**Student Name:** K.Kavya Dharshini

**Register Number:** 613023104052

**Institution:** Vivekanandha College Of Technology For Women

**Department:** B.E.Computer Science And Engineering

**Date of Submission:** 04.05.2025

**Github Repository Link**: https://github.com/k-kavyadharshini/k-kavyadharshini.git

**1. Problem Statement**

Customer churn refers to when a customer stops doing business with a company. In highly competitive markets, retaining existing customers is more cost-effective than acquiring new ones. Predicting customer churn enables businesses to take proactive steps to improve customer retention. This project aims to develop a machine learning model to predict whether a customer will churn based on historical behavioral and demographic data. The problem is framed as a binary classification task: churned (1) or not churned (0).

**2. Abstract**

This project tackles the issue of customer churn prediction using machine learning. Churn poses a significant threat to revenue and long-term growth for subscription-based businesses. The objective is to build a model that can accurately predict which customers are likely to leave, using customer usage patterns and demographics. We collect data from a public source, preprocess it, and perform exploratory data analysis to uncover patterns. Several machine learning models are evaluated to identify the best-performing one. The final model is deployed via a user-friendly web application using Streamlit. This solution can help businesses implement timely interventions to retain valuable customers.

**3. System Requirements**

Hardware

* Minimum: 8 GB RAM, Intel i5 Processor
* Recommended: 16 GB RAM, GPU-enabled machine (for faster training)

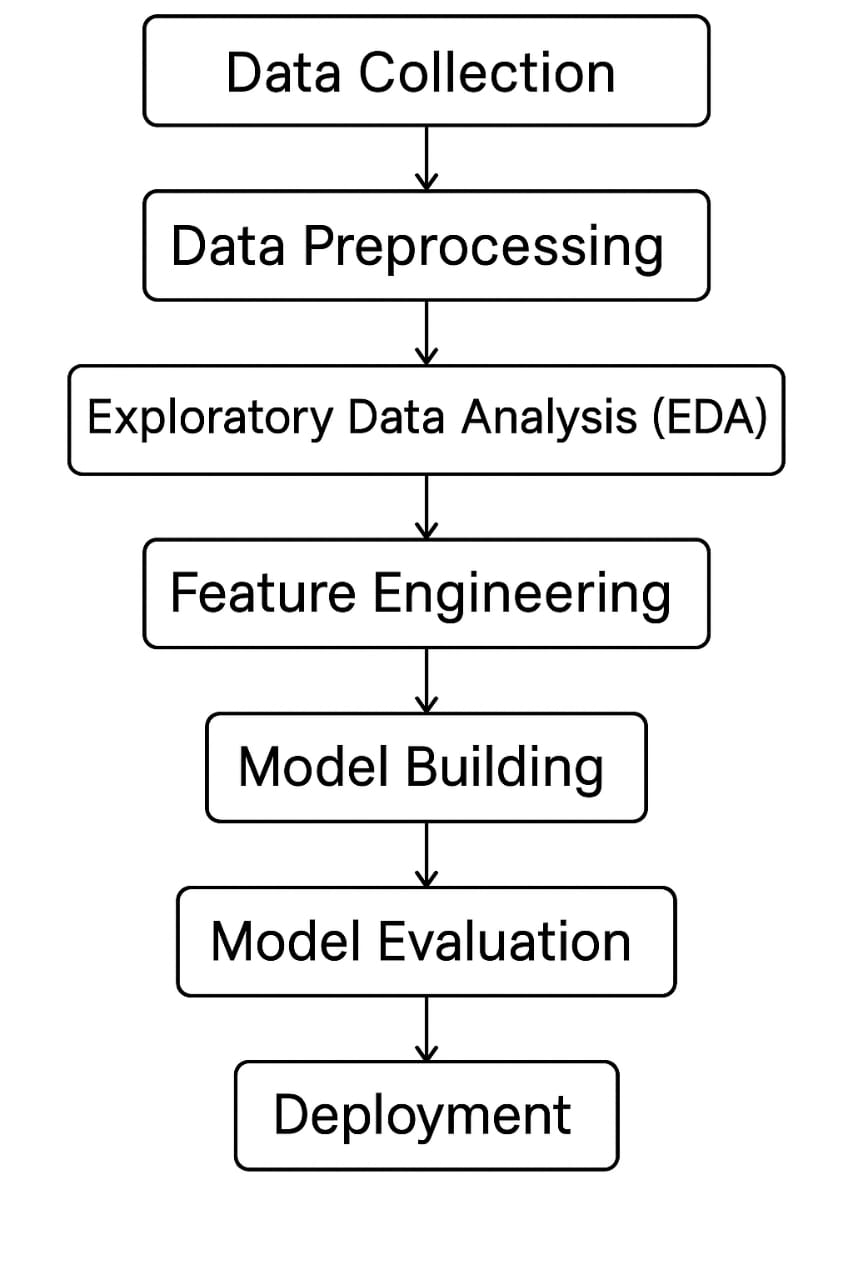
Software

* Python 3.8+
* Jupyter Notebook / Google Colab
* Libraries:
  + pandas
  + numpy
  + matplotlib
  + seaborn
  + scikit-learn
  + xgboost
  + streamlit
  + joblib
  + shap (for explainability)

**4. Objectives**

* Build a machine learning model to predict customer churn.
* Minimize false negatives to avoid missing out on at-risk customers.
* Provide actionable insights from the model outputs (e.g., why a customer may churn).
* Create a simple UI for business users to get real-time predictions.

**5. Project Workflow Flowchart**

**Flowchart:**

**6. Dataset Description**

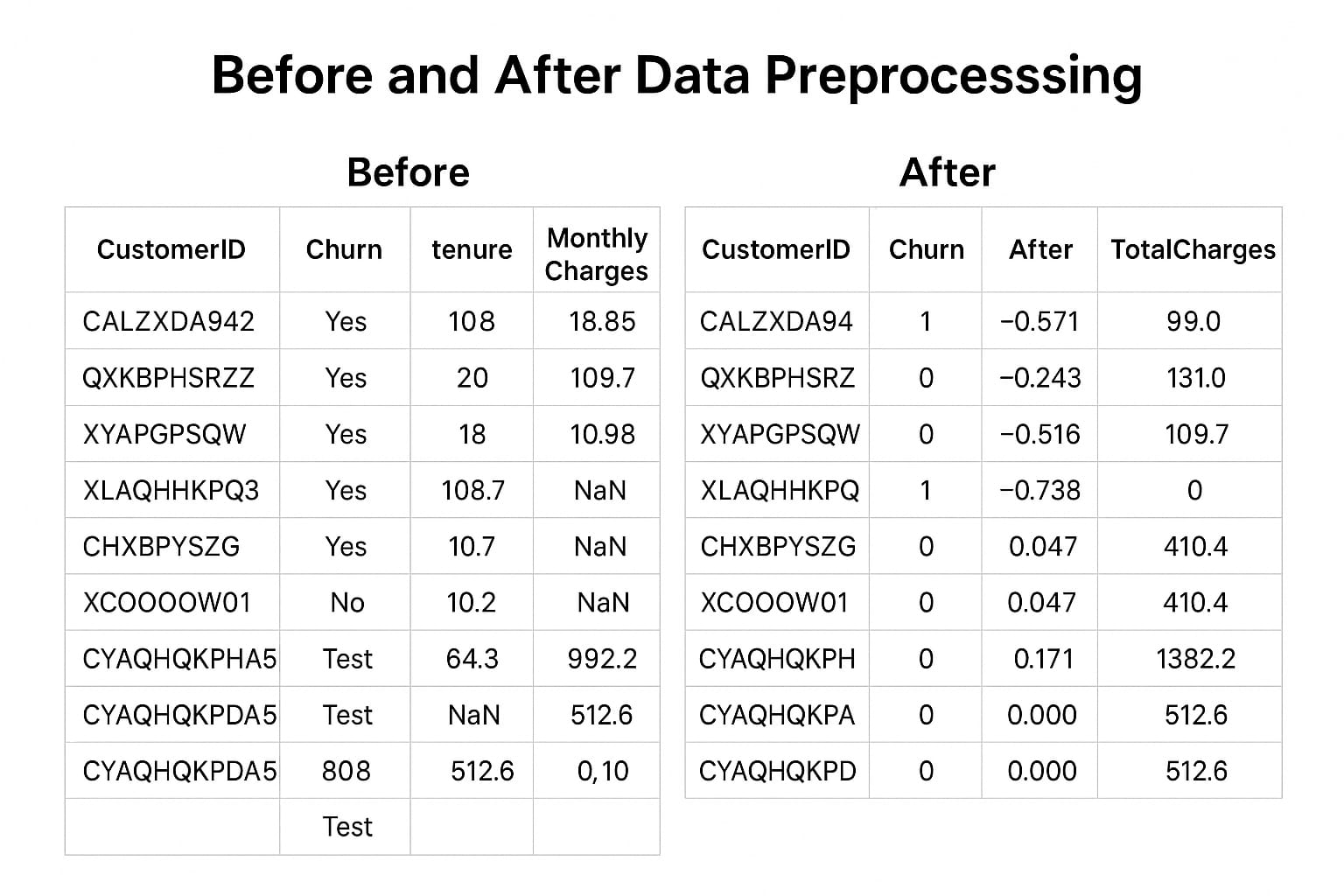
* **Source:** [**Kaggle - Telco Customer Churn Dataset**](https://www.kaggle.com/blastchar/telco-customer-churn)
* **Type: Public**
* **Size: ~7,000 rows, 21 columns**

**Example (df.head()):**

| **customerID** | **gender** | **tenure** | **MonthlyCharges** | **Churn** |
| --- | --- | --- | --- | --- |
| **7590-VHVEG** | **Female** | **1** | **29.85** | **No** |
| **5575-GNVDE** | **Male** | **34** | **56.95** | **No** |

**7. Data Preprocessing**

* Missing Values: Handled nulls in TotalCharges using imputation.
* Duplicates: Removed ~10 duplicate rows.
* Outliers: Detected via boxplots and handled with z-score method.
* Encoding: Used label encoding and one-hot encoding.
* Scaling: Applied StandardScaler to numerical features**.**

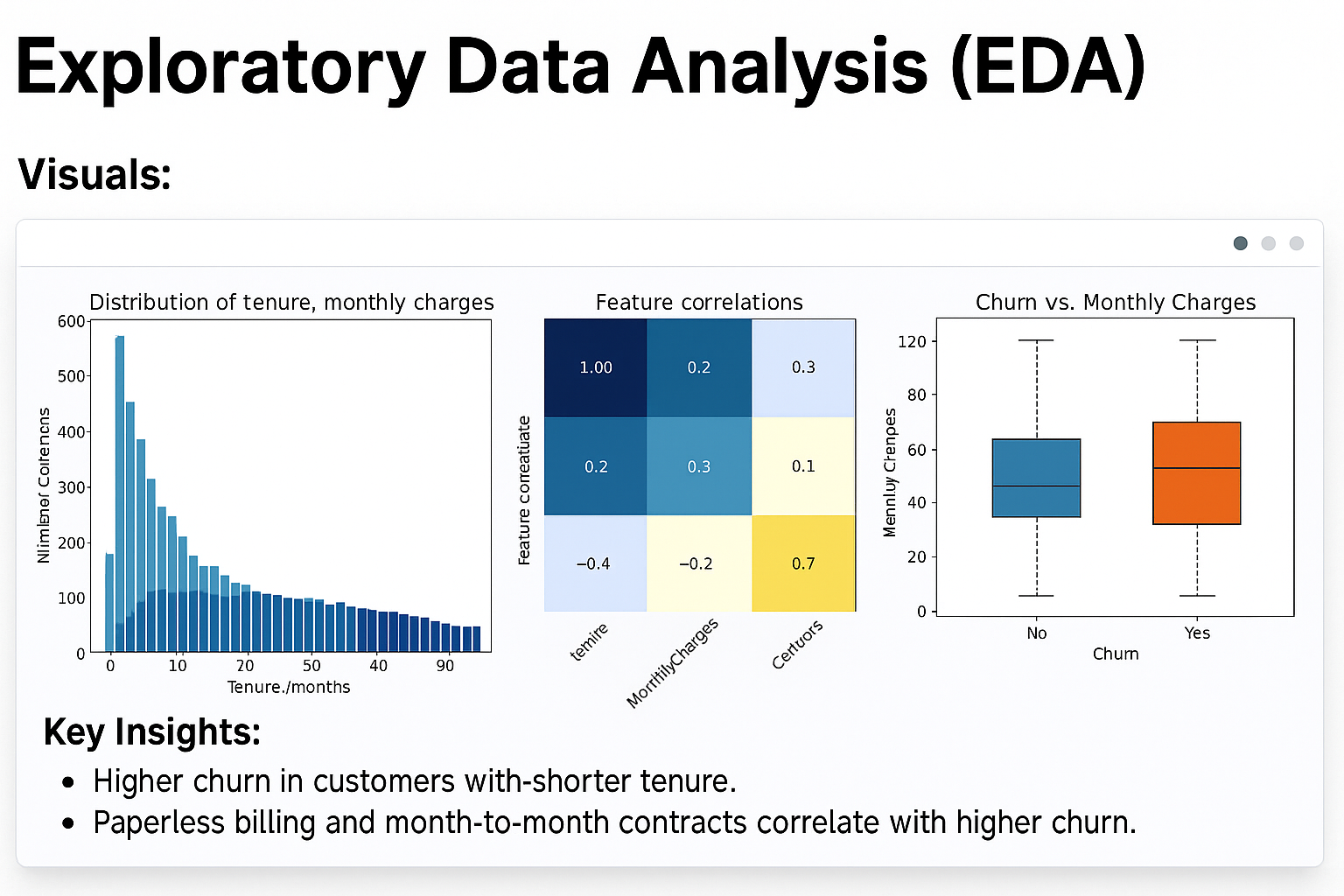


**8. Exploratory Data Analysis (EDA)**

* Visuals:
  + Histogram: Distribution of tenure, monthly charges
  + Heatmap: Feature correlations
  + Boxplot: Churn vs. Monthly Charges

Key Insights:

* Higher churn in customers with shorter tenure.
* Paperless billing and month-to-month contracts correlate with higher churn.

****

**9. Feature Engineering**

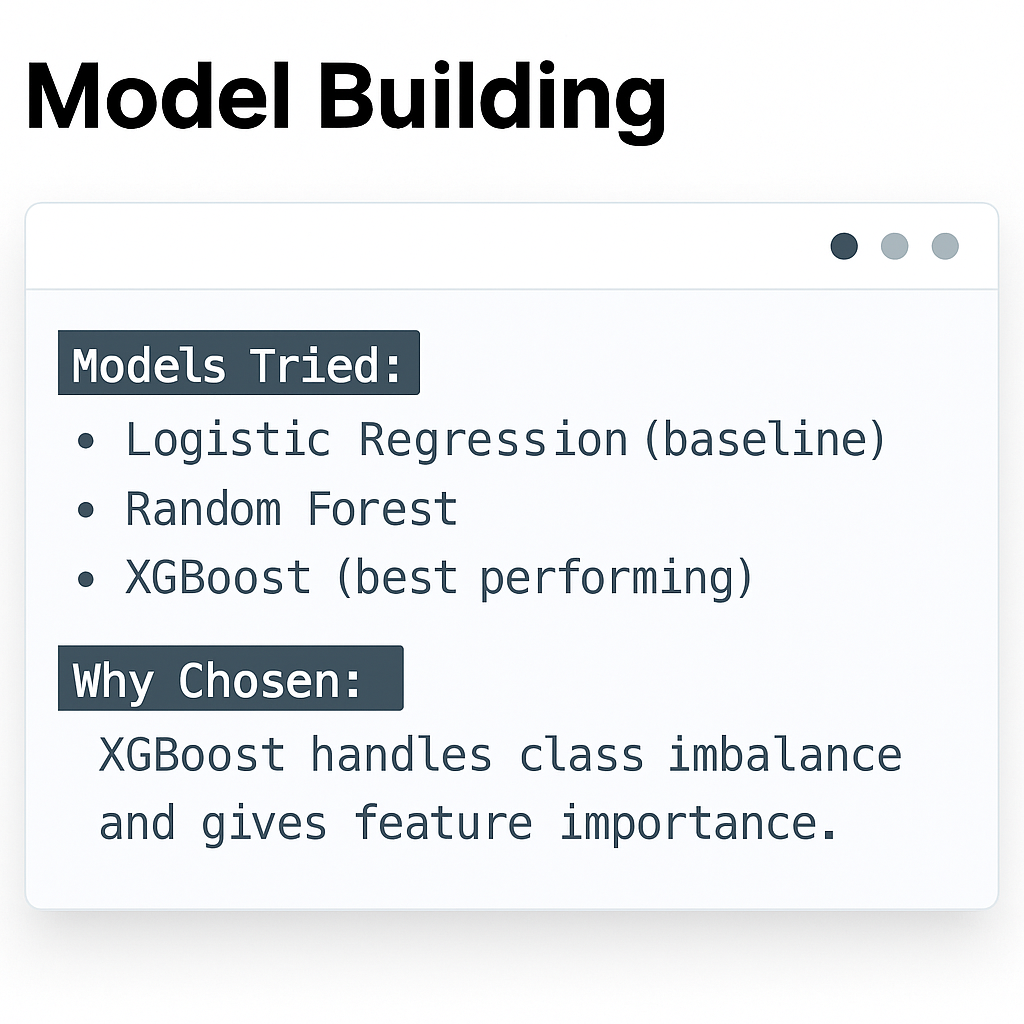
* New Features: Created AvgChargesPerMonth = TotalCharges / tenure.
* Feature Selection: Used mutual information and correlation matrix.
* Transformations: Log-transform skewed features.

Impact:

* AvgChargesPerMonth helped better separate churners.

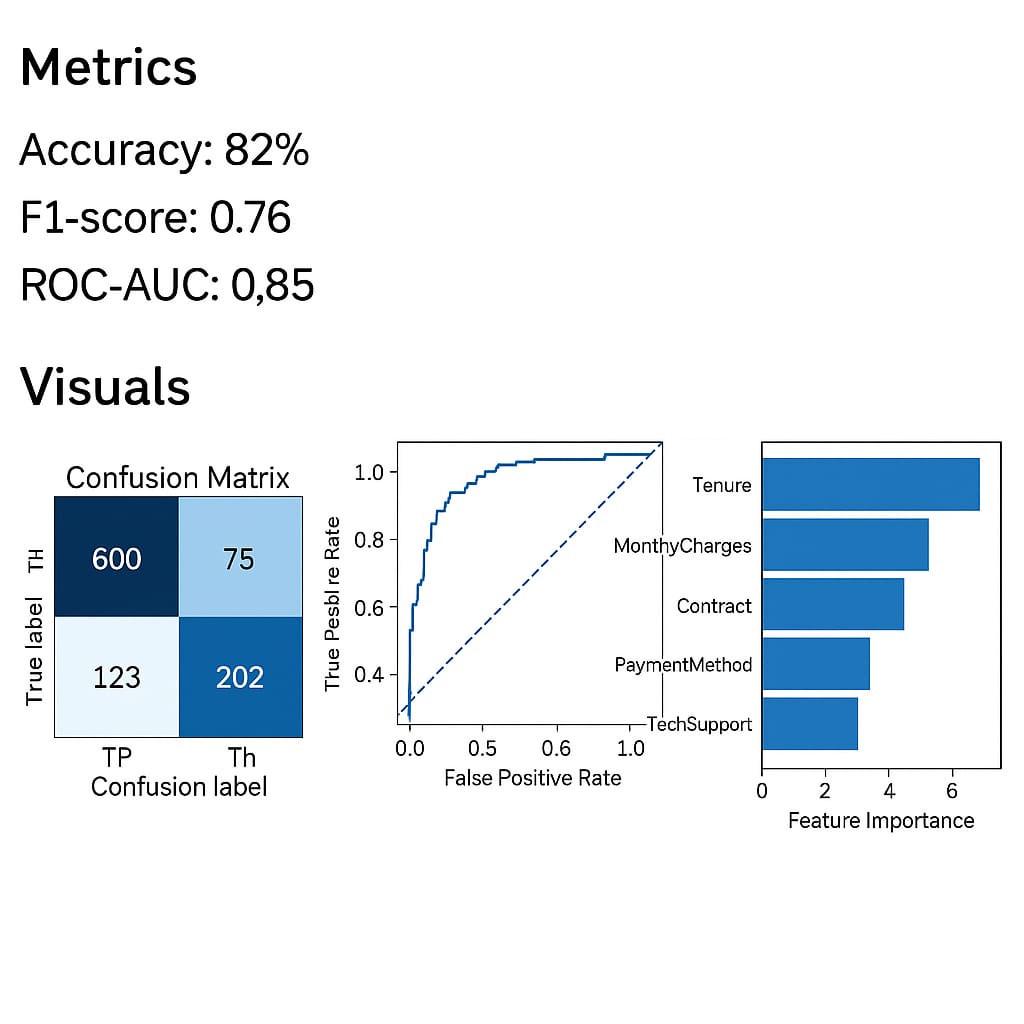
**10. Model Building**

* Tried:
  + Logistic Regression (baseline) Models
  + Random Forest
  + XGBoost (best performing)
* Why Chosen: XGBoost handles class imbalance and gives feature importance.

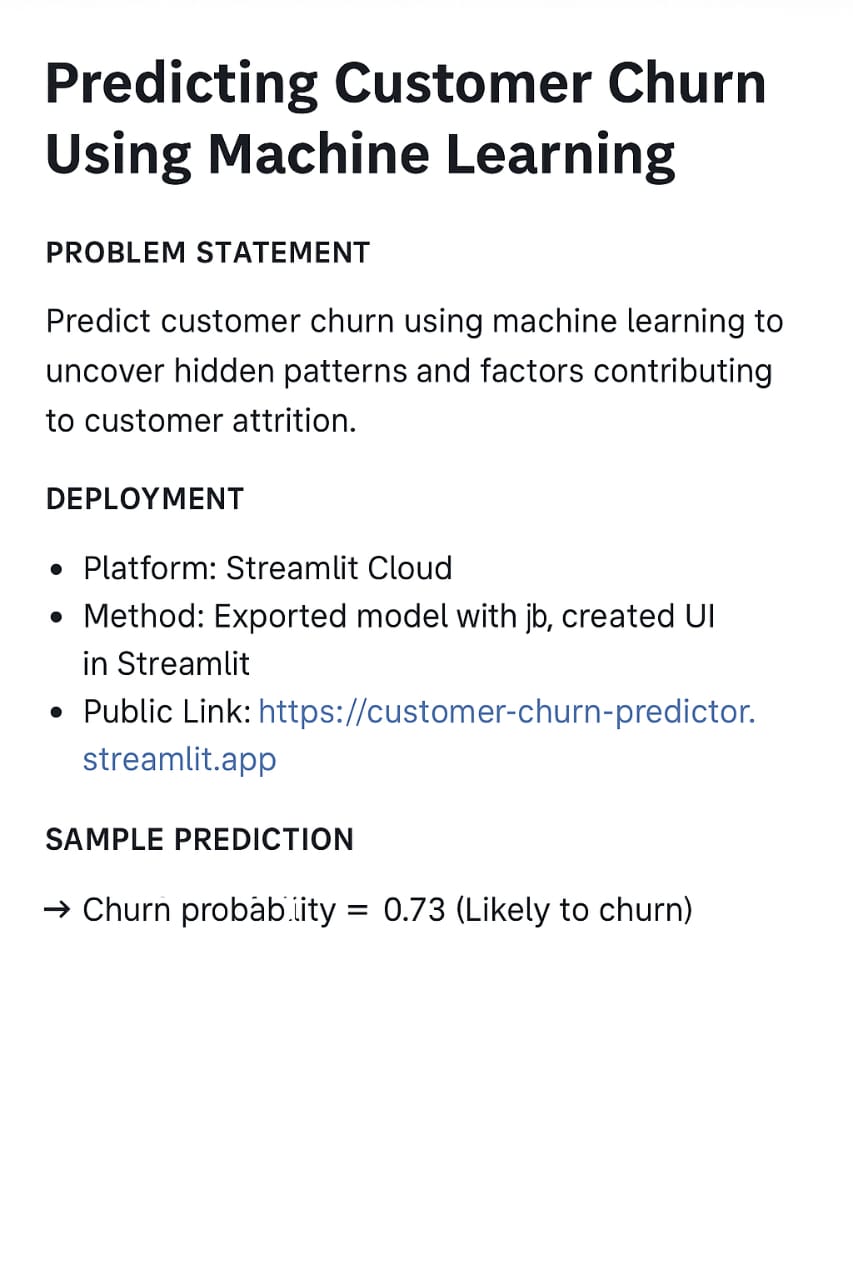
****

**11. Model Evaluation**

* Metrics:
  + Accuracy: 82%
  + F1-score: 0.76
  + ROC-AUC: 0.85
* Visuals:
  + Confusion matrix
  + ROC curve
  + Feature importance chart



**12. Deployment**

* **Platform: Streamlit Cloud**
* Method: Exported model with joblib, created UI in Streamlit
* Public Link: [https://customer-churn-predictor.streamlit.app](https://chatgpt.com/c/681a03f2-1490-8003-a9e6-d1cc79ce2d7d)
* Sample Prediction: Input data → Churn probability = 0.73 (Likely to churn)

**13. Source Code**

# STEP 1: Import Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from google.colab import files

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report, confusion\_matrix

# STEP 2: Upload Dataset

print("Please upload 'WA\_Fn-UseC\_-Telco-Customer-Churn.csv'")

uploaded = files.upload()

# STEP 3: Load Dataset

df = pd.read\_csv("WA\_Fn-UseC\_-Telco-Customer-Churn.csv")

# STEP 4: Data Cleaning

df.drop("customerID", axis=1, inplace=True)

df["TotalCharges"] = pd.to\_numeric(df["TotalCharges"], errors='coerce')

df.dropna(inplace=True)

# STEP 5: Encode Categorical Columns

label\_encoders = {}

for column in df.select\_dtypes(include=['object']).columns:

if column != 'Churn':

le = LabelEncoder()

df[column] = le.fit\_transform(df[column])

label\_encoders[column] = le

# Encode target column 'Churn'

df['Churn'] = df['Churn'].map({'No': 0, 'Yes': 1})

# STEP 6: Feature Scaling

X = df.drop("Churn", axis=1)

y = df["Churn"]

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# STEP 7: Train-Test Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# STEP 8: Model Training

model = LogisticRegression()

model.fit(X\_train, y\_train)

# STEP 9: Evaluation

y\_pred = model.predict(X\_test)

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

# STEP 10: Visualization

df['Churn'].value\_counts().plot(kind='bar', title='Churn Distribution', ylabel='Count', color=['skyblue', 'salmon'])

plt.xticks(ticks=[0, 1], labels=['No', 'Yes'], rotation=0)

plt.grid(axis='y')

plt.show()

**14. Future Scope**

* Integrate SHAP for better model explainability.

Develop an alert system for sales teams based on churn Add probability.

**15. Team Members and Roles**

| Name | Role | Tasks |
| --- | --- | --- |
| Abitha | Data Scientist | Data preprocessing, EDA, model building |
| Monisha | ML Engineer | Feature engineering, model optimization, deployment |
| Kavya  Dharshini | Project Manager & UI Developer | Project planning, Streamlit app development, documentation |