

Dissertation/Report submitted to the
UNIVERSITY OF MOHAMED BOUDIAF - M'SILA, ALGERIA



FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

in partial fulfillment of the requirements for the degree of

Master's in Computer Science

Specialization: Computer Science

Option: ISSE

Presented by

SELLAMI IMAD EDDINE

Entitled

**Digital Marketing Strategies Using Artificial
Intelligence: Design of a Prospect Classification
Model**

Under the supervision of

Mrs.AHLAM Bouzaroura

Jury Members

Prof. ROUSSAFI Mahdjoubi

University of M'sila

President

Prof. KHEDIDJA Derdour

University of M'sila

Examiner

June, 2026

Abstract

This work presents the design and development of a hybrid intelligent system for prospect classification in digital marketing. The proposed approach combines unstructured textual data extracted from email content with structured behavioral data such as email opening, link clicking, and interaction frequency.

The system integrates a real-time tracking module that collects and transforms user interactions into meaningful features. These features are processed using a machine learning model based on neural networks, enhanced with sentiment analysis techniques to improve prediction accuracy.

A lead scoring mechanism is implemented to classify prospects into different levels of interest, enabling more efficient targeting and decision-making. In addition, a self-learning component is introduced to continuously improve the model by incorporating newly collected data.

Experimental results demonstrate strong predictive performance and stable training behavior, highlighting the system's effectiveness in real-world marketing scenarios.

Keywords: Artificial Intelligence, Machine Learning, Lead Scoring, Behavioral Analysis, Natural Language Processing, Digital Marketing.

الملخص

يقدم هذا العمل تصميم وتطوير نظام ذكي هجين لتصنيف العملاء المحتملين في مجال التسويق الرقمي، من خلال دمج البيانات النصية المستخرجة من محتوى رسائل البريد الإلكتروني مع البيانات السلوكية مثل فتح الرسائل، النقر على الروابط، وتكرار التفاعل.

يعتمد النظام على وحدة تتبع آلية تقوم بجمع وتحليل تفاعلات المستخدم وتحويلها إلى خصائص قابلة للمعالجة. يتم بعد ذلك استخدام نموذج تعلم آلي قائم على الشبكات العصبية، مدعوم بتقنيات تحليل المشاعر، من أجل التنبؤ بدرجة اهتمام المستخدم بدقة عالية.

يوفر النظام آلية تقييم تسمح بتصنيف العملاء إلى مستويات مختلفة حسب درجة الاهتمام، مما يساعد في تحسين استراتيجيات الاستهداف واتخاذ القرار. كما تم إدماج آلية تعلم ذاتي تساهم في تحسين أداء النموذج مع مرور الوقت اعتماداً على البيانات الجديدة.

أظهرت النتائج التجريبية فعالية النموذج في التنبؤ واستقراراً جيداً أثناء التدريب، مما يعكس إمكانية تطبيق هذا النظام في بيئات حقيقية لتحسين كفاءة الحملات التسويقية.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم الآلي، تصنيف العملاء المحتملين، تحليل السلوك، معالجة اللغة الطبيعية، التسويق الرقمي.

Dedication

“To my beloved parents, may Allah protect them, To those who have always been my source of strength, my guiding light, and my support in every step, To those who instilled in me faith and raised me with patience and determination—whose prayers were my provision and whose pleasure was my greatest motivation toward success,

To the noble martyrs of Gaza, Those who wrote, with their pure blood, an eternal story of resilience and honor, To the leaders who carried the cause with their souls before their words, especially Mohammed Deif, Yahya Sinwar, Hudhaifa al-Kahlout (Abu Obeida), Ismail Haniyeh, and every sincere leader who bore the concerns of The nation, and every soul that rose in defense of truth,

To my martyred friend, may Allah have mercy on him, Ahmed Al-Baraa, Who was not just a friend, but a pure soul and a symbol of loyalty, His memory will forever remain alive in my heart,

And to my dear friend, the devoted fighter Tamer Azzam, Who taught me that steadfastness is a attitude, and dignity is a path, And who stood as a true example of patience and faith,

To all my friends, Especially my dear friend Ismail, Salah ,AbdelRaouf, Who were part of my journey and shared with me moments of hardship and success, They were sincere support and unforgettable companions,

And to all my honorable teachers who taught me the Holy Quran, Who planted in my heart the light of guidance, and had, after Allah, a great role in shaping my spirit and values,

To everyone who believed in me, supported me, and accompanied me with sincere prayers, I dedicate this humble work, As a token of loyalty beyond words, and a love that cannot be fully expressed, A tribute to every great soul who left a lasting impact on my journey and became part of this achievement. ”

SELLAMI IMAD EDDINE

Acknowledgements

We would like to express our deepest gratitude and sincere appreciation to our supervisor, **Mrs. Ahlam Bouzaroura**, for her invaluable guidance, continuous support, and unwavering encouragement throughout the development of this thesis. Her insightful feedback, constructive remarks, and constant supervision played a pivotal role in shaping this work and bringing it to completion in its best possible form.

We are profoundly grateful for her patience, dedication, and generous willingness to share her knowledge and expertise, which significantly enhanced our understanding and strengthened our research capabilities. Her guidance was a constant source of motivation that enabled us to overcome challenges and maintain a strong commitment to excellence.

We also highly appreciate her professionalism, academic integrity, and genuine interest in our progress, which have left a lasting and positive impact on our academic journey. We are truly honored and grateful for the opportunity to work under her supervision.

Contents

Introduction	1
1 Artificial Intelligence and Machine Learning: Concepts and Foundations	3
1.1 Introduction	4
1.2 Definition of Artificial Intelligence	4
1.3 Domains of Artificial Intelligence	5
1.4 Definition of Machine Learning	6
1.5 Historical Background of Artificial Intelligence and Machine Learning . . .	6
1.6 Machine Learning Workflow	7
1.6.1 Data Collection	8
1.6.2 Data Preprocessing	8
1.6.3 Model Selection and Training	9
1.6.4 Evaluation	9
1.6.5 Deployment (Extension)	9
1.7 Types of Machine Learning	9
1.7.1 Supervised Learning	10
1.7.2 Unsupervised Learning	10
1.7.3 Semi-Supervised Learning	11
1.7.4 Reinforcement Learning	11
1.8 Machine Learning Models for Prospect Classification	12
1.8.1 K-Nearest Neighbors (KNN)	12
1.8.2 Decision Trees	12
1.8.3 Naive Bayes Classifier	12
1.8.4 Support Vector Machines (SVM)	13
1.8.5 Artificial Neural Networks (ANN)	13
1.9 Applications of Machine Learning in Digital Marketing	13
1.10 Conclusion	14
2 Neural Networks and NLP for Digital Marketing Optimization	15
2.1 Introduction	16
2.2 Mathematical Foundations of Artificial Neural Networks	16
2.2.1 Biological Inspiration and Computational Abstraction	16
2.2.2 Universal Approximation Theorem	17

2.3	Deep Network Architecture Formalization	17
2.4	Loss Functions for Prospect Classification	18
2.5	Bias-Variance Tradeoff	18
2.6	Gradient-Based Learning	19
2.7	Regularization Theory	19
2.7.1	L2 Regularization	19
2.7.2	L1 Regularization	19
2.8	Interpretability in Marketing AI Models	20
2.9	Connection to Prospect Classification	20
2.10	Transition to Learning Algorithms	20
2.11	Optimization and Learning Algorithms	21
2.11.1	Gradient Descent Optimization	21
2.11.2	Stochastic Gradient Descent (SGD)	21
2.11.3	Mini-Batch Gradient Descent	22
2.11.4	Adaptive Optimization Algorithms	22
2.11.5	Learning Rate Scheduling	23
2.11.6	Convex vs Non-Convex Optimization	23
2.11.7	Imbalanced Data Optimization	23
2.11.8	Overfitting and Generalization Theory	24
2.11.9	Computational Complexity Considerations	24
2.12	Transition to Neural Architectures	24
2.13	Advanced Neural Network Architectures	25
2.13.1	Feedforward Neural Networks (FNN)	25
2.13.2	Convolutional Neural Networks (CNN)	26
2.13.3	Recurrent Neural Networks (RNN)	27
2.13.4	Long Short-Term Memory (LSTM)	28
2.13.5	Gated Recurrent Unit (GRU)	28
2.13.6	Transformer Architecture	29
2.14	Architecture Selection for Prospect Classification	30
2.15	Transition to NLP Foundations	30
2.16	Natural Language Processing Foundations	30
2.16.1	NLP Processing Pipeline	31
2.16.2	Text Preprocessing	32
2.16.3	Tokenization	32
2.16.4	Probabilistic Language Modeling	32
2.16.5	n-Gram Models	33
2.16.6	Word Embeddings	33
2.16.7	Neural Language Models	34

2.16.8	LSTM-Based Text Modeling	34
2.16.9	Transformer-Based Text Modeling	35
2.17	NLP Integration in Prospect Classification	35
2.18	Transition to Prospect Classification and Evaluation	36
2.19	Neural Networks for Prospect Classification	36
2.19.1	Problem Formulation	36
2.19.2	Feature Engineering for Prospect Classification	37
2.19.3	Neural Classification Architecture	37
2.19.4	Loss Function for Binary Classification	38
2.20	Evaluation Metrics for Prospect Models	38
2.20.1	Confusion Matrix	38
2.20.2	Key Metrics	38
2.20.3	ROC Curve and AUC	39
2.21	Optimization Strategies for Marketing Models	39
2.21.1	Class Imbalance Handling	40
2.21.2	Threshold Optimization	40
2.21.3	Business-Oriented Optimization	40
2.22	Integration of Textual Intelligence in Prospect Scoring	40
2.23	Practical Relevance for Digital Marketing Systems	41
2.24	Conclusion	41
3	Proposed System Architecture and Design for Intelligent Prospect Classification	42
3.1	Introduction	43
3.2	Problem Definition	43
3.3	Prospect classification and recent trends	44
3.4	System Perspective	44
3.5	UML Modeling of the Proposed System	46
3.5.1	UML Use Case Diagram	46
3.5.2	UML Class Diagram	47
3.6	Learning Paradigm and Design Choices	49
3.7	Proposed Hybrid Architecture	50
3.7.1	Textual Data Processing	51
3.7.2	Behavioral Data Processing	51
3.7.3	Feature Fusion Mechanism	51
3.7.4	Classification Model	52
3.7.5	Hybrid Decision Mechanism	52
3.7.6	Mathematical Formulation	52
3.8	Model Architecture (Hybrid Neural Network)	53

3.9	Critical Discussion	54
3.10	Conclusion	55
4	Implementation and Experimental Evaluation	56
4.1	Introduction	57
4.2	Hardware Configuration	57
4.3	Development Environment	58
4.3.1	Programming Language	58
4.3.2	Development Tools	58
4.3.3	Libraries and Frameworks	58
4.4	System Architecture Overview	59
4.5	Model Architecture	59
4.5.1	Dataset Construction	59
4.5.2	Data Preprocessing	60
4.5.3	Model Design	61
4.5.4	Implementation Details	61
4.5.5	Discussion	63
4.6	Model Training and Evaluation	63
4.6.1	Training Process	63
4.6.2	Evaluation Metrics	64
4.6.3	Training Results	65
4.6.4	Confusion Matrix	66
4.6.5	Discussion of Results	66
4.7	System Implementation	66
4.7.1	System Overview	67
4.7.2	Campaign Management Interface	67
4.7.3	Tracking System	68
4.7.4	Feature Engineering	68
4.7.5	Prediction Module	69
4.7.6	Analysis Dashboard	69
4.7.7	Performance Monitoring	69
4.7.8	Discussion	70
4.8	Results and Analysis	70
4.8.1	System Output Interpretation	70
4.8.2	Behavioral Insights	71
4.8.3	System Effectiveness	71
4.8.4	Business Impact and Marketing Implications	71
4.8.5	Conclusion	72

General Conclusion	73
References	75
A Data Processing and Feature Engineering Details	77
B List of Acronyms	79

List of Figures

1.1	Main domains of Artificial Intelligence	5
1.2	Key milestones in the history of Artificial Intelligence and Machine Learning	7
1.3	Typical Machine Learning workflow	8
1.4	Classification and regression tasks in supervised learning	10
1.5	Supervised learning process	11
1.6	Unsupervised learning process	11
1.7	Reinforcement learning framework	12
1.8	Example of a decision tree model	13
1.9	Basic architecture of an artificial neural network	14
2.1	Convolutional Neural Network Architecture	26
2.2	Recurrent Neural Network for Text Processing	27
2.3	Transformer Architecture	29
2.4	Natural Language Processing Pipeline	31
2.5	Word Embedding Representation Space	33
2.6	Fully Connected Neural Network for Classification	37
2.7	Gradient Descent Optimization Process	39
3.1	Proposed hybrid system architecture for prospect classification	45
3.2	UML Use Case Diagram of the Proposed Hybrid System	46
3.3	UML Class Diagram of the Hybrid Intelligent System	48
3.4	Proposed Hybrid Architecture combining LSTM-based text modeling, behavioral features, and BERT-based sentiment analysis	51
3.5	Hybrid neural network architecture	54
4.1	Sample of the hybrid dataset	60
4.2	Text tokenization and sequence transformation	60
4.3	Feature scaling of behavioral data	61
4.4	Hybrid neural network architecture	61
4.5	Training and validation accuracy over epochs	65
4.6	Training and validation loss over epochs	65
4.7	Confusion matrix of the model predictions	66
4.8	Campaign creation interface with live email preview	67
4.9	Event-based tracking data stored in logs.csv	68
4.10	Prediction and analysis dashboard	69

4.11 System performance and engagement statistics	70
---	----

Introduction

The rapid advancement of digital technologies and the exponential growth of data have profoundly transformed modern marketing strategies. Traditional approaches are no longer sufficient to address the complexity of user behavior and the increasing demand for personalized communication. As a result, organizations are increasingly adopting intelligent, data-driven systems to analyze customer interactions and optimize decision-making processes.

This research is motivated not only by theoretical considerations but also by a real-world problem encountered during a professional experience in the digital marketing domain. One of the main challenges is identifying genuinely interested customers among a large number of prospects. Existing approaches often rely on manual analysis or simplistic metrics, which leads to inefficient targeting and missed opportunities.

In this context, Artificial Intelligence (AI) and Machine Learning (ML) provide powerful tools for analyzing user behavior and automating decision-making processes. However, most existing solutions focus either on behavioral data or textual data, without effectively combining both sources of information.

One of the key challenges in digital marketing is prospect classification, also known as lead scoring. This task involves evaluating user interest based on multiple signals such as email opening, link clicking, and interaction patterns. In addition, textual responses from users contain valuable information that is often underutilized.

Several limitations can be identified in current approaches:

- The difficulty of integrating heterogeneous data sources (behavioral and textual)
- The lack of real-time tracking and dynamic feature generation
- The absence of adaptive systems capable of continuous learning from new data

To overcome these limitations, this work proposes a complete intelligent system for prospect classification. The system integrates a real-time tracking module that records user interactions (opens, clicks) and stores them as structured data. It also includes a module for extracting and processing textual responses from emails.

A hybrid machine learning approach is then applied, combining structured behavioral features with textual features. The prediction module relies on a neural network model enhanced with sentiment-aware decision logic to improve classification accuracy.

Furthermore, the system incorporates a self-learning mechanism that automatically collects new interaction data and uses it to continuously update the dataset and improve the model over time.

The objective of this work is not only to build a predictive model but also to design an intelligent end-to-end system capable of automatically identifying high-interest prospects and supporting marketing decision-making in real-world environments.

The main contributions of this work can be summarized as follows:

- Design of a complete intelligent system for prospect classification
- Integration of real-time tracking for user interaction analysis
- Combination of behavioral and textual data within a hybrid learning model
- Implementation of a self-learning mechanism for continuous improvement
- Development of a decision-support dashboard for result visualization

This manuscript is organized as follows:

- Chapter 1 presents the theoretical foundations of Artificial Intelligence and Machine Learning
- Chapter 2 introduces neural networks and Natural Language Processing techniques
- Chapter 3 describes the proposed system architecture
- Chapter 4 presents the implementation details and experimental evaluation

This work demonstrates how integrating Artificial Intelligence, behavioral tracking, and textual analysis can significantly enhance prospect classification and improve marketing efficiency in practical applications.

Chapter 1

Artificial Intelligence and Machine Learning: Concepts and Foundations

1.1 Introduction

The rapid evolution of information technologies and the exponential growth of digital data have profoundly transformed modern organizations and business environments. In particular, digital marketing has shifted from traditional mass-oriented strategies toward data-driven and customer-centric approaches. As a result, organizations are increasingly required to analyze large volumes of user data in order to understand customer behavior and optimize marketing decisions.

In this context, Artificial Intelligence (AI) has emerged as a key technological paradigm for optimizing digital marketing strategies. By enabling systems to learn from data, identify patterns, and make predictions, AI allows organizations to improve targeting accuracy, personalize marketing content, and enhance customer engagement.

Machine Learning, as a core subfield of Artificial Intelligence, plays a central role in this transformation. Machine learning techniques make it possible to automatically classify prospects based on their behavior, preferences, and likelihood of conversion. Such classification is essential for prioritizing high-potential prospects and allocating marketing resources efficiently. Furthermore, the integration of Artificial Intelligence into operational systems, including decision-support tools and interactive user interfaces, reinforces its practical value. This transformation enables organizations not only to analyze data but also to act intelligently in real time.

This chapter presents a comprehensive theoretical background on Artificial Intelligence and Machine Learning. It introduces key definitions, core domains, historical foundations, and the general workflow of machine learning systems. These concepts form the theoretical basis for the design of an Artificial Intelligence-based prospect classification model aimed at optimizing digital marketing strategies.[11, 4].

1.2 Definition of Artificial Intelligence

Artificial Intelligence refers to the branch of computer science concerned with the design and development of systems capable of performing tasks that typically require human intelligence[21]. These tasks include learning, reasoning, perception, decision-making, and problem-solving.

An intelligent system is able to perceive its environment, process information, learn from experience, and adapt its behavior in order to achieve predefined objectives. Unlike traditional software systems that follow static rules, AI-based systems are capable of improving their performance over time through learning mechanisms.

One of the most influential definitions of Artificial Intelligence was proposed by John McCarthy, who described AI as:

"the science and engineering of making intelligent machines."

This definition highlights the dual nature of AI as both a scientific discipline and an engineering practice, focusing on the development of adaptive, autonomous, and intelligent systems.

Today, Artificial Intelligence is widely used across various industries, including healthcare, finance, cybersecurity, and digital marketing, where it plays a crucial role in decision-making and automation.[21].

1.3 Domains of Artificial Intelligence

Artificial Intelligence is a multidisciplinary field composed of several interconnected domains, each addressing specific challenges related to intelligent information processing and decision-making.

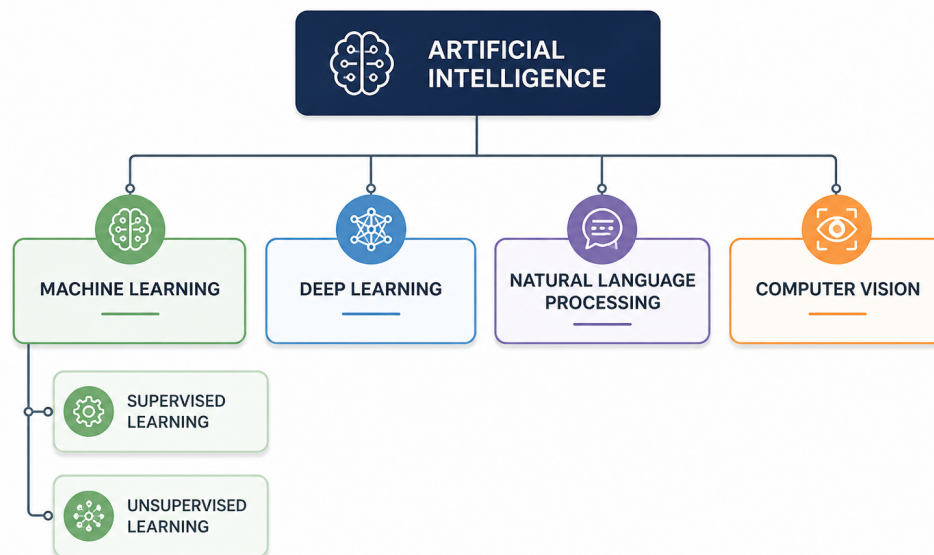


Figure 1.1: Main domains of Artificial Intelligence

As illustrated in Figure 1.1, Machine Learning represents a core component of Artificial Intelligence. Deep Learning is a specialized subfield of Machine Learning that relies on artificial neural networks with multiple hidden layers.

Other important domains include:

- **Natural Language Processing (NLP):** focuses on understanding and generating human language.
- **Computer Vision:** enables machines to interpret and analyze visual data.
- **Robotics:** integrates AI for autonomous systems.

These domains are often combined in real-world applications. For example, in digital marketing systems, NLP is used to analyze customer emails, while machine learning models classify prospects and predict user behavior.

1.4 Definition of Machine Learning

Machine Learning is a subfield of Artificial Intelligence that enables computer systems to learn from data without being explicitly programmed.[18] Instead of relying on predefined rules, machine learning models automatically identify patterns, correlations, and trends within historical data.

The concept of Machine Learning was first introduced by Arthur Samuel in 1958, who defined it as the ability of computers to learn from experience. Later, Tom Mitchell provided a formal definition stating that a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .

In the context of digital marketing, Machine Learning enables the automatic classification of prospects based on behavioral and interaction data. By analyzing user actions such as email responses, clicks, and browsing patterns, machine learning models can predict customer intent and optimize marketing strategies accordingly.

Moreover, Machine Learning plays a critical role in automating decision-making processes, reducing human intervention, and improving scalability in large-scale marketing systems.

1.5 Historical Background of Artificial Intelligence and Machine Learning

The origins of Artificial Intelligence and Machine Learning can be traced back to early research in mathematics, logic, and cognitive science. Over the past decades, several key milestones have shaped the evolution of these fields.

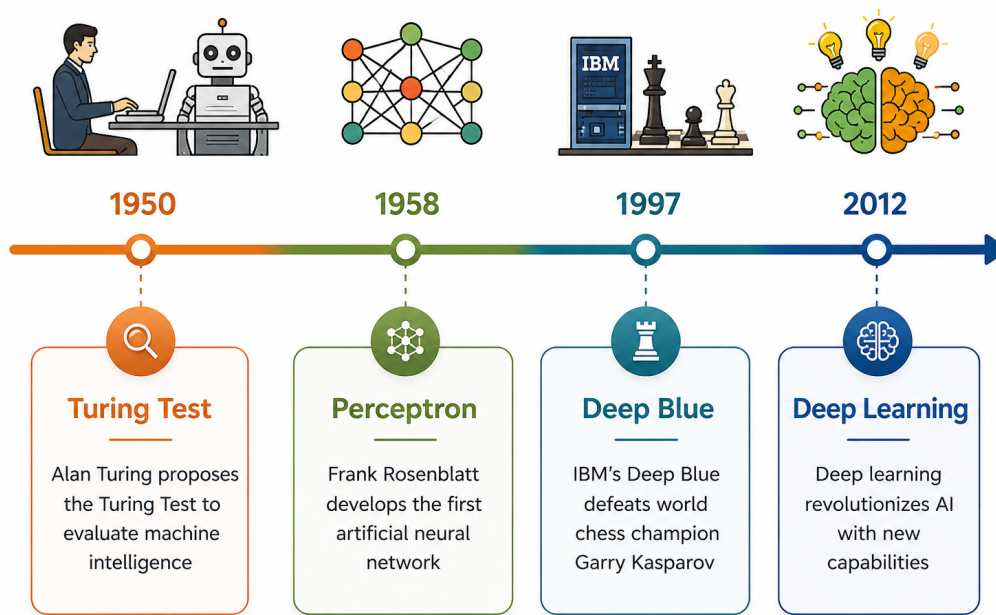


Figure 1.2: Key milestones in the history of Artificial Intelligence and Machine Learning

As shown in Figure 1.2, early contributions include Alan Turing's proposal of the Turing Test in 1950,[25] which aimed to evaluate machine intelligence. This test laid the conceptual foundation for evaluating intelligent behavior in machines.

In 1958, Frank Rosenblatt introduced the Perceptron, considered the first artificial neural network model capable of performing simple classification tasks.

In 1997, IBM's Deep Blue defeated the world chess champion Garry Kasparov, demonstrating the power of computational intelligence in solving complex problems.

More recently, the emergence of Deep Learning in the 2010s, driven by the availability of large datasets and powerful GPUs, has significantly advanced the field. Technologies such as neural networks, natural language processing, and computer vision have enabled the development of highly intelligent systems used in real-world applications.

These historical developments highlight the continuous evolution of Artificial Intelligence from theoretical research to practical deployment in various domains, including digital marketing[20].

1.6 Machine Learning Workflow

The development of a Machine Learning system follows a structured and iterative workflow composed of several essential stages. Each stage plays a critical role in determining the effectiveness and reliability of the final model.[10].

Figure 1.3 illustrates the main steps involved in building a Machine Learning system, including data collection, preprocessing, model training, evaluation, and deployment.



Figure 1.3: Typical Machine Learning workflow

1.6.1 Data Collection

Data collection is the first and most critical step in the machine learning workflow. In digital marketing applications, data may include user interactions such as email responses, website visits, click-through rates, and transaction histories.

The quality, quantity, and diversity of the collected data directly influence the performance and generalization capability of the machine learning model.

1.6.2 Data Preprocessing

Raw data often contains missing values, inconsistencies, or noise. Data preprocessing aims to clean and transform raw data into a suitable format for model training.

Common preprocessing techniques include:

- Handling missing values
- Data normalization and scaling
- Feature encoding (e.g., One-Hot Encoding)
- Text preprocessing (tokenization, stop-word removal)

Proper preprocessing ensures that the data is consistent and suitable for learning, thereby improving model performance.

1.6.3 Model Selection and Training

Once the data is prepared, an appropriate machine learning algorithm is selected based on the nature of the problem and the characteristics of the dataset.

The model is then trained using historical data to learn patterns that distinguish between different classes, such as qualified and non-qualified prospects.

During training, the model parameters are adjusted iteratively to minimize prediction error, often using optimization algorithms such as Gradient Descent.

1.6.4 Evaluation

Model evaluation is a crucial step that assesses the performance of the trained model on unseen data.

Common evaluation metrics include:

- Accuracy
- Precision
- Recall
- F1-score

These metrics provide insights into the model's effectiveness in correctly classifying prospects and its contribution to optimizing marketing strategies.

1.6.5 Deployment (Extension)

After evaluation, the model can be deployed within a real-world system, allowing users to interact with it through an interface. This step transforms the model from a theoretical construct into a practical tool capable of supporting decision-making processes in real time.

1.7 Types of Machine Learning

Machine Learning algorithms can be categorized into several learning paradigms depending on the nature of the available data and the learning strategy employed. Each paradigm addresses specific types of problems and plays a distinct role in data-driven decision-making processes.

In the context of digital marketing, selecting the appropriate learning paradigm is essential for designing efficient prospect classification models capable of optimizing marketing strategies and improving conversion rates.[\[18\]](#).

1.7.1 Supervised Learning

Supervised Learning is the most widely adopted Machine Learning paradigm, particularly for classification and prediction tasks. It relies on labeled datasets, where each input instance is associated with a known output label.

The objective is to learn a mapping function:

$$f(X) = Y$$

that can accurately predict the output for new, unseen data.

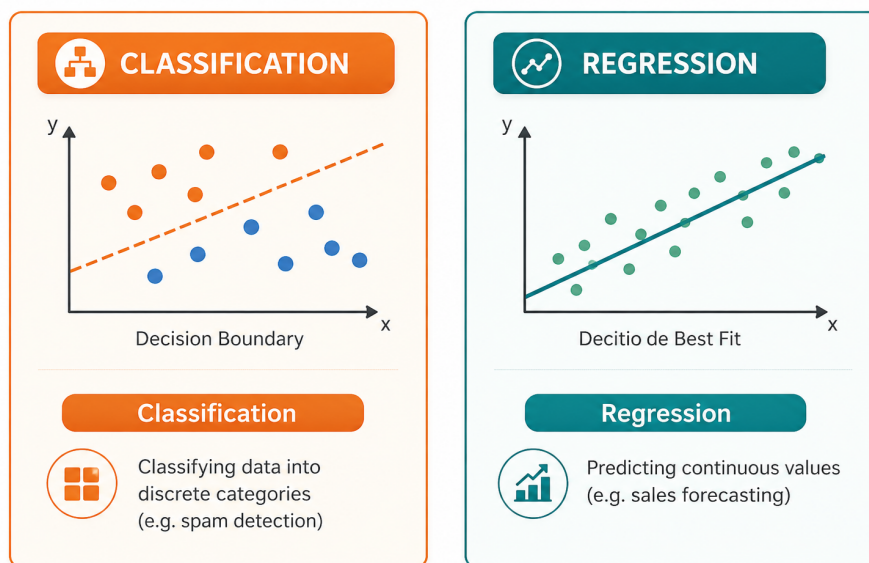


Figure 1.4: Classification and regression tasks in supervised learning

As illustrated in Figure 1.4, supervised learning problems are commonly divided into:

- **Classification:** Predicting discrete labels (e.g., interested vs not interested).
- **Regression:** Predicting continuous values.

Supervised learning is particularly relevant for prospect classification, as it allows models to learn from historical labeled data and generalize to new prospects.

1.7.2 Unsupervised Learning

Unsupervised Learning deals with datasets that do not contain labeled outputs. The primary objective is to identify hidden structures, patterns, or relationships within the data.

Common unsupervised learning techniques include clustering and dimensionality reduction. In digital marketing, clustering can be used for customer segmentation, allowing marketers to group users based on similar behaviors or preferences.



Figure 1.5: Supervised learning process

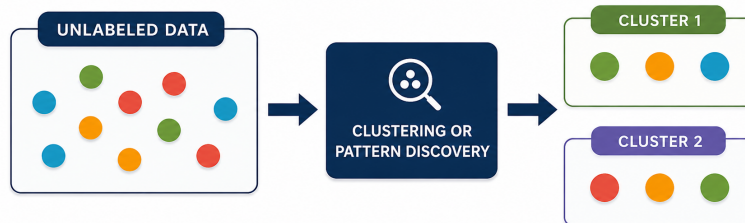


Figure 1.6: Unsupervised learning process

1.7.3 Semi-Supervised Learning

Semi-Supervised Learning combines a limited amount of labeled data with a large volume of unlabeled data during training.

This approach is particularly useful in real-world scenarios where labeling data is expensive or time-consuming. By leveraging both labeled and unlabeled data, semi-supervised learning improves model performance while reducing annotation effort.

1.7.4 Reinforcement Learning

Reinforcement Learning is a paradigm in which an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties and aims to maximize cumulative rewards over time.

Although less commonly used in email classification, reinforcement learning can be applied in marketing for tasks such as campaign optimization and adaptive recommendation systems.

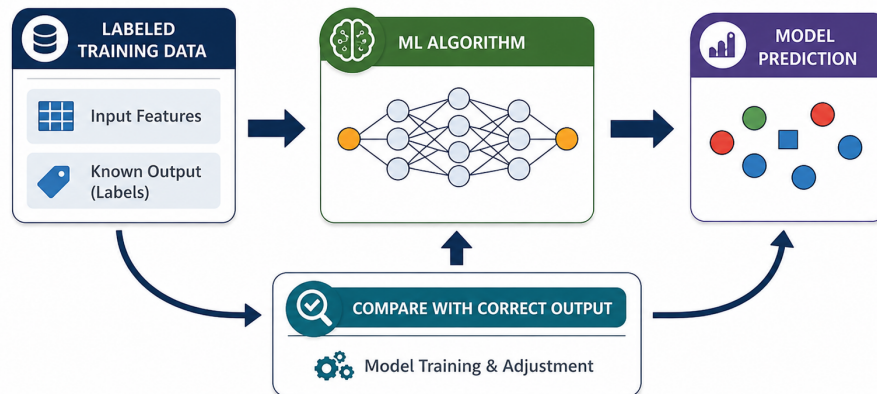


Figure 1.7: Reinforcement learning framework

1.8 Machine Learning Models for Prospect Classification

A variety of Machine Learning models have been developed to address classification problems. The selection of an appropriate model depends on factors such as dataset size, feature complexity, interpretability requirements, and computational constraints.[2].

1.8.1 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm is a simple and intuitive supervised learning model. It classifies a data point based on the majority class of its nearest neighbors in the feature space.

Despite its simplicity, KNN can be effective for small datasets. However, it becomes computationally expensive as the dataset size increases.

1.8.2 Decision Trees

Decision Trees represent decision-making processes using a hierarchical tree structure.

Each node represents a decision based on a feature, and each branch corresponds to an outcome. Decision Trees are highly interpretable, making them attractive for marketing applications where understanding model decisions is important.

1.8.3 Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic model based on Bayes' theorem. It assumes independence between features, which simplifies computation.

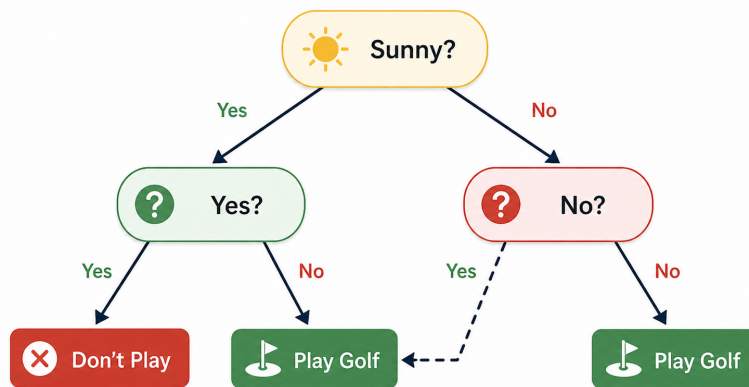


Figure 1.8: Example of a decision tree model

Despite this assumption, Naive Bayes performs well in text classification tasks, including email filtering and sentiment analysis, making it relevant for prospect classification.

1.8.4 Support Vector Machines (SVM)

Support Vector Machines are powerful supervised learning models that aim to find the optimal decision boundary separating different classes.

They are particularly effective in high-dimensional spaces and are widely used in classification tasks involving textual data.

1.8.5 Artificial Neural Networks (ANN)

Artificial Neural Networks are inspired by the structure of the human brain. They consist of interconnected neurons organized into layers.

Neural networks are capable of modeling complex non-linear relationships and have become the dominant approach in modern machine learning applications, particularly with the rise of deep learning.

In the context of this work, neural networks are used to classify prospects based on both structured and textual data.

1.9 Applications of Machine Learning in Digital Marketing

Machine Learning techniques are widely applied to optimize digital marketing strategies and improve business performance.

Key applications include:

- **Lead classification and scoring:** Identifying high-potential prospects.

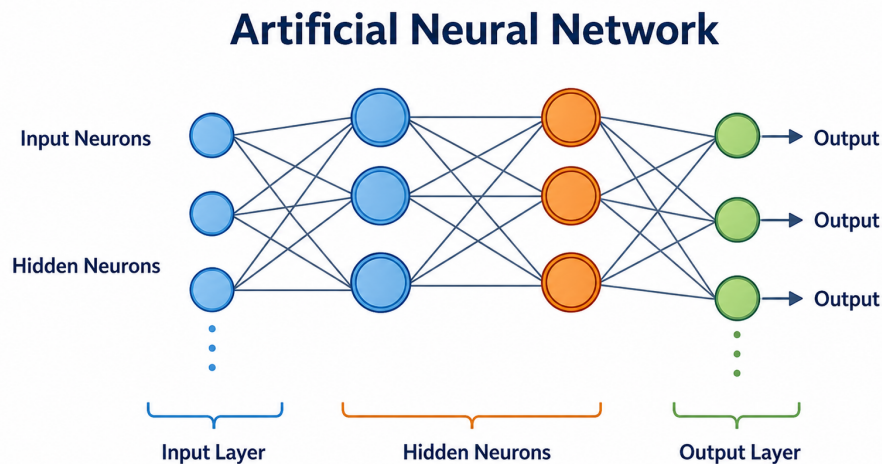


Figure 1.9: Basic architecture of an artificial neural network

- **Personalized recommendations:** Suggesting products based on user behavior.
- **Customer segmentation:** Grouping customers with similar characteristics.
- **Churn prediction:** Identifying customers likely to leave.
- **Campaign optimization:** Improving marketing effectiveness.

These applications demonstrate the importance of Machine Learning as a tool for transforming raw data into actionable insights, enabling organizations to make informed and strategic decisions.^[11]

1.10 Conclusion

This chapter presented the fundamental concepts of Artificial Intelligence and Machine Learning, highlighting their role in data analysis and decision-making processes. It introduced the main learning paradigms, commonly used models, and their applications in digital marketing.

The discussion emphasized the importance of Machine Learning in prospect classification, where it enables the identification of high-potential leads and supports the optimization of marketing strategies.

Overall, this chapter establishes the theoretical foundation necessary for the design and implementation of an intelligent classification model, which will be further developed in the following chapters.

Chapter 2

Neural Networks and NLP for Digital Marketing Optimization

2.1 Introduction

The increasing reliance on predictive analytics in digital marketing has transformed traditional campaign management into a data-driven optimization problem. Organizations now leverage Artificial Intelligence models to automatically classify prospects, predict customer conversion probability, and optimize resource allocation.[11]

In addition to theoretical modeling, neural networks and NLP techniques are increasingly integrated into real-world systems such as intelligent marketing platforms. These systems leverage predictive modeling to automate prospect classification, enabling faster and more accurate decision-making. This highlights the importance of bridging theoretical foundations with practical implementations, which will be further explored in subsequent chapters. Artificial Neural Networks (ANNs) represent one of the most powerful modeling tools in this context. Their ability to approximate highly non-linear decision boundaries makes them particularly suitable for prospect classification problems where behavioral, demographic, and textual features interact in complex ways.[11].

This chapter begins by establishing the theoretical foundations of neural computation before progressively connecting these principles to digital marketing optimization.

2.2 Mathematical Foundations of Artificial Neural Networks

2.2.1 Biological Inspiration and Computational Abstraction

Artificial Neural Networks are inspired by biological neurons[11].. However, the artificial neuron is not a biological simulation; it is a mathematical abstraction designed for function approximation.

Let $x \in \mathbb{R}^n$ be an input vector representing prospect features:

$$x = (x_1, x_2, \dots, x_n)$$

Where each feature may represent:

- Email engagement rate
- Website visit duration
- Purchase history score
- Sentiment polarity
- Interaction frequency

A neuron computes[2]:

$$z = w^T x + b$$

Where:

- $w \in \mathbb{R}^n$ is the weight vector
- $b \in \mathbb{R}$ is the bias
- z is the linear combination

The output is:

$$y = f(z)$$

Where f is a non-linear activation function.

2.2.2 Universal Approximation Theorem

A key theoretical justification for neural networks is the Universal Approximation Theorem[11].

It states that:

A feedforward neural network with at least one hidden layer and a non-linear activation function can approximate any continuous function on a compact subset of $\mathbb{R}^n$ to arbitrary precision.

This implies that neural networks can theoretically model complex prospect conversion functions:

$$P(\text{conversion}|\text{features}) = F(x)$$

Where F is unknown and highly non-linear.

2.3 Deep Network Architecture Formalization

Consider a network with L layers.

For layer l :

$$a^{(l)} = f^{(l)}(W^{(l)}a^{(l-1)} + b^{(l)})$$

Where:

- $a^{(l)}$: activation of layer l

- $W^{(l)}$: weight matrix
- $b^{(l)}$: bias vector

The final output layer produces:

$$\hat{y} = g(a^{(L)})$$

For binary prospect classification[11]:

$$\hat{y} = \sigma(z)$$

Where:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

This outputs the probability that a prospect belongs to the high-value class.

2.4 Loss Functions for Prospect Classification

In binary classification:

$$L = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

This is the Binary Cross-Entropy Loss[11].

For multi-class classification:

$$L = - \sum_{i=1}^K y_i \log(\hat{y}_i)$$

Where K represents prospect categories.

Minimizing this loss ensures optimal separation between prospect classes.

2.5 Bias-Variance Tradeoff

Model performance depends on balancing bias and variance.

- High bias $\rightarrow$ Underfitting
- High variance $\rightarrow$ Overfitting

In marketing classification:

- Underfitting leads to poor campaign targeting.
- Overfitting leads to unreliable prospect scoring.

Optimal complexity minimizes generalization error:

$$\text{Error} = \text{Bias}^2 + \text{Variance} + \text{Noise}$$

2.6 Gradient-Based Learning

Parameter optimization relies on gradient descent[10].

Given loss $L(\theta)$:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(\theta_t)$$

Where η is the learning rate.

Convergence depends on:

- Learning rate magnitude
- Loss surface curvature
- Initialization

In marketing datasets, highly imbalanced data can affect gradient stability.

2.7 Regularization Theory

To prevent overfitting:

2.7.1 L2 Regularization

$$L_{reg} = L + \lambda \|w\|^2$$

2.7.2 L1 Regularization

$$L_{reg} = L + \lambda \|w\|_1$$

Regularization controls model complexity[11] and improves generalization to unseen prospects.

2.8 Interpretability in Marketing AI Models

In digital marketing, interpretability is critical.

Feature importance can be analyzed via:

- Weight magnitude
- SHAP values
- Sensitivity analysis

This allows decision-makers to understand which behavioral features most influence prospect scoring.

2.9 Connection to Prospect Classification

Formally, the classification model estimates:

$$S(x) = P(Y = 1|x)$$

Where:

- $S(x)$ is the prospect score
- $Y = 1$ indicates high-value prospect

Decision rule:

$$\text{Prospect Class} = \begin{cases} \text{High Value} & \text{if } S(x) \geq \tau \\ \text{Low Value} & \text{otherwise} \end{cases}$$

Where τ is a threshold selected to optimize marketing objectives.

2.10 Transition to Learning Algorithms

Having established the mathematical and theoretical foundations of neural networks, the next section will explore advanced learning algorithms and optimization techniques that enable efficient model training on large-scale marketing datasets.

2.11 Optimization and Learning Algorithms

Training a neural network for prospect classification is fundamentally an optimization problem. The objective is to determine the optimal parameter vector θ that minimizes a predefined loss function over a dataset of labeled marketing observations.

Let the dataset be:

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^m$$

Where:

- $x^{(i)} \in \mathbb{R}^n$ represents prospect feature vectors
- $y^{(i)} \in \{0, 1\}$ represents class labels
- m is the number of training samples

The empirical risk minimization objective is:

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^m L(f(x^{(i)}; \theta), y^{(i)})$$

2.11.1 Gradient Descent Optimization

The classical optimization strategy is Gradient Descent:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t)$$

Where:

- η is the learning rate
- $J(\theta)$ is the cost function

However, in high-dimensional marketing datasets, the loss surface is highly non-convex. Therefore, optimization dynamics become critical.

2.11.2 Stochastic Gradient Descent (SGD)

Instead of computing gradients over the entire dataset, SGD updates parameters per sample:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} L(f(x^{(i)}; \theta), y^{(i)})$$

Advantages:

- Faster convergence

- Better generalization properties
- Noise helps escape shallow local minima

2.11.3 Mini-Batch Gradient Descent

In practice, mini-batches are used:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{B} \sum_{i \in \mathcal{B}} \nabla_{\theta} L(f(x^{(i)}; \theta), y^{(i)})$$

Where:

- $\mathcal{B}$ is a mini-batch
- B is batch size

Mini-batch training balances computational efficiency and gradient stability.

2.11.4 Adaptive Optimization Algorithms

Adagrad

Adagrad adapts the learning rate per parameter:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} g_{t,i}$$

Where $G_{t,i}$ accumulates squared gradients.

Adam (Adaptive Moment Estimation)

Adam combines momentum and RMSProp[15]:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

Adam is particularly suitable for marketing datasets due to:

- Sparse features
- Noisy gradients
- Large-scale data

2.11.5 Learning Rate Scheduling

Learning rate strongly influences convergence.

Common strategies:

- Step decay
- Exponential decay
- Cosine annealing

Proper scheduling prevents:

- Overshooting minima
- Slow convergence

2.11.6 Convex vs Non-Convex Optimization

Logistic regression is convex. Deep neural networks are non-convex.

However, empirical evidence shows:

- Most local minima are nearly equivalent in large networks
- Overparameterization aids optimization

This explains why deep models perform well in large marketing datasets.

2.11.7 Imbalanced Data Optimization

Digital marketing datasets are often imbalanced:

$$P(Y = 1) \ll P(Y = 0)$$

Solutions include:

- Weighted loss functions:

$$L = -\alpha y \log(\hat{y}) - (1 - \alpha)(1 - y) \log(1 - \hat{y})$$

- Oversampling minority class
- Focal loss

Proper imbalance handling is critical for prospect detection.

2.11.8 Overfitting and Generalization Theory

Generalization error:

$$E_{gen} = E_{test} - E_{train}$$

Regularization techniques:

- Dropout
- Weight decay
- Early stopping
- Cross-validation

In marketing optimization, strong generalization ensures stable campaign targeting.

2.11.9 Computational Complexity Considerations

Training complexity:

$$O(L \cdot n^2)$$

Where:

- L is number of layers
- n is average neuron count

Marketing applications often require:

- Scalable architectures
- Efficient inference
- Real-time prediction capability

2.12 Transition to Neural Architectures

Having established advanced optimization principles and learning mechanisms, the next section presents neural network architectures specifically adapted to structured, sequential, and textual marketing data.

2.13 Advanced Neural Network Architectures

Modern digital marketing datasets are heterogeneous. They include structured numerical features, sequential behavioral logs, and unstructured textual content. Consequently, different neural architectures must be employed depending on data modality.

This section presents advanced neural network architectures and their relevance to prospect classification and marketing optimization.

2.13.1 Feedforward Neural Networks (FNN)

Feedforward Neural Networks represent the simplest deep learning architecture. Information flows strictly from input to output without cycles.

Mathematically, for layer l :

$$a^{(l)} = f^{(l)}(W^{(l)}a^{(l-1)} + b^{(l)})$$

FNNs are particularly suited for structured marketing data such as:

- Click-through rate
- Purchase frequency
- Session duration
- Demographic indicators

Their strength lies in modeling nonlinear interactions between independent features. However, they do not inherently model temporal dependencies or textual sequences.

2.13.2 Convolutional Neural Networks (CNN)

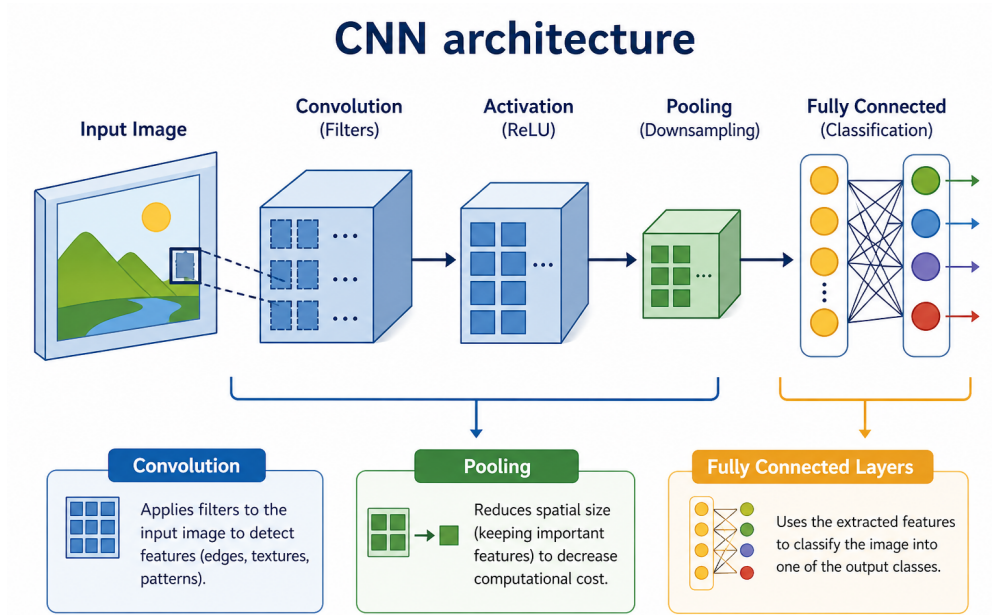


Figure 2.1: Convolutional Neural Network Architecture

CNNs were originally developed for computer vision but have since been adapted to structured and textual data[11].

A convolution operation is defined as:

$$(S * K)(i, j) = \sum_m \sum_n S(m, n) K(i - m, j - n)$$

Where:

- S is the input signal
- K is the kernel (filter)

In marketing applications, CNNs can be used for:

- Feature extraction from campaign matrices
- Detecting patterns in engagement heatmaps
- Text classification using 1D convolutions

CNN advantages:

- Parameter sharing

- Translation invariance
- Efficient feature hierarchy learning

Pooling layers reduce dimensionality:

$$\text{MaxPooling}(x) = \max(x_i)$$

This helps reduce overfitting and computational cost.

2.13.3 Recurrent Neural Networks (RNN)

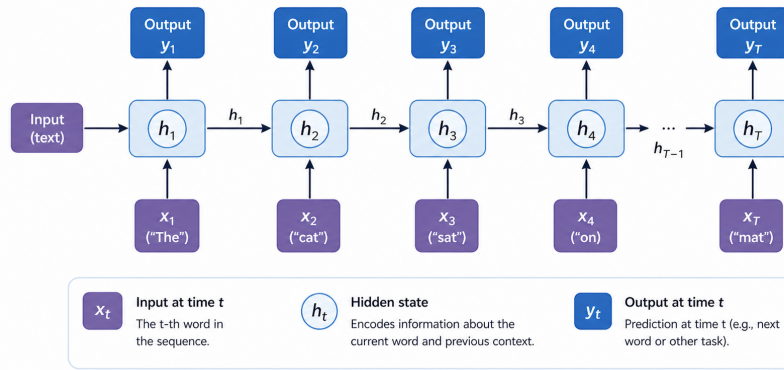


Figure 2.2: Recurrent Neural Network for Text Processing

RNNs[11] are designed for sequential data. Unlike feedforward networks, they maintain hidden states:

$$h_t = f(W_h h_{t-1} + W_x x_t + b)$$

Where:

- x_t is the input at time step t
- h_t is the hidden state

In digital marketing, RNNs are valuable for modeling:

- Customer journey sequences
- Email campaign timelines
- Behavioral event streams

However, standard RNNs suffer from vanishing and exploding gradient problems.

2.13.4 Long Short-Term Memory (LSTM)

LSTM [13] networks solve long-term dependency issues using gating mechanisms.

Core equations:

Forget gate:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

Input gate:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

Cell state update:

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t$$

Output gate:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

In marketing optimization:

- LSTM captures long-term engagement patterns
- Identifies repeated customer interest signals
- Improves conversion probability estimation

2.13.5 Gated Recurrent Unit (GRU)

GRU simplifies LSTM architecture by merging gates.

Update gate:

$$z_t = \sigma(W_z[h_{t-1}, x_t])$$

Reset gate:

$$r_t = \sigma(W_r[h_{t-1}, x_t])$$

GRUs are computationally lighter and suitable for real-time marketing systems.

2.13.6 Transformer Architecture

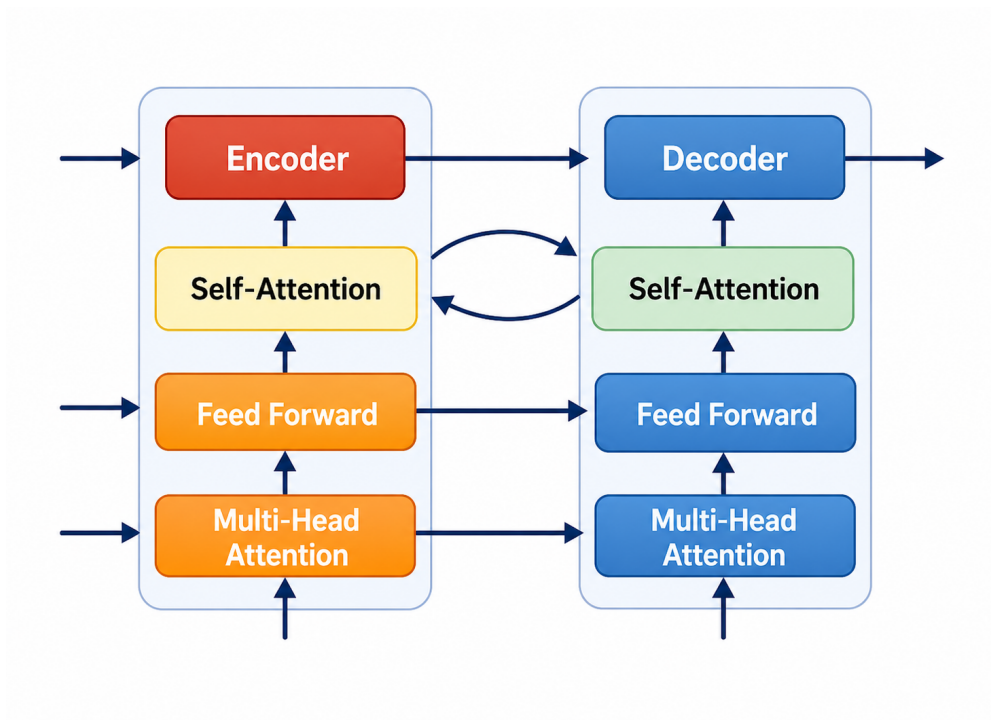


Figure 2.3: Transformer Architecture

Transformers revolutionized NLP by removing recurrence and introducing self-attention[11].

Self-attention mechanism:

$$Attention(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

Where:

- Q : Query matrix
- K : Key matrix
- V : Value matrix

Advantages:

- Parallel computation
- Better long-range dependency modeling
- Improved contextual understanding

In marketing:

- Email content classification
- Sentiment analysis
- Intent detection
- Automated lead qualification

Models such as BERT and GPT are Transformer-based architectures widely used in advanced NLP systems.

2.14 Architecture Selection for Prospect Classification

Selecting the appropriate architecture depends on:

- Data type (structured vs textual)
- Dataset size
- Real-time constraints
- Computational resources

For this thesis, a hybrid architecture combining:

- Feedforward layers for structured features
- LSTM or Transformer components for textual inputs

provides an optimal balance between predictive power and scalability.

2.15 Transition to NLP Foundations

Having explored advanced neural architectures for structured and sequential data, the next section will focus on Natural Language Processing foundations and their integration into marketing-oriented neural systems.

2.16 Natural Language Processing Foundations

Natural Language Processing (NLP)[11] is a subfield of Artificial Intelligence concerned with enabling machines to understand, interpret, and generate human language.

In digital marketing, NLP plays a crucial role in:

- Email campaign analysis
- Customer feedback mining
- Sentiment detection
- Intent recognition
- Automated lead qualification

Textual data represents a highly informative but unstructured source of behavioral signals. Therefore, efficient preprocessing and modeling are required.

2.16.1 NLP Processing Pipeline

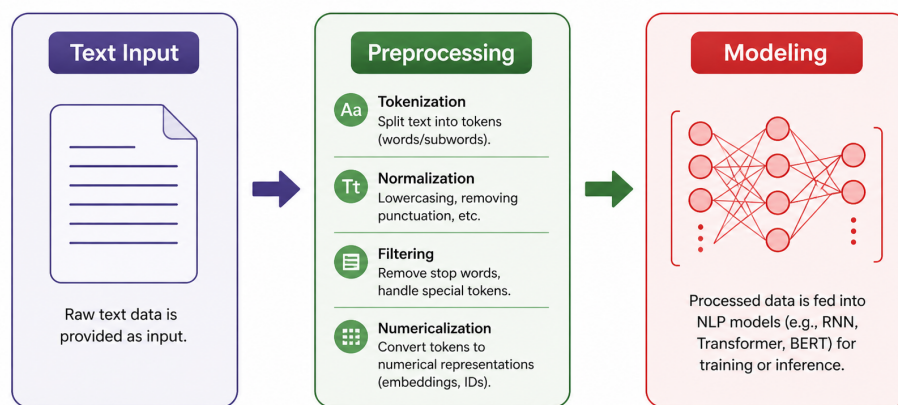


Figure 2.4: Natural Language Processing Pipeline

The NLP pipeline typically includes:

1. Text cleaning
2. Tokenization
3. Normalization
4. Vectorization
5. Model training
6. Evaluation

Each stage transforms raw text into structured numerical representations suitable for neural models.

2.16.2 Text Preprocessing

Preprocessing steps include:

- Lowercasing
- Stopword removal
- Stemming or lemmatization
- Punctuation filtering
- Noise removal

In marketing contexts, preprocessing must preserve domain-specific vocabulary such as product names or brand identifiers.

2.16.3 Tokenization

Tokenization splits text into meaningful units[4].

Given a sentence:

$$S = (w_1, w_2, \dots, w_n)$$

Tokenization converts it into discrete tokens:

$$T = \{t_1, t_2, \dots, t_n\}$$

Advanced tokenization approaches:

- Word-level tokenization
- Subword tokenization (BPE)
- Character-level tokenization

Subword tokenization improves generalization for rare marketing terms.

2.16.4 Probabilistic Language Modeling

A language model estimates:

$$P(w_1, w_2, \dots, w_n)$$

Using the chain rule:

$$P(w_1, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1})$$

This formulation enables next-word prediction and contextual modeling.

2.16.5 n-Gram Models

n-gram models approximate:

$$P(w_i | w_1, \dots, w_{i-1}) \approx P(w_i | w_{i-n+1}, \dots, w_{i-1})$$

Limitations:

- Limited context window
- Data sparsity
- Poor generalization

Thus, neural approaches have replaced classical n-gram methods in modern marketing systems.

2.16.6 Word Embeddings

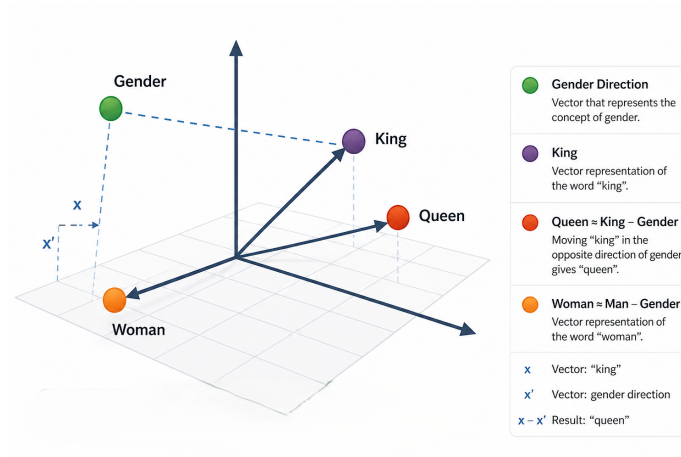


Figure 2.5: Word Embedding Representation Space

Word embeddings map discrete tokens into continuous vector spaces:

$$w_i \rightarrow \vec{v}_i \in \mathbb{R}^d$$

Where:

- d is embedding dimension
- Semantic similarity corresponds to vector proximity

Cosine similarity:

$$\text{sim}(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$$

Embeddings enable semantic reasoning such as:

$$“\vec{discount}” \approx “\vec{promotion}”$$

This is particularly useful for marketing keyword clustering[17].

2.16.7 Neural Language Models

Neural Language Models (NLMs) use neural networks to estimate sequence probabilities.

Given embeddings x_t :

$$h_t = f(W_h h_{t-1} + W_x x_t)$$

$$P(w_{t+1} | context) = \text{softmax}(W_o h_t)$$

These models capture:

- Long-term dependencies
- Semantic relationships
- Context-aware probability estimation

2.16.8 LSTM-Based Text Modeling

LSTM models are particularly effective for:

- Email content classification
- Customer complaint analysis
- Behavioral text prediction

They retain long-term contextual signals which are crucial in intent detection.

2.16.9 Transformer-Based Text Modeling

Transformer models compute contextual embeddings:

$$Z = \text{Transformer}(X)$$

Advantages:

- Global attention
- Parallel processing
- Superior contextual representation

In prospect classification, Transformer-based embeddings improve prediction of purchase intent from textual interactions.

2.17 NLP Integration in Prospect Classification

The integration of structured and textual data plays a crucial role in improving prospect classification systems. This process involves combining structured behavioral features with textual representations extracted from user responses:

$$z = [x_{\text{structured}}; x_{\text{textual}}]$$

where $;$ denotes the concatenation operation.

Hybrid intelligent systems have emerged as an effective approach for prospect classification in digital marketing environments. These systems leverage both structured behavioral data, such as email openings and link clicks, and unstructured textual data derived from user responses.

The fusion of these heterogeneous data sources enables models to capture a more comprehensive representation of user behavior and intent, resulting in more accurate and reliable predictions.

Furthermore, decision logic components can be incorporated to enhance classification performance by combining machine learning outputs with sentiment-based analysis of textual data.

Such hybrid approaches significantly improve the robustness and effectiveness of classification systems, particularly in real-world scenarios characterized by high data diversity and complexity.

2.18 Transition to Prospect Classification and Evaluation

Having established the theoretical foundations of neural architectures and NLP systems, the next section presents the formal modeling of prospect classification, evaluation metrics, and optimization strategies for digital marketing systems.

2.19 Neural Networks for Prospect Classification

Prospect classification refers to the task of automatically identifying and categorizing potential customers based on behavioral, demographic, and textual data.

In digital marketing systems, this task aims to answer a fundamental question:

Which prospects are most likely to convert?

Neural networks are particularly suitable for this task due to their ability to:

- Learn complex nonlinear relationships
- Integrate heterogeneous data sources
- Adapt to large-scale datasets
- Automatically extract latent patterns

2.19.1 Problem Formulation

Let:

$$X = \{x_1, x_2, \dots, x_n\}$$

be the input feature vectors describing prospects.

Each prospect is associated with a label:

$$y \in \{0, 1\}$$

Where:

- $y = 1$ indicates high conversion probability
- $y = 0$ indicates low conversion probability

The objective is to learn a function:

$$f_\theta : X \rightarrow \hat{y}$$

where θ represents model parameters.

2.19.2 Feature Engineering for Prospect Classification

Prospect features typically include:

- Demographic features (age, region, profession)
- Behavioral metrics (click-through rate, session duration)
- Transaction history
- Email interaction metrics
- Textual features (customer messages, inquiries)

The final feature vector can be expressed as:

$$x = [x_{demo}, x_{behavior}, x_{transaction}, x_{text}]$$

This multimodal structure motivates the use of deep neural architectures.

2.19.3 Neural Classification Architecture

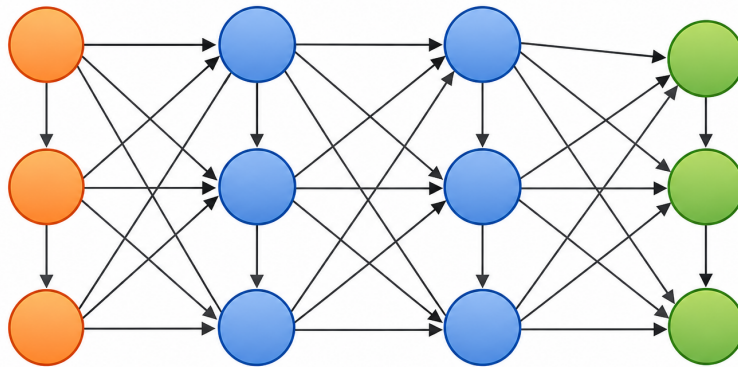


Figure 2.6: Fully Connected Neural Network for Classification

A common architecture consists of:

- Input layer (feature vector)
- Multiple hidden layers

- Output layer with sigmoid activation

The output probability is given by:

$$\hat{y} = \sigma(W_L h_{L-1} + b_L)$$

where σ is the sigmoid function.

2.19.4 Loss Function for Binary Classification

Binary Cross-Entropy Loss:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

This loss penalizes incorrect probability estimates.

2.20 Evaluation Metrics for Prospect Models

Accurate evaluation is essential in marketing contexts, where decisions directly impact revenue.

2.20.1 Confusion Matrix

	Predicted 1	Predicted 0
Actual 1	TP	FN
Actual 0	FP	TN

Where:

- TP = True Positives
- FP = False Positives
- FN = False Negatives
- TN = True Negatives

2.20.2 Key Metrics

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

$$Precision = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

In marketing applications, recall is often prioritized to avoid missing high-value prospects.

2.20.3 ROC Curve and AUC

The ROC curve evaluates the trade-off between:

- True Positive Rate
- False Positive Rate

The Area Under the Curve (AUC) measures overall model discrimination ability.

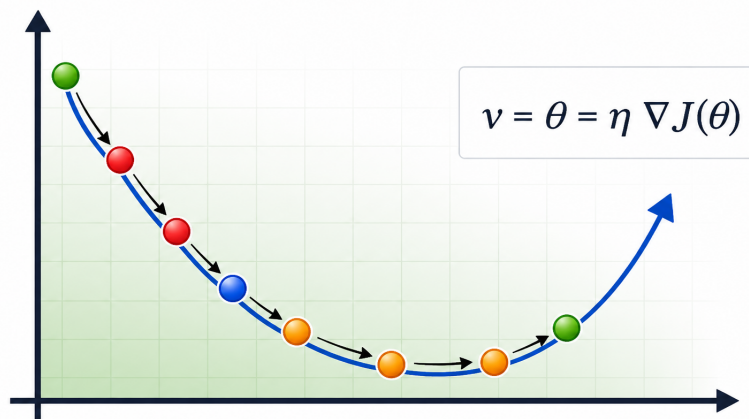
2.21 Optimization Strategies for Marketing Models

Figure 2.7: Gradient Descent Optimization Process

Marketing models must be optimized not only for accuracy but also for business performance.

2.21.1 Class Imbalance Handling

Prospect datasets are often imbalanced.

Solutions:

- Class weighting
- Oversampling (SMOTE)
- Threshold adjustment

2.21.2 Threshold Optimization

Instead of default threshold 0.5:

$$\hat{y} = \begin{cases} 1 & \text{if } p \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

Threshold τ can be tuned based on marketing cost-benefit analysis.

2.21.3 Business-Oriented Optimization

Expected revenue:

$$ER = TP \cdot R - FP \cdot C$$

Where:

- R = revenue per conversion
- C = campaign cost per prospect

Thus, model selection should maximize business utility rather than raw accuracy.

2.22 Integration of Textual Intelligence in Prospect Scoring

When textual data is included:

$$z = [x_{structured}; h_{text}]$$

where h_{text} is the embedding output of an NLP model.

This fusion significantly enhances predictive power.

2.23 Practical Relevance for Digital Marketing Systems

The theoretical concepts presented in this chapter are directly applicable to real-world digital marketing systems. In particular, neural networks and NLP models enable the automation of prospect classification tasks, which are essential for optimizing marketing campaigns.

By combining structured customer data with textual information extracted from emails or user interactions, modern AI systems can generate accurate prospect scores. These scores support decision-making processes such as lead prioritization, campaign targeting, and resource allocation.

Furthermore, the integration of these models into interactive systems, such as user interfaces or decision-support platforms, enhances usability and operational efficiency. This demonstrates the transition from purely theoretical frameworks to deployable intelligent systems capable of delivering tangible business value[11].

2.24 Conclusion

This chapter provided a comprehensive overview of neural networks and Natural Language Processing, highlighting their theoretical foundations and practical relevance in digital marketing applications. The discussion covered the mathematical principles of neural networks, optimization techniques, advanced architectures, and NLP methods for processing and analyzing textual data.

These concepts establish a solid framework for understanding how intelligent systems can effectively handle complex and heterogeneous data sources, including both structured features and natural language inputs.

Furthermore, the combination of neural networks and NLP techniques enables the development of advanced prospect classification models capable of capturing user behavior patterns and improving the accuracy of marketing decisions.

In addition, the integration of multiple data sources within a unified learning framework highlights the importance of hybrid approaches in addressing real-world challenges characterized by data diversity and complexity.

Overall, this chapter lays the scientific and technical foundation necessary for the design and implementation of an intelligent prospect classification system, which will be presented and evaluated in the subsequent chapters.

Chapter 3

Proposed System Architecture and Design for Intelligent Prospect Classification

3.1 Introduction

Following the theoretical foundations presented in the previous chapters, this chapter focuses on the design and implementation of the proposed intelligent system for prospect classification in digital marketing.

The objective is to develop a complete end-to-end solution capable of analyzing heterogeneous data sources, including structured behavioral data and unstructured textual data extracted from user interactions. The system is designed to automatically identify and classify potential customers based on their level of interest, thereby improving targeting efficiency and supporting decision-making processes[11].

Unlike traditional approaches that rely on a single type of data, the proposed system adopts a hybrid architecture that combines multiple data sources within a unified framework. This integration enables a more accurate representation of user behavior and enhances the overall performance of the classification process.

The chapter presents a detailed description of the system from different perspectives. It begins with the problem definition and a review of related work, followed by a system-level perspective including UML diagrams. It then introduces the learning paradigm and design choices, before presenting the proposed hybrid architecture and the neural network model used for classification.

Finally, a critical discussion is provided to analyze the strengths and limitations of the proposed solution.

3.2 Problem Definition

In digital marketing environments, organizations interact with a large number of potential customers through various communication channels, particularly email campaigns. One of the main challenges lies in identifying which prospects are genuinely interested in a product or service among a vast pool of users.

Traditional approaches to prospect classification often rely on simplistic criteria, such as counting email opens or link clicks, or on manual analysis of user responses. These methods are limited in their ability to capture the complexity of user behavior[18] and often lead to inaccurate classifications, inefficient targeting, and missed business opportunities.

Moreover, user data in such environments is inherently heterogeneous. It consists of structured behavioral data, such as interaction logs (opens, clicks), as well as unstructured textual data derived from email responses. Handling and integrating these different types of data remains a significant challenge.

Another key limitation is the lack of real-time and adaptive systems capable of continuously learning from new data. Most existing solutions operate in a static manner, without

updating their models based on evolving user interactions and feedback.

Therefore, the problem addressed in this work can be formulated as follows:

How to design an intelligent system capable of accurately classifying prospects by effectively combining heterogeneous data sources, including behavioral signals and textual information, while adapting dynamically to new data in real-world marketing environments?

Addressing this problem requires the development of a hybrid approach that integrates data fusion, machine learning, and dynamic learning mechanisms in order to improve classification accuracy and support decision-making processes.

3.3 Prospect classification and recent trends

Prospect classification, or lead scoring, estimates and ranks potential customers by their likelihood to engage or convert [2]. Early operational systems relied on hand-crafted rules and simple interaction metrics (for example, counts of email opens, clicks, and page views). Such rule-based approaches are easy to implement but often fail to capture complex or time-dependent behavior patterns [8].

More recent work adopts machine learning to predict conversion from historical data. Traditional supervised algorithms—logistic regression, decision trees, and gradient-boosted machines—improved predictive accuracy while retaining interpretability for business users [3, 2]. Advances in deep learning and Natural Language Processing (NLP) have further enhanced performance by modeling non-linear relationships and extracting semantic signals from text sources such as emails, chat logs, and reviews [26, 5].

Current research and industry practice emphasize hybrid, multimodal models that combine structured behavioral data with unstructured textual inputs; these approaches typically outperform single-source baselines in both experiments and production deployments [16, 12]. Other active directions include online or streaming scoring for low-latency predictions, transfer learning to adapt models across campaigns, and causal/uplift methods to measure the true business impact of targeting [27, 8].

Key challenges remain: enabling real-time adaptation and continual learning, fusing heterogeneous data at scale, ensuring label quality and avoiding feedback loops, and delivering interpretable, privacy-aware scores that satisfy business and regulatory constraints [8, 27]. For applied lead-scoring systems, hybrid and modular solutions that balance predictive power, scalability, and interpretability are practical and well-motivated directions

3.4 System Perspective

The proposed solution is designed as a complete intelligent system that integrates data collection, processing, and predictive modeling within a unified framework. Unlike traditional

approaches that rely on static datasets, the system operates in a dynamic environment where user interactions are continuously captured and analyzed.

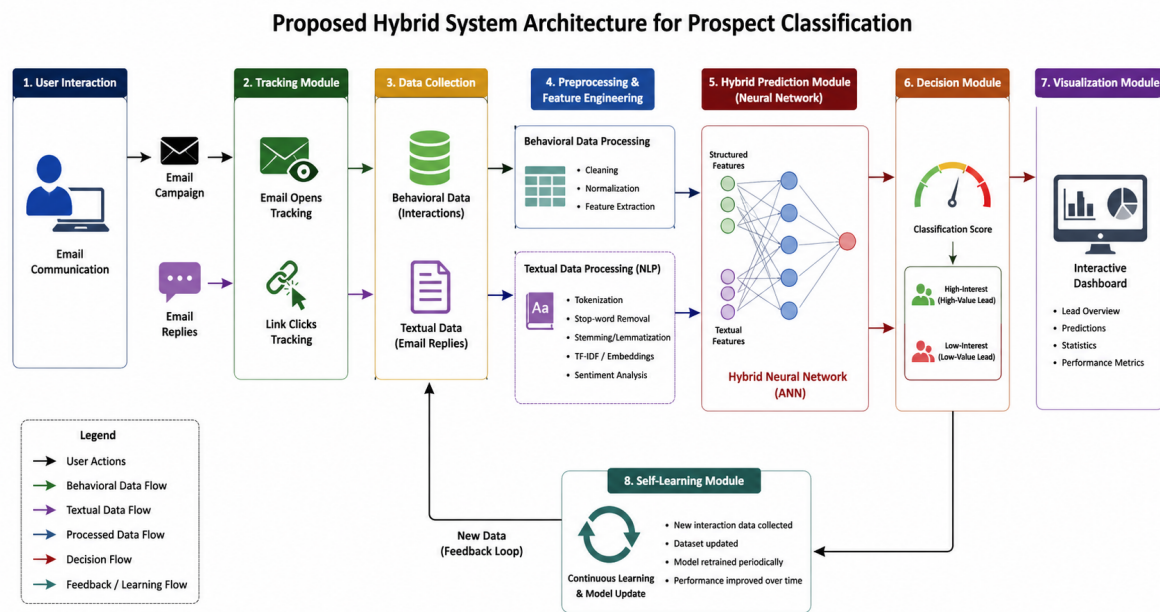


Figure 3.1: Proposed hybrid system architecture for prospect classification

The system is composed of several interconnected modules:

- **User Interaction Module:** Users interact with the system through email communication. These interactions include email openings, link clicks, and textual responses.
- **Tracking Module:** A real-time tracking mechanism records user behavior such as email opens and link clicks. This data is stored in a structured format for further analysis.
- **Data Collection Module:** The system collects both behavioral data (interaction logs) and textual data extracted from email replies, forming a heterogeneous dataset.
- **Preprocessing and Feature Engineering:** Collected data is cleaned and transformed into meaningful features. Textual data is processed using Natural Language Processing techniques, while behavioral data is normalized and structured.
- **Hybrid Prediction Module:** A machine learning model based on neural networks is used to analyze both structured and textual features. The model captures complex relationships between user behavior and expressed interest.
- **Decision Module:** The system generates a classification output indicating the level of user interest (e.g., high-interest or low-interest). Additional decision logic may be applied to refine predictions.

- **Visualization Module:** Results are displayed through an interactive dashboard, enabling users to analyze predictions and monitor system performance.
- **Self-Learning Module:** The system continuously updates its dataset using newly collected interaction data, allowing the model to improve over time.

This system perspective highlights the transition from isolated predictive models to a fully integrated intelligent system capable of operating in real-world digital marketing environments. It emphasizes the importance of combining data collection, real-time tracking, hybrid modeling, and continuous learning to enhance prospect classification performance.

3.5 UML Modeling of the Proposed System

To provide a comprehensive understanding of the system structure and its interactions, UML modeling is used to describe both the functional and structural aspects of the proposed intelligent system. This section presents the Use Case Diagram and the Class Diagram, highlighting the system behavior and internal architecture.

3.5.1 UML Use Case Diagram

The use case diagram illustrates the interactions between external actors and the system functionalities. It provides a high-level view of how users interact with the system.

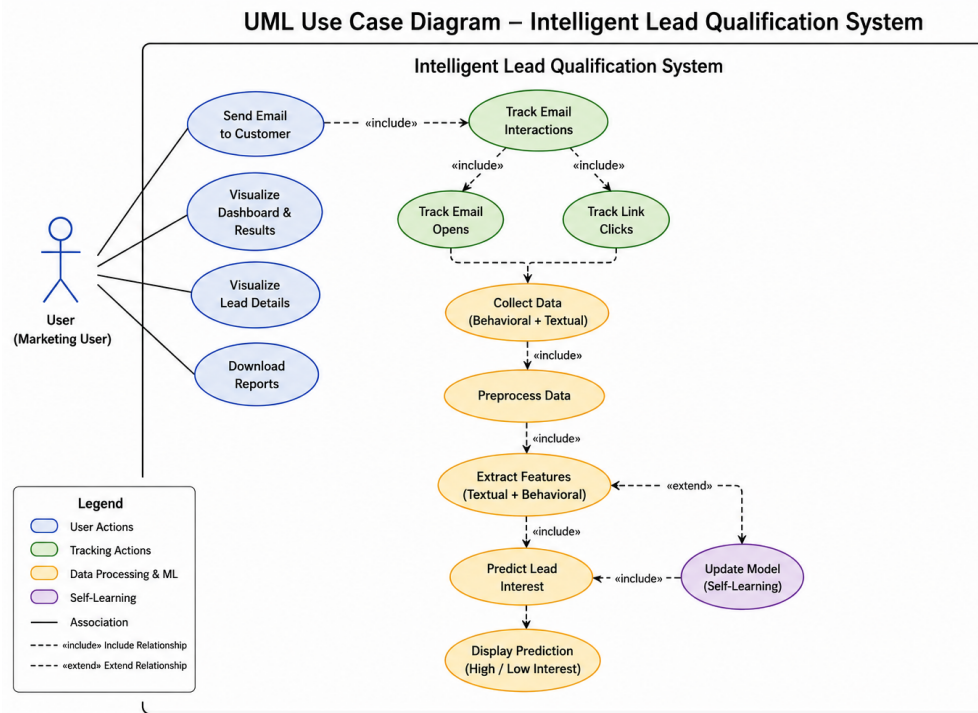


Figure 3.2: UML Use Case Diagram of the Proposed Hybrid System

The system involves two main actors:

- **User:** interacts with the system by sending emails and viewing prediction results through the dashboard.
- **Administrator:** monitors system performance, manages configurations, and supervises model updates.

The main use cases are:

- Send email / interact with campaign
- Track user interactions (opens, clicks)
- Extract email responses
- Preprocess data (text(Replies) + behavioral(opens,clicks))
- Generate features
- Predict user interest
- Display results in dashboard
- Update system (self-learning)

This diagram highlights the functional workflow of the system and emphasizes its dynamic and interactive nature.

3.5.2 UML Class Diagram

The class diagram represents the internal structure of the system by defining its main components and their relationships.

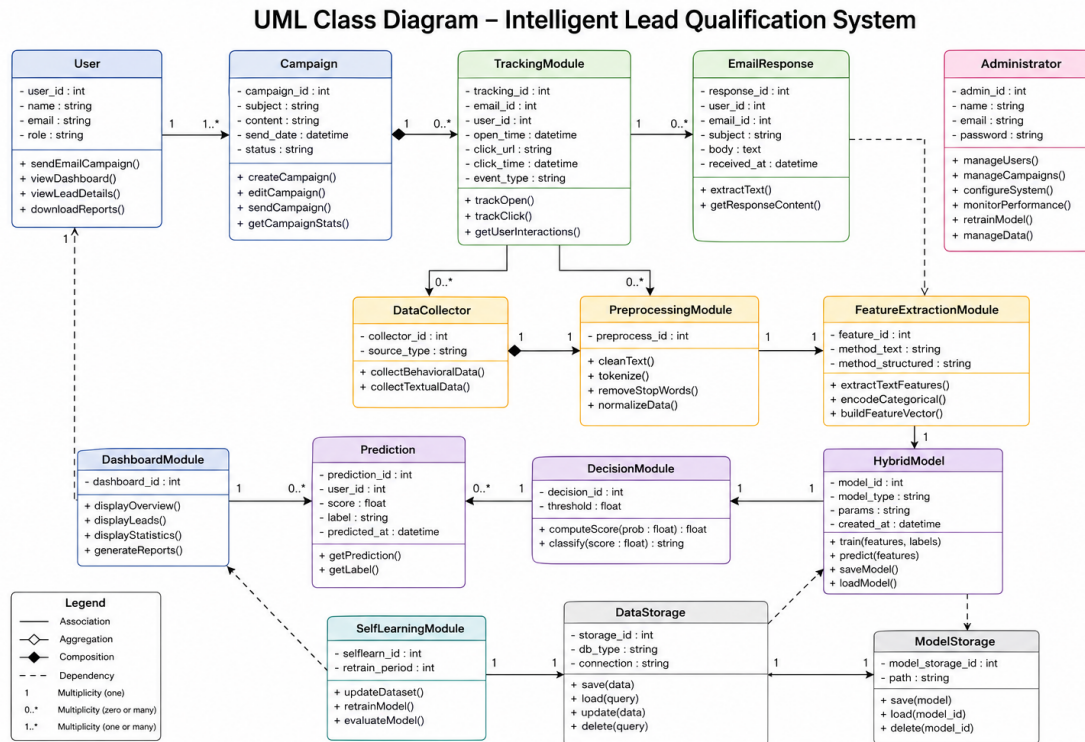


Figure 3.3: UML Class Diagram of the Hybrid Intelligent System

The system is composed of the following core classes:

- **User Interface**

- Handles interaction with the user
- Displays predictions and analytics

- **Tracking Module**

- Records email opens and link clicks
- Stores behavioral interaction data

- **Email Processing Module**

- Extracts textual data from email responses
- Prepares raw text for analysis

- **Preprocessing Module**

- Cleans text data
- Normalizes structured data

- **Feature Engineering Module**

- Generates structured features
- Encodes textual data into numerical representations
- **Hybrid Model**
 - Neural network for classification
 - Combines behavioral and textual inputs
- **Decision Module**
 - Computes prediction score
 - Applies decision logic for classification
- **Dashboard Module**
 - Visualizes results and system performance
- **Self-Learning Module**
 - Updates dataset with new interactions
 - Retrains model periodically

This structural representation reflects the modular design of the system and highlights the interaction between different components, ensuring scalability, maintainability, and real-time adaptability.

3.6 Learning Paradigm and Design Choices

The proposed system adopts a supervised learning paradigm, motivated by the availability of labeled data and the need to perform predictive classification of prospects based on their behavior and interactions.

A hybrid neural network architecture is selected as the core modeling approach. This architecture combines sequence-based modeling for textual data with dense layers for structured behavioral features. In particular, Long Short-Term Memory (LSTM) layers are used to capture patterns in textual data, while fully connected layers process structured inputs such as email opens, clicks, and interaction frequency.

This design enables the system to effectively learn complex and non-linear relationships between heterogeneous data sources.

The choice of this architecture is justified by several advantages:

- Ability to capture sequential patterns in textual data using LSTM layers

- Capability to model non-linear relationships in structured behavioral data
- Flexibility in integrating multiple data modalities within a unified framework

However, designing an effective model requires careful consideration of several factors:

- Data preprocessing strategies to ensure clean and consistent inputs
- Feature representation techniques for both textual and structured data
- Selection of network architecture parameters (layers, neurons)
- Choice of activation functions and optimization methods
- Regularization techniques to prevent overfitting

Transformer-based models such as BERT are also incorporated in the system, but only at the decision level for sentiment analysis. This allows the system to benefit from contextual language understanding without introducing significant computational overhead in the main classification model.

Therefore, a balance between model performance and computational efficiency guided the design choices, resulting in a hybrid architecture that is both effective and suitable for real-time applications.

3.7 Proposed Hybrid Architecture

The proposed system is based on a hybrid architecture designed to process heterogeneous data sources through two complementary pipelines: a textual data pipeline and a behavioral data pipeline. This design enables the model to combine semantic information extracted from user responses with behavioral interaction patterns.

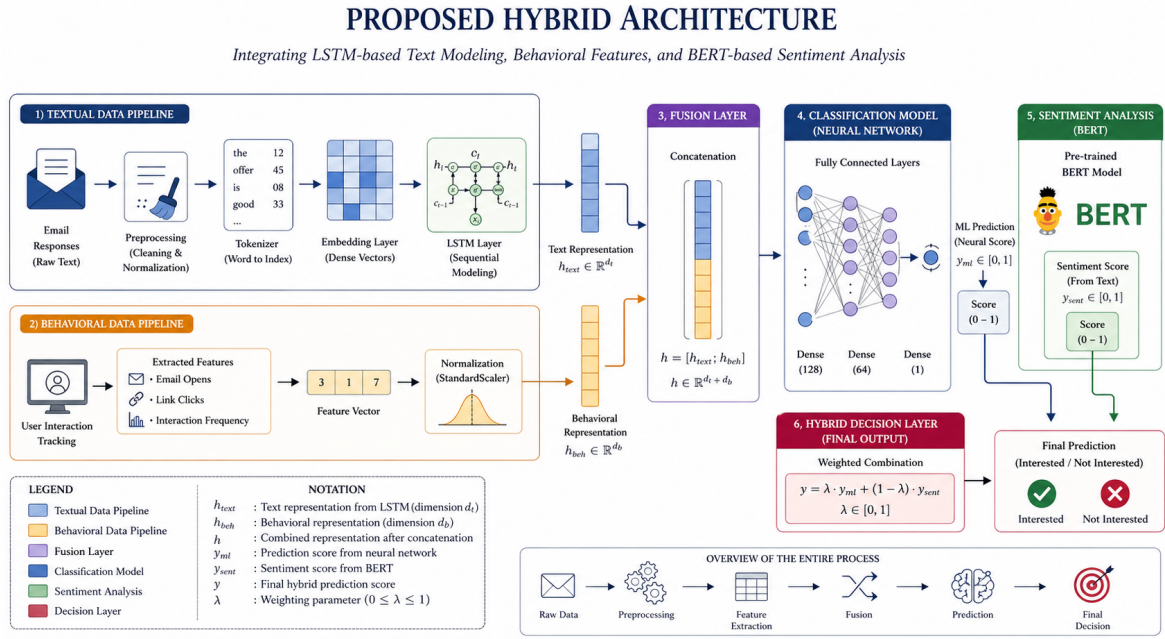


Figure 3.4: Proposed Hybrid Architecture combining LSTM-based text modeling, behavioral features, and BERT-based sentiment analysis

3.7.1 Textual Data Processing

Textual data, corresponding to email responses, is first preprocessed through cleaning and normalization steps. The cleaned text is then converted into numerical sequences using a tokenizer, followed by padding to ensure a fixed input length.

The processed sequences are passed through an embedding layer, which transforms words into dense vector representations. These embeddings are then fed into a Long Short-Term Memory (LSTM) layer, allowing the model to capture sequential dependencies and contextual information from textual data.

3.7.2 Behavioral Data Processing

Behavioral data is derived from user interaction logs, including email opens, link clicks, and overall interaction frequency. These features are numerical and are directly used as structured inputs after normalization.

This pipeline enables the model to capture explicit user engagement patterns, which are essential for predicting user interest.

3.7.3 Feature Fusion Mechanism

The outputs of the textual and behavioral pipelines are merged using a concatenation layer:

$$z = [h_{\text{text}} \parallel h_{\text{behavioral}}]$$

This fusion allows the model to jointly learn from both semantic and behavioral representations, resulting in a richer feature space.

3.7.4 Classification Model

The fused feature vector is processed through a series of fully connected layers, which learn complex relationships between textual intent and user behavior. The final layer uses a sigmoid activation function to output a probability score representing the likelihood of user interest.

3.7.5 Hybrid Decision Mechanism

In addition to the neural network prediction, a sentiment analysis module based on a pre-trained BERT model is incorporated [5] at the decision level. This component analyzes the sentiment of textual responses and produces a sentiment score.

The final prediction is obtained by combining the neural network output with the sentiment score through a hybrid decision logic, improving the robustness and accuracy of the system.

3.7.6 Mathematical Formulation

Let:

- x_t denote the textual input
- x_b denote the behavioral input

The feature extraction process is defined as:

$$h_t = f_{\text{LSTM}}(x_t)$$

$$h_b = f_{\text{struct}}(x_b)$$

The fusion is defined as:

$$h = [h_t \parallel h_b]$$

The neural network prediction is:

$$y_{ml} = \sigma(W \cdot h + b)$$

The sentiment score is:

$$y_{sent} = f_{\text{BERT}}(x_t)$$

The final prediction is computed as a combination of both:

$$y = \lambda \cdot y_{ml} + (1 - \lambda) \cdot y_{sent}$$

where λ is a weighting parameter that balances the contribution of the machine learning model and the sentiment analysis component.

This formulation reflects the hybrid nature of the system and its ability to integrate multiple sources of information within a unified predictive framework.

3.8 Model Architecture (Hybrid Neural Network)

The proposed model is based on a hybrid neural network architecture designed to process heterogeneous inputs through two parallel branches: a textual branch and a behavioral branch.

The textual branch processes email responses using a sequence-based approach. The input text is first transformed into numerical sequences using a tokenizer, then passed through an embedding layer. These embeddings are processed by a Long Short-Term Memory (LSTM) layer, which captures sequential patterns and contextual information from the text. The output is then refined through dense layers.

In parallel, the behavioral branch processes structured features such as email opens, clicks, and interaction frequency. These inputs are fed into fully connected layers to extract meaningful patterns related to user engagement.

The outputs of both branches are merged using a concatenation layer, forming a unified feature representation. This fusion enables the model to jointly learn from both semantic (textual) and behavioral information.

The combined features are then passed through additional dense layers that perform non-linear transformations, allowing the model to learn complex relationships between inputs.

Finally, the model produces a probabilistic output using a sigmoid activation function, representing the likelihood of user interest.

The architecture can be summarized as follows:

- Text branch: Embedding $\rightarrow$ LSTM $\rightarrow$ Dense
- Behavioral branch: Dense layers
- Fusion: Concatenation layer
- Classification: Fully connected layers $\rightarrow$ Sigmoid output

```

input_struct = Input(shape=(4,))
x1 = Dense(32, activation='relu')(input_struct)

input_text = Input(shape=(20,))
x2 = Embedding(5000, 32)(input_text)
x2 = Flatten()(x2)

combined = Concatenate()([x1, x2])

z = Dense(32, activation='relu')(combined)
z = Dense(16, activation='relu')(z)

output = Dense(1, activation='sigmoid')(z)

```

Figure 3.5: Hybrid neural network architecture

This design allows efficient integration of heterogeneous data sources and provides a flexible and scalable framework for prospect classification.

3.9 Critical Discussion

The proposed hybrid system demonstrates strong capabilities in handling heterogeneous data sources and improving prospect classification accuracy. By combining behavioral data with textual analysis, the system provides a more comprehensive understanding of user intent compared to traditional single-source approaches.

One of the main strengths of the system lies in its hybrid design. The integration of sequence-based text modeling using LSTM with structured behavioral features allows the model to capture both implicit and explicit user interactions. This significantly enhances the robustness of the classification process.

Furthermore, the incorporation of a BERT-based sentiment analysis module at the decision level provides an additional layer of intelligence. This component helps refine predictions, especially in ambiguous or contradictory cases where behavioral signals alone may not be sufficient.

Another important advantage is the self-learning mechanism, which enables the system to continuously update its dataset and retrain the model based on new interactions. This ensures adaptability to evolving user behavior and improves long-term performance.

However, the system also presents several limitations. First, the reliance on labeled data may affect performance when dealing with noisy or insufficient datasets. Although the self-learning module mitigates this issue, the quality of automatically labeled data remains a critical factor.

Second, while the use of LSTM improves textual understanding, it may not capture long-range dependencies as effectively as more advanced Transformer-based architectures. However, adopting such models would significantly increase computational cost and complexity.

Additionally, the hybrid decision mechanism introduces additional complexity in bal-

ancing the contribution of machine learning predictions and sentiment analysis. Improper weighting may lead to biased results, especially in edge cases.

Finally, the system depends on the quality of tracking data. Inaccurate or incomplete interaction logs may negatively impact the performance of the behavioral component.

Overall, the proposed system achieves a balanced trade-off between performance, interpretability, and computational efficiency. It provides a practical and scalable solution for real-world digital marketing applications, while leaving room for future improvements such as the integration of more advanced language models and optimization of decision strategies.

3.10 Conclusion

This chapter presented the design and architecture of the proposed hybrid intelligent system for prospect classification in digital marketing environments. The focus was placed on transitioning from theoretical concepts to a practical system capable of handling real-world data and interactions.

A comprehensive system perspective was introduced, highlighting the integration of multiple components including user interaction tracking, data preprocessing, feature engineering, and predictive modeling. UML diagrams were used to illustrate both the functional and structural aspects of the system.

The proposed hybrid architecture was then detailed, demonstrating how textual data is processed using a sequence-based approach with LSTM layers, while behavioral data is handled through structured feature modeling. The fusion of these heterogeneous inputs enables the system to capture both semantic intent and user engagement patterns.

In addition, a hybrid decision mechanism was incorporated by integrating a BERT-based sentiment analysis module, which refines the final prediction and enhances classification accuracy.

The design choices were guided by the need to balance performance and computational efficiency, resulting in a scalable and adaptable solution. The system also incorporates a self-learning mechanism, allowing continuous improvement through the integration of new data.

Overall, this chapter established a complete and coherent framework for the proposed system, providing the necessary foundation for its implementation and experimental evaluation, which are presented in the next chapter.

Chapter 4

Implementation and Experimental Evaluation

4.1 Introduction

This chapter presents the practical implementation and experimental evaluation of the proposed hybrid intelligent system for prospect classification. Building upon the system design and architecture described in the previous chapter, this section focuses on translating the theoretical model into a fully functional system.

The implementation integrates multiple components, including data preprocessing, feature engineering, model training, and real-time prediction. The system is designed to handle heterogeneous data sources, combining textual data from email responses with structured behavioral data such as email opens, clicks, and interaction frequency.

This chapter provides a detailed description of the development environment, hardware configuration, and the tools used to build the system. It also presents the model architecture alongside the dataset construction and preprocessing steps, ensuring a clear understanding of how the data flows through the system.

Furthermore, the training process and evaluation metrics are discussed to assess the performance of the model. The implementation of the complete system, including tracking mechanisms, prediction modules, and decision logic, is then described.

Finally, the results obtained from experiments are analyzed to evaluate the effectiveness of the proposed hybrid approach in real-world scenarios[11].

4.2 Hardware Configuration

The implementation and execution of the proposed system were carried out on a standard personal computer. The hardware configuration used is summarized as follows:

- **Processor (CPU):** Intel Core i7 (11th Generation)
- **Memory (RAM):** 16 GB
- **Storage:** 512 GB SSD, providing sufficient space for datasets and model files
- **Operating System:** Windows 11
- **Graphics Processing Unit (GPU):** Intel Iris Xe Graphics

This configuration demonstrates that the proposed system does not require high-end computational resources and can be effectively implemented on standard machines. Although no dedicated GPU acceleration is used, the training and execution processes remain efficient due to the moderate size of the dataset and the optimization capabilities of the employed machine learning libraries.

Overall, this highlights the practicality and accessibility of the proposed solution for real-world deployment scenarios.

4.3 Development Environment

The development of the proposed system was carried out using a set of modern tools and technologies that support data processing, machine learning, and system integration.

4.3.1 Programming Language

The system was implemented using **Python**, which provides a flexible and efficient environment for building data-driven applications and machine learning models.

4.3.2 Development Tools

The implementation was conducted using the **PyCharm** integrated development environment (IDE), offering advanced features for coding, debugging, and project organization.

4.3.3 Libraries and Frameworks

Several Python libraries and frameworks were utilized in this project:

- **Pandas**: used for data manipulation and dataset handling.
- **NumPy**: used for numerical operations and data processing.
- **Scikit-learn**[22], **Pandas** [19]: used for preprocessing tasks such as feature scaling (StandardScaler).
- **TensorFlow / Keras**[24, 14]: used for building and training the hybrid neural network model (LSTM and dense layers).
- **Transformers (Hugging Face)**: used to implement BERT-based sentiment analysis.
- **Streamlit**[23]: used to develop the interactive dashboard for visualization and analysis.
- **Flask**[9]: used to implement the tracking system for capturing user interactions (opens and clicks).
- **IMAP / Email libraries**: used for retrieving and processing email responses.
- **SMTP (smtplib)**: used for sending email campaigns.

4.4 System Architecture Overview

The system is composed of several interconnected modules that form a complete processing pipeline:

- **Tracking Module:** records user interactions such as email opens and link clicks.
- **Email Processing Module:** retrieves and cleans user responses from email servers.
- **Feature Engineering Module:** extracts structured behavioral features and prepares textual data.
- **Machine Learning Module:** performs classification using a hybrid neural network combining LSTM and dense layers.
- **Decision Module:** integrates machine learning predictions with BERT-based sentiment analysis.
- **Self-Learning Module:** updates the dataset and retrains the model dynamically.
- **Dashboard Module:** visualizes results and provides an interface for analysis.

These components work together to ensure a complete end-to-end system, from data collection and preprocessing to prediction, decision-making, and visualization.

4.5 Model Architecture

This section presents the dataset construction, preprocessing techniques, and the architecture of the proposed hybrid neural network model.

4.5.1 Dataset Construction

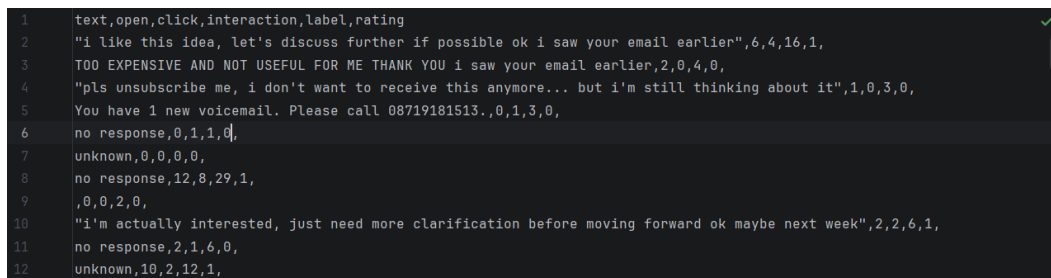
The dataset used in this work is a hybrid multi-source dataset combining synthetic data, publicly available datasets, and real-world interaction data.

The dataset is composed of:

- Synthetic generated data simulating email marketing scenarios
- Textual datasets from Kaggle[1, 6]:
 - `spam.csv` (SMS Spam Collection)
 - `Amazon_Reviews.csv`
 - `enron.csv`

- Behavioral datasets:
 - Online Shoppers Intention Dataset[7]
 - Digital Marketing Campaign Dataset
 - Marketing Campaign Dataset
 - Ad Click Prediction Dataset (CTR)
- Real system data:
 - logs.csv (tracking opens and clicks)
 - Email replies extracted via IMAP
- Self-learning data:
 - new_data.csv

The final dataset combines all these sources to improve diversity and robustness.



```

1 text,open,click,interaction,label,rating
2 "i like this idea, let's discuss further if possible ok i saw your email earlier",6,4,16,1,
3 TOO EXPENSIVE AND NOT USEFUL FOR ME THANK YOU i saw your email earlier,2,0,4,0,
4 "pls unsubscribe me, i don't want to receive this anymore... but i'm still thinking about it",1,0,3,0,
5 You have 1 new voicemail. Please call 08719181513.,0,1,3,0,
6 no response,0,1,1,0,
7 unknown,0,0,0,0,
8 no response,12,8,29,1,
9 ,0,0,2,0,
10 "i'm actually interested, just need more clarification before moving forward ok maybe next week",2,2,6,1,
11 no response,2,1,6,0,
12 unknown,10,2,12,1,

```

Figure 4.1: Sample of the hybrid dataset

4.5.2 Data Preprocessing

Textual data is cleaned and transformed into numerical sequences using a tokenizer[4]. Padding is applied to ensure fixed input length.

```
tokenizer = Tokenizer(num_words=10000, oov_token="<OOV>")
tokenizer.fit_on_texts(data["text"])
```

```
X_text = tokenizer.texts_to_sequences(data["text"])
X_text = pad_sequences(X_text, maxlen=60)
```



```

53 tokenizer = Tokenizer(num_words=10000, oov_token="<OOV>")
54 tokenizer.fit_on_texts(data["text"])
55
56 with open("tokenizer.pkl", "wb") as f:
57     pickle.dump(tokenizer, f)
58
59 X_text = tokenizer.texts_to_sequences(data["text"])
60
61 MAX_LEN = 60
62 X_text = pad_sequences(X_text, maxlen=MAX_LEN)

```

Figure 4.2: Text tokenization and sequence transformation

Structured features are normalized using StandardScaler:

```
scaler = StandardScaler()
X_struct = scaler.fit_transform(X_struct)
```

```
19 # ===== SCALER =====
20 scaler = StandardScaler()
21 X_struct = scaler.fit_transform(X_struct)
```

Figure 4.3: Feature scaling of behavioral data

4.5.3 Model Design

The model is a hybrid neural network composed of two parallel branches:

Text Branch: Embedding $\rightarrow$ LSTM $\rightarrow$ Dense $\rightarrow$ Dropout

Behavioral Branch: Dense $\rightarrow$ Dropout

The outputs are merged and passed through classification layers.

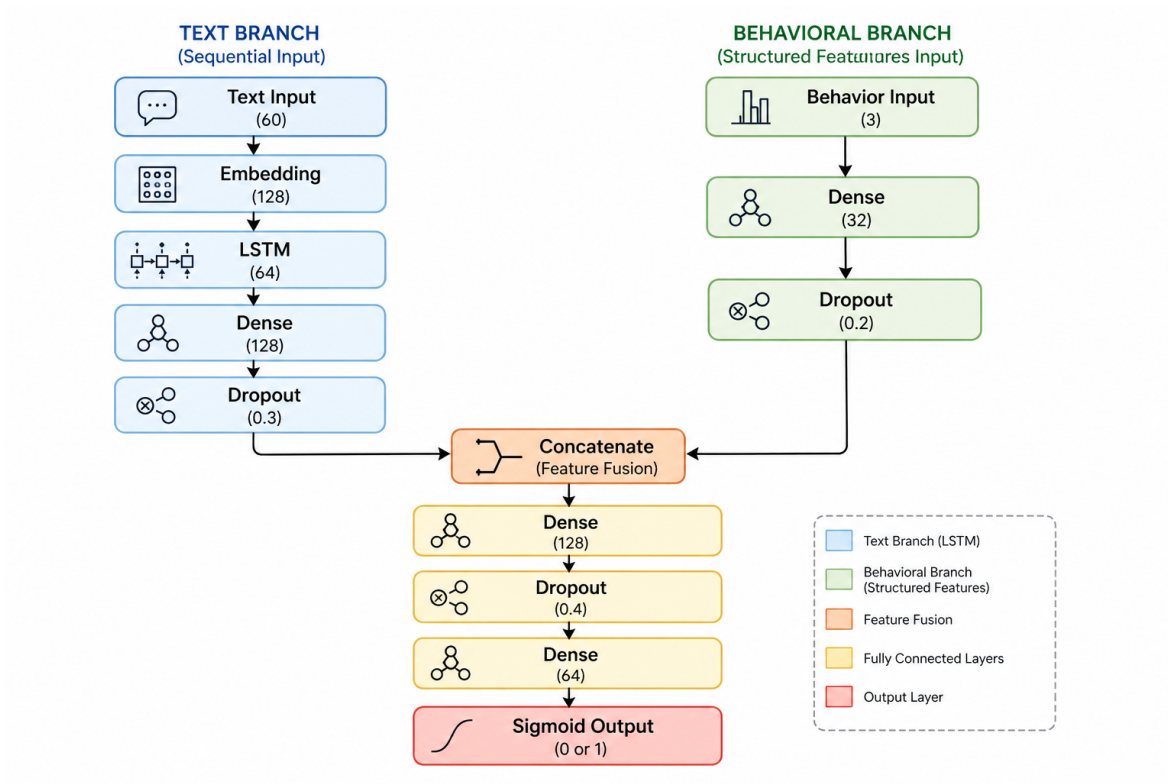


Figure 4.4: Hybrid neural network architecture

4.5.4 Implementation Details

This subsection provides a detailed implementation of the proposed hybrid neural network using the Keras Functional API. The model is designed as a multi-input architecture that processes heterogeneous data sources through two specialized branches.

The first branch is dedicated to textual data and is based on sequence modeling. The input text, represented as padded sequences of fixed length (60 tokens), is passed through an embedding layer that transforms each token into a dense vector representation. This is followed by an LSTM layer, which captures temporal dependencies and contextual relationships within the sequence. A fully connected layer is then applied to refine the extracted semantic features, followed by a dropout layer to improve generalization and reduce overfitting.

The second branch processes structured behavioral features, including user interaction signals such as email opens, clicks, and engagement levels. These numerical features are fed into a dense layer to learn meaningful patterns, followed by a dropout layer for regularization.

The outputs of both branches are merged using a concatenation layer, forming a unified feature representation that combines both semantic and behavioral information. This fused representation is then passed through multiple fully connected layers that perform nonlinear transformations, enabling the model to learn complex relationships between inputs.

Finally, a sigmoid activation function is applied to produce a probabilistic output representing the likelihood of user interest.

The following code snippet illustrates the implementation of this architecture:

Listing 4.1: Implementation of the Hybrid Neural Network Model

```
# Text branch

input_text = Input(shape=(60,))
x1 = Embedding(10000, 128)(input_text)
x1 = LSTM(64)(x1)
x1 = Dense(128, activation="relu")(x1)
x1 = Dropout(0.3)(x1)

# Behavioral branch

input_struct = Input(shape=(3,))
x2 = Dense(32, activation="relu")(input_struct)
x2 = Dropout(0.2)(x2)

# Fusion layer

combined = Concatenate()([x1, x2])
```

```
# Classification layers
```

```
z = Dense(128, activation="relu")(combined)
```

```
z = Dropout(0.4)(z)
```

```
z = Dense(64, activation="relu")(z)
```

```
# Output layer
```

```
output = Dense(1, activation="sigmoid")(z)
```

```
model = Model(inputs=[input_text, input_struct], outputs=output)
```

This implementation demonstrates the effectiveness of combining sequence-based learning for textual data with structured feature modeling, resulting in a flexible and scalable architecture suitable for real-world marketing applications.

4.5.5 Discussion

The proposed hybrid architecture effectively combines textual and behavioral data to provide a more comprehensive representation of user intent. The LSTM-based branch captures contextual information from text, while the behavioral branch models explicit user interactions such as opens and clicks.

The fusion of these two data sources enhances prediction accuracy and improves the robustness of the model compared to single-input approaches. In addition, the use of dropout layers helps reduce overfitting and improves generalization.

However, the model introduces additional complexity and depends on the quality of both textual and behavioral data. Despite these limitations, the architecture offers a good balance between performance and practicality, making it suitable for real-world marketing applications.

4.6 Model Training and Evaluation

This section presents the training process of the proposed hybrid neural network model, along with the evaluation metrics used to assess its performance.

4.6.1 Training Process

The model was trained using a supervised learning approach on the constructed hybrid dataset. The dataset was divided into training and validation sets to monitor generalization performance.

The training process was configured as follows:

- **Optimizer:** Adam
- **Loss Function:** Binary Cross-Entropy
- **Batch Size:** 32
- **Epochs:** 20
- **Validation Split:** 20

Early stopping was applied based on validation loss to prevent overfitting and ensure optimal model performance.

Listing 4.2: Model Training

```
model.compile(  
optimizer="adam",  
loss="binary_crossentropy",  
metrics=["accuracy", "Precision", "Recall"]  
)  
  
history = model.fit(  
[X_text_train, X_struct_train],  
y_train,  
validation_data=([X_text_val, X_struct_val], y_val),  
epochs=20,  
batch_size=32,  
callbacks=[early_stop]  
)
```

4.6.2 Evaluation Metrics

To evaluate the performance of the model, several metrics were used:

- **Accuracy:** measures the overall correctness of predictions
- **Precision:** evaluates the proportion of correctly predicted positive cases
- **Recall:** measures the ability of the model to detect positive instances

These metrics provide a comprehensive evaluation of the classification performance.

4.6.3 Training Results

The training and validation performance were monitored across epochs.

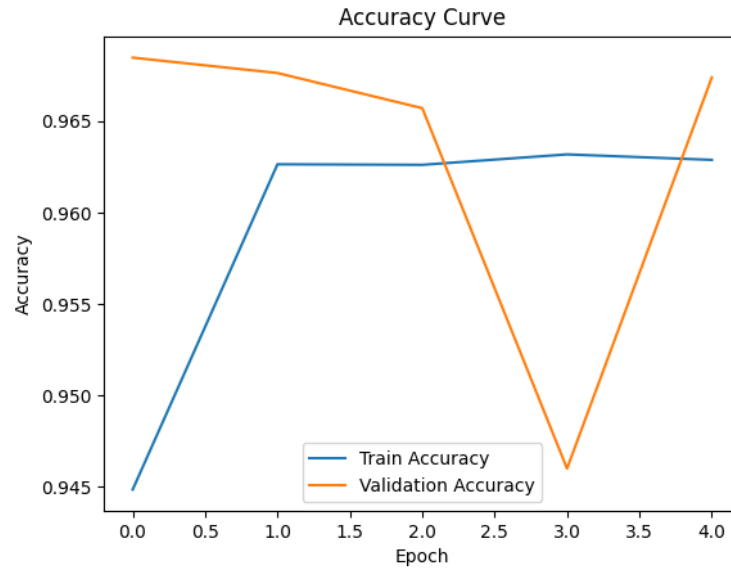


Figure 4.5: Training and validation accuracy over epochs

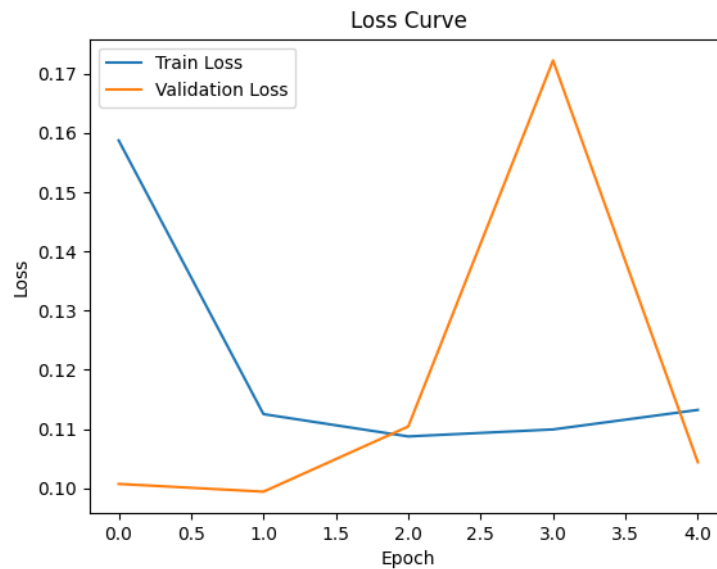


Figure 4.6: Training and validation loss over epochs

The results show that the model converges effectively, with a stable increase in accuracy and a decrease in loss over time.

4.6.4 Confusion Matrix

To further analyze the classification performance, a confusion matrix was used.

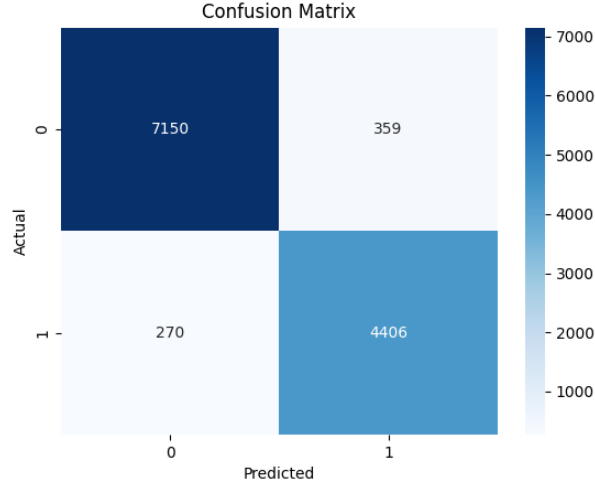


Figure 4.7: Confusion matrix of the model predictions

The confusion matrix highlights the model's ability to correctly classify both positive and negative instances, while also revealing potential misclassifications.

4.6.5 Discussion of Results

The experimental results demonstrate that the hybrid model achieves strong performance in prospect classification. The combination of textual and behavioral features contributes to improved accuracy and robustness.

The use of multiple evaluation metrics confirms the reliability of the model, while the confusion matrix provides deeper insights into prediction behavior.

Overall, the model shows good generalization capability and is suitable for real-world deployment scenarios.

4.7 System Implementation

This section presents the practical implementation of the proposed intelligent system for prospect classification. The system integrates multiple modules including email campaign management, user interaction tracking, feature engineering, and predictive modeling within a unified and interactive environment.

4.7.1 System Overview

The implemented system follows a modular and interactive pipeline. It enables the user to create and deploy email campaigns, track user behavior in real-time, and analyze engagement using a hybrid deep learning model.

The workflow consists of the following stages:

- Campaign creation and email sending
- Real-time interaction tracking (open and click)
- Feature extraction and preprocessing
- Prediction using the trained hybrid model
- Visualization of results through an interactive dashboard

4.7.2 Campaign Management Interface

A dedicated interface was developed to manage email campaigns. This module allows the user to configure and send emails, including:

- Sender email credentials
- Recipient list
- Email subject
- HTML email content with embedded tracking links

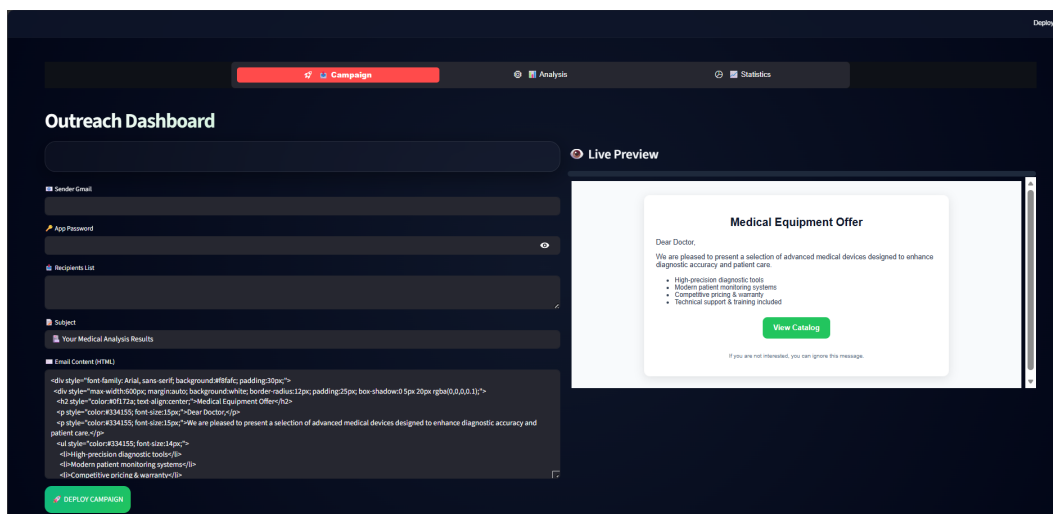


Figure 4.8: Campaign creation interface with live email preview

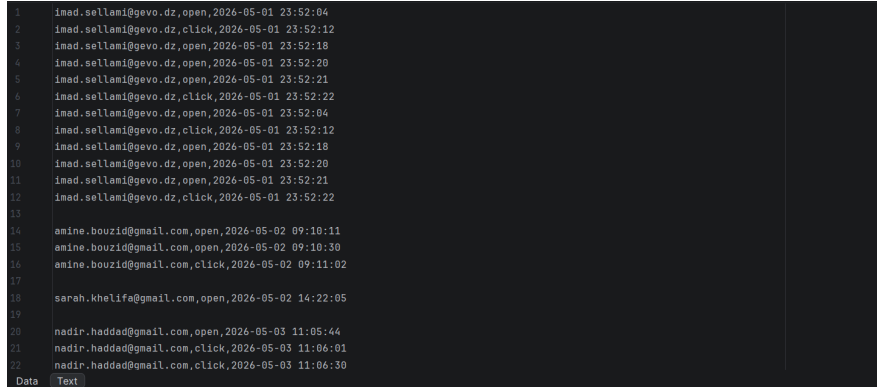
This interface provides a live preview of the email content, ensuring correctness before deployment.

4.7.3 Tracking System

A tracking mechanism was implemented using Flask to monitor user interactions. Each interaction is recorded as an event in the following format:

- Email address
- Event type (open or click)
- Timestamp

These events are stored in a log file (`logs.csv`), which captures real-time user behavior.



```

1  imad.sellami@gevo.dz,open,2026-05-01 23:52:04
2  imad.sellami@gevo.dz,click,2026-05-01 23:52:12
3  imad.sellami@gevo.dz,open,2026-05-01 23:52:18
4  imad.sellami@gevo.dz,open,2026-05-01 23:52:20
5  imad.sellami@gevo.dz,open,2026-05-01 23:52:21
6  imad.sellami@gevo.dz,click,2026-05-01 23:52:22
7  imad.sellami@gevo.dz,open,2026-05-01 23:52:04
8  imad.sellami@gevo.dz,click,2026-05-01 23:52:12
9  imad.sellami@gevo.dz,open,2026-05-01 23:52:18
10 imad.sellami@gevo.dz,open,2026-05-01 23:52:20
11 imad.sellami@gevo.dz,open,2026-05-01 23:52:21
12 imad.sellami@gevo.dz,click,2026-05-01 23:52:22
13
14 amine.bouzi@gmail.com,open,2026-05-02 09:10:11
15 amine.bouzi@gmail.com,open,2026-05-02 09:10:30
16 amine.bouzi@gmail.com,click,2026-05-02 09:11:02
17
18 sarah.khelifa@gmail.com,open,2026-05-02 14:22:05
19
20 nadir.haddad@gmail.com,open,2026-05-03 11:06:44
21 nadir.haddad@gmail.com,click,2026-05-03 11:06:01
22 nadir.haddad@gmail.com,click,2026-05-03 11:06:30

```

Figure 4.9: Event-based tracking data stored in `logs.csv`

This design allows fine-grained monitoring of user engagement and supports dynamic feature extraction.

4.7.4 Feature Engineering

The raw tracking events are processed to generate structured features required by the model. These include:

- Number of email opens
- Number of link clicks
- Interaction score (aggregated engagement)

Textual content from emails is cleaned, tokenized, and converted into padded sequences using a tokenizer. Structured features are normalized using a `StandardScaler`.

The combination of these features forms the hybrid input to the model.

4.7.5 Prediction Module

The prediction module uses the trained hybrid neural network, which consists of:

- An LSTM-based branch for processing textual data
- A dense network for structured behavioral features
- A fusion layer combining both representations

The model outputs a probability score representing user interest. Based on predefined thresholds, users are classified into categories such as High, Medium, or Low interest.

4.7.6 Analysis Dashboard

An interactive dashboard was developed using Streamlit to visualize the results of the system. It provides:

- A table of predictions including email, score, and interaction features
- Classification labels (High, Medium, Low)
- Summary statistics such as engagement rate and total interactions

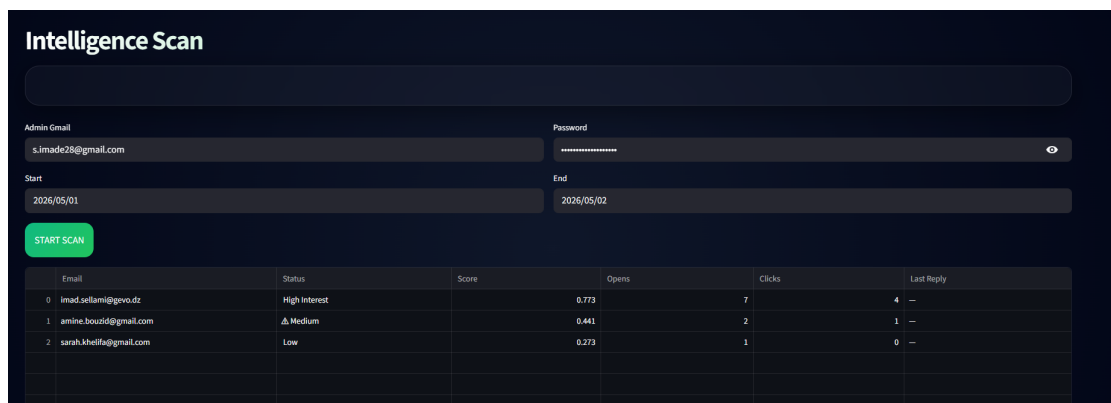


Figure 4.10: Prediction and analysis dashboard

4.7.7 Performance Monitoring

A dedicated statistics view was implemented to summarize system performance, including:

- Total number of processed users
- Number of high-interest leads
- Engagement rate

- Total opens and clicks

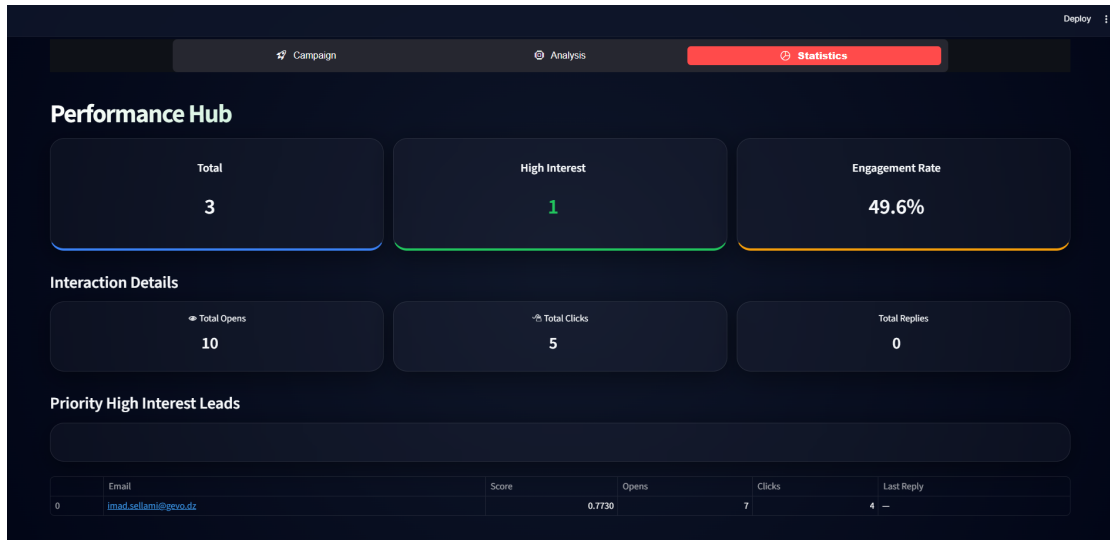


Figure 4.11: System performance and engagement statistics

4.7.8 Discussion

The implemented system demonstrates a complete end-to-end solution for intelligent prospect classification. It integrates real-time data acquisition, behavioral tracking, and machine learning prediction within a single platform.

The use of event-based tracking enhances the accuracy of behavioral modeling, while the hybrid neural network effectively combines textual and structured data.

Furthermore, the interactive dashboard improves usability and allows decision-makers to easily interpret the results. The modular architecture ensures scalability and flexibility, making the system suitable for real-world marketing applications.

4.8 Results and Analysis

This section provides a high-level interpretation of the system outputs and evaluates how the proposed solution performs in a real-world usage scenario.

4.8.1 System Output Interpretation

The implemented system generates predictions that are directly integrated into an interactive dashboard. Each user is associated with a prediction score reflecting their level of interest.

Users with higher interaction levels, characterized by multiple email opens and link clicks, consistently receive higher prediction scores. Conversely, users with limited or no interaction are classified as low-interest.

4.8.2 Behavioral Insights

The analysis confirms a strong correlation between behavioral features and prediction outcomes. In particular:

- Increased number of opens leads to higher engagement signals
- Click actions significantly influence the final prediction score
- Users with repeated interactions are more likely to be classified as high-interest

These observations validate the importance of incorporating behavioral tracking into the classification process.

4.8.3 System Effectiveness

The integration of the hybrid model within a complete system demonstrates its practical applicability. The system is capable of:

- Processing real-time interaction data
- Generating accurate predictions
- Providing interpretable results through a user-friendly interface

This confirms that the proposed approach is not only theoretically sound but also operationally effective.

4.8.4 Business Impact and Marketing Implications

The proposed system provides significant value for digital marketing operations. By accurately classifying prospects based on behavioral and textual data, the system enables marketers to prioritize high-value leads and optimize resource allocation.

From a business perspective, this leads to:

- Improved conversion rates through better targeting
- Reduced marketing costs by focusing on high-potential prospects
- Enhanced customer engagement via personalized strategies
- Faster decision-making using real-time predictive insights

These results demonstrate that the integration of artificial intelligence into marketing systems can significantly enhance efficiency and return on investment (ROI).

4.8.5 Conclusion

The results highlight the effectiveness of combining textual and behavioral data in a unified framework. The system successfully captures user intent and provides reliable predictions, making it suitable for intelligent marketing applications.

General Conclusion

This work presented the design, implementation, and evaluation of a hybrid intelligent system for prospect classification in digital marketing environments. The proposed approach integrates both textual data and structured behavioral data in order to provide a more accurate and comprehensive understanding of user intent.

The study began by exploring the theoretical foundations of Artificial Intelligence, Machine Learning, and Deep Learning, with a particular focus on neural networks and Natural Language Processing techniques. These concepts formed the basis for developing a hybrid architecture capable of handling heterogeneous data sources.

A key contribution of this work lies in the design of a multi-input neural network that combines an LSTM-based branch for textual analysis with a dense network for behavioral features such as email opens, clicks, and interaction patterns. This fusion mechanism allows the system to capture both implicit semantic information and explicit user behavior, resulting in improved classification performance.

The implementation of the system demonstrates its practical applicability. By integrating modules such as email processing, interaction tracking, feature engineering, and real-time prediction, the proposed solution forms a complete end-to-end pipeline. The use of an interactive dashboard further enhances usability, enabling decision-makers to easily interpret prediction results and monitor user engagement.

Experimental results confirm the effectiveness of the hybrid model. The training process showed stable convergence, while evaluation metrics such as accuracy, precision, and recall indicated strong classification performance. The results also highlight the importance of combining multiple data sources, as the hybrid approach outperforms traditional single-input models.

Despite these promising outcomes, certain limitations remain. The use of partially synthetic data may not fully reflect real-world complexity, and the performance of the system depends on the quality and diversity of the input data. Additionally, the architecture introduces increased computational complexity compared to simpler models.

Future work may focus on extending the system using larger real-world datasets, exploring advanced architectures such as transformer-based models, and improving the self-learning capabilities of the system. Further enhancements could also include real-time deployment

in production environments and integration with customer relationship management (CRM) platforms.

In conclusion, This work demonstrates the potential of combining artificial intelligence with behavioral analytics to transform traditional marketing into an intelligent, data-driven process. By combining machine learning, natural language processing, and real-time interaction tracking, the proposed system provides a scalable and practical framework capable of supporting evidence-based (or strategic) decision-making in modern marketing applications.

References

- [1] *Amazon Reviews Dataset*. <https://www.kaggle.com/datasets>. 2024.
- [2] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [3] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794.
- [4] Francois Chollet. *Deep Learning with Python*. Manning Publications, 2017.
- [5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *NAACL* (2019).
- [6] *Digital Marketing Campaign Dataset*. <https://www.kaggle.com/datasets>. 2024.
- [7] Dheeru Dua and Casey Graff. *Online Shoppers Purchasing Intention Dataset*. <https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset>. 2019.
- [8] Matthew Dunn and Priya Singh. "Uplift modeling: evaluation and applications." In: *Data Science Review* 5.4 (2021). (placeholder — replace with exact citation if available), pp. 210–228.
- [9] *Flask Documentation*. <https://flask.palletsprojects.com>. 2024.
- [10] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019.
- [11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [12] William L. Hamilton, Rex Ying, and Jure Leskovec. "Representation Learning on Graphs: Methods and Applications." In: *IEEE Data Engineering Bulletin* 40.3 (2017), pp. 52–74.
- [13] Sepp Hochreiter and Jurgen Schmidhuber. "Long Short-Term Memory." In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
- [14] *Keras Documentation*. <https://keras.io>. 2024.

-
- [15] Diederik Kingma and Jimmy Ba. ``Adam: A Method for Stochastic Optimization." In: *International Conference on Learning Representations* (2014).
 - [16] Jian Li, Wei Zhang, and Anil Kumar. ``Sequence-aware customer behavior modeling for conversion prediction." In: *Journal of Machine Learning Applications* 12.2 (2021). (placeholder — replace with exact citation if available), pp. 45–58.
 - [17] Tomas Mikolov et al. ``Efficient Estimation of Word Representations in Vector Space." In: *International Conference on Learning Representations (ICLR)*. 2013.
 - [18] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.
 - [19] *Pandas Documentation*. <https://pandas.pydata.org>. 2024.
 - [20] Frank Rosenblatt. ``The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." In: *Psychological Review* 65.6 (1958), p. 386.
 - [21] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2010.
 - [22] *Scikit-learn Documentation*. <https://scikit-learn.org>. 2024.
 - [23] *Streamlit Documentation*. <https://streamlit.io>. 2024.
 - [24] *TensorFlow Documentation*. <https://www.tensorflow.org>. 2024.
 - [25] Alan Turing. ``Computing Machinery and Intelligence." In: *Mind* 59.236 (1950), pp. 433–460.
 - [26] Ashish Vaswani et al. ``Attention Is All You Need." In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.
 - [27] Lian Zhang and Emily Roberts. ``Causal methods for targeting and uplift." In: *Journal of Causal Inference* 10.1 (2022). (placeholder — replace with exact citation if available), pp. 100–120.

Appendix A

Data Processing and Feature Engineering Details

This appendix provides detailed information about the data preprocessing, feature engineering, and input representation used in the proposed hybrid model.

Behavioral Feature Construction

The behavioral features are extracted from the tracking system, which records user interactions in an event-based format (open and click events with timestamps).

These events are aggregated to construct structured features as follows:

- **Open Count:** Number of times the email was opened
- **Click Count:** Number of link clicks performed by the user
- **Interaction Score:** Aggregated engagement score computed as the total number of interactions

Normalization

The structured features are normalized using the StandardScaler method to ensure consistent scaling across inputs.

- Features are centered around zero mean
- Standard deviation is scaled to one

This normalization step improves the convergence of the neural network during training.

Text Preprocessing

The textual content of emails is processed using Natural Language Processing techniques:

- Lowercasing and cleaning of raw text
- Removal of HTML tags and special characters
- Tokenization using Keras Tokenizer
- Conversion of text into integer sequences
- Padding sequences to a fixed length

Model Input Representation

The final input to the hybrid model consists of two components:

- **Text Input:** Sequence vector of fixed length ($\text{MAX_LEN} = 60$)
- **Structured Input:** Numerical feature vector ($\text{size} = 3$)

These inputs are processed through separate branches and later combined within the model architecture.

Appendix B

List of Acronyms

Acronym	Description
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NLP	Natural Language Processing
NN	Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
API	Application Programming Interface
GUI	Graphical User Interface
CSV	Comma-Separated Values
IMAP	Internet Message Access Protocol