GREAT YIELDS
a boutique investment house

Roy Madpis - 319091526
Rawaa Makhoul - 316114370
Alexandra Fatyeyeva – 336540950

Working Paper №1

GreatYields is a boutique investment house that invests in stocks and bonds. Till now the company produced low returns of approximately 2% annually. For this reason, the CIO, Walter, is considering P2P investments. As the analytical team in the company, we need to provide answers to several questions, that will enable Walter to decide whether to invest in peer-to-peer lending.

The first question we wish to address is: What are the expected realized returns[1] for different loan grades (A-G) and how are they distributed? Our approach in addressing this question:

First, we wish to calculate for each loan independently its' realized return. This is feasible as we have past data and thus know the outcome of each loan (default / fully paid) and more importantly, we have the total amount that was paid (column total_pymnt) (technically we can only use loans with a duration of 36 months and loans with a duration of 60 months with the status "Charged off" or "fully paid", otherwise the loan's status and the total_pymnt may change in the future, therefore those other loans can't be used). If the loan was **fully paid / partly paid** [and assuming its monthly payments were paid on time] the realized return is: $100\% \times (\frac{total\ amount\ paid\ [2]}{total\ original\ loan} - 1)$. If the loan **defaulted** and **nothing was paid**, then the realized return will be: $100\% \times \left(100\% \cdot \frac{0}{total\ original\ loan} - 1\right) = -100\%$

Second, we will group the data by the grade column and add a weight column which will be the quotient of each loan's total original amount with the sum of the loans of the specific grade.
Third, we will calculate the realized returns for each grade by performing a weighted average using the weight and the realized return while grouping the data by the grade.
Finally, we will create a Histogram of the realized return of each grade independently.

We have two approaches regarding the calculation of the **expected** realized return. Each approach is based on a different type of model. The first approach is building a **classification model** - to calculate the **expected** realized return for each grade independently, we need to calculate, for each grade, the **probability** to default (Charged-off) and to be fully paid. This can be done based on previous data - we can build a classification model (let's say a decision tree) and predict the probability to default/fully paid for each grade independently. Second, after having the probabilities to default/fully paid, we can calculate the realized expected return for each loan separately by multiplying the default probability with -100% and the fully paid probability with the interest of the loan. By adding these two numbers we can get the expected realized return (for each loan separately).

Lastly, we will calculate the expected realized returns for each grade by performing a weighted average using the weight and the expected realized return while grouping the data by the grade.
The main drawback of this approach is its' inability to take into account loans that were partly paid[3].

[1] The realized return is the return earned by investing in a loan.

[2] The total amount paid contains the actual interest paid. We can use column 41 "total_pymnt" for the calculation.

[3] And for example, among the 133,887 loans of Q1, in 2019, 23,655 were charged off, among them only 49 had the value "0" in total_pymnt (meaning **-100%** realized return). This means we probably shouldn't use the naïve approach of classifying only to default / not default.

The Second approach is building a **Regression model** - to calculate the **expected** realized return for each grade independently we can build a regression model to predict the **total amount paid** (column total_pymnt). For each loan independently, the model will predict its' total_pymnt. Then we will use the following formula to calculate the expected realized return for each loan: $100\% \times (\frac{total\ amount\ paid^{4}}{total\ original\ loan} - 1)$. Lastly, we will calculate the expected realized returns for each grade by performing a weighted average using the weight and the expected realized return while grouping the data by the grade.

As mentioned above, our answers will be based on the previous loan data that we have. In order to know whether the data at hand is informative and can help to select loans to invest in, we have to further investigate the data and determine which columns are indeed informative and which are not. We will do so by considering two perspectives:

- **Business-wise**: We will look at the data from the investor's point of view and locate the columns that may be involved in the investing consideration process (columns that the investors are probably looking at for selecting loans to invest in)
  Columns / features that might be informative: employment / title / purpose of the loan / grade etc.
- **Data analysis-wise**: We can use correlations analysis: first, we wish to observe whether there's a correlation between specific columns [features] (This will help us understand the relations between the variables and the combination of which features would be most informative) and the correlation between each feature and the target variable, which can be the total_pymnt/realized return/grade.

One way to do so is using a correlation table that can help determine which features have high correlation with each other (and thus may be redundant) and which features have high correlation with our target variable. Another way to understand the importance of the variables is using feature selection.

After discovering if the data is informative, we want to know what increased performance can be expected compared to a baseline of simply selecting loans based on their grades. As Walter claimed, the company's average return for stocks and bonds was 2% annually. This is the "baseline" performance. In other words, the company will invest only in bonds that their expected return will be 2% or above. The baseline method relies on selecting loans based on their grades.

To explain what increased performance can be expected, we will build two prediction models that will help us decide in which loan we should invest - both will aim at predicting the expected realized return of each loan independently. The first model, called the "baseline", will be based only on the given loan grades. The second model will be based on a combination of multiple columns (The informative columns which we will choose at the near future). The output of both models will be the expected realized return of each given loan. Therefore, as the company's average return on stocks and bonds was 2% annually, we would invest only in loans whose expected realized return is higher than 2%.

As the model will be tested with a dedicated test set, we will be able to determine how "well" the two models performed, i.e., if we would have invested in all the loans with expected realized return > 2% (According to each model's output), what would have been the actual annual realized return gain? We would compare the performance of the "baseline model" with our second "fancier" model.

[4] The Regression Model's output.

Two notes to consider in the future:

1. Does one of the models able to gain an annual realized return > 2% (meaning - does Walter's company should enter P2P lending)?

2. How well our "fancier" model is performing compared to the baseline model? (does it worth building and investing in an elaborate model that performs only x% better than a simple model? Or maybe the differences are significant enough?)

Regarding the two models mentioned above - both will be based on predicting the total amount paid (total_pymnt) via a regression model (and this will enable calculating the expected return of each given loan) or will be based on predicting the probability to default and not default (and those probabilities will enable calculating the expected return of each given loan). If we use the classification model, the model will output default / not default with a threshold of 50% - in order to gain maximum accuracy score. (True = the loan won't default, False = default) We may wish to alter this threshold, and probably use a higher threshold, so only loans with high probability to not-default, will be classified as so by the model. Doing so will probably lower our model accuracy but may increase the total revenues. For choosing the best threshold we can use a profit curve. To generate the profit curve, we need a cost benefit matrix:

Cost benefit matrix		Actual Class	
		P (not default)	n (default)
Predicted Class	Y (prediction = not default)	+ The given interest rate of the loan	**-100%**
	N (prediction = default)	- The given interest rate of the loan	0%

As mentioned above, If we use a classification model, we will probably use a profit curve to decide the best threshold to use (for the model predicting default/not default and their probabilities) This threshold will maximize the expected realized returns of the loans. Then, in order to calculate the **average returns** - we will perform a weighted average. We will filter the loans so we will have only the loans with expected return > 2%. This will be our "***suggested portfolio***". Afterwards, we will calculate the weight of each loan in our "suggested portfolio" (The weight = the quotient of each loan's total original amount with the sum of the loans in the suggested portfolio). Finally, we will perform a weighted average using the weight column and the expected return column. The outcome will be the average return GreatYeilds can expect from investing in peer lending loans.

Having the "suggested portfolio", we can calculate the variance and standard error of the expected return column, this may serve as a measure of the risk level entailed in peer to peer lending.

Important note: The specific average return and variance/standard error (of the expected returns) that we will get will be influenced directly from the randomly chosen test set. For this reason, we wish to use **cross validation**. In this Manner, we will get K different test sets. From each, independently, we will filter expected return > 2% and get K "suggested portfolios". Then on each we will perform a weighted average and get K "average" returns and K different variances. We would like to analyze the range (max - min), mean, variance and standard deviation of the K different averages and Variances. If the range will be statistically small enough, we may conclude that the real average return and variance (of the population) may be close to the mean of the average returns and variances we got in the K folds.