

**Github Link :** <https://github.com/HardikAbinesh/Project-Predicting-Customer-Churn.git>

## **Project Title : Predicting Customer Churn Using Machine Learning to Uncover Hidden Patterns**

### **PHASE-2**

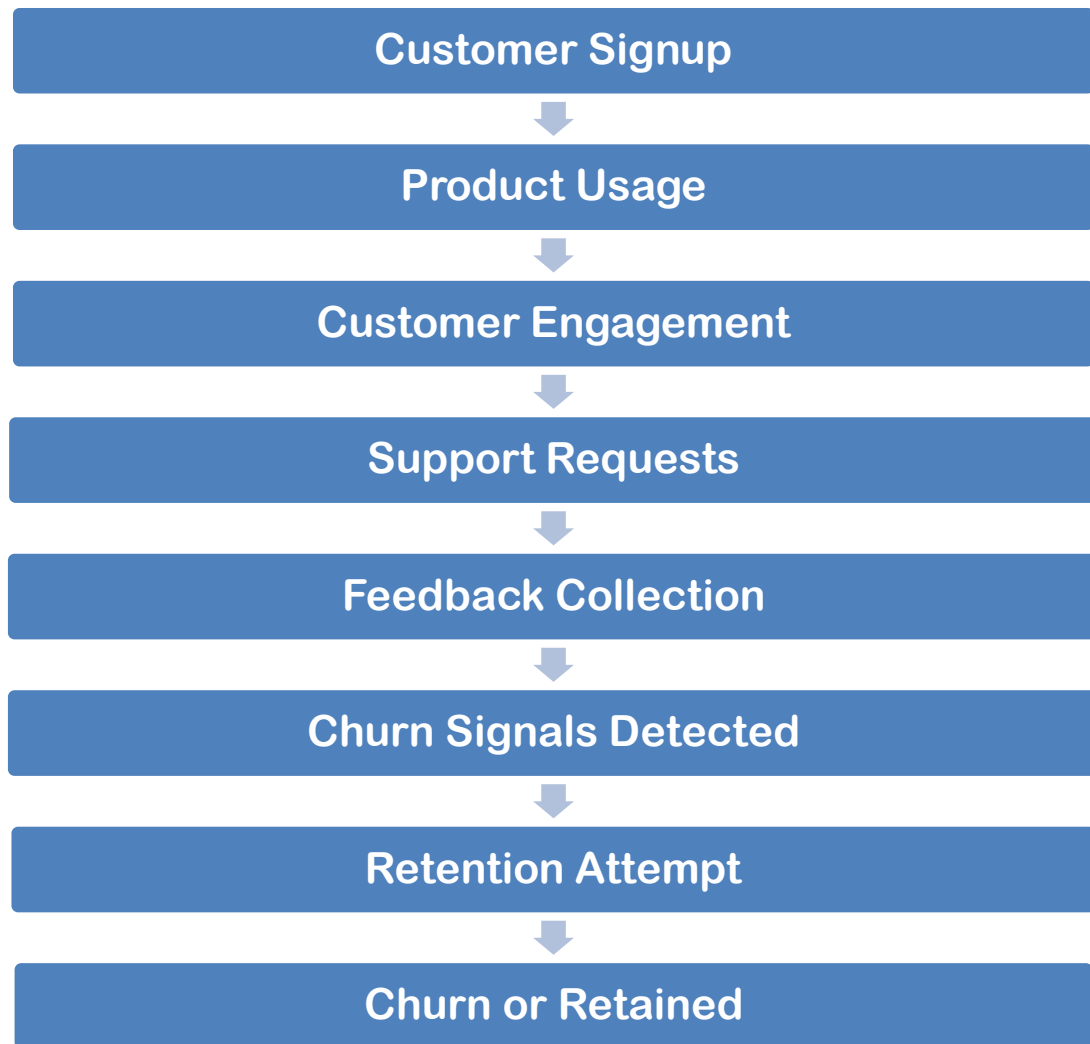
#### **1. Problem Statement**

- Customer churn poses a significant challenge for businesses, directly impacting revenue and growth. Traditional methods often fail to detect early signals of churn, especially when hidden within complex customer behavior patterns.
- This project aims to develop a machine learning-based solution to accurately predict customer churn by uncovering subtle and non-obvious patterns in customer data.

#### **2. Project Objectives**

- Collect and preprocess customer data to ensure quality and consistency for model training.
- Analyze the data to identify patterns and trends related to customer churn.
- Engineer relevant features that effectively represent customer behavior and engagement.
- Develop and compare machine learning models to accurately predict customer churn.
- Uncover hidden patterns and key factors influencing churn using advanced analytics techniques.

#### **3. Flowchart of the Project Workflow**



#### 4. Data Description

- **Dataset Name:** Credit Card Customer Churn Prediction
- **Source:** IBM Sample Data Repository / Kaggle / Other public repositories
- **Type of Data:** Structured tabular data
- **Records and Features:** 10,127 customer records and 23 features (combination of categorical and numerical)
- **Target Variable:** Attrition\_Flag (binary: Existing Customer / Attrited Customer)
- **Static or Dynamic:** Static dataset

- **Attributes Covered:**

- **Demographics:** Customer\_Age, Gender, Dependent\_count, Education\_Level, Marital\_Status, Income\_Category
- **Account Information:** Customer\_ID, Card\_Category, Months\_on\_book, Total\_Relationship\_Count
- **Credit Behavior:**
  - Credit\_Limit, Total\_Revolving\_Bal, Avg\_Open\_To\_Buy
- **Transaction Behavior:**
  - Total\_Trans\_Amt, Total\_Trans\_Ct, Total\_Ct\_Chng\_Q4\_Q1
- **Utilization and Risk Indicators:**
  - Avg\_Utilization\_Ratio, Contacts\_Count\_12\_mon
- **Dataset Link:** <https://www.kaggle.com/datasets/rjmanoj/credit-card-customer-churn-prediction>

## 5. Data Preprocessing

- **Handling Missing Values:** Impute or remove missing values in the dataset.
- **Data Type Conversion:** Convert columns to appropriate data types.
- **Encoding Categorical Variables:** Apply one-hot encoding or label encoding to categorical features.
- **Feature Scaling:** Standardize or normalize numerical features for consistency.
- **Outlier Detection and Feature Selection:** Identify outliers and remove irrelevant features for model accuracy.

## 6. Exploratory Data Analysis (EDA)

- **Univariate Analysis:**
  - **Histogram of 'Churn'** to understand the class distribution and assess class imbalance.
  - **Boxplots** for numerical variables such as monthly charges, tenure, and total charges to detect outliers and understand their distribution relative to churn status.
  - **Count plots** for categorical features (e.g., contract type, internet service, payment method) to explore frequency distributions and potential churn patterns.

- ***Bivariate & Multivariate Analysis:***

- **Correlation matrix** shows that tenure and monthly charges have moderate relationships with churn likelihood, while total charges correlate highly with tenure.
- **Bar plots and boxplots** comparing features like contract type, tech support, and payment method against churn status reveal notable trends—e.g., customers on month-to-month contracts churn more frequently.
- **Stacked/grouped bar charts** highlight how services like online security, streaming services, and tech support affect churn behavior.
- **Pair plots and scatter plots** identify clusters and trends among continuous variables segmented by churn.

- ***Key Insights:***

- **Contract type** is a strong predictor—month-to-month customers are far more likely to churn than those on longer-term contracts.
- **Tech support and online security** are associated with lower churn—customers with these services tend to stay.
- **Higher monthly charges** slightly increase churn risk, especially for short-tenure customers.
- **Tenure** inversely correlates with churn—longer-tenured customers are less likely to leave.

## **7. Feature Engineering**

- **Created interaction features:**

- `total_services` = count of services subscribed (e.g., internet, phone, streaming)
- `has_streaming_and_support` = flag indicating if a customer has both streaming and tech support services

- **Derived binary features:**

- `is_month_to_month` = 1 if the contract type is "Month-to-month", else 0
- `is_senior_citizen` = converted from numeric (0/1) to meaningful binary representation

- **Handled multicollinearity** by removing or combining highly correlated variables like `total_charges` and `monthly_charges × tenure`

- **Performed label encoding** for binary categorical variables (e.g., Partner, Dependents, PaperlessBilling)
- **One-hot encoded** multi-class categorical features like Contract, PaymentMethod, and InternetService
- **Scaled numeric features** such as tenure, monthly\_charges, and total\_charges using **StandardScaler** to standardize input for machine learning models

## **8. Model Building**

### **• Algorithms Used:**

- **Logistic Regression:** as a baseline linear classifier
- **Random Forest Classifier:** for capturing non-linear relationships and feature importance
- **XGBoost Classifier** (*optional addition*): for high performance and better handling of imbalanced data

### **• Model Selection Rationale:**

- **Logistic Regression:** simple, fast, and interpretable; establishes a benchmark
- **Random Forest:** robust to overfitting, works well with both categorical and numerical data, and provides feature importance scores
- **XGBoost:** optimized for classification with imbalanced datasets and offers better performance through boosting

### **• Train-Test Split:**

- **80% training, 20% testing**
- Used `train_test_split` with `random_state` for reproducibility

### **• Evaluation Metrics:**

- **Accuracy:** Overall correctness of predictions
- **Precision & Recall:** Especially important due to the cost of false positives/negatives
- **F1 Score:** Balances precision and recall in a single metric
- **ROC-AUC Score:** Measures model's ability to distinguish between churners and non-churners

## **9. Visualization of Results & Model Insights**

### • ***Feature Importance:***

- **Bar plots from Random Forest and XGBoost** revealed key drivers of churn:
  - Contract type, tenure, and tech support ranked highest
  - Features like monthly charges and payment method also showed moderate influence

### • ***Model Comparison:***

- **Plotted evaluation metrics** (Accuracy, F1 Score, ROC-AUC) for each model:
  - **Random Forest and XGBoost** outperformed **Logistic Regression**, especially in ROC-AUC and F1 Score
  - Visualized using bar and line plots for clarity

### • ***Confusion Matrix & ROC Curve:***

- **Confusion matrix** highlighted true positives and false negatives, helping assess model reliability
- **ROC Curve** plotted for all models to visualize trade-offs between sensitivity and specificity

### • ***User Testing:***

- **Built an interactive Gradio or Streamlit web interface:**
  - Allowed users to input customer details and instantly receive churn prediction
  - Helpful for stakeholders to explore model behavior and test scenarios

## **10. Tools and Technologies Used**

☐ **Programming Language:** Python 3

☐ **Notebook Environment:** Google Colab (for development and experimentation)

☐ **Key Libraries & Frameworks:**

- **pandas, numpy** : for efficient data manipulation and analysis
- **matplotlib, seaborn, plotly** : for exploratory and result visualizations
- **scikit-learn** : for preprocessing, model training, and evaluation

- **XGBoost** (*optional*) : for gradient boosting classifier implementation
- **Gradio** or **Streamlit** : to build an interactive web app for churn prediction

## **11. Team Members and Contributions**

### ***1. R.Abinesh – Model Development & Evaluation***

- Implemented Logistic Regression, Random Forest, and XGBoost models.
- Tuned hyperparameters and performed train-test splits.
- Evaluated models using accuracy, precision, recall, F1 score, and ROC-AUC metrics.

### ***2. R.Abishek – Model Development & Evaluation***

- Implemented Logistic Regression, Random Forest, and XGBoost models.
- Tuned hyperparameters and performed train-test splits.
- Evaluated models using accuracy, precision, recall, F1 score, and ROC-AUC metrics.

### ***3. K.Arulkumaran – Data Collection & Cleaning***

- Responsible for loading and cleaning the customer churn dataset.
- Handled missing values, inconsistencies, and prepared the dataset for analysis.
- Ensured proper encoding and formatting of categorical variables.

### ***4. V.Bavithran – Visualization & Interface Deployment***

- Visualized feature importance, confusion matrices, and ROC curves.
- Built an interactive churn prediction interface using Gradio.
- Documented insights and assisted with report preparation and presentation.