Predicting Credit Default - Home Credit

Rohan Prabhakar Agale
Springboard Capstone 2

Table of Contents

Introduction

Home Credit is an international consumer finance provider. It is on a mission to provide a safe, simple & fast borrowing experience to unbanked and underserved population using a powerful technological platform. On this mission, it wants to explore and utilize the predictive power of ML to extend credit access to deserving population and avoid defaults. Identifying debt paying capacity at the time of application is a win-win situation for Home Credit and applicants because it helps

1. Home Credit to maintain a healthy portfolio
2. Applicants to make an informed decision and avoid debt traps.

Thus, our **problem statement is to identify the potential clients as those**

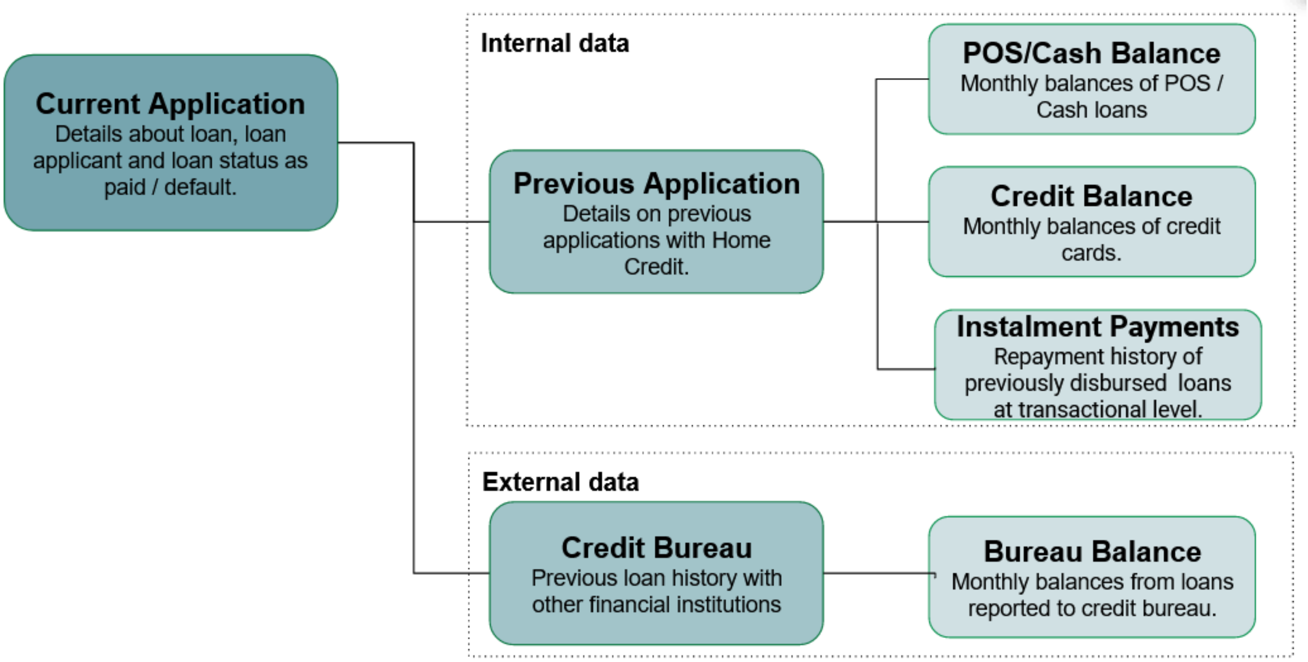
1. **who can repay the loans and**
2. **who can default on loans**

so that people capable of repayment are not rejected and the company maintains a healthy portfolio. **Predicting the severity of default is out of scope.**

Data

Home Credit has provided the data at Kaggle. The data contains information about application and applicant's details, her credit history with Home Credit and her credit history, if any, with other institutions which is provided by the Credit Bureau.

Data Schematics


Internal data
Current Application
Details about loan, loan applicant and loan status as paid / default.
Previous Application
Details on previous applications with Home Credit.
POS/Cash Balance
Monthly balances of POS / Cash loans
Credit Balance
Monthly balances of credit cards.
Instalment Payments
Repayment history of previously disbursed loans at transactional level.
External data
Credit Bureau
Previous loan history with other financial institutions
Bureau Balance
Monthly balances from loans reported to credit bureau.

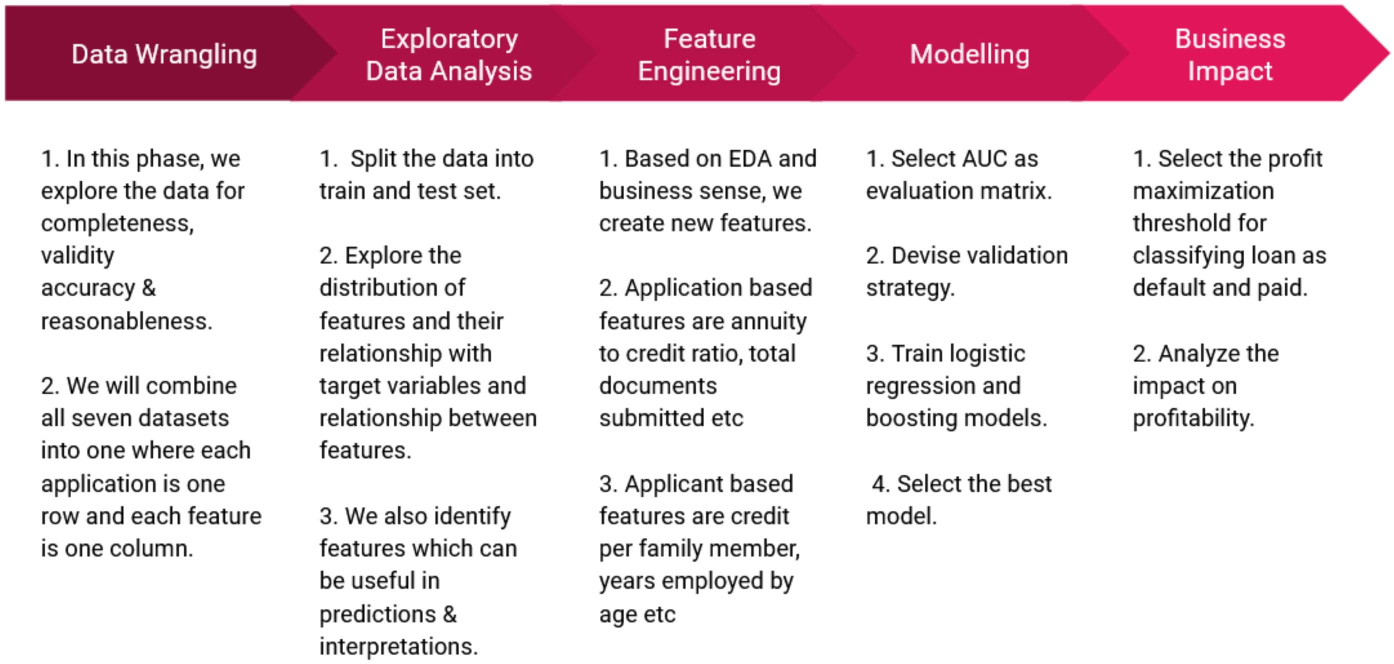
All this information is present in 7 csv files.

1. **application_train.csv**- The main file with details about credit and loan applicants. Each row represents an unique loan and whether it was repaid or defaulted on. We do not have records of rejected applications.
2. **previous_application.csv** - All previous applications for Home Credit loans of clients who have loans in our sample. Each row is one previous application.
3. **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 (# of loans in sample * # of relative previous credits * # of months in which we have some history observable for the previous credits) rows.
4. **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 (# of loans in sample * # of relative previous credit cards * # of months where we have some history observable for the previous credit card) rows.
5. **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.
6. **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 the number of credits the client had in the Credit Bureau before the application date.
7. **bureau_balance.csv** - Monthly balances of previous credits in the 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.

We also have a Data Dictionary with a short description of each variable. The details can be found at the **Data Description notebook**.
As the data is from Kaggle, our understanding is limited to what is described and discussed in the Kaggle forum. We may not get additional clarification.

Process

We will follow the below process for this project. Main steps in each stage are described in the diagram.



Data Wrangling

1. In this phase, we explore the data for completeness, validity accuracy & reasonableness.

2. We will combine all seven datasets into one where each application is one row and each feature is one column.

Exploratory Data Analysis

1. Split the data into train and test set.

2. Explore the distribution of features and their relationship with target variables and relationship between features.

3. We also identify features which can be useful in predictions & interpretations.

Feature Engineering

1. Based on EDA and business sense, we create new features.

2. Application based features are annuity to credit ratio, total documents submitted etc

3. Applicant based features are credit per family member, years employed by age etc

Modelling

1. Select AUC as evaluation matrix.

2. Devise validation strategy.

3. Train logistic regression and boosting models.

4. Select the best model.

Business Impact

1. Select the profit maximization threshold for classifying loan as default and paid.

2. Analyze the impact on profitability.

Once all 5 stages are done, we will communicate results using technical report and business presentation. Now let us deep dive into each stage.

1. Data Wrangling

In Data Wrangling, we explore the data for **completeness, validity, accuracy and reasonableness.** And we will combine all datasets into one dataset where each application is one record. And each variable/feature that has information about the application will be one column. You can refer to the **Data Wrangling notebook** for details. Below are some important points.

Following table contains the number of observations and number of variables in each dataset.

Data Dimensions

#	Dataset Name	# Observations	# Variables
1	application	307,511	122
2	prev_application	1,670,214	37
3	pos_cash_balance	10,001,358	8
4	credit_card_balance	3,840,312	23
5	installments_payments	13,605,401	8
6	bureau	1,716,428	17
7	bureau_balance	27,299,925	3
Total	-	58,441,149	218

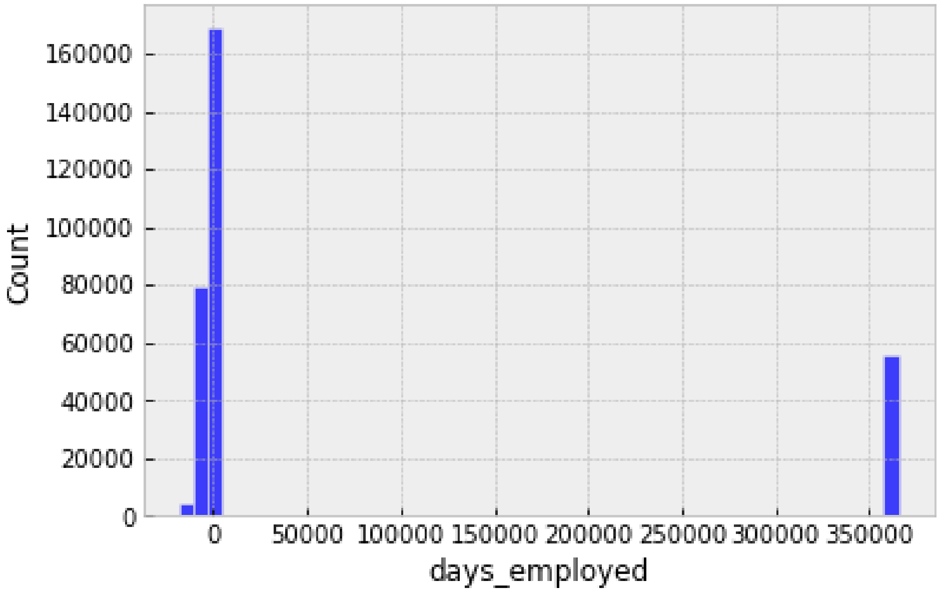
Using **pandas_profiling** library, we generated the data **summary reports**. We use these reports to gain a broad understanding and to identify any major issues in the data. We will document only the interesting findings and issues.

1. Application

Application data has missing values in 67 variables/features. We have the following.

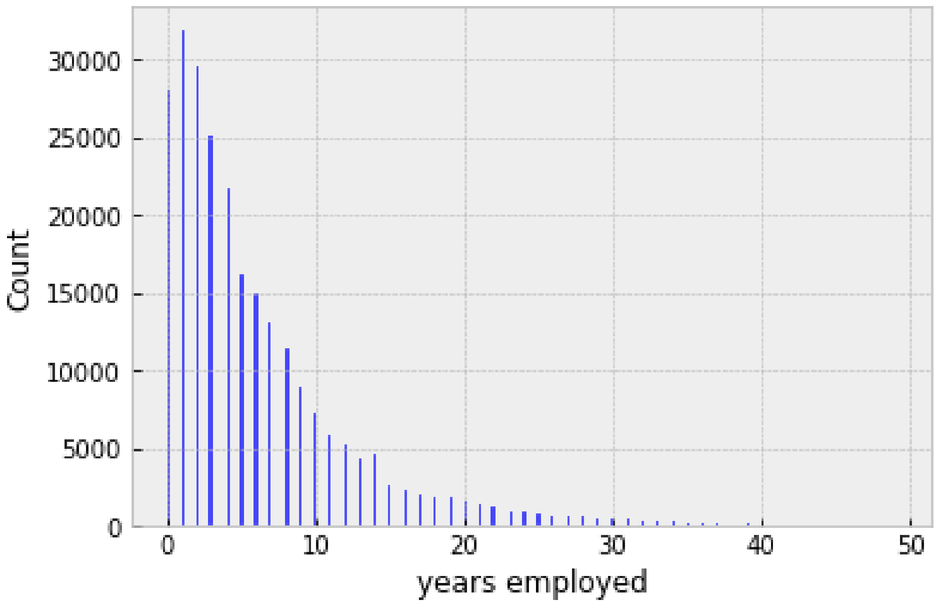
1. **target -** variable is present in all records. This is what we are trying to predict. target is 1 in case of default and 0 when loan is paid back.
2. **amt_annuity -** is missing in 12 observations. Annuity is an important feature and ideally, it should be present for all loans. We **delete these 12 records** without affecting our analysis.
3. **Property related features** - We have 47 features related to property and surrounding area eg housetype_mod, years_build_avg, apartments_avg etc. These features have missing value % ranging from 47% to 70%.
4. **code_gender -** We have 4 records where gender is XNA. Hence, any analysis will not be credible for XNA. So, **we combine XNA with F**(female).

5. **name_family_status -** Family status has two values to indicate married status. I) Married # 196432 records II) Civil Marriage # 29775 records
   It is worth checking the **accuracy** of these two categories with business. Whether all clients were presented with these two options and/or they were aware of the difference between two options or there is a way to verify this field. For now, we keep these two categories distinct.
6. **amt_credit > amt_goods_price** - Usually, the client pays some downpayment and borrows the remaining part of the goods price as a loan. Hence, the credit amount should be less than the price. Home Credit clarifies in the discussion that it is quite possible as insurance cost is added to the loan amount. If it is so, then how high can the insurance cost be with respect to a good's price ? Let's say insurance costs can go upto 50% of a good's price. With this modified check, we just have 2564 observations. These are < 1% of total loans. It would be worth to run a check with business to get a better understanding of these records. For now, **we accept these records as it is.**
7. **days_employed -** is the number of days an applicant was employed at the time of application and it is relative to the time of application. Hence, negative. Let's see the distribution of days employed.
   Days Employed at the time of application



+365243 days or ~1000 years does not seem to be a correct value as days are relative to application date, hence should be negative. Let us check what income type this value represents. Days employed are coded as 365243 for **pensioners**(# 55352) and **unemployed**(# 22) people. It is clear that days_employed information is provided only for currently employed people. Hence, we can **replace this odd value 365243 with nan** because we do not have actual information and days employed could be different for different pensioners. Let's **exclude these records and convert days to years** for reasonableness and validity check on the distribution.

Years Employed at the time of application



The distribution seems sensible with minimum 0 and maximum 49 years. And days_employed < days_birth for all observations.

8. **days_birth, days_registration, days_id_publish are negative** - Just like days_employed, all these dates are relative to the time of application. Hence, negative.

2. Previous Application

We have missing data in 16 features in this data. Below are some important observations.

1. **Name_contract_status -** We have all types of previous applications; approved, cancelled, refused and unused offers.
2. **amt_annuity -** has 22% missing values. Annuity is missing for most but not for all of the cancelled and unused loans. It may be because non-approval can happen at different stages of application. Against ~250k refused loans, annuity amount is present.
3. **amt_credit** - 20%(# 336768) loans have zero credit amount against them. Only 1551 are approved loans.
4. **days_first_due, days_last_due_1st_version, days_termination, days_last_due, days_first_drawing** - features have 40% missing values and are missing for the same observations. These are largely Non- Approved loans. All the dates are relative to the time of application and are negative. For some observations, **+365243** days i.e. ~1000 years in the future is present in some dates features. This is because of many reasons like revolving loans or clients returning the goods soon after taking the loan etc. We r**eplace this odd value 365243 with nan.**
5. **33% of current applications are having more than 5 previous loan applications.** One customer has 77 applications with whooping 64 refusals.

3. POS Cash Balances

POS Cash balances data is complete except that it has 2 count installment features with 0.3% missing values. POS data has only count information and no amount information.

4. Credit Card Balances

1. We have **20% missing** values in drawing amount & count features. These are **amt_drawings_atm_currnet, amt_drawings_pos_current, cnt_drawings_atm_current** etc. **amt_payment_current** feature has 20% missing values. All these features are largely missing for the same observations.
2. **amt_balance** - Amount balance is negative in 0.06% cases. These customers overpaid what they own. Hence, we get a negative amount.

5. Installment

While pos/cash & credit data are snapshot on certain day of the month, installment data is **transactions at day level**. This data does not have missing values.

6. Bureau

This data is received from the credit bureau. It contains information about loans from other financial institutions.

1. **amt_annuity** - amt_annuity is missing in 71.5% of observations.
2. **credit_credit_enddate** - We have extreme values in days corresponding to 115 years in the past and 85 years in the future.This values do not make sense. We have a similar issue in **days_enddate_fact** & **days_credit_update**.
3. **amt_credit_max_overdue** - 65% values are missing. We have an extremely skewed distribution with a median of 0 and a maximum of 116 million.

7. Bureau Balance

We have 3 variables- ID, monthly balance and status- in this dataset. This data does not have missing values.

1. **months_balance** - This indicates the month of balance relative to the application date. Like days related fields, this is also negative.

8. Combining the data

Now we will combine all the datasets into one and we will follow below merge and summarization strategies.

1. Merge strategy

We will do this in two steps for both internal and external datasets.
For **Internal Data**

1. **POS Cash balances, Credit balances** and **Installment** are monthly or transactional details of each previous loan application. Hence, firstly we summarize the experience of each loan from these datasets so that we have one row per loan. Then merge this data with **Previous Application** data. Let's call this prev_app_merged data.
2. Secondly, we summarize prev_app_merged so that we have one row per current application. Finally, we will merge the summarized prev_app_merged data with the **Application** dataset.

We follow same process for **External Data**

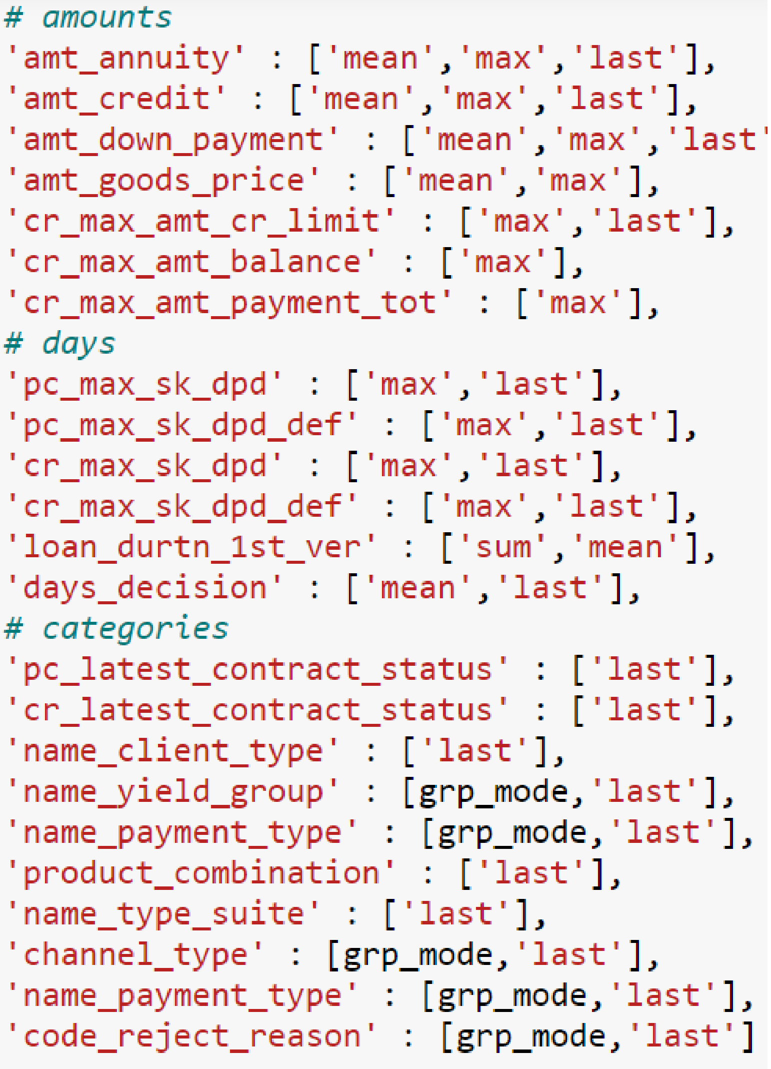
1. Firstly, we will summarize **Bureau Balance** data at each Bureau loan. And then merge this data with **Bureau** data. Let's call this bc_merged.
2. Secondly, we summarize bc_merged so that we have one row per current application. Finally, we will merge the summarized bc_merged data with the **Application** dataset.

2. Summarization strategy

Before combining, we need to summarize the datasets. We will apply business logic to do so. We are concerned about the applicant's loan paying capacity. We will use historical data in 2 ways to predict it.

1. We will summarize total loan experience using mean, maximum and most frequent functions.
2. The applicant's recent financial well being may differ from her financial well being in the past. Hence, we will also create features from the last (most recent) loan experience.

We have created a custom function **grp_mode** which can find the most frequently occurring value in a categorical variable. If the most frequently occurring value is nan, then the function will return nan.
Here is a code snippet of aggregation logic for some features.



```
# amounts
'amt_annuity' : ['mean','max','last'],
'amt_credit' : ['mean','max','last'],
'amt_down_payment' : ['mean','max','last'
'amt_goods_price' : ['mean','max'],
'cr_max_amt_cr_limit' : ['max','last'],
'cr_max_amt_balance' : ['max'],
'cr_max_amt_payment_tot' : ['max'],
# days
'pc_max_sk_dpd' : ['max','last'],
'pc_max_sk_dpd_def' : ['max','last'],
'cr_max_sk_dpd' : ['max','last'],
'cr_max_sk_dpd_def' : ['max','last'],
'loan_durtn_1st_ver' : ['sum','mean'],
'days_decision' : ['mean','last'],
# categories
'pc_latest_contract_status' : ['last'],
'cr_latest_contract_status' : ['last'],
'name_client_type' : ['last'],
'name_yield_group' : [grp_mode,'last'],
'name_payment_type' : [grp_mode,'last'],
'product_combination' : ['last'],
'name_type_suite' : ['last'],
'channel_type' : [grp_mode,'last'],
'name_payment_type' : [grp_mode,'last'],
'code_reject_reason' : [grp_mode,'last']
```

3. Feature Engineering

We have also engineered new **behavioral and financial** features while summarizing the data. Below is the logic behind creating these features using one example.
payment_delay - This feature indicates delay in payment from the due date of installment. This delay is counted in number of days. It is zero if the loan is paid before the due date.
in_sum_payment_delay - We sum **payment_delay** to get total delay for the loan.
rt_payment_delay - We normalize **in_sum_payment_delay** by total loan duration.

Similarly we have derived **payment_advance**(paid before due date), **payment_deficit**(shortfall in the required installment amount), **payment_surplus**(surplus in the required installment amount) and many more features. With this, we end the Data Wrangling stage with **307499 observations and 230 variables** including target.

2. Exploratory Data Analysis (EDA)

In this section, we will

1. Split the data into train and test set
2. Distribution of features and their relationship with target variables.
3. Relationship between features.

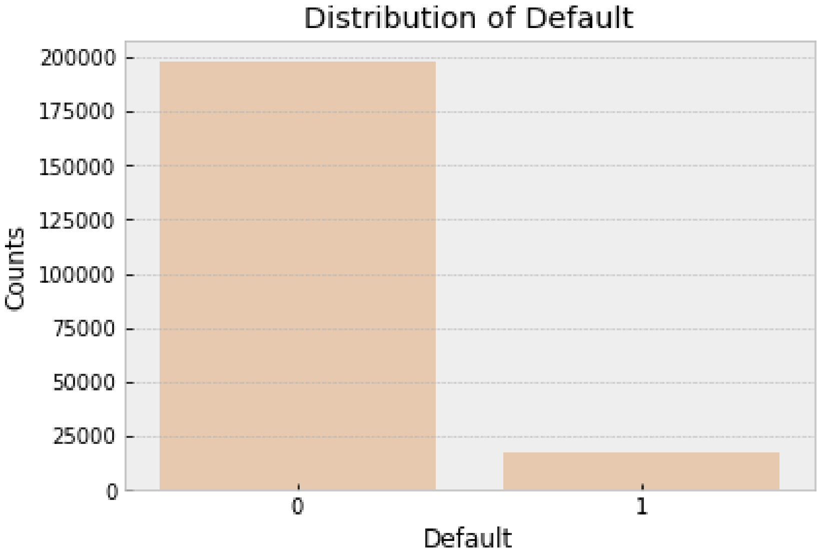
and identify features which can be useful in predictions & interpretations.

1. Train Test Split

We split the combined data into train and test sets. Train set is 70% with 215249 observations and **Test set is 30%** with 92250 observations. Using stratification, we ensured that both test and test sets have the same ie 8.07 % default loans.
Following analysis is based on the train dataset.

0. Target Variable

We are classifying the customers who can repay the loans and customers who will default on the loan. Our response variable is ‘target’ with 0/1 value indicating loan paid or loan defaulted respectively. We have 17,377 observations with defaults i.e. **8.07%** default rate.



2. Distribution of features

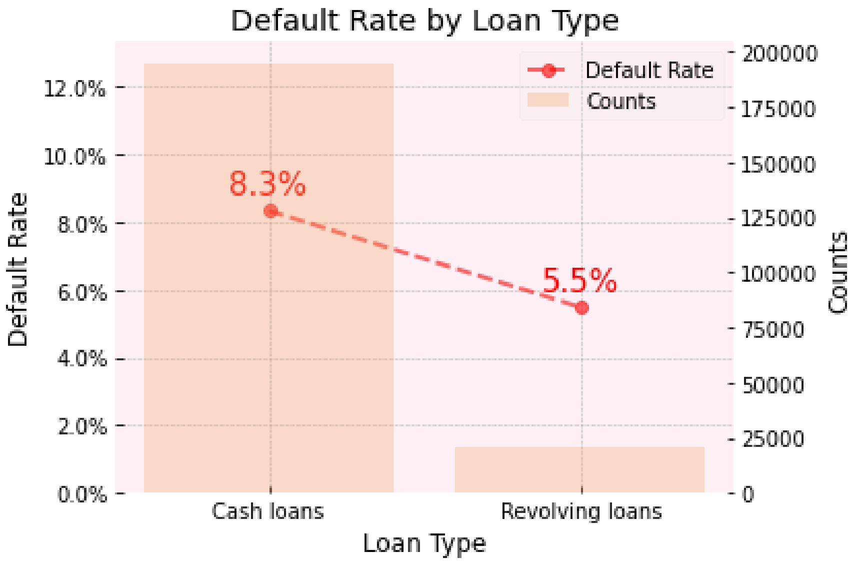
In this section, we will analyse the distribution of each feature and how the default rate varies across that feature. We will document only the important and interesting features which may be useful in predictions.

For **categorical features**, we will plot the feature on x-axis and default rate on primary y-axis. We will also plot the total number of observations(counts) in each category on the secondary y-axis. With counts, we can gauge the credibility/reliability of the default rate for that category.

For **continuous features**, we will first bin the feature into a number of groups. This will smoothen out the trend and we will get a better picture. We may remove 1 percentile outliers in case of extreme values for the smoothness and clarity. This is just for visualization purposes and we are going to use the original continuous feature for modelling. Now, we plot bins on x-axis and default rate on primary y-axis. We will also plot the total number of observations(counts) in each bin on the secondary y-axis. We will exclude missing data as well.

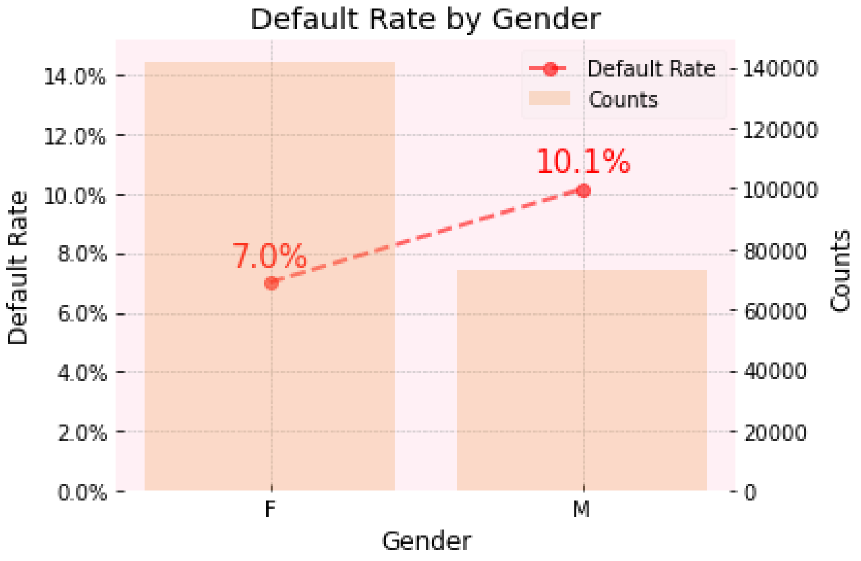
With this framework in mind, let us see what loan portfolio we have.

1. Type of Loan



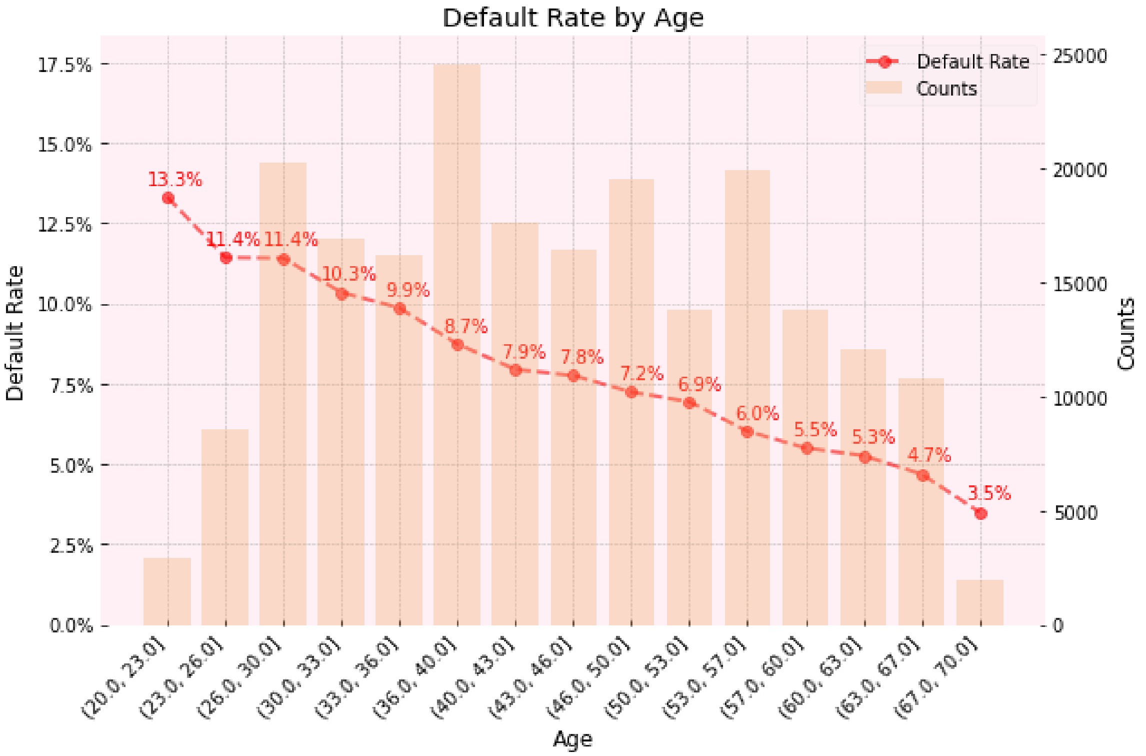
90% loans are cash loans and cash loans have a higher default rate.
Now we will explore the types of clients we are lending to.

2. Gender



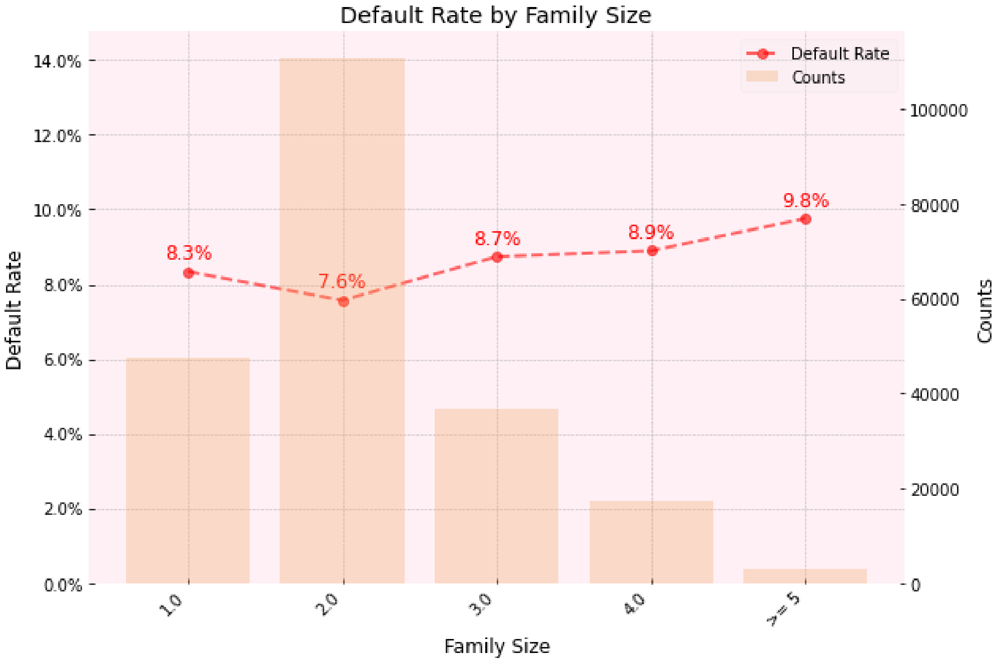
~65% of applicants are females. Females are on average 3% points better than males in paying back the loans. In general, females are conservating when it comes to managing finances and the trend is indicating the same.

3. Applicant’s Age



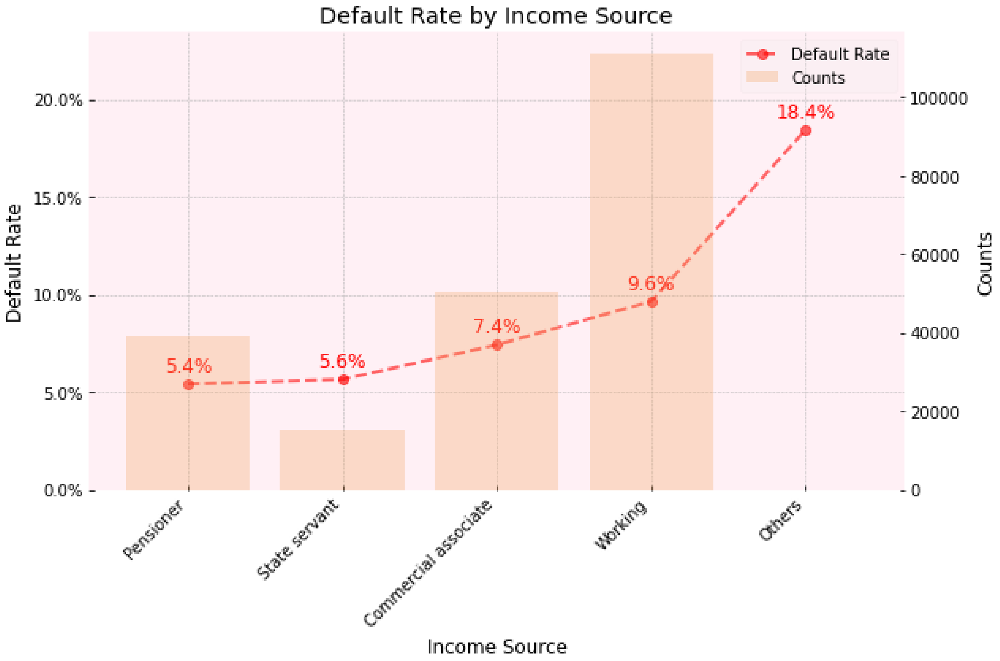
Raw feature is days_birth. We convert it to age. Applicant's age is within a reasonable range of 21-69. Home credit lends to clients across all ages. We can clearly see that the default rate decreases as age increases. Average risk differentiation between the 2nd youngest and the 2nd last oldest groups is more than 7% points.

4. Family Size



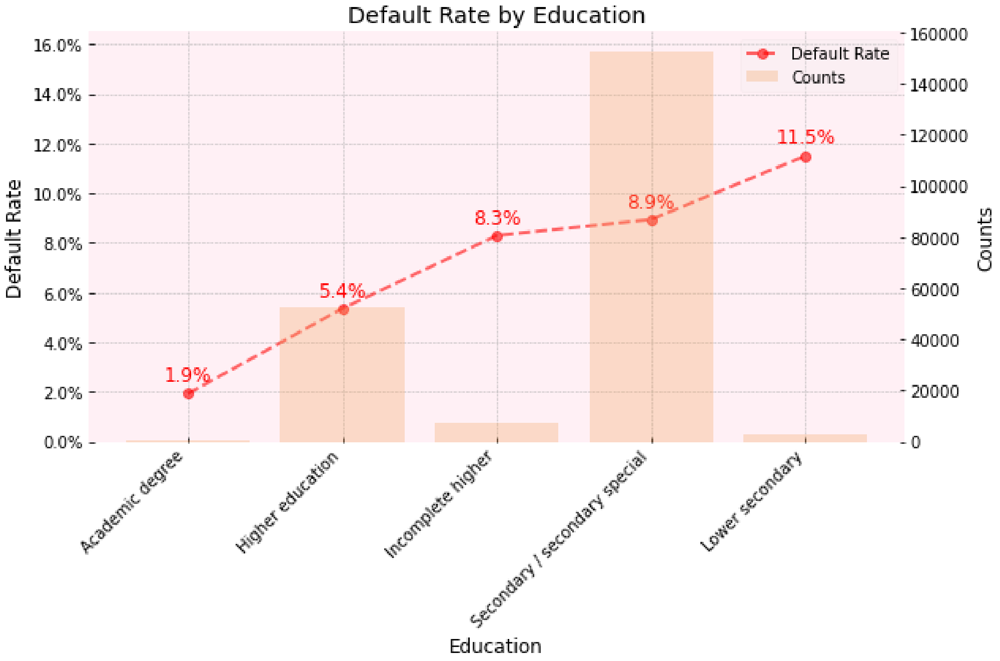
~50% of the clients come from a family of 2. And this group seems to have the lowest default rate.

5. Income source



More than 50% loans are given to working professionals. Pensioners and State servants are having much lower average default rate than working professionals.

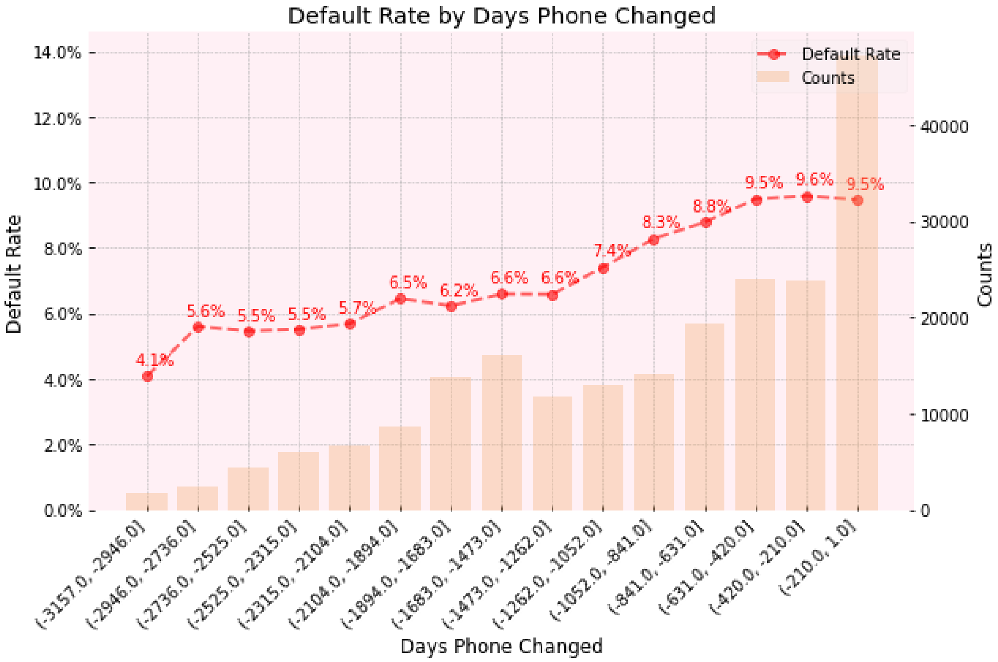
6. Education



~ 70% of the clients fall into the secondary education category and ~25% of clients fall into the higher education category. Higher the education, lower the default rate.

7. Days Phone Change

The number of days since the client changed the phone.



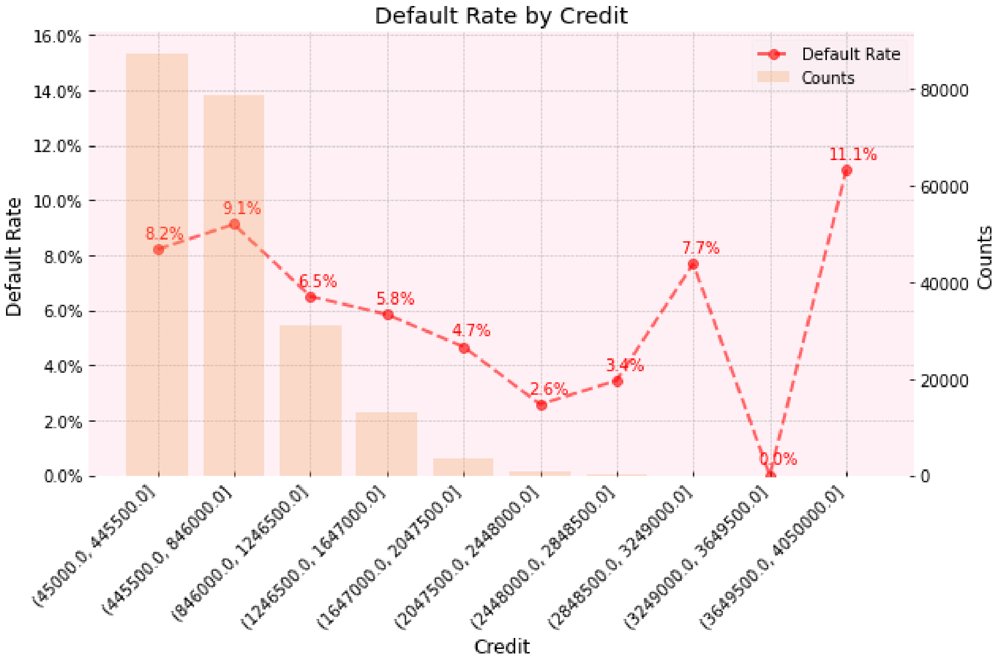
Days are relative to the day of application, hence negative. On average, the older the phone, the lower the default rate. This feature may be related to age.

Let us explore **loan amounts related features**. These are

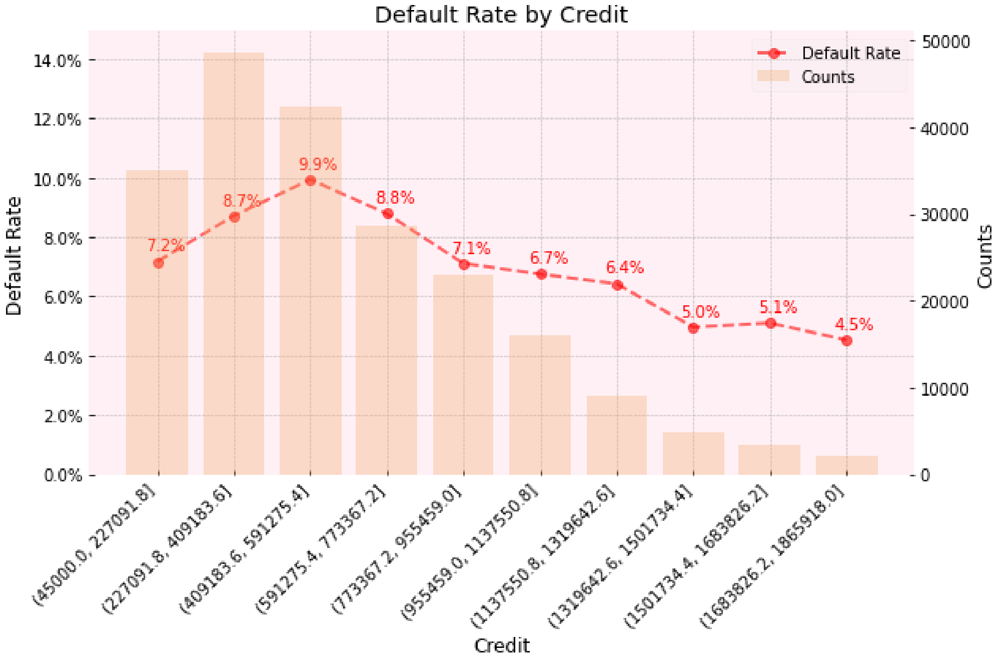
1. Credit Amount - Actual loan amount.
2. Application Amount - Loan amount the client had applied for.
3. Annuity Amount - Monthly installment amount to be paid.

Let us explore how credit amount affects the default rate.

8. Credit Amount

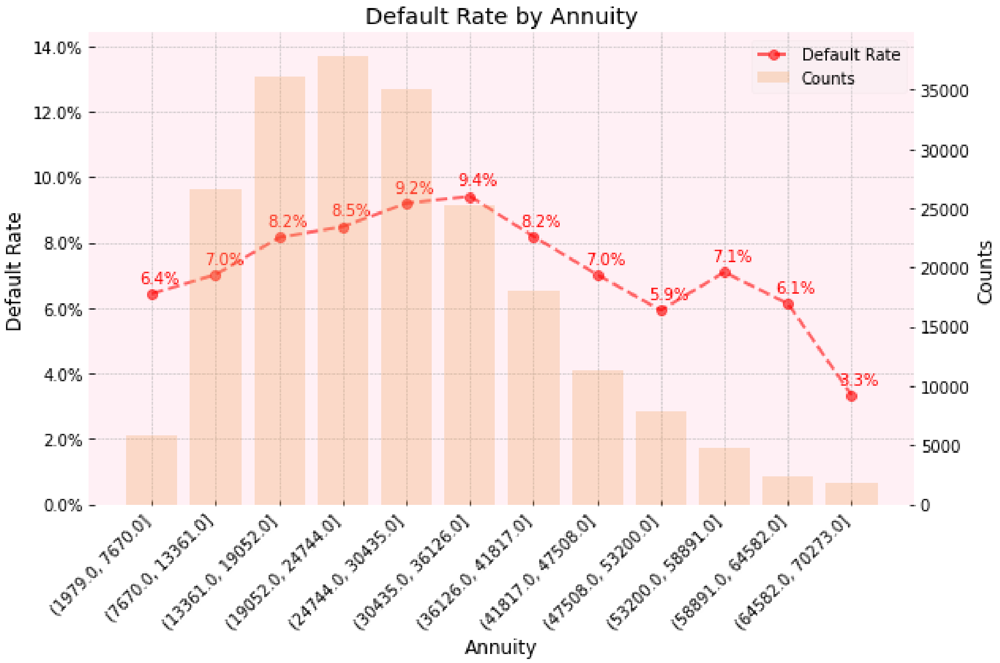


We have some extreme outliers with less observations at higher credit amounts. This is leading to high volatility in the last 4-5 bins. And given so few observations, it will be difficult to rely on the trend. The idea is to observe the general trend for credit feature. Therefore, rather than combining the top 1 percentile with lower percentiles, we would remove the top 1 percentile so that the overall trend is not distorted. We will follow the same logic for other continuous features with extreme outliers at the bottom/top end of the distribution.
We plot the credit amount after removing the top 1 percentile.



We can see an inverted V type trend. Initially, with an increase in credit amount, the default rate increases and it peaks around the median. Then it starts decreasing again.
For higher credit amounts, the underwriting process can be stricter than usual. We can ask business about the validity of these trends. We see **similar distribution and trends in goods price amounts.**

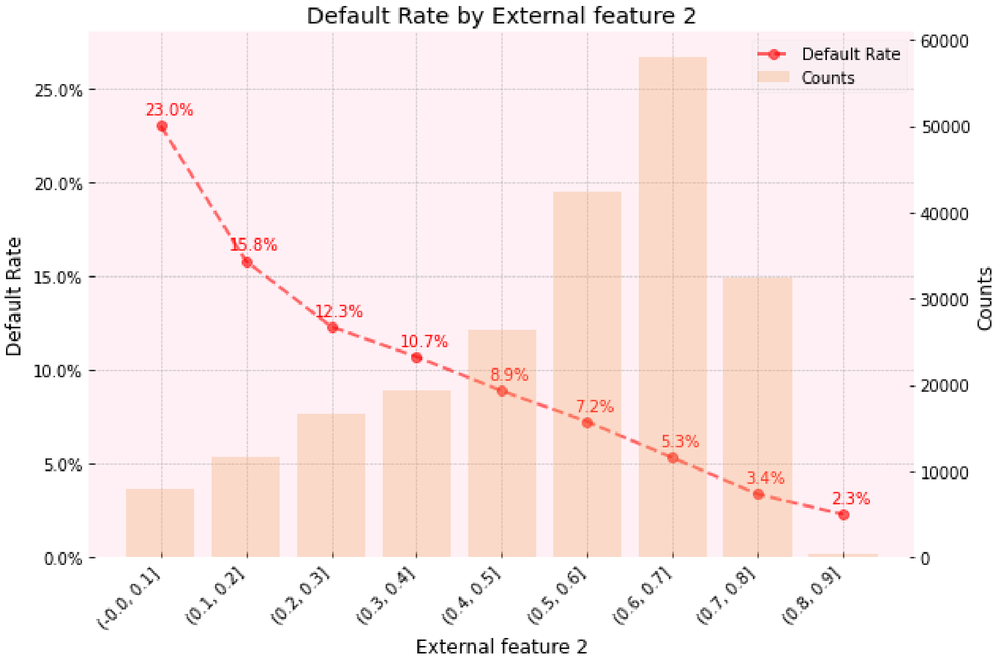
9. Annuity



In Annuity amount, it is more like an inverted U trend after removing the top 1 percentile.

We use external data to evaluate loan applications. There are 3 anonymised features.
Does the data help us in predicting the risk of the applications ?

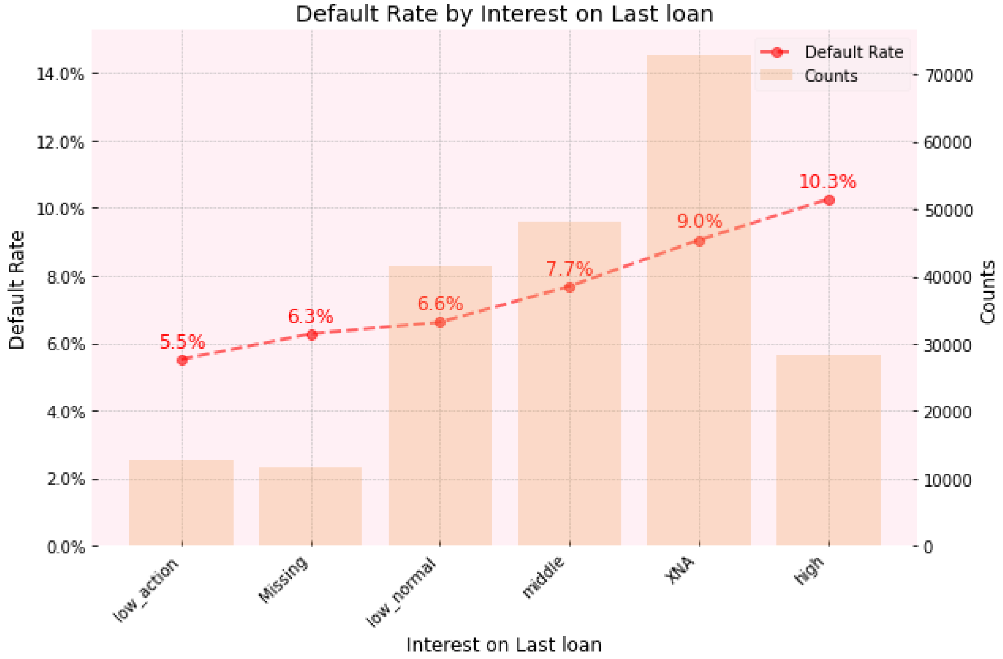
10. External feature 2



This looks like a normalized feature. As the value of this feature starts approaching 1, the default rate starts decreasing. This feature is exhibiting a strong downward trend with excellent risk differentiation. We also have external features 1 & 3 with similar trends. But they have 56% and 20% missing values respectively. Whereas the external feature 2 has just 0.2% missing values. Hence, we prefer external feature 2.

We have a **history** of the previous applications. The features from history are limited and somewhat different from features in current application.
Can the previous loan history help us predict default rate ?

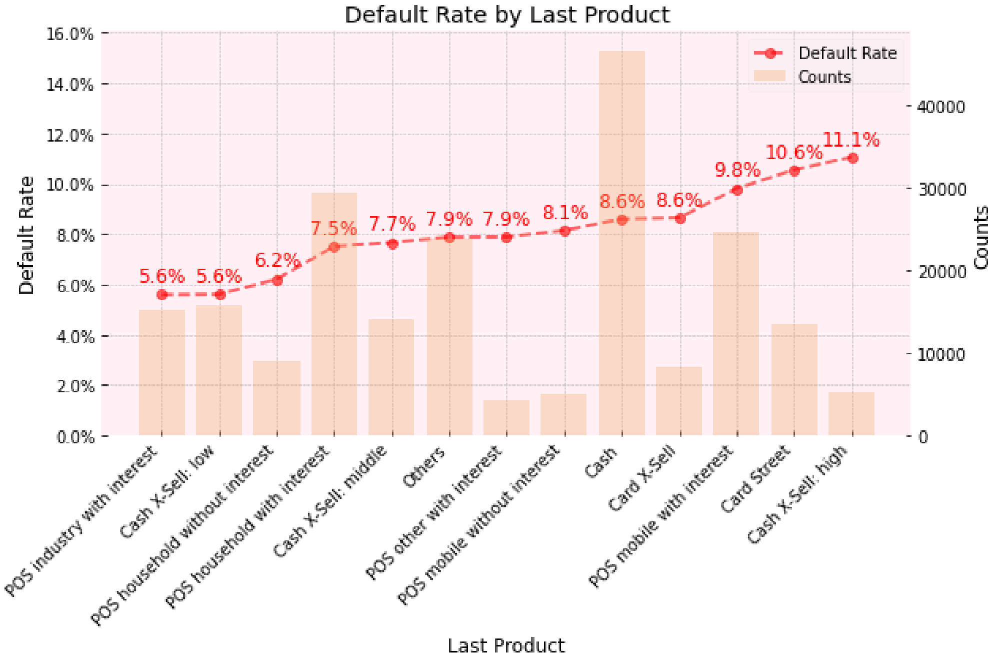
11. Interest on last loan



XNA interest rate is not defined in the data dictionary. Default rate is higher for higher interest offered. This is in some way a confirmation of the existing lending process.

12. Last Product

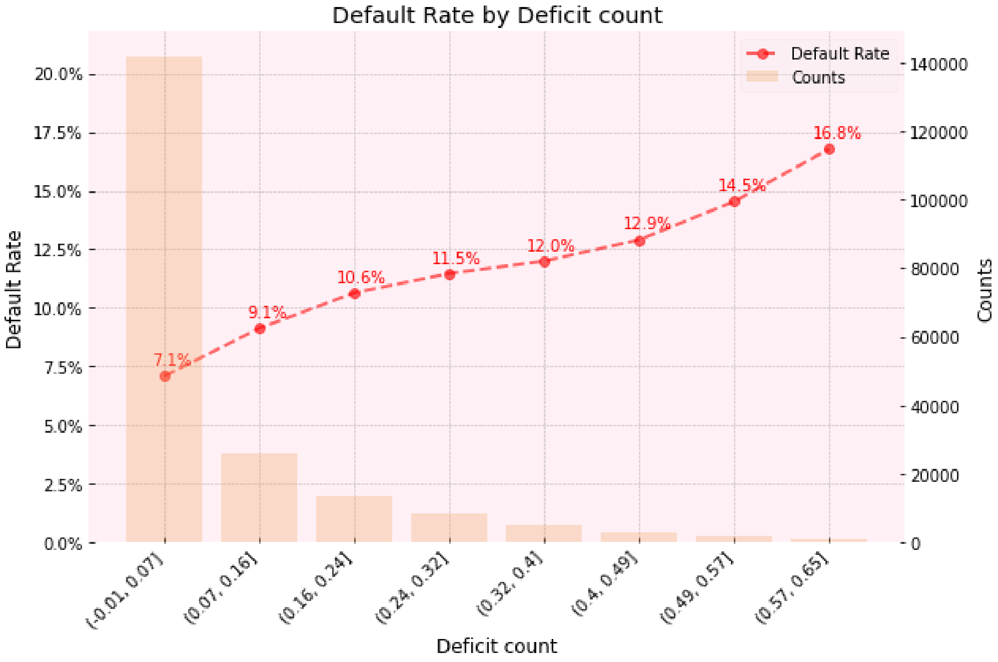
What product client had applied for last time ?



Default rate varies by products. It will be good to have an understanding of products. Are they offered based on risk ?

13. Deficit Count - Normalized

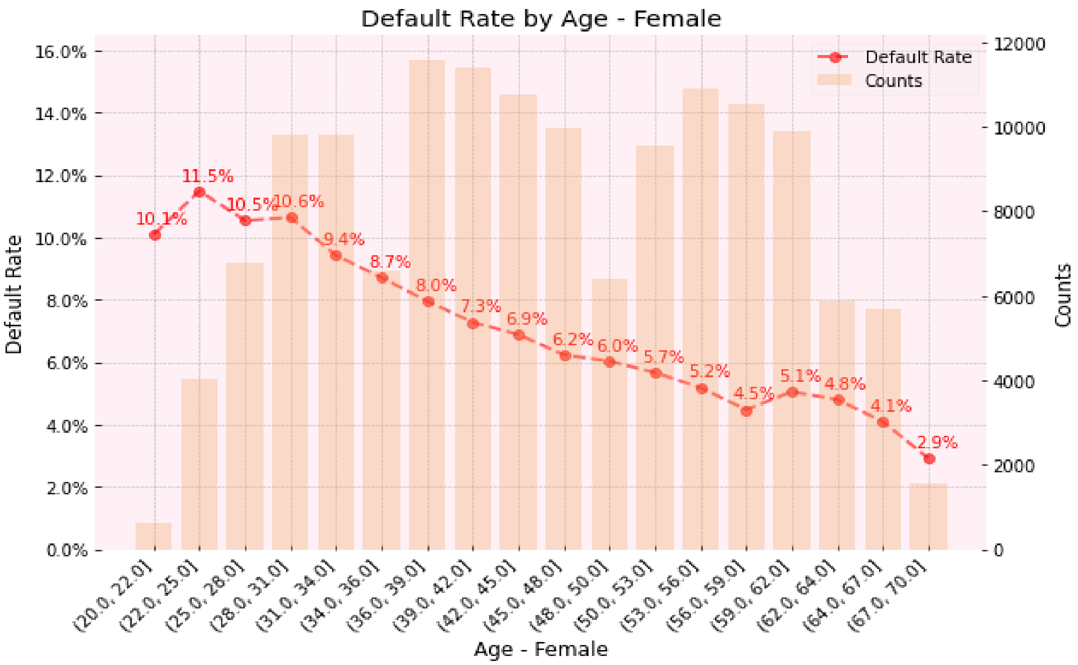
The number of times there was a deficit in installment payment in past loans. It is normalized by total installment counts.



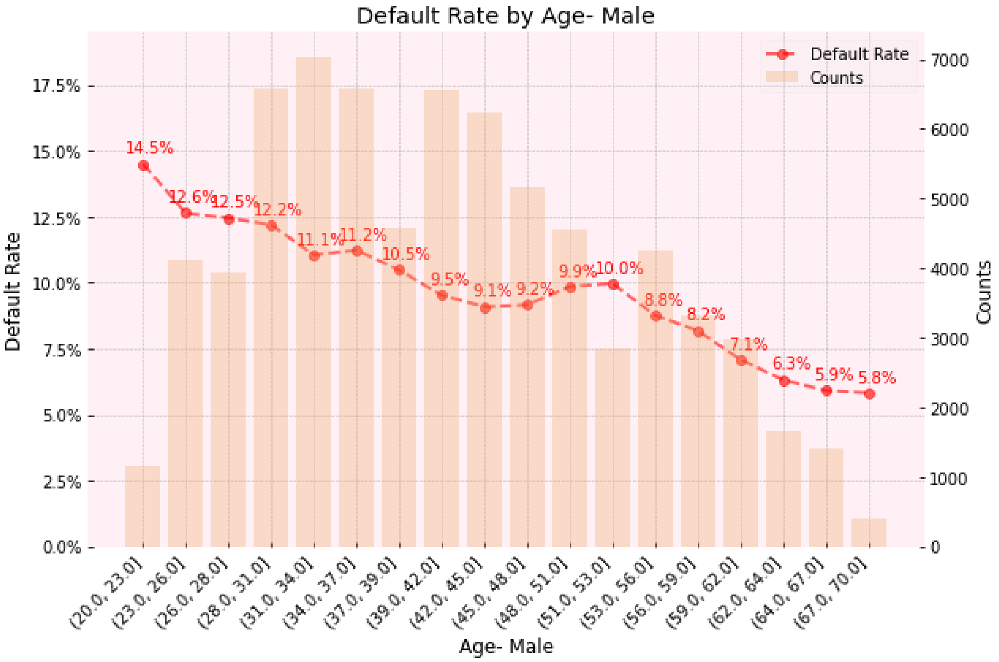
This is a derived feature from installment dataset. We count the number of times there was a shortfall in minimum installment payment. Then we normalize this count by the total number of installments. 55% applications have paid required minimum installments amounts or more, indicated by 1st bin where all observed values are zeros. But as the deficit count increases, the default rate increases. This is intuitive as someone who is unable to pay the minimum installment amount is more likely to default. (~6% applications do not have past history and are missing and we are excluding top 1 percentile observations.)

14. Age x Gender

We will look at the effect of age on the default rate separately for females and males. This will help us understand whether the trend in age depends on gender. This is known as interaction.
Let us plot the default rate by age separately for females and males.



Default Rate by Age - Female



Default Rate by Age- Male

Just as we had observed for gender feature, the default rate is lower for females than for males. Apart from that, from age 28 to 56, default rate steeply decreases as female age increases. But, the decrease in default rate is slower for males in the same age range. This is indicating an interaction between age and gender features.

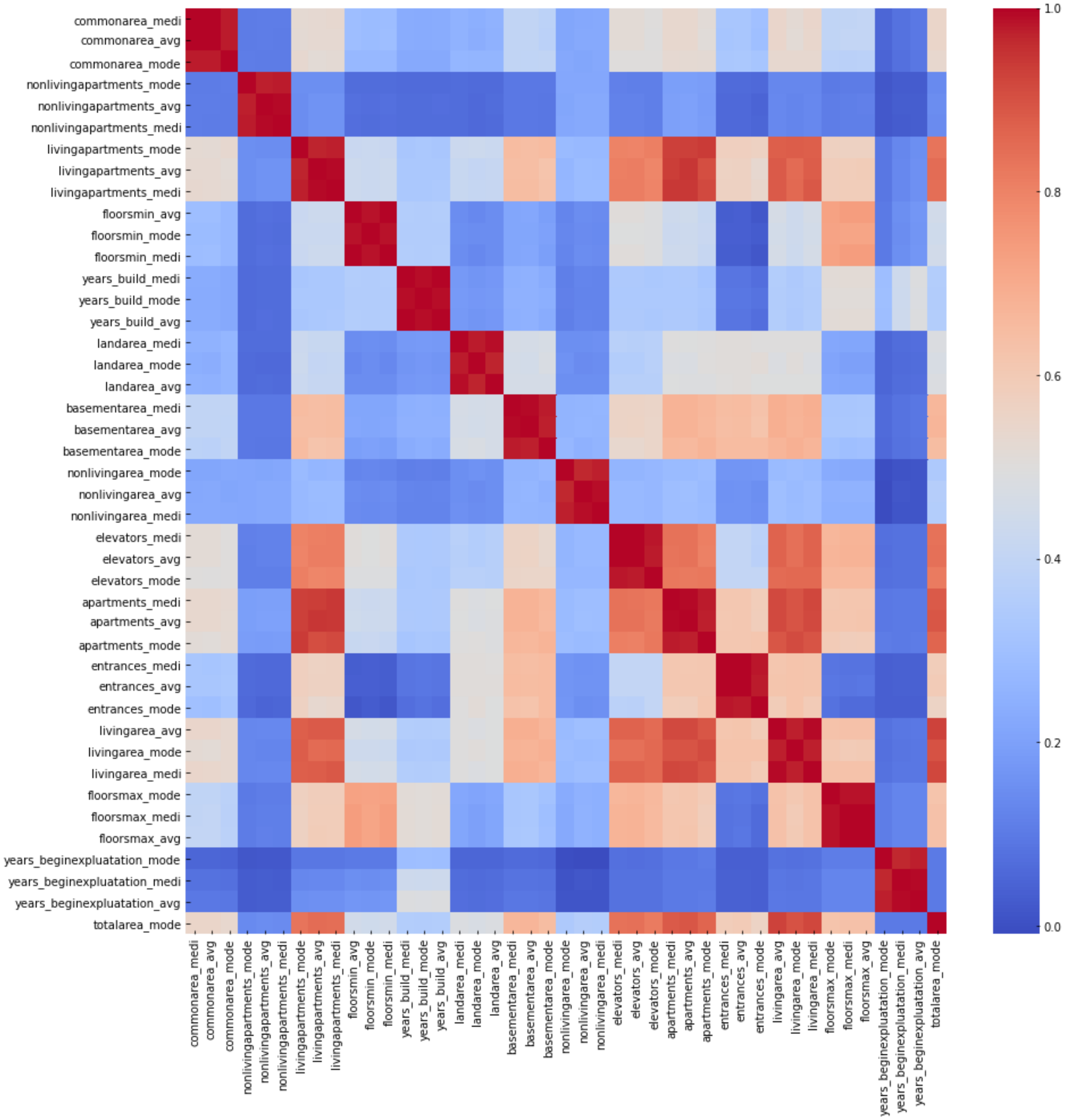
Now that we have a sense of what features can be predictive of loan default rate, we can look at the relationship between features using correlation. Along the way, we identify features to drop based on high correlation and high missing values.

3. Correlations

1. Continuous features

We have 43 continuous property features corresponding to house, apartment and area. These features have missing values in the range 47% to 69%. With so many missing values, we would look at correlation to reduce the number of features rather than doing PCA.

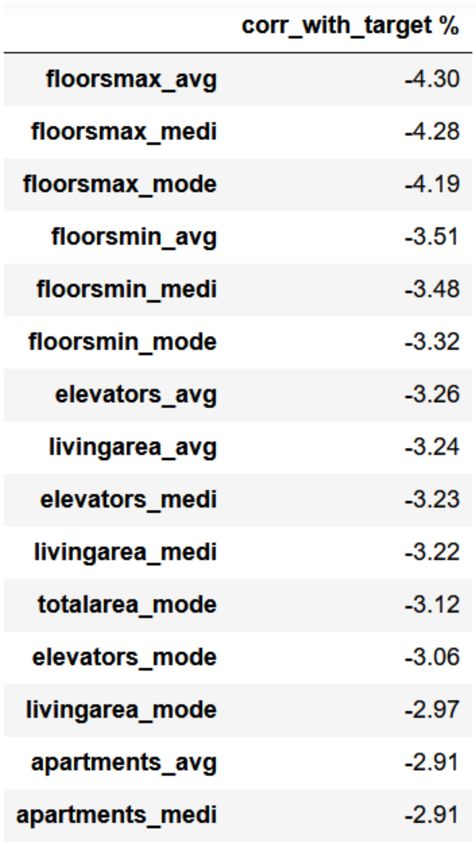
Property Features :



Looking at the correlation heatmap, 3 things are clear.

1. We have a positive correlation between features.
2. We have 3 different derivations of each feature : median, mode, average. These 3 derivations are more than 95% correlated with each other.
3. Living area, living apartments, elevators, apartments and total area are ~90% correlated with each other. Total area has just the mode feature.

Let's pick the feature which is more correlated to the target to represent each group.



	corr_with_target %
floorsmax_avg	-4.30
floorsmax_medi	-4.28
floorsmax_mode	-4.19
floorsmin_avg	-3.51
floorsmin_medi	-3.48
floorsmin_mode	-3.32
elevators_avg	-3.26
livingarea_avg	-3.24
elevators_medi	-3.23
livingarea_medi	-3.22
totalarea_mode	-3.12
elevators_mode	-3.06
livingarea_mode	-2.97
apartments_avg	-2.91
apartments_medi	-2.91

1. Of the 3 derivations, average is more correlated with target than mode and median.

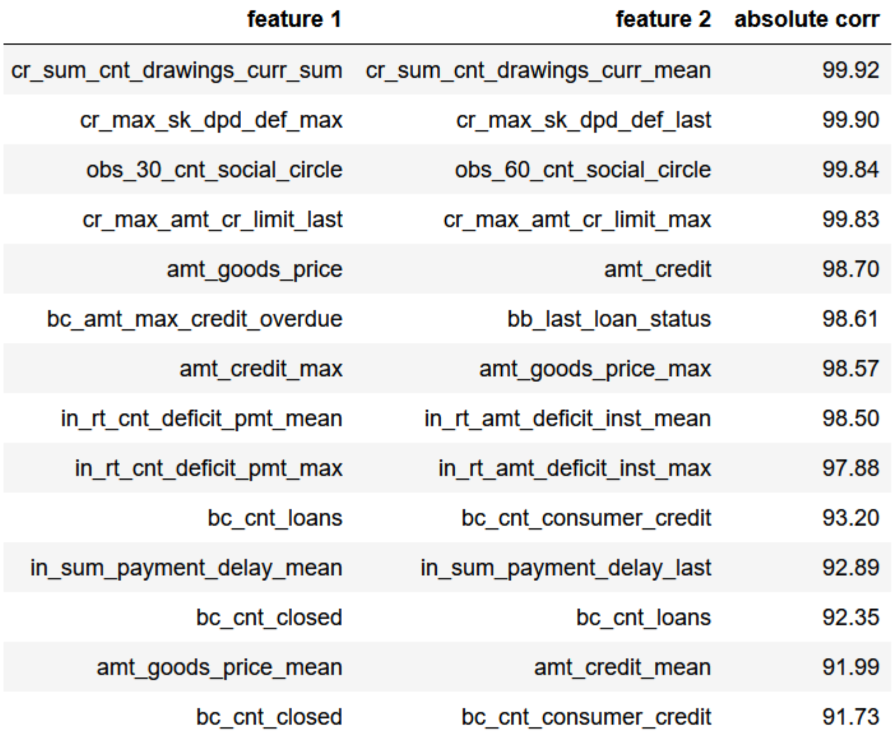
2. We can pick livingarea_avg feature to represent those 5 features because of high correlation with target and better distribution spread.

3. We drop commonarea_avg, nonlivingapartments_avg, livingapartments_avg features as they have 69% missing values.

4. Finally, **we have only the following features.**
'basementarea_avg',
'floorsmax_avg',
'floorsmin_avg',
'landarea_avg',
'livingarea_avg',

Note : The correlation mentioned is Persons. Results of Point Biserial & Pearson correlation are identical when the categorical feature is binary. These correlations are statistically significant.

Remaining non-property features :
Now let's explore correlation between remaining non-property features. As there are 126 features, it is difficult to visualize. Hence, we will list features which have greater than 90% absolute correlation. The 90% threshold is judgemental. Idea is to understand highly correlated features and make sense of them.



feature 1	feature 2	absolute corr
cr_sum_cnt_drawings_curr_sum	cr_sum_cnt_drawings_curr_mean	99.92
cr_max_sk_dpd_def_max	cr_max_sk_dpd_def_last	99.90
obs_30_cnt_social_circle	obs_60_cnt_social_circle	99.84
cr_max_amt_cr_limit_last	cr_max_amt_cr_limit_max	99.83
amt_goods_price	amt_credit	98.70
bc_amt_max_credit_overdue	bb_last_loan_status	98.61
amt_credit_max	amt_goods_price_max	98.57
in_rt_cnt_deficit_pmt_mean	in_rt_amt_deficit_inst_mean	98.50
in_rt_cnt_deficit_pmt_max	in_rt_amt_deficit_inst_max	97.88
bc_cnt_loans	bc_cnt_consumer_credit	93.20
in_sum_payment_delay_mean	in_sum_payment_delay_last	92.89
bc_cnt_closed	bc_cnt_loans	92.35
amt_goods_price_mean	amt_credit_mean	91.99
bc_cnt_closed	bc_cnt_consumer_credit	91.73

We observe that highly correlated features are

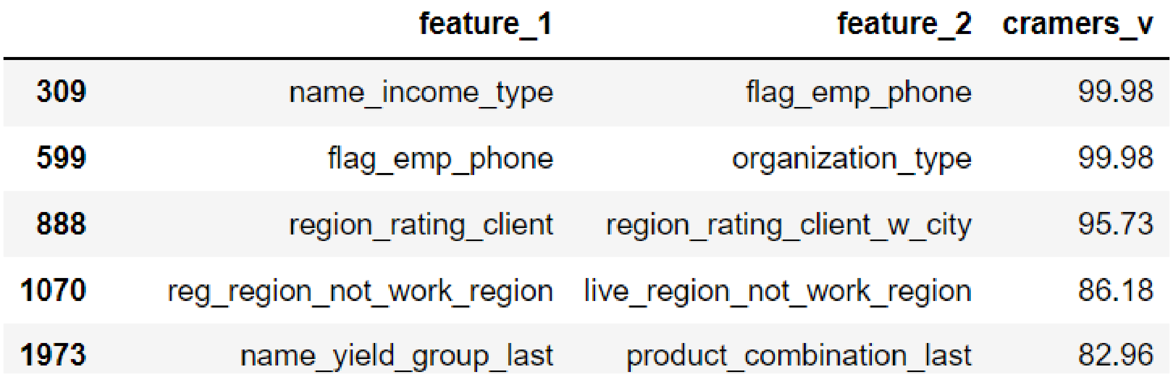
1. Derivations of same feature eg sum & mean, mean & last etc
2. Derivations of same pair of features eg (amt_goods_price_max , amt_credit_max), (amt_goods_price_mean , amt_credit_mean)

It makes sense to drop one feature from each pair. And we will drop the feature which has more missing values within the pair. But we will keep both amt_goods_price & amt_credit in raw form.

Dropped features :
'cr_sum_cnt_drawings_curr_mean',
'amt_goods_price_mean',
'cr_max_sk_dpd_def_max',
'obs_60_cnt_social_circle',
'cr_max_amt_cr_limit_max',
'amt_goods_price_max',
'in_rt_cnt_deficit_pmt_mean',
'in_rt_cnt_deficit_pmt_max',
'bc_cnt_loans',
'in_sum_payment_delay_mean',
'bc_cnt_loans',
'bc_cnt_consumer_credit'

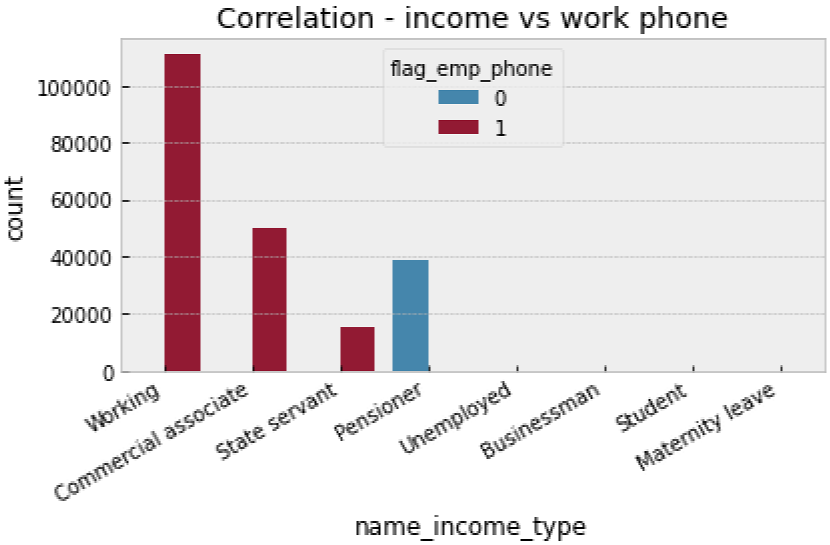
2. Categorical features

We will use Cramer’s V correlation to understand the relationship between categorical features.
We have 64 categorical features. Below are the top 5 correlation pairs.

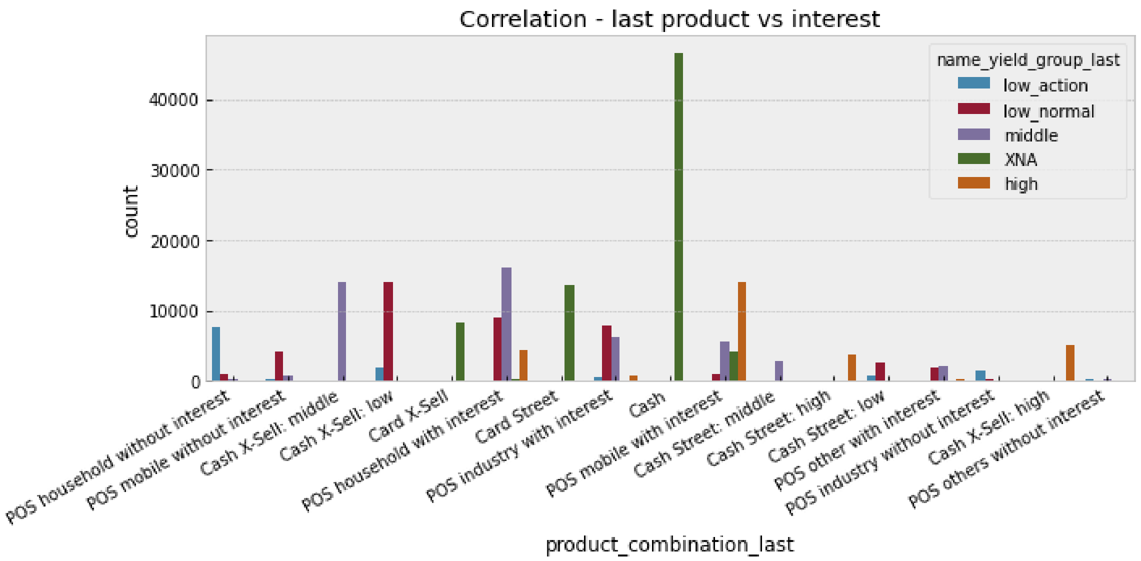


	feature_1	feature_2	cramers_v
309	name_income_type	flag_emp_phone	99.98
599	flag_emp_phone	organization_type	99.98
888	region_rating_client	region_rating_client_w_city	95.73
1070	reg_region_not_work_region	live_region_not_work_region	86.18
1973	name_yield_group_last	product_combination_last	82.96

What is the relationship between work phone & income source ?



Pensioners did not provide employer provided work phone numbers in credit applications. Similar logic applies to organization type.
Region rating & region rating with the city are providing almost the same information.
Interesting correlation is between last applied product & interest offered. Are products offered based on risks ?



Each product has one dominating interest type.

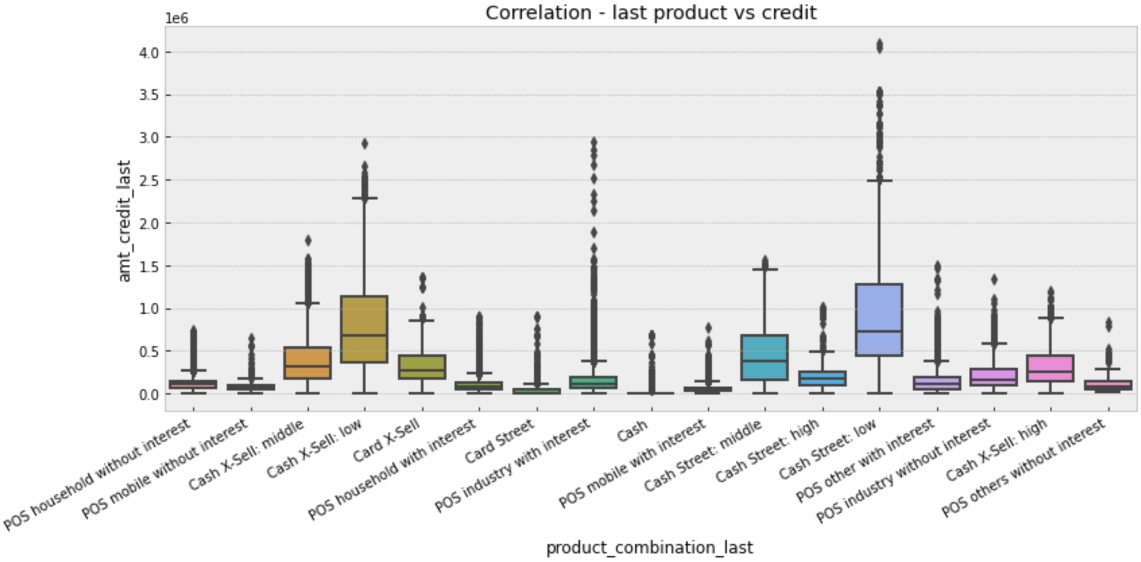
Based on the above analysis, we will **drop** the employer provided phone flag & region rating features.

Finally, using correlation ratio, we will find correlation between continuous & categorical features.

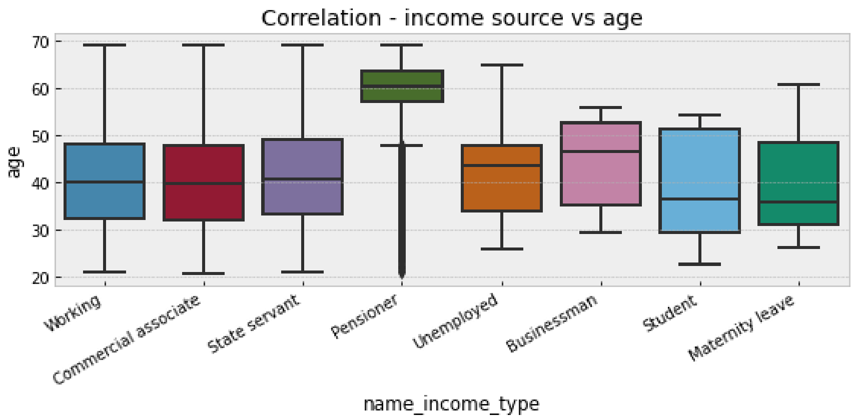
3. Continuous & Categorical features

Top 5 correlation pair are below

Let us explore correlation between last product and credit amount.



Median POS loan is a small amount loan whereas median Cash : low loan is a much high amount loan.



As expected, the median age of pensioners is higher and the median age of students is lower. Median age of businessmen is higher than the median age of working and commercial associates.

We are not dropping any features based on the above analysis.

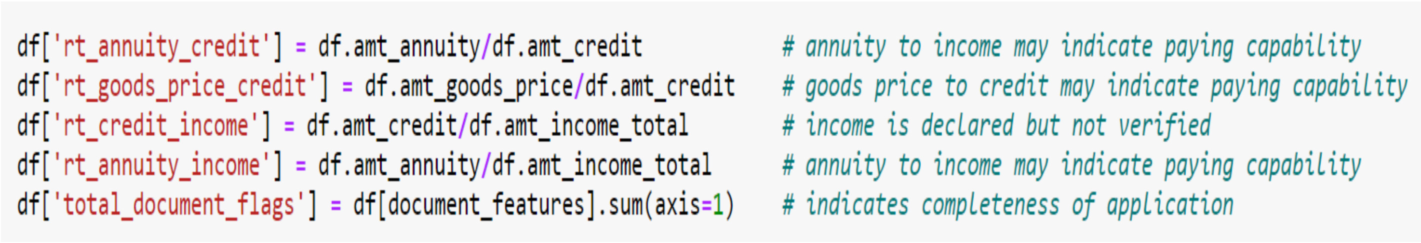
Finally, after dropping features based on high correlations, missing values and distributions, we have **177 features.**

3. Feature Engineering

Based on EDA analysis and common sense, we will create additional features which might be useful in predicting credit default. We have created new features in Data Wrangling when we combined Home Credit Historical data and Bureau data with current application data. This notebook focuses on adding more features based on EDA analysis and current application data.

1. Application details based features

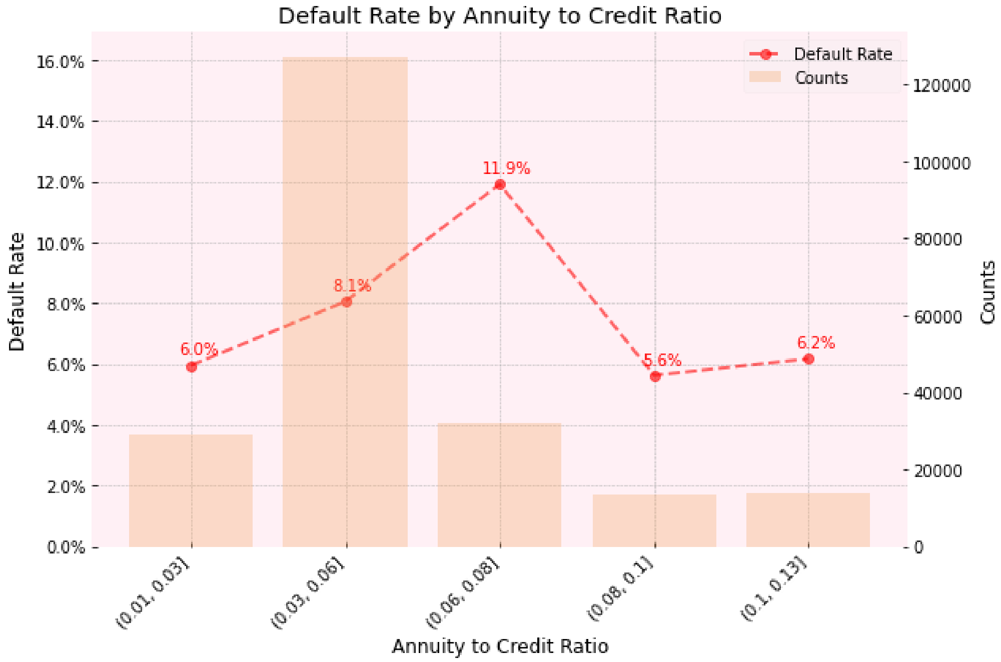
We have created the ratio features based on application details. Below is a code snippet for the same.



```
df['rt_annuity_credit'] = df.amt_annuity/df.amt_credit          # annuity to income may indicate paying capability
df['rt_goods_price_credit'] = df.amt_goods_price/df.amt_credit  # goods price to credit may indicate paying capability
df['rt_credit_income'] = df.amt_credit/df.amt_income_total      # income is declared but not verified
df['rt_annuity_income'] = df.amt_annuity/df.amt_income_total    # annuity to income may indicate paying capability
df['total_document_flags'] = df[document_features].sum(axis=1)  # indicates completeness of application
```

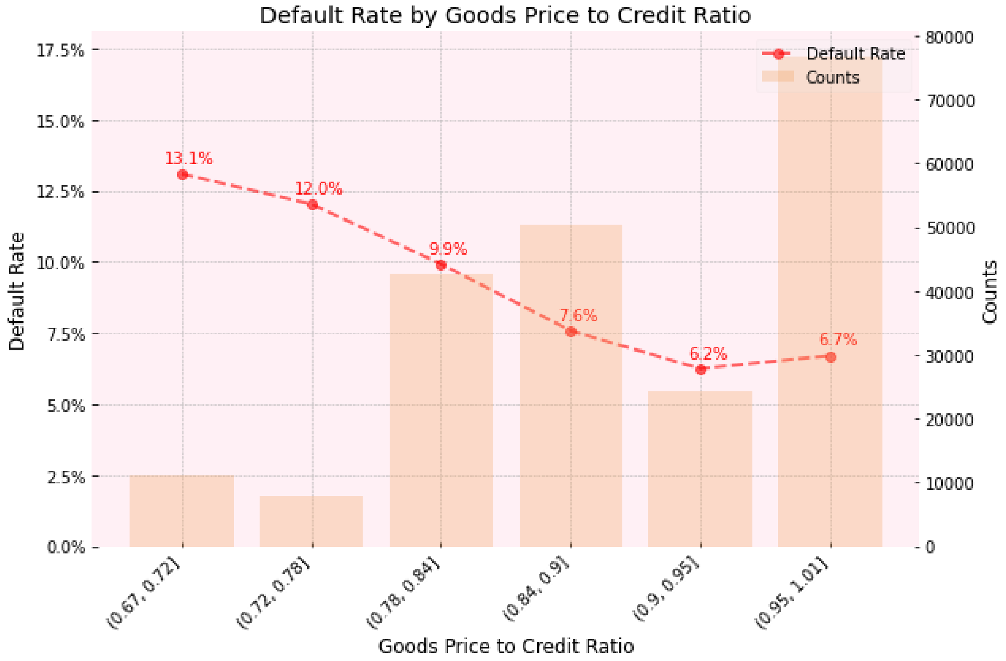
Let us explore a couple of them.

1. Annuity to Credit Amount ratio



This feature is a ratio of annuity to credit amount in current application. Interestingly, we have lower default rate at tails and higher default rate in the middle.

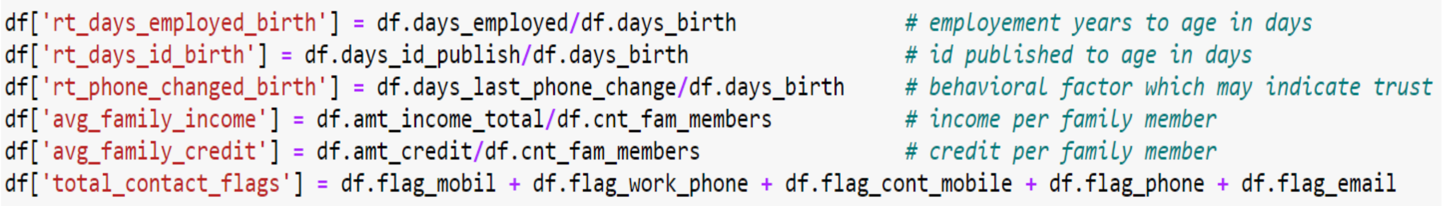
2. Goods Price to Credit Amount ratio



Default rate decreases with increase in goods price to credit ratio. This may indicate that underwriting practices are strong where higher risks are given lower loan amounts(as % of goods price). We do not have a bifurcation of credit into different components (eg insurance costs, downpayments, processing fees etc). This bifurcation may help in interpretation.

2. Applicant details based features

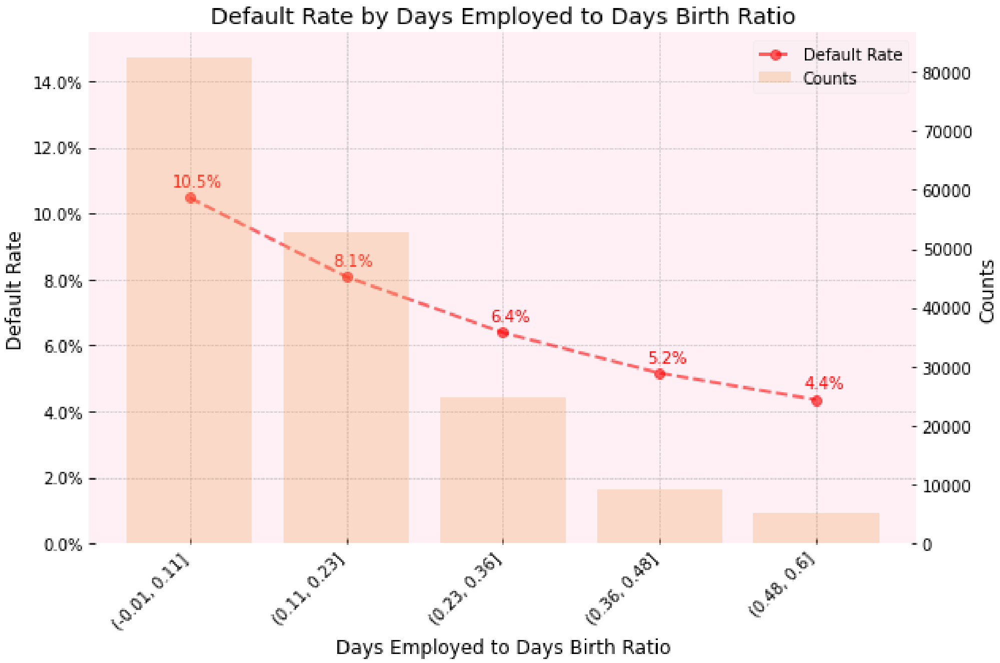
Similarly, we have created ratio features based on applicant details as listed below.



```
df['rt_days_employed_birth'] = df.days_employed/df.days_birth              # employement years to age in days
df['rt_days_id_birth'] = df.days_id_publish/df.days_birth                  # id published to age in days
df['rt_phone_changed_birth'] = df.days_last_phone_change/df.days_birth     # behavioral factor which may indicate trust
df['avg_family_income'] = df.amt_income_total/df.cnt_fam_members           # income per family member
df['avg_family_credit'] = df.amt_credit/df.cnt_fam_members                 # credit per family member
df['total_contact_flags'] = df.flag_mobil + df.flag_work_phone + df.flag_cont_mobile + df.flag_phone + df.flag_email
```

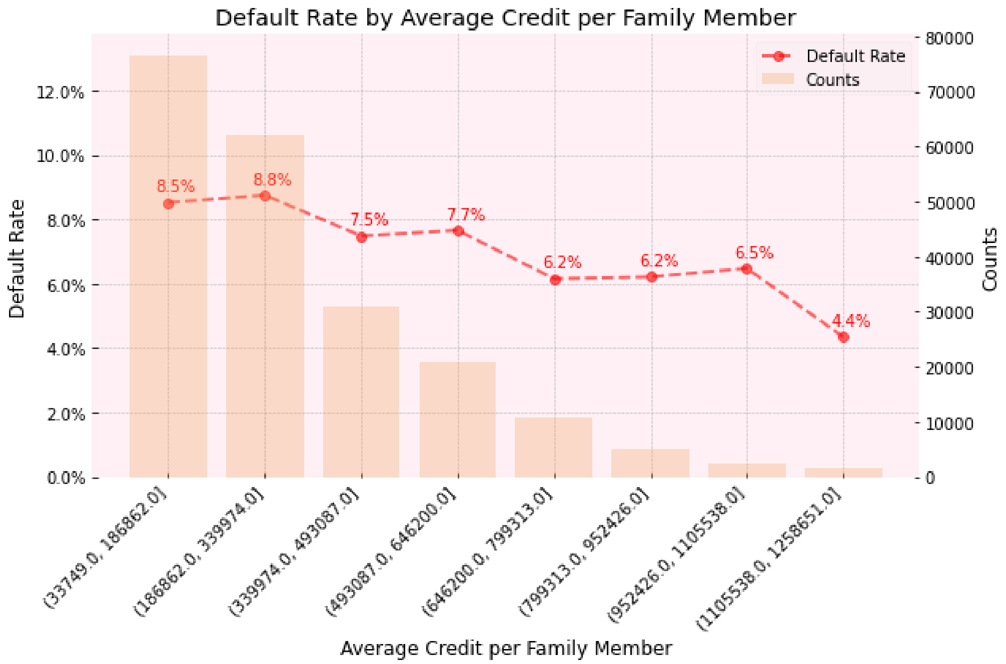
Let us explore a couple of them.

1. Days Employed to Days Birth ratio



Higher the number of days someone is in their current job as a proportion of age, lower the default rate.

2. Credit Amount per Family Member



Smaller credit per family member indicates higher default rate. This may be because of higher expenses incurred in bigger family sizes. Finally, we have **188 features** for the modelling stage.

4. Modeling

In the modelling stage, we will select an appropriate **evaluation matrix**, devise our **modelling and validation strategy** and select the **best model**. The Modelling Notebook can be referred for details.

1. Evaluation Matrix

For Home Credit, it is equally important to

1. Avoid customers who are likely to default on the loan and
2. Increase customers who are likely to pay back the loan

Given this business objective, we will be using the **roc_auc** matrix as it considers both the classes (default/no default). The pr_auc matrix mainly cares about positive class(default). Hence, we will avoid it. Other important considerations in model evaluation will be **prediction time** and **training time** in that order.

2. Modelling Strategy

HomeCredit is more interested in accurate predictions than interpretations. Hence, we consider following algorithms for our classification task.

1. **Logistic Regression with l1 regularization** - Logistic regression is a simple linear algorithm with decent performance. As we have 180+ features and many are correlated, we will use **L1 regularization** for feature selection. This will also be our base model.
2. **XgBoost** - XgBoost is known to outperform linear models on tabular data in many cases.
3. **LightGBM** - LightGBM is faster than XgBoost and tends to perform well on tabular data.

Logistic regression with manual/automatic feature selection resulting in parameter coefficients would be a better choice if transparency and simplicity was the preference.

Handling missing values

As we do not have business insights, we will avoid imputing missing values as much as possible.
1. **Categorical** features - Create a seperate nan category/level for missing value. We will do this at the time of one-hot-encoding using pandas get_dummy function.
2. **Numerical** features - For logistic regression, we will impute missing values with median because numerical features are highly skewed as observed in EDA. XgBoost & LightGBM can handle missing in a supervised way which optimizes model performance. Hence, we will not impute numerical features for these two algorithms.

3. Validation Strategy

We will use a **5-fold cross-validation** strategy on train data for model tuning and model selection. We will train these 3 models on the same set train-validation datasets. To select the best hyperparameters, we will use **RandomSearchCV** technique. And we will compare the performance of the best model of each of the three algorithms. Best of the bests will be our final model.
And finally, we will test the performance of the final model on the test dataset. This will give us a better idea of how well the model performs on unseen data.

s4. Best Model

1. Logistic Regression

We will use median imputation to fill missing values for numerical features using column transformers. As we are using l1 regularization and for greater convergence speed, we will scale these features using StandardScaler. We will tune the C (regularization) parameter using RandomSearchCV with 10 iterations. The range for C is

param_dists = { 'classifier_lr__C': loguniform(1e-2, 1e2) }

For model convergence and faster execution, tolerance level was changed to 0.01.

2. XgBoost Classifier

As XgBoost has more hyperparameters to tune, we will use 50 iterations in RandomSearch cross-validation. I have enabled GPU support for XgBoost training. **50 iterations are chosen considering the time and hardware availability.** Hyperparameter tuning range is

```
xgb_param_dists = {'learning_rate': loguniform(0.001, 1),
          'max_depth' : [1,4,8,16],
          'min_child_weight' : [1,4,8],
          'colsample_bytree' : [0.6,0.8],
          'subsample' : [0.6,0.8],
          'n_estimators' : [200,400,800,1600,6400],
          'reg_alpha' : loguniform(0.001, 10) }
```

3. Light GBM Classifier

Similar to XgBoost, LightGBM has more hyperparameters to tune. Hence, we will use 50 iterations in RandomSearch cross-validation. LightGBM is faster on multicore CPUs and enabling GPU support for LightGBM was challenging and unsuccessful. Hyperparameter tuning range is

```
lgb_param_dists = {
          'learning_rate': loguniform(0.01, 1),
          'n_estimators': [200,400,800,1600,6400],
          'num_leaves': [4,8,16,50],
          'min_child_weight' : [1,4,8,16],
          'colsample_bytree' : [0.6,0.8],
          'subsample' : [0.6,0.8,1],
          'reg_alpha': loguniform(0.001, 1)}
```

4. Final Model

Below table compares the best hypertuned model of each algorithm.

Best Model from RandomSearchCV	Iterations	Mean roc_auc score	Standard deviation of roc_auc score	Prediction Time*	Training Time
Logistic Regression	10	0.7643	0.0039	NA**	8min 35s

XgBoost	**50**	**0.7826**	**0.0047**	**20.4s**	**14h 6min 10s**
Light GBM	50	0.7814	0.0049	36.8s	1h 50min 49s

* Prediction time is calculated on the train dataset.

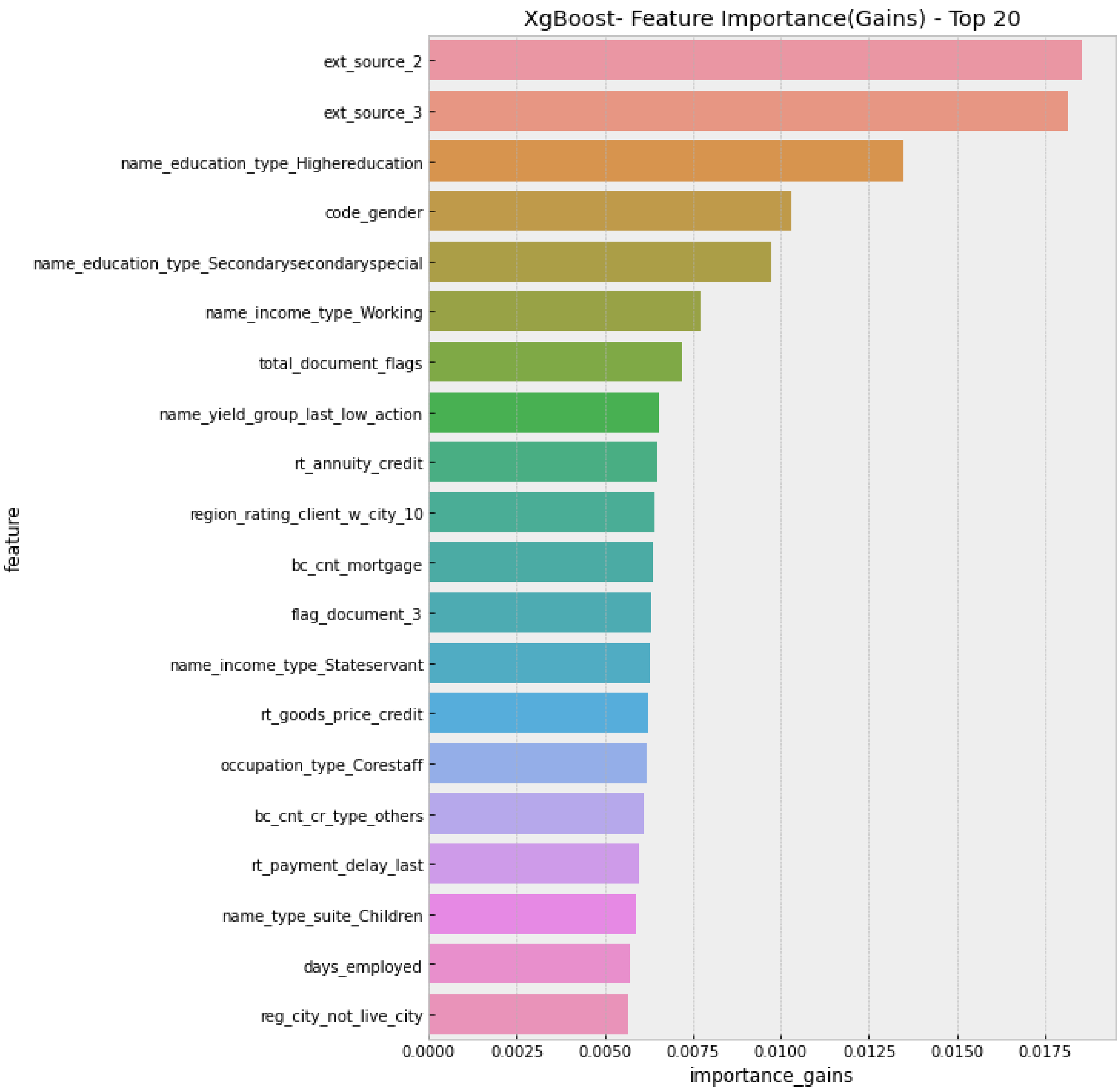
** As roc_auc score was not within 3 standard deviations of XgBoost model, it was not evaluated.

While XgBoost is slightly better than Light GBM, roc_auc scores of both the models are within 1 standard deviation of each other. XgBoost prediction is 1.8 times faster than LightGBM prediction. LightGBM with 5 CPU cores is 7 times faster than XgBoost and 6GB GPU. **Hence, we pick XgBoost as our final model.** The parameters of final model are

```
xgb_final_params = {'learning_rate': 0.02332 ,
        'max_depth' : 8,
        'min_child_weight' : 1,
        'colsample_bytree' : 0.6,
        'subsample' : 0.8,
        'n_estimators' : 800,
        'reg_alpha' : 8.6410 }
```

1. Feature Importance

We will be using Gains based feature importance and below is a list of top 20 features.



External feature 2 & 3 are the top two features. In EDA, we had observed that these features were exhibiting a strong downward trend with excellent risk differentiation(~ 24% to 2%). Education and gender are also important predictors of default rate. Interestingly, 3 feature engineered ratio factors are among the top 20. These are annuity to credit, goods price to credit and payment delay to duration for last loan. We also have 2 features from bureau data. They are counts of mortgage and others credit types. It is assuring that we observed trends in many of the top 20 features during EDA.

2. Performance on Test Data

Now we evaluate the model on test data. We get an AUC of **0.7841** which is slightly higher than **0.7814** mean AUC of cross-validation but well within the range **(0.7861, 0.7767)** of 1 standard deviation. The XgBoost model has been well generalized on the test set.

We will select the appropriate threshold and evaluate the business impact in the next chapter.

5. Business Impact

Assume that HomeCredit wants to **maximize profit** as a business objective and we have been given following parameters from business.

1. **5% of credit as a final profit** on loans which are fully paid - this is the average % profit after removing commission, expenses, cost of capital etc.
2. **40% of credit as a loss** on loans which are defaulted - this is the average % loss after considering outstanding amount, recoveries etc. 90% of loans are cash loans and losses tend to be higher on cash loans.

The details of business impact analysis are towards the end of the **Modelling Notebook**.

With this in mind, let us optimize the threshold for maximum profit and calculate the confusion matrix. Now we will:

1. Calculate average loan size for paid and default loans.
2. Simulate different scenarios and plot profitability curves.
3. Select the best point that maximizes profit.

Average loan size for a paid loan (mean_credit_paid) = 601969.81.
Average loan size for a default loan (mean_credit_default) = 554235.12

We get the above average loan sizes. We have
tn - true negative (paid loans which are predicted as paid)
tp - true positives (default loans which are predicted as defaults)
fp - false positive (paid loans which are predicted as default)
fn - false negative (default loans which are predicted as paid)

Now we define the formula for expectancy as
expectancy =

+ tn * mean_credit_paid * profit_on_paid_loans (profit from fully paid loans)

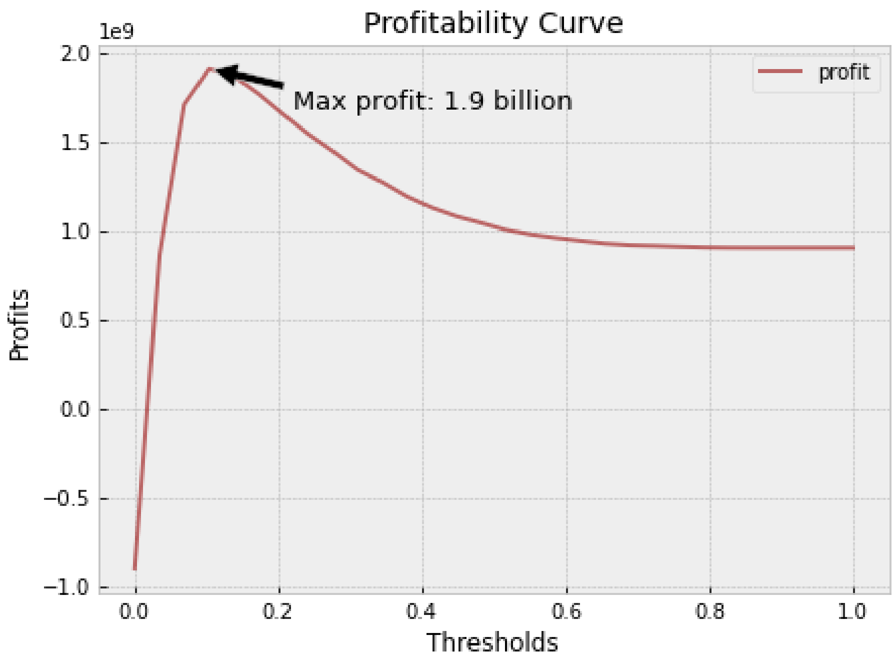
+ tp * mean_credit_default * loss_on_default_loans (notional profit from avoiding defaults)

- fp * mean_credit_paid * profit_on_paid_loans (notional loss from avoiding paid back loans)

- fn * mean_credit_default * loss_on_default_loans (loss from defaults on loans)

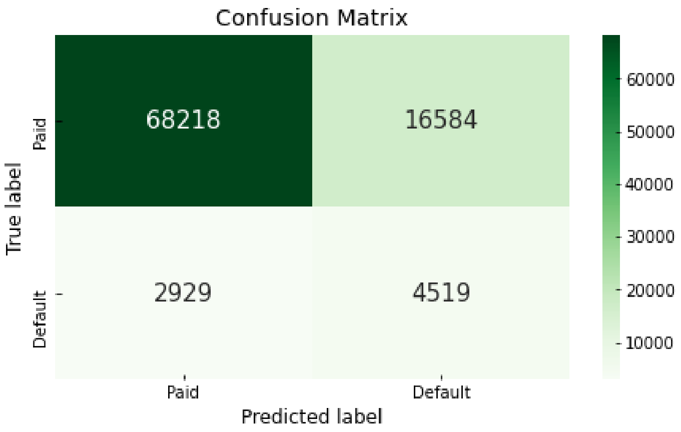
1. Profitability Curve

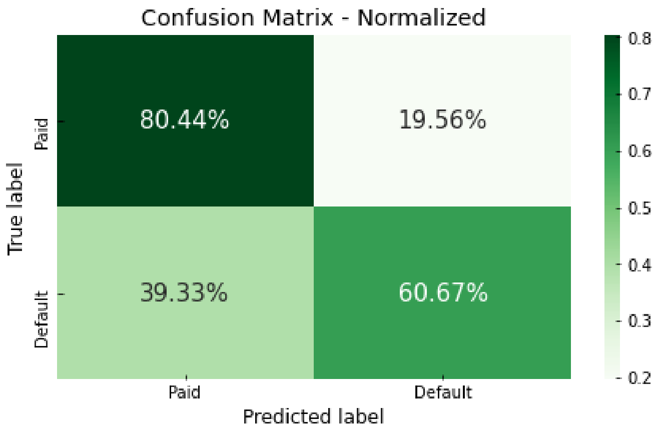
After simulating various thresholds for expectancy, we get **0.1034** as profit maximizing threshold with 1.9 Billion as maximum profit based on test data (30%). The profitability curve is plotted below.



2. Confusion Matrix

The confusion matrix for the profit maximizing threshold is





*Normalization is done as a % of true labels.

With a 0.1034 threshold, we provide loans to 71147 (77.12%) applications. So, we will give loans to 80.44% of the paying applicants and accommodate 39.33% of default applicants. This essentially translated into a **book profit of 3.29%** (as a % of total credit extended).

6. Next Steps

Following ideas can be explored in the future

1. Modelling applicants with and without credit card separately - We have just about 28% applicants who had credit cards with HomeCredit but the records are credible(65k). This leads to missing values in most of the records. Also, applicants without credit cards have a 1.2% higher default rate. Separating these two groups may lead to better accuracy.
2. Bayesian Hyperparameter Optimization - We can explore Bayesian Hyperparameter Optimization which is a targeted search strategy based on model improvements. This may lead to improved tuning at a lesser time.
3. Model Explainability - HomeCredit is more focused on prediction. With model Interpretation/Explainability analysis, we can demonstrate the impact of the individual feature for easy buy-in of the model by business.