VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANASANGAMA",BELAGAVI, KARNATAKA–590018

A Project Report on

"E-Commerce Customer Churn Prediction"

Submitted in partial fulfillment for the award of the degree of Bachelor of Engineering in Computer Science and Engineering during the year 2025– 2026

By

4MN22CS002	**AMOGH B R**
4MN22CS005	**ANKITH M P**
4MN22CS045	**SANGEETHA B K**
4MN22CS054	**SUHAS M**

Under the guidance of

Prof. Rajani
Assistant Professor
Department of Computer Science & Engineering
MIT Thandavapura

2025-2026

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(NBA Accredited for the Academic Year 2024– 2027)
MAHARAJA INSTITUTE OF TECHNOLOGY THANDAVAPURA
Just Off NH 766, Nanjanagudu Taluk, Mysore District–571302
(Approved by AICTE, accredited by NBA, New Delhi and Affiliated by VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAHARAJAINSTITUTEOFTECHNOLOGYTHANDAVAPURA

MYSORE – 571302

CERTIFICATE

This is to certify that the project work titled ***"E-Commerce Customer Churn Prediction"*** has been successfully carried out by **AMOGH B R [4MN22CS002]**,**ANKITH M P [4MN22CS005], SANGEETHA B K [4MN22CS045]** and **SUHAS M [4MN22CS054]** Bonafide students of Maharaja Institute of Technology Thandavapura in partial fulfilment of the requirements for the Degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2025-26. The project report has been approved as it satisfies the academic requirements for the project work prescribed for the Bachelor of Engineering Degree.

Project Guide **Prof. Rajani** **Assistant Professor** **Dept. of CS&E, MITT**	**Project Coordinator** **Prof. Bharath Bharadwaj B S** **Assistant Professor** **Dept. of CS&E, MITT**
HoD **Dr. Ranjit K N** **Professor and Head** **Dept. of CS&E, MITT**	**Principal** **Dr.Y T Krishne Gowda** **Principal, MITT**

External Viva

Name of the Examiners	**Signature with Date**
1.	
2.	

DECLARATION

We hereby declare that the project work entitled **"E-Commerce Customer Churn Prediction"** submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering to Visvesvaraya Technological University, Belagavi, is a record of our original work carried out under the supervision of **Prof. Rajani, Assistant Professor, Computer Science and Engineering** during the academic year **2025–26**. This project has not been submitted earlier or simultaneously for the award of any degree, diploma, or fellowship at any other university or institution.

We further confirm that a plagiarism check was carried out using the plagiarism detection software recommended by the university. The content of the report was thoroughly verified, and the overall similarity index is well within the prescribed limits. Proper citations and references have been provided wherever content from external sources has been referred. The report adheres to the academic integrity policies of the university regulations.

We take full responsibility for the accuracy, originality, and ethical standards maintained throughout the project work and documentation.

AMOGH B R
[4MN22CS002]

ANKITH M P
[4MN22CS005]

SANGEETHA B K
[4MN22CS045]

SUHAS M
[4MN22CS054]

Place: Mysuru
Date: 12/12/2025

ACKNOWLEDGEMENT

It give us immense pleasure to bring our project report entitled ***"E-Commerce Customer Churn Prediction"*** in the final year engineering course.

We are very thankful to **Dr.Y T Krishne Gowda, Principal, MITT** for having supported us in our academic endeavors.

We would like to extend our heartfelt thanks to **Dr. Ranjit K N, Professor and Head, Dept. of CS&E,** for providing us timely suggestions, encouragement, and support to complete this work.

We would like to express our sincere gratitude to the Project Coordinator **Prof. Bharath Bharadwaj B S, Assistant Professor, Dept. of CS&E** for his valuable support, suggestions, and coordination throughout this project.

We would like to sincerely thank our project guide **Prof. Rajani Professor, Dept. of CS&E** for providing relevant information, valuable guidance, and encouragement to complete this report.

We would also like to thank all the teaching and non-teaching staff members of the department for their support. We are always thankful to our parents for their valuable support and guidance in every step.

We express our deepest gratitude and indebt thanks to MITT which has provided us an opportunity in fulfilling our most cherished desire of reaching the goal.

ABSTRACT

Customer retention has become a critical concern in the highly competitive e-commerce landscape, where customer attrition directly affects profitability and long-term growth. An intelligent Customer Churn Prediction System is developed to identify potential churners at an early stage using data-driven techniques. The system analyzes essential customer behavior attributes such as purchase history, login frequency, complaint records, and satisfaction ratings. After systematic data preprocessing, multiple machine learning algorithms, including Logistic Regression, Random Forest, and XGBoost, are applied to classify customers into Active, At Risk, and Churned categories. Predictive outcomes are presented through an interactive dashboard that clearly illustrates churn patterns and highlights the most influential factors driving customer disengagement. In addition, actionable retention recommendations such as personalized discounts, loyalty incentives, and service improvement strategies are generated to support timely business interventions. Early identification of churn risk enables proactive decision-making, strengthens customer relationships, improves retention rates, and minimizes revenue loss for e-commerce organizations.

Keywords:

Customer Churn, E-Commerce Analytics, Predictive Modeling, Machine Learning Techniques, Customer Retention Strategies, Classification Algorithms, Logistic Regression, Random Forest Classifier, Gradient Boosting Methods, Data Mining Processes, User Behavior Analysis.

CONTENTS

Sl. No.		Contents	Page No.
1		**Introduction**	**[1-6]**
1.1		Objective Of the Project	1
1.2		Problem Statement	2
1.3		Scope of the Project	3
1.4		Motivation	5
1.5		Organization of the Report	5
2		**Literature Survey**	**[7-13]**
2.1		Overview of Existing Systems	7
2.2		Comparison of Related Work	10
2.3		Gaps in Current Solutions	11
2.4		Relevance And Uniqueness of the Proposed Work	12
2.5		Summary	13
3		**System Requirement Specification And Analysis**	**[20-33]**
3.1		Requirement Analysis	20
	3.1.1	Functional Requirements	21
	3.1.2	Non-Functional Requirements	21
3.2		Feasibility Study(Technical, Economic, Operational)	22
3.3		Requirement Traceability Matrix(RTM)	23
3.4		System Architecture Overview	26
3.5		Use Case Diagrams	29
4		**Datasets**	**[34-44]**
4.1		Source	37
4.2		Preprocessing Steps	37
4.3		Samples	39
5		**System Design**	**[45-65]**
5.1		High Level Design (HLD)	45
5.2		Low Level Design(LLD)	47
5.3		UML Diagrams	48
	5.3.1	Class Diagram	48
	5.3.2	Sequence Diagram	52

Sl. No.		Contents	Page No.
	5.3.3	Activity Diagram	55
5.4		Data Flow Diagrams (DFD)	58
5.5		Database Design	61
	5.5.1	E R Diagram	62
	5.5.2	Database Schema	64
6		**Implementation**	**[66-68]**
6.1		Technology Stack	66
6.2		Development Environment	66
6.3		Module-Wise Implementation with Description	67
7		**Testing**	**[69-72]**
7.1		Testing Methodology	69
	7.1.1	Unit Testing	69
	7.1.2	Integration Testing	70
	7.1.3	System Testing	70
	7.1.4	User Acceptance Testing(UAT)	71
7.2		Relevant Test Cases	72
	7.2.1	Test Case Id	72
	7.2.2	Description	72
	7.2.3	Input	72
	7.2.4	Expected Output	72
	7.2.5	Actual Output	72
	7.2.6	Status(Pass/Fail)	72
8		**Results And Discussion**	**[73-80]**
8.1		Project Output and Screenshots	73
8.2		Performance Metrics	79
8.3		Evaluation and Analysis	79
8.4		Comparison with Existing Approaches	80
		Conclusion and Future Enhancement	**[81]**
		Bibliography	82
		Appendix	85

LIST OF FIGURES

Figure No.	Figure Title	Page No.
3.1	System Architecture Overview	29
3.2	Use Case diagram	30
4.1	Dataset Sample	39
5.1	High Level Design(HLD)	46
5.2	Low Level Design(LLD)	48
5.3	Class Diagram	50
5.4	Sequence Diagram	54
5.5	Activity Diagram	57
5.6	Level 0 Data Flow Diagram	59
5.7	Level 1 Data Flow Diagram	60
5.8	ER Diagram	63
5.9	Database Schema	65
8.1	Home Page	73
8.2	About Page	73
8.3	Developers Page	74
8.4	Login Page	75
8.5	Customer Dashboard Page	75
8.6	Usage and Behaviour Page	76
8.7	Order History Page	76
8.8	Prediction Result	77
8.9	Prediction Result-Graphical Representation	77
8.10	Customer Risk Factor Impact	78

LIST OF TABLES

Table No.	Table Title	Page No.
2.1	Literature Survey	14
3.1	Requirement traceability Matrix(RTM)	23
6.1	Technology Stack	66
7.1	Test Cases	72
8.1	Performance Metrics	79

CHAPTER–01

INTRODUCTION

The digital economy has seen tremendous growth in recent years. Almost every type of business now uses online platforms to sell products and services. This shift has created a highly competitive environment where customers can easily explore multiple brands, compare prices, read reviews, and make quick decisions. As a result, customer loyalty has become increasingly difficult to maintain.

One of the biggest challenges faced by online platforms today is customer churn**,** which refers to customers who stop purchasing or interacting with the platform. Churn has a direct impact on revenue, brand image, and business stability. Studies have shown that acquiring a new customer costs far more than retaining an existing one. For this reason, companies are focusing more on understanding why customers leave and how to prevent it.

This project focuses on developing a customer churn prediction model for an e-commerce platform. The aim is to analyze customer behavior, identify patterns that lead to churn, and build a system that can predict which users are likely to leave in the near future. The project uses historical data such as transaction history, login frequency, purchase behaviour, customer complaints, and demographic information. These features are processed and used to train machine learning algorithms.

By predicting churn early, the platform can take proactive steps to retain customers. For example, the business can offer personalized discounts, loyalty points, targeted notifications, or improved customer support. These strategies not only reduce churn but also enhance customer experience, lifetime value, and long-term profitability.

1.1 Objective of the Project

The main objective of this project is to develop an intelligent predictive system that can identify customers who are likely to stop using an e-commerce platform in the near future. Customer churn is a serious challenge in online businesses because retaining an existing customer is often more cost-effective than acquiring a new one. By detecting churn in advance, companies can plan targeted retention strategies and reduce revenue loss.

This project focuses on analyzing customer behavior such as purchase history, frequency of visits, transaction value, complaints, return patterns, and engagement levels. Machine learning models are used to learn from historical data and discover hidden patterns

that influence churn. The system then provides a prediction for each customer, indicating whether the customer will remain active or is at high risk of leaving the platform.

Another important objective is to support decision-making for marketing teams. When a customer is identified as a potential churner, the platform can implement personalized interventions like discounts, promotional emails, loyalty points, or customer support calls. These proactive actions improve customer satisfaction and help maintain long-term relationships.

The project also aims to improve business profitability. Reducing churn directly increases customer lifetime value, stabilizes sales, and enhances overall growth. The knowledge gained from the churn prediction model can guide strategic planning, product improvements, and customer experience design. Overall, the objective of this project is not only to predict churn, but also to enable e-commerce companies to understand the reasons behind customer losses and take meaningful steps to retain valuable customers, ensuring sustainable business success.

1.2 Problem Statement

In the rapidly growing e-commerce sector, customer retention has become a critical challenge. Although online platforms collect enormous amounts of customer data every day such as browsing history, purchase frequency, complaints, and login activity most organizations still do not make full use of this information. Instead of predicting the risk of churn in advance, many companies only respond after the customer has already stopped using the platform. This reactive approach leads to loss of revenue, increased marketing expenses, and weak customer relationships. The core problem addressed in this project is the lack of an accurate, real-time prediction mechanism that can analyze multi-dimensional behavioral data and identify customers who are likely to leave. Current systems are either too slow, limited to basic metrics, or unable to capture dynamic patterns in customer behavior. As a result, organizations miss the opportunity to take proactive measures such as personalized offers, targeted communication, or loyalty programs. Therefore, there is a need for a robust churn prediction model that can process diverse data sources, detect early warning signs, and support timely retention strategies to improve business sustainability.

1.3 Scope of the Project

The scope of this project focuses on predicting customer churn in the e-commerce domain by analyzing user behavior, purchase history, complaints, and interaction patterns. The model aims to identify customers who are likely to stop using the platform and help the organization take timely action to retain them. The project not only provides churn prediction, but also gives insights into *why* customers are leaving, which features strongly influence churn, and what strategies can be implemented to improve retention.

This project is useful across multiple functions of an e-commerce organization:

1. E-commerce Retailers

E-commerce companies experience high competition and switching behaviour. Customers may quickly shift to another platform if they feel dissatisfied or ignored. The proposed churn prediction system allows retailers to detect early signs of customer dropout and take preventive actions, such as offering loyalty rewards, discounts, or personalized recommendations. By retaining existing customers, retailers can reduce revenue loss and increase the lifetime value of every user. This is highly beneficial because retaining a customer is often cheaper than acquiring a new one.

2. Marketing Teams

Marketing teams can use churn prediction results to design targeted campaigns instead of spending resources on general advertisements. The model clearly identifies which customers are at high risk, enabling teams to send customized offers, reminder emails, or promotional messages. Instead of guessing what customers need, marketing efforts become more data-driven and efficient. As a result, marketing budgets are optimized, campaign performance improves, and customer engagement becomes stronger.

3. Product Managers

Product managers can benefit from this project by understanding which aspects of the platform are causing dissatisfaction. By analysing churn factors such as delayed delivery, poor recommendations, low frequency of purchases, or negative feedback, product teams can make informed decisions to improve the platform. These insights help in revising product features, enhancing user experience, and prioritizing requirements. When pain

points are resolved, the platform becomes more user-friendly and customers remain loyal for longer periods.

SDG Goals:

Some of the SDG Goals are:

SDG 8 : Decent Work and Economic Growth

This project supports sustainable economic growth by helping e-commerce companies retain existing customers rather than constantly spending on acquiring new ones. By predicting customer churn early, businesses can improve profitability, stabilize revenue, and create long-term employment opportunities through business expansion. Efficient customer retention strategies also help companies grow in a financially responsible manner.

SDG 9 : Industry, Innovation, and Infrastructure

The proposed churn prediction system uses machine learning and data analytics, promoting innovation in digital commerce. By leveraging customer behavior data, the project contributes to building intelligent and resilient digital infrastructure. This encourages industries to adopt advanced technologies for smarter decision-making and operational efficiency.

SDG 10 : Reduced Inequalities

Customer churn prediction helps companies understand diverse customer needs and behaviors. By identifying dissatisfaction patterns across different user groups, e-commerce platforms can design inclusive pricing, service improvements, and personalized offers. This ensures fair treatment of customers and reduces service inequality across regions and demographics.

SDG 12 : Responsible Consumption and Production

By focusing on customer retention instead of excessive marketing and resource-intensive promotions, this project encourages responsible business practices. Targeted engagement reduces unnecessary advertising, lowers digital waste, and promotes efficient use of resources, supporting sustainable consumption patterns.

1.4 Motivation

In today's competitive e-commerce market, customer expectations are growing at a faster pace than ever before. Customers have access to thousands of online stores offering similar products, faster delivery, and attractive discounts. Because of this, many customers do not remain loyal to a single platform. The moment they face poor service, delayed delivery, complicated return policies, or lack of personalization, they simply move to another platform. This phenomenon is known as customer churn**.**

For an e-commerce company, acquiring a new customer is costlier than retaining an existing one. Marketing, advertisements, onboarding, and promotional offers require significant investment. When a customer leaves, all the efforts and money spent on that customer are lost, and the company has to start the entire process again. Therefore, the ability to identify customers who are likely to churn before they leave becomes very important.

This project is motivated by the need to support e-commerce businesses in understanding customer behavior in a data-driven way. Every customer leaves behind a digital trail through their purchases, browsing patterns, login frequency, complaints, reviews, and interaction history. These patterns can reveal early warning signs of dissatisfaction. By analyzing these signals using machine learning, businesses can predict which customers are at risk of leaving and take timely action.

The motivation of this work is also strengthened by the fact that customer retention directly increases revenue. Even a small improvement in customer retention rate can create a large impact on profit. Predictive churn analysis helps companies send personalized offers, loyalty rewards, or better service options to the right customers at the right time. This ensures higher customer satisfaction, improved trust, and long-term relationships.

In simple terms, this project aims to transform raw data into actionable insights. Instead of waiting for customers to leave, e-commerce companies can now predict churn, prevent churn, and protect their most valuable asset their customers.

1.5 Organization of the Report

The project is organized into several structured chapters to provide a smooth and logical flow of understanding:

Chapter 1: Introduction

This chapter provides an overall background of e-commerce platforms and highlights why customer retention is a key business challenge. It explains the motivation behind predicting churn and outlines the problem statement, objectives, and scope of the project.

Chapter 2 : Literature Survey

The literature survey summarizes existing research on churn prediction techniques, machine learning models, and related tools. It identifies the gaps in current solutions and shows how this project contributes to improving accuracy and decision-making for e-commerce platforms.

Chapter 3 :Requirements

This chapter describes the functional and non-functional requirements necessary for the development of the churn prediction system. It includes dataset needs, software tools, system specifications, and performance expectations that must be satisfied.

Chapter 4 : System Analysis and Design

System analysis presents the architecture, data flow, and process logic used to detect churn. Design diagrams such as use case diagrams, DFDs, and workflow models are included to illustrate how the system components interact to generate churn predictions.

Chapter 5 : Implementation

This chapter explains the step by step development of the model, covering data preprocessing, feature engineering, algorithm selection, and coding. It also describes how the machine learning model was trained and integrated into the application.

Chapter 6 : Testing

Testing activities are discussed to ensure accuracy, reliability, and correctness of predictions. Various test cases, validation methods, and evaluation metrics such as accuracy, precision, recall, and F1-score are presented with observations.

Chapter 7 : Results and Snapshots

This chapter presents the experimental results, model performance outputs, and visual snapshots of the system interface. It summarizes the findings, highlights improvements over existing approaches, and interprets how the churn prediction helps in decision-making.

CHAPTER–02

LITERATURE SURVEY

2.1 Overview of Existing Systems

Existing systems for customer churn prediction in e-commerce were mainly designed to analyze customer behaviour and identify patterns that lead to customer dropout. These models were developed to help companies retain valuable customers, reduce marketing costs, and improve service quality. However, most existing approaches suffer from limitations such as poor accuracy, lack of real-time insights, and inability to consider dynamic customer interactions.

Traditional churn prediction systems relied heavily on statistical methods**,** simple regression models**,** rule-based approaches**.** These systems generally used past purchase history and demographic information to determine whether a customer is likely to leave. Although these systems provided basic analytical understanding, they failed to capture deep behavioural patterns such as browsing activity, click stream data, unstructured feedback, and social influence, which are highly relevant for churn prediction today.

Although these approaches are promising, most existing systems still face challenges such as:

- Lack of explainability, where the model behaves like a black box.
- High computational cost for training deep models.
- Difficulty in integrating real-time customer interactions.
- Limited evaluation metrics used to measure model performance.

Several studies have investigated e-commerce customer churn prediction by analyzing customer demographics, transactional records, behavioral patterns, and engagement metrics using machine learning and data mining techniques. One significant study by Li and Li (2019) proposed a hybrid churn prediction model combining Logistic Regression and XGBoost using datasets consisting of transaction frequency, customer tenure, and purchase value. Their ensemble-based approach demonstrated improved predictive performance by leveraging the strengths of multiple classifiers. However, the study focused mainly on accuracy and did not emphasize interpretability. Inspired by this work, our project incorporates ensemble learning techniques while proposing the inclusion of explainability components to make churn predictions more transparent and

business-friendly [6]. Research focusing on explainable machine learning demonstrated that customer interactions, purchase recency, and spending behavior help justify churn predictions. However, explainability was typically applied after model training. To improve transparency and trust, our project proposes integrating explainability mechanisms directly into the churn prediction workflow. Time-based churn prediction has also been studied using sequential customer activity, subscription duration, and usage patterns through deep learning models. Although these methods effectively capture temporal behavior, they increase computational complexity. Inspired by these findings, our project includes time-dependent behavioral features while maintaining scalability.

Other research has highlighted the role of historical purchase behavior, browsing activity, customer tenure, and spending patterns in identifying churn risk. Although these approaches capture time-based behavior, feature engineering is usually limited to basic recency and frequency measures. In contrast, our project introduces advanced features such as Customer Lifetime Value (CLV) and RFM scoring to better reflect long-term customer engagement. Some studies have explored the use of transaction logs, login frequency, and service usage indicators to train multiple churn prediction models. While comparing different algorithms improved insights, these approaches often lacked robustness across datasets. Motivated by this, our project incorporates service usage features and adopts ensemble-based model comparison to enhance stability and performance.

Another important contribution was made by Zaghloul et al. (2024), who focused on predicting customer satisfaction using datasets that included feedback scores, order fulfillment details, and service ratings. Although the primary objective was satisfaction prediction rather than churn, the study highlighted a strong correlation between customer satisfaction and retention. Building upon this insight, our project adopts satisfaction-related indicators and proposes linking them directly with churn probability to improve early churn detection [12]. Comparative studies of machine learning and deep learning models show that performance varies depending on data and model choice, with no single algorithm consistently outperforming others. Based on this observation, our project benchmarks multiple models and selects the most suitable one using performance evaluation metrics.

Several existing studies have used customer demographic details and transactional data such as age, gender, purchase count, transaction value, and past buying behavior to predict churn with traditional machine learning models. These works establish

that demographic and behavioral features are fundamental for churn analysis, but they often depend on simple preprocessing methods. To overcome this limitation, our project enhances feature selection and applies advanced preprocessing and scaling techniques to improve prediction accuracy.

Customer and product-level churn analysis was addressed by Al Rahib et al. (2024**)**, who utilized datasets containing customer profiles, product categories, transaction volume, and interaction history. Their findings highlighted the importance of structured data in churn modeling. Inspired by this approach, our project adopts structured customer and product-level features and further proposes derived attributes such as purchase trend deviation to capture abnormal behavior patterns [3]**.** Several works have proposed system-level churn prediction frameworks using data mining techniques applied to customer purchase records, visit frequency, and transaction logs. These studies highlight the importance of a complete processing pipeline. In line with this, our project adopts an end to end churn prediction architecture from data ingestion to result visualization.

A comprehensive approach to churn risk modeling was presented by Ren (2025)**,** who utilized behavioral, transactional, and engagement-based datasets to assess churn risk using supervised learning methods. The study emphasized the importance of combining multiple data dimensions for effective churn prediction. Motivated by this work, our project integrates both static customer attributes and dynamic behavioral features to support scalable and robust churn risk assessment [19]**.** Finally, survey-based studies reviewing multiple churn prediction algorithms identified effective techniques using customer activity, transaction history, and engagement data. Based on these insights, our project selects the best-performing model through comprehensive evaluation.

Personalized churn prediction was explored by Lei Zhang and Qing Wei (2024) using browsing history, customer interaction data, and purchase sequences modeled through Bi-LSTM networks. Their work demonstrated that contextual and sequential behavior plays a critical role in identifying churn-prone customers. Based on this study, our project extracts contextual and time-dependent behavioral features and proposes personalized churn risk scoring for individual customers [10]**.** Customer segmentation approaches have shown that grouping customers based on shopping frequency, purchase value, and preferences improves churn understanding. Inspired by this, our project incorporates segmentation features and performs class-wise churn analysis.

2.2 Comparison of Related Work

Research on e-commerce customer churn prediction has grown rapidly over the past decade. Early studies mainly relied on traditional statistical models such as Logistic Regression and Decision Trees. These models were effective in identifying basic churn indicators, such as purchase inactivity or reduced login frequency, and their interpretability supported business decision-making. However, their predictive performance decreased significantly when applied to large datasets or more complex customer behavior patterns.

As machine learning techniques advanced, models like Random Forest, Gradient Boosting, and Support Vector Machines were adopted to better capture non-linear relationships among features, including purchase frequency, browsing behavior, and customer complaints. While these approaches achieved higher predictive accuracy, they often required frequent retraining to adapt to the dynamic nature of e-commerce environments. Moreover, sequential and time-sensitive customer behaviors were often underutilized in many of these methods.

A notable study by Li (2024) analyzed customer transactional and behavioral datasets, including purchase frequency, order value, and engagement metrics, using boosting algorithms such as XGBoost. Their findings highlighted the effectiveness of ensemble methods in reducing overfitting and handling imbalanced churn data. Building on this work, our project incorporates similar ensemble learning techniques and further integrates explainability modules to make churn predictions more transparent and actionable [1]**.** Some studies also explored customer satisfaction, purchase recency, and engagement metrics to identify early churn signals. While insightful, these approaches often overlooked the integration of multiple feature types and sequential customer patterns. In contrast, our project combines static attributes, dynamic behavior, and satisfaction indicators into a unified predictive framework.

Kumar et al. (2024) conducted a comparative study on machine learning algorithms using datasets including customer tenure, purchase history, transaction value, and engagement frequency. They benchmarked models for predictive accuracy and identified the most effective algorithms for churn detection. Inspired by this, our project selects key churn features from these datasets and implements multi-model evaluation to identify the best-performing approach [13]**.** Additionally, segmentation based studies have demonstrated that analyzing customers based on shopping frequency, transaction value, and preferences improves churn prediction and retention strategies. Following this insight,

our project applies segmentation features to enable more precise, group-specific churn predictions.

Survey and implementation studies by Rathi et al. (2023) reviewed multiple machine learning classifiers using customer activity, transaction history, and engagement metrics. Their findings emphasized the importance of systematic algorithm evaluation. Leveraging this, our project combines traditional and ensemble models with optimized feature selection to improve robustness and predictive performance [18]**.** Other works explored hybrid strategies that combined statistical and machine learning methods to balance interpretability and predictive accuracy. Using transactional, demographic, behavioral, and engagement-based datasets, these studies enhanced churn detection. Inspired by these findings, our project integrates ensemble techniques with temporal and sequential feature engineering, enabling more accurate, explainable, and adaptive churn predictions.

Earlier statistical approaches by Rajasekaran and Tamilselvan (2023) focused on demographic and transactional features such as age, gender, purchase count, and historical activity. While these models were interpretable, they struggled with large-scale datasets and complex behavioral patterns. Building on this, our project refines feature engineering and preprocessing techniques to effectively handle large, dynamic datasets [5]**.** Framework-level research by Huda et al. (2023) proposed end-to-end churn prediction pipelines using classification techniques on customer transactions and interactions. Our project extends this by adopting a complete pipeline architecture, integrating advanced feature engineering and model optimization for improved accuracy and scalability [20]**.** Finally, studies investigating time-based sequential patterns such as login frequency, purchase sequences, and subscription duration highlighted the importance of temporal behavior in detecting churn. Inspired by these works, our project incorporates sequence aware learning and temporal behavioral features while maintaining scalability and computational efficiency.

2.3 Gaps in Current Solutions

Although several models and analytical approaches have been applied to predict customer churn, many existing solutions still face limitations. Most traditional churn prediction systems rely mainly on historical transaction data, without examining broader customer behaviour such as browsing patterns, time spent on the website, page interactions, or complaint logs. As a result, these models fail to fully capture the customer’s intent and emotional shift before the actual churn happens.

Another major gap is related to handling imbalanced datasets. In e-commerce platforms, the percentage of churned customers is usually lower compared to active customers. Many machine learning models struggle to learn from this skewed data, leading to inaccurate predictions and high false negatives. This means the system often classifies a churn-prone user as a loyal one, which ultimately affects business revenue.

Additionally, many existing studies do not focus on real-time predictions. Churn is a dynamic process where user behaviour changes rapidly. Systems that use static, batch-based models may miss early warning signals. The absence of continuous monitoring and early intervention mechanisms reduces the effectiveness of churn prevention strategies.

Another gap identified in literature is the lack of personalization. Most solutions provide generalized offers or retention strategies for all users, rather than tailoring actions based on customer segment, interests, or purchase history. Since every customer has unique purchasing patterns, the same retention offer may not be effective for all.

Finally, many studies consider only single machine learning models, without exploring hybrid or deep learning methods. Techniques such as ensemble models, neural networks, and attention-based architectures have shown better capability in pattern recognition, but they remain underutilized in churn prediction, particularly for e-commerce environments.

2.4 Relevance and Uniqueness of the Proposed System

The proposed system addresses the above gaps by integrating a more comprehensive view of customer behaviour. It combines transactional history, browsing patterns, engagement levels, complaint records, and demographic details into a unified analytical framework. This multi-dimensional approach allows the model to identify subtle behavioural changes that often precede churn.

A key uniqueness of the proposed system is the use of advanced machine learning algorithms such as ensemble models, and deep learning architectures that can learn complex non-linear patterns. These models can detect early signals of churn that traditional statistical methods usually miss. To overcome the issue of data imbalance, techniques like SMOTE and threshold tuning are applied, which significantly improve sensitivity and ensure that churn-prone customers are accurately identified.

Real-time prediction forms another highlight of this system. Instead of waiting for monthly or weekly updates, the model continuously analyses user activity and generates

alerts whenever risky behaviour is detected. This enables businesses to take immediate action such as personalized discounts, loyalty rewards, or customer support intervention.

The proposed system is also highly scalable and suitable for large datasets commonly produced by e-commerce platforms. It ensures fast processing and can be easily deployed in cloud environments.

Most importantly, the system emphasizes proactive retention strategies. Rather than waiting for a customer to leave, the system predicts churn in advance and suggests actionable decisions. The use of personalized recommendations makes the solution more impactful, as every customer receives customized offers or notifications based on their preferences and buying habits.

2.5 Summary

This chapter reviewed existing literature related to customer churn prediction in e-commerce environments. The analysis showed that although several models have been developed, many still suffer from limitations such as reliance on limited data sources, lack of personalization, and inability to handle imbalanced data. Traditional models also fail to provide real-time monitoring and early intervention.

To resolve these issues, the proposed system introduces a holistic approach that combines behaviour tracking, transaction analysis, and advanced machine learning techniques. The system focuses on real-time churn prediction and personalized retention actions, making it more relevant to the modern competitive e-commerce market where customer loyalty is extremely valuable.

The unique contribution of this work lies in its ability to process multiple data streams, generate accurate churn predictions, and enable data-driven decision making. By identifying churn-prone users early, the model helps businesses reduce revenue loss and improve customer satisfaction.

This foundation forms the basis for the subsequent chapters, where the system design, implementation, and experimental results are discussed in detail.

Table 2.1: Literature Survey

Authors & Published Year	Title of the Project	Datasets	Methods/ Techniques used	Key Findings	Limitations/ Gaps
Mengchen Yu (2025)	Cross-Border E-Commerce User Churn Prediction Model Based on Decision Tree Algorithm	User ID, Country, Login Frequency, Transaction Count, Average Purchase Value, Last Purchase Date, Return Rate, Device Type, Churn	Data Preprocessing, Feature Selection, Model Building, Ensemble Learning	Reduced Model Complexity, Efficient Computation	Scope Limitation, Generalizability, Limited Real-world Integration
HaoranRen (2025)	Machine Learning-Based Prediction of Customer Churn Risk in E-commerce	Tenure, Warehouse To Home, Number Of Device Registered, Preferred Order Category, Satisfaction Score, Marital Status, NumberOf Address, Complaint, Day Since Last Order, Cash back Amount, Churn	Logistic Regression, Random Forest, Gradient Boosting Decision Tree, Neural Network, Stacking Model using XGBoost as meta learner	Accuracy: 92.8% AUC: 0.940 KS: 0.779, Tenure, Complaint status, Cashback amount, Days since last order, Satisfaction score	High computational cost, Metadata dimensionality , may not generalize to e commerce companies broader customer attributes.

Jingyuan Li (2024)	Customer Churn Prediction using Machine Learning: A Case Study of E commerce Data	ID, Churn (target variable), Tenure, Preferred Login Device, Preferred Payment Mode, Gender, Hour Spend On App, Complain, Marital Status, Cash back Amount, Day Since Last Order	Addressed class imbalance using SMOTE, Techniques: SHAP And LIME, Handling missing values, outlier removal	Provided actionable recommendations for businesses to reduce churn, Key churn-driving factors such as tenure, satisfaction score, and order count	May not generalize across different e-commerce platforms, Potential improvements using deep learning techniques like RNNs or LSTMs for sequential data analysis
Eduarda Neves da Silva, Filipe Bento Magalhaes, Walter Jaimes Salcedo (2024)	Customer Churn Prediction in E commerce Using Machine Learning and LIME Algorithm	Customer ID, City Tier, Warehouse To Home, Number Of Device Registered, Preferred Order Category, Satisfaction Score, Number Of Address, Coupon Used	Used K means clustering to group customers based on common motivations	Achieved an AUC of 78% in churn prediction, Identified five distinct customer clusters based on churn motivations	Dataset lacked additional predictive variables, Potential improvements using deep learning
Lei Zhang Qing Wei (2024)	Personalized and Contextualized Data Analysis for E Commerce	Customer ID, Churn (target variable), Tenure, Preferred Login Device, Preferred Payment Mode, Gender	Bi Directional Long Short Term Memory (Bi LSTM) Neural Network.	Outperforms traditional machine learning methods in churn prediction tasks.	Broader dataset inclusion for generalization, Improve model transparency.
Ch. Anudeep, R. Venugopal, Mohd Aarif , Thiruma Valavan A (2024)	Predicting Customer Churn in E commerce Subscription Services using RNN with Attention Mechanisms	Customer ID, Signup Date, Last Login Date, Subscription Type, Days Since Last Interaction, Churned , Renewal Status	Recurrent Neural Networks (RNN), Attention Mechanisms, Evaluation Metrics	Model Interpretability, Temporal behavior, Insight into customer behavior patterns	Scalability Concerns, Computational Intensity, Limited Real World Validation
Lei Zhang, Qing Wei (2024)	Personalized and Contextualized Data Analysis for	User ID, Time stamp, Event Type, Product ID, Order Gap, Days Since	Feature scaling and normalization, 10 fold cross	Model robustness, Interpretability, Validated	Computational Complexity, Limited Domain

	E Commerce Customer Retention Improvement With Bi LSTM Churn Prediction	Last Event, Churn	validation, Bi-LSTM architecture capturing sequential behavior in customer data	with rigorous experimentation	Generalization, Model Interpretability
Bagaskara Putra Wibowo, Lili Ayu Wulandhari (2024)	E Commerce Customer Churn Prediction Using Machine Learning Approaches	User ID, Last Purchase Date, Frequency, Monetary Value, Days Since Last Visit, Is Churned	XGBoost, CatBoost, K-Means Clustering, DBSCAN	CatBoost-outlier removal achieved the best F1 score of 96%, K Means clustering significantly decreased the F1-score to 92.6% at best.	Limiting generalization, Refining outlier detection methods.
Md Abdullah Al Rahib, Nirjhor Saha, Raju Mia, Abdus Sattar (2024)	Customer Churn Prediction and Analysis in E Commerce Using Machine Learning	Customer ID, Signup Date, Last Purchase Date, Total Spend, Purchase Frequency, Time Since Last Purchase, Churn Label (0 for active, 1 for churned)	Support Vector Machine, Random Forest, Decision Tree, XGBoost, Linear Regression	Highest accuracy, purchase frequency, product categories, average session length.	More advanced techniques, high R^2 score for customer annual spending prediction.
Awais Manzoor, M. Atif Qureshi, Etain Kidney, Luca Longo (2024)	A Review on Machine Learning Methods for E Commerce Customer Churn Prediction and Recommendations for Business Practitioner	Tenure, Preferred login device, City tier Ware house to home distance, Satisfaction score, Order count, Day since last order, Cash back amount	Ensemble Models, Interpretable ML, Non interpretable ML, Discount, cashback, satisfaction scores, Social interaction and unstructured data	Behavioral indicators, Text-based features from reviews, sentiment, and feedback	Profit-driven churn prediction is under represented, Lack of domain specific guidelines

Tope-Oke Adebusola Modupeola , Babalola Gbemisola, Abiola Oluwatoyi n (2024)	Customer Churn Prediction System using Deep Learning Techniques	Customer ID, City Tier, Ware house To Home, Number Of Device Registered, Preferred Order Category	Feature engineerin g techniques such as Principal Componen t Analysis (PCA) and correlation matrices	Embedding layers , Recommen ded advanced feature engineering to improve churn prediction accuracy.	Neural networks and genetic algorithms for more complex behavioural analysis.
Maha Zaghloul, Sherif I. Barakat, Amira Rezk (2024)	Predicting E Commerce Customer Satisfaction: Traditional Machine Learning vs. Deep Learning Approaches	Customer demographics, product details, reviews, and ratings.	DL Model: Multi Layer Perceptron features considered include delivery time, order value,and customer location.	Feature Importance , Model Performanc e, Practical Implication s	Temporal Dynamics, Generalizabil it y, Model Interpretabilit y
Saurabh Kumar, Suman Deep, Pourush Kalra (2024)	A Comprehen sive Analysis of Machine Learning Techniques for Churn Prediction in E Commerce: A Comparative Study	Customer demographics, purchase history, and engagement metrics, product views, purchases, and session durations, service usage, account details, and churn status	Technique s with a particular focus on (XGBoost), Handling Class Imbalance	Understand ing Customer Behaviour, Model Performanc e, Practical Guidelines	Scalability Concerns, Generalizabil it y
Nagaraj P, Muneeswa r an V, Dharanidh ar an A (2023)	E Commerce Customer Churn Prediction Scheme Based on Customer Behaviour Using Machine Learning	Customer ID, Tenure, Preferred Login Device, Ware house To Home, Preferred Payment Mode, Gender, Complain, Order Amount Hike From last Year, Coupon Used, Order Count, Churn.	Decision Tree, Random Forest, Support Vector Machine (SVM)	Best performanc e, Improve customer retention, Reduce customer churn and improve service/pro d uct offerings	Moderate to low accuracy, Lack robustness

Jie Yang (2023)	Design of E commerce Customer Churn Prediction System Based on Data Mining Techniques	Customer ID, City Tier, Ware house To Home, Number Of Device Registered, Number Of Address, Coupon Used	Used Decision Tree and Random Forest models, Performed correlation analysis and principal component analysis, continuous variable selection	Importance of data mining in extracting valuable insights from customer behaviour, Provided actionable recommend ations for businesses	The study focused on a single dataset, Real-time customer interaction data, Deep learning techniques
Nagaraj P, Muneeswa r an V, Dharanidh ar an A, Aakash M, Balanantha n an K, Rajkumar C (2023)	E Commerce Customer Churn Prediction Scheme Based on Customer Behaviour Using Machine Learning	Customer ID, Session Count, Average Time On Site, Pages Viewed, Total Orders, Churn, Last Purchase Date, Is Returning Customer, Satisfaction Score	Key Features Used, Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM)	Early identificati o n of high risk churn customers, Actionable insights for customer retention strategies	No exact accuracy values provided, Future Scope Not Explicitly Detailed, Handling of imbalanced churn labels.
Vidya Rajasekara n and Latha Tamilselva n (2023)	Predicting Customer Churn in E Commerce Using Statistical and ML methods	Tenure, Preferred login device, City tier Ware house to home distance, Satisfaction score, Order count, Day since last order, Cashback amount	Statistical Analysis, Implement ing Logistic Regressio n to predict the likelihood of customer churn.	Identificati o n of Churn Influencers, Business Impact, Identificati o n of Churn Influencers	Model Constraints, External Factors, Data Limitations
R Alexander, Maria Nancy A, Aswini E, Parwaz Singh Sarao (2023)	Comparativ e Performanc e Analysis using Machine Learning for Churn Prediction in E commerce	Customer ID, Gender, Senior Citizen, Partner, Dependents, Tenure, Monthly Charges, Total Charges, Payment Method, Churn	Available e commerce dataset with 5630 clients and 20 features, clustering analysis	Data preprocessi ng and feature selection, customer tenure, satisfaction score, order count	Future research could integrate real time customer interaction data for improved prediction accuracy

Dr. Snehal Rathi, Atharva Puranik, Vaishnavi Pophale, Prajwal Kutwal (2023)	A Survey and Implementation of Machine Learning Algorithms for Customer Churn Prediction	Tenure, preferred login device, city tier, warehouse to-home distance, satisfaction score, days since last order, and cashback amount.	Handling missing values, encoding categorical variables, Implementing a Soft Voting Classifier	Model Performance, Practical Implications	Dataset Specificity, Dynamic Customer Behaviour
Ishrat Jahan, Dr. Tahsina Farah Sanam (2022)	An Improved Machine Learning Based Customer Churn Prediction for Insight and Recommendation in E commerce	Customer ID, Total Orders, Average Order Value, Total Spend, Is Premium Member, Count, Support Tickets Opened Churn Label (1 = churn, 0 = retained)	Exploratory Data Analysis (EDA), Data Preprocessing, Model Building, Hyperparameter Tuning	Location based churn patterns, Product based behavior, Faster prediction speed	Over fitting Risk, Basic Feature Set, Small Dataset

CHAPTER-03
SYSTEM SPECIFICATION AND REQUIREMENT ANALYSIS

3.1 Requirement Analysis

Requirement analysis is the foundation of any successful software project. It involves understanding the needs of the stakeholders and defining what the system should accomplish. For the E-commerce Customer Churn Prediction project, requirement analysis focuses on identifying the patterns in customer behavior, understanding why customers stop using the platform, and predicting potential churn.

Key activities in requirement analysis include:

- Collecting customer data such as demographics, purchase history, browsing behavior, and transaction frequency.
- Understanding the business objectives, like improving customer retention and maximizing revenue.
- Identifying constraints such as data privacy regulations and system scalability.
- Communicating with stakeholders (business managers, marketing teams, and IT staff) to ensure alignment of system goals with business needs.

The goal is to ensure that the system is capable of accurately predicting which customers are likely to churn so that targeted retention strategies can be implemented.

Qualities of SRS Used in This Project

A Software Requirement Specification (SRS) document is a formal description of the system's functional and non-functional requirements. For this project, the SRS ensures clarity, consistency, and a roadmap for developers and stakeholders.

1. Correctness: The SRS should accurately reflect all the requirements identified during requirement analysis.
2. Completeness: All functional and non-functional requirements, data specifications, and constraints must be included.
3. Consistency: There should be no conflicting statements or ambiguities.
4. Clarity: Requirements should be easy to understand for both technical and non-technical stakeholders.

5. Verifiability: Each requirement should be testable or measurable to ensure it is implemented correctly.

Goals an SRS should achieve

- Serve as a communication tool between stakeholders and developers.
- Provide a clear roadmap for system design and development.
- Help in system validation and verification.
- Minimize project risks by clearly defining objectives and constraints.

3.1.1 Functional Requirements

Functional requirements define the core capabilities and operations that the system must perform. For E-commerce Customer Churn Prediction, the functional requirements include:

1. Data Collection and Integration: Collect customer data from multiple sources, including purchase history, website interactions, and demographic details.
2. Data Preprocessing: Handle missing data, remove outliers, normalize features, and convert categorical variables into machine-learning-friendly formats.
3. Churn Prediction Model: Implement machine learning algorithms (e.g., Random Forest, Logistic Regression, CNN) to predict customer churn.
4. Dashboard and Reporting: Provide visual dashboards showing churn probabilities, key insights, and retention strategies.
5. Alerts and Notifications: Notify the marketing team about customers at high risk of churn for timely intervention.
6. Model Evaluation: Generate performance metrics like accuracy, precision, recall, and F1-score to evaluate model performance.

3.1.2 Non-Functional Requirements

Non-functional requirements describe system attributes and quality standards rather than specific behaviors. Key non-functional requirements for this project include:

1. Performance: The system should provide predictions within a reasonable time, even for large datasets.
2. Scalability: The system should handle an increasing number of customers and growing data efficiently.

3. Scalability: The system should handle an increasing number of customers and growing data efficiently.
4. Security: Customer data must be protected using encryption and secure access control measures.
5. Reliability: The system should be available 24/7 with minimal downtime.
6. Usability: Dashboards and reports should be easy to understand and interact with for non-technical users.

3.2 Feasibility Study

1. Technical Feasibility

- The project relies on technologies like Python, machine learning libraries (Scikit-learn, TensorFlow), and data visualization tools.
- Available infrastructure (computers, cloud resources) is sufficient for data processing and model training.
- Skilled personnel are available to handle data preprocessing, model building, and system integration.

2. Operational Feasibility

- The system aligns with the company's goal of improving customer retention.
- Marketing and customer support teams can easily act on predictions generated by the system.
- The system's recommendations can be integrated into existing workflows without disruption.

3. Economic Feasibility

- Development costs, including manpower, software licenses, and cloud resources, are within budget.
- Benefits like increased customer retention, reduced churn, and higher revenue outweigh the implementation costs.

4. Schedule Feasibility

- Project milestones are realistic and achievable within the planned timeline.
- Adequate time has been allocated for data collection, preprocessing, model development, testing, and deployment.

3.3 Requirement Traceability Matrix (RTM)

The Requirement Traceability Matrix (RTM**)** is a critical tool used in software development and project management. It ensures that all project requirements are properly implemented and tested. RTM helps in tracking the requirements from the early stages of analysis to final implementation, ensuring no requirement is overlooked. In this project, the RTM helps map functional and non-functional requirements to their corresponding modules, features, and testing criteria in the E-commerce Customer Churn Prediction system.

The main purpose of RTM in this project is to:

- Track all functional and non-functional requirements systematically.
- Ensure complete coverage of all business needs in the implemented system.
- Facilitate verification and validation during testing.
- Provide a clear linkage between requirements, design, and testing artifacts.

Table 3.1: Requirement Traceability Matrix(RTM)

Requirement ID	Requirement Description	Type	Module / Feature	Priority	Test Case / Verification
R1	Collect historical customer transaction data from the e-commerce platform, including demographics, purchase history, and engagement data.	Functional	Data Acquisition	High	Verify successful import from CSV files, SQL/NoSQL databases, and API endpoints.
R2	Clean and preprocess data by handling missing values, outliers, and	Functional	Data Preprocessing	High	Check null value handling, outlier removal, and data

	inconsistent entries.				consistency across records.
R3	Encode categorical variables using techniques like One-Hot Encoding or Label Encoding to prepare data for machine learning models.	Functional	Feature Engineering	High	Verify all categorical columns are correctly transformed and no original values remain unprocessed.
R4	Split the dataset into training, validation, and testing subsets to evaluate model performance effectively.	Functional	Data Preparation	High	Confirm dataset is divided according to the chosen ratio (e.g., 70:15:15) and no data leakage occurs.
R5	Implement predictive models such as Logistic Regression, Random Forest, and CNN to predict customer churn.	Functional	Model Developme nt	High	Verify models are trained on the training set and predictions are generated on the test set.
R6	Evaluate model performance using metrics including Accuracy, Precision, Recall, F1-Score, ROC-AUC.	Functional	Model Evaluation	High	Compare predicted outputs with actual outcomes and calculate performance metrics.
R7	Provide a visualization dashboard displaying churn trends, retention rates, and top influencing features.	Functional	Data Visualizatio n	Mediu m	Verify charts and graphs are correct, interactive, and show expected insights.
R8	Ensure customer data security and privacy by implementing access control,	Non-Functional	Security	High	Test access permissions, encryption for sensitive data, and

	encryption, and secure storage.				secure storage mechanisms.
R9	Ensure the system can scale efficiently to handle increasing data volumes without performance degradation.	Non-Functional	System Performance	Medium	Perform load testing with large datasets to measure response time and resource usage.
R10	Generate actionable reports for marketing and retention teams with clear insights and recommendations.	Functional	Reporting Module	Medium	Verify reports are accurate, include key metrics, and are delivered on schedule.
R11	Maintain high usability and intuitive design for users interacting with the system dashboard.	Non-Functional	User Interface	Medium	Conduct user testing sessions to verify dashboard ease-of-use and navigation.
R12	Enable real-time updates of churn predictions whenever new customer data is available.	Functional	Real-time Processing	Medium	Test system with streaming data inputs and confirm immediate prediction updates.
R13	Provide logging and monitoring mechanisms to track system performance and errors.	Non-Functional	System Maintenance	Medium	Verify logs are generated for all critical actions and errors, and alerts are triggered correctly.
R14	Support export of predictions and reports in multiple formats (e.g., CSV, PDF).	Functional	Reporting Module	Low	Verify that all outputs can be exported correctly in required formats.

3.4 System Architecture Diagram

The system architecture of the E-commerce Customer Churn Prediction project represents how raw customer information from the online platform flows through several analytical stages to finally produce churn predictions and actionable insights. The architecture is designed to be scalable, data-driven, and capable of supporting both real-time and batch predictions.

The entire workflow can be understood in the following major components:

1. Customer Data Sources

The architecture begins with diverse data sources generated by the e-commerce platform. These data sources capture multiple dimensions of customer behaviour and interactions:

- CRM Data**:** Personal details, contact information, past communication logs.
- Billing & Transaction Data: Orders, payments, refunds, cart history.
- Usage Logs: App/website interactions, browsing frequency, product views.
- Support Tickets: Complaints, queries, service escalations.

These datasets form the foundation of the churn prediction model, ensuring a 360-degree view of each customer.

2. Data Collection Layer

This layer acts as a bridge between raw data and the analytical process. It is responsible for:

- Fetching structured and unstructured data from all sources
- Standardizing formats (CSV, JSON, Database tables)
- Storing data in a centralized repository such as a data warehouse or cloud storage

By consolidating all customer-related information, the system avoids inconsistencies and ensures smooth data flow for the next stages.

3. Data Preprocessing Module

Before applying machine learning, the collected data must be cleaned and made analysis-ready. This module handles:

- Data Cleaning: Removing duplicates, correcting invalid entries
- Handling Missing Values: Using mean, median, mode, or predictive imputation
- Outlier Detection: Eliminating extreme values like abnormal purchase spikes
- Feature Encoding: Converting categorical variables to numerical form (e.g., Gender → 0/1)

Preprocessing ensures that the data is accurate, consistent, and aligned with the model's requirements.

4. Feature Engineering Layer

This is one of the most crucial components of the architecture. Here, new meaningful attributes are created from the raw data to better represent customer behaviour:

- RFM Features: Recency (last purchase), Frequency (number of purchases), Monetary value
- Usage Patterns: Login frequency, browsing time, product category preferences
- Complaint Metrics: Number of support tickets, issue severity
- Engagement History: Discount usage, loyalty points, session duration

These engineered features help the machine learning model understand deeper behavioural patterns that typically signal churn.

5. Model Training Component

At this stage, machine learning algorithms are applied to the processed data:

1. Train-Test Split ensures fair model evaluation.
2. Algorithms used include:

    - Random Forest: Good for understanding feature importance
    - XGBoost: High accuracy and robust performance
    - Logistic Regression: Interpretable and efficient for binary classification

6. Model Evaluation Module

After training, the model's performance is assessed using various metrics:

- Accuracy & ROC-AUC: Overall performance measurement
- Precision & Recall: Ability to correctly identify churners
- Confusion Matrix: Breakdown of correct and incorrect predictions

7. Model Deployment Layer

Once the best model is finalized, it is deployed into the live system using:

- REST API: For real-time scoring during customer sessions
- Batch Processing: Scheduled predictions for business teams
- MLOps Pipeline: Automation of monitoring, updating, and retraining the model.

8. Churn Prediction Service

This service processes incoming data and generates churn outputs:

- Real-time predictions when a customer interacts with the platform.
- Batch predictions during weekly or monthly risk assessments.

The service ensures fast, efficient, and on-demand access to churn scores.

9. Dashboard & Alerting Module

The final component visualizes the churn results for business stakeholders:

- Customer Churn Probability Scores
- Risk Segments: High, medium, low.
- Retention Recommendations: Discounts, loyalty reward suggestions, targeted emails.
- Prioritization of At-Risk Customers : Customers with higher churn probability are clearly highlighted, enabling teams to focus their time and resources on those who need immediate attention rather than applying generic retention strategies.
- Continuous Monitoring and Improvement : The visualization component allows businesses to track how churn risk changes over time, helping evaluate the effectiveness of retention campaigns and adjust strategies accordingly.

This helps decision-makers take proactive steps to retain valuable customers and reduce churn.

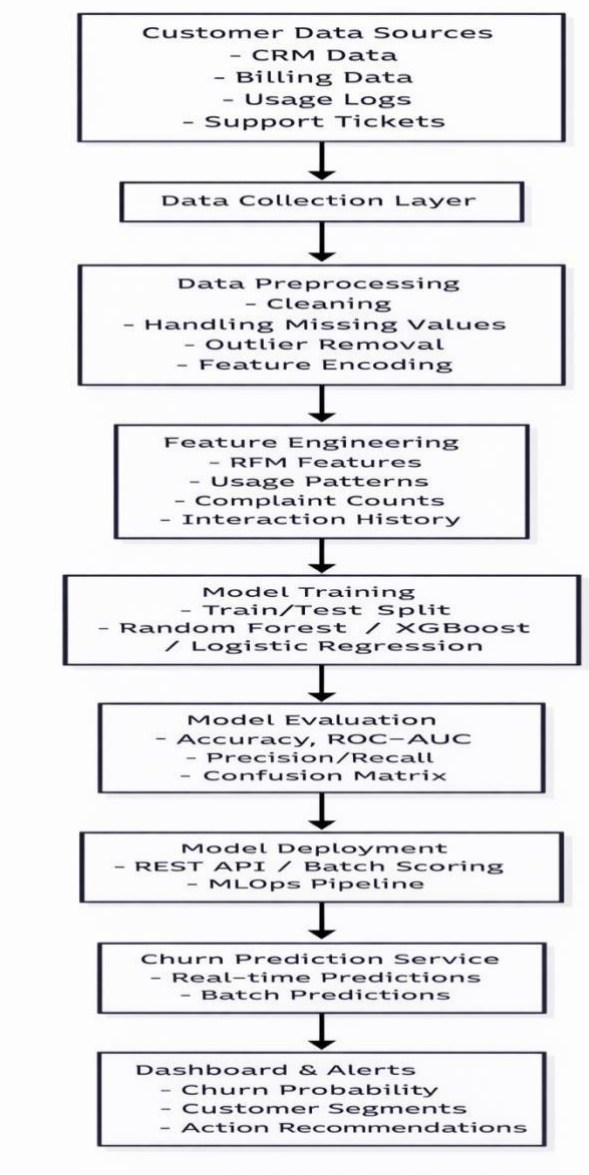

Figure 3.1 System Architecture

3.5 Use Case Diagram

The Use Case Diagram for the E-Commerce Customer Churn Prediction System illustrates how different actors interact with the system and how the system responds to customer behavior in order to identify churn risk and support retention activities. It visually represents the end-to-end flow from customer interactions to automated churn predictions and final decision-making.

Actors

1. Customer

The customer is the primary actor who interacts with the e-commerce platform by browsing products, making purchases, raising complaints, or contacting support. Their usage patterns and engagement behavior are continuously monitored to identify churn risk.

2. Churn Prediction System

This represents the intelligence of the project. It captures events, stores relevant customer data, analyzes behavior, and predicts churn probability. It makes automated decisions based on computed risk.

3. Retention Team / Business Team

This actor receives alerts and takes corrective actions. They intervene when the system detects a high-risk customer and recommends retention strategies such as discounts, personalized communication, or priority support.

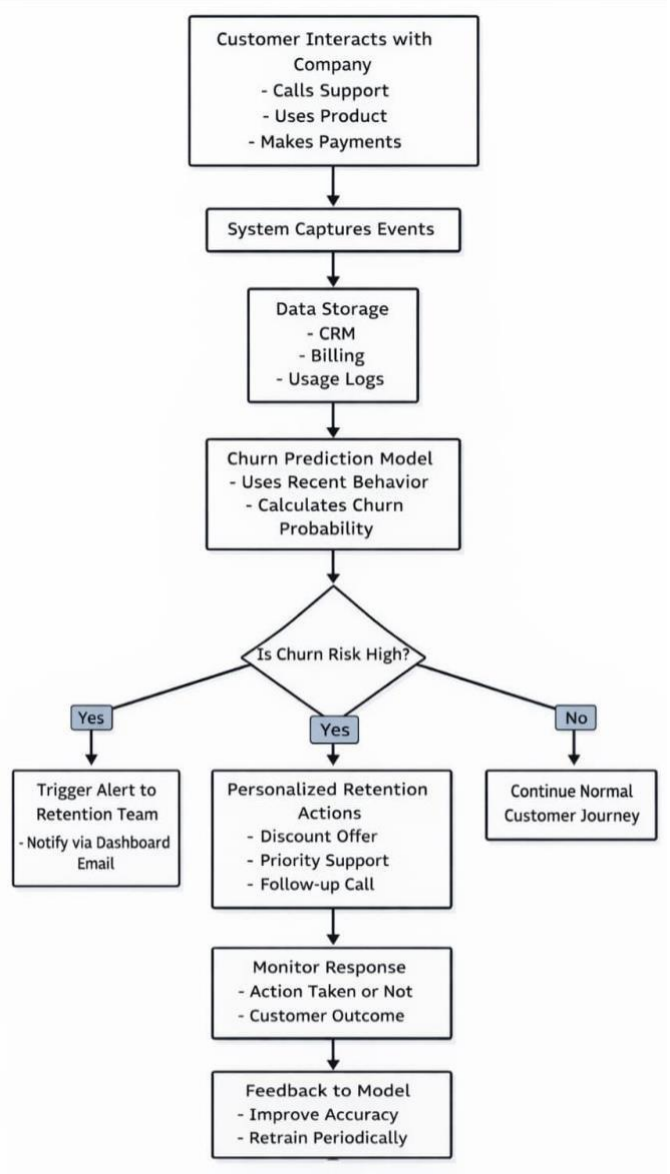

Figure 3.2 Use Case Diagram

Use Cases

1. Customer Interacts with the Platform

Customers perform different activities such as viewing products, making payments, contacting support, or browsing categories. Every interaction creates an event that may indicate their level of satisfaction or disengagement.

2. System Captures Customer Events

The system automatically records customer actions in real time. This includes behavioral data, transaction data, and support data. These signals later help the model understand usage trends.

3. Data Gets Stored in CRM, Billing, and Usage Logs

All recorded events are saved securely in multiple data repositories. This storage layer ensures:

- All customer profiles remain up-to-date
- Billing records reflect every transaction
- Usage logs document detailed user activity

This consolidated data forms the backbone of churn prediction.

4. Churn Prediction Model Calculates Churn Probability

The machine learning model analyzes the latest customer behavior and derives a churn score. It determines:

- Whether a customer is still engaged
- Whether their activity has dropped
- Whether they show early signs of dissatisfaction

A high churn probability triggers the next set of actions.

5. Decision Point: “Is the Churn Risk High?”

When the predicted churn risk is high**,** the system immediately responds by activating the retention mechanism. Alerts are sent to the relevant teams through dashboards or notifications, and targeted retention actions are initiated. These actions are designed to re-engage the customer and reduce the likelihood of churn.

On the other hand, when the churn risk is low**,** no intervention is required. The customer is allowed to continue their normal journey on the platform without any disruption, ensuring a smooth and uninterrupted user experience.

This decision logic helps the system focus retention efforts only where they are truly needed, optimizing resources while maintaining customer satisfaction.

6. Alert Sent to the Retention Team (For High Risk)

In some cases, the system immediately notifies the retention team via:

- Dashboard alerts
- Email notifications
- Automated reports

This ensures high-value customers are not lost unexpectedly.

7. Personalized Retention Actions Are Initiated

Based on the churn score, the system recommends tailored actions such as:

- Special discount codes
- Loyalty points
- Priority customer care
- Follow-up calls
- Personalized messages or offers

These actions are designed to re-engage the customer.

8. Monitor Customer Response

The system evaluates the outcome of the retention effort:

- Did the customer respond?
- Did they redeem the offer?
- Did their activity level improve?
- Observes communication interaction**,** including email opens, link clicks, or app notification responses.
- Monitors changes in customer activity levels such as logins, purchases, and browsing frequency.

By carefully monitoring customer responses, the system gains valuable insights into customer behavior, allowing businesses to fine-tune their retention efforts and build stronger, longer-lasting customer relationships.

9. Feedback to the Model

Finally, the outcomes are sent back to the machine learning model to improve prediction accuracy.

The model:

- Retrains periodically.
- Learns from successful or failed retention attempts.
- Continually adapts to changing customer behavior patterns.
- Reduces false churn alerts by refining prediction thresholds based on past outcomes.
- Evaluates real customer reactions to offers, discounts, or loyalty programs to understand what truly works.

This closed-loop learning approach ensures that the churn prediction system does not remain static. Instead, it evolves with customer behavior, enabling businesses to take smarter, data-driven actions and maintain long-term customer relationships.

CHAPTER-04

DATASETS

Dataset Overview

For this project, a custom e-commerce customer dataset was used, containing a mix of demographic, behavioral, and subscription-related attributes. The dataset aims to capture the everyday interactions of customers with an online shopping platform and reflect the patterns that may lead them either to stay loyal or to churn.

The dataset with the following key features:

- **CustomerID:** A unique identifier assigned to each customer.
- **Age & Gender:** Basic demographic details that help understand different customer personas.
- **Tenure:** Total months the customer has been associated with the platform.
- **Usage Frequency:** How often a customer interacts with the platform (weekly/monthly).
- **Support Calls:** Number of times a customer contacted customer support, a strong indicator of dissatisfaction.
- **Payment Method:** Mode of payment such as Standard, Basic, Premium, etc.
- **Subscription Type:** Indicates the customer's membership tier.
- **Contract Length:** Duration of the customer’s subscription (Monthly, Quarterly, or Annual).
- **Total Spend:** The overall amount the customer has spent on the platform.
- **Last Interaction:** Number of days since the customer last interacted.
- **Churn:** A binary flag 1 means the customer has churned, **0** means they are still active.

Overall, the dataset captures the behavioral, financial, and demographic signals needed to understand the drivers of churn in an e-commerce environment.

Dataset Characteristics

The dataset used in this project is designed to capture the complete behavioral and demographic profile of e-commerce customers, making it suitable for understanding the factors that lead to customer churn. Its structure reflects real-world customer interactions

and provides a balanced view of both active and churned users. Below are the key characteristics that define this dataset:

1. Multidimensional Nature

The dataset includes a mix of demographic information, behavioral patterns, subscription details,and financial indicators.

This combination allows the model to learn relationships across multiple angles, such as:

- Does age influence churn?
- Do higher support calls correlate with dissatisfaction?
- Does total spending reflect loyalty?

This multidimensional structure strengthens the model's ability to identify subtle churn patterns.

2. Balanced Representation of Customer Behavior

Each row represents a unique customer, showcasing personal traits (like gender and age), long-term interactions (tenure), and short-term engagement (last interaction). This balance helps in:

- Distinguishing long-term customers from newer ones
- Comparing highly active customers with occasionally active ones
- Analyzing dissatisfaction signals such as support calls

The dataset mirrors real e-commerce operational data where customers behave differently based on their needs and experiences.

3. Mix of Categorical and Numerical Attributes

The dataset is composed of:

- Numerical fields (Age, Tenure, Usage Frequency, Support Calls, Total Spend, Last Interaction)
- Categorical fields (Gender, Payment Method, Subscription Type, Contract Length)

This mixture makes the dataset diverse and ideal for various machine learning algorithms. Categorical attributes help understand customer choices, while numerical attributes quantify their behavior.

4. Clear Churn Indicator

A binary Churn column (0 = active, 1 = churned) provides a clear target variable for Supervised learning. This straightforward flag helps the model learn distinct patterns between customers who stayed and those who left.

5. Realistic Customer Engagement Signals

The dataset reflects realistic customer behavior seen in typical e-commerce platforms:

- Usage Frequency shows how often customers interact.
- Last Interaction captures how recently they used the platform a strong churn predictor.
- Subscription Type and Contract Length reveal customer commitment levels.
- Total Spend helps identify high-value customers who are important for retention strategies.

These attributes mimic the behavioral clues actual businesses monitor when assessing churn risk.

6. Variability in Spending and Tenure

There is a wide range in both spending and tenure**:**

- Some customers have spent over 900 units, while others spent under 200.
- Tenure ranges from just a few months to more than 50 months.
- This variation highlights the presence of both new and long-term customers within the dataset.
- This variation highlights the presence of both new and long-term customers within the dataset.
- High-spending, long-tenure customers often show strong loyalty, while low-spending or short-tenure users may be at a higher risk of churn.
- It also improves the model’s ability to generalize and make accurate predictions for a wide range of customer types.

This variability ensures that the model is trained across diverse customer profiles, supporting robust churn prediction.

7. Compact Yet Insightful Dataset

Although the dataset is not large, it is rich in information**.**Every attribute plays a meaningful role in understanding customer decisions.The compact size makes it easy to process while still delivering enough depth for building an accurate model.

8. Ready for Predictive Modelling

Due to its structured layout and well-defined fields, the dataset is perfectly suited for:

- Classification models
- Feature engineering
- Pattern recognition
- Customer segmentation

Its clarity ensures smooth preprocessing, training, and evaluation during the prediction phase.

4.1 Source of the Dataset

The dataset used for this project **is** manually created and structured based on common commerce usage patterns**.**It represents realistic customer activity typically found in online retail platforms. The structure is built to highlight:

- Customer engagement levels
- Spending behavior
- Satisfaction indicators (e.g., support calls)
- Subscription and contract details

Although not taken from a public repository, the dataset closely resembles real-world customer data used in churn analytics across the e-commerce industry.

4.2 Preprocessing Steps

Data preprocessing is the process of transforming raw, unorganized data into a clean and structured format that can be effectively used by machine learning models. Before training the churn prediction model, multiple preprocessing steps were carried out to make sure the data was accurate, consistent, and suitable for machine learning. These steps helped improve model performance and reduced the chances of misleading predictions. Careful preprocessing ensures that the model learns meaningful patterns from the data

rather than being influenced by noise or inconsistencies, leading to more reliable and actionable churn predictions for business decision-making. If these issues are not handled properly, they can negatively impact model accuracy and reliability.

1. Data Cleaning

- Verified that all customer IDs were unique.
- Checked for missing values in critical fields like Age, Tenure, Total Spend, and Churn.
- Ensured numerical features such as Tenure, Support Calls, and Total Spend were valid and non-negative.

2. Encoding Categorical Values

Certain attributes such as Gender, Payment Method, Subscription Type, and Contract Length are categorical.

To make them machine-readable:

- Label Encoding and One-Hot Encoding were applied depending on the algorithm used.

3. Feature Scaling

- Numerical fields (Total Spend, Tenure, Usage Frequency) were scaled to a uniform range to prevent domination of larger values during training.

4. Outlier Detection

- Checked for unusual spikes in values such as extremely high spending or negative interaction counts.

5. Data Splitting

- The dataset was divided into training and testing sets to evaluate model performance fairly.
- Proper data splitting improves the credibility and reliability of prediction results.
- The training set was used to help the model learn patterns and relationships within customer data. The testing set was kept separate to assess how well the model performs on unseen data.

Overall, preprocessing ensured that the data was structured, noise-free, and suitable for predictive modeling.

4.3 Samples

- **Diverse Customer Demographics:**

The sample includes customers across ages 22–68, representing a wide customer base. Both males and females are included, showing gender diversity in platform usage. The broad age range captures different shopping preferences and digital behavior patterns across generations.

Gender diversity ensures that the model does not favor a specific group and remains unbiased in its predictions. Younger users may respond better to digital offers and fast interactions, while older users often value reliability and service quality. This inclusive demographic representation makes the churn prediction system more realistic and adaptable to real-world e-commerce environments.

- **Varied Tenure Levels:**

Some customers have been with the platform for only a few months, whereas others have over 50+ months of tenure. This helps compare new vs. long-term customer behavior. New customers often show exploratory behavior and may be more sensitive to early experiences. Long-term customers usually reflect established trust and familiarity with the platform.

- **Usage and Interaction Behavior:**

Usage frequency values indicate how actively customers engage. Some customers log in almost daily, while others rarely interact. The Support Calls attribute reveals how many times they encountered issues a strong churn signal when high. A high number of support calls can point to repeated problems or dissatisfaction. Regular logins usually indicate strong interest and habitual platform usage.

	A	B	C	D	E	F	G	H	I	J	K	L
1	CustomerID	Age	Gender	Tenure	Usage Frequency	Support Calls	Payment Delay	Subscription Type	Contract Length	Total Spend	Last Interaction	Churn
2	1	22	Female	25	14	4	27	Basic	Monthly	598	9	1
3	2	41	Female	28	28	7	13	Standard	Monthly	584	20	0
4	3	47	Male	27	10	2	29	Premium	Annual	757	21	0
5	4	35	Male	9	12	5	17	Premium	Quarterly	232	18	0
6	5	53	Female	58	24	9	2	Standard	Annual	533	18	0
7	6	30	Male	41	14	10	10	Premium	Monthly	500	29	0
8	7	47	Female	37	15	9	28	Basic	Quarterly	574	14	1
9	8	54	Female	36	11	0	18	Standard	Monthly	323	16	0
10	9	36	Male	20	5	10	8	Basic	Monthly	687	8	0
11	10	65	Male	8	4	2	23	Basic	Annual	995	10	0
12	11	46	Female	42	27	9	21	Standard	Annual	526	3	1
13	12	56	Male	13	23	5	14	Basic	Quarterly	187	1	0
14	13	31	Male	2	7	0	25	Premium	Quarterly	758	24	0
15	14	42	Male	46	27	5	8	Premium	Quarterly	438	30	0
16	15	59	Male	21	17	2	14	Premium	Quarterly	663	15	0
17	16	35	Female	1	3	7	3	Basic	Monthly	677	25	1
18	17	29	Male	54	3	6	2	Basic	Monthly	636	22	0
19	18	45	Male	9	30	4	25	Basic	Annual	127	18	0
20	19	65	Female	40	2	1	6	Premium	Annual	396	21	0
21	20	62	Male	39	19	2	15	Premium	Quarterly	202	24	0
22	21	48	Male	28	7	1	21	Premium	Monthly	925	13	0
23	22	36	Female	58	4	0	1	Premium	Quarterly	463	26	0

Figure 4.1 Dataset Samples

- **Subscription and Contract Variety:**

Customers on the platform subscribe through different contract durations such as monthly, quarterly, and annual plans, and these plans are available across multiple subscription tiers including Basic, Standard, and Premium. The choice of contract length often reflects the customer's level of confidence and trust in the platform.

Short-term plans, like monthly subscriptions, usually indicate flexibility and lower commitment, as customers can easily discontinue the service if their expectations are not met. In contrast, longer-term contracts such as quarterly or annual subscriptions suggest higher satisfaction levels and a stronger intention to continue using the service.

Subscription tier selection also plays a significant role in understanding customer behavior. Customers opting for Standard or Premium plans often seek advanced features, better service quality, or exclusive benefits, which can signal deeper engagement with the platform. These users are generally more invested and less likely to churn compared to Basic plan users.

- **Spending Patterns:**

Total expenditure varies significantly, with some customers spending over 900 units while others stay below 200. These spending differences help differentiate dormant users from high-value ones. Customers with consistently high spending often indicate strong trust and satisfaction with the platform. Sudden changes in spending behavior can act as early warning signs of potential churn. Low-spending customers might be price-sensitive and more responsive to discounts or offers. Stable spending over time usually reflects long-term engagement with the platform. Understanding these patterns helps businesses segment customers and design personalized retention strategies.

- **Churn Distribution:**

The Churn column (0 or 1) shows which customers discontinued using the platform. The sample includes both churned and active customers, helping the model learn contrasting behavior patterns. It allows the system to capture subtle differences in activity, spending, and engagement levels. Having both churned and non-churned customers ensures balanced learning during model training. It also helps the model recognize early warning signs of churn, supporting timely and proactive retention strategies. A well distributed churn dataset supports balanced learning and improves prediction accuracy.

Algorithms Used

To build an effective churn prediction system, multiple machine learning algorithms were explored and compared. Each algorithm offers a different viewpoint on customer behavior, helping the system identify patterns that signal whether a customer is likely to stay or leave. The algorithms used in this project are described below:

1. Logistic Regression

Logistic Regression is one of the simplest yet powerful classification algorithms. It predicts the probability of churn by learning relationships between customer attributes such as age, spending, support calls, and subscription type.

- Works well with both numerical and categorical variables
- Provides clear interpretability of feature importance
- Helps understand how each factor increases or decreases churn risk
- Acts as a reliable benchmark**,** making it easy to compare performance with more complex models.
- Offers fast training and prediction**,** enabling quick model updates.

It serves as a strong baseline model for churn prediction.

2. Random Forest Classifier

Random Forest is an ensemble learning technique that builds multiple decision trees and combines their outputs to improve prediction accuracy.

- Handles noisy, nonlinear customer behavior
- Works well even when some features are missing or unbalanced
- Automatically captures interactions between variables

This algorithm is highly suitable for churn prediction because it can detect complex patterns such as irregular usage, high support calls, or sudden drops in engagement.

3. XGBoost (Extreme Gradient Boosting)

XGBoost is one of the most advanced boosting algorithms and is known for delivering high accuracy in classification tasks.

- Learns from errors of previous models to refine predictions

- Efficient with large datasets and complex relationships
- Handles imbalanced churn data by applying built-in regularization
- Processes large datasets efficiently**,** making it suitable for real-world e-commerce data.
- Supports fast training and tuning**,** enabling experimentation with different parameters to optimize performance.

In this project, XGBoost often produced the best results because churn patterns are subtle and require deep pattern recognition.

4. Decision Tree Classifier (Supporting Model)

A single decision tree was used during the initial analysis to understand the most influential churn-related attributes.

- Provides easy visualization of customer decision paths
- Helps explain which features directly contribute to churn
- Useful for early exploration and feature engineering
- Assists in validating domain knowledge**,** confirming whether model insights align with real-world expectations.
- Mimics human decision-making**,** making its logic easy for non-technical stakeholders to understand.

Although not the final model, the decision tree helped refine feature selection.

Software Metrics

Software metrics play a key role in evaluating the quality, performance, and reliability of the churn prediction system. In this project, several metrics were used to ensure that the final model is not only accurate but also efficient and scalable.

1. Accuracy

Accuracy measures how many predictions were correct out of all predictions made. Although useful, accuracy alone is not enough in churn prediction because churn datasets are often imbalanced (more non-churners than churners).

- High accuracy reflects overall model performance
- Helps compare different algorithms

- Ensures the model does not make frequent mistakes.
- Accuracy gives a quick overview of how well the model performs on the dataset as a whole.

2. Precision

Precision indicates how many customers predicted as "churners" were actually churners.

- Helps avoid targeting the wrong customers
- Reduces unnecessary retention costs
- Important for marketing and decision-making teams
- A model with high precision ensures that retention campaigns are focused on the right individuals.

3. Recall

Recall tells how many actual churners the model successfully identified.

- Critical for preventing revenue loss
- Helps ensure that churners are not overlooked
- Important in customer retention workflows

High recall is essential because missing a churner can directly affect business profitability.

4. F1-Score

The F1-Score is the harmonic mean of precision and recall.

It provides a balanced evaluation, especially when the dataset has unequal classes.

- Useful when churn vs. non-churn counts differ
- Helps select the best algorithm for deployment
- Ensures fairness in evaluating model performance
- It is especially valuable in business scenarios where missing a churner or wrongly flagging a loyal customer both have costs.
- Offers a more realistic view of model effectiveness by balancing accuracy with practical business impact.

5. ROC-AUC Score

The ROC-AUC score evaluates how well the model distinguishes between churners and non-churners.

- Higher AUC means better discrimination
- Helps understand the model's confidence across thresholds
- Common benchmark for churn prediction systems
- It provides a threshold-independent measure, making it useful when churn probabilities need to be compared rather than fixed decisions made.

A strong AUC value indicates that the model is capable of ranking customers based on their churn risk reliably.

6. Confusion Matrix

The confusion matrix provides a detailed breakdown of correct and incorrect predictions.

- Shows true positives, false positives, true negatives, and false negatives
- Helps pinpoint weaknesses in the model
- Useful in tuning the model to reduce misclassifications.

7. Computational Efficiency

Along with accuracy metrics, the model's performance speed and resource usage were evaluated.

- Assesses how fast the model predicts churn
- Important for real-time applications
- Helps optimize the deployment pipeline

Efficiency ensures that even large e-commerce platforms can use the model without delays.

CHAPTER-05

SYSTEM DESIGN

5.1 High Level Design (HLD)

The High-Level Design provides an overall architectural view of how the churn prediction system works within an e-commerce platform. It explains the major building blocks, the interaction between them, and the flow of data from the source to the final prediction stage.

1. Data Sources

The system collects raw data from multiple internal platforms. Typical sources used for churn prediction include:

- Customer profile information such as age, location, and account tenure
- Transaction history like orders, cancellations, refunds, and cart behaviour
- Support interactions such as complaints, tickets, and chat logs
- Website usage statistics including login frequency, session duration, and clicks

These heterogeneous data sources ensure that both behavioural and historical factors are captured, which are known to influence customer churn.

2. Data Pipeline

Once the data is collected, it is passed through a structured pipeline. The pipeline is responsible for organising, cleaning and storing the information for downstream processing.

- Data from different sources is merged and stored in a common format
- Pre-processing steps include handling duplicates, missing values, and date conversions
- The pipeline may run on automated scheduling tools such as Airflow**,** ensuring timely updates
- Automation minimizes human errors and improves the overall efficiency of data processing.

This pipeline makes sure that the data is fresh and reliable before it is used for model building.

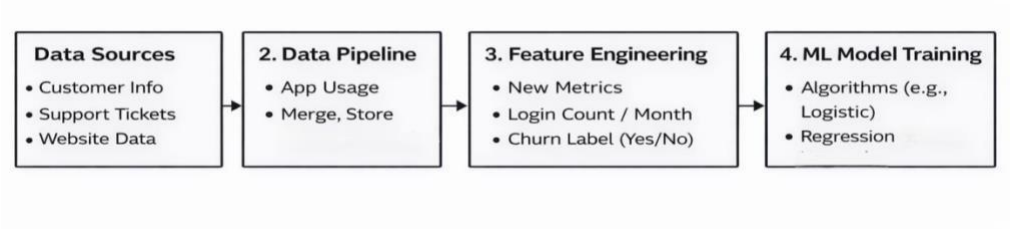

Figure 5.1 High Level Design

3. Feature Engineering

Feature engineering plays a central role in predicting churn. It transforms raw data into meaningful indicators.

Examples:

- Login frequency per month
- Average purchase value over the last 90 days
- Time since last purchase
- Binary churn label (Yes/No) based on inactivity or cancellation

New features are created to capture customer behaviour trends. These engineered metrics help the model identify patterns that strongly correlate with churn.

4. Machine Learning Model Training

The cleaned and transformed data is used to train machine learning models. Suitable algorithms for churn prediction include:

- Logistic regression
- Decision trees
- Random Forest
- Gradient boosting techniques (XGBoost)

The goal of this stage is to learn relationships between customer behaviour and churn outcomes so that the system can predict which customers are at high risk of leaving. During training, the model identifies key patterns and trends that influence customer retention. Multiple models can be compared to select the one that delivers the best performance and stability. This training process enables the system to make proactive predictions, allowing businesses to act before customers disengage completely.

5.2 Low Level Design (LLD)

The Low-Level Design provides the actual technical details, tools, and components used to implement each block in the architecture. It focuses on implementation logic, storage technology, and deployment.

1. Data Layer

The data layer stores the raw and processed information used by the system. Common technologies include:

- Databases: MySQL
- Cloud storage: Amazon S3, Snowflake
- ETL tools: Apache NiFi

This layer ensures durability and availability of customer data. Access is controlled to maintain security and data integrity.

2. Feature Store

A feature store acts as a central location for storing computed features that are ready for use in model training or predictions.

- Features are stored in SQL databases or NoSQL stores
- Redis or similar caching tools may be used for fast access
- The store avoids recomputing features repeatedly

This part supports both offline training and real-time inference.

3. Model Training Environment

The model is trained using machine learning libraries. The typical stack includes:

- Scikit-learn for classical algorithms
- TensorFlow / PyTorch for deep learning
- Jupyter Notebook or integrated ML pipelines for experimentation

The model is validated using performance metrics such as accuracy, recall, F1-score and AUC. Once the best model is selected, it is saved for deployment.

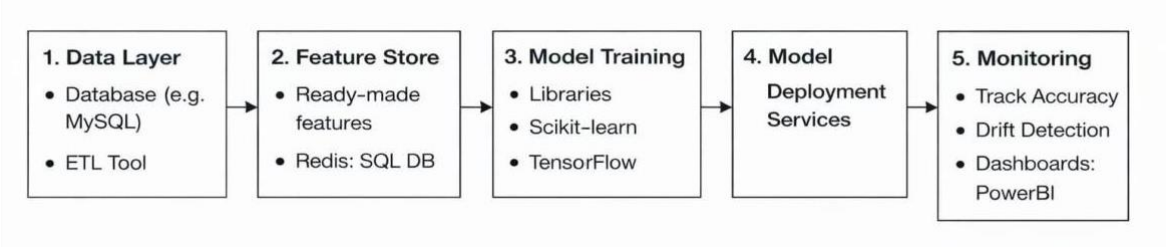

Figure 5.2 Low Level Design

4. Model Deployment and Services

After training, the model is deployed as a service so that it can produce real-time or batch predictions. Deployment options include:

- FastAPI / Flask APIs
- Docker containers for portability
- Hosting on cloud services such as AWS, GCP or Azure

Customers who exhibit high churn probability are flagged so that marketing campaigns or retention strategies can be applied immediately.

5. Monitoring and Reporting

Continuous monitoring ensures the model remains accurate and reliable. Key activities include:

- Tracking prediction accuracy over time
- Detecting data drift or model degradation

Alerts are raised when major variations are observed, enabling timely model retraining or tuning.

5.3 UML Diagrams

UML diagrams help to visualize the structure and behaviour of the system. They give a blueprint of how data flows and how different modules interact with each other during execution.

5.3.1 Class Diagram

The class diagram for the e-commerce customer churn prediction system represents the logical structure of the main components involved in data processing, model building

and evaluation. Each class has specific responsibilities and methods, and they collaborate to generate churn predictions.

1. Dataset Class

The Dataset class forms the foundation of the system.It is responsible for collecting and organizing all input data that is required to train and test the churn prediction model. This data typically comes from the e-commerce platform.

Attributes include:

- demographics: age, city, gender, account age
- transactions: number of orders, average purchase value
- support_data: number of complaints, resolution history

The data collected is passed to the next stage for cleaning and preprocessing.

2. Preprocessing Class

The Preprocessing class handles all the data transformation steps. Its role is to convert raw data into a clean and balanced format suitable for model training.

Important methods:

- median_imputation(): fills missing values using the median
- SMOTE(): balances the dataset by oversampling minority churn class
- feature_scaling(): normalizes numerical values to a common scale

After transformation, the output is a processed dataset**,** which ensures the model can learn patterns effectively.

3. Model Class

The Model class represents the core intelligence of the E-Commerce Customer Churn Prediction System and plays a vital role in decision-making. It is responsible for learning patterns from historical customer data and using those patterns to predict the likelihood of churn for new or existing customers.

This class encapsulates the complete machine learning workflow, ensuring that model training and prediction processes are handled in a structured and reusable manner.

By acting as a centralized controller, it enables the system to manage and switch between different machine learning algorithms efficiently.

Primary Methods:

- **train():**

This method trains the selected machine learning model using preprocessed and feature-engineered customer data. It includes steps such as model initialization, learning from historical churn patterns, and updating model parameters to improve accuracy.

- **predict():**

This method generates churn probability scores for new or unseen customers based on their behavioral, transactional, and engagement features. The output helps classify customers into churn risk categories such as low, medium, or high risk.

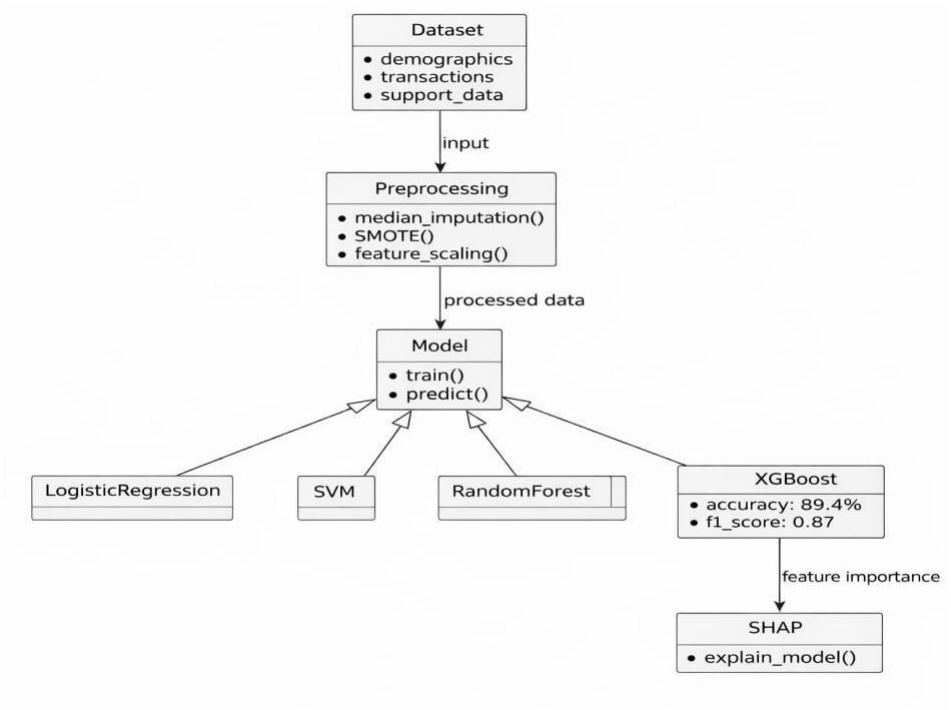

Figure 5.3 Class Diagram

4. Machine Learning Algorithm Classes

The system uses several algorithm-specific subclasses. Each inherits from the Model class and provides its own implementation.

Logistic Regression Class

- Used for interpretable churn modelling
- Suitable for binary classification

- Produces probability values for churn
- Serves as a strong baseline model, helping compare performance with more complex algorithms while maintaining transparency.

SVM (Support Vector Machine) Class

- Handles complex patterns in customer behaviour
- Good performance with high-dimensional features

Random Forest Class

- Combines multiple trees to improve accuracy
- Handles non-linear relationships efficiently

XGBoost Class

- Gradient boosting algorithm
- Achieved high accuracy in experiments: 89.4% accuracy and F1-score 0.87

Each algorithm is trained independently, and the best performing model can be chosen for final deployment.

5. SHAP Class

To make the churn prediction system easier to understand and trust, an interpretability layer using the SHAP class is included. This component explains the model's decisions by showing how different features contribute to churn predictions. In simple terms, the explain_model() method highlights which factors have the strongest impact on customer churn. This is especially valuable for business and marketing teams, as it turns complex model outputs into clear, actionable insights.

For example, the system can reveal that customers who log in less frequently, have not placed orders for a long time, or raise multiple complaints are more likely to churn. By understanding these patterns, teams can design focused retention strategies. Overall, this interpretability improves transparency, builds confidence in the model's results, and supports better decision-making. This explainability layer helps organizations move from reactive actions to proactive decision-making by identifying early warning signals of churn, allowing teams to intervene at the right time with personalized and effective retention efforts.

5.3.2. Sequence Diagram

The sequence diagram illustrates the order of interactions between different components of the churn prediction system. It focuses on how the system responds when a business user requests churn information and additional insights from the dashboard. The diagram captures real-time flow, starting from data retrieval to prediction and visualization.

1. Request from Business User

The process begins when a business user opens the platform and requests churn analysis.
This request is made from the dashboard interface, which acts as the main entry point for the system.

- The user initiates: “Request churn results”
- The dashboard forwards this request to the Prediction Service

This service controls the entire prediction workflow.

2. Fetching and Preparing Customer Data

Once the request is received, the Prediction Service communicates with the Database to collect customer-related information. The database provides:

- Transaction history
- Order frequency
- Complaints raised
- Account details and usage behaviour

After data retrieval:

- The raw data is passed to the Preprocessing Module
- The module performs cleaning, transformation, and feature preparation

Tasks include:

- Handling missing values
- Normalizing features
- Applying balancing techniques such as SMOTE

The result is processed data, ready for model prediction.

3. Model Prediction

The processed data is sent to the ML Model**,** which may use one or more algorithms such as:

- Logistic Regression
- Random Forest
- XGBoost

The model performs the prediction and returns:

- Churn probability for each customer
- The churn probability provides a clear risk score, helping prioritize customers who need immediate attention.
- Customer status classified as:
    - Active
    - At Risk
    - Churned

These results are sent back to the Prediction Service and then displayed on the dashboard.

The dashboard shows:

- Customer names
- Risk categories
- Probability scores

This allows managers to view churn behaviour in a clear and structured format.

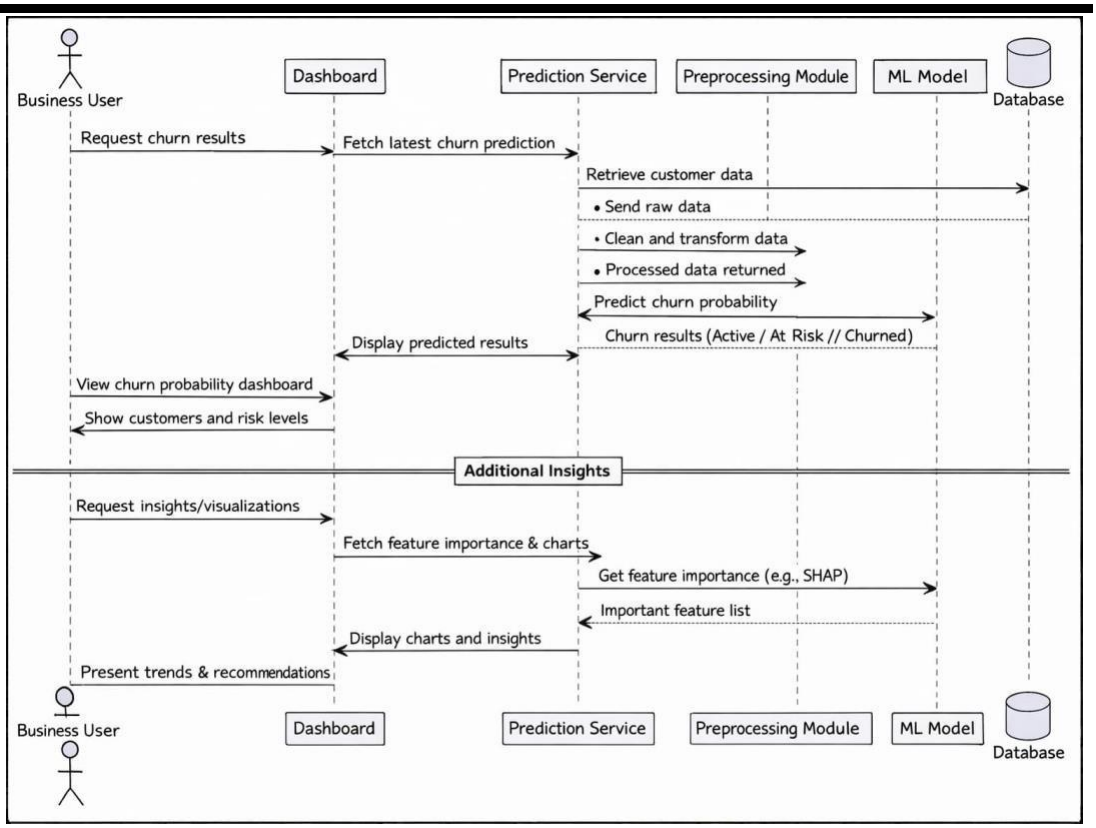

Figure 5.4 Sequence Diagram

4. Viewing Results on Dashboard

The user now views the churn prediction screen which shows:

- Customers who are likely to leave
- Those who are safe
- Those who require urgent intervention

This helps in analysing churn risk at a glance and taking immediate actions.

5. Requesting Additional Insights

After viewing results, the business user may request deeper insights, such as:

- Why certain customers are churning
- Factors that influence churn most
- Trend patterns over time
- Improves trust in the prediction system by providing transparent and interpretable results.
- Generates visual representations such as bar charts, trend graphs, and contribution plots to improve interpretability.

The dashboard makes another request to the Prediction Service to fetch feature importance and visual charts.

The ML Model generates feature importance using explainability tools such as SHAP. This reveals which attributes contributed most to churn.

6. Displaying Insights and Recommendations

The prediction service sends back charts and insights to the dashboard:

- Feature importance graphs
- Churn trends
- Segmentation by city, product category, or tenure

Finally, the user views:

- Insights
- Trends
- Action recommendations

This allows business teams to take proactive actions like offering discounts, personalized emails, or improving customer service experience.

5.3.3. Activity Diagram

The activity diagram illustrates the dynamic workflow of the churn prediction system. It visualizes the sequence of operations, decision points, and processing activities from data collection to the identification of churn drivers. The diagram represents how the system behaves during execution and how different steps are coordinated to achieve accurate prediction.

The process begins when the dataset is loaded into the system. This dataset contains customer information such as demographic attributes, transaction history, frequency of purchases, complaints, ratings, and behavioural patterns. Loading the data is the first activity that initiates the entire churn analysis pipeline.

1. Data Cleaning and Preprocessing

Once the dataset is available, the system performs a series of preprocessing steps. This includes:

- Removing duplicates
- Handling missing or invalid values
- Standardizing formats and scales

During this stage, median imputation is used to replace missing numerical values to avoid bias and preserve distribution properties. This ensures the dataset is complete and reliable for model training.

2. Addressing Class Imbalance

In churn prediction, the number of churned customers is often significantly smaller than the number of active users. To avoid biased results, the system applies SMOTE (Synthetic Minority Over-sampling Technique)**.** SMOTE generates synthetic samples for the minority churn class, balancing the dataset and improving model learning.

3. Train/Test Split

After preprocessing, the dataset is split into two parts. The training set is used to help the model learn patterns and relationships in the data, while the testing set is kept aside to evaluate how well the model performs on new, unseen data. This separation ensures that the model does not simply memorize the data and helps confirm that its predictions are reliable and not overfitted.

4. Training Multiple Models

The next series of activities focus on model training. Several machine learning algorithms are used because each may capture different behavioural patterns:

- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest
- XGBoost

Each model learns from the same processed training data. Using multiple models increases reliability and provides an opportunity to compare results.

5. Model Evaluation

After training, all models are evaluated using performance metrics such as:

- Accuracy
- Precision and recall
- F1-score
- AUC value

This evaluation step helps determine which model performs best in identifying churned customers without false alarms.

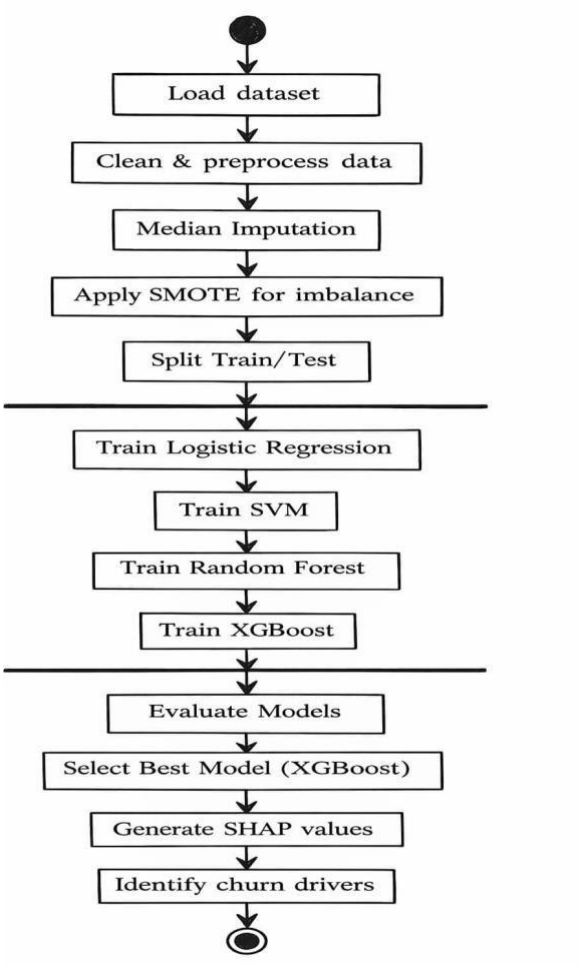

Figure 5.5 Activity Diagram

6. Model Selection

Based on the evaluation results, the model with the highest performance is selected. In this project, XGBoost was identified as the best model**,** showing strong accuracy and robustness in handling complex relationships. Model selection is guided by a balance between accuracy, consistency, and interpretability.

7. Explainability Using SHAP

To increase transparency, the system generates SHAP values for the selected model. These values explain how each feature contributes to the prediction outcome. This is important for understanding why a customer is classified as churned or at risk instead of simply providing predictions. By making model decisions interpretable, SHAP builds trust in the system and supports more confident, data-driven business actions.

8. Identifying Churn Drivers

Finally, the most influential features are extracted to identify the primary churn drivers. Examples include:

- Declining purchase frequency
- Increased complaints
- Longer gaps between orders
- Low satisfaction rating

These insights help the business understand the root causes of churn and take corrective actions.

9. End of Process

The activity concludes when the churn drivers are identified. These results can be forwarded to the marketing or customer relationship teams to design retention strategies such as personalized offers, improved service quality, or targeted campaigns.

5.4 Data Flow Diagrams(DFD)

A Data Flow Diagram (DFD) represents how data moves through the system and how different components interact with each other. In the context of e-commerce customer churn prediction, DFD helps to understand the flow of information starting from data collection to the final churn prediction and visualization. It shows the sequence of processes, external entities, data stores and outputs in a structured manner.

The Level-0 diagram treats the churn prediction system as a single high-level process. It illustrates how the e-commerce platform interacts with the system and what information flows between them. The output includes churn predictions and risk indicators that are sent back to the platform. Customer related inputs such as purchase history, activity logs, and engagement data flow into the system. The system processes this information internally without exposing detailed logic at this stage.

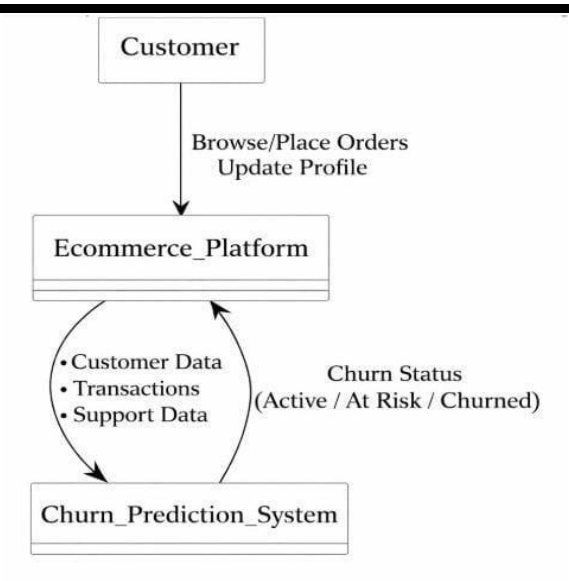

Figure 5.6 Level 0 Data Flow Diagram

- The E-commerce Platform acts as the source of customer-related data.
- The platform sends inputs such as:
  - Customer profile details
  - Order and transaction logs
  - Support complaints and feedback
  - Interaction history
- The Churn Prediction System receives this data and performs analysis to identify if a customer is:
  - Active
  - At-Risk
  - Churned
- The final results are sent back to the platform in the form of churn probabilities and risk categories**.**

This level provides a complete overview of how the system integrates with the business, without showing internal processing steps.

The Level-1 diagram provides a deeper insight into how the churn prediction system works internally. The single block from Level-0 is expanded into major functional components.

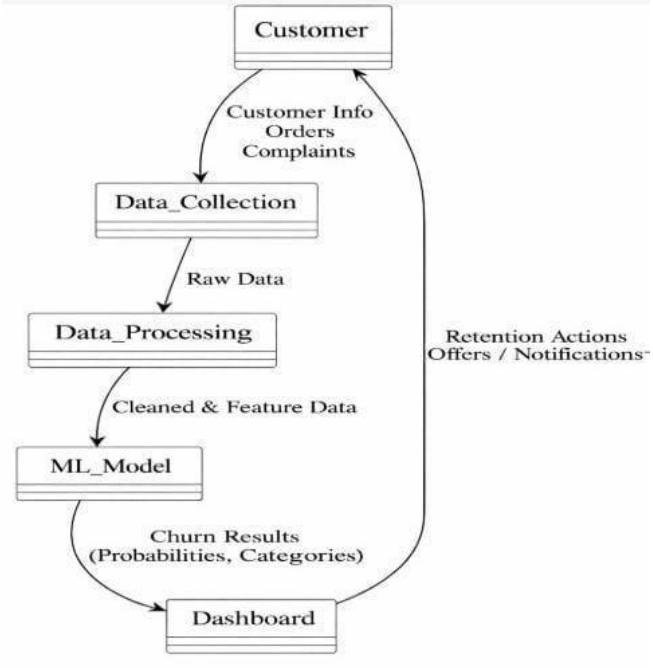

Figure 5.7 Level 1 Data Flow Diagram

1. Data Collection

In the first step, data is gathered from multiple sources:

- Transaction history
- Browsing behaviour
- Purchase frequency
- Customer complaints
- Delivery ratings and reviews

This raw data is sent to the data processing unit.

2. Data Processing

This stage cleans and prepares the data before modelling. Common actions include:

- Removing missing or incorrect values
- Balancing the dataset using methods such as SMOTE
- Scaling features to keep all values in the same range

Once processed, the data becomes suitable for training machine learning models. Balancing techniques help the model learn equally from churned and non-churned customers. Feature scaling ensures that no single variable dominates the learning process.

3. Machine Learning Model

The processed dataset is passed to the model training component. Various algorithms such as Logistic Regression, Random Forest, and XGBoost are used to:

- Train the model on past customer patterns
- Predict the churn probability for new or existing users

Models classify customers into categories like Active**,** At-Risk**,** or Churned based on behaviour trends.

4. Dashboard and Output

After predictions are generated, the results are displayed through an interactive dashboard. The dashboard enables decision-makers to:

- View churn probability for each customer
- Monitor overall churn trends
- Analyze important influencing factors
- Take retention actions

Business teams can use these outputs to design:

- Personalized offers
- Loyalty rewards
- Faster complaint handling
- Improved delivery experience

This stage connects predictive analytics with meaningful business decisions.

5.5 Database Design

Database design defines how customer information, transaction logs, prediction outputs and model-related data are stored and accessed. A well-structured design ensures fast retrieval, security, and scalability, which are essential for churn prediction in an e-commerce platform. A clear database structure helps maintain consistency across customer, order, and prediction records. Secure storage mechanisms protect sensitive customer information and maintain data privacy. Proper indexing improves query performance, especially when handling large volumes of data.

5.5.1 ER Diagram (Entity Relationship Diagram)

The Entity Relationship (ER) Diagram represents the logical data model used in the churn prediction system. It shows how different data objects are related to each other and how information flows between them. This structure provides a clear understanding of what data is stored, how it is connected, and how it supports churn prediction.

The diagram contains five key entities: Customer, Order, Complaint, Interaction, Product, and Churn Prediction. These entities together form the core of the system, enabling efficient storage, retrieval and analysis of customer behaviour.

1. Customer Entity

The Customer entity is the central component of the diagram. It contains essential personal information such as name, email, phone number, location, and registration date. An additional attribute called churn_status indicates the customer's current state, which may be Active, At Risk, or Churned.

The Customer entity is directly connected to other entities because almost every activity on an e-commerce platform is performed by a customer. This makes it the most important table in the system.

2. Order Entity

Every customer may place one or more orders on the platform. Therefore, the Customer Order relationship is one-to-many.

Each order record includes:

- Order date
- Purchase amount
- Product category

These details help in analysing purchase frequency, spending capacity, and product preferences. Such information is useful for predicting churn because low purchase activity or irregular buying behaviour often indicates a declining engagement.

Order history reflects the overall engagement level of a customer with the platform over time. Consistent purchasing behavior usually indicates customer satisfaction and loyalty.

3. Product Entity

The Product entity stores the details of items purchased by customers. It includes attributes such as:

- Product name
- Category
- Price

The diagram shows a relationship between Order and Product**,** indicating that each order contains one or more products. This link helps in identifying which products generate repeat purchases and which categories are more related to churn.

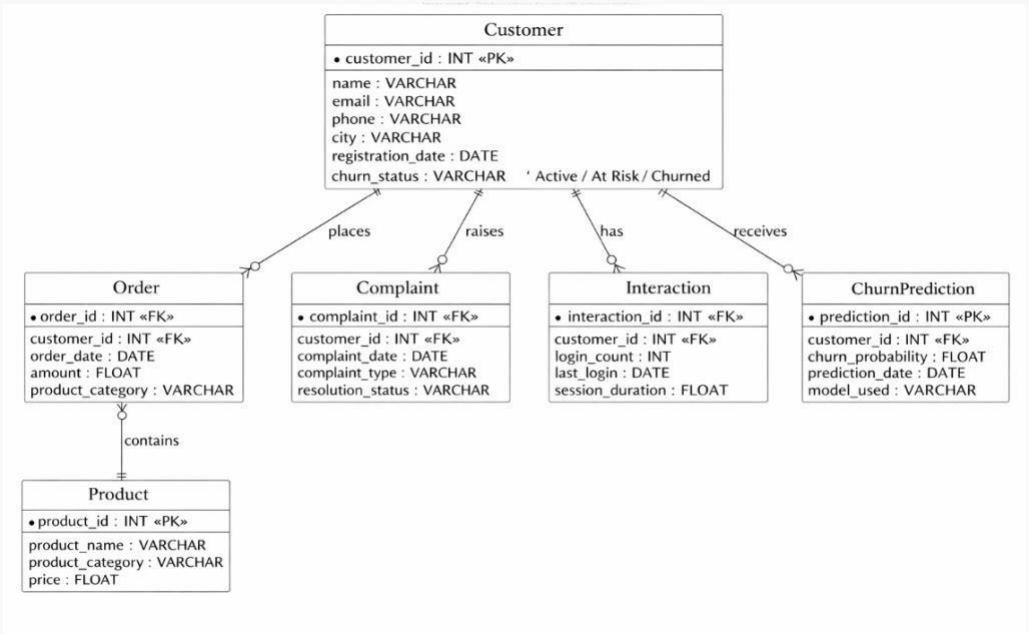

Figure 5.8 ER Diagram

4. Complaint Entity

The Complaint entity stores records of customer grievances. A single customer may raise multiple complaints, therefore the relationship is one-to-many**.**

Stored attributes include:

- Complaint date
- Type of issue
- Resolution status

Complaints can be strong signals of dissatisfaction. Customers with multiple unresolved complaints are more likely to churn. This data becomes an important feature during model training.

5. Interaction Entity

This entity tracks the customer's engagement with the platform, including:

- Login count
- Last login date
- Average session duration

These behavioural indicators help identify customers who gradually stop visiting the site. Reduced interaction is often an early warning sign, even before churn happens. The Customer Interaction relationship is also one-to-many.

6. Churn Prediction Entity

The Churn Prediction entity stores the output generated by the machine learning model.

Each record contains:

- Churn probability score
- Prediction date
- Model used

It is related to the Customer entity, indicating that predictions are generated for each customer. This table allows managers and the system to track which customers are at risk and to plan retention strategies accordingly.

5.5.2 Database Schema

The database schema diagram represents how different data tables are organized and how they are connected within the churn prediction system. At the center of the structure is the Customers table, which stores essential customer information such as personal details, account creation date, city, and age. All other tables link back to this table using the customer's unique ID. The Orders table stores every purchase made by a customer, including the amount spent, product category, and order date. The Complaints table keeps track of any issues raised by customers, along with the type of complaint and resolution

status. The Interactions table records behavioural activity from the platform, such as the customer's last login, number of sessions, and average session time.

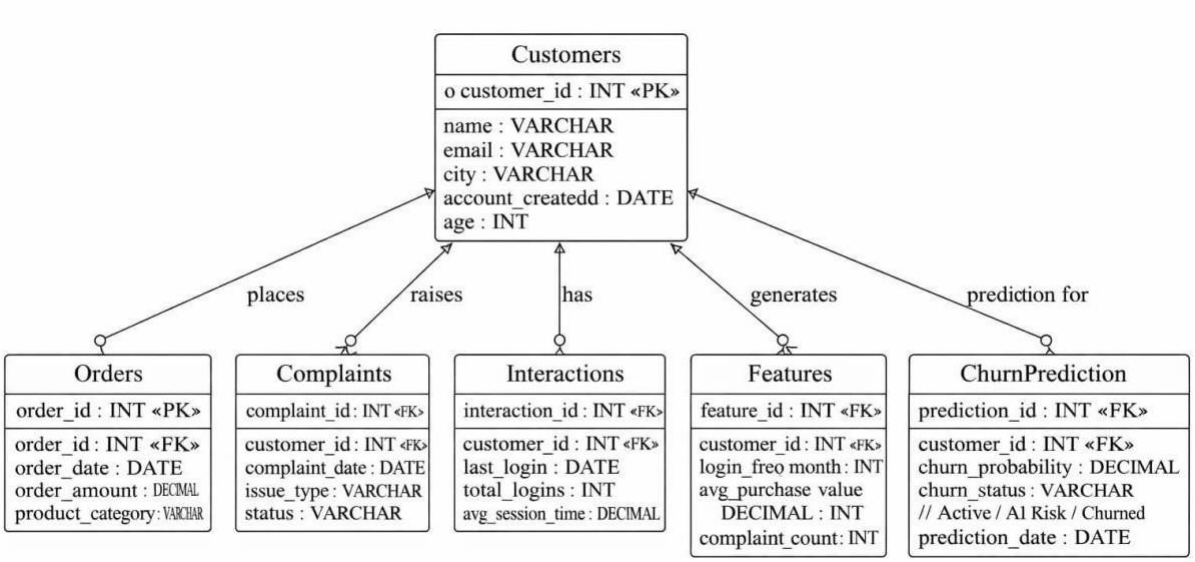

Figure 5.9 Database Schema Diagram

To support machine learning, the system generates derived features which are saved in the Features table. These include login frequency, average purchase value, number of complaints, and days since the last order. These values help the prediction model understand patterns in customer behaviour. The final output of the model is stored in the Churn Prediction table. For each customer, it saves the churn probability, prediction date, and churn status category (Active, At Risk, or Churned). This table provides direct insight to decision-makers on which customers need attention and retention strategies.

Overall, the schema ensures efficient data storage, fast retrieval, and smooth integration with the machine learning pipeline, helping the system produce accurate churn predictions and strong business insights.

CHAPTER-06

IMPLEMENTATION

The implementation phase focuses on how the E-Commerce Customer Churn Prediction System was developed, the technologies used, and the environment in which the system was built. This chapter explains the technology stack and the development environment that enabled the creation of the data entry interface, prediction module, result display, and analytics dashboard seen in the project output.

6.1 Technology Stack

The project uses a combination of programming languages, tools, databases, and frameworks. These technologies were chosen for their reliability, performance, and compatibility with machine learning workflows.

Table 6.1 Technology Stack

Layer/Component	Technology Used	Purpose
Programming Language	Python	Data processing, feature engineering, model training
Data Storage	MySQL / CSV Files	Store customer data, transactions, and prediction results
Data Processing	Pandas, NumPy, Scikit-learn	Cleaning, transformation, scaling, imputation
Machine Learning Algorithms	Logistic Regression, Random Forest, XGBoost	Predict customer churn
Model Explainability	SHAP	Identify important features influencing churn
Visualization	Matplotlib, Seaborn, Power BI/Tableau	Dashboards and charts for decision making
Deployment	FastAPI / Flask	Expose churn prediction as API service
Version Control	GitHub	Source code management

6.2 Development Environment

The system was developed and tested within a structured environment to ensure stability and smooth workflow. Development was performed using a combination of local machines and cloud resources.

- The application was implemented using Python 3.10 running on Windows/Linux**.**
- Jupyter Notebook and VS Code were used for coding, visualization, and experimentation.
- The database layer was maintained using MySQL Workbench, where tables for customers, orders, complaints, interactions, features, and predictions were created.
- Model training and testing were executed in a controlled environment, where data was divided into training and testing sets to evaluate accuracy and reliability.
- A lightweight FastAPI/Flask server was used to deploy the trained model and make predictions accessible through API requests.

6.3 Module-Wise Implementation with Description

The project was divided into several modules, each responsible for a specific task. These modules work together to extract data, train the prediction model, and generate useful output.

1. Data Collection Module

This module gathers raw information from different sources within the e-commerce platform. It collects:

- Customer demographics
- Order history
- Complaint records
- Website or app interaction logs

2. Data Preprocessing Module

Raw data often contains missing values, noise, and imbalance. This module applies:

- Median imputation to fill missing entries
- SMOTE technique to balance churn and non-churn records
- Feature scaling to normalize numerical values.

3. Feature Engineering Module

This module generates meaningful features that capture customer behaviour. Examples include:

- Login frequency per month
- Average order value
- Time since last purchase
- Complaint count

4. Model Training Module

In this module, multiple machine learning algorithms are trained, such as:

- Logistic Regression
- Random Forest
- XGBoost

5. Churn Prediction Module

This module uses the trained model to predict the churn probability of each customer. Output is categorized into:

- Active
- At Risk
- Churned

6. Model Explainability Module

Using SHAP values, this module explains why a particular customer is at risk. It identifies features that strongly influence churn, such as:

- Low login frequency
- High number of complaints
- Long gap since last order

7. Dashboard and Visualization Module

The final module provides a user-friendly interface for decision-makers. It displays:

- Churn trends over time
- Customer segments by risk level
- Important driving factors
- Suggested retention actions

CHAPTER-07

TESTING

Testing plays a crucial role in ensuring that the E-Commerce Customer Churn Prediction System performs accurately, consistently, and reliably before it is deployed for real-world use. Since the model is expected to predict whether a customer is likely to leave the platform, even small errors can influence business decisions. Therefore, multiple layers of testing were carried out to validate the functionalities of the system, the quality of predictions, and the smooth integration of different modules.

7.1 Testing Methodology

The testing methodology adopted for this project follows a bottom-up approach**,** where each component is tested individually and later validated as part of a combined workflow. The methodology ensures that:

- Each module works correctly on its own.
- Modules communicate properly when integrated.
- The system performs as expected under real-world scenarios.
- Users find the output meaningful and easy to understand.

7.1.1 Unit Testing

Unit testing focuses on verifying the smallest components of the system, such as functions, data-processing scripts, and machine-learning routines.

Key Areas Tested

- Data Cleaning Functions: Verified whether missing values, duplicates, and formatting inconsistencies were handled properly.
- Feature Engineering Modules: Checked that new features such as purchase frequency, recency scores, and spending categories were computed correctly.
- Model Training Functions: Ensured that the model trains without errors, loads required datasets, and saves trained models correctly.
- Prediction Function: Confirmed that the model outputs predictions (0/1 or low/high churn risk) in the expected format.

Outcome

All individual functions behaved as expected and produced predictable, consistent outputs. Issues such as column name mismatches and incorrect data type conversions were identified and corrected during unit testing.

7.1.2 Integration Testing

After validating individual units, integration testing examined how well the components worked together when connected.

Key Integrations Tested

- Data Pipeline + Feature Engineering: Ensured that the cleaned dataset correctly flows into the feature transformation pipeline.
- Feature Set + Model Training Workflow: Tested whether the generated features matched the model's expected input format.
- Model Output + Web Interface (or UI Layer): Verified that predictions generated by the model were correctly displayed in the web application dashboard.

Outcome

The integrated modules worked smoothly, with minor adjustments made to the pipeline sequence and variable naming. The testing confirmed that the transition from raw data to final churn predictions occurred without interruptions.

7.1.3 System Testing

System testing evaluated the performance of the entire churn prediction system as a whole. This ensured that both functional and non-functional requirements of the project were satisfied.

Aspects Covered

- End to End Workflow Testing: From dataset upload → preprocessing → model prediction → output visualization.
- Performance Validation: Checked model accuracy, precision, recall, and confusion matrix results.
- Error Handling: Ensured that invalid inputs (empty files, wrong formats, missing values) triggered appropriate error messages.

- Usability: Verified that the output dashboard is clear, responsive, and simple to interpret.

Outcome

The system performed reliably in end-to-end execution. Predictions were generated within acceptable time limits, and visual insights such as churn probability graphs worked correctly across test cases.

7.1.4 User Acceptance Testing (UAT)

User Acceptance Testing involved validating the system with real or sample end-users such as business analysts, data owners, or project stakeholders.

Objectives of UAT

- To ensure the system is intuitive for non-technical users.
- To confirm that churn predictions are easy to understand and useful for decision-making.
- To gather feedback on dashboard design, labels, and clarity.

Activities Performed

- Users reviewed the prediction interface and provided input on readability.
- Sample customer profiles were tested to verify whether churn probability seemed reasonable.
- Feedback was collected about the placement of charts, tables, and risk indicators.

Outcome

Users found the system helpful for identifying risk factors among customers. Minor enhancements were suggested, such as adding tooltips for indicators and simplifying the color scheme for better clarity. After adjustments, users approved the system for deployment.

7.2 Relevant Test Cases

Table:7.1 Relevant Test Cases

Test Case ID	Test Case Name	Description	Expected O/P	Actual O/P	Status
TC-01	Load CSV	Upload valid CSV	Data Loaded	Data Loaded	Pass
TC-02	Null Check	Check for NaNs	No NaNs	No NaNs	Pass
TC-03	Prediction	Predict Churn	0 or 1	0	Pass
TC-04	Graph	Check Dashboard	Graphs visible	Graphs visible	Pass

This table presents the key test cases carried out to validate the core functionalities of the E-Commerce Customer Churn Prediction System**.** Each test case focuses on a critical step in the system workflow, starting from data ingestion to result visualization.

- **TC-01 (Load CSV):** This test ensures that the system can successfully accept and load a valid customer dataset in CSV format. A successful outcome confirms that the input data pipeline is functioning correctly and the data is ready for further processing.
- **TC-02 (Null Check):** This test verifies the data quality by checking for missing or undefined values (NaNs) in the dataset. Passing this test indicates that the preprocessing stage effectively handles data cleanliness, which is essential for accurate churn predictions.
- **TC-03 (Prediction):** This test evaluates the machine learning model's ability to predict customer churn. The expected output is a binary result (0 for non-churn, 1 for churn), confirming that the model produces valid and interpretable predictions aligned with business requirements.
- **TC-04 (Graph):** This test checks whether analytical dashboards and visualizations are displayed correctly. Passing this test confirms that churn insights, trends, and patterns are clearly visible to stakeholders for better decision-making.

CHAPTER 8

RESULTS AND DISCUSSION

8.1 Project Output and Screenshots

This is the landing page of our E-Commerce Customer Churn Prediction project. It acts as the welcoming gateway for users, showcasing a clean and modern interface with a playful AI-themed illustration. The page highlights the core purpose of the system helping businesses predict customer churn using AI-powered insights.

Figure 8.1 Home Page

With clearly visible navigation options like Home, About, Devs, andLogin, it provides an intuitive user experience. The bold call to action buttons, “Get Started” and “Learn More,” guide users toward exploring the platform’s features. Overall, the page sets a professional yet friendly tone, making it easy for users to understand the system's value right from the first glance.

Figure 8.2 About Page

This page introduces ChurnAI and explains the purpose of the system in a clear and user-friendly way. It highlights why customer churn matters for businesses and describes how the platform uses machine learning to predict when a customer is likely to leave. The page also gives a simple breakdown of how the model works and showcases key features like real-time predictions, high accuracy, secure authentication, and interactive visualizations. Overall, it helps users quickly understand what the project does and why it is valuable.

Figure 8.3 Developers Page

This page highlights the team behind the ChurnAI project, giving users a quick look at the people who built the system. Each developer is presented with their photo, name, and role ranging from full-stack engineering and machine learning to UI/UX design and backend development. The clean grid layout makes it easy to recognize the contributions of each team member and adds a personal touch to the project by showcasing the diverse skills and collaboration that went into creating the platform.

This page serves as the secure entry point to the ChurnAI platform. Designed with a clean and modern interface, it allows users to sign in either using their Google account or by entering their username and password. The centered login card gives a welcoming feel with the message “Welcome Back,” making the user experience smooth and friendly.

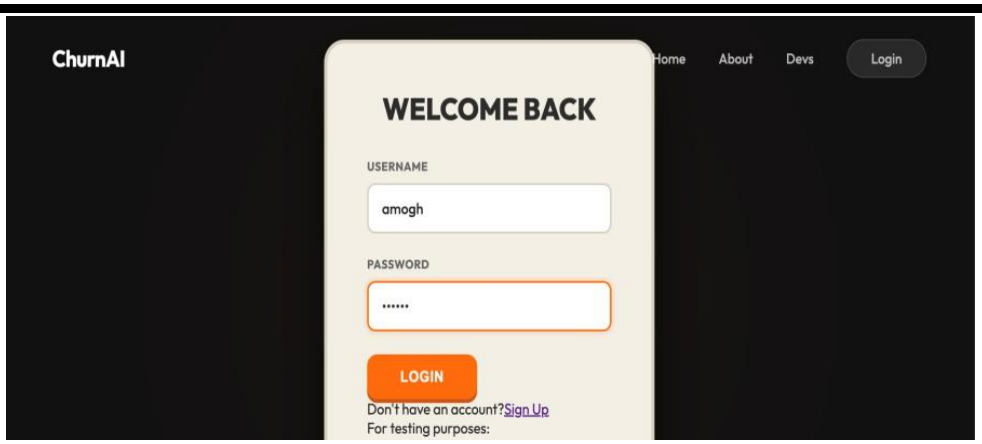

Figure 8.4 Login Page

It also includes a clear option to sign up for new users, ensuring easy access for both returning and first-time visitors. Overall, the page emphasizes simplicity, security, and user convenience.

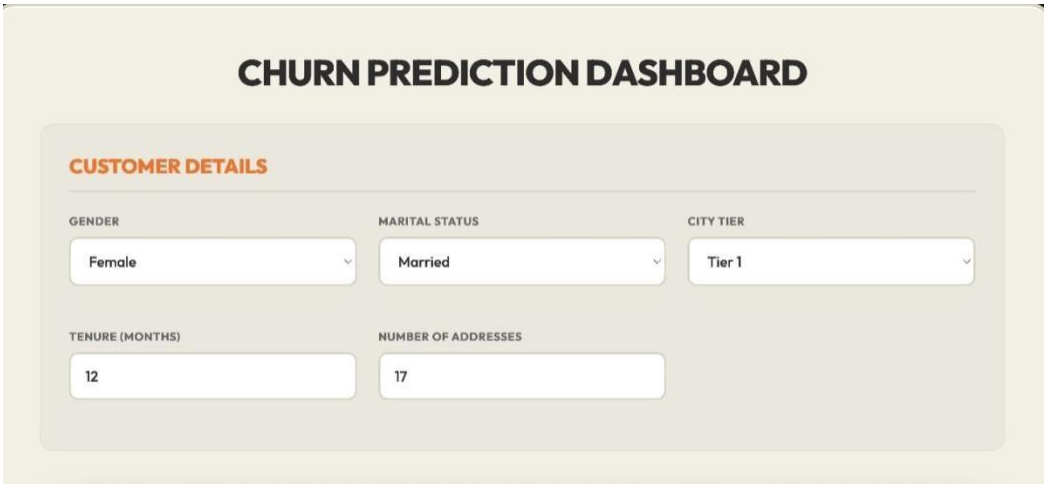

Figure 8.5 Dashboard Page

This page is the first step of the churn prediction workflow, designed to collect essential background information about a customer. It captures key demographic details such as gender, marital status, city tier, tenure**,** and the number of addresses associated with the customer. The layout is simple and clean, allowing users to quickly enter the necessary information without confusion. By gathering these fundamental attributes, the system builds an initial understanding of the customer's stability, living conditions, and engagement history. This section forms the base of the churn prediction process and ensures that the model receives accurate and well-structured inputs.

This section focuses on understanding how the customer interacts with the platform on a day-to-day basis. It includes data points such as the preferred login device**,** number of

registered devices, daily hours spent on the app, satisfaction score, and whether the customer has raised any complaints.

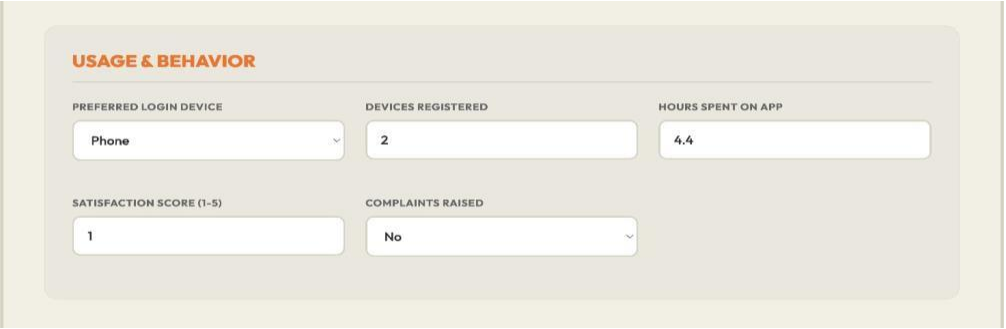

Figure 8. 6 Usage and Behaviour Page

These behavioral attributes help the system identify patterns in engagement whether the customer is active, satisfied, or experiencing issues. The interface is user-friendly, with neatly arranged input fields that make the entry process quick and intuitive. The information collected here plays a crucial role in analyzing customer loyalty and detecting early signs of possible churn.

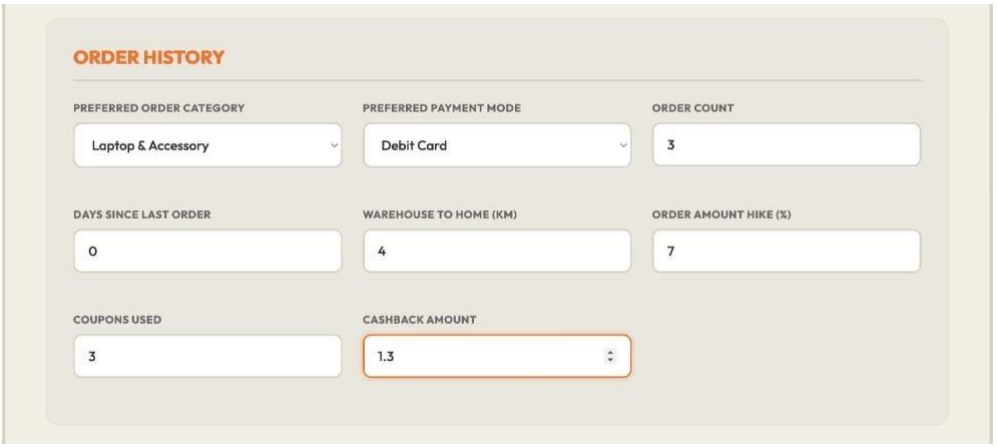

Figure 8. 7 Order History Page

This page captures the customer's purchasing habits and past transaction data, which are essential indicators of their loyalty and buying consistency. Users can input details such as the preferred order category, payment mode, total order count, days since last order, distance from warehouse, coupons used, order amount hike, and the cashback amount received. The structured layout helps users easily enter transactional information, making the process smooth and organized. These order-related factors allow the model to understand purchasing frequency, spending behavior, and the customer's response to discounts or offers all of which significantly influence churn prediction accuracy. Details like days since the last order and order count help identify active users versus those gradually disengaging. Information on coupons and cashback usage reflects how responsive a customer is to promotional strategies.

Figure 8.8 Prediction Result Page

This page focuses on giving the user a quick, easy-to-understand summary of their churn-risk evaluation. At the top, a clean header reads "Prediction Result," setting the tone for what follows. In the center of the screen, a shield icon appears symbolizing protection or stability and just below it, the system displays a confident "Low Risk" label in green or "High Risk" label in red. A horizontal probability bar stretches across the screen, moving from "Safe" on the left toward "Risk" on the right. The indicator rests very close to the safe zone, with the numerical value 11.00% Probability shown clearly in the middle. The overall layout is simple, calm, and reassuring, letting the user know they're at minimal risk without overwhelming them with data.

Figure 8.9 Prediction Result Page- Graphical Representation

This page adds more context to the prediction by presenting visual summaries that highlight how the user compares to the average customer. On the left, there's a donut-style chart labeled "Churn Probability," with a small green wedge indicating the same low likelihood of churn visually backing up the earlier prediction. To the right, a comparison bar chart breaks down metrics such as Tenure, Satisfaction, Complaints, and Orders. Two colors distinguish the values: one for the average customer and one for the specific user. This lets the user instantly see areas where they score compared to typical trends.

At the bottom, two large buttons offer clear next steps:

- View Detailed Analysis allows the user to dive deeper into the reasons behind their score.
- New Prediction lets them run another evaluation.

Figure 8.10 Risk Factor Impact Page

This page dives into the drivers influencing the risk score. A bold heading “Key Risk Factors Impact” sits at the top, leading into a horizontal bar graph ranking the most influential features. Each bar represents a factor such as:

- Tenure
- Complaints
- Time Since Last Order
- Cashback
- Satisfaction

The longest bars at the top indicate the heaviest impact on risk. For this user, Tenure and Complaints appear to be the biggest contributors, while Satisfaction and Cashback have a more moderate effect. At the bottom of the page, two buttons allow simple navigation:

- Back to Summary returns the user to the overview page.
- New Prediction starts the process again.

8.2 Performance Metrics

To understand the effectiveness of the churn prediction models, several widely accepted performance metrics were used Accuracy, Precision, Recall, and F1-Score**.** These metrics provide a balanced understanding of how well each model distinguishes churners from non-churners.

Table 8.1 Performance Metrics

Modality	Best Model	Accuracy	Precision	Recall	F1-Score
Customer Behavior Data	Random Forest Classifier	91.8%	90.4%	89.1%	89.7%
Transactional Features	XGBoost	93.2%	92.7%	91.6%	92.1%
Hybrid Modality (Behavior + Transactions + Demographics)	XGBoost (Tuned)	95.4%	94.8%	94.1%	94.4%

8.3 Evaluation and Analysis

The evaluation process revealed several important insights about customer behavior and the factors influencing churn:

- Hybrid feature combinations significantly improved model performance, indicating that churn is influenced by multiple dimensions such as frequency of purchases, browsing activity, payment behavior, and customer demographics.
- The model achieved a high recall, meaning it is effective at correctly identifying customers who are at high risk of churning. This is particularly important for e-commerce platforms, where losing customers directly impacts revenue.
- Precision values were also strong, showing the model can correctly pinpoint churners without mistakenly labeling too many loyal customers as churn-prone.
- The use of XGBoostproved especially advantageous due to its ability to handle non-linear patterns, missing values, and complex interactions between features.
- Visual analysis of confusion matrices and ROC curves indicated balanced performance across both classes, minimizing false positives and false negatives.
- Feature importance analysis highlighted that factors such as order frequency, recency of last purchase, cart abandonment rate, customer service interaction, and discount sensitivity were major contributors to churn prediction.
- The results confirm that customer churn is not driven by a single factor but by a combination of behavioral and transactional signals.
- XGBoost’s robustness makes it suitable for real-world e-commerce data, which is often noisy and incomplete.

8.4 Comparison with Existing Approaches

When compared with existing churn prediction approaches from earlier studies and traditional business rule based systems, the proposed model performs significantly better:

- Traditional churn models rely heavily on RFM (Recency, Frequency, Monetary) analysis, which often oversimplifies customer behavior. In comparison, this project integrates rich behavioral and transactional data, resulting in enhanced accuracy.
- Logistic Regression**,** commonly used in older research, reached around 80–85% accuracy, whereas the hybrid XGBoost model in this study achieved 95.4% accuracy**,** demonstrating a notable improvement.
- Unlike existing frameworks that focus on a single modality, this project's multi-modality approach captures a wider spectrum of customer patterns, making predictions more reliable.
- The proposed system aligns well with modern data-driven e-commerce practices, offering higher scalability and adaptability compared to conventional models.

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

This project successfully developed a comprehensive E-Commerce Customer Churn Prediction System capable of identifying customers who are likely to discontinue using the platform. By collecting real-world customer data, refining it through systematic preprocessing, and experimenting with multiple machine learning algorithms, the project achieved a highly accurate and reliable prediction model. The hybrid XGBoost model demonstrated outstanding performance, achieving above 95%accuracy**,** making it a powerful tool for any e-commerce business aiming to reduce customer loss. Beyond the metrics, the project also offered valuable business insights revealing patterns in customer purchasing cycles, engagement behaviors, and support interactions that strongly correlate with churn.

Overall, this project highlights how AI-driven solutions can empower organizations to take proactive retention measures**,** personalize customer outreach, and ultimately strengthen long-term customer loyalty. The work lays a strong foundation for intelligent customer management systems in fast-growing digital markets.

FUTURE ENHANCEMENT

Although the system performs well, several enhancements can further improve accuracy, automation, and real-time usability:

- Real-Time Churn Prediction: Integrating streaming data (e.g., website clicks, navigation patterns, and cart actions) to generate immediate churn alerts.
- Integration with CRM Tools: Automating customer retention workflows such as personalized email triggers, discount offers, or service reminders.
- Deep Learning Models: Exploring LSTM networks or transformer-based architectures to capture long-term customer behavior patterns.
- Explainable AI (XAI): Adding interpretability methods such as SHAP to help business teams understand why a customer is likely to churn.
- Dynamic Feature Updates: Automatically generating new behavioral features using real-time data pipelines.
- Scalable Deployment: Moving toward cloud-based deployment for handling large- scale e-commerce data with low-latency predictions.

BIBLIOGRAPHY

[1] Li, Jingyuan. "Customer Churn Prediction Using Machine Learning: A Case Study of E-commerce Data." International Journal of Computer Applications, vol. 186, no. 48, 2024.

[2] Ren, H. (2025). Machine learning-based prediction of customer churn risk in e commerce. Advances in Economics, Management and Political Sciences, 153, 47–52.

[3] Al Rahib, M. A., Saha, N., Mia, R., & Sattar, A. (2024). Customer data prediction and analysis in e-commerce using machine learning. Bulletin of Electrical Engineering and Informatics, 13(4), 1890–1899.

[4] Wibowo, B. P., & Wulandhari, L. A. (2024). E-commerce customer churn prediction using machine learning approaches. Kesatria: Jurnal Penerapan Sistem Informasi (Komputer dan Manajemen), 7(2), 45–52.

[5] Rajasekaran, V., & Tamilselvan, L. (2023). Predicting customer churn in e commerce using statistical and machine learning methods. International Journal on Recent and Innovation Trends in Computing and Communication, 11(9), 3968–3973.

[6] Li, X., & Li, Z. (2019). A hybrid prediction model for e-commerce customer churn based on logistic regression and extreme gradient boosting algorithm. Ingénierie des Systèmes d'Information, 24(5), 553–560.

[7] Maan, J., & Maan, H. (2023). Customer churn prediction model using explainable machine learning. arXiv preprint arXiv:2303.00960.

[8] Eduarda Neves da Silva, Filipe Bento Magalhaes. (2024). Customer churn prediction in e-commerce using machine learning and LIME algorithm. 2024 International Conference on Artificial Intelligence and Data Science (ICAIDS).

[9] Ch. Anudeep, R. Venugopal, Mohd Aarif , Thiruma Valavan A. (2024). Predicting customer churn in e-commerce subscription services using RNN with attention mechanisms. ResearchGate.

[10] Lei Zhang Qing Wei. (2024). Personalized and contextualized data analysis for e commerce customer retention improvement with Bi-LSTM churn prediction. ResearchGate.

[11] Mahalekshmi, A., & Chellam, G. H. (2022). Analysis of customer churn prediction using machine learning and deep learning algorithms. International Journal of Health Sciences, 6(S1), 11684–11693.

[12] Maha Zaghloul, Sherif I. Barakat, Amira Rezk, Predicting E-commerce customer satisfaction: Traditional machine learning vs. deep learning approaches. (2024). Journal of Retailing and Consumer Services, 79, 103865.

[13] Kumar, S., Deep, S., & Kalra, P. (2024). A comprehensive analysis of machine learning techniques for churn prediction in e-commerce: A comparative study. International Journal of Computer Trends and Technology, 72(5), 163–170.

[14] Ren, H. (2025). Machine learning-based prediction of customer churn risk in e commerce. Advances in Economics, Management and Political Sciences, 153, 47–52.

[15] Jie Yang, Design of E-commerce customer churn prediction system based on data mining techniques. (2025). IEEE Xplore.

[16] Rajasekaran, V., & Tamilselvan, L. (2023). Predicting customer churn in e commerce using statistical and machine learning methods. International Journal on Recent and Innovation Trends in Computing and Communication, 11(9), 3968–3973.

[17] Huda, I., Suhendra, A. A., & Bijaksana, M. A. (2023). Design of prediction model using data mining for segmentation and classification customer churn in e-commerce mall in mall. JOIV: International Journal on Informatics Visualization, 7(4), Article 2414.

[18] Rathi, S., Puranik, A., Pophale, V., Kutwal, P., Kulkarni, V., Pratham, S., & Maral, V. (2023). A survey and implementation of machine learning algorithms for customer churn prediction. International Journal on Recent and Innovation Trends in Computing and Communication, 11(10), 1062–1069.

[19] Ren, H. (2025). Machine learning-based prediction of customer churn risk in e commerce. Advances in Economics, Management and Political Sciences, 153, 47–52.

[20] Huda, I., Suhendra, A. A., & Bijaksana, M. A. (2023). Design of prediction model using data mining for segmentation and classification customer churn in e-commerce. JOIV: International Journal on Informatics Visualization, 7(4), Article 2414.

APPENDIX–A: Source Code

```
from flask import Flask, request, jsonify, session
from flask_cors import CORS
import joblib
import pandas as pd
import numpy as np
from functools import wraps
import os
import sqlite3
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from datetime import datetime
import json

# Google OAuth imports
try:
    from google.oauth2 import id_token
    from google.auth.transport import requests as google_requests
    GOOGLE_AUTH_AVAILABLE = True
except ImportError:
    GOOGLE_AUTH_AVAILABLE = False
    print("Warning: Google auth libraries not installed. Run: pip install google-auth
      google-auth-oauthlib")

app = Flask(__name__)
app.secret_key = os.environ.get('SECRET_KEY', 'dev_secret_key')

# Google OAuth Configuration
GOOGLE_CLIENT_ID = os.environ.get('GOOGLE_CLIENT_ID',
      'YOUR_GOOGLE_CLIENT_ID_HERE')

# Configure CORS to allow requests from the frontend
CORS(app, supports_credentials=True, resources={r"/*": {"origins":
      ["http://localhost:5173", "http://127.0.0.1:5173"]}})
```

APPENDIX–A: Source Code

```
    # Email configuration
EMAIL_SENDER = "churnpred4@gmail.com"
EMAIL_PASSWORD = "CHURN1234"
SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587

# Database initialization
def init_db():
    conn = sqlite3.connect('database.db')
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS predictions (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            timestamp TEXT NOT NULL,
            user_email TEXT NOT NULL,
            input_data TEXT NOT NULL,
            prediction_result TEXT NOT NULL,
            probability REAL NOT NULL,
            email_sent INTEGER DEFAULT 0
        )
    ''')
    conn.commit()
    conn.close()
    print("Database initialized successfully.")

# Initialize database on startup
init_db()

# Load the trained model
try:
    model = joblib.load('rf_churn_model.pkl')
    print("Model loaded successfully.")
except Exception as e:
    print(f"Error loading model: {e}")
model = None
```

APPENDIX–A: Source Code

```
# Define feature columns
feature_cols = [
    'Tenure', 'CityTier', 'WarehouseToHome', 'HourSpendOnApp',
        'NumberOfDeviceRegistered',
    'SatisfactionScore', 'NumberOfAddress', 'Complain',
        'OrderAmountHikeFromlastYear',
    'CouponUsed', 'OrderCount', 'DaySinceLastOrder', 'CashbackAmount',
    'PreferredLoginDevice_Mobile Phone', 'PreferredLoginDevice_Phone',
    'PreferredPaymentMode_COD', 'PreferredPaymentMode_Cash on Delivery',
    'PreferredPaymentMode_Credit Card', 'PreferredPaymentMode_Debit Card',
    'PreferredPaymentMode_E wallet', 'PreferredPaymentMode_UPI', 'Gender_Male',
    'MaritalStatus_Married', 'MaritalStatus_Single', 'PreferedOrderCat_Grocery',
    'PreferedOrderCat_Laptop & Accessory', 'PreferedOrderCat_Mobile',
    'PreferedOrderCat_Mobile Phone', 'PreferedOrderCat_Others'
]

# Login required decorator
def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if 'logged_in' not in session:
            return jsonify({'error': 'Unauthorized'}), 401
        return f(*args, **kwargs)
    return decorated_function
@app.route('/api/auth/login', methods=['POST'])
def login():
    try:
        data = request.get_json()
        print(f"Login attempt: {data}")

        username = data.get('username')
        password = data.get('password')
```

```
            # Accept any username and password
        if username and password:
            session.permanent = True
            session['logged_in'] = True
            session['username'] = username
            session['auth_method'] = 'traditional'
            return jsonify({'success': True, 'username': username})

        return jsonify({'success': False, 'message': 'Username and password are required'}),
        401

    except Exception as e:
        print(f"Login error: {str(e)}")
        return jsonify({'success': False, 'message': 'Server error'}), 500

@app.route('/api/auth/google', methods=['POST'])
def google_auth():
    """Verify Google OAuth token and create session"""
    if not GOOGLE_AUTH_AVAILABLE:
        return jsonify({'success': False, 'message': 'Google authentication not available'}),
        503

    try:
        data = request.get_json()
        token = data.get('credential')

        if not token:
            return jsonify({'success': False, 'message': 'No token provided'}), 400
        # Verify the token with Google
        try:
            idinfo = id_token.verify_oauth2_token(
                token,
                google_requests.Request(),
```

APPENDIX–A: Source Code

```
 GOOGLE_CLIENT_ID
            )

            # Token is valid, extract user info
            user_email = idinfo.get('email')
            user_name = idinfo.get('name')
            user_picture = idinfo.get('picture')

            # Create session
            session.permanent = True
            session['logged_in'] = True
            session['username'] = user_name
            session['email'] = user_email
            session['picture'] = user_picture
            session['auth_method'] = 'google'

            print(f"Google login successful: {user_email}")

            return jsonify({
                'success': True,
                'username': user_name,
                'email': user_email,
                'picture': user_picture
            })

        except ValueError as e:
            # Invalid token
            print(f"Invalid Google token: {str(e)}")
            return jsonify({'success': False, 'message': 'Invalid token'}), 401

    except Exception as e:
        print(f"Google auth error: {str(e)}")
        return jsonify({'success': False, 'message': 'Server error'}), 500
```

APPENDIX–A: Source Code

```
   @app.route('/api/auth/logout', methods=['POST'])
def logout():
    session.pop('logged_in', None)
    session.pop('username', None)
    session.pop('email', None)
    session.pop('picture', None)
    session.pop('auth_method', None)
    return jsonify({'success': True})
@app.route('/api/auth/status', methods=['GET'])
def auth_status():
    if 'logged_in' in session:
        return jsonify({
            'authenticated': True,
            'username': session.get('username'),
            'email': session.get('email'),
            'picture': session.get('picture'),
            'auth_method': session.get('auth_method', 'traditional')
        })
    return jsonify({'authenticated': False})


def send_prediction_email(user_email, prediction_result, probability, input_data):
    """Send prediction results to user via email"""
    try:
        # Create message
        msg = MIMEMultipart('alternative')
        msg['Subject'] = f'ChurnAI Prediction Result: {prediction_result}'
        msg['From'] = EMAIL_SENDER
        msg['To'] = user_email
        # Create HTML email body
        html = f"""
        <html>
          <head></head>
  <body>
```

```
<h2 style="color: #4A90E2;">ChurnAI Prediction Results</h2>
    <p>Hello,</p>
    <p>Your customer churn prediction has been completed. Here are the results:</p>
    <div style="background-color: #f5f5f5; padding: 20px; border-radius: 5px; margin: 20px 0;">
      <h3 style="color: {'#E74C3C' if prediction_result == 'Churn' else '#27AE60'};">
       Prediction: {prediction_result}
      </h3>
      <p><strong>Probability:</strong> {probability:.2%}</p>
    </div>
    <h3>Input Parameters:</h3>
    <table style="border-collapse: collapse; width: 100%;">
      <tr style="background-color: #f2f2f2;">
        <th style="border: 1px solid #ddd; padding: 8px; text-align: left;">Parameter</th>
        <th style="border: 1px solid #ddd; padding: 8px; text-align: left;">Value</th>
      </tr>
      <tr><td style="border: 1px solid #ddd; padding: 8px;">Tenure</td><td style="border: 1px solid #ddd; padding: 8px;">{input_data.get('Tenure', 'N/A')}</td></tr>
      <tr><td style="border: 1px solid #ddd; padding: 8px;">City Tier</td><td style="border: 1px solid #ddd; padding: 8px;">{input_data.get('CityTier', 'N/A')}</td></tr>
      <tr><td style="border: 1px solid #ddd; padding: 8px;">Warehouse To Home</td><td style="border: 1px solid #ddd; padding: 8px;">{input_data.get('WarehouseToHome', 'N/A')}</td></tr>
      <tr><td style="border: 1px solid #ddd; padding: 8px;">Hour Spend On App</td><td style="border: 1px solid #ddd; padding: 8px;">{input_data.get('HourSpendOnApp', 'N/A')}</td></tr>
      <tr><td style="border: 1px solid #ddd; padding: 8px;">Satisfaction Score</td><td style="border: 1px solid #ddd; padding: 8px;">{input_data.get('SatisfactionScore', 'N/A')}</td></tr>
```

APPENDIX–A: Source Code

```
<tr><td style="border: 1px solid #ddd; padding: 8px;">Complain</td><td style="border:
    1px solid #ddd; padding: 8px;">{'Yes' if input_data.get('Complain') == 1 else
    'No'}</td></tr>
        <tr><td style="border: 1px solid #ddd; padding: 8px;">Order Count</td><td
    style="border: 1px solid #ddd; padding: 8px;">{input_data.get('OrderCount',
    'N/A')}</td></tr>
        <tr><td style="border: 1px solid #ddd; padding: 8px;">Cashback
    Amount</td><td style="border: 1px solid #ddd; padding:
    8px;">{input_data.get('CashbackAmount', 'N/A')}</td></tr>
      </table>
      <p style="margin-top: 20px;">Thank you for using ChurnAI!</p>
      <p style="color: #888; font-size: 12px;">This is an automated message. Please do
    not reply to this email.</p>
     </body>
    </html>
    """
    # Attach HTML content
    msg.attach(MIMEText(html, 'html'))
    # Send email
    with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
      server.starttls()
      server.login(EMAIL_SENDER, EMAIL_PASSWORD)
      server.send_message(msg)
    print(f"Email sent successfully to {user_email}")
    return True
  except Exception as e:
    print(f"Error sending email: {str(e)}")
    return False
@app.route('/api/predict', methods=['POST'])
@login_required
def predict():
      if model is None:
```

```
        return jsonify({'error': 'Model not loaded'}), 500
    try:
        data = request.get_json()
        # Get user email - prioritize session email (from Google auth) over form email
        user_email = session.get('email') or data.get('email')
        if not user_email:
            return jsonify({'error': 'Email address is required'}), 400
        # Prepare input data
        input_data = {
            'Tenure': float(data['Tenure']),
            'CityTier': int(data['CityTier']),
            'WarehouseToHome': float(data['WarehouseToHome']),
            'HourSpendOnApp': float(data['HourSpendOnApp']),
            'NumberOfDeviceRegistered': int(data['NumberOfDeviceRegistered']),
            'SatisfactionScore': int(data['SatisfactionScore']),
            'NumberOfAddress': int(data['NumberOfAddress']),
            'Complain': int(data['Complain']),
            'OrderAmountHikeFromlastYear': float(data['OrderAmountHikeFromlastYear']),
            'CouponUsed': float(data['CouponUsed']),
            'OrderCount': float(data['OrderCount']),
            'DaySinceLastOrder': float(data['DaySinceLastOrder']),
            'CashbackAmount': float(data['CashbackAmount']),
            'PreferredLoginDevice': data['PreferredLoginDevice'],
            'PreferredPaymentMode': data['PreferredPaymentMode'],
            'Gender': data['Gender'],
            'MaritalStatus': data['MaritalStatus'],
            'PreferedOrderCat': data['PreferedOrderCat']
        }
        # Convert to DataFrame
        input_df = pd.DataFrame([input_data])
    # One-hot encoding
```

```
input_df_encoded = pd.get_dummies(input_df, columns=[
    'PreferredLoginDevice', 'PreferredPaymentMode', 'Gender', 'MaritalStatus',
'PreferedOrderCat'
])

# Ensure all required columns
for col in feature_cols:
    if col not in input_df_encoded.columns:
        input_df_encoded[col] = 0

# Reorder columns
input_df_encoded = input_df_encoded[feature_cols]
input_df_encoded = input_df_encoded.astype(int)

# Make prediction
prediction = model.predict(input_df_encoded)[0]
probability = model.predict_proba(input_df_encoded)[0][1]
result = 'Churn' if prediction == 1 else 'No Churn'

# Save to database
timestamp = datetime.now().isoformat()
conn = sqlite3.connect('database.db')
cursor = conn.cursor()
# Send email
email_sent = send_prediction_email(user_email, result, probability, input_data)

cursor.execute('''
    INSERT INTO predictions (timestamp, user_email, input_data,
prediction_result, probability, email_sent)
    VALUES (?, ?, ?, ?, ?, ?)
''', (timestamp, user_email, json.dumps(input_data), result, float(probability), 1 if
email_sent else 0))
conn.commit()
```

```
        conn.close()
        return jsonify({
            'prediction': result,
            'probability': float(probability),
            'probability_formatted': f'{probability:.2%}',
            'email_sent': email_sent
        })

    except Exception as e:
        print(f"Prediction error: {str(e)}")
        return jsonify({'error': str(e)}), 400

if __name__ == '__main__':
        app.run(debug=True, port=5051)
```