**GithubLink:** [**https://github.com/hariganesh200311/Predicting-customer-churn-using-machine-learning-to-uncover-hidden-patterns-.git**](https://github.com/hariganesh200311/Predicting-customer-churn-using-machine-learning-to-uncover-hidden-patterns-.git)

**Project Title: Predicting customer churn using machine learning to uncover hidden patterns**

**Student Name:** HARI GANSEH.B

**Register Number:** 513523106301

**Institution:** AMCET

**Department:** ECE

**Date of Submission:** 5.5.2025

**GithubRepositoryLink:** <https://github.com/hariganesh200311/Predicting-customer-churn-using-machine-learning-to-uncover-hidden-patterns-.git>

**1. Problem Statement**

*In today’s competitive market, retaining existing customers is more cost-effective than acquiring new ones. However, many businesses struggle to identify which customers are likely to churn before it’s too late. Despite the availability of large volumes of customer data, traditional analytical methods often fail to capture complex, nonlinear patterns that lead to churn.*

*The objective of this project is to develop a predictive machine learning model that can accurately forecast customer churn by analyzing historical behavioral, transactional, and demographic data.*

*The model should not only predict churn probability but also uncover hidden patterns and key factors that contribute to customer attrition. These insights will enable businesses to take proactive, data-driven retention strategies and improve overall customer lifetime value.*

**2. Project Objectives**

***1.Data Acquisition & Preparation***

* *Collect and consolidate customer data from relevant sources (e.g., CRM, billing systems, user activity logs).*
* *Clean, preprocess, and transform the data to ensure quality and consistency for analysis and modeling.*

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

* *Analyze customer behavior and characteristics to understand trends and patterns.*
* *Identify correlations and features that may influence churn.*

***3.Feature Engineering***

* *Create new features from raw data to enhance model performance (e.g., average usage, time since last activity).*
* *Encode categorical variables and scale numerical features as needed.*

***4.Model Development***

* *Train multiple machine learning models (e.g., Logistic Regression, Random Forest, XGBoost) to predict customer churn.*
* *Use cross-validation and hyperparameter tuning to optimize model performance.*

***5.Model Evaluation***

* *Evaluate model performance using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.*
* *Handle imbalanced data with appropriate techniques (e.g., SMOTE, class weighting).*

***6.Model Interpretation***

* *Use explainable AI tools (e.g., SHAP, LIME) to understand key drivers of churn.*
* *Provide actionable insights to business stakeholders.*

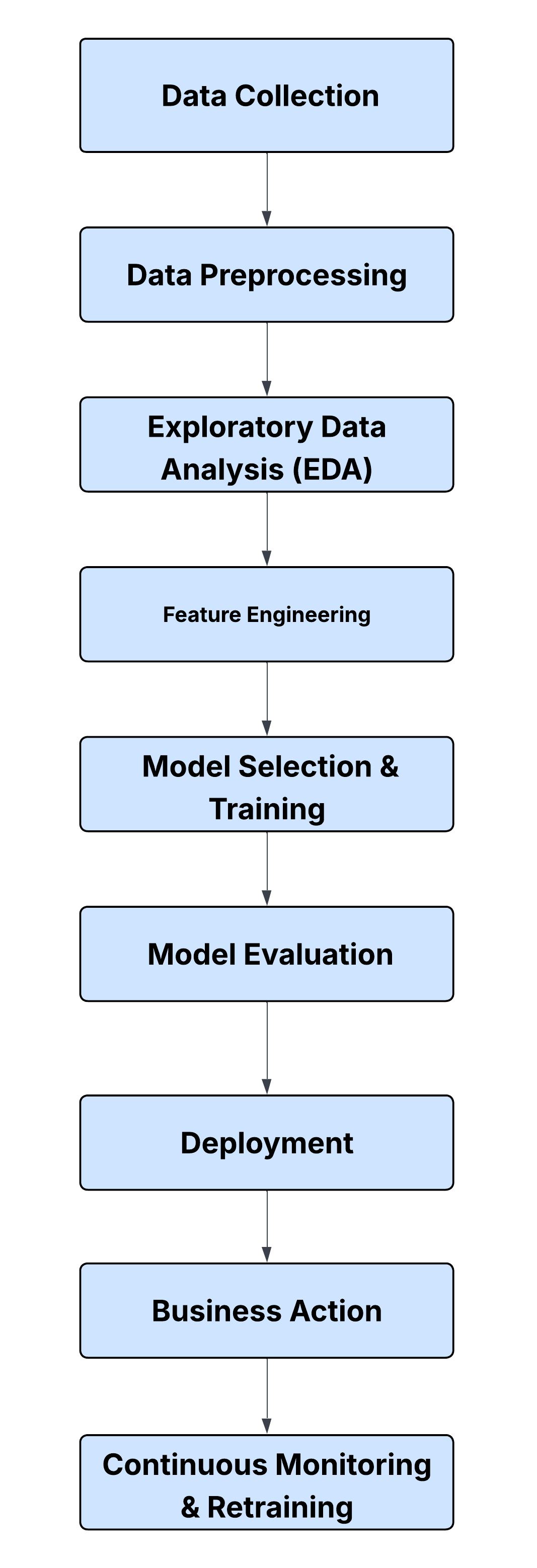
***7.Deployment Strategy***

* *Design a plan to integrate the churn prediction model into existing business systems or dashboards.*
* *Enable real-time or periodic scoring of customers for churn risk.*

***8.Business Impact & Recommendations***

* *Recommend data-driven strategies for customer retention based on model results.*
* *Estimate potential cost savings or revenue improvement from reducing churn.*

**3. Flowchart of the Project Workflow**



**4. Data Description**

***1.Customer Demographics***

* *Basic information such as customer ID, age, gender, and location.*

***2. Account & Subscription Details***

* *Plan type, contract type, monthly charges, and payment method.*

***3. Usage Behavior***

* *Login frequency, session duration, number of transactions, and support ticket activity.*

***4. Engagement Metrics***

* *Email open rates, last login date, and app usage patterns.*

***5. Target Variable***

* *Churn (binary): 1 if the customer has churned, 0 otherwise.*

**5. Data Preprocessing**

1. ***Handle Missing Values*** *– Fill or remove missing data to ensure consistency.*
2. ***Encode Categorical Variables*** *– Convert categories (e.g., gender, plan type) into numeric format.*
3. ***Scale Numerical Features*** *– Normalize features like age and monthly charges for uniformity.*
4. ***Extract Time-Based Features*** *– Derive values like “days since last login” from date fields.*
5. ***Treat Outliers*** *– Detect and manage extreme values to reduce noise.*
6. ***Balance the Dataset*** *– Address class imbalance using techniques like SMOTE or class weighting.*
7. ***Split Data*** *– Divide into training and test sets for model development and evaluation.*

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

***1.Understand Data Structure***

* *Review data types, distributions, and summary statistics of all features.*

***2.Visualize Churn Distribution***

* *Check how many customers have churned vs. stayed to assess class balance.*

***3. Univariate Analysis***

* *Analyze individual features (e.g., age, tenure, monthly charges) using histograms, boxplots, and density plots.*

***4. Bivariate Analysis***

* *Compare features with the churn variable using:*
* *Bar plots for categorical features (e.g., plan type vs. churn rate)*
* *Scatter plots or boxplots for numerical features (e.g., tenure vs. churn)*

***5. Correlation Analysis***

* *Generate a correlation matrix to identify relationships between numerical variables.*

***6. Detect Patterns and Insights***

* *Identify key behaviors or attributes (e.g., short tenure, low usage) that are common among churned customers.*

**7. Feature Engineering**

***1. Create New Predictive Features***

* *Generate meaningful features such as:*
* *Customer tenure (e.g., months since signup)*
* *Average usage per session*
* *Days since last login or transaction*

***2. Transform Date Features***

* *Convert raw dates into numerical formats (e.g., time differences, recency metrics).*

***3. Aggregate Behavioral Metrics***

* *Summarize activity over time (e.g., total logins in the past 3 months, number of support tickets).*

***4. Group Rare Categories***

* *Combine infrequent categorical values into “Other” to reduce noise.*

***5. Interaction Features***

* *Combine multiple features (e.g., monthly charges × tenure) to capture non-linear effects.*

***6. Binarize Features***

* *Convert some variables into binary flags (e.g., "high engagement" = 1 if logins > threshold).*

**8. Model Building**

***1. Select Algorithms***

* *Choose suitable machine learning models, such as:*
* *Logistic Regression – for interpretability*
* *Random Forest / Decision Trees – for handling non-linear data*
* *XGBoost / LightGBM – for high performance*
* *Neural Networks – for complex patterns (if data is large enough)*

***2. Split the Dataset***

* *Divide the data into training and test sets (e.g., 80/20 or 70/30) to validate performance.*

***3. Train the Model***

* *Fit models using the training set with optimized features.*

***4. Hyperparameter Tuning***

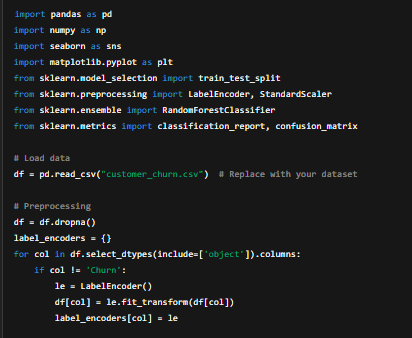
* *Use techniques like Grid Search or Random Search to improve model accuracy and generalization.*

***5. Cross-Validation***

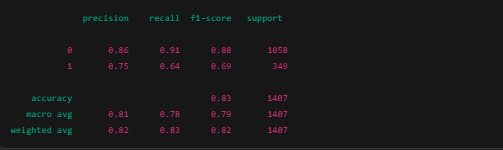
* *Apply k-fold cross-validation to reduce overfitting and assess stability.*

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

**Program:**



**Output:**



**10. Tools and Technologies Used**

**1. Programming Language**

* *Python – for data analysis, preprocessing, modeling, and visualization.*

**2. Libraries & Frameworks**

* *Pandas, NumPy – for data manipulation.*
* *Matplotlib, Seaborn, Plotly – for data visualization.*
* *Scikit-learn – for machine learning models and evaluation.*
* *XGBoost, LightGBM – for advanced gradient boosting algorithms.*
* *SHAP, LIME – for model interpretability.*

**3. Development Environment**

* *Jupyter Notebook / Google Colab – for interactive coding and analysis.*

**4. Version Control**

* *Git – for tracking code changes.*
* *GitHub / GitLab – for collaboration and repository management.*

**5. Deployment (optional)**

* *Flask / FastAPI – for creating a web API for the model.*
* *Streamlit / Dash – for building simple data dashboards.*

**6. Data Storage (optional)**

* *CSV / Excel / SQL Databases – for storing and retrieving data****.***

**11. Team Members and Contributions**

**SANTHOSH KUMAR K – Project Lead / Data Scientist**

* *Defined the problem statement and project scope*
* *Performed data preprocessing, feature engineering, and model development*
* *Interpreted model results and prepared final report*

**SANTHOSH G – Data Analyst**

* *Conducted exploratory data analysis (EDA)*
* *Visualized churn trends and patterns*
* *Assisted in identifying key features affecting churn*

**HARINI S – ML Engineer**

* *Implemented and fine-tuned machine learning models*
* *Handled model evaluation and cross-validation*
* *Supported model deployment using [Flask / Streamlit / other tool]*

**HARI GANESH B – Documentation & Presentation**

* *Compiled project documentation*
* *Designed visuals for model insights and final presentation*
* *Managed version control and collaboration via GitHub*