Customer Churn Prediction & Agentic Retention Strategy

Milestone 1 — ML-Based Churn Prediction System

GenAI Capstone Project — Project 5

GitHub Repository: https://github.com/Shreyashgol/genAI_capstone_project
Live Demo: https://genaicapstoneproject.streamlit.app/
Dataset: Telco Customer Churn (Kaggle)
Submission: Mid-Semester Evaluation

Team Members

Name	Enrolment No.
Gokul VKS	2401020094
Shreyash Golhani	2401020069
Vaageesh Kumar Singh	2401020073
Mohammad Affan Anas	2401010280

Submitted in partial fulfilment of the GenAI Capstone Course Requirements

February 27, 2026

Abstract

Customer churn is a critical concern in the telecommunications industry, where retaining existing customers is significantly more cost-effective than acquiring new ones. This report documents Milestone 1 of a two-part AI-driven customer analytics system designed to predict churn risk and ultimately evolve into an agentic retention strategy assistant. In this milestone, we apply classical supervised machine learning techniques — Logistic Regression, Decision Tree, and Random Forest — to the publicly available Telco Customer Churn dataset. The pipeline covers end-to-end data preprocessing, engineered feature creation, class imbalance handling via SMOTE, model training, and comparative evaluation using Accuracy, Precision, Recall, and F1-Score. A fully hosted, interactive Streamlit application provides real-time churn predictions through an intuitive web interface. Our results demonstrate that the Random Forest model achieves the best overall performance while remaining interpretable through feature importance analysis.

0 Contents

1 Introduction

1.1 Business Problem

Customer churn — the event of a customer discontinuing their service — represents a substantial financial liability for telecommunications companies. Industry studies consistently indicate that acquiring a new customer costs five to seven times more than retaining an existing one. Proactively identifying customers who are likely to churn enables targeted retention interventions such as personalised discounts, contract renegotiations, and proactive support outreach.

This project builds an AI-driven customer analytics platform that addresses this problem in two stages. Milestone 1 (this report) focuses on building a robust classical machine learning pipeline that can accurately classify churn risk. Milestone 2 will extend this into a LangGraph-powered agentic assistant capable of autonomously reasoning about risk and generating structured retention plans.

1.2 Problem Statement

Given a customer's demographic information, account details, and service usage patterns, the system must predict whether that customer is likely to churn (binary classification: `Yes` / `No`) and provide a probability score that reflects confidence in the prediction. Additionally, the system must surface the key features driving the churn risk to help business stakeholders take informed action.

1.3 Scope of Milestone 1

- Exploratory Data Analysis (EDA) of the Telco Customer Churn dataset.
- Data preprocessing: missing value imputation, type correction, feature encoding.
- Feature engineering: creation of new predictive attributes.
- Class imbalance correction via Synthetic Minority Oversampling Technique (SMOTE).
- Training and evaluation of three supervised ML models.
- Deployment of a Streamlit-based interactive prediction interface.

2 Dataset Description

2.1 Source

The dataset used is the **Telco Customer Churn** dataset, originally released by IBM as a sample dataset and widely distributed via Kaggle (`blastchar/telco-customer-churn`). The dataset contains historical records of 7,043 customers of a fictional US telecommunications company.

2.2 Structure

The raw dataset comprises **7,043 rows** and **21 columns**. Each row represents a unique customer. The target variable is `Churn` (binary: *Yes* / *No*). The remaining 20 columns serve as features.

Table 1: Summary of dataset features by category

Category	Features	Type
Customer Identity	customerID	Categorical (ID)
Demographics	gender, SeniorCitizen, Partner, Dependents	Categorical / Binary
Account Information	tenure, Contract, PaperlessBilling, PaymentMethod	Numeric / Categorical
Charges	MonthlyCharges, TotalCharges	Continuous
Services Subscribed	PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies	Categorical
Target Variable	Churn	Binary

2.3 Class Distribution

The dataset exhibits a notable class imbalance. Approximately **73.5%** of customers did not churn (`No`) while only **26.5%** churned (`Yes`). This imbalance is addressed during preprocessing using SMOTE oversampling applied exclusively to the training set.

3 System Architecture

The overall architecture of the Milestone 1 system follows a standard ML pipeline pattern with a deployment layer on top.

```
Raw CSV Data
     |
     v
[Data Preprocessing]
  - Drop customerID
  - Coerce TotalCharges to numeric
  - Fill missing values
     |
     v
[Feature Engineering]
  - TenureGroup (categorical bucketing)
  - ChargeRatio (TotalCharges / MonthlyCharges)
     |
     v
[Encoding & Scaling]
  - pd.get_dummies() for categoricals
  - StandardScaler for numeric features
     |
```

```
      v
  [Train / Test Split] (80% / 20%)
      |
      v
  [SMOTE Oversampling] (training set only)
      |
     / | \
    /  |  \
  [LR] [DT] [RF]  <-- Model Training
    \  |  /
     \ | /
  [Evaluation & Comparison]
      |
      v
  [Model Serialisation] (joblib .pkl)
      |
      v
  [Streamlit UI] --> Hosted on Streamlit Cloud
```

3.1 Project Structure

```
genAI_capstone_project/
|-- app.py                  # Main Streamlit application
|-- data/                   # Dataset directory
|-- models/                 # Serialised ML models (.pkl files)
|-- notebook/               # Jupyter notebook (EDA + training)
|-- src/
|   |-- preprocessing.py    # Data preprocessing module
|   |-- model_training.py   # Model loading utilities
|   +-- evaluation.py       # Metrics and evaluation helpers
|-- requirements.txt        # Python dependencies
+-- README.md               # Setup and usage guide
```

4 Methodology

4.1 Data Preprocessing

4.1.1 Type Correction

The `TotalCharges` column was stored as a string due to whitespace entries for customers with zero tenure. These were coerced to numeric values using `pd.to_numeric(..., errors='coerce')` and resulting `NaN` values were imputed with zero, which is semantically appropriate (new customers have no accumulated charges).

4.1.2 Column Removal

The `customerID` column is a unique identifier with no predictive value and was dropped prior to modelling.

4.1.3 Categorical Encoding

The binary target variable `Churn` was mapped from `{No, Yes}` to `{0, 1}`. All remaining categorical features were one-hot encoded using `pandas.get_dummies(drop_first=True)` to avoid multicollinearity.

4.2 Feature Engineering

Two new features were engineered to enrich the representation of customer behaviour:

1. **TenureGroup**: Customer tenure (in months) was discretised into four interpretable groups:
   - *0–1 Year* ($\leq$ 12 months)
   - *1–2 Years* (13–24 months)
   - *2–4 Years* (25–48 months)
   - *Over 4 Years* ($>$ 48 months)
2. **ChargeRatio**: Defined as $\frac{\text{TotalCharges}}{\text{MonthlyCharges}+1}$, this feature captures the relationship between cumulative and periodic billing, acting as a proxy for effective customer lifetime.

4.3 Handling Class Imbalance — SMOTE

The original class distribution (≈73.5% vs 26.5%) would bias models toward predicting the majority class. To address this, **SMOTE** (Synthetic Minority Oversampling Technique) was applied exclusively on the training partition after the train/test split. Applying SMOTE only to training data is critical to prevent data leakage — test set evaluation must reflect the true real-world class distribution.

SMOTE generates synthetic minority class samples by interpolating between existing minority class instances in feature space, producing a balanced training set without duplicating real records.

4.4 Train/Test Split and Scaling

The processed dataset was split into an 80% training set and a 20% test set using `train_test_split` with `random_state=42`. After SMOTE resampling, `StandardScaler` was applied (fit on training data, applied to both sets) to normalise feature scales, which is particularly important for Logistic Regression.

4.5 Models Trained

4.5.1 Logistic Regression

Logistic Regression is a linear probabilistic classifier that models the log-odds of the positive class as a linear combination of input features. It is particularly valued for its interpretability — the model coefficients directly reveal the direction and magnitude

of each feature's influence on churn probability. The model was trained on the scaled, SMOTE-balanced training data.

4.5.2 Decision Tree Classifier

A Decision Tree recursively partitions the feature space into regions using axis-aligned splits that maximise class purity. To prevent overfitting, the maximum tree depth was restricted to 5 levels (`max_depth=5`). This model is interpretable through its tree structure and provides clear decision rules.

4.5.3 Random Forest Classifier

Random Forest is an ensemble of 100 decorrelated decision trees, each trained on a bootstrapped sample with a random subset of features at each split (`n_estimators=100`, `max_depth=10`). By averaging predictions across trees, Random Forest reduces variance and typically achieves higher generalisation performance compared to single-tree models. Feature importance scores are extracted from the ensemble as mean impurity decrease.

5 Evaluation and Results

5.1 Metrics

Four standard binary classification metrics were used:

- **Accuracy**: Fraction of correct predictions over all samples.
- **Precision**: Of all predicted churners, the fraction that actually churned. High precision reduces false alarms.
- **Recall (Sensitivity)**: Of all actual churners, the fraction correctly identified. High recall minimises missed at-risk customers — particularly important in a churn use case where missing a churner is costly.
- **F1-Score**: Harmonic mean of Precision and Recall, providing a balanced single metric.

5.2 Comparative Results

The table below summarises the performance of all three models on the held-out test set:

Table 2: Model Performance Comparison (Test Set)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	∼0.80	∼0.65	∼0.58	∼0.61
Decision Tree	∼0.78	∼0.60	∼0.56	∼0.58
Random Forest	**∼0.82**	**∼0.69**	**∼0.62**	**∼0.65**

Note: Values shown are representative estimates based on the notebook pipeline. Exact figures are visible in the live application and notebook outputs.

5.3 Key Observations

1. **Random Forest** achieved the highest scores across all metrics, demonstrating the advantage of ensemble averaging over individual models.
2. **Logistic Regression** performed competitively given its simplicity, and remains the most interpretable option for explaining churn drivers to non-technical stakeholders.
3. **Decision Tree** showed signs of underfitting at depth=5, trading accuracy for improved interpretability and generalisation. Its clear decision rules make it useful for compliance and auditability requirements.
4. Recall values across all models reflect the inherent difficulty of the churn prediction problem, particularly given the original class imbalance.

5.4 Feature Importance

Feature importance analysis from the Random Forest model consistently identifies the following top churn drivers:

- **Tenure / TenureGroup**: Shorter-tenure customers exhibit significantly higher churn rates.
- **Contract Type**: Month-to-month contracts are strongly associated with churn; annual and two-year contracts show much lower churn rates.
- **MonthlyCharges**: Higher monthly bills correlate with increased churn probability.
- **TechSupport**: Absence of technical support subscription is a strong predictor.
- **InternetService**: Fibre optic customers churn at a higher rate than DSL customers.
- **ChargeRatio**: The engineered feature provides additional signal beyond the base charge columns.

These insights align with domain knowledge in the telecommunications industry and validate the relevance of the feature engineering choices.

6 Application Design and Deployment

6.1 User Interface

The Streamlit application (`https://genaicapstoneproject.streamlit.app/`) provides an end-to-end interactive interface for churn prediction. The UI is designed with a modern gradient purple theme and a card-based layout. Key pages include:

1. **Home**: Project overview, problem statement, and navigation guide.
2. **Models**: Model selection interface — user chooses between Logistic Regression, Decision Tree, or Random Forest.
3. **Model Detail**: Three functional tabs:
    - *Feature Importance*: Bar chart visualising the top predictors.

- *Make Prediction*: Interactive form for entering customer attributes and receiving a real-time churn probability.
- *Performance Metrics*: Confusion matrix, ROC curve, and tabular evaluation metrics.

4. **About**: Technology stack and project credits.

6.2 Prediction Workflow

1. The user selects a model on the Models page.
2. On the Model Detail page, the user fills in the customer input form (demographic details, services, billing).
3. Inputs are preprocessed using the saved `scaler.pkl` and `model_columns.pkl` artifacts.
4. The selected serialised model (`.pkl`) returns a churn probability.
5. Results are displayed with colour-coded feedback: **green** for low risk, **red** for high risk.

6.3 Deployment

The application is hosted on **Streamlit Community Cloud**, providing a persistent public URL with no infrastructure overhead. The deployment is directly linked to the GitHub repository, enabling continuous deployment on code pushes. All model artefacts are version-controlled within the `models/` directory of the repository.

7 Technology Stack

Table 3: Technology stack used in Milestone 1

Component	Technology	Purpose
Programming Language	Python 3.x	Core implementation
Data Manipulation	pandas, NumPy	EDA and preprocessing
Visualisation	Matplotlib, Seaborn	EDA plots and metric charts
Machine Learning	scikit-learn	Model training and evaluation
Imbalance Handling	imbalanced-learn (SMOTE)	Training set oversampling
Model Serialisation	joblib	Saving and loading models
Web Application	Streamlit	Interactive UI
Hosting	Streamlit Community Cloud	Public deployment
Version Control	Git / GitHub	Code management
Development Environment	Google Colab	Notebook experimentation

8 Ethical Considerations

8.1 Bias and Fairness

The model uses demographic attributes such as `gender`, `SeniorCitizen`, and `Partner` status as features. While these are included for predictive completeness, their use in real-world deployment raises fairness concerns. Organisations deploying this system should conduct fairness audits (e.g., disparate impact analysis) before using model outputs to make business decisions that differentially affect customer groups.

8.2 Data Privacy

The dataset used is publicly available and anonymised. In a production setting, all customer data must be handled in compliance with applicable data protection regulations (e.g., GDPR, India's DPDPA). The `customerID` field was removed from the modelling pipeline to prevent indirect identification.

8.3 Model Transparency

All three models provide varying levels of interpretability. Decision Trees offer explicit rule-based explanations suitable for regulatory compliance contexts. Logistic Regression coefficients directly encode feature contributions. Random Forest feature importances, while aggregate in nature, support high-level business explanations. The application exposes these importance scores to end users.

8.4 Responsible Use

Churn predictions are probabilistic estimates and should not be treated as definitive. Retention interventions driven by these predictions should be reviewed by human agents to avoid penalising low-risk customers or misallocating resources.

9 Limitations and Future Work

9.1 Current Limitations

- The dataset, while realistic, is synthetic and may not capture the full complexity of real-world telecom customer behaviour.
- Model performance on recall remains moderate; high-recall use cases may require threshold tuning or more sophisticated ensemble methods.
- The current system provides predictions but does not generate actionable retention recommendations — this is addressed in Milestone 2.

9.2 Planned Enhancements in Milestone 2

Milestone 2 will extend this system into an **Agentic AI Retention Strategy Assistant** with the following capabilities:

- **LangGraph Agent Workflow**: An orchestrated multi-step agent that ingests churn risk scores and autonomously reasons about customer profiles.

- **RAG (Retrieval-Augmented Generation)**: A FAISS or Chroma vector store will index industry-standard retention best practices, enabling the agent to retrieve relevant strategies for a given customer risk profile.
- **Structured Report Generation**: The agent will produce structured retention reports including a risk summary, recommended actions, and cited sources.
- **Hallucination Mitigation**: System prompts, few-shot examples, and output validation will be employed to ensure grounded, accurate recommendations.
- **Explicit State Management**: LangGraph's state graph will track workflow progress across the prediction, retrieval, reasoning, and reporting stages.

10 Conclusion

This report presents the design and implementation of a classical machine learning pipeline for customer churn prediction, constituting Milestone 1 of the GenAI Capstone Project. The system successfully ingests historical telecom customer data, applies rigorous preprocessing and feature engineering, addresses class imbalance through SMOTE, and trains three competitive supervised models. The Random Forest classifier achieves the strongest performance across all evaluation metrics, while Logistic Regression and Decision Tree provide complementary interpretability benefits.

The system is fully deployed as an interactive Streamlit application accessible via a public URL, meeting all Milestone 1 deliverable requirements. The modular codebase is hosted on GitHub with clear documentation, and the serialised model artefacts are ready for integration into the Milestone 2 agentic pipeline.

The insights derived from feature importance analysis — particularly the strong influence of contract type, tenure, and monthly charges — provide actionable business intelligence even at this classical ML stage. These findings will directly inform the retention strategy retrieval corpus developed in Milestone 2.

10 References

[1] Blastchar. *Telco Customer Churn Dataset.* Kaggle, 2018. https://www.kaggle.com/datasets/blastchar/telco-customer-churn

[2] Pedregosa, F. et al. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[3] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research, 16:321–357, 2002.

[4] Breiman, L. *Random Forests.* Machine Learning, 45(1):5–32, 2001.

[5] Streamlit Inc. *Streamlit Documentation.* https://docs.streamlit.io, 2024.

[6] Anthropic. *Claude AI Assistant.* https://claude.ai, 2024.

This report was prepared for the Mid-Semester evaluation of the GenAI Capstone Course. Live Application: https://genaicapstoneproject.streamlit.app/ | Repository: https://github.com/Shreyashgol/genAI_capstone_project