Customer Churn Prediction & Agentic Retention Strategy

Project 5 — Milestone 1: Mid-Semester Submission

GenAI Course — Machine Learning Pipeline Report

Dataset	IBM Telco Customer Churn (Kaggle)
Framework	Scikit-learn, XGBoost, Gradio
Hosting	Hugging Face Spaces
Phase	Milestone 1 — Classical ML

Team Members
Aaloke Eppalapalli – 2401010005
Arina Ali – 2401020114
Hemanth Tenneti – 2401010186
Shrihari K N – 2401010448

March 2, 2026

Contents

1. Abstract

Customer churn—the phenomenon whereby subscribers voluntarily discontinue their service—represents one of the most consequential operational challenges in the telecommunications industry. Acquiring a new customer typically costs five to seven times more than retaining an existing one [1], making accurate churn prediction a high-value business problem. This report documents the design and implementation of a classical machine learning pipeline applied to the IBM Telco Customer Churn dataset comprising 7,043 customer records and 20 predictive features.

Five supervised learning models were systematically benchmarked: Linear Regression (baseline), Logistic Regression, Decision Tree, Random Forest, and XGBoost. Logistic Regression was selected as the production model, achieving the highest F1-Score of 0.6092 and a ROC-AUC of 0.8416 on the held-out test set. The trained model is deployed as an interactive Gradio web application, allowing users to adjust 20 customer attributes across three input panels and receive an instant, colour-coded churn probability estimate.

This submission constitutes Milestone 1 of a two-phase project. Milestone 2 will extend the system into an agentic AI retention strategy assistant using LangGraph and Retrieval-Augmented Generation (RAG).

2. Introduction

2.1. Business Context and Motivation

The global telecommunications market is characterised by intense competition, commoditised pricing, and low switching costs for consumers. In this environment, customer retention is not merely a commercial priority—it is a financial imperative. Research by Reichheld and Sasser [1] demonstrates that a five-percentage-point improvement in customer retention can increase profitability by 25 to 95 percent, depending on the industry. For telecoms specifically, the cost of acquiring a new subscriber through advertising, incentives, and on-boarding routinely exceeds the cost of retaining an at-risk customer by several multiples.

Churn prediction models address this problem by transforming historical behavioural and transactional data into probabilistic risk scores. These scores enable customer success and marketing teams to prioritise retention interventions toward high-value customers who exhibit credible signals of disengagement.

2.2. Problem Statement

Given a customer's demographic profile, service subscriptions, account tenure, and billing characteristics, the objective is to construct a machine learning classifier that outputs:

- A binary churn label: *Churn* (1) or *No Churn* (0).
- A continuous churn probability score $\hat{p} \in [0, 1]$.
- An identification of the most influential features driving disengagement.

2.3. Scope of Milestone 1

The current milestone is scoped exclusively to classical machine learning methods. No large language models, generative AI, or agentic frameworks are used in this phase. The deliverables include a training notebook with full EDA, preprocessing and model benchmarking; a serialised production model; and a hosted Gradio web application accessible via a public URL.

2.4. Dataset

The IBM Telco Customer Churn dataset [2] is a widely used benchmark for binary churn classification. It contains records for 7,043 customers of a fictional California-based telecommunications provider and comprises 21 columns: one identifier (`customerID`), 19 feature columns, and one binary target (`Churn`). The feature set spans three broad categories:

- **Demographics:** gender, senior citizen status, partner, dependents.
- **Services subscribed:** phone service, multiple lines, internet service type, online security, online backup, device protection, tech support, streaming TV, streaming movies.
- **Account information:** tenure (months), contract type, paperless billing, payment method, monthly charges, total charges.

The target variable is moderately imbalanced, with approximately 26.5% of customers labelled as churned and 73.5% as retained.

3. Methodology

3.1. Data Preprocessing

Raw data ingestion and cleaning were performed in the following sequence:

3.1.1. Type Correction and Missing Value Imputation

The `TotalCharges` column is stored as an `object` dtype in the raw CSV, because 11 records contain blank strings rather than numeric values. These were coerced to `NaN` using `pd.to_numeric(..., errors='coerce')` and subsequently filled with the column median (1397.475), which is robust to the right-skew present in the distribution.

3.1.2. Identifier Removal

The `customerID` column is a non-predictive alphanumeric key. It was dropped prior to modelling to prevent any spurious pattern learning.

3.1.3. Target Encoding

The binary target `Churn` was mapped from the string values {`'Yes'`, `'No'`} to integer values {1, 0} respectively.

3.1.4. Categorical Feature Encoding

All remaining object-typed columns were one-hot encoded using pandas.get_dummies with drop_first=True to avoid multicollinearity. This expanded the feature space from 20 columns to 30 binary or numeric columns.

3.1.5. Train–Test Split

The dataset was partitioned into an 80% training set (5,634 samples) and a 20% held-out test set (1,409 samples) using stratified sampling on the target variable (random_state=42). Stratification preserves the original 73.5/26.5 class ratio in both splits.

3.1.6. Feature Scaling

A StandardScaler was fitted exclusively on the training set and applied to both splits. Scaling is mandatory for distance- and gradient-sensitive models (Logistic Regression, Linear Regression) but was deliberately withheld from the tree-based models (Decision Tree, Random Forest, XGBoost), which are invariant to monotone feature transformations.

3.2. Model Selection

Five model families were trained and evaluated to provide a comprehensive benchmark.

Table 1: Summary of models trained in Milestone 1

Model	**Key Hyperparameters**	**Notes**
Linear Regression	Default	Regression baseline, thresholded at 0.5
Logistic Regression	max_iter=1000, random_state=42	Trained on scaled features
Decision Tree	max_depth=10, random_state=42	Trained on unscaled features
Random Forest	n_estimators=100, random_state=42	Trained on unscaled features
XGBoost	eval_metric='logloss', random_state=42	Trained on unscaled features

3.3. Evaluation Metrics

Model performance was assessed on the held-out test set using four complementary metrics:

- **Accuracy:** Proportion of all correctly classified samples. Useful as a headline figure but susceptible to class imbalance.
- **Precision (Churn class):** Of all customers predicted to churn, the fraction who actually churned. High precision reduces unnecessary retention spend on non-churners.
- **F1-Score (Churn class):** Harmonic mean of precision and recall. The primary selection criterion, as it balances the cost of false positives and false negatives under moderate class imbalance.

- **ROC-AUC:** Area under the Receiver Operating Characteristic curve. Measures discriminative ability across all classification thresholds, independent of the chosen decision boundary.

3.4. Model Export and Deployment

The best-performing model (Logistic Regression) was serialised to disk using `joblib` for consumption by the Gradio application. The Gradio UI exposes three input tabs—Demographics, Account & Billing, and Services—and returns a churn probability score alongside a colour-coded half-circle risk gauge (green $< 30\%$, amber 30–60%, red $> 60\%$).

4. Results

4.1. Model Performance Comparison

Table 2 presents the complete performance comparison across all five benchmarked models, sorted by ROC-AUC in descending order.

Table 2: Model performance on the held-out 20% test set (1,409 samples)

Model	Accuracy	Precision	F1-Score	ROC-AUC
Logistic Regression	**0.8070**	**0.6584**	**0.6092**	**0.8416**
Linear Regression	0.7963	0.6426	0.5773	0.8301
Random Forest	0.7864	0.6237	0.5501	0.8251
XGBoost	0.7850	0.6079	0.5690	0.8214
Decision Tree	0.7559	0.5387	0.5486	0.7607

4.2. Selected Model: Logistic Regression

Logistic Regression was selected as the production model on the basis of its leading performance across all four evaluation metrics. Notably, it achieves the highest ROC-AUC of 0.8416, indicating strong discriminative ability across classification thresholds, alongside the best F1-Score of 0.6092 on the minority (Churn) class.

The dominance of a linear model over ensemble methods such as Random Forest and XGBoost is consistent with the characteristics of this dataset. The Telco churn dataset is relatively small (7,043 samples), moderately imbalanced, and contains features with largely linear relationships to the target. Ensemble methods benefit most from large, complex, high-dimensional datasets; on smaller, cleaner datasets they tend to overfit or offer negligible improvement over regularised linear classifiers [3].

4.3. Classification Report — Logistic Regression

Table 3 presents the per-class precision, recall, and F1-Score for the selected model.

Table 3: Classification report for Logistic Regression on the test set

Class	Precision	Recall	F1-Score	Support
No Churn (0)	0.85	0.89	0.87	1,035
Churn (1)	0.66	0.57	0.61	374
Weighted Avg	0.80	0.81	0.80	1,409

The model correctly identifies 57% of all actual churners (recall = 0.57), with a precision of 0.66 on the Churn class. This trade-off reflects the underlying class imbalance: the majority No-Churn class is well-represented in training, resulting in a slight conservative bias. The confusion matrix, presented conceptually in Table 4, reveals 212 true positives and 162 false negatives.

Table 4: Confusion matrix for Logistic Regression (test set)

	Predicted: No Churn	Predicted: Churn
Actual: No Churn	925 (TN)	110 (FP)
Actual: Churn	162 (FN)	212 (TP)

4.4. Key Churn Drivers

Logistic Regression provides interpretable coefficient weights that can be used to identify the most influential features driving churn. Based on the model coefficients and established domain knowledge for this dataset, the principal churn drivers are:

1. **Contract type:** Month-to-month customers exhibit significantly higher churn propensity compared to those on one- or two-year contracts. The absence of lock-in reduces switching friction.
2. **Tenure:** Newer customers (low tenure) are disproportionately likely to churn. Customers who have remained for longer periods have higher sunk-cost investment and familiarity with the service.
3. **Fibre optic internet service:** Counterintuitively, customers with fibre optic subscriptions show higher churn rates, likely reflecting higher monthly charges and elevated price sensitivity in this segment.
4. **Monthly charges:** Higher monthly bills are positively correlated with churn probability.
5. **Tech support and online security:** Customers who subscribe to these add-on services show notably lower churn rates, suggesting that value-added services improve perceived utility and retention.
6. **Electronic check payment:** Customers paying via electronic check churn more frequently than those using automatic payment methods, possibly reflecting lower engagement or financial uncertainty.

5. System Architecture and Deployment

5.1. Folder Structure

The project repository is organised as follows:

```
project/
 app.py                         # Gradio web application
 CustomerChurnPrediction.ipynb  # Training notebook
 README.md                      # Setup and usage instructions
 requirements.txt               # Python dependencies
 dataset/                       # Data downloaded at runtime via kagglehub
 models/
     model.pkl                  # Serialised production model
```

5.2. Technology Stack

Table 5: Technology stack used in Milestone 1

Component	Technology
Data Processing	`pandas`, `numpy`
ML Models	`scikit-learn` (Logistic Regression, Decision Tree, Random Forest), `xgboost`
Evaluation	`scikit-learn` metrics, `matplotlib`, `seaborn`
Model Serialisation	`joblib`
Web UI	`Gradio`
Dataset Access	`kagglehub`
Hosting	Hugging Face Spaces / Streamlit Community Cloud

5.3. Gradio Application

The Gradio web application provides a public-facing interface for real-time churn prediction. On first launch, the application downloads the Telco dataset from Kaggle via `kagglehub` to reconstruct the `StandardScaler`, then loads the serialised Logistic Regression model from `models/model.pkl`. The UI is structured across three input tabs:

- **Demographics:** gender, senior citizen flag, partner, dependents, tenure (months).
- **Account & Billing:** contract type, paperless billing, payment method, monthly charges, total charges.
- **Services:** phone service, multiple lines, internet service type, online security, online backup, device protection, tech support, streaming TV, streaming movies.

Upon submission, the right panel renders a half-circle risk gauge coloured green ($< 30\%$ risk), amber (30–60% risk), or red ($> 60\%$ risk), alongside the exact probability and a plain-English risk label. Three pre-built example profiles—*High-Risk*, *Loyal*, and *New Senior*—are provided for demonstration purposes.

6. Discussion

6.1. Interpretation of Results

The benchmark results in Table 2 yield several noteworthy observations. First, all five models achieve ROC-AUC scores above 0.76, confirming that the feature set carries substantial predictive signal. Second, the ordering of models by ROC-AUC (Logistic Regression > Linear Regression > Random Forest > XGBoost > Decision Tree) is somewhat counter-intuitive given that ensemble methods are generally assumed to outperform linear classifiers. This finding can be explained by the moderate size of the dataset and the relatively linear separability of churn patterns when features are properly scaled.

The Decision Tree's comparatively low ROC-AUC of 0.7607 suggests that a single, unpruned tree at `max_depth=10` is overfitting to training-set noise, a common failure mode on tabular datasets of this size.

6.2. Limitations

1. **Class imbalance:** The 73.5/26.5 class split is addressed through stratified splitting but not through oversampling (e.g., SMOTE) or class-weight adjustment. This contributes to the modest recall of 0.57 on the Churn class.
2. **No hyperparameter optimisation:** Models were trained with default or manually chosen hyperparameters. Cross-validated grid search or Bayesian optimisation could yield measurable improvements.
3. **Feature engineering depth:** No interaction features, polynomial expansions, or domain-specific aggregations were created beyond one-hot encoding.

6.3. Roadmap to Milestone 2

Milestone 2 will extend the system into a fully agentic AI retention assistant. The planned architecture includes a LangGraph workflow for state-managed, multi-step reasoning; a Chroma or FAISS vector store for RAG-based retrieval of retention best practices; an LLM (free-tier, open-source) for generating structured retention recommendations; and explicit ethical disclaimers embedded in the system output.

7. Conclusion

This report has documented the end-to-end design, implementation, and evaluation of a classical machine learning pipeline for customer churn prediction in the telecommunications domain. The pipeline ingests raw customer data, applies principled preprocessing and feature engineering, benchmarks five model families, and surfaces predictions through a publicly hosted Gradio web application.

Logistic Regression was identified as the optimal model, achieving an F1-Score of 0.6092 and a ROC-AUC of 0.8416—the best performance across all evaluated architectures. The results confirm that structured behavioural and billing data carries strong predictive signal for churn, and that interpretable linear classifiers can match or exceed ensemble methods on datasets of this scale and dimensionality.

The primary avenue for improvement lies in addressing class imbalance through SMOTE or class-weight regularisation, which is expected to meaningfully increase Churn-class recall without significant precision sacrifice. Milestone 2 will build upon this foundation, transforming the prediction system into an autonomous retention strategist capable of reasoning about risk, retrieving evidence-based interventions, and producing structured recommendations for customer success teams.

8. Team Contributions

All four team members contributed equally to the overall project. The division of responsibilities across the deliverables was as follows.

Hemanth Tenneti (2401010186) led the technical implementation of the project. He was responsible for the complete machine learning pipeline — data preprocessing, feature engineering, model training, benchmarking across all five model families, and evaluation. He authored the training notebook `CustomerChurnPrediction.ipynb` in its entirety, including the EDA, the StandardScaler pipeline, the model export logic, and all visualisations including the ROC curve comparison and confusion matrices.

Aaloke Eppalapalli (2401010005) was responsible for the deployment and user interface. He designed and built the Gradio web application (`app.py`), implementing the three-tab input layout covering Demographics, Account & Billing, and Services, the real-time churn prediction output panel, the half-circle risk gauge visualisation, and the three pre-built example customer profiles. He also managed the hosting and ensured the application was publicly accessible via a live URL.

Arina Ali (2401020114) was responsible for the project report. She authored this document in LaTeX, structuring it across the Abstract, Introduction, Methodology, Results, System Architecture, Discussion, and Conclusion sections. She ensured all evaluation metrics, results tables, confusion matrices, and citations were accurately represented, and maintained the overall clarity and professional typesetting of the report.

Shrihari K N (2401010448) was responsible for the project video. He coordinated and produced the five-minute demonstration video, scripting the walkthrough, managing the screen recording, and overseeing the final edit. He ensured the video covered the problem statement, notebook walkthrough, live application demo, GitHub repository overview, and the Milestone 2 roadmap within the allotted time.

References

[1] Reichheld, F. F. and Sasser, W. E. (1990). *Zero defections: Quality comes to services.* Harvard Business Review, 68(5), 105–111.

[2] IBM. (2018). *Telco Customer Churn* [Dataset]. Kaggle. Retrieved from https://www.kaggle.com/datasets/blastchar/telco-customer-churn

[3] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

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

[5] Chen, T. and Guestrin, C. (2016). *XGBoost: A scalable tree boosting system.* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM.