this competition will be an algorithm that computes delinquency probabilities (in the range [0.0,1.0]) for each of the 48,744 Home Credit borrowers in the dataset's test segment. These predictions will be stored in a CSV file that adheres to the format described in the Problem Statement above in Section 2. This algorithm will take as inputs features from the seven data tables outlined above. It will be trained on the 307,511 Home Credit borrower records that comprise the training segment. The winning algorithm will predict the highest probabilities of delinquency for the borrowers who had at least one late payment, and will predict the lowest probabilities of delinquency for the borrowers who ultimately repaid their loans with no late payments. More formally, a winning algorithm will have the highest true positive rate, or sensitivity, (correctly guessing who will be delinuent), while at the same time having a minimum false positive rate (as seldom as possible making an incorrect guess that an individual will have difficulty paying back their loan).

5 Benchmark Model

Solutions to this competition will be ranked on Kaggle by the area under the ROC curve between their predicted proabilities and the observed targets (whether or not the borrower was actually delinquent).

I will train a Gaussian Naive Bayes classifier on a fully preprocessed dataset and use its ROC area under curve score as my primary benchmark. Gaussian Naive Bayes is ideal for this because it needs no hyperparameter input nor tuning. It also trains faster than most other models and I expect I should have no difficulty training it on the dataset's 300,000+ rows locally on my own machine. I am optimistic that with appropriate model selection and hyperparameter tuning I will eventually be able to build a learning algorithm that outperforms this out-of-the-box Gaussian Naive Bayes model.

In order to guage how my solution compares to others in the Kaggle community, I will use the average ROC curve area of all entries on the competition's public leaderboard[Kagc] as of June 2, 2018 as a secondary, personal benchmark. As of this date, there were 5,398 total entries from 1,515 participants. The average area under the ROC curves of all entries was `0.7379`. I hope to design an algorithm that will meet or exceed this score.

It would also be nice to know the area under ROC curve of the prediction model employed by Home Credit's data scientists prior to this competition's debut on Kaggle. However, since Home Credit possibly considers this model and its ROC curve area to be proprietary information, it is understandable that it was not disclosed.

6 Evaluation Metric

The area under the ROC (receiver operating characteristic) curve[Wikb] will be the evaluation metric for my solution to this competition. This area can range from a minimum value of 0, or 0%, to a maximum of 1, or 100%. The size of the area under an ROC curve indicates how good a job a classifier does of identifying a certain target segment in a dataset. An area of 1 indicates that the classifier is perfect – that it can find every true positive (exhibiting perfect sensitivity, or recall) without making any mistakes (not accidentally labelling some true negatives as false positives), thus also giving it perfect specificity, which leads to a false positive rate (1 - specificity), of zero. An area of 0 indicates that the classifier isn't able to find and properly label any of the true positives. An area of 0.5 is the level of performance we'd expect from a classifier that randomly labels each point in the dataset – on the whole, this classifier would make just as many mistakes (false positives) as it makes correct predictions (true positives).

Because the goal of this competition is to create an algorithm that correctly identifies applicants who were delinquent in repaying their loans, area under the ROC curve is a perfectly appropriate measure of my algorithm's performance. While my target benchmark is the score of `0.7379` that I mentioned above, at a bare minimum, my classifier must achieve a score of better than 0.5 – anything less would mean that my classifier performs only as good as, or worse than, a random classifier.

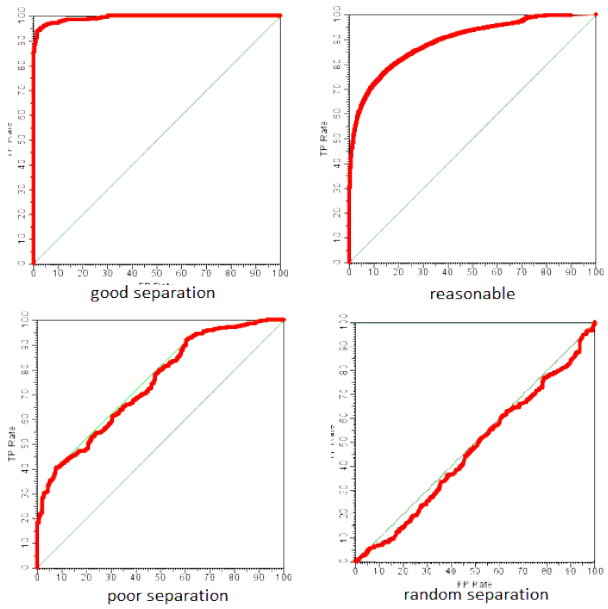

Figure 2: ROC Curve Examples[Wika]

7 Project Design

Although Home Credit provides seven data tables for this competition, for the purposes of this project I will focus most of my attention on the main data table. However, at a minimum, I will engineer at least one feature from the remaining six data tables. I may dive deeper into these other six data tables if time permits. I will structure my approach to creating a high-performing classification algorithm according to the steps in the following provisional workflow:

I **Data Exploration**

(a) Compile lists and short descriptions of each feature in all seven of the competition's data tables.

(b) Display number of entries (rows) in each data table.

(c) Display number of features in each data table.

(d) Generate and display 5-row samples of each data table.

(e) Generate and display statistical descriptions of the numerical features of each data table that will be used by my algorithm.

(f) Investigate whether any features in any of the data tables have 'NaN' entries that will need to be handled, and whether some features have sparse enough data such that the entire feature should be removed.

(g) Also investigate whether any borrowers (rows) in the main data table have 'NaN' for the majority of their feature entries. Some of these datapoints may also need to be removed.

(h) Review the statistical descriptions of each numerical feature, and ensure that there are no other unexpected numerical entries in these features, eg. a feature having some negative numbers among its entries, where only positive numbers would be expected based on how the feature was defined by Home Credit.

(i) List the numerical features in the main data table that are not already identified as being normally distributed (they may eventually need to be log-normalized).

(j) List the categorical features in the main data table that will need to be one-hot encoded.

(k) List the categorical features in the data table that are already one-hot encoded.

(l) Define at least one feature that could be engineered from the data contained in the other six data tables outside of the main data table.

II Exploratory Visualization

(a) Plot histograms of the numerical features in the main data table, in order to confirm which ones are indeed already normally distributed.

(b) Plot histograms of the numerical features in the main data table that are supposedly not already normally distributed.

(c) Identify any features containing outliers that will need to be addressed.

III Algorithms and Techniques to Consider

(a) **Dimensionality Reduction**:

i. Feature transformation and dimensionality reduction using Principle Component Analysis (PCA)

ii. Dimensionality reduction using Sparse Random Projection (in case using PCA is too taxing on my compute resources)

(b) **Learning Algorithms**:

Any algorithm that I use will need to not only be able to simply segment the data points into the "delinquent repayers" and "on-time repayers" categories, but must also be able to compute the probability that each data point in the test set belongs to the "delinquent repayers" segment. These probabilities will hopefully be higher for points that the algorithm labels as "delinquent repayers," and will be lower for the points that are labeled as "on-time repayers." This requirement automatically eliminates algorithms such as SVM, K-Means, and Hierarchical Clustering from consideration.

Below are several algorithms that meet the above criteria:

i. Classifier Algorithms:
   - Multi-layer Perceptron Classifier
   - Logistic Regression Classifier
   - Gaussian Naive Bayes Classifier

ii. Ensemble Methods:
   - AdaBoost Classifier
   - Bagging Classifier
   - Extra Trees Classifier
   - Gradient Boosting Classifier

(c) **Other Techniques**:

i. Gridsearch CV for hyperparameter tuning.

ii. Breaking the training set up into an 80%-20% train-test split, or use K-Fold Cross-Validation, for model selection.

iii. SelectKBest for feature selection.

IV Data Preprocessing

(a) Separate out training and testing target data into separate variables. Drop the targets column from both training and testing sets of the main data table.

(b) Apply one-hot encoding to all non-numerical features in the main data table.

(c) Apply non-linear feature scaling using the natural logarithm to any numerical features in the main data table that aren't already normally distributed.

(d) Scale all numerical features are to the range [0.0,1.0].

(e) Impute 'NaN' values for numerical features.

(f) Replace 'NaN' with 0 for all binary categorical features.

(g) **Engineer** at least one feature based on some of the data contained in the six other data tables and append it to the main data table.

(h) Build a preprocessing pipeline that can be used to apply the above steps in order to preprocess test sets before a classifier makes predictions on them.

V Implementation

(a) **First and foremost**, it is absolutely necessary to set aside a validation segment from the training set in order to compare the different algorithms I attempt and to avoid overfitting. This should be about 20% of the size of the original training set. Alternatively, K-Fold Cross-Validation could also be used.

(b) Run PCA or Sparse Random Projection on the features in the main data table.

(c) Plot the feature weights and display the explained variances of the dimensions outputted by the chosen dimensionality reduction algorithm, in order to get an idea of how many dimensions to use when classifying the data points.

(d) Create different classifiers, using some or all of the different classifier algorithms listed above.

(e) Create a training and prediction pipeline that outputs probability predictions for each data point in test segment to a CSV file, and also calculates and outputs the area under the ROC curve of the predictions.

VI **Refinement**

(a) Experiment with adjusting hyperparameters on the best performing classifier.

(b) Try feature selection methods such as SelectKBest.

VII **Model Evaluation and Validation**

(a) Generate a table that ranks the test set scores (area under ROC curve) of all the different algorithms and featuresets that I experiment with.

(b) Plot ROC curves of the results from various classifier algorithms

(c) Choose the algorithm/featureset combo that had the highest score on the testing set.

(d) Conduct sensitivity analysis by manipulating some of the training data in order to ascertain the solution's robustness.

(e) Save the testing set predictions to a CSV file, and submit on Kaggle.

(f) Compare my final algorithm's score to the benchmark score, and observe my submission's public rank on Kaggle.

Appendices

A Data Table Descriptions

From the data description page on the Home Credit Kaggle competition's website[Kaga].

1 **application_{train | test}.csv**

- This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET).
- Static data for all applications. One row represents one loan in our data sample.

2 **bureau.csv**

- All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in our sample).

- For every loan in our sample, there are as many rows as number of credits the client had in Credit Bureau before the application date.

3 **bureau_balance.csv**

- Monthly balances of previous credits in Credit Bureau.
- This table has one row for each month of history of every previous credit reported to Credit Bureau – i.e the table has (#loans in sample # of relative previous credits # of months where we have some history observable for the previous credits) rows.

4 **previous_application.csv**

- All previous applications for Home Credit loans of clients who have loans in our sample.
- There is one row for each previous application related to loans in our data sample.

5 **POS_CASH_balance.csv**

- Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
- This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample # of relative previous credits # of months in which we have some history observable for the previous credits) rows.

6 **installments_payments.csv**

- Repayment history for the previously disbursed credits in Home Credit related to the loans in our sample.
- There is a) one row for every payment that was made plus b) one row each for missed payment.
- One row is equivalent to one payment of one installment OR one installment corresponding to one payment of one previous Home Credit credit related to loans in our sample.

7 **credit_card_balance.csv**

- Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
- This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample # of relative previous credit cards # of months where we have some history observable for the previous credit card) rows.

B Data Table Features

From the HomeCredit_columns_description.csv file on the data description page on the Home Credit Kaggle competition's website[Kaga].

B.1 Main Data Table Features (application_{train | test}.csv)

1 **SK_ID_CURR**: ID of loan in our sample

2 **TARGET**: Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)

3 **NAME_CONTRACT_TYPE**: Identification if loan is cash or revolving

4 **CODE_GENDER**: Gender of the client

5 **FLAG_OWN_CAR**: Flag if the client owns a car

6 **FLAG_OWN_REALTY**: Flag if client owns a house or flat

7 **CNT_CHILDREN**: Number of children the client has

8 **AMT_INCOME_TOTAL**: Income of the client

9 **AMT_CREDIT**: Credit amount of the loan

10 **AMT_ANNUITY**: Loan annuity

11 **AMT_GOODS_PRICE**: For consumer loans it is the price of the goods for which the loan is given

12 **NAME_TYPE_SUITE**: Who was accompanying client when he was applying for the loan

13 **NAME_INCOME_TYPE**: Clients income type (businessman, working, maternity leave,Ö)

14 **NAME_EDUCATION_TYPE**: Level of highest education the client achieved

15 **NAME_FAMILY_STATUS**: Family status of the client

16 **NAME_HOUSING_TYPE**: What is the housing situation of the client (renting, living with parents, ...)

17 **REGION_POPULATION_RELATIVE**: Normalized population of region where client lives (higher number means the client lives in more populated region) – normalized

18 **DAYS_BIRTH**: Client's age in days at the time of application – time only relative to the application

19 **DAYS_EMPLOYED**: How many days before the application the person started current employment – time only relative to the application

20 **DAYS_REGISTRATION**: How many days before the application did client change his registration – time only relative to the application

21 **DAYS_ID_PUBLISH**: How many days before the application did client change the identity document with which he applied for the loan – time only relative to the application

22 **OWN_CAR_AGE**: Age of client's car

23 **FLAG_MOBIL**: Did client provide mobile phone (1=YES, 0=NO)

24 **FLAG_EMP_PHONE**: Did client provide work phone (1=YES, 0=NO)

25 **FLAG_WORK_PHONE**: Did client provide home phone (1=YES, 0=NO)

26 **FLAG_CONT_MOBILE**: Was mobile phone reachable (1=YES, 0=NO)

27 **FLAG_PHONE**: Did client provide home phone (1=YES, 0=NO)

28 **FLAG_EMAIL**: Did client provide email (1=YES, 0=NO)

29 **OCCUPATION_TYPE**: What kind of occupation does the client have

30 **CNT_FAM_MEMBERS**: How many family members does client have

31 **REGION_RATING_CLIENT**: Our rating of the region where client lives (1,2,3)

32 **REGION_RATING_CLIENT_W_CITY**: Our rating of the region where client lives with taking city into account (1,2,3)

33 **WEEKDAY_APPR_PROCESS_START**: On which day of the week did the client apply for the loan

34 **HOUR_APPR_PROCESS_START**: Approximately at what hour did the client apply for the loan rounded

35 **REG_REGION_NOT_LIVE_REGION**: Flag if client's permanent address does not match contact address (1=different, 0=same, at region level)

36 **REG_REGION_NOT_WORK_REGION**: Flag if client's permanent address does not match work address (1=different, 0=same, at region level)

37 **LIVE_REGION_NOT_WORK_REGION**: Flag if client's contact address does not match work address (1=different, 0=same, at region level)

38 **REG_CITY_NOT_LIVE_CITY**: Flag if client's permanent address does not match contact address (1=different, 0=same, at city level)

39 **REG_CITY_NOT_WORK_CITY**: Flag if client's permanent address does not match work address (1=different, 0=same, at city level)

40 **LIVE_CITY_NOT_WORK_CITY**: Flag if client's contact address does not match work address (1=different, 0=same, at city level)

41 **ORGANIZATION_TYPE**: Type of organization where client works

42 **EXT_SOURCE_1**: Normalized score from external data source – normalized

43 **EXT_SOURCE_2**: Normalized score from external data source – normalized

44 **EXT_SOURCE_3**: Normalized score from external data source – normalized

45 **APARTMENTS_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

46 **BASEMENTAREA_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

47 **YEARS_BEGINEXPLUATATION_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

48 **YEARS_BUILD_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

49 **COMMONAREA_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

50 **ELEVATORS_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

51 **ENTRANCES_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

52 **FLOORSMAX_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

53 **FLOORSMIN_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

54 **LANDAREA_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

55 **LIVINGAPARTMENTS_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

56 **LIVINGAREA_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

57 **NONLIVINGAPARTMENTS_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

58 **NONLIVINGAREA_AVG**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

59 **APARTMENTS_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

60 **BASEMENTAREA_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

61 **YEARS_BEGINEXPLUATATION_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

62 **YEARS_BUILD_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

63 **COMMONAREA_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

64 **ELEVATORS_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

65 **ENTRANCES_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

66 **FLOORSMAX_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

67 **FLOORSMIN_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

68 **LANDAREA_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

69 **LIVINGAPARTMENTS_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

70 **LIVINGAREA_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

71 **NONLIVINGAPARTMENTS_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

72 **NONLIVINGAREA_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

73 **APARTMENTS_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

74 **BASEMENTAREA_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

75 **YEARS_BEGINEXPLUATATION_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

76 **YEARS_BUILD_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

77 **COMMONAREA_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

78 **ELEVATORS_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

79 **ENTRANCES_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

80 **FLOORSMAX_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

81 **FLOORSMIN_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

82 **LANDAREA_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

83 **LIVINGAPARTMENTS_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

84 **LIVINGAREA_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

85 **NONLIVINGAPARTMENTS_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

86 **NONLIVINGAREA_MEDI**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

87 **FONDKAPREMONT_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

88 **HOUSETYPE_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

89 **TOTALAREA_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

90 **WALLSMATERIAL_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

91 **EMERGENCYSTATE_MODE**: Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor – normalized

92 **OBS_30_CNT_SOCIAL_CIRCLE**: How many observation of client's social surroundings with observable 30 DPD (days past due) default

93 **DEF_30_CNT_SOCIAL_CIRCLE**: How many observation of client's social surroundings defaulted on 30 DPD (days past due)

94 **OBS_60_CNT_SOCIAL_CIRCLE**: How many observation of client's social surroundings with observable 60 DPD (days past due) default

95 **DEF_60_CNT_SOCIAL_CIRCLE**: How many observation of client's social surroundings defaulted on 60 (days past due) DPD

96 **DAYS_LAST_PHONE_CHANGE**: How many days before application did client change phone

97 **FLAG_DOCUMENT_2**: Did client provide document 2

98 **FLAG_DOCUMENT_3**: Did client provide document 3

99 **FLAG_DOCUMENT_4**: Did client provide document 4

100 **FLAG_DOCUMENT_5**: Did client provide document 5

101 **FLAG_DOCUMENT_6**: Did client provide document 6

102 **FLAG_DOCUMENT_7**: Did client provide document 7

103 **FLAG_DOCUMENT_8**: Did client provide document 8

104 **FLAG_DOCUMENT_9**: Did client provide document 9

105 **FLAG_DOCUMENT_10**: Did client provide document 10

106 **FLAG_DOCUMENT_11**: Did client provide document 11

107 **FLAG_DOCUMENT_12**: Did client provide document 12

108 **FLAG_DOCUMENT_13**: Did client provide document 13

109 **FLAG_DOCUMENT_14**: Did client provide document 14

110 **FLAG_DOCUMENT_15**: Did client provide document 15

111 **FLAG_DOCUMENT_16**: Did client provide document 16

112 **FLAG_DOCUMENT_17**: Did client provide document 17

113 **FLAG_DOCUMENT_18**: Did client provide document 18

114 **FLAG_DOCUMENT_19**: Did client provide document 19

115 **FLAG_DOCUMENT_20**: Did client provide document 20

116 **FLAG_DOCUMENT_21**: Did client provide document 21

117 **AMT_REQ_CREDIT_BUREAU_HOUR**: Number of enquiries to Credit Bureau about the client one hour before application

118 **AMT_REQ_CREDIT_BUREAU_DAY**: Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application)

119 **AMT_REQ_CREDIT_BUREAU_WEEK**: Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application)

120 **AMT_REQ_CREDIT_BUREAU_MON**: Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application)

121 **AMT_REQ_CREDIT_BUREAU_QRT**: Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application)

122 **AMT_REQ_CREDIT_BUREAU_YEAR**: Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application)

B.2 Bureau Data Table Features (bureau.csv)

1 **SK_ID_CURR**: ID of loan in our sample - one loan in our sample can have 0,1,2 or more related previous credits in credit bureau – hashed

2 **SK_ID_BUREAU**: Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application) – hashed

3 **CREDIT_ACTIVE**: Status of the Credit Bureau (CB) reported credits

4 **CREDIT_CURRENCY**: Recoded currency of the Credit Bureau credit – recoded

5 **DAYS_CREDIT**: How many days before current application did client apply for Credit Bureau credit – time only relative to the application

6 **CREDIT_DAY_OVERDUE**: Number of days past due on CB credit at the time of application for related loan in our sample

7 **DAYS_CREDIT_ENDDATE**: Remaining duration of CB credit (in days) at the time of application in Home Credit – time only relative to the application

8 **DAYS_ENDDATE_FACT**: Days since CB credit ended at the time of application in Home Credit (only for closed credit) – time only relative to the application

9 **AMT_CREDIT_MAX_OVERDUE**: Maximal amount overdue on the Credit Bureau credit so far (at application date of loan in our sample)

10 **CNT_CREDIT_PROLONG**: How many times was the Credit Bureau credit prolonged

11 **AMT_CREDIT_SUM**: Current credit amount for the Credit Bureau credit

12 **AMT_CREDIT_SUM_DEBT**: Current debt on Credit Bureau credit

13 **AMT_CREDIT_SUM_LIMIT**: Current credit limit of credit card reported in Credit Bureau

14 **AMT_CREDIT_SUM_OVERDUE**: Current amount overdue on Credit Bureau credit

15 **CREDIT_TYPE**: Type of Credit Bureau credit (Car, cash,...)

16 **DAYS_CREDIT_UPDATE**: How many days before loan application did last information about the Credit Bureau credit come – time only relative to the application

17 **AMT_ANNUITY**: Annuity of the Credit Bureau credit

B.3 Bureau Balance Data Table Features (bureau_balance.csv)

1 **SK_ID_BUREAU**: Recoded ID of Credit Bureau credit (unique coding for each application) - use this to join to CREDIT_BUREAU table – hashed

2 **MONTHS_BALANCE**: Month of balance relative to application date (-1 means the freshest balance date) – time only relative to the application

3 **STATUS**: Status of Credit Bureau loan during the month (active, closed, DPD0-30,Ö [C means closed, X means status unknown, 0 means no DPD, 1 means maximal did during month between 1-30, 2 means DPD 31-60,Ö 5 means DPD 120+ or sold or written off])

B.4 Previous Application Data Table Features (previous_application.csv)

1 **SK_ID_PREV**: ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loan applications in Home Credit, previous application could, but not necessarily have to lead to credit) – hashed

2 **SK_ID_CURR**: ID of loan in our sample – hashed

3 **NAME_CONTRACT_TYPE**: Contract product type (Cash loan, consumer loan [POS] ,...) of the previous application

4 **AMT_ANNUITY**: Annuity of previous application

5 **AMT_APPLICATION**: For how much credit did client ask on the previous application

6 **AMT_CREDIT**: Final credit amount on the previous application. This differs from AMT_APPLICATION in a way that the AMT_APPLICATION is the amount for which the client initially applied for, but during our approval process he could have received different amount - AMT_CREDIT

7 **AMT_DOWN_PAYMENT**: Down payment on the previous application

8 **AMT_GOODS_PRICE**: Goods price of good that client asked for (if applicable) on the previous application

9 **WEEKDAY_APPR_PROCESS_START**: On which day of the week did the client apply for previous application

10 **HOUR_APPR_PROCESS_START**: Approximately at what day hour did the client apply for the previous application – rounded

11 **FLAG_LAST_APPL_PER_CONTRACT**: Flag if it was last application for the previous contract. Sometimes by mistake of client or our clerk there could be more applications for one single contract

12 **NFLAG_LAST_APPL_IN_DAY**: Flag if the application was the last application per day of the client. Sometimes clients apply for more applications a day. Rarely it could also be error in our system that one application is in the database twice

13 **RATE_DOWN_PAYMENT**: Down payment rate normalized on previous credit – normalized

14 **RATE_INTEREST_PRIMARY**: Interest rate normalized on previous credit – normalized

15 **RATE_INTEREST_PRIVILEGED**: Interest rate normalized on previous credit – normalized

16 **NAME_CASH_LOAN_PURPOSE**: Purpose of the cash loan

17 **NAME_CONTRACT_STATUS**: Contract status (approved, cancelled, ...) of previous application

18 **DAYS_DECISION**: Relative to current application when was the decision about previous application made time only relative to the application

19 **NAME_PAYMENT_TYPE**: Payment method that client chose to pay for the previous application

20 **CODE_REJECT_REASON**: Why was the previous application rejected

21 **NAME_TYPE_SUITE**: Who accompanied client when applying for the previous application

22 **NAME_CLIENT_TYPE**: Was the client old or new client when applying for the previous application

23 **NAME_GOODS_CATEGORY**: What kind of goods did the client apply for in the previous application

24 **NAME_PORTFOLIO**: Was the previous application for CASH, POS, CAR, Ö

25 **NAME_PRODUCT_TYPE**: Was the previous application x-sell o walk-in

26 **CHANNEL_TYPE**: Through which channel we acquired the client on the previous application

27 **SELLERPLACE_AREA**: Selling area of seller place of the previous application

28 **NAME_SELLER_INDUSTRY**: The industry of the seller

29 **CNT_PAYMENT**: Term of previous credit at application of the previous application

30 **NAME_YIELD_GROUP**: Grouped interest rate into small medium and high of the previous application – grouped

31 **PRODUCT_COMBINATION**: Detailed product combination of the previous application

32 **DAYS_FIRST_DRAWING**: Relative to application date of current application when was the first disbursement of the previous application – time only relative to the application

33 **DAYS_FIRST_DUE**: Relative to application date of current application when was the first due supposed to be of the previous application – time only relative to the application

34 **DAYS_LAST_DUE_1ST_VERSION**: Relative to application date of current application when was the first due of the previous application – time only relative to the application

35 **DAYS_LAST_DUE**: Relative to application date of current application when was the last due date of the previous application – time only relative to the application

36 **DAYS_TERMINATION**: Relative to application date of current application when was the expected termination of the previous application – time only relative to the application

37 **NFLAG_INSURED_ON_APPROVAL**: Did the client requested insurance during the previous application

B.5 POS CASH Balance Data Table Features (POS_CASH_balance.csv)

1 **SK_ID_PREV**: ID of previous credit in Home Credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit)

2 **SK_ID_CURR**: ID of loan in our sample

3 **MONTHS_BALANCE**: Month of balance relative to application date (-1 means the information to the freshest monthly snapshot, 0 means the information at application - often it will be the same as -1 as many banks are not updating the information to Credit Bureau regularly) – time only relative to the application

4 **CNT_INSTALMENT**: Term of previous credit (can change over time)

5 **CNT_INSTALMENT_FUTUR**: Installments left to pay on the previous credit

6 **NAME_CONTRACT_STATUS**: Contract status during the month

7 **SK_DPD**: DPD (days past due) during the month of previous credit

8 **SK_DPD_DEF**: DPD during the month with tolerance (debts with low loan amounts are ignored) of the previous credit

B.6 Installments Payments Data Table Features (installments_payments.csv)

1 **SK_ID_PREV**: ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) – hashed

2 **SK_ID_CURR**: ID of loan in our sample – hashed

3 **NUM_INSTALMENT_VERSION**: Version of installment calendar (0 is for credit card) of previous credit. Change of installment version from month to month signifies that some parameter of payment calendar has changed

4 **NUM_INSTALMENT_NUMBER**: On which installment we observe payment

5 **DAYS_INSTALMENT**: When the installment of previous credit was supposed to be paid (relative to application date of current loan) – time only relative to the application

6 **DAYS_ENTRY_PAYMENT**: When was the installments of previous credit paid actually (relative to application date of current loan) – time only relative to the application

7 **AMT_INSTALMENT**: What was the prescribed installment amount of previous credit on this installment

8 **AMT_PAYMENT**: What the client actually paid on previous credit on this installment

B.7 Credit Card Balance Data Table Features (credit_card_balance.csv)

1 **SK_ID_PREV**: ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit) – hashed

2 **SK_ID_CURR**: ID of loan in our sample – hashed

3 **MONTHS_BALANCE**: Month of balance relative to application date (-1 means the freshest balance date) – time only relative to the application

4 **AMT_BALANCE**: Balance during the month of previous credit

5 **AMT_CREDIT_LIMIT_ACTUAL**: Credit card limit during the month of the previous credit

6 **AMT_DRAWINGS_ATM_CURRENT**: Amount drawing at ATM during the month of the previous credit

7 **AMT_DRAWINGS_CURRENT**: Amount drawing during the month of the previous credit

8 **AMT_DRAWINGS_OTHER_CURRENT**: Amount of other drawings during the month of the previous credit

9 **AMT_DRAWINGS_POS_CURRENT**: Amount drawing or buying goods during the month of the previous credit

10 **AMT_INST_MIN_REGULARITY**: Minimal installment for this month of the previous credit

11 **AMT_PAYMENT_CURRENT**: How much did the client pay during the month on the previous credit

12 **AMT_PAYMENT_TOTAL_CURRENT**: How much did the client pay during the month in total on the previous credit

13 **AMT_RECEIVABLE_PRINCIPAL**: Amount receivable for principal on the previous credit

14 **AMT_RECIVABLE**: Amount receivable on the previous credit

15 **AMT_TOTAL_RECEIVABLE**: Total amount receivable on the previous credit

16 **CNT_DRAWINGS_ATM_CURRENT**: Number of drawings at ATM during this month on the previous credit

17 **CNT_DRAWINGS_CURRENT**: Number of drawings during this month on the previous credit

18 **CNT_DRAWINGS_OTHER_CURRENT**: Number of other drawings during this month on the previous credit

19 **CNT_DRAWINGS_POS_CURRENT**: Number of drawings for goods during this month on the previous credit

20 **CNT_INSTALMENT_MATURE_CUM**: Number of paid installments on the previous credit

21 **NAME_CONTRACT_STATUS**: Contract status (active signed,...) on the previous credit

22 **SK_DPD**: DPD (Days past due) during the month on the previous credit

23 **SK_DPD_DEF**: DPD (Days past due) during the month with tolerance (debts with low loan amounts are ignored) of the previous credit

References

[Yu17] Xiaojiao Yu. "Machine learning application in online lending risk prediction". In: *ArXiv e-prints* (July 2017). arXiv: 1707.04831 [q-fin.RM].

[Gro] Home Credit Group. *Home Credit Group Website*. URL: http://www.homecredit.net/. (accessed: 06.01.2018).

[Kaga] Kaggle. *Home Credit Default Risk Competition Data*. URL: https://www.kaggle.com/c/home-credit-default-risk/data. (accessed: 06.01.2018).

[Kagb] Kaggle. *Home Credit Default Risk Competition Overview*. URL: https://www.kaggle.com/c/home-credit-default-risk/. (accessed: 06.01.2018).

[Kagc] Kaggle. *Home Credit Default Risk Competition Public Leaderboard*. URL: https://www.kaggle.com/c/home-credit-default-risk/leaderboard. (accessed: 06.02.2018).

[Wika] ML Wiki. *ML Wiki ROC Analysis*. URL: http://mlwiki.org/index.php/ROC_Analysis. (accessed: 06.04.2018).

[Wikb] Wikipedia. *Receiver operating characteristic*. URL: https://en.wikipedia.org/wiki/Receiver_operating_characteristic. (accessed: 06.07.2018).