INSTITUTO DE TECNOLOGIA E LIDERANÇA – INTELI

OPTIMIZING THE ACHIEVEMENT OF NNM TARGET FOR FINANCIAL ADVISORS

RAPHAEL LISBOA ANTUNES

SÃO PAULO

2025

RAPHAEL LISBOA ANTUNES

OPTIMIZING THE ACHIEVEMENT OF NNM TARGET FOR FINANCIAL ADVISORS

Report presented to the Instituto de Tecnologia e Liderança as an artifact of Module 13 of the Bachelor's Degree in Computer Science, in the corporate track.

Supervisor: Prof. Dr. Rafael Will Macedo de Araujo

São Paulo

2025

ABSTRACT

In the financial market, especially in the B2C segment, investment advisors encounter significant challenges in achieving their net fundraising targets, such as *Net New Money* (NNM). This project seeks to solve the lack of predictability in target attainment by implementing an algorithm using *Machine Learning* and Statistical Models. Beyond forecasting, the solution will offer optimized guidelines for advisors to refine their strategies by analyzing the behaviors of top-performing professionals. This approach aims to enhance efficiency and strategic decision-making for both advisors and team leaders.

Keywords: Net New Money, Predictability, Machine Learning, Statistical Models, Investment Advisors.

LIST OF ABBREVIATIONS AND ACRONYMS

NNM	Net New Money
AUC	Assets under Custody
B2C	Business to Consumer
SDR	Sales Development Representative
MTD	Month to Date
B3	Brazilian Stock Exchange
HR	Human Resources
D-1	Previous day
MVP	Minimum Viable Product
CGE	Code used to anonymize the advisor's identity.
KNN	K-Nearest Neighbors

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Problem Statement

In the financial market, particularly in the retail segment within the B2C model, clients with an invested wealth exceeding a minimum threshold gain access to an investment advisor. These professionals play a crucial role in assisting clients in selecting the best strategies for resource management, guiding them in investment allocation according to their financial targets, risk profile, and time horizon.

As a result, advisors receive monthly targets distributed across various financial products such as Fixed Income, Variable Income, and Banking, aiming to drive the institution's results. Among these targets, Net New Money (NNM) — a metric representing the net balance between funds raised and withdrawn by clients — stands out as one of the biggest challenges, as achieving it requires acquiring new funds. Unlike other targets, which can be met through strategic restructuring of clients' portfolios.

In addition to advisors, professionals in leadership roles, known as Team Leaders, must have access to a forecast of target achievement at the end of the month based on daily behavior throughout the period. This allows for identifying deviations, helping an advisor who initially would not meet their target to realign their strategy and, therefore, have a chance to achieve the objective by the end of the period.

1.2 Project Objectives

To address the proposed issue, this project aims to solve the pain point of advisors and team leaders who lack an accurate prediction of target achievement at the end of the month. Therefore, an algorithm will be developed using Machine Learning and Statistical Models.

With statistically reliable forecasts, the second phase of the project focuses on providing optimized guidelines for advisors who need to adjust their strategies. By analyzing the behavior of top-performing advisors, key features contributing to success will be extracted, helping those with insufficient results adopt more effective practices and enhance their performance.

2 MAPPING AND ANALYSIS OF THE CURRENT PROCESS

2.1 Target Achievement

Currently, to achieve the NNM target, advisors use two main strategies. The first involves their existing client base. In this case, the advisor contacts clients to assess whether they have funds invested in other institutions. Based on this analysis, the advisor attempts to convince them to transfer their investments to their institution. During the conversation, the advisor may not secure an immediate transfer but establishes a commitment with the client to make a deposit at some point in the future. This potential future gain is recorded in a management platform, referred to as the Pipeline. However, as expected, not all recorded amounts are effectively deposited into the institution. This practice serves primarily as an approximate control tool for estimating the amount to be captured over time.

The second strategy involves the SDR (Sales Development Representative) commercial prospecting, consisting of professionals who function as call center representatives. They bridge the gap between potential clients with substantial financial capacity who do not yet have an account with the institution and investment advisors. This way, advisors can meet the NNM target by acquiring new clients.

2.2 Predictability in target achievement

For predictability, a *Forecast* metric is currently used. It is a linear metric calculated daily, where the returned value represents the percentage of target achievement at the end of the month based on the funds raised until the previous day (D-1). It is calculated considering only business days, meaning only the days the advisor works. Therefore, in determining the total number of days, the following are excluded: weekends, B3-recognized holidays, and the advisor’s pre-approved vacation period set with the institution’s HR department.

The formula for calculating this metric is:

$$VP = VR \times \frac{BD}{WD} \qquad \text{Forecast} = \frac{VP}{MT}$$

Where the variables represent:

- **VP** – Total projected captured volume for the end of the month.
- **VR** – Financial volume raised by the advisor throughout the month.
- **BD** – Number of business days in the month.
- **WD** – Number of days worked by the advisor.
- **MT** – Monthly target.

2.3 Target Achievement Flowchart

The flowchart detailing the target achievement process is presented in **Figure A.1** in **Appendix A**.

3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

3.1 Functional requirements

- **Access Control**: The system must be able to differentiate access levels for each user, restricting advisors to view only their own numbers and Team Leaders to view only their designated advisors.
- **Target Prediction**: The Team Leader must be able to view, through a dashboard, the predicted target achievement for each of their advisors at any time, based on their behavior throughout the month. Likewise, advisors should be able to see their own predictions.
- **Daily Updates**: The system must recalculate the prediction daily based on historical data from the beginning of the month until the previous day (*Month to Date - MTD*).
- **Performance Alerts**: The Team Leader must be able to identify which advisors have a predicted percentage below a predetermined threshold and view other segmentations based on predicted percentage.
- **Strategy Suggestions**: The system must recommend optimized strategic actions based on top-performing advisors to those at risk of not meeting their targets.

3.2 Non-functional requirements

- **Scalability**: The system must be capable of processing and analyzing large volumes of data without performance degradation.
- **Reliability**: The model must demonstrate statistically significant accuracy to ensure reliable predictions.
- **Intuitive Interface**: The dashboard must present numbers and results clearly to avoid misinterpretation.
- **Continuous Updates**: The algorithm must be periodically updated to reflect new market trends.
- **Computational Efficiency**: The model must be optimized to run efficiently without requiring excessively expensive infrastructure.

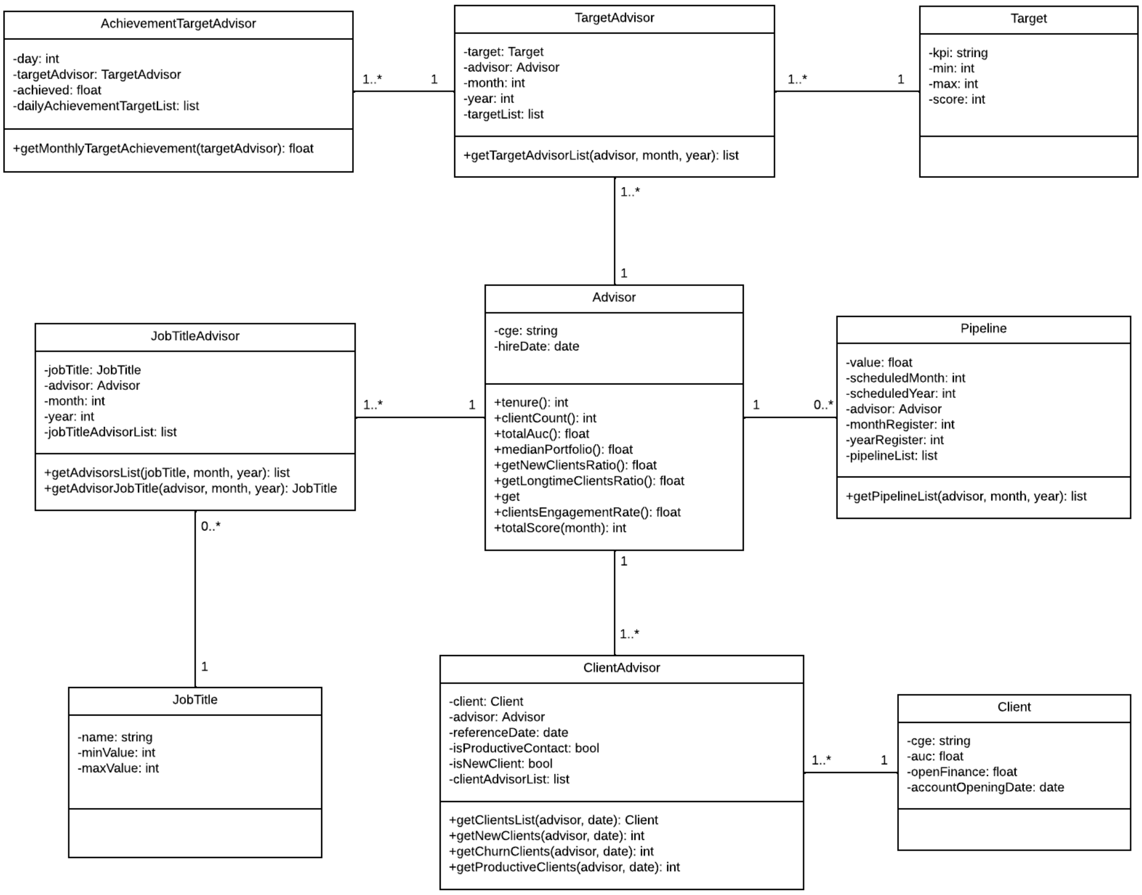
4 SYSTEM ARCHITECTURE

This section describes the system architecture, detailing its organizational structure, the types of information users can interpret from the presented data, and how these data elements are grouped and interact. To represent these components, we employ two standard UML diagrams: the use case diagram and the class diagram. These diagrams are essential tools for visually modeling software systems, helping improve clarity, communication, and understanding of system behavior and structure.

UML (Unified Modeling Language) combines techniques from data modeling, business modeling, object modeling, and component modeling to provide a standardized methodology that simplifies system design and documentation (Lucidchart, n.d.).

4.1 Class Diagram

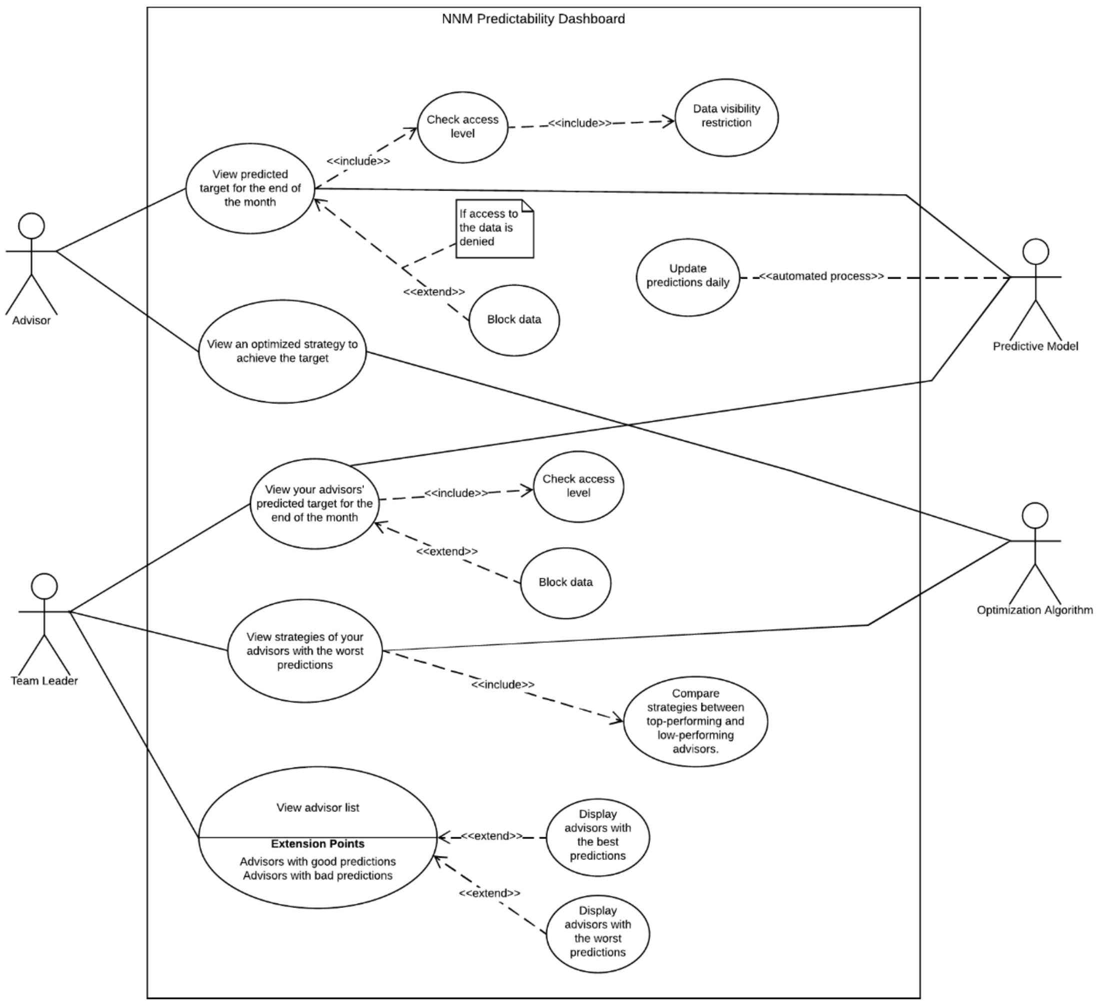
In a class diagram, the primary entities of our data structure are represented as classes, each encapsulating attributes and methods that define their properties and behaviors. These classes are interconnected, illustrating their relationships and dependencies, which are often characterized by specific cardinalities. A class diagram is “an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML)” (TechTarget, 2019).



Given that the goal is to periodically monitor the advisor's performance throughout the days of the month, to predict the achievement of their targets by the end of the period, the data structure is naturally centered around the advisor. This is evident from the fact that all other classes are connected to the Advisor class, indicating its role as the core element of the system. Some information has been organized into specific classes to provide greater flexibility and avoid data redundancy — for example, the Job Title class, which allows the advisor's position to be represented across different months without unnecessary duplication. Similarly, information related to clients has been encapsulated in dedicated entities, allowing the advisor to have an aggregated view of their client base while preserving the individuality and specificity of each client. The same can be observed in the cases of Pipeline and Target, which, despite having monthly individuality, were structured to support monitoring at different time levels — whether monthly or daily — a feature that was envisioned during the development of the system's data structure.

4.2 Use Case Diagram

Unlike the class diagram discussed in section 4.1, which focuses on organizing data and elucidating their interrelationships, the primary objective of the use case diagram is to illustrate the interactions between the system and its users. This diagram clarifies who will utilize the system, the manner of usage, and the functionalities accessible to the user, thereby providing a comprehensive understanding of how the previously discussed data will be presented to the end user. A use case diagram “depicts the functionality or behavior of a system from the user’s perspective” (Creately, 2022).



The main goal of this project is to make the advisor's life easier by allowing them to monitor their performance in a personalized way and continuously find ways to optimize it. To achieve this, it is essential to build a system that delivers this information in an accessible manner with good usability, both for the advisor and the team leader. Good UX/UI design is crucial for ensuring that data is clearly understood, since, as FEW (2006) points out, effective data visualization relies more on how users perceive and process visual information than on purely graphic design skills. It is the understanding of data and the ability to communicate visually and precisely that ensures a functional design.

The diagram presented represents the user's journey when using the final dashboard, which was designed to convey this information clearly and efficiently. It shows how the user interacts with the agents responsible for processing the data — represented by the predictive model and the optimization algorithm — reinforcing the importance of a well-structured visual design to facilitate this interaction.

5 CONCEPTUAL MODEL

The idea of this chapter is to present a Minimum Viable Product (MVP). In "The Lean Startup", RIES (2011) introduces the concept of the MVP as the simplest version of a product that allows a team to begin the learning process as quickly as possible. This approach enables startups to test their hypotheses with minimal resources, gather valuable customer feedback, and make informed decisions about product development. By focusing on delivering just enough features to attract early adopters, companies can avoid building products that customers do not want, thereby reducing wasted effort and resources.

In the case presented for this project, the goal is to identify potential issues that may arise soon during the development process, as well as to evaluate the performance of some machine learning models for this type of problem. For this first simplified solution, the database was structured based on the architecture proposed in Chapter 4, where the main features were designed for training four simple predictive models: Linear Regression, KNN, Decision Tree, and Random Forest.

5.1 Database

In "The Master Algorithm", DOMINGOS (2015) emphasizes the critical role of data in training effective machine learning models. He asserts that machine learning algorithms derive their predictive capabilities from the data they are trained on; thus, the quality and quantity of this data are paramount. Domingos highlights that the more relevant data an algorithm processes, the better it can learn and make accurate predictions. Based on this, for the development of the presented MVP, the following features were selected to ensure good model training, aiming for optimal performance.

- **Advisor:** Anonymized by a code (CGE), it represents the advisor responsible for all the features that will be mentioned below.
- **Job Title:** Advisor’s Job Title
- **Reference Date:** Within a range between January 2024 and January 2025, the dataset provides daily granularity for each advisor.
- **AuC:** Represents the advisor's total assets, determined by summing the assets of all clients.
- **Number of Clients:** The total number of clients for a specific advisor. The number changes based on the advisor's job title.
- **Average AuC:** Represents the ratio between the total AuC and the number of clients.
- **Tenure:** Time worked in months.
- **Target:** Represents the amount the advisor is expected to achieve during the month.
- **Target Achievement:** Represents the percentage of the target achieved.
- **KPI:** Financial product tied to the target.
- **New Accounts Ratio:** Represents the percentage of clients who have been with the advisor for three months or less.
- **Old Accounts Ratio:** Represents the percentage of clients who have been with the advisor for more than three months.
- **Productive Clients:** A productive client is one with whom the advisor managed to hold a phone call for more than 30 seconds, received a response to a

WhatsApp message within less than 24 hours or accepted a product recommendation through the investment institution's app.

- **Base Coverage:** The ratio between the number of productive clients and the total number of clients.
- **Pipeline Count:** Number of pipelines.
- **Pipeline Volume:** The total financial volume of pipelines.

5.2 Model Training

In this initial version, the model is trained using data from four advisors throughout the entire year of 2024. Once trained, the models will be compared against each other and against the Forecast method — previously introduced in Chapter 2.2 as the current approach for predicting goal achievement. Validation will be performed using unseen data from January 2025, presented one day at a time to check the model's end-of-month prediction. This step is crucial to assess the models' ability to generalize, helping to prevent overfitting — a phenomenon where the model memorizes training data patterns so precisely that it fails to perform well on new data (GOODFELLOW; BENGIO; COURVILLE, 2016).

5.2.1 Linear Regression

"Regression analysis is one of the most widely used techniques for analyzing multifactor data. Its broad appeal and usefulness result from the conceptually logical process of using an equation to express the relationship between a variable of interest (the response) and a set of related predictor variables" (MONTGOMERY; PECK; VINING, 2012, p. 15).

Owing to its simplicity, Linear Regression is the only model among those tested that does not rely on tunable parameters affecting training performance. As such, the training metrics for its single configuration are presented below.

R^2	Mean Error
0.9803	0.0263

5.2.2 KNN (K-Nearest Neighbors):

In "Pattern Recognition and Machine Learning", BISHOP (2006) introduces KNN as a straightforward, instance-based learning algorithm used for classification and regression tasks. The core idea is that a data point is classified based on the majority class among its 'k' nearest neighbors in the feature space. This method relies on a distance metric, commonly the Euclidean distance, to determine the proximity between data points. Additionally, Bishop addresses the importance of the choice of 'k' and the distance metric, as these factors significantly influence the algorithm's performance. He also discusses the computational challenges associated with KNN, particularly when dealing with large datasets, and suggests potential solutions to mitigate these issues.

To select the best model, four different values of K were evaluated.

	R^2	Mean Error
K = 2	0.9910	0.0082
K = 3	0.9916	0.0110
K = 5	0.9853	0.0190
K = 7	0.9803	0.0242

Based on the highest R^2 value, the model with the parameter K = 3 was chosen as the most efficient in training.

5.2.3 Decision Tree

In “Understanding Machine Learning: From Theory to Algorithms”, SHALEV-SHWARTZ; BEN-DAVID (2014) explains that decision trees function by recursively partitioning the input space into regions, each associated with a specific prediction. This partitioning is achieved through a series of tests on feature values, forming a tree-like structure where each internal node represents a decision rule based on a feature, each branch corresponds to an outcome of the rule, and each leaf node signifies a predicted output. They also discuss how the maximum depth of a decision tree influences its performance. A shallow tree may underfit by failing to capture essential patterns, leading to lower accuracy. Conversely, an overly deep tree can be overfit by modeling noise in the training data, reducing its generalization to unseen data. Therefore, it is crucial to balance tree depth to ensure the model generalizes well.

As previously explained, to find the depth that maximizes the model's performance, four different max depths were tested.

	R^2	**Mean Error**
Max depth = 2	0.8289	0.0805
Max depth = 3	0.9248	0.0510
Max depth = 10	0.9972	0.0012
Max depth = 20	0.9999	0.0001

Based again on the R^2 metric, the best-performing model was the one with max depth = 20.

5.2.4 Random Forest

According to "Random Forests", BRIEMAN (2001) introduces an ensemble learning method that enhances the accuracy and stability of classification and regression models by combining multiple decision trees. In this approach, each tree is constructed using a random vector sampled independently, ensuring diverse tree structures. The final prediction is determined by aggregating the outputs of all trees, typically through majority voting for classification or averaging for regression. For instance, a study by OSHIRO *et al.* (2012) suggests that while adding more trees can improve performance, there exists a threshold beyond which additional trees do not significantly enhance accuracy and only increase computational cost. They recommend using between 64 and 128 trees to achieve a balance between performance and resource usage.

To determine the best model, three different numbers of trees were tested.

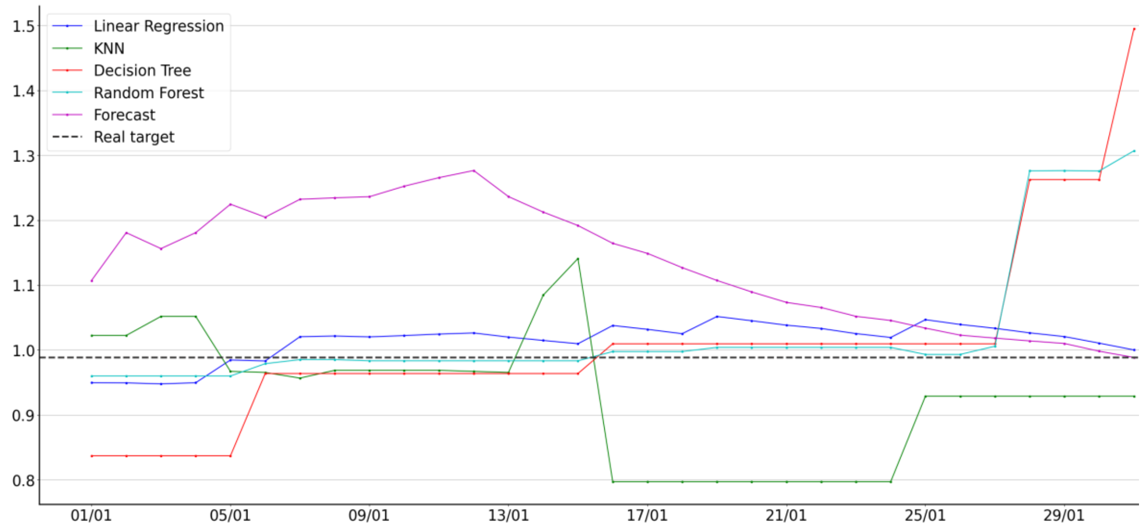
	R^2	Mean Error
N Tree = 20	0.9990	0.0016
N Tree = 100	0.9992	0.0014
N Tree = 200	0.9991	0.0015

Once again, using the R^2 metric to determine the best model, the chosen Random Forest model was the one with N Tree = 100

5.3 Comparison Between Models

To analyze the performance of the previously mentioned models, the January 2025 data — never seen by the models before — was presented to them iteratively. The models had to predict the end-of-month result based on the data available for each day.

The results obtained were:



At first glance, Linear Regression achieved a result even superior to Forecast. However, when evaluating the same model with other advisors, the results proved to be inconsistent, showing variable performance depending on the advisor, leading to the belief that the models did not actually learn meaningful patterns, but rather performed well by chance in certain cases.

5.4 Results Obtained

Some models, depending on the case, may appear to deliver effective results. However, when evaluated with different advisors, it becomes evident that these results are often coincidental, and the model does not consistently perform well. Therefore, we conclude that all models still prove to be less efficient than the Forecast. Several hypotheses are proposed to explain this behavior:

- **Overfitting:** As previously explained, overfitting occurs when a model memorizes the data instead of learning meaningful patterns. This can happen due to several factors, such as high similarity between data points, low data volume, or weak correlations. This issue is noticeable in the model training phase, where despite parameter adjustments, all models exhibit an R^2 value very close to 1. In "Five Reasons Why Your R-squared can be Too High", FROST (2013) explains that in regression analysis, R-squared (R^2) measures the proportion of variability in the dependent variable that is explained by the independent variables. A high R^2 indicates that the model accounts for a large portion of the variability in the response variable. However, an excessively high R^2 can sometimes be a sign of overfitting.
- **Order of data during training**: The model was trained using a common machine learning technique called Train-Test Split, which randomly separates a portion of the dataset — typically 70% for training and 30% for testing — ensuring the model is evaluated with unseen data. However, in this specific case, chronological order is crucial. To accurately predict the end-of-month target, the model needs to learn the advisor’s behavior day by day. Training with randomly shuffled data can introduce confusion and noise.

6 CONCLUSIONS

This study primarily aimed to address the inherent challenges in forecasting the achievement of the NNM target for financial advisors by integrating machine learning techniques and statistical modeling. Initially, a detailed analysis of the current process was conducted, identifying the limitations of traditional forecasting methods and the need for a more dynamic approach adapted to the evolving behavior of advisors throughout the month.

The implementation of the MVP allowed for experimentation with various predictive models which, despite exhibiting high efficiency in tests — with R^2 values close to 1 — revealed significant issues during practical application. Notably, the challenges encountered include the phenomenon of overfitting and the inadequacy of the temporal separation of data during model training. These factors compromise the generalizability of the forecasts, underscoring the necessity to expand and refine the dataset used.

Additionally, the proposal to incorporate a recommendation system based on the analysis of high-performing advisors' performance patterns appears promising. Although this functionality is still in its initial stage, it offers interesting prospects for optimizing operational strategies, contributing to the continuous improvement of results and more assertive decision-making by both advisors and team leaders.

Finally, the results obtained highlight the importance of adopting methodologies that consider the sequential nature of the data and making more precise adjustments in model parameterization. Therefore, it is recommended that future research incorporates temporal approaches — such as time series models and recurrent neural networks — as well as an expanded sampling scope to achieve more robust and representative forecasts of advisor behavior in different contexts.

In summary, the integration of advanced data analysis techniques in the context of financial target forecasting proves to be a promising path for the development of tools that foster more effective and sustainable resource management, offering relevant support for the evolution of market practices in the financial sector.

REFERENCES

LUCIDCHART. ***What is Unified Modeling Language***. Lucidchart, n.d. Available at: < https://www.lucidchart.com/pages/tutorial/uml>. Accessed on: 9 Apr. 2025.

TECHTARGET. ***What is a class diagram?***. TechTarget, May 2019. Available at: <https://www.techtarget.com/searchapparchitecture/definition/class-diagram>. Accessed on: 9 Apr. 2025.

CREATELY. ***Use Case Diagram Tutorial (Guide with Examples)***. Creately, Dec 2022. Available at: <https://creately.com/guides/use-case-diagram-tutorial/>. Accessed on: 9 Apr. 2025.

FEW, Stephen. ***Information Dashboard Design: The Effective Visual Communication of Data***. O'Reilly Media, 2006.

RIES, Eric. ***The Lean Startup: How today's entrepreneurs use continuous innovation to create radically successful businesses****.* New York: Crown Business, 2011.

DOMINGOS, Pedro. ***The Master Algorithm: How the quest for the ultimate learning machine will remake our world****.* New York: Basic Books, 2015.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. ***Deep learning***. Cambridge: MIT Press, 2016.

MONTGOMERY, Douglas C.; PECK, Elizabeth A.; VINING, G. Geoffrey. ***Introduction to Linear Regression Analysis***. 5. ed. Hoboken: Wiley, 2012.

BISHOP, Christopher M. ***Pattern Recognition and Machine Learning***. New York: Springer, 2006.

SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. ***Understanding Machine Learning: From Theory to Algorithms***. Cambridge: Cambridge University Press, 2014.

BREIMAN, Leo. ***Random Forests. Machine Learning***, v. 45, n. 1, 2001.

OSHIRO, Thais Mayumi; PEREZ, Pedro Santoro; BARANAUSKAS, José Augusto. ***How Many Trees in a Random Forest?***. In: *Proceedings of the 8th International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Berlin: Springer, 2012. p. 154–168.

FROST, Jim. ***Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?***, 2013, Available at: <https://blog.minitab.com/en/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit.> Accessed on: 9 Apr. 2025

APPENDIX A - Target Achievement Flowchart

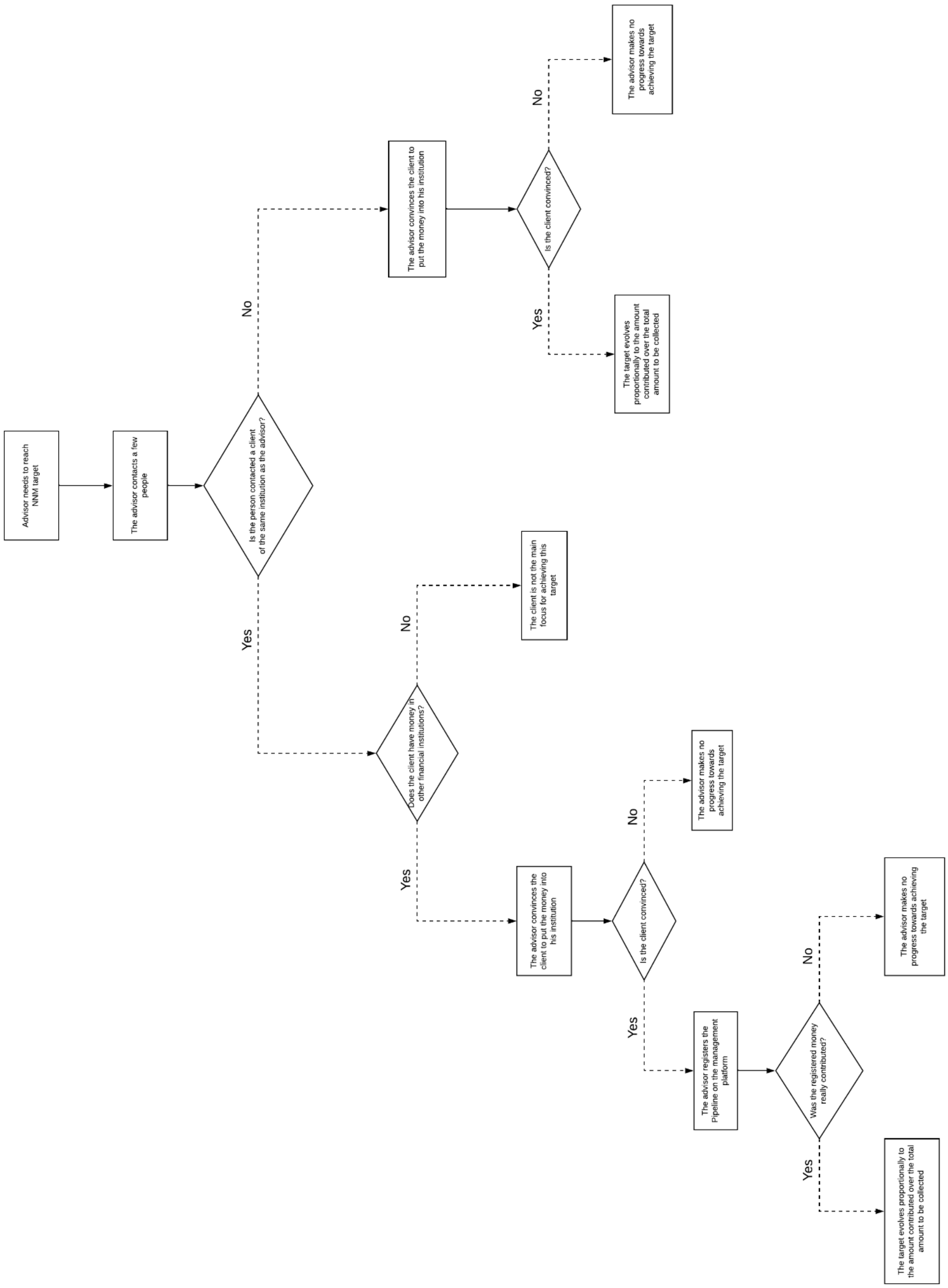


Figure A.1 – Detailed Flowchart of Target Achievement
Source: LISBOA (2025).