Sir Syed University of Engineering & Technology (SSUET)
Department of Software Engineering

Course Name: Artificial Intelligence (SE-314L)

Semester: 5th
Batch: 2023F
Section: "C"

PROJECT REPORT

Project Title: AI-Based Customer Churn Prediction & Action Recommendation System

Submitted To:
Engr. Dur-e-Shawar Agha
Engr. Samrah Shafiq

Submitted By:

STUDENT NAME	ROLL NO
Muhammad Sumama Karim (GL)	2023F-BSE-151
Taimoor Abrar	2023F-BSE-369
Wasay Faisal	2023F-BSE-200
Ali Hussain	2023F-BSE-180

TABLE OF CONTENTS

S. No.	TOPICS	PAGE No.
1	PROBLEM DOMAIN	1
2	PROPOSED TREATMENT	2
3	PLAN OF WORK	3
4	PROJECT SCHEDULING	4
5	SOFTWARE AND HARDWARE SPECIFICATIONS	5
6	BLOCK DIAGRAM	6
7	SYSTEM FLOW DIAGRAM	7
8	USER GUIDE	8

TEAM PROFILE

1. **Muhammad Sumama Karim (GL)** **(2023F-BSE-151)**
   Project Lead & ML Model Development

2. **Taimoor Abrar** **(2023F-BSE-369)**
   Frontend Development & UI Design

3. **Wasay Faisal** **(2023F-BSE-200)**
   Data Processing & Analytics

4. **Ali Hussain** **(2023F-BSE-180)**
   Reports & Documentation

1. PROBLEM DOMAIN:

Subscription-based businesses are becoming very popular today. These include mobile network companies, streaming platforms like Netflix or YouTube Premium, and online software services known as SaaS (Software as a Service)—for example, apps like Zoom, Canva, or cloud tools. For these businesses, keeping customers is extremely important for long-term success.

Customer churn happens when a customer stops using a service and cancels their subscription. Many studies show that getting a new customer is much more expensive—5 to 25 times more costly—than keeping an existing one. This means that even a small reduction in churn can help a company save money and make much higher profits.

The main problem is that most companies use reactive strategies, meaning they take action only after it is too late. Usually, businesses only realize a customer is unhappy when they have already cancelled. Traditional data analysis methods often fail to detect complex and non-linear behavior patterns that could give early warnings about a customer leaving.

Another challenge is that even if a company knows a customer is at risk, they often don't understand the exact reason behind the customer's dissatisfaction. The customer might be upset because of high pricing, poor engagement, lack of interest, or technical problems—but without this insight, companies find it hard to offer the right solution or support.

The goal of this project is to solve these problems by building an Intelligent and Proactive Customer Churn Prediction System. This system will not only identify customers who might leave with high accuracy, but it will also explain why the prediction was made. This helps business teams trust the system and understand the issue clearly.

On top of that, the system will also give automatic and personalized business suggestions, like offering discounts, improving engagement, or resolving technical issues. These suggestions will help companies take action early and prevent churn before a customer cancels their subscription.

In simple words, this system will help businesses act faster, understand customers better, and keep more users, leading to stronger growth and more profit.

2. PROPOSED TREATMENT:

To tackle the challenges described in the Problem Domain, we propose a sophisticated AI solution that combines Machine Learning (ML) for prediction, Explainable AI (XAI) for transparency, and a Rule-Based Expert System for actionable recommendations. This solution is designed as a desktop application, allowing business analysts to interact easily with advanced models without needing deep technical knowledge.

1. Predictive Modeling Core

At the heart of the system is a robust machine learning pipeline. The model uses and compares multiple algorithms, such as Logistic Regression and Random Forest Classifiers, to identify the most effective method for predicting customer churn. The model analyzes ten important behavioral and demographic features, including days since the last login, payment failures, watch time, support calls, and customer tenure. Based on these features, the system calculates a churn probability score ranging from 0 to 100 percent, indicating the likelihood of a customer leaving the service. This allows businesses to identify at-risk customers early and take proactive measures to retain them.

2. Explainability Layer (SHAP Integration)

One of the main challenges with AI is that users often do not understand why a model makes a specific prediction. To solve this, our system integrates Shapley Additive explanations (SHAP). SHAP evaluates the contribution of each feature to the final prediction and generates a clear, personalized explanation for every customer. For example, the system might report that a customer is at high risk because their payment failed twice and they have not logged in for 45 days. This transparency builds trust in the AI system and enables non-technical staff to understand the root causes of customer dissatisfaction.

3. Action Recommendation Engine

Predictions are only useful if they lead to real actions. Our system includes a logic-based recommendation engine that converts the churn probability and identified risk factors into practical business strategies. For instance, if a customer is at high risk due to price sensitivity, the system may recommend offering a targeted discount. If the risk is caused by low engagement, it could suggest launching a content re-engagement campaign. This integration ensures that insights from data analysis are immediately translated into actionable steps, helping businesses prevent churn before customers cancel their subscriptions.

3. PLAN OF WORK:

The development of this project follows a structured System Development Life Cycle (SDLC) approach, divided into six distinct phases to ensure robustness and quality:

Phase 1: Requirement Analysis & Data Simulation:
We began by defining the key factors that influence customer churn in a subscription context. Since real-world proprietary data is often inaccessible, we developed a sophisticated data generation script ('generate_data.py') to create a synthetic dataset of 5,000 customers. This dataset was engineered to contain realistic correlations (e.g., users with more payment failures are more likely to churn) to ensure the model learns meaningful patterns.

Phase 2: Data Preprocessing & Exploratory Data Analysis (EDA):
Raw data is rarely ready for modeling. This phase involved cleaning the data, handling missing values, and performing feature engineering. Categorical variables like 'Gender' and 'Subscription Type' were transformed using Label Encoding, and numerical features were normalized using Standard Scaling to ensure that algorithms like Logistic Regression perform optimally.

Phase 3: Model Training & Evaluation:
We implemented a training pipeline to test multiple algorithms. The dataset was split into training (80%) and testing (20%) sets. We evaluated models based on Accuracy, Precision, Recall, and F1-Score. The best-performing model was then serialized (saved) using 'joblib' for later use in the application.

Phase 4: Explainability Implementation:
We integrated the SHAP library to interpret the trained model. This involved creating an explainer object that could take a specific data point and decompose the prediction into feature importance values, distinguishing between factors that push risk up versus those that pull it down.

Phase 5: User Interface (UI) Development:
To make the system accessible, we developed a modern Graphical User Interface (GUI) using Python's Tkinter framework. The UI was designed with a focus on User Experience (UX), featuring a sidebar navigation, clean forms for data entry, and visual indicators for risk levels.

Phase 6: Integration & Testing:
The final phase involved connecting the backend ML models with the frontend UI. Rigorous testing was conducted to ensure that inputs from the UI were correctly processed, predictions were accurate, and the recommendations displayed were logically consistent with the risk factors.

4. PROJECT SCHEDULING:

To ensure the timely and organized execution of the project, a detailed schedule has been prepared, outlining all tasks, their durations, and the team members responsible for each activity. The Gantt chart below (Figure 4.1) visually represents the project timeline, showing the sequence of tasks, overlapping activities, and deadlines for each phase of the AI Customer Churn Prediction project.

AI Customer Churn Prediction Project Timeline

Task List	Week 1	Week 2	Days	Person
Phase 1: Data Collection & Preparation				
Gather customer dataset	Week 1	Week 1	5 days	Team
Data cleaning & preprocessing	Week 1	Week 2	7 days	Taimoor Abrar
Exploratory Data Analysis (EDA)	Week 2	Week 2	5 days	Abdul Wasay
Feature engineering	Week 2	Week 3	5 days	Ali Hussain
Phase 2: Model Development				
Implement Logistic Regression	Week 3	Week 3	4 days	Sumama Karim
Implement Naive Bayes	Week 3	Week 4	4 days	Taimoor Abrar
Implement Random Forest	Week 4	Week 4	4 days	Abdul Wasay
Model training & evaluation	Week 4	Week 5	5 days	Team
Select best model	Week 5	Week 5	2 days	Sumama Karim
Phase 3: UI Development				
Design UI wireframes	Week 5	Week 5	3 days	Ali Hussain
Create Dashboard page	Week 5	Week 6	4 days	Sumama Karim
Create Predict page	Week 6	Week 6	4 days	Taimoor Abrar
Create Upload & Data Results pages	Week 6	Week 7	5 days	Abdul Wasay
Create Analytics page	Week 7	Week 7	4 days	Sumama Karim
Create Reports page (PDF export)	Week 7	Week 8	5 days	Ali Hussain
Create Settings page	Week 8	Week 8	3 days	Taimoor Abrar
Phase 4: Integration & Testing				
Integrate ML model with UI	Week 8	Week 9	5 days	Sumama Karim
SHAP explainability integration	Week 9	Week 9	4 days	Abdul Wasay
Unit testing	Week 9	Week 10	5 days	Team
Bug fixes & optimization	Week 10	Week 10	5 days	Team
Phase 5: Documentation & Delivery				
Write project documentation	Week 10	Week 11	5 days	Ali Hussain
Create user manual	Week 11	Week 11	3 days	Taimoor Abrar
Final testing & review	Week 11	Week 12	5 days	Team
Project submission	Week 12	Week 12	2 days	Sumama Karim (GL)

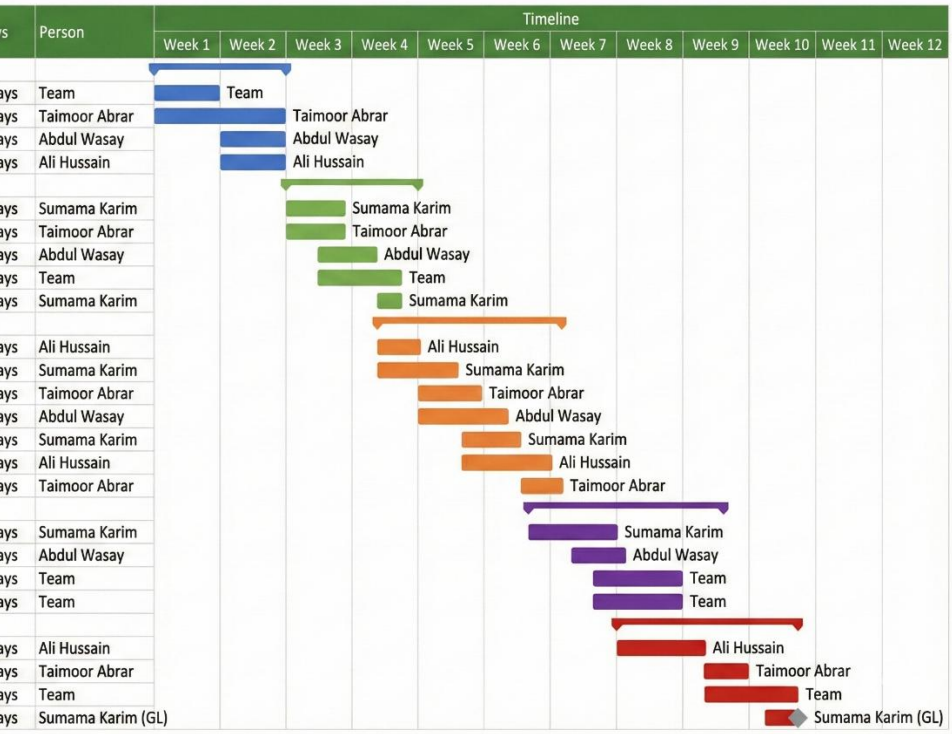


Figure 4.1: Gantt Chart

5. SOFTWARE AND HARDWARE SPECIFICATIONS:

The project relies on a modern, open-source technology stack selected for its reliability, performance, and ease of deployment.

Software Requirements:

1. Programming Language: Python 3.8+

Chosen for its extensive ecosystem of data science and machine learning libraries.

2. Core Libraries & Frameworks:

- Pandas & NumPy: Utilized for high-performance data manipulation, structure, and numerical analysis.
- Scikit-Learn: The primary machine learning library used for training algorithms (Logistic Regression, Random Forest), preprocessing data, and evaluating metrics.
- SHAP (SHapley Additive exPlanations): A game-theoretic approach to explain the output of the machine learning model.
- Joblib: Used for object serialization, allowing us to save the trained model and encoders to disk and load them instantly during application runtime.
- Tkinter: Python's standard GUI library, used to create the desktop application interface without requiring a web browser.

3. Development Environment:

- Visual Studio Code or PyCharm as the Integrated Development Environment (IDE).

Hardware Requirements:

To ensure smooth training and inference, the following minimum specifications are recommended:

- Processor: Intel Core i5 or equivalent (Minimum 2.0 GHz)
- RAM: 8 GB or higher (to handle dataset loading and model training in memory)
- Storage: 500 MB free space (for environment, libraries, and datasets)
- Display: 1366x768 resolution or higher for optimal UI viewing.

6. BLOCK DIAGRAM:

The block diagram illustrates the overall architecture of the AI Customer Churn Prediction system, showing how data flows through different modules from data collection and preprocessing to prediction, explainability, and actionable recommendations. It provides a clear visual overview of the system's components and their interactions.

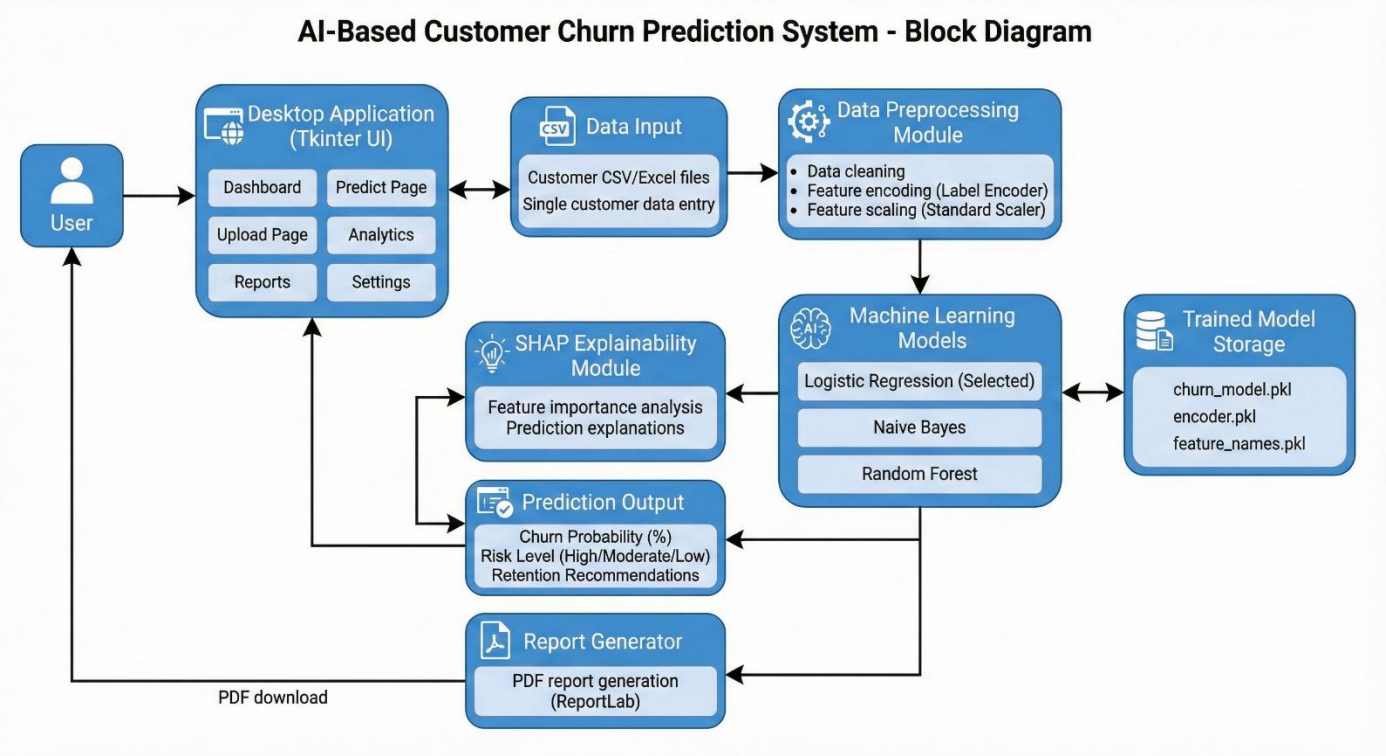


Figure 6.1: Block Diagram

7. SYSTEM FLOW DIAGRAM:

The system flow diagram depicts the step-by-step process of the AI Customer Churn Prediction system, showing how data moves through the system from input to output. It outlines the sequential flow starting from customer data collection, followed by preprocessing, feature extraction, machine learning-based prediction, explainability through SHAP, and finally generating actionable recommendations. This diagram helps to visualize the dynamic interactions between different modules, making it easier to understand the system's operational workflow and how each component contributes to predicting and preventing customer churn.

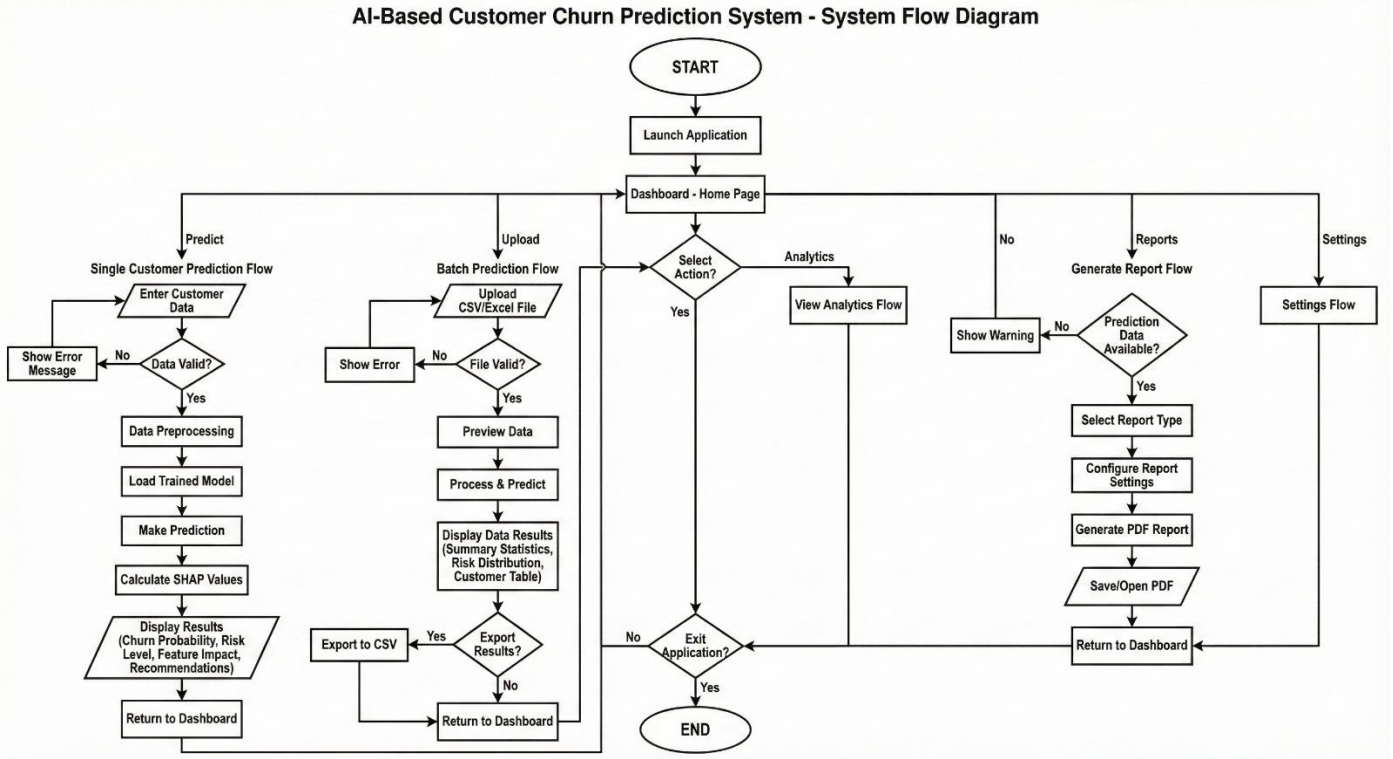


Figure 7.1: System Flow Diagram

8. USER GUIDE:

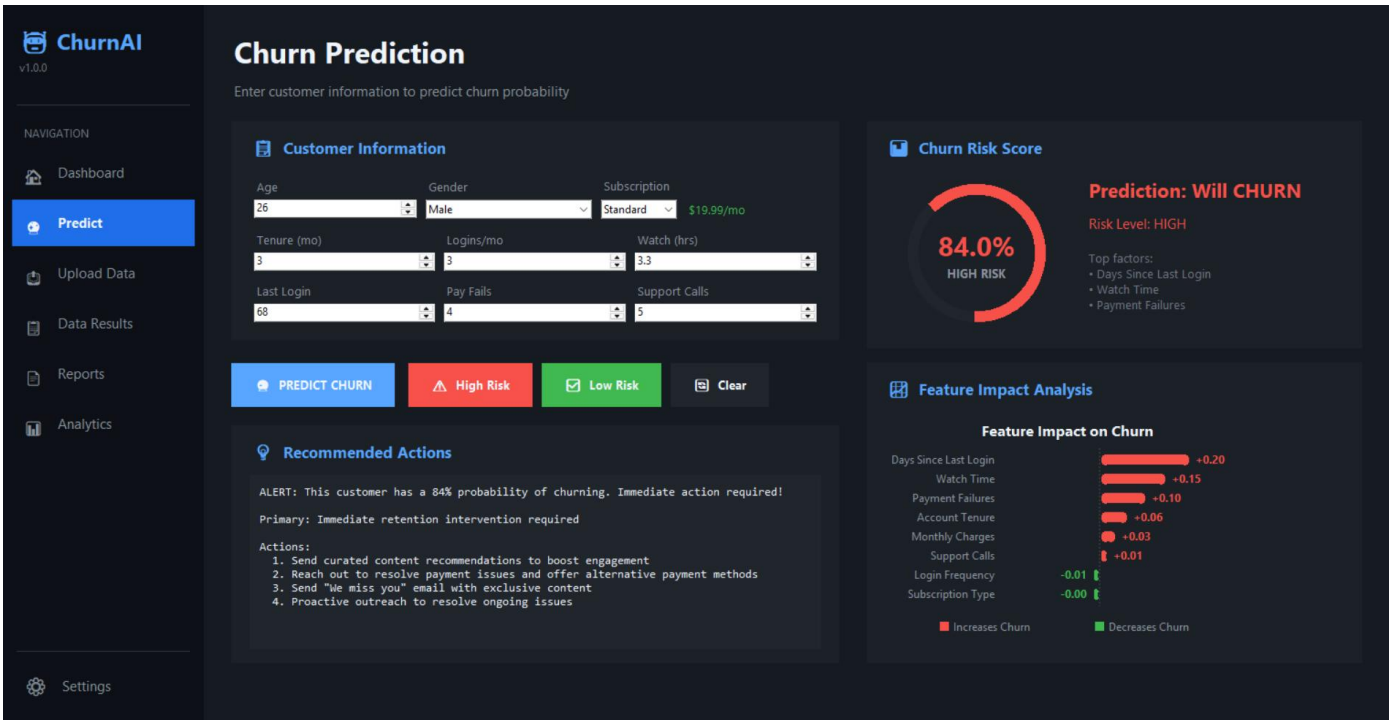
This User Guide provides a comprehensive walkthrough of the AI-Based Customer Churn Prediction System. The application is divided into several modules, each accessible via the left-hand navigation sidebar.

1. Dashboard Overview (Home Page):

- Upon launching the application ('python ui/app.py'), you are greeted by the Home Dashboard.
- This page serves as the central hub, providing a brief overview of the system's capabilities and quick access buttons to the main features (Predict, Upload, Reports).

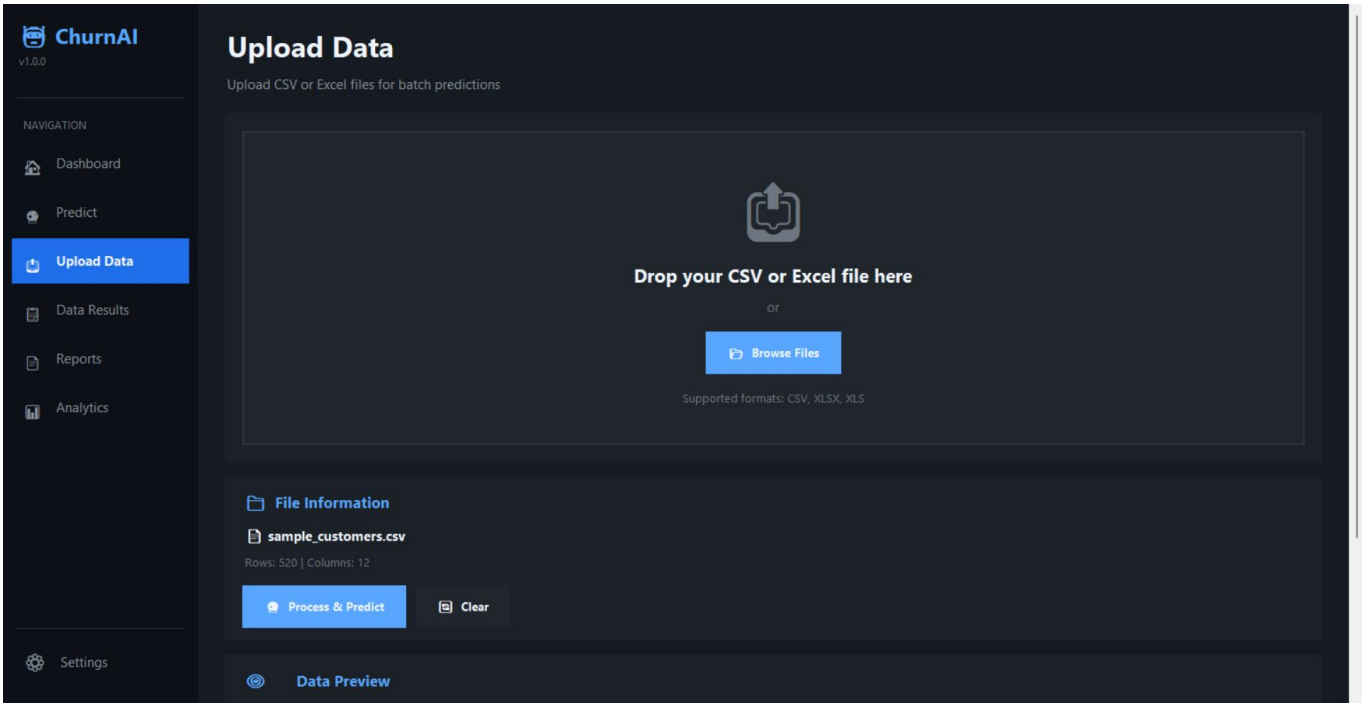
2. Individual Customer Prediction (Predict Page):

- Navigate to the 'Predict' tab to analyze a single customer.
- Input Data: Fill in the customer details manually (Age, Tenure, Monthly Charges, etc.) or click 'Load Sample Data' to auto-populate the form for testing.
- Analyze: Click 'Predict Churn'. The AI analyzes the data in real-time.
- Results: The system displays the Churn Probability (%), Risk Level (Low/Med/High), and SHAP-based explanations ('Why this prediction?').
- Action Plan: Review the tailored business recommendations at the bottom (e.g., 'Offer 20% Discount').



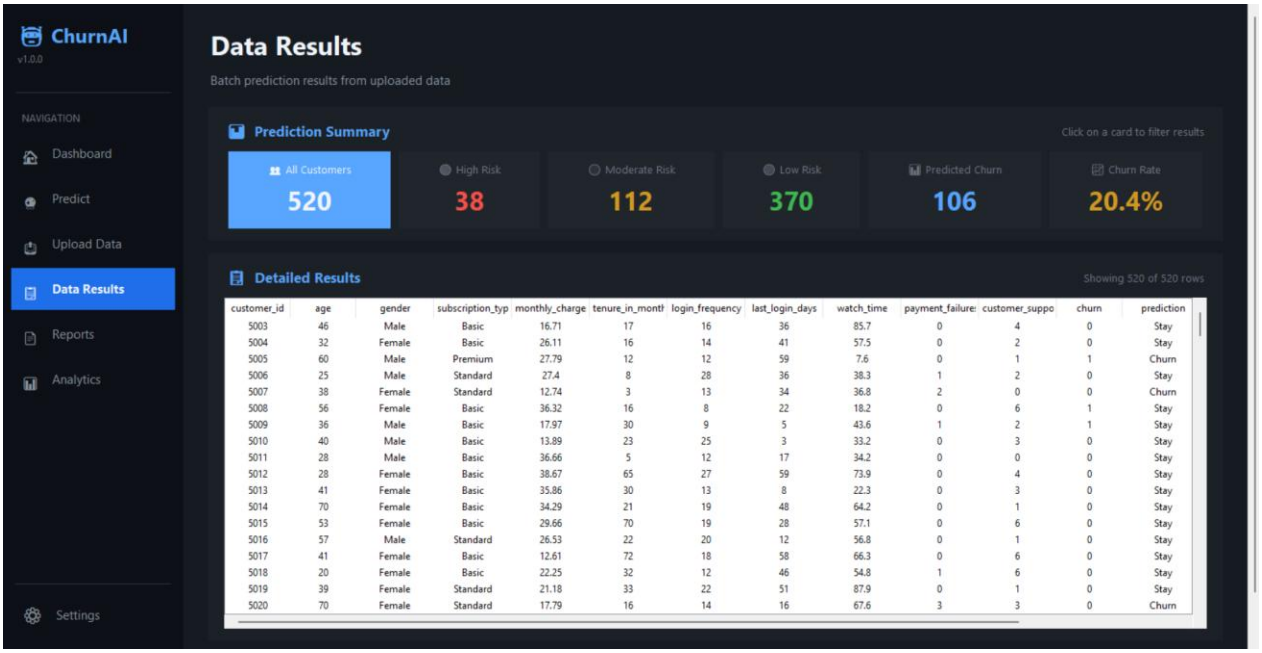
3. Batch Data Processing (Upload Page):

- For analyzing multiple customers at once, go to the 'Upload' tab.
- Click 'Browse' to select a CSV file containing customer records.
- The system validates the file and processes all records simultaneously.
- Results are displayed in a tabular format, showing the predicted risk for every customer in the file.



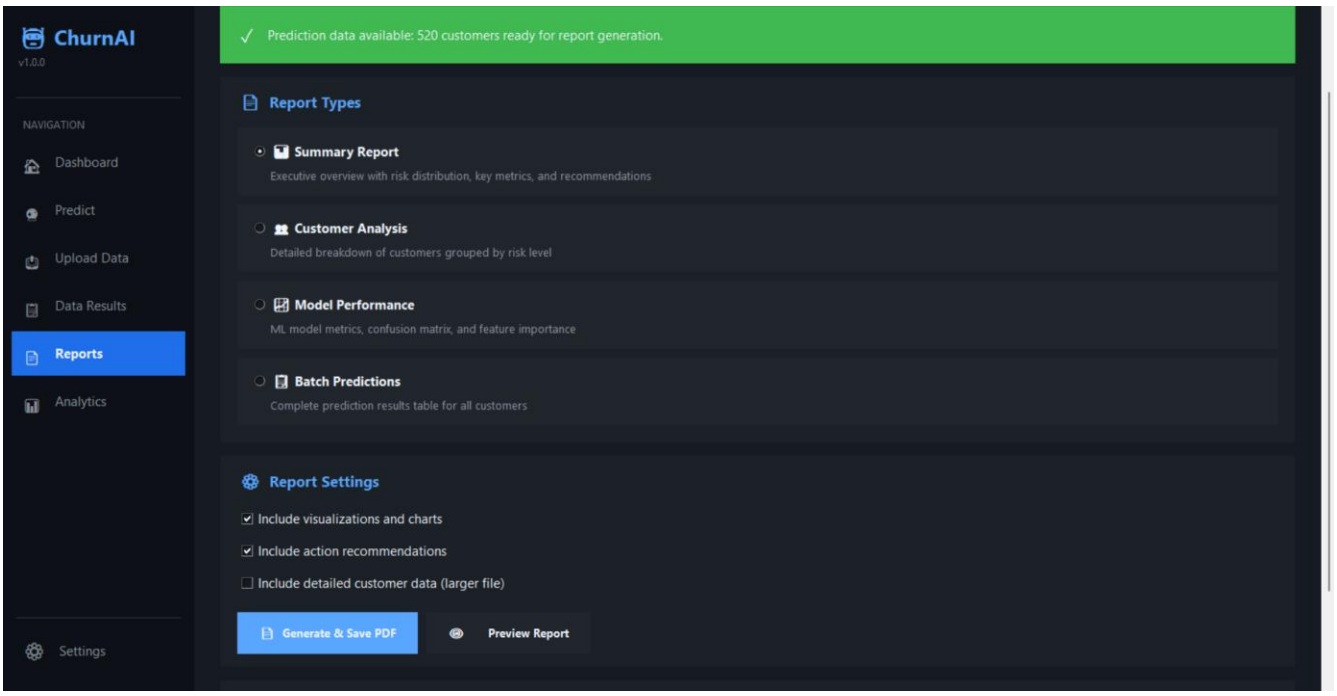
4. Data Analytics & Visualization (Charts Page):

- The Data Results Page shows a clear summary of churn predictions with top cards displaying total customers and risk levels (high, moderate, low). Users can filter the results table by risk category to view detailed information, including Customer ID, Churn Probability, Prediction, and Risk Level. An Export to CSV option allows downloading the results for further analysis.



5. Report Generation (Reports Page):

- Navigate to the 'Reports' tab to document findings.
- You can generate a comprehensive PDF or Text report summarizing recent predictions and overall system performance.
- This feature is essential for sharing insights with stakeholders who do not have access to the app.

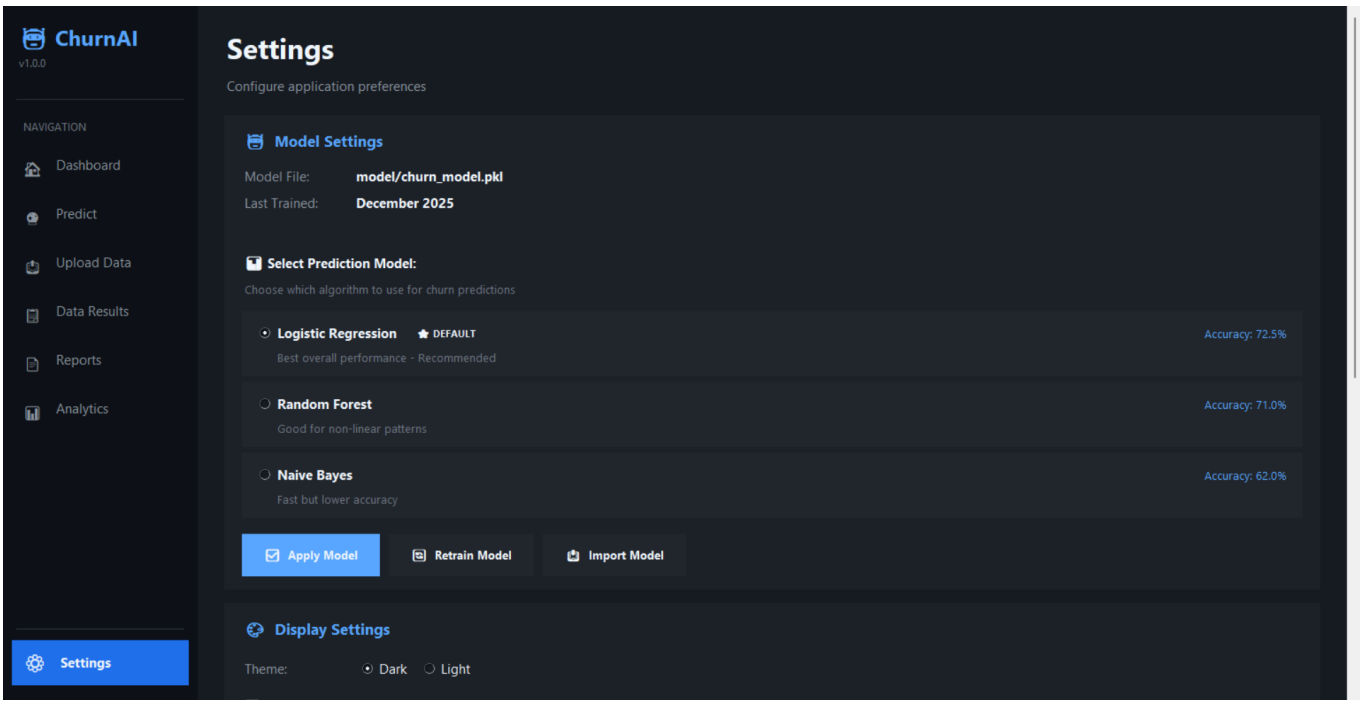


6. Model Performance Analytics (Analytics Page):

The "Analytics" tab provides transparency into the AI's performance. You can switch between different algorithms (Logistic Regression, Random Forest) to see their Accuracy, Precision, and Recall charts. It also features a Confusion Matrix to show how many correct vs. incorrect predictions were made during testing.

7. System Configuration (Settings Page):

- The 'Settings' tab allows you to customize the AI's behavior.
- Model Selection: You can switch the active machine learning model (e.g., from 'Logistic Regression' to 'Random Forest') to compare performance or use a more complex algorithm.



REFERENCES

Libraries & Frameworks

1. **scikit-learn** - Machine Learning Library
   Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
   https://scikit-learn.org/

2. **SHAP** - Explainable AI
   Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30.
   https://github.com/slundberg/shap

3. **Pandas** - Data Manipulation
   McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*.
   https://pandas.pydata.org/

4. **NumPy** - Numerical Computing
   Harris, C. R., et al. (2020). Array programming with NumPy. *Nature*, 585, 357-362.
   https://numpy.org/

5. **ReportLab** - PDF Generation
   https://www.reportlab.com/

6. **Tkinter** - Python GUI Framework
   https://docs.python.org/3/library/tkinter.html

Machine Learning Algorithms

7. **Logistic Regression**
   Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). Wiley.

8. **Random Forest**
   Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.

9. **Naive Bayes**
   Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Customer Churn Research

10. **Customer Churn Prediction**
Verbeke, W., et al. (2012). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 39(12), 10972-10984.

11. **Churn Analysis in Subscription Services**
Hadden, J., et al. (2007). Computer assisted customer churn management. *Computers & Operations Research*, 34(10), 2902-2917.

Online Resources

12. **Python Documentation**
https://docs.python.org/3/

13. **scikit-learn User Guide**
https://scikit-learn.org/stable/user_guide.html

14. **Stack Overflow** - Programming Q&A
https://stackoverflow.com/

15. **GitHub** - Code Repository
https://github.com/

INSTRUCTIONS

1. You have to implement AI based algorithms or libraries in your project.
2. Try to implement the project on your own, must not be cheat or copy from anywhere and must believe that you will achieve the target!
3. The font size should be 14 for heading and 12 for the rest of text, font style should be Times New Roman. The font color should be black only.
4. The header must contain the heading “Artificial Intelligence Project Report” in the middle and the footer must contain the heading “SIR SYED UNIVERSITY OF ENGINEERING AND TECHNOLOGY” on left side and page numbers on the right side through out in the report. The page number should start from “Problem Domain”.
5. There can be only single line spacing between the lines.