Customer Churn Prediction using Machine Learning

Project Overview

This project involves creating a predictive system that identifies customers likely to leave a telecom provider (churn). Students will develop a machine learning model that analyzes customer behavior and subscription features. The project emphasizes classification techniques and business insights.

Objective

Build a binary classifier that predicts whether a customer will churn (YES/NO) based on demographic, usage, and billing data.

Dataset

IBM Telco Customer Churn Dataset
Public & Open Source - provided with the problem statement.

Dataset Details

- Rows: ~7,000 customers
- Features: 21 (Categorical & Numerical)
- Target column: Churn (Yes / No)

Examples of input features:

- Contract type
- Tenure (months)
- Internet service type
- Monthly charges
- Payment method

Tools & Technologies

Category	Options
Programming	Python
ML Libraries	Scikit-learn, XGBoost, pyTorch, TensorFlow, any ML library
Data Handling	Pandas, NumPy, any machine learning library

Category	**Options**
Visualization	Matplotlib, Seaborn
UI Options	Flask / Tkinter / Streamlit/ Any other

Project Components

1️⃣ Data Preprocessing

- Load and explore dataset (missing values, data types)
- Encode categorical variables
    - One-Hot Encoding / Label Encoding
- Normalize numerical features
- Train/Test split (e.g., 80/20)
- Handle imbalanced classes
    - Class weights or SMOTE oversampling

2️⃣ Model Building

You can train and evaluate any models including but not limited to:

- Logistic Regression
- Random Forest
- Support Vector Machine
- Gradient Boosting
- Neural network

Evaluate using:

- Accuracy
- F1-score (important due to imbalance)
- Confusion matrix
- ROC-AUC curve

3️⃣ User Interface

Students must create a simple GUI

UI choices allowed:

- Flask web app
- Tkinter desktop app

- Streamlit dashboard

4️⃣ Documentation & Insights

- Write one page document for methodology, result, and conclusion. Use UML diagram, class diagram, confusion matrix to explain the result.

Expected Deliverables

Component	Requirement
Code	Clean, modular Python scripts
UI	Fully functional demo
Report	Methodology, results, business conclusion
Visuals	Confusion matrix, performance charts

Submission Guidelines

1. Upload the source code as zip folder.
2. PDF documentation