Architecture Design Document

NoMoreChurn – Telco Risk Intelligence

Date: 25/5/2025

Submitted By :-

Soumen Baidya

Gourisankar Maity

Disha Jaiswal

Contain

Introduction

What is Architecture Design Document?

The Architecture Design Document (ADD) provides a comprehensive description of the overall system architecture used to implement the "No More Churn – Telco Risk Intelligence" solution. It outlines how various components of the system interact, the data flow, technical frameworks, modelling strategy, and deployment architecture. It serves as a technical blueprint for developers, data scientists, and DevOps teams.

Scope

This document covers the architectural structure of the end-to-end churn prediction solution using machine learning and Power BI. It includes data ingestion, preprocessing, feature engineering, model training (Random Forest, XG Boost, Ensemble Voting), probability scoring, and final reporting using Power BI dashboards.

Architecture

Architectural Overview

The architecture is modular and consists of five layers:

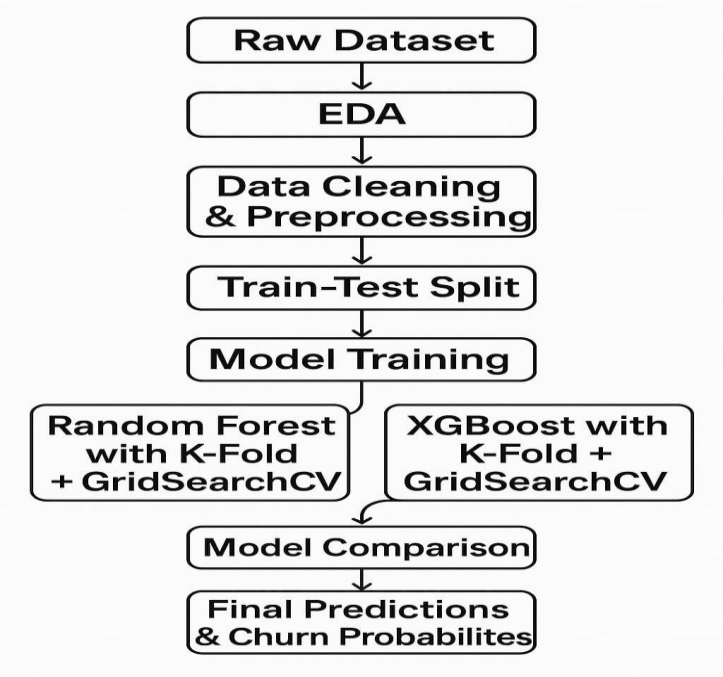
Data Ingestion Layer: **Responsible for sourcing the raw telco customer dataset.**

Data Processing Layer: **Handles cleaning, preprocessing, feature engineering.**

Modelling Layer: **Performs train-test split, model building, evaluation, and selection.**

Scoring & Prediction Layer: Applies trained models to test data and generates churn probabilities.

Visualization Layer: Consumes final prediction output and visualizes it using Power BI.



Data Ingestion

Source: Static CSV file (Telco-Customer-Churn.csv)

Tool: Pandas (Python)

Validations: Column consistency, null value checks

Data Processing

Null value handling (e.g., Total Charges empty values replaced with median)

Label encoding for categorical variables (e.g., Gender, Contract)

Standardization using Standard Scaler for numerical fields

Feature Engineering

Created tenure bins (0–12, 13–24, etc.)

Total services count per customer

Monthly to total charge ratio

One-hot encoding for multiclass features

Modelling Architecture

Random Forest:

Used with 5-Fold Cross Validation

Hyperparameter tuning using Grid SearchCV

Metrics: Accuracy, Precision, Recall, AUC-ROC

XG Boost:

Tuned with learning rate, n\estimators, max\depth

Used early stopping and Grid Search CV

Voting Classifier (Ensemble):

Combined Random Forest and XG Boost using soft voting Selected final model based on best AUC-ROC (0.86)

Scoring & Probability Banding

Final model used to predict churn probability on test set

Generated binary prediction (Yes/No) and probability scores

Probability bands: 0–0.2, 0.2–0.4, 0.4–0.6, 0.6–0.8, 0.8–1.0

Data Export Layer

Final test set included:

customerID

Actual Churn

Predicted Churn

Churn Probability

Tenure, MonthlyCharges, etc.

Exported to: final_powerbi_churn_data.csv

Visualization Architecture

Tool: Power BI Desktop

Data imported from exported CSV

Dashboard structured into 3 sheets:

1. Churn Overview & Model Performance

2. Churn Analysis by Customer Attributes

3. Churn Risk & Ticket Impact

Advanced features:

Churn persona slicer cards

Probability slicers and heatmaps

Drill-through for customer-level insights

Deployment

Development Environment

Python (Jupyter Notebook)

Libraries: pandas, numpy, sklearn, xgboost, matplotlib, seaborn

Model Export

Models trained and optimized in Python

Predictions saved into a structured test set

Power BI Deployment

Data: CSV file from modeling pipeline

Imported as static dataset into Power BI

Interactive filters and drill-through enabled

Dashboard distributed as PBIX file

Scheduled Refresh (Optional/Future Scope)

Weekly or monthly refresh via scheduled Python job

Refreshable data source linked to Power BI Service (future deployment

Storage & Access

Data stored locally or on internal network

PBIX file accessible to management/reporting stakeholders

*Source code versioned via Git