National Economics University

Faculty of Economical Mathematics

Bachelor Thesis

Revolutionizing Bank Marketing: Machine Learning Approaches for Predicting Term Deposit Subscriptions

Student: Tran Phuong Anh

Student ID: 11219258

Class: Data Science in Economics & Business 63

Instructor: PhD. Thi Khuyen LE

Ha Noi, May 2025

SUMMARY

As financial institutions face increasing pressure to personalize services and optimize the effectiveness of marketing campaigns, this study explores how machine learning can transform bank marketing by predicting customer subscription to term deposit products. By leveraging real-world data from a banking campaign including behavioral patterns, interaction history, and socio-economic indicators the research aims to uncover customer decision-making patterns and propose a scalable, data driven solution to improve targeting accuracy.

To achieve this objective, a comprehensive machine learning pipeline was developed, encompassing exploratory data analysis, one hot encoding for categorical variables, outlier detection and treatment, and min max feature scaling. Six supervised learning models were implemented, including Logistic Regression, Naive Bayes, Random Forest, Gradient Boosting, LightGBM, and CatBoost. The model training process was designed systematically, incorporating cross validation to ensure model stability and generalization capability, and hyperparameter tuning using randomized search to optimize predictive performance. In addition, model interpretability techniques such as SHAP and LIME were employed to ensure transparency and explainability at the individual prediction level.

The results show that ensemble models, particularly CatBoost, outperform traditional models in terms of balanced accuracy and probability calibration. Notably, CatBoost achieved a balanced accuracy of 0.8901 and a ROC-AUC of 0.9469, while also demonstrating stronger performance in handling uncertain or misclassified cases. These models provided valuable insights into key decision-driving features such as call duration, previous campaign outcomes, and macroeconomic indicators.

This research opens new directions for data informed decision making in the banking sector, while highlighting the role of combining machine learning models with explainable artificial intelligence to identify factors influencing customer subscription behavior. Accurate prediction of term deposit subscription not only improves marketing campaign efficiency but also contributes to higher conversion rates and more effective resource

allocation. Future research may focus on deploying models in real time environments, integrating multi channel behavioral data, and experimenting with advanced techniques such as ordered target encoding and Bayesian optimization to further enhance accuracy and overall model performance.

ACKNOWNLEDGMENT

This thesis marks the culmination of a long journey of learning, research, and personal growth. Throughout this process, I have been fortunate to receive tremendous support, guidance, and encouragement from my lecturers, family, friends, and colleagues. I would like to express my deepest gratitude to everyone who has contributed to this meaningful journey.

First and foremost, I would like to express my sincere appreciation to Ms. Khuyen, my academic advisor, who has been wholeheartedly dedicated and supportive from the very beginning. With her patience, sense of responsibility, and invaluable academic insights, she not only helped me shape a clear direction for this thesis but also inspired me with academic passion and confidence to overcome challenges during the research process. Even when I recognized my own shortcomings, she consistently encouraged and supported me, creating the best possible conditions for me to complete this thesis to the best of my ability. I am truly grateful for her dedication and enthusiasm.

My heartfelt thanks also go to the lecturers of the Faculty of Mathematical Economics at the National Economics University. Their knowledge, passion, and enthusiastic teaching have laid a strong academic foundation that empowered me to pursue this research with confidence.

I am especially thankful to my family for their unconditional love and constant encouragement, which gave me the strength to keep going. I am equally grateful to my friends and colleagues for their support, conversations, and companionship that kept me motivated.

Lastly, I sincerely thank everyone who has contributed, directly or indirectly, to the completion of this thesis. Every act of support, encouragement, and companionship has been a precious source of strength, helping me overcome difficulties and stay committed to my goals. I deeply appreciate all the kindness and support I have received throughout this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1.

INTRODUCTION

1.1 Background & Motivation

In the context of an increasingly competitive and dynamic financial sector, banks are under constant pressure to optimize their marketing efforts and retain valuable customers. One of the most critical products offered by banks is the term deposit, which provides not only a stable source of funding but also reflects the level of trust and long-term commitment customers have toward a financial institution. Understanding the factors that influence a customer's decision to subscribe to a term deposit has, therefore, become a strategic imperative for banks (Moro et al., 2014).

Term deposits, also known as time or fixed deposits, play a fundamental role in modern banking systems. For customers, they offer a low-risk investment option with a guaranteed return over a specified period. For banks, they serve two essential functions. First, they provide a stable and predictable source of funding. Unlike demand deposits, term deposits are locked in for a set duration, which enables banks to manage liquidity and asset-liability more effectively (Bikker and Metzemakers, 2005). Second, term deposits are widely regarded as a signal of customer loyalty and trust. Clients who choose to commit their funds for extended periods typically exhibit stronger engagement with the institution, creating opportunities for cross-selling and long-term relationship management.

However, many banks continue to rely on traditional marketing approaches, such as demographic segmentation and basic statistical analysis, which often fall short in capturing the intricate patterns of customer behavior. These limitations have become more evident in a digital age where customer preferences are rapidly evolving, and a one-size-fits-all strategy is no longer effective (Sharma, 2024). Moreover, traditional methods are generally static, unable to adapt in real-time to shifts in customer sentiment, financial behavior, or external market influences.

To address these challenges, the integration of machine learning (ML) in bank marketing

has emerged as a promising solution. ML algorithms, by nature, are capable of handling large-scale datasets with high dimensionality, uncovering complex, non-linear relationships between variables that are often missed by conventional techniques. For instance, predictive models such as Decision Trees, Random Forests, Support Vector Machines, Naïve Bayes, and Neural Networks have been employed to forecast customer subscription behavior, yielding encouraging results in terms of accuracy, recall, and F1-score.

A prominent case study demonstrating this application is the Bank Marketing Dataset from the UCI Machine Learning Repository, which contains real-world data collected from direct marketing campaigns of a Portuguese bank. (Moro et al., 2014) used this dataset to build predictive models and provided strong evidence of ML's potential in improving targeting efficiency. Building on this foundation, subsequent studies have applied more sophisticated techniques and optimization strategies to further enhance model performance.

In the Vietnamese context, the trend toward digital transformation in the banking sector further highlights the relevance of this research direction. Recent studies show that many Vietnamese commercial banks, especially the Big 4 (Vietcombank, VietinBank, BIDV, and Agribank), are actively investing in AI and ML technologies to streamline operations and personalize customer engagement (Thu and Quan, 2023). The shift from intuition-based marketing toward data-driven decision-making is no longer optional but essential for maintaining competitiveness in the digital economy.

Based on this, the study focuses on systematically exploring and analyzing data (EDA) to assess the relationship between input features and the likelihood of customers subscribing to term deposits. Next, several popular machine learning models are implemented to compare classification performance, with metrics such as balanced accuracy, sensitivity, and specificity. In addition, SHAP analysis is employed to interpret the contributions of each feature to the model's predictions, providing insights into how individual factors influence the likelihood of subscription. The study also delves into the analysis of misclassified cases and observations with uncertain prediction probabilities, thereby shedding light on the limitations of the models and suggesting improvement directions. While it does not aim to develop a large-scale practical system, the research is expected to provide a practical perspective on the application of ML in banking marketing and serve as a reference for similar problems in the financial sector.

1.2 Object of the Study

This study aims to improve the predictive capacity of banking marketing systems by applying advanced machine learning techniques to forecast customer behavior, particularly in relation to term deposit subscription decisions. The first objective is to extract actionable insights from the marketing dataset by analyzing patterns and correlations between customer attributes and subscription outcomes. Based on these insights, the study will propose suitable data preprocessing strategies and feature engineering techniques to improve model input quality.

The second objective is to develop, implement, and evaluate a range of supervised machine learning models such as Logistic Regressions, Naive Bayes, Random Forest, LightGBM, Gradient Boosting and CatBoost with the goal of accurately predicting whether a given customer will subscribe to a term deposit. The comparative performance of these models will be assessed using relevant evaluation metrics.

A final and equally important objective is to ensure model interpretability and transparency. To this end, the study will incorporate Explainable Artificial Intelligence (XAI) approaches, specifically LIME (Local Interpretable Model-Agnostic Explanations) or SHAP (SHapley Additive exPlanations), to provide clear explanations of how input variables influence model predictions. This not only fosters trust in the model outputs but also supports strategic decision-making within the banking context.

1.3 Research Questions

This study is designed to address the following research questions:

- How does data quality and preprocessing affect the performance of machine learning models in predicting customer subscriptions to term deposit products?
- Which machine learning models demonstrate the highest predictive effectiveness in the context of bank marketing campaigns?
- What are the most influential features that contribute to a model's ability to accurately predict whether a customer will subscribe to a term deposit?
- Which techniques and training strategies can be applied to enhance the predictive performance and generalizability of machine learning models?

- How can Explainable AI (XAI) techniques, such as LIME or SHAP, be utilized to interpret model outputs and explain the rationale behind classification decisions?

1.4 Thesis Scope

This study focuses on the application of supervised machine learning models to predict whether a customer will subscribe to a term deposit product based on marketing campaign data. The research is limited to the analysis of the Bank Marketing Dataset provided by the UCI Machine Learning Repository, which contains data collected from a Portuguese bank's direct marketing campaigns.

The study concentrates on the classification task, using only structured tabular data with no additional customer behavioral data beyond the scope of the original dataset. The research does not cover other banking products or customer decisions unrelated to term deposits.

In terms of methodology, the scope is limited to a selection of supervised machine learning algorithms, including Naive Bayes, Logistic Regression, Random Forest, LightGBM, Gradient Boosting and CatBoost. The study emphasizes preprocessing techniques, model evaluation using metrics such as balanced accuracy, sensitivity, and specificity, and the application of Explainable AI (XAI) methods, specifically LIME or SHAP to enhance model interpretability.

The research is exploratory in nature and is conducted in an offline experimental environment. It does not aim to develop a full-scale deployment-ready system but rather seeks to generate analytical insights that can guide future development and implementation in real-world banking systems.

1.5 Thesis Structure

This thesis is structured into six main chapters, each addressing a specific aspect of the research process:

- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Material & Methodology

- Chapter 4: Results & Findings
- Chapter 5: Discussion
- Chapter 6: Conclusions & Perspective

CHAPTER 2.

LITERATURE REVIEW

2.1 Overview of Bank Marketing & Customer Behavior

In the evolving landscape of financial services, effective bank marketing has shifted from a one-size-fits-all approach to personalized and data-driven strategies. Understanding customer behavior is now at the core of modern marketing, especially as competition increases and customer expectations become more sophisticated (Kotler and Keller, 2016).

Customer relationship management (CRM) plays a foundational role in how banks collect, manage, and utilize customer information. In banking, CRM refers not only to software systems, but also to strategic processes that aim to build long-term relationships with clients by offering timely and relevant services based on their financial behavior and preferences (Peppers and Rogers, 2011).

By integrating CRM platforms with transaction history, interaction logs, and digital engagement data, banks can generate a 360-degree view of the customer (David Stone and David Woodcock, 2014). This enables better segmentation, personalized recommendations, and more efficient handling of marketing campaigns. In the context of term deposit subscription, CRM systems can help identify individuals with a propensity to save or invest, based on prior interactions or similar customer profiles.

As CRM systems become increasingly sophisticated, they are often enhanced with analytics and machine learning modules (Wang and Kim, 2017). These technologies transform CRM from a static customer database into a predictive engine - supporting proactive decision-making, early identification of potential subscribers, and optimal timing for outreach.

While demographic and geographic segmentation have long been used in bank marketing, behavioral segmentation offers a more actionable approach by grouping customers based on actual behaviors such as product usage, responsiveness to marketing, channel preference, or transaction patterns (Wedel and Kamakura, 1999). In contrast to static

attributes like age or occupation, behavioral data reflects dynamic and often predictive indicators of customer intent.

In the context of term deposit campaigns, behavioral segmentation allows banks to distinguish between passive account holders and those who frequently save large amounts or interact positively with prior promotions. These insights enable targeted marketing efforts, ensuring that resources are focused on the most promising customer segments.

Moreover, behavioral segmentation serves as a bridge to machine learning - based prediction models. Features derived from behavioral data (e.g., number of contacts, response history, prior outcomes) are often used as input variables in predictive algorithms (Moro et al., 2014). These models can automate the segmentation process, forecast likelihood to subscribe, and support more scalable, consistent marketing decisions.

In sum, behavioral segmentation and CRM form the analytical backbone for personalized bank marketing. When combined with machine learning, they empower financial institutions to predict customer behavior at scale-making them directly relevant and integral to the research conducted in this thesis.

2.2 Traditional Approaches for Predicting Customer Behavior in Marketing

As previously discussed, Customer Relationship Management (CRM) and customer segmentation serve as critical foundations for enabling businesses to understand customer needs and implement personalized marketing strategies. However, to operationalize these strategies in practice particularly in the banking sector, organizations have long relied on traditional predictive methods. Prior to the widespread adoption of machine learning, these conventional techniques were extensively used due to their simplicity, interpretability, and suitability for structured data.

Nevertheless, in the current data landscape characterized by increasing complexity and volume, the effectiveness of traditional approaches has gradually revealed limitations and now faces significant challenges in terms of efficiency and scalability.

Logistic Regression Among traditional techniques, logistic regression has been one of the most widely used tools in marketing analytics. Estimates the probability of a binary outcome, such as whether a customer will respond to a marketing campaign, based on

a linear combination of explanatory variables (Hosmer et al., 2013). In the context of banking, logistic regression has been applied to credit scoring, churn prediction, and customer targeting due to its ease of implementation, statistical interpretability, and relatively low data requirements (Malthouse and Blattberg, 2005). Nonetheless, the method assumes linearity in the log-odds and independence among predictors, which may not hold in real-world behavioral data. Its performance diminishes when faced with high-dimensional datasets, multicollinearity, and nonlinear relationship conditions often observed in complex customer interactions (Coussement and Van den Poel, 2008).

Rule-based Segmentation Another prevalent approach is rule-based segmentation, in which customers are divided into groups based on predefined rules such as age brackets, income thresholds, or product usage. These heuristics are typically derived from expert knowledge or past marketing outcomes (Bailey et al., 2009). For instance, a rule might define a "priority client" as someone with a monthly balance above a given threshold and no outstanding loans. While rule-based segmentation is simple and interpretable, it lacks adaptability. The static nature of the decision rules prevents the model from responding to changing customer behaviors or integrating real-time data. Furthermore, rule-based systems tend to oversimplify the diversity of behavioral patterns, potentially missing nuanced but meaningful trends.

Expert Systems Expert systems aim to codify human decision-making logic into structured sets of "if–then" rules. In financial services, they were historically used in advisory services, risk assessment, or service recommendations. These systems depend heavily on domain experts and are not capable of autonomous learning. Consequently, maintaining and updating such systems is time-consuming and prone to obsolescence as customer data evolves (Turban et al., 2005).

Limitations of Traditional Approaches While traditional methods laid the groundwork for early predictive analytics, they are increasingly inadequate for handling the complexity, scale, and dynamism of modern customer datasets. Key limitations include:

- Poor performance with nonlinear or high-dimensional data;
- Strong dependency on manual rule design and expert assumptions;

- Inability to learn from new data without intervention;
- Lack of scalability and adaptability in dynamic marketing environments.

These drawbacks highlight the need for more robust and intelligent approaches. As customer behavior becomes more complex and fast-changing, especially in financial services, machine learning methods offer a powerful alternative. They not only address the limitations of traditional techniques but also enable more accurate, scalable, and automated predictions. This transition from conventional tools to data-driven algorithms forms the methodological foundation of this thesis, which focuses on applying advanced models to predict customer subscription to term deposits.

2.3 Applications of Machine Learning in Marketing & Customer Prediction

Rise of Data-Driven Marketing In the banking industry, data-driven strategies are especially valuable due to the richness of transactional and behavioral data. Financial institutions now employ machine learning to extract actionable insights from large volumes of structured and unstructured data, enabling a more dynamic, personalized approach to customer engagement. This shift aligns with the broader trend of digital transformation in banking, which emphasizes automation, customer-centricity, and predictive intelligence (Nguyen, 2024).

Common Machine Learning Models used in Banking A wide range of machine learning algorithms have been adopted in marketing analytics and customer behavior modeling. Among the most widely used are:

- **Support Vector Machines (SVM)**: Suitable for binary classification tasks such as predicting campaign response. SVMs perform well in high-dimensional spaces and are effective in identifying nonlinear boundaries (Cortes and Vapnik, 1995).
- **Random Forest**: A robust ensemble method based on decision trees, often used for feature importance analysis and predictive modeling due to its low risk of overfitting and high accuracy (Breiman, 2001).
- **Extreme Gradient Boosting (XGBoost)**: An advanced boosting algorithm known

for its efficiency and performance in handling structured data and imbalanced classification problems, such as predicting rare events (Chen and Guestrin, 2016).

These models are capable of automatically capturing complex patterns in customer data and are increasingly integrated into intelligent CRM systems.

Applications in Marketing & Customer Analytics Machine learning has demonstrated strong potential in a variety of marketing-related applications across the banking sector, including:

- **Churn prediction**: Identifying customers likely to leave the bank, allowing proactive retention strategies (Verbeke et al., 2011).
- **Loan approval and risk assessment**: Enhancing credit scoring models by learning from historical approval data (Lessmann et al., 2015).
- **Cross-selling and upselling**: Recommending relevant products based on behavioral patterns and transaction history (Qiu et al., 2009).
- **Term deposit subscription prediction**: Forecasting the likelihood of customer response to savings campaigns - a key focus of this thesis (Moro et al., 2014).

By learning directly from data, machine learning models offer substantial advantages in predictive accuracy, adaptability, and scalability. These characteristics make them ideal for modern marketing environments where customer preferences are diverse and constantly evolving.

2.4 Overview of Related Work

The UCI Bank Marketing Dataset, originally provided by (Moro et al., 2014), has become a benchmark for research in predictive marketing, particularly in banking. It includes rich information on customer demographics, campaign interactions, and macroeconomic indicators, making it suitable for various classification and customer response prediction tasks. This section reviews several studies that employed this dataset and evaluates their methods, findings, and limitations.

Summary of Studies & Model used One of the earliest and most influential studies was conducted by Moro et al. (2014), who applied data mining techniques such as Logistic Regression, Support Vector Machines (SVM), Decision Trees, and Naive Bayes. Their study highlighted that advanced machine learning methods outperformed traditional statistical models in predictive performance.

Building on this foundation, recent studies have focused on improving model performance through structured data preprocessing pipelines and the application of ensemble learning methods. For example, Borugadda et al. (2021) and Peter et al. (2025) proposed workflows that include categorical variable encoding, feature normalization, and class imbalance handling using techniques such as SMOTE. These steps have been shown to enhance model stability and generalization, especially in imbalanced classification problems.

Logistic regression remains a reliable baseline model due to its simplicity and interpretability, achieving accuracy levels up to 92.72% (Borugadda et al., 2021). However, ensemble methods such as Gradient Boosting, XGBoost, and particularly Stacking have demonstrated superior predictive performance. In the study by Peter et al. (2025), the stacking model achieved 91.88% accuracy, an F1-score of 0.5972, and a ROC-AUC of 0.9491, outperforming all other tested models.

In terms of model evaluation techniques, most studies adopt 10-fold cross-validation. A notable exception is the study by Jiang (2018), which implemented models using the R programming language and employed an 80/20 train-test split to compare the performance of logistic regression, SVM, neural networks, decision trees, and Naïve Bayes.

Model evaluation is not limited to accuracy alone; common metrics also include precision, recall, and F1-score. Recent studies further incorporate advanced indicators such as the Matthews Correlation Coefficient (MCC) and Cohen's Kappa, which provide deeper insights into model performance under imbalanced data conditions (Peter et al., 2025).

While Naïve Bayes generally yields lower accuracy (ranging from 66% to 86%), it often delivers relatively high recall, making it valuable in scenarios where minimizing false negatives is critical Palaniappan et al. (2017) and Huang (2024). Models such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) also show competitive performance when supported by proper data scaling and hyperparameter tuning, though

they may face scalability limitations when applied to large datasets.

Comparison of Performance & Findings Most studies agree that tree-based ensemble models (e.g., Random Forest, XGBoost) outperform traditional classifiers such as Naive Bayes or Logistic Regression in terms of accuracy and F1-score. However, there is limited consensus on evaluation metrics. While some prioritize accuracy, others emphasize precision, recall, or AUC, depending on whether the business objective is minimizing false positives or false negatives. Furthermore, few studies report robust cross-validation techniques or perform hyperparameter tuning systematically. This inconsistency raises concerns about the reproducibility and generalizability of results across different settings.

Identified Research Gaps Despite the extensive application of machine learning models on this dataset, certain research gaps remain:

- **Lack of explainability**: Most studies focus on accuracy but neglect model interpretability, which is crucial in banking contexts where transparency is essential for regulatory compliance and customer trust;
- **Limited use of explainable AI (XAI) techniques**: Few studies incorporate methods such as SHAP or LIME to interpret model outputs;
- **Restricted evaluation metrics**: Many works do not evaluate sensitivity, specificity, or balanced accuracy, which are critical for imbalanced datasets;
- **Absence of post-hoc error analysis**: Misclassified cases are rarely investigated, which limits insight into customer segments that consistently yield prediction errors.

CHAPTER 3.

MATERIAL & METHODOLOGY

3.1 Material

3.1.1 Data Description

The dataset utilized in this study is the Bank Marketing dataset sourced from the UCI Machine Learning Repository (Moro et al., 2014). It contains information related to a direct marketing campaign conducted by a Portuguese banking institution, aiming to encourage customers to subscribe to term deposit products.

The dataset includes **41,188 observations** and **21 variables**, comprising 20 input features and 1 binary output target. The input features include both categorical and continuous variables, capturing information related to client demographics, previous banking interactions, and external macroeconomic conditions.

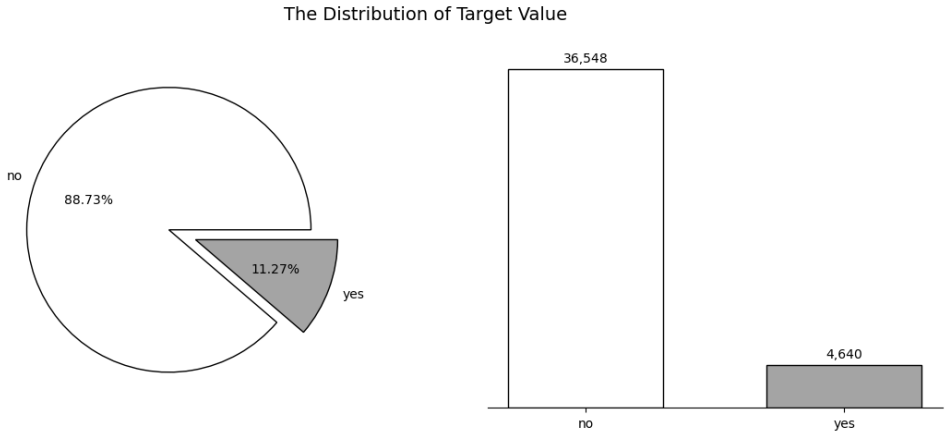


Figure 3.1: Distribution of the target value.

The target variable "y" indicates whether a client has subscribed to a term deposit, categorized as either "yes" or "no". Notably, the dataset exhibits a significant class imbalance, with only approximately 11% of the instances corresponding to positive responses (see in Figure 3.1). This imbalance necessitates the use of appropriate resampling strategies and evaluation metrics to ensure reliable model performance.

No.	Column name	Data type	Description
Bank client data			
1	age	Numeric	Age
2	job	Categorical	Type of job
3	marital	Categorical	Marital status
4	education	Categorical	Level of education
5	default	Categorical	Has credit in default?
6	housing	Categorical	Has housing loan?
7	loan	Categorical	Has personal loan?
Last contact of the current campaign			
8	contact	Categorical	Contact communication type
9	month	Categorical	Last contact month of year
10	day_of_week	Categorical	Last contact day of the week
11	duration	Numeric	Last contact duration, in seconds
Other attributes			
12	campaign	Numeric	Number of contacts performed during this campaign and for this client
13	pdays	Numeric	Number of days that passed by after the client was last contacted from a previous campaign
14	previous	Numeric	Number of contacts performed before this campaign and for this client
15	poutcome	Categorical	Outcome of the previous marketing campaign
Social & Economic context attributes			
16	emp.var.rate	Numeric	Employment variation rate – quarterly indicator
17	cons.price.idx	Numeric	Consumer price index – monthly indicator
18	cons.conf.idx	Numeric	Consumer confidence index - monthly indicator
19	euribor3m	Numeric	Euribor 3 month rate - daily indicator
20	nr.employed	Numeric	Number of employees - quarterly indicator
Target variable			
21	y	Binary	Has the client subscribed to a term deposit?

Table 3.1: Summary of dataset features.

To provide greater clarity on the structure of the dataset, a detailed summary of all input features including their names, data types, descriptions is presented in Table 3.1.

3.1.2 Exploratory Data Analysis (EDA)

This part presents an exploratory data analysis (EDA) process examining in detail each input variable in the banking dataset. The primary objective is to assess the distribution of variables, identify correlations with the target variable "y" (customer term deposit subscription status), and derive profound insights to support the predictive modeling process.

To begin, a t-SNE (t-distributed Stochastic Neighbor Embedding) visualization is applied to provide an overall view of the data structure and uncover potential separability between subscribed and non-subscribed customers in the reduced feature space.

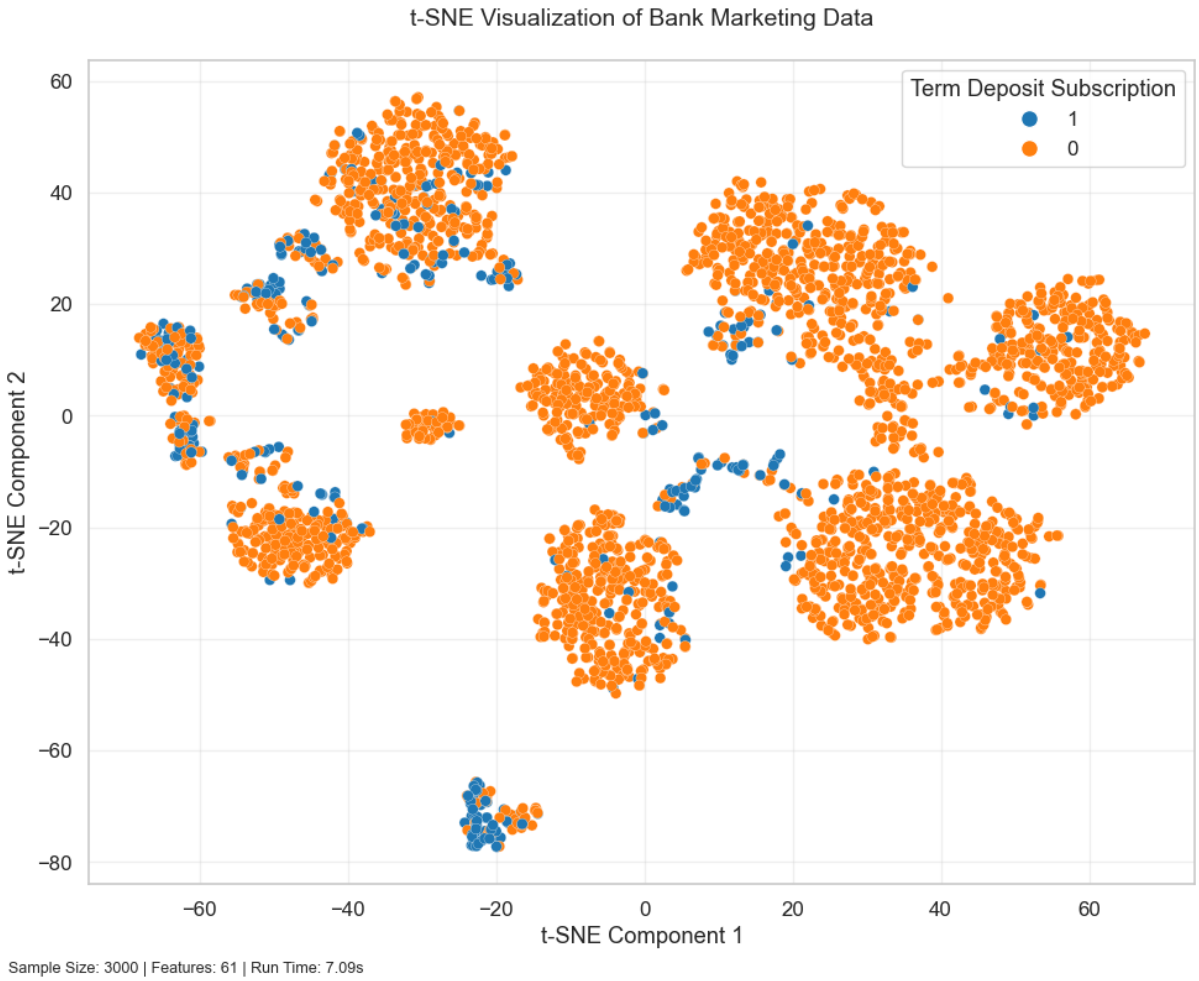


Figure 3.2: t-SNE visualization with 3000 random samples.

Figure 3.2 reveals several visible clusters, with subscribed customers mostly concentrated

in certain regions, while non-subscribers are more dispersed. This suggests that the input features may capture useful patterns for distinguishing between the two groups.

Univariate Analysis Following the overall structural view, a univariate analysis is conducted to explore the distribution of each individual feature.

Regarding the `job` variable, the most common occupations are `blue-collar`, `manage -ment`, and `technician`, whereas categories such as `student`, `unemployed`, and `housemaid` appear less frequently. In terms of the `marital` variable, the majority of customers are `married`, followed by `single` and `divorced` individuals. For `education`, most customers have university-level qualifications, while a smaller proportion completed only primary education. The `default` variable shows that most customers reported no credit default, although a considerable portion of entries are marked as `unknown`. Similarly, most customers do not hold `housing` or `loan` accounts, suggesting a relatively low level of financial obligation. Additionally, the `contact` method is predominantly `cellular`, indicating a move away from traditional landlines. `May` is the month with the highest volume of contact attempts, possibly linked to campaign strategy. Contact frequency across the `day_of_week` variable is relatively uniform. As for the outcome of previous campaigns, the `poutcome` distribution is heavily skewed toward a single category, implying limited prior interaction with most customers (see in Figure 3.3).

In addition to categorical variables, the distribution of numerical features is also examined to identify skewness, outliers, and potential data preprocessing needs.

Figure 3.3: Distribution of categorical features.

Among the numerical features, variables such as `age`, `campaign`, and `duration` exhibit right-skewed distributions, indicating the presence of several extreme values. The variable `pdays`, which represents the number of days since the client was last contacted

in a previous campaign, shows an extremely imbalanced distribution, with most values being 999 — likely a placeholder indicating no prior contact. Similarly, `previous`, which records the number of previous contacts before the current campaign, is heavily concentrated at zero, suggesting that most clients had not been approached in earlier campaigns. Furthermore, macroeconomic indicators such as `euribor3m`, `emp.var.rate`, `cons.price.idx`, `cons.conf.idx`, and `nr.employed` appear in distinct clusters, likely due to their periodic reporting intervals (monthly or quarterly), which results in a limited set of unique values (illustrated in Figure 3.4).

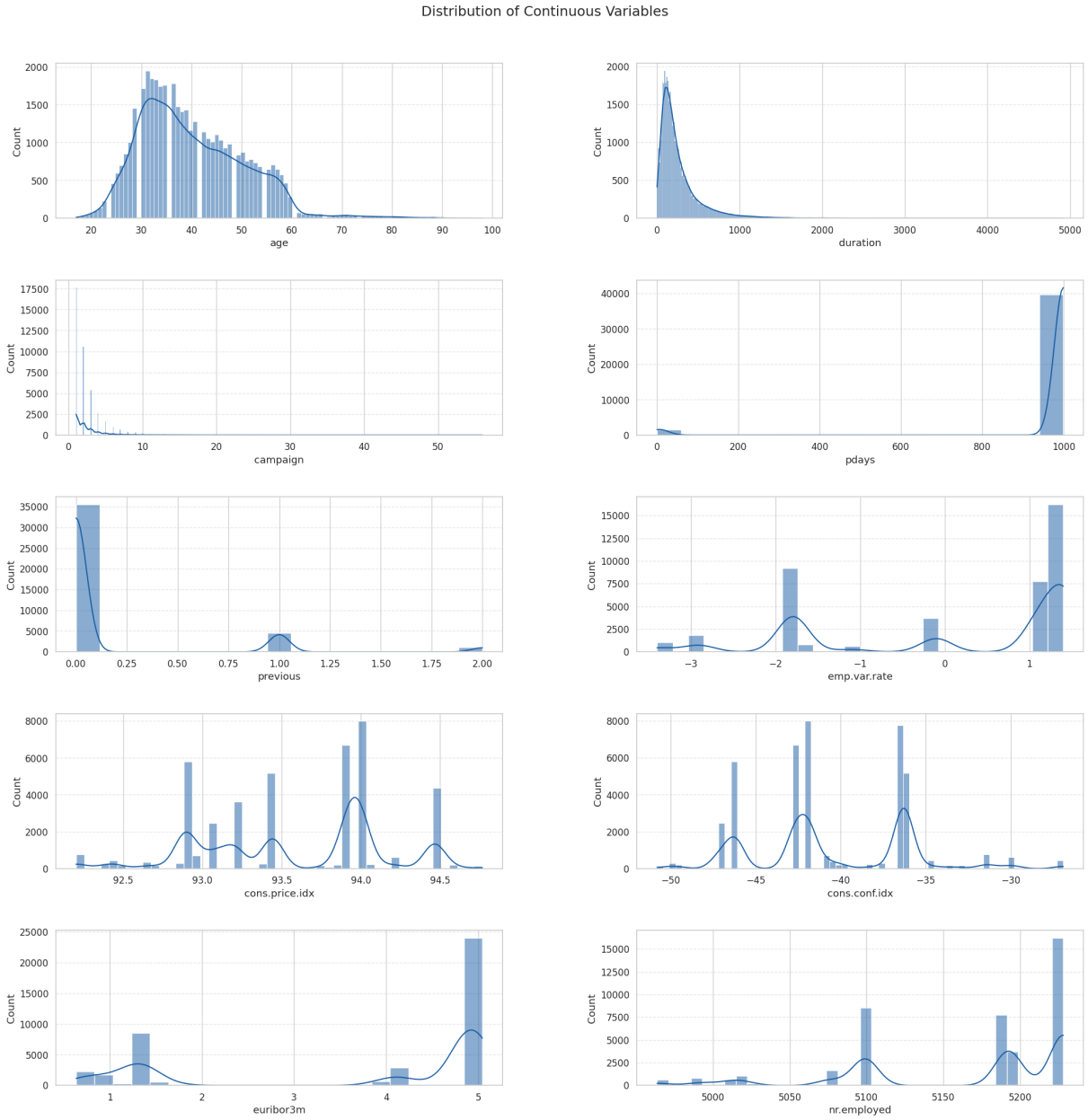


Figure 3.4: Distribution of numerical features.

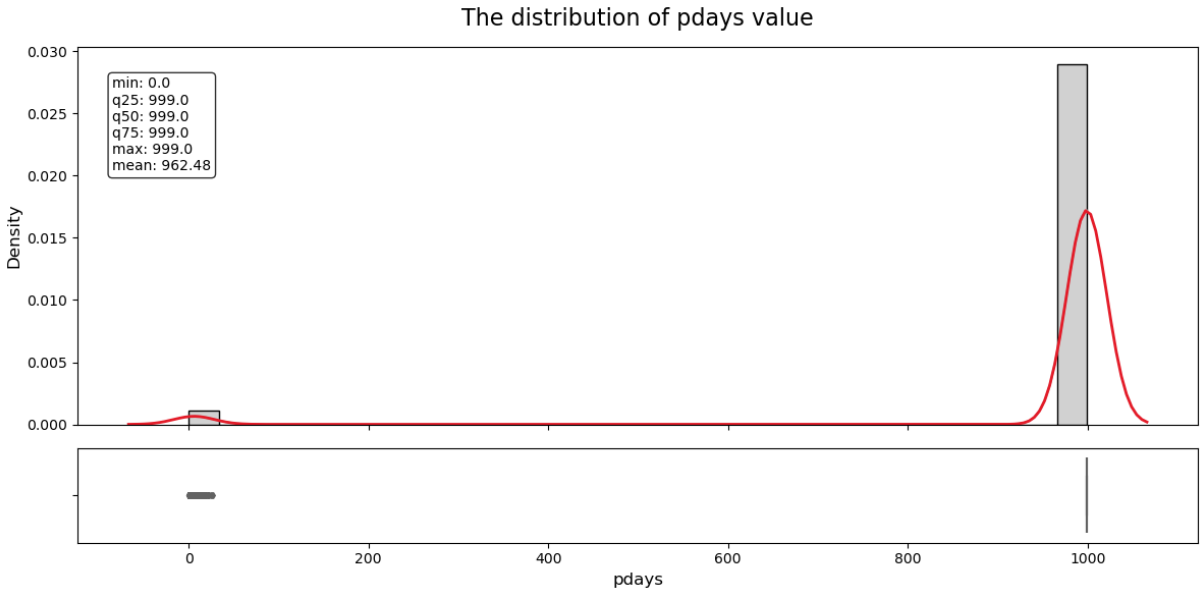


Figure 3.5: Distribution of `pdays` variable.

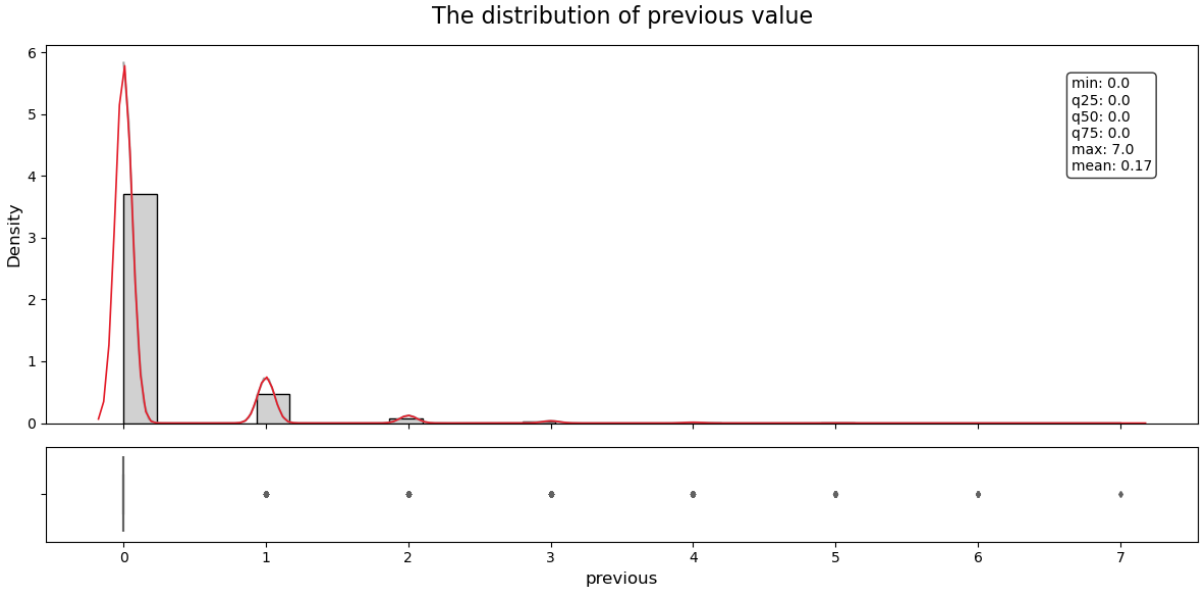


Figure 3.6: Distribution of `previous` variable.

Figure 3.5 and Figure 3.6 provides a closer look at the distribution of the `pdays` variable, alongside the previously discussed `previous` variable. As shown, the vast majority of `pdays` values are equal to 999, which is likely used as a placeholder to indicate that the client was never contacted in a previous campaign. This leads to an extremely imbalanced distribution. Similarly, the `previous` variable is highly concentrated at zero, reinforcing the observation that most clients had no prior interaction. In the data processing stage, appropriate transformation techniques will be applied to handle these special cases and

ensure better model interpretability.

Bivariate Analysis After examining the distribution of individual features, the analysis proceeds with bivariate exploration to investigate how each predictor variable relates to the target variable `y`.

Compared to the univariate distribution, it is notable that although categories like `student` and `retired` represent a relatively small portion of the dataset, they account for a disproportionately high number of positive responses (see in Fig. 3.7). In contrast, more prevalent groups such as `blue-collar` and `services` show lower subscription rates. Regarding marital status, `single` clients appear to be more likely to subscribe than `married` or `divorced` ones. In terms of education, clients with a `university.degree` or `high.school` qualification tend to show higher subscription rates. Additionally, clients with no `housing` or `personal loan`, and those contacted via `cellular`, are more likely to subscribe. As for the `month` variable, most positive responses occurred in `May`, indicating a potential campaign effect. Finally, the variable `poutcome` shows that clients who had a successful outcome in a previous campaign are much more likely to subscribe again.

Following the analysis of categorical variables, the relationship between numerical features and the target variable `y` is further investigated using boxplots to detect potential trends and separability between classes.

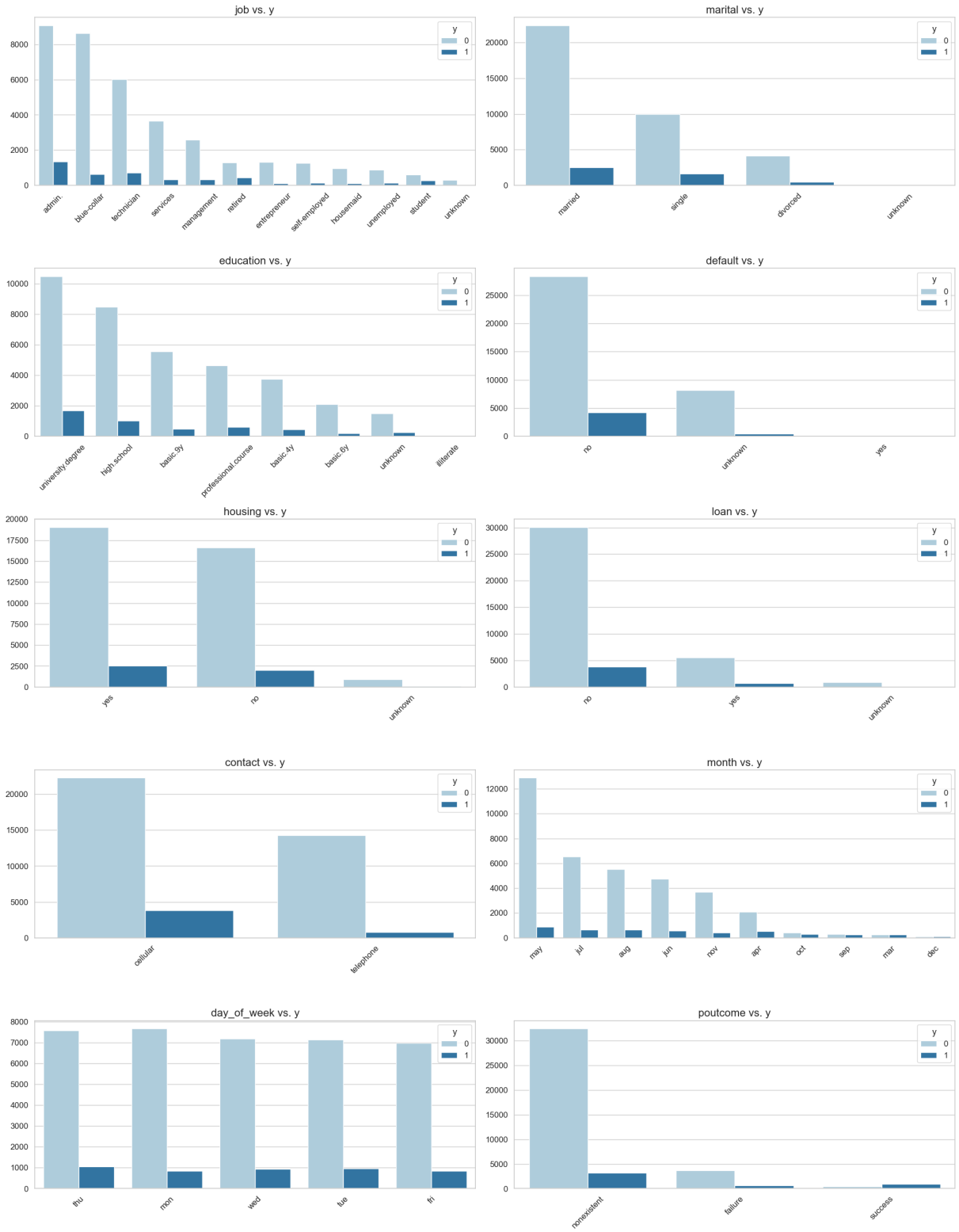


Figure 3.7: Distribution of categorical variables by subscription status.

Notably, clients who subscribed to term deposits tend to have longer `duration` values, indicating that successful outcomes are often associated with longer conversations. Moreover, those with higher `previous` values are more likely to subscribe, while the vari-

able `pdays` again shows a clear split between contacted and non-contacted individuals. Some economic indicators such as `euribor3m`, `emp.var.rate`, and `cons.conf.idx` also show observable shifts in central tendency between the two groups, suggesting possible correlations with subscription behavior (illustrated below in Fig. 3.8).

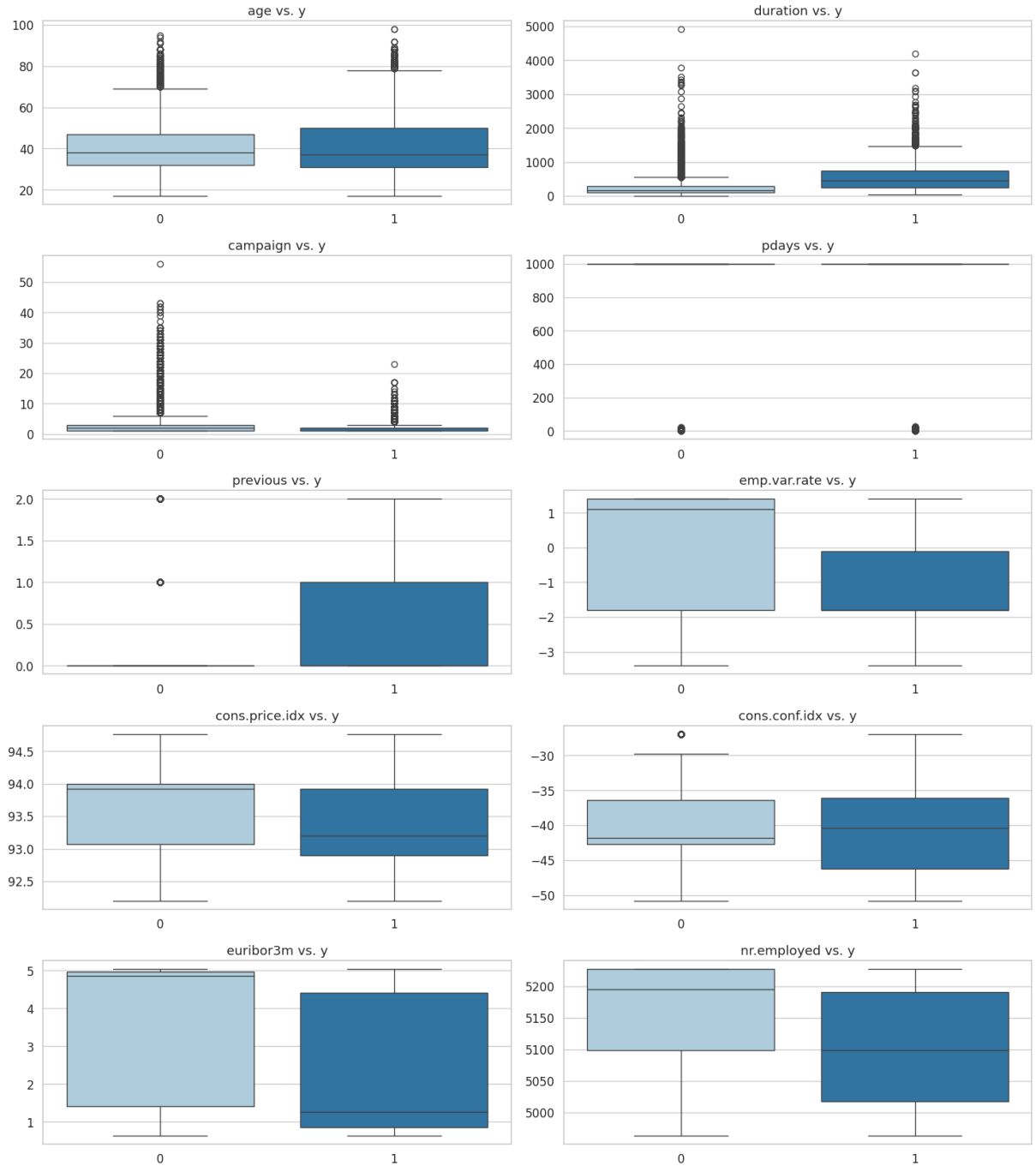


Figure 3.8: Boxplots of numerical features grouped by target.

Correlation of Numerical Features To further understand the interdependencies between numerical features, a correlation matrix is computed to examine linear relationships

and identify potential multicollinearity.

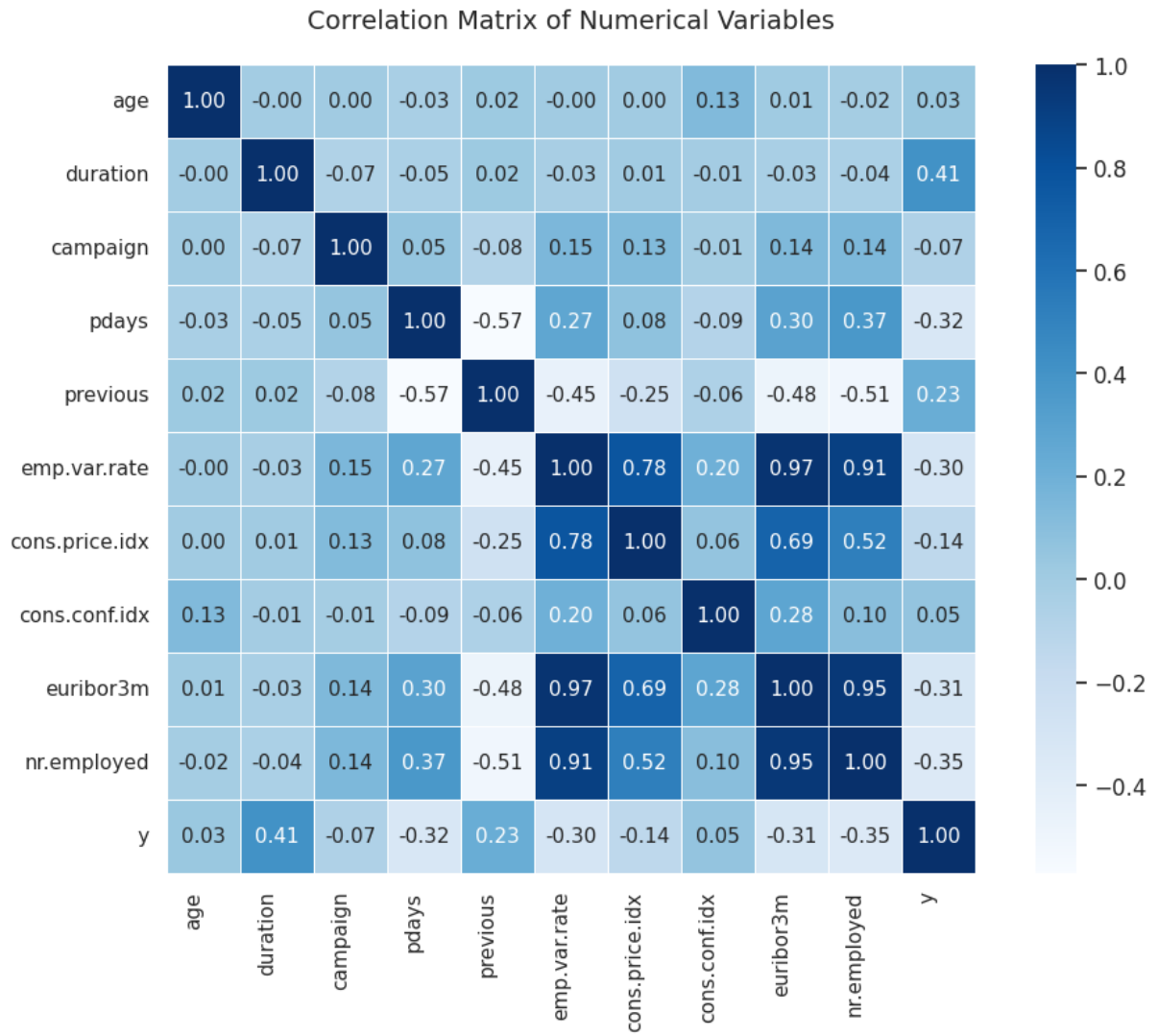


Figure 3.9: Correlation matrix of numerical features.

Figure 3.9 reveals several strong correlations among economic indicators. Notably, `euribor3m`, `nr.employed`, and `emp.var.rate` are highly correlated with one another (coefficients > 0.90), indicating possible multicollinearity. Meanwhile, the variable `duration` shows the strongest positive correlation with the target variable `y` ($r = 0.41$), consistent with previous boxplot findings. In contrast, variables such as `pdays`, `euribor3m`, and `nr.employed` are negatively correlated with `y`, suggesting that lower values of these indicators are associated with higher subscription rates.

Temporal Patterns To gain deeper insights into the correlations among socio-economic variables, this section analyzes their temporal trends, highlighting how these indicators

have changed over time and their potential relationship with customer subscription behavior.

The dataset bank-additional-full.csv comprises `41,188 observations` with `20 input variables`, chronologically ordered from May 2008 to November 2010 (Moro et al., 2014). Notably, this period coincides with the global financial crisis, creating a unique context for the research. Since the target variable directly relates to customers' financial decisions, it can be inferred that each time unit (month/year) carries distinct socio-economic conditions, significantly influencing users' financial behavior.

	age	job	marital	education	default	housing	loan	contact	month	year	...
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	2008	...
1	57	services	married	high.school	unknown	no	no	telephone	may	2008	...
2	37	services	married	high.school	no	yes	no	telephone	may	2008	...
3	40	admin.	married	basic.6y	no	no	no	telephone	may	2008	...
4	56	services	married	high.school	no	no	yes	telephone	may	2008	...

Figure 3.10: Illustration of the dataset after adding the auxiliary variable `year`.

To fully leverage the temporal aspect of the dataset, an auxiliary variable named year was derived from the existing date-related information, as illustrated in Figure 3.10. It is important to emphasize that this variable is used solely for exploratory analysis purposes and is excluded from the model training process in order to preserve the objectivity and generalizability of the predictive model.

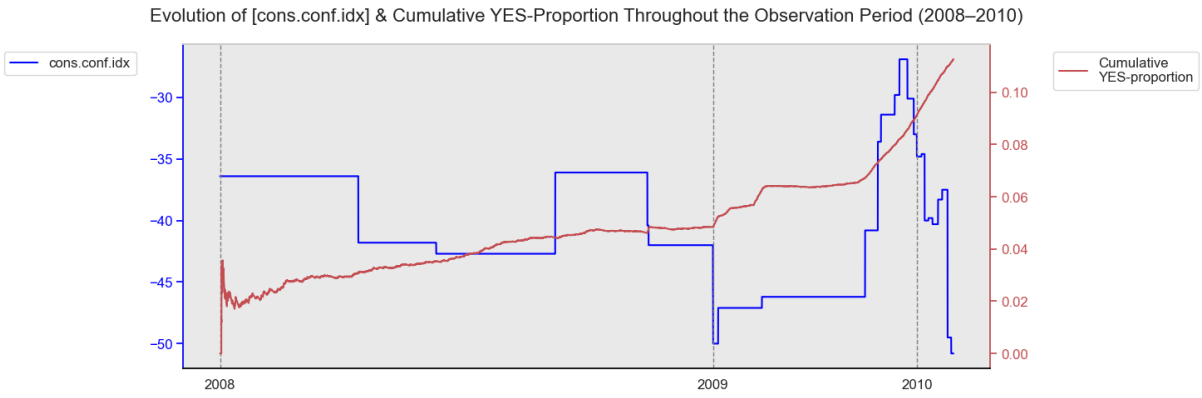


Figure 3.11: Trend of consumer confidence index (CCI) & cumulative term deposit subscription rate (2008–2010).

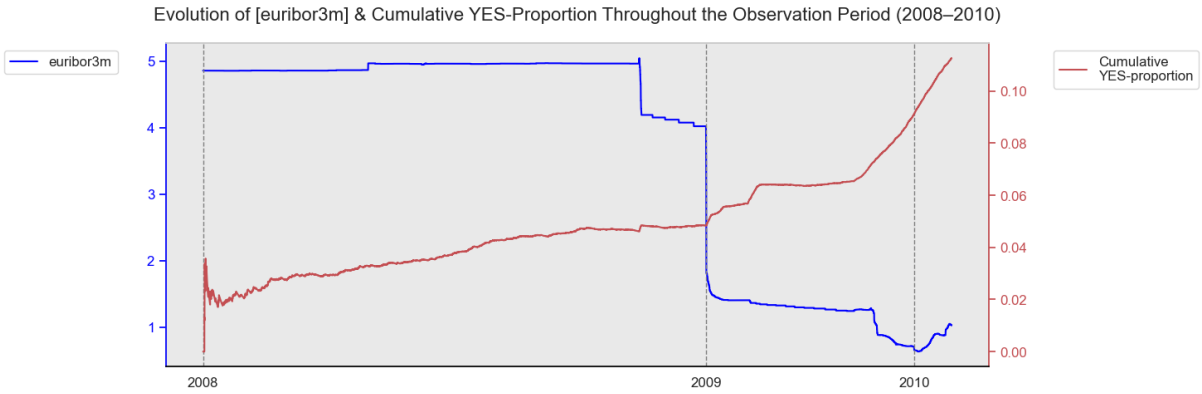


Figure 3.12: Trend of Euribor 3-month rate & cumulative term deposit subscription rate (2008–2010).

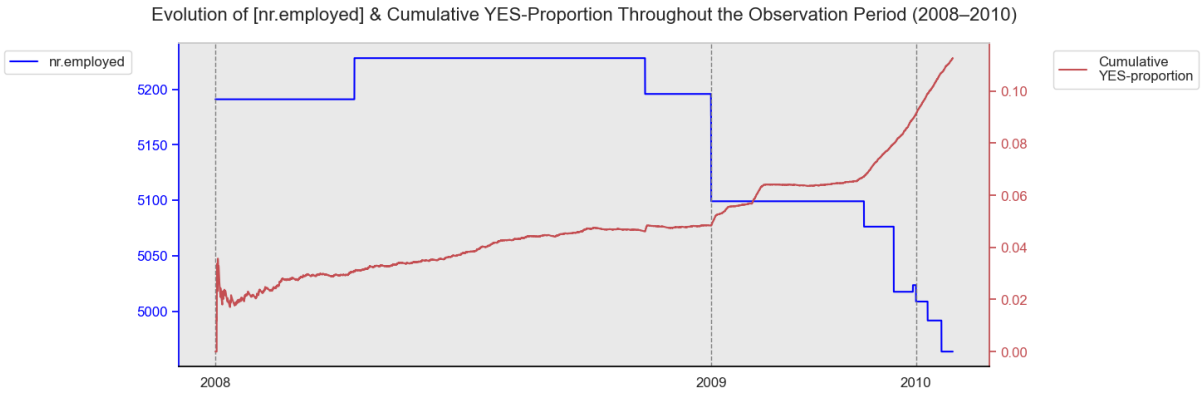


Figure 3.13: Evolution of number of employees & cumulative term deposit subscription rate (2008–2010).

Across the five socio-economic indicators analyzed, distinct temporal patterns emerge in relation to the cumulative term deposit subscription rate. Notably, three variables includes `euribor3m` (Fig. 3.12), `nr.employed` (Fig. 3.13), and `emp.var.rate` (Fig. 3.14) exhibit an inverse trend compared to the subscription rate. As these indicators decline over time, the proportion of successful term deposit subscriptions increases. In contrast, `cons.conf.idx` (consumer confidence Iindex - Fig. 3.11) and `cons.price.idx` (consumer price index - Fig. 3.15) show a similar upward trend to that of the subscription rate, particularly during the final phase of the observation period.

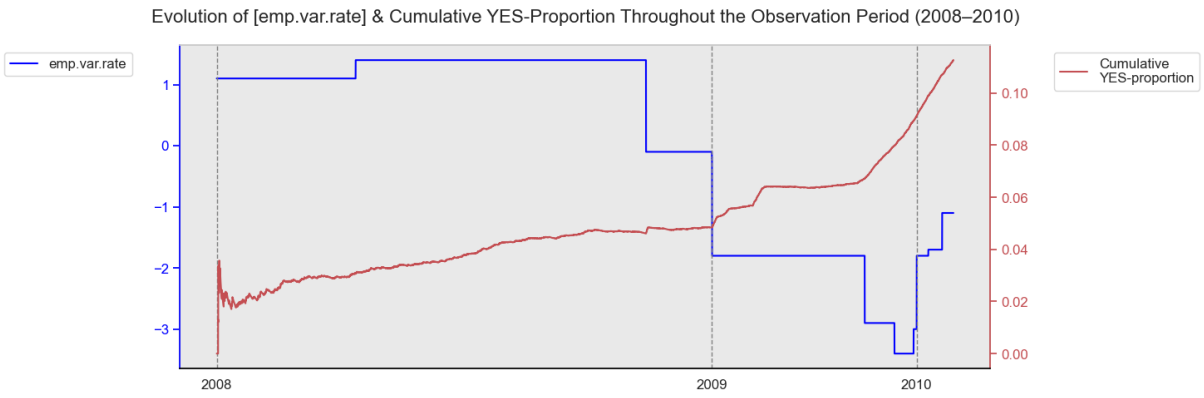


Figure 3.14: Quarterly employment variation rate & cumulative term deposit subscription rate (2008–2010).

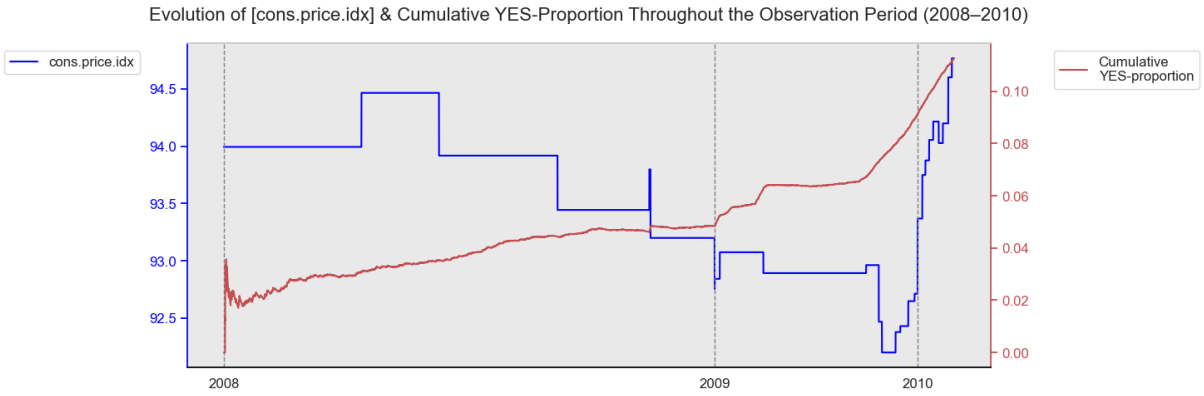


Figure 3.15: Trend of consumer price index (CPI) & cumulative term deposit subscription rate (2008–2010).

These contrasting directions suggest that macroeconomic conditions such as declining interest rates, employment shifts, and changes in consumer sentiment may collectively influence customers' financial decisions. The combination of indicators thus provides a valuable macro-level context for interpreting variations in campaign outcomes.

This section continues to examine the annual distribution of term deposit subscription outcomes. As illustrated in Figure 3.1, the dataset is unevenly distributed across years, with 2008 accounting for the majority of total records. However, it is important to note that data collection in 2008 started only in May, meaning the records cover slightly more than half of the year. Despite this, the subscription rate in 2008 remained relatively low. In contrast, 2010, although containing significantly fewer observations and ending in November, exhibited a notably higher conversion rate. This contrast highlights the critical importance of input variables, particularly socio-economic indicators and information

from previous campaigns. Moreover, since the data was collected during the period of the global financial crisis, it further supports the idea that macroeconomic conditions can significantly influence customer behavior.

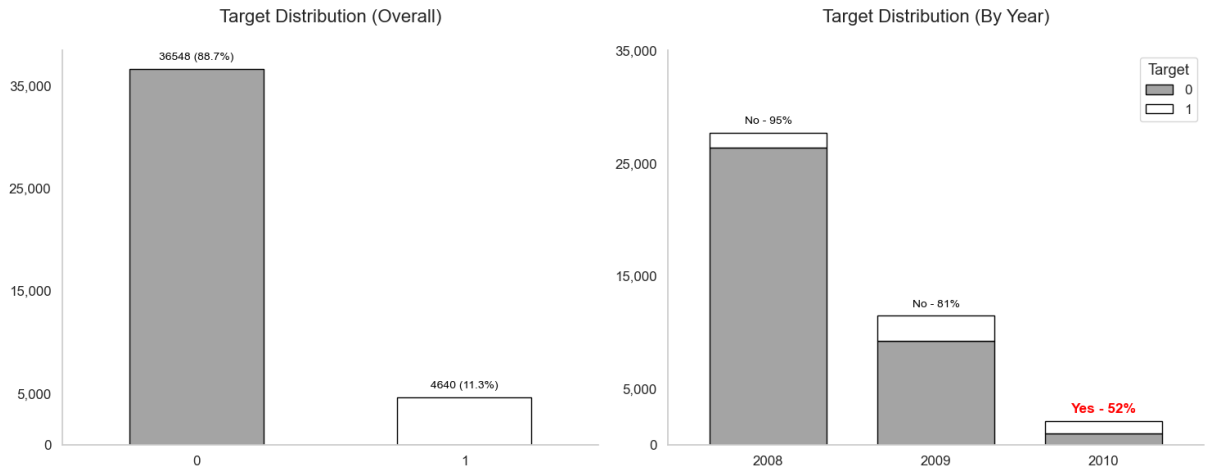


Figure 3.16: Distribution of term deposit subscription outcomes overall & by year.

In summary, the exploratory data analysis has revealed several important patterns and insights. Both categorical and numerical features show varying degrees of association with the subscription outcome, while certain socio-economic indicators exhibit strong temporal trends that align or diverge from customer behavior. These findings serve as a valuable foundation for subsequent preprocessing steps, including variable transformation, outlier handling, and the selection of relevant input features for model training. By addressing the observed data characteristics, the next phase aims to enhance model performance and interpretability.

3.1.3 Data Preprocessing

Handling Missing Values As an initial step in data preprocessing, the dataset was examined for missing values using summary statistics and visual checks. The inspection confirmed that no missing data were present in any variable. Therefore, no imputation techniques were required.

Handling Duplicate Values During the data collection process from the bank marketing system, the occurrence of duplicate records is unavoidable. Such duplicate entries can introduce bias into the model and distort prediction results. In this study, duplicate values were identified using the `duplicated()` function from the `pandas` library,

which detects observations sharing identical attribute values. To address this issue, a `First Occurrence Keep` strategy was applied only the first instance of each duplicate group was retained, while subsequent redundant entries were removed. This procedure ensures the representativeness and independence of the observations, thereby enhancing the reliability of the dataset for model training.

Handling Outliers Outlier values in variables can distort statistical relationships and negatively impact model performance. This research applies the Interquartile Range (IQR) method to identify and process outliers in the dataset.

The Interquartile Range (IQR) method was employed to detect outliers in numerical data, following the approach illustrated in Figure 3.17 below:

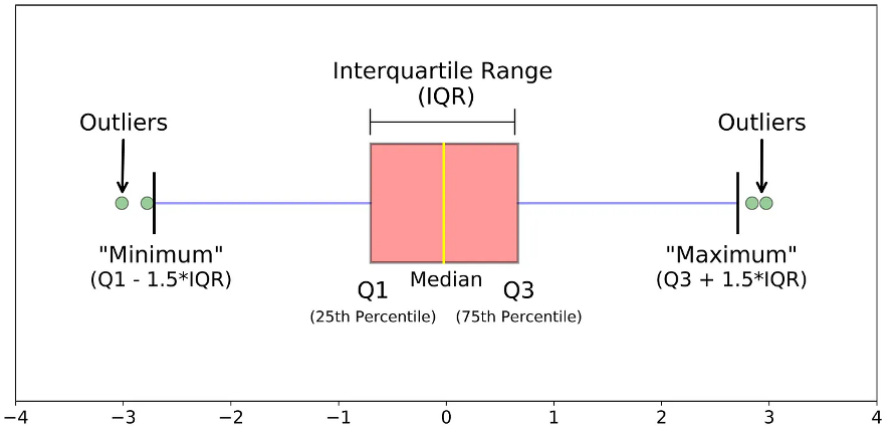


Figure 3.17: Illustration of the IQR method for outlier detection (Pham, 2020).

For each numerical variable, the first quartile (Q1 - 25th percentile) and third quartile (Q3 - 75th percentile) were calculated, and the IQR was defined as:

$$\text{IQR} = Q3 - Q1 \tag{3.1}$$

Using this, the lower and upper bounds were established as:

$$\text{Lower bound} = Q_1 - 1.5 \times \text{IQR} \tag{3.2}$$

$$\text{Upper bound} = Q_3 + 1.5 \times \text{IQR} \tag{3.3}$$

Any data points falling outside this range were flagged as outliers. After detecting

outliers, two primary approaches were considered:

- **Removal**: Eliminating observations containing outlier values from the dataset.
- **Winsorization**: Replacing extreme values with less extreme ones to reduce the impact of outliers while preserving the observation.

After evaluating these methods, the winsorization technique was prioritized based on the calculated IQR thresholds:

- Values below the lower bound were replaced with the lower bound value.
- Values above the upper bound were replaced with the upper bound value.

This approach allows for limiting extreme values within a defined threshold, rather than completely removing observations outside the interquartile range. Using winsorization helps preserve the original data while minimizing the impact of anomalous values on model performance. This is especially important in the context of banking and financial data, where distributions are often skewed and outliers may represent customer groups with special behaviors or high values, such as high-net-worth individuals. Retaining these observations not only maintains the integrity of the data distribution but also ensures that machine learning models can still extract valuable information from real-world data.

Feature Transformation In the dataset, the quantitative variable `pdays` (number of days since the last contact with the customer) contains a special value that requires processing. Specifically, the value `999` is used to indicate customers who have never been contacted in any previous marketing campaign. Therefore, this value does not represent an actual time period, but rather carries a binary classification meaning about the "never previously contacted" status.

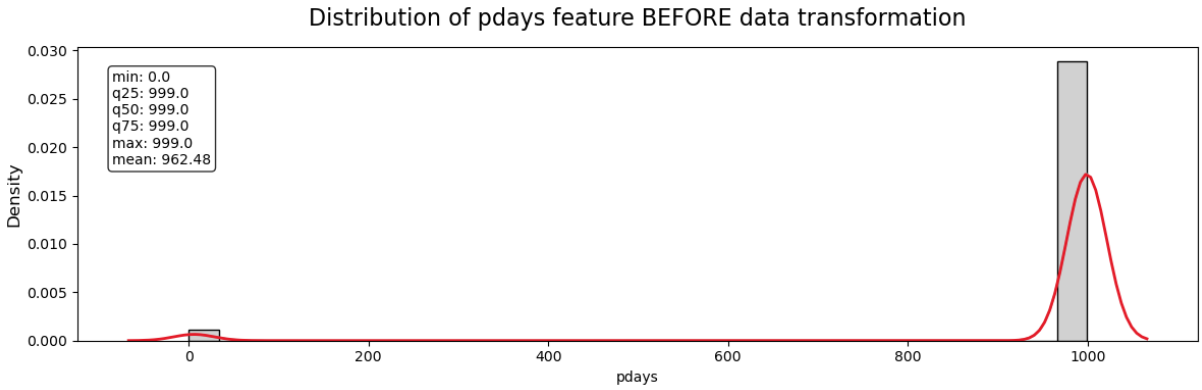


Figure 3.18: Distribution of the `pdays` feature before data transformation.

As illustrated in Figure 3.18, the value 999 constitutes a large portion of the distribution of the `pdays` variable, causing severe data skewness. Maintaining the value 999 in its numeric form could lead the model to misinterpret it as a large actual time period (e.g., 999 days since the last contact), thereby negatively affecting the learning process.

To address this issue, a transformation was performed by reassigning the value -1 to all observations with `pdays=999`. The result of this transformation process is shown in Figure 3.19. The purpose of this modification is to help the model clearly distinguish between two customer groups: (1) those who were not previously contacted and (2) those who have been approached in past campaigns. Simultaneously, this method helps minimize the skewness in the data distribution while preserving important information about customers' interaction history.

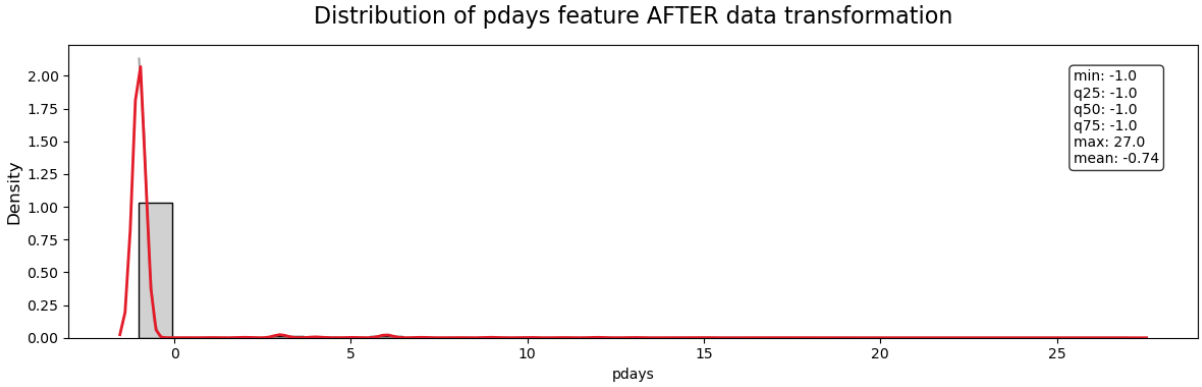


Figure 3.19: Distribution of the `pdays` feature after data transformation.

Feature Scaling During the data preprocessing phase, standardizing the scale of quantitative variables represents a critical step to ensure that features with different value

ranges do not disproportionately influence machine learning model performance. Min-Max Scaler is a data normalization method that scales the range of feature values to a fixed interval, typically [0, 1]. The transformation formula of Min-Max Scaler for a data point x on feature j is:

$$x'_j = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{3.4}$$

- x'_j is the normalized value of feature j;
- x_j is the original value of feature j;
- $\min(x_j)$ is the minimum value of feature j in the training dataset;
- $\max(x_j)$ is the maximum value of feature j in the training dataset.

The application of Min-Max Scaler facilitates bringing all quantitative variables to the same scale, thereby minimizing the possibility that features with larger values dominate during model training, especially with scale-sensitive algorithms such as Gradient Descent, Support Vector Machines.

In this research, Min-Max Scaler was applied to the following set of quantitative variables: `age`, `duration`, `campaign`, `pdays`, `previous`, `emp.var.rate`, `cons.price.idx`, `cons.conf.idx`, `euribor3m`, and `nr.employed`. This procedure was implemented after addressing the specific issues of the `pdays` variable (as described in the Feature Transformation section previously) to ensure all quantitative features reside within a consistent value range.

Feature Encoding Machine learning algorithms typically require input data in numerical format. Categorical variables cannot be used directly and need to be transformed. One-Hot Encoding is a popular encoding technique used to convert nominal categorical variables (those without a natural order) into a numerical format suitable for machine learning models.

This method works by creating new binary columns (values of `0` or `1`) for each unique category in the original categorical variable. For each observation, only the column corresponding to the category to which that observation belongs is assigned a value of `1`, while all other new columns have a value of `0`.

The main advantage of One-Hot Encoding is that it does not implicitly assign an ordinal relationship or weighting to categories, which is crucial for categorical variables without inherent ordering. This helps prevent the model from misinterpreting relationships between categories, which can occur if other encoding methods such as Label Encoding are used for nominal variables.

In this study, One-Hot Encoding was applied to the following set of categorical variables: `job`, `marital`, `education`, `default`, `housing`, `loan`, `contact`, `month`, `day_of_week`, and `poutcome`. This process transformed the categorical features into numerical representations, ready for input into machine learning models.

Handling Imbalanced Data The dataset used in this research, collected from marketing campaigns aimed at predicting customers' likelihood to subscribe to bank term deposits, faces a common challenge of class imbalance. This characteristic manifests itself in a significant disparity in the number of samples between the outcome classes. Specifically, the class of customers who did not subscribe ("`no`") constitutes approximately `11%` of total observations, forming the minority class, while the class of customers who subscribed ("`yes`") is the majority class with the overwhelming remaining proportion. This uneven distribution is illustrated in detail in Figure 3.1.

The severe imbalance between the sample sizes of these two classes creates certain difficulties in the predictive modeling process. Traditional classification machine learning algorithms tend to optimize performance on the majority class (the class of subscribing customers), leading to the potential of overlooking or misclassifying cases belonging to the minority class (the class of non-subscribing customers). Accurately predicting both classes is crucial for a comprehensive understanding of customer behavior and development of effective marketing strategies; therefore, appropriate processing methods are necessary to ensure the model is not biased.

Within the scope of this research, Random Oversampling, a technique belonging to the resampling group, was selected to address the imbalance in the training dataset. This technique operates by randomly duplicating samples from the minority class, thereby increasing the quantity of this class to achieve a more balanced distribution ratio between minority and majority classes in the training dataset.

To implement Random Oversampling accurately, the `RandomOverSampler` class provided by the `imblearn` library in the Python programming environment was utilized.

This process was performed only on the training dataset after the data had been divided into training and testing sets. Limiting the application of resampling techniques to the training set is necessary and mandatory to prevent data leakage from the test set into the model training process, thereby ensuring the objectivity and reliability of model performance evaluation results on previously unprocessed data. By randomly duplicating samples of the minority class in the training set, Random Oversampling helps increase the quantity of this class, thereby rebalancing the class distribution and providing the model with more data to learn the characteristics of the minority class, thus improving the ability to identify and predict cases belonging to this class.

Although Random Oversampling is a simple and easily implementable method, especially with the support of the `imblearn` library, this technique has the potential risk of leading to overfitting for the minority class. This stems from the fact that the method simply duplicates existing data samples without creating new information, causing the model to potentially learn too closely from repetitive samples. However, for this problem, it is a reasonable starting approach and has been applied to provide a more balanced training dataset for machine learning algorithms, with the expectation of improving classification capabilities and thereby enhancing the overall performance of the model across both classes.

3.2 Methodology

3.2.1 Model Selection

3.2.1.1 Baseline Models

To provide a performance reference point for advanced models, three widely-used classification algorithms with different operating principles were selected as baseline models in this research. These models include Logistic Regression and Naive Bayes.

Logistic Regression Logistic Regression is a foundational statistical and machine learning model widely applied and highly effective in binary classification problems. This model operates by estimating the probability that a given input observation $\mathbf{x}$ belongs to a specific class (typically the class of interest, denoted as `1`) based on a linear combination of the input features.

The core distinction from linear regression lies in its use of the logistic function (also known as the sigmoid function) to map the output of this linear combination to a probability value bounded between `0` and `1`.

The formula for calculating the probability that an observation $x = (x_1, x_2, \ldots, x_n)$ belongs to the positive class ($y = 1$) is expressed as follows:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}} \tag{3.5}$$

- x_i represents the value of the i-th feature;
- β_0 is the intercept term;
- β_i ($i = 1, \ldots, n$) are the coefficients (weights) corresponding to each feature.

These model parameters β_i are optimized during the training process, typically using the Maximum Likelihood Estimation (MLE) method, which seeks to find the set of parameters that maximizes the probability of observing the given class labels in the training data under the model. After computing the probability $P(y = 1 \mid \mathbf{x})$, the model makes a final class prediction by comparing this probability to a predefined decision threshold, commonly set at 0.5. If $P(y = 1 \mid \mathbf{x}) \geq 0.5$, the observation is classified into class 1; otherwise, it is classified into class 0.

Owing to its structural simplicity, computational efficiency, and the clear interpretability provided by its coefficients (indicating the influence of each feature on the log-odds of the outcome), Logistic Regression often serves as an effective and reliable baseline model for comparison against more complex classification algorithms. However, a limitation of this model is its assumption of a linear relationship between the features and the log-odds of the outcome, which may constrain its predictive performance on datasets exhibiting complex structures with strong non-linear patterns.

Naive Bayes Naive Bayes is a probabilistic classification algorithm derived from Bayes' Theorem, known for its efficiency and effectiveness in various real-world problems, particularly in text classification. The foundation of the algorithm is Bayes' Theorem, which allows calculating the probability of an event (a class label) occurring given that other events (feature values) are known. In the context of classification, Bayes' Theorem is formally stated as follows:

$$P(C_k \mid \mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{P(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid C_k) \cdot P(C_k)}{P(\mathbf{x}_1, \ldots, \mathbf{x}_n)} \tag{3.6}$$

- $P(C_k \mid \mathbf{x}_1, \ldots, \mathbf{x}_n)$ represents the posterior probability of class C_k given the features $\mathbf{x}_1, \ldots, \mathbf{x}_n$;
- $P(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid C_k)$ is the likelihood or the conditional probability of the features given class C_k;
- $P(C_k)$ is the prior probability of class C_k;
- $P(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the probability of the features (evidence).

To simplify the computation of $P(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid C_k)$, especially for high-dimensional feature spaces, the Naive Bayes algorithm introduces a "naive" assumption: that the features are conditionally independent given the class. Formally, this means

$$P(x_i \mid x_j \neq i, C_k) = P(x_i \mid C_k) \quad \text{for all } i \neq j. \tag{3.7}$$

With this assumption, the likelihood term $P(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid C_k)$ can be simplified into a product of individual conditional probabilities:

$$P(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid C_k) = \prod_{i=1}^{n} P(x_i \mid C_k) \tag{3.8}$$

Substituting this back into Bayes' Theorem, and noting that $P(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is a constant for a given input, the posterior probability is proportional to:

$$P(C_k \mid \mathbf{x}_1, \ldots, \mathbf{x}_n) \propto P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k) \tag{3.9}$$

The model predicts the class C_k that maximizes this posterior probability. The necessary probabilities ($P(C_k)$ and $P(x_i \mid C_k)$) are estimated directly from the training data, typically using Maximum Likelihood Estimation based on frequency counts.

To prevent the issue of zero probabilities for feature class combinations not observed in the training data which would make the entire product zero—smoothing techniques such as Laplace smoothing are commonly applied.

Naive Bayes is characterized by its computational efficiency, fast training time, and relatively low memory usage. It performs well in practice even when the conditional independence assumption is violated, particularly in domains where the assumption is approximately true or when dealing with high-dimensional data like text.

However, its strong assumption can limit its performance compared to more complex models in datasets with strong feature dependencies.

3.2.1.2 Ensemble Learning

Ensemble learning is a powerful technique that combines predictions from multiple individual machine learning models commonly referred to as base learners or weak learners into a single, aggregated model with superior performance compared to any individual constituent model. The fundamental idea is to leverage the diversity among the base models to reduce both bias and variance, thereby improving the generalization ability on unseen data.

Overview of Bagging & Boosting model The two most widely used ensemble approaches are Bagging and Boosting, which differ significantly in the way they train and aggregate the base models.

Bagging (Bootstrap Aggregating) trains multiple base learners in parallel on different bootstrap samples random subsets of the original dataset drawn with replacement. The primary objective of Bagging is to reduce model variance, making it particularly effective for high-variance models such as deep decision trees (see in Figure 3.20). In classification tasks, the final prediction is typically made via majority voting, whereas for regression tasks, averaging is used.

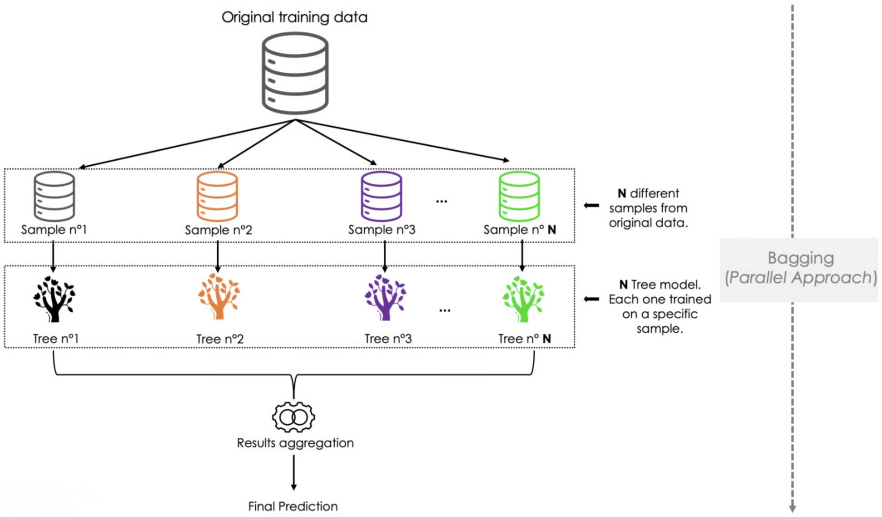


Figure 3.20: Illustration of Bagging model (DataCamp, 2025).

In contrast, Boosting constructs base learners sequentially, with each new model attempting to correct the errors made by its predecessors (see in Figure 3.21). This is achieved by assigning higher weights to misclassified instances in the training data. Boosting primarily aims to reduce bias and is especially effective when applied to weak base learners. Popular Boosting algorithms include AdaBoost, Gradient Boosting Machines (GBM), and advanced variants such as XGBoost, LightGBM, and CatBoost.

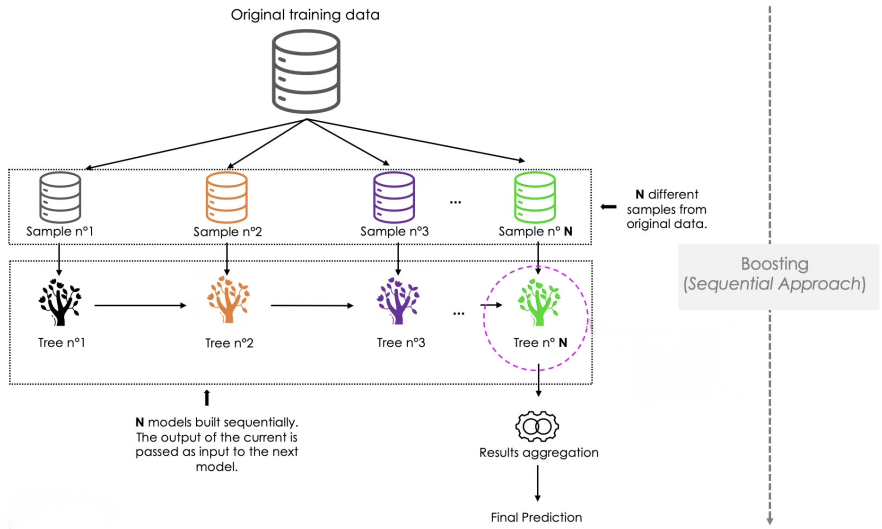


Figure 3.21: Illustration of the Boosting model (DataCamp, 2025).

In the context of this study, four representative ensemble algorithms were selected for implementation and evaluation: Random Forest as a Bagging-based model, and Gradient Boosting, CatBoost, and LightGBM as Boosting-based models.

Random Forest Random Forest is a powerful ensemble learning algorithm based on the Bagging (Bootstrap Aggregating) paradigm, introduced by Breiman (2001). It enhances both the accuracy and generalization capability of individual decision trees by aggregating the outputs of multiple models trained independently on different bootstrap samples. As a result, it effectively reduces model variance while maintaining low bias, making it highly suitable for complex classification tasks.

The training process of Random Forest involves three key stages: (1) generating multiple bootstrap datasets from the original training data using sampling with replacement; (2) independently training decision trees on each bootstrap sample, where at each node only a random subset of features is considered for splitting, which increases tree diversity; and (3) aggregating predictions from all decision trees through **majority voting** (for classification) or **averaging** (for regression).

The Figure 3.22 below presents the architecture of a typical Random Forest: the original dataset is divided into several bootstrapped samples, each used to train a separate decision tree. These trees then generate individual predictions, which are aggregated using Majority Voting to form the final output. This parallel training structure characterizes Bagging and capitalizes on model diversity to boost robustness.

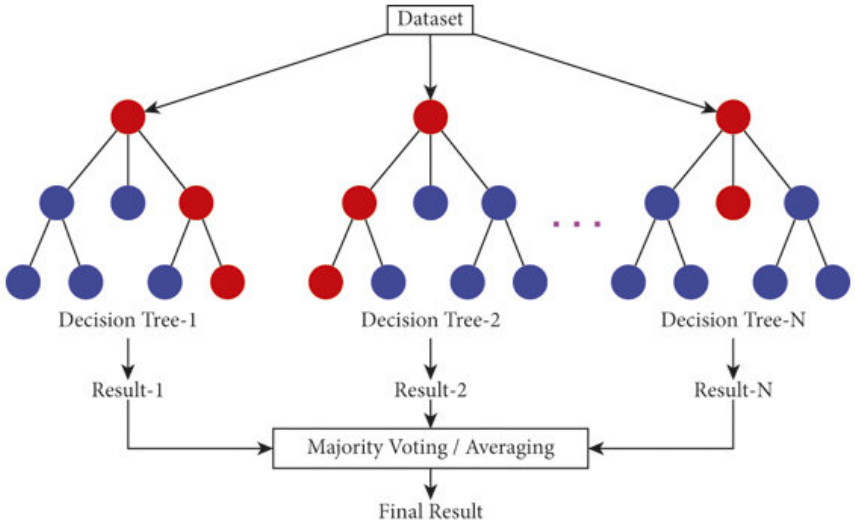


Figure 3.22: Illustration of Random Forest trees (Khan et al., 2021).

Majority Voting in Random Forest

A core aspect of the Random Forest algorithm is the use of Majority Voting to synthesize

predictions from the ensemble. After training N decision trees on distinct bootstrap samples, each tree T_i independently predicts the class of an unseen instance $\mathbf{x}$. The final predicted class $\hat{y}$ is then determined as the one that receives the most votes:

$$\hat{y} = \arg\max_{c \in C} \sum_{i=1}^{N} \mathbb{I}(T_i(\mathbf{x}) = c) \tag{3.10}$$

- C denotes the set of possible class labels;
- $T_i(\mathbf{x})$ is the prediction from the i-th tree;
- $\mathbb{I}(\cdot)$ is the indicator function.

In the context of this study, where the goal is to predict customer subscription to a term deposit, majority voting helps to enhance classification robustness. Since each tree is trained on slightly different data and feature subsets, their predictions capture diverse decision patterns. Aggregating these predictions mitigates the risk of overfitting or being influenced by noisy or atypical training points.

Furthermore, majority voting aligns well with the Stratified K-Fold Cross-Validation framework applied in this research. For each model and fold, predictions on the test set are collected from all sub-models trained on different training splits. Final predictions are then determined via majority voting across the sub-models, mimicking the core idea of Random Forest at a meta-model level. This approach ensures more reliable out-of-sample performance and offers a stable benchmark to compare with boosting-based ensemble methods.

Within the context of this thesis, forecasting customer responses to term deposit offers Random Forest serves as a foundational model due to its flexibility, robustness, and strong performance on structured datasets. It is particularly well-suited for problems involving non-linear relationships and interactions among variables, which are common in behavioral prediction tasks like financial decision-making.

Moreover, Random Forest provides interpretable outputs such as feature importance scores, which help identify key factors influencing customer decisions. This insight is critical for marketing teams seeking to optimize targeting strategies. Additionally, its resistance to overfitting, even when faced with noisy or imbalanced data, makes it a reliable choice in operational settings.

Given its ensemble-based architecture, ability to handle mixed data types, and robustness against overfitting, Random Forest stands as a strong candidate for baseline comparison in this study. It not only delivers competitive predictive performance but also offers interpretability and stability qualities that are highly valued in practical banking applications.

CatBoost CatBoost is a state-of-the-art gradient boosting algorithm developed by Yandex, designed specifically to handle categorical features efficiently and robustly. Unlike traditional boosting algorithms that require extensive preprocessing of categorical data (e.g., one-hot or label encoding), CatBoost natively supports categorical variables by employing an innovative technique known as ordered boosting and a sophisticated method for encoding categories via target statistics, while avoiding target leakage and overfitting.

CatBoost follows the fundamental framework of boosting, where decision trees (base learners) are built sequentially. Each successive tree is trained to correct the prediction errors made by the previous ensemble of trees. However, CatBoost introduces several novel enhancements to standard gradient boosting frameworks:

- **Ordered boosting**: To avoid target leakage in small datasets or datasets with high-cardinality categorical features, CatBoost uses a permutation-driven training process that ensures that the prediction for each sample is based only on prior observations. This preserves the causality and reliability of the model.
- **Efficient categorical encoding**: CatBoost transforms categorical variables into numerical representations using target statistics (e.g., mean target encoding) while maintaining robustness against overfitting by applying Bayesian priors and permutation-based techniques.
- **Symmetric tree structure**: Unlike traditional gradient boosting algorithms such as XGBoost or LightGBM that build asymmetric trees, CatBoost constructs symmetric trees, where all leaves at a given depth are split by the same feature. This design leads to faster inference and better model regularization.

In the context of thesis, where the goal is to predict customer subscriptions to term deposits based on banking telemarketing data, CatBoost is particularly advantageous due to the dataset's high proportion of categorical features (e.g., `job`, `marital`, `education`, `contact`). The model's ability to natively process these variables without extensive manual encoding streamlines the preprocessing pipeline and reduces potential information

loss.

Moreover, CatBoost has been shown in prior empirical studies to outperform other gradient boosting models in tasks involving tabular data with mixed types. This makes it an ideal candidate for the predictive modeling task at hand, as it combines both accuracy and interpretability.

Majority Voting in CatBoost

Though CatBoost is fundamentally a boosting algorithm which typically uses weighted sums of tree outputs rather than explicit majority voting its decision-making process can be interpreted through a leaf-wise aggregation mechanism. Each tree contributes a real-valued output (log-odds or raw score), and the cumulative output is obtained via summation across all trees. This aggregated value is then passed through a sigmoid function to generate the final probability prediction.

In ensemble evaluation settings such as Stratified K-Fold Cross-Validation used in this study CatBoost models trained on different folds can be combined through meta-level majority voting. That is, when multiple CatBoost sub-models are trained on different folds, their final class predictions can be combined using plurality voting to determine the most frequent predicted class across models. This serves to mitigate overfitting and enhance the robustness of out-of-sample predictions.

The schematic illustration in Figure 3.23 below demonstrates the sequential learning structure of CatBoost. Each tree is trained to minimize the residuals of the ensemble formed by previous trees, and the process continues iteratively until convergence.

The figure also illustrates how the outputs from each tree, denoted as $(o_1, o_2, \ldots, o_n)$, are combined to produce the final model output.

Key advantages of CatBoost that justify its use in this study include:

- Superior handling of categorical features without extensive preprocessing.
- Robust performance on small datasets due to ordered boosting and regularization.
- Fast inference enabled by symmetric trees.
- Reduced overfitting via advanced encoding and permutation strategies.
- Interpretability through SHAP value support, facilitating explainable AI integration.

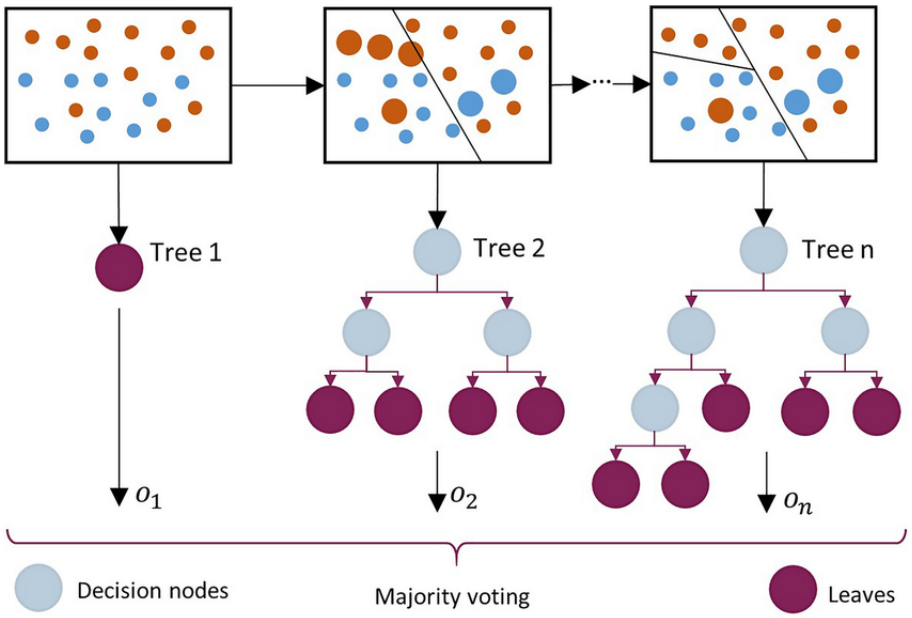


Figure 3.23: Schematic of the CatBoost tree construction (Yousefzadeh et al., 2024).

Given these strengths, CatBoost serves as a powerful and reliable model in the comparative analysis of ensemble learning approaches for predicting customer behavior in term deposit campaigns.

LightGBM Light Gradient Boosting Machine is a high-performance machine learning framework developed by Microsoft, based on the principles of Gradient Boosting Decision Trees (GBDT). This algorithm stands out for its ability to efficiently process large-scale datasets while maintaining high accuracy and rapid training speeds. LightGBM implements two key innovative techniques: Gradient-based One-Side Sampling (GOSS) for intelligent sampling and Exclusive Feature Bundling (EFB) for handling sparse data. A distinctive characteristic of this algorithm is its leaf-wise tree growth approach, which focuses on optimizing the leaf node with the highest information gain instead of developing trees level by level as in traditional methods.

The LightGBM model can be represented by the following fundamental mathematical formulation:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i) \tag{3.11}$$

- $\hat{y}_i$ is the predicted value for the i-th sample;
- K is the total number of decision trees;

- f_k is the prediction function of the k-th tree;
- $\mathbf{x}_i$ is the feature vector of the i-th sample.

The training process optimizes the following objective function:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.12}$$

- $l(y_i, \hat{y}_i)$ is the loss function measuring the difference between the actual value y_i and the predicted value $\hat{y}_i$;
- $\Omega(f_k)$ is the regularization term to control model complexity;
- n is the number of samples.

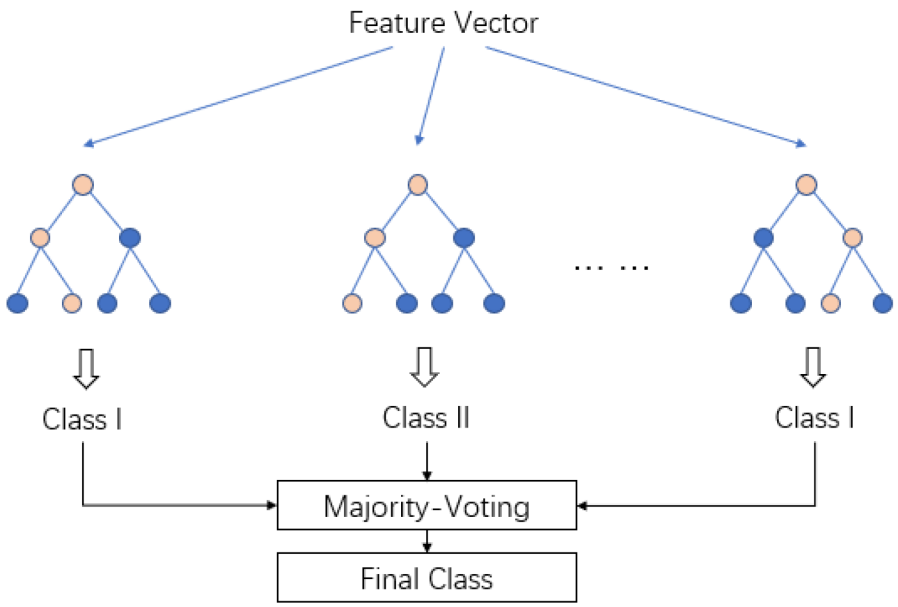


Figure 3.24: Architecture of majority voting in LightGBM (Kılıç, 2023).

Majority Voting in LightGBM

A crucial aspect of LightGBM's architecture is the majority voting mechanism, as illustrated in Figure 3.24. After the input feature vector is processed through an ensemble of decision trees, each tree produces a prediction about the class (e.g., Class I, Class II). Instead of simply aggregating prediction values as in traditional GBM, LightGBM employs a majority voting method to determine the final class.

The mathematical formulation for this mechanism is:

$$C(\mathbf{x}) = \text{mode}\{c_1(\mathbf{x}), c_2(\mathbf{x}), \ldots, c_K(\mathbf{x})\} \tag{3.13}$$

- $C(\mathbf{x})$ is the final predicted class for input sample $\mathbf{x}$;
- $c_k(\mathbf{x})$ is the class predicted by the k-th tree for sample $\mathbf{x}$;
- mode is the function returning the most frequently occurring value.

The probability of prediction for each class is calculated as:

$$P(y = j \mid \mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}(c_k(\mathbf{x}) = j) \tag{3.14}$$

- $P(y = j \mid \mathbf{x})$ is the probability that the sample $\mathbf{x}$ belongs to class j;
- $\mathbb{I}(c_k(\mathbf{x}) = j)$ is the indicator function, which returns 1 if the k-th decision tree predicts that the sample $\mathbf{x}$ belongs to class j, and 0 otherwise.

In the majority voting system of LightGBM, each decision tree is considered an independent "expert", and the prediction of each tree counts as a vote for the corresponding class. The class that receives the most votes (appears most frequently in the predictions of the trees) is selected as the final classification result. This mechanism helps LightGBM minimize the influence of "outlier" trees (those with deviant predictions), while enhancing the stability and generalization capability of the model across diverse datasets.

Majority voting in LightGBM is particularly effective when dealing with complex classification problems with non-linear decision boundaries. This method also contributes significantly to the algorithm's ability to handle data imbalance, as minority classes still have the opportunity to be detected through this voting mechanism.

Gradient Boosting Gradient Boosting is a powerful ensemble technique that constructs predictive models in a sequential manner by combining multiple weak learners typically decision trees to form a strong composite learner. Unlike parallel ensemble approaches such as Bagging, Gradient Boosting builds models iteratively, with each new tree trained to correct the residual errors of the ensemble generated so far. This additive model aims

to minimize a specified loss function by applying gradient descent in function space, hence the name "Gradient Boosting".

Formally, the prediction function $F(\mathbf{x})$ at iteration m is updated as follows:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \gamma_m h_m(\mathbf{x}) \tag{3.15}$$

- $F_{m-1}(\mathbf{x})$ is the current model's prediction;
- $h_m(\mathbf{x})$ is the weak learner fitted to the negative gradients (residuals);
- γ_m is a learning rate that controls the contribution of h_m.

This procedure continues for a predefined number of iterations or until convergence. The final prediction is an ensemble of all weak learners, each addressing errors of its predecessors, resulting in a model that balances bias and variance effectively.

In this study, Gradient Boosting is employed to predict customer subscription to term deposit products based on the UCI Bank Marketing dataset. Its ability to model complex, non-linear relationships and focus learning on the most informative samples makes it particularly well-suited for the given classification task especially in the presence of class imbalance and mixed-type variables.

Figure 3.25 below illustrates the structure of the Gradient Boosting process. Initially, the first decision tree is trained on the full dataset, making both correct and incorrect predictions. The errors from this tree are used to adjust the training data distribution such that the next tree focuses more on previously misclassified instances. This iterative process continues until the ensemble reaches a desired number of trees or convergence. The final prediction is made by aggregating the outputs from all the trees, often using a weighted sum with weights $w_1, w_2, \ldots, w_n$, determined during training.

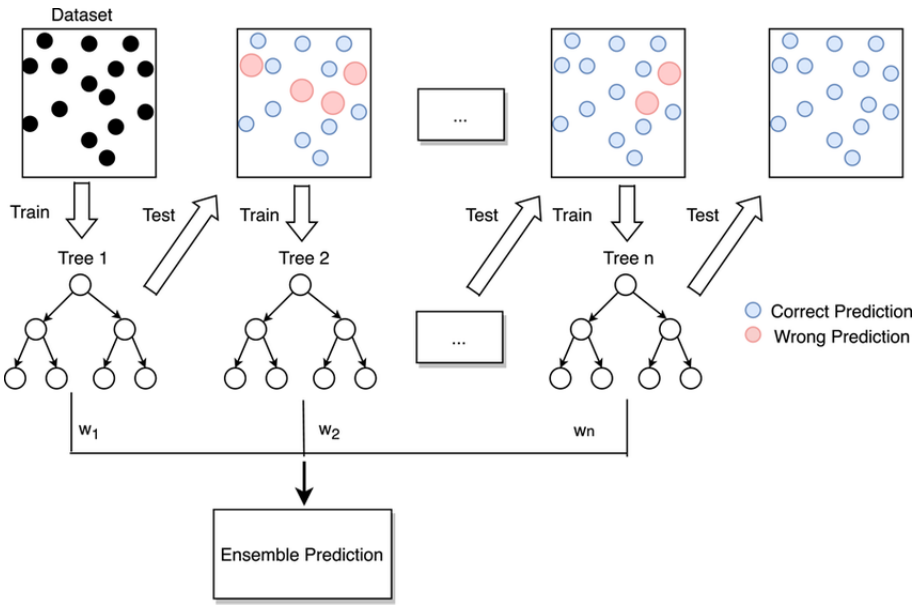


Figure 3.25: Flow diagram of Gradient Boosting method (Li et al., 2024).

In addition, while Gradient Boosting inherently does not employ majority voting (as used in Bagging methods like Random Forest), in this research, model robustness is further enhanced by applying meta-level majority voting across multiple Gradient Boosting sub-models trained using Stratified K-Fold Cross-Validation. This ensemble of ensembles approach helps mitigate overfitting and provides a stable prediction framework.

3.2.2 Training Techniques

To ensure the generalization performance and reliability of the machine learning models deployed in this study, a series of training techniques were adopted, including data splitting, cross-validation, and hyperparameter optimization. These techniques play a critical role in mitigating overfitting and enhancing model performance on unseen data.

Data Splitting The dataset, comprising `41,188` observations, was partitioned into two mutually exclusive subsets following an `80:20` ratio. Specifically, `80%` of the data (`32,950` records) was allocated for training purposes, while the remaining `20%` (`8,238` records) was retained as an independent test set for final model evaluation.

The splitting procedure was executed using the `train_test_split()` function from the `Scikit-learn` library, with a fixed `random_state=42` parameter to ensure reproducibility and consistency across different experimental runs.

Importantly, the data splitting was performed prior to any preprocessing steps to prevent information leakage, thereby ensuring that the test set remains entirely unseen and unaffected by transformation procedures applied to the training data.

Cross-Validation During model training, the training set was further partitioned using Stratified K-Fold Cross-Validation with `k=5`. Unlike conventional K-Fold Cross-Validation, the stratified variant preserves the original class distribution of the target variable within each fold. This is particularly important in the context of this study, where the dataset is subject to significant class imbalance.

More specifically, the training data (after initial train-test split) was divided into five equally sized subsets. In each iteration, one fold served as the validation set, while the remaining four folds were used to train the model. This process was repeated five times, with each fold serving as the validation set exactly once. The final performance metric was computed as the average of the evaluation results across the five folds.

A crucial methodological detail is that all preprocessing procedures such as outlier handling, feature scaling, and categorical encoding were strictly confined to each fold. In other words, transformation functions were fitted solely on the training portion of each fold and then applied to the corresponding validation subset. This rigorous approach helps prevent data leakage across folds and ensures the integrity and validity of cross-validation results.

The logic of stratified k-Fold cross-validation employed in this study is visually illustrated in Figure 3.26 below.

Figure 3.26: Diagram of stratified k-fold cross-validation (Duan, 2023).

Hyperparameter Tuning Hyperparameter tuning is a critical process in building robust and high-performing machine learning models. Unlike model parameters (e.g., regression coefficients) that are learned directly from data during training, hyperparameters must be defined before the learning process begins. These include settings such as learning rate, maximum depth, number of estimators, and regularization terms, which significantly influence model behavior in terms of accuracy, generalization, convergence speed, and overfitting tendencies.

In this research, where multiple machine learning models are evaluated to predict customer subscription to term deposits, a binary classification problem with class imbalance, hyperparameter tuning plays a central role in achieving optimal performance under a fair and reproducible framework.

There are several widely-used strategies for hyperparameter optimization:

- `GridSearchCV`: This method performs an exhaustive search over a predefined set of hyperparameter combinations. As shown in Figure 3.27 (left panel), it evaluates all combinations within the defined grid, ensuring that the best value in the grid is identified. However, this method becomes computationally expensive and inefficient when the number of parameters or their value ranges increases (curse of dimensionality).
- `RandomizedSearchCV`: Instead of evaluating every point on the grid, Randomized Search randomly samples a fixed number of parameter combinations from specified distributions. As illustrated in Figure 3.27 (right panel), it offers a more efficient and scalable approach, particularly when only a few hyperparameters strongly affect the model's performance. By selecting fewer yet diverse combinations, it often finds nearly optimal results with significantly lower computational cost.

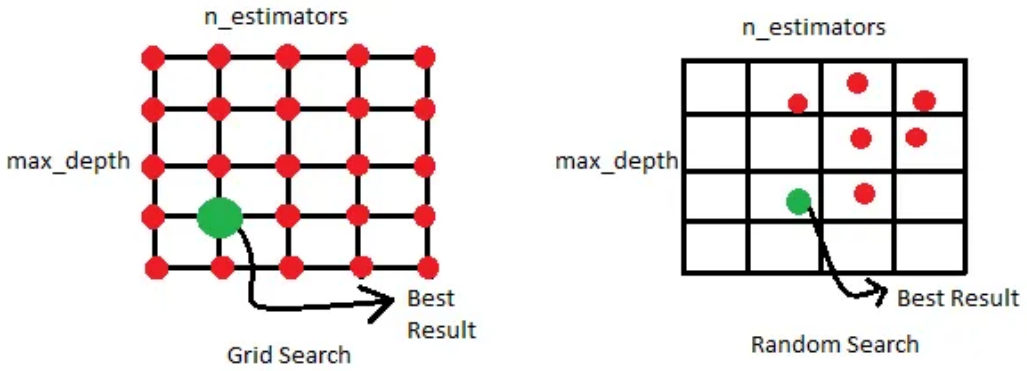


Figure 3.27: GridSearch & RandomizedSearch (Chaurasiya, 2022).

Considering the trade-off between computational efficiency and performance, this study adopts `RandomizedSearchCV` as the standard tuning technique across all models. The reasons are:

- **Scalability**: It handles a large hyperparameter space efficiently without incurring the full cost of `GridSearch`.
- **Exploration**: It explores more diverse combinations than grid-based search, which is particularly useful when tuning complex ensemble methods like CatBoost and Gradient Boosting.
- **Integration**: It integrates smoothly with stratified k-fold cross-validation, preserving consistency and fairness during model evaluation.
- **Stability**: It helps reduce overfitting by avoiding redundant parameter evaluations and focusing on generalizable configurations.

While more sophisticated approaches such as `Bayesian Optimization` or `Optuna` offer promising directions for future research, this thesis maintains consistency and reproducibility by applying `RandomizedSearchCV` as the exclusive hyperparameter tuning method for all experiments.

3.2.3 Evaluation Metrics

Given the class imbalance in the dataset where the proportion of customers subscribing to term deposits is relatively low this study avoids relying solely on overall accuracy, which

can be misleading. Instead, a set of more robust and informative evaluation metrics is employed, including Balanced Accuracy, Sensitivity (Recall), Specificity, ROC-AUC, and the Confusion Matrix. These metrics collectively provide a comprehensive view of model performance, especially in imbalanced classification settings, and serve as the basis for comparing model effectiveness and supporting reliable business decisions in telemarketing campaigns.

3.2.3.1 Confusion matrix

The confusion matrix is a 2 x 2 table used to evaluate the performance of binary classification models. In our case, it assesses whether the model correctly predicts if a customer will subscribe to a term deposit. Each row shows the actual status, and each column shows the predicted status.

In this specific context, the confusion matrix measures instances where a customer's subscription decision is misclassified. Each column in the matrix represents a predicted subscription status, while each row corresponds to the actual subscription status of the customer.

The confusion matrix for our bank deposit subscription classification includes the following components:

- True Positive (TP): Correctly predicted subscribers
- False Positive (FP): Non-subscribers wrongly predicted as subscribers
- True Negative (TN): Correctly predicted non-subscribers
- False Negative (FN): Subscribers wrongly predicted as non-subscribers

		Actual	
		Subcriber	Not-Subcriber
Predicted	Subcriber	True Positive (TP)	False Negative (FN)
	Not-Subcriber	False Positive (FP)	True Negative (TN)

Table 3.2: Confusion matrix for binary classification.

3.2.3.2 *Sensitivity*

Sensitivity, also known as `Recall` or `True Positive Rate (TPR)`, measures the model's ability to correctly identify positive instances. It is particularly important in scenarios where missing a positive case carries high consequences, such as disease diagnosis.

Sensitivity is calculated as:

$$Sensitivity = Recall = True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{3.16}$$

- **TP**: the number of **True Positives**
- **FN**: the number of **False Negatives**

This metric ranges from 0 to 1, with higher values indicating better performance in correctly identifying positive cases. A sensitivity of 0.1 indicates that the model correctly identifies all positive instances in the dataset.

3.2.3.3 *Specificity*

Specificity, or **True Negative Rate (TNR)**, quantifies the model's ability to correctly identify negative instances. This metric is crucial in contexts where false alarms are costly or may lead to unnecessary interventions.

Specificity is calculated as:

$$Specificity = True\ Negative\ Rate = \frac{TN}{TN + FP} \tag{3.17}$$

- **TN**: the number of **True Negatives**
- **FP**: the number of **False Positives**

Like sensitivity, specificity ranges from 0 to 1, with higher values indicating better performance in correctly identifying negative cases. A specificity of 1.0 indicates that the model correctly identifies all negative instances in the dataset.

3.2.3.4 *Balanced Accuracy*

Balanced accuracy provides a more representative measure of model performance when dealing with imbalanced datasets. Unlike standard accuracy, which can be misleading when class distributions are skewed, balanced accuracy calculates the average of sensitivity and specificity.

Balanced accuracy is calculated as:

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2} \tag{3.18}$$

This metric effectively addresses the bias that can occur in traditional accuracy measurements when one class significantly outnumbers the other. A balanced accuracy of 0.5 indicates performance equivalent to random guessing, while a score of 1.0 represents perfect classification.

3.2.3.5 *Receiver Operating Characteristic AUC (ROC–AUC)*

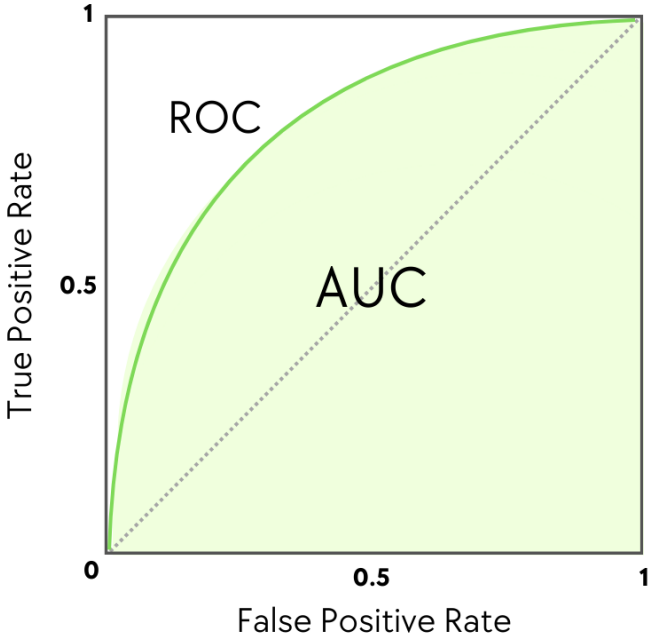


Figure 3.28: Illustrated of ROC-AUC curve (Chatterjee, 2025).

The Receiver Operating Characteristic Area Under Curve (ROC-AUC) is a performance measurement that evaluates a model across various threshold settings. The ROC curve

plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at different classification thresholds.

The AUC represents the probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative instance. This makes it particularly valuable for evaluating model performance independent of any specific classification threshold.

The ROC-AUC score ranges from 0 to 1:

- A score of 0.5 indicates performance equivalent to random guessing.
- Scores below 0.5 suggest worse-than-random performance.
- Scores above 0.5 indicate increasingly better discrimination, with 1.0 representing perfect classification.

3.2.4 Explainability Methods

In this research, we employ explainability methods to better understand the underlying factors that influence customer decisions to subscribe to bank deposits. These methods help reveal the "black box" nature of complex machine learning models, providing insights into which features significantly impact prediction outcomes and how they interact with each other.

3.2.4.1 Feature Importance

Feature importance analysis quantifies how much each input contributes to the model's predictive performance. This helps identify which customer attributes and interaction variables are most influential in determining subscription outcomes. In this study, normalized importance values are used to rank features in four ensemble models (Random Forest, Gradient Boosting, LightGBM, and CatBoost). Across these models, variables related to the economic context (e.g., `euribor3m`, `nr.employed`) and campaign interactions (e.g., `duration`, `poutcome_success`) repeatedly emerge as highly influential.

3.2.4.2 SHapley Additive exPlanations (SHAP)

To complement the global view from feature importance, SHAP values are used to explain model predictions in greater detail. While traditional feature importance methods offer a general overview of which input variables are most influential, they often fall short in explaining the underlying mechanisms behind specific predictions. To overcome this limitation, the study incorporates SHAP (SHapley Additive exPlanations), which enables both global and individual-level interpretability. SHAP is grounded in cooperative game theory and computes the marginal contribution of each feature to the model's output, thereby reflecting the true influence of individual inputs on the prediction.

One of the key distinctions between SHAP and conventional feature importance is that SHAP not only quantifies how important a feature is, but also clarifies the direction of its impact that is, whether it increases or decreases the likelihood of subscription. Furthermore, SHAP allows for local explanations, making it possible to interpret model decisions at the level of individual customers. This capability is especially valuable in the context of banking marketing, where decisions must be personalized. In addition, SHAP can uncover complex non-linear interactions between features, which are often overlooked by standard importance metrics, especially in ensemble or boosting models.

This study presents two SHAP-based visualizations:

- **SHAP Summary Plot**: Shows overall feature impact across the dataset, including both direction (positive or negative influence) and magnitude.
- **SHAP Waterfall Plot**: Provides local explanations for individual cases by visualizing how each feature contributed to a specific prediction.

Although more advanced SHAP visualizations such as dependence or interaction plots are not included, the summary and waterfall plots already offer valuable interpretation. They help identify global drivers of model behavior and provide case-level reasoning, which can support targeted marketing strategies and customer engagement decisions.

3.2.4.3 Local Interpretable Model-agnostic Explanations (LIME)

Objective & Working Principle Local Interpretable Model-agnostic Explanations (LIME) is a post-hoc interpretability technique designed to explain the predictions of

any machine learning model by approximating its behavior locally. While many models operate as black boxes, LIME seeks to understand how individual predictions are made by learning a simple surrogate model such as linear regression or decision tree—around the instance being explained.

To achieve this, LIME perturbs the input data around a specific instance, obtains predictions from the original model, and then fits a locally weighted interpretable model to mimic the complex model's behavior in that local region. This allows practitioners to observe which features were most influential in determining a particular prediction.

Advantages & Applications

- **Model-agnostic**: LIME works with any machine learning model, including ensemble and neural networks.
- **Local explanations**: It offers instance-level interpretability, clarifying decisions for individual cases rather than general global trends.
- **Broad applicability**: LIME can be applied to tabular, text, and image data.
- **Decision support**: Particularly useful in domains such as banking, healthcare, and legal systems, where transparent and accountable AI decisions are essential.

Limitations

- **Instability**: The results may vary depending on how the perturbed samples are generated or weighted.
- **Linearity assumption**: LIME assumes that a simple linear model can approximate complex model behavior in the local vicinity, which may not always hold in highly non-linear regions.
- **Computational cost**: Generating perturbed samples and training surrogate models repeatedly can be computationally expensive for large datasets.

CHAPTER 4.

RESULTS & FINDINGS

4.1 Model Performance

4.1.1 Evaluation before Hyperparameter tuning

Table 4.1 below presents the balanced accuracy scores of six classification models across the training, validation, and test sets prior to hyperparameter tuning. The results highlight notable differences in model performance and generalization capabilities.

Model	train_score_mean	val_score_mean	test_score_voting
LightGBM	**0.931197**	**0.891731**	**0.890573**
CatBoost	**0.951400**	**0.874679**	**0.881021**
Gradient Boosting	0.896847	0.887256	0.879868
Logistic Regression	0.881649	0.879993	0.866645
Random Forest	0.999962	0.753932	0.759260
Naive Bayes	0.753903	0.751100	0.757261

Table 4.1: Balanced accuracy scores on train, val & test before hyperparameter tuning.

Among the evaluated models, **LightGBM** achieved the highest balanced accuracy on the test set (**0.8906**), with relatively consistent scores on the training (**0.9312**) and validation sets (**0.8917**). This suggests that LightGBM not only learned effectively from the training data but also maintained its predictive ability on unseen samples. Similarly, **CatBoost** and **Gradient Boosting** also demonstrated strong and stable performance, with test balanced accuracy scores of **0.8810** and **0.8799**, respectively. These models showed minimal variance between training and test performance, indicating good generalization and robustness.

Logistic Regression, despite being a linear model, delivered competitive performance (test score: **0.8666**) and served as a reliable baseline. Its performance across the three

sets remained relatively stable, further supporting its reliability.

By contrast, **Random Forest** exhibited signs of severe overfitting. While the model almost perfectly fit the training data (train score: **0.9999**), its performance on the validation and test sets was considerably lower (**0.7539** and **0.7592**, respectively). This discrepancy suggests that the model memorized patterns in the training data rather than capturing generalizable relationships.

Naive Bayes and **Random Forest** obtained the lowest balanced accuracy scores on the test set, indicating limited suitability for the classification problem under study.

Model	Balanced_Acc	Sensitivity	Specificity	ROC-AUC
LightGBM	**0.890573**	**0.912461**	**0.868685**	**0.946819**
CatBoost	**0.881021**	**0.872297**	**0.889745**	**0.947776**
Gradient Boosting	0.879868	0.915551	0.844184	0.942787
Logistic Regression	0.866645	0.893924	0.839367	0.931052
Random Forest	0.759260	0.568486	0.950034	0.938451
Naive Bayes	0.757261	0.771370	0.743152	0.821360

Table 4.2: Comparative performance metrics of models before tuning.

While **LightGBM** achieved the highest balanced accuracy score, it also exhibited a noticeable gap between sensitivity (**0.9125**) and specificity (**0.8687**). A similar trend is observed in **Gradient Boosting**, where sensitivity outperforms specificity by a significant margin. This discrepancy suggests that both models tend to favor the positive class (i.e., predicting term deposit subscription), potentially at the cost of increased false positives.

By contrast, **CatBoost** and **Logistic Regression** demonstrated a more balanced trade-off between sensitivity and specificity, indicating that these models make more calibrated predictions across both classes. As illustrated in Table 4.2, these two models maintained relatively stable and symmetric performance. This balance is especially important in real-world banking applications where misclassification in either direction (false positive or false negative) could have strategic implications.

4.1.2 Evaluation after Hyperparameter tuning

Model	Balanced_Acc	Sensitivity	Specificity	ROC-AUC
CatBoost	**0.890112**	**0.923790**	**0.856435**	**0.946877**
LightGBM	**0.887049**	**0.924820**	**0.849277**	**0.947259**
Gradient Boosting	0.886712	0.918641	0.854783	0.944371
Random Forest	0.880039	0.916581	0.843496	0.937428
Logistic Regression	0.868397	0.897013	0.839780	0.931278
Naive Bayes	0.759275	0.784758	0.733792	0.821091

Table 4.3: Comparative performance metrics of models after tuning.

Summary performance After applying hyperparameter tuning, all ensemble models showed improved performance, particularly in terms of balanced accuracy and ROC-AUC on the test set. As shown in Table 4.3, **CatBoost** outperformed all other models with a balanced accuracy of **0.8901** and ROC-AUC of **0.9469**, followed closely by **LightGBM** (**0.8870**) and **Gradient Boosting** (balanced accuracy: **0.8867**). These top three models also demonstrated high sensitivity and specificity values, indicating strong capability in identifying both classes with minimal trade-off.

Notably, **Random Forest**, which initially showed signs of overfitting, achieved a substantial improvement after tuning. Its balanced accuracy reached **0.88** with strong sensitivity and specificity, indicating that hyperparameter optimization significantly enhanced its generalization capability.

In contrast, **Naive Bayes** and **Random Forest** remained the weakest performers, even after tuning. Their relatively low balanced accuracy and ROC-AUC scores indicate limited ability to distinguish between the two classes effectively.

Naive Bayes The best hyperparameters for the model were:

```
{
  'model__priors': [0.4, 0.6],
  'model__var_smoothing': 1.1462107403425044e-09
}
```

Naive Bayes was among the least effective models in this study. After hyperparameter tuning, the evaluation metrics showed only marginal improvements (see in Tab. 4.4), with balanced accuracy increasing slightly from **0.7573** to **0.7593** and sensitivity reaching **0.7848**, while specificity declined slightly (**0.7432** → **0.7338**).

	Balanced Acc	**Sensitivity**	**Specificity**	**ROC-AUC**
Before tuning	0.757261	0.771370	0.743152	0.821360
After tuning	0.759275	0.784758	0.733792	0.821091

Table 4.4: Performance comparison of Naive Bayes before & after tuning.

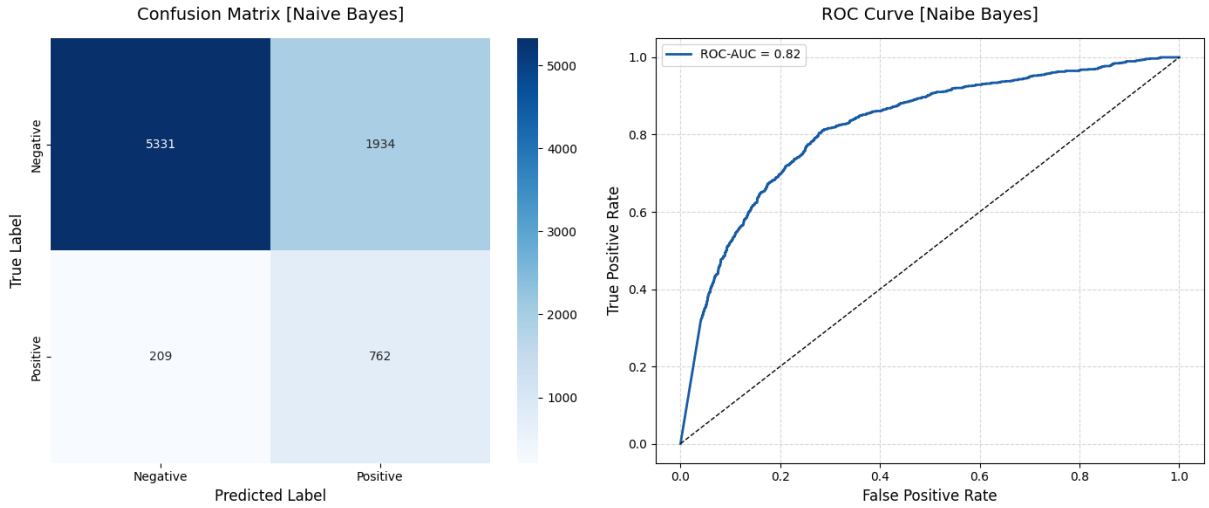


Figure 4.1: Confusion matrix & ROC curve of Naïve Bayes model.

The model produced **209** false negatives, indicating limited capability in detecting the positive class. The ROC curve, with an AUC of **0.8211**, further confirms the model's relatively limited discriminative power under class imbalance (see Fig 4.1).

Logistic Regression Logistic Regression continues to affirm its role as a reliable baseline model in this study. After hyperparameter tuning, evaluation metrics such as balanced accuracy (increased from **0.8666** to **0.8684**), sensitivity (from **0.8939** to **0.8970**), and specificity (from **0.8394** to **0.8398**) all showed slight improvements, indicating that the model maintained stable performance on the test set (see in Table 4.5). Although the increases were modest, they reflect the model's inherent generalization capability.

	Balanced Acc	Sensitivity	Specificity	ROC-AUC
Before tuning	0.866645	0.893924	0.839367	0.931052
After tuning	0.868397	0.897013	0.839780	0.931278

Table 4.5: Performance comparison of Logistic Regression before & after tuning.

The best hyperparameters found through randomized search were as follows:

```
{
  'model__C': 1.1462107403425035,
  'model__class_weight': 'balanced',
  'model__l1_ratio': 0.13949386065204183,
  'model__max_iter': 500,
  'model__penalty': 'l2',
  'model__solver': 'sag',
  'model__tol': 0.0001
}
```

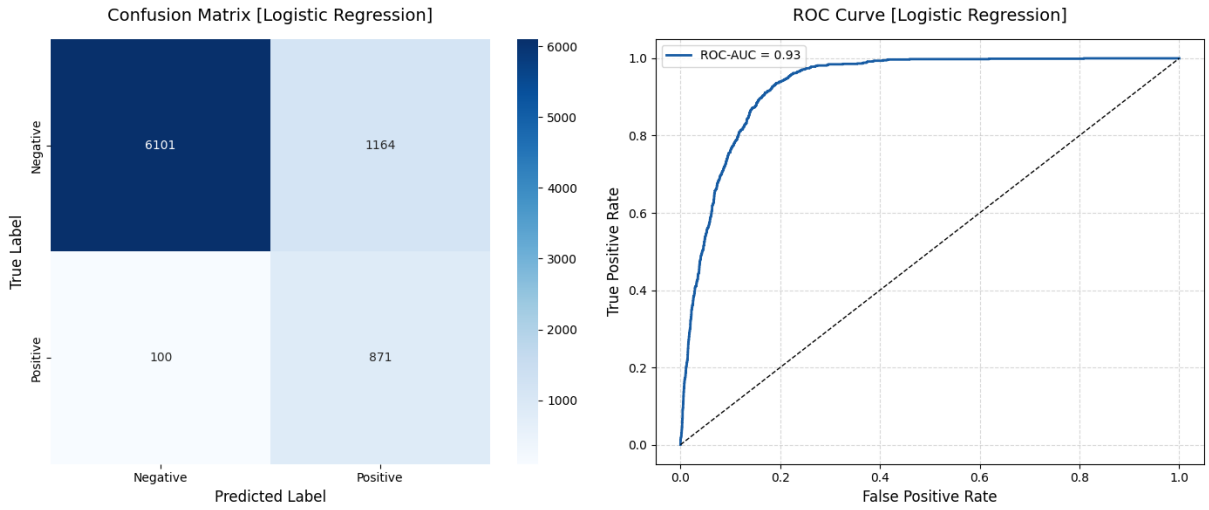


Figure 4.2: Confusion matrix & ROC curve of Logistic Regression model.

The confusion matrix (see in Fig. 4.2) shows a small number of false negatives (**100**), and the ROC curve illustrates strong overall performance with an AUC of **0.9313**, validating its discriminative ability.

Random Forest Before hyperparameter tuning, the Random Forest model suffered from severe overfitting, evidenced by its extremely high balanced accuracy on the training set (**0.9999**) compared to significantly lower performance on validation (**0.7539**) and test sets (**0.7593**)

	Balanced Acc	Sensitivity	Specificity	ROC-AUC
Before tuning	0.759260	0.568486	0.950034	0.938451
After tuning	0.880039	0.916581	0.843496	0.937428

Table 4.6: Performance comparison of Random Forest before & after tuning.

The best hyperparameters for the model were:

```
{
  'model__n_estimators': 100,
  'model__min_samples_split': 2,
  'model__min_samples_leaf': 4,
  'model__max_features': 'sqrt',
  'model__max_depth': 10,
  'model__bootstrap': False
}
```

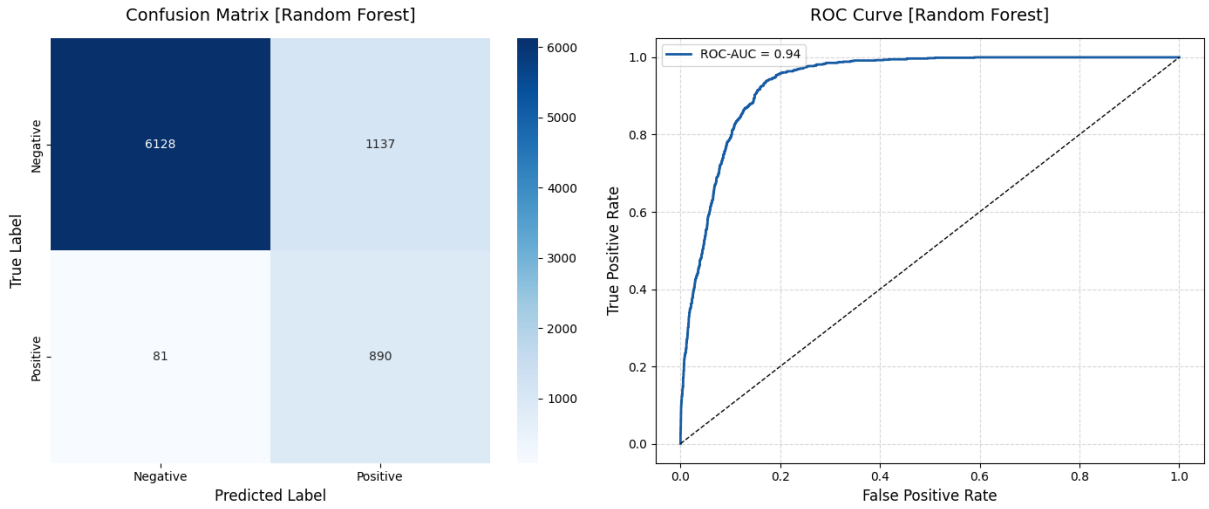


Figure 4.3: Confusion matrix & ROC curve of Random Forest model.

After tuning, the model's generalization improved remarkably: test balanced accuracy increased to **0.88**, and sensitivity rose from **0.5685** to **0.9166**, reducing the gap with

specificity (**0.8435**). Although the ROC-AUC slightly decreased to **0.9374** after tuning, it remained high, confirming strong class discrimination. The confusion matrix also shows a more balanced distribution between false positives and false negatives (see in Fig. 4.3).

Gradient Boosting Gradient Boosting continued to perform robustly after hyperparameter tuning, with balanced accuracy slightly improving from **0.8799** to **0.8867**. Both sensitivity (**0.9186**) and specificity (**0.8548**) remained high, indicating its ability to effectively detect both positive and negative classes. The ROC-AUC also increased slightly to **0.9444**, confirming the model's stable and strong discriminative power.

	Balanced Acc	**Sensitivity**	**Specificity**	**ROC-AUC**
Before tuning	0.879868	0.915551	0.844184	0.942787
After tuning	0.886712	0.918641	0.854783	0.944371

Table 4.7: Performance comparison of Gradient Boosting before & after tuning.

The optimized parameters listed below:

```
{
  'model__subsample': 1.0,
  'model__n_estimators': 100,
  'model__min_samples_split': 2,
  'model__min_samples_leaf': 3,
  'model__max_depth': 3,
  'model__learning_rate': 0.2
}
```

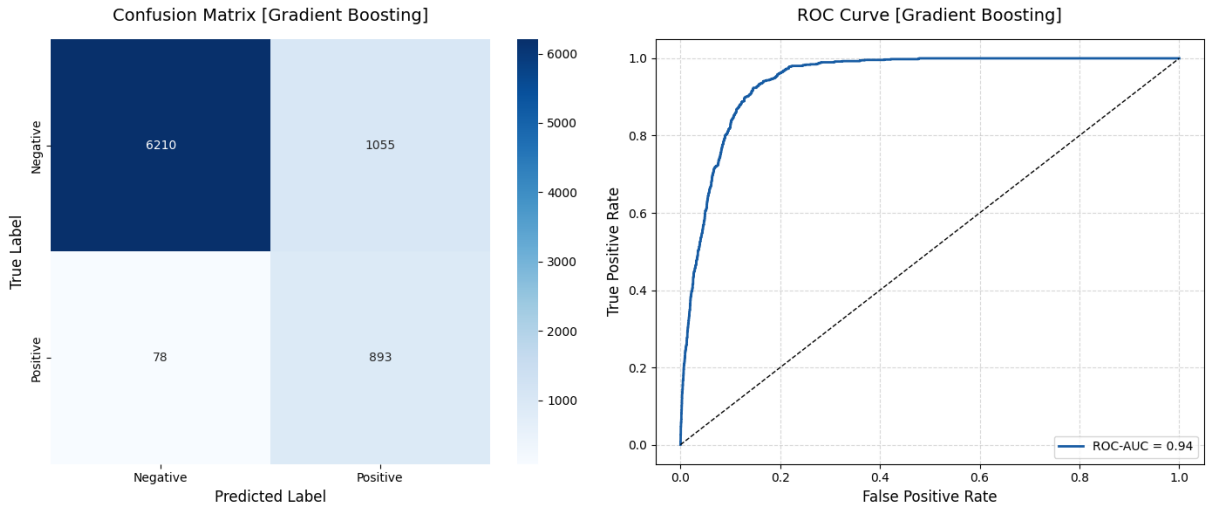


Figure 4.4: Confusion matrix & ROC curve of Gradient Boosting model.

LightGBM Among the ensemble models, LightGBM initially achieved the highest balanced accuracy (**0.8906**) before hyperparameter tuning. However, after tuning, despite achieving the highest sensitivity (**0.9248**) and a strong ROC-AUC of 0.9473, its balanced accuracy slightly declined to **0.8870**, placing it second after CatBoost. This suggests that while tuning enhanced LightGBM's ability to detect positive cases (fewer false negatives), it slightly compromised its balance across classes. The confusion matrix also reflects this shift, showing improved sensitivity but a marginal drop in specificity.

	Balanced Acc	**Sensitivity**	**Specificity**	**ROC-AUC**
Before tuning	0.890573	0.912461	0.868685	0.946819
After tuning	0.887049	0.924820	0.849277	0.947259

Table 4.8: Performance comparison of LightGBM before & after tuning.

The set of optimized hyperparameters that enhanced LightGBM's predictive performance is presented below.

```
{
  'model__subsample': 0.8,
  'model__reg_lambda': 0.5,
  'model__reg_alpha': 1.0,
  'model__num_leaves': 15,
  'model__n_estimators': 100,
```

```
    'model__min_child_samples': 10,
    'model__max_depth': -1,
    'model__learning_rate': 0.1,
    'model__colsample_bytree': 0.6
}
```

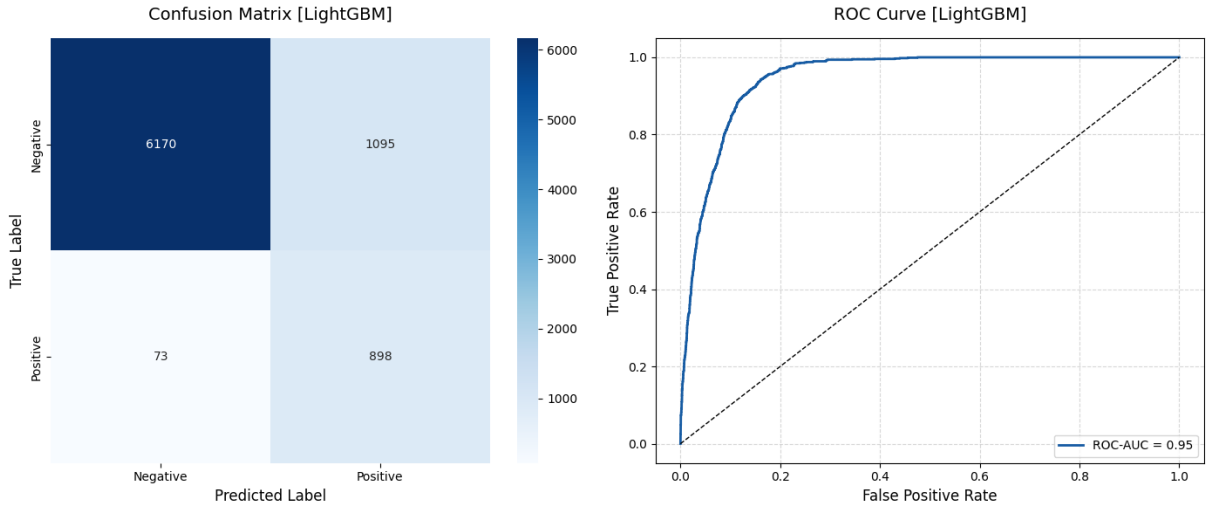


Figure 4.5: Confusion matrix & ROC curve of LightGBM model.

CatBoost Following hyperparameter tuning, CatBoost emerged as the top-performing model, slightly surpassing its prior results and other contenders in all key metrics. With a balanced accuracy of 0.8901 and a ROC-AUC of **0.9469**, it demonstrated both high classification precision and robust discriminative power. The sensitivity increased markedly to **0.9238**, showing the model's improved capacity to correctly identify positive instances. This enhancement was achieved with only a modest decrease in specificity, suggesting an effective balance between recall and false positive control.

	Balanced Acc	**Sensitivity**	**Specificity**	**ROC-AUC**
Before tuning	0.881021	0.872297	0.889745	0.947776
After tuning	0.890112	0.923790	0.856435	0.946877

Table 4.9: Performance comparison of CatBoost before & after tuning.

The tuned hyperparameters that contributed to this improvement are listed below.

```
{
```

```
    'model__learning_rate': 0.1,
    'model__l2_leaf_reg': 9,
    'model__iterations': 200,
    'model__depth': 4,
    'model__border_count': 128,
    'model__bagging_temperature': 0
}
```

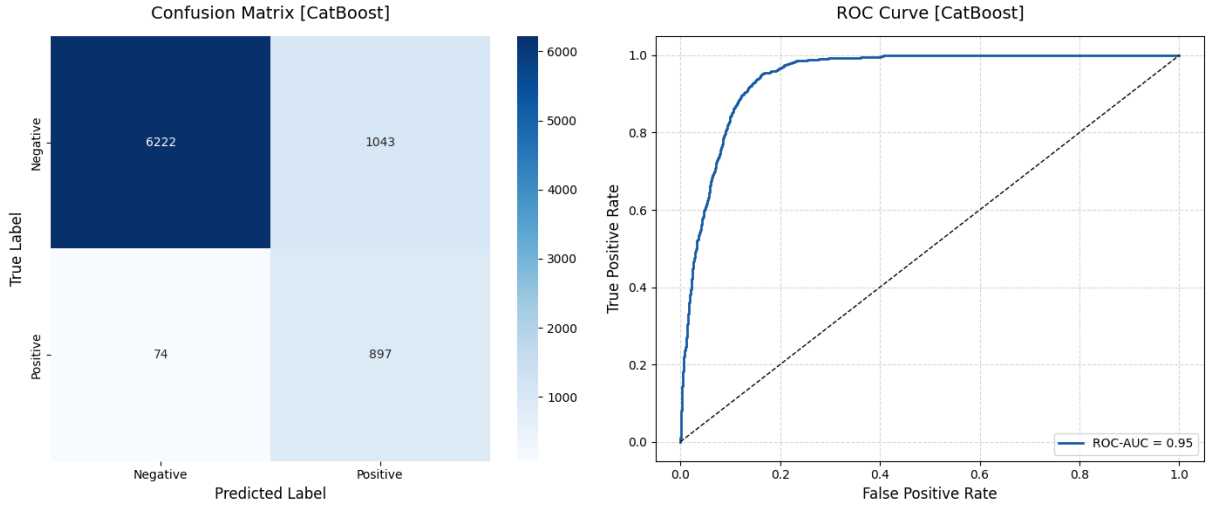


Figure 4.6: Confusion matrix & ROC curve of CatBoost model.

4.2 Model Explainability Insights

This section analyzes the factors determining the performance of optimized models. By exploring Feature Importance, applying SHAP and LIME methodology, the research clarifies decision-making mechanisms and identifies variables with the strongest influence on prediction outcomes. This analysis is conducted on the highest-performing models after the tuning process.

4.2.1 Feature Importance Analysis

The feature importance analysis reveals significant patterns across the four ensemble models (Random Forest, Gradient Boosting, LightGBM, and CatBoost) as illustrated in Figure 4.7 Several key variables consistently emerge as influential predictors across all models, with some notable variations in their relative importance.

Across all four ensemble models, the most influential features can be broadly categorized

into two groups: (1) interaction-related variables from previous campaign, and (2) social and economic context indicators.

- `duration` emerged as the most important variable in all models. This feature captures the length of the most recent call in the campaign, reflecting a direct relationship between interaction duration and the likelihood of customers subscribing to a term deposit. Its consistent ranking across models suggests that the time a customer is willing to invest in a conversation is a strong predictor of conversion. This aligns with existing marketing literature indicating that higher customer engagement often correlates with increased conversion rates.
- Features representing the social and economic context, such as `nr.employed`, `euribor3m`, `emp.var.rate`, `cons.price.idx`, consistently ranked within the top five. These results underscore the substantial influence of broader socio-economic conditions on financial decision-making. Customers are often sensitive to market signals such as short-term interest rates and employment indicators, which directly affect their financial capacity and willingness to invest. This finding also aligns with the results from the correlation analysis in EDA section, where many economic and social variables showed strong associations with the target variable `y`.

The consistent dominance of these contextual variables across all ensemble models highlights their pivotal role in shaping customer behavior. Specifically, short-term rates (e.g., `euribor3m`) and employment conditions (e.g., `nr.employed`, `emp.var.rate`) appear to significantly influence customers' readiness to commit funds to long-term financial products.

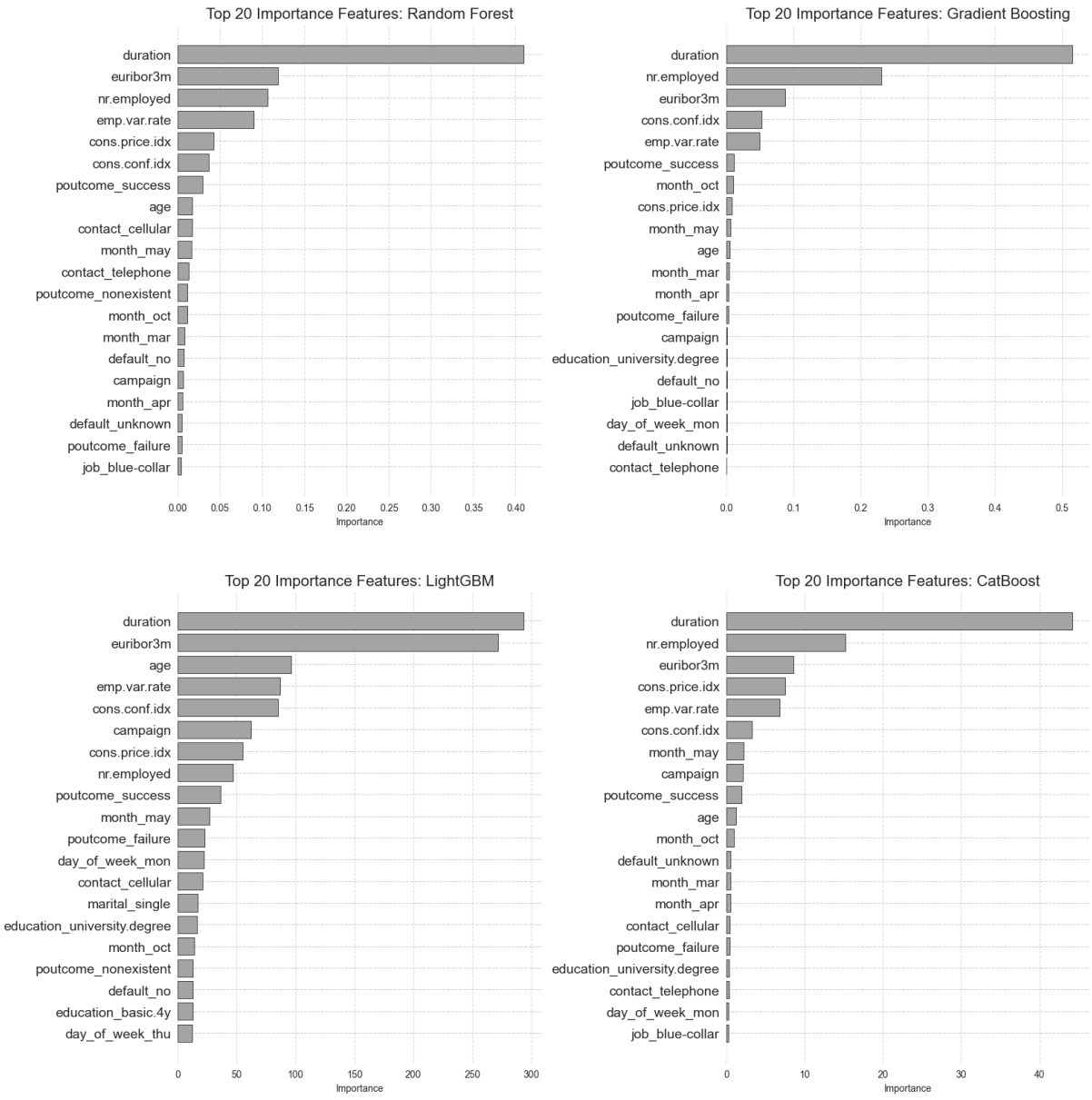


Figure 4.7: Comparison of top 20 features importance across ensemble models.

In addition to contextual features, several customer profile attributes and historical campaign performance variables also proved to be highly predictive:

- **Customer information:** The variable `age` stood out, ranking third in the LightGBM model and appearing in the top 10 across most other models. This reflects how different age groups exhibit varied financial behaviors, older customers may prefer secure investments like term deposits, while younger customers may prioritize flexibility and liquidity.
- **Credit status:** The features `default_no` and `default_unknown` appeared in the

top 20 of several models, suggesting that while not the primary drivers, customers' credit history still exerts a notable influence on deposit decisions.

- **Previous campaign outcomes:** The feature `poutcome_success` consistently ranked within the top 10 across all models, while `poutcome_failure` was also frequently observed. This indicates that past customer responses serve as strong indicators of future behavior customers who previously responded positively are more likely to do so again.

A further interesting pattern concerns the timing of contact. Features like `month_may`, `month_oct`, and `month_mar` often appeared in the top 15, which may reflect seasonal dynamics in campaign effectiveness or periodic changes in interest rate policies.

The use of diverse ensemble models has not only helped identify the most influential predictors but also revealed nuanced differences in how algorithms interpret feature–outcome relationships. These insights offer a solid foundation for refining marketing strategies and enhancing data-driven decision-making in the banking sector.

4.2.2 SHAP Analysis

To obtain deeper insights into model interpretability, this study conducts a SHAP (SHapley Additive exPlanations) analysis. Specifically, we begin by analyzing the CatBoost model, which currently demonstrates the best predictive performance. The subsequent analysis will examine whether similar feature importance patterns are consistently observed in the remaining two models Gradient Boosting and LightGBM, thereby addressing the generalizability of the findings across different ensemble techniques.

CatBoost As shown in Figures 4.8 and 4.9, the CatBoost model reveals a clear and interpretable pattern of feature contributions to the prediction of term deposit subscriptions. In particular, duration emerges as the most influential variable by a significant margin. The SHAP summary plot shows that higher values of `duration` (indicated in red) are strongly associated with higher SHAP values, thereby pushing the model output towards a positive prediction (i.e., the customer will subscribe). This confirms the critical role of the duration of call engagement in determining customer response, a result that is in line with practical understanding in direct marketing contexts.

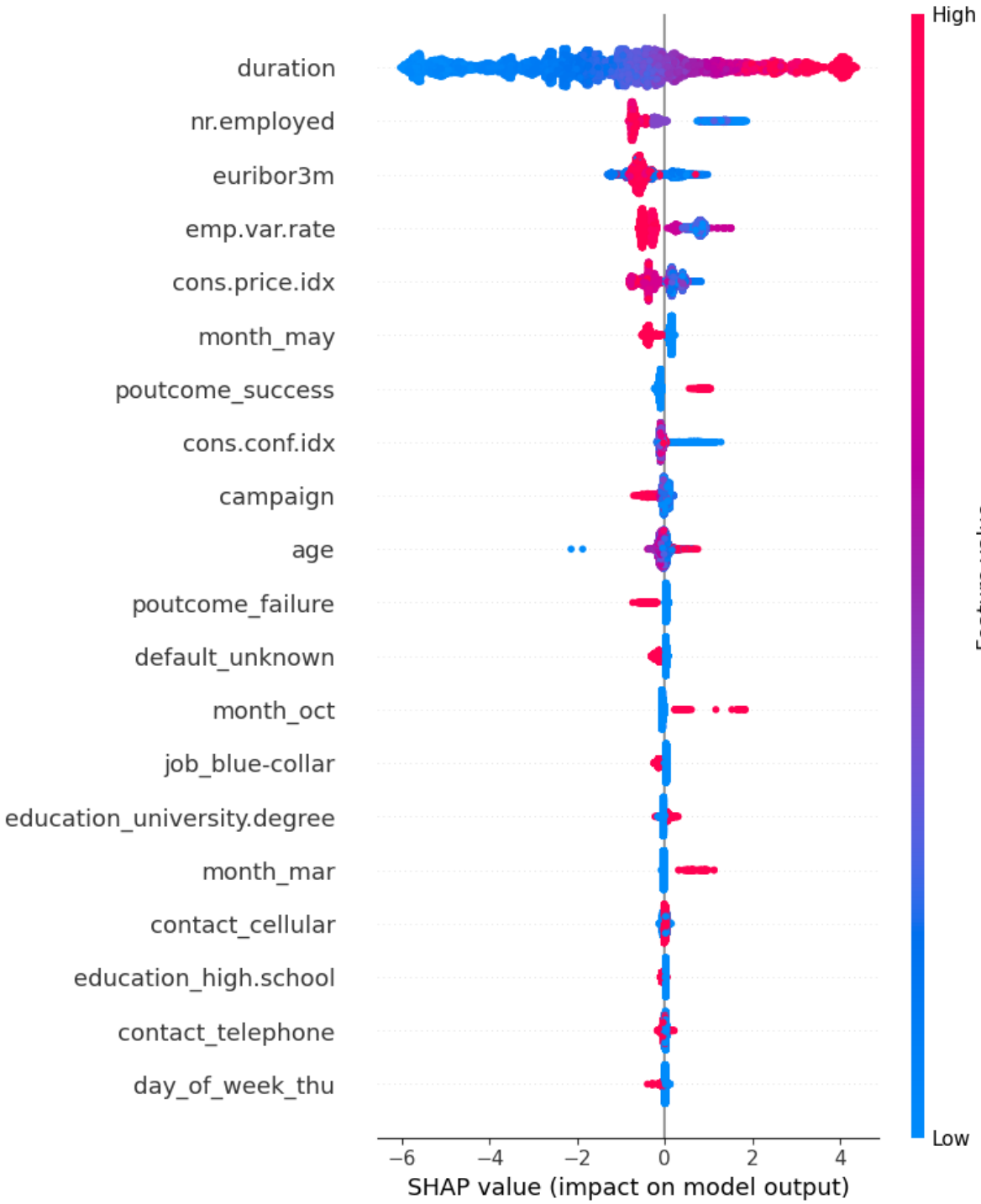


Figure 4.8: SHAP summary plot of CatBoost model.

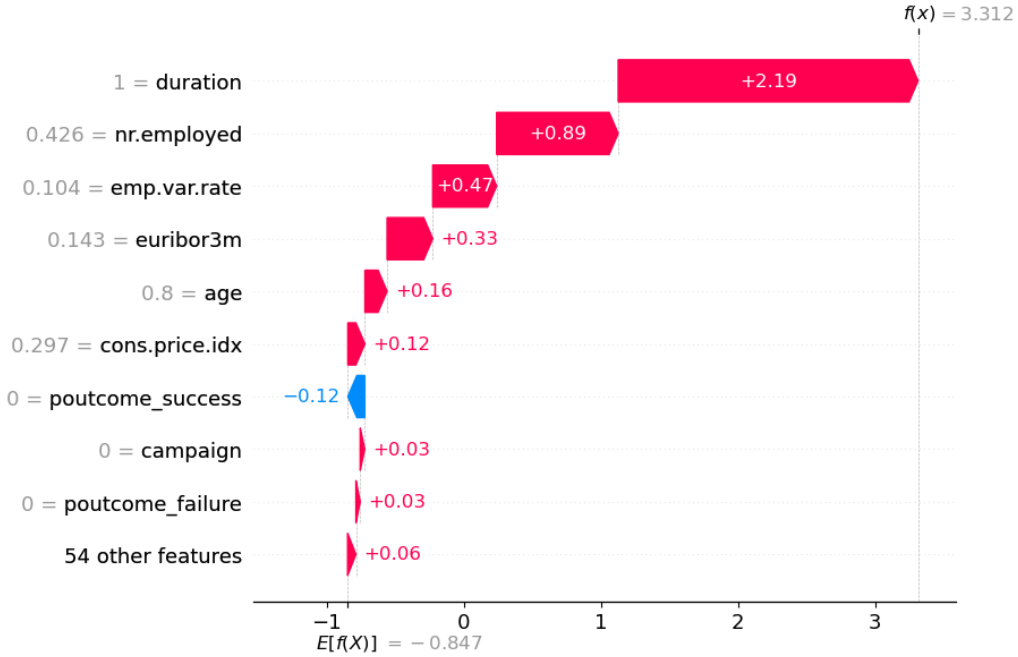


Figure 4.9: SHAP waterfall plot of CatBoost model.

Following `duration`, the next most impactful features include key macroeconomic indicators such as `nr.employed`, `emp.var.rate`, and `euribor3m`. These features consistently exhibit positive SHAP values when their levels are high, suggesting that customers are more likely to subscribe during periods of economic stability and favorable employment conditions. This highlights the sensitivity of customer behavior to broader economic trends, implying that marketing campaigns may achieve greater success when timed with positive economic indicators.

Additionally, features like `age` and `cons.price.idx` show moderate contributions, often interacting subtly with other variables. Interestingly, the binary feature `poutcome_success` — indicating whether the customer had a successful outcome in a previous campaign - contributes negatively when set to zero, as reflected in its negative SHAP value in the waterfall plot. This underlines the importance of the history of the campaign, showing that customers without prior positive interactions tend to have a lower predicted probability of subscribing.

Variables such as campaign, `poutcome_failure`, and `month_may` have relatively minor but non-negligible effects. For example, while campaign count increases may slightly raise the prediction score, this relationship is not strong enough to override the influence of more dominant variables like `duration`.

The SHAP analysis for CatBoost confirms that both behavioral interaction characteristics (e.g., call duration) and economic context features play a pivotal role in influencing subscription decisions.

Gradient Boosting & LightGBM Following the SHAP analysis of all three models (CatBoost, LightGBM, and Gradient Boosting) it is evident that the feature importance patterns identified in CatBoost are highly consistent across the other two models. Specifically, the feature `duration`, which dominated in the CatBoost model, also ranks as the most influential variable in both LightGBM and Gradient Boosting. In all cases, higher call durations are strongly associated with increased SHAP values, clearly pushing the prediction toward a positive subscription outcome.

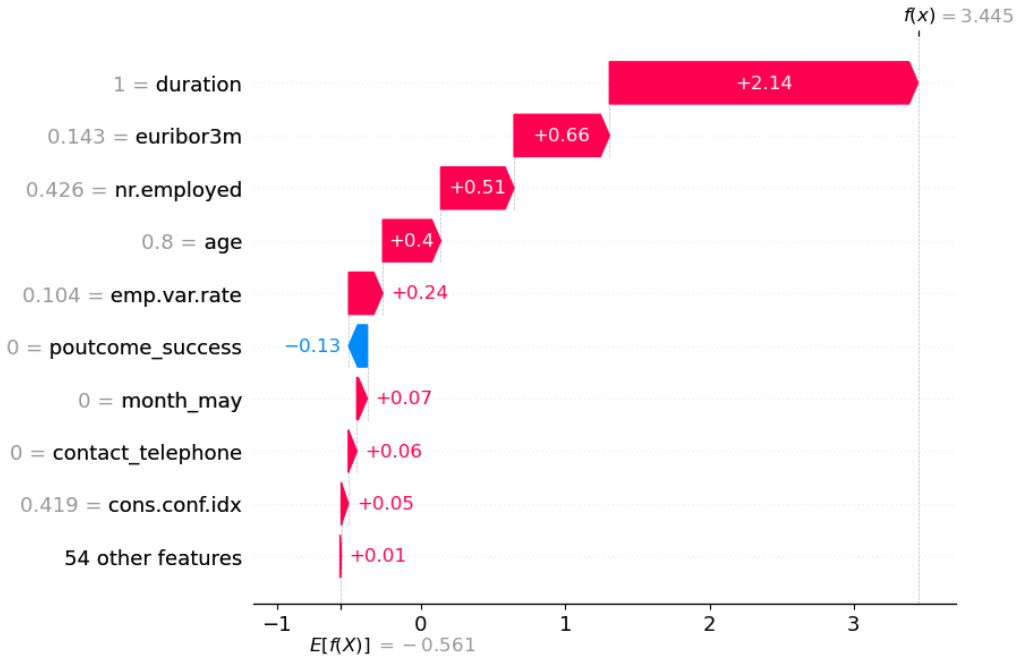


Figure 4.10: SHAP waterfall plot of Gradient Boosting model.

Moreover, a similar trend is observed with macroeconomic indicators such as `emp.var.rate`, `nr.employed`, `euribor3m`, and `cons.conf.idx`. Regardless of the learning algorithm used, these features consistently exhibit positive SHAP values when favorable, indicating that customers are more inclined to subscribe under positive economic conditions. This reinforces the explanatory power of CatBoost's results, demonstrating their relevance beyond a single model.

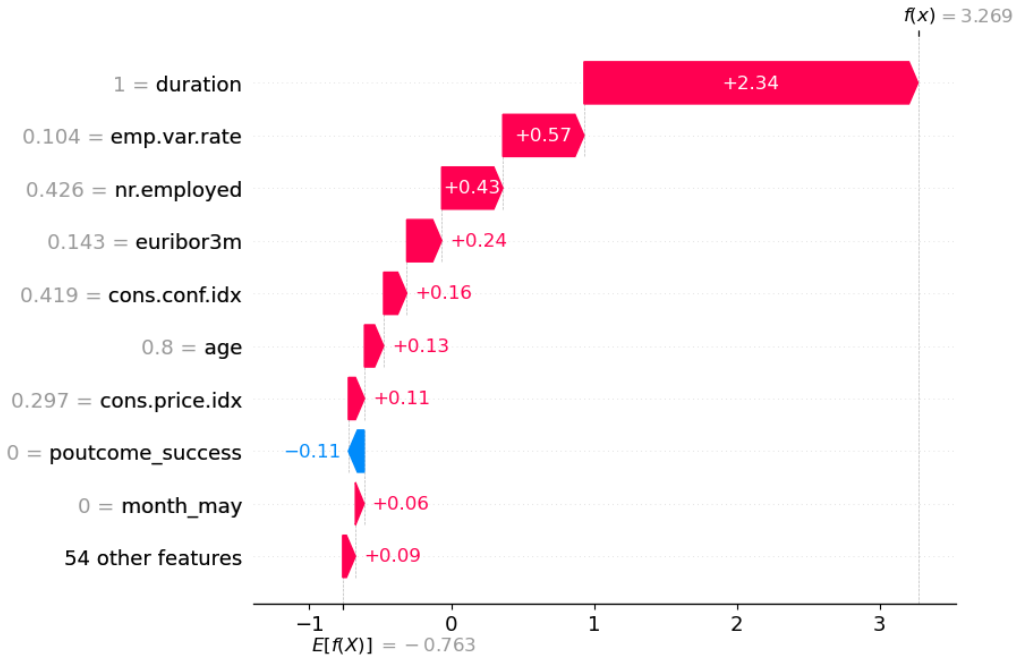


Figure 4.11: SHAP waterfall plot of LightGBM model.

Of particular interest is the binary variable `poutcome_success`, which when set to 0 (i.e., no prior campaign success) consistently yields negative SHAP values in all three models. This pattern, visible in the waterfall plots, underscores the importance of campaign history in predicting customer conversion and validates the insights originally drawn from the CatBoost model.

Furthermore, features such as `month_may`, `campaign`, and `age` exhibit moderate or low influence across all models, yet their repeated appearance among the top features affirms the robustness of CatBoost's interpretation.

The explanatory patterns revealed by CatBoost are not specific to that model alone, but are consistently replicated in both LightGBM and Gradient Boosting. This repetition not only enhances the reliability of the insights but also confirms that marketing strategies built upon these key features are grounded in model-agnostic evidence, thus supporting their generalizability and operational relevance.

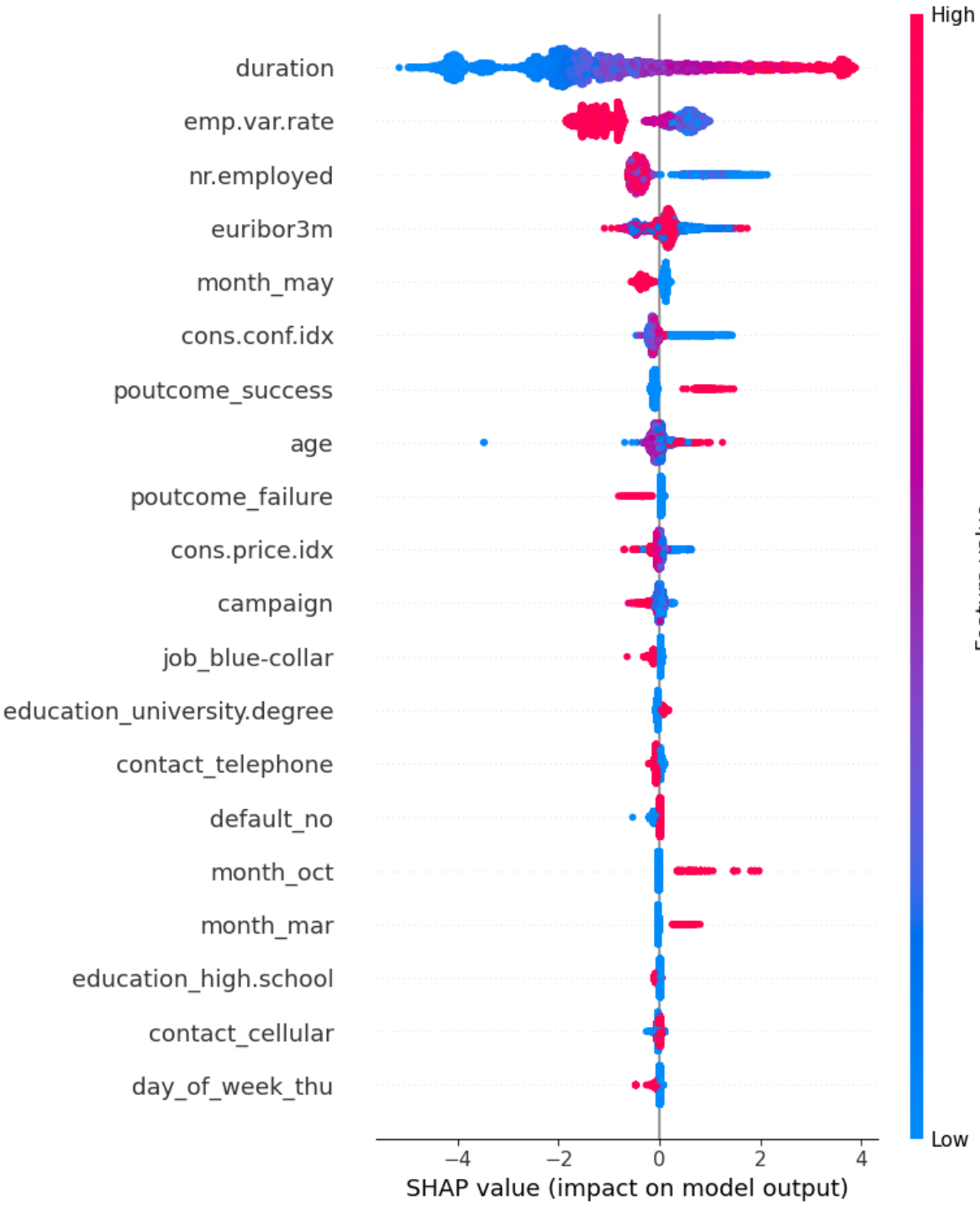


Figure 4.12: SHAP summary plot of Gradient Boosting model.

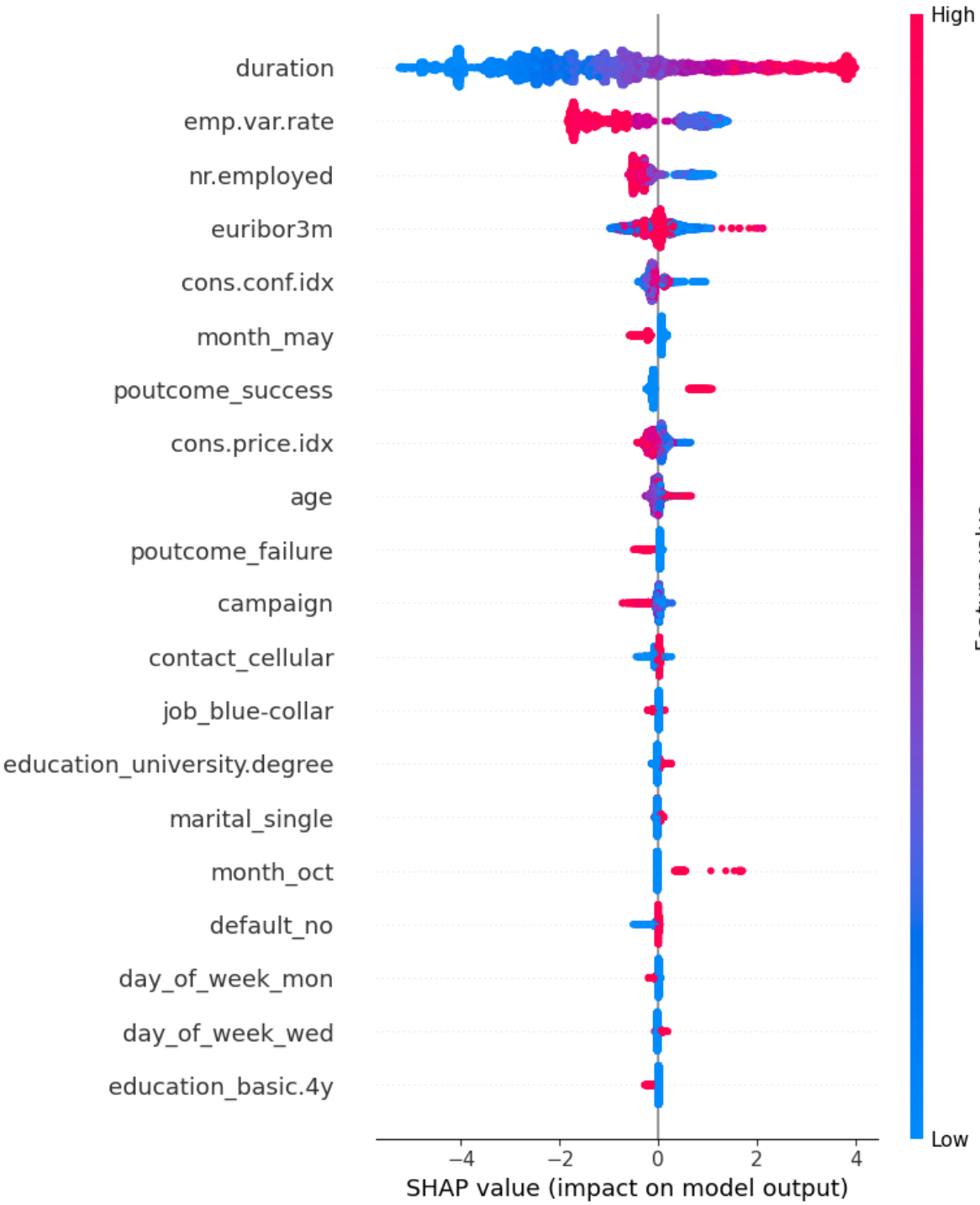


Figure 4.13: SHAP summary plot of LightGBM model.

4.2.3 LIME Analysis

Figure 4.14 presents the LIME explanation for a specific prediction made by the CatBoost model, offering a localized interpretation of why this instance was classified as a non-subscriber with a high probability of 96%. While SHAP provided a broader view of

global and individual feature influence, LIME focuses on a linear approximation of the model's behavior within the local neighborhood of the selected observation.

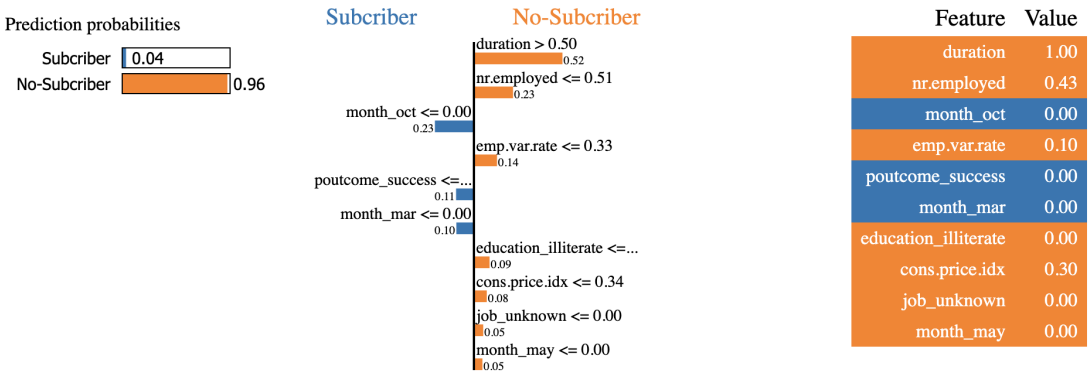


Figure 4.14: LIME for CatBoost model.

CatBoost In this particular case, the most influential factor pushing the prediction toward **non-subscription** is `duration > 0.50`, contributing 0.52 to the final decision (show in Figure 4.14. This suggests that the call, while present, was relatively short, as high `duration` values in previous analyses were associated with positive conversion. Additionally, a relatively low value of `nr.employed` (0.43) and a modest `emp.var.rate` (0.10) further support the non-subscriber classification, indicating that unfavorable labor market conditions may have discouraged the customer from committing to a deposit.

Other features such as `education_illiterate`, `cons.price.idx`, and `job_unknown` also reinforce the non-subscription prediction, albeit with smaller weights. These variables reflect either low financial literacy or economic uncertainty, both of which can reduce the propensity to engage in long-term financial products.

On the opposite side, only a few features act in favor of the subscriber prediction, including `month_oct`, `poutcome_success`, and `month_mar`, all of which have values of 0. These appear as weak signals, slightly increasing the probability of subscription but not strong enough to outweigh the dominant non-subscriber indicators.

Overall, this LIME analysis confirms and complements the SHAP-based insights: shorter interactions, coupled with weak economic conditions and lack of prior campaign success, significantly decrease the likelihood of subscription. The consistency between global and local interpretability methods enhances trust in the model's reasoning and highlights actionable variables for marketing refinement.

4.3 Analysis of Misclassified & Uncertainty Cases

This section further investigates misclassified instances and prediction uncertainty zones (defined as probability scores between 0.45 and 0.55) to assess the consistency and robustness of model performance. This analysis serves to reinforce earlier findings and provides a more comprehensive perspective on the reliability of the classification outcomes.

4.3.1 Misclassified Cases

Analyzing misclassified cases is a critical step in thoroughly evaluating model performance and identifying potential weaknesses that warrant improvement. Figure 4.15 presents an overview of the misclassification rates across the six evaluated models, based on a test set comprising 8, 236 samples.

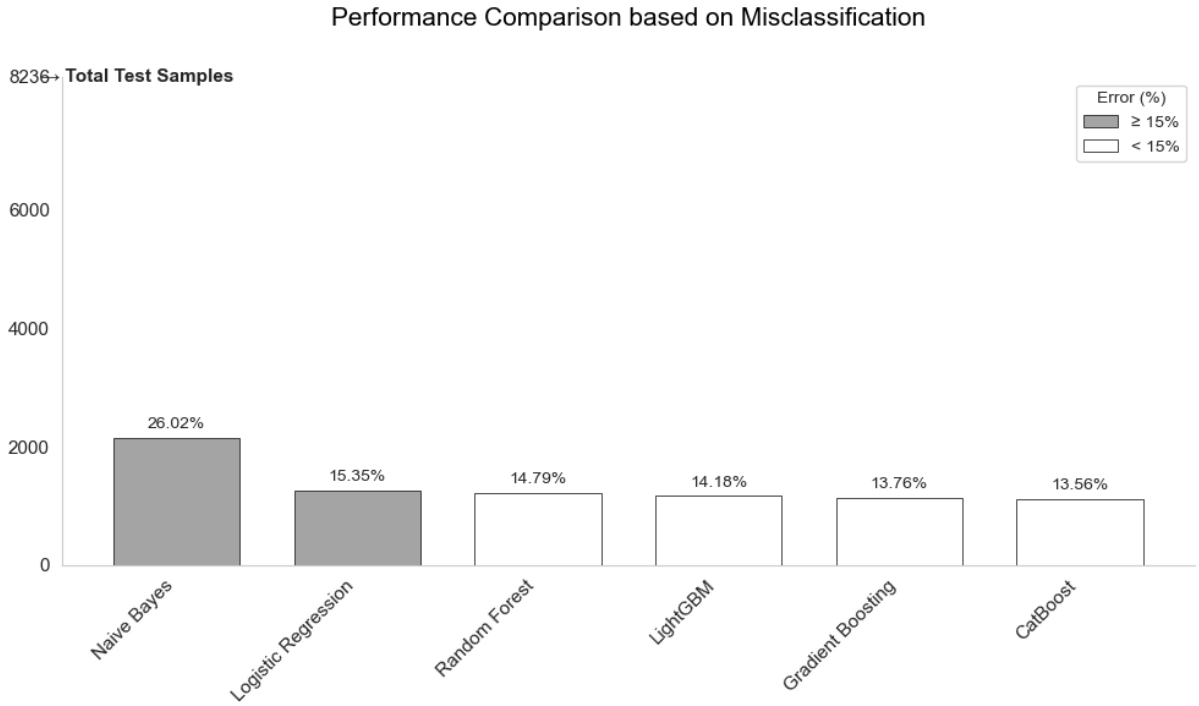


Figure 4.15: Performance comparison based on misclassification.

The results displayed in the chart reveal that ensemble models demonstrate superior performance, with significantly lower misclassification rates compared to traditional models. Specifically, **CatBoost** achieves the lowest error rate at **13.56%**, indicating the highest classification accuracy, followed by **Gradient Boosting** (13.77%), **LightGBM**

(14.18%), and **Random Forest** (14.79%). In contrast, conventional models such as **Logistic Regression** (15.35%) and particularly **Naive Bayes** (26.02%) exhibit notably higher error rates. This highlights the advantage of ensemble methods in capturing complex and nonlinear relationships in the data, which traditional statistical models often fail to model effectively.

These misclassification results align closely with previous feature importance and SHAP analyses, reinforcing the superiority of ensemble approaches especially CatBoost in addressing the problem of predicting term deposit subscriptions. The substantial performance gap between Naive Bayes (26.02%) and CatBoost (13.56%), amounting to **12.46%**, clearly illustrates the value of leveraging advanced machine learning algorithms in this context.

From an error analysis perspective, it can be inferred that misclassified cases are partly attributable to the model's heavy reliance on the variable `duration`, which, according to prior SHAP analysis, exerts the strongest influence on prediction outcomes. Instances where the interaction duration is long but the customer still declines the offer, or conversely, where the duration is short yet the customer agrees to subscribe, can pose significant challenges for the model in making accurate classifications.

In addition, macroeconomic indicators such as `nr.employed`, `emp.var.rate`, and `euribor3m` were also identified as highly influential through SHAP analysis. Irregular fluctuations in these variables may give rise to atypical scenarios that the model struggles to classify correctly. For example, during periods of unusual economic volatility, customer behavior may deviate from the patterns the model has learned, leading to mispredictions.

Another potential source of classification error lies in the inherent **class imbalance** present in the banking dataset, where the proportion of customers agreeing to subscribe to term deposits is significantly lower than those who decline. This class imbalance may hinder the model's ability to effectively learn the characteristics of the minority class, thereby resulting in a higher rate of **false negatives** cases where customers who actually subscribed were incorrectly predicted as non-subscribers.

4.3.2 Uncertainty Cases

The analysis of uncertainty cases provides an additional perspective on the performance and reliability of the models. Uncertainty cases are defined as predictions with probabilities falling within the range of **0.45** to **0.55**, representing instances where the model cannot make a confident decision due to the predicted probability being too close to the classification threshold of **0.5**.

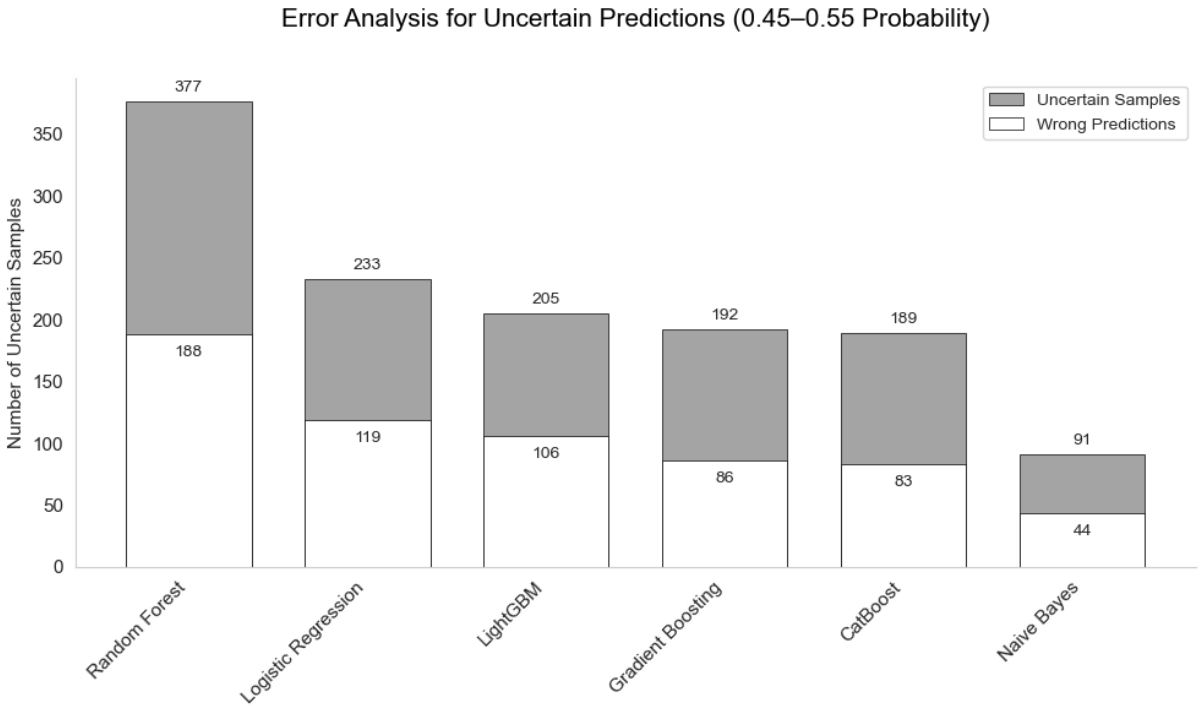


Figure 4.16: Error analysis for uncertain predictions (0.45-0.55 probability).

As illustrated in Figure 4.16, there are notable differences across models in handling uncertainty cases. **Random Forest** shows the highest number of uncertain samples (377), with 188 of them misclassified, corresponding to an error rate of approximately **49.9%**. This suggests that, despite its overall good performance with a total error rate of **14.79%** as discussed earlier, Random Forest struggles with borderline cases where the decision boundary is unclear.

Logistic Regression and **LightGBM** also exhibit high error rates within the uncertainty zone, at **50.9%** (119/234) and **51.7%** (106/205), respectively. This aligns with earlier feature importance analyses, indicating that while these models are capable of capturing key relationships in the data, they still face challenges in dealing with complex, ambiguous cases near the classification threshold.

Notably, **CatBoost** — the best-performing model overall with the lowest total error rate (**13.56%**) also demonstrates superior handling of uncertainty cases, with an error rate of only **43.9%** (83/189) in this zone. This further reinforces CatBoost's advantage not only in optimizing predictive accuracy but also in probability calibration, resulting in more reliable decision-making.

Combining this analysis with the results of misclassified cases and the earlier SHAP analysis, it can be inferred that uncertainty cases often arise when:

- Important features like `duration` fall within a medium range, providing no clear direction for prediction;
- There is conflicting information between key features, such as a long interaction duration (typically indicating a "`yes`") but unfavorable macroeconomic indicators (typically suggesting a "`no`");
- The customer belongs to a minority segment underrepresented in the training data.

Overall, the analysis of uncertainty cases provides an important perspective on model reliability, especially in business decision-making contexts where both accuracy and confidence levels are critical. These findings highlight **CatBoost**'s strengths not only in overall performance but also in well-calibrated probability outputs, making it a compelling candidate for real-world deployment.

CHAPTER 5.

DISCUSSION

Insights from Data Analysis

Through in-depth data analysis and preprocessing, several features were identified as being strongly associated with customers' decisions to subscribe to term deposit products. Among them, the variable `duration` emerged as the most influential factor affecting subscription outcomes. Additionally, socioeconomic indicators also played a significant role, reflecting the influence of macroeconomic conditions on consumer financial behavior. Specifically, in a stable economic environment characterized by high employment rates and favorable interest rates, customers tend to be more inclined to invest in long-term financial products. In contrast, during periods of economic uncertainty, they often adopt a more cautious approach, prioritizing liquidity and avoiding long-term financial commitments. Notably, the variable `poutcome` showed a clear and positive impact, underscoring the importance of maintaining continuous engagement with customers who have previously responded positively to marketing efforts.

Insights from Classification Results

The classification results from six different machine learning models provided valuable insights into the ability to predict customer behavior in the banking and finance sector. Notably, ensemble models demonstrated superior performance, with CatBoost achieving the highest balanced accuracy (**0.89**), while LightGBM attained the highest ROC-AUC (**0.947**) and sensitivity (**0.925**). The significant performance gap between these ensemble methods and traditional models such as Logistic Regression and Naive Bayes clearly illustrates the advantages of advanced machine learning techniques in capturing complex relationships within financial data.

Hyperparameter tuning also played a critical role in enhancing model performance. For CatBoost, optimizing parameters such as `depth`, `learning_rate`, and `l2_leaf_reg` enabled the model to achieve an ideal balance between generalization capability and model complexity. Similarly, for Gradient Boosting and Random Forest, tuning parameters like the number of trees (`n_estimators`) and maximum depth (`max_depth`) significantly

improved overall performance. These findings emphasize the importance of a rigorous hyperparameter optimization process in tailoring machine learning models to the characteristics of real-world financial datasets.

Insights from Misclassified & Uncertainty Cases

A deeper analysis of misclassified and uncertain cases provides valuable insights into the limitations of the models and opportunities for improvement. In misclassified cases, a key contributing factor is the models' strong reliance on the variable `duration`. Customers with long interaction durations who ultimately declined to subscribe, or conversely, those with short durations who agreed to subscribe, were often misclassified. This phenomenon suggests that the relationship between interaction duration and the final decision is not entirely linear and may be influenced by psychological or contextual factors that the current models fail to capture.

Macroeconomic indicators such as `nr.employed`, `emp.var.rate`, and `euribor3m` also played a significant role in misclassifications. During periods of economic volatility, customer behavior may deviate from historical patterns learned by the model, emphasizing the importance of regularly updating and retraining the model to reflect current economic conditions especially in the dynamic financial environment.

Data imbalance is another noteworthy factor, with a significantly lower proportion of customers subscribing to term deposits compared to those who declined. This imbalance may cause the model to favor majority class predictions (i.e., rejection), resulting in a higher number of false negatives—instances where customers who actually subscribed were incorrectly predicted as non-subscribers.

In the case of uncertain predictions (i.e., those with predicted probabilities in the range of 0.45–0.55), the analysis revealed notable differences across models. Random Forest produced the largest number of uncertain samples (377), while CatBoost not only had fewer uncertain cases but also exhibited a lower error rate within this group (43.9% compared to 49.9% for Random Forest). This highlights CatBoost's superior probability calibration, a critical factor for assessing prediction reliability in real-world applications. Uncertain cases often arise from conflicting signals among key features or ambiguous average feature values that do not clearly favor one decision over the other. From a marketing perspective, identifying this group is especially valuable. These are potential yet hesitant customers individuals whose behaviors are ambiguous and thus should be

prioritized for personalized care or targeted follow-up campaigns (e.g., special offers, direct consultations). Focusing marketing resources on this segment, rather than applying mass-marketing strategies, can significantly improve conversion rates and optimize campaign efficiency.

Role of XAI in Explaining the Classification Results.

The application of Explainable Artificial Intelligence (XAI) techniques particularly Feature Importance Analysis and SHAP (SHapley Additive exPlanations) plays a pivotal role in understanding the inner workings of the classification models employed in this study. While evaluation metrics such as balanced accuracy and ROC-AUC provide insights into model performance, they do not reveal how predictions are made. XAI bridges this gap by clearly identifying which features influence the model's decisions and to what extent.

Notably, SHAP enables both global and local interpretability. It quantifies the contribution of each input feature to the prediction outcome not only across the entire dataset but also at the level of individual customers. This capability is especially critical in domains such as banking, where transparency and explainability directly impact trust and regulatory compliance.

Through XAI visualizations including SHAP, LIME, and feature importance plots, the study reaffirmed the dominant role of variables such as `duration` and the broader group of social and economic context features in classification outcomes. These findings not only strengthen the model's credibility but also offer valuable insights to business stakeholders seeking to better understand customer behavior and develop more effective engagement strategies.

CHAPTER 6.

CONCLUSIONS & PERSPECTIVE

6.1 Conclusions

In this thesis, machine learning techniques were employed to predict customer subscription to term deposit products in the context of direct marketing campaigns conducted by a Portuguese bank. The analysis was based on a publicly available dataset from the UCI Machine Learning Repository, comprising 41,188 observations. Notably, only approximately 11% of customers subscribed to the term deposit, highlighting a significant class imbalance in the dataset.

To address this challenge and ensure the model's ability to learn meaningful patterns, exploratory data analysis (EDA) was conducted, followed by essential preprocessing steps. These included outlier handling, encoding of categorical variables using One-Hot Encoding, and normalization of numerical features using MinMaxScaler to ensure consistent feature scaling. Such standardization is critical to prevent the model from being biased by differing value ranges and contributes to a stable training pipeline. Subsequently, the study implemented and compared the performance of six machine learning models: traditional models such as Logistic Regression and Naive Bayes, as well as advanced ensemble models including Random Forest, Gradient Boosting, LightGBM, and CatBoost. Hyperparameter tuning was applied to optimize model performance. Furthermore, Explainable AI (XAI) techniques were integrated to interpret classification outcomes and support decision-making in practical applications.

Among the evaluated models, ensemble methods particularly CatBoost demonstrated superior performance. **CatBoost** achieved a balanced accuracy of **0.8901** and a ROC-AUC score of **0.9469**, indicating a strong ability to distinguish between the two classes and delivering consistent results across both test sets and uncertainty scenarios. These findings underscore the advantages of ensemble models in capturing non-linear relationships and handling the complex data structures characteristic of customer behavior in the financial services sector.

Through feature importance analysis and XAI techniques, the study identified the most influential variables affecting customers' decisions to subscribe. Key features included `duration`, macroeconomic indicators such as `nr.employed` and `euribor3m`, and `poutcome`, which represents the outcome of previous marketing campaigns. Notably, the importance of these features was consistently observed across all evaluated models. This indicates that both socioeconomic context and the length of interaction during marketing calls play a critical and recurring role in customer decision-making. These insights suggest that customer decisions are influenced not only by short-term behavior but also by broader economic conditions and prior engagement history with the bank.

Moreover, training strategies such as Stratified K-fold cross-validation, random oversampling to address class imbalance, and hyperparameter tuning have played a pivotal role in enhancing the model's generalization ability, mitigating overfitting, and improving predictive accuracy. Notably, the integration of interpretability techniques for misclassified cases and predictions with uncertain probabilities has helped elucidate the model's decision-making mechanisms, thereby improving its transparency and reliability. These insights are not only technically valuable but also hold significant practical implications, enabling banks to design marketing strategies that are more aligned with customer behavior, thereby optimizing conversion rates and delivering more effective personalized experiences to targeted customer segments.

6.2 Limitations & Future works

Although the study has achieved positive results in applying machine learning to predict customers' likelihood of subscribing to term deposit products, several limitations should be addressed to improve future research.

One important aspect that was not considered is the analysis of training time, computational cost, and the feasibility of deploying models in practical environments. These factors are especially relevant in the financial sector, where systems are often expected to deliver responses in real time or near real time. Ignoring such considerations may limit the practical applicability of the models, particularly in large-scale operational settings.

In addition, the dataset used in this study was obtained from a historical banking campaign. Although it was thoroughly preprocessed, it may not accurately reflect changing customer behavior or current market conditions. To enhance the robustness and real-

world relevance of future models, researchers should consider incorporating more diverse and up-to-date data sources, such as real-time transaction data or behavioral information collected from multiple channels.

Furthermore, the study applied only basic encoding techniques for categorical variables and did not explore alternative methods such as target encoding, frequency encoding, or embedding representations, which may offer improved performance. The process of hyperparameter tuning was also limited to a single method, without examining more sophisticated optimization approaches such as Bayesian optimization or evolutionary algorithms. These techniques have the potential to improve model accuracy while reducing computational cost, thereby increasing overall modeling efficiency.

REFERENCES

Bailey, C., Baines, P. R., Wilson, H., and Clark, M. (2009). Segmentation and customer insight in contemporary services marketing practice: why grouping customers is no longer enough. *Journal of Marketing Management*, 25(3-4):227–252.

Bikker, J. and Metzemakers, P. (2005). Bank provisioning behaviour and procyclicality. *Journal of International Financial Markets, Institutions and Money*, 15(2):141–157.

Borugadda, P., Nandru, P., and Madhavaiah, C. (2021). Predicting the success of bank telemarketing for selling long-term deposits: An application of machine learning algorithms. 7:91–108.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chatterjee, S. (2025). Auc-roc analysis: Understanding model discrimination. Accessed: 2025-05-10.

Chaurasiya, P. (2022). Hyperparameter tuning using gridsearchcv and randomsearchcv. Accessed on Medium.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Coussement, K. and Van den Poel, D. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34(1):313–327.

DataCamp (2025). Ensemble learning in python. Accessed: 2025-05-10.

David Stone, M. and David Woodcock, N. (2014). Interactive, direct and digital marketing: A future that depends on better use of business intelligence. *Journal of Research in Interactive Marketing*, 8(1):4–17.

Duan, X. (2023). Automatic identification of conodont species using fine-grained convolutional neural networks. *Frontiers in Earth Science*, 10:1046327.

Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied Logistic Regression*. John Wiley & Sons, Hoboken, NJ, 3 edition.

Huang, S. (2024). A comparative analysis of machine learning algorithms for predicting the telemarketing campaigns of portuguese banking institutions. In *Proceedings of ICFTBA 2024 Workshop: Finance in the Age of Environmental Risks and Sustainability*, pages 44–53. EWA Direct. Open Access under CC BY 4.0.

Jiang, Y. (2018). Using logistic regression model to predict the success of bank telemarketing. *International Journal on Data Science and Technology*, 4(1):35–41.

Khan, M. Y., Qayoom, A., Nizami, M., Siddiqui, M. S., Wasi, S., and Syed, K.-U.-R. R. (2021). Automated prediction of good dictionary examples (gdex): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques. *Complexity*.

Kotler, P. and Keller, K. L. (2016). *Marketing Management*. Pearson Education, 15th edition.

Kılıç, (2023). Light gbm: A powerful gradient boosting algorithm. Accessed on Medium.

Lessmann, S., Baesens, B., Seow, H. V., and Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136.

Li, H., Zhang, X., Fan, Y., Peng, S., Zhang, L., Xiang, D., Liao, J., Zhang, J., and Meng, Z. (2024). A hybrid approach to mountain torrent-induced debris flow prediction combining experiments and gradient boosting regression. *Water*, 16(23).

Malthouse, E. C. and Blattberg, R. C. (2005). Can we predict customer lifetime value? *Journal of Interactive Marketing*, 19(1):2–16.

Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.

Nguyen, H. (2024). IMPACTS OF DIGITAL TRANSFORMATION ON FINANCIAL PERFORMANCE: EVIDENCE FROM VIETNAM. *Financial and credit activity problems of theory and practice*, 5(58):175–184.

Palaniappan, S., Mustapha, A., Mohd Foozy, C. F., and Atan, R. (2017). Customer profiling using classification approach for bank telemarketing. *International Journal on Informatics Visualization*, 1(4-2):214–217. Available via UTHM or research repositories.

Peppers, D. and Rogers, M. (2011). *Managing Customer Relationships: A Strategic Framework*. John Wiley & Sons, Hoboken, NJ, 2 edition.

Peter, M., Mofi, H., Likoko, S., Sabas, J., Mbura, R., and Mduma, N. (2025). Predicting customer subscription in bank telemarketing campaigns using ensemble learning models. *Machine Learning with Applications*, 19:100618.

Pham, P. (2020). Outlier detection and removal using the iqr method. Accessed on Medium.

Qiu, D. H., Wang, Y., and Zhang, Q. F. (2009). A model for a bank to identify cross-selling opportunities. In *2009 International Conference on Computational Intelligence and Software Engineering*. IEEE.

Sharma, S. (2024). Marketing in the digital age: Adapting to changing consumer behavior. *International Journal of Management and Business Intelligence*, 2(1):1–14.

Thu, N. A. and Quan, T. T. (2023). Impact of digital transformation on financial decision making at Big4 banks in Vietnam.

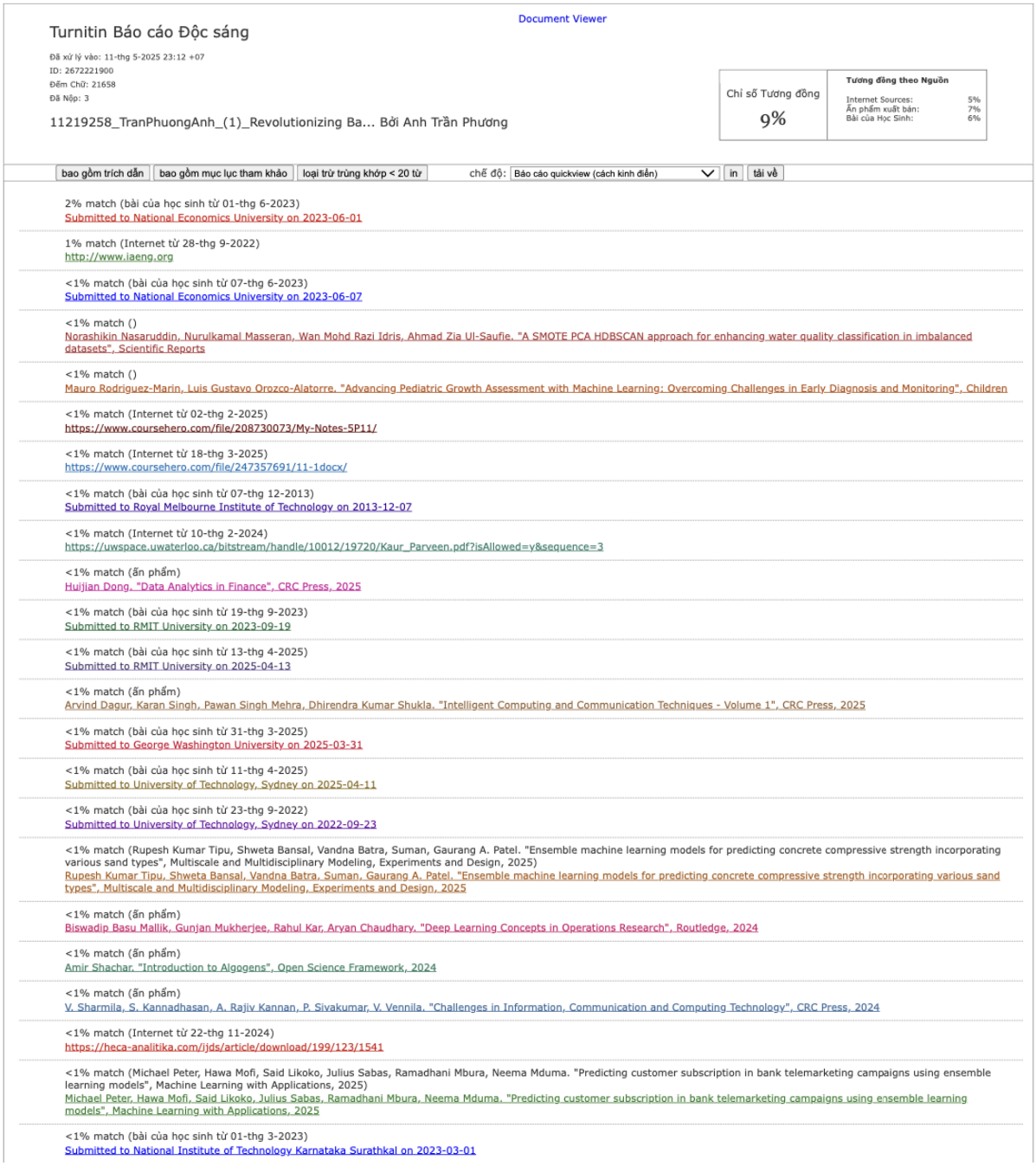
Turban, E., Aronson, J. E., and Liang, T.-P. (2005). *Decision Support Systems and Intelligent Systems*. Prentice Hall, Upper Saddle River, NJ, 7 edition.

Verbeke, W., Martens, D., Mues, C., and Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Syst. Appl.*, 38:2354–2364.

Wang, Z. and Kim, H. G. (2017). Can social media marketing improve customer relationship capabilities and firm performance? a dynamic capability perspective. *Journal of Interactive Marketing*, 39(1):15–26.

Wedel, M. and Kamakura, W. A. (1999). *Market Segmentation: Conceptual and Methodological Foundations*. Kluwer Academic Publishers, Dordrecht/Boston/London, 2 edition.

Yousefzadeh, R., Kazemi, A., and Al-Maamari, R. (2024). Application of power-law committee machine to combine five machine learning algorithms for enhanced oil recovery screening. *Scientific Reports*, 14.



Document Viewer

Turnitin Báo cáo Độc sáng

Đã xử lý vào: 11-thg 5-2025 23:12 +07
ID: 2672221900
Đếm Chữ: 21658
Đã Nộp: 3

11219258_TranPhuongAnh_(1)_Revolutionizing Ba... Bởi Anh Trần Phương

Chỉ số Tương đồng	Tương đồng theo Nguồn	
9%	Internet Sources:	5%
	Ấn phẩm xuất bản:	7%
	Bài của Học Sinh:	6%

bao gồm trích dẫn | bao gồm mục lục tham khảo | loại trừ trùng khớp < 20 từ | chế độ: Báo cáo quickview (cách kinh điển) | in | tải về

2% match (bài của học sinh từ 01-thg 6-2023)
Submitted to National Economics University on 2023-06-01

1% match (Internet từ 28-thg 9-2022)
http://www.iaeng.org

<1% match (bài của học sinh từ 07-thg 6-2023)
Submitted to National Economics University on 2023-06-07

<1% match ()
Norashikin Nasaruddin, Nurulkamal Masseran, Wan Mohd Razi Idris, Ahmad Zia Ul-Saufie. "A SMOTE PCA HDBSCAN approach for enhancing water quality classification in imbalanced datasets", Scientific Reports

<1% match ()
Mauro Rodriguez-Marin, Luis Gustavo Orozco-Alatorre. "Advancing Pediatric Growth Assessment with Machine Learning: Overcoming Challenges in Early Diagnosis and Monitoring", Children

<1% match (Internet từ 02-thg 2-2025)
https://www.coursehero.com/file/208730073/My-Notes-5P11/

<1% match (Internet từ 18-thg 3-2025)
https://www.coursehero.com/file/247357691/11-1docx/

<1% match (bài của học sinh từ 07-thg 12-2013)
Submitted to Royal Melbourne Institute of Technology on 2013-12-07

<1% match (Internet từ 10-thg 2-2024)
https://uwspace.uwaterloo.ca/bitstream/handle/10012/19720/Kaur_Parveen.pdf?isAllowed=y&sequence=3

<1% match (ấn phẩm)
Huijian Dong. "Data Analytics in Finance", CRC Press, 2025

<1% match (bài của học sinh từ 19-thg 9-2023)
Submitted to RMIT University on 2023-09-19

<1% match (bài của học sinh từ 13-thg 4-2025)
Submitted to RMIT University on 2025-04-13

<1% match (ấn phẩm)
Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Intelligent Computing and Communication Techniques - Volume 1", CRC Press, 2025

<1% match (bài của học sinh từ 31-thg 3-2025)
Submitted to George Washington University on 2025-03-31

<1% match (bài của học sinh từ 11-thg 4-2025)
Submitted to University of Technology, Sydney on 2025-04-11

<1% match (bài của học sinh từ 23-thg 9-2022)
Submitted to University of Technology, Sydney on 2022-09-23

<1% match (Rupesh Kumar Tipu, Shweta Bansal, Vandna Batra, Suman, Gaurang A. Patel. "Ensemble machine learning models for predicting concrete compressive strength incorporating various sand types", Multiscale and Multidisciplinary Modeling, Experiments and Design, 2025)
Rupesh Kumar Tipu, Shweta Bansal, Vandna Batra, Suman, Gaurang A. Patel. "Ensemble machine learning models for predicting concrete compressive strength incorporating various sand types", Multiscale and Multidisciplinary Modeling, Experiments and Design, 2025

<1% match (ấn phẩm)
Biswadip Basu Mallik, Gunjan Mukherjee, Rahul Kar, Aryan Chaudhary. "Deep Learning Concepts in Operations Research", Routledge, 2024

<1% match (ấn phẩm)
Amir Shachar. "Introduction to Algogens", Open Science Framework, 2024

<1% match (ấn phẩm)
V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024

<1% match (Internet từ 22-thg 11-2024)
https://heca-analitika.com/ijds/article/download/199/123/1541

<1% match (Michael Peter, Hawa Mofi, Said Likoko, Julius Sabas, Ramadhani Mbura, Neema Mduma. "Predicting customer subscription in bank telemarketing campaigns using ensemble learning models", Machine Learning with Applications, 2025)
Michael Peter, Hawa Mofi, Said Likoko, Julius Sabas, Ramadhani Mbura, Neema Mduma. "Predicting customer subscription in bank telemarketing campaigns using ensemble learning models", Machine Learning with Applications, 2025

<1% match (bài của học sinh từ 01-thg 3-2023)
Submitted to National Institute of Technology Karnataka Surathkal on 2023-03-01