Peer Lending Project

Stage 3 - Data Preparation/Modeling

Group C

In this stage we prepared our data for modeling using methods to deal with missing values, outliers and encoding of categorical variables as well as checking the option of scaling the data. We started creating our models for proof of concept.

The steps we took:

1. Changes from the previous stage:

a. We returned two attributes that we dropped due to irrelevance - “earliest_cr_line” and “issue_d”, we used those to create a new attribute “cr_hist” - the length of a person's credit history at the time the loan is issued. We will check if the new attribute improves our model.
b. We decided to drop the “emp_title” attribute since it has too many unique values (approx 80,000) and will not help our model.

2. Dealing with missing values:

We treated the attributes by their N/A Fraction:

- Less than 1% - dropped those N/As, since it's negligent.
- More than 1% - the traditional approach : replacing with the median, which is not biased by the values at the far end of the distribution, therefore, the median is a good representation of the majority of the values in the variable.

We've decided on those methods in order not to lose many instances in our initial modeling stage. In the next step we will try to use a KNN-imputation to replace missing values (also the ones we dropped for now) and see if it improves our model (See Appendix 1 for the N/A fractions).

3. Dealing with categorical attributes:

We had to change the categorical attributes since Machine Learning algorithms require that attributes will be numeric. We used few different methods to encoding those attributes:

a. “loan_status” & “initial_list_status” :

These attributes have only 2 possible values, therefore, they will be encoded with the values as "0" and "1" (Appendix 2.1).

b. Encoding of categorical attributes with more than 2 unique values:

- **Hierarchical attributes -** “grade”, “purpose” (after manipulation).
  Encoded by their order (Appendix 2.1, 2.6 for “grade” and Appendix 2.1, 2.3, 2.4, 2.5 for “purpose”) .
- **Non Hierarchical attributes** - “addr_state”, “home_ownership”,”verification_status”
  One-hot encoding: Some attributes had many possible values so we grouped them to decrease the number of values (Appendix 2.1 and Appendix 2.2 for “home_ownership”).

As we move forward if an encoded attribute will not contribute to our model, it will be dropped.

4. Dealing with outliers:

Before detecting outliers we want to mention that the distribution of the realized return between fully paid and defaulted loans - 99% of fully paid loans have realized return of more than 2% whereas only 5% of the defaulted loans yield more than 2%. This is worth mentioning since the company currently yields around 2% yearly and we want to detect the loans with a better return.

We understand that the majority of loans we want to detect and avoid are defaulted, with only 5% of them with a better realized return then the company generates now. This means that if we do a good job detecting fully paid loans, 99% of them will yield more than 2%.

We used both the empirical[1] and the IQR[2] rule to detect the absolute number and fraction of outliers for each attribute (Appendix 3.1).
We saw that when using the IQR rule we have much more outliers, hence removing outliers based on this criteria will result in a massive loss of data. Therefore, we will focus on the empirical rule results.

To understand the nature of the outliers better, we checked the division between fully paid and defaulted loans for every attribute's outliers (Appendix 3.2). Our target column - realized_return, has the highest number of outliers, however, all of them are defaulted loans - as we mentioned above these are the loans we want to detect, removing them will skew the data even more than it already is (only 13% defaulted). Therefore we will not ignore or replace these outliers.

Outliers are a sensitive subject. On one hand, they might bias our future model but on the other hand, they might contain relevant information. For now, we will not remove or change outliers, we will check each attribute's outliers separately, and treat them only if it improves our model before making any hasty decisions.

5. Scaling

Scaling the data is an important step since some models require the data to be scaled so it will not assign unproportional weights for attributes. For now we are testing our models with and without scaling, at this time we will use a standard scaler, when we proceed we will try different scaling methods and see if it improves our models.

6. Modeling:

Approach A - Regression model for predicting realized return:

We used linear regression and Gradient Boosting regression to predict the realized return for each loan. We then compared the results to our baseline model - investing only based on grade (appendix 4.1). Using grade we can choose loans that fit our desired risk. Loans graded 'A' has the highest average return and the lowest standard deviation - thus being the safest investment. In the second extreme, loans graded 'F' and 'G' offer average return close to zero but high standard deviation - meaning a chance for high return. Our goal is either to predict loans with the same realized return and lower standard deviation or a higher return with the same standard deviation.

Key evaluation metrics:

RMSE - it indicates the absolute fit of the model to the data – how close the observed data points are to the model's predicted values

R^2 - the proportion of the variance in the dependent variable that is predictable (explained) from the independent variables.

---

[1] 3 SD from the mean

[2] below Q1 − 1.5 IQR or above Q3 + 1.5 IQR

Both models performed poorly. The RMSE is too large, meaning that many loans with predicted positive return can have negative return, exactly the type of error we want to avoid - hence we can't trust the predictions. We will attempt to improve our model and consider dropping it if we can't reach better results.

Approach B - Classification model on loan status:

We used 3 different classification methods to determine whether a loan is fully paid or defaulted. We then created an expected return equation as follows:

$$E(loans) = P(fully\ paid) * E(return\ fully\ paid) + P(default) * E(return\ default)$$

Where:

$P(fully\ paid)$ - The proportion of loans predicted as fully paid that were actually fully paid

$P(default)$ - The proportion of loans predicted as fully paid that were actually default (1 - precision)

$E(return\ fully\ paid)$ - The average realized return of fully paid loans calculated from the training set

$E(return\ default)$ - The average realized return of defaulted loans calculated from the training set

In our equation, we multiply $P(fully\ paid)$ with the mean realized return of fully paid loans and deduct ("punish") $P(default)$ multiplied by the mean realized return of defaulted loans to estimate the average return for loans our model predicts as fully paid. Using this method, we don't predict the realized return per loan, rather we show the average expected realized return for investing in all loans our model predicts as fully paid. As we mentioned above, 99% of fully paid loans yield more than 2%. Thus, predicting correctly the maturity of the loans can yield a higher return. We decided to try this different approach since the regression models performed poorly, this might suggest that our data is not suitable for regression.

We will use cross validation and other metrics to raise our confidence in the estimate of the precision our model produces since it's crucial for our expected return equation. (Appendix 4.2)

Key evaluation metric:

Recall - The ratio of correct positive predictions to the total positives examples. Defaulted loans classified as fully paid generate losses where fully paid loans classified as default prevent gain. One way to increase recall is to lower the threshold for classifying loans as default, however this will also result in loss of fully paid loans with positive return. We will try different thresholds and choose the one with the highest return suitable to a specific budget constraint.

AUC- The Area Under the Curve is the measure of the ability of a classifier to distinguish between classes (loan status) better than a random model.

Appendices

Appendix 1 - missing values Fraction

Columns	Missing value fraction
total_bal_il	0.0166
max_bal_bc	0.0166
all_util	0.0226
revol_util	0.0667
realized_return	0.1514
bc_open_to_buy	1.1464
percent_bc_gt_75	1.1551
bc_util	1.1969
il_util	13.9307
emp_length	14.8805

Appendix 2: Dealing with Categorical attributes

Appendix 2.1 - Encoding description for categorical attributes

Attribute	Encoding Description
"loan_status"	Can get the values Fully Paid\Charged Off
"initial_list_status"	Can get the values w\f
"verification_status"	We will merge "Source Verified" and "Verified" to "Verified" since we believe they have the same meaning. we will encode it as 0 and 1
"grade"	Hierarchical attribute. Since the different grades are ordered where A is the highest, we will replace G to A with 1 to 7
'purpose'	The values "wedding" and "renewable_enery" appear 1 and 204 times respectively in our data. Since these won't be helpful for our model, we decided to remove the rows containing these values. For each of the remaining values, we checked their default rate as well as the distribution of their realized return. The results showed that the realized return distribution is more or less the same for each

	purpose while the default rate can be divided into three subgroups, “under 16.36%”, “between 16.36% and 19.5%” and “over 19.5%”. We decided to group the values into these 3 subgroups, labeled 1,2 and 3 (Appendix 2.3). After this process, “purpose” is a hierarchical attribute
“addr_state”	Contains values for 50 states so we will group the states to regions and use one hot encoding
“home_ownership”	Has 4 unique values, the value "ANY" has 58 inctenses, it is negligent so we will drop these instances and use one hot encoding for the column
“sub_grade”	There is an overlap of information with the rankings column. In addition it contains 35 different variables.We will currently drop the column and return it if necessary according to the model's performance.

Appendix 2.2 - Home Ownership Attribute values count:

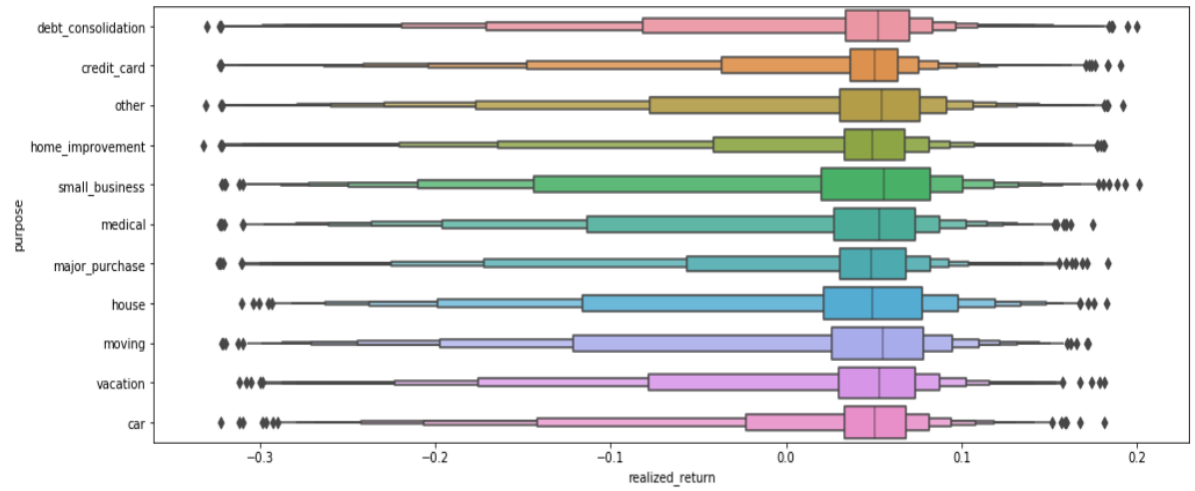
Home ownership	Value count
ANY	58
MORTGAGE	121648
OWN	33539
RENT	109574

Appendix 2.3 - Purpose Attribute values count:

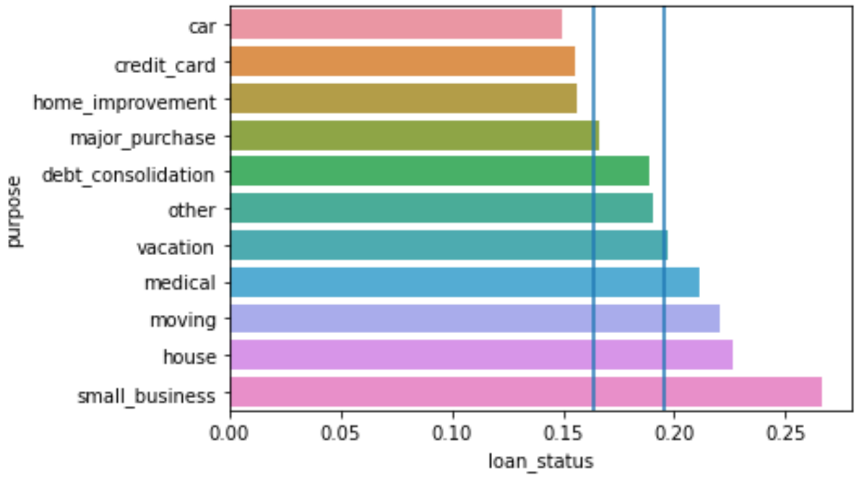
Purpose	Value count
car	3305
credit_card	57098
debt_consolidation	146266
home_improvement	19116
house	1259
major_purchase	6749
medical	3618
moving	2328
other	19380
Renewable_energy	204
small_business	3029
vacation	2408
wedding	1

Appendix 2.4 - Realized return distribution for each purpose

We can see the realized return distribution is more or a less the same for each purpose.



Appendix 2.5 - Loan status for each purpose



car, credit_card, home_improvement, major_purchase → encode 1

debt_consolidation,other,vacation → encode 2

medical,moving,house,small_business → encode 3

Appendix 2.6 - Grades realized return distribution

Grade	Mean	Median	Std	Min	Max
G	-0.0039	0.0332	0.1541	-0.3197	0.1942
F	0.0081	0.0579	0.1453	-0.3305	0.2010
E	0.0158	0.0677	0.1306	-0.3215	0.1813
D	0.0189	0.0683	0.1138	-0.3214	0.1575
C	0.0248	0.0659	0.0936	-0.3319	0.1301
B	0.0301	0.0558	0.0714	-0.3303	0.1222
A	0.0320	0.0410	0.0461	-0.3227	0.1007

Appendix 3 - Outliers

Appendix 3.1: Method comparison

Columns	IQR Outliers	Prop IQR Outliers	Gaussian Outliers	Prop Gauss Outliers
realized_return	44529	0.168315971	7546	0.028523262
bc_open_to_buy	22389	0.084628585	5726	0.021643811
pct_tl_nvr_dlq	15417	0.058274997	5161	0.019508157

total_bc_limit	16660	0.062973435	5147	0.019455238
total_bal_il	16640	0.062897836	5042	0.019058347
total_bal_ex_mort	15864	0.05996462	4876	0.01843088
num_bc_sats	13071	0.049407309	4785	0.018086908
avg_cur_bal	15722	0.059427872	4765	0.01801131
mort_acc	9679	0.036585827	4761	0.01799619
total_il_high_credit_limit	13743	0.051947414	4655	0.017595519
delinq_2yrs	55811	0.210961006	4327	0.016355705
total_rev_hi_lim	16444	0.062156972	4218	0.015943694
max_bal_bc	16918	0.063948654	4125	0.015592162
fico_range_high	9096	0.034382135	4090	0.015459865
fico_range_low	9096	0.034382135	4090	0.015459865
tot_cur_bal	11104	0.04197221	4068	0.015376707
num_actv_bc_tl	6088	0.023012141	3812	0.014409048
tot_hi_cred_lim	10836	0.040959192	3709	0.014019716
num_bc_tl	6121	0.023136878	3702	0.013993257
pub_rec	51963	0.196415882	3676	0.013894979
num_rev_tl_bal_gt_0	7165	0.027083113	3672	0.013879859

open_acc	6849	0.025888659	3661	0.01383828
revol_bal	18613	0.070355615	3648	0.013789141
num_op_rev_tl	12068	0.045616051	3643	0.013770241
num_sats	10208	0.038585403	3576	0.013516987
num_rev_accts	7244	0.027381726	3493	0.013203254
il_util	11673	0.044122983	3299	0.01246995
pub_rec_bankruptcies	35323	0.133518045	2841	0.010738747
int_rate	7003	0.026470766	2758	0.010425014
grade	11279	0.042633696	2575	0.009733289
cr_hist	8087	0.030568197	2525	0.009544293
tot_coll_amt	47489	0.179504528	2455	0.009279699
installment	9675	0.036570707	1954	0.00738596
annual_inc	13103	0.049528266	1881	0.007110026
loan_amnt	10069	0.038059995	1555	0.005877773
dti	906	0.003424606	729	0.00275556
all_util	1599	0.006044089	327	0.001236033
bc_util	18	6.80E-05	23	8.69E-05
revol_util	19	7.18E-05	19	7.18E-05

RENT	0	0	0	0
Southwest	33017	0.124801554	0	0
South East	63249	0.239076037	0	0
North East	60317	0.227993317	0	0
Midwest	47919	0.181129893	0	0
1	0	0	0	0
0	0	0	0	0
MORTGAGE	0	0	0	0
OWN	33517	0.126691513	0	0
percent_bc_gt_75	0	0	0	0
emp_length	0	0	0	0
loan_status	47310	0.178827923	0	0
purpose	0	0	0	0
initial_list_status	0	0	0	0
West	60054	0.226999199	0	0

Appendix 3.2: Outliers ratio of fully paid/default

Columns	Fully_paid	Default	frac
realized_return	0	7546	1

int_rate	1482	1276	0.462654097
grade	1411	1164	0.452038835
installment	1348	606	0.31013306
all_util	236	91	0.278287462
dti	534	195	0.267489712
revol_util	14	5	0.263157895
num_rev_tl_bal_gt_0	2798	874	0.238017429
pub_rec_bankruptcies	2212	629	0.221400915
num_actv_bc_tl	2982	830	0.217733473
bc_util	18	5	0.217391304
delinq_2yrs	3393	934	0.21585394
num_op_rev_tl	2886	757	0.207795773
pub_rec	2926	750	0.204026115
num_sats	2867	709	0.198266219
open_acc	2936	725	0.198033324
pct_tl_nvr_dlq	4156	1005	0.194729704
il_util	2668	631	0.191270082
cr_hist	2044	481	0.19049505

num_bc_sats	3899	886	0.185161964
num_rev_accts	2854	639	0.182937303
num_bc_tl	3041	661	0.178552134
total_bal_il	4242	800	0.158667196
tot_coll_amt	2069	386	0.157230143
total_il_high_credit_limit	3972	683	0.146723953
total_bal_ex_mort	4161	715	0.146636587
loan_amnt	1349	206	0.132475884
max_bal_bc	3651	474	0.114909091
annual_inc	1669	212	0.112706007
revol_bal	3238	410	0.112390351
mort_acc	4228	533	0.111951271
tot_cur_bal	3648	420	0.103244838
tot_hi_cred_lim	3339	370	0.099757347
total_rev_hi_lim	3798	420	0.099573257
avg_cur_bal	4317	448	0.094018888
total_bc_limit	4673	474	0.092092481
bc_open_to_buy	5324	402	0.070206078

fico_range_low	3887	203	0.049633252
fico_range_high	3887	203	0.049633252

Appendix 4 -Modeling

4.1 Approach A – Regression model on realized return

MODEL	R Squared	RMSE
Baseline model linear regression only by grade	0.75327	0.04130
Linear Regression	0.01189	0.08282
KNN	0.18461	0.09068
Gradient Boosting Regressor	0.02078	0.08244
Linear Regression - scaled	0.01188	0.08282
KNN - scaled	0.16875	0.09007
Gradient Boosting Regressor - scaled	0.02076	0.08245

4.2 Approach B – classification model on loan status

Baseline model - predicting by grade: confusion matrix

	Fully Paid	Charged Off
Fully Paid	43424	26
Charged Off	9455	7

Model recall: 0.00074

Model precision score: 0.21212

Model roc_auc_score: 0.50007

Model accuracy score: 0.82082

Logistic regression: confusion matrix

	Fully Paid	Charged Off
Fully Paid	43432	18
Charged Off	9460	2

Model recall: 0.00021

Model precision score: 0.1000

Model roc_auc_score: 0.49990

Model accuracy score: 0.82087

KNN: confusion matrix

	Fully Paid	Charged Off
Fully Paid	41299	2151
Charged Off	8820	642

Model recall: 0.06785

Model precision score: 0.22986

Model roc_auc_score: 0.50917

Model accuracy score: 0.79266

Random forest: confusion matrix

	Fully Paid	Charged Off
Fully Paid	43216	234
Charged Off	9222	240

Model recall: 0.02536

Model precision score: 0.50633

Model roc_auc_score: 0.50999

Model accuracy score: 0.82129

Logistic regression scaled: confusion matrix

	Fully Paid	Charged Off
Fully Paid	43103	347
Charged Off	9116	346

Model recall: 0.03657

Model precision score: 0.49928

Model roc_auc_score: 0.51429

Model accuracy score: 0.82116

KNN scaled: confusion matrix

	Fully Paid	Charged Off
Fully Paid	41137	2313
Charged Off	8507	955

Model recall: 0.10093

Model precision score: 0.29223

Model roc_auc_score: 0.52385

Model accuracy score: 0.79551

Random forest scaled: confusion matrix

	Fully Paid	Charged Off
Fully Paid	43201	294
Charged Off	9179	238

Model recall: 0.02527

Model precision score: 0.44736

Model roc_auc_score: 0.50999

Model accuracy score: 0.82096