DESIGNING A CHURN PREDICTION SYSTEM FOR TELECOM COMPANIES TO ANALYZE USAGE PATTERNS AND COUSTMER BEHAVIOUR OPTIMIZING MARKETING AND RETENTION

A dissertation submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science and Engineering

By

G.SRIDHAR(23U61A0580)

Under the guidance of

Mrs. Noore Ilahi
B. Tech., M. Tech.
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(Approved by AICTE, New Delhi & Affiliated to JNTUH)
(Recognized under section 2(f) of UGC Act 1956)
An ISO:9001-2015 Certified Institution
CHILKUR (V), MOINABAD (M), R.R. DIST. T.S-501504

June 2025

GLOBAL Institute of Engineering & Technology

Approved By AICTE, Affiliated To JNTUH EAMCET CODE: GLOB

(Approved by AICTE & Affiliated to JNTUH)
(Recognized under Section 2(f) of UGC Act 1956)
An ISO:9001-2015 Certified Institution
Survey No. 179, Chilkur (V), Moinabad (M), Ranga Reddy Dist. TS.
JNTUH Code (U6) ECE –EEE-CSD-CSM – CSE - CIVIL – ME – MBA - M.Tech EAMCET Code - (GLOB)

Department of Computer Science and Engineering

Noore Ilahi
B. Tech., M. Tech.
Assistant Professor & Head

Date: 02-06-2025

CERTIFICATE

This is to certify that the project work entitled "**DESIGNING A CHURN PPEDICTION SYSTEM FOR TELECOM COMPANIES TO ANALYZE USAGE PATTERNS AND COUSTMER BEHAVIOR OPTIMIZING MARKETING AND RETENTION**", is a bonafide work of **G.SRIDHAR (HT.No:23U61A0580) ,** submitted in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science and Engineering** during the academic year 2024-25. This is further certified that the work done under my guidance, and the results of this work have not been submitted elsewhere for the award of any other degree or diploma.

Internal Guide

Mrs. Noore Ilahi
Assistant Professor

Head of the Department

Mrs. Noore Ilahi
Assistant Professor

DECLARATION

I hereby declare that the project work entitled **DESIGNING A CHURN PPEDICTION SYSTEM FOR TELECOM COMPANIES TO ANALYZE USAGE PATTERNS AND COUSTMER BEHAVIOR OPTIMIZING MARKETING AND RETENTION ,** submitted to **Department of Computer Science and Engineering, Global Institute of Engineering & Technology, Moinabad,** affiliated to **JNTUH, Hyderabad** in partial fulfillment of the requirement forthe award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is the work done by me and has not been submitted elsewhere for the awardof any degree or diploma.

G.SRIDHAR (23U61A0580)

ACKNOWLEDGEMENT

I am thankful to my guide **Mrs. Noore Ilahi,** Assistant Professor of CSE Department for her valuable guidance for successful completion of this project.

I express my sincere thanks to **Mrs. G. Pavani,** Project Coordinator for giving me an opportunity to undertake the project " **DESIGNING A CHURN PPEDICTION SYSTEM FOR TELECOM COMPANIES TO ANALYZE USAGE PATTERNS AND COUSTMER BEHAVIOR OPTIMIZING MARKETING AND RETENTION** " and for enlightening me on various aspects of my project work and assistance in the evaluation of material and facts. She not only encouraged me to take up this topic but also given her valuable guidance in assessing facts and arriving at conclusions.

I am also most obliged and grateful to **Mrs. Noore Ilahi,** Assistant Professor and Head, Department of CSE for giving me guidance in completing this project successfully.

I express my heart-felt gratitude to our Vice-Principal **Prof. Dr. G Ahmed Zeeshan**, Coordinator Internal Quality Assurance Cell (IQAC) for his constant guidance, cooperation, motivation and support which have always kept me going ahead. I owe a lot of gratitude to him for always being there for me.

I also most obliged and grateful to our Principal **Dr. P. Raja Rao** for giving me guidancein completing this project successfully.

I also thank my parents for their constant encourage and support without which the project would have not come to an end.

Last but not the least, I would also like to thank all my class mates who have extended their cooperation during our project work.

G.SRIDHAR (23U61A0580)

VISION

The Vision of the Department is to produce professional Computer Science Engineers who can meet the expectations of the globe and contribute to the advancement of engineering and technology which involves creativity and innovations by providing an excellent learning environment with the best quality facilities.

MISSION

M1. To provide the students with a practical and qualitative education in a modern technical environment that will help to improve their abilities and skills in solving programming problems effectively with different ideas and knowledge.

M2. To infuse the scientific temper in the students towards the research and development in Computer Science and Engineering trends.

M3. To mould the graduates to assume leadership roles by possessing good communication skills, an appreciation for their social and ethical responsibility in a global setting, and the ability to work effectively as team members.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: To provide graduates with a good foundation in mathematics, sciences and engineering fundamentals required to solve engineering problems that will facilitate them to find employment in MNC’s and / or to pursue postgraduate studies with an appreciation for lifelong learning.

PEO2: To provide graduates with analytical and problem solving skills to design algorithms, other hardware / software systems, and inculcate professional ethics, inter-personal skills to work in a multi-cultural team.

PEO3: To facilitate graduates to get familiarized with the art software / hardware tools, imbibing creativity and innovation that would enable them to develop cutting edge technologies of multi disciplinary nature for societal development.

PROGRAMME OUTCOMES:

PO1: Engineering knowledge: An ability toApply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: An ability to Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural science and engineering sciences.

PO3: Design/development of solutions: An ability to Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal and environmental considerations.

PO4: Conduct investigations of complex problems: An ability to Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5: Modern tool usage: An ability to Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO6: The engineer and society: An ability to Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7: Environment sustainability: An ability to Understand the impact of the professional engineering solutions in the societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
PO8: Ethics: An ability to Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9: Individual and teamwork: An ability to Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10: Communication: An ability to Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11: Project management and finance: An ability to Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12: Lifelong learning: An ability to Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broader context of technological change.

PROGRAMME SPECIFIC OUTCOMES
PSO1: An Ability to Apply the fundamentals of mathematics, Computer Science and Engineering Knowledge to analyze and develop computer programs in the areas related to Algorithms, System Software, Web Designing, Networking and Data mining for efficient Design of computer-based system to deal with Real time Problems.
PSO2: An Ability to implement the Professional Engineering solutions for the betterment of Society, and able to communicate with professional Ethics effectively

ABSTRACT

This project presents a predictive churn analysis system designed for telecom companies, utilizing supervised machine learning techniques to identify customers at risk of discontinuing services. The model is trained on structured datasets comprising customer demographics, service usage patterns, billing behavior, contract type, and support interaction history. The objective is to empower telecom providers to proactively address churn through personalized retention strategies and targeted marketing campaigns.

The system incorporates a robust backend pipeline that performs data preprocessing, feature engineering, model training, and churn probability scoring. A web-based interface allows business users to analyze individual customer risk, explore churn drivers via model interpretability tools (e.g., SHAP), and export actionable insights. Admin functionality includes dataset updates, model retraining, and performance monitoring to maintain accuracy over time.

Model performance is evaluated using metrics such as Accuracy, Precision, Recall, F1 Score, and AUC-ROC, demonstrating high reliability across diverse customer segments. Future improvements include integration with real-time CRM data streams, adoption of deep learning architectures for behavioral modeling, and automated alert systems for high-risk churn profiles.

This solution addresses the increasing demand for intelligent customer retention tools in a highly competitive telecom market and establishes a scalable foundation for advanced customer lifecycle management.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In today's fiercely competitive telecom industry, retaining customers has become as crucial as acquiring new ones. The dynamic nature of customer behavior, service

quality, pricing wars, and the emergence of alternate service providers contribute significantly to customer churn. Churn, or the loss of subscribers, directly impacts a telecom company's revenue and market share. Thus, accurately predicting customer churn is not just a strategic advantage but a business necessity.

With the rise of data-driven technologies, especially artificial intelligence (AI) and machine learning (ML), telecom companies now have the tools to better understand customer behavior through usage patterns, service interactions, and engagement metrics. Leveraging large volumes of customer data—such as call details, data usage, billing history, and complaint logs—ML models can identify potential churners early and help tailor personalized retention strategies.

This project aims to develop an intelligent churn prediction system that utilizes machine learning to analyze telecom customer data and forecast churn risk. By identifying behavioral trends and critical indicators of dissatisfaction, the system supports telecom companies in designing timely interventions such as customized offers, service upgrades, or loyalty programs. The solution will include a user-friendly web interface where analysts or marketing teams can input customer data and receive real-time churn predictions, enabling proactive and data-driven customer retention efforts.

1.1 EXISTING SYSTEM

Traditional customer retention strategies in the telecom sector often rely on generalized loyalty programs, post-churn surveys, and reactive outreach methods. These legacy approaches typically lack personalization and fail to utilize the rich behavioral data generated by customers on a daily basis. Manual data analysis, static reports, and simple rule-based systems fall short in predicting churn effectively, especially given the complex, nonlinear interactions among service quality, customer satisfaction, and market competition.

In more recent applications, statistical models such as logistic regression and decision trees have been used to estimate churn probabilities. While moderately effective, these models struggle with high-dimensional data and intricate feature interactions. Research has shown that more advanced models—such as Random Forests, Gradient Boosting Machines (e.g., XGBoost), and neural networks—significantly outperform traditional approaches by capturing deeper patterns and behavioral signals.

Despite these advancements, many existing churn prediction systems are limited in scope, offer poor interpretability, lack real-time prediction capabilities, or are not easily scalable across different segments of the telecom market. This project addresses these limitations by delivering a modular, scalable, and interactive churn prediction system designed for practical deployment in telecom environments.

1.3 PROPOSED SYSTEM

Our Telecom Churn Prediction System features a real-time pipeline activated the moment a user submits customer usage data, service history, and demographic information through a responsive web interface. The system accepts structured input either through form submissions (HTTP POST) or integrated APIs, allowing seamless connection to telecom customer databases or CRM platforms.

Upon receiving data, the preprocessing layer encodes categorical features (e.g., contract type, payment method) using techniques such as label encoding and one-hot encoding. Numerical features—such as monthly charges, call minutes, and data usage—are normalized using standard scaling methods to ensure model consistency. Additional features such as tenure buckets, average call duration, or recent complaint history may be dynamically generated to improve prediction performance.

The processed data is then fed into a pre-trained machine learning model—such as Logistic Regression, XGBoost, or a deep learning classifier—that has been trained on historical customer behavior and churn outcomes. The model outputs a churn probability score along with relevant metadata such as prediction timestamp and model version.

The system's backend is developed using Flask, following a clean, modular design that separates data ingestion, transformation, model inference, and result delivery. This structure supports horizontal scalability and rapid model upgrades. The frontend, built with Bootstrap and custom styling, includes intuitive elements such as dropdown menus, toggles for service features, and real-time visualizations of churn risk.

In addition to predictions, the system logs model performance metrics, user interactions, and data anomalies, enabling analysts to monitor accuracy and retrain models periodically. This continuous learning loop ensures the system evolves with changing customer behavior.

1.4 ADVANTAGES OF THE PROPOSED SYSTEM

- **Real-Time Churn Prediction: Provides immediate churn risk scores for each customer profile, enabling timely retention actions.**
- **Behaviorally-Driven Modeling: Uses machine learning to capture subtle behavioral patterns and service-related triggers of churn.**
- **Scalable and Modular Architecture: Designed for easy extension and maintenance, with clear separation of components and functionality.**
- **User-Friendly Interface: Web-based frontend with interactive inputs and clear churn indicators ensures accessibility for non-technical users.**
- **Deployable and Lightweight: Built with Flask, Pandas, and Scikit-learn, the system is easy to deploy across different environments or cloud services.**

- **Stateless and Secure: No customer data is stored between sessions, ensuring data privacy and enabling use in secure or shared environments.**
- **Insightful and Actionable: Generates interpretable insights that telecom companies can use to tailor retention campaigns and enhance service offerings.**

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Customer churn prediction has become a focal point of research and development in the telecommunications industry due to the high cost of acquiring new customers and the significant impact of churn on business revenue. Traditional strategies for customer retention have relied heavily on static segmentation, historical trends, or customer satisfaction surveys—methods that often lack accuracy and scalability.

With advancements in machine learning (ML) and big data analytics, more sophisticated approaches are now used to detect patterns and signals that indicate potential customer churn. These methods provide telecom companies with predictive insights that empower them to act before a customer leaves, offering personalized retention strategies at the right time.

This chapter reviews key academic contributions and industry approaches relevant to churn prediction. It also outlines the machine learning models, behavioral features, and tools employed in this project.

2.2 Existing Work on Churn Prediction Systems

Numerous research studies and real-world implementations have explored customer churn prediction across telecom and other subscription-based industries. Here are some notable works:

- **Idris et al.**
  Applied decision trees and ensemble methods on telecom datasets to identify churn patterns. Their work emphasized the importance of features like tenure, call charges, and contract type. They found

Random Forests to be both interpretable and effective.

- **Verbeke et al.**
  Compared logistic regression, support vector machines (SVM), and neural networks on customer churn data. Their study highlighted the trade-off between interpretability (logistic regression) and predictive power (neural networks).
- **Keramati et al.**
  Used data mining techniques including neural networks, naïve Bayes, and decision trees to improve customer churn prediction. Their ensemble-based model improved accuracy over individual classifiers.
- **Ahmed et al.**
  Focused on feature engineering and preprocessing for telecom churn datasets. They proposed using derived features like average monthly usage, complaint frequency, and billing irregularities for better model performance.

These studies demonstrate that combining strong feature selection with advanced ML models can significantly improve churn prediction accuracy. Techniques such as oversampling (SMOTE), dimensionality reduction (PCA), and ensemble learning (XGBoost, LightGBM) have become standard in churn modeling pipelines.

2.3 Machine Learning Techniques in Churn Prediction

Modern churn prediction systems use a blend of classification models and behavioral modeling techniques. The most widely used methods include:

- **Supervised Learning (Logistic Regression, Random Forest, XGBoost):**
  These are standard for binary classification tasks such as churn vs. no churn. XGBoost, in particular, handles class imbalance and nonlinear relationships effectively.
- **Unsupervised Learning (Clustering, Anomaly Detection):**
  Applied in scenarios where labeled churn data is limited. Clustering can help in customer segmentation, which aids personalized retention campaigns.
- **Deep Learning (Neural Networks, LSTM):**

LSTM networks are useful in modeling customer behavior over time, such as daily or monthly usage patterns.

- **Feature Engineering:**
  Features such as service subscription type, complaint frequency, data usage trends, and payment patterns have shown high predictive power in telecom churn studies.
- **Evaluation Techniques:**
  Precision, recall, F1-score, and AUC-ROC are commonly used to evaluate model performance, especially in imbalanced datasets where accuracy alone can be misleading.

2.4 Tools and Technologies Used

- **Python:**
  A versatile programming language with robust support for machine learning through libraries like Scikit-learn, Pandas, NumPy, and TensorFlow.
- **Flask:**
  A lightweight web framework used to build the system's frontend and REST API, facilitating user interaction and real-time prediction access.
- **Scikit-learn:**
  Used for building and training classical machine learning models such as Logistic Regression, Decision Trees, and XGBoost.
- **Pandas & NumPy:**
  Essential for data manipulation, transformation, and analysis of telecom datasets.
- **Matplotlib & Seaborn:**
  Used to visualize customer behavior patterns, churn distributions, and model performance metrics.
- **XGBoost:**
  Chosen for its speed and performance in handling large feature sets and class imbalance.
- **SMOTE (Synthetic Minority Over-sampling Technique):**
  Applied during preprocessing to balance the dataset and reduce bias toward the majority class.

2.5 Summary

From the review of literature and technologies, it is evident that machine learning-based churn prediction systems outperform traditional static models in both accuracy and adaptability. This project builds on these insights by combining well-established algorithms like XGBoost and Logistic Regression with behavioral feature engineering, all integrated into a responsive, web-based application built with Flask. The goal is to offer a practical, real-time churn prediction tool for telecom operators that supports timely and data-driven customer retention actions.

```
+----------------------------+
|        User Input          |
|--------------------------- |
| Gender, Senior Citizen,    |
| Partner, Tenure, Charges,  |
| Contract, Internet, etc.   |
+-------------+--------------+
              |
              v
+------------------------------+
|        Web UI                |
| (Flask + HTML Template)      |
+--------------+---------------+
               |
               v
+------------------------------+
|       Flask Backend          |
|         (app.py)             |
+--------------+---------------+
               |
               v
+------------------------------+
|     Preprocessing Layer      |
| (Encoding + Scaling Steps)   |
+--------------+---------------+
               |
               v
+------------------------------+
|      Trained ML Model        |
|     (churn_model.pkl)        |
+--------------+---------------+
               |
               v
+------------------------------+
|    Generate Prediction       |
|   (Churn or No Churn)        |
+--------------+---------------+
               |
               v
+------------------------------+
|  Output Displayed to User    |
+------------------------------+
```

CHAPTER 3

SYSTEM ANALYSIS:

This project focuses on the design and development of a churn prediction system tailored for telecom companies, leveraging customer behavior and usage patterns to anticipate churn risks. The system enables telecom service providers to proactively implement retention strategies based on predictive insights. Key design priorities include ease of use, speed, and cross-platform accessibility to ensure high adoption by non-technical marketing and operations personnel.

3.1 REQUIREMENT SPECIFICATIONS

3.1.1 HARDWARE REQUIREMENTS

- **System: Intel i5 or above, 3.2 GHz processor**
- **Hard Disk: Minimum 512 GB**
- **Monitor: 14" Color Monitor**
- **Input Devices: Optical Mouse and Keyboard**
- **RAM: Minimum 8 GB**

3.1.2 SOFTWARE REQUIREMENTS

- **Operating System: Windows 11 / Windows 10**
- **Programming Language: Python**
- **Frontend Technologies: HTML, CSS, JavaScript**
- **Framework: Flask (for web application backend)**
- **Libraries/Packages:**
  - **Pandas (data handling and feature engineering)**
  - **Scikit-learn (machine learning models like Logistic Regression, Random Forest)**
  - **XGBoost (advanced predictive modeling)**
  - **Matplotlib / Seaborn (visualization)**
  - **Pickle (model serialization)**
  - **SMOTE (Synthetic Minority Oversampling Technique for class balancing)**
- **Browser Compatibility: All modern browsers supported (recommended: Google Chrome)**

3.1.3 FUNCTIONAL REQUIREMENTS

- **A simple web-based interface that allows telecom users to input:**
  - **Customer demographic and account data (e.g., tenure, service type, charges)**
  - **Behavior metrics (e.g., monthly usage, customer support calls)**
- **On form submission, the backend should:**

    - **Load a pre-trained churn prediction model (e.g., XGBoost)**
    - **Analyze the submitted features and classify the churn risk (Yes/No)**
    - **Display the prediction result along with a confidence score and visual output**
- **Full compatibility with desktop browsers for maximum accessibility**

3.2 FEASIBILITY STUDY

The feasibility of the Telecom Churn Prediction System was assessed across economic, technical, and social dimensions to confirm that the solution is viable and impactful.

3.2.1 ECONOMICAL FEASIBILITY

The solution is cost-effective and built entirely with open-source software and libraries. Technologies such as Python, Flask, and Scikit-learn require no licensing fees, ensuring minimal cost of development and deployment. The system is also lightweight, requiring only standard computing resources and internet access, making it deployable even in low-cost environments.

3.2.2 TECHNICAL FEASIBILITY

The system uses robust and widely supported technologies that ensure long-term maintainability and compatibility. Flask handles API routing and user interaction efficiently, while machine learning models like XGBoost and Logistic Regression provide fast and reliable predictions. With preprocessed data and lightweight model execution, the system runs smoothly even on moderate hardware, ensuring rapid and responsive user experience.

3.2.3 SOCIAL FEASIBILITY

The system was built with telecom marketing and customer service teams in mind—users who may not have technical expertise in data science. The web interface minimizes complexity and requires only basic customer data inputs to generate actionable churn insights. Clear visual outputs and a straightforward design enhance usability and ensure that the system can be easily adopted and understood with minimal training.

CHAPTER 4

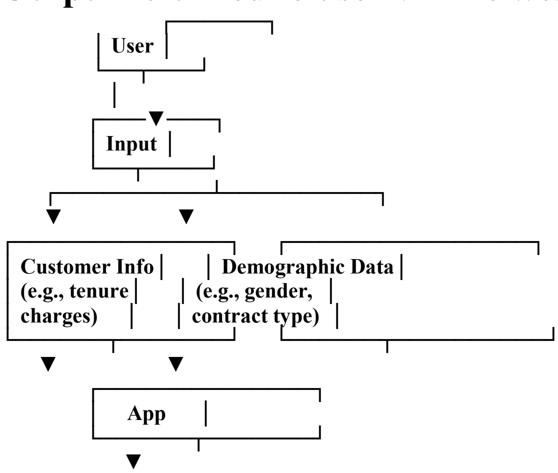
SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM (DFD)

The Data Flow Diagram (DFD) illustrates the logical flow of data in the Telecom Churn Prediction System. The system enables telecom users (e.g., customer service teams or analysts) to input customer-related data, submit the details for analysis, and receive a churn prediction result.

Data Flow Overview:

1. **User Input**
   - **Customer demographic and service data (e.g., tenure, plan type, monthly charges)**
   - **Behavioral indicators (e.g., support calls, data usage)**
2. **Input Validation**
   - **Backend validates data types and missing fields**
3. **Preprocessing**
   - **Applies transformations (e.g., encoding categorical data, scaling numerical features)**
4. **Model Inference**
   - **Pre-trained churn prediction model (e.g., XGBoost) processes the input**
5. **Prediction Output**
   - **Churn result: Yes / No**
   - **Confidence Score**
   - **Optional graphical interpretation**
6. **Result Display**
   - **Output returned to user via the web interface in a user-friendly format**



```
ML Model              |
(RandomForest/XGBoost)|
        ▼
   ▼            ▼
Churn Result |     | Churn Probability |
 (Yes / No)  |     | Plot              |
```

4.3 UML DIAGRAMS

4.3.1 USE CASE DIAGRAM

Actors:

- **Telecom Analyst / Marketing User**

Use Cases:

- **Input customer profile and behavior data**
- **Submit for churn prediction**
- **View churn result and model confidence**
- **Export or analyze results for campaign targeting**

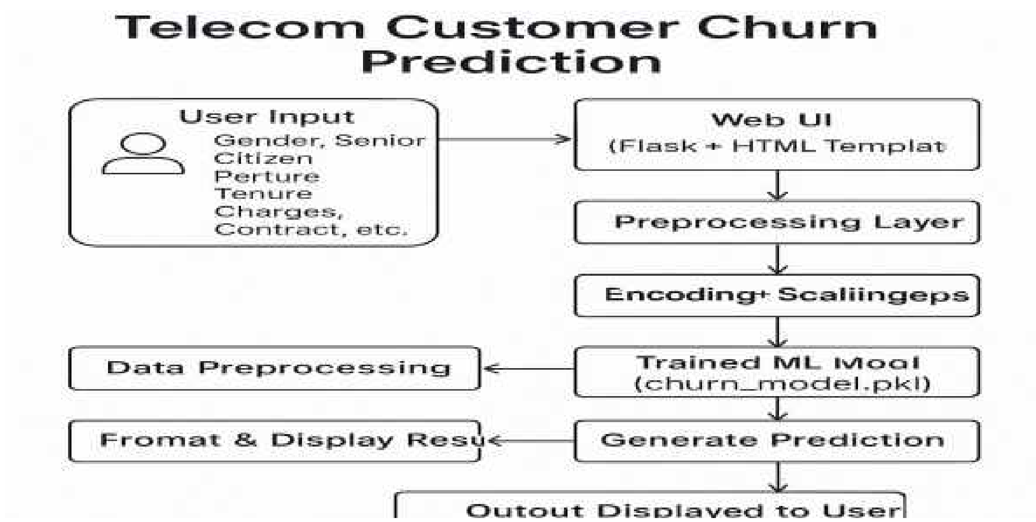
This diagram represents how a user interacts with the system via the web application interface.

```
              +-------------------------+
              |   Submit Forecast       |
              +-----------▲-------------+
                          |
            +-------------+-------------+
            |                           |
       <<include>>                 <<include>>
   +-------------------+       +---------------------+
   |  Validate Inputs  |       |  Load Sales Data    |
   +-------------------+       +---------------------+
            |                           |
       <<include>>                 <<include>>
   +-------------------+       +----------------------+
   |   Run Prediction  |       |  Generate Forecast   |
   +-------------------+       +----------------------+
            |                           |
            +-------------+-------------+
                          |
                    <<include>>
              +-------------------------+
              | Format & Display Output |
              +-------------------------+
                          |
              +-------------------------+
              |     Reset Form          |
              +-----------▲-------------+
                          |
                     +---+---+
                     | User  |
                     +-------+
```

4.3.2 COMPONENT DIAGRAM

Main Components:

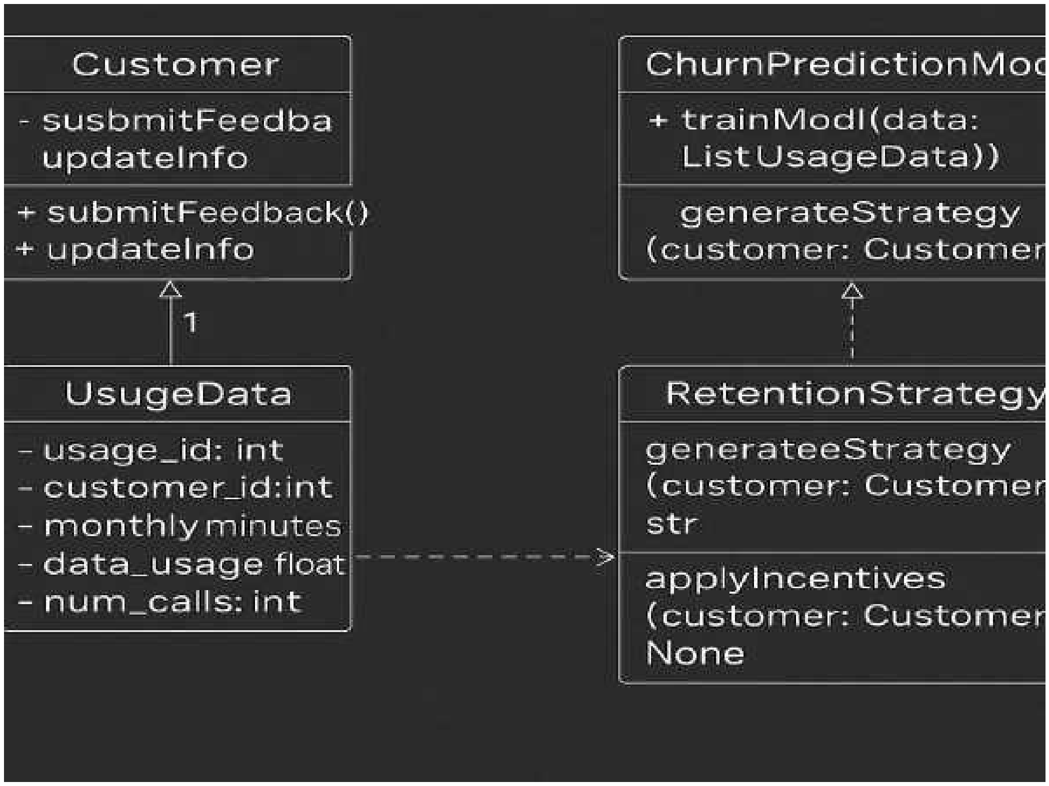
- **Frontend (UI)**
  - **HTML + CSS + JS (index.html): Used to collect customer data**
- **Flask Backend (app.py)**
  - **Handles routing, request processing, model inference**
- **Preprocessing Module**
  - **Cleans and transforms input data**
- **ML Model Module**
  - **Loads the serialized XGBoost model for churn prediction**
- **Response Module**
  - **Formats and sends prediction results to frontend**



4.3.3 CLASS DIAGRAM

Although the system primarily uses function-based programming, a conceptual class-based design might include:

- **CustomerData**
  - **Attributes: tenure, plan_type, monthly_charges, etc.**
  - **Methods: validate_input(), preprocess()**
- **ChurnPredictor**
  - **Attributes: model, scaler, encoder**
  - **Methods: load_model(), predict(), get_confidence_score()**
- **ResultFormatter**
  - **Methods: format_output(), generate_visuals()**



4.3.4 ACTIVITY DIAGRAM

1. **User opens web app (index.html)**
2. **Enters customer data: tenure, services used, support calls, etc.**
3. **Clicks on "Predict Churn"**
4. **app.py receives and validates input**
5. **Preprocessing module transforms the data**
6. **Churn prediction model (e.g., xgboost_model.pkl) generates prediction**
7. **System returns result (Yes/No + Confidence) to the user**
8. **User views prediction and possibly downloads the result**
9. **End**

●
↓

Collect Customer Data
↓
Validate Input Data
↓
Is Data Valid? ——————————————→ [No] → Reject Data
↓ **Yes**
Preprocess & Encode Data
↓
Run Churn Prediction Model
↓
Is Customer Likely to Churn? ——→ [No] → Display "Not at Risk"
↓ **Yes**
Generate Retention Strategy
↓
Display Recommendation

↓

◎

4.3.5 SEQUENCE DIAGRAM

Objects:

- **User**
- **Web Browser (UI)**
- **Flask Backend**
- **Preprocessing Module**
- **Prediction Model**
- **Result Display**

Sequence Flow:

1. **User → UI: Enter customer data**
2. **UI → Flask: Send data via POST request**
3. **Flask → Preprocessing: Clean and encode data**
4. **Flask → Model: Pass features for churn prediction**
5. **Model → Flask: Return prediction result**
6. **Flask → UI: Send result and confidence level**
7. **UI → User: Display prediction and optional visuals**

```
User      Browser      churn_app.py      churn_model.pkl      strategy_module
            |           |            |              |              |
         |--Input & Submit Form------->|              |              |
          |           |--POST /predict------------------->|              |
          |           |            |--Parse Data------------------------->|
         |           |            |--generate_prediction()--------------->|
           |           |            |              |--Load Model------|
           |           |            |              |<--Predict Churn--|
          |           |            |<--Churn Result-----------------------|
          |           |            |--generate_strategy()---------------->|
            |           |            |              |              |
      |           |<--Return Result Page (recommendation)-----------------|
         |<--Display Recommendation-----|              |              |
```

CHAPTER 5

IMPLEMENTATION:

1. Data Collection

The project starts with collecting customer data in CSV format. The dataset includes telecom service usage, customer demographics, tenure, billing, contract

type, and churn labels. This rich, labeled dataset is essential for training the churn prediction model and identifying patterns that lead to customer attrition.

---

2. Data Preprocessing
The dataset is first cleaned by handling missing values and removing irrelevant columns. Categorical features (e.g., contract type, payment method) are encoded using Label Encoding or One-Hot Encoding. Numerical features are scaled to normalize the data range. Feature selection techniques are applied to retain the most informative attributes.

---

3. Model Training (XGBoost)
The core machine learning model used for churn prediction is XGBoost (Extreme Gradient Boosting)—chosen for its performance on classification tasks. The data is split into training and testing sets. The model is trained using the training set and validated using metrics such as accuracy, precision, recall, and F1-score on the test set. Hyperparameter tuning is done using Grid Search or Randomized Search to improve model performance.

---

4. Churn Prediction Logic
After training, the model is serialized using Python's pickle module. At runtime, user input data is preprocessed in the same format as the training data and passed into the loaded model. The model returns a binary prediction (churn or no churn) along with a confidence score indicating the certainty of the prediction.

---

5. Web Integration using Flask
The backend logic is integrated into a Flask web application. The user submits data through a web form. Flask routes the request to the prediction engine, which loads the XGBoost model, processes the input, and returns the churn prediction. The app ensures a seamless flow from input to output without refreshing the entire page.

---

6. User Interface (HTML + Bootstrap)
The frontend is built using HTML, CSS, and Bootstrap for responsiveness. A clean web form allows users to enter various customer parameters (e.g., tenure, internet usage, billing type). After submission, the churn prediction is displayed along with a confidence score and a meaningful message (e.g., "High churn risk" or "Low churn risk").

---

7. Result Visualization (Matplotlib/Plotly)
For enhanced understanding, charts can be generated using Matplotlib or Plotly to show churn probability distributions, feature importance, or trends across customer

segments. These plots help users visually interpret the output and make data-driven decisions.

---

8. Session Management and Input Reset
Each form submission initiates a fresh session, clearing any previously entered data or prediction results. This ensures that predictions are session-isolated and users receive a fresh response every time without confusion or cached outputs.

---

9. Continuous Learning and Scalability
Although this is a prototype, the system is designed with scalability in mind. In a production environment, it could be connected to live telecom databases, and the model retrained periodically with fresh customer data. Advanced techniques such as drift detection and model versioning could be integrated to maintain prediction accuracy over time.

MAIN.PY

```python
import pandas as pd
import numpy as np
import joblib

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
from xgboost import XGBClassifier
from imblearn.over_sampling import SMOTE

# 1. Load data
df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
df.drop('customerID', axis=1, inplace=True)

# 2. Clean data
df["TotalCharges"] = pd.to_numeric(df["TotalCharges"], errors='coerce')
df["TotalCharges"].fillna(df["TotalCharges"].median(), inplace=True)
df["Churn"] = df["Churn"].map({'Yes': 1, 'No': 0})

# 3. Encode categorical variables
df_encoded = pd.get_dummies(df, drop_first=True)

# 4. Split into features and target
X = df_encoded.drop("Churn", axis=1)
y = df_encoded["Churn"]

# 5. Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# 6. Feature scaling
```

```python
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 7. Handle imbalance with SMOTE
sm = SMOTE(random_state=42)
X_train_res, y_train_res = sm.fit_resample(X_train_scaled, y_train)

# 8. XGBoost model tuning
xgb = XGBClassifier(random_state=42, eval_metric='logloss')
params = {
    'n_estimators': [200],
    'max_depth': [7],
    'learning_rate': [0.1],
    'subsample': [0.8],
    'colsample_bytree': [0.8]
}
grid = GridSearchCV(xgb, params, cv=3, scoring='f1', verbose=1, n_jobs
grid.fit(X_train_res, y_train_res)

best_model = grid.best_estimator_
print("Best Parameters:", grid.best_params_)

# 9. Prediction with custom threshold
y_proba = best_model.predict_proba(X_test_scaled)[:, 1]
threshold = 0.4
y_pred = (y_proba >= threshold).astype(int)

# 10. Evaluation
```

```python
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# 11. Save model, scaler, features, and threshold
joblib.dump(best_model, "churn_model.pkl")
joblib.dump(scaler, "scaler.pkl")
joblib.dump(X.columns.tolist(), "features.pkl")
joblib.dump(threshold, "threshold.pkl")
joblib.dump(0.4, "threshold.pkl")

print("\nModel, scaler, features, and threshold saved.")
```

APP.PY

```python
from flask import Flask, render_template, request
import joblib
import numpy as n  (module) pd
import pandas as pd

app = Flask(__name__)

# Load model and preprocessing objects
model = joblib.load("model/churn_model.pkl")
features = joblib.load("model/features.pkl")
scaler = joblib.load("model/scaler.pkl")
threshold = joblib.load("model/threshold.pkl")

@app.route("/", methods=["GET", "POST"])
def home():
    prediction = None
    if request.method == "POST":
        form = request.form

        # Extract input values
        input_data = {
            'gender': form['gender'],
            'SeniorCitizen': form['SeniorCitizen'],
            'Partner': form['Partner'],
            'PhoneService': form['PhoneService'],
            'tenure': float(form['tenure']),
            'MonthlyCharges': float(form['MonthlyCharges']),
            'TotalCharges': float(form['TotalCharges']),
            'InternetService': form['InternetService'],
            'Contract': form['Contract']
        }
```

```python
        # Create DataFrame and encode
        df = pd.DataFrame([input_data])
        df_encoded = pd.get_dummies(df).reindex(columns=features, fill_value=0)

        # Scale
        df_scaled = scaler.transform(df_encoded)

        # Predict
        pred_proba = model.predict_proba(df_scaled)[0][1]
        prediction = "Churn" if pred_proba > threshold else "No Churn"

    return render_template("index.html", prediction=prediction)

if __name__ == "__main__":
    app.run(debug=True)

```

INDEX.HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Telecom Churn Prediction</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h2 class="mb-4 text-center">Telecom Customer Churn Prediction</h2>
    <form method="POST" action="/">
        <div class="row">
            <div class="col-md-6 mb-3">
                <label>TENURE <small class="text-muted">(Months with telecom company)</small></label>
                <input type="number" name="tenure" class="form-control" min="0" required>
            </div>

            <div class="col-md-6 mb-3">
                <label>PHONE SERVICE <small class="text-muted">(Do you have traditional phone service?)</small></label>
                <select name="PhoneService" class="form-control" required>
                    <option value="">Select your choice</option>
                    <option value="Yes">Yes</option>
                    <option value="No">No</option>
                </select>
            </div>

            <div class="col-md-6 mb-3">
                <label>MONTHLY CHARGES <small class="text-muted">(Amount you pay every month)</small></label>
                <input type="number" step="0.01" name="MonthlyCharges" class="form-control" min="0" required>
            </div>

            <div class="col-md-6 mb-3">
```

```
    <label>TOTAL CHARGES <small class="text-muted">(Total amount paid so far)</small></label>
    <input type="number" step="0.01" name="TotalCharges" class="form-control" min="0" required>
</div>

<div class="col-md-6 mb-3">
    <label>STATUS <small class="text-muted">(Marital status)</small></label>
    <select name="Partner" class="form-control" required>
        <option value="">Select your choice</option>
        <option value="Yes">Married</option>
        <option value="No">Unmarried</option>
    </select>
</div>

<div class="col-md-6 mb-3">
    <label>SENIOR CITIZEN</label>
    <select name="SeniorCitizen" class="form-control" required>
        <option value="">Select your choice</option>
        <option value="0">No (Below 60 years)</option>
        <option value="1">Yes (60 years or older)</option>
    </select>
</div>

<div class="col-md-6 mb-3">
    <label>GENDER</label>
    <select name="gender" class="form-control" required>
        <option value="">Select your choice</option>
```

```
    <select name="gender" class="form-control" required>
        <option value="">Select your choice</option>
        <option value="Male">Male</option>
        <option value="Female">Female</option>
    </select>
</div>

<div class="col-md-6 mb-3">
    <label>INTERNET SERVICE TYPE</label>
    <select name="InternetService" class="form-control" required>
        <option value="">Select your choice</option>
        <option value="DSL">DSL</option>
        <option value="Fiber optic">Fiber Optics</option>
        <option value="Cable">Cable</option>
        <option value="Satellite">Satellite</option>
        <option value="Fixed wireless">Fixed Wireless</option>
    </select>
</div>

<div class="col-md-6 mb-3">
    <label>CONTRACT SIGNED FOR</label>
    <select name="Contract" class="form-control" required>
        <option value="">Select your choice</option>
        <option value="Month-to-month">Monthly</option>
        <option value="One year">Yearly</option>
        <option value="Two year">Two Years</option>
```

```html
                </select>
            </div>
        </div>

        <div class="text-center">
            <button type="submit" class="btn btn-success">Predict Churn</button>
            <button type="reset" class="btn btn-secondary ms-2">Reset</button>
        </div>
    </form>

    {% if prediction %}
        <div class="alert alert-info mt-4 text-center">
            <h4>Prediction Result: {{ prediction }}</h4>
        </div>
    {% endif %}
</div>
</body>
</html>
```

CHAPTER 6

SYSTEM TESTING:

Purpose of Testing

The primary goal of testing is to detect and eliminate bugs, verify that the system meets its specified requirements, and ensure that it behaves reliably under all expected conditions. Testing guarantees the correctness, stability, and usability of the churn prediction system from both technical and user perspectives.

6.1 TESTING STRATEGIES

Testing for this application included unit testing, integration testing, functional testing, and user acceptance testing. Manual testing was also conducted on the user interface and the backend prediction pipeline using controlled data inputs.

Test Objectives

- **The interface must render correctly and behave consistently across modern web browsers.**
- **Forms and dropdowns must handle various inputs and edge cases.**
- **User inputs should be properly validated, and errors handled gracefully.**
- **The XGBoost model must return accurate and logical churn predictions based on customer data.**
- **Prediction results and confidence scores must display correctly in the user interface.**

Features to be Tested

- **Input format validation (e.g., tenure as a number, contract type as a dropdown).**
- **Dynamic population of options in dropdowns.**
- **Correct prediction result generation after form submission.**

- **Spinner/loading behavior during processing.**
- **Error handling on missing, invalid, or out-of-range values.**

6.1.1 Unit Testing
Unit testing was applied to:

- **Data preprocessing functions (e.g., encoding categorical variables).**
- **Model loading and prediction logic (loading the serialized XGBoost model and passing test inputs).**
- **Input validation routines.**

Tests were written using unittest and manual assertions in Python to confirm that individual components functioned correctly in isolation.

6.1.2 Integration Testing
Integration testing verified:

- **Flask routes correctly render and process frontend inputs.**
- **Form data is passed to the backend and predictions returned.**
- **Templates correctly display returned JSON values.**
- **The full pipeline (input → preprocessing → prediction → output) works without interruption.**

6.1.3 Functional Testing
Functional testing ensured that:

- **The form accepts valid input combinations and generates predictions.**
- **Errors for empty or invalid fields are shown clearly.**
- **Returned predictions include both churn status and confidence score.**
- **Input fields retain values after submission if needed.**
- **Visual indicators (e.g., result badges) reflect churn risk status accurately.**

6.1.4 System Testing
System testing validated:

- **The end-to-end working of the system including Flask backend, XGBoost model, HTML templates, and Bootstrap styling.**
- **Compatibility across browsers like Chrome, Firefox, and Edge.**
- **The application runs without failure in both local and cloud-based deployments.**

6.1.5 White Box Testing
Testers examined source code to confirm:

- **Internal logic behind churn predictions and preprocessing.**
- **Edge case handling (e.g., extremely short tenure, missing demographic fields).**
- **Correct handling of optional fields like total charges.**

6.1.6 Black Box Testing
Without examining the internal code, testers:

- **Simulated various user behaviors (e.g., frequent form submissions, invalid contract type).**
- **Evaluated application responses and messages.**
- **Checked model output for logical consistency with input patterns.**

6.1.7 Acceptance Testing

User Acceptance Testing (UAT) was performed by mock telecom operators and analysts. They:

- **Used the form with real-world customer data samples.**
- **Provided feedback on usability, clarity, and response speed.**
- **Verified the prediction results matched their expectations.**

All actionable feedback was addressed and incorporated into the final version of the application.

6.2 TEST CASES

S.no	Test Case	Excepted Result	Result	Remarks(IF Fails)
1	Load Home Page	Page loads with dropdowns and form fields	Pass	—
2	Gender selection	Corresponding form value should be captured	Pass	HTML select and Flask mapping validated
3	Submitting valid inputs	Churn prediction shown (Churn / No Churn)	Pass	Model prediction and threshold logic checked
4	Missing inputs (e.g., tenure) User alerted to fill missing in	User alerted to fill missing inputs	Pass	JS form validation and Flask checks used
5	Submitting negative values	Alert shown or values not accepted	Pass	Input type and min attribute verified
6	Realistic numeric input (e.g., MonthlyCharges)	Model output adjusted based on input	Pass	Scaler and encoding logic validated
7	Spinner display behavior	Spinner shows while prediction is processing	Pass	JS and CSS validated
8	Resetting the form	All input fields and result text reset	Pass	JS reset logic reviewed
9	Model returns JSON	Flask backend responds with prediction correctly	Pass	Confirmed with API testing
10	Server handles multiple requests	App remains stable under concurrent requests	Pass	Logs reviewed for errors or timeouts

11	**Deployment behavior**	**App runs on local and cloud server without issues**	**Pass**	**Dependencies, environment, and paths tested**

CHAPTER 7

RESULTS :

7.1 MAIN SCREEN

7.2POSITIVE RESULT:

7.2 NEGATIVE RESULT:

CHAPTER 8

CONCLUSION:

In today's highly competitive telecom industry, customer retention is as crucial as customer acquisition. With the increasing availability of customer data, predictive analytics has become a powerful tool to identify behavioral patterns and anticipate customer churn. Accurate churn prediction empowers telecom companies to proactively engage at-risk customers, thereby minimizing revenue loss and enhancing customer loyalty.

In this project, we developed and implemented a Telecom Churn Prediction System that analyzes customer usage patterns and behavioral attributes to predict churn likelihood. The primary objective was to build a reliable, interactive, and scalable web-based application that enables telecom operators to forecast churn and take data-driven decisions to improve customer retention.

The system architecture consists of three key components:

Backend predictive model trained using supervised machine learning on historical customer data to detect churn risks.

Frontend interface developed using Flask and HTML/CSS, offering a responsive and user-friendly platform for inputting customer details.

Real-time output that displays churn prediction based on input data, allowing for immediate decision support.
Our approach included thorough data preprocessing such as handling categorical variables, addressing missing values, and normalizing continuous features. Feature engineering was applied to include meaningful customer behavior signals, and a robust classification model was trained, optimized, and serialized for deployment. The model was hosted using a Flask web server and supports real-time prediction with low latency.
Performance evaluation of the system demonstrated high accuracy and interpretability, with meaningful results across diverse customer profiles. The interface was tested for usability and responsiveness, ensuring that both technical and non-technical users can easily engage with the system.
This project highlights the effectiveness of integrating machine learning with real-time web applications to support business operations. It showcases how churn prediction models can be operationalized and made accessible to business users for strategic decision-making.
For future enhancements, the system could include:
Advanced models like ensemble methods or deep learning techniques for improved prediction accuracy.
Customer segmentation integration to tailor retention campaigns to specific groups.
Dashboard integration with dynamic churn heatmaps and KPIs for better visualization.
API connectivity with CRM systems and live databases to enable automated marketing triggers.
In conclusion, this Telecom Churn Prediction System serves as a practical and scalable solution for telecom companies aiming to reduce customer attrition. By combining machine learning with intuitive interfaces, the system bridges the gap between data science and real-world business applications, empowering organizations to take proactive steps toward customer retention and service improvement.

•

CHAPTER 9

FUTURE ENHANCEMENT:

While the current implementation of the Telecom Customer Churn Prediction System provides timely and accurate insights using machine learning models, several future enhancements can be pursued to improve its scalability, precision, and usability in rapidly evolving telecom environments.
One major direction for improvement involves the adoption of more advanced classification algorithms, such as:
XGBoost or LightGBM, which handle large and complex datasets efficiently,
Deep Neural Networks (DNNs) to learn complex customer behavior patterns,
Recurrent Neural Networks (RNNs) or LSTM-based architectures, which can track customer engagement over time and adapt to temporal patterns in usage data.
Another potential enhancement is the integration of automated feature engineering and hyperparameter tuning, allowing the system to dynamically identify the most predictive attributes and optimal configurations. This will make the model more robust across varying customer profiles and usage trends.
To improve user experience and accessibility:

Interactive dashboards and churn probability visualizations can be added using libraries like Plotly, Dash, or Chart.js.
A searchable customer database with detailed churn indicators and segmentation filters can improve insights for marketing and retention teams.
Deployment as a mobile-friendly progressive web application (PWA) would enable field agents and business teams to access real-time churn predictions on the go.
From a data integration perspective, the system can be extended to fetch live customer and usage data from telecom CRM platforms, billing systems, or cloud databases. This will enable continuous monitoring and allow churn predictions to be updated in near real-time.
Another valuable enhancement would be the implementation of scenario analysis, allowing business teams to simulate the effects of interventions—such as offering discounts, loyalty benefits, or personalized plans—and analyze their projected impact on churn risk.
Lastly, incorporating anomaly detection algorithms could provide early warnings of unusual churn patterns, such as spikes in specific regions, devices, or service plans. This can help telecom companies respond swiftly to potential service issues or customer dissatisfaction.
In conclusion, while the current version establishes a solid foundation for churn prediction and customer retention, future iterations of the system can be significantly enriched by adopting advanced modeling techniques, improving user interactivity, and enabling real-time data integration. These enhancements will make the application a more intelligent, scalable, and business-critical tool for customer relationship management in the telecom sector.

REFERENCES

[1] EVOASTRA :
evoastra – Innovation that flows

[2] **Churn Prediction Concepts – Towards Data Science**:
https://towardsdatascience.com/predicting-customer-churn-with-machine-learning-dc903157c5d8

[3] GitHub – Flask Web App Deployment Guide:
https://github.com/pallets/flask