PROJECT ABSTRACT

Customer Churn Prediction using Machine Learning

Acquiring a customer costs far more than retaining one, yet most businesses spot churn only after cancellation lands — when win-back is expensive and often futile.

01 Title

Customer Churn Prediction using Machine Learning

02 Introduction / Background

Acquiring a customer costs far more than retaining one, yet most businesses spot churn only after cancellation lands — when win-back is expensive and often futile. This project builds an early-warning system on telecom-style records: a model learning the signatures that precede cancellation — short tenure, rising charges, repeated support tickets — and flagging at-risk customers in time for retention action.

03 Problem Statement

Given tenure, charges, contract type, support tickets and services, predict which customers will churn. The hurdle is class imbalance: churners are a minority (~25–30%), so a naive model scores high accuracy by predicting that nobody ever leaves — while every prediction must stay explainable enough to guide retention decisions.

04 Objectives

- Flag likely churners with a binary classifier on usage, billing, contract and support features.
- Handle imbalance: compare scale_pos_weight against SMOTE on stratified splits.
- Tune the decision threshold for churn-class recall — a missed churner costs more than a false alarm.
- Ship a ChurnGuard Streamlit dashboard with risk segments, SHAP drivers and a what-if simulator.

05 Proposed System / Working

Stack: Python, scikit-learn, XGBoost (XGBClassifier), imbalanced-learn (SMOTE), SHAP, Streamlit. Records are cleaned and encoded, then split with stratification so the churn minority keeps its share in train and test. XGBoost trains with scale_pos_weight set from the class ratio, tuned via cross-validation; SMOTE runs as a tabulated alternative, and the threshold is scanned on the validation set to maximize churn-class recall within a precision floor. SHAP gives global and per-customer driver breakdowns; models ship as .pkl files for offline demos.

The tuned XGBoost model learns interpretable churn signatures with precision, recall, F1, ROC-AUC and PR-AUC reported from the run, and the scale_pos_weight-vs-SMOTE table defends the imbalance strategy with numbers. The ChurnGuard demo shows risk segments with revenue at risk, per-customer SHAP charts, and the what-if simulator — a contract-upgrade offer moving a high-risk customer's score: a classifier becoming a business tool.

06 Technologies / Components Used

Python 3 · pandas · NumPy · stratified splits · pipelines · precision/recall/F1 · ROC and PR curves · XGBoost · imbalanced-learn · SHAP · Streamlit · Plotly

07 Key Features

- Per-customer churn probability scoring with adjustable decision threshold.
- Risk segmentation (high / medium / low) with customer counts and revenue-at-risk rollups.
- SHAP-ranked drivers: month-to-month contract, short tenure, high charges, frequent support tickets.
- What-if simulator: change contract, tenure or support calls and watch risk move.
- Full metrics view: precision, recall, F1, ROC-AUC and PR curves.

08 Applications / Use Cases

- Telecom operators prioritize retention campaigns with churn scoring
- banks and insurers apply the same pattern to policy lapse and card attrition
- SaaS firms predict cancellation from usage telemetry

10 Conclusion

This project fuses machine learning with business analytics: imbalance handling, threshold economics and SHAP interpretability in a dashboard a retention manager could use. Its honest limits — correlational drivers, not proven causes; behaviour drift demanding retraining — are exactly what makes it viva-ready and industry-credible.