**25/05/2021**

# Peer Lending Project

## *Stage 4 - Model Selection and Setup*

## *Group C*

In this step of the project, we continued to manipulate the data and dropped attributes to make it a better fit for our model. We built a few classification models in order to find the best model for our use. Some of the steps we took are changes from the last stage.

**Model Approach**

In the last stage, we've examined two methods of modeling, linear regression for predicting the realized return and classification model for loan status. We compared the approaches by checking the average return and standard deviation per X of the top loans. For the regression model, the top loans are the ones with the highest predicted realized return. While for the classification model, it is the loans that have the lowest probability to default. (e.g. top 5000 loans, top 10000). We saw that the results in both models are similar, whereas the regression usually got a higher average return but also a higher standard deviation. Therefore, we decided to continue with the second approach- The classification model (See Appendix 1 for the comparison).

**Realized return calculation**

In the previous stage, we offered a way to calculate the expected return using a certain equation, but we've decided that this is not the most accurate way and it's better to use the average return of the loans which will be predicted as fully paid by our model. We assume that the loans we have are a decent representation of the loans we will deploy in our model in the future, hence the average return on fully paid loans predicted by our model should be a good representation of the actual average return of future loans.

## The steps we took:

**Changes from the last step:**

1. **N/A's Handling:**
   For attributes with less than 1% N/A's, we decided to replace missing values with the median, since it's negligent and will not harm our model. For attributes with more than 1% N/A's, we dropped those instances. We can't prove that the attributes are normally distributed, therefore by replacing many missing values with the median we fear it will damage the true presentation of the population and hence harm our model.

2. **Attributes:**
   a. After a correlation check, we decided to drop the encoded attribute "addr_state" since it had nearly 0 correlation (both Pearson and Spearman) with our key attributes - realized return and loan status (See appendix 2.1+2.2 for the correlation matrix)
   b. We kept the encoded attribute "purpose" since it had some correlation with our key attributes and a low correlation with the rest of the predictors (Appendix 2.1+2.2). We will keep it for now and see if it passes the feature selection process in the next stage.

**Correlations check**

The problem is that even after these steps, we still have a large number of attributes – 49. We want to reduce the dimensionality of our data.
To do so, we checked for attributes with collinearity higher than 0.75 (both Pearson and Spearman). After reviewing it, we dropped one of the highly correlated attributes, because it holds similar information that won't contribute to our model's performance. Even though they had a relatively high correlation, we kept the attributes realized return and loan status since they are important to our model. We chose the threshold of 0.75 because we believe it's high enough to be redundant. **Dropped 15 Attributes due to high correlation (Appendix 2.3).**

**Modeling**

We started the modeling process with 35 attributes and 194,147 rows.
During this stage, we began to check different thresholds for our model by looking at the average realized return for each threshold. For now, we created a plot (Appendix 3.1) to examine the results for the different thresholds, but this is an issue which we will further explain in the next stage.

We split the data into train-test (70% train and 30% test) and used the same train set to fit all models (for logistic regression, we scaled the data), we checked 3 classification models - Logistic Regression, Random Forest, and Gradient Boosting. We compared the model using these metrics:

- AUC : How are my models perform better than a random classifier (Appendix 3.2)
- Fraction of loans (or absolute amount) invested VS average return (Appendix 3.3+ 3.4)
- Fraction of loans (or absolute amount) invested VS standard deviation of return (Appendix 3.5 + 3.6)

We compared our models to our baseline grade model (i.e. choosing loans based only on grades). Our goal is, for a specific investment or a specific fraction of loans we want to invest in, to get the same average return with a lower standard deviation or higher average return with the same standard deviation. The models assign for each loan a probability to default for a given threshold we choose. Our model will predict all loans assigned with a probability above this threshold as defaulted. As we raise the threshold, we invest in more loans but the average return decreases because we invest in more defaulted loans predicted as fully paid. The threshold should be tuned carefully and according to a combination of the amount we want to invest and the return we want to receive.

**Results**

After comparing the models on several train-test splits, it seems like all 3 models have similar performance. The AUC is between 0.69 to 0.7. For the top 30% of loans (i.e. the 30% of loans our models assigned the lowest probability to default), the Random Forest has a slightly higher average realized return (Appendix 3.1+3.3). As we increase the threshold and invest in more loans, the Gradient Boosting model has a small edge in the average realized return. Looking at the fraction of loans invested VS standard deviation of return, we see that the Random Forest model has the highest SD for every fraction (Appendix 3.4). The gap between the SD's narrows as we increase the fraction, However, this cancels the edge Random Forest has for the best 20% of loans.

**Comparison to the grade model**

Compared to the grade model (Appendix 4.1) we get a good feeling that our model can outperform it. For a specific average return and fraction of loans invested (e.g. Grade A), we get a slightly higher average return with a slightly lower standard deviation. We believe that in the next stage after we use feature selection and tune the hyperparameters of our model we can obtain better results that will outperform the grade model.

**Conclusions and next steps**

Based on the results we got , we've decided to keep tuning and improving the Random Forest and Gradient boosting models, to see if one can outperform the other in general or in a specific fraction of loans we want to invest in. We may recommend different models based on the company's restrictions.

# Appendices

## Appendix 1: Regression VS Classification

| Top number of loans | Regression average return | Classification average return | Regression SD of return | Classification SD of return |
|---|---|---|---|---|
| 5000 | 0.043 | 0.033 | 0.066 | 0.032 |
| 10000 | 0.04 | 0.034 | 0.062 | 0.04 |
| 15000 | 0.039 | 0.034 | 0.062 | 0.046 |
| 20000 | 0.037 | 0.034 | 0.062 | 0.05 |
| 25000 | 0.037 | 0.034 | 0.063 | 0.054 |
| 30000 | 0.036 | 0.034 | 0.065 | 0.058 |
| 35000 | 0.035 | 0.034 | 0.067 | 0.062 |
| 40000 | 0.034 | 0.033 | 0.069 | 0.066 |
| 45000 | 0.033 | 0.032 | 0.072 | 0.07 |
| 50000 | 0.031 | 0.031 | 0.075 | 0.074 |
| 55000 | 0.029 | 0.029 | 0.079 | 0.079 |

**Appendix 2: Correlation**

West/South West… stands for the the addr_state attribute after encoding(based on geographical location)

Appendix 2.1 - Pearson

| | purpose | West | Southwest | South East | North East | Midwest |
|---|---|---|---|---|---|---|
| **loan_status** | 0.048122 | -0.00540222 | 0.001927612 | 0.009553507 | 0.001136606 | -0.0076 |
| **realized_return** | -0.01617 | -0.00444589 | -0.00208154 | -0.006905774 | 0.006736079 | 0.006933 |

Appendix 2.2 -Spearman

| | purpose | West | Southwest | South East | North East | Midwest |
|---|---|---|---|---|---|---|
| **loan_status** | 0.047513 | -0.00540222 | 0.001927612 | 0.009553507 | 0.001136606 | -0.0076 |
| **realized_return** | 0.045101 | -0.01589078 | -0.00076726 | 0.000204054 | 0.00570971 | 0.011497 |

Appendix 2.3 - Correlation pairs

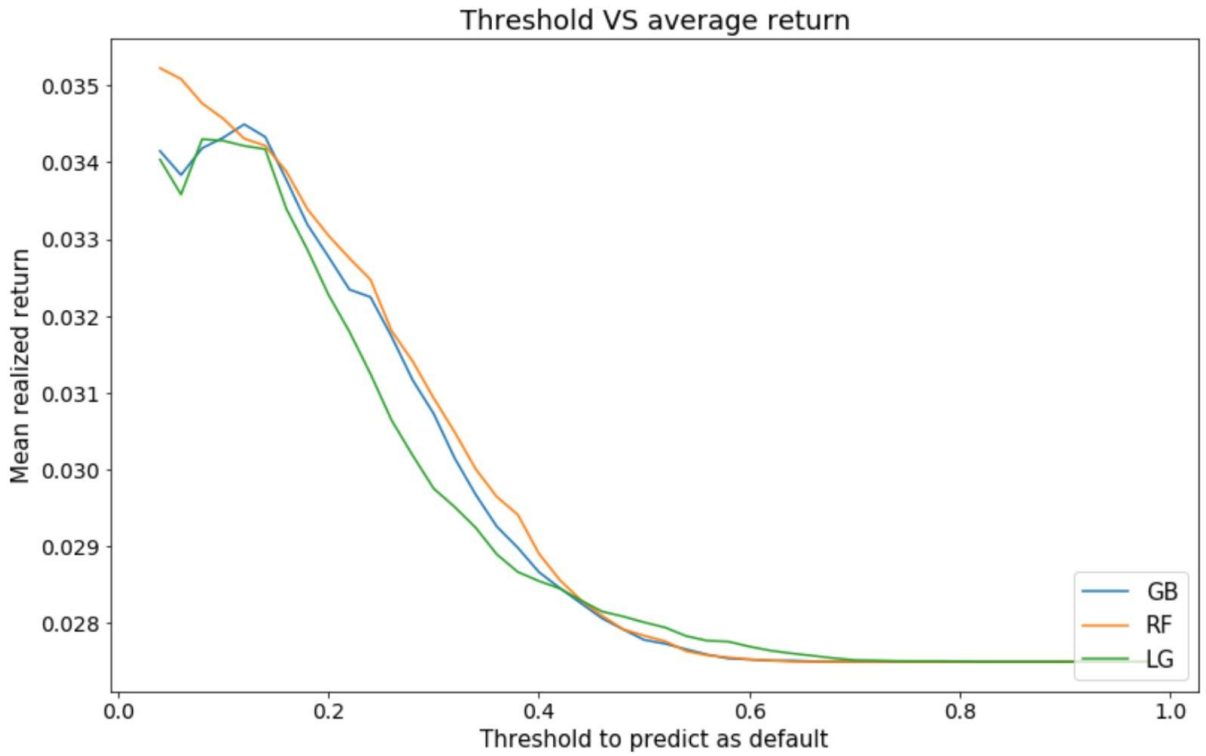
| Correlation pairs | |
|---|---|
| 'revol_util' | bc_util', |
| fico_range_low' | fico_range_high' |
| 'num_bc_sats' | num_bc_tl', |
| 'num_bc_tl' | num_rev_accts', |
| 'num_actv_bc_tl' | num_rev_tl_bal_gt_0', |
| 'num_op_rev_tl' | num_sats', |
| 'open_acc' | num_sats', |
| 'bc_util' | percent_bc_gt_75', |
| 'tot_cur_bal' | tot_hi_cred_lim', |
| 'avg_cur_bal' | tot_hi_cred_lim', |
| 'bc_open_to_buy' | total_bc_limit', |
| 'total_rev_hi_lim' | total_bc_limit', |
| 'total_bal_ex_mort' | total_il_high_credit_limit' |
| 'total_bal_il' | total_il_high_credit_limit', |
| 'revol_bal' | total_rev_hi_lim', |

Appendix 2.4 -Columns dropped due to high collinearity

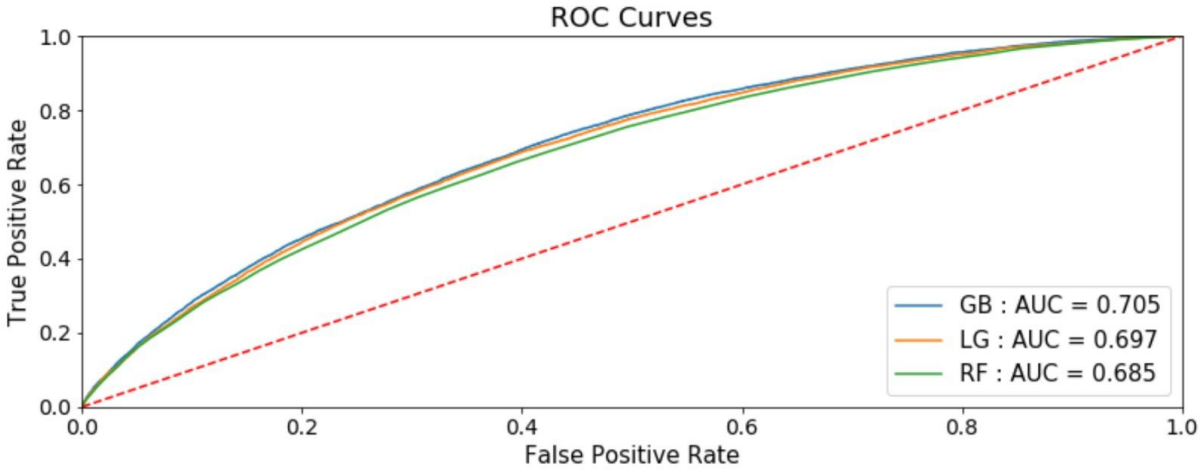
avg_cur_bal',
'bc_util',
'fico_range_high',
'grade',
'num_bc_sats',
'num_op_rev_tl',
'num_rev_accts',
'num_rev_tl_bal_gt_0',
'num_sats',
'percent_bc_gt_75',
'tot_hi_cred_lim',
'total_bal_ex_mort',
'total_bc_limit',
'total_il_high_credit_limit',
'total_rev_hi_lim']

## Appendix 3 - Models comparison

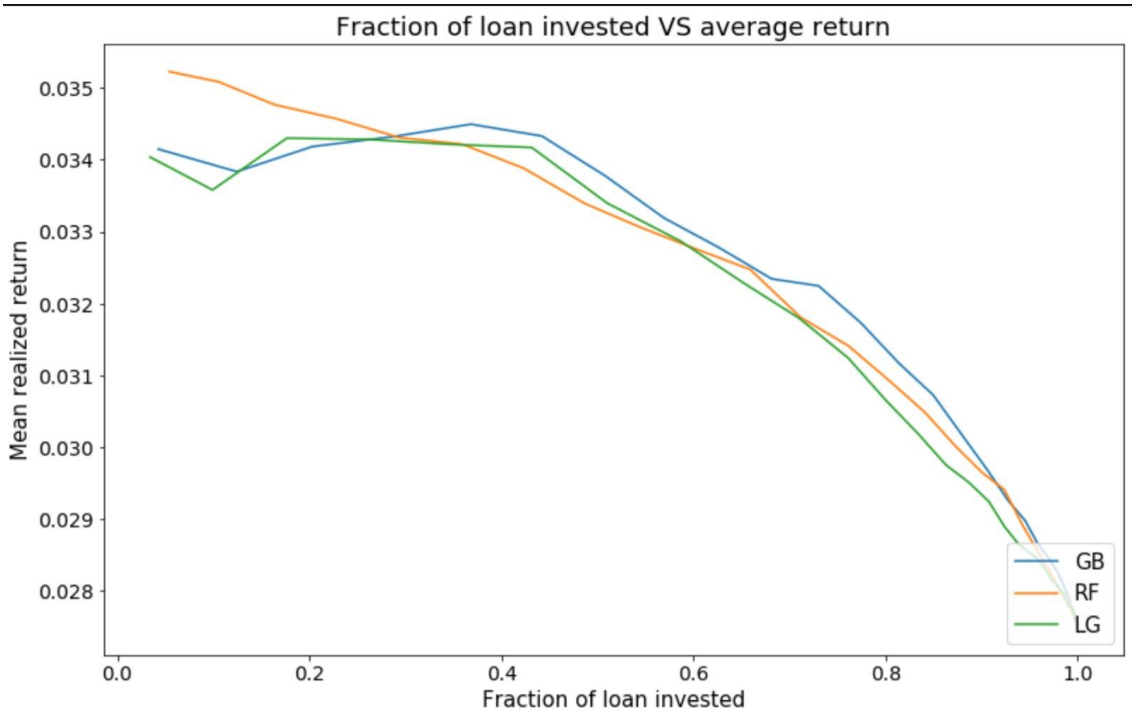
Appendix 3.1 -Threshold VS average return



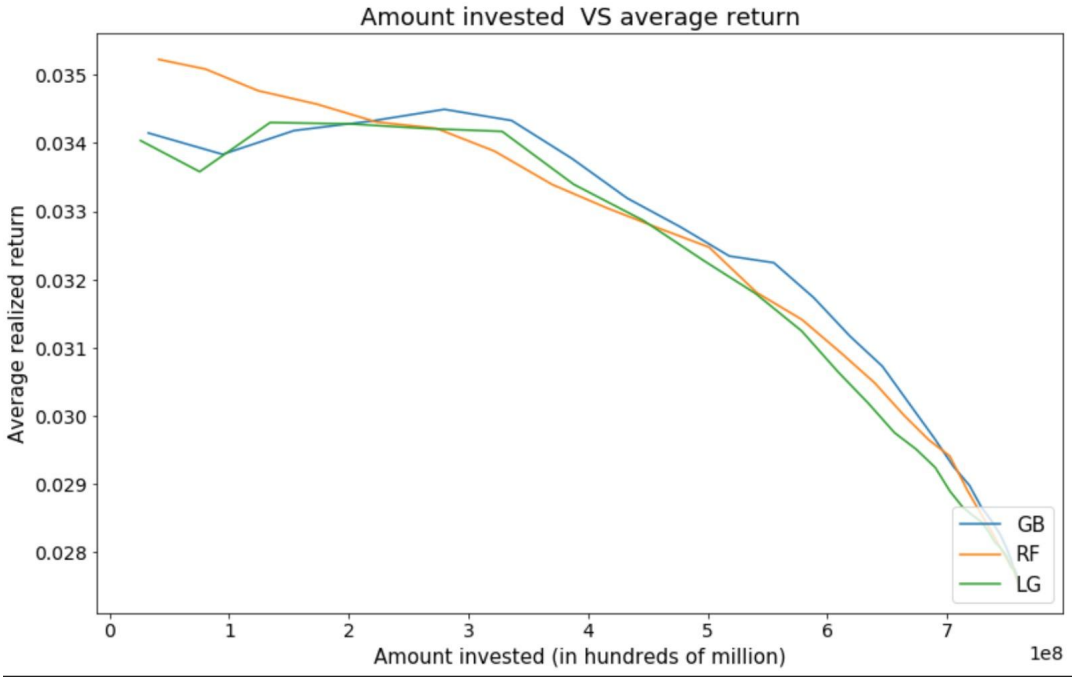

Appendix 3.2 - ROC curves



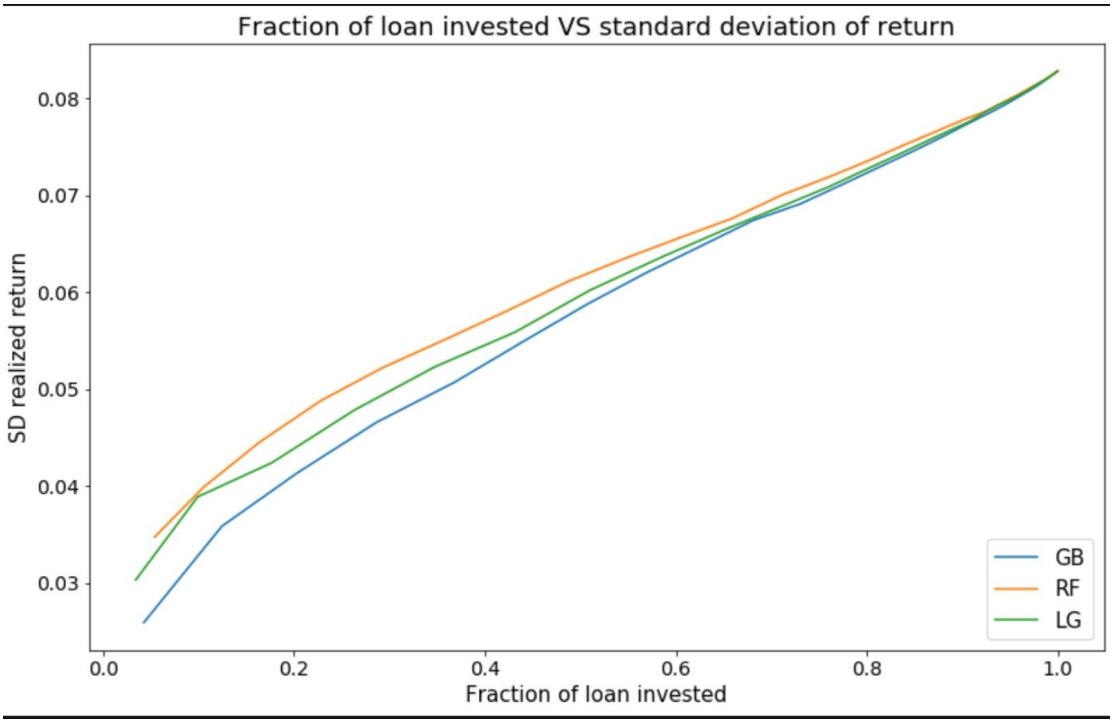

Appendix 3.3 - Fraction of loan invested VS average return



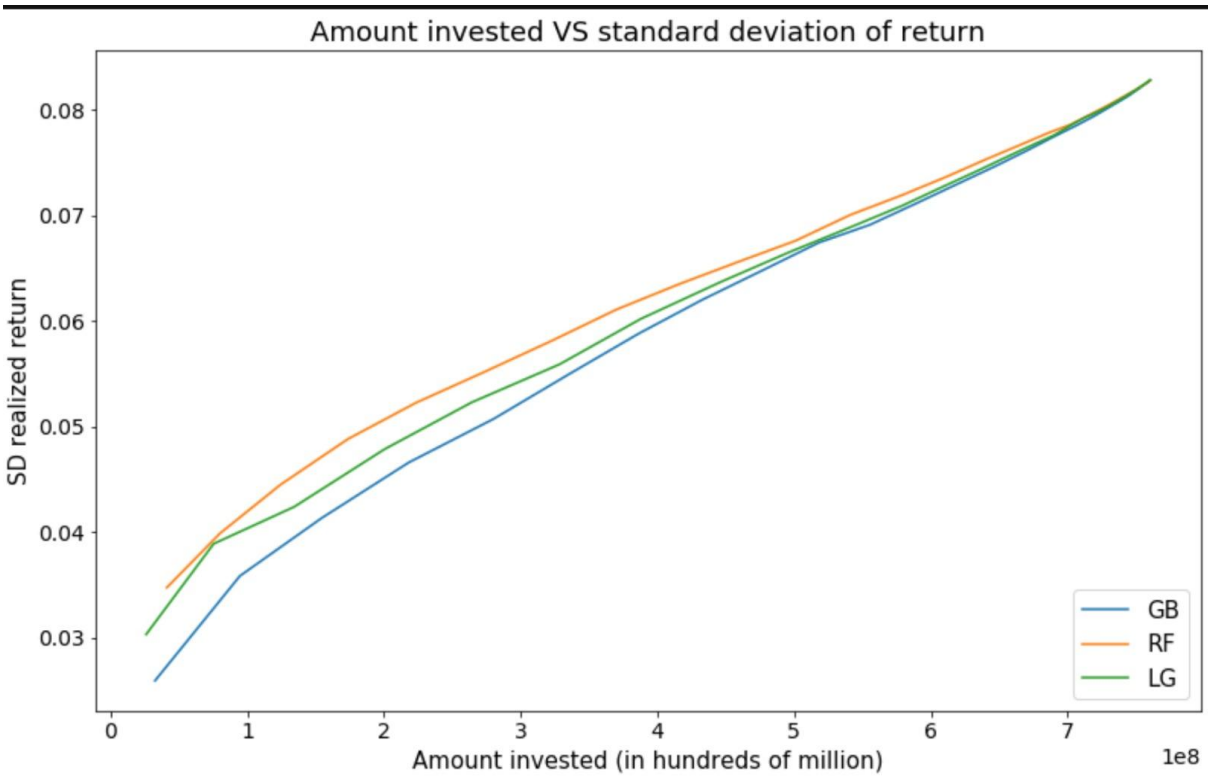

Appendix 3.4 - Amount invested VS average return




Appendix 3.5 - Fraction of loan invested VS SD of return




Appendix 3.6 - Amount invested VS SD of return




## Appendix 4 - Models probabilities distribution

Appendix 4.1 - Grade baseline

| grade/ Realized Return | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| mean | 0.0323 | 0.0305 | 0.0257 | 0.0194 | 0.0162 | 0.0046 | -0.002 |
| median | 0.0409 | 0.0558 | 0.0658 | 0.0679 | 0.0666 | 0.0514 | 0.04 |
| std | 0.0451 | 0.0708 | 0.0925 | 0.113 | 0.1296 | 0.144 | 0.1536 |
| min | -0.3227 | -0.3303 | -0.3235 | -0.3212 | -0.321 | -0.3199 | -0.3197 |
| max | 0.0957 | 0.1108 | 0.1294 | 0.1575 | 0.1813 | 0.201 | 0.1942 |
| count | 41611 | 67470 | 54659 | 22288 | 6352 | 1434 | 333 |
| loans_frac | 0.2143 | 0.3475 | 0.2815 | 0.1148 | 0.0327 | 0.0074 | 0.0017 |