Customer Churn Prediction Model

Project Documentation

1. Introduction

Customer churn happens when a customer decides to stop using a company's services. For example, in a telecom company, if a customer cancels their subscription, we call it churn.

For businesses, preventing churn is very important because retaining existing customers is always cheaper than acquiring new ones. This project focuses on building a **machine learning solution** that predicts whether a customer will churn or not, based on their service usage and billing information.

The final model is deployed as an easy-to-use **Streamlit web app**, so that business teams can get predictions and insights without touching the code.

2. Problem Statement

The main goal of this project is:

- To analyze customer data and predict churn (Yes/No).
- To identify the factors driving churn.
- To help businesses take preventive steps (discounts, offers, better customer support).
- To build an interactive application that demonstrates the predictions visually.

3. Dataset Description

The dataset contains customer information with details about their services, contracts, and billing.

Features included:

- **Demographics**: Gender, SeniorCitizen, Partner, Dependents
- **Services**: PhoneService, InternetService, OnlineSecurity, TechSupport, StreamingTV, StreamingMovies
- **Billing & Contracts**: Contract type, Payment method, MonthlyCharges, TotalCharges
- **Tenure**: Number of months the customer has been with the company
- **Target variable**: **Churn** (Yes = customer left, No = customer stayed)

This combination of features helps the model learn patterns of customers who are more likely to churn.

4. Data Preprocessing

Before training the model, the dataset went through cleaning and preparation steps:

- Handled missing values by filling or removing them.
- Converted categorical values (like Gender, Contract type) into numeric format using encoding.
- Scaled numerical features (MonthlyCharges, TotalCharges) so that values are in a comparable range.
- Split the dataset into **training (80%)** and **testing (20%)** sets for evaluation.

5. Machine Learning Models

Different algorithms were trained and compared:

- **Logistic Regression**
  - A simple but effective model that gives the probability of churn.
  - Easy to interpret, which makes it useful for business explanations.
- **Decision Tree**
  - Use a series of if-else rules to classify customers.
  - Example: *If contract = month-to-month and tenure < 2 → churn likely.*
- **Random Forest**
  - Combines multiple decision trees to reduce errors and improve accuracy.
  - More stable than a single decision tree.
- **XGBoost**
  - A powerful boosting algorithm that focuses on improving weak predictions.
  - Known for high accuracy and widely used in industry.

6. Model Evaluation

To check how well the models perform, several metrics were used:

- **Accuracy** → Percentage of correct predictions.
- **Confusion Matrix** → Breaks predictions into 4 groups:

Prediction	**Churn**	**No Churn**
Actual Churn	TP	FN
Actual No Churn	FP	TN

- **Precision** → Of the customers predicted as churn, how many churned?
- **Recall** → Of all customers who churned, how many did we correctly predict?
- **F1 Score** → Balance between precision and recall.
- **ROC Curve & AUC** → Graph that shows how well the model separates churners and non-churners. AUC close to 1 means the model performs well.

7. Exploratory Data Analysis (EDA)

To better understand the data, several visualizations were created:

- Bar chart → Number of churn vs non-churn customers
- Pie chart → Distribution of contract types
- Box plot → Monthly charges vs churn
- Heatmap → Correlation between features
- Confusion matrix heatmap → Errors made by the model
- ROC curve → Model performance visualization
- Feature importance plot → Shows which factors contribute most (e.g., Contract type, Tenure, Monthly Charges)

These graphs are also displayed in the final Streamlit app.

8. Tools and Libraries

- **Python** → Programming language
- **pandas, numpy** → Data cleaning and manipulation
- **matplotlib, seaborn** → Visualizations
- **scikit-learn** → ML algorithms (Logistic Regression, Decision Tree, Random Forest)
- **xgboost** → Gradient boosting model
- **pickle/joblib** → Save and load trained models
- **Streamlit** → Web app deployment framework

9. Streamlit Application

The model was deployed using **Streamlit** to make it interactive and user-friendly.

Features of the app:

- Users can upload a CSV file or manually enter customer details.
- The model predicts whether each customer is likely to churn.
- Shows the churn probability along with prediction results.
- Displays important graphs like confusion matrix, ROC curve, and feature importance.

This makes the solution useful for business teams who don't have technical expertise.

10. Workflow

1. Data collection and preprocessing
2. Exploratory Data Analysis (EDA) with visualizations
3. Model training using Logistic Regression, Decision Tree, Random Forest, XGBoost
4. Model evaluation with Accuracy, Precision, Recall, F1-score, ROC-AUC
5. Selection of the best model based on performance
6. Saving the model with pickle
7. Deployment with Streamlit including visual dashboards

11. Business Value

- Helps identify customers most likely to leave.
- Businesses can take early action by offering discounts, personalized offers, or better support.
- Improves customer retention and long-term revenue.
- Provides data-driven insights into why customers churn.

12. Conclusion

This project demonstrates how machine learning can be applied to predict customer churn. Logistic Regression gave interpretable insights, while Random Forest and XGBoost delivered higher accuracy.

The Streamlit web app makes predictions and insights accessible to business users in an interactive way. By identifying churn-prone customers in advance, companies can reduce customer loss and increase overall profitability.