International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-8, Issue-5, May 2022
ISSN: 2395-3470
www.ijseas.com

Robo Advisory: On using XGBoost algorithms for customer's Wealth and portfolio Management

Mohsen A. Hassan[1] , Aliaa AA Youssif[2] , Osama Imam[3] , Amr S. Ghoneim[4],

[1] Researcher –PHD-, Faculty of Computers & Artificial Intelligence, Helwan University (HU), Cairo, Egypt.

[2] Professor of Computer Science, Arab Academy for Science, Technology and maritime transportation (AASTMT), Cairo, Egypt.

[3] Professor of information system, Faculty of Computers & Artificial Intelligence, Helwan University (HU), Cairo, Egypt.

[4] Assistant Professor of Computer Science, Faculty of Computers & Artificial Intelligence, Helwan University (HU), Cairo, Egypt.

Abstract

Robo advising is becoming increasingly popular among ordinary investors. Researchers are always working to develop machine learning and deep learning algorithms that can reliably anticipate stock price movement with minimal error. The emergence of Robo-advisor wealth management services raises the question of whether they can take the place of human-based advice, specifically how to handle a client's behavioral biases in an automated manner. One method would be to use machine learning technologies throughout the client profiling process. A trained neural network, on the other hand, is sometimes regarded as a black box, which may cause anxiety among customers and regulators because most systems do not explain or exhibit their thinking. Also A well-constructed questionnaire (Markowitz, 1968) is the traditional method for calculating the risk profile, which is also employed by Robo-advising systems (Barsky et al., 1997). Such questionnaires ask questions to measure a client's willingness to risk as an opportunity to make money if the client seeks thrills in having high returns on investments if the client gets a thrill out of investing, and if the client would invest money a high return on their investment, even though it means accepting a high degree of risk, is the most important aspect. We address this issue by proposing a possible criterion for assessing the influence of factors when providing advice. The goal of this work is to present a wealth management machine learning adviser and to improve the automated wealth management process. **The accuracy of our model was 0.9830**.

Keywords: Wealth Management advice , Gradient Boosting, Long Short-Term Memory (LSTM), Stock Price Prediction,

1. Introduction

In recent decades, Machine Learning (ML) has been widely applied to a variety of fields. The three major components of machine learning, supervised learning, unsupervised learning, and reinforcement learning, have led to creative techniques, particularly in the finance industry. Financial time series forecasting, for example, can benefit from supervised learning approaches, whether or not deep neural networks are used (Ahmed et al., 2010). Portfolio selection and optimization have been aided by unsupervised learning techniques such as clustering (Bontempi et al., 2012). Given a large number of retail and institutional investors and increasing inflows in the asset management industry, the use of machine learning in investment management is particularly promising (Wang and Zhou, 2020). However, there are a number of difficulties in integrating machine learning techniques into investment management. In contrast to high-frequency trading, where microstructure data is abundant, data for training machine learning algorithms in the asset management industry is scarce, particularly for long-term low-frequency monthly or quarterly rebalancing investments. Understanding the investment objectives and risk profiles of investors, which is a critical first step for advising retail clients, is another problem. This amounts to learning the risk-return tradeoff parameter, commonly known as risk aversion/tolerance, in the basic single-period mean-variance MV portfolio optimization context (Wang and Yu, 2021) or equivalently, the target return, or the target variance. A properly assessed risk preference will result in a portfolio that meets the client's risk-return tradeoff, but an incorrectly calculated objective may result in devastating deviations from the client's long-term investment goals.

A well-constructed questionnaire (Markowitz, 1968) is the traditional method for calculating the risk profile, which is also employed by Robo-advising systems (Barsky et al., 1997). Such questionnaires ask questions to measure a client's willingness to risk as an opportunity to make money if the client seeks thrills in having high returns on investments if the client gets a thrill out of investing, and if the client would invest money a high return on their investment, even though it means accepting a high degree of risk, is the most important aspect. On a digital platform, Robo Advisor is an algorithm-driven software that provides automated financial planning services. It's built to handle everything from registration to client needs and goals analysis, advising, asset allocations, and re-balancing. A human advisor may or may not be involved. Because of the ambiguity and bias, such a technique is frequently difficult (Yu et al., 2020). Clients' subjective responses in hypothetical scenarios, in particular, may not accurately reflect their intrinsic risk preferences. The use of online inverse optimization is a more quantitatively based, systematic method to learning clients' risk profiles. By directly collecting and analyzing a client's answers to questionnaires, we can deduce a rational client's risk preference and give a piece of tailored advice for the client. This approach enables for evaluation of embedded portfolio risks, as well as an automatic portfolio adjustment decision process when appropriate, ensuring that real portfolio risk is aligned with the investor's true risk assessment. The automation of wealth management operations that were previously manual is a key feature of our proposed approach. Traditional financial advisors may devote a large amount of time to routine operations like re-balancing a portfolio to its target weights or identifying stocks that are trading at a loss for tax purposes. Automating these operations eliminates the danger of human error and allows them to be conducted in real-time across several accounts at a cheap cost. Utilizing Robo-advisors is less expensive than using traditional financial advisors (Kaya et al., 2017). Most Robo-advisors charge 0.25% to 0.50% in advisory fees, compared to about 1% for a typical advisor (Grealish and Kolm, 2021), making them appealing to individual investors. Furthermore, the use of automation and passive investing methods by Robo-advisors reduces the likelihood of internal agency conflicts and conflicts of interest that can occur between financial advisors and their clients.

After stating the motivations for Robo advising, the utilization of Machine and Deep learning for employ them in the process, the remaining part of the paper is organized as follows. Different Machine/Reinforcement Learning approaches are reviewed in Section 2. In Section 3, shows the proposed models and dataset.

The results are presented in Section 4. Experiments and result analysis for different architectures are shown in section 5. The conclusion and further work are found in Section 6.

2. Related Works

(Capponi et al., 2021) present a two-ML agent-based full-cycle data-driven investment Robo advising architecture. The first agent, an inverse portfolio 3 agent, uses online inverse optimization to deduce an investor's risk preference and projected return directly from historical allocation data. The second agent, a deep reinforcement learning (RL) agent, collects the inferred sequence of predicted returns to design a new multi-period mean-variance portfolio optimization problem that deep RL techniques can answer. The planned investment pipeline has continuously outperformed the SP 500 benchmark portfolio that represents the aggregate market optimal allocation using real market data from April 1, 2016, to February 1, 2021.

The performance of several forecasting algorithms that can be utilized in the Robo advising framework was compared by (Potdar and Pande, 2021). In this study, the RMSE (Root Mean Square Value) value is used to compare the performance of algorithms because it is a good and straightforward loss function. Newer algorithms, such as those presented by Facebook, have been shown to be capable of accurately predicting stock values, although ARIMA surpassed all of the algorithms studied in this study.

(Gerhana et al., 2021) used the Markowitz Method to find acceptable mutual fund investment products in the Robo-Advisor application. This study is a test to see which mutual fund portfolio is the best. Danareksa Flexible Orchid (DAF), Bahana Trailblazer Fund (BTF), CIMB Principal Total Return Equity (CIMBPTRE), and Danareksa Mawar Consumer 10 were the Mutual Funds chosen (DMK). From 2014 to 2018, a total of 60 data points were used. The first portfolio has a higher return rate of 0.55%, while the second portfolio has a return rate of 0.54%, according to the estimates.

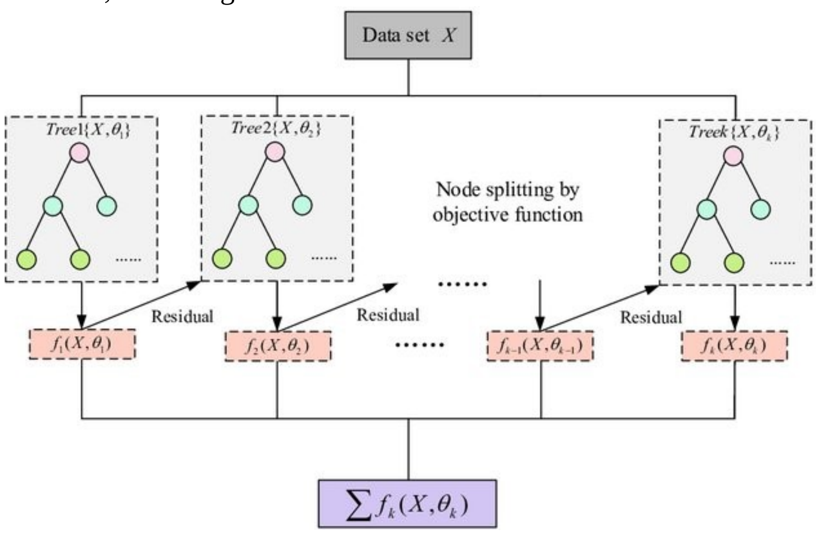

Figure 1: illustrates an example of boosting in XGBoost

(Wang et al., 2019) present a Robo-advisor system that combines trend forecasting, portfolio management, and a recommendation mechanism. Three multi-objective RankNet kernels used in a gated neural network structure could rate target financial goods and recommend the top-n stocks to investors. The gated neural network learns to select or weight each RankNet in order to incorporate the most essential partial network inputs, such as profits per share, market index, and hidden time-series information. Experiments show that our suggested Robo-advice advisor's outcomes can exceed existing algorithms at the moment.

3. Proposed Methodology

4. Proposed Model

For our model, we used XGBoost. Extreme Gradient Boosting XGBoost is an open-source package that implements the gradient boosting technique efficiently and effectively. Quickly after its development and initial release, XGBoost became the go-to method for classification and regression issues in machine learning contests and was frequently the crucial component in winning solutions. Gradient boosting (Chen and Guestrin, 2016; Yang et al., 2020) is a machine learning technique that builds a prediction model from an ensemble of weak prediction models, most often decision trees, for regression, classification, and other tasks. Gradient boosted trees are the consequence of a decision tree that is a bad learner, and it outperforms random forest in most circumstances. It builds the model in the same step-by-step fashion like other boosting methods, but it expands the scope by allowing optimization of any differentiable loss function. Gradient boosting attempts to forecast a target variable by combining the estimates of a series of simpler, weaker models. Figure 1 Illustrate the procedure.

4.1. Dataset

For our dataset, we collected questionnaire answers from clients and advice from experts for each client. We collected 1843 records. The dataset contains answers for a client's investment amount, an investment period, a client's monthly liability, the risk level, the partial period, the partial withdrawal, and the expert's advice. We managed to create 81 unique pieces of advice, spanning multiple permutations for each client's answers. This allows for better learning methods for the model since it shows multiple approaches for tackling a clients' specific case, ensuring the model's robustness. One challenge in the proposed dataset thats the imbalanced dataset, since not all advices are equally distributed. This challenge was solved using the class-weight in the cross-entropy loss. Samples of the data are shown in Table.1.

5. Results

We train on 80% of each dataset, while 10% is for validation, and 10% for testing. For prediction, the input is a sample containing the values of the questions and the output in the prediction of the advice. We trained with a maximum depth of 10, 100 estimators, and a balanced class weight. The
XGBoost model yielded an accuracy of 0.9830

Table 1: Sample Data

Invest Period	Part Period	Partial Withdraw	Invest Amount	Risk Level	Advice
36.0	6.0	0.5	250K	75.0	57.0
36.0	1.0	0.25	170K	60.0	14.0
12.0	3.0	0.25	200K	25.0	19.0
36.0	9.0	0.5	150K	65.0	39.0
7.0	0.0	0.0	40K	25.0	55.0
36.0	3.0	0.75	195K	25.0	66.0
12.0	6.0	0.5	120K	25.0	21.0

6. Discussion

6.1. Feature Importance

To show which feature was the most impactful on the model's decisions, we calculated the feature importance (Wojtas and Chen, 2020; Casalicchio et al., 2018) for each variable in our dataset. The decrease in node impurity is weighted by the likelihood of accessing that node to compute feature significance. The number of samples that reach the node divided by the total number of samples yields the node probability. The more significant the feature, the higher the value. In other words, the relevance is explicitly determined for each property in the dataset, allowing attributes to be ordered and contrasted. The amount that each attribute split point improves the performance measure, 5weighted by the number of observations the node is responsible for, is used to assess the importance of a single decision tree. The purity used to pick the split points or any more precise error function could be utilized as a performance measure. The feature importances are then averaged over the entire model's decision trees. In general, significance assigns a score to each feature that shows how useful or important it was in the creation of the model's enhanced decision trees. The higher the relative relevance of an attribute, the more it is used to make

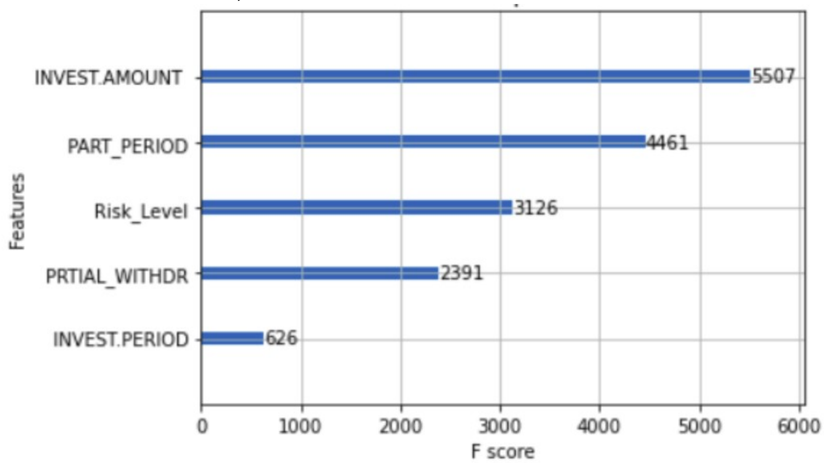

Figure 2: Feature Importance of XGBOOST

crucial judgments with decision trees. As shown in figure 2, we have illustrated the ranking of each feature in our model, the higher the F-1 score, the more impactful the feature is. From our observation, the model sees that the client's investment amount is the most impactful to the model's prediction. While there is no proof by experts that this feature is the most impactful in their experience, we believe that this could be a start of a standard for assessing clients since none exist since most experts rely on subjective assessments and experience to properly advise clients.

6.2. Experiments and Result Analysis

In testing, we tried different models. We began with LightGBM Light Gradient Boosted Machine (Ke et al., 2017; Nabi et al., 2020), an open-source gradient boosting implementation that attempts to be more efficient and effective than existing methods. LightGBM is based on Gradient-based One-Side Sampling GOSS, a variant of the gradient boosting strategy that concentrates on training samples that result in a higher gradient, which speeds up learning and reduces the method's computing cost. When used in conjunction with Exclusive Feature
Bundling EFB, a technique for bundling sparse mostly zero mutually exclusive features, such as one-hot encoded categorical variable inputs. As a result, it's a type of automatic feature selection. LightGBM yielded an F1-Score of 0.9830. We then followed by testing CatBoost (Prokhorenkova et al., 2018). CatBoost is an open-source machine learning algorithm that is relatively recent. One of CatBoost's key advantages is its ability to combine a range of data kinds, such as photos, audio, and text, into a single framework. However, unlike the bulk of other machine learning algorithms, CatBoost has its unique approach of dealing with categorical data, requiring only a minimal amount of categorical feature transformation. The transition from a non-numeric state to numeric values can be a time-consuming and difficult job in feature engineering, however, CatBoost eliminates this step. CatBoost is based on decision tree and gradient boosting theory. The primary principle behind boosting is to sequentially integrate multiple weak models that perform marginally better than a chance to generate a strong competitive prediction model using greedy search. Because gradient boosting sequentially fits decision trees, the fitted trees will learn from previous trees' failures and thereby reduce errors. This procedure of adding additional functions to old ones is repeated until the loss function chosen is no longer minimized. CatBoost does not use similar gradient boosting models in the decision tree growth process. Instead, CatBoost develops oblivious trees, which are trees that are produced by requiring that all nodes at the same level

must test the same predictor with the same condition, allowing for the calculation of a leaf's index using bitwise operations. The oblivious tree technique has a simple fitting strategy and is CPU-efficient, while the tree structure acts as a regularisation to discover the best solution and avoid overfitting. CatBoost yielded an F1-Score of 0.9830.

Finally, we tried ensemble learning, the act of systematically generating and combining various models, such as classifiers or experts, to address a computational intelligence challenge. Ensemble learning is largely used to improve a model's performance classification, prediction or to lessen the risk of an unintentional poor model selection. We ensembled all 3 proposed models, but the model's F1-Score decreased, only yielding an F1-Score of 0.9660.

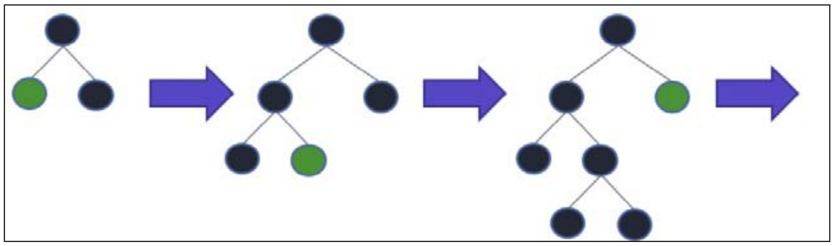

Figure 3: Shows the difference in tree growth between XGBoost and LightGBM. Leaf-wise splits increase complexity and may result in overfitting; however, this can be avoided by specifying the max-depth option, which specifies the depth to which splitting will occur.

7. Conclusion and Future Works

In this paper, we proposed a machine learning model for wealth management advisory while presenting a possible standard for ranking variables that could affect the decision-making of advisors to better aid their clients. Results show that our XGBoost model yielded the best accuracy.

We aim to further improve this model by training on a larger dataset to ensure the model's robustness and push the model as an industry standard.

AlsoWe aim to perform fake news analysis on stock market-related news which we believe could add to the robustness of the model's prediction.

In addition, Adding more financial features relevant to the customer could add more accuracy of the model predication; select the most fitting facilities to customer and obtaining more profit to customer.

Also expanding the scope to include another investment industry rather stock market, such as gold industry, petroleum industry, real state industry …etc.

References

Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H., 2010. An empirical comparison of machine learning models for time series forecasting. Econometric reviews 29, 594–621.

Barsky, R.B., Juster, F.T., Kimball, M.S., Shapiro, M.D., 1997. Preference parameters and behavioral heterogeneity: An experimental approach in the health and retirement study. The Quarterly Journal of Economics 112, 537–579.

Bontempi, G., Ben Taieb, S., Borgne, Y.A.L., 2012. Machine learning strategies for time series forecasting, in: European business intelligence summer school, Springer. pp. 62–77.

Capponi, A., Olafsson, S., Zariphopoulou, T., 2021. Personalized robo-advising: Enhancing investment through client interaction. Management Science .

Casalicchio, G., Molnar, C., Bischl, B., 2018. Visualizing the feature importance for black box models, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. pp. 655–670.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794.

Gerhana, Y., Umam, M., Atmadja, A., 2021. Implementation of the markowitz method in the robo-advisor application for best investment selection, in: IOP Conference Series: Materials Science and Engineering, IOP Publishing. p. 032051.

Grealish, A., Kolm, P.N., 2021. Robo-advisory: From investing principles and algorithms to future developments.

Grealish, Adam and Petter N. Kolm, 2021, "Robo-Advisory: From Investing Principles and Algorithms to Future Developments" in Machine Learning in Financial Markets: A Guide to Contemporary Practice (2021). Edited by A. Capponi and C.A. Lehalle. Cambridge University Press.

Kaya, O., Schildbach, J., AG, D.B., Schneider, S., 2017. Robo-advice–a true innovation in asset management. Deutsche Bank Research , 1–16.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems 30.

Markowitz, H.M., 1968. Portfolio selection. Efficient Diversification of Investments. Yale university press. https://doi.org/10.12987/9780300191677

Nabi, R.M., Soran Ab M, S., Harron, H., 2020. A novel approach for stock price prediction using gradient boosting machine with feature engineering (gbm-wfe). Kurdistan Journal of Applied Research 5, 28–48.

Potdar, A., Pande, M., 2021. Comprehensive analysis of machine learning algorithms used in robo-advisory services, in: Journal of Physics: Conference Series, IOP Publishing. p. 062105.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A., 2018. Catboost: unbiased boosting with categorical features. Advances in neural information processing systems 31.

Wang, H., Yu, S., 2021. Robo-advising: Enhancing investment with inverse optimization and deep reinforcement learning, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE. pp. 365–372.

Wang, H., Zhou, X.Y., 2020. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. Mathematical Finance 30, 1273–1308.

Wang, P.Y., Liu, C.S., Yang, Y.C., Huang, S.H., 2019. A robo-advisor design using multiobjective ranknets with gated neural network structure, in: 2019 IEEE International Conference on Agents (ICA), IEEE. pp. 77–78.

Wojtas, M., Chen, K., 2020. Feature importance ranking for deep learning. Advances in Neural Information Processing Systems 33, 5105–5114.

Yang, J., Zhao, C., Yu, H., Chen, H., 2020. Use gbdt to predict the stock market. Procedia Computer Science 174, 161–171.

Yu, S., Chen, Y., Dong, C., 2020. Learning time varying risk preferences from investment portfolios using inverse optimization with applications on mutual funds .