GithubLink:https://github.com/RamyaV-29/predicting-customer-chural.git

PROJECT TITLE: PREDICTING CUSTOMER CHUR USING MACHINE LEARNING TO UNCOVER HIDDEN PATTERNS

PHASE-2

STUDENT NAME:RAMYA.V

REGISTER NUMBER:623023104042

INSTITUTION: Tagore Institute Of Engineering and Technology

DEPARTMENT:Computer Science And Engineering

DATE OF SUBMISSION:08/05/2025

1.PROBLEM STATEMENT:

Customer churn—when clients stop using a company's product or service—is a critical issue that directly impacts profitability and long-term sustainability. Traditional churn prediction models often fail to detect subtle behavioral signals and complex interactions that lead to customer attrition.

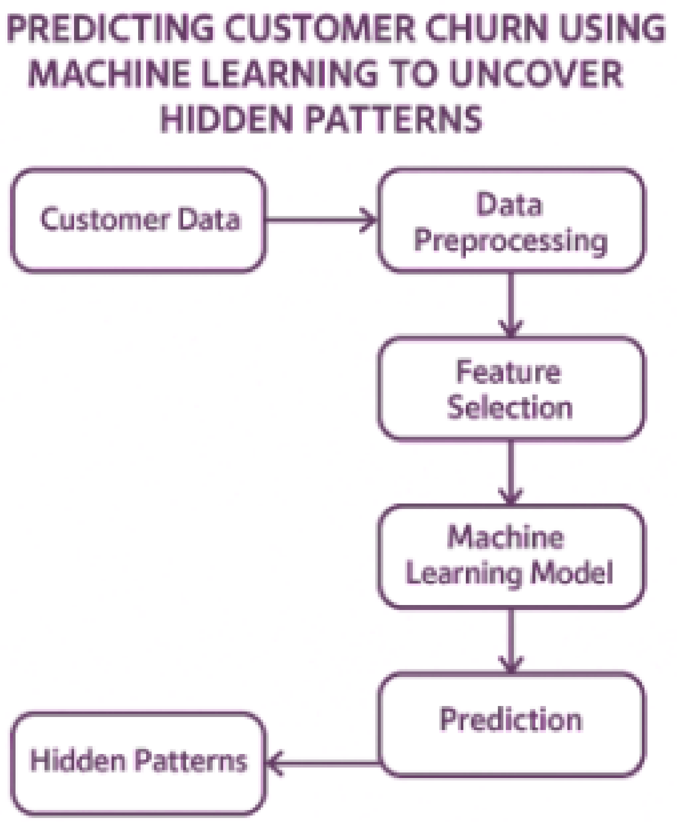
This project aims to leverage machine learning techniques to analyze customer data, uncover hidden patterns, and accurately predict churn.

By identifying high-risk customers early, businesses can develop targeted retention strategies. The ultimate goal is to reduce churn rates, improve customer satisfaction, and enhance business performance through data-driven decision-making.

2. PROJECT OBJECTIVES:

- **To collect and preprocess historical customer data** for effective feature extraction and model training.
- **To apply various machine learning algorithms** (e.g., logistic regression, decision trees, random forests, etc.) to predict customer churn.
- **To uncover hidden patterns and behavioral trends** in customer data that contribute to churn decisions.
- **To evaluate model performance** using appropriate metrics such as accuracy, precision, recall, and F1-score.
- **To develop actionable insights and recommendations** that help businesses implement

3.FLOW CHART FOR THE WORKFLOW:



4. DATA DESCRIPTION:

- **Dataset name:** IBM Telco Customer Churn Dataset.
- **Source:** koogle.
- **Types of data:** Demographic, behavioral, transactional, subscription, feedback.
- **Records and features :** Customer churn prediction using ML.
- **Target Variable:** Churn status (customer retention prediction).
- **Static or Dynamic:** Dynamic, based on customer behavior.
- **Attributes and covered**: Attributes include customer demographics, usage patterns, and interaction history, covering behavior, engagement, and service satisfaction.

- Dataset Link:https://www.kaggle.com/datasets/rjmanoj/credit-card-customer-churn-prediction

5.DATA PREPROCESSING:

- **Data Collection**: Gather customer-related data such as

demographics, transaction history,customer service interactions, and usage patterns to form a comprehensive dataset for analysis.

- **Data Cleaning**: Handle missing values, remove duplicates, and correct inconsistencies to ensure the dataset is reliable and ready for analysis.
- **Feature Engineering**: Identify and create relevant features, such as customer tenure, product usage frequency, or complaint history, that may have an impact on churn prediction.
- **Data Normalization/Scaling**: Standardize or normalize numerical features to bring all variables to a comparable scale, improving the performance of machine learning algorithms.
- **Data Splitting**: Split the data into training and testing sets to evaluate the model's performance on unseen data, ensuring that the churn prediction model generalizes well.

6. EXPLORATORY DATA ANALYSIS (EDA):

- **Univariate Analysis:**

  Distribution of Churn by Customer Demographics: Analyze individual features like age, gender, and income to see how they correlate with churn rates, helping identify demographic segments that are more likely to churn.

  Churn vs. Product Usage Patterns: Explore how factors like frequency of product use or service interactions impact churn, identifying disengaged customers who may be at a higher risk of leaving.

- **Bivariate & Multivariate Analysis**:

  Bivariate analysis examines the relationship between two variables, such as tenure and monthly spend, to identify churn patterns. Multivariate analysis explores multiple features together, uncovering complex interactions that provides a more accurate churn prediction.

- **Key Insights**:

  Identifying Risk Factors: Key insights reveal critical factors like low engagement, frequent customer service issues, or short tenure that are strong indicators of churn risk.

  Segmentation for Retention: By uncovering hidden patterns, machine learning helps segment customers based on churn likelihood, enabling tailored retention strategies for high-risk groups.

7. FEATURE ENGINEERING:

- **Customer Tenure**: Create a feature representing the length of time a customer has been with the company. Shorter tenures may correlate with higher churn rates, while longer tenures can indicate customer loyalty.
- **Frequency of Customer Support Interactions**: Add features based on the number of

support interactions or complaints a customer has made. Frequent interactions, especially negative ones, often signal dissatisfaction and increased churn risk.

- **Product Usage Frequency**: Develop features that measure how often a customer uses the product or service. Decreased usage may indicate disengagement, which is a strong predictor of churn.
- **Spending Patterns**: Create features based on a customer's monthly or yearly spend. Declining spending over time can be a precursor to churn, as customers may scale back or disengage with services.
- **Customer Demographics**: Generate features based on age, location, income level, or other demographic information. These features can help identify specific customer segments that are more or less likely to churn.

8. MODEL BUILDING:

- **Algorithms Used**:

Logistic Regression: A commonly used algorithm for churn prediction, where it models the probability of a customer churning based on various features, providing interpretable results.

Random Forest: A powerful ensemble learning algorithm that uses multiple decision trees to classify churn, capturing complex patterns and interactions in the data for improved accuracy.

- **Model Selection Rationale**:

Accuracy and Interpretability: Logistic regression is often chosen for churn prediction due to its simplicity and interpretability, making it easy to understand which features contribute to churn risk.

Handling Complex Relationships: Random Forest is selected when dealing with non-linear relationships and interactions between features, as it can effectively capture complex patterns without overfitting.

- **Train-Test Split**:

Training Set: Typically, 70-80% of the data is used for training the model, allowing the algorithm to learn patterns and relationships in the data. **Test Set**: The remaining 20-30% of the data is set aside for testing, providing an unbiased evaluation of the model's performance on unseen data and ensuring it generalizes well.

- **Evaluation Metrics**:

Precision and Recall: These metrics evaluate how accurately the model identifies churned customers (precision) and how effectively it detects all actual churn cases (recall), balancing false positives and false negatives.

AUC-ROC: This metric measures the model's ability to distinguish between churned and non-churned customers, with a higher AUC indicating better model performance across different thresholds.

9. VISUALIZATION OF RESULT & MODEL INSIGHTS:

- **Feature Importance**:

Customer Tenure and Usage Patterns: These features often have high importance, as longer-tenured customers or those with consistent usage are less likely to churn, while decreasing usage may signal higher churn risk.

Customer Support Interactions and Spending Behavior: Frequent support interactions and declining spending are critical indicators, with higher interaction rates or lower spend suggesting dissatisfaction and a higher likelihood of churn.

- **Model Comparison**:

Performance Metrics Comparison: Models like Logistic Regression offer high interpretability but may underperform on complex data, whereas ensemble models like Random Forest or XGBoost generally provide higher accuracy and better recall.

Overfitting and Generalization: Simpler models generalize better on small datasets, while complex models may capture intricate patterns but risk overfitting if not properly tuned or validated.

- **Residual Plots**:

A residual plot helps visualize the difference between predicted and actual churn values, revealing patterns or biases in the model's predictions and indicating whether the model fits the data well.

- **User Testing**:

User Testing Insight: User testing involves validating the churn prediction model with real user data or feedback to assess its practical effectiveness, ensuring the model provides actionable insights for retention strategies.

10. TOOLS AND TECHNOLOGY USED:

- **Programming Language**:Python Language.
- **Notebook Environment**: Jupyter Notebook.
- **Key Libraries**:
  - pandas, numpy for data handling
  - matplotlib, seaborn, plotly for visualizations
  - scikit-learn for preprocessing and modeling
  - Gradio for interface deployment

11. TEAM MEMBERS AND CONTRIBUTIONS:

1. **[Saranya] – Project Coordinator**

- Organizes team meetings, sets project timelines, and tracks progress toward milestones.

- Maintains comprehensive documentation and formats the final project report. - Oversees integration of all team outputs into a cohesive final submission and assists with deployment coordination.

2. **[Ramya] – Data Collection & Preprocessing**

   - Acquires the **CICIDS2017** dataset from official sources

   - Handles **data preprocessing**: cleaning, missing value treatment,

   encoding, normalization.

   - Ensures data is ready for analysis and contributes to early-stage exploratory data

   analysis.

3. **[Vinothini] – Exploratory Data Analysis (EDA)**

   - Conducts EDA using tools like Matplotlib, Seaborn, and pandas profiling. - Identifies key patterns, correlations, and anomalies in the dataset.

   - Provides insights to guide feature engineering and model selection.

4. **[Sanguzhali] – Feature Engineering**

   - Engineers new features to improve model accuracy and performance.

   - Implements and fine-tunes anomaly detection models: Isolation Forest,

5. [rajalakshmi] – Model Evaluation, Visualization

   - Evaluates model performance using metrics like ROC-AUC, precision,

   recall, F1-score, and confusion matrix.

   - Creates interactive visualizations using Plotly or Dash.

   - Develops a demo interface using Streamlit or Flask.

   - Prepares final dashboards and supports presentation efforts.