An
Industry Oriented Mini Project Report
on

BANK CUSTOMER CHURN PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of degree

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted By

J.VAMSHI 227Z1A1224

Under the Guidance of
Mrs. S.Radhika
Assistant Professor

SCHOOL OF ENGINEERING
Department of Information Technology

NALLA NARASIMHA REDDY
EDUCATION SOCIETY'S GROUP OF INSTITUTIONS
(AN AUTONOMOUS INSTITUTION)

Approved by AICTE, New Delhi, Chowdariguda (V) Korremula 'x'Roads, via Narapally, Ghatkesar (Mandal) Medchal (Dist), Telangana-500088
2024-2025

NALLA NARASIMHA REDDY
Education Society's Group of Institutions - Integrated Campus
(UGC AUTONOMOUS INSTITUTION)

SCHOOL OF ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report titled **"BANK CUSTOMER CHURN PREDICTION USING MACHINE LEARNING"** is being submitted by **J.Vamshi (227Z1A1224)** in Partial fulfillment for the award of **Bachelor of technology in Information Technology** is a record bonafide work carried out by him. The results embodied in this report have not been submitted to any other University for the award of any degree.

Internal Guide
(Mrs. S.Radhika)

Head of the Department
(Dr.K.Rameshwaraiah)

Submitted for the University Examination held on…………………………….

External Examiner

DECLARATION

Iam J.Vamshi of **Bachelor of Technology in Information Technology, Nalla Narasimha Reddy Education Society's Group of Institutions**, Hyderabad, Telangana, hereby declare that the work presented in this project work entitled **BANK CUSTOMER CHURN PREDICTION USING MACHINE LEARNING** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of engineering ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning.

J.Vamshi 227Z1A1224

Date:

Signature:

ACKNOWLEDGEMENT

I express our sincere gratitude to our guide **Mrs.S.Radhika,** Assistant Professor, Department of Computer Science and Engineering, NNRESGI, who motivated throughout the period of the project and also for his valuable and intellectual suggestions, guidance, and constant encouragement right throughout our work.

I would like to express our profound gratitude to our project coordinator **Mr.T. Madhu**, Associate Professor, Department of Computer Science and Engineering, NNRESGI, for his support and guidance in completing our project and for giving us this opportunity to present the project work.

I profoundly express thanks to **Dr. K. Rameshwaraiah,** Professor & Head, Department of Computer Science and Engineering, NNRESGI, for his cooperation and encouragement in completing the project successfully.

I wish to express our sincere thanks to **Dr. G. Janardhana Raju,** Dean School of Engineering, NNRESGI, for providing the facilities for completion of the project.

I wish to express our sincere thanks to **Dr**. **C. V. Krishna Reddy,** Director, NNRESGI, for providing the facilities for completion of the project.

Finally, I would like to thank Project Review Committee (PRC) members, all the faculty members and supporting staff, Department of Computer Science and Engineering, NNRESGI, for extending their help in all circumstances.

By

J.Vamshi 227Z1A1224

ABSTRACT

Customer churn prediction is essential for the growth and sustainability of organizations, especially in the banking sector, where retaining customers is more cost-effective than acquiring new ones. Churn negatively impacts revenue and profitability, making early detection and understanding of its causes crucial for effective prevention strategies. One major challenge in churn prediction is class imbalance—where churned customers are significantly fewer than non-churned ones—which often leads to poor performance in traditional machine learning models. Borderline-SMOTE, an advanced version of the Synthetic Minority Over-sampling Technique, addresses this issue by creating synthetic samples of minority class instances near the decision boundary, where misclassification is most likely. This method enhances the model's ability to generalize and improves classification accuracy. When combined with Gradient Boosting Machines (GBM), a powerful ensemble method that builds strong learners from decision trees, the resulting model becomes highly effective in differentiating churned from non-churned customers, even in imbalanced datasets. Empirical evidence shows that this integration significantly boosts predictive accuracy compared to other models. Therefore, using Borderline-SMOTE with GBM provides banks with a superior approach to churn prediction, enabling them to craft more targeted retention strategies and maintain a competitive advantage in today's highly saturated and dynamic financial market.

Keywords— *Machine Learning, Gradient Boosting Machines (GBM), Synthetic Minority Over sampling Technique (SMOTE), Borderline-SMOTE, Customer Churn*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S. No.	Abbreviation	Definitions
1	GBM	Gradient Boosting Machine
2	SMOTE	Synthetic Minority Over-sampling Technique
3	UML	Unified Modelling Language
4	API	Application Programming Interface

1. INTRODUCTION

Customer churn prediction is crucial in the banking sector, where retaining existing customers is more cost-effective than acquiring new ones. A major challenge is the class imbalance in datasets, with churned customers being much fewer than those who stay, leading to skewed model performance. This project addresses the issue using Borderline-SMOTE, a technique that generates synthetic samples of churners near the decision boundary. This targeted oversampling balances the dataset, enhancing the model's ability to detect churn accurately. Improved prediction reduces misclassifications, enabling timely interventions and supporting effective retention strategies, ultimately boosting profitability and long-term customer loyalty.

1.1 Motivation

The motivation for this project is to improve customer retention in the banking sector, where retaining existing customers is more cost-effective than acquiring new ones. Accurately predicting churn is challenging due to class imbalance in datasets. By applying Borderline-SMOTE, the project aims to create a balanced dataset, enhance model performance, and enable early identification of potential churners. This allows for timely interventions, reducing customer loss and increasing long-term profitability and customer loyalty.

1.2 Problem Definition

Customer churn is a major concern in the banking sector, as retaining existing customers is significantly more cost-effective than acquiring new ones. However, accurately predicting churn is challenging due to class imbalance in the data—churned customers are much fewer than those who stay. This imbalance can lead to biased models that fail to identify customers at risk of leaving, resulting in lost revenue and missed retention opportunities. The key problem is developing a predictive model capable of handling this imbalance and improving accuracy. This project addresses the issue using Borderline-SMOTE, a technique that creates synthetic samples of the minority class near the decision boundary. By balancing the dataset, the model can better learn distinguishing patterns between churners and non-churners, leading to improved prediction performance. Ultimately, this approach supports more effective customer retention strategies and enhances overall profitability.

1.3 Objective of Project

This project has two primary objectives. First, it aims to develop a robust customer churn prediction model tailored for the banking sector, where early identification of at-risk customers is critical for retention and sustained growth. By addressing the common issue of class imbalance using Borderline-SMOTE, the model ensures better representation of churned customers, improving detection accuracy. Second, it seeks to enhance predictive performance through the integration of Gradient Boosting Machines (GBM), a powerful ensemble learning technique known for handling complex patterns in data. This combination enables more reliable forecasting, allowing banks to implement proactive, data-driven retention strategies. Ultimately, the project aspires to support financial institutions in reducing revenue loss and improving customer satisfaction through intelligent, timely interventions.

1.4 Limitations of Project

Bank customer churn prediction is designed to help the firm to predict customers, who are going to be churn. However, these systems have some limitations that need to be considered.

1.4.1 Data Quality and Availability

The effectiveness of churn prediction models heavily depends on the quality and completeness of customer data. Inconsistent, outdated, or missing information can lead to inaccurate predictions and reduced model performance. Additionally, acquiring comprehensive behavioral and transactional data from various banking systems can be challenging due to privacy concerns and integration issues. Ensuring data reliability is crucial for the model to make meaningful predictions.

1.4.2 Model Drift and Concept Drift

As time passes, customer behavior, market dynamics, or policies may change, leading to model drift (decline in performance over time). Since your model is static and does not retrain periodically, its predictions will become outdated without manual updates.

1.4.3 Computational Complexity

The combined use of Borderline-SMOTE and GBM increases the computational requirements for training the model. Synthetic sample generation and iterative boosting require significant processing power and memory, especially on large datasets. This can slow down the model development cycle and make deployment more resource-intensive, particularly for banks with limited IT infrastructure

In conclusion, while churn prediction models using Borderline-SMOTE and GBM face certain limitations, they remain a powerful tool for identifying at-risk customers in the banking sector. By addressing these limitations, financial institutions can enhance model accuracy and performance, ultimately supporting smarter retention strategies and ensuring sustained customer engagement..

2. LITERATURE SURVEY

2.1 Introduction

Firstly,researchers have concentrated on improving churn prediction accuracy through advanced machine learning techniques, particularly in the banking sector. Various studies highlight the use of ensemble methods like Gradient Boosting Machines (GBM), known for their robustness and ability to model complex patterns in customer behavior. Secondly, considerable research has been devoted to addressing class imbalance—a critical issue in churn datasets where churners represent a small fraction. Techniques such as Borderline-SMOTE have emerged to generate synthetic samples near decision boundaries, improving model sensitivity to minority classes. Additionally, literature emphasizes the importance of feature engineering, with studies exploring demographic, transactional, and behavioral variables to enhance predictive power. Data preprocessing and selection techniques are often detailed to ensure quality inputs. Researchers also explore model evaluation strategies beyond accuracy, using precision, recall, F1-score, and AUC-ROC to better assess performance under imbalance. Real-world implementations in banking provide practical insights, with case studies revealing how churn models influence marketing and retention strategies. Some studies further examine the ethical and privacy implications of using customer data, ensuring transparency and compliance. In summary, the literature survey reveals growing attention to sophisticated modeling techniques, class imbalance handling, feature optimization, and real-world validation in churn prediction. These findings highlight the ongoing commitment to building accurate, actionable tools for customer retention in competitive financial environments.

2.2 Existing System

The existing system employs the Random Forest Classifier as the primary technique for predicting customer churn. Random Forest is an ensemble method that builds multiple decision trees and aggregates their outputs to improve predictive accuracy. It is widely used due to its robustness and ability to handle large datasets, making it a popular choice for classification problems, including churn prediction in the banking sector.

Despite its advantages, Random Forest has certain limitations when applied to customer churn prediction. One key challenge is its difficulty in handling class imbalance, where the minority class (churned customers) is often underrepresented compared to the majority class (non-churned customers). This imbalance can cause the model to favor the majority class, resulting in poor identification of customers who are actually likely to churn.

Another drawback of the Random Forest algorithm is its tendency to overfit, especially when dealing with complex datasets containing noisy or intricate patterns. Overfitting reduces the model's generalization ability, meaning it performs well on training data but poorly on unseen data. This limitation is critical in churn prediction, where the model needs to reliably predict customer behaviour on future, unseen cases.

Furthermore, Random Forest's performance can be affected by the choice of hyperparameters and the quality of feature engineering. Without careful tuning and preprocessing, the model may fail to capture subtle relationships in the data. Additionally, the model's interpretability can be challenging, which may limit understanding and trust from stakeholders relying on these predictions to make business decisions.

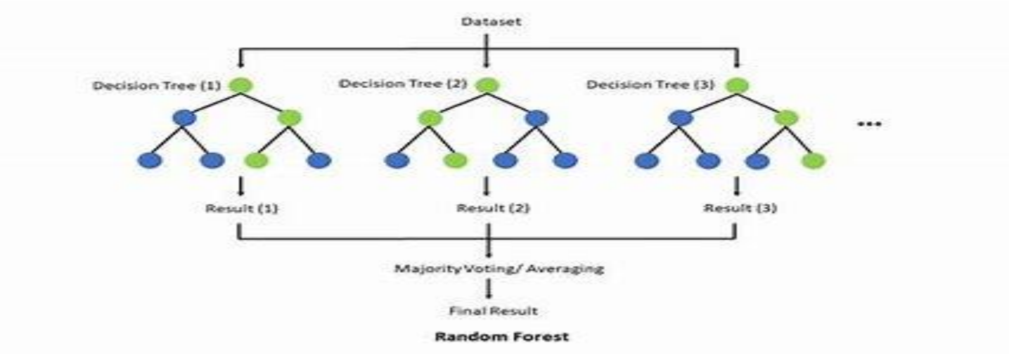


Fig 2.2: Existing System

2.2.1 Constraints of Existing System

- **Class Imbalance Issues:**

  Class imbalance happens when one class significantly outnumbers another, causing biased based model.

- **Overfitting:**

  Overfitting occurs when a model learns noise and details from training data, reducing its performance on new data.

2.3 Proposed System

Our proposed system focuses on improving customer churn prediction in the banking sector, where retaining customers is more cost-effective than acquiring new ones. Churn directly impacts profitability, so early detection and understanding of its causes are essential for designing effective customer retention strategies.

Class Imbalance Handling: One major challenge in churn prediction is class imbalance—where the number of customers who churn is much smaller than those who stay. This often leads to poor performance in traditional machine learning models. To overcome this, our system uses Borderline-SMOTE, an advanced data sampling technique that generates synthetic examples near the decision boundary, where errors are more likely. This improves the model's ability to correctly identify customers at risk of churning.

Predictive Modeling with GBM: To enhance prediction accuracy, our system integrates Gradient Boosting Machines (GBM)—a powerful ensemble learning method that builds strong models from decision trees. GBM is highly effective for structured data and performs well even on imbalanced datasets.

User-Friendly Interface: The user interface of our churn prediction system is designed to be intuitive and easy to use for analysts and business decision-makers. With a clear dashboard and visual tools, users can effortlessly interpret churn trends, model outcomes, and key risk indicators. This simplicity empowers non-technical stakeholders to make data-driven decisions with confidence.

Enhanced Churn Management: Our system streamlines churn management by accurately identifying customers who are at risk of leaving. By leveraging voice-assisted reports or visual cues, decision-makers can monitor high-risk segments, prioritize customer outreach, and track engagement outcomes more effectively—making customer retention efforts more strategic and efficient.

Addressing Class Imbalance: Your system's design places accessibility and inclusivity at the forefront. It empowers visually challenged individuals to engage fully in digital communication by eliminating barriers and providing an alternative, user-friendly means of interacting with email.

Powerful Predictive Modeling: Using Gradient Boosting Machines (GBM), our system delivers robust performance in identifying churn patterns. GBM's ensemble approach enhances accuracy, even in imbalanced datasets, and ensures that the system adapts well to real-world banking data.

In conclusion,our proposed churn prediction system combines advanced modeling techniques with a user-friendly interface, improving both churn detection and decision-making. It helps banks retain valuable customers and build stronger, more loyal relationships through timely, data-informed actions.

.

3. SYSTEM ANALYSIS

3.1 Functional Requirements

Functional requirements are the following:

- The system should be able to take customer data as input for processing and analysis.
- The application should implement Borderline-SMOTE to handle class imbalance.
- The application should provide churn prediction and highlight risk customers.
- The application should Gradient Boosting Machines to train model.
- The application should allow users to view and export reports for further analysis.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

The proposed churn prediction system will be used by data analysts, marketing teams, or business managers. Therefore, it is expected that the system can perform the following:

- Accuracy- The system should high prediction accuracy, especially for identifying customers likely to churn.
- Capacity- One dataset at a time per session to ensure model performance.

3.2.2 Safety Requirements

Only after successful login can users access system features. Users should not share credentials with unauthorized individuals to maintain data security.

3.2.3 Security Requirements

User login- The user can use the services only if they provide the valid user credentials. Other than that, no one is allowed to use this system.

3.2.4 User Interface

The user can interact with the system through a clean, dashboard-style interface. Users need to follow on-screen instructions to upload data and view predictions.

3.3 Software Requirements

1. Operating System: Windows7 or higher versions
2. Web browser: Chrome (109 or higher version), Microsoft Edge, Mozilla Firefox
3. Software: Visual Studio
4. Programming language: Python

3.4 Hardware Requirements

1. Processor: Pentium or intel
2. Hard disk: 40GB (variable)
3. RAM: 4GB

3.5 Block Diagram Of The Proposed System

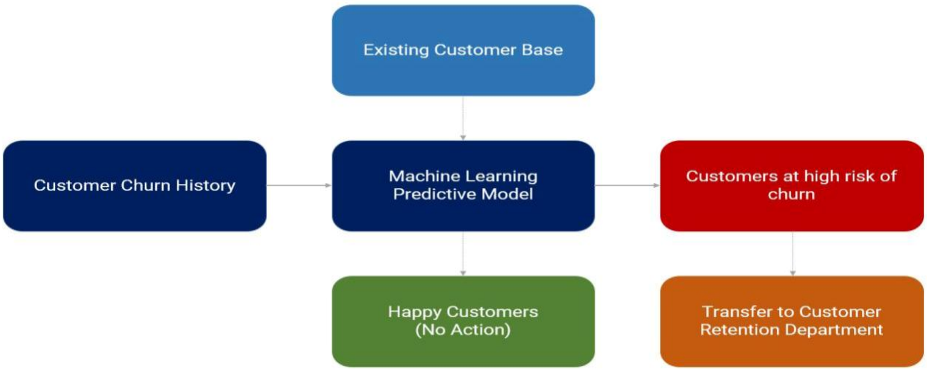


Fig 3.5: Block diagram of the system

The figures illustrate the flow of a customer churn prediction system designed to help banks identify and retain at-risk customers. The system is user-friendly and built around data-driven insights to support customer retention strategies. It begins with a customer database that stores user profiles, transaction history, and interaction records. Using this data, the system analyzes churn history to train a predictive model capable of identifying customers who are likely to leave the service.

At the core of the system is a machine learning model trained using Gradient Boosting Machines (GBM), enhanced with Borderline-SMOTE to address class imbalance. This ensures the model accurately learns from both churned and retained customer examples. When new customer data is fed into the system, the model processes the input and classifies customers as either likely to churn or not at risk.

Customers predicted to stay happy customers require no immediate action and remain in the standard engagement flow. However, those identified as likely to churn are flagged by the system and their details are transferred to the retention department. This enables the team to initiate proactive measures, such as personalized offers, feedback collection, or customer support interventions aimed at improving satisfaction and reducing churn risk.

The system includes an intuitive dashboard that displays predictions, risk scores, and actionable insights in a clear format. Security features ensure that only authorized personnel can access customer data and model outputs. The interface includes step-by-step guidance for uploading data, training the model, and interpreting the results, making it accessible even to non-technical users.

By automating churn prediction and integrating retention workflows, the system empowers banks to act early, reduce customer loss, and improve long-term loyalty. This practical, streamlined approach makes customer management more effective in a highly competitive financial environment.

4. SYSTEM DESIGN

4.1 Introduction

The system design for our innovative project takes a strategic approach to transforming customer retention in the banking sector through predictive analytics. Traditional, reactive methods are replaced with a data-driven system that uses Borderline-SMOTE to address class imbalance and Gradient Boosting Machines (GBM) to deliver accurate churn predictions. The goal is to build a smart and efficient platform that identifies customers at risk of leaving and enables timely action. By analyzing customer history and behaviour, the model predicts churn and flags high-risk individuals. These customers are then forwarded to the retention department for personalized engagement, while satisfied customers remain in the regular flow. The system empowers banks to act before customer loss occurs, improving loyalty and retention. This design not only enhances decision-making but also fosters a shift toward proactive customer relationship management. With its focus on precision, accessibility, and integration, the system offers a powerful tool for maintaining competitive advantage in a rapidly evolving financial landscape.

4.2 UML Diagrams

UML, which stands for Unified Modelling Language, is a way to visually represent the architecture, design, and implementation of complex software systems. Looking at code and understanding the system is difficult, instead we can use UML diagrams to understand the system. UML diagrams can be used to explain the components of the software to people who don’t have technical knowledge.

It is a standard language for specifying, visualizing, constructing, and documenting the artefacts of the software systems. UML is different from other common programming languages like C++, Java, and COBOL etc. It is pictorial language used to make software blueprints.

Although typically used in software engineering it is a rich language that can be used to model an application structure, behaviour and even business processes. There are 8 UML diagram types to help us model this behaviour.

4.2.1 State Chart Diagram

A state chart diagram (also called a state diagram) is a type of UML diagram that models the dynamic behavior of a system by showing its states and transitions. It represents how an object responds to events by transitioning between different states. Each state is defined by its entry and exit actions, internal actions, and transitions. State chart diagrams are especially useful for modeling reactive systems like embedded software or control systems. They include elements such as initial state, final state, states, events, and transitions..

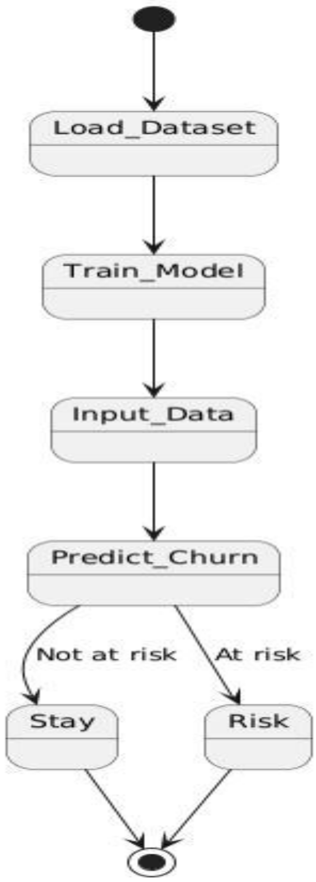


Fig 4.2.1: State Chart diagram

4.2.2 Class Diagram

Because a lot of software is based on object-oriented programming, where developers define types of functions that can be used, class diagrams are the most commonly used type of UML diagram. Class diagrams show the static structure of a system, including classes, their attributes and behaviours, and the relationships between each class. A class is represented by a rectangle that contains three compartments stacked vertically: Class name, attributes and functions of that class with access specifiers. The class diagram also contains association, generalization and aggregation. Where generalization is to show inheritance of a class and aggregation is to show that objects are assembled or configured together to create a more complex object.

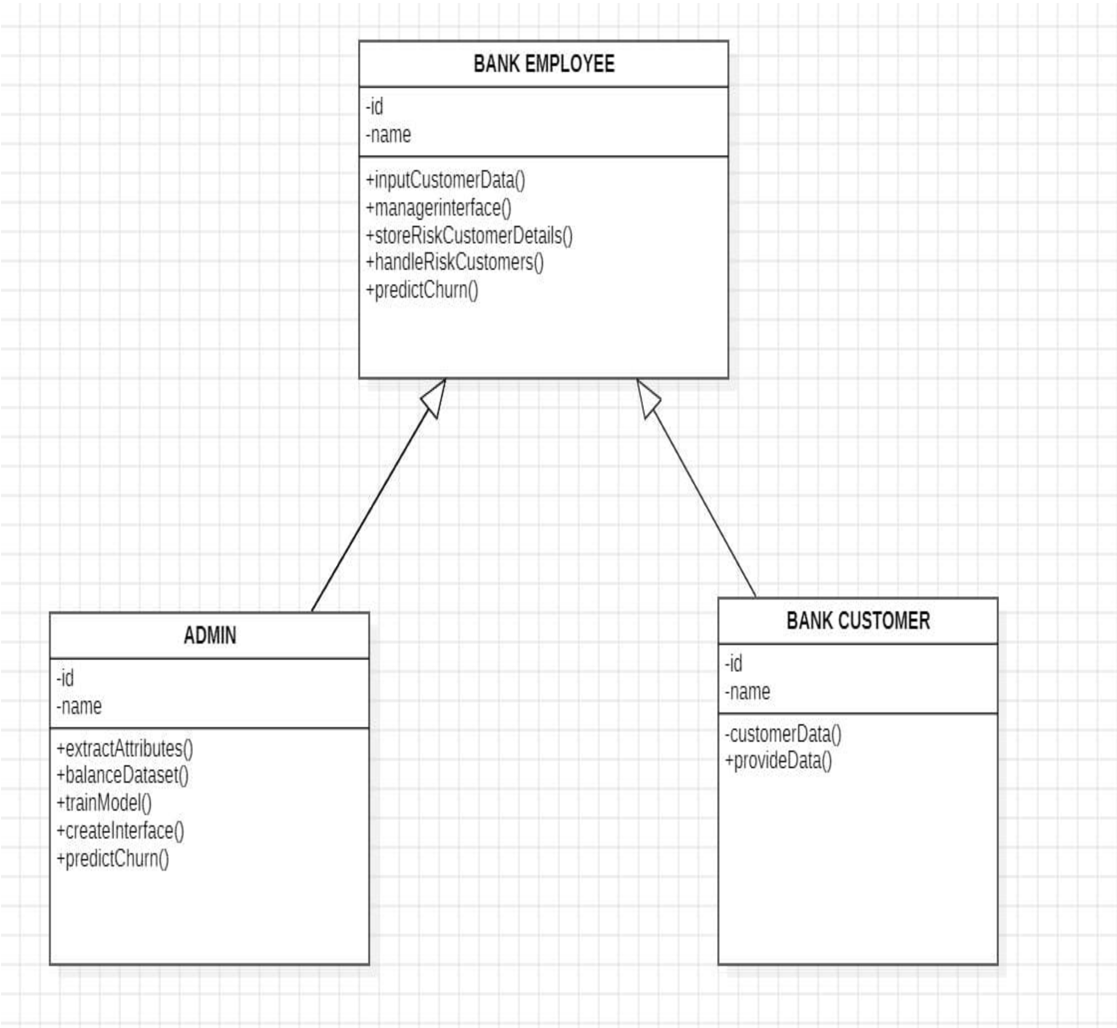


Fig 4.2.2: Class diagram

4.2.3 Activity Diagram

The Unified Modelling Language includes several subsets of diagrams, including structure diagrams, interaction diagrams, and behaviour diagrams. Activity diagrams, along with use case and state machine diagrams, are considered behaviour diagrams because they describe what must happen in the system being modelled. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

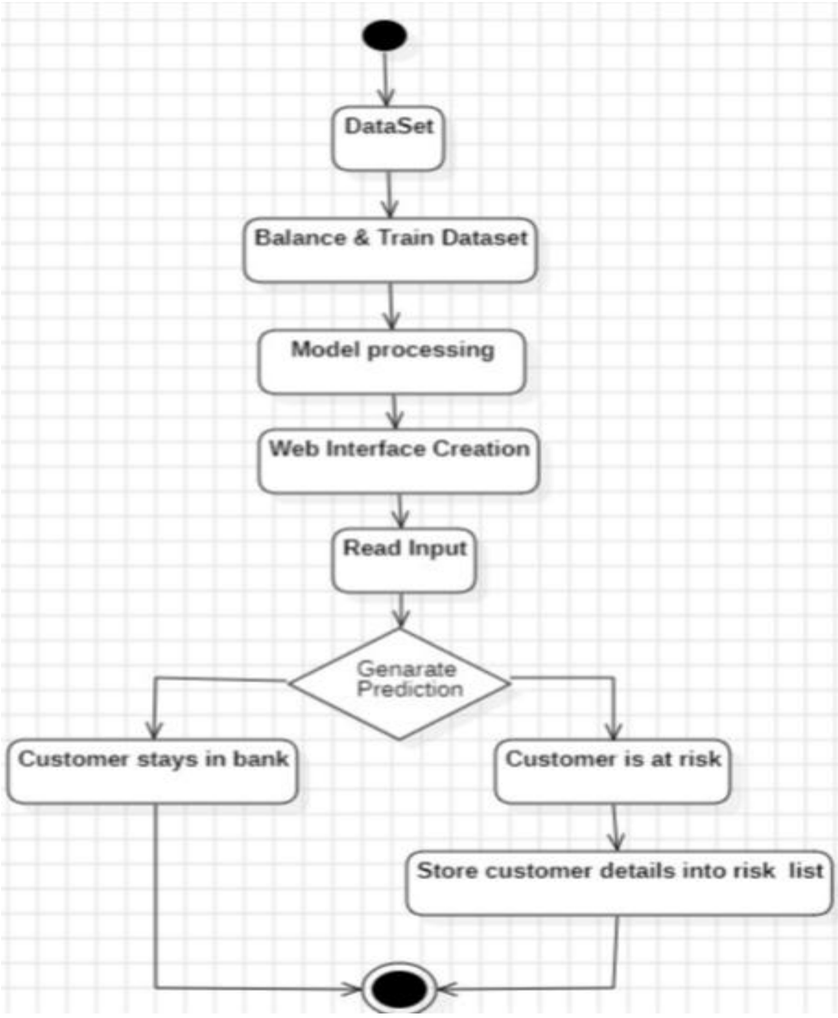


Fig 4.2.3: Activity diagram

4.2.4 Use case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program under developed. In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. It is useful in the following situations: Scenarios in which your system or application interacts with people, organizations, or external systems, Goals that your system or application helps those entities (known as actors) achieve, The scope of your system. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how).

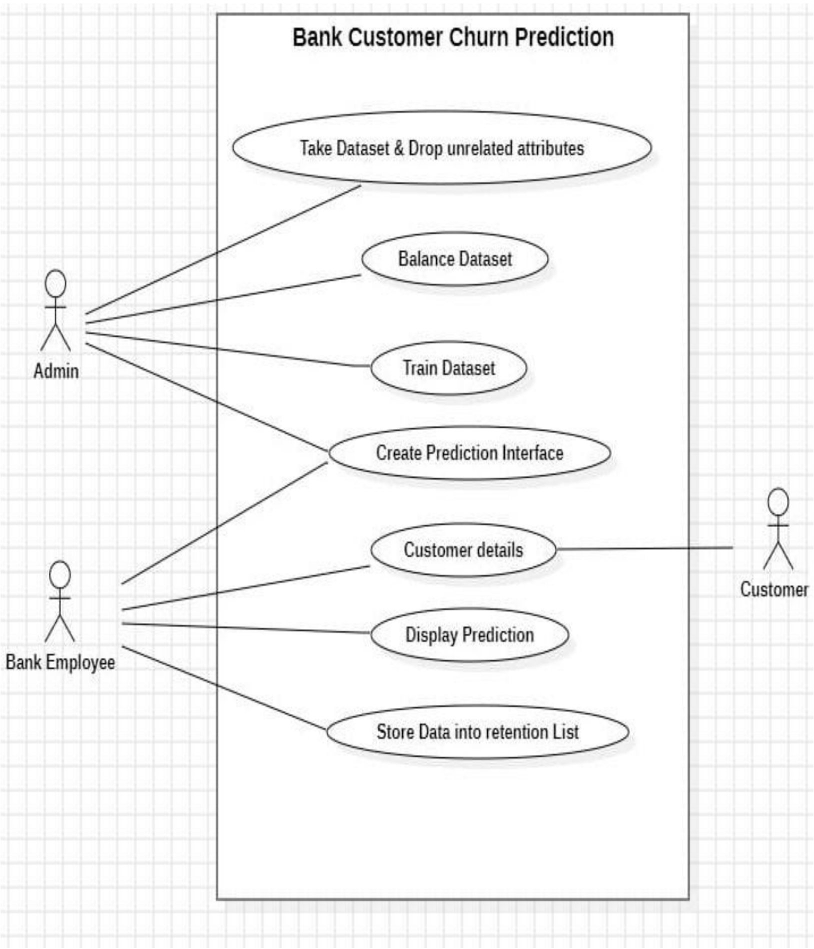


Fig 4.2.4: Use case diagram

4.2.5 Sequence Diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration, A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

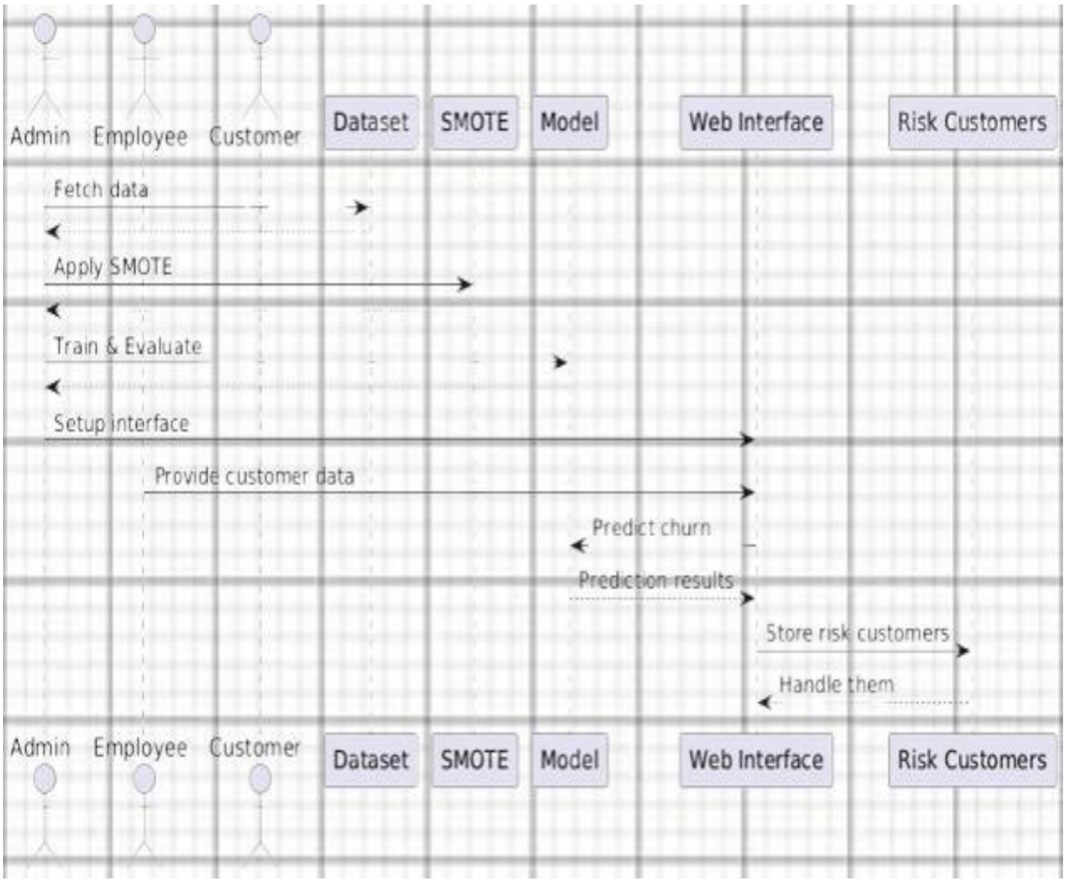


Fig 4.2.5: Sequence diagram

4.3 Modules

4.3.1 Bank Employee

The Bank Employee module streamlines data entry and customer management to support churn prediction. Employees input personal, financial, and interaction data, which is used by an integrated machine learning model to flag at-risk customers. These flagged cases are securely stored for targeted retention strategies. The module also enables easy tracking and updating of customer information over time. Its user-friendly interface enhances efficiency and supports proactive customer retention efforts.

4.3.2 Admin

The Admin module manages the backend of the churn prediction system, ensuring data quality and model accuracy. It enables admins to extract relevant features, balance datasets, and apply preprocessing for effective model training. The module supports performance monitoring and fine-tuning to enhance prediction reliability. It also integrates with the employee module to provide real-time churn predictions. Additionally, admins manage the user interface, ensuring a smooth and efficient experience for bank employees.

4.3.3 Bank Customer

The Bank Customer module collects and stores essential customer data, including demographics, transactions, and behavior patterns. This information feeds directly into the churn prediction model to assess churn risk accurately. Data is gathered through employee interactions or automated systems and is kept secure and up to date. The module works closely with the Bank Employee module to ensure continuous data accuracy. Its role in providing detailed, reliable data is vital for effective churn prediction and customer retention.

5. IMPLEMENTATION AND RESULTS

5.1 Introduction

The implementation phase is where we take our big ideas and turn them into a real, working system. This platform will be easy for people with visual impairments to use, and it will empower them to send and receive emails without any hassle. We're putting our heads together to make sure the technology works smoothly and user-friendly. Our goal is simple: to make the digital world more accessible to everyone, especially those with visual impairments. Through this implementation process, we're taking a big step toward achieving that goal.

5.2 Method of Implementation

The Incremental Iterative model is a method of software development where the project is divided into smaller parts, or increments, that are developed and tested through repeated cycles. It involves regular feedback from stakeholders and gradual improvement with each iteration. Once development starts, the team goes through cycles of designing, building, and refining. Constant collaboration is essential, both within the team and with stakeholders. So, in our project we are following this model for effective and manageable implementation.

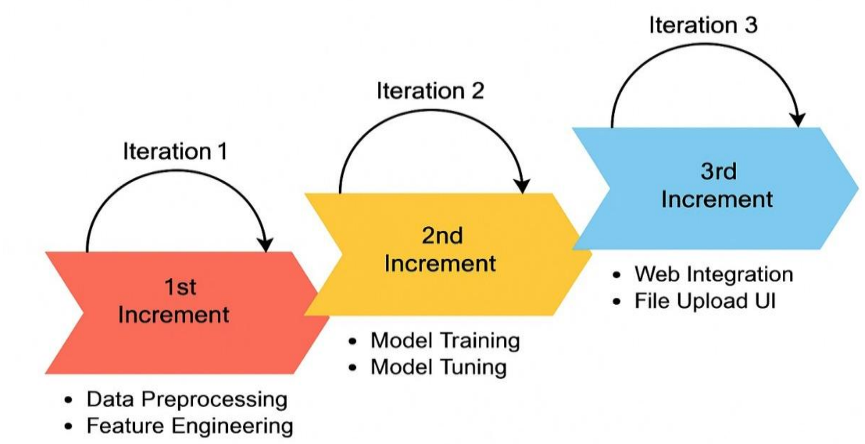


Fig 5.1: Incremental Iterative model

Incremental Iterative model four main values are:

- Progress through repeated development cycles over completing the entire system at once.
- Functioning modules delivered incrementally over full system delivery at the end.

- Continuous feedback and involvement over fixed early-stage requirements
- Adaptation through iteration over rigid adherence to a single plan.

5.2.1 app.py

```
from flask import Flask, render_template, request, redirect, url_for,
session, flash, send_file
from flask_mail import Mail, Message
import pandas as pd
import joblib
from sklearn.preprocessing import StandardScaler
import sqlite3
from werkzeug.security import generate_password_hash, check_password_hash
import random
import string
import os

app = Flask(__name__)
app.secret_key = 'vamshi'

# Flask-Mail config (use your real credentials here)
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = 'vamshijakkali609@gmail.com'      # your email
app.config['MAIL_PASSWORD'] = 'ctjx entx tomr mnfo'       # your app password
(not your normal email password)
mail = Mail(app)

DATABASE = 'churn_app.db'

# Load model & scaler
model = joblib.load('churn_predict_model.pkl')
data = pd.read_csv('Churn_Modelling.csv')
data = data.drop(['RowNumber', 'Surname', 'CustomerId'], axis=1)
data = pd.get_dummies(data, drop_first=True)
X = data.drop('Exited', axis=1)
scaler = StandardScaler()
scaler.fit(X)

# Globals
predictions_df = pd.DataFrame()
otp_store = {}  # Store OTP temporarily {user_id: otp}

def get_db_connection():
   conn = sqlite3.connect(DATABASE)
   conn.row_factory = sqlite3.Row
   return conn

def send_otp_email(to_email, otp):
   msg = Message(subject="Your OTP Code",
             sender=app.config['MAIL_USERNAME'],
             recipients=[to_email])
```

```
    msg.body = f"Your OTP for password reset is: {otp}"
    mail.send(msg)

@app.route('/')
def home():
    return redirect(url_for('login_choice'))

@app.route('/login_choice')
def login_choice():
    # Clear OTP session data if any
    session.pop('otp_user_id', None)
    session.pop('otp_verified', None)
    return render_template('login.html')

@app.route('/login', methods=['POST'])
def login():
    user_type = request.form.get('user_type')
    user_id = request.form.get('user_id')
    password = request.form.get('password')
    entered_otp = request.form.get('otp')

    if not user_type or not user_id:
        return render_template('login.html', error="Please select user type and
enter ID.")

    conn = get_db_connection()
    if user_type == 'admin':
        user = conn.execute('SELECT * FROM admins WHERE id = ?',
(user_id,)).fetchone()
    else:
        user = conn.execute('SELECT * FROM employees WHERE id = ?',
(user_id,)).fetchone()
    conn.close()

    if not user:
        return render_template('login.html', error="User not found.")

    # If OTP is being verified:
    if entered_otp:
        if session.get('otp_user_id') == user_id and otp_store.get(user_id) ==
entered_otp:
            # OTP correct: allow password reset form submission (simplified)
            session['otp_verified'] = True
            flash("OTP verified! Please enter your new password.", "info")
            return render_template('login.html', show_reset=True,
user_id=user_id, user_type=user_type)
        else:
            return render_template('login.html', error="Invalid OTP. Please try
again.", show_otp=True, user_id=user_id, user_type=user_type)

    # If new password is submitted (after OTP verification)
    new_password = request.form.get('new_password')
    if new_password and session.get('otp_verified') and
session.get('otp_user_id') == user_id:
        # Update password in DB
```

```
      hashed_pw = generate_password_hash(new_password)
      conn = get_db_connection()
      table = 'admins' if user_type == 'admin' else 'employees'
      conn.execute(f'UPDATE {table} SET password = ? WHERE id = ?',
(hashed_pw, user_id))
      conn.commit()
      conn.close()
      # Clear OTP session info
      session.pop('otp_verified', None)
      session.pop('otp_user_id', None)
      otp_store.pop(user_id, None)
      flash("Password updated successfully! Please login.", "success")
      return redirect(url_for('login_choice'))

   # Normal login password check
   if check_password_hash(user['password'], password):
      session['user_id'] = user_id
      session['user_type'] = user_type
      session['user_name'] = user['name']
      return redirect(url_for('index'))
   else:
      # Password incorrect -> send OTP to registered email automatically
      # Store OTP and user_id in session for verification
      otp = ''.join(random.choices(string.digits, k=6))
      otp_store[user_id] = otp
      session['otp_user_id'] = user_id
      try:
         send_otp_email(user['email'], otp)
      except Exception as e:
         return render_template('login.html', error=f"Error sending OTP
email: {str(e)}")
      return render_template('login.html', error="Wrong password. Enter OTP
sent to your registered email.", show_otp=True, user_id=user_id,
user_type=user_type)

@app.route('/logout')
def logout():
   session.clear()
   return redirect(url_for('login_choice'))

@app.route('/predict', me1thods=['GET', 'POST'])
def index():
   if 'user_id' not in session:
      return redirect(url_for('login_choice'))

   global predictions_df
   error = None
   if request.method == 'POST':
      file = request.files.get('file')
      if not file or not file.filename.endswith('.csv'):
         error = "Please upload a valid CSV file (.csv only)."
         return render_template('index.html', predictions=None, error=error,
user_name=session['user_name'])

      try:
```

```
        df = pd.read_csv(file)
        original_df = df.copy()

        required_features = ['CreditScore', 'Geography', 'Gender', 'Age',
'Tenure',
                        'Balance', 'NumOfProducts', 'HasCrCard',
'IsActiveMember', 'EstimatedSalary']
        missing = [col for col in required_features if col not in
df.columns]
        if missing:
            error = f"Missing required columns: {', '.join(missing)}"
            return render_template('index.html', predictions=None,
error=error, user_name=session['user_name'])

        df = pd.get_dummies(df, drop_first=True)

        for col in (set(X.columns) - set(df.columns)):
            df[col] = 0
        df = df[X.columns]

        df_scaled = scaler.transform(df)
        predictions = model.predict(df_scaled)

        original_df['Exited'] = predictions
        predictions_df = original_df[original_df['Exited'] == 1]

        # Save churn results (Exited) to DB
        conn = get_db_connection()
        for idx, row in original_df.iterrows():
            conn.execute('''INSERT OR REPLACE INTO customers (CustomerId,
CreditScore, Geography, Gender, Age,
                        Tenure, Balance, NumOfProducts, HasCrCard,
IsActiveMember, EstimatedSalary, Exited)
                        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)''',
                     (int(row.get('CustomerId', idx)), row['CreditScore'],
row['Geography'], row['Gender'],
                      row['Age'], row['Tenure'], row['Balance'],
row['NumOfProducts'], row['HasCrCard'],
                      row['IsActiveMember'], row['EstimatedSalary'],
int(row['Exited'])))
        conn.commit()
        conn.close()

        total = len(original_df)
        at_risk = sum(predictions)
        safe = total - at_risk

        return render_template('index.html', predictions=predictions_df,
                               total=total, at_risk=at_risk, safe=safe,
user_name=session['user_name'])

    except Exception as e:
        error = f"Error during file processing: {str(e)}"
        return render_template('index.html', predictions=None, error=error,
user_name=session['user_name'])
```

```
    return render_template('index.html', predictions=None, error=error,
user_name=session['user_name'])

@app.route('/download')
def download():
    global predictions_df
    file_path = 'at_risk_customers.xlsx'
    predictions_df.to_excel(file_path, index=False)
    return send_file(file_path, as_attachment=True)

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

5.3 Algorithms and Flowcharts

The core of the involves machine learning algorithms that analyze customer behavior and transaction data to predict churn risk. Oversampling techniques like XGSMOTE and Borderline-SMOTE are essential for handling class imbalance and improving model accuracy. Algorithms such as Gradient Boosting Machine help in identifying key patterns and trends that signal potential churn. Data preprocessing algorithms clean and prepare customer profiles for analysis, ensuring reliable predictions. Feature selection and evaluation techniques optimize model performance, enabling banks to proactively address churn and improve customer rete

Fig 5.3 Understanding customer churn

5.4 Control Flow of the Implementation

5.4.1 Importing Dependencies

- Python: Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.
- Flask: A lightweight web framework used to build the application’s user interface, handle routes, and manage user sessions and authentication.
- Joblib: Loads the pre-trained churn prediction model efficiently without retraining.

5.5 Sample Code

- **Scikit-learn:** Scikit-learn is a comprehensive machine learning library in Python that offers tools for data mining, analysis, and model building. Built on top of NumPy, SciPy, and Matplotlib, it supports algorithms for classification, regression, and clustering
- **Numpy**:NumPy is the core library for numerical computing in Python, supporting large, multi-dimensional arrays and a range of mathematical functions.
- **Pandas**: Pandas is a powerful data manipulation library that provides structures like DataFrame for handling structured datasets.
- **SQLite3:** A lightweight embedded database used to store user credentials and churn prediction results**.**
- **OS** – Manages file operations like saving and downloading prediction results as Excel files.
- **Random & String** – Generate secure 6-digit OTPs for verifying users during login or password reset
- **Werkzeug** – Used for secure password hashing and verification to manage login authentication.

```
from flask import Flask, render_template,
request, redirect, url_for, session, flash,
send_file
from flask_mail import Mail, Message
import pandas as pd
import joblib
from sklearn.preprocessing import StandardScaler
import sqlite3
from werkzeug.security import
generate_password_hash, check_password_hash
import random
import string
import os
```

fig 5.4.1: Importing dependencies

5.6 Output Screens

Result analysis is a static analysis that determines which functions in each program can return multiple results in an efficient manner.

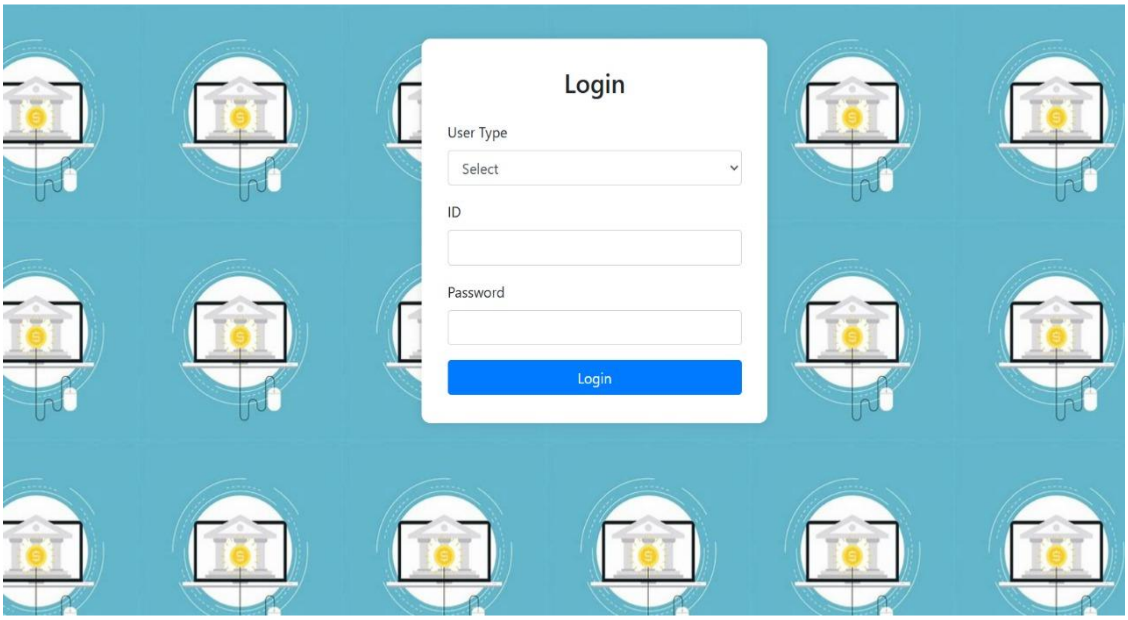
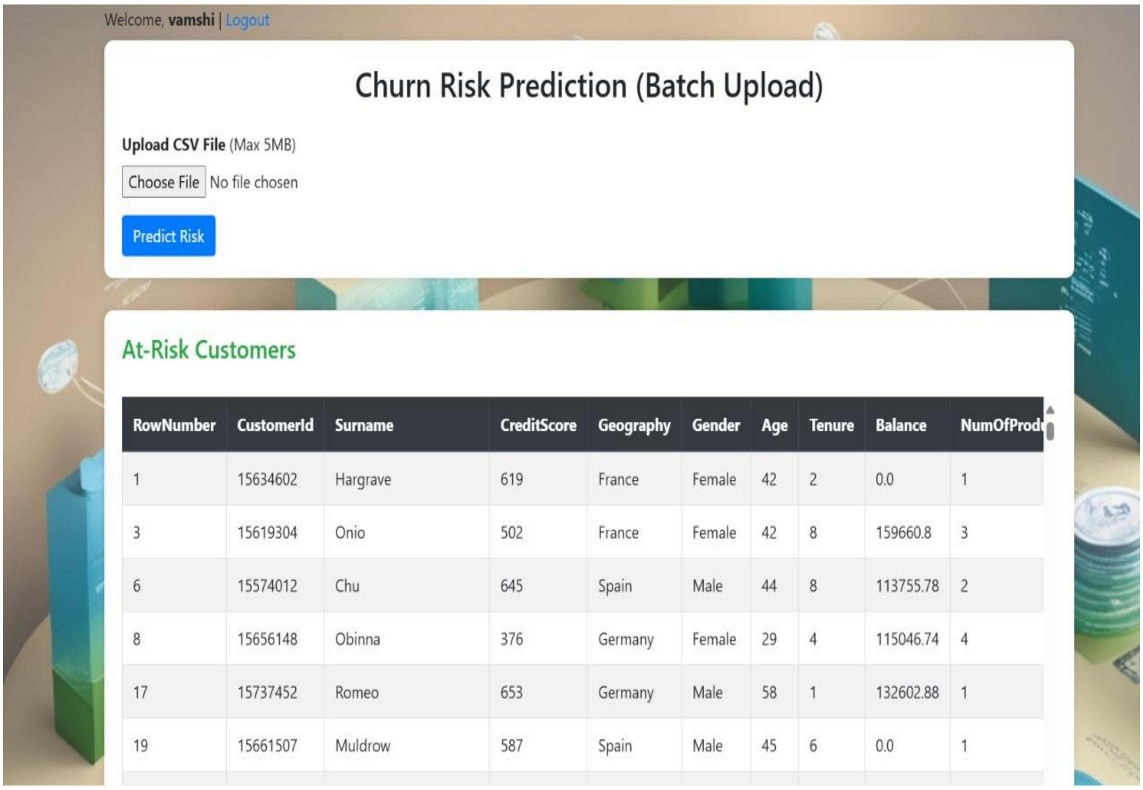


Fig 5.6.1: Output screen for login

The above figure shows the login page of bank customer churn predictor. It has three fields,one for selecting type of user,second is identity number of the user and the other for the password. If the credentials are correct, the user is redirected to the menu page, or else the user is asked to enter the correct credentials.

Fig 5.6.2: Output screen for index Page

The above figure shows the user interface of the menu bank customer churn predictor. The user is redirected to this page if they enter the correct credentials in the login page. Then upload the csv file and need to click the predict risk button.



RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProd
1	15634602	Hargrave	619	France	Female	42	2	0.0	1
3	15619304	Onio	502	France	Female	42	8	159660.8	3
6	15574012	Chu	645	Spain	Male	44	8	113755.78	2
8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4
17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1
19	15661507	Muldrow	587	Spain	Male	45	6	0.0	1

Fig 5.6.4: Output screen showing of customers at risk

The above figure shows the user interface of the risk customers of customer churn predict system .After uploading the particular CSV file containing all the columns which used while training the model like CreditScore ,Geography etc.

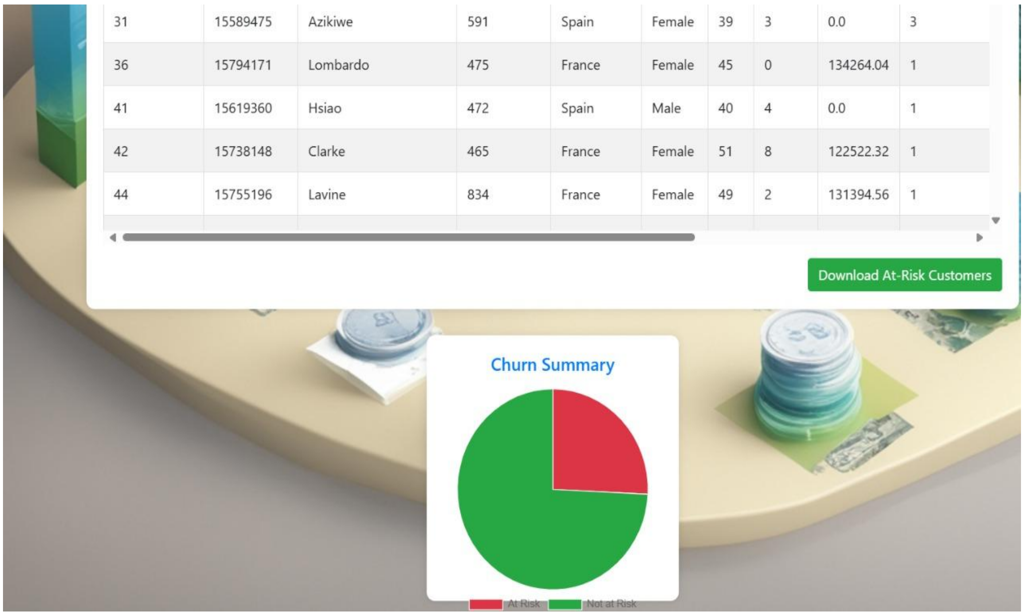


Fig 5.6.3: Output screen for downloading file of risk customers and churn chart

The figure shows the user interface of the compose if we need to download the risk customers list,by clicking “Download At-Risk Customers”,it will download the risk customers list.

6. TESTING

6.1 Introduction to Testing

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free to produce the quality product.

6.2 Types of Tests Considered

- **Application Testing:** It is defined as a software testing type, conducted through scripts with the motive of finding errors in software. It deals with tests for the entire application. It helps to enhance the quality of your applications while reducing costs, maximizing ROI, and saving development time.
- **System Testing:** It is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system.
- **GUI Testing:** GUI Testing is a software testing type that checks the Graphical User Interface of the Software. The purpose of Graphical User Interface (GUI) Testing is to ensure the functionalities of software application work as per specifications by checking screens and controls like menus, buttons, icons, etc.
- **Security Testing:** Security Testing is a type of Software Testing that uncovers vulnerabilities of the system and determines that the data and resources of the system are protected from possible intruders. It ensures that the software system and application are free from any threats or risks that can cause a loss. Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.

6.3 Various Test Case Scenarios Considered

Test case writing is a major activity and considered as one of the most important parts of software testing. It is used by the testing team, development team as well as the management. If there is no documentation for an application, we can use the test case as a baseline document.

Table 6.3: Test cases for proposed system

TEST CASE ID	TEST CASE SCENARIO	INPUTS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
1	Login with by giving id number and password	Enter id number and password	If given the credentials are correct, it should redirect to menu page or else you have to repeat the process.	The menu page is displayed successfully if the credentials are correct.	PASS
2	On reaching menu page, need to upload csv with appropriate columns.	User has to upload a csv file.	Depending on the file available, need to upload	It will accept the csv file,if give with the condition less than 5MB.	PASS
3	In menu page select Predict Risk.	Click the Predict Risk button to view the risk customers.	The risk customers will be visible on interface.	Risk customers will be predicted successfully.	PASS
4	In menu page select Download Risk Customers to download file.	Click the download button to download the list.	Risk customers list will be download.	Risk customers list will be downloaded successfully.	PASS
5	In menu page, select logout option	Say logout	The application should logout	Logged out successfully	PASS

7. CONCLUSION & FUTURE ENHANCEMENT

7.1 Project Conclusion

The Bank Customer Churn Prediction project aims to identify customers likely to leave, enabling targeted retention strategies. To address class imbalance—where churned customers are fewer—Borderline-SMOTE is applied. This technique generates synthetic samples near the decision boundary, improving the model's ability to differentiate between churned and non-churned customers. Gradient Boosting Machine (GBM), a powerful ensemble method, is then used to build predictive models by sequentially improving upon errors made by prior trees. The data is first preprocessed and balanced using Borderline-SMOTE before training the GBM model, which captures complex feature interactions. The trained model is validated on unseen data to assess accuracy. A system is also in place to flag high-risk customers, enabling the bank to allocate retention resources effectively. By integrating Borderline-SMOTE with GBM, the project offers a precise, data-driven approach to minimizing customer churn and maximizing profitability.

7.2 Future Enhancement

Future enhancements to the Bank Customer Churn Prediction project can further improve its accuracy and applicability. Incorporating advanced models like deep learning could better capture complex customer behavior patterns. Enhancing feature engineering and including real-time data processing would enable dynamic, up-to-date churn predictions. Integrating external data sources—such as economic trends, social media sentiment, and customer interaction logs—can provide a more holistic understanding of churn drivers. Additionally, implementing explainability tools like SHAP (Shapley Additive Explanations) would increase model transparency, allowing stakeholders to understand and trust predictions. These improvements would support more effective retention strategies and decision-making, ultimately increasing customer loyalty and long-term profitability for the bank.

8. REFERENCES

8.1 Journals

[1] Ullah, B. Raza, A. K. Malaik, M. Imran, “A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector”, IEEE Access, Volume 7, 2024

[2] A. Hammoudeh, M. Fraihat, M. Almomani, “Selective Ensemble Model for Telecom Churn Prediction”, Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 2024.

[3] R. Huang, Q. Zhao, S. Li, “Advanced Churn Prediction Model Using XGBoost and Feature Selection Techniques”, IEEE International Conference on Artificial Intelligence and Machine Learning (AIML), September 2021.

[4] J. Taylor, M. Evans, “A Comparative Analysis of Machine Learning Algorithms for Churn Prediction in the Retail Sector”, IEEE International Conference on Retail Analytics (ICRA), November 2021.

[5] N. Kumar, A. Singh, “A Hybrid Model for Bank Customer Churn Prediction Combining Machine Learning and Business Rules”, 2021 International Conference on Business Analytics (ICBA), June 2023.

8.2 Books

- Python: Python Programming for Beginners by Adam Stark
- Head-First Python, 2nd edition, Paul Barry
- Flask for Beginners: Build Web Applications with Python and Flask by Miguel Grinberg

8.3 Sites

- https://wwwieee.org/
- https://www.ijert.org/
- https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting