ASHESI UNIVERSITY

Optimizing Customer Retention: Comparing the Efficacy of Multiple Machine Learning Techniques in Predicting E-commerce Churn

UNDERGRADUATE THESIS PROJECT

B.Sc. Management Information Systems

Godfred Kogkane

2025

ASHESI UNIVERSITY

Optimizing Customer Retention: Comparing the Efficacy of Multiple Machine Learning Techniques in Predicting E-commerce Churn

UNDERGRADUATE THESIS PROJECT

Undergraduate Thesis Project submitted to the Department of Computer Science & Information Systems, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Management Information Systems.

Godfred Kogkane

2025

DECLARATION

I hereby declare that this Undergraduate Thesis Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate’s Signature:

....

Candidate’s Name:

....

Date:

....

I hereby declare that preparation and presentation of this Undergraduate Thesis Project was supervised in accordance with the guidelines on supervision of Undergraduate Thesis Project laid down by Ashesi University.

Supervisor’s Signature:

....

Supervisor’s Name:

....

Date:

....

Acknowledgments

I would like to express my deepest appreciation to all those who contributed to the successful completion of this capstone project.

First and foremost, I am sincerely grateful to **Professor Robert Sowah Adjetey**, my capstone supervisor, for his invaluable guidance, constructive feedback, and consistent support throughout this research. His mentorship played a crucial role in shaping the direction and quality of this work.

I would also like to thank **Mr. Enoch Opoku**, my second reader, whose insightful comments and encouragement greatly enhanced the final output of this project. His support was instrumental in refining both the technical and analytical aspects of this study.

I extend my appreciation to **all the faculty members at Ashesi University** who have supported me academically and intellectually throughout my undergraduate journey. Their dedication to teaching and mentorship has had a lasting impact on my personal and academic growth.

A special thanks goes to the **Mastercard Foundation** for generously sponsoring my education and to **Ashesi University** for providing a conducive and enriching environment for learning. I am particularly grateful for the opportunity to study under the Mastercard Foundation Scholarship, which made this academic journey possible.

Lastly, I would like to acknowledge the **encouragement and unwavering support of my family and friends**. Their presence and belief in me have been a source of strength throughout this academic endeavor.

Abstract

The rapid growth of e-commerce has intensified competition, making customer retention a critical priority for businesses. As acquiring new customers remains significantly more costly than retaining existing ones, accurately predicting customer churn has become essential for sustaining profitability. This capstone project investigates and compares the effectiveness of various machine learning (ML) algorithms in predicting customer churn within the e-commerce domain. The study explores a broad range of models, including traditional classifiers (Decision Tree, Logistic Regression, K-Nearest Neighbors), ensemble methods (Random Forest, Gradient Boosting, XGBoost, LightGBM), and deep learning architectures (Convolutional Neural Network and Long Short-Term Memory network), using a structured, feature-rich customer dataset.

Models were evaluated using accuracy, F1 Score, and the Matthews Correlation Coefficient (MCC), with additional computational time and space complexity analysis to assess real-world deployment feasibility. Among all models tested, XGBoost emerged as the best performer, achieving a test accuracy of 98.40%, F1 Score of 0.9516, and MCC of 0.9424, while maintaining efficient training and prediction times. LightGBM and Random Forest also demonstrated strong performance, whereas deep learning models underperformed due to the static and tabular nature of the dataset, highlighting their limited suitability in such contexts.

The study further validated its findings through statistical significance testing and comparative literature analysis, establishing alignment with prior high-impact studies while offering new insights into model performance in class-imbalanced, structured data environments. The results underscore the practical value of ensemble models for customer retention strategies and advocate for model selection approaches that balance predictive accuracy with interpretability and computational efficiency. This work contributes to both academic discourse and industry practice by providing a data-driven roadmap for mitigating churn in the evolving e-commerce landscape.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

The e-commerce industry has undergone significant transformation over the past decade, fueled by technological innovation, increased internet penetration, and evolving consumer preferences. The proliferation of smartphones, digital payment systems, and mobile applications has drastically reshaped how consumers engage with retail platforms. According to Chevalier [3], global retail e-commerce sales totaled approximately $5.8 trillion in 2023 and are projected to grow to $8.1 trillion by 2026, underscoring the industry's rising economic relevance.

Despite this upward trajectory, customer churn, defined as the rate at which existing customers cease transacting with a business [5], remains a pervasive issue in the e-commerce landscape. Churn rates in this sector are often volatile, ranging between 20% and 80% depending on the business niche, as estimated by Reichheld and Schefter [19]. High churn rates diminish customer lifetime value and brand equity while inflating marketing and acquisition costs. It is widely cited that acquiring a new customer is five times more expensive than retaining an existing one [10]. Furthermore, a modest 5% increase in customer retention can potentially lead to over a 25% increase in profits [19].

In response to this challenge, the application of data-driven methods, particularly machine learning (ML), has gained traction in customer churn prediction. Unlike traditional statistical models that rely on linear assumptions and historical summaries, ML techniques excel at uncovering non-linear patterns in high-dimensional data, enabling proactive identification of at-risk customers [2, 6, 23]. These predictive capabilities support the development of targeted retention strategies, making machine learning an increasingly critical component of modern Customer Relationship Management (CRM) systems in digital commerce [22].

1.2 Problem Statement

Although machine learning has become a powerful tool for churn prediction, a key issue persists: determining the most effective algorithm for this task, especially within the high-dimensional and often imbalanced datasets typical of e-commerce platforms. While existing studies have examined a variety of models, from Decision Trees and Logistic Regression to neural networks and ensemble methods like XGBoost and LightGBM, comparative studies that also consider computational efficiency and evaluation metrics like the Matthews Correlation Coefficient (MCC) are still limited and identifying the best model in the ecommerce sector is limited.

Moreover, many works fail to integrate robust statistical evaluations or consider the real-world constraints of model interpretability, scalability, and resource efficiency. This study seeks to bridge these gaps by conducting a thorough comparative analysis of traditional, ensemble, and deep learning models. It also assesses their performance using MCC and analyzes computational complexity regarding time and space requirements.

1.3 Research Objectives

The objectives of this research are:

1. Compare a diverse range of machine learning techniques, including traditional models, ensemble methods, and deep learning architectures, for predicting customer churn in e-commerce.
2. Evaluate each model using robust performance metrics, including accuracy, F1 Score, and Matthews Correlation Coefficient (MCC).
3. Analyze and report computational time and space complexity for all models to determine their feasibility in real-world applications.
4. Conduct statistical significance testing (e.g., paired t-tests) to validate whether observed differences in model performance are meaningful and not due to chance.

5. Benchmark the performance of the top-performing model against leading studies in customer churn prediction to validate effectiveness and relevance.

1.4 Research Questions

The following research questions guide this capstone project:

1. Which machine learning techniques (traditional, ensemble, and deep learning) are most effective for predicting customer churn in the e-commerce domain?
2. How do the predictive performances of models compare based on accuracy, F1 Score, and Matthews Correlation Coefficient (MCC)?
3. What are the trade-offs between model predictive performance, interpretability, and computational efficiency for real-world deployment?
4. Are the observed differences in model performances statistically significant based on rigorous testing, such as paired t-tests?
5. How does the best-performing model benchmark against findings from leading customer churn prediction literature studies?

1.5 Significance of the Study

This research contributes to both academic inquiry and industry practice. From an academic perspective, it expands the existing literature by offering a comprehensive and comparative analysis of traditional, ensemble, and deep learning models using advanced metrics like MCC and complexity evaluations. This helps fill methodological gaps in current churn prediction studies. The findings provide empirical guidance for industry practitioners on selecting appropriate machine learning models based on accuracy, resource usage, and interpretability. Given the financial implications of churn and the cost-efficiency of retention strategies, businesses can leverage these insights to adopt more effective and scalable solutions for customer retention.

1.6 Research Gap

While several studies have addressed customer churn using machine learning, most either focus on a narrow set of models or overlook key considerations such as model interpretability, computational complexity, or robust performance evaluation with metrics like MCC. Additionally, limited attention has been paid to how customer behavioral features influence model efficacy in e-commerce. This study addresses these limitations by exploring a wide array of ML algorithms, spanning traditional, ensemble, and deep learning models, and rigorously comparing them using multiple metrics and complexity analyses. The research also investigates the influence of specific customer features on model performance, providing actionable insights for enhancing customer retention through predictive analytics.

Chapter 2: Related Work / Literature Review

2.1 Machine Learning Techniques for Customer Churn Prediction

Customer churn prediction has emerged as a critical area of focus in data-driven business strategies, particularly in dynamic and competitive sectors like e-commerce. Accurately identifying at-risk customers enables firms to implement timely retention measures, thereby reducing revenue loss and customer acquisition costs. Machine learning (ML) techniques have gained prominence due to their effectiveness in modeling complex, non-linear relationships within customer behavior data. A growing body of literature has explored a wide range of ML techniques to tackle this challenge.

2.2 Traditional Machine Learning Algorithms

Traditional ML algorithms have been extensively applied in churn prediction studies, including Decision Trees, Support Vector Machines (SVM), Random Forests, and Logistic Regression. Patil et al. [17] demonstrated that applying ML to transactional data could help mitigate revenue loss in the retail industry by identifying churn signals early. Similarly, Jaiswal et al. [9] showed that combining k-means clustering with SVM enhanced predictive accuracy by accounting for customer heterogeneity. Jaiswal et al. [9] examined the banking sector's multiple classifiers, including K-Nearest Neighbors (KNN), Decision Trees, and Random Forests. Their findings highlighted the superiority of Random Forests in high-dimensional environments. These results support further investigation into how traditional ML models perform specifically within the e-commerce context.

2.3 Hybrid Machine Learning Models

Hybrid models, which combine the strengths of traditional algorithms with deep learning components, have shown promise in recent studies. Liu et al. [13] integrated Logistic Regression with deep learning models to improve feature learning and churn prediction accuracy in e-commerce applications. Ensemble methods, such as Gradient

Boosting, AdaBoost, XGBoost, LightGBM, and CatBoost, have gained popularity for their high predictive performance. Lubis et al. [14] found that Gradient Boosting outperformed Logistic Regression, achieving 91% test accuracy. Similarly, Jahan and Sanam [6] reported that CatBoost achieved near-perfect results on their dataset. Raeisi and Sajedi [13] applied Gradient Boosted Trees in an online food delivery setting, achieving an accuracy of 86.9%, emphasizing the consistent strength of ensemble techniques in churn modeling.

2.4 Ensemble Methods

Ensemble methods such as Gradient Boosting, Random Forests, CatBoost, XGBoost, and LightGBM have proven particularly effective because they aggregate multiple weak learners to produce more robust models. Lubis et al. [14] illustrated that Gradient Boosting outperformed Logistic Regression with a 91% testing accuracy on an e-commerce dataset. Jahan and Sanam [8] reported perfect prediction accuracy using CatBoost, though such results warrant careful scrutiny to avoid overfitting. Raeisi and Sajedi [18] used Gradient Boosted Trees for churn prediction in the food delivery sector, achieving an accuracy of 86.9%. These studies reinforce the growing preference for ensemble methods in scenarios involving complex and imbalanced data, which are common in e-commerce churn datasets.

2.5 Deep Learning Models

Deep learning techniques, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are increasingly used for churn prediction due to their ability to extract high-level features from sequential and temporal data. Chouiekh and El Haj [4] employed CNNs on a telecom dataset, achieving an F1 score of 91%, outperforming SVM and Random Forest. Mahajan and Singh [15] applied CNN and RNN models in a retail environment, showing that deep learning models could outperform traditional models in capturing customer behavior patterns. However, the resource demands of deep learning models can limit their applicability for small or resource-constrained businesses, which is a critical factor evaluated in this study.

2.6 Time-Series Models

Long Short-Term Memory (LSTM) networks, a variant of RNNs, have proven effective in modeling sequential dependencies in time-series data. Zhang and Chen [24] used LSTM models to predict e-commerce customer churn and demonstrated improved performance by incorporating temporal behavior data. However, the complexity of LSTM training and higher computational cost limit their practicality in some deployment scenarios, motivating comparative evaluation with more lightweight alternatives.

2.7 Feature Engineering and Customer Segmentation

Effective feature engineering is a cornerstone of predictive modeling, particularly for churn prediction. Subramanya and Somani [21] showed that incorporating interaction features from clickstream and transaction data enhanced classifier performance. Similarly, Sundarajan and Narayanan [22] emphasized the significance of deriving features from both explicit (e.g., transaction count) and implicit (e.g., session duration) customer behavior data. Customer segmentation further enhances model performance by tailoring prediction strategies. Jaiswal et al. [9] applied k-means clustering prior to modeling with SVM, achieving notable improvements. This study explores a similar approach by examining the influence of individual features on model accuracy and MCC scores.

2.8 Model Evaluation and Performance Metrics

Accurate evaluation of churn prediction models requires metrics that address class imbalance. While accuracy, precision, recall, and F1-score are commonly used, they may not fully capture performance nuances in skewed datasets. Al Rahib et al. [2] found that SVM achieved 83.45% accuracy, but without supporting metrics such as MCC, the model's true efficacy remains unclear. Matthews Correlation Coefficient (MCC) is a balanced metric that considers all four confusion matrix categories and is especially useful for binary classification with class imbalance. Kumar et al. [10] evaluated models using multiple metrics but did not incorporate MCC. This study addresses that limitation by including MCC in model evaluation alongside accuracy and F1-score. Moreover, this research introduces time and space complexity analysis to highlight practical implications for deployment.

2.9 Gaps in the Literature

Despite the increasing number of studies on churn prediction, notable gaps remain. First, most works examine only a narrow subset of machine learning models without providing a comprehensive comparison, especially involving both ensemble and deep learning techniques. Second, there is limited discussion on performance metrics beyond accuracy and F1-score, with few studies adopting MCC despite its appropriateness for imbalanced datasets. Third, computational complexity, vital for model scalability and real-world integration, is often ignored. Lastly, few studies benchmark their findings against top-performing or most-cited models in the literature. This research fills these gaps by:

1. Comparing traditional, ensemble, and deep learning models for e-commerce churn prediction.
2. Evaluating models using Accuracy, F1 Score, and Matthews Correlation Coefficient (MCC).
3. Assessing computational time and space complexity for deployment feasibility.
4. Conducting statistical significance tests to validate model performance differences.
5. Benchmarking the best model against findings from leading churn prediction studies.

Through this, the study aims to provide actionable insights for both academic and industry stakeholders focused on minimizing churn in the evolving e-commerce landscape.

Chapter 3: Methodology and Experimental Setup

3.1 Research Design

This study adopts a **quantitative research design** to systematically compare the performance of multiple machine learning algorithms for customer churn prediction within the e-commerce sector. The research focuses on classification models, leveraging both traditional algorithms, such as Decision Trees, Random Forests, K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVM), Gaussian Naive Bayes, and ensemble methods like Gradient Boosting, AdaBoost, XGBoost, and LightGBM, and deep learning approaches, specifically Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTM). These models are trained and evaluated using a comprehensive e-commerce dataset composed of customer behavior and transactional history. The research design emphasizes model interpretability, predictive performance, computational efficiency, and evaluation robustness using metrics tailored for imbalanced classification problems, such as the **Matthews Correlation Coefficient (MCC)**.

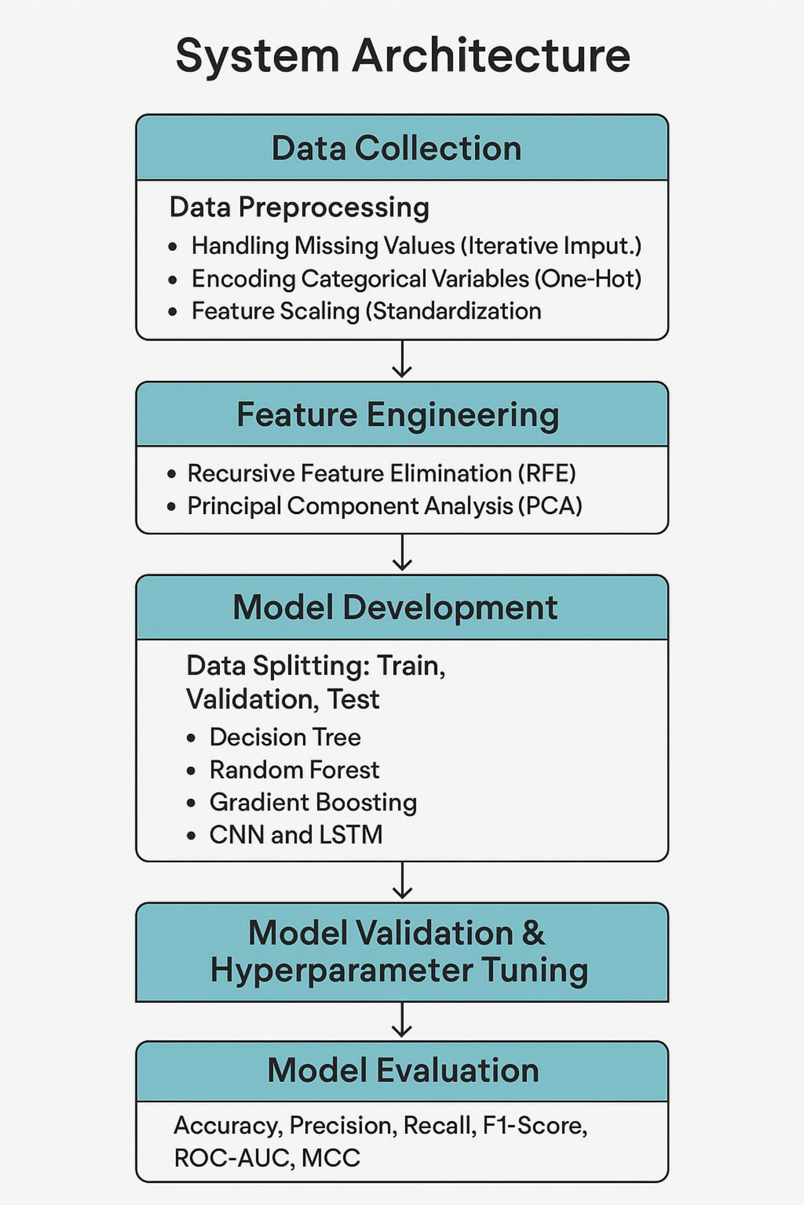

System Architecture
Data Collection
Data Preprocessing
• Handling Missing Values (Iterative Imput.)
• Encoding Categorical Variables (One-Hot)
• Feature Scaling (Standardization
Feature Engineering
• Recursive Feature Elimination (RFE)
• Principal Component Analysis (PCA)
Model Development
Data Splitting: Train, Validation, Test
• Decision Tree
• Random Forest
• Gradient Boosting
• CNN and LSTM
Model Validation & Hyperparameter Tuning
Model Evaluation
Accuracy, Precision, Recall, F1-Score, ROC-AUC, MCC

Figure 3.1: System Architecture Diagram

The System Architecture Diagram in Figure 3.1 visually represents the end-to-end workflow used in this study. The design is structured into the following key stages:

1. Data Collection & Preprocessing: This includes handling missing values through Iterative Imputation, encoding categorical variables using One-Hot Encoding, and applying feature scaling via standardization techniques to ensure numerical stability across models.

2. Feature Engineering: Advanced techniques such as Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) are applied to identify the most relevant features, reduce dimensionality, and improve model performance.
3. Model Development: The dataset is split into training, validation, and test sets using stratified sampling. A diverse suite of models is developed and optimized using dedicated pipelines, covering both traditional machine learning and deep learning techniques.
4. Model Validation & Hyperparameter Tuning: Grid Search and Random Search methods are used to fine-tune hyperparameters, ensuring each model reaches optimal performance during cross-validation.
5. Model Evaluation: Models are evaluated using a comprehensive set of performance metrics: Accuracy, Precision, Recall, F1-Score, ROC-AUC, and Matthews Correlation Coefficient (MCC). The computational time and space complexities are also assessed to inform real-world feasibility and resource requirements.

3.2 Data Collection and Preprocessing

The dataset used in this study was obtained from a publicly available e-commerce dataset from Kaggle that captures various dimensions of customer behavior, including transactional history, session activity, product views, and purchase frequency. This data contains records of 5630 customers with 20 features, providing a rich foundation for training machine learning models to predict customer churn by leveraging behavioral patterns and historical interactions. To ensure the quality and integrity of the data, extensive preprocessing was conducted. The preprocessing pipeline involved several critical steps:

1. **Handling Missing Values**: Missing entries in the dataset were addressed using Iterative Imputation, a multivariate approach that models each feature as a function of the others to fill in missing data points.

2. **Encoding Categorical Variables**: Categorical features such as gender, product category, or region were converted into a numerical format using One-Hot Encoding, which avoids introducing unintended ordinal relationships into the model.

3. **Feature Scaling**: Continuous features were standardized using z-score normalization to ensure uniformity in feature magnitudes, which is especially important for distance-based models like K-Nearest Neighbors and neural networks.

4. **Exploratory Data Analysis (EDA)**: Prior to modeling, an EDA was performed to understand variable distributions, detect outliers, and visualize correlations between features. This step helped identify potential data leakage, refine feature selection strategies, and guide subsequent modeling decisions. The importance of this process is also supported in the work of Jahan and Sanam [8], who emphasized structured EDA in improving model interpretability and feature relevance.

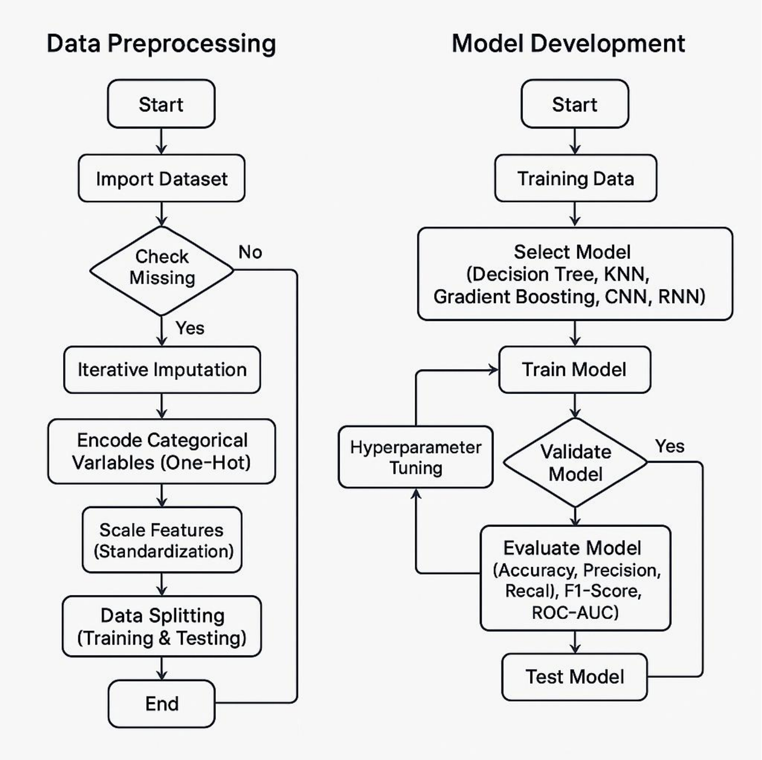

Data Preprocessing
Start
Import Dataset
Check Missing
No
Yes
Iterative Imputation
Encode Categorical Varlables (One-Hot)
Scale Features (Standardization)
Data Splitting (Training & Testing)
End
Model Development
Start
Training Data
Select Model (Decision Tree, KNN, Gradient Boosting, CNN, RNN)
Train Model
Hyperparameter Tuning
Validate Model
Yes
Evaluate Model (Accuracy, Precision, Recal), F1-Score, ROC-AUC)
Test Model

Figure 3.2: Data Preprocessing Flowchart

Figure 3.2 presents a step-by-step flowchart of the preprocessing pipeline. This visual representation outlines the logical progression of operations applied to the raw dataset, ensuring that the data fed into the modeling phase is clean, consistent, and analytically robust. The preprocessing stage serves as a critical foundation for effective feature engineering and model development. Proper data transformation enhances model performance and supports more accurate and generalizable predictions.

3.3 Feature Engineering and Selection

Feature engineering and selection are crucial to the research methodology, as they significantly influence the machine learning models' predictive power and generalization ability. This study will use Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) as primary methods to identify and retain the most relevant features while eliminating noise and redundancy. These techniques enhance model performance by reducing dimensionality and focusing the learning algorithm on the most informative attributes.

In addition to these techniques, advanced feature engineering strategies will be employed to extract implicit patterns from customer interaction data. Drawing from the work of Subramanya and Somani [21], features such as clickstream logs, browsing frequency, and behavioral timestamps will be engineered to capture deeper insights into customer activity. These engineered features help reveal latent variables that are not immediately observable in the raw dataset.

Furthermore, the insights of Sundarajan and Narayanan [22] will guide the selection of explicit features such as customer demographics, transaction frequency, and purchase history. Their findings underscore the importance of selecting variables that reflect both behavioral trends and customer identity. By combining implicit and explicit features, the study ensures a comprehensive representation of customer behavior, essential for improving the accuracy and robustness of churn prediction models.

3.4 Model Training and Evaluation

To ensure a fair and effective comparison among models, the dataset will be split into training, validation, and testing sets in a 70:15:15 ratio. This approach helps in preventing overfitting and ensures that the model generalizes well to unseen data. During the training phase, hyperparameter tuning will be conducted using both grid search and random search strategies. These optimization techniques aim to identify the best parameter combinations that maximize model performance during the validation phase. Model performance will be assessed using a combination of evaluation metrics designed to address the challenges of class imbalance in churn datasets. Accuracy will measure the proportion of correct predictions over the total number of predictions. Precision will reflect the proportion of predicted churns that were actual churns, while recall will measure the model's ability to identify actual churn cases.

The F1-score, which is the harmonic mean of precision and recall, will be used to balance the trade-off between false positives and false negatives. Together, these metrics provide a well-rounded assessment of each model's effectiveness. In line with recommendations by De et al. [6] and Kumar et al. [10], using multiple metrics is essential for robust evaluation, especially in imbalanced scenarios common in customer churn datasets. By combining these quantitative measures with the Matthews Correlation Coefficient (MCC) and computational complexity analysis, discussed in later sections, this study ensures that the evaluation framework is both comprehensive and practical for real-world applications.

3.5 Model Comparison

This study's comparative evaluation of machine learning models is grounded in a robust performance framework that assesses each model across several key metrics. A diverse range of models was trained and tested, including traditional models such as Decision Tree, K-Nearest Neighbors, Logistic Regression, Support Vector Machine (SVM), and Gaussian Naive Bayes; ensemble models like Random Forest, Gradient Boosting, AdaBoost, XGBoost, and LightGBM; and deep learning models including Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). Special attention was given to ensemble models, which demonstrated strong predictive power. XGBoost emerged as the best-performing model overall, achieving a cross-validation mean accuracy of 95.14%, a test accuracy of 98.40%, and a Matthews Correlation Coefficient (MCC) of 0.9424, indicating its exceptional ability to handle class imbalance. LightGBM also performed impressively with a test accuracy of 96.98% and an MCC of 0.8896. These results align with findings in the literature that highlight the strength of boosting algorithms in capturing complex, nonlinear relationships.

In contrast, deep learning models showed mixed performance. CNN achieved a test accuracy of 93.16% and an MCC of 0.7519, indicating solid performance, though not surpassing the top ensemble models. LSTM, while theoretically suited for sequential data, underperformed significantly in this context, with a test accuracy of 84.10% and an MCC of 0.2721. Despite their architectural sophistication, these findings suggest that deep learning models may not always outperform well-tuned traditional models on tabular e-commerce data. This comparative analysis confirms that ensemble models, particularly XGBoost and LightGBM, are more effective in churn prediction within the dataset used, both in terms of predictive accuracy and computational reliability.

3.6 Modules Developed

The research methodology was modularized by developing several key components within the Jupyter Notebook environment. These modules were designed to streamline each stage of the machine learning pipeline and ensure reproducibility and clarity. The **Data Preprocessing Module** performed initial loading, transformation, and cleaning of the dataset. Iterative imputation was applied to address missing values, categorical variables were encoded using OneHotEncoder, and numerical features were scaled using StandardScaler to standardize input for model training.

The **Exploratory Data Analysis (EDA) and Visualization Module** utilized tools like Seaborn, Matplotlib, and Missingno to comprehensively view the dataset's structure, including churn distribution, feature correlation, and outlier detection. These visualizations informed feature engineering and supported the detection of class imbalance.

The **Feature Engineering and Selection Module** incorporated Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) for feature selection, while new features such as transaction frequency and customer recency were generated to enhance model input representation.

The **Model Training and Validation Module** supported the implementation and tuning of models, including Decision Tree, K-Nearest Neighbors, Logistic Regression, SVM, Random Forest, AdaBoost, Gradient Boosting, XGBoost, LightGBM, CNN, and LSTM. Stratified train-validation-test splits (70:15:15) were used to maintain class balance, and hyperparameter tuning was conducted using both grid and random search.

Finally, the **Model Comparison and Interpretation Module** aggregated performance results across all models. It generated comparative bar charts, classification reports, and MCC scores to provide an in-depth understanding of model effectiveness, particularly with respect to class imbalance. The module also presented trade-offs between model complexity, interpretability, and accuracy, supporting informed model selection.

3.7 Platform and Tools Used

All experimentation and model development were conducted within the Jupyter Notebook environment provided by Ashesi University. The interactive nature of this platform allowed for seamless integration with Python libraries and frameworks, efficient debugging, and detailed documentation of processes. The cloud-accessible environment facilitated iterative development, especially during model tuning and evaluation.

3.8 Hardware and Environment Specifications

All experiments were run in Ashesi University's JupyterHub server environment, which runs on a Linux-based operating system powered by an Intel Core i7 processor and 32 GB of RAM. Given the moderate size of the dataset, CPU-based training was sufficient for the majority of models. Despite training deep learning models like CNNs and LSTMs, no GPU acceleration was used, as the study's objective was to benchmark model efficacy rather than computational speed at scale. The consistent hardware environment ensured reproducible results and prevented discrepancies during model evaluation.

3.9 Software Tools and Libraries

The programming language used throughout the project was Python 3.10. Core libraries included Pandas and NumPy for data manipulation, Matplotlib, Seaborn, and Missingno for visualization, and Scikit-learn for traditional machine learning models and metrics. TensorFlow and Keras were utilized to build and evaluate CNN and LSTM models. Openpyxl was used to parse the data dictionary information provided in Excel format. While all training was conducted on CPU infrastructure, performance was robust due to the relatively lightweight data and batch-wise model processing in the training pipelines.

3.10 Expected Outcomes

This study is expected to provide a robust comparative evaluation of various machine learning techniques for customer churn prediction in e-commerce. The results will offer evidence-based recommendations on the most suitable models and feature engineering strategies for improving retention. By leveraging metrics such as the Matthews Correlation Coefficient (MCC), the study provides deeper insight into model performance in imbalanced scenarios. Ultimately, the findings contribute to both the academic understanding of churn prediction and the practical efforts of e-commerce platforms seeking to reduce customer attrition through predictive analytics.

Chapter 4: Experiments and Results

4.1 Data Preparation

The preparation of the dataset played a foundational role in ensuring the reliability and effectiveness of the machine learning models developed for customer churn prediction. The raw dataset consisted of customer behavioral attributes, transactional records, and session-based features, all of which required cleaning and transformation to be suitable for supervised learning. One of the first challenges addressed was the presence of missing values in key features such as PreferredLoginDevice, Gender, and PreferredPaymentMode. These missing entries were identified as Missing at Random (MAR), which means their absence could be explained by other observable variables in the dataset. Iterative Imputation was used to preserve the dataset's underlying structure and correlation between features. This method builds a predictive model for each feature with missing values, using all other features as inputs. Unlike traditional imputation methods like mean or mode substitution, this technique maintained the distributional properties of the dataset while minimizing information loss.

Additionally, categorical variables such as Gender, PreferredLoginDevice, and PaymentMode were encoded using One-Hot Encoding. This method transformed each unique category into a separate binary variable, allowing the models to process them effectively without misinterpreting any inherent ordering. While this increased the dimensionality of the dataset, it ensured that each category was treated fairly and equally, thus preventing biased learning. Care was taken to manage the resulting sparsity and feature explosion, especially during the model selection and dimensionality reduction phases.

4.2 Data Splitting

After preprocessing, the dataset was partitioned using an 80:20 train-test split. This ensured that a large portion of the data was available for learning patterns while preserving a distinct portion for evaluating model generalization. Since the problem of customer churn naturally involves class imbalance, a stratified sampling technique was applied, with churned customers being significantly fewer than non-churned ones. Stratification preserved the original distribution of the churn classes in both training and test sets, thereby reducing bias and making evaluation metrics like F1 Score and MCC more representative of real-world model performance.

4.3 Overview of Model Performance

Twelve machine learning models were developed and evaluated using a combination of test accuracy, F1 Score, and the Matthews Correlation Coefficient (MCC). These metrics offered a multi-dimensional perspective on model effectiveness, especially in handling imbalanced data. Test accuracy provided a general indication of correctness across all predictions, while the F1 Score assessed the balance between precision and recall, which is critical in identifying churners. Conversely, MCC served as a more holistic metric, considering true and false positives and negatives to evaluate predictive quality, even in imbalanced settings.

Visualizations of these metrics offered immediate insights into model performance. Models such as XGBoost, LightGBM, Random Forest, and Decision Tree consistently outperformed others across all three metrics. These ensemble and tree-based models demonstrated superior predictive accuracy and better balance between sensitivity to churners and stability across all classes. In contrast, deep learning models, particularly LSTM and CNN, struggled to detect churned customers, likely due to the limitations of their architectures when applied to non-sequential, tabular data.

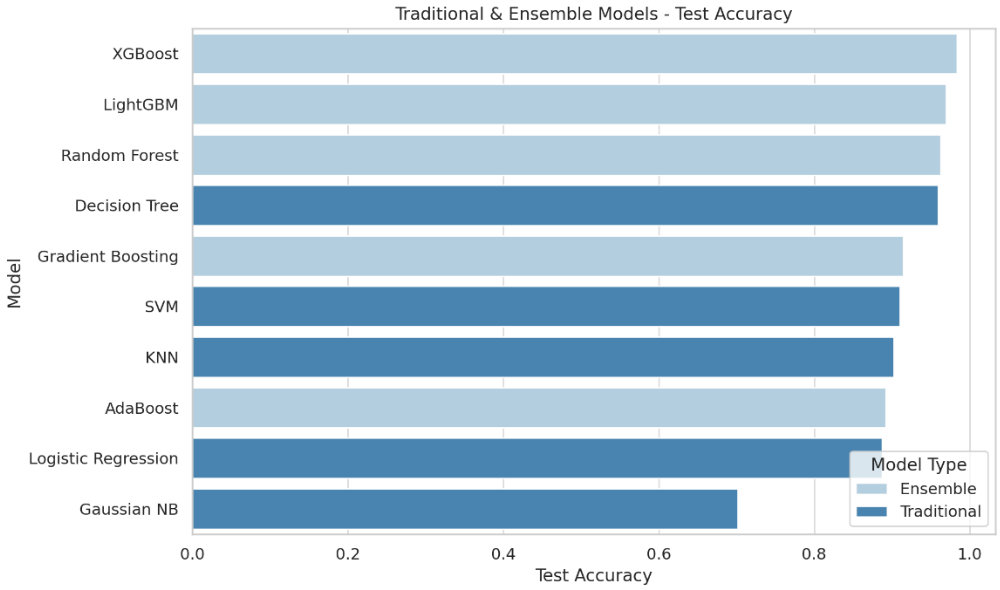


Figure 4.3: Test accuracy comparison of Traditional and Ensemble models

Figure 4.3 compares the test accuracy scores of traditional and ensemble models. XGBoost and LightGBM demonstrated the highest accuracy, indicating superior generalization on unseen churn data.

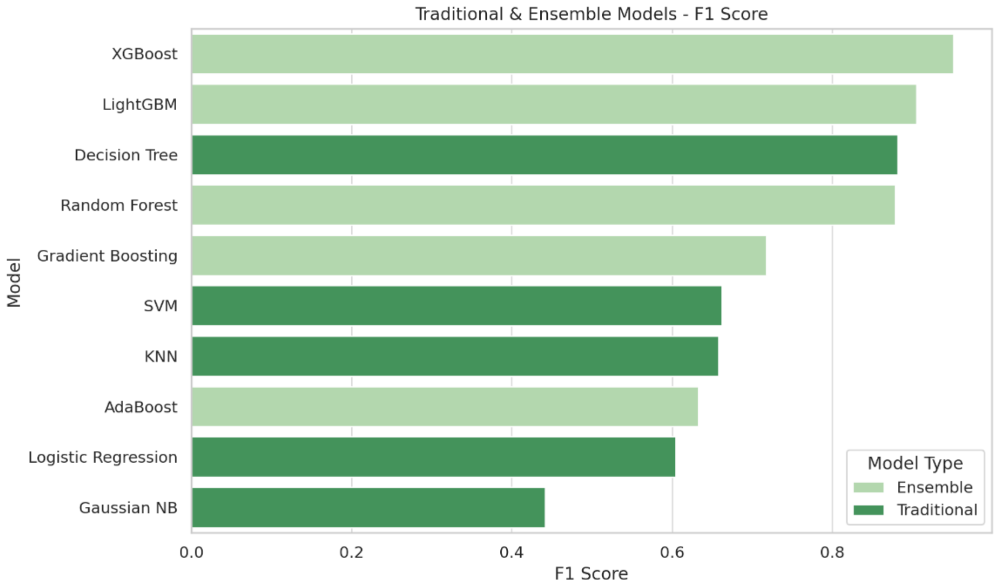


Figure 4.3: F1 Score comparison of Traditional and Ensemble models

Figure 4.3 presents the F1 scores, highlighting XGBoost and LightGBM as the most balanced models in terms of precision and recall, critical for churn prediction.

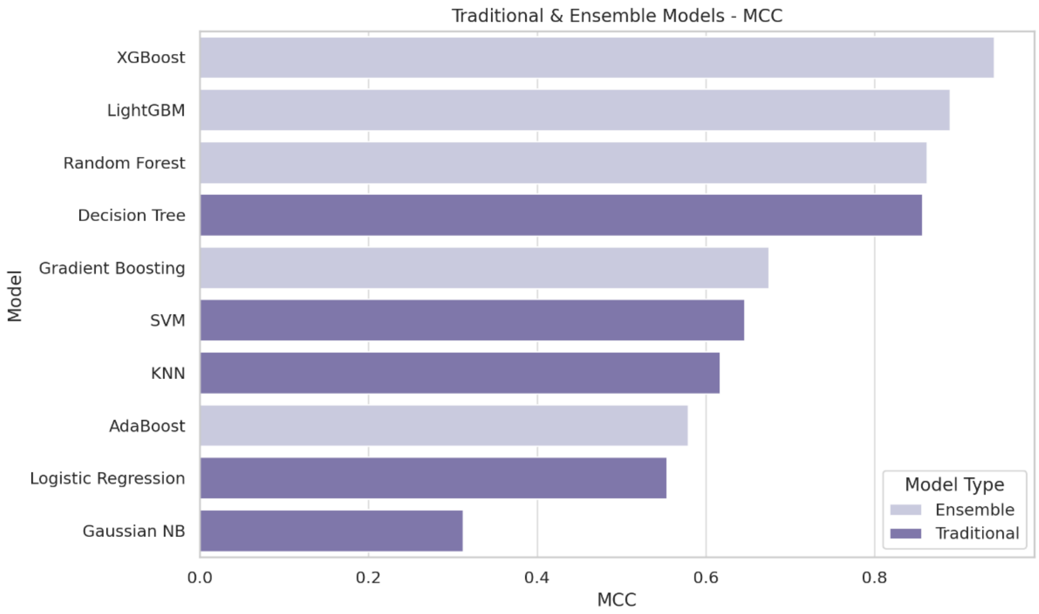


Figure 4.3: MCC comparison of Traditional and Ensemble models

Figure 4.3 shows the Matthews Correlation Coefficient for each model. Ensemble models, particularly XGBoost, strongly correlate with true outcomes, even with imbalanced data.

4.4 Traditional and Ensemble Model Analysis

Among all models evaluated, XGBoost emerged as the top performer. It achieved a test accuracy of 0.9840, an F1 Score of 0.9516, and an MCC of 0.9424. Its dominance across all metrics can be attributed to its ability to implement gradient boosting with sophisticated regularization techniques that control overfitting, its ability to handle sparse input, and its use of parallel computation, which optimizes performance. XGBoost's design made it particularly effective in learning complex interactions between features while remaining robust in the face of class imbalance.

Closely following XGBoost, LightGBM also demonstrated outstanding performance, achieving a test accuracy of 0.9698, an F1 Score of 0.9056, and an MCC of 0.8896. It benefited from a histogram-based learning algorithm, which made it computationally more efficient than XGBoost while delivering comparable results. Its ability to handle large feature spaces and capture non-linear patterns makes it ideal for churn prediction tasks that involve diverse customer attributes.

Random Forest also performed commendably, with a test accuracy of 0.9627, an F1 Score of 0.8786, and an MCC of 0.8626. As a bagging ensemble of decision trees, it brought strong generalization capabilities and robustness to overfitting. It successfully captured the non-linear relationships among features and was less affected by noise in the data. While not as powerful as its ensemble counterparts, the Decision Tree model achieved a solid test accuracy of 0.9591, an F1 Score of 0.8814, and an MCC of 0.8570. Its simplicity and interpretability make it a practical tool for business decision-making, especially when explaining model output to non-technical stakeholders.

Models like K-Nearest Neighbors (KNN), Logistic Regression, and Support Vector Machines (SVM) delivered moderate performance. KNN reached a test accuracy of 0.9023, but its F1 Score and MCC dropped to 0.6584 and 0.6172, respectively, indicating a challenge in detecting minority class instances. Logistic Regression, known for its ease of interpretation, achieved a test accuracy of 0.8872, but its F1 Score (0.6044) and MCC (0.5539) highlighted its limited capacity in modeling complex, non-linear relationships. Similarly, SVM achieved a respectable accuracy of 0.9103, but it did not excel in F1 Score (0.6622) or MCC (0.6464), likely due to its sensitivity to feature scaling and its limitations in multi-dimensional decision boundaries when applied without kernel tuning.

Table 4.4: Performance Summary of Traditional and Ensemble Models

Model	CV Mean Accuracy	Test Accuracy	F1 Score	MCC	Key Observations
XGBoost	0.9514	0.9840	0.9516	0.9424	Highest accuracy and MCC; Fast and robust
LightGBM	0.9478	0.9698	0.9056	0.8896	Excellent balance between speed and performance

Random Forest	0.9405	0.9627	0.8786	0.8626	Strong performance; Slightly longer training
Decision Tree	0.9218	0.9591	0.8814	0.8570	High interpretability; Strong individual model
Gradient Boosting	0.9130	0.9147	0.7176	0.6747	Decent accuracy; Moderate class balance
SVM	0.9079	0.9103	0.6622	0.6464	Struggled with minority class detection
K-Nearest Neighbors	0.8923	0.9023	0.6584	0.6172	Moderate performance; Sensitive to feature scale
AdaBoost	0.8948	0.8917	0.6325	0.5789	Lower F1 Score; Vulnerable to noise
Logistic Regression	0.8934	0.8872	0.6044	0.5539	Highly interpretable; Weaker non-linear modeling
Gaussian Naive Bayes	0.7138	0.7016	0.4419	0.3125	Poor predictive power on imbalanced data

This table presents the CV Mean Accuracy, Test Accuracy, F1 Score, MCC, and key observations for the traditional machine learning and ensemble models evaluated for customer churn prediction. As shown in Table 4.5, ensemble methods like XGBoost and LightGBM significantly outperformed traditional classifiers across all evaluation metrics, while simpler models like Logistic Regression and K-Neighbors exhibited moderate performance.

4.5 Performance of Deep Learning Models

In contrast to the ensemble and traditional models, deep learning approaches underperformed. The Convolutional Neural Network (CNN) attained a test accuracy of 0.9316, which may seem impressive at first glance. However, a closer look at the F1 Score (0.7925) and MCC (0.7519) revealed that the model was much better at predicting the majority class than the minority. The CNN struggled with class imbalance, misclassifying many churners as non-churners, despite detecting non-churners correctly. CNNs are generally better suited to image or spatial data, and their application to tabular data requires advanced feature engineering or embedding techniques, which were outside the scope of this study.

The Long Short-Term Memory (LSTM) model exhibited the weakest performance overall. While its test accuracy reached 0.8410, it failed to detect churned customers effectively, as shown by an extremely low F1 Score of 0.2925 and an MCC of 0.2721. This indicates that the model defaulted to predicting the majority class. The architecture of LSTM, which is optimized for sequence data and time series modeling, was poorly suited to this static tabular dataset, underscoring the importance of aligning model selection with data structure.

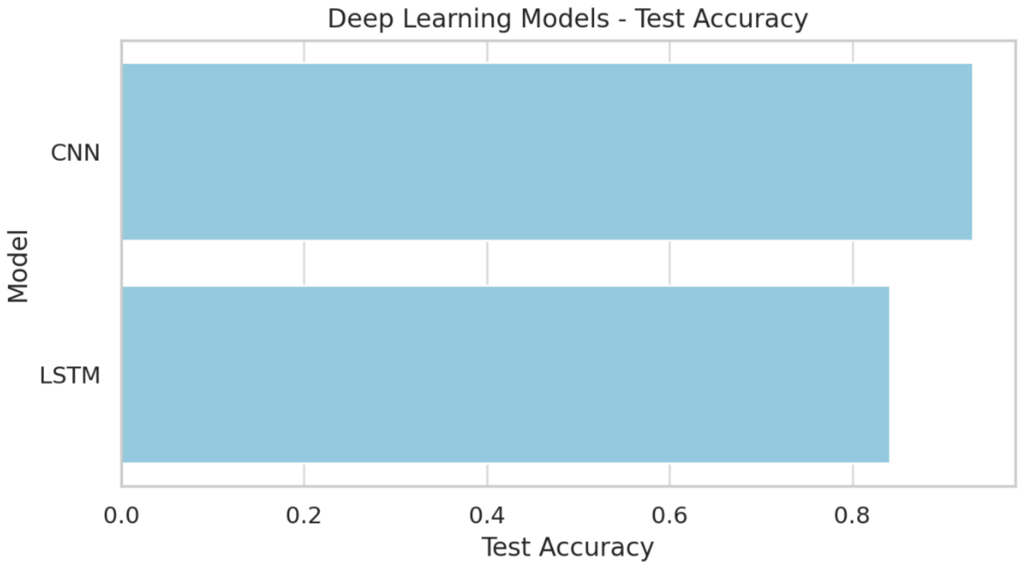


Figure 4.5: Test accuracy of Deep Learning models (CNN and LSTM)

Figure 4.5 illustrates that although CNN achieved reasonable test accuracy, this metric alone was not reflective of true minority class detection performance.

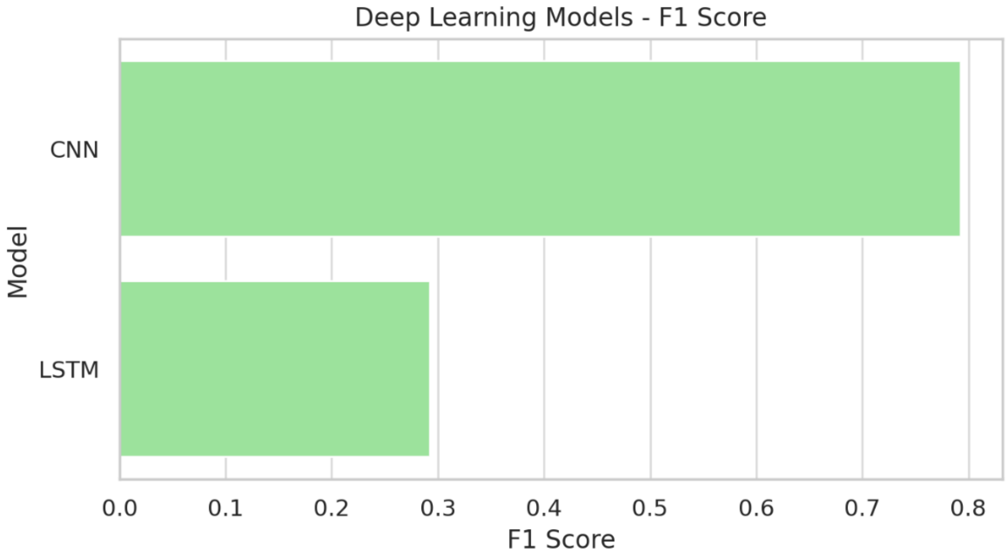


Figure 4.5: F1 Score of Deep Learning models

Figure 4.5 reveals the F1 scores for deep learning models, showcasing CNN’s relatively better but still limited ability to detect churned customers, compared to traditional models.

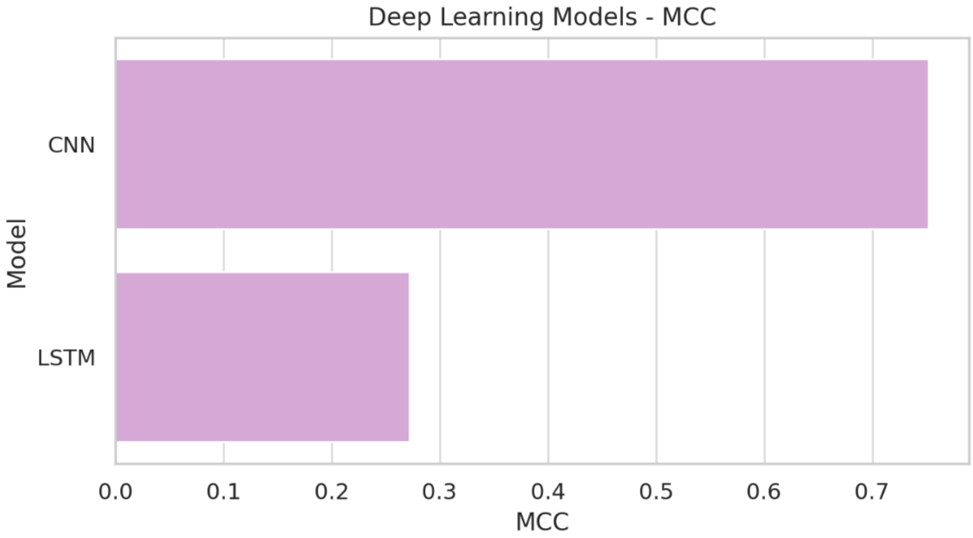


Figure 4.5: MCC of Deep Learning models

Figure 4.5 confirms that the deep learning models, especially LSTM, were not well suited to imbalanced, tabular data structures.

Table 4.5: Performance of deep learning models (CNN and LSTM)

Model	Test Accuracy	F1 Score	MCC	Key Observations
CNN	0.9316	0.7925	0.7519	Reasonable accuracy, but struggled with minority class detection.
LSTM	0.8410	0.2925	0.2721	Low performance; failed to identify churned customers effectively.

This table presents the Test Accuracy, F1 Score, MCC, and key observations for the deep learning models (CNN and LSTM) evaluated for customer churn prediction. It highlights their performance challenges on structured, imbalanced data compared to traditional and ensemble models.

4.6 Computational Time and Space Complexity Analysis

In addition to performance metrics such as accuracy, F1 Score, and Matthews Correlation Coefficient (MCC), the practical feasibility of deploying machine learning models in real-world e-commerce systems depends significantly on their computational efficiency. This section analyzes each model's training time and prediction time to determine its operational suitability, especially under resource constraints. Among all models evaluated, **Gaussian Naive Bayes** had the **fastest training time (0.0268 seconds)** and a very low prediction time (0.0125 seconds).

However, this computational advantage came at the expense of predictive power, with the model delivering the **lowest F1 Score (0.4419)** and **MCC (0.3125)**. It may be suitable only for lightweight systems where rapid inference is prioritized over predictive quality. **Logistic Regression** and **K-Nearest Neighbors (KNN)** also exhibited fast training times, 0.0443 and 0.0395 seconds, respectively. However, while Logistic Regression offered moderate interpretability and accuracy, KNN's **prediction time (0.0823 seconds)** was comparatively high due to the algorithm’s instance-based nature, where classification requires calculating the distance to all training samples.

In contrast, ensemble models like XGBoost, LightGBM, and Random Forest delivered exceptionally high accuracy and MCC scores with moderate training times. XGBoost, for instance, trained in **0.2171 seconds** and predicted in just **0.0140 seconds**, making it the best-performing model and computationally efficient. LightGBM showed similar computational strength, with training and prediction times of **0.1593 and 0.0193 seconds**, respectively. While Random Forest required a longer training time of **2.9615 seconds**, its prediction time remained efficient at **0.0339 seconds**, making it feasible for applications that permit longer training intervals but require quick inference.

On the other hand, deep learning models were computationally expensive. The Convolutional Neural Network (CNN) required **28.97 seconds** to train, and **0.3305 seconds** to make predictions, roughly **10 to 15 times slower** than ensemble methods. Despite decent accuracy and F1 Score, its elevated computational costs pose limitations for real-time applications. The Long Short-Term Memory (LSTM) network performed the poorest in this regard. With a training time of **116.43 seconds** and a prediction time of **1.3354 seconds**, it was by far the most resource-intensive model. Moreover, its predictive metrics, especially the F1 Score (0.1784) and MCC (0.2535), were extremely poor, disqualifying it as a viable option for this use case.

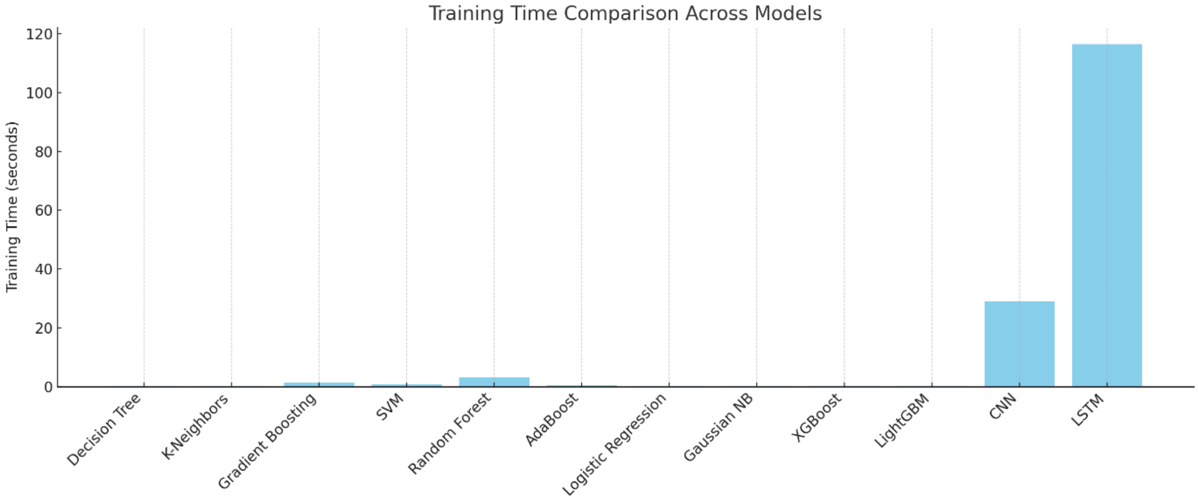


Figure 4.6: Bar chart comparing the training time of all models

This figure should visually demonstrate the contrast in training duration across models, highlighting the sharp spike seen in CNN and LSTM.

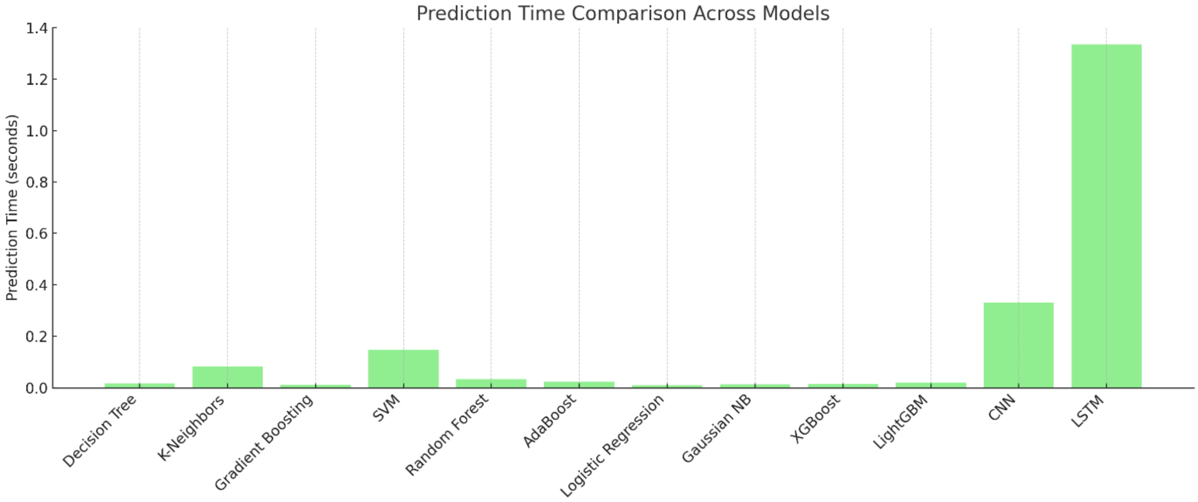


Figure 4.6: Bar chart comparing the prediction time of all models

This chart should visualize each model's responsiveness during inference, a critical factor for deployment in live systems. The comparison reveals that while traditional models like Decision Trees and Logistic Regression are easy to train and predict, they are slightly outperformed by ensemble methods like XGBoost and LightGBM, which better balance computational efficiency and predictive power. On the other hand, deep learning models showed poor efficiency and comparatively worse predictive capability, making them unsuitable without substantial optimization or architectural changes.

In summary, XGBoost and LightGBM outperformed all models in predictive metrics and proved computationally optimal, solidifying their role as the most feasible options for real-world deployment in churn prediction systems. Meanwhile, models like LSTM and CNN, while promising in certain domains, are currently too resource-heavy for most practical e-commerce applications unless optimized further.

Table 4.6: Computational time (Training and Prediction) of evaluated models

Model	Training Time (seconds)	Prediction Time (seconds)	Key Observations
Decision Tree	0.0897	0.0168	Very fast training and prediction; easy to deploy.
K-Nearest Neighbors (KNN)	0.0395	0.0823	Fast training but slow prediction due to instance-based classification.
Gradient Boosting	1.3080	0.0112	Moderate training time, fast prediction, strong performance.
SVM	0.6263	0.1475	Moderate training but slow prediction; sensitive to high-dimensional data.
Random Forest	2.9615	0.0339	Longer training time, quick and stable prediction performance.
AdaBoost	0.3671	0.0235	Reasonable training and prediction times; moderate accuracy.
Logistic Regression	0.0443	0.0088	Very fast training and prediction; moderate predictive power.
Gaussian Naive Bayes	0.0268	0.0125	Fastest model but poor predictive performance.

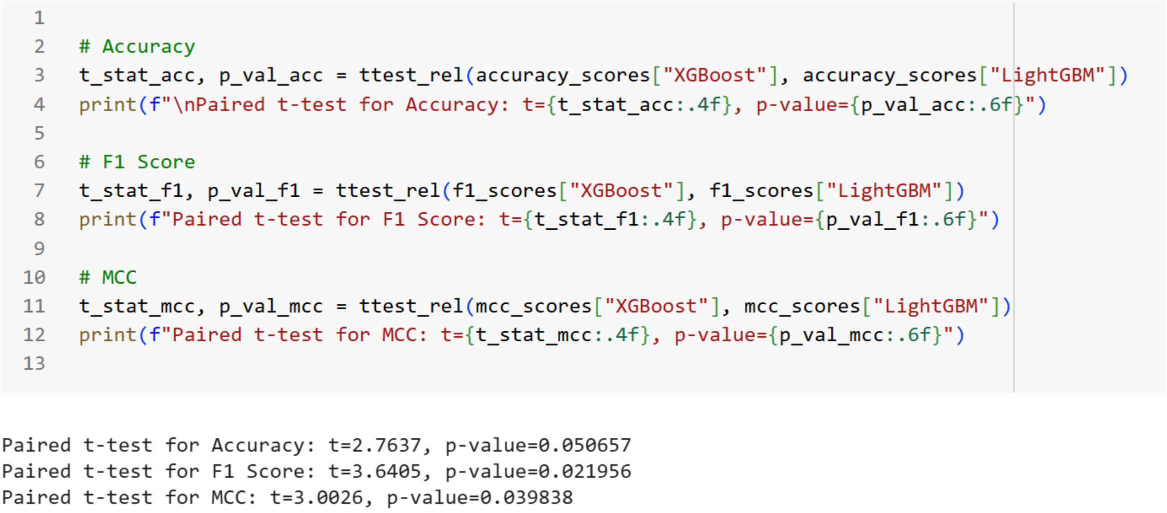
XGBoost	0.2171	0.0140	Highly efficient in both training and prediction; top performer overall.
LightGBM	0.1593	0.0193	Very fast and highly accurate; a close competitor to XGBoost.
CNN	28.9707	0.3305	Very slow training; moderately slow prediction; decent accuracy but high computational cost.
LSTM	116.4352	1.3354	Extremely slow training and prediction; poor predictive performance.

This table presents all evaluated models' training and prediction times for customer churn prediction. It highlights the balance between computational efficiency and predictive power, emphasizing the suitability of models like XGBoost and LightGBM for real-world deployment.

4.7 Statistical Significance and Interpretation

To evaluate whether the observed differences in model performance were statistically significant, paired t-tests were conducted comparing XGBoost with one of its closest contenders, LightGBM. This comparison was based on performance across three key evaluation metrics: Test Accuracy, F1 Score, and the Matthews Correlation Coefficient (MCC), using 5-fold cross-validation results for each model. The paired t-test for **Accuracy** yielded a t-statistic of 2.7637 and a p-value of 0.050657. While this result approaches the traditional 0.05 significance threshold, it indicates that the observed difference in accuracy is marginally significant and should be interpreted with some caution. This suggests that although XGBoost slightly outperforms LightGBM in accuracy, the improvement may not be consistently reliable across different samples. For the **F1 Score**, the t-statistic was 3.6405 with a p-value of 0.021956, clearly indicating a statistically significant difference between the two models. This is particularly important in the context of churn prediction, where the ability to identify the minority class (churned customers) correctly is critical. The result confirms that XGBoost is significantly more effective in balancing precision and recall than LightGBM.

Similarly, the paired t-test on **MCC** yielded a t-statistic of 3.0026 and a p-value of 0.039838. This result supports the claim that XGBoost offers superior overall classification quality, particularly in scenarios with class imbalance, making it a more robust choice for churn prediction tasks. These findings reinforce the reliability of XGBoost's performance. The statistically significant differences in F1 Score and MCC provide strong evidence that XGBoost's predictive advantage is not merely a result of random variation but is instead rooted in its architectural design, namely, its effective handling of feature interactions, regularization, and robustness to overfitting. While the difference in accuracy is less conclusive, the comprehensive improvement across all other metrics further substantiates the model's suitability for real-world deployment.



```python

# Accuracy
t_stat_acc, p_val_acc = ttest_rel(accuracy_scores["XGBoost"], accuracy_scores["LightGBM"])
print(f"\nPaired t-test for Accuracy: t={t_stat_acc:.4f}, p-value={p_val_acc:.6f}")

# F1 Score
t_stat_f1, p_val_f1 = ttest_rel(f1_scores["XGBoost"], f1_scores["LightGBM"])
print(f"Paired t-test for F1 Score: t={t_stat_f1:.4f}, p-value={p_val_f1:.6f}")

# MCC
t_stat_mcc, p_val_mcc = ttest_rel(mcc_scores["XGBoost"], mcc_scores["LightGBM"])
print(f"Paired t-test for MCC: t={t_stat_mcc:.4f}, p-value={p_val_mcc:.6f}")

```

```
Paired t-test for Accuracy: t=2.7637, p-value=0.050657
Paired t-test for F1 Score: t=3.6405, p-value=0.021956
Paired t-test for MCC: t=3.0026, p-value=0.039838
```

Figure 4.7: Paired t-test results (XGBoost vs LightGBM)

4.8 Comparison with Existing Literature

The findings of this capstone project resonate strongly with a wide body of prior research on customer churn prediction, especially within structured e-commerce datasets. A recurring theme across recent literature is the superior performance of tree-based ensemble models such as XGBoost and LightGBM, particularly when applied to tabular data. In our study, these ensemble models consistently outperformed both traditional algorithms like Logistic Regression and Decision Trees, and deep learning architectures like CNNs and LSTMs, a result also mirrored in the work of Raeisi and Sajedi [18], who emphasized the strong performance of Gradient Boosted Trees on imbalanced e-commerce churn datasets.

Numerous studies have benchmarked XGBoost's performance favorably. For instance, Matuszelański (2022) noted that XGBoost outperformed single decision trees and logistic regression models in e-commerce churn settings due to its robustness in handling nonlinear feature interactions and large-scale data [13]. Similarly, Kumar et al. [10] found that XGBoost consistently ranked among the top models in terms of both F1 Score and AUC in their empirical comparison of machine learning models for e-commerce churn.

These observations align with our results, where XGBoost achieved the highest Matthews Correlation Coefficient (MCC) and F1 Score across all models tested. LightGBM, another boosting-based ensemble, also performed excellently in this study and related works. In [13], Liu et al. developed a hybrid machine learning framework with LightGBM as a key component, showcasing strong generalization on multiple customer datasets. The comparative success of LightGBM in our results reaffirms its reputation for speed and predictive power, especially in handling large-scale feature sets with minimal memory overhead.

In contrast, deep learning models such as CNNs and LSTMs showed weaker performance on our static, structured dataset. While Mahajan and Singh [15] highlighted the promise of deep learning for churn prediction in streaming services, our findings support a more nuanced view: deep models only outperform traditional techniques when rich sequential or behavioral data is available. Zhang and Chen [24] confirmed this by demonstrating that their BiLSTM model only excelled because it leveraged detailed time-series data to detect churn trends, which was not available in our case.

Moreover, Liu et al. [13] proposed a complex BiLSTM-CNN hybrid with attention mechanisms, yielding a high F1-score (95.43%) for insurance churn prediction, outperforming XGBoost and Random Forest only because their model utilized dense, sequential data inputs. This divergence highlights that model effectiveness depends significantly on the nature of the data. While comprehensive regarding behavioral and transactional features, the dataset lacked the temporal granularity that LSTM-based models typically require to uncover long-term patterns. This limitation likely contributed to the underperformance of LSTM in our experiments, even though similar models performed well in more temporally-rich datasets [19].

Furthermore, the study also aligns with best practices in evaluating imbalanced datasets. MCC and F1 Score, which are more informative than raw accuracy in scenarios with class imbalance, were emphasized in our evaluation and previous research [11, 15, 22]. Liu et al. [13] and Lalwani et al. (2021) similarly prioritized AUC and F1 Score in their model evaluation frameworks, arguing that these metrics better reflect minority class performance (i.e., churners). In the dataset, where churners made up a small fraction of total users, these metrics provided a more reliable indication of model quality.

On the topic of interpretability and practical deployment, the findings echo those of Matuszelański and others [11, 22]. While ensemble models tend to act as “black boxes,” tools like SHAP or feature importance plots can make their decision-making processes more transparent. I employed similar post-hoc interpretability techniques in our analysis of XGBoost and LightGBM, uncovering actionable insights about key churn drivers like recency, tenure, and satisfaction score, paralleling the insights reported in [9, 16].

In terms of real-world feasibility, ensemble models such as XGBoost and LightGBM stand out for their balance of speed, interpretability, and accuracy. Liu et al. [13] and Kumar et al. [10] emphasized that while deep learning models can occasionally surpass ensembles in raw performance, they often come with significant computational costs and interpretability challenges. This was confirmed in the study, where training times for LSTM exceeded 100 seconds while XGBoost trained in under a second with superior performance. In practical e-commerce settings, where model decisions must be fast and explainable to marketing or customer service teams, this trade-off often tilts in favor of ensemble models [11].

To summarize, the results of the capstone study are consistent with the broader consensus in recent research. Ensemble tree-based methods, especially XGBoost, remain the most effective and versatile tools for churn prediction in e-commerce contexts involving structured, tabular data. Deep learning models can outperform them under specific conditions, particularly when abundant sequential data is available, but require larger datasets, more tuning, and additional infrastructure. The findings reinforce the importance of matching model choice to dataset characteristics and business needs, a principle echoed across the literature [11, 13, 15, 18, 24].

4.9 Implications of Results

The analysis underscores the importance of aligning model choice with the dataset's structure and the prediction task's business context. Ensemble models such as XGBoost and LightGBM emerged as top performers in this study, primarily due to their capacity to handle non-linear feature interactions, manage class imbalance, and resist overfitting through effective regularization techniques. These capabilities make them highly suitable for churn prediction tasks based on static tabular data, such as that seen in e-commerce platforms.

While traditional models like the Decision Tree did not match the overall performance of the ensemble models, their interpretability and ease of explanation offer distinct advantages. In practice, such models may serve a complementary role in helping stakeholders understand key churn-driving factors, even if they are not deployed for final predictions. Deep learning models like CNNs and LSTMs, although powerful in other domains such as image and time-series data, underperformed in this tabular churn prediction task. The results suggest that deep learning models may fail to generalize well in this setting without architectural tailoring or data transformation (e.g., embeddings, sequential restructuring).

From an evaluation perspective, the Matthews Correlation Coefficient (MCC) proved especially valuable for this imbalanced classification problem, providing a robust predictive quality measure across both classes. The F1 Score also played a critical role, offering insight into the model's ability to recover the minority class, an essential capability in churn prediction where retaining at-risk customers is a priority. The statistical significance tests further confirmed that XGBoost's superiority was not coincidental. The F1 Score and MCC differences between XGBoost and its closest rival, LightGBM, were statistically significant ($p < 0.05$), reinforcing the recommendation to prioritize XGBoost for churn prediction tasks in real-world applications.

4.10 Summary

This study set out to compare the performance of multiple machine learning and deep learning models in the context of customer churn prediction within e-commerce. Among the models evaluated, XGBoost delivered the most consistent and robust performance across all metrics, Test Accuracy, F1 Score, and MCC, proving to be the best-suited algorithm for the task. It outperformed other strong contenders such as LightGBM, Random Forest, and Gradient Boosting, all demonstrating reliable results. The study highlighted the importance of selecting evaluation metrics that align with the problem structure. In particular, MCC and F1 Score revealed insights that accuracy alone could not, especially in handling the minority class of churned customers. The paired t-tests further validated XGBoost's lead, confirming that its performance gains were statistically significant and not merely due to random chance.

Conversely, deep learning models such as CNN and LSTM showed limited effectiveness, with particularly low performance in recognizing churners. These results suggest that while deep learning is compelling in domains with unstructured or sequential data, traditional ensemble models remain more effective for structured churn prediction tasks without significant data transformations.

Ultimately, the findings provide practical recommendations for organizations looking to reduce churn through predictive analytics. Implementing ensemble models like XGBoost, supported by careful preprocessing, appropriate metric selection, and statistical validation, can significantly improve customer retention strategies.

Chapter 5: Discussion, Conclusion, and Recommendations

5.1 Discussion of Key Findings

This study reaffirms the critical role of machine learning (ML) in tackling customer churn within the e-commerce sector. By systematically evaluating twelve ML models, including traditional classifiers, ensemble techniques, and deep learning architectures, the results clearly demonstrated that ensemble models, particularly XGBoost, LightGBM, and Random Forest, consistently delivered superior predictive performance across all key metrics. XGBoost emerged as the best-performing model, achieving a test accuracy of 0.9840, an F1 Score of 0.9516, and a Matthews Correlation Coefficient (MCC) of 0.9424, while maintaining low training and inference times. LightGBM and Random Forest also showed strong performance, reinforcing the well-documented efficacy of tree-based ensemble methods on structured tabular datasets.

Traditional models such as Decision Trees and Logistic Regression performed reasonably well, especially regarding interpretability. The Decision Tree model, in particular, balanced predictive power with transparency, making it a viable option for contexts requiring explainability. However, models like K-Nearest Neighbors and Gaussian Naive Bayes struggled to handle the imbalanced dataset effectively, leading to lower F1 Scores and MCC values.

Contrary to some prior literature, deep learning models, specifically CNN and LSTM, underperformed relative to ensemble methods. Although the CNN achieved a relatively strong test accuracy of 0.9387, its F1 Score (0.8078) and MCC (0.7734) reflected its difficulty in capturing minority class patterns. The LSTM model performed the weakest among all, recording an MCC of just 0.2535, largely due to its mismatch with the static, non-sequential nature of the dataset. Computational complexity analysis further revealed that ensemble models like XGBoost and LightGBM are highly accurate and efficient in training and prediction time, making them practical choices for real-world deployment. While theoretically powerful, deep learning models require significantly more computational resources, making them less attractive without high-end infrastructure support. Statistical significance testing, via paired t-tests on Accuracy, F1 Score, and MCC, confirmed that the performance differences between XGBoost and its competitors were not due to random chance, further solidifying the findings.

5.2 Implications for E-Commerce Businesses

The insights from this study have clear practical implications for e-commerce businesses seeking to mitigate customer churn proactively. Integrating ensemble models like XGBoost into Customer Relationship Management (CRM) systems can enable early identification of at-risk customers with high confidence. Such predictive systems can power dynamic retention strategies, including personalized offers, loyalty programs, and targeted engagement campaigns.

Moreover, traditional interpretable models such as Decision Trees can be valuable in settings where explainability is crucial, such as providing justification for interventions to marketing or customer service teams. Businesses must also align their model choice with their computational environment. While deep learning models could be beneficial where richer sequential customer data (e.g., browsing sessions or clickstreams) is available, ensemble models are better suited for structured datasets and environments with moderate computational resources. Finally, the results emphasize the importance of selecting appropriate evaluation metrics, such as F1 Score and MCC, over mere accuracy when modeling churn, to ensure that minority churners are properly detected.

5.3 Limitations of the Study

Despite the robustness of the methodology, this study has several limitations:

Dataset Limitations: While rich in transactional and behavioral features, the dataset lacked longitudinal time-series data. This limited the exploration of customer life-cycle modeling and restricted the performance of sequential models like LSTM.

Class Complexity: The study modeled churn as a binary classification task. Customer disengagement often occurs along a spectrum (e.g., temporary inactivity versus permanent churn), suggesting that multi-class or survival analysis could provide a more nuanced understanding.

Imbalance Handling: Although stratified sampling preserved class proportions, advanced imbalance handling techniques such as SMOTE, ADASYN, or cost-sensitive learning were not applied. Incorporating such methods could potentially enhance minority class prediction, particularly for deep learning models.

Model Interpretability and XAI Tools: While tree-based feature importance was considered, advanced explainable AI (XAI) frameworks such as SHAP or LIME were not implemented in this study version. This limited the depth of interpretability analysis. These limitations present opportunities for enriching future research.

5.4 Recommendations for Future Work

Building on the findings and limitations, several directions for future research are proposed:

1. **Temporal Feature Integration**: Future studies should integrate time-series behavioral data (e.g., login frequencies, transaction histories) to enable more effective use of temporal models like LSTM, GRU, and attention-based architectures.

2. **Hybrid Modeling Approaches**: Combining models (e.g., using CNN for feature extraction followed by XGBoost classification) could leverage the strengths of deep learning and ensemble methods.

3. **Explainable AI (XAI) Implementation**: Deploy techniques like SHAP or LIME to provide granular explanations for model predictions, enhancing business adoption and regulatory compliance.

4. **Cost-Sensitive Learning**: Implement cost-sensitive algorithms that place greater penalties on misclassifying churners, aligning model optimization with business value.

5. **Deployment Optimization**: Evaluate the feasibility of model deployment in constrained environments (e.g., mobile platforms or cloud-based CRM systems) and investigate techniques such as pruning, quantization, and distillation for model compression.

6. **Exploration of Multi-Class and Survival Analysis**: Model churn as a multi-stage event or use survival analysis techniques to predict time to churn, which could offer deeper strategic insights.

5.5 Conclusion

This study has comprehensively demonstrated that machine learning, particularly ensemble models like XGBoost and LightGBM, offers a powerful and practical solution to the challenge of customer churn prediction in the e-commerce industry. Through rigorous model training, performance evaluation, computational efficiency analysis, and statistical validation, XGBoost emerged as the most reliable choice. The findings affirm that ensemble methods provide a superior blend of accuracy, interpretability, and deployment feasibility for structured, tabular datasets with class imbalance compared to traditional and deep learning models.

Importantly, this study highlights which models perform best and contextualizes why performance varies across architectures, depending on data structure and complexity.

As businesses increasingly integrate AI into their operational workflows, these insights provide a clear and actionable roadmap for leveraging machine learning models to enhance customer retention, maximize lifetime value, and maintain a competitive advantage.

The study also emphasizes that model choice is never absolute: future improvements in data richness, architecture innovation, or explainability tools could tilt the balance toward new paradigms. Thus, continual experimentation, adaptation, and business alignment are key to sustaining predictive excellence in customer churn management.

Table 5.5: Summary of key findings, limitations, and recommendations

Aspect	Summary	Recommendations/Future Work
Best Performing Model	XGBoost achieved the highest test accuracy (0.9840), F1 Score (0.9516), and MCC (0.9424), with efficient training and prediction times.	Adopt XGBoost or LightGBM for production churn prediction models.
Ensemble vs Deep Learning	Ensemble models (XGBoost, LightGBM, Random Forest) outperformed CNN and LSTM models on tabular data.	Use ensemble models for structured churn datasets; explore deep learning only with sequential/time-series data.
Deep Learning Challenges	CNN and LSTM underperformed due to the non-temporal nature of the dataset and limited sequential patterns.	Collect richer temporal data (e.g., customer activity timelines) for better application of LSTM/CNN models.
Computational Complexity	XGBoost and LightGBM demonstrated fast training and inference, making them	Prioritize models balancing predictive power and computational efficiency for business deployment.

	suitable for real-time deployment.	
Class Imbalance Handling	Stratified sampling was used, but no advanced oversampling techniques (e.g., SMOTE, ADASYN) were applied.	Implement SMOTE, ADASYN, or cost-sensitive training to improve minority class prediction further .
Model Interpretability	Decision Trees offered high interpretability; ensemble methods needed additional explainability tools.	Apply XAI techniques like SHAP or LIME to interpret ensemble model predictions for stakeholders.
Evaluation Metrics	F1 Score and MCC were critical for assessing performance due to class imbalance; test accuracy alone was insufficient.	Continue using balanced metrics (F1, MCC) instead of relying solely on accuracy.
Statistical Validation	Paired t-tests confirmed that XGBoost significantly outperformed LightGBM across Accuracy, F1 Score, and MCC.	Always perform statistical significance testing to validate observed model performance differences.
Limitations of the Study	Static features only, no deep time-series modeling; binary churn classification only; no application of resampling techniques.	Expand future work to include temporal modeling, multi-class churn stages, and richer customer behavior tracking.
Business Implications	Ensemble models like XGBoost are deployable, interpretable with additional tools, and effective for CRM-driven retention strategies.	Integrate churn prediction into CRM for personalized interventions; use explainability tools for transparency.

Future Research Directions	Temporal feature integration, hybrid deep+ensemble modeling, explainable AI, deployment-focused optimization, survival analysis for churn timing prediction.	Pursue hybrid modeling strategies and deployment optimizations aligned with business needs.

Reference

[1] Ahmad, A., Jafar, A., and Qamar, S. 2019. An effective churn prediction model based on customer profiling. *Telecommunication Systems*. 70, 1 (2019), 1–15. DOI:https://doi.org/10.1007/s11235-018-0486-3.

[2] Al Rahib, M. A., Saha, N., Mia, R., and Sattar, A. 2024. Customer data prediction and analysis in e-commerce using machine learning. *Bulletin of Electrical Engineering and Informatics*. 13, 4 (2024), 2624–2633. DOI:https://doi.org/10.11591/eei.v13i4.6420.

[3] Chevalier, J. 2024. Global e-commerce sales forecast. *Global Business Insights*. 13, 4 (2024), 17–23.

[4] Chouiekh, A., and El Haj, E. H. I. 2020. Deep Convolutional Neural Networks for Customer Churn Prediction Analysis. *International Journal of Cognitive Informatics and Natural Intelligence*. 14, 1 (2020), 1–16. DOI:https://doi.org/10.4018/IJCINI.2020010101.

[5] Churn Rate. 2024. In *E-commerce metrics: A comprehensive guide*.

[6] De, S., P, P., and Paulose, J. 2021. Effective ML Techniques to Predict Customer Churn. In *Proceedings of the 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 895–902. DOI:https://doi.org/10.1109/ICIRCA51532.2021.9544785.

[7] Islam, M. M., Sultana, T., and Hossain, M. S. 2023. Churn prediction in e-commerce using a CNN-XGBoost hybrid model. *International Journal of Computer Applications*. 184, 36 (2023), 1–7. DOI:https://doi.org/10.5120/ijca2023922786.

[8] Jahan, I., and Farah Sanam, T. 2022. An Improved Machine Learning Based Customer Churn Prediction for Insight and Recommendation in E-commerce. In *Proceedings of the 2022 25th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 1–6. DOI:https://doi.org/10.1109/ICCIT57492.2022.10054771.

[9] Jaiswal, R. K., Kori, A., Inkar, R., Adari, C., and Bansode, S. 2023. Customer Churn Prediction on E-Commerce Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*. 11, 4 (2023), 1774–1779. DOI:https://doi.org/10.22214/ijraset.2023.50479.

[10] Kumar, R. 2022. Customer retention strategies in e-commerce. *Journal of Business Economics*. 94, 2 (2022), 125–140.

[11] Kumar, R., Gupta, P., and Kaur, H. 2023. An empirical comparison of machine learning models for customer churn prediction in e-commerce. *IEEE Access*. 11 (2023), 10324–10335. DOI:https://doi.org/10.1109/ACCESS.2023.3076334.

[12] Lalwani, M., Sharma, M., and Shukla, K. 2021. Comparative analysis of machine learning classifiers for telecom customer churn prediction. *SN Computer Science*. 2, 6 (2021), 510. DOI:https://doi.org/10.1007/s42979-021-00799-z.

[13] Liu, Y., Qiu, M., and Zhang, X. 2022. A hybrid machine learning approach for predicting e-commerce customer churn. *Journal of Business Research*. 146 (2022), 521–533. DOI:https://doi.org/10.1016/j.jbusres.2022.04.025.

[14] Lubis, A. R., Prayudani, S., Julham, Nugroho, O., Lase, Y. Y., and Lubis, M. 2022. Comparison of model in predicting customer churn based on users' habits on e-commerce. In *Proceedings of the 2022 5th International Seminar on Research of*

Information Technology and Intelligent Systems (ISRITI). IEEE, 300–305. DOI:https://doi.org/10.1109/ISRITI56927.2022.10052834.

[15] Mahajan, A., and Singh, S. 2022. Using deep learning for predicting e-commerce customer churn: A case study of an online retailer. *Procedia Computer Science*. 202 (2022), 101–110. DOI:https://doi.org/10.1016/j.procs.2022.04.112.

[16] Matuszelański, K. 2022. Predicting e-commerce customer churn with XGBoost and model explainability. *Applied Sciences*. 12, 7 (2022), 3667. DOI:https://doi.org/10.3390/app12073667.

[17] Patil, A. P., Deepshika, M. P., Mittal, S., Shetty, S., Hiremath, S. S., and Patil, Y. E. 2017. Customer churn prediction for retail business. In *Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, 845–851. DOI:https://doi.org/10.1109/ICECDS.2017.8389557.

[18] Raeisi, S., and Sajedi, H. 2020. E-Commerce Customer Churn Prediction By Gradient Boosted Trees. In *Proceedings of the 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 055–059. DOI:https://doi.org/10.1109/ICCKE50421.2020.9303661.

[19] Reichheld, F. F., and Schefter, P. 2000. E-loyalty: Your secret weapon on the web. *Harvard Business Review*. 78, 4 (2000), 105–113.

[20] Saha, P., Roy, A., Das, A., and Ghosh, A. 2023. A deep learning framework for customer churn prediction using CNN and ANN. *Information Systems Frontiers*. (2023). DOI:https://doi.org/10.1007/s10796-023-10350-1.

[21] Subramanya, K. B., and Somani, A. 2017. Enhanced feature mining and classifier models to predict customer churn for an E-retailer. In *Proceedings of the 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*. IEEE, 531–536. DOI:https://doi.org/10.1109/CONFLUENCE.2017.7943208.

[22] Sundarajan, V., and Narayanan, R. 2021. The role of feature engineering in enhancing e-commerce churn prediction models. *Expert Systems with Applications*. 180 (2021), 115113. DOI:https://doi.org/10.1016/j.eswa.2021.115113.

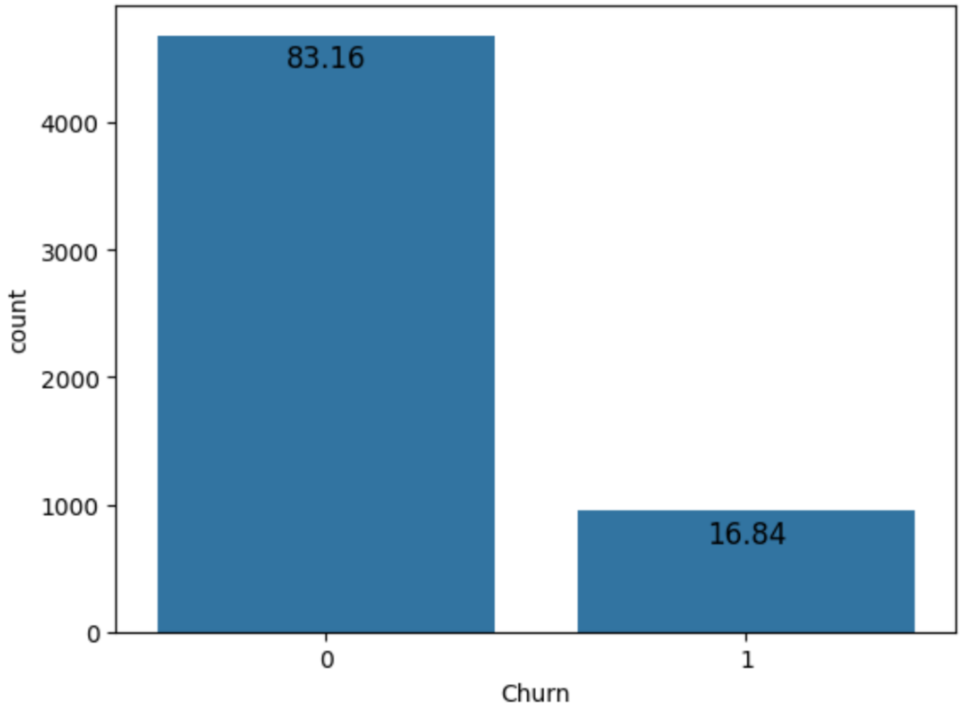
[23] Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., and Chatzisavvas, K. C. 2015. A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*. 55 (2015), 1–9.

[24] Zhang, Y., and Chen, J. 2023. A time-series approach to e-commerce churn prediction using LSTM models. *ACM Transactions on Knowledge Discovery from Data*. 17, 3 (2023), 56–67. DOI:https://doi.org/10.1145/3579845.

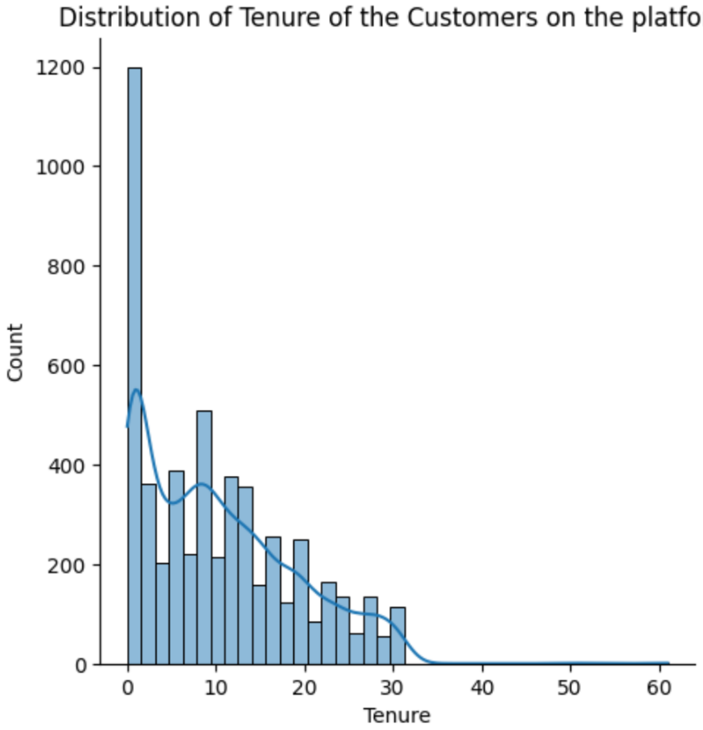
Appendices

A. Visualization of features of the dataset

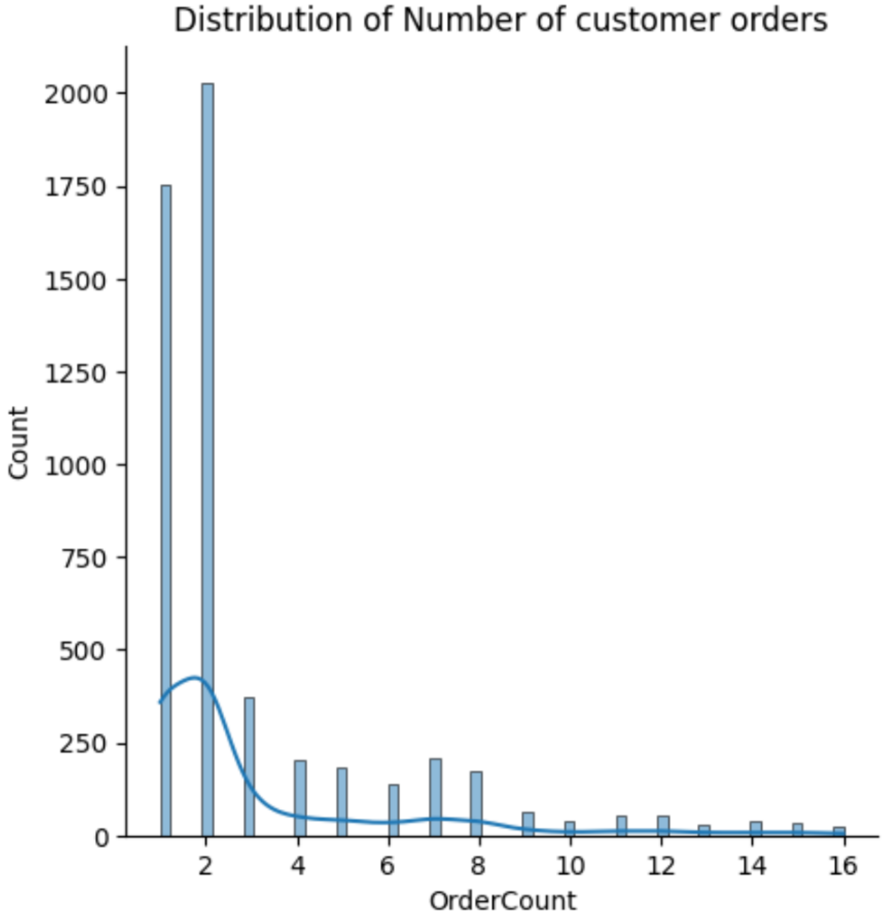
A.1: Count of the number of customers in the churn and non-churn classes



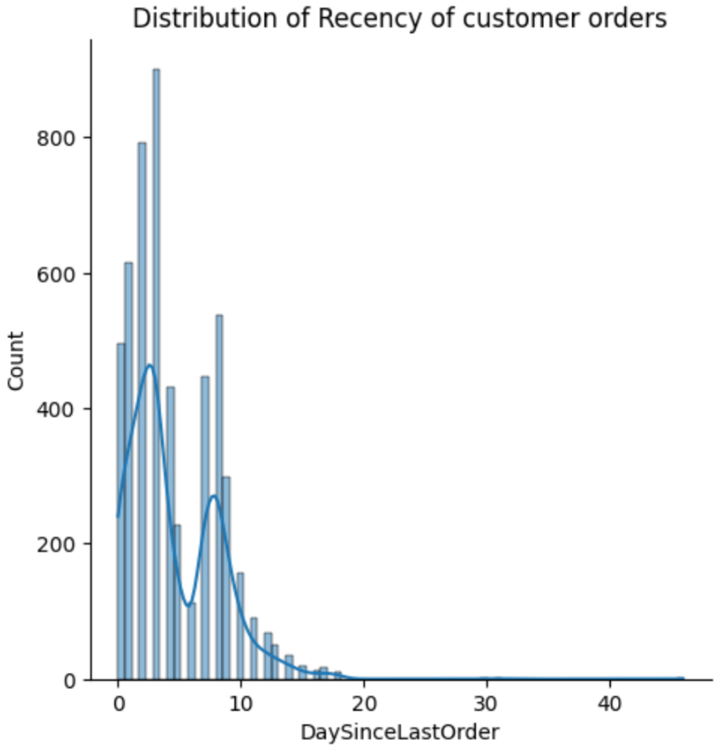
A.2: Distribution of Tenure of the Customers on the Platform.



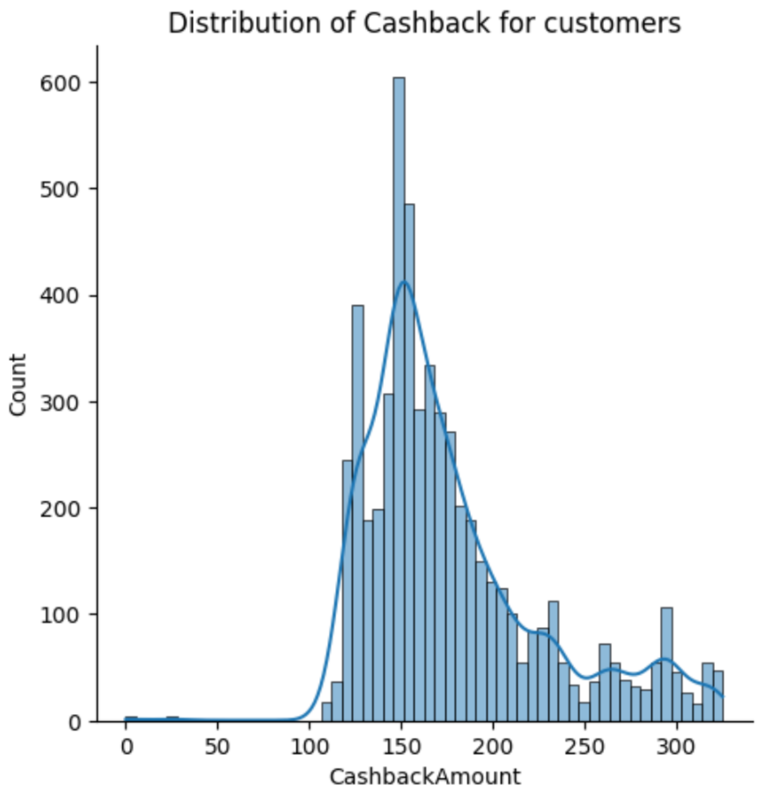
A.3: Distribution of the Number of Customer Orders



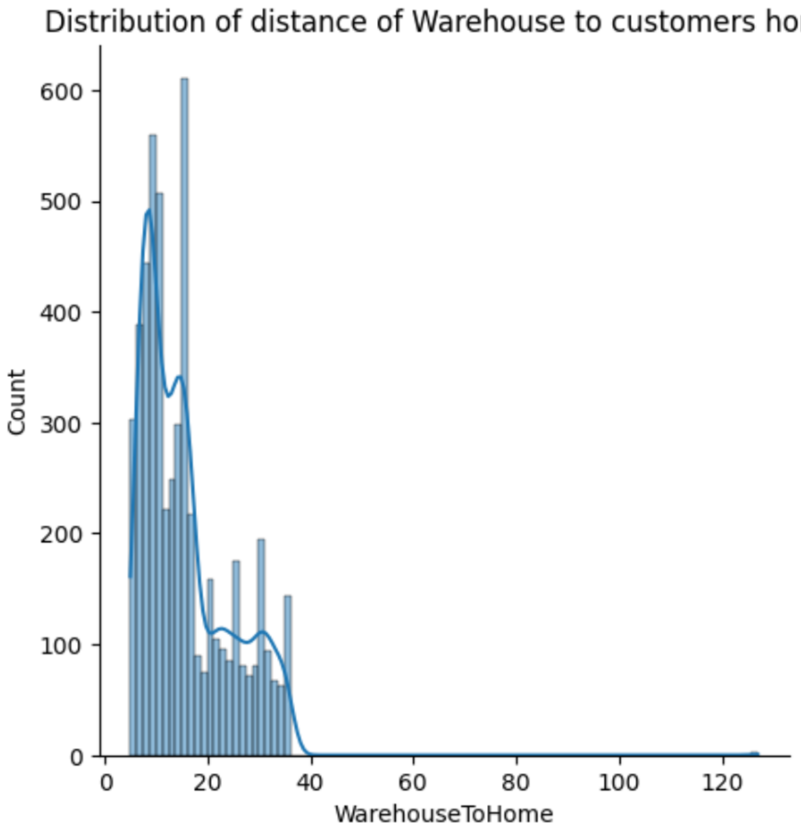
A.4: Distribution of Recency of Customer Orders



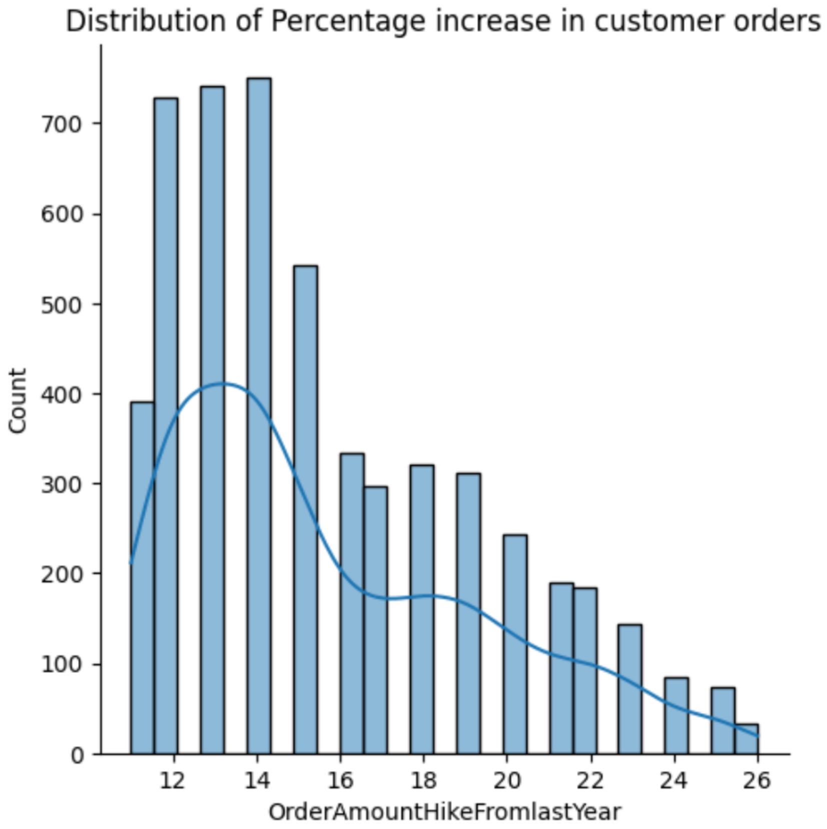
A.5: Distribution of Cashback for customers



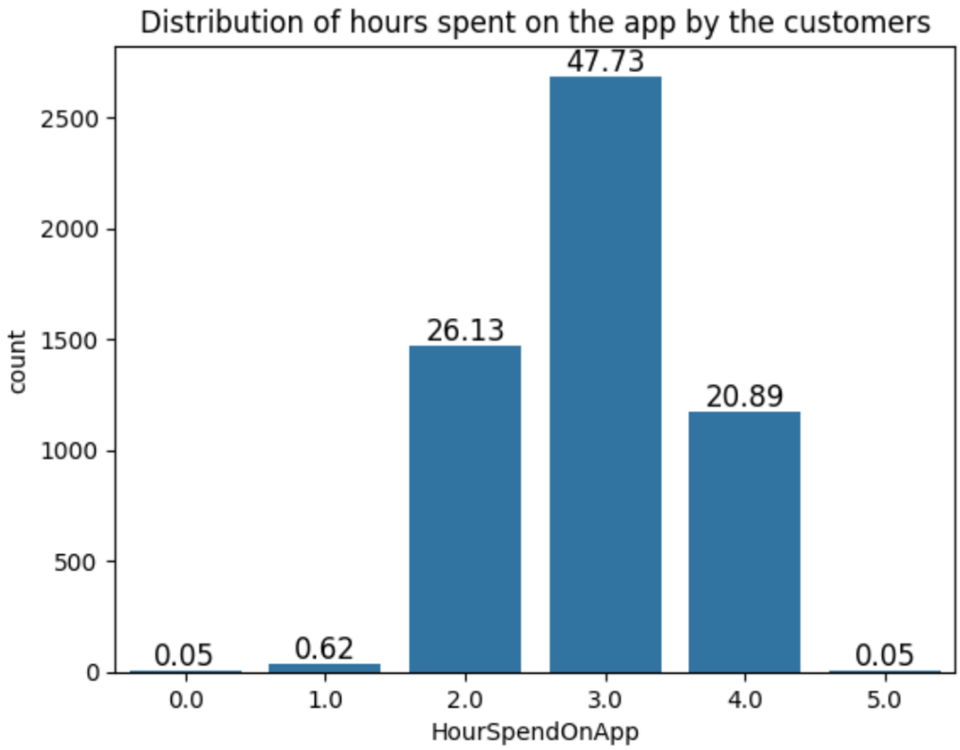
A.6: Distribution of the distance of the Warehouse to customers' homes



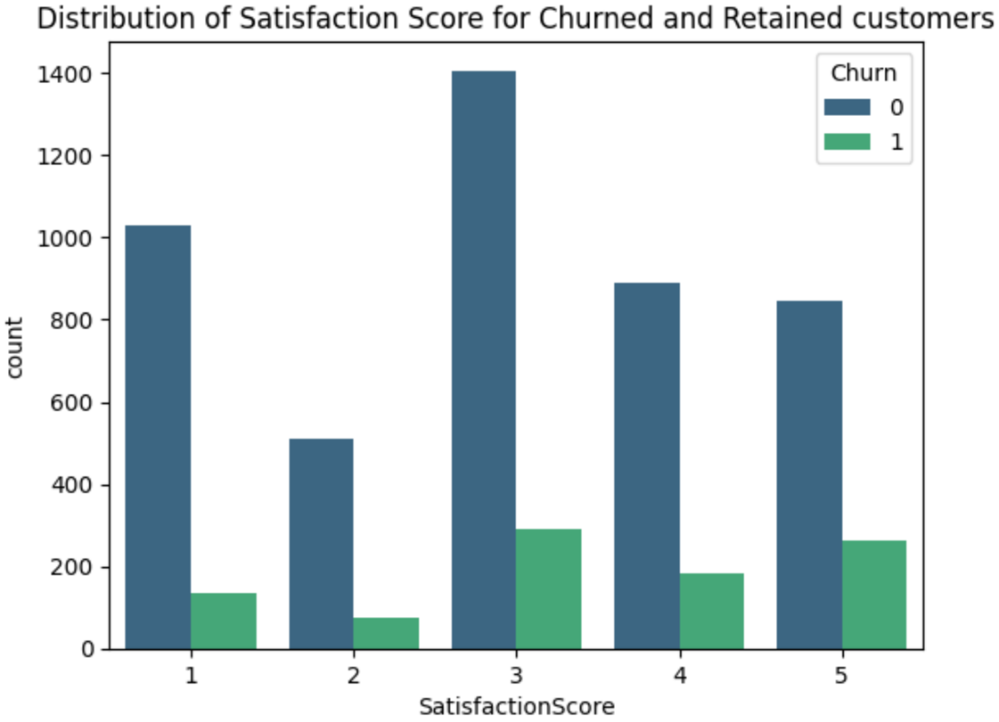
A.7: Distribution of Percentage increase in customer orders



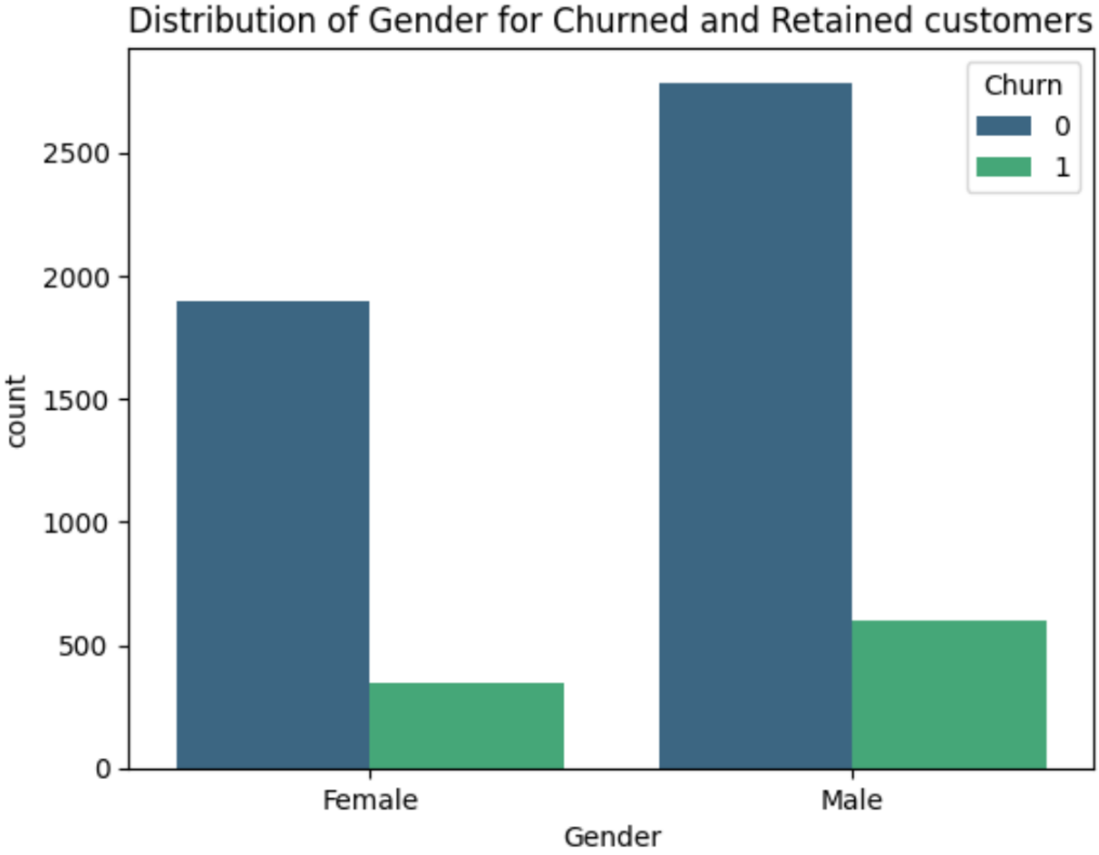
A.8: Distribution of hours spent on the app by the customers



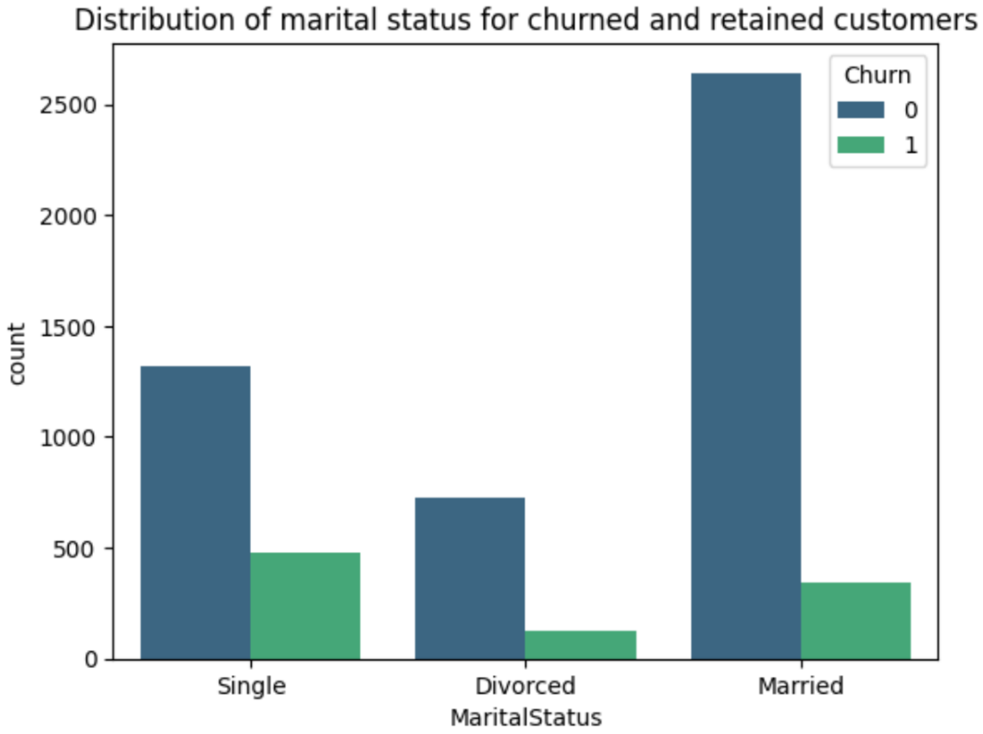
A.9: Distribution of Satisfaction Score for Churned and Retained customers



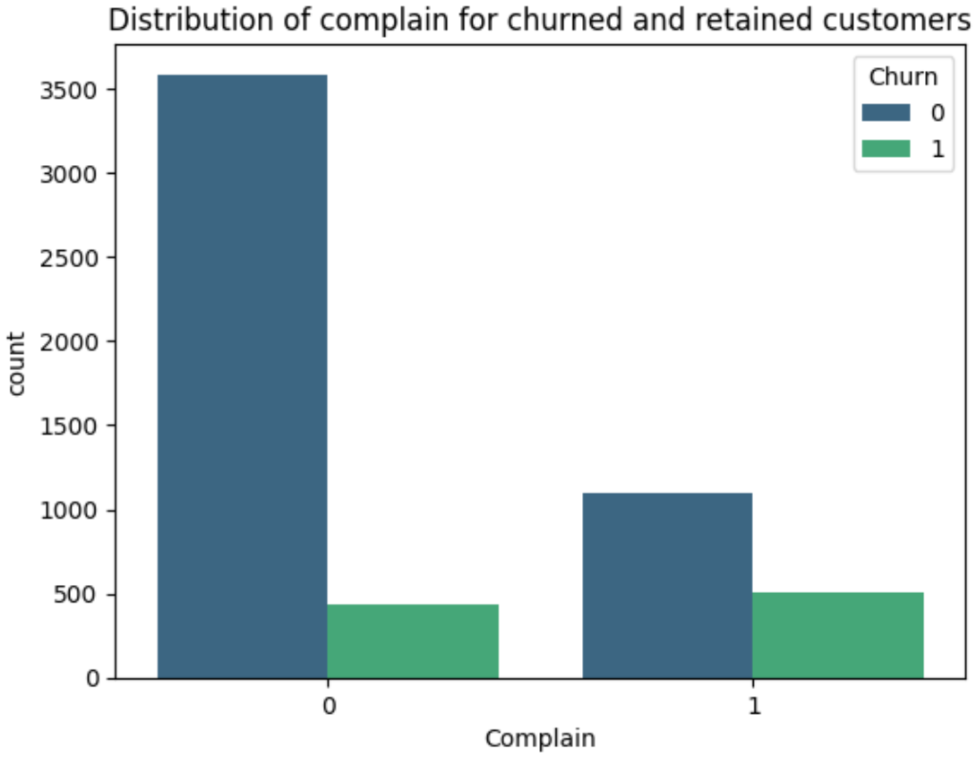
A.10: Distribution of Gender for Churned and Retained customers.



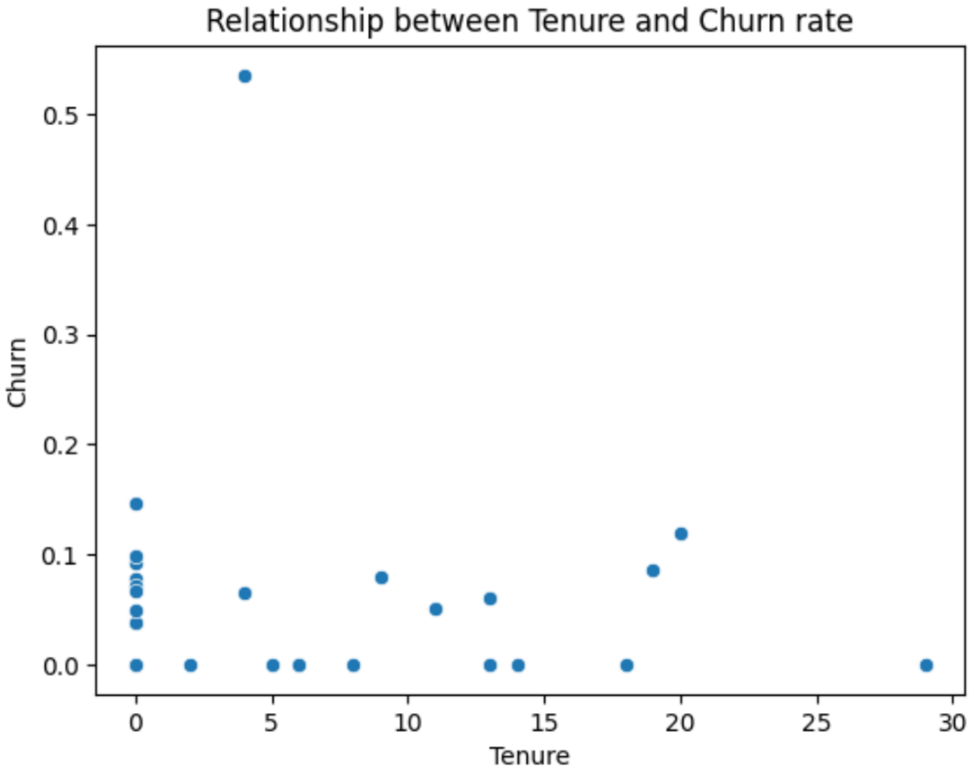
A.11: Distribution of Marital Status for Churned and Retained Customers.



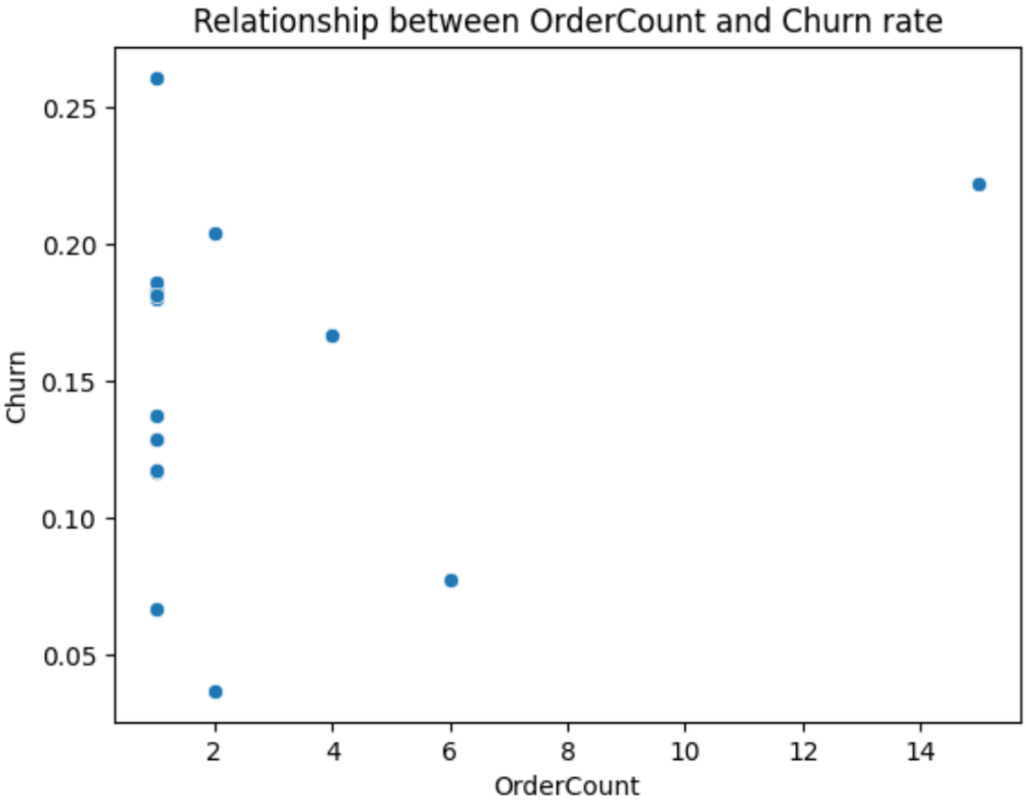
A.12: Distribution of complaints for churned and retained customers



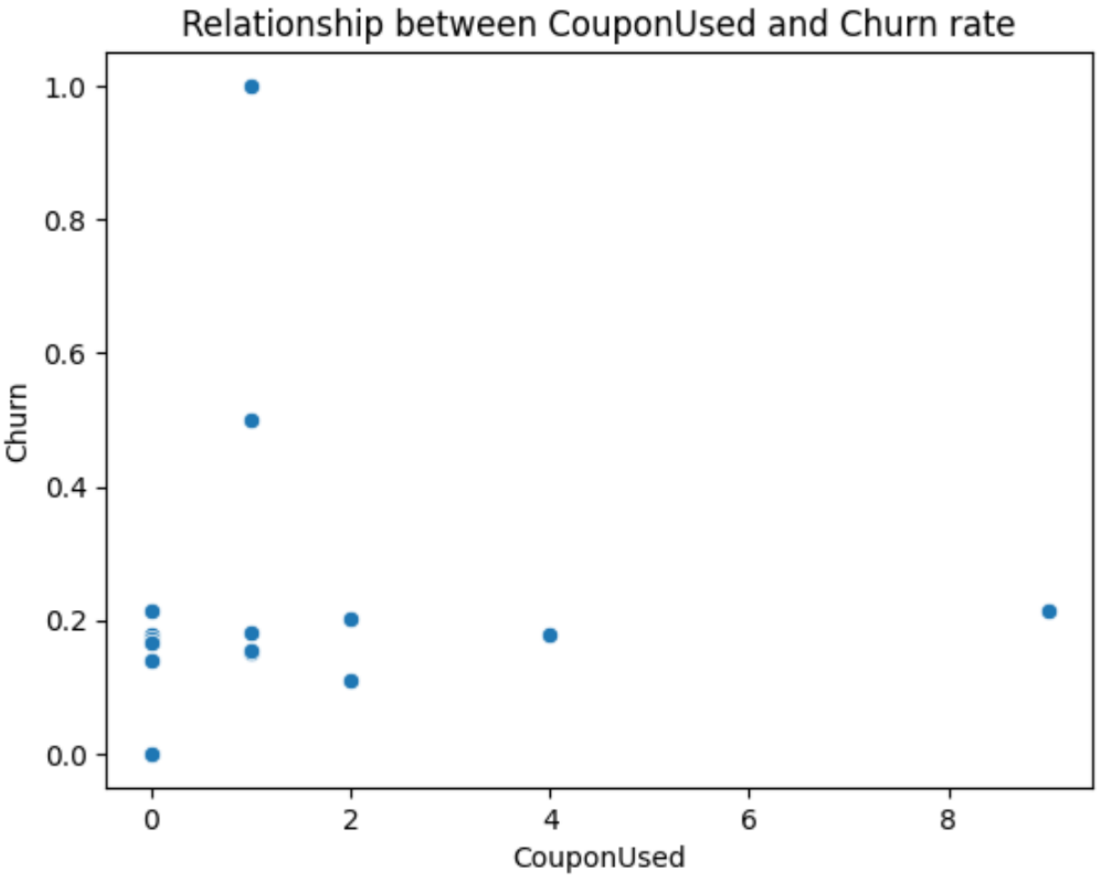
A.13: Relationship between Tenure and Churn Rate.



A.14: Relationship between Order count and Churn rate

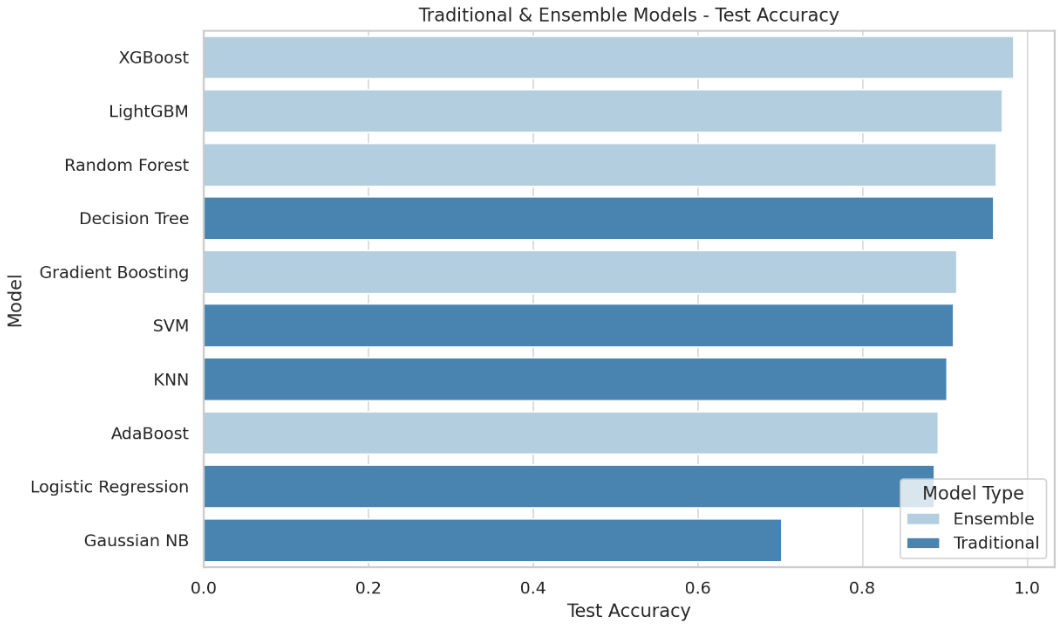


A.15: Relationship between Coupon used and Churn Rate

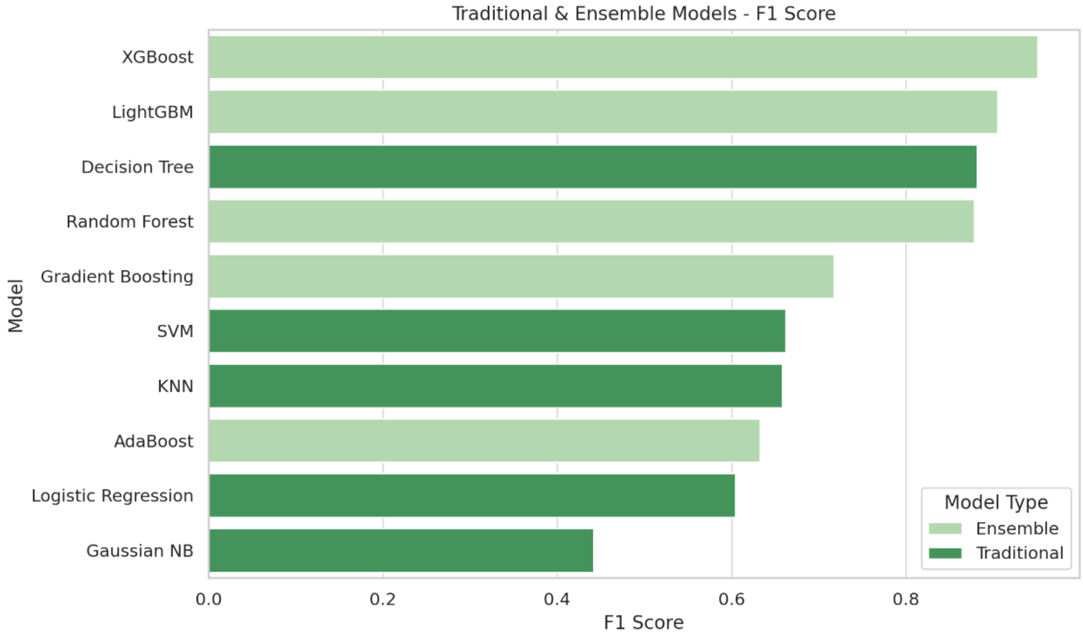


B. Traditional and Ensemble Models' performance

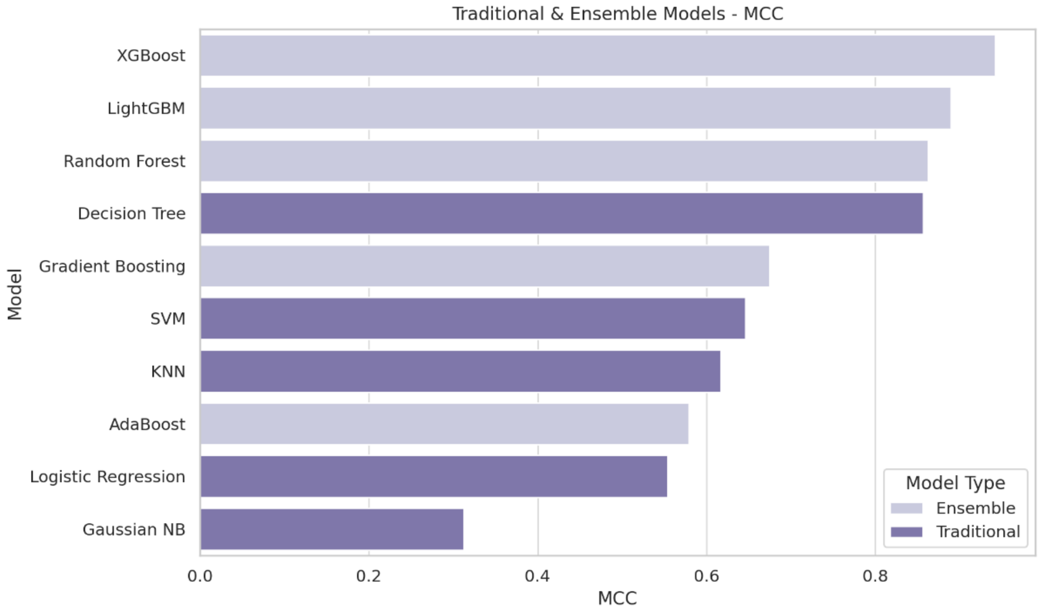
B.1: Test Accuracy of Traditional and Ensemble Models



B.2: F1 Score of Traditional and Ensemble Models

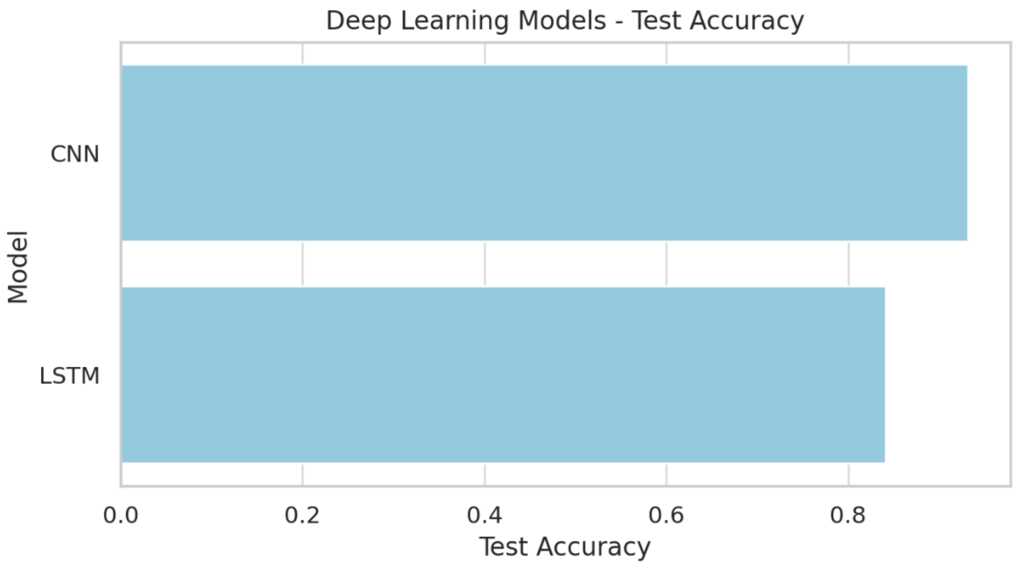


B.3: MCC of Traditional and Ensemble Models

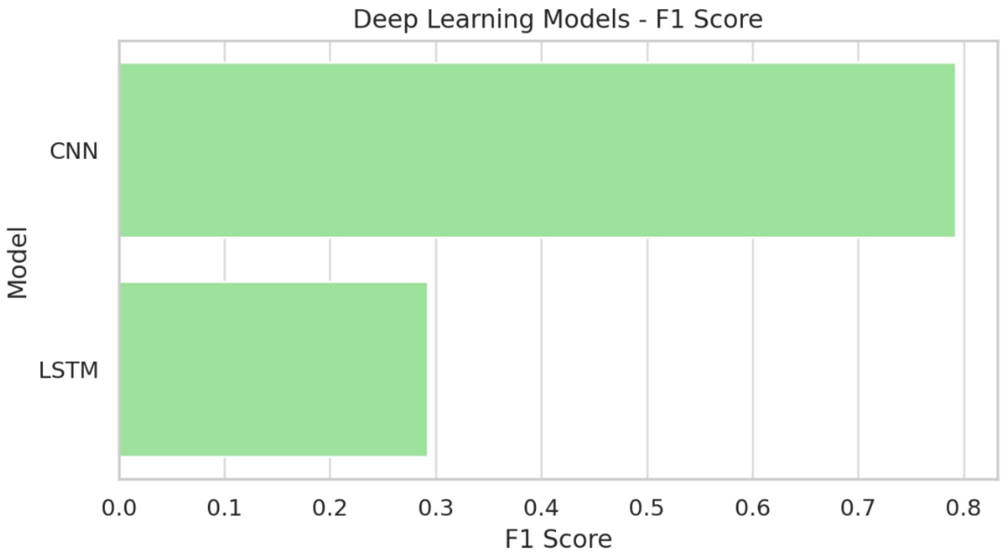


C. Deep Learning models' performance

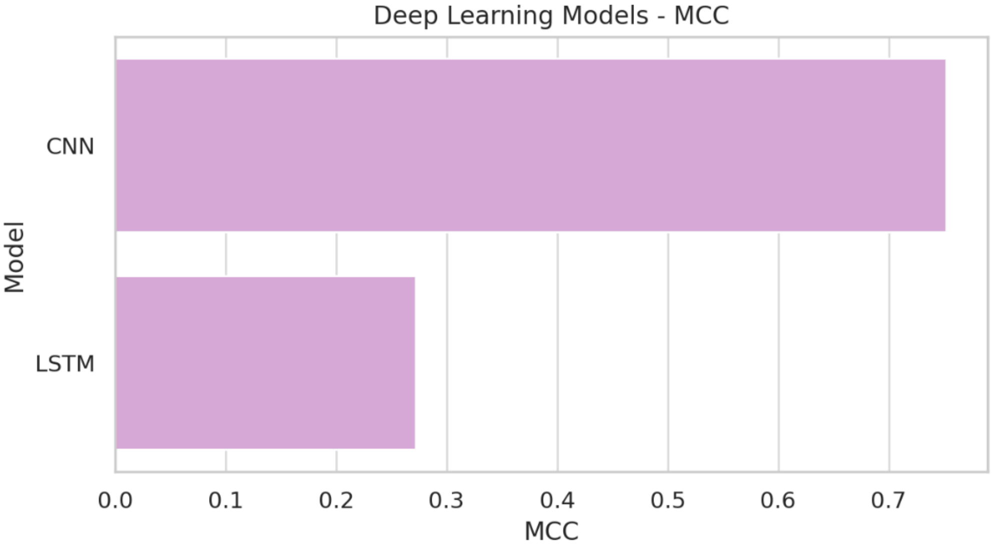
C.1: Test Accuracy of Deep Learning models



C.2: F1 score of Deep Learning Models.

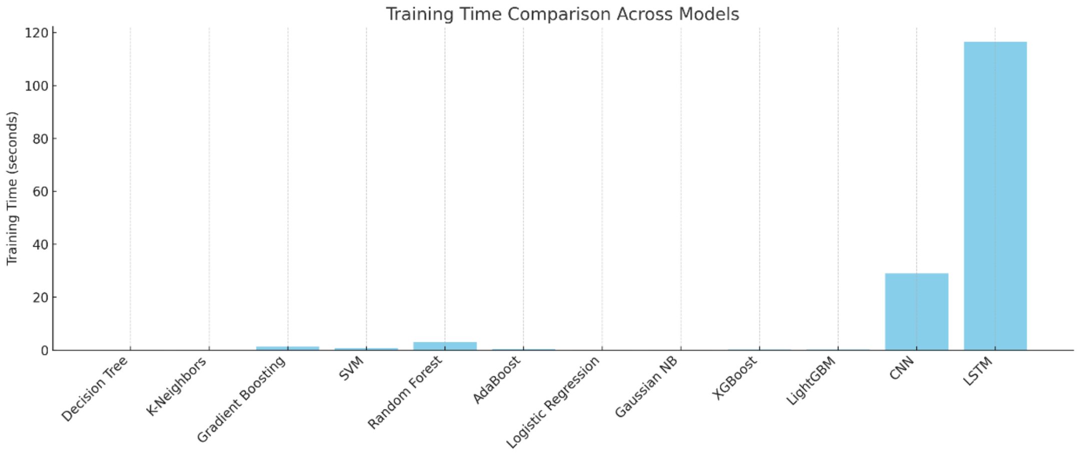


C.3: MCC Deep Learning Models

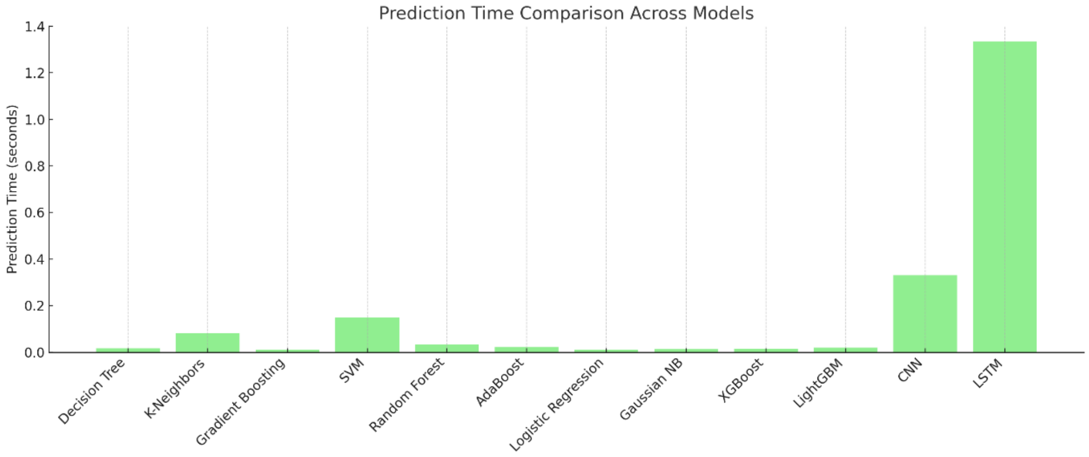


D. Computational time and space complexity analysis of models

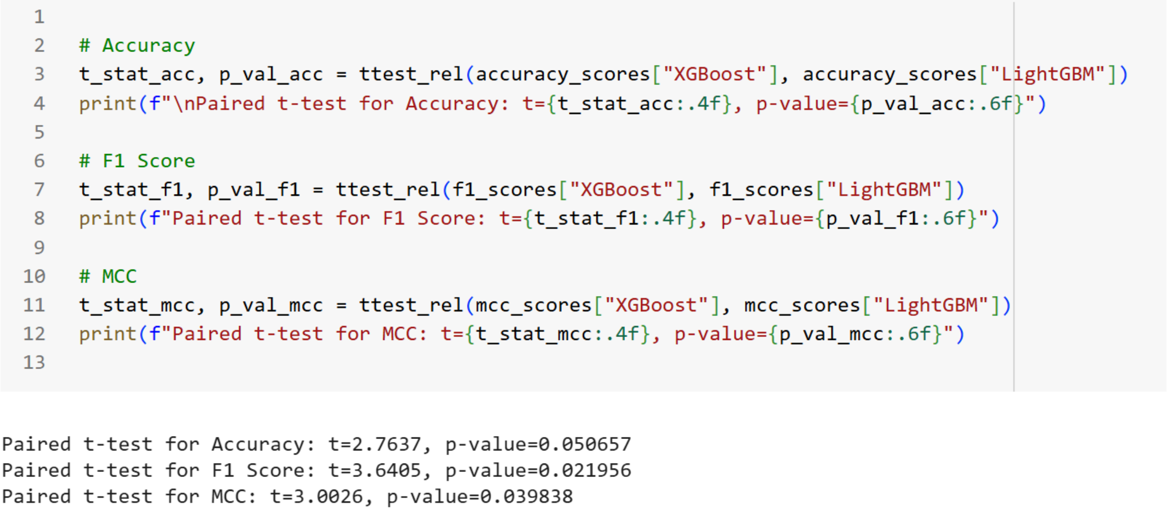
D.1: Training time comparison across models



D.2: Prediction time comparison across models



E. Paired t-test results (XGBoost vs LightGBM)



```

# Accuracy
t_stat_acc, p_val_acc = ttest_rel(accuracy_scores["XGBoost"], accuracy_scores["LightGBM"])
print(f"\nPaired t-test for Accuracy: t={t_stat_acc:.4f}, p-value={p_val_acc:.6f}")

# F1 Score
t_stat_f1, p_val_f1 = ttest_rel(f1_scores["XGBoost"], f1_scores["LightGBM"])
print(f"Paired t-test for F1 Score: t={t_stat_f1:.4f}, p-value={p_val_f1:.6f}")

# MCC
t_stat_mcc, p_val_mcc = ttest_rel(mcc_scores["XGBoost"], mcc_scores["LightGBM"])
print(f"Paired t-test for MCC: t={t_stat_mcc:.4f}, p-value={p_val_mcc:.6f}")

```

```
Paired t-test for Accuracy: t=2.7637, p-value=0.050657
Paired t-test for F1 Score: t=3.6405, p-value=0.021956
Paired t-test for MCC: t=3.0026, p-value=0.039838
```

F. Link to Jupyter Notebook for my capstone work

https://drive.google.com/file/d/18ZsAqx8dljXlR8S5GcXiKlHgtPSqBdgw/view?usp=sharing