# PYCE LOANS

***Team: Nicholas Pittavino, Denver Yu, João Pedro Khair Cunha, Emmanuel Enoh***

## STAGE 1

***Preprocessing***: Our team began by understanding and cleaning the raw data of past loan applications (e.g., converting categorical data, creating dummy variables, focusing social2 profiles). By separating the factors (sex, employment, marital status, income, and the social parameter) against the dependent variable of interest (default or not), we may begin to prepare a model for estimating the probability of default (PD).

***Model building (calculating PD)***: Using the past loan data, we used three Machine Learning techniques (Linear regression, Logistic regression, and Random forest) to generate three different models, and we found the Random forest model to be the most appropriate predictor for the project. It gave the most logical and accurate predictors of past default rates.

The available data to train the model was restricted by the fact that, unlike other groups, we didn't have information on social media 1 and 3, which could potentially improve the accuracy of predicting default probabilities. Hence, in a real-life situation, a first strategy would be to further develop our company's Data Analytics tools so that we could use metrics from all the applicants' social media platforms. However, for the assignment, we simply trained the model without social1 and social3 information as this is the best we could do given the constraints.

***Offering loans (calculating interest rates)***: Then, we advanced to the final step of predicting interest rates. For each applicant, the interest rate would be the rate to breakeven plus a fixed minimum rate, depending on the probability of default from our Random Forest model. The formula was "*Breakeven Rate = PD/(1-PD)*".

We used the break-even formula to visualize which interest rate PYCE Loans should give to avoid losing money. Then, to each interest rate found, we added 4% for two reasons: (1) to not only break even but to have a profit despite lacking competitive information; (2) to have a margin that could sustain possible inaccuracies from our model. Using that strategy, we could give specific interest rates based on each probability of default, personalizing our service and increasing our chances of giving the best loan possible given the default rate.

Even with the sum of 4%, our interest rates were marginally greater than our breakeven rate. This reflects the strategy adopted by PYCE Loans: ***to give fewer loans and try to win the loans with low-interest rates***. With this strategy, we planned to minimize the number of defaulted loans and thus not have to worry about compensating these losses via high-interest loans.

## STAGE 2

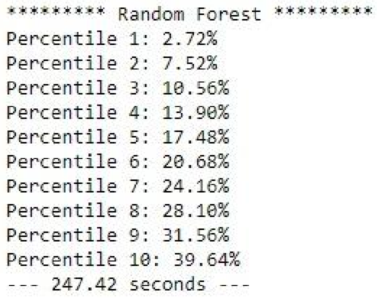
***Analyzing results***: According to the results from stage 1, PYCE Loans was the seventh team out of twenty, losing a total of €1,217,340. Having the detailed table with the results, the group could analyze why we lost money and how we could change that scenario for Stage 2.

With our strategy not to give a lot of loans, we ended up offering a total of 20,908 loans, but only 12,005 were chosen by the applicants (***57.4%***). Compared to the random competitors 1 and 3, we offered 77.7% and 71.5% less loans, respectively. However, as shown below, our acceptance rates were comparatively strong. PYCE Loans' borrowers defaulted a total of 805 times (***6.70%***), which is not a high number, but suggests room for improvement given that we ended up losing money.

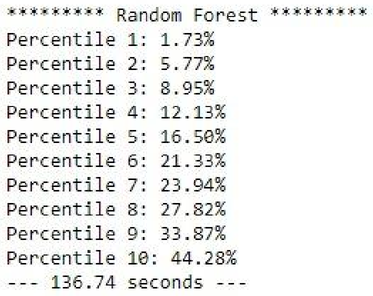
If we had won all the bids from our competitors, we would have made a profit of €1,553,109 with a 4.84% default rate, which validates the accuracy of our model, from which we expected a default rate of around 5.37% (based on the analysis on past loans). Our main oversight here was not accounting for game theory, as the applicants could choose from other companies depending on the rates and their preferences.

***Improving the model (calculating PD)***: By analyzing the data, we could see that our model was well built, as well as part of our strategy. This is represented in our scrutiny of what our profit would be if we had won every loan we offered. That shows that the model works, and that theoretically we are on the right track to have profits.

Our first modification was in improving our model. We optimized the parameters using the GridSearchCV function from the python package sklearn, choosing the model with the lowest root-mean-squared-error (rmse). In the process, we compared 144 combinations from our 7 parameters to reach the best model possible. Consequently, comparing the default rates of the 10 deciles, we observe that the model is better at distinguishing lower default probabilities from the riskier loans. This is beneficial as we can more accurately decide which loans align with our profit strategy.



```
********* Random Forest *********
Percentile 1: 2.72%
Percentile 2: 7.52%
Percentile 3: 10.56%
Percentile 4: 13.90%
Percentile 5: 17.48%
Percentile 6: 20.68%
Percentile 7: 24.16%
Percentile 8: 28.10%
Percentile 9: 31.56%
Percentile 10: 39.64%
--- 247.42 seconds ---
```



```
********* Random Forest *********
Percentile 1: 1.73%
Percentile 2: 5.77%
Percentile 3: 8.95%
Percentile 4: 12.13%
Percentile 5: 16.50%
Percentile 6: 21.33%
Percentile 7: 23.94%
Percentile 8: 27.82%
Percentile 9: 33.87%
Percentile 10: 44.28%
--- 136.74 seconds ---
```

***Left***: Stage 1 Model
***Right***: Stage 2 Model

***Simulating profits (calculating interest rates)***: The remaining efforts of the project is to determine our profit strategy. We can either proceed with our old strategy (aim for high bid acceptance by offering low-interest loans only to the best applicants) or we could change our strategy based on a simulation of market conditions given by Stage 1. This alternative strategy takes into consideration our offered interest rate, the competitor's offered interest rates, and the borrower type (0, 1, 2, or 3). We developed a simulating function to calculate, for each applicant, how much profit we would hypothetically earn.

For the market simulation, we tested 5050 combinations of 2 parameters:

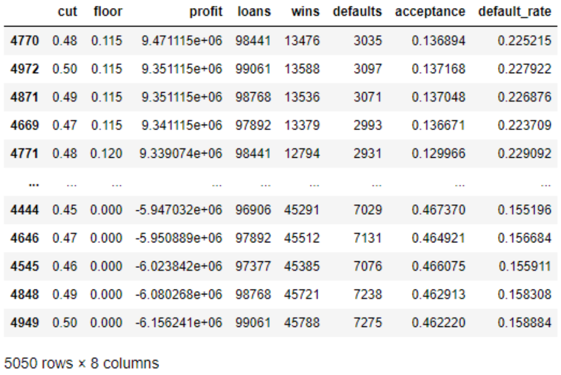
- ***Cut***: the maximum PD, above which we deny loans. This is effectively quality control.
    - Between 0 < x <= 50%. The lower bound exists because a 0% cut means we do not provide loans to anyone. A 50% loan, according to the break-even formula, would yield a 100% interest rate (plus the floor).
- ***Floor***: the minimum interest rate, theoretically given to applicants with 0% PD.
    - Between 0 and 50%. Interest rates higher than 100% are lowered to 100%.

- This drives the acceptance rate, since a high floor equals higher rates and consequently lower acceptance from borrowers. On the other hand, it may increase profits if combined with low default rates.

The interest rate calculation for each scenario followed the same rules as in Stage 1:

```
if  PD > cut:
        No loan given
else
        Interest rate = PD/(1-PD) + floor
```

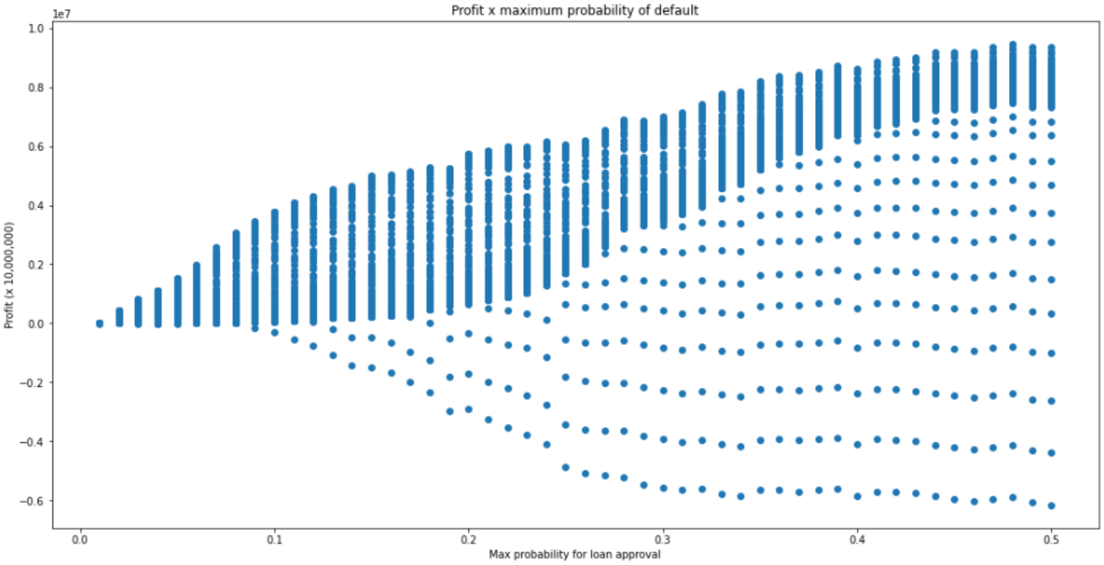
The simulation results were very interesting, as shown in the table on the right:



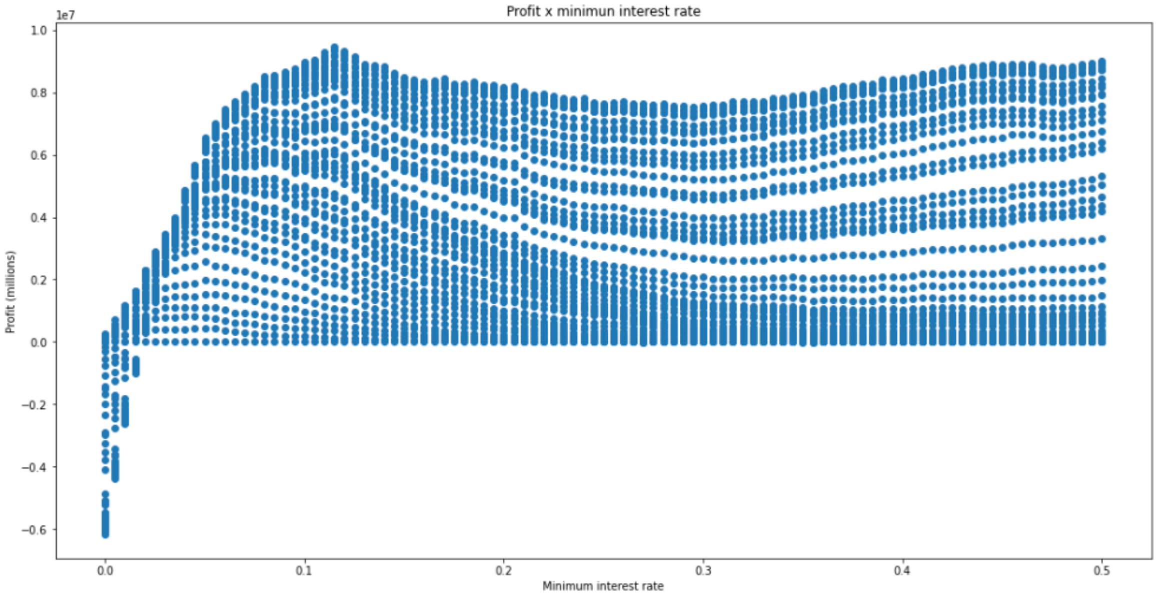
| | cut | floor | profit | loans | wins | defaults | acceptance | default_rate |
|---|---|---|---|---|---|---|---|---|
| **4770** | 0.48 | 0.115 | 9.471115e+06 | 98441 | 13476 | 3035 | 0.136894 | 0.225215 |
| **4972** | 0.50 | 0.115 | 9.351115e+06 | 99061 | 13588 | 3097 | 0.137168 | 0.227922 |
| **4871** | 0.49 | 0.115 | 9.351115e+06 | 98768 | 13536 | 3071 | 0.137048 | 0.226876 |
| **4669** | 0.47 | 0.115 | 9.341115e+06 | 97892 | 13379 | 2993 | 0.136671 | 0.223709 |
| **4771** | 0.48 | 0.120 | 9.339074e+06 | 98441 | 12794 | 2931 | 0.129966 | 0.229092 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **4444** | 0.45 | 0.000 | -5.947032e+06 | 96906 | 45291 | 7029 | 0.467370 | 0.155196 |
| **4646** | 0.47 | 0.000 | -5.950889e+06 | 97892 | 45512 | 7131 | 0.464921 | 0.156684 |
| **4545** | 0.46 | 0.000 | -6.023842e+06 | 97377 | 45385 | 7076 | 0.466075 | 0.155911 |
| **4848** | 0.49 | 0.000 | -6.080268e+06 | 98768 | 45721 | 7238 | 0.462913 | 0.158308 |
| **4949** | 0.50 | 0.000 | -6.156241e+06 | 99061 | 45788 | 7275 | 0.462220 | 0.158884 |

5050 rows × 8 columns

The best results were from scenarios with high cut (which means loans to almost all applicants) and intermediate floor values. We ran scatter plots to better visualize the profit sensitivity. The first one is between cut (x-axis) and profit (y-axis), and shows that both the profits and losses are intensified in high cuts.



The second scatter plot is between floor (x-axis) and profit (y-axis), and confirms that the optimal floor point is around 11%, although high floor points also lead to great returns. It is curious to see that the loss scenarios are all concentrated in the lower end of the floor range. In this market, we would only face a loss with low floor values (< 2%).



***Final strategy:*** We decided to change our pricing strategy based on our simulation of the market's conditions. Instead of giving few, low-interest loans, we will provide a greater number of loans (given that, in this assignment, we don't have any constraints of available cash to lend) with higher interest rates. At first, this seems a tad counterintuitive, since higher rates would drive acceptance rates down (applicants would prefer the competitors' rates), but it's not unusual for Machine Learning models bring unexpected insights. It was interesting to see that neither acceptance rates nor default rates were key metrics for reaching the best results: out of the 5050 scenarios, the most profitable scenario has the 2727th lowest default rate and the 1080th highest acceptance rate. Therefore, profit is a more complex function that depends on our rate, the competitors' rates and the borrower's preferences. Nevertheless, it is important to highlight that we are assuming that there won't be any major strategy changes from our competitors, keeping the accuracy of our market simulation.

In conclusion, in Stage 2 we performed a parameter optimization on our Machine Learning model and an automated market simulation to find the best way to calculate the interest rate. All our code is available here. Unless there is a complete shift of the market's conditions, we are confident that our results in Stage 2 will be better than in Stage 1.