Data cleaning and pre-processing

Denis Tsvetkov and Lukas Ostrovskis

January 2023

1 Filtering on loan status

We consider loans with the status *Current*, *Late (31-120 days)*, *In Grace Period* and *Late (16-30 days)* to be still running and thus not usable for model training, so we remove them.
While *Charged Off* is the most interesting status for loans that will not be repaid, *Default* also indicates a very high probability that a loan will not be repaid so we encode both these statuses as 0 ("not repaid"). The final status is *Fully Paid*, we encode that as 1 ("has been repaid").

2 Preventing leakage

We performed an extensive search in the provided dictionary for features that are not known at the time of applying for a loan and we found a total of 6 that were not removed yet. Namely, the features:

- **issue_d** - The month which the loan was funded;
- **last_credit_pull_d** - The most recent month LC pulled credit for this loan;
- **next_pymnt_d** - Next scheduled payment date - not known as it follows from **issue_d**, however, we decided not to drop it here as it is full of NaNs and it will be filtered out later;
- **last_pymnt_d** - Last month payment was received;
- **last_pymnt_amnt** - Last total payment amount received;
- **debt_settlement_flag** - Flags whether or not the borrower, who has charged-off, is working with a debt-settlement company.

It is clear that all of the features above are related to still ongoing (*in limbo*) loans and thus cannot be used for training our machine learning algorithms.

3 Dropping features of no or little predictive value

The feature set contains multiple features that will have little predictive value to the models we will be training. Below you can find the list of features we deemed of little value, their definition, and our reason for dropping them:

Feature	Definition	Reason
id	A unique LC assigned ID for the loan listing.	Unique id for each data point is not useful for model training.
member_id	A unique LC assigned ID for the borrower member.	Unique id for each data point is not useful for model training.

disimbursement_method	Method for receiving the loan (e.g. in cash or via bank transfer).	Receiving the loan in cash vs via a bank transfer should not have a major influence on repayment status.
url	URL for the LC page with listing data.	URLs are unique and don't provide useful information.
desc	Loan description provided by the borrower.	Loan descriptions can vary too much and be too unique to provide useful information during training (It also overlaps more or less with **title**).
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.	First 3 digits of a US zip code designate a city or a larger rural area. That's essentially equivalent to the **addr_state** feature.
emp_title	The job title supplied by the Borrower when applying for the loan.	There are too many unique job titles in the data for it to be a useful feature for model training.
bc_open_to_buy	Total open to buy on revolving bankcards.	There are too many unique values in the data for it to be a useful feature for model training.
max_bal_bc	Maximum current balance owed on all revolving accounts.	There are too many unique balance values in the data for it to be a useful feature for model training.
mths_since_recent_bc	Months since most recent bankcard account opened.	We don't think this has a huge influence on the outcome of a loan status.
grade	LC assigned loan grade.	This feature is already covered by the **sub_grade** feature.
title	The loan title provided by the borrower.	This feature has values that are equivalent to the **purpose** feature and more missing values.

Table 1: Features of no or little predictive value

4 Unbalanced features

After going over the values for each of the remaining features, we identified 7 "biased" ones with above 95% of the instances having the same values. In table 2, under the *Values - counts* column, we provide the 2 most common values with the number of their occurrences (note that the number of loans left in our data is 146775). These features can be split into 2 categories: they either have too many identical values (all but one or two instances), or the loan statuses for the most common dominated value (for example *Joint App* for **application_type**) are split the same way as the whole data (roughly 77% to 23% in favour of the paid out loans). In both of these cases, the features will not have any beneficial impact on our future models, hence we decided to drop all of them to further simply our data.

Feature	Values - counts
application_type	Individual - 144062 Joint App - 2713
policy_code	1.0 - 146775 2.0 - 0
out_prncp	0.00 - 146767 14928.28 - 1
collections_12_mths_ex_med	0.00 - 143860 1.0 - 2685
chargeoff_within_12_mths	0.00 - 145535 1.0 - 1138
tax_liens	0.00 - 140437 1.0 - 4327
hardship_flag	N - 146775 Y - 0

Table 2: Unbalanced features with more than 95% of identical values

5 Highly correlated features

We were not sure how many of the correlated features to drop, as it is not always a bad thing to have them, especially if they are correlated towards the target [1]. Therefore, we opted for a high correlation coefficient of 0.8 to only drop a couple features to reduce the dimensionality of the data and increase the speed of our future models. To identify the highly correlated remaining features we generated a correlation matrix. The list of feature pairs that we identified, in addition to the appropriate resolution method taken can be found in table 3.

Feature Pair	Correlation Coefficient	Method of Resolution
open_acc - num_sats	1	Drop open_acc
fico_range_low - fico_range_high	1	Combine into fico_range_avg
num_actv_rev_tl - num_rev_tl_bal_gt_0	0.98	Drop num_rev_tl_bal_gt_0
tot_cur_bal - tot_hi_cred_lim	0.96	Drop tot_hi_cred_lim
loan_amnt - installment	0.96	Drop installment
mo_sin_old_rev_tl_op - earliest_cr_line	0.91	Drop mo_sin_old_rev_tl_op
revol_util - bc_util	0.86	Drop revol_util
acc_open_past_24mnths - open_rv_24m	0.84	Drop open_rv_24m
open_acc - num_op_rev_tl	0.84	Drop num_op_rev_tl
bc_util - percent_bc_gt_75	0.84	Drop percent_bc_gt_75

Table 3: Highly correlated feature pairs

6 Getting rid of the remaining missing values

The data set contains a lot of variables with missing values. Features with more than 80% missing values may not be useful for model training at all so we have decided to completely remove them. The reason for this is the following: as the majority of the values are missing, it would be impossible

[1] https://datascience.stackexchange.com/questions/24452/in-supervised-learning-why-is-it-bad-to-have-correlated-features

to impute or predict them based on the information that we have, without introducing biases in the data (see Saar-Tsechansky). The list of features that we have identified to miss values for over 80% of the data points are the following:

Feature	Percentage of Missing Values
sec_app_fico_range_low	100.00
sec_app_open_act_il	100.00
sec_app_open_acc	100.00
sec_app_mort_acc	100.00
revol_bal_joint	100.00
sec_app_chargeoff_within_12_mths	100.00
sec_app_collections_12_mths_ex_med	100.00
sec_app_mths_since_last_major_derog	100.00
sec_app_num_rev_accts	100.00
sec_app_revol_util	100.00
sec_app_fico_range_high	100.00
sec_app_earliest_cr_line	100.00
sec_app_inq_last_6mths	100.00
next_pymnt_d	99.99
orig_projected_additional_accrued_interest	99.46
hardship_end_date	99.13
hardship_last_payment_amount	99.13
hardship_payoff_balance_amount	99.13
hardship_loan_status	99.13
hardship_dpd	99.13
hardship_length	99.13
hardship_status	99.13
hardship_start_date	99.13
hardship_amount	99.13
deferral_term	99.13
hardship_reason	99.13
hardship_type	99.13
payment_plan_start_date	99.13
dti_joint	98.15
verification_status_joint	98.15
annual_inc_joint	98.15
debt_settlement_flag_date	96.21
settlement_status	96.21
settlement_date	96.21
settlement_amount	96.21
settlement_percentage	96.21
settlement_term	96.21

Table 4: Features with too many missing values

The rest of the features have a significantly lower percentage of missing values (13% and below) so we cannot justify dropping them based on that. These features might be useful during model training, so we decided to replace the missing values for the 4 features mentioned in the table below and drop the rows (instead of the columns) for the ones that we could not easily replace (because they would cause changes in the covariance and the correlation between the features). Dropping the rows in this case will not affect the future models, as the missing data is at least MAR (missing at random), based on the frequency of the loan states of it compared to the whole data set.

Feature	Number of Missing Values	Replacement value
il_util	19132	71.8982 (mean)
emp_length	9486	$<$ 1 year
mths_since_rcnt_il	3760	0.0
mo_sin_old_il_acct	3730	0.0

Table 5: Features with missing values and the replacement value we used

And our reasoning for the replacement values is summarised in the list below:

- **il_util** - for this feature we decided to use mean imputation as in most cases it produces unbiased estimates for MACR (missing completely at random) data (see Enders, 2010, ch. 9, p. 177), which is what we have as the frequency for the loan statuses of the missing values is the same of the whole data. Furthermore, the values for the data are not skewed at all as most of them are in the range of 65 to 90;

- **emp_length** - we figured that this field would have been left out empty if the person who was applying for a loan was unemployed at that moment, so in that case an employment length of $<$ 1 year made sense;

- **mths_since_rcnt_il** - as this feature corresponds to the the number of months since the most recent installment accounts were opened, we decided to map its missing values to -1, as it indicated to us that maybe that person simply did not have such accounts. The -1 value is quite different from the rest (as they are all non-negative), however, the NaN columns had the same frequency for the loan statuses as the rest of the data. Therefore, we concluded that it will not make our model biased in some way;

- **mo_sin_old_il_acct** - same reason as for the previous feature.

7 Dates

The only feature that represents a date that has not been dropped yet is **earliest_cr_line**.
We decided that it is important to our model so instead of dropping we converted it into a numerical feature. For that we used a mapping that extracts the year integer value of a feature (e.g. "Sep-2002" into 2002) and subtracts it from the latest year value in the data set (2013 in our case).

References

[1] Enders, C. K. (2010). Applied Missing Data Analysis. Methodology In The Social Sciences. (Chapter 9, p. 177)

[2] Saar-Tsechansky, M., Provost, F. (2007). Handling Missing Values when Applying Classification Models. Journal of Machine Learning Research 8, 1625-1657