Republic of Tunisia
Ministry of Higher Education and Scientific Research
University of Carthage
Carthage Institute of Higher Commercial Studies

End of Studies Internship Report

Optimizing SMS Deliverability Using Artificial Intelligence

Submitted in partial fulfillment of the requirements for the degree of:

Bachelor of Business Computing Specializing in Business Intelligence

Prepared by:

Houwayda Bouaziz & Saif Allah Bounawara

Academic Supervisor
Dr. Ghorbel Molka

Professional Supervisor
Mr. Youssef Iheb

In partnership with:

TRITUX Group

Academic year: 2024/2025

Dedications

I dedicate this work

To my father Raouf, my mother Aida and my brother Omar,
You are the pillars of my life and surely my greatest supporters. Having you by my side is the best thing I could wish in my whole life. Words cannot describe my feelings towards you. I can only say that I love you more than anyone in this life and thank God for blessing me with this family.
Last but not least I have a special thank to my brother for buying me what I want from Switzerland. It means the world to me.

To Joy, my furry coach and therapeutic cuddle expert,
Thank you for nibbling on my worksheets (yes, even the project one), for watching over my feet during study time. You were the only one who never doubted me especially when I was talking to myself in front of the computer. My first loyal "audience."

To Saif, my project partner,
Thank you for your collaboration during this project. Your consistency and problem-solving were truly appreciated.

To all the people who hold a special place in my life
I dedicate this work

Houwayda BOUAZIZ

Dedications

I dedicate this work

To my mother the heart of my journey
For every sacrifice, and every moment you placed my dreams above your own. Your unwavering love and strength have carried me further than .

To my father
for being my quiet force. For all your efforts, for providing without hesitation, and for standing by me with patience and belief.

To my friends, Nour and Aziz
for your support, your encouragement, and for being a steady presence throughout this challenging year.

To all the people who hold a special place in my life
I dedicate this work

Saif Allah BOUNAWARA

Acknowledgment

We want to express our deep gratitude to all the persons who have contributed to the completion of this end-of-studies project.

We thank warmly our professional supervisor , Mister Iheb Youssef , for his disponibility, his precious tips and his support throughout our internship within the compagny « Tritux Group».

We would like to express our sincere gratitude to Mr. Mohamed Amine Kilani for his invaluable support throughout our internship. His guidance, expertise, and willingness to share his knowledge played a crucial role in the success of our project. We'd like to

start by warmly thanking our academic supervisor, Dr. Molka Ghorbel. Her thoughtful guidance, sharp insights, and constant support truly helped shape our work and pushed us to think deeper and aim higher throughout this project. A big thank you also goes to

Tritux Group for welcoming us into their team and giving us the chance to bring our academic knowledge to life in a real-world setting. The experience has been both enriching and eye-opening. We're also incredibly grateful to all the professors who've

guided us along the way. Their dedication, passion for teaching, and encouragement have played a big part in our growth, both academically and personally. A special thank

you to the ***Institut des Hautes Études Commerciales de Carthage (IHEC)*** for providing a space where we could learn, grow, and prepare ourselves for the professional world. It's been a meaningful journey. And finally, our sincere thanks to the jury

members for taking the time to review our project. We truly appreciate their constructive feedback, thoughtful questions, and the opportunity to learn from their perspectives.

Table of Contents

List of Figures

List of Tables

Acronyms

- **AI** : Artificial Intelligence
- **ML** : Machine Learning
- **TSO** : Time Sent Optimization
- **UI** : User Interface

General Introduction

In digital marketing, SMS is well and alive as a channel to reach customers fast and efficiently. But, messages in saturated inboxes and much sharper security filters put genuinely adventitious stakes for businesses to make sure their messages get across - not just reaching the recipients but also piquing their interest. If a text message happens not to go through or lands with an equally disengaged recipient, it's sheer wastage of time, money, and opportunities. An SMS campaign will be successful only if the messages are

delivered successfully. Many times, messages are sent at the wrong times or to customers who simply ignore them. Traditional methods, based on rigid rules, are no longer effective recipient behaviors have evolved, and improving message deliverability now requires more refined and intelligent approaches.

Given this worrying situation, optimizing SMS deliverability is a priority. These days, it's no longer simply a matter of sending messages; it's also about ensuring they actually reach their recipients, at the right time, and that they generate genuine interest. Traditional, even conventional, approaches are showing their limitations and even inadequacies in the face of increasingly shifting customer behavior.

With this innovative focus, our final year project aims to develop an intelligent solution for optimizing SMS deliverability. Our main objective is to implement a smart system capable of analyzing historical data from SMS communication campaigns to predict the optimal time for each send. This approach will transform campaigns into more relevant and effective retention campaigns, while maximizing their impact.

To accomplish the deliverable for this task and thus move to the next iteration, we are using a regression model that predicts the ideal time for an SMS to each contact. To do this, we trained various supervised algorithms and evaluated them on a dataset that was provided to us by the host organization. We were provided with a semi-structured dataset that is stored in MongoDB so we went through data wrangling procedures to put the data in a modelling appropriate format. Then, we implemented the chosen method as a learning algorithm with an interface for ease-of-use and dashboard capabilities for visually examining campaign performance leveraging insights and recommendations. This makes results very easy to report and act on for users.

This report will summarize the successive stages in the practice of the project in four chapters.

Chapter 1: Overall framework and project management approach
In this chapter we provide an overview of the project. We discuss a brief initial analysis, the host group , need for the project and method used .

Chapter 2 : Understanding the business problem and the data
In this chapter, we focus on understanding the business problem and the available data. Our attention is on analyzing customer behavior and examining the data at hand for the analysis.

Chapter 3 :Data Modeling and Evaluation
This chapter summarizes our technical approach, detailing the AI algorithms selected for their effectiveness and those that were trained.

Chapter 4: Deployment and Interface Development
This chapter outlines the implementation of the project. We describe how the solution is integrated into an operational and user-friendly interface.

To wrap things up, our project is all about improving SMS communication by using AI to figure out the best time to deliver messages. By looking at past data and the actions of recipients, we've created a smart solution that boosts message deliverability and relevance. The goal is to turn SMS campaigns into conversations people actually welcome sent at the right time to have a real impact, while still respecting the recipients privacy.

Chapter 1

Overall Framework and Project Management Approach

1.1 Introduction

The first chapter introduces the general outline for how we ended up creating our project. We collaborated with Tritux Group, our partner with the digital artisan's touch, to revolutionize their flow of message into highly anticipated meetings. Through exploring their campaigns, we found the history: texts lost in the noise, targets underestimated. AI became our collaborator in interpreting the complex silences of the data those moments where people truly give themselves, those words that count. Moreover, we searched for a few existing SMS campaign platforms and analyzed their advantages and limitations to better understand our objective. Before selecting a methodology, we also investigated several data science approaches to determine which one aligned best with the nature of our project, ultimately concluding that CRISP-DM was the most suitable. Through illustrations and the CRISP-DM methodology, you'll find fewer technical guide pages and more cookbook recipe: ingredients (data), knowledge (algorithms), and above all, the chef's flair to serve the dish at the moment it's craved. For a successful text message is only an utterance of, "We heard. And we chose this precise timing... for you."

1.2 Project Framework

Our final goal is to develop a solution for predicting the best time to send SMS messages in collaboration with the host company **Tritux Group**. This project represents the culmination and crowning achievement of our three years of studies in Business Intelligence at the Institut des Hautes Études Commerciales de Carthage (IHEC) and makes a significant contribution to the marketing of advertising campaigns.

Throughout this project, we are making every effort to mobilize all of our knowledge and skills acquired over time in order to design an effective solution that guarantees the right time to send SMS messages to recipients, while offering high reliability in predicting optimal moments.

1.3 Presentation of the Host Organization

Tritux Group is a software publisher with specialized knowledge in software engineering, IT consulting, and outsourcing. A major player in the digital transformation of businesses in various public and private sectors, Tritux Group offers a range of digital services, software engineering solutions, and innovative high-tech products.

With over 16 years of expertise in supporting and deploying client's digitalization processes, Tritux Group offers a tailored approach based on business needs [1].

Figure 1.1: Logo Tritux Group

Table 1.1: Tritux Group – Core Areas of Expertise and Descriptions [2]

Expertises	Description
Software Engineering	Modern, optimized, and customized information systems.
Audit and Consulting	Expertise that meets the needs of our client partners.
Solution Integration	Assistance in defining your project specifications.
Operations and Maintenance	Anticipation and provision of the best solution, at the right time.
Outsourcing Support and Training	Talent incubator for your digitalization projects. Specific and customized training and support services.

Table 1.2: Tritux Group – Solutions and Service Descriptions [3]

Solutions	Description
IoT	A wide variety of IoT solutions meeting the specific needs of businesses.
Telecom Solutions	A complete suite of innovative roaming and telecom solutions.
IT Solutions	Multifunctional tools that manage operational and business challenges.
Public Solutions	Innovative solutions to boost your brand image and maximize your revenue.

1.4 Preliminary study

Before initiating this project, it is important, even essential, to lay the foundations with a precise study. This key step, shown here, is based on three pillars: a problem defining the challenges to be addressed, a carefully considered solution, and a needs analysis anchored in reality. Far from being just a rule, this step by step represents our way of thinking: grasp before acting, question before proposing; it therefore draws a guide that is both hard and real, which reflects your commitment to a concrete response to the challenges seen.

1.5 Problem statement

At some point, every marketer has asked themselves the following question: What is the optimal time to send text messages? Are recipients more likely to open messages in the morning or late evening? What's on the agenda for Tuesday at noon? This problem is called TSO. Why is it so important? Because it allows marketers to send text messages at the optimal time for each contact. It also allows for more effective interaction with contacts, capturing their attention when they are traditionally most attentive to their texts. Imagine you want to launch an advertising campaign. The more text messages are opened, the more revenue the company will earn and the easier it will be to promote a product. Not only is the goal to increase the open rate, but you also need to ensure that the user reads the text message. This means interacting with the text message, whether by scrolling through it or clicking on the links it offers. Sending time optimization, for the reasons mentioned above, is already being practiced by many large and medium sized companies. However, they do not disclose their methods for commercial reasons. If a technique for achieving this were made available online, everyone would rush to adopt it, and the goal would then become futile, because it is very likely that at some point, each user would be overwhelmed with numerous messages from various companies, which would harm the TSO's profits.

1.6 Study of Existing SMS Campaign Platforms

In this section, we present a selection of widely used SMS campaign platforms and analyze their main characteristics. These platforms, such as EasyBulkSMS and BulkSMS, are commonly adopted by marketing teams and organizations for mass messaging purposes. We conduct a comparative study of these tools by examining their features, strengths, and limitations in terms of usability, automation, analytics, and scalability.

1.6.1 Current SMS Campaign Workflow

Right now the majority of SMS campaigns use traditional bulk SMS service providers, who allow companies to upload a list of contacts, and write the message content, with a basic interface which allows users to schedule a send time. The process typically follows these steps:

- **Contact List Preparation:** Marketer/Campaign Manager prepares an Excel type file or database, containing customer phone numbers, customer detail, and segments.

- **Messages Writing:** Marketer composes a standard SMS message, which is often the same for every customer.
- **Start Time Selecting:** Campaigns are usually scheduled manually, at a time thought to be appropriate, and not based on any actual user behavior.
- **Sending Using Bulk SMS Platforms:** The bulk SMS services, such as EasyBulk or BulkSMS, send the message to all contacts.
- **Post-Campaign Analysis:** Once sent, some basic post send statistics are collected, such as delivery status, and the bounce rate. Very few analytics are provided to users, and none can inform predictive or personalized changes to be made.

Potential Flaws with the Workflow:

- No intelligent targeting or personalization.
- No selection of optimal send time based on user actions and behaviours.
- Little or no insight as to how users interacted with or responded to messages.

The workflow above presently provides a clear opportunity for improvement leveraging AI to assist in the delivery of messaging and in general, effectiveness of campaigns.

1.6.2 Existing SMS Campaign Platforms Description

In order to better understand the current practices and identify areas for improvement, we conducted a study of existing SMS campaign management platforms. This section presents a comparative analysis of two widely used solutions: BulkSMS and EasyBulkSMS.

- **EasyBulkSMS:** is a user-friendly, cloud-based solution specifically designed to simplify the process of sending bulk SMS messages, particularly for marketing campaigns. Its intuitive interface is tailored to non-technical users, allowing businesses and marketing teams to quickly create, schedule, and send messages without requiring programming knowledge. The platform emphasizes speed and ease of execution, making it ideal for small to medium-sized enterprises seeking fast outreach capabilities. EasyBulkSMS often includes essential features such as contact list management, delivery reports, and message templates, streamlining campaign management from end to end [4].

Figure 1.2: Logo EasybulkSMS

- **BulkSMS:** is a more established and globally recognized SMS gateway provider. It caters to a wider range of users from small businesses to large enterprises and supports both web-based and API-integrated message dispatch. This allows developers to automate SMS sending directly from their systems or applications. BulkSMS offers advanced features such as message scheduling, delivery tracking, two-way messaging, and support for international delivery, making it a robust solution for businesses with diverse communication needs. Its reliability and versatility have earned it a strong reputation in industries requiring large-scale or time-sensitive messaging. [5].

Figure 1.3: Logo BulkSMS

1.6.3 Critical Analysis of Existing SMS Campaign Platforms

Several SMS campaign platforms exist today to assist businesses in reaching out to large audiences effectively. Among the most widely used are EasyBulkSMS and BulkSMS. While these tools provide a functional foundation for sending bulk messages, they are often limited when it comes to advanced features like AI-powered delivery time optimization, personalized contact targeting, and real-time dashboard analytics.

The following table summarizes the main advantages and limitations of these two platforms, providing a critical perspective on their capabilities and outlining the motivations for developing a more intelligent, data-driven SMS campaign solution.

Table 1.3: Comparative Analysis of Existing SMS Platforms [6]

SMS Platform	Advantages	Limitations
EasyBulkSMS	- Intuitive and user-friendly interface suitable for marketing teams with limited technical knowledge. - Fast and efficient message dispatch system. - Fully cloud-based with no setup needed. - Supports basic features like message templates and contact list import/export.	- Does not support personalized or intelligent scheduling based on recipient behavior. - Lacks integration with predictive models or AI. - No advanced data visualization or dashboards. - Limited message analytics.
BulkSMS	- Reliable and robust infrastructure for sending large volumes of SMS worldwide. - Offers flexible API integration for automated messaging workflows. - Supports message scheduling and delivery reports. - Compatible with various CRM systems and third-party platforms.	- No support for personalized delivery time optimization based on user interaction patterns. - Dashboards are basic and do not provide real-time insights. - Does not leverage machine learning for improved engagement rates. - Requires technical knowledge for advanced features (e.g., API usage).

This comparative analysis clearly shows that while both EasyBulkSMS and BulkSMS are capable platforms for basic SMS campaigns, they lack intelligent automation, adaptive targeting, and predictive capabilities.

These missing features are critical in the modern digital marketing landscape, where personalization and timing significantly influence customer engagement. The proposed system in this thesis aims to fill this gap by incorporating AI-driven optimization and real-time insights through interactive dashboards, all integrated into a unified, centralized, and shared application for multi-user access.

1.7 Proposed solution

Given the constraints seen in the way we currently send SMS, we have thought of a clever concept to enhance the deliverability of messages using artificial intelligence methods. The idea is based on a well-structured methodology that directs the entire process journey from understanding the business problem all the way to deploying a working solution. After reviewing CRISP-DM and its structured approach, and comparing it with other methodologies such as KDD and SEMMA, we made an informed decision that CRISP-DM fits best for our project.

Our methodology includes data extraction and cleaning, analyzing the data, predicting when the best deliverability time is using supervised learning; and finally building it into an intuitive interface for deployment and dashboards so that the model can be used in real-time. This approach not only enhances the deliverability of messages using artificial intelligence methods, but also provides real-time insights through interactive dashboards, integrated into a unified, centralized, and shared application for multi-user access. All these steps ensure that our solution is both technically sound and attuned to real business goals.

1.8 Adopted Methodology

When it comes to making informed decisions, data mining has become an essential step for businesses. Over the years, several methodologies have been developed to facilitate this task. Among them, the CRISP-DM methodology is considered one of the most popular and effective. It provides a structured and iterative framework for managing data mining projects, which has become a benchmark model for most companies. In the following section, we will take an in-depth look at the CRISP-DM methodology to better understand how to successfully conduct a data mining project.

1.8.1 Choice of Methodology and Justification

Selecting a methodology is essential to any data science project's success. A clear framework gives your project a chance to be completed on time, and with the capability to reach technical and business objectives. Although the organization that was hosting us uses the CRISP-DM methodology when a data-based project is using data for deployment-focused projects, we decided to research and compare other well-used methodologies: KDD, SEMMA, and CRISP-DM.

The comparative analysis allowed us to understand the strengths and weaknesses of their approaches towards our specific end goals. This way, the decision to choose a methodology is structural, and not only positioned to identify industry trends, but methodological consistency and suitability related to the problem from the context analysis. From the analysis you can also justify and validate your final selected methodology.

Knowledge Discovery in Databases (KDD): KDD is a structured process designed to extract valuable insights and patterns from large datasets. It focuses on turning raw data into useful knowledge that can support business or scientific decision-making [7]. The process is composed of five main steps:

1. **Selection:** Identifying and extracting the relevant data from larger datasets.
2. **Preprocessing:** Cleaning the data to remove noise and inconsistencies.
3. **Transformation:** Converting data into suitable formats for analysis, often through normalization or dimensionality reduction.
4. **Data Mining:** Applying analytical techniques or algorithms to discover patterns or build predictive models.
5. **Interpretation/Evaluation:** Assessing the discovered results to ensure they are valid, useful, and aligned with the project goals.

While KDD provides a solid technical framework for data exploration, it does not explicitly cover business understanding or deployment phases, which limits its use in end-to-end data science projects.

SEMMA: Developed by SAS, SEMMA (Sample, Explore, Modify, Model, Assess) is a sequential process focused on the technical side of data mining. It provides a practical workflow but lacks emphasis on business understanding or deployment phases, which limits its application in end-to-end projects [8]. The SEMMA process consists of five key steps:

1. **Sample:** Selecting a representative subset of the data to ensure efficient analysis and modeling.
2. **Explore:** Visualizing and exploring the data to identify patterns, trends, anomalies, or relationships.
3. **Modify:** Preparing the data by cleaning, transforming, and engineering features to improve model performance.
4. **Model:** Applying statistical or machine learning algorithms to develop predictive or descriptive models.
5. **Assess:** Evaluating the model's accuracy, stability, and usefulness based on relevant performance metrics.

Unlike KDD or CRISP-DM, SEMMA assumes that business objectives and deployment strategies are already defined outside the methodology, which makes it less comprehensive for projects that require alignment between business needs and technical execution.

CRISP-DM (Cross-Industry Standard Process for Data Mining): CRISP-DM is one of the most widely adopted and comprehensive frameworks for managing data science and data mining projects. It provides a structured and iterative process that integrates both the technical and business aspects of a project, making it suitable for end-to-end implementations from understanding business objectives to deploying the final solution [9]. The CRISP-DM process is composed of six well-defined phases:

1. **Business Understanding:** Defining the business objectives and converting them into a data mining problem definition.
2. **Data Understanding:** Collecting initial data and exploring it to identify quality issues, patterns, or anomalies.
3. **Data Preparation:** Transforming and preparing the raw data for modeling, including cleaning, integration, and formatting.
4. **Modeling:** Selecting appropriate modeling techniques and building models based on the prepared data.
5. **Evaluation:** Interpreting the results of the models to ensure they meet business objectives and assessing their reliability.

6. **Deployment:** Implementing the model in a production environment and integrating it into the business process.

CRISP-DM stands out for its iterative nature and strong focus on business alignment and deployment, which makes it highly effective for real-world projects that require both analytical rigor and operational impact.

Table 1.4: Comparative Table of Methodologies

Phase / Step	**KDD**	**SEMMA**	**CRISP-DM**
Business understanding	—	—	Business understanding
Data Selection / Sampling	Selection	Sample	Data Understanding
Data Preprocessing	Preprocessing	Explore	Data Preparation
Data Transformation	Transformation	Modify	Data Preparation
Modeling	Data Mining	Model	Modeling
Evaluation	Interpretation	Assess	Evaluation
Deployment	—	—	Deployment
Suitability for This Project	Limited (no deployment, no UX)	Too technical (no business context)	Complete & Structured

Justification for Choosing CRISP-DM as Methodology

CRISP-DM methodology has been adopted as an appropriate methodology for developing our project, which involves facilitating communication through SMS transmission via a computer to reach intended recipients. In data-driven projects, CRISP-DM is known for its rigor, flexibility, and efficiency.

The reason behind this is that the approach guarantees a proper understanding of the business problem, ensures adequate model evaluation, and emphasizes proper data preparation; hence predictive models for SMS optimization can be constructed well. This also means model customization according to user requirements becomes easier. The place where it acts greatly is in ensuring valid and applicable results where applicability matters most; for instance, in determining inactive numbers or deciding when to send or split contacts.

Last but not least, CRISP-DM is recognized in the industry as having real-world deployment; hence it serves well when aiming for this solution's long-term success and scalability.

Additionally, it is important to note that the host organization we collaborated with, Tritux Group, applies the CRISP-DM methodology in its data science and deployment projects. This alignment with their internal processes not only helped streamline our work but also ensured our solution followed industry-recognized practices for effective development and operational integration.

1.8.2 CRISP-DM Process According to Our Project

The basic principle of the CRISP-DM methodology is to follow an iterative and cyclical process to solve problems related to data exploitation. The methodology is structured into six interdependent phases that include domain understanding, data collection, data preparation, modeling, evaluation, and deployment. Each phase is designed to provide useful results to the next phase while ensuring data quality and model validity.

The methodology also allows for great flexibility to adapt to the specific needs of each project. The following figure represents the life cycle of the CRISP-DM methodology.

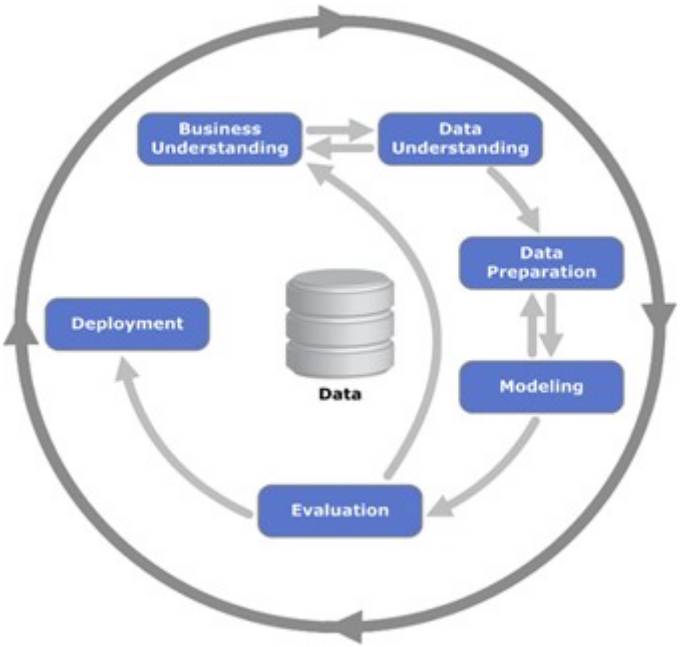

Figure 1.4: CRISP-DM Lifecycle

The CRISP-DM project management methodology is a standardized seven-step framework for data science project management:

1. **Business Understanding:** Before moving on to the technical aspects, it is essential to fully understand the business context. In our case, this means analyzing how SMS campaigns are currently managed, identifying objectives (such as improving deliverability or better targeting contacts), and also identifying any specific constraints or expectations the company has. This first step lays the foundation for the project and guides all future work.

2. **Data Understanding:** Once the framework is defined, we turn our attention to the available data. We gather information related to previous campaigns (such as sending times, campaign name, deliverability rates, etc.), then assess their quality and usefulness. This step allows us to get a clear idea of what we have at hand and begin to identify the most interesting variables to use next.

3. **Data Preparation:** This is where we really get into the practical phase. We clean the data, process missing values, transform certain columns so they can be used by the algorithms, and encode anything categorical (such as days of the week). The goal is to put the data in an optimal format for training the models.

4. **Modeling:** At this stage, we begin building our artificial intelligence models. This could involve, for example, predicting whether a number is active, estimating the best times to send a text message, or classifying contacts based on their behavior. We test different algorithms, train them on our data, and look for the ones that offer the most promising results.

5. **Evaluation:** Once the models are created, we need to ensure their performance. We use performance metrics to compare the different approaches (RMSE, R^2, etc.), and adjust the parameters if necessary. This step is crucial to validate the reliability of the predictions before considering a real life deployment.

6. **Deployment:** Finally, the chosen model is put into production. This means it is integrated into the company's work environment, accessible to marketing teams, and used to manage campaigns. A dashboard can also be provided to visualize results and monitoring can be implemented to ensure the system remains efficient over time.

It is worth noting that the CRISP-DM method does not follow a linear approach, but rather an iterative one. This characteristic allows for continuous improvement of the model and adaptation to changes that occur in the business environment over time.

1.9 Study of the Hardware and Software Environment

Every decision that relates to hardware and software was made with care. We chose each component with precision to make sure it met our specific needs. In this section, we will introduce the key technological elements that played an important role in bringing our innovative vision to life.

1.9.1 Hardware Development Environment

To successfully carry out our project, we used machines with the following specifications as our working environment:

Characteristics of Machine 1

Table 1.5: Characteristics of Machine 1

Brand	Lenovo IdeaPad 3 series
Processor	11^{th} Gen Intel(R) Core(TM) i5
RAM Memory	8.00 GB
Operating System (OS)	Windows 10

Characteristics of Machine 2

Table 1.6: Characteristics of Machine 2

Brand	MSI
Processor	Intel Core i5-13420H, 13th generation
RAM Memory	24 GB DDR4
Operating System (OS)	Windows 10

1.9.2 Programming Language and Development Tools

- **Anaconda:** Anaconda is an open source data science and artificial intelligence distribution platform for Python and R programming languages. With Anaconda, the users benefit from a huge flexibility to develop applications, perform data analysis and create models. Its package manager called *conda* simplifies the installation and update processes, allowing developers to work on different projects with specific dependencies without conflicts [10].

Figure 1.5: Anaconda Logo

- **Jupyter:** Jupyter is a web application used to program in more than 40 programming languages, including Python and R. It is a community project whose goal is to develop free software, open formats, and services for interactive computing. Jupyter is an evolution of the Python project. It allows you to create notebooks and programs. These notebooks are used in particular in data science to explore and analyze data [11].

Figure 1.6: Jupyter Logo

- **Python :** Python is a high-level programming language celebrated for its simplicity and readability. It has become widely utilized in web development, data science,

machine learning, automation, and more, thanks to its straightforward syntax and comprehensive standard library. Its interpreted nature facilitates rapid development and debugging, making it an ideal choice for both novice and seasoned developers [12].

Figure 1.7: Python Logo

- **Visual Studio Code :** Microsoft created Visual Studio Code, known as VS Code, is an integrated development environment for web browsers, Linux, macOS, and Windows. Debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and integrated version control using Git are among the features. Users can install functional extensions, modify the theme, keyboard shortcuts, and preferences[13].

Figure 1.8: Visual Studio Code Logo

- **Postman:** is a user-friendly platform for developing and testing APIs. It allows users to send requests, view responses, and automate tests. Available on desktop and web, it supports collaboration, environment management, and API documentation, making it a valuable tool for backend and frontend developers[14].

Figure 1.9: Postman Logo

1.9.3 Web Development Framework

- **Django :** Django is an open-source web framework written in Python. Its primary goal is to simplify web application development by promoting code reusability and rapid development. Initially developed in 2003 for a local newspaper in Lawrence, Kansas, Django was officially released under the BSD license in July 2005. Since June 2008, the development and promotion of Django have been overseen by the Django Software Foundation [15].

Figure 1.10: Django Logo

- **React JS :** React.js is an open-source JavaScript library created by Meta (formerly Facebook) for building dynamic and responsive user interfaces, especially in single-page applications (SPAs). It is based on the concept of reusable components, which helps developers organize and structure their code in a modular and maintainable way. React uses a Virtual DOM to efficiently update and render only the parts of the page that change, leading to improved performance and a smoother user experience. It also supports one-way data binding, hooks, and state management, making it a powerful and flexible tool for developing modern web applications[16].

Figure 1.11: React JS Logo

1.9.4 Conception Tool

- **StarUML** StarUML is a software modeling tool that helps design and visualize system architectures using UML diagrams. It supports use case, class, sequence, and other diagrams, making it easier to plan and document complex systems clearly and efficiently[17].

Figure 1.12: StarUML Logo

1.9.5 Word processing and document composition

- **LaTeX :** LaTeX is a document typesetting system widely used in the scientific and academic fields. It offers advanced features for creating technical documents, such as research articles, theses, reports, and books. LaTeX stands out for its ability to handle complex mathematical formulas, bibliography, tables of contents, and aesthetic layout. It allows users to focus on the content of their document while letting LaTeX handle the formatting. LaTeX is highly regarded for its typographic quality and flexibility[18].

Figure 1.13: LaTeX Logo

1.9.6 Communication and Collaboration Tool

- **Google Meet :** Google Meet is a video conferencing and communication tool which was created by Google. It allows users to make online meetings, webinars, and collaborate online with superior video and audio quality[19].

Figure 1.14: Google Meet Logo

- **GitHub :** GitHub is a web-based service that uses Git for version control. It allows developers to collaborate on projects while maintaining a history of their work. GitHub supports public and private repositories, making it useful for both open-source and proprietary projects [20].

Figure 1.15: GitHub Logo

1.10 Conclusion

In this chapter, we have placed the project in its overall framework and presented an in-depth state of the art. Thanks to an in depth analysis, we were able to clarify the objectives of our application and highlight its added values. Based on this study, we will address in the next chapter the generalities relating to the intelligent systems of our project, as well as our in depth understanding of the domain and the associated data, by providing a detailed specification. Moreover , we highlighted the importance of the right technical setting, encompassing both hardware and software, for implementing our AI-based SMS optimization tool. Through careful identification of essential tools, choosing the right ones, and designing a clear, user-friendly interface that includes dashboards and the deployment of the prediction model, we achieved a highly effective experience.

Chapter 2

Understanding the business problem and data

2.1 Introduction

The second chapter of this in depth study begins with a fascinating exploration of the general concepts of intelligent systems. We cover the fundamentals of these systems, how they work, and their ability to autonomously process large amounts of data. Data collection, a crucial step in the development of intelligent systems, is then examined in detail. We explore the method used to create the dataset.

2.2 General information on intelligent systems

Before delving into the methods and techniques used for data collection, it's important to understand the fundamental concepts of intelligent systems. By understanding these key concepts, we can better understand how they work and how these systems are designed to reason and make decisions.

- **Artificial Intelligence :**
  Artificial Intelligence (AI) is a field of computer science that enables machines to perform tasks requiring human intelligence, such as learning, reasoning, and decision-making. It is used in areas like image recognition, chatbots, autonomous vehicles, and healthcare [21].

- **Data Science :**
  Data science combines statistics, computing, and domain knowledge to analyze and extract insights from data. It involves collecting, cleaning, analyzing, and visualizing data to support better decisions across many industries [22].

- **Machine Learning :**
  Machine Learning (ML) is a branch of AI where models learn patterns from data to make predictions or decisions without being explicitly programmed [23].

There are two main types of machine learning:

- **Supervised learning :**
  Uses labeled data to train models for tasks like classification and regression. Common algorithms include decision trees and neural networks [24].

- **Unsupervised learning :**
  Works with unlabeled data to find patterns or groupings. It's used in clustering, segmentation, and anomaly detection [25].

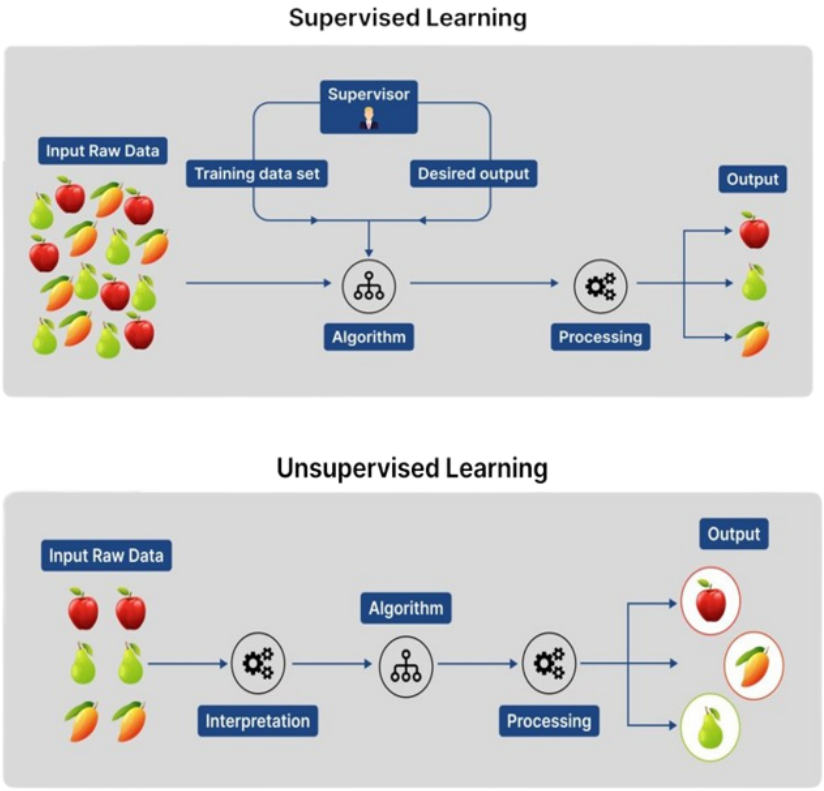

Figure 2.1: Supervised vs. Unsupervised Learning: A Comparison

2.3 Data Extraction and Preprocessing

In this section, we present how we created our dataset based on semi-structured data. The raw data in this project is initially stored in a NoSQL database: MongoDB. It was chosen not only for its flexibility and efficiency in modeling semi-structured data, but also for its ability to handle large volumes of documents like JSON efficiently.

Description of MongoDB Fields:

Figure 2.2: MongoDB data (1)

Figure 2.3: MongoDB data (2)

Each document in the MongoDB collection represents a record of an SMS communication. The key fields are classified into two categories:

➥ **Historical behavioral data:**

- **id :** Unique ID automatically created by MongoDB for each document.
- **Name :** The name of the contact hashed.
- **Statut_History :** A list that contains the history of all SMS status updates for this contact.

- **Status :** The state of the message (Sent, Received, Bounced) as a number code.
- **Status_id :** An ID that identifies each status update inside the history.
- **Status_time :** The date and time when the message changed status (was sent, received, etc).
- **Status_communication_id :** ID that tells which type of message was sent (promo, invitation, etc.).
- **Status_message :** The actual message content that was sent.

Since we are using these fields in the prediction process, we transformed them from semi-structured data in MongoDB into a structured format in an `.xlsx` file.

➥ **Personal and Demographic Information:**

- **Age:** The age of the contact.
- **Gender:** The contact's gender (e.g., Male, Female, Other).
- **Occupation:** The general profession or type of work the contact does.
- **Company:** The name of the company the contact works for.
- **City:** The city where the contact lives or works.
- **Region:** The larger area or region the contact is in.
- **Language Preference:** The preferred language of the contact for communication.
- **Country:** The contact's country of residence.
- **Phone_Os:** The operating system of the contact's phone (e.g., Android, iOS).
- **Device_Type:** The type of device used (e.g., smartphone, tablet).
- **Cost:** The total cost related to SMS sent to this contact.
- **Per_Message:** The cost for sending one SMS to this contact.
- **Encoding:** The type of character encoding used for the message (e.g., GSM, Unicode).
- **Status_history:** A list of all past SMS statuses (sent, received, bounced) for this contact.
- **Job_title:** The specific job position of the contact (e.g., Marketing Manager).
- **Working_Hours:** The usual hours during which the contact works.
- **Type_of_industry:** The sector the contact works in (e.g., Healthcare, Education, Finance).

Since this information is personal and private, we are going to keep it in MongoDB and not use it for now.

➥ **Transformation Process from MongoDB to `.xlsx` File:**

Given that we will use Python for analysis and model training, we had to make the data easier to work with. For this reason, we converted it from its original semi-structured format in MongoDB to a structured Excel (`.xlsx`) file. This process included the following steps:

- **Connecting to MongoDB:**
  Connecting to our local MongoDB database and accessing the collection containing all the pertinent data was accomplished using a Python script with the `pymongo` library.
- **Extracting the Data:**
  At this point, we imported every document from the database into a `pandas` DataFrame.
- **Exporting to Excel:**
  Once the data was prepared, we exported it into an `.xlsx` file with `pandas.to_excel()` to simplify its use in the rest of the project.

➥ **The Transformed Data:**

The figure below illustrates a snapshot of the structured dataset after transformation into Excel format:

	EventName	EventDate	HashMessage	HashContact	CommunicationName	Subject
2	Sent	2025-01-01 08:22:00	msg39952	contact1000	Limited Time Deal	Get 20% Off This Winter!
3	Sent	2025-01-01 11:57:00	msg64109	contact1000	Limited Time Deal	Exclusive Deals Just for You!
4	Sent	2025-01-01 14:39:00	msg39325	contact1000	Limited Time Deal	Don't Miss out the next Event
5	Sent	2025-01-02 06:14:00	msg50153	contact1000	Limited Time Deal	Get 20% Off This Winter!
6	Sent	2025-01-02 08:20:00	msg65772	contact1000	Product Launch	Exclusive Deals Just for You!
7	Sent	2025-01-02 08:23:00	msg20016	contact1000	Limited Time Deal	Don't Miss out the next Event
8	Sent	2025-01-02 10:14:00	msg44412	contact1000	Product Launch	Don't Miss out the next Event
9	Sent	2025-01-02 13:27:00	msg1343	contact1000	Event Invitations	Don't Miss out the next Event
10	Received	2025-01-02 14:36:00	msg1343	contact1000	Event Invitations	Don't Miss out the next Event
11	Sent	2025-01-03 12:36:00	msg48973	contact1000	Limited Time Deal	Check Out Our New Products!
12	Bounced	2025-01-23 07:36:00	msg70355	contact1000	Loyalty Rewards	Get 20% Off This Winter!
13	Sent	2025-01-03 14:28:00	msg68591	contact1000	Limited Time Deal	Check Out Our New Products!

Figure 2.4: Dataset in Excel File

2.4 Exploratory Data Analysis (EDA)

This chapter emphasizes the process of exploring and investigating the dataset to extract valuable insights, discover patterns, spot anomalies, and test assumptions using various statistical methods and graphical representations, while focusing on our main goal: predicting the best sent time of the SMS for each user for a specific communication.

2.4.1 Data Inspection

The dataset used in this study comprises SMS campaign logs collected from a telecommunications platform over a defined period. Each record in the dataset typically includes:

- **EventType:** Represents the condition of the SMS in accordance with the company and the contact. The condition has three states:
  - **Sent:** The SMS has been sent to the user.
  - **Received:** The reception of the SMS by the user was successful.
  - **Bounced:** The SMS did not reach the user due to technical issues.
- **EventDate:** Represents the date and time from which the `EventType` is successfully set.
- **HashContact:** Represents the user's unique ID.
- **HashMessage:** Represents the SMS unique ID.
- **CommunicationName:** Represents the name of the communication of the SMS.
- **Subject:** Represents the topic of the SMS.

The dataset is composed of 36k events and 259 distinct users.

2.4.2 Data Cleaning

Once the dataset was converted into a structured format (.xlsx file), we carried out few data cleaning operations to guarantee the quality and reliability of the data used in our predictive models.

The main steps of the cleaning process were:

To ensure dataset quality and suitability for analytical and training purposes, the raw dataset required significant processing. First of all, we started by standardizing the inconsistent formatting (For example, "received", "Received", "RECEIVED" should be standardized). Further, for subsequent analysis, we converted the EventDate column from object (string) type to proper datetime format whereby we can extract additional attributes such as the sent hour and day of the week. Additionally, we transformed the communication name values from numerical codes to their corresponding labels (for example : converting "16" to "Limited Time Deal") to enhance data readability and interpretability.

2.4.3 Data Visualization and Statistics

This section presents key statistics and visualizations to better understand the dataset, in order to extract features for our model, which is critical for achieving our prediction goals.

Current Reception and Bounce Rates

This part is dedicated to examining the reception and bounce rates to establish a clear starting point for the project. By analyzing these fundamental metrics, we can better understand the extent of the problem and identify opportunities for improvement.

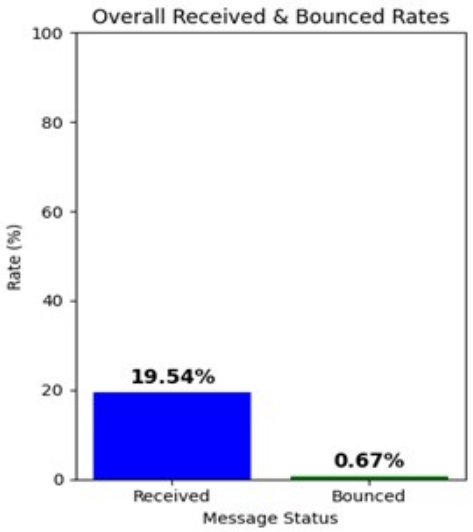

Figure 2.5: Overall Received and Bounced Rates

Through this bar chart we are able to witness a striking disparity between the receive rate and the amount of SMS sent. The receive rate appears to be significantly lower than expected, which underscores the challenge of the project and the large margin of enhancement that we have. This section clarifies further the main goal of our model and the main metrics for its evaluation in the future. It's also noticeable that the bounce rate is negligible, which emphasizes the rarity of unsent messages due to technical challenges.

Receive ratio per week day

Another metric that seems to have a significant impact on the performance of any SMS campaign is the day of the week on which the SMS is sent. People's availability and phone habits alternate significantly according to the day of the week.

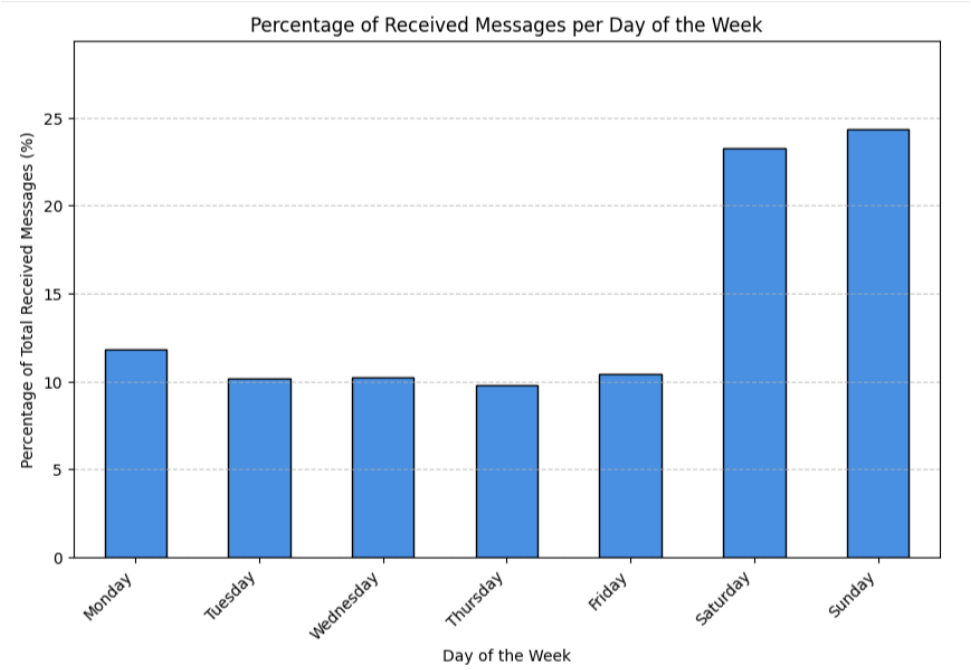

Figure 2.6: Receive Ratio per day of the week

The bar graph demonstrating the receive rate by day of week indicates the majority of SMSes are being received during weekends. This would be due to increased availability during weekends as people are not constrained by normal office hours.
This finding serves to reinforce further the significant influence which day of the week can have on user behavior, and thus further validates its value as a potentially valuable feature to keep in mind throughout the feature engineering process.

Time slot of message reception

This section shows the temporal distance between the sent and the reception of the messages that were actually received, This would be a crucial feature to capture patterns leading to accurate predictions.
The chart shows a very interesting distribution for 40 hours after sending the message.

- **Phase 1 (0-2 hours)**: Reception rates peak dramatically, with the first two hours accounting for a remarkably concentrated surge in the reception.
- **Phase 2 (2-12 hours)**: Engagement plummets to significantly lower levels, though residual activity persists at a near constant rate.
- **Phase 3 (12-27 hours)**: Reception rates continue to drop.
- **Phase 4 (27-40 hours)**: Reception rates dwindle to negligible levels.

These temporal dynamics provide a foundation for developing a weighted decay function aligned with observed engagement trends.

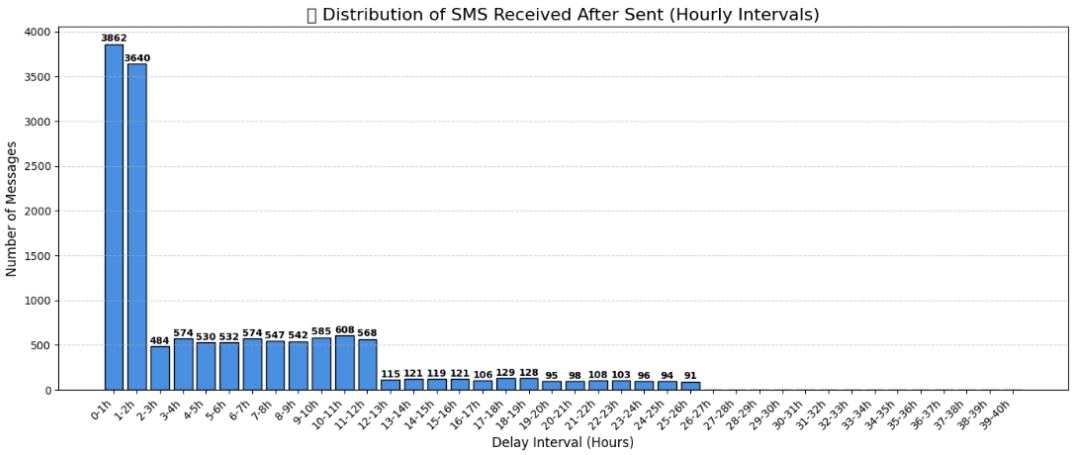

Figure 2.7: Distribution of SMS Received After Sent

Average Response Time

In the previous analysis, we observed that the frequency of receiving SMS decreases significantly approximately two hours after sending the message. This further supports the requirement to examine the mean receiving times of the users so that we can identify time intervals of the day which are unfavorable and possess a lower likelihood of message receipt.

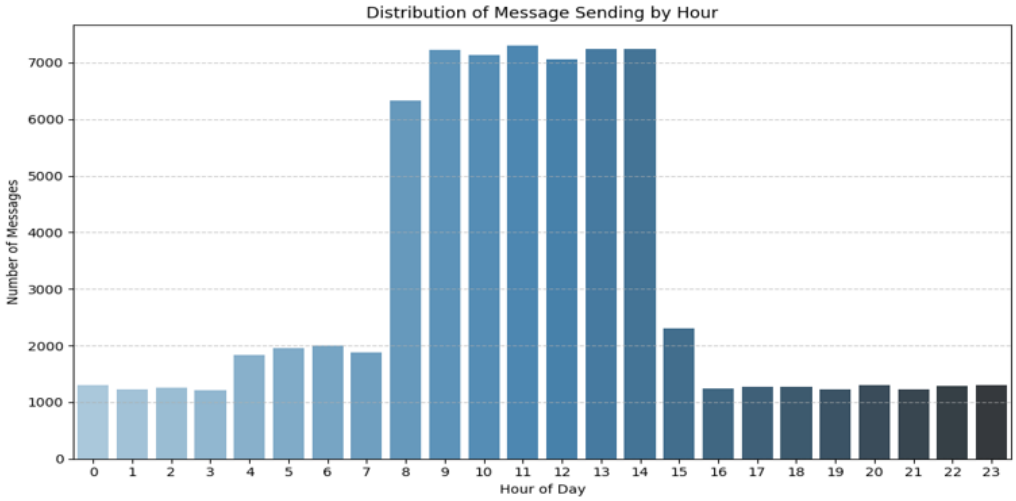

Figure 2.8: Distribution of Message Sending by Hour

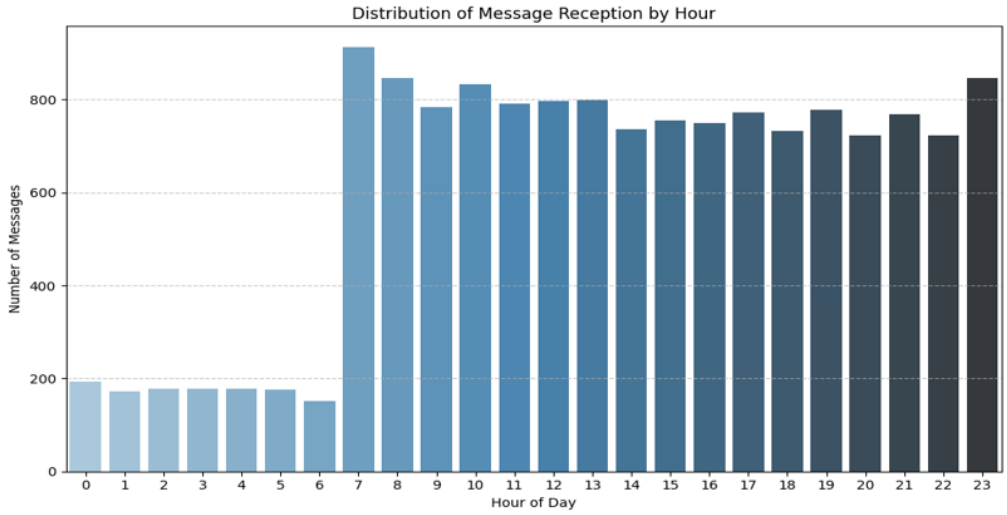

Figure 2.9: Distribution of Message Reception by Hour

Another point of interest is related to the distribution of SMS sending times. Most of the messages are being sent in the range from 08:00 to 14:00, which also overlaps with the peaks of high reception rates and can be considered an optimal move. But one thing that promises to see better improvement is the underuse of the 15:00 to 23:00 time period. Although a large portion of the messages was delivered successfully during this period, the overall SMSes dispatched during these hours remains relatively low.

This disparity suggests that the company's current strategy is perhaps overlooking a possibility to optimize deliverability timing during late afternoon and evening hours, when user activity and engagement still appear high. By adjusting the sending distribution so that it includes more SMSes during this time, the company could potentially boost overall reception rates and campaign performance.

Reception ratio per communication

We know that each SMS is related to a specific communication. Here we want to check if some communication has a higher or lower reception rate. If so, we can conclude that the type of communication would be a feature that can significantly influence the performance of our feature predictions. The following chart represents the percentage of opened SMSes among all the sent SMSes for each distinct communication.

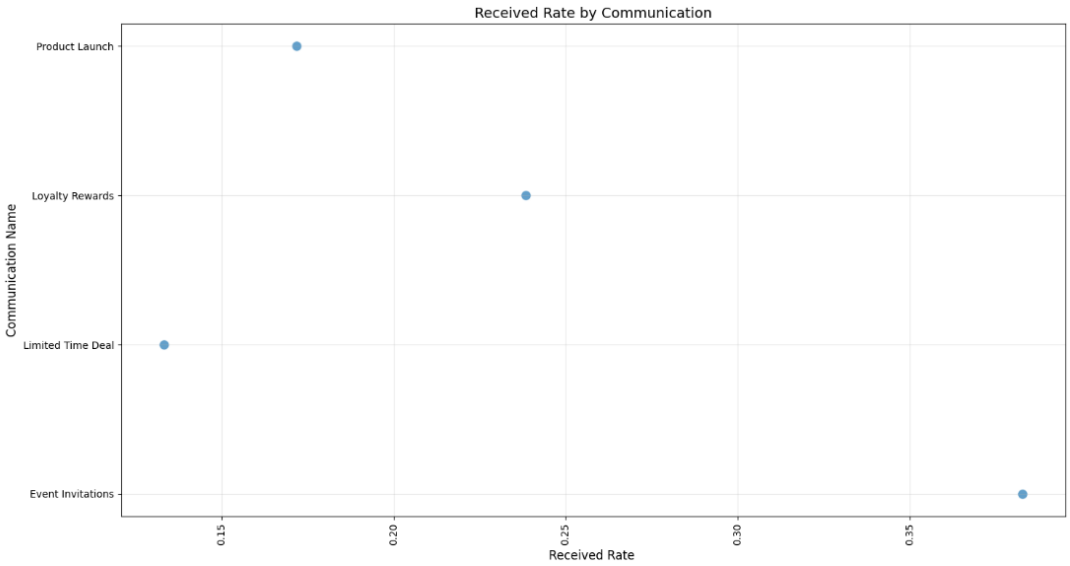

Figure 2.10: Received Rate per Communication

The plot indicates huge variation in the rate of reception between different forms of communication. While some of the communication types have very low rates of reception, nearly 0.12, others realize reception rates as high as nearly 0.43. The huge variation indicates that the effect of message deliverability is very much dependent on the form of communication being employed, indicating the necessity for adjustments based on communication type.

This is a fascinating finding that leads to a useful insight: the type of communication appears to have a measurable impact on whether the SMS is received or not. Therefore, it becomes important to include communication type as a feature in the subsequent feature engineering step because it might enhance the ability of the model to learn and predict the optimal sending times better.

2.5 Feature Engineering

Feature engineering is a preprocessing step in machine learning which transforms raw data into a more effective set of inputs. Each input comprises several attributes, known as features. By providing models with relevant information, feature engineering significantly enhances their predictive accuracy and decision-making capability.

You can say that: the more your features are well prepared and chosen, the better results you will receive. While this is largely true, it's important not to oversimplify the process. Model performance is the result of several interdependent components, including: the model you choose, the data you have available and the features you prepared, even your framing of the problem and objective measures you are using to estimate accuracy play a part. So, in this chapter, we are going to discuss which are the features that we have chosen among the possible set of features that we experimented with along with the choice of the label.

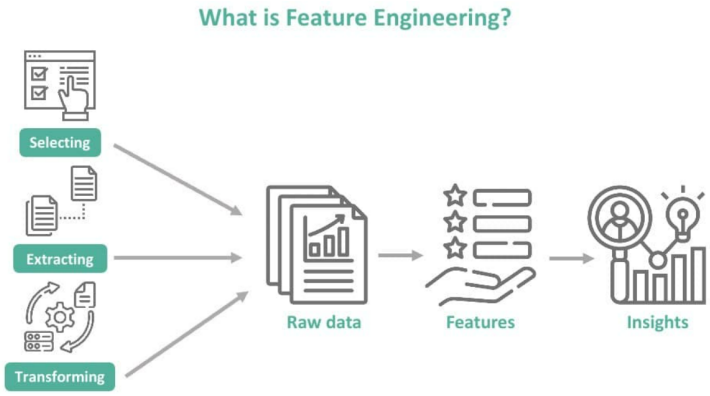

Figure 2.11: The Feature Engineering Process

2.5.1 Contact Receive Rate

This is arguably the most important feature since it has a direct influence on the target variable and the model's overall predicting performance. To maximize the value extracted from this information, we engineered this feature.

- **Receive Rate:** Here the main objective is to provide a beneficial, clear, structured and informative representation for the user's historical SMS reception behavior to assist the model in catching patterns. In order to do so, it is essential to fragment this feature into 24 distinct features, each representing an hour of the day. Each feature reflects the receive rate for that particular hour for that specific user. The purpose is providing the model with personalized hourly profile by identifying the hours from which each user mostly receives the SMS.

2.5.2 Cyclical Sent Time

From a human perspective, it's clear to us that the transition between 23:59 and 00:00 is one minute. However, machine learning models interpret numeric values linearly. The model might incorrectly perceive hour 0 and hour 23 as being far apart, rather than adjacent in time. This information is crucial and pivotal in the predictions. The following cyclical transformations were applied:

a. **Hour Sine/Cosine:** Encodes the hour of reception as sinusoidal waves, ensuring 23:00 and 00:00 occupy proximate positions in the feature space.

b. **Minute Sine/Cosine:** Similarly captures minute level granularity, reflecting the cyclical nature of intra hour intervals.

Table 2.1: Contact Receive Features

EventDate	hour	minute	hour_sin	hour_cos	minute_sin	minute_cos
25-04-21 00:00:00	0	0	0.0000	1.0000	0.0000	1.0000
25-04-21 06:00:00	6	0	1.0000	0.0000	0.0000	1.0000
25-04-21 12:00:00	12	0	0.0000	-1.0000	0.0000	1.0000
25-04-21 18:00:00	18	0	-1.0000	0.0000	0.0000	1.0000
25-04-21 23:00:00	23	0	-0.2588	0.9659	0.0000	1.0000
25-04-21 23:59:00	23	59	-0.2588	0.9659	-0.1045	0.9945

We can observe that although 00 and 23 may appear distant numerically, the use of cyclical encoding brings them closer together in the feature space.

2.5.3 Communication Receive Rate

In the exploratory data analysis conducted in the first chapter, we have discovered that the type of communication influences the reception of the SMS, which affirms the crucial role that the communication features may play in the prediction process. So we want to add two features related to communication to our dataset.

- **Communication receive rate:** As in the previous feature, here we do not have just a single column, but we have a column for each time slot, so 24 columns. Each time slot represents which is the reception rate for that communication in that specific time slot.

2.5.4 Fitness Sent Average Response Hour

This feature represents a single normalized value between 0 and 1, aimed at identifying whether the sent time chosen by the company for an SMS for a specific user is effective or not. The idea is simple: we calculate the elapsed time between the sent event and the average response hour of the user, and from then it is clear that as much as the time difference is small, the sent time adopted is reliable. Otherwise, if the gap is large, the sent time adopted is poor.

We are providing the model with the opportunity to distinguish between SMSes that are poor and those that are optimal. We already mentioned that the value will be a number between 0 and 1, but how are we going to ensure that? First, we take the elapsed time in minutes and we pass it to a decreasing exponential function that squashes the obtained number into a value between the range of 0 and 1. The used decreasing exponential function is represented below:

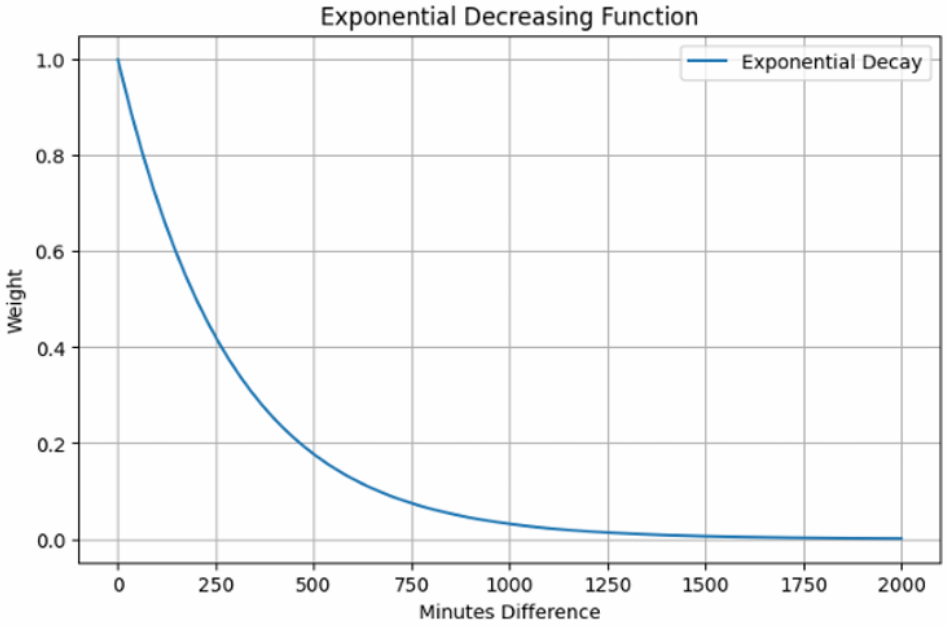

Figure 2.12: Exponential Decreasing Function

We shaped this function so that it gives us the maximum value namely 1 in correspondence with 0 on the x-axis, which indicates that if the elapsed minutes is 0, then the score is at its maximum value which is 1. Otherwise, if the SMS has been received one minute later, then the exponential decreasing function will start returning the relative on the y-axis. We have also encoded that if the elapsed time surpasses 1500 minutes, then the fitness is assigned 0.

Also, this function has another hyperparameter which is the slope of the decrease. We have tested a range of slopes, even a slope which is approximated to a straight line, and we came up with the one shown in the above figure because, in the end, it gave us the best results. This feature played a significant role in the prediction process by enabling models to recognize decreasing exponential patterns.

2.5.5 Is Weekend

Weekends generally break the weekly pattern of daily life, giving users more leisure time and increased availability and usage of their mobile phones. This increased usage significantly increases the chances of receiving an SMS during the weekend. We therefore define a binary feature indicating whether the day of sending the SMS is a weekend. We anticipate this to make a significant contribution to the model's predictions by identifying user behavioral patterns related to the calendar week.

2.5.6 Working Hours

In like manner, working hours normally between 8:00 and 18:00 hours have shown a tangible impact upon the users' behavior. Users' patterns of message reception during these hours vary due to work distractions, meetings, or commuting time. As a result, we shall project this information as a feature, stating whether the SMS was received during

normal working hours. This will allow the model to adjust its predictions in line with the assumed degree of user availability over those hours.

2.6 Building Dataset

Now that we engineered our features, we are going to build our dataset. The dataset we are going to construct is not structured like the one we started with, in which every row denoted an event. Here, the values of all the previously calculated features are contained in each row for each contact and message.

We conducted a number of trials to determine the most suitable label for our predictive model. After evaluating the performance of different options, we ultimately selected the one that yielded the most accurate and reliable results.

2.6.1 Dataset and Label Choice 1

The initial label was the SMS receive time normalized to a value between 0 and 1. To compute this, we figured out the number of minutes after midnight (from 0 to 1439) and normalized it to the [0, 1] interval. This label maintained the granularity of hours and minutes in an effort to link predictions and actual receive times intimately.

Yet this was at the expense of a high resolution prediction space (1,440 potential values) that was difficult for the model to learn due to the increased complexity and sparsity. As such, this label was eventually discarded due to low predictive quality.

2.6.2 Dataset and Label Choice 2

Our discussion here revolves around the second label we made, which is actually the one that gave us the best results. We have also introduced label 1 since it's logical to give it a try, depending on the type of data that we have. It has the potential to work better than this one. For this label, instead of having one value as the label, we have the possible time slots from 00:00 to 00:00, which are 24 time slots. But that's not it, we took each time slot and divided it further into 4 pieces. Each one represents a quarter of an hour.

Let's use an example to better grasp it: if the SMS is opened at 11:30, we begin by calculating the number of minutes that have passed since the first quarter, which ends at 0:00. Let's say that these minutes are x. As with the fitnesses, we take this x and feed it to a Gaussian function to obtain a number. This figure corresponds to the quarter 00:00. Then, at 00:15 for the following quarter, we continue in the same manner, and so forth. This allows us to label a histogram that shows the user's level of fitness over all conceivable hours. Therefore the goal now translates into learning this fitness distribution. We have used two labels of this kind:

- Since the function we used to represent the label's values is a diminishing exponential, it is evident that the function that is drawn is what we would expect: an exponential. The level of fitness increases as we approach the open quarter hour. Because we only want to minimize in one direction, you'll notice that we have 0 everywhere after the open quarter hour. This indicates that we would want our send prediction to occur prior to the actual open rather than following it.

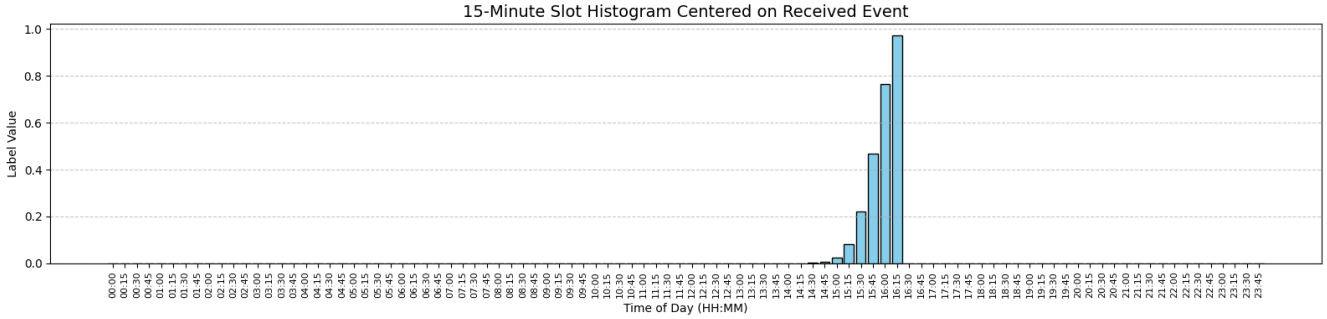

Figure 2.13: First Label Choice

- Although the first logical thinking is theoretically correct, there's a practical nuance worth addressing: If an SMS is received at 12:00 and the model predicts 12:15, it's still a very close and potentially optimal prediction. Therefore, completely assigning zero fitness to predictions after the actual receive time may penalize reasonable predictions too harshly. As such, during evaluation and later stages of model tuning, we accounted for such edge cases to ensure that nearby predictions (± 15 minutes) were not unfairly treated as poor.

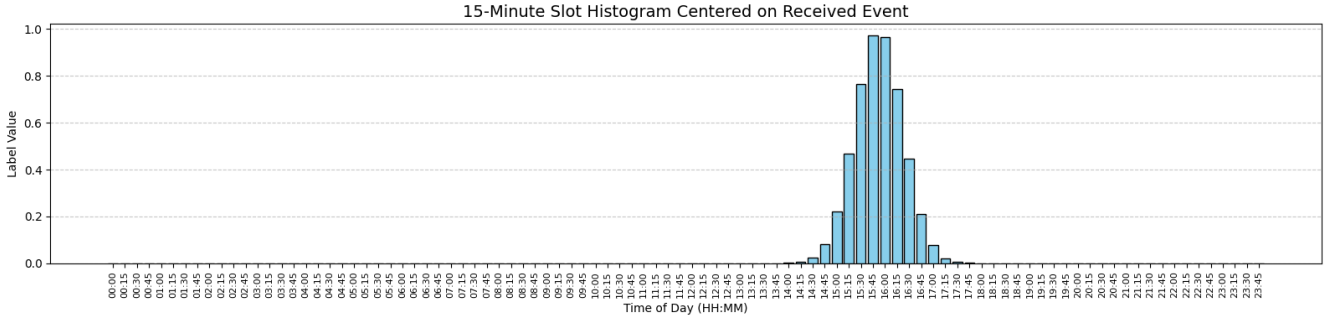

Figure 2.14: Second Label Choice

2.7 Conclusion

In this chapter, we took a deep dive into understanding both the business need and the dataset. We looked at how users interact with SMS campaigns and uncovered important trends like the impact of time of day, Weekdays vs. Weekends, and even the type of communication. These patterns helped us understand when users are more likely to receive messages. From there, we moved on to creating meaningful features that could help a model make smarter predictions. We built personalized profiles for users, captured cyclical time patterns, and even introduced a "fitness" score to measure how effective a send time really is. All of this sets the stage for the next step: building and testing models that can learn from this data and help us predict the best time to send each SMS.

Chapter 3

Data Modeling and Evaluation

3.1 Introduction

In this chapter, we pay very close attention to the critical step of preprocessing the data. We elaborately describe a range of advanced techniques designed to clean and transform the data. We then move on to the modeling phase, rightly explaining why we've selected our algorithm and providing an extensive comparison with other approaches so that you see why they're less suitable for our specific case study. Finally, we conclude the chapter by detailing comparative analysis of models and evaluation of our developing system.

3.2 Modeling

Once we completed data preprocessing, we then moved on to the modeling phase of our project. Our ultimate aim was to create a model as accurate as possible which could reliably predict the optimal SMS send time for any user. To do this, we used different algorithms, then compared accuracy of predictions and results to identify the most effective model for optimizing SMS deliverability. The following sections lay out the different stages of our modeling approach, the algorithms used, and a summary of results for our final model decision. A complete evaluation and critique of all results allowed us to determine the most accurate and suitable model, with the best degree of performance.

3.2.1 Modeling Steps

The following steps describe the modeling process in a detailed and straightforward manner:

1. **Importing Libraries and Loading Data:** To start the modeling process, we first imported key libraries such as `pandas`, `numpy`, as well as `Scikit-learn`, which present the essential tools we need for machine learning. After that, we loaded the preprocessed dataset. In this case, we have ensured that all features were clean, structured, and ready for use. This step is important and even crucial, as it prepares the data for applying algorithms effectively and ensures consistency throughout the SMS deliverability optimization process.

2. **Exporting Data from MongoDB to Excel format:** Our data was in a MongoDB database. To make the data suited for already-existing models, we exported

the appropriate data into a Excel structure so that we could train machine learning models, making it easier to manipulate as well as introduce the data to standard tools and mentioned libraries. Transforming raw event logs to "clean" tabular data provided an end-to-end and continuous workflow for feature engineering, model training, and model evaluation.

3. **Splitting the Data into Training and Test Sets:** Splitting data into training and test sets is an indispensable part of the modeling process. It involves separating the data into two parts: one for training the model and the other used to evaluate how well your model performs on unseen data. Since we are working on our SMS optimization project, this step ensures that the model generalizes well and doesn't simply memorize past patterns. We used a 70/30 split, which is a widely used principle when working with medium to large datasets. To maintain balanced evaluation, the split was done randomly while ensuring that each subset retained a representative distribution of key features and user behaviors.

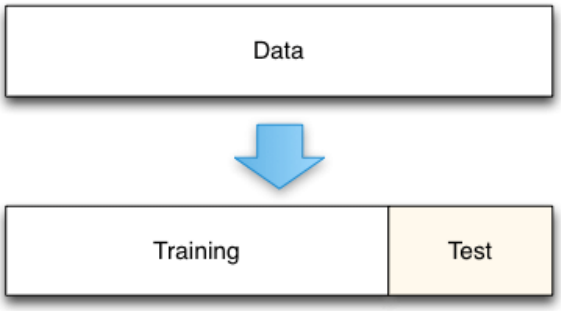

Figure 3.1: Train/Test Data Split

4. **Training the Model on the Training Set:**The training phase entails fitting machine learning models to the training data in order to learn patterns that can accurately predict optimal SMS deliverability times. In our project, we applied regression algorithms such as **Random Forest, Linear Regression and Neural Networks** to model and optimize SMS deliverability, aiming to predict the best time to send messages based on user behavior. Once the models were trained, they were used to generate predictions on the test set. This phase, while critical for model performance, can be computationally intensive, especially when working with large-scale datasets and complex temporal features.

5. **Predicting Labels on the Test Set:** Once the model has been trained, the next step is to generate predictions on the test set. In the context of our project, this involves using the trained model to forecast optimal SMS send times and determine which users are most likely to engage with a given campaign. The predicted outcomes are then compared to the actual data to evaluate the model's accuracy and effectiveness.

6. **Model Performance Evaluation:** It is essential to measure the performance of the model to find out how good it predicts the success of SMS deliverability. Various metrics such as mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R2) are used to measure performance. Nevertheless,

these metrics should not be interpreted in isolation but rather analyzed collectively to get a well-rounded view of the performance and the effectiveness of the model. Also, having a clear and separate validation set before deployment guarantees the model's reliability and ability to generalize to new unseen data, thus optimizing its overall performance.

7. **Model Saving:** Saving the trained model is a crucial step for future reuse without needing to retrain from scratch. In our project, we used standardized formats such as Python's `pickle` to store the model. This allows us to preserve the training configuration and facilitates comparisons with other models in the future to ensure that we always use the most accurate and efficient version for SMS deliverability.

3.2.2 Unsupervised Learning

The first method we applied to address the problem was unsupervised learning. This approach involves using machine learning algorithms to analyze data that lacks predefined labels, allowing the system to automatically uncover patterns or natural groupings. In the context of SMS deliverability , unsupervised learning is especially useful for identifying user segments, deliverability behavior clusters, or hidden trends in message performance without requiring prior manual categorization. Below we will define each learning method and highlight common algorithms and approaches to conduct them effectively.

3.2.2.1 Clustering

One of the key methods used in unsupervised learning is clustering, which involves grouping users with similar characteristics or response behaviors. In our case, clustering helped identify users who tend to interact with SMS messages during similar time windows or exhibit similar reception rates. These insights are crucial for personalizing send times and optimizing campaign effectiveness [26].

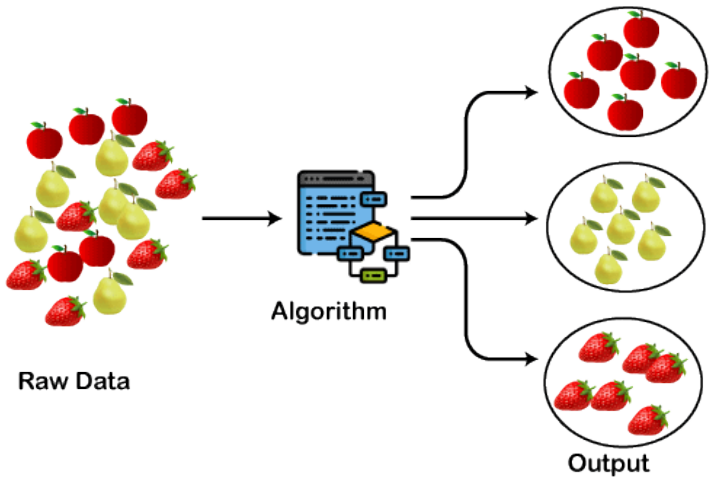

Figure 3.2: Clustering Process

K-Means

K-means clustering is a common example of an exclusive clustering method where data points are assigned into K groups, where$K$ represents the number of clusters based on the distance from each groups centroid. The data points closest to a given centroid will be clustered under the same category. A larger K value will be indicative of smaller groupings with more granularity whereas a smaller K value will have larger groupings and less granularity. K-means clustering is commonly used in market segmentation, document clustering, image segmentation, and image compression. Overlapping clusters di ers from exclusive clustering in that it allows data points to belong to multiple clusters with separate degrees of membership.[27]

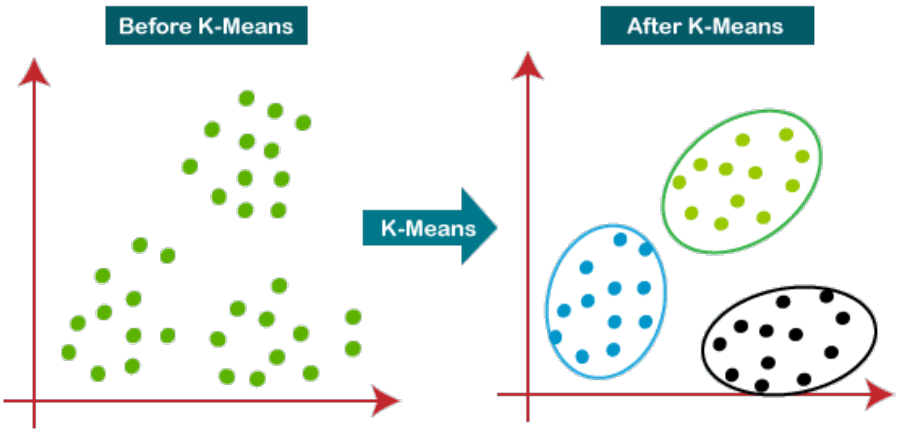

Figure 3.3: K-Means Process

3.2.2.2 Hierarchical Clustering

Hierarchical clustering is a method of cluster analysis in data mining that creates a hierarchical representation of the clusters in a dataset. The method starts by treating each data point as a separate cluster and then iteratively combines the closest clusters until a stopping criterion is reached. The result of hierarchical clustering is a tree-like structure, called a dendrogram, which illustrates the hierarchical relationships among the clusters.[28]

3.2.3 Supervised Learning

Since the unsupervised learning algorithms proved to be ineffective for our type of problem and dataset, we decided to switch to the supervised learning world. Supervised learning is a category of machine learning that uses tagged data sets to train algorithms to predict outcomes and recognize trends. Unlike unsupervised learning, supervised learning algorithms are trained in a labeled way to learn the relationship between an input and the corresponding outputs. Supervised machine learning algorithms make it easier for companies to create complex models that can make accurate predictions. Therefore, they are widely used in various sectors and fields including health, marketing, financial services.

3.2.3.1 Regression

Regression allows us to learn the relationship between independent and dependent variables and hence allows us to train models that uphold the correlation between the send time. Regression is also commonly utilized for predicting purposes, for example., for revenue of sales for a given company.
Linear regression, logistic regression, and polynomial regression are popular regression algorithms. Now, we will discuss the different regression algorithm we tried.[29]

3.2.3.1.1 Linear Regression

Linear regression analysis is used in estimating the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable. This kind of analysis computes the coefficients of the linear equation, one or more independent variables that best predict the value of the dependent variable. Linear regression graphs a straight line or plane that best reduces the disparities between predicted and actual output values.[30]

$$\boldsymbol{y} = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 x_1 + \boldsymbol{\beta}_2 x_2 + \cdots + \boldsymbol{\beta}_n x_n + \boldsymbol{\varepsilon} \tag{3.1}$$

Where:

- y: predicted value (target)
- $x_1, x_2, \ldots, x_n$: input features
- β_0: intercept
- $\beta_1, \ldots, \beta_n$: coefficients
- ε: error term

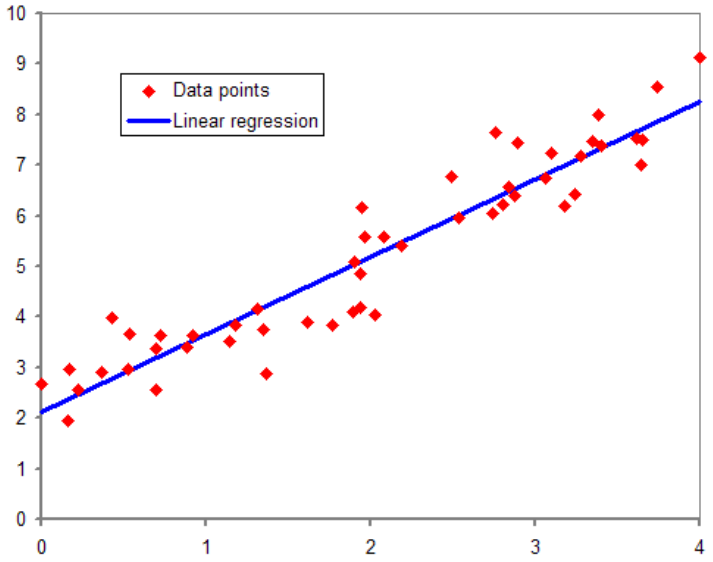

Figure 3.4: Linear Regression in ML

3.2.3.1.2 Random Forest

Random forest, as indicated by the name, consists of a large number of individual decision trees operating as an ensemble. Each decision tree in the random forest spews out a class prediction and the class which receives the most votes is our model's prediction, see the figure below :

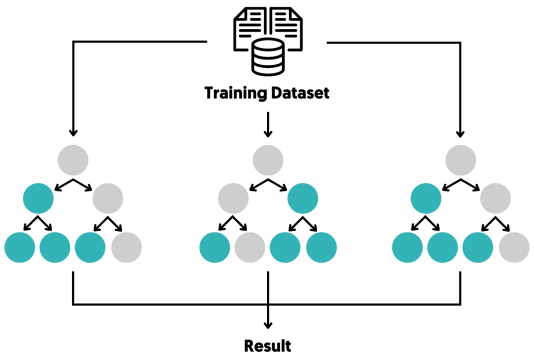

Figure 3.5: Random Forest in ML

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained by the "bagging" methodology.
The general idea of the bagging method is that the ensemble of learning models improves the final result. That is: random forest builds an ensemble of decision trees and combines them to get a more accurate and stable prediction. One of the best advantages of random forest is that both classification and regression problems, which are the majority of current machine learning systems, can be addressed by it.
Random forest shares nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there is no need to stack a decision tree with a bagging classifier because you can just use the classifier-class of random forest. You can also tackle regression problems with random forest by using the algorithm's regressor. Random forest adds additional randomness to the model, while developing the trees. Instead of searching for the best feature to split a node, it searches for the best feature from a random subset of features. This results in a wide diversity that tends to give a better model.In random forest, therefore, only a random subset of the features is considered by the algorithm when splitting a node. You can even go trees more random by also using random thresholds for each feature rather than searching for the best possible thresholds (as a normal decision tree would). [31]

3.2.3.1.3 Neural Network

Neural networks, or artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are the foundation of deep learning algorithms. They are named and constituted after the human brain and function by mimicking the way biological neurons interact with one another. Artificial neural networks (ANNs) are made up of a node layer, such as an input layer, one or more hidden layers, and an output layer. Every node, or artificial neuron, is linked to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value,

that node is activated, sending data to the next layer of the network. Otherwise, no data is passed to the next layer of the network. [32]

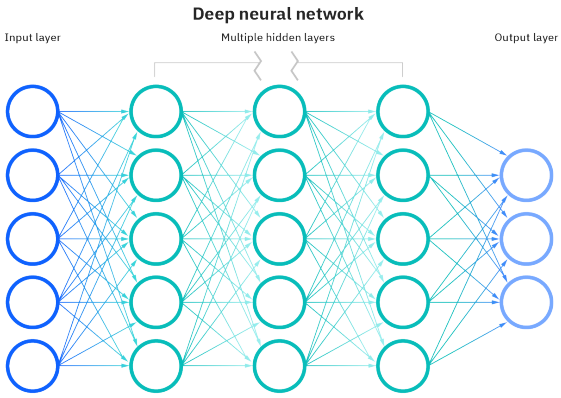

Figure 3.6: Neural Network in ML

Neural networks learn from training data and improve over time. But after these learning algorithms are optimized for precision, they are extremely valuable tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high rate. Tasks in speech or image recognition will take minutes compared to hours against the human expert manual identification.

The architecture of the neural network used in our case is the following:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	14,592
batch_normalization (BatchNormalization)	(None, 256)	1,024
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 96)	6,240

Total params: 188,002 (734.39 KB)

Trainable params: 62,496 (244.12 KB)

Non-trainable params: 512 (2.00 KB)

Optimizer params: 124,994 (488.26 KB)

Figure 3.7: The architecture of the Neural Network

These are the different components of our neural network algorithm:

- **Input Layer**: The input layer receives a vector of shape (input_dim,), the feature number.

- **Hidden Layers**: The network has three hidden layers with 256, 128, and 64 neurons respectively. All hidden layers make use of the ReLU activation function, which is widely used because it is computationally efficient with hidden layers and can address the vanishing gradient problem.

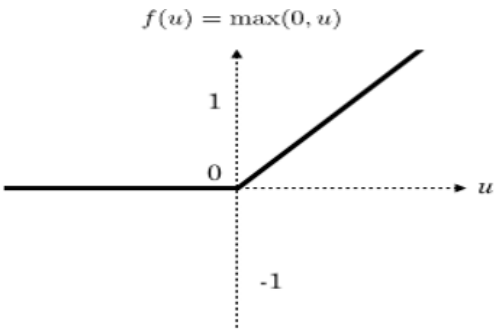

Figure 3.8: Convolutional Neural Network

- **Dropout Layer**: To avoid overfitting and enhance generalization, the model incorporates Dropout layers with increasingly decreasing rates of 0.4, 0.3, and 0.2 after each hidden layer.
- **Batch Normalization**: it is also used after the first layer to normalize layer inputs and stabilize and accelerate training.
- **Output Layer**: The output layer has output_dim neurons and utilizes the Softmax activation to output a probability distribution for time slots, aligning with the objective of the task to choose the most probable time of engagement.

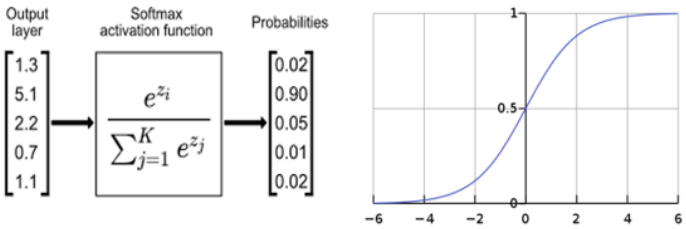

Figure 3.9: Softmax Activation Function in Neural Networks

- **Adam Optimizer**: The Adam optimizer with a learning rate of 0.0005 is used to optimize the model, adapting learning rates throughout training to converge quickly.
- **The Kullback-Leibler Divergence (KLD)**: loss function is employed since it computes the divergence between model-generated and true probability distributions, thus being ideally optimized for histogram-based outputs like these employed in this time-slot prediction task.

3.2.3.1.4 XGBoost

XGBoost (eXtreme Gradient Boosting) is a distributed, open-source machine learning library that uses gradient boosted decision trees, a supervised learning boosting algorithm that makes use of gradient descent. It is known for its speed, efficiency and ability to scale well with large datasets.

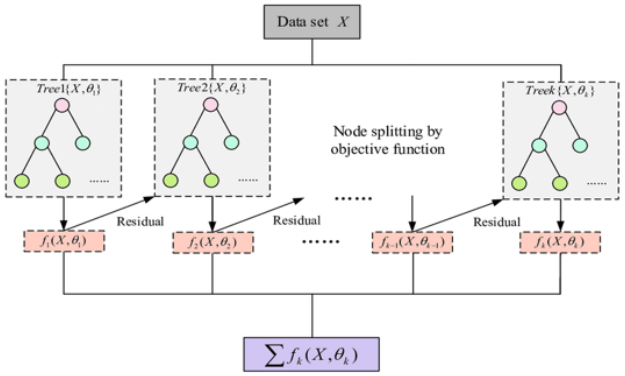

Figure 3.10: Architecture of the XGBoost Model

It builds decision trees sequentially with each tree attempting to correct the mistakes made by the previous one. The process can be broken down as follows:

- Start with a base learner: The first model decision tree is trained on the data. In regression tasks this base model simply predicts the average of the target variable.
- Calculate the errors: After training the first tree the errors between the predicted and actual values are calculated.
- Train the next tree: The next tree is trained on the errors of the previous tree. This step attempts to correct the errors made by the first tree.
- Repeat the process: This process continues with each new tree trying to correct the errors of the previous trees until a stopping criterion is met.
- Combine the predictions: The final prediction is the sum of the predictions from all the trees.

3.2.3.1.5 Comparison of Algorithms

Table 3.1: Comparison of Algorithms

Algorithm	**Type**	**Handle non linearity**	**Strengths**
Linear Regression	Parametric	Poor	Simple, fast, interpretable, good for linear data.
Random Forest	Ensemble (Decision Trees)	Excellent	Handles non-linearity, robust to noise.
XGBoost	Gradient Boosting	Excellent	High performance, handles missing values, effective for tabular structured data.
Neural Network	Deep Learning	Good	Learns complex patterns, adaptable to high-dimensional and non-linear data.

3.3 Time Sent Optimization Algorithm

3.3.1 TSO 1 Algorithm

The first attempt at making predictions about the sent time of SMS messages was an unsupervised learning task. The premise of this approach is that users who share similar behavior patterns should preferably receive messages around the same time. Therefore, if we could identify user clusters based on these patterns, we can assign send times.

To do this, we applied our dataset to a clustering process that grouped users into clusters depending on how they responded to previous SMSes. Once clustered, we analyzed each cluster in terms of whether members within a cluster received or did not receive messages. This produced three categories of clusters:

- **Green Cluster**: Reception of all SMS messages sent to them.
- **Orange Cluster**: Users who received some of the SMS messages, but not all.
- **Red Cluster**: Users who did not receive any SMS messages.

The algorithm then applied specific rules for each cluster:

- **Green Cluster**: Since users here already received all the messages, the present sending time is already optimal. We keep the same sending time for sending to such users in the future.
- **Orange Cluster**: For users who did get some of the messages, we calculate the mean send hour of successful deliveries. We take the mean time as the new send time for messages that were not received by similar users in this cluster.

- **Red Cluster**: This is the most difficult case since no messages were received by these users. Here, we check the send times used for these users and confirm that they didn't work.Calculate the Euclidean distance of this red cluster to the other clusters (orange or green). Identify the closest non-red cluster :
  - If the closest cluster is orange, we use the mean successful send time for this orange cluster.
  - If the closest cluster is green, we randomly select a send time from the send times employed for users who are in the green cluster.

It is a simple, rule-based algorithm. It is not learning-based but a reference point for the predictive task. It may be helpful in some circumstances but was not helpful in our case. The set was very dispersed and had lots of small, fractured clusters, which made this technique ineffective. Nonetheless, we chose to describe this method because of its ease and potential application in other, less broken data sets. It's a speedy method to try out, and it's a good starting point before trying more complex models.

3.3.2 TSO 2 Algorithm

We here describe the second Time Sent Optimization (TSO) algorithm, a regression based method tailored to our SMS dataset. The algorithm's central objective is to predict the most appropriate time slot for sending an SMS, by learning from historical user interaction data and optimizing for maximum message reception likelihood.

The algorithm operates in the following key steps:

1. A regression model is trained over the labeled data, where each sample is a historical SMS event with added features such as `is_weekend`, working hours, communication type, and other engineered features discussed above.

2. Once the model has been trained, it is applied to a filtered subset of the dataset defined by a specific `HashContact` and `CommunicationName`. If no such match exists, the model generalizes using all data points that belong to the contact.

3. For each relevant message, the model forecasts a distribution over 96 time slots (each 15 minutes). These distributions are then averaged out to obtain a smoothed probability curve showing the best times of day to send an SMS.

4. 4. The algorithm calculates the peak of this distribution, the time slot with the maximal predicted probability as the most optimal time to send the message. This peak is plotted alongside the whole probability curve to facilitate understanding the model recommendation.

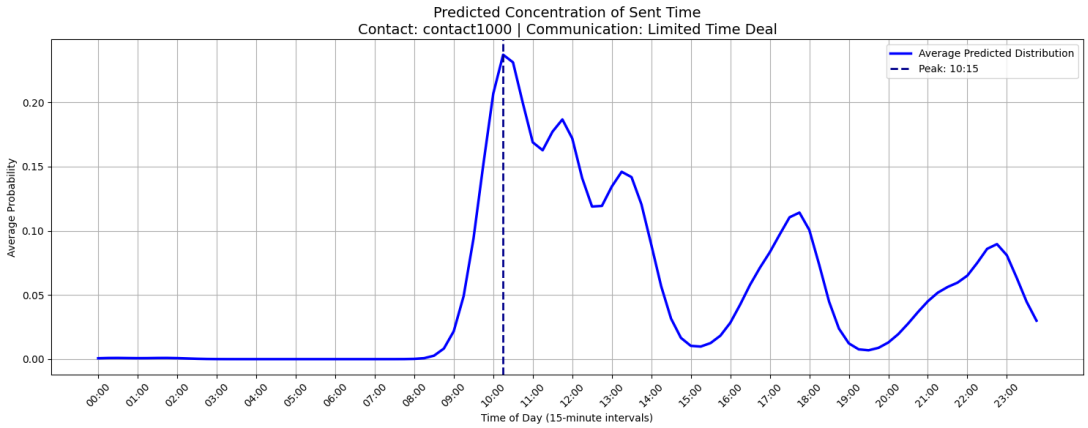

Figure 3.11: Predicted Concentration of Sent Time

3.4 Metrics and Evaluation

Before actually building and running our models, the million-dollar question is: once a model's trained, how do we know it's good? We need to have a good, useful collection of metrics that we can use to compare what the performance of the different models we experiment with are. And, importantly, the metrics need to reflect what will happen in real production. For example, if our model is expected to increase the open rate by 15%, this improvement should be realistic and observable once the model is deployed.

Without appropriate evaluation metrics, the alternative would be to train and deploy a model, then wait several months to gather new data and assess its performance an approach that is both time-consuming and impractical.

Because our task is given as a regression problem that involves predicting the best time to fire an SMS, we select the right evaluation measures to most accurately reflect the correctness of the model in real life. These are:

- **R^2 (Coefficient of Determination)**:It calculates the extent to which the predictions of regression are close to the real data. R^2 that is closer to 1 represents improved model performance.
- **MAE (Mean Absolute Error):** : The average of the absolute value of the differences between predicted and actual values. Lower MAE is a sign of higher predictive accuracy.
- **RMSE (Root Mean Squared Error)**:Comparable to MAE but places more emphasis on larger errors. A lower RMSE is an indication of more regular predictions.

These steps allow us to evaluate and contrast models effectively and therefore select the one that is optimally balanced. We must also consider that because we want to

maximize reception time, only SMS messages actually received are considered when we evaluate the model. This ensures that it gets trained and validated upon significant and actionable data, which provides more reliable and realistic performance estimates.

3.4.1 Results for TSO Algorithm

In this section we are going to show the result of the Time sent Optimization Algorithm applied on label 2 of the models we have cited before taking into account the metric described before. The results on the test set of the dataset are the following:

Table 3.2: Model Performance Comparison for TSO Algorithms

Algorithm	**R^2 Score**	**MAE**	**RMSE**
Linear Regression	0.06	1.31	2.65
Random Forest	0.86	0.22	0.89
XGBoost	0.69	0.58	1.44
Neural Network	0.19	0.65	2.46

After we tried all the previous algorithms on our data here's what we observed:

- **Linear Regression** did poorly on all of the measures, especially with a very low R^2 measure of 0.06 and a high RMSE of 2.65. This confirms that it fails to represent the intricate non-linear relationships of the data.
- **Neural Network** characterized by low R^2 (0.19) and comparatively high RMSE (2.46), demonstrating inconsistent prediction quality and larger spread of errors. This is brought about by too small a dataset or model complexity not being adequately set.
- **XGBoost** good R^2 score (0.69), but its MAE (0.58) and RMSE (1.44) were higher than Random Forest and therefore it is very slightly less dependable.
- **Random Forest** highest R^2 value (0.86), and lowest error rates (MAE = 0.22, RMSE = 0.89). Random Forest handled the non-linear pattern of the data quite nicely and had good generalization abilities on the test set.

3.5 Conclusion

The present chapter describes a machine learning method that enhances SMS campaign decision-making processes. Our research approach combined supervised and unsupervised learning techniques to discover the best message sending times. The results enabled us to predict at what time should each contat receive the SMS according to its commuication . The evaluation of models included MAE together with RMSE and R^2 scores which showed positive performance results. The implementation of predictive tools enables SMS deliverability methods to operate both effectively and data-driven.

Chapter 4

Deployment and Interface Development

4.1 Introduction

This chapter shows the key elements that define how our system was structured. We begin by specifying the project's requirements both functional and non-functional to ensure the solution meets real-world needs. Then, we present few essential diagrams that help visualize how the system works. Finally, we introduce the core features of the application, offering a clear view of the tools and functionalities made available to users and administrators. All of these components work together to lay the groundwork for an efficient and intelligent SMS campaign management platform.

4.2 Model Deployment and User Interface Development

4.2.1 Requirements Specification

This section presents a full range of requirements to ensure our prediction tool is both effective and reliable. Here is a list of functional requirements:

- **Optimal Send Time Prediction :** The system must identify the most suitable time slots for each contact, those when they are most likely to receive and open their SMS messages.
- **Campaign Performance Enhancement :** The tool must improve overall SMS campaign performance by ensuring messages are sent to the right people at the right time.
- **Real-Time and Shared Application Access:** The system should provide a centralized platform enabling authorized users to access and interact with real-time dashboards. These dashboards present communication analytics, SMS performance metrics, and comprehensive contact information to support data-driven decision-making and efficient collaboration. The platform also includes a dedicated interface that allows users to predict the optimal time to send SMS messages to each contact, enhancing delivery effectiveness. Additionally, it should offer user profile management features such as updating their credentials, and securely reset their passwords.

The system's non-functional requirements are as follows:

- **Security:** The system must effectively protect user and advertising campaign data, as well as prediction results.
- **Usability:** The system must be intuitive and user-friendly. Users shouldn't need a manual to figure out how to use it—everything should feel natural. The interface needs to be clean, responsive, and accessible from any device, whether desktop or mobile. If users can navigate and complete tasks easily, they'll adopt the platform more quickly and use it more efficiently, which directly supports our marketing goals.
- **Scalability:** The system must be robust. It must easily handle a large volume of data and efficiently process multiple analysis requests simultaneously.
- **Performance:** The system must respond quickly, very quickly, even. Especially when it comes to generating predictions like the best sending time or ideal segmentation. This way, we can make decisions quickly and be responsive. After all, that's essential in marketing.
- **Maintainability:** The application must be easy to update and improve. Whether it's fixing bugs, adding new features, or adapting to changes in business needs, the codebase should remain clean, modular, and well-documented. This allows developers to work faster and more confidently, without breaking existing functionalities. Since the project may evolve over time, maintainability ensures we can build on solid foundations without having to start from scratch.
- **Adaptability:** The system must be flexible and scalable over time. Behaviors change, operators change too, and marketing strategies never stop evolving. Therefore, the tool must be able to integrate new data and improve its artificial intelligence models regularly. There's no choice but to adapt!

In conclusion, it is necessary to consider both functional and non-functional needs in the development of a reliable and effective prediction tool. By combining these two types of requierments, we can be sure that this system will offer a complete and user friendly solution.

4.2.2 UML Diagrams of the Application

4.2.2.1 Use Case Diagram

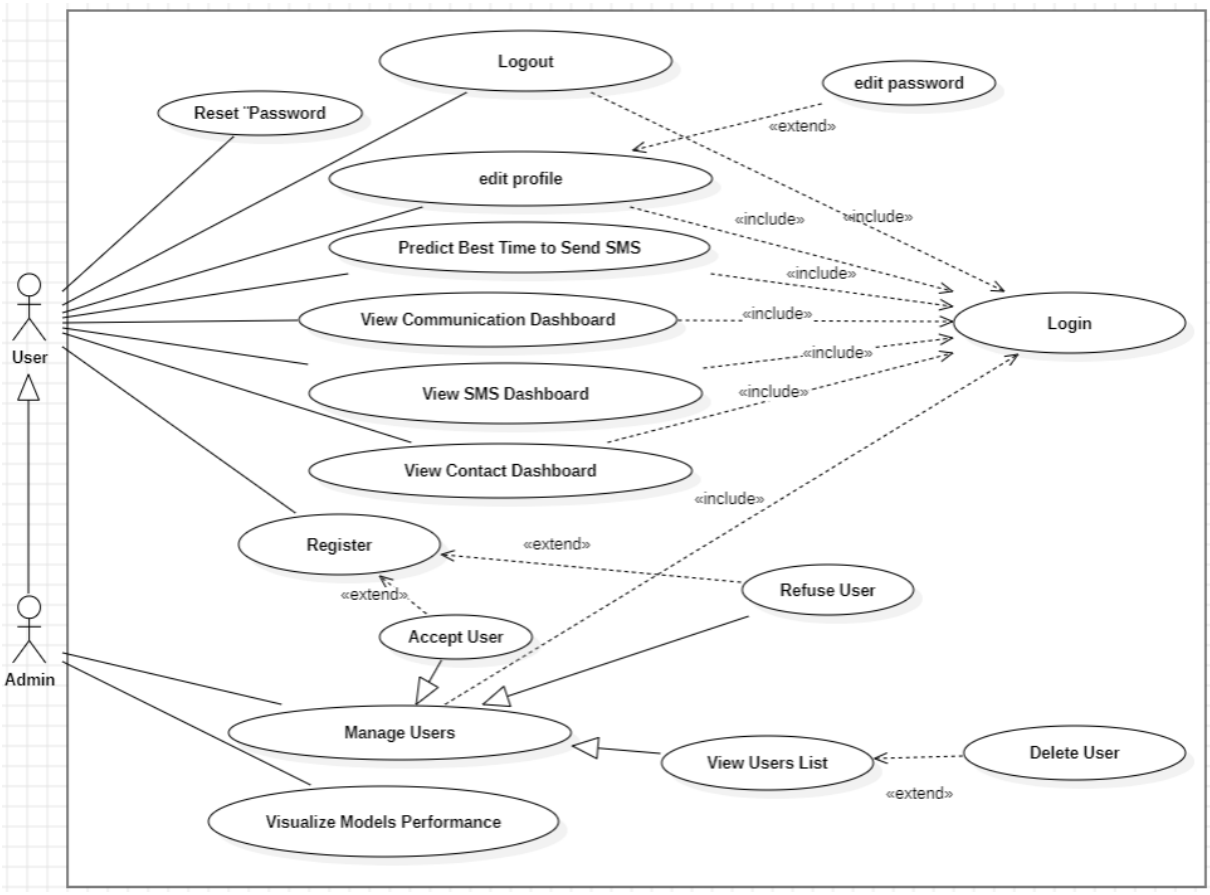

Figure 4.1: Use Case Diagram

- **Use Case Diagram Description :**

  This Use Case Diagram models the interactions between two system actors: User and Admin.

- The **User** can log in, edit their profile, reset or change their password, predict the best time to send SMS, and view three dashboards: SMS, Communication, and Contact.

- The **Admin** inherits all User rights and has additional privileges: managing user registrations (accept/refuse), viewing the list of users, deleting users, and visualizing the performance of predictive models.

  Note: Although the Admin cannot register for themselves, as their credentials are predefined in the system's database, they have the ability to register (add) new users to the platform and manage their access.

The diagram uses **«include»** relationships to indicate **mandatory** actions:

- Features such as **Predict Best Time to Send SMS** and all **Dashboard views** include **Login**, meaning users must be authenticated before accessing them.
- **Edit Profile** also includes Login, ensuring secure data management.

It also uses **«extend»** relationships to show optional or conditional actions:

- **Edit Password** extends **Edit Profile**, since changing the password is optional when editing the profile.
- **Accept User** and **Refuse User** extend **Register**, as Admin approval is conditionally required after a user registers.
- **Delete User** extends **View Users List**, triggered only when an Admin chooses to remove a user after listing them.

Overall, this diagram highlights role-based access control, required authentication, and optional flows for enhanced system flexibility and security.

4.2.2.2 Class Diagram

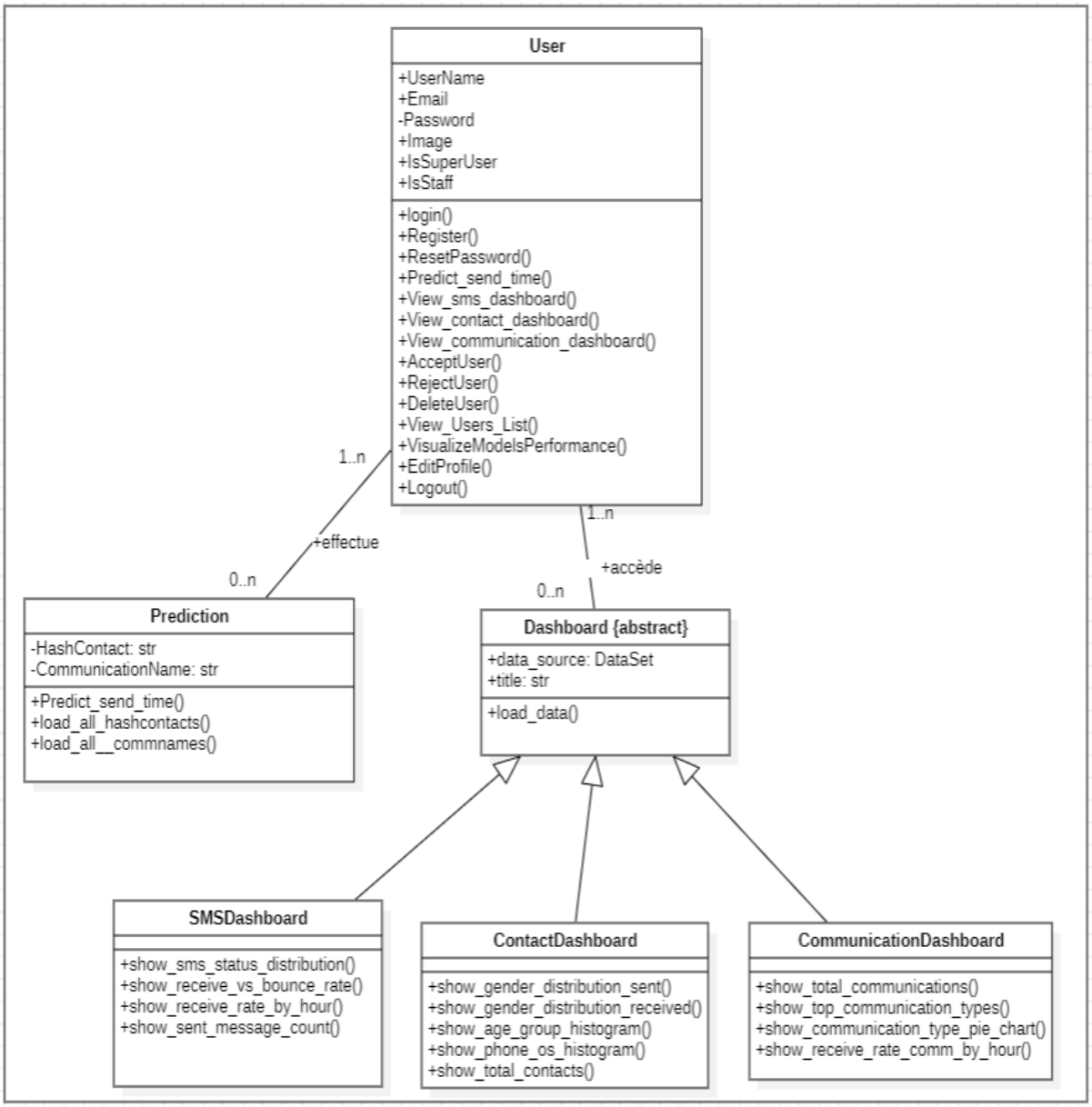

Figure 4.2: Class Diagram

This class diagram illustrates the core composition of the SMS Communication System, highlighting user roles, prediction functionality, and dashboard access.

- **User:** The User class represents a system user and includes attributes like name, email, password, image, isSuperUser which is a Boolean attribute that grants full system permissions and isStaff which is Boolean attribute that allows a user to access the staff interface with limited permissions.

  Users can depending to their role :

  * Log in
  * Register
  * Reset their password

    - view users list
    - Delete user
    - Accept user
    - Reject user
    - Delete user
    - VisualizeModelsPerformance
    - Launch predictions for optimal SMS sending times
    - Access multiple dashboards (SMS, Contact, Communication)
    - Edit Profile
    - Logout

  Each user can create several predictions and view multiple dashboards.

- **Prediction :** This class enables users to generate predictions based on a contact hash and communication name. It includes the method `Predict_send_time()` to determine the best time to send SMS.
- **Dashboard (Abstract):** A generic class representing a visual analytics panel. It is extended by three specific dashboards and includes:
    - `data_source` and `title` attributes
    - A method to load data
- **SMSDashboard:** Displays SMS-specific metrics and insights such as delivery status, bounce rate, send counts, and hourly reception rates.
- **ContactDashboard:** Focuses on demographic insights like gender, age group, and phone OS distributions.
- **CommunicationDashboard:** Shows aggregated communication statistics including total interactions, top types, and type distribution charts.

This design applies object-oriented principles such as inheritance, abstraction, and modularity to ensure maintainability.

4.2.2.3 Sequence Diagram

Login

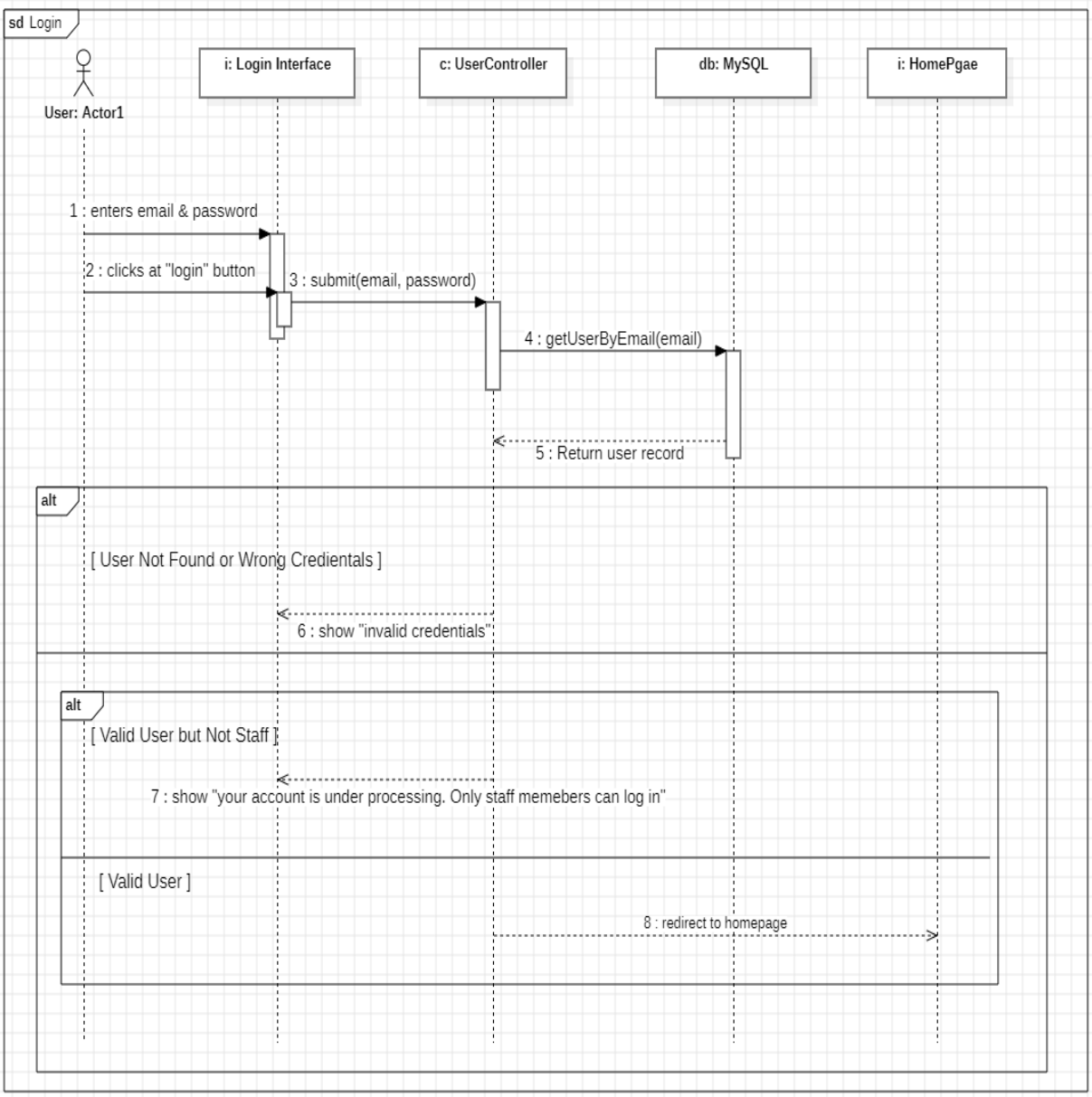

Figure 4.3: Login Sequence Diagram

This sequence diagram illustrates the authentication process followed when a user (either an admin or a standard user) attempts to log into the system.

Actors and Components

– **User (Actor)**: The primary actor who initiates the process by entering login credentials (email and password) to access the platform.

- **Login Interface**: The graphical user interface (UI) that allows the user to input login information. It also displays messages indicating whether the login attempt was successful or failed or if the credentials are invalid.
- **UserController**: The system component responsible for handling login logic. It receives login requests from the interface, validates the credentials by querying the database, and determines whether to grant or deny access.
- **MySQL (Database)**: Stores all user-related information, including email addresses, passwords, user roles (admin or staff). It is queried during the login process for credential verification.
- **HomePage**: The landing page to which the user is redirected upon successful authentication.

Process Flow

1. **User Input**: The user initiates the process by entering their email and password in the login form presented on the interface (UI), then clicks the `"Login"` button.
2. **Request Submission**: The UI sends the login credentials to the backend Controller, which handles authentication logic.
3. **Authentication Check**: The Controller interacts with the database (MySQL) to verify whether a user record exists with the provided email and password.
4. **Conditional Response**:
   - If the credentials are incorrect (email or password don't match any user in the database): The Controller returns an error indicating invalid credentials. The UI displays a standard login failure message.
   - If the credentials are correct but the user does not have staff privileges: The Controller rejects the login attempt. The UI triggers a pop-up message: `"Your account is under processing,only staff members can log in"`
   - If the credentials are correct and the user has staff privileges: The Controller confirms authentication success and user role. It sends a positive response to the UI, which then redirects the user to his Home Page.
5. **Outcome**: The user either gains access to the platform or is informed that their account is not yet authorized to log in.

Reset Password

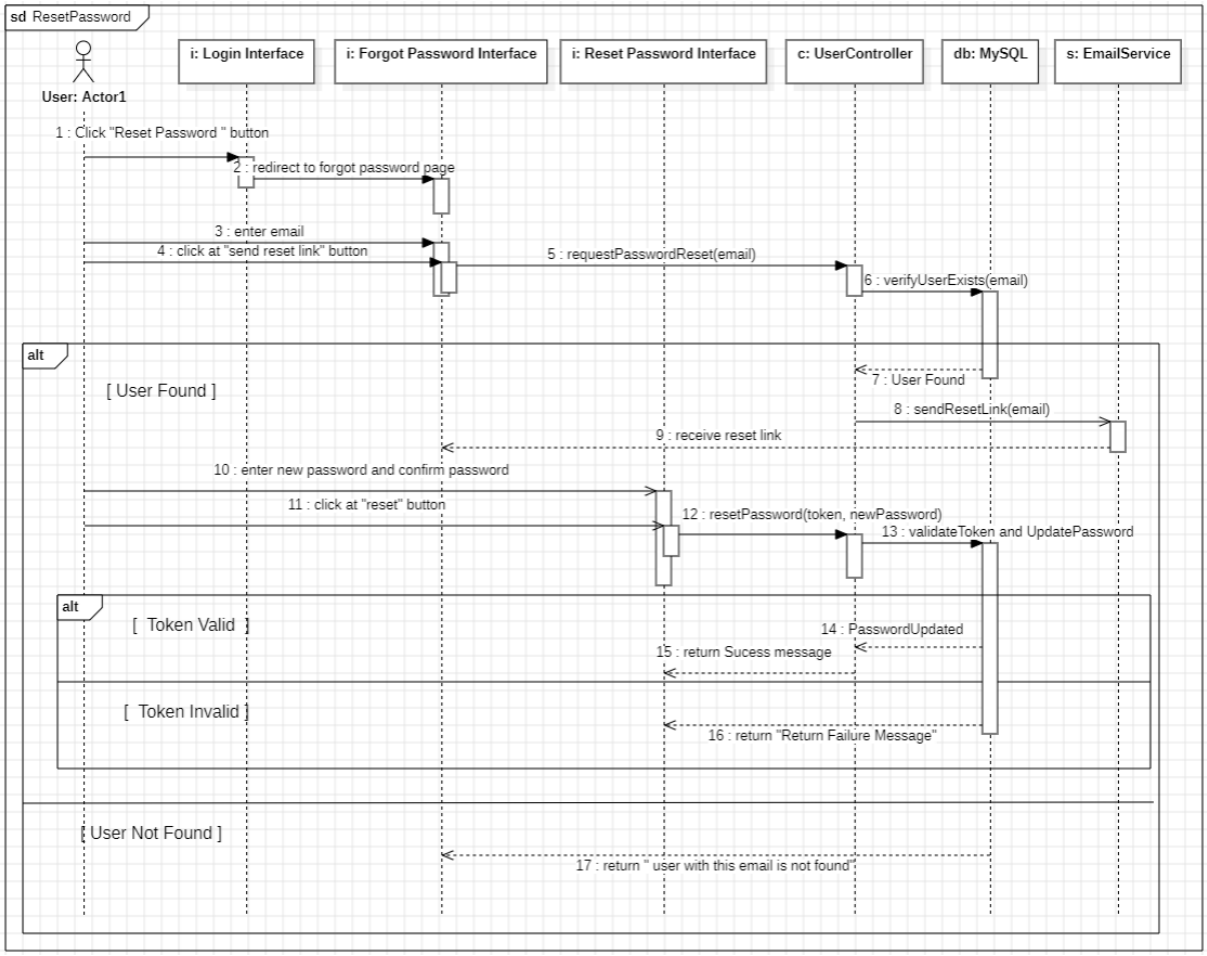

Figure 4.4: Reset Password Sequence Diagram

This sequence diagram illustrates the interaction between the user and system components during the password reset process.

Actors and Components:

- **User**: The end-user requesting a password reset.
- **Login Interface, Forgot Password Interface, Reset Password Interface**: Front-end interfaces.
- **UserController**: Handles logic and validation.
- **MySQL (Database)**: Stores user credentials.
- **EmailService**: Sends reset emails.

Process Flow:

1. **Initiation:** via "Reset Password" button.
2. **Redirection to Forgot Password Interface:**The system redirects the user to the Forgot Password Interface.
3. **Email Submission:** The user enters their email address.
4. **Send Reset Link Request:**The user clicks the “Send Reset Link” button, triggering a backend request: requestPasswordReset(email).
5. **Email Verification:** The UserController calls verifyUserExists(email) in the Database.
   - User Found: If the email exists, the system proceeds to steps 6-> 11.
   - User Not Found: If email does not exist, the Forgot Password Page triggers a pop-up message "user with this email is not found"
6. **Send Reset Link :** The UserController calls sendResetLink(email) via the EmailService.
7. **Email Delivery :** The User receives the reset link through their EmailClient.
8. **Reset Link Activation :** The user clicks the reset link, which opens the Reset Password Interface.
9. **Password Entry:** The user enters and confirms a new password.
10. **Reset Request Submission:** Clicking the "Reset" button triggers the call : resetPassword(token, newPassword).
11. **Token Validation and Password Update :** The UserController validates the reset token and updates the password in the Database: validateToken and UpdatePassword.
    - Confirmation: Once the token is valid and the update is successful, the system returns a success message to the user.
    - Failure: If the Token is invalid, the system returns "password couldn’t be changed".

Predict Best Sent Time

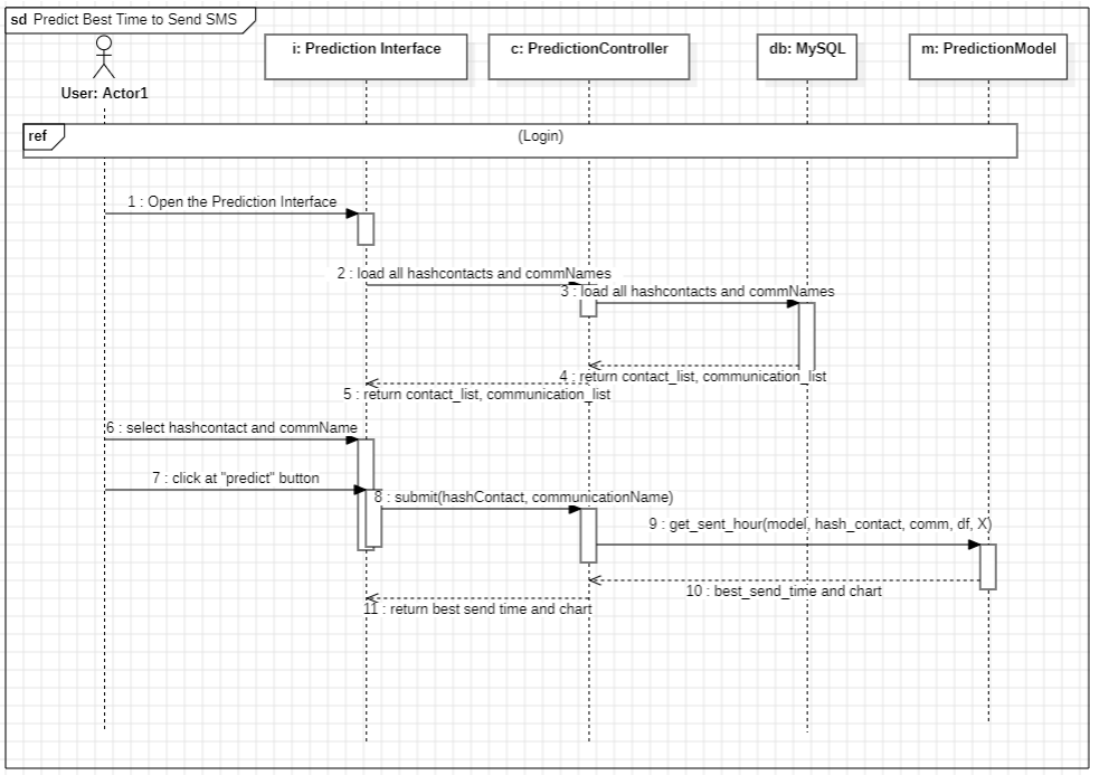

Figure 4.5: Predict Best Time to Send SMS Sequence Diagram

This sequence diagram illustrates the complete interaction between the User, the system's Prediction Interface, the Prediction Controller, the Database, and the Prediction Model to determine the optimal time to send an SMS based on user-selected hashcontact and communication type.

Actors and Components:

- **User (Actor)**: Initiates the prediction process.
- **Prediction Interface**: UI allowing the user to log in, input data, and view prediction results.
- **Prediction Controller**: Core logic unit managing user requests, database access, and model interaction.
- **MySQL (database)**: Stores contact information, communication data, and extracted features.
- **Prediction Model**: Processes features to calculate the optimal SMS send time using ML.

Process Flow:

1. **User Login**: The user must be logged into the system (represented as a reference fragment labeled Login).

2. **Form Access & Data Loading:**
   - The user opens the prediction interface.
   - The Prediction Interface sends a request to the Prediction Controller to load and retrieve HashContacts and CommunicationNames.
   - The Prediction Controller queries the Database, which returns the relevant data.
   - The data is returned and displayed in dropdown menus.
3. **User Selection:** The user selects a specific contact (HashContact) and communication type.
4. **Prediction Request Submission:** The user clicks the "Predict" button to initiate the prediction.
5. **Best Time Prediction:**
   - The Prediction Interface sends the selected values to the Prediction Controller.
   - The Prediction Controller calls the function `get_sent_hour(model, hashcontact, commname, df, x)` from the prediction model. .
6. **Displaying the Prediction:** The model returns the best send time and a chart, which are then passed by the prediction controller to the Prediction Interface.

4.2.3 Interface Features : Prediction and Dashboarding

In this section, we present the user interface designed to support and visualize the outcomes of our SMS deliverability optimization system. The interface was developed with usability, clarity, and practical insights in mind, enabling both technical and non-technical users to easily access core functionalities such as predictions, campaign analysis, and contact segmentation.

Each functionality is detailed below, accompanied by interface screenshot and explanation.

User Authentication: Login and Registration :

→ **Login :**

The application kicks off with a safe login setup to make sure that only authorized individuals can get to the platform's features. When landing on the main page, both admins and users are directed to a login interface where they are required to type their email and password, then hit the "Login" button. Upon successful authentication, the user gets access to dashboards, prediction tools, and other parts of the system. This check layer acts as a first security shield protecting important information, data and functions from unapproved and unauthorized access.

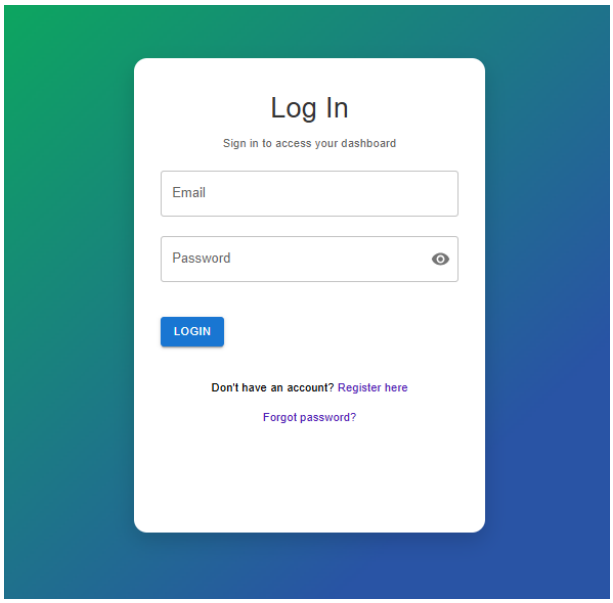

Figure 4.6: Login Page

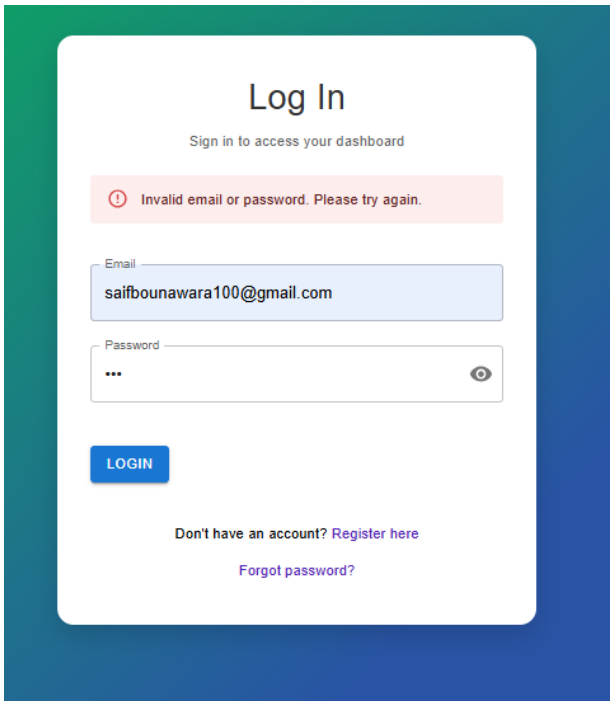

Figure 4.7: Wrong Credentials

For security and scalability purposes, we implemented a token based authentication system using Django Knox. Users receive a unique token after successful authentication that must be presented in subsequent API requests. This approach enables stateless communication, improved security, and simpler management of user sessions.

Any user who signs up cannot use the system immediately after signing up. One is only able to access the system after being specifically added to the staff

by a superuser. This is a controlled access feature to the platform and is one of the user management functions that are restricted to the superuser only, which are described in a later section.

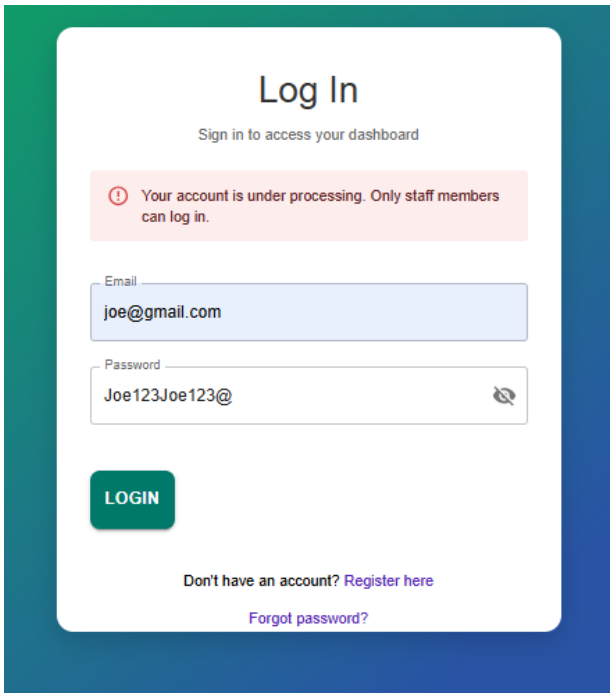

Figure 4.8: Login Validation : Access Denied for Non-Staff Accounts

→ **Registration Request :**

Since administrator credentials are already predefined in the database, administrators do not register through the interface. However, users who don't have an account in the system can request registration. Upon clicking the "Register here" button, users get redirected to a form where they are required to input their email address, username, password, to confirm password and to choose a profile image to upload. Once a user submits a registration form by clicking on "Register" button, then registration request gets sent to the administrator. The admin reviews the requests and either approves or rejects them; therefore maintaining full control over who gets access to the system.

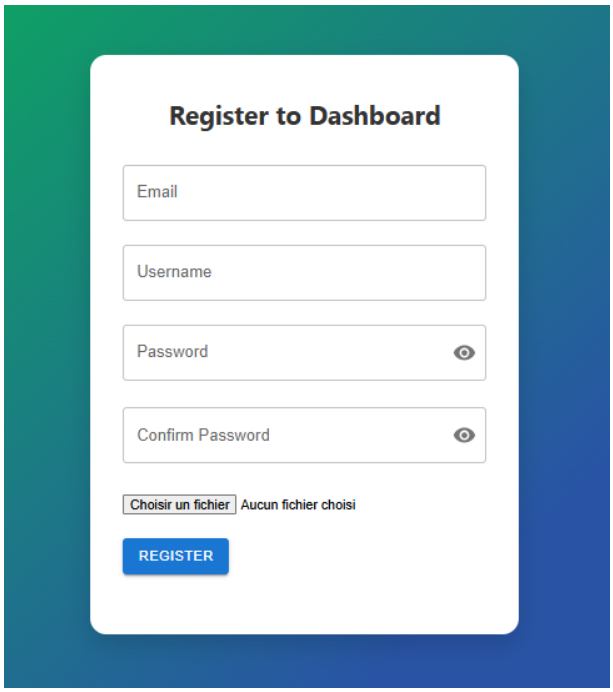

Figure 4.9: Register Page

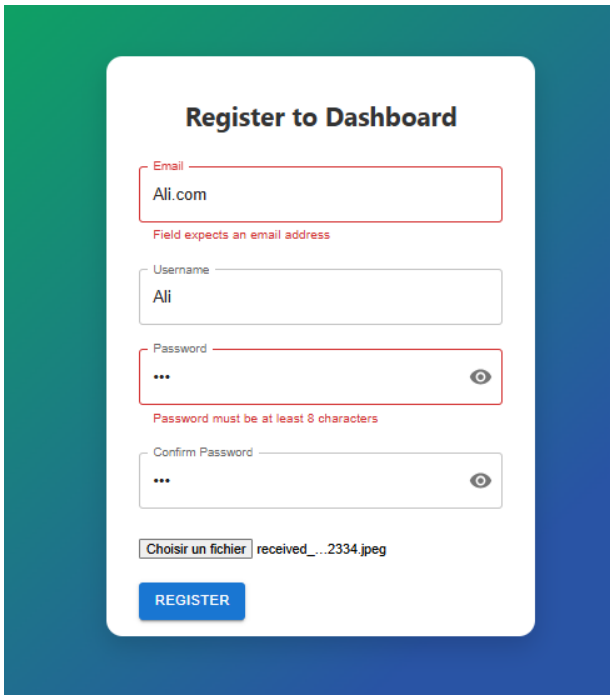

Figure 4.10: Register Page if invalid fields

Access rights are tightly managed within the system. Admins alone can approve new user registrations and other users' access rights. This provides complete control over who gets access and interact with the system, making it very apt for team-based settings where roles and responsibilities are varied.

→ **Password Reset:**

In the event of a user or admin forgetting a password, the system provides a "Reset Password" directly on the login page. On clicking the button, the system generates an email to the registered email address associated with the account. The email contains a new password that can be used to restore access. This password recovery provides an easy and safe way for users to recover their accounts without compromising on the security of the platform.

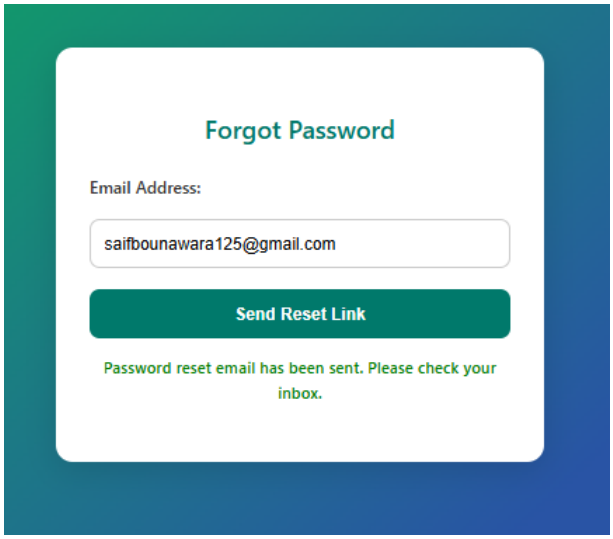

Figure 4.11: Forget Password

Figure 4.12: Resetting Password Email Sent to User

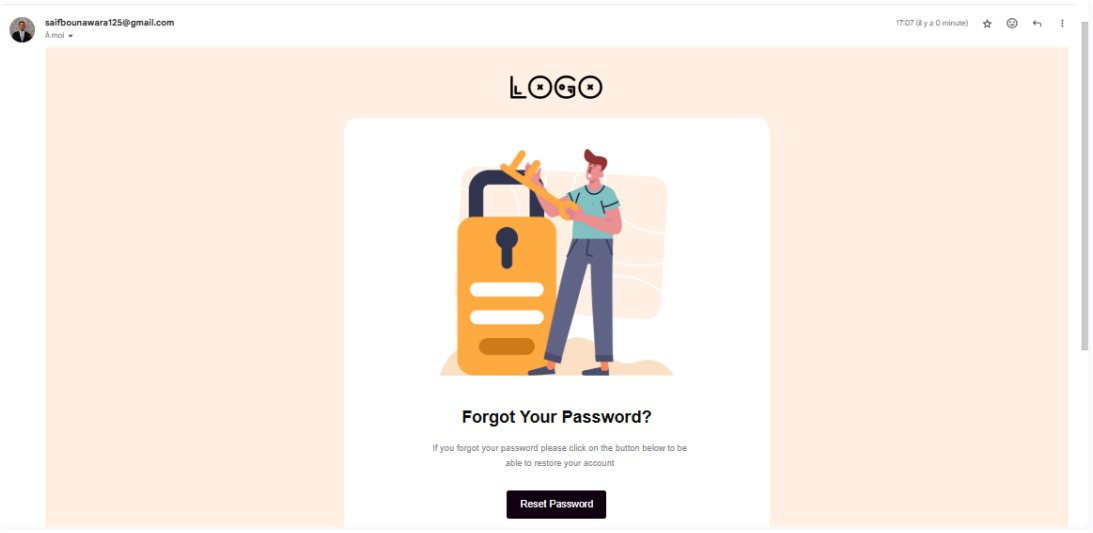

Figure 4.13: Reset Password Page in Gmail

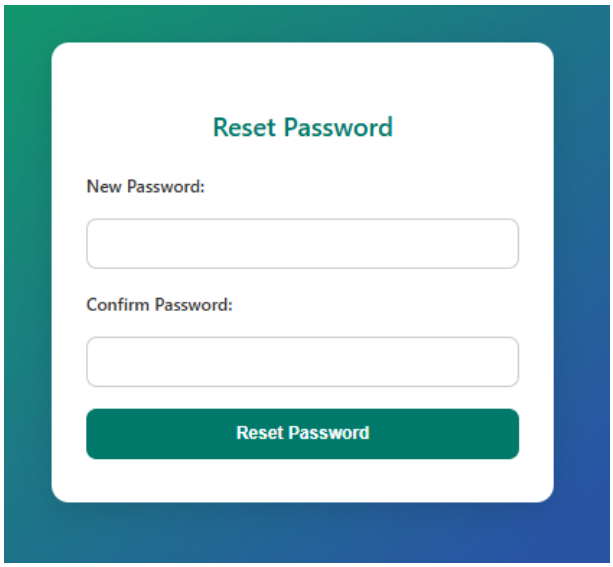

Figure 4.14: Reset Password Page

By combining this method for authentication, it allows for user access that is traceable and manageable as well as secure. This authentication method sets the stage for adding additional role-based access control in future releases of the application which is useful in a collaborative space where the use of the system falls upon multiple users within a team monitoring SMS campaigns, providing predictions, or managing user directories.

→ **User and Admin Home Pages :**

Each one, User and Admin, has a specific home page where they can access features according to their role: the User can view dashboards, use prediction tools, and edit their profile, while the Admin has additional access to user management functionalities

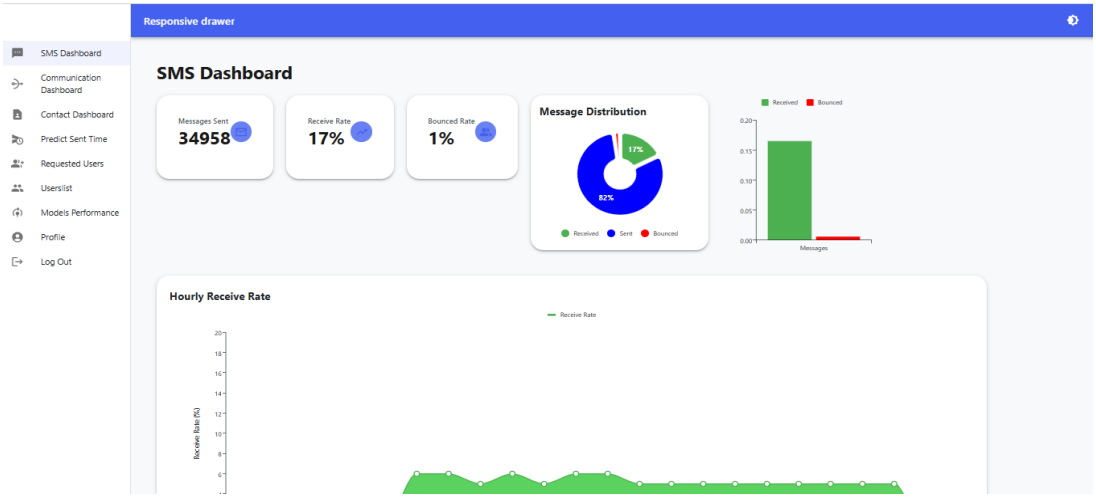

Figure 4.15: Admin HomePage

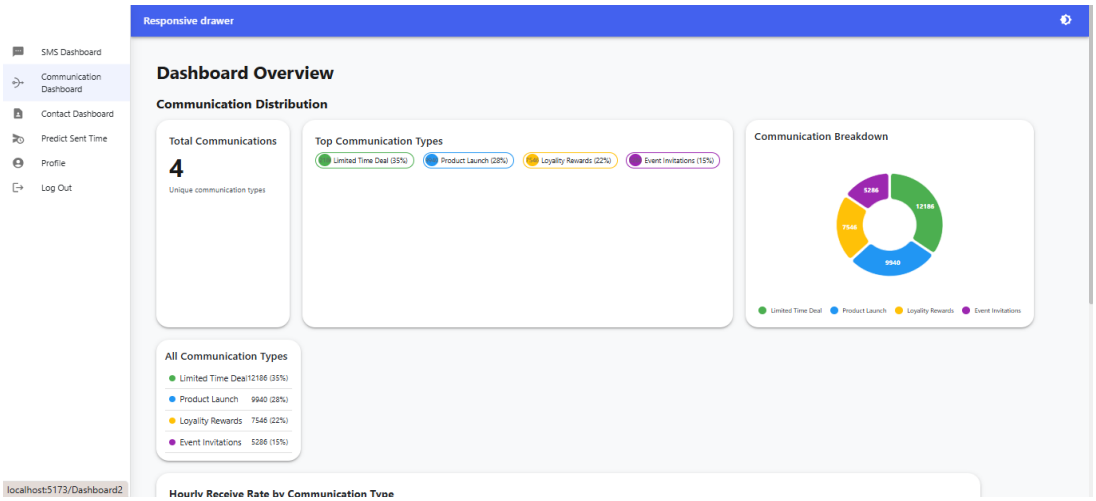

Figure 4.16: User HomePage

→ Edit Profile

Both Users and Admins have the right to edit their profile through a dedicated interface. They can not only update their username, change their email address and upload a new profile image from their desktop but also edit their password by resetting it. These options mentionned here allow each user or super user (admin) to manage and personalize their account easily while ensuring data remains secure and up to date.

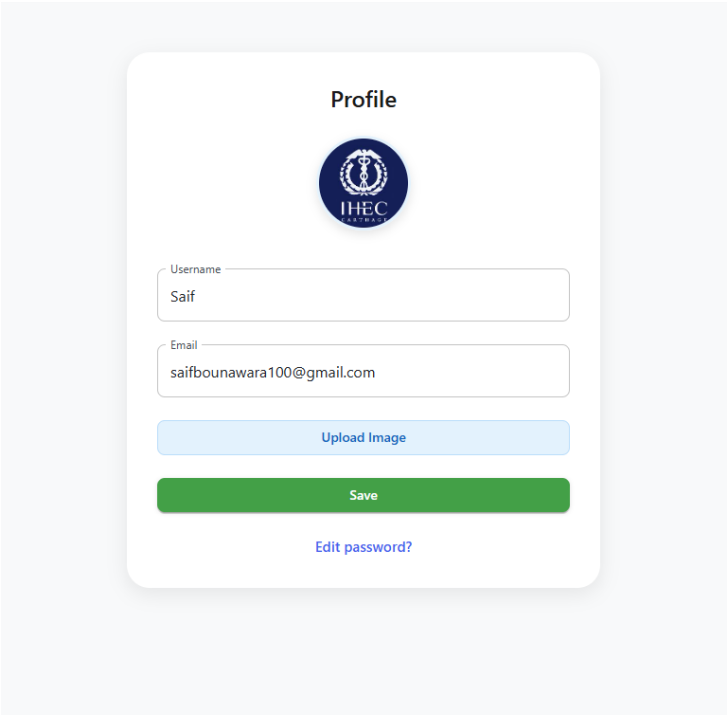

Figure 4.17: Edit Profile Page

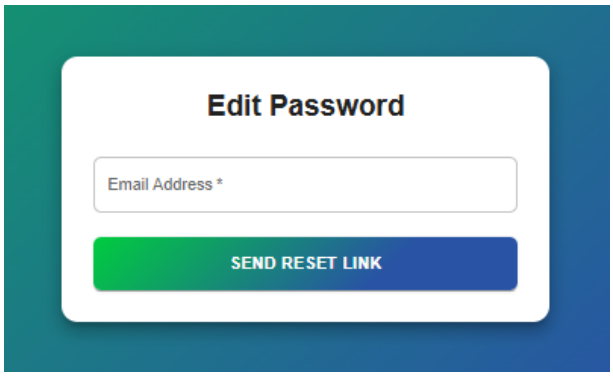

Figure 4.18: Edit Password Page

→ **SMS Dashboard :**

The SMS Dashboard makes it easy to view SMS campaign statistics at a high level. Users can watch performance metrics such as deliverability and engagement stats that are critical to analyze the effectiveness of messaging strategies and to quickly identify possible deliverability problems or behavioral trends. The SMS Dashboard includes:

- → A pie chart illustrating the distribution of SMS statuses (sent, received, and bounced). This provides an easy and clear view of message outcomes across all campaigns at a glance.
- → A histogram illustrating the percentage rates of received vs. bounced SMS. This aids in identifying inconsistencies and performance bottlenecks.
- → A line or bar chart illustrating the receive rate per hour of day. This will show the best times to send SMS based on actual engagements of users.
- → A numerical summary of the total number of messages sent, total received rate, and total bounced rate for a quick operational snapshot.

We encourage the reader to leverage these visual data insights to make data-driven decisions by identifying delivery trends and underperforming time slots. Marketers will be able to change their strategies on the fly in search of some sweet spot for maximizing reach and response.

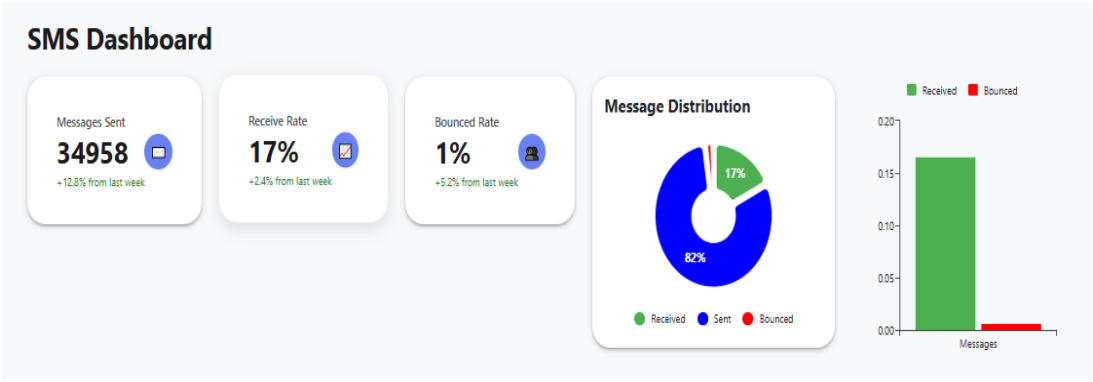

Figure 4.19: A preview of the SMS Dashboard (1)

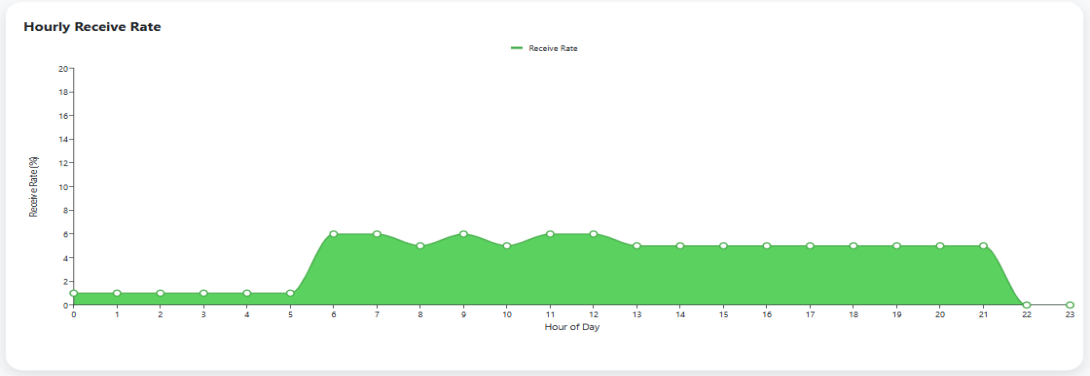

Figure 4.20: A preview of the SMS Dashboard (2)

→ **Contact Dashboard :**

The Contact Dashboard offers comprehensive demographic-based insights that furnish users with a greater understanding of the people in their audience and how to later refine targeting. This dashboard offers several statistical visualizations focused on being effective for segmentation and personalization in campaign design. In summary, this dashboard includes :

- → Pie chart for the distribution of sent messages by gender, which shows the breakdown of whether the design of the messages is reaching proportionally to males, females, and other segments of demographic groups.
- → Pie chart for the distribution of received messages by gender, which may indicate differences in engagement rates for gendered groups.
- → Summary card for total messages sent, which offers a general sense of volume of communication.
- → Histogram breaking down message statistics by age group, which allows you to see which age groups are being more engaged or perhaps more actively targeted.
- → Histogram for message statistics by phone operating system (OS), which allows you to understand the technological basis of your recipients and allows you to adjust your messages based on the platform your recipients are using.

Taken all together, these visualizations allow marketing teams to design message content, timing, and channels according to the demographic and technical profile of their audience, and hence improve campaign effectiveness.

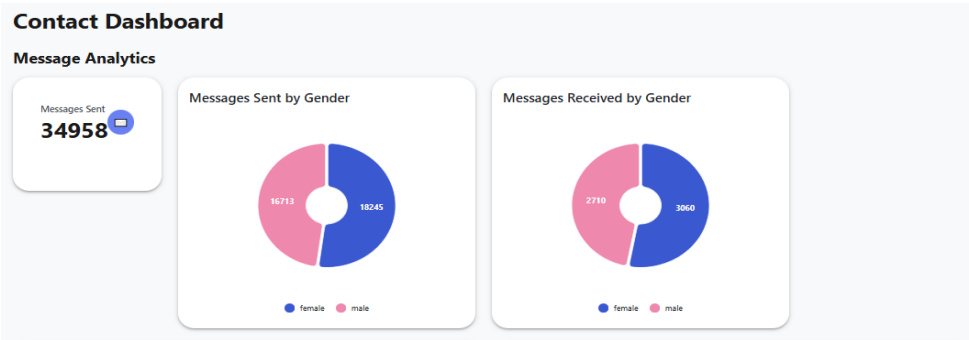

Figure 4.21: A preview of the Contact Dashboard (1)

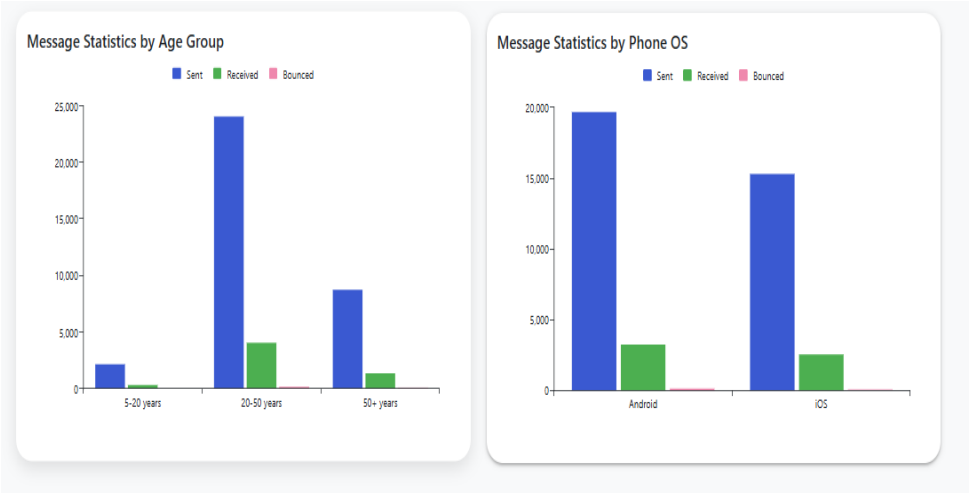

Figure 4.22: A preview of the Contact Dashboard (2)

→ **Communication Dashboard :**

This dashboard presents important communication data, enabling meaningful exploration into how different types of communication perform across campaigns. Overall, there are main statistics that are included in this dashboard:

- → Communications – The total number of communications as an overall view of messaging activity.
- → Top 4 Communications Types, including rates, for the comparative benefits of that communication across the company.
- → A pie chart that outlines the communication names, demonstrating the percentage of frequency and amount used in previous campaigns.

Each statistic and visualization is valuable in learning users' preferences, but they moreover highlight key engagements by showing the categories of the communications that had more successful strategies. Marketing teams are assisted in recognizing their content, what formats they were using to deliver messages, and the next best approach they could take toward maximizing the impact of their future campaigns.

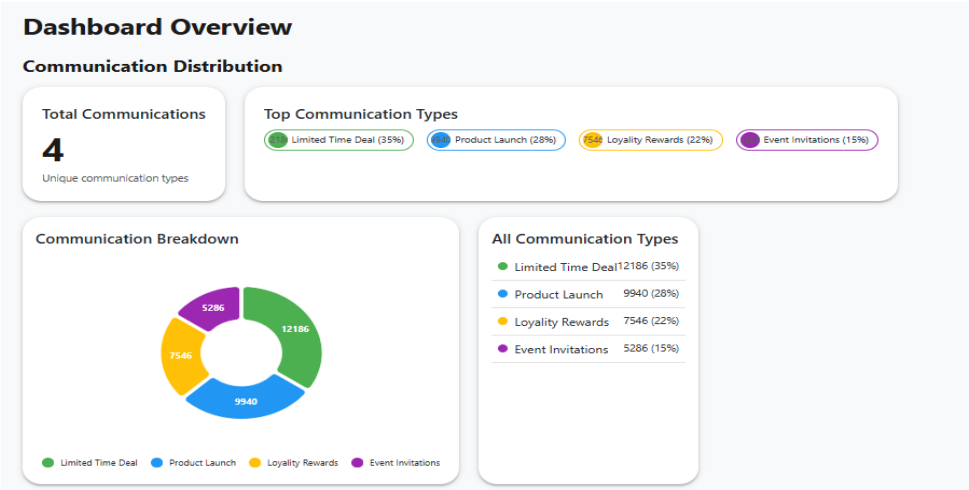

Figure 4.23: A preview of the Communication Dashboard

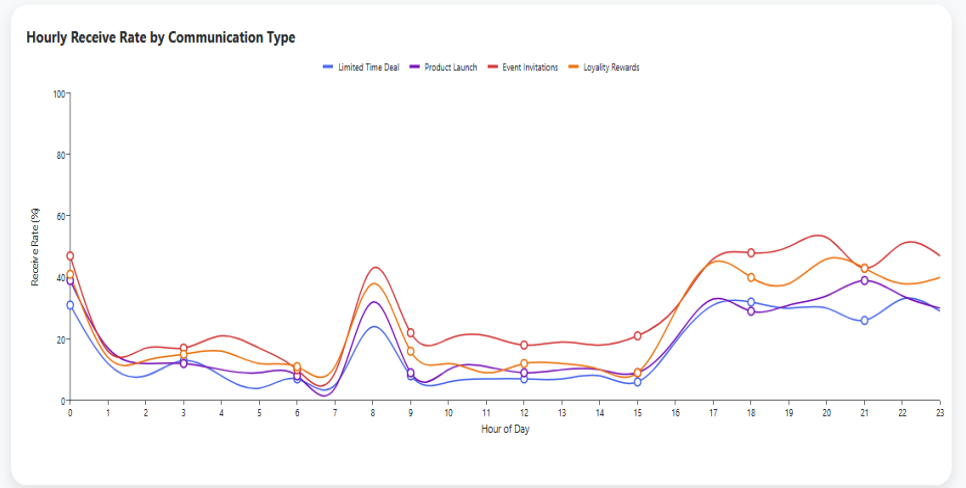

Figure 4.24: A preview of the Communication Dashboard (2)

→ **Predict Sent Time :**

The Prediction module is a significant feature of the application. The process begins with the user entering both the HashContact and the Communication-Name. Then, after clicking on the predict button, the application returns the optimal time slot to deliver the SMS to this contact. This functionality is driven by the deployed ML model and helps marketers maximize the deliverability and engagement of their SMS campaigns by targeting their users at the correct time.

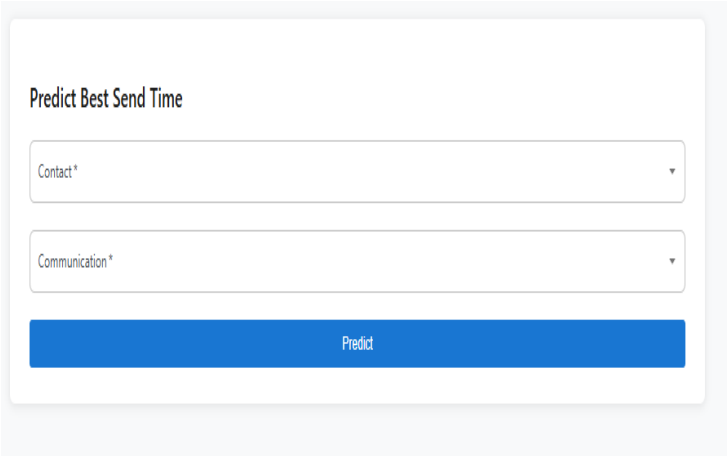

Figure 4.25: A preview of the Best Send Time Prediction Page

Requested Users Directory: Accept and Reject Staff :

Administrators have access to a management section where they can:

→ Accept requests sent and add new users or staff members who will interact with the system.

→ Reject users requests from the directory.

This ensures secure role management and interface governance.

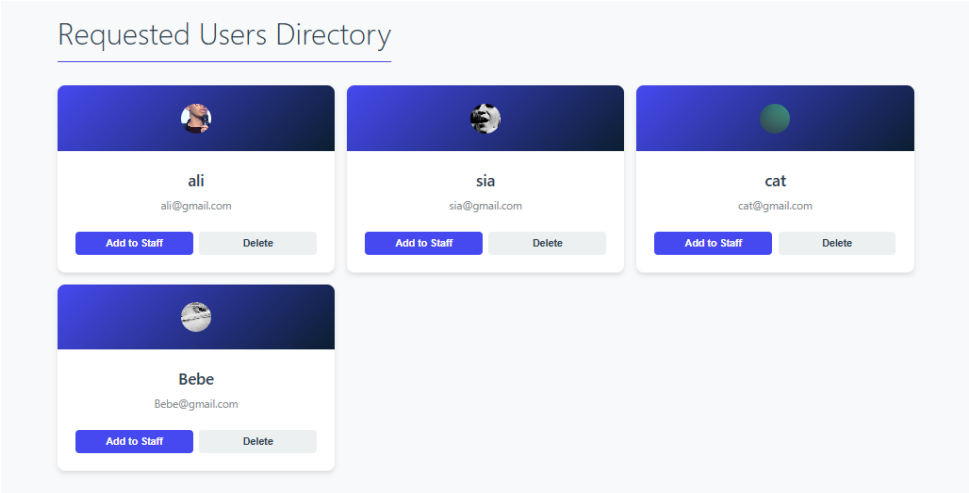

Figure 4.26: A preview of the Requested Users Directory

→ **Users List**

This page displays a list of all users with access to the system so the administrator can monitor and manage the use of the platform. The information of each user, such as email and username, is shown for auditing. The administrator is also capable of deleting users from the system if necessary, for example, as a result of inactivity or unauthorized use. This aids in ensuring that only legitimate users can interact with the system, maintaining security and control.

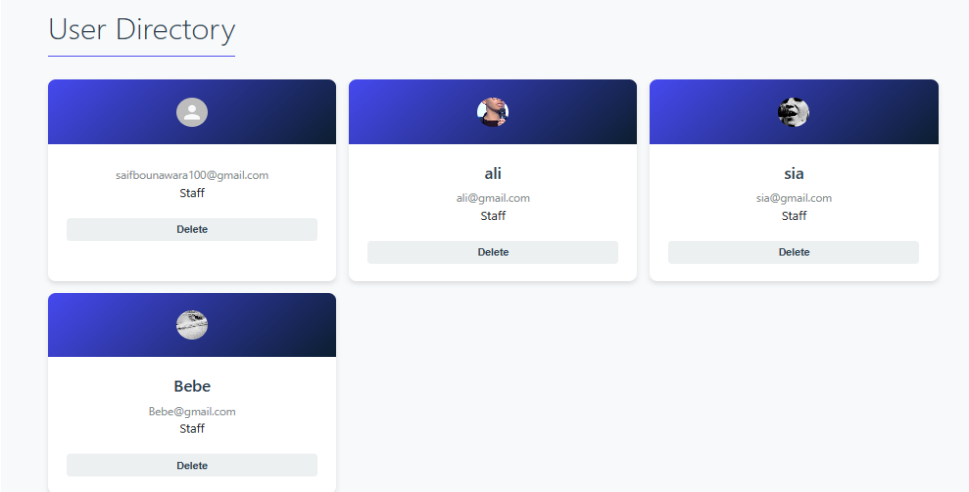

Figure 4.27: A preview of the Users List

→ **Visualize Performance of Models**

This module allows for model performance to be compared effortlessly and understanding of strengths and weaknesses of each algorithm in the framework of send time prediction. The output provides transparency and serves as a foundation for selecting the best performing model.

Model	MAE (minutes)	RMSE (minutes)	R^2 Score
random_forest	0.22	0.89	0.86
linear_regression	1.31	2.65	0.06
xgboost	0.58	1.44	0.69
neural_network	0.65	2.46	0.19

Figure 4.28: A preview of the Models Performance

→ **Log Out**

The Log Out feature enables the users to exit the application securely, their authentication token is invalidated once they log out, ending effectively the session and preventing further access to the system without reauthentication.

4.3 Conclusion

In summary, this chapter presented the requirements specification of our project, covering both functional and non-functional aspects. We also introduced the architecture of the user interface, supported by essential diagrams such as the detailed use case diagram to illustrate its structure and operation. Finally, we described the main features of the application, , which included explanations and images of functionality illustrating their role within the system. These details have given a solid foundation that can give a clear understanding of how the platform has been designed to support SMS campaign management and to do that as efficiently as possible, allowing the platform to be the incumbent if questions do arise.

Conclusion

In this report , we have presented various potential methods of solving the Time Sent Optimization problem . From a commercial perspective, possessing such a tool is important , even crucial for an organization or a company , as it enables companies to enhance and ameliorate the efficiency of their marketing campaigns since more users receive the company's communications and open it .

During this internship , we have found that this task itself is not particularly difficult . Nevertheless , unlike other domains , this project lacks existing literature and remains largely unexplored . Therefore we need to invent the algorithm completely from scratch and by ourselves .

In particular , it is crucial to comprehend the process of creating the dataset , determining the appropriate features to employ , understanding the structure of the label and finally selecting the which metric for the evaluation need to be used . our initial attempt aimed to address this issue within the realm of unsupervised learning. Yet , our dataset was too scattered. In other words , the data points are spread out and not concentrated in a clear pattern or structure which make it difficult to perform operations of the unsupervised learning algorithms like clustering. Thus we switched our focus to supervised learning. Still, moving to supervised learning doesn't seem like a straightforward step , as it is essential to understand the way our label will be structured, given that various options are available. After multiple tries, we ultimately decided to opt for a label which represents a histogram reflecting the users' fitness across all potential hours. Conversely our objective requires the to be related and dependent. Therefore we moved to solve a regression problem. We developed two distinct algorithms and tested several regression models which we have already presented in this thesis , finding out that the most effective one was RandomForest model.

Perspectives

This specific problem is highly reliant on data. In fact, an increased volume of real data enhances the outcomes. Therefore, as a key direction for future endeavors, it is essential to persist in collecting additional data from the users and retrain the model, followed by testing it using the metric introduced in this thesis.

Despite the fact that we reached a higher open rate compared to the baseline, the improvement in the engagement rate was minimal. This outcome arose because we concentrated mainly on improving the receive rate, assuming that an uptick in the receive rate would lead to a corresponding rise in the engage rate. This is true in part, but not in general.

Thus, in future work, we intend to focus directly on increasing the engage rate as well. This could be achieved by developing a model capable of generating compelling and personalized SMS subjects that are more likely to capture the user's attention. This represents a distinct challenge in the field of Natural Language Processing, different from the challenge of optimizing message timing. However, we believe that combining both solutions could lead to even better results, especially in terms of user engagement.

Furthermore, we plan to enhance targeting by applying a filtering process to our dataset. This will involve identifying and removing inactive contacts (e.g., unreachable or unresponsive numbers), allowing the system to focus solely on active users. This refinement is expected to dilute the noise within the data and reinforce the effectiveness of the proposed approach.

Bibliography

[1] Tritux Group, *Notre histoire*, https://www.tritux.com/notre-histoire/, Retrieved on March 18, 2025.

[2] About "Tritux Group – Core Areas of Expertise and Descriptions", https://www.tritux.com/notre-histoire/, Retrieved on 18/03/2025.

[3] About "Tritux Group – Solutions and Service Descriptions", https://www.tritux.com/notre-histoire/, Retrieved on 18/03/2025.

[4] About "EasybulkSMS", https://easybulksms.co.in/about.php, Retrieved on 19/03/2025.

[5] About "bulkSMS", https://www.bulksms.com/company/, Retrieved on 19/03/2025.

[6] About "Comparative Analysis of Existing SMS Platforms", https://www.bulksms.com/company/, https://easybulksms.co.in/about.php, Retrieved on 19/03/2025.

[7] About "KDD Methodology", https://www.datascience-pm.com/kdd-and-data-mining/, Retrieved on 19/03/2025.

[8] About "SEMMA Methodology", https://www.datascience-pm.com/semma/, Retrieved on 19/03/2025.

[9] About "CRISP-DM Methodology", https://www.datascience-pm.com/crisp-dm-2/, https://www.sv-europe.com/crisp-dm-methodology/, Retrieved on 20/03/2025.

[10] About "Anaconda", https://domino.ai/data-science-dictionary/anaconda, Retrieved on 16/05/2025.

[11] About "Jupyter", https://en.wikipedia.org/wiki/Project_Jupyter, Retrieved on 16/05/2025.

[12] About "Python", https://fr.wikipedia.org/wiki/Python_(langage), Retrieved on 16/05/2025.

[13] About "Visual Studio Code", https://fr.wikipedia.org/wiki/Python_(langage), Retrieved on 16/05/2025.

[14] About "Postman", https://www.postman.com/product/what-is-postman/, Retrieved on 16/05/2025.

[15] About "Django", https://fr.wikipedia.org/wiki/Django_(framework), Retrieved on 18/05/2025.

[16] About "React Js", https://fr.wikipedia.org/wiki/React, Retrieved on 18/05/2025.

[17] About "StarUML", https://staruml.fr.softonic.com/, Retrieved on 18/05/2025.

[18] About "LaTeX", https://fr.wikipedia.org/wiki/LaTeX, https://www.latex-project.org/, Retrieved on 21/05/2025.

[19] About "Google Meet", https://apps.apple.com/us/app/google-meet/id1096918571, Retrieved on 21/05/2025.

[20] About "Github", https://www.techtarget.com/searchitoperations/definition/GitHub, Retrieved on 21/05/2025.

[21] About "Artificial Intelligence", https://cloud.google.com/learn/what-is-artificial-intelligence, Retrieved on 03/04/2025.

[22] About "Data Science", https://datascientest.com/data-science-definition, Retrieved on 03/04/2025.

[23] About "Machine Learning", https://datascientest.com/machine-learning-tout-savoir, Retrieved on 03/04/2025.

[24] About "Supervised Learning", https://cloud.google.com/discover/what-is-supervised-learning, https://www.ibm.com/think/topics/supervised-learning, Retrieved on 06/04/2025.

[25] About "Unsupervised Learning", https://cloud.google.com/discover/what-is-unsupervised-learning, https://www.ibm.com/think/topics/unsupervised-learning, Retrieved on 06/04/2025.

[26] About "Clustering", https://www.geeksforgeeks.org/clustering-in-machine-learning/, https://www.ibm.com/think/topics/clustering, Retrieved on 25/04/2025.

[27] About "K-Means", https://www.ibm.com/think/topics/k-means-clustering, Retrieved on 25/04/2025.

[28] About "Hierarchical Clustering", https://www.ibm.com/think/topics/hierarchical-clustering, Retrieved on 26/04/2025.

[29] About "Regression", https://www.geeksforgeeks.org/regression-in-machine-learning/, Retrieved on 28/04/2025.

[30] About "Linear Regression", https://www.geeksforgeeks.org/ml-linear-regression/, https://www.ibm.com/fr-fr/think/topics/linear-regression, Retrieved on 28/04/2025.

[31] About "Random Forest", https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/, https://www.intelligence-artificielle-school.com/ecole/technologies/random-forest-en-machine-learning/, Retrieved on 02/05/2025.

[32] About "Neural Network", https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/, https://www.ibm.com/think/topics/neural-networks, Retrieved on 02/05/2025.

[33] About "XGBoost", https://www.ibm.com/fr-fr/think/topics/xgboost, Retrieved on 02/05/2025.

Abstract

This document is milestone work of a graduation project for a Bachelor of Science in Business Intelligence at the Institute of Higher Commercial Studies of Carthage **(IHEC Carthage)**. The graduation project has taken place in the company **Tritux Group**.

The main goal of our project was to create an intelligent system that increases the effectiveness of SMS marketing campaigns while improving the deliverability of SMS campaigns. With the help of ML techniques, one of our objectives was to create predictive models to predict the best time to send SMS for each contact. The platform uses predictive data derived from customer data (from MongoDB) and follows the CRISP-DM data science methodology. The final project also included a user-friendly web interface that allows the marketer to view the predictive dashboards and live prediction for effective and actionable marketing insight.

The document includes the outline of each phase of the project and technical and functional aspects of the solution we implemented.

Keywords: Python, MongoDB, CRISP-DM, API, machine learning, Django, data.

Résumé

Ce document constitue un travail d'étape d'un projet de fin d'études pour une licence en informatique décisionnelle à l'Institut des Hautes Études Commerciales de Carthage **(IHEC Carthage)**. Ce projet s'est déroulé au sein de l'entreprise **Tritux Group**.

L'objectif principal de notre projet était de créer un système intelligent permettant d'accroître l'efficacité des campagnes marketing par SMS tout en améliorant leur diffusion. Grâce aux techniques de Machine Learning, nous avons notamment cherché à créer des modèles prédictifs pour prédire le meilleur moment pour envoyer un SMS à chaque contact. La plateforme utilise des données prédictives issues des données clients (de MongoDB) et suit la méthodologie de science des données CRISP-DM. Le projet final comprenait également une interface web conviviale permettant aux marketeurs de consulter les tableaux de bord prédictifs et les prévisions en temps réel pour des analyses marketing efficaces et exploitables.

Ce document détaille chaque phase du projet ainsi que les aspects techniques et fonctionnels de la solution mise en œuvre.

Mots-clés : Python, MongoDB, CRISP-DM, API, ML, Django, Data.