# Customer Project Steps

# Churn Prediction – Data Science

## 1. Understand theBusiness Problem

Customer churn is a critical business metric in the telecom industry. Retaining customers is often more cost-effective than acquiring new ones. This project aims to predict whether a customer is likely to leave the service (churn) based on their usage patterns, complaints, support history, and service preferences.

## 2. Data Collection and Preparation

We use a synthetic dataset containing 10,000 customer records. Features include:

- Customer Age
- Monthly Charges
- Total Charges
- Contract Type
- Internet Service Type
- Technical Support Availability
- Tenure in Months
- Payment Method
- Streaming Services
- Customer Support Calls
- Target: Churn (0 = Stay, 1 = Churn)

The dataset includes 5% null values in each feature (excluding the target). Handle missing values, encode categorical variables, and scale numerical data as part of preprocessing.

## 3. Exploratory Data Analysis (EDA)

Use visualizations to uncover patterns and correlations:

- Countplot: Churned vs Non-Churned Customers
- Countplot: Contract Type vs Churn
- Bar Chart: Internet Service vs Churn
- Violin Plot: Monthly Charges vs Churn
- Histogram: Tenure Distribution
- Heatmap: Correlation between numerical variables
- Boxplot: Customer Support Calls vs Churn

## 4. Feature Engineering

Encode categorical features (One-Hot Encoding or Label Encoding), impute missing values using mean/mode, and normalize numeric columns like charges and tenure. Optionally, derive features such as 'is_long_term_customer'.

## 5. Model Selection

Begin with Logistic Regression for baseline results. Then explore models like Random Forest, Decision Tree, or XGBoost to improve accuracy.

## 6. Model Training

Split data into train-test sets (80/20). Train the selected model using features and labels.

## 7. Model Evaluation

Evaluate the model using the following metrics:

- Accuracy
- Precision
- Recall
- F1 Score
- Confusion Matrix
- ROC Curve

## 8. Model Saving and Deployment

After evaluation, save the model using `joblib` or `pickle`. Deploy it through a web service using Flask or FastAPI, with a front-end UI to allow inputs and view predictions.