Faculty of Computing
University of Sri Jayewardenepura

Proposal of Machine Learning

On

PREDICTING CUSTOMER CHURN IN TELECOM USING ML AND STREAMLIT-BASED DEPLOYMENT

for

CCS3032 - Machine Learning

by

Group Number 06

FC110531/FC211004 - A. K. G. N. Deshapriya – Project Lead

FC110550/FC211023 - G. P. S. Weerakoon - Data Handler

FC110537/FC211010 - M. L. H. Muthuarachchi - Model Trainer

FC110541/FC211014 – E. W. U. Kalyana - Model Trainer

FC220548/FC211021 – W. A. D. S. S. Wickramasinghe – Deployment lead

1. Problem Statement

The telecommunications industry faces significant challenges with customer churn, where customers discontinue their services. This project aims to predict customer churn for a telecommunications company using historical customer data. The goal is a binary classification task to identify customers likely to churn, enabling proactive retention strategies. This is significant as reducing churn can improve revenue and customer satisfaction.

2. Dataset Information

1. Dataset Name and Source

The dataset used for this project is titled **“Telco Customer Churn”**, and it is publicly available on **Kaggle**, contributed by **Blastchar**. The original data comes from **IBM Sample Data Sets**.

- **URL**: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

2. Dataset Size and Features

This data set has 7,043 records(customer entities) and also it has 21 columns, including the customer ID and target variable. The target variable is **Churn** (binary classification: “Yes” or “No”).

Among the features:

- 13 are categorical (gender, InternetService, Contract, etc.)
- 3 are numerical (tenure, MonthlyCharges, TotalCharges)
- 5 are binary (SeniorCitizen, Partner, Dependents, etc.)

The dataset is self-contained (not multi-table) and does not require external data merging. It is relatively small in size (< 1 MB), making it manageable for exploratory data analysis and model experimentation.

Main Feature Categories:

- **Customer Demographics**:
  - gender, SeniorCitizen, Partner, Dependents
- **Customer Account Information**:
  - tenure, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges

- **Services Signed Up**:
  - PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies

3. Suitability for the Project

The Telco Customer Churn dataset is highly suitable for a churn prediction project due to some reasons. They are **Binary Classifications , Balanced Feature Diversity, Business Relevance and Also this dataset is Well-Structured.**

The target variable (Churn) makes this dataset ideal for developing supervised learning models. It includes both numerical and categorical data, which allows the use of various preprocessing and feature engineering techniques.

4. Preprocessing and Cleaning Plan

To prepare the data for analysis and modeling, several preprocessing steps will be applied. First, any blank or null values in the **TotalCharges** column will be converted to NaN and handled appropriately, either through imputation or removal, depending on their frequency and impact. Since **TotalCharges** is initially stored as a string, it will be converted to a numeric data type to allow for mathematical operations and model compatibility.

Next, categorical variables such as **Contract, PaymentMethod**, and other non-numeric features will be encoded using either label encoding or one-hot encoding, depending on the nature of the variable and the algorithm being used. Numerical features like **MonthlyCharges, TotalCharges, and tenure** will be normalized or standardized to ensure consistent scales across features, which is especially important for distance-based models and those sensitive to feature magnitude.

The target variable Churn, currently represented as "Yes" and "No", will be mapped to binary values 1 and 0 respectively to facilitate binary classification. Additionally, optional feature engineering may be performed, such as grouping the tenure feature into categories (e.g., short-term, mid-term, and long-term customers) to capture different customer lifecycle stages, and creating composite service indicators (e.g., a binary feature representing whether a customer subscribes to any streaming services). These transformations will help enhance model interpretability and performance.

3. Machine Learning Approach

We hope to assess the performance of various classification models on the churn prediction task. The models we will evaluate include:

- **Logistic Regression**: A simple linear model for binary classification, suitable as a baseline due to its interpretability and efficiency.
- **Random Forest**: An ensemble method that builds multiple decision trees, effective for capturing complex feature interactions and handling imbalanced data.
- **Support Vector Machine (SVM)**: A powerful model that finds the optimal hyperplane for classification, effective for high-dimensional data with clear class separation.

Justification:

- Logistic Regression provides a benchmark for performance comparison.
- Random Forest excels in handling non-linear relationships and feature importance analysis.
- SVM is chosen for its ability to maximize the margin between classes. Potentially improving performance with proper kernel selection (e.g., RBF kernel). We will use cross-validation to compare models and select the best performer based on evaluation metrics.

Data Splitting:

- The dataset will be split into training, validation, and testing sets using a 70-15-15 ratio (approximately 4,930 training, 1,056 validation, and 1,057 testing samples).
- We will use stratified splitting to maintain the same proportion of churn/non-churn instances across all sets, addressing potential class imbalance.

4. Evaluation Metrics

We will use the following evaluation metrics to assess model performance:

- **Accuracy**: The proportion of correct predictions out of total predictions, providing an overall measure of model performance.
- **Precision**: The proportion of true positive predictions out of all positive predictions, critical for minimizing false positives in churn prediction.
- **Recall**: The proportion of true positive predictions out of all actual positive cases, essential for identifying most churners.

- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of model performance.
- **ROC-AUC**: The area under the ROC curve, measuring the model's ability to distinguish between classes, suitable for imbalanced datasets.
- **Confusion Matrix**: A table summarizing true positives, true negatives, false positives, and false negatives to provide a detailed view of classification performance.

Success Definition: A model with ROC-AUC ≥ 0.85, high recall (to identify most churners), and reasonable precision (to avoid excessive false positives).

5. Deployment Plan

Deployment Tool

We will deploy the trained model using Streamlit, a lightweight framework ideal for creating interactive web applications for machine learning with minimal overhead. It enables rapid development of dashboards and data input portals with a user-friendly interface.

Environment Setup

To ensure consistency and portability across platforms, we will use Docker to containerize the application along with all dependencies.

Model Integration

The Streamlit application will include:

- Data Preprocessing: Apply the same preprocessing steps as used during training.
- Prediction Logic: Use the model to generate predictions on new input data.

User Interface Design

The user interface will feature two key components:

- Dashboard: Displays key performance metrics and visualizations.
- Data Input Portal: Allows users to submit input data and view model predictions.

UI Development Steps:

- Design the dashboard using Streamlit to display KPIs and visualizations.
- Implement the input form to collect user data.
- Display prediction results with clear outputs and relevant insights.
- Deployment Workflow
- Develop the full application, integrating UI and model logic.

- Test thoroughly to ensure correctness and robustness.
- Build the Docker image for the complete application.
- Run the Docker container to verify local deployment.
- Deploy to cloud using a platform like AWS, Google Cloud, or Heroku for public access.

Maintenance and Updates

- Monitor application performance and gather user feedback.
- Retrain and update the model periodically with fresh data.
- Debug and patch any issues that arise during production usage.

6. Project Timeline

Week 1: Data Collection and Preprocessing

Week 2: Model Training and Evaluation

Week 3: Deployment and Documentation

7. Challenges and Risks

RISK	IMPACT	LIKELIHOOD	MITIGATION STRATEGY
Data quality issues	High	Medium	Implement robust data cleaning, handle missing values, outliers, and validate inputs.
Model overfitting	High	High	Use cross-validation, regularization, and systematic hyperparameter tuning.
Deployment complexity	Medium	Medium	Conduct end-to-end integration testing and follow best practices in containerization.
Resource constraints	Medium to High	Medium	Optimize code, batch processing, and utilize cloud platforms for scalability.
Ui/ux design challenges	Medium	Medium	Follow user-centered design, perform usability testing, and iterate based on feedback.
Maintenance and updates	Medium	High	Define a lifecycle management plan with scheduled model retraining and performance checks.

Table 9.1 – Challenges and Risks

08. References

- **Dataset**: Telco Customer Churn dataset from Kaggle: https://www.kaggle.com/datasets/blastchar/telco-customer-churn