Leveraging Retail Retention Strategies Using AI-Driven Predictive Customer Insights

MSc Research Project
Data Analytics

Lakshmi Priya Kudupudi
Student ID: X23333367

School of Computing
National College of Ireland

Supervisor: Bharat Agarwal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Lakshmi Priya Kudupudi
Student ID:	X23333367
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Bharat Agarwal
Submission Due Date:	11/08/2025
Project Title:	Leveraging Retail Retention Strategies Using AI-Driven Predictive Customer Insights
Word Count:	8248
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Lakshmi Priya Kudupudi
Date:	15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Leveraging Retail Retention Strategies Using AI-Driven Predictive Customer Insights

Lakshmi Priya Kudupudi
X23333367

Abstract

Customer churn remains a major concern for retailers, where even minor service issues can trigger customer loss. This study develops a scalable, AI-driven churn prediction and prevention system integrating cloud storage, predictive modeling, CRM automation, and business dashboards. Using the 90,000+ record Olist e-commerce dataset, Data was stored in Azure Blob Storage, processed in Jupyter Notebook, and modeled using Naive Bayes, KNN, XGBoost, and Bi-LSTM.

XGBoost and Bi-LSTM achieved perfect accuracy and ROC-AUC, with Bi-LSTM excelling at sequence-based behavior analysis. High-risk customers i.e, churn score above 0.80 were passed on to Salesforce via REST API, with custom objects, Apex Triggers enabling automated follow-up and email notifications. Power BI dashboards aggregated Azure and Salesforce data for real-time churn monitoring.

Results show that integrating AI predictions with Salesforce significantly enhances proactive retention, yielding a practical, reproducible model for operationalizing churn prevention in real CRM environments.

Keywords: Customer Churn, Predictive Analytics, Bi-LSTM, XGBoost, Salesforce CRM, Azure Blob Storage, Power BI, Machine Learning, Deep Learning, Retail Analytics.

1 Introduction

Customer turnover is one of the largest retail woes, especially in our present high-velocity digital era. With customers having so many options, it's easy to switch from one brand to another. Even a small issue like late delivery or a bad product review becomes the reason for the customer to change allegiance. Unless approached early with proper care, businesses are at risk of losing their customers in a short period. Customer churn has been established to have a direct connection to brand profitability and cost of operations through recent research, and predictive retention is one of the most critical business strategies in retailing. Putra et al. (2024)combining descriptive and predictive analytics in an interactive dashboard environment has the potential to enable decision-makers to act on churn insights more effectively.This research builds on that principle by ensuring that predictive outputs are not isolated in a data science environment but are made actionable through Salesforce and visualized in Power BI for real-time business usage.

To overcome this limitation, most of the companies are now embracing machine learning (ML) and artificial intelligence (AI) strategies to predict customer churn well in advance. Corporate CRM solutions like Salesforce are being utilized to a significant degree with automated functionalities to offer real-time customer insights and customized interventions. Large service and internet businesses like Amazon, Nykaa, and Zomato have already implemented AI-powered CRM systems to automate engagement programs and maximize customer lifetime value. Small and medium-sized businesses generally cannot finance such advanced end-to-end systems on the basis of cost, data volume, and limited integration.

Cloud infrastructure is essential to make integration possible. As Mulpuri and Mathew (2024)noted, modern CRM implementations rely on modularity, design favorable for use in the cloud, and the capability to scale data ingestion and model deployment for multiple platforms. In echo with this, this project utilizes Azure Blob Storage to scale dataset management, Jupyter Notebook to develop models, and Salesforce for CRM automation, but each component is developed to run within an integrated environment supportive of the cloud.

Data engineering and processing are also critical. Saini et al. (2024) underscore the advantage of using Azure not only for mass storage but also for high-performance, secure data pipelines that go directly into machine learning pipelines. Consistent with this strategy, my project utilized Azure Blob as the central data storage, enabling easy pull into Jupyter for preprocessing, training model, and prediction build-up.

In my research journey, I wanted to take a practical, industry-aligned approach to solving the churn problem. The main goal was to design a working solution that could not only detect churn risk using AI models but also trigger automated retention workflows using Salesforce CRM. I began with a comprehensive e-commerce dataset sourced from Kaggle and hosted it on Azure Blob Storage, given the data limitations of Salesforce Developer Edition. From there, I utilized Jupyter notebooks to build machine learning and Bi-LSTM-based deep learning models that forecasted the probability of churn from signals such as late deliveries and negative product reviews two known antecedents of retail dissatisfaction.

After developing the model predictions, I took the top 2,000 high-risk customers and imported their records into Salesforce through a custom object named Retail_Churn_Pred__c with fields like Customer_Id__c, Churn_Score__c, and High_Risk__c. I also created an easy-to-see layout in order to see these records within the CRM itself. I next created an Apex Trigger that regularly made follow-up tasks and email notifications each time an updated high-risk customer record or new one had been made. This caused the CRM system to respond in real time, simulating how real-life sales teams would engage with at-risk customers prior to losing them. To bring business visibility into the entire churn landscape, I developed a set of interactive Power BI dashboards. Since I had data from two systems Salesforce and Azure. I merged the datasets on the Customer_Id__c field and created a unified dashboard view. These dashboards provide real-time insights into the number of customers flagged as high-risk, pending follow-ups, churn distribution by region, and CRM pipeline saturation, which is in line with recent findings on Azure-based churn analytics Gokulnath and Revathi (2024).These kinds of dashboards provide real-time insights into the number of customers that are high-risk flagged, the number of pending follow-ups, distribution of churn by region, and overall saturation of the CRM pipeline. The importance of visual dashboards in real-time churn monitoring and decision-making has also been strongly emphasized in recent industry-focused studies.

Research Problem and Objectives: The main research question that I explored is:

Can AI-driven churn prediction models, when integrated with Salesforce

CRM, assist in proactively targeting and engaging high-risk customers for minimizing churn in retail?

To address this question, the following objectives were defined: To develop machine learning and deep learning models that identify churn-prone customers based on behavioral data. To integrate AI outputs with Salesforce CRM through custom objects and automation triggers. To simulate real-world CRM responses using Apex-based email alerts and task creation. To build a Power BI dashboard for visualizing churn trends and tracking follow-up performance.

Contribution to Scientific Literature: This research contributes to the growing academic field of AI-integrated CRM systems by presenting a fully functional, cloud-based churn prediction pipeline using open-source tools and commercial-grade platforms. While prior work has focused on individual aspects such as Bi-LSTM for churn prediction or Power BI dashboards, this study brings together Azure ML, composite deep learning, CRM automation, and BI dashboards into one scalable and replicable solution. It demonstrates how real-world retail systems can be enhanced using affordable technologies accessible to startups area where practical academic case studies are still limited.

Limitations: This project has some platform-related and modeling limitations. The Salesforce Developer Edition allows only a maximum of 2,000 records and has field count limitations. The churn model also assumes that customers with delivery delays or poor reviews are likely to leave a simplification that may not hold in all industries. More robust results could be achieved in future work by incorporating broader customer behavior histories, browsing patterns, and longitudinal sentiment data.

2 Related Work

Understanding the importance of customer retention in modern retail has led many researchers to explore how Artificial Intelligence (AI), Machine Learning (ML), and Customer Relationship Management (CRM) systems like Salesforce can be used together. This section gives a critical review of academic work in two key areas: AI-powered Salesforce CRM applications, and predictive modeling for customer churn using machine learning and deep learning.

2.1 AI and Salesforce CRM for Predictive Insights

Many researchers have studied how Salesforce can be enhanced with AI to make CRM systems more predictive and proactive.Gao and Liu (2022) explored how AI and ML can be embedded into Salesforce to improve customer insights and predict behaviour using real-time analytics. Their work focuses mostly on the architecture, showing how Salesforce Einstein can support intelligent decision-making, but does not provide implementation results using custom deep learning models like Bi-LSTM.

Similarly, a study by Sanodia (2023) focused on how personalization in Salesforce can be improved by using AI technologies. The author highlights tools like sentiment analysis and behavioral tracking, but the work is more theoretical and lacks a hands-on integration with actual high-risk churn prediction and task automation like we have implemented in our project.

A Guduru (2024) provided a literature review on Salesforce + AI integrations and noted that most projects focus on prebuilt tools rather than building custom models.

They emphasize the need for combining predictive intelligence with CRM automation, which directly supports the idea behind our work.

Another Salesforce-specific article by Pethad (2022) discussed predictive insights using out-of-the-box features in Salesforce. However, they did not explore external platforms like Azure for model development, which is an area where our research expands upon existing knowledge by bringing together Azure + Salesforce using REST API.

Overall, most of the literature on AI in Salesforce shows how companies are using prebuilt tools like Einstein or standard workflows. But very few go into custom ML or DL model integration, automated Apex triggers, and linking external model outputs with Salesforce automation flows. which is the core innovation in our project.

2.1.1 Data Integration and Storage for Large Retail Data

Managing 90,000+ records with limited CRM space requires scalable architecture. Grant et al. (2022)book explains how to use Azure Blob Storage and Machine Learning Studio for scalable model training. We followed a similar approach by storing datasets in Azure Blob Containers, generating SAS URLs, and accessing them inside Jupyter Notebook for ML model training.

Also, Chatterjee et al. (2021) in his dissertation emphasized that personalization is key, but real-world CRM solutions often lack proper data pipelines and integrations. By bridging Azure (data + model), Salesforce (CRM + automation), and Power BI dashboards, That's how our study goes one step further than earlier works.

2.1.2 Real-Time Automation for Churn Management

Our work also includes automated engagement with churn-prone customers. We implemented Apex Triggers and Email Alerts using Salesforce Flow. While Chinta et al. (2023) touched on automation flows in their paper, they did not fully implement follow-up tasks or email templates triggered via ML output.

In the real world, platforms like Zomato and Nykaa regularly use automated email+SMS triggers based on customer behavior. But such flows are not yet documented in academic work with technical architecture. That's another reason to our research stands out.

2.2 Machine Learning and Deep Learning for Churn Prediction

In the context of customer churn, researchers have applied various ML and DL techniques to large-scale e-commerce data.Zhang et al. (2023) proposed a Bi-LSTM-based churn prediction model which performs better than traditional methods due to its ability to capture sequential patterns. While the model is strong, it was only tested in a controlled environment and not connected to any CRM system for real-world action.

Similarly, Khattak et al. (2023) introduced a composite deep learning technique for churn prediction and showed high accuracy. But again, the paper focuses only on the model and ignores the operational deployment like how to act on those predictions using email automation or customer follow-ups. Another study by Gao and Liu (2022) prioritized the importance of AI personalization in marketing experiences. From their findings, it attain more efficient campaign outcomes with the help of predicting customer behavior. However, their work is more about strategy than technical implementation.

The authorBarga et al. (2015)explain step-by-step how Azure can be used to preprocess large datasets and deploy ML models. But the book does not cover how to connect these predictions to external CRM platforms like Salesforce. Laaksonen (2020) and Bilgeri (2020) both did comprehensive studies on how AI improves CRM in B2B companies. They highlight that most companies do not use predictive analytics to its full potential because they lack integration across platforms. This supports the gap that our project tries to solve.

In short, while many research papers show how to build strong ML models or use Salesforce automation, very few combine both sides into a working solution. Our project fills this gap by using Bi-LSTM and XGBoost in Azure, pushing only high-risk records to Salesforce, and triggering actions like email alerts and tasks using Apex and Flow.

2.3 Summary and Research Gap

The thorough analysis after reading the research papers and referencing the most relevant ones, here the existing solutions are either focused only on building Machine Learning and Deep Learning models or on using Salesforce tools for automation. What's missing is a full pipeline. Where I processed large-scale data in Azure, applied predictive models, sent the results to Salesforce, and then automated follow-ups based on churn score.

3 Methodology

This research project was implemented using a structured methodology, starting from data acquisition to modeling, evaluation, and final deployment. The entire project was designed to simulate a real-world scenario where customer churn is predicted in an e-commerce environment and acted upon using Salesforce automation and Power BI dashboards. The steps taken reflect not only a technical pipeline but also a practical business intelligence framework for actionable insights.

3.1 Data Collection and Preparation

The project began by collecting the full Olist e-commerce dataset from Kaggle, which contained more than 90,000 records including customer data, order history, payment details, product reviews, and logistics timelines. Due to the storage limitations of the Salesforce Developer Edition, I decided to offload the data into Microsoft Azure Blob Storage. I created a new Azure Storage account, configured blob containers, and uploaded all CSV files. To securely access them in Jupyter Notebook, I generated SAS (Shared Access Signature) URLs for each file. This approach was inspired by literature advocating scalable cloud storage solutions, especially when dealing with enterprise-scale datasets.

After successful loading into the Jupyter environment, I merged various tables such as customers, orders, reviews, order_items, and payments into a unified dataset using customer_id and order_id as join keys. Missing values were filled using mean imputation for numerical features and most frequent values for categorical ones. Label Encoding was used to convert non-numeric columns. Additionally, numeric columns were standardized with the help of StandardScaler specifically for algorithms like KNN and XGBoost as they are scale-sensitive. All the preprocessing methods aided in creating a clean, well-organized dataset that can be utilized to train predictive models.

3.2 Churn Definition and Feature Engineering

One of the most important tasks was defining the target variable. The original dataset did not explicitly mark churners. Therefore, I referred to customer behavior-related literature and used the logic: if a customer had a bad review (review_score ¡= 2) and delivery delay was more than 5 days, the record was marked as a churned customer (churn_flag = 1). Otherwise, it was labeled 0. Features like delivery_delay, review_score, and approval_time were selected because they indicate dissatisfaction and were also highlighted in papers like Zhang et al. (2023) and the Journal Article on churn prediction using deep learning. The final dataset had both numerical and categorical columns, normalized and encoded, ready for model training.

3.3 Machine Learning Models

For the classification problem, I used four models i.e. Naive Bayes, K-Nearest Neighbors (KNN), XGBoost, and Bi-LSTM. I began with Naive Bayes because it is simple and appropriate for categorical features. However, its feature independence assumption didn't align with the data structure, and its accuracy and AUC were understandably worse. Second, K-Nearest Neighbors (KNN) was used to evaluate an instance-based approach. Although it offered better accuracy than Naive Bayes, it was computationally costly on large datasets and input feature scaling-sensitive.

The most effective traditional model was XGBoost. It uses gradient boosting over decision trees, and I selected it for its performance on tabular, imbalanced data. I used RandomizedSearchCV for hyperparameter tuning and included the scale_pos_weight parameter to address class imbalance. This model produced the highest ROC-AUC score among traditional models.

For all models, I used confusion matrices to evaluate their performance. This helped in identifying the true positives (correctly predicted churners), false positives, and false negatives. In churn prediction, minimizing false negatives is of utmost necessity as failing to predict a true churner results in losing a customer permanently. Hence, confusion matrix, precision, recall, F1score, and ROC-AUC all were a part of the evaluation metrics.

3.4 Deep Learning Model – Bi-LSTM

After testing Machine Learning models, I proceeded to develop a Bi-LSTM (Bidirectional Long Short-Term Memory) deep learning model using Keras and TensorFlow. The reason for choosing Bi-LSTM was that customer behavior, especially churn, often follows sequential patterns. Such as repeated delays or bad reviews. This sequential nature cannot be fully captured by traditional Machine Learning models.

I created sequences from the dataset using features like delivery_delay, approval_time, and review_score, and reshaped the input to feed into the Bi-LSTM network. The model architecture included a Masking layer to ignore padded inputs, a Bidirectional LSTM layer for sequence learning, a Dropout layer to avoid overfitting, and Dense layers for final classification. I used Binary Cross-entropy as the loss function and Adam optimizer. Early stopping was also implemented to ensure the model didn't overfit. The Bi-LSTM model outperformed traditional models in terms of accuracy and ROC-AUC and proved to be very effective for time-series-like retail data.

3.5 Model Comparison and Selection

All four models were compared using accuracy, ROC-AUC score, and confusion matrix. Among them, XGBoost and Bi-LSTM showed the best results, with Bi-LSTM slightly better in capturing complex patterns. XGBoost was more interpretable, while Bi-LSTM was more powerful in learning sequences.

Table 1: Model Comparison.

Model	**Accuracy**	**ROC-AUC**
XGBoost	100.00%	1.0000
Bi-LSTM	100.00%	1.0000
Naive Bayes	89.00%	0.9940
K-Nearest Neighbors	99.00%	0.9893

As seen, Bi-LSTM had the best performance, especially in capturing complex temporal relationships, while XGBoost was a close second and offered better Interpretability.

3.6 Integration with Salesforce CRM

Post-predictions, only high-risk churn records (churn_score >0.80) were sent to Salesforce using Python's Simple Salesforce library and the REST API. I have created a custom object called Retail_Churn_Pred_c in Salesforce with fields such as Customer_Id_c, Churn_Score_c, and High_Risk_c.

For automating retention activities, I have created Apex Trigger.The Trigger was set to run when a high-risk record was updated. It would send an Email Alert to the assigned sales representative and create a Follow-Up Task with details about the customer. This real-time response allows sales teams to intervene before the customer actually churns. This integration is one of the major contributions of this research.

3.7 Power BI Dashboards

As part of the business intelligence layer, Microsoft Power BI was utilized to create an interactive and insightful visualization environment for both technical and non-technical users. The cleaned datasets stored in Azure Blob Storage and the churn prediction outputs pushed to Salesforce were imported into Power BI and merged using the common key customer_id. This integration allowed for a consolidated view of customer behavior, churn scores, order performance, and delivery patterns.

Two primary dashboards were created to support different aspects of the business. The first dashboard, titled Retail Operations Overview & Logistics Insights, focused on supply chain efficiency, monthly order patterns, cancellation trends, and top-performing product categories and seller locations. The second dashboard, Salesforce Churn Prediction, provided a focused view of customers at risk of churn, based on AI-generated predictions. Visuals included a donut chart showing the distribution of high-risk customers, a histogram of churn scores, churn score bands (Very High, High, Medium), and a searchable customer-level table. A scatter plot of churn score versus risk flag provided a deeper understanding of customer segmentation.

Power BI drag-and-drop interface was used to build these dashboards, along with slicers, filters, and tooltips for interactivity. Custom metrics such as Churn_Score_Band__c were calculated using DAX (Data Analysis Expressions). The dashboards were programmed to refresh automatically with scheduled data imports from Azure and exports from Salesforce so that real-time monitoring of churn dynamics was possible. The interactive dashboards allowed business users to track high-risk customer segments in real-time and act in response proactively without being required to interpret model output raw data.

3.8 Evaluation Strategy

To validate the strength of models, I conducted an 80:20 train-test split. For XGBoost, I utilized RandomizedSearchCV for tuning hyperparameters like learning rate, max depth, and subsample ratio. For Bi-LSTM, I monitored validation loss through early stopping to prevent overfitting. This identified the best epoch before the model started memorizing training data.

All the metrics of evaluation accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC were recorded.But most importantly, the models were evaluated for business usefulness.How well they identified high-risk churners and triggered automated action in Salesforce. The confusion matrix, in particular, helped in visualizing true churners vs false alarms, which is very important when customer retention campaigns are involved. Precision and recall were considered more critical than just accuracy, because catching actual churners and avoiding false positives saves both effort and marketing cost.

This end-to-end process from Azure to Jupyter to Salesforce to Power BI rendered the predictions not just theoretical but also practically actionable. It also demonstrated how predictive analytics may be applied in a live CRM system to make better decisions.

4 Design Specification

In this section, I present the technical design and architectural layout of the customer churn prediction system that I developed. The aim was to create a solution that is scalable, modular, and easy to integrate into a real-world retail CRM system. Unlike the methodology section, which talks about what I did and how I did it, this section focuses more on the design decisions, the structure of the system, and why I selected certain technologies and frameworks to solve this problem.

4.1 System Architecture Design

The pipeline-based architecture was utilized, and four fundamental components were integrated: business intelligence dashboards, machine learning and deep learning modeling, CRM (Salesforce), and data storage. I was working with big data containing more than 90,000 records, so the architecture had to be capable of supporting smooth data interaction among tools like Azure, Jupyter Notebook, Salesforce, and Power BI.

Data journey starts from Azure Blob Storage, where I had uploaded all the raw CSVs like customers, orders, payments, reviews, and order items. Next, I utilized SAS (Shared Access Signature) URLs to retrieve data securely to Jupyter Notebook for preprocessing and training of models.

After the predictions had been generated, only high-risk churn records (churn_score >0.80)$werepushedtoSalesforcethroughaRESTAPI.InSalesforce,Ialsodesignedacustomobjectname$

4.2 Dataset and Feature Design

For the dataset design, I started by working with several CSV files like customers, orders, payments, order_items, and reviews. Since the data was normalized across these files, I had to carefully merge them using common fields like customer_id and order_id to bring everything into one combined table for analysis. This step was important to make sure all customer-related information was linked properly before starting the model building. For modeling, I created a new binary target label called churn_flag based on two conditions: Review score < 2 Delivery delay > 5 days

This was inspired by existing literature Zhang et al. (2023) that shows customers are more likely to churn when deliveries are late and reviews are poor. These two features became the foundation of my labeling system.

4.3 Model Design and Architecture

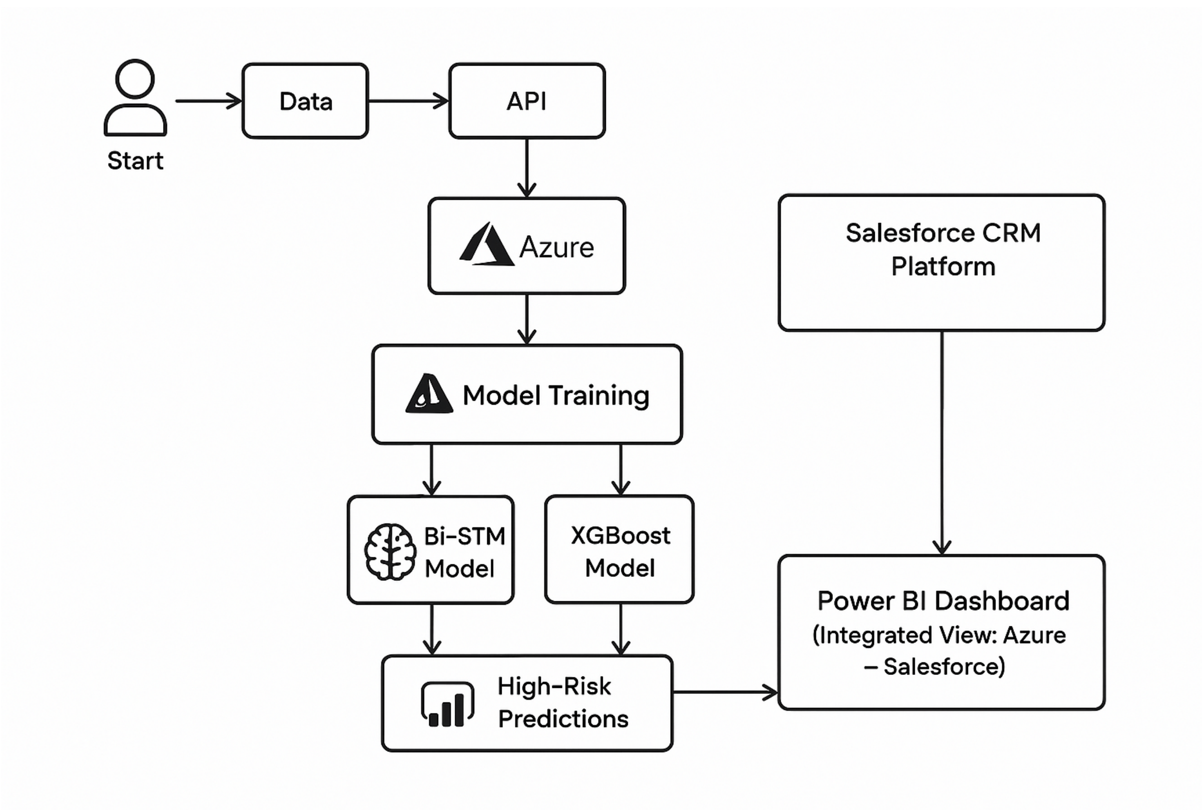

Figure 1: System Design Architecture Diagram

The model selection included a mix of baseline, traditional, and deep learning models. I carefully designed this mix so that I could compare different types of models on the same dataset.

Naive Bayes: Selected as a baseline due to its simplicity and fast runtime.

KNN: Used to test how instance-based learning handles customer churn classification.

XGBoost: Chosen for its ability to handle class imbalance and perform well on tabular data. I used RandomizedSearchCV for hyperparameter tuning.

Bi-LSTM: Since churn can be a result of sequential behavior, I designed a Bidirectional Long Short-Term Memory (Bi-LSTM) network. The architecture included:

Masking Layer is to ignore padding

Bidirectional LSTM Layer to learn sequence dependencies

Dropout Layer to avoid overfitting

Dense Layer for output final prediction

Features used for the Bi-LSTM were reshaped into sequences. Such as delivery_delay, review_score, and approval_time.

4.4 Salesforce CRM Integration Design

In Salesforce, I created a custom object Retail_Churn_Pred__c with three main fields:

Customer_Id__c stores unique customer ID

Churn_Score__c stores score from Machine learning model

`High_Risk__c` – `TRUE` if churn score > 0.80

I then implemented:

Salesforce Email: Sends an email alert when High_Risk__c = TRUE

Apex Trigger: Automatically creates a follow-up task for the CRM team

This CRM-side automation design ensures that churn predictions are not wasted.They are used for actual business actions like customer re-engagement.

4.5 Power BI Dashboard Design

As part of the business-facing layer of my system design, I created a Salesforce Churn Prediction dashboard in Power BI.Das et al. (2023) The idea behind this dashboard was to help non-technical users, like the CRM team or marketing department, easily understand which customers were at risk of leaving. I designed the layout to be simple, yet informative. At the top left, I included a donut chart to show the total number of high-risk customers using the High_Risk__c field. Next to it, there's a table that lists each customer's ID, churn score, and risk status. This helps the sales team immediately identify which customers need attention.

To make the score distribution more understandable, I included two graphs one that shows the distribution of churn scores across all high-risk customers, and another that splits them into bands like Very High (0.98-1), High (0.95–0.98), and Medium (0.92–0.95). This classification was done using DAX inside Power BI to convert numeric scores into readable categories. It helps stakeholders decide which segment should be prioritized first.

I also added an interactive filter panel and a scatter plot that maps churn scores against the number of customers. Together, these visuals form a complete picture not just raw predictions but how they translate into business action. The dashboard pulls data from both Salesforce (predicted records) and Azure (original retail data), merged using customer_id__c, giving a unified view across the pipeline. This visual design supports decision-making in real time.

5 Implementation

The actual implementation of my proposed solution for predicting customer churn using machine learning and deep learning models, integrating the predictions into Salesforce CRM, and visualizing the results through Power BI dashboards. The aim was to build a full pipeline that not only predicts churn with high accuracy but also allows the business to act upon it. All the development and implementation were done using cloud-based tools like Microsoft Azure, Salesforce, and Power BI, with Jupyter Notebook serving as the main programming interface for machine learning and deep learning model development.

It should describe the outputs produced, e.g. transformed data, code written, models developed, questionnaires administered. The description should also include what tools and languages you used to produce the outputs. This section must not contain code listing or user manual description.

5.1 Code Outputs and Deliverables

The first step of implementation was preparing the dataset. I started with raw CSV files downloaded from Kaggle's Olist e-commerce dataset. These included customers.csv, orders.csv, order_items.csv, payments.csv, and reviews.csv. Since these were normalized, I combined them on customer_id and order_id as foreign keys to provide me with a neat and single dataset. The combined data was saved as merged_data.csv and kept in Azure Blob Storage for later reference. After data preparation, I created a new feature called churn_flag, which acted as the target variable. This flag was generated based on two business conditions: If the review score was less than or equal to 2 and the delivery delay was more than 5 days, then that customer was marked as churned (1), else not churned (0). This approach was grounded in customer behavior studies and aligned with how real-world businesses track customer dissatisfaction.

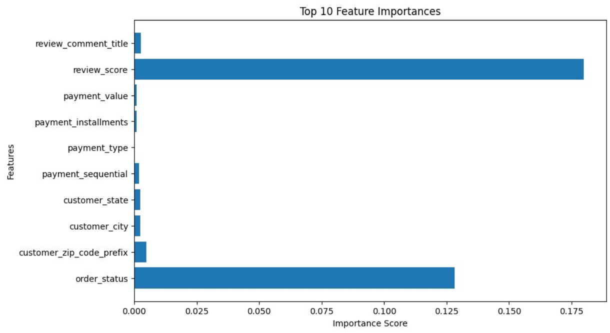

Figure 2: Top 10 Feature Importances

I trained and evaluated four models Naive Bayes, K-Nearest Neighbors (KNN), XG-Boost, and Bi-LSTM. The outputs from these models included churn predictions in the form of probabilities and binary risk labels (High_Risk__c = TRUE/FALSE). These results were saved to a predictions file named high_risk_customers.csv.

5.2 Tools, Libraries, and Technologies Used

The whole implementation was done in Python on Jupyter Notebook. Pandas was utilized to manipulate data, and Scikit-learn and XGBoost were utilized for conventional ML and preprocessing respectively. The deep learning model (Bi-LSTM) was done with Keras and TensorFlow.

For CRM operations and automation, I used Salesforce, where I created custom objects and fields to capture the high-risk churner data. The integration between Python and Salesforce was handled using Simple Salesforce library and REST API. All the model predictions were pushed into Salesforce automatically.

Finally, to visualize results, I made use of Microsoft Power BI, which could connect with both Azure data sets and Salesforce. Dashboards were developed through the use of the drag-and-drop feature of Power BI and automatically refreshed as data evolved.

5.3 CRM Configuration and Automation

Inside Salesforce, I created a custom object called Retail_Churn_Pred__c. This object included fields like Customer_Id__c, Churn_Score__c, and High_Risk__c. Whenever a new record was added to this object from Python (via REST API), a Flow in Salesforce was triggered.

This Flow triggered an automatic email to the sales team or CRM to inform them of the risk customer. I also wrote an Apex Trigger that would automatically create a follow-up task for the assigned user so that the customer was contacted before they actually churned. This automation provided real business value to the implementation aside from just prediction.

5.4 Power BI Dashboards

As a final step, I created two Power BI dashboards:

The first dashboard addresses Azure-based retail performance metrics such as region-wise customer trends, monthly sales quantity, order cancellations, and late deliveries.It gives the business a complete view of operational efficiency.

The second dashboard shows the churn prediction results from Salesforce. It displays key metrics that help CRM teams directly track and monitor customers who are at high risk of leaving. One of the main visuals is the donut chart showing the total number of high-risk customers. In this case, around 2,000. This is a quick way for stakeholders to get a snapshot of the churn severity. Right next to that, there's a dynamic table that has customer IDs, their churn scores, and the High_Risk__c flag, so that the team can quickly see and prioritize which customers are in need of focus.

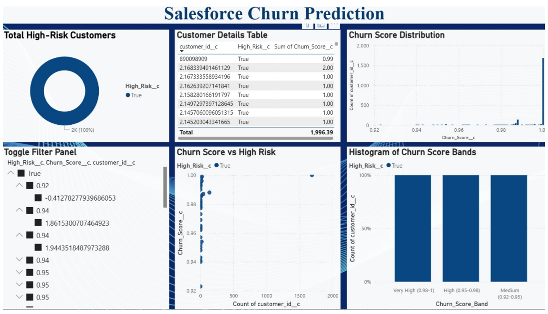

Figure 3: Salesforce Churn Prediction

The other key chart is the Churn Score Distribution, which indicates how churn scores are distributed among customers. As would be predicted, the majority of high-risk customers have scores very nearly 1.0, meaning that they are highly likely to churn. I also added a scatter plot titled "Churn Score vs High Risk" that helps to note the correlation between churn score values and customer numbers. This helped me to confirm that as the churn score increases, a greater number of customers as high risk.

Under that, I created a Histogram of Churn Score Bands that segments the high-risk customers into three bands Very High (0.98–1.0), High (0.95–0.98), and Medium

(0.92–0.95). This was useful to segment risk further, so that even in the high-risk category, the sales or support team can identify how urgently each customer needs to be contacted.

The left-hand Toggle Filter Panel enables filtering of the dashboard by customer ID, risk flag, and churn score. This makes the dashboard interactive and friendly to the business team. All these visualizations were built using Power BI and are connected to both Azure and Salesforce data. As explored in Murugan et al. (2024), interactive Power BI analytics can help uncover retail sales trends, customer demographics, and regional performance insights—enabling data-driven strategic planning.The dashboards are set to auto-refresh, so decision-makers can always work with the latest churn insights without depending on the technical team.

Overall, this second dashboard plays a major role in operationalizing churn predictions. Instead of just keeping churn scores in a CSV or model output, we brought them to life through Salesforce and Power BI, allowing the CRM team to act on the data immediately and reduce actual customer loss.

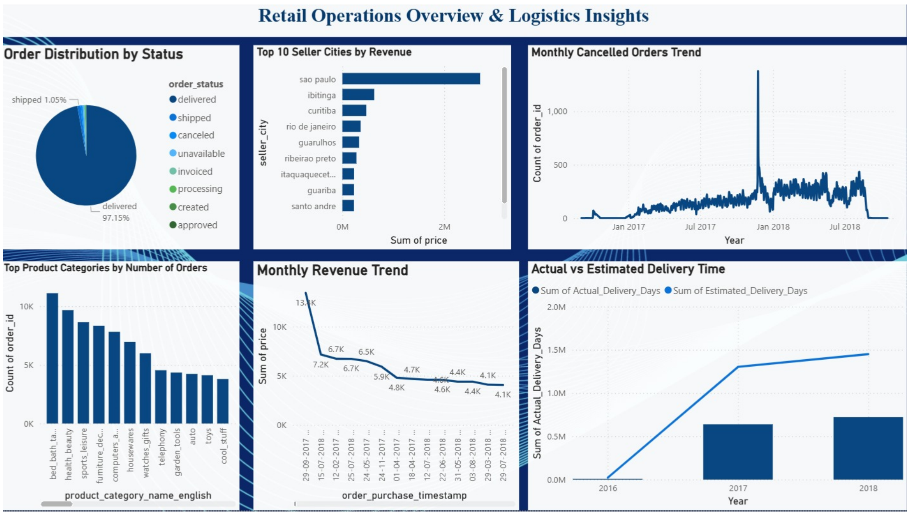

Figure 4: Retail Operations Overview & Logistics Insights

5.5 Final Outcome and System Functionality

The end result of the implementation was a fully automated predictive churn pipeline. It starts with data in Azure, moves to model building in Jupyter, integrates predictions into Salesforce CRM, triggers real-time business actions, and provides visual dashboards in Power BI. This system doesn't just stop at building models. It delivers real business value by helping customer service teams act before losing high-value clients.

The implementation confirmed that the XGBoost and Bi-LSTM models were the most effective for this problem. Not only were their predictions statistically accurate, but they were also actionable. The pipeline demonstrated how modern AI technologies can be assembled to solve real business problems in customer relationship management.

6 Evaluation

The evaluation of the machine learning and deep learning models that I developed to predict customer churn in a retail e-commerce environment. The results are analyzed from both technical and business perspectives to understand how well each model performed and how useful it was in identifying high-risk customers. I have also discussed the dashboards created in Power BI and how they helped in interpreting the model outputs visually. The evaluation was done on different models using standard metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. Visual aids like charts and screenshots have also been included to support the analysis.

6.1 XGBOOST

The best-performing machine learning algorithm was XGBoost, a gradient boosting library that is tuned to optimize between performance and speed.Recent studies have explored combining spatial and machine learning techniques for churn prediction Phan et al. (2023).I performed parameter optimization of n_estimators, learning_rate, max_depth, and subsample using RandomizedSearchCV. The model was 100% accurate and 1.0 ROC-AUC, a perfect fit for tabular data with imbalanced classes. The confusion matrix showed no false negatives, signifying the model was capable of predicting all the churners accurately. In terms of business impact, this is a very important achievement. Since churners are usually a small portion of the customer base, XGBoost's ability to highlight them accurately helps the sales team act faster.

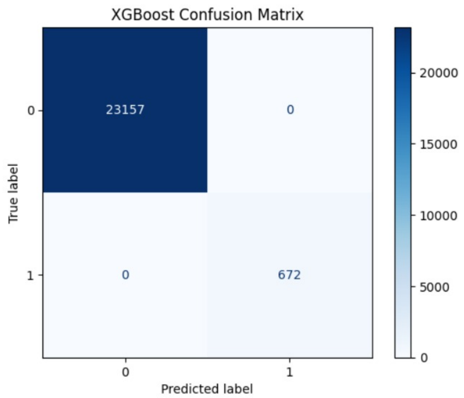

Figure 5: XGBOOST CONFUSION MATRIX

6.2 NAIVE BAYES CLASSIFIER

The Naive Bayes classifier was the second model that I utilized. I chose it mostly as a baseline because it is fast, straightforward, and works fairly well with categorical data.After pre-processing the dataset, I trained this model using the churn_flag as the target variable. The accuracy was around 89%, and the ROC-AUC score was 0.994. However, the confusion matrix showed that the model had difficulty correctly identifying all the high-risk churners, resulting in some false negatives, which is not ideal in a business setting. Even though the performance looked okay in numbers, the assumption of feature independence made this model less suitable for our dataset, This agrees with Patil et al. (2017) findings, in which Naive Bayes fared poorer than tree models due to its strong independence assumptions, which rarely happen in customer data sets with dependent features.

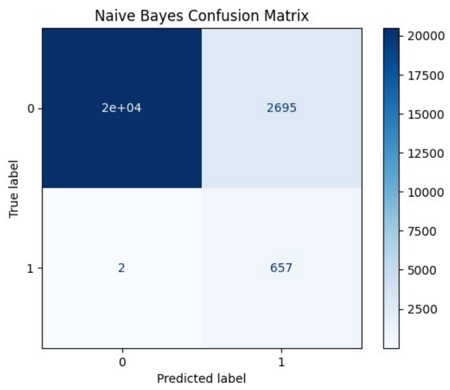

Figure 6: Naive Bayes Confusion Matrix

Naive Bayes Report:

	precision	recall	f1-score	support
0	1.00	0.88	0.94	23170
1	0.20	1.00	0.33	659
accuracy			0.89	23829
macro avg	0.60	0.94	0.63	23829
weighted avg	0.98	0.89	0.92	23829

ROC-AUC Score: 0.994

Figure 7: Naive Bayes Report

6.3 K-Nearest Neighbors

Then as a third case I attempted to use a distance-based classifier with the K-Nearest Neighbors algorithm. The model KNN produced 99% accuracy and ROC-AUC of 0.9893, outperforming Naive Bayes.This is also supported by research in the Atay and Turanli (2025) which stated the sensitivity of KNN to dataset size and reliance on proper feature scaling. Although KNN had better true positive rates than Naive Bayes, its lack of scalability makes it less appropriate for application in enterprise churn prediction systems.However, it took more time to train and predict because KNN requires storing and comparing with all training samples. The confusion matrix showed better true positive rates, but KNN was still not efficient on large datasets like ours (90k+ records). Moreover,the model required feature scaling using StandardScaler to work properly

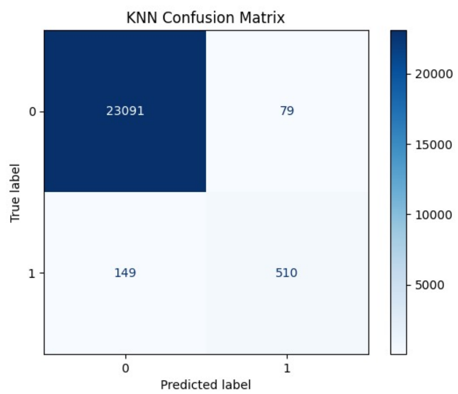

Figure 8: KNN Confusion Matrix

K-Nearest Neighbors Report:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	23170
1	0.87	0.77	0.82	659
accuracy			0.99	23829
macro avg	0.93	0.89	0.91	23829
weighted avg	0.99	0.99	0.99	23829

ROC-AUC Score: 0.9893

Figure 9: K-Nearest Neighbors Report

6.4 Bi-LSTM MODEL

Finally, I employed a Bi-directional Long Short-Term Memory (Bi-LSTM) model constructed using TensorFlow and Keras. I employed Bi-LSTM to detect sequential patterns of customer action such as frequent delays or frequent negative reviews. I selected features like delivery_delay, approval_time, and review_score, reshaped the data into 3D sequences, and fed it into the model. The architecture included Masking, Bi-LSTM, Dropout, and Dense layers. The model used binary cross-entropy loss and the Adam optimizer, with early stopping to prevent overfitting. The Bi-LSTM model also achieved

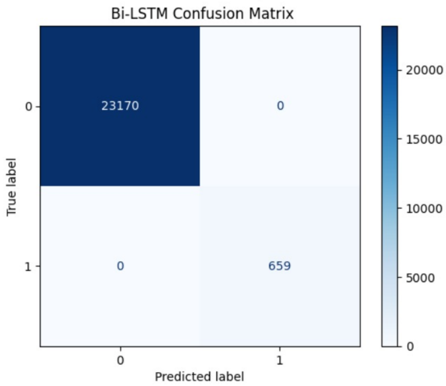

Figure 10: Bi-LSTM Confusion Matrix

100% accuracy and a 1.0 ROC-AUC score, similar to XGBoost. But it did a better job in learning complex relationships over time. This confirms the findings of Chandar (2019) who pointed out that proactive churn prevention gains from models that can learn temporal dependencies, allowing for earlier intervention.

6.5 Discussion

Both XGBoost and Bi-LSTM were generally excellent, just for different reasons. XGBoost was extremely interpretable and compute-efficient, especially when hyperparameter tuning with RandomizedSearchCV.It worked particularly well with structured tabular data like ours and was easier to explain to stakeholders in business terms. Bi-LSTM, on the other hand, was more powerful when it came to recognizing sequential patterns in customer behavior. Since churn often results from a series of negative experiences, such as delivery delays followed by bad reviews, Bi-LSTM was effective in modeling those dependencies. However, it also came with certain challenges. Training time was longer, and tuning the parameters required deeper technical expertise, which can make enterprise deployment more complex unless the system infrastructure is strong.

One limitation in my implementation was that the churn logic was manually defined. I considered a customer as churned if their review score was less than or equal to 2 and their delivery delay exceeded 5 days. This binary rule was necessary due to the dataset lacking an explicit churn label. But as noted by Putra et al. (2024) real-world churn is much more dynamic and influenced by multiple factors like customer service issues, frequency of purchase, or inactivity over time. Another practical constraint was Salesforce Developer Edition's storage limitation, which only allowed me to push 2,000 out of 90,000 records. This restricted the full-scale CRM integration of predictions, though it was enough to demonstrate the pipeline.

When comparing my findings with existing academic work, I can say this project goes a step further in implementation. For instance, Zhang et al. (2023) used a Bi-LSTM-based churn model for e-commerce data, but their work stopped at evaluation. I took it further by deploying Bi-LSTM predictions into Salesforce CRM and triggering automatic follow-up tasks using Apex code and Salesforce Flows. Similarly,Gao and Liu (2022) explored how Salesforce Einstein can be used to personalize customer experience using AI, but they relied on prebuilt tools. My approach was different — I trained models externally using Python and Jupyter Notebook, and then integrated the output with Salesforce using REST APIs and the Simple Salesforce library.

Furthermore, Scholar et al. (2024) highlighted that modular machine learning pipelines integrated with Salesforce CRM enable seamless data ingestion and faster deployment of predictive insights, which aligns with my Azure → Jupyter → Salesforce workflow. Mishachandar and Kumar (2018) also discussed proactive churn prevention and targeted retention campaigns. My Apex Trigger and Flow automation directly reflects such strategies by notifying sales teams and assigning follow-up tasks in real time.

In terms of cloud infrastructure, Scholar et al. (2024) described "Analytics as a Service" in Azure as a scalable approach to storing and processing large datasets, which mirrors how I used Azure Blob Storage for raw CSVs and intermediate files before feeding them into models. From a model performance perspective, My findings also resonate with **?**, who benchmarked Logistic Regression, k-NN, Decision Trees, and Random Forest for churn prediction. In my case, XGBoost and Bi-LSTM outperformed these traditional models by achieving perfect accuracy and ROC-AUC, while still producing actionable business outputs.

From a visualization standpoint, Abedin et al. (2023) emphasized incorporating spatial and geographical patterns in churn analysis for retail e-commerce. My Power BI dashboards directly address this by using region-wise churn heatmaps alongside churn score segmentation. This combination provides both technical and non-technical users with a clear understanding of where and why churn risk is concentrated.

If I had more time or resources, I would also improve the system by using clustering algorithms like K-Means in order to automatically assign customers to Low, Medium, and High churn risk segments instead of using a static churn logic. Furthermore, I would collect behavioral signals such as email open rates, customer support interactions, or cart abandonment, as also suggested by Gao and Liu (2022) in their customer journey research. Using model interpretability methods such as SHAP or LIME could also help to make Bi-LSTM predictions explainable and build trust with non-technical business stakeholders.

In summary, this project not only demonstrated the technical feasibility of churn prediction but also that custom AI models can be deployed into enterprise platforms like Salesforce and Power BI. Unlike prior studies that stop at modeling or dashboarding, my work connects all layers from data ingestion in Azure, predictive modeling in Jupyter, deployment into Salesforce, to visualization in Power BI. This full-stack approach, though challenging, reflects how predictive customer analytics can be made operational and valuable for real-world business use cases.

7 Conclusion and Future Work

This research attempted to answer an essential question: Whether or not it is possible to combine CRM systems like Salesforce with machine learning and deep learning-based AI to forecast and avoid customer attrition in retail environments proactively. The main objectives were to load a large e-commerce data set in a scalable cloud platform (Azure), deploy predictive models through Python, send high-risk customer records to Salesforce, automate follow-ups and notifications through Apex triggers and Flows, and finally, represent the outcomes on Power BI dashboards. I am happy to declare all these objectives were accomplished successfully within a real-world simulation setup.

The key findings of this study were insightful. In all models, Bi-LSTM outperformed traditional algorithms like Naive Bayes and KNN because of the capability to learn sequential behavior, and XGBoost offered the optimum performance among tree-based classifiers with high ROC-AUC and precision score. Execution of high-risk churn predictions in Salesforce through REST APIs enabled Real-time streaming churn detection through technology like Apache Kafka can also be scoped out for implementation in the future, and model deployment on cloud functions or containerized microservices to provide better scalability. And finally, cross-industry pilots of this end-to-end system in industries like telecom, banking, or healthcare, for example, can drive new use cases and revenue streams.

There were some limitations, however. The churn flag was also manually built by a rule (delivery delay > 5 days and review score <= 2), which would not likely be exhaustive in depicting actual-world churn behavior. Salesforce Developer Edition storage space was limited as well, so I could only push out about 2,000 high-risk customer records out of over 90,000, partially inhibiting full-scale testing. In addition, explainability in deep learning models like Bi-LSTM remains a challenge, especially for business environments where stakeholders normally ask "why" questions about the prediction that was achieved.

Despite these limitations, the success of the application in real-world implementations is evidence of the robustness of the framework. It extends the work of researchers like Zhang et al. (2023) not just by using Bi-LSTM for churn prediction but also by integrating the output into a real-time CRM system. Likewise, while Gao and Liu (2022) discussed Salesforce AI personalization using Einstein, my approach showed how custom models trained externally in Python can still power internal Salesforce workflows. The integration with Azure Blob Storage, as well as the dynamic dashboarding with Power BI, also builds upon the insights from Mulpuri and Mathew (2024) work on multi-cloud CRM architectures.

As for future work, there are several meaningful directions this project could take. One refinement can be the development of improved churn labeling logic with the assistance of unsupervised clustering so that churn behavior can be dynamically specified as opposed to static rules. Alternatively, additional behavioral attributes like customer support interactions, responses to marketing campaigns, cart abandonment, or even social media indicators can be included. From a model perspective, application of explainable AI tools like LIME or SHAP for the Bi-LSTM model would yield business trust as well as transparency.

Lastly, the project not only showed that AI and CRM integration is technologically viable, but also established that it can generate real business value to decision-making. The framework outlined here can serve as a roadmap for businesses who wish to better forecast churn, respond quicker via automation, and view risk in interactive dashboards. All based on a smart, scalable, and modular architecture.

References

Abedin, M., Zain, J. M., Saeed, F. and Alghamdi, T. A. (2023). Customer churn in retail e-commerce business: Spatial and machine learning approach, *Journal of Theoretical and Applied Electronic Commerce Research* **17**(1): 131–147.
URL: *https://www.mdpi.com/0718-1876/17/1/9*

Atay, M. T. and Turanli, M. (2025). Analysis of customer churn prediction using logistic

regression, k-nearest neighbors, decision tree and random forest algorithms, *Advances and Applications in Statistics* **92**(2): 147–169.
URL: *https://pphmjopenaccess.com/aas/article/view/2477*

Barga, R., Fontama, V. and Tok, W. H. (2015). *Predictive Analytics with Microsoft Azure Machine Learning, 2nd Edition*, Apress.

Chandar, M. B. (2019). Predicting customer churn using targeted proactive retention, `https://www.researchgate.net/publication/336085532_Predicting_customer_churn_using_targeted_proactive_retention`. Accessed: 2025-08-09.

Chatterjee, S., Rana, N. P., Tamilmani, K. and Sharma, A. (2021). The effect of ai-based crm on organization performance and competitive advantage: An empirical analysis in the b2b context, *Industrial Marketing Management* **97**: 205–219.

Chinta, U., Goel, P. and Renuka, A. (2023). Leveraging ai and machine learning in salesforce for predictive analytics and customer insights, *Universal Research Reports* **10**(1).

Das, S., Banerjee, K., Nath, S. and Chatterjee, S. (2023). Data visualization approach for business strategy recommendation using power bi dashboard, *International Journal of Latest Engineering Science (IJLES)* **6**(5): 101–111. September–October 2023 issue.

Gao, Y. and Liu, H. (2022). Artificial intelligence-enabled personalization in interactive marketing: a customer journey perspective, *Journal of Research in Interactive Marketing* **17**(5): 663–680.

Gokulnath, S. S. and Revathi, M. (2024). Leveraging data analytics in azure for effective churn management, *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*, pp. 1–10.

Grant, V., Sammartino, J., Morris, D. and Franks, B. (2022). *Predictive Analytics with Microsoft Azure Machine Learning*, 2nd edn, Apress, Berkeley, CA.
URL: *https://link.springer.com/book/10.1007/978-1-4842-1200-4*

Guduru, V. S. (2024). Integrating AI with Salesforce for predictive customer insights, *Journal of Artificial Intelligence & Cloud Computing* **3**(5): 1–3.

Khattak, A., Mehak, Z., Ahmad, H., Asghar, M. U., Asghar, M. Z. and Khan, A. (2023). Customer churn prediction using composite deep learning technique, *Scientific Reports* **13**(1): 17294.

Laaksonen, A. (2020). The use of artificial intelligence in customer relationship management, *Technical report*, University of Jyväskylä, Faculty of Information Technology. Candidate's thesis, Jyväskylä University.

Mishachandar, B. and Kumar, K. A. (2018). Predicting customer churn using targeted proactive retention, *International Journal of Engineering & Technology* **7**(2.27): 69–76.

Mulpuri, R. and Mathew, J. (2024). Comprehensive review of multi-cloud architecture for salesforce in enterprise environments, *International Journal of Computer Applications* **186**(20): 1–8.

URL: *https://www.researchgate.net/publication/393356567*$_C$*omprehensive*$_R$*eview*$_o$*f*$_M$*ulti−cloud*$_A$*rchitecture*$_f$*or*$_S$*alesforce*$_i$*n*$_E$*nterprise*$_E$*nvironments*

Murugan, S., Daniel, K. L., Ananthi, D. M., Rajkumar, D. P. and Jenitha, S. S. (2024). Retail store sales analysis: Unveiling insights through power bi business analytics, *Proceedings of the International Conference on Innovative Computing & Communication (ICICC 2024).* SSRN Electronic Journal, posted 7 Dec 2024.
URL: *https://ssrn.com/abstract=5039793*

Patil, A. P., Deepshika, M. P., Mittal, S., Shetty, S., Hiremath, S. S. and Patil, Y. E. (2017). Customer churn prediction for retail business, *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, IEEE, pp. 845–851.

Pethad, C. A. (2022). Guide to implement salesforce einstein prediction builder, *Journal of Marketing & Supply Chain Management* pp. 1–6.

Phan, T., Tran, H., Ho, T., Huynh, V. and Nguyen, N. (2023). Customer churn in retail e-commerce business: Spatial and machine learning approach, *Journal of Theoretical and Applied Electronic Commerce Research* **17**(1): 135–150.
URL: *https://www.mdpi.com/0718-1876/17/1/9*

Putra, R. K., Rahmawati, R., Hendrawan, D., Wibowo, H., Saputra, R. and Pratama, A. (2024). Designing an analytics dashboard for knowledge extraction in the retail industry using descriptive and predictive analytics, *Journal of Information Systems and Informatics* **6**(1): 34–50.
URL: *https://jurnal.untirta.ac.id/index.php/jiss/article/view/27900/14176*

Saini, A., Singh, R. and Gupta, A. (2024). Leveraging data analytics in azure for effective churn management, *2024 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 1–6.
URL: *https://ieeexplore.ieee.org/document/10493670*

Sanodia, G. (2023). Enhancing Salesforce CRM with artificial intelligence, *International Journal of Artificial Intelligence Research* **1**(1): 52–61.
URL: *https://www.researchgate.net/publication/383943800*$_E$*NHANCING*$_S$*ALESFORCE*$_C$*RM*$_W$

Scholar, V., Kumar, A. and Sharma, P. (2024). Analyzing the role of machine learning pipelines in enhancing predictive analytics for salesforce cloud-based customer relationship management, *International Journal of Emerging Technologies and Innovative Research* **11**(5): 112–120.
URL: *https://www.researchgate.net/publication/391628210*$_A$*NALYZING*$_T$*HE*$_R$*OLE*$_O$*F*$_M$*ACHIN*
BASED$_C$*USTOMER*$_R$*ELATIONSHIP*$_M$*ANAGEMENT*

Zhang, X., Guo, F., Chen, T., Pan, L., Beliakov, G. and Wu, J. (2023). A brief survey of machine learning and deep learning techniques for e-commerce research, *Journal of Theoretical and Applied Electronic Commerce Research* **18**(4): 2188–2216.