**Investment Strategy Analysis in Peer-to-Peer Lending Platform**

Jiagan Ye

**1. Introduction**

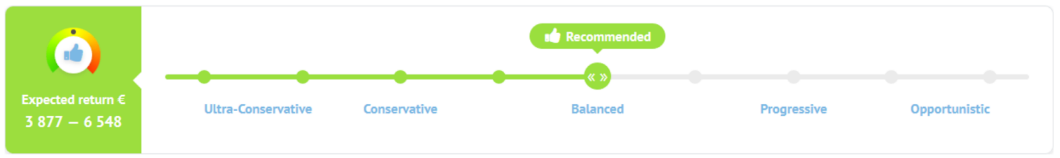
**1.1 Introduction to Bondora**

Building a profitable investment strategy can come in many ways and, when done correctly, can produce great wealth. For some, investment strategies may seem challenging to develop, but anything is indeed possible with a bit of data. In our analysis, we will dive into the FinTech space of peer-to-peer (P2P) lending and some investment schemes built from defaulted loans and machine learning algorithms. P2P lending platforms enable anyone to invest in loans or ask for a loan. Distinct from a traditional bank, P2P companies go through their platform rather than banks. The platform then takes similar actions to a bank and gives a credit rating based on the borrower and the particular loan. The P2P space is quite exciting for investors as it garnishes them with a new way to invest without the need for a bank and produces less hassle and a great chance for their loan to be accepted and funded. For our project we will dive into a P2P lending company named Bondora.

Bondora, established in 2008, is one of the oldest and well-respected loan lenders in Europe. Housing markets for Estonia, Finland, and Spain, Bondora has become one of the most prominent leaders in non-bank digital loans in Europe. Bondora relies on its fair and low-cost scheme to a population that can see some well-earned results. Thus, for Bondora, they approve borrowers and lenders from low-to-mid income and low-to-mid credit customers. The strategy achieved here allows a more significant population to become investors and investees; however, the risk applied results in a higher probability of loan default. Therefore, with data provided by Bondora, we will build an investment strategy that can yield significant returns.

Our models will build on the framework provided by Bondora and push their limits further. Bondora offers multiple products for customers to use when creating their investment models. From these, we will be focusing on two, "Go and Grow" and "Portfolio Manager." "Go and Grow" promises a return of up to ~6.75% on investment. "Go and Grow" is fully automated and allows quicker liquidity than the other products. Bondora built this product to enable any lender to invest in multiple loans quickly. Compared to Bondora's different products, one does not have to list your loans and wait for them to materialize. "Go and Grow" is for those that need quick cash at their promised rate and are allowed to cash out at a moment's notice. As well as, "Go and Grow" does not need the entire loan amount to invest, making it an excellent plan for investors.

"Portfolio Manager" compared to "Go and Grow" is semi-automated. The product offered here changes slightly from "Go and Grow," it does not provide a steady return. Instead, the "Portfolio Manager" product makes investment decisions based on one's risk levels. Once you deposit money into your account, the product will quickly make investments based on your choices. The risk level ranges from “Ultra-Conservative” to “Opportunistic.” From the risk level set, Bondora will calculate an approximate return rate. Bondora uses the scale shown below to calculate not only your risk level, but also the credit rating of loans that will soon be invested in.



Any investment chosen has great return possibilities. Best of all, Bondora has allowed multiple sectors of people, varying from different social and economic classes, to partake in a new investment strategy. However, while Bondora offers extraordinary returns, can we beat their system and allow an even better investment strategy?

**1.2 Importance and Uniqueness of Project**

Achieving a solid investment strategy can be difficult; thus, our goal is to beat Bondora's established models. We will be acting as consultants to investors that use Bondora to maximize their potential gains. Our models may be feared by borrowers as we may predict their loans from defaulting, thus we are one-sided in our goal to make investors more profitable. While many have created default predictions strategies, our investment plans go beyond that. To gain a more comprehensive and more diverse approach, we will be implementing filters to our loan models. The filters will allow a more varied strategy than just picking a risk level. The more diverse a loan can be, the more fine-tuned any single loan investment can become.

First, we will have to implement our default prediction models using machine learning. The four machine learning methods used to predict loan default are Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors Algorithm (KNN), and Random Forest. Once our models have been built, we will assess the best model and compare its returns to Bondora's "Go and Grow" product, which promises a ~6.75% return rate.

Once our model is in place, we will then assess our model to that of the "Portfolio Manager." As stated, the "Portfolio Manager" allows more settings to be in place and emphasizes risk levels for customers. Our model will be introduced as a filter, allowing more specific fine-tuning and a more effective selection process for loan investments. We will be focusing on aspects of risk and credit ratings. Selection of various risk levels will give our customers to gain on their risks while minimizing loss freely. Our model protects the customer by having a better default prediction and thus can make better riskier picks when selecting loans.

Creating a diverse and calibrated investment strategy will allow investors and investees more freedom and security. The plan is set based on great returns while increasing risk and personalization. Like Bondora, loans will be selected automatically based on one's settings but will create greater profits tailored to one's immediate or long-term needs.

**2. Method**

**2.1 Data Cleaning**

The loan data was downloaded from Bondora's website (https://www.bondora.com/marketing/media/LoanData.zip), which contains different types of loans from March 1, 2015, to January 27, 2020. Our dataset has 182,017 rows, 112 columns in CSV format.

The raw data has many null values that may not suit model building. Therefore, features with more than 50% of null values are considered as not usable and dropped from the dataset. The remaining columns with missing values, depending on the situation, are filled with either 0 or mean values of the remaining values. For example, in missing values in the feature, "age" is replaced with the mean of the remaining. And for the transaction amount, a value of 0 is placed, assuming the loan has no transaction.

Besides, among 112 features, some features are meaningless, such as AuctionID, Auction Name, which will be dropped. Also, there are redundant columns, such as age and date of birth, which basically refer to the same thing, among which only one of them will be kept.

To help predict whether loans will default or not, a target variable, "Default," is defined as "default" if the loan is overdue for more than 90 days without new repayment, otherwise as "not default."

Variables with definite meaning but present in the numerical form are converted back to categorical variables for convenience of following analysis, such as "Verification Type," "Language Code," "Gender", "Use of Loan".

**2.2 Feature Selection**

A correlation matrix is built to explore the relationship between "Default" and other features. At first, all the categorical data are excluded and only the rest of them are used to construct the correlation matrix. By setting 0.25 as boundary, the following variables: "PrincipleBalance", "LanguageCode", "InterestAndPenaltyBalance", "RecoverageStage" and "Interest" are found to be relevant with the target variable "Default", which can be potential features as input variables for following machine learning models (Figure 1). Other variables with correlation value below 0.25 but by common sense is closely related with “Default” are also included, and with consideration of elimination of overlapping impact between variables, a raw\_features list is generated for further analysis, that is ["PrincipalBalance", "Country", "InterestAndPenaltyBalance", "Interest", "LoanDuration", "Gender", "NewCreditCustomer", "Restructured", "AppliedAmount", "Age", "VerificationType", "HomeOwnershipType", "LiabilitiesTotal", "IncomeTotal", "CreditScoreEeMini", "Education", "MaritalStatus", "EmploymentStatus", "Default"].

Table

Description automatically generated

*Figure 1. Correlation between features and default*

Based on the raw\_features list, a further round of selection is conducted.

As the debt-to-income ratio is a more interpretable indicator for default possibility, a new variable, "DebtToIncomeRatio," is generated by the division of "LiabilitiesTotal" by "IncomeTotal." In contrast, the variable "LiabilitiesTotal" is dropped.

Due to the belief that customers who don't verify their income statements are less trustable, and therefore will be more likely to default, the variable "VerificationType" is analyzed to decide whether to be kept. "VerificationType" is defined as the method used for loan application data verification. First, the numbers are replaced by the actual names of the verification status for readability, as "1.0" : "NotVerified" "2.0" : "VerifiedByPhone" "3.0" : "VerifiedByOtherDocument" "4.0" : "VerifiedByBankStatement". Then they are converted to categorical, and the distribution of default and not default state across different verification types is calculated. Given the similar proportions in default for each verification type (Figure 2), except for verified by phone, which only makes up a small fraction of the dataset. It would be safe to remove "VerificationType" from the dataset for the sake of reducing the noise that could impact our models.

Table

Description automatically generated

*Figure 2. Proportions of default for each verification type*

The variables “Gender”, “Education”, “MaritalStatus”, “Restructured”, “CreditScoreEeMini”, “HomeOwnershipType” are analyzed in the same manner as above (the distribution of default state amount different categories for each feature is present as Figure 3-8). Given the similar proportions in terms of default for different categories within a specific variable, those variables are removed.

Text, table

Description automatically generated

*Figure 3. Proportions of default for each gender*

Table

Description automatically generated

*Figure 4. Proportions of default for each education category*

Table

Description automatically generated

*Figure 5. Proportions of default for each marital status*

Table

Description automatically generated

*Figure 6. Proportions of default for each reconstructured category*

Table

Description automatically generated

*Figure 7. Proportions of default for each CreditScoreEeMini category*

Graphical user interface

Description automatically generated with low confidence

*Figure 8. Proportions of default for each home ownership type*

For the variable “EmploymentStatus”, the numbers are replaced by the actual names of the employment status, as "1.0" : "Unemployed", "2.0" : "Partially employed", "3.0" : "Fully employed", "4.0" : "Self-employed", "5.0" : "Entrepreneur", "6.0" : "Retiree". The distribution of default and not default state across employment status is calculated (Figure 9). Since the status “Retiree” and “Self-employed” have similar default and not default distribution, which significantly differs from other categories, they are combined as one group [“Employment”] = 0. And the rest of the employment status categories are combined as [“Employment”] = 1. Thus, the variable “EmploymentStatus” is converted to a binary feature “Employment” and kept.

Table

Description automatically generated

*Figure 9. Proportions of default for each employment status*

For the variables “Country” and “NewCreditCustomer”, as the distribution of default and not default state has obvious different patterns between different categories (Figure 10&11), they are kept as a predictor.

Table

Description automatically generated

*Figure 10. Proportions of default for country*

Table

Description automatically generated with medium confidence

*Figure 11. Proportions of default for each NewCreditCustomer category*

As a result, 11 predictors are finalized (Table 1, the default as the outcome). Numerical variables are normalized and converted to categorical bins, while categorical variables are converted to dummy variables. And all of them are converted to binary categorical for following model building.

|  |  |
| --- | --- |
| Predictor Chosen | Potential indication for prediction |
| PrincipalBalance | Principal that still needs to be paid by the borrower. High principal balance may be an indicator of heavy burden for borrower and affects the repayment. |
| Country | Residency of the borrower. Lenders may prefer loans from certain countries with concern of macroeconomic condition of that country. |
| InterestAndPenaltyBalance | Unpaid interest and penalties. High unpaid interest and penalties may discourage borrowers from repayment. |
| Interest | Maximum interest rate accepted in the loan application. The acceptance of interest rates may give information about borrowers’ affordability or urgency for loans. |
| LoanDuration | Current loan duration in months. Loan duration is a relevant factor influencing borrowers’ ability for repayment. |
| NewCreditCustomer | Did the customer have prior credit history in Bondora 0 Customer had at least 3 months of credit history in Bondora 1 No prior credit history in Bondora. Customers with a prior credit history may be more reliable for repayment. |
| AppliedAmount | The amount borrower applied for originally. Applied loan amount presents borrowers’ need for money. Large amount of applied loan, that is huge unmet need for capital, may be a potential indicator for high probability of default. |
| Age | The age of the borrower when signing the loan application. Age may to some extent be related to borrowers’ ability to earn money, thus affecting repayment ability. |
| IncomeTotal | Borrower's total income can be a direct predictor for repayment ability. |
| DebtToIncomeRatio | Total debt to income ratio reveals the capital burden of the borrowers, can be more straightforward in repayment prediction. |
| Employment | Separate borrowers in more stable employment status (entrepreneur, fully employed or partially employed) and less stable employment status (retiree or self-employed) |

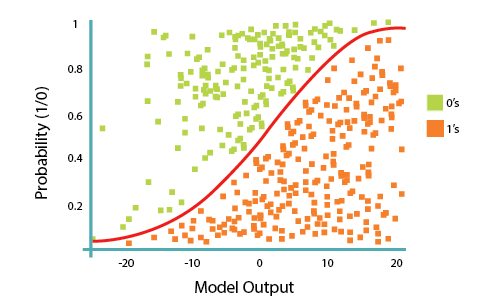
*Table 1. Predictors kept for model building and their potential indication for prediction*

**2.3 Model Building**

11 predictors chosen from the feature selection step are used in model building. "Default," defined as a loan overdue for more than 90 days, is used as the prediction outcome. Here, logistic regression, support vector machine, K-nearest neighbor algorithm, and random forest models will be tested and compared to derive an optimized model as our investment strategy. And the whole dataset is split into train set and test set in the ratio of 7:3.

**2.3.1 Logistic Regression**

Logistic regression is a great choice to predict default as it identifies the relationship of the dependent variable to one or more independent variables. We use the logistic regression model to predict the likelihood of default in any given loan.

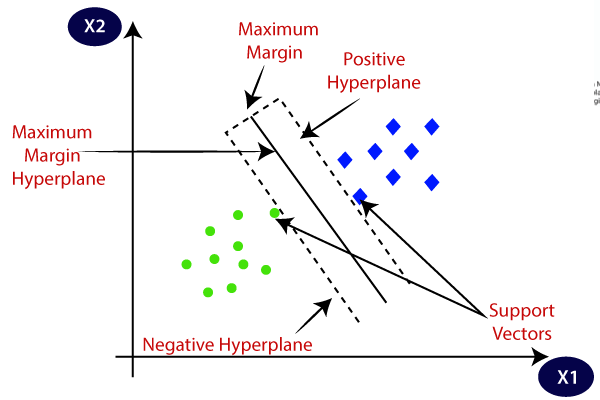


*Figure 11. Illustration for logistic regression model*

The logistic regression takes in only numeric or categorical data. Thus, the model needs even further cleaning and setting up work. Fortunately, the only feature left to convert to categorical is the country feature. After conversion, we can now construct our logistic regression model. First, we split our dataset between our training and test set and ultimately fit it into our model. After the model has been fit, we test our model through statistic scores and a confusion matrix visualization to better represent our results. All of these computations were made possible by the sci-kit-learn library in Python.

**2.3.2 Support Vector Machine (SVM)**

SVM is a supervised machine learning model used for classification, outliers' detection, and regression. In the SVM model, we plot each data as a physical point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (www.analyticsvidhya.com). Based on the graph below, our job is to find the hyperplane that maximizes the margin between two classes (the green and blue points in the graph).

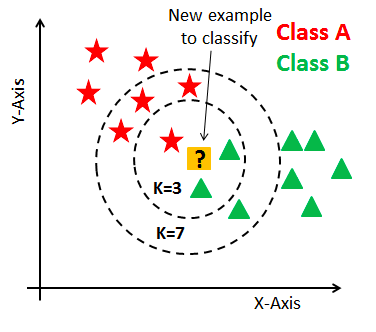


*Figure 12. Illustration for support vector machine model*

The parameters of SVM model are fine-tuned using grid search method. The optimized SVM model found has the following parameters: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}. The train set is used to train the SVM model, and the test set is used for the calculation of accuracy, precision, recall, and ROC.

**2.3.3 K-Nearest Neighbors Algorithm (KNN)**

K-nearest neighbor algorithm (KNN) is a non-parametric classification method. By majority voting, the object is assigned to the class that is most common among its k nearest neighbors. The "K" is the number of nearest neighbors we wish to take the vote from. Taking the below graph, for example, when k equals to 3, out of 3 closest points to the target, 2 of them are green triangles, hence we can conclude the target point belongs to class B. But when k increases to 7, out of 7 closest points to the target, 4 of them are red stars. Therefore we can conclude that the target points belong to class A this time. So the result may vary as the value of k changes.



*Figure 13. Illustration for K-nearest neighbors model*

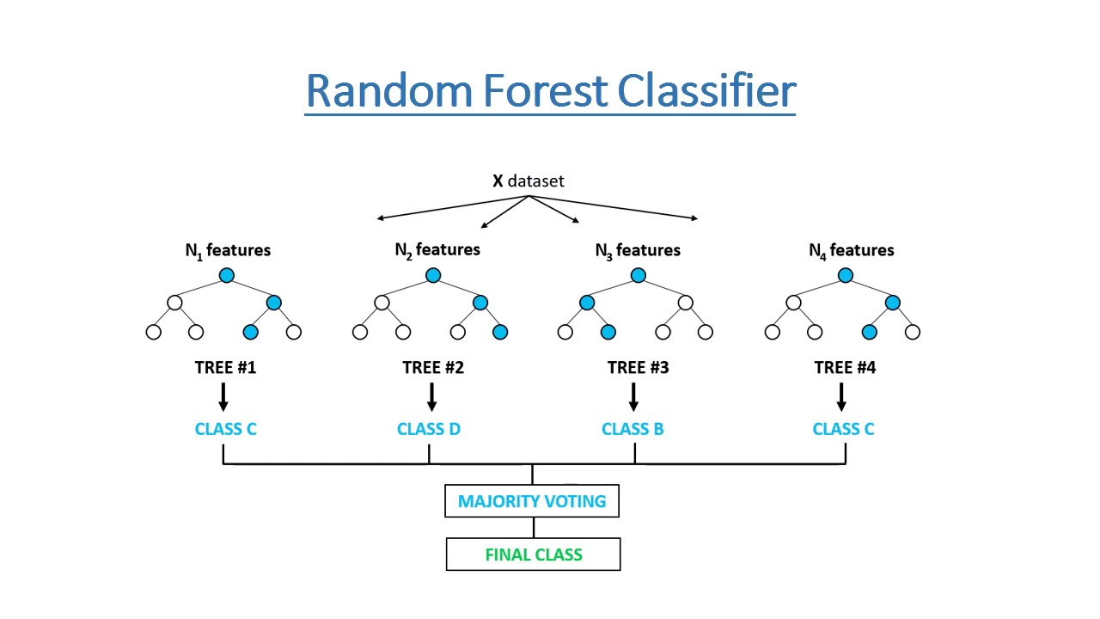
Since this algorithm relies on physical distance for classification, our features represent different physical units and vary a lot in scales. We need to [normalize](https://en.wikipedia.org/wiki/Normalization_(statistics)) the features first before putting them in our model. Also, we want to include dummy variables, such as employment, new credit customer in our model to increase the dimension, and therefore to improve the overall prediction accuracy of our models.

KNN only has one hyperparameter K, which is the number of neighbors to consider. The best choice of k depends upon the data; typically, A small value of k means that noise will have a stronger influence on the result. A larger value of k reduces the effect of the noise on the classification. But it will make the model computationally expensive and makes the result very hard to explain.

The parameters of KNN model are fine-tuned using the grid search method. The optimized KNN model found has the following parameters: {'n\_neighbors': 11}. The train set is used to train the KNN model, and the test set is used for the calculation of accuracy, precision, recall, and ROC.

**2.3.4 Random Forest**

Random forest is another powerful classification method. It is an ensemble of a large number of individual decision trees. Each individual tree makes a class prediction and the majority vote side will become our model’s prediction.



*Figure 14. Illustration for random forest model*

The parameters of random forest model is fine-tuned using grid search method. The optimized random forest model found has the following parameters: {'bootstrap': False, 'criterion': 'gini', 'max\_depth': 15, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 11, 'n\_estimators': 280}. And the explanation of hyperparameters is as follow:

* Bootstrap is false, which means the whole dataset is used to build each tree.
* Criterion equals ‘Gini’ meaning Gini impurity is used to measure the quality of each split.
* Max\_depth is the maximum number of levels in each tree.
* max\_features': 'sqrt' means the tree will not stop searching for a split until the sqrt of the number of node samples is found.
* The min\_sample\_split is the minimum number of samples required to split at a node. The default value is 2, which means the tree will keep on splitting until the nodes only have 2 subnodes. As a result, the tree grows in size and therefore overfits the data. Hence we set the test range kind of high here to prevent overfitting risk.
* Min\_samples\_leaf is the minimum number of samples required to be at a leaf node. The number is one here, which means a subnode will only be considered if it contains at least 1 training sample.
* N\_estimators is just the number of trees in the forests.

The train set is used to train the Random Forest model, and the test set is used for calculation of accuracy, precision, recall and ROC.

**2.4 Comparison of Our Model and Bondora’s Investment Strategy**

**2.4.1 Generation of Our Model**

Among the 4 machine learning models, random forest has the best accuracy, recall and ROC, thus is selected for building our own investment strategy. Based on the random forest model, every loan can be predicted as default or not. A “predicted non-default pool” is generated, which helps filter out the loans that are likely to default based on our model. And our investment strategy is to only choose loans from our “predicted non-default pool”.

**2.4.2 Simulation of Bondora’s “Go & Grow” Investment Strategy and Comparison with our Model**

By definition, Go & Grow is an automated investment strategy that will guarantee a 6.75% return to investors. Thus, simulation of this strategy can simply be done by setting the average return rate as 6.75%. And the average return of loans in our “predicted non-default pool” is calculated and compared with the value of 6.75% to see whether our model outperforms the Go & Grow strategy.

**2.4.3 Simulation of Bondora’s “Portfolio Manager” Investment Strategy and Comparison with Our Model**

Next, comparison of our model with Bondora’s “Portfolio Manager” strategy, which helps select loans with different risk preferences will be conducted.

We will simulate the “Portfolio Manager” strategy by random drawing 30 samples from each credit rating loan (7 ranks in total). And each rating group will be a representation of a risk preference: A and above, B, C, D, E, F, HR rating group. The expected return of the portfolio will be calculated as the sum of the products of return and default or not state (0 or 1).

For application of our model, the “predicted non-default pool” will be used as a filter. The random selection of samples from each rating group as above will be restricted to the pool, and the return will be calculated in the same manner.

Simulation for each strategy and each rating group will be conducted 500 times to check how many times our model can beat the “Portfolio Manager” and how much improvement in return can be achieved by using our model.

**3. Results**

**3.1 Data Cleaning**

The original raw datasets have more than 110 features. After data cleaning, only 49 features were left.

**3.2 Feature Selection**

11 predictors are finalized: “PrincipalBalance”, “Country”, “InterestAndPenaltyBalance”, “Interest”, “LoanDuration”, “NewCreditCustomer”, “AppliedAmount”, “Age”, “IncomeTotal”, “DebtToIncomeRatio”, “Employment”.

Table

Description automatically generated

*Figure 11. Predictors and outcome variable for model building*

**3.3 Machine Learning Model Building and Selection**

The logistic regression model has the accuracy of 85.92%, precision of 78.27%, recall of 74.11% and ROC (Receiver Operating Characteristic) of 0.9394.

Chart, treemap chart

Description automatically generatedChart

Description automatically generated

*Figure 12. Confusion matrix and ROC curve of linear regression model*

The SVM model has the accuracy of 89.92%, precision of 81.00%, recall of 86.65% and ROC of 0.9581.

Chart, treemap chart

Description automatically generatedA picture containing graphical user interface

Description automatically generated

*Figure 13. Confusion matrix and ROC curve of SVM model*

The KNN model has the accuracy of 87.10%, precision of 79.63%, recall of 76.47% and ROC of 0.9172.

KNN model doesn’t have many true positive cases, but the biggest amount of false negative cases, which gives it a decent accuracy rate. Since it also has a lot of true negative cases and false positive cases, in which the model made the wrong prediction, hence the precision score and recall score is not good. Overall, compared with previous models, KNN has the lowest score of roc curve, which indicates its bad performance of distinguishing the default and not default classes.

Chart, treemap chart

Description automatically generatedChart

Description automatically generated

*Figure 14. Confusion matrix and ROC curve of KNN model*

The random forest model has the accuracy of 90.46%, precision of 79.49%, recall of 91.85% and ROC of 0.9704.

The random forest model is actually very effective in discrimination because it made a lot of correct predictions, including true positive and false negative cases, which gives it the highest accuracy score, 90.46%. It also has the highest recall score due to its lowest number of true negative cases. Although the precision score is kind of low, overall, random forest is still excellent in default prediction and it has the highest score in the roc curve.

Chart, treemap chart

Description automatically generatedChart

Description automatically generated

*Figure 14. Confusion matrix and ROC curve of random forest model*

Among the 4 machine learning models, the random forest models have leading positions in both accuracy, recall and ROC, thus is selected as our final model for making investment strategy. Based on the random forest models, the “predicted non-default pool” is generated, including the loans that we predicted to be unlikely to default.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistic Regression | SVM | KNN | Random Forest |
| Accuracy | 85.92% | 89.92% | 87.10% | 90.46% |
| Precision | 78.27% | 81.00% | 79.63% | 79.49% |
| Recall | 74.11% | 86.65% | 76.47% | 91.85% |
| ROC | 0.9394 | 0.9581 | 0.9172 | 0.9704 |

*Table 2. Comparison of 4 machine learning models*

Chart, bar chart

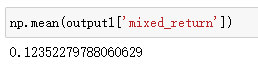
Description automatically generated

*Figure 15. Comparison of 4 machine learning models*

**3.4 Comparison of Investment Strategies**

**3.4.1 Comparison with “Go & Grow” Investment Strategy**

The average return of our “predicted non-default pool” is 12.35%, which obviously outperforms the Go & Grow strategies, which only guarantees a return of 6.75%.



*Figure 16. Average return of the “predicted non-default pool”*

**3.4.2 Comparison with “Portfolio Manager” Investment Strategy**

Since “Portfolio Manager” strategy allows investors to choose a portfolio of loans based on different risk preference, simulation of the strategy can be achieved by drawing loans from each rating of loans respectively (i.e. A or above, B, C, D, E, F, HR). Portfolio simulation from each rating here is confined to drawing of 30 random loans from that rating group. And we compare the portfolio return of direct drawing from each rating group of all loans and that of drawing from each rating group restricted to our “predicted non-default pool”. In other words, our “predicted non-default pool” serve as a filter to sieve out loans with high probability to default before drawing of loans from each rating group. The result suggests that our “predicted non-default pool” strategy successfully improved the return of portfolio: for group A or above, B, C , D, E, F and HR, our model beat “Portfolio Manager” model for 99.8%, 98%, 99.2%, 100%, 100%, 99.8% and 62.8% of all the simulation (out of 500 times of simulation) and increase the return by 4.7%, 4%, 4.8%, 4.7%, 4.9%,4.8%, 2.9%, respectively.

Besides, a stratified rating group is generated by combining 30 samples from each rating group together in representation of a well-diversified investment strategy. In this scenario, our model wins all the time and improved the return by 4.1% on average.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 30 samples each | | | | | | | 30\*7 samples |
| Rating group | A or above | B | C | D | E | F | HR | Stratified |
| Number of wins | 499 | 490 | 496 | 500 | 500 | 499 | 314 | 500 |
| Win rate | 99.8% | 98.0% | 99.2% | 100.0% | 100.0% | 99.8% | 62.8% | 100.0% |
| Improved return | 4.7% | 4.0% | 4.8% | 4.7% | 4.9% | 4.8% | 2.9% | 4.1% |

*Table 3. Comparison of our “predicted non-default pool” strategy with “Portfolio Manager” strategy*

One thing to be noted in the result table above is the relatively low improved return for HR group (only 2.9% compared to others all above 4%). However, it is reasonable since HR referring to the loan with the highest credit quality, offering high safety for timely payment of debt obligations are less likely to default. Therefore, our strategy of filtering out likely-default loans have little impact on this kind of loans.

**4. Conclusion**

As seen, we have already reached our first goal of creating a better strategy than Bondora’s primary tactics. Having earned 12.35% in returns on average, results in a 5.6% better return than promised by Bondora (6.75%). Thus we have derived a model that can outperform and become a better investment strategy for customers that use the P2P platform.

Our machine learning model can achieve 90% accuracy in default loans prediction for Bondora platform. And adding our “predicted non-default pool” as a filter on Bondora’s “Portfolio Manager” strategy can help investors get 4% more returns on average. However, the addition of the filter is never limited to the “Portfolio Manager” strategy, investors who is more dedicated to the portfolio management could also apply the result of our model to design a variety of financial products that will have a steady high return.

**5. Discussion and Next Steps**

Considering the satisfactory result of our model on Bondora and similitude of many P2P lending platforms, our model is generalizable to other P2P lending platforms, such as Lending Club and Prosper. Investors can also use our strategy to get higher returns by avoiding loans with high probability of default.

Furthermore, the use of the process of machine learning for default prediction in this project could even be extended to the realm of banking business, since loan default predict from P2P lending and credit default prediction from banks have similar logic.

However, our model is not perfect yet, and there is still some uncertainty about its stability and robustness. On one hand, the hyperparameters of the machine learning model can be tested in a broader range for performance improvement. On the other hand, we will keep monitor the model’s performance when new records come in. Seasonal adjustment is necessary once the model’s performance drop below baseline to accommodate for the change caused by external factors.(https://towardsdatascience.com/)

**6. Bibliography**

“Investing Made Simple: Bondora.” *Bondora.com*, <https://www.bondora.com/en>.

*SVM: Support Vector Machine Algorithm in machine learning*. Analytics Vidhya. (2021, August 26). Retrieved December 3, 2021, from <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>.

Xu, Z. (J. (2021, November 2). *Loan default prediction for profit maximization*. Medium. Retrieved December 3, 2021, from <https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b>.