Phase-2

Student Name: Murugesh M

Register Number: 410723104049

Institution: Dhanalakshmi College of Engineering

Department: Computer Science Engineering

Date of Submission: 06-05-2025

Github Repository Link:

https://github.com/Murugesh5756/NM_MURUGESH

PREDICTING CUSTOMER CHURN USING MACHINE LEARNING TO UNCOVER HIDDEN PATTERN

1. Problem Statement

Customer churn is a critical concern for businesses, directly impacting revenue and growth. With the increasing availability of customer interaction and transaction data, it has become feasible to predict churn before it happens. Traditional methods often fail to capture subtle patterns leading to customer attrition.

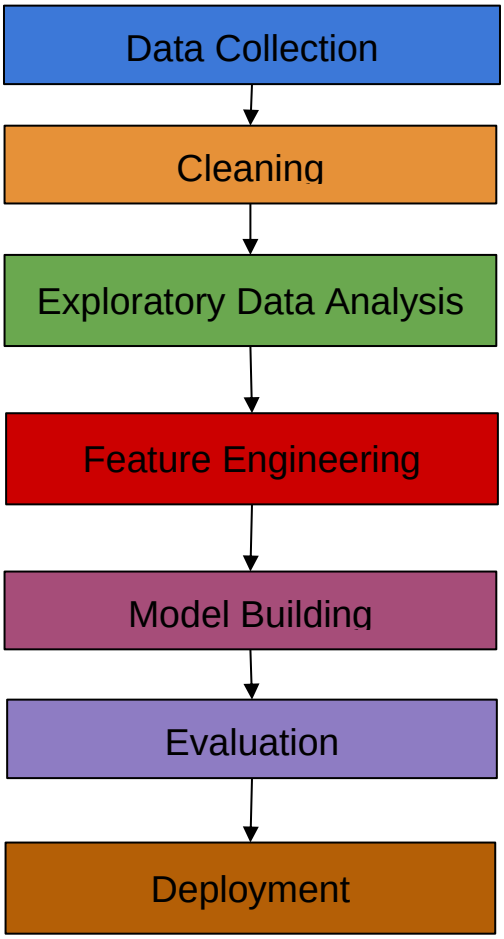
This project aims to develop a machine learning-based solution that accurately. predicts customer churn by leveraging historical data and identifying previously. undetected behavioral patterns. The model will help businesses:

- Detect high-risk customers before they churn.
- Understand the key factors contributing to churn.
- Design targeted retention strategies based on data-driven insights.

2. Project Objectives

- To predict customer churn by building a machine learning model that learns from historical customer data and accurately forecasts whether a customer is likely to leave.
- To analyze behavioral trends and interaction history to detect early warning signs of churn—such as reduced usage, frequent complaints, or overdue payments.
- To identify and rank the key features (such as contract type, usage pattern, customer support interaction, etc.) that most significantly influence a customer's decision to stay or leave.
- To empower businesses with actionable insights so they can design personalized retention campaigns, improve customer experience, and proactively address churn triggers.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name:** Telco Customer Churn Dataset (Kaggle)
- **Type:** Structured tabular data
- **Records and Features:** Varies (70+ records, 20+ features)
- **Target Variable:** Churn (Yes/No)
- **Attributes:** Demographic (gender, senior citizen, partner, dependents), Service (phone, internet, streaming), Billing (monthly charges, tenure), Support (complaints, tech support)

5. Data Preprocessing

- **Missing Values:** Missing values in TotalCharges were imputed with the median after converting empty strings to NaN.
- **Duplicate Records:** Duplicate rows were identified and removed to ensure data uniqueness.
- **Outliers:** Outliers in MonthlyCharges and TotalCharges were handled using statistical techniques and log transformations where needed.
- **Data Types:** All columns were checked; TotalCharges was converted from object to numeric, others were appropriate.
- **Encoding Categorical Variables:** Binary features were label encoded, and multi-class categorical variables were one-hot encoded.
- **Class Imbalance:** SMOTE was applied to balance the churn classes in the training data.
- **Normalization:** Tenure, MonthlyCharges, and TotalCharges were standardized using StandardScaler for consistent scaling.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
numerical_cols = ['tenure', 'MonthlyCharges', 'TotalCharges']
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])
```

6. Exploratory Data Analysis (EDA)

- ***Univariate Analysis:***
  - Histograms, boxplots to study feature distributions.
  - Churn rate by demographics and service usage.
- ***Bivariate/Multivariate Analysis:***
  - Correlation matrix for numeric features.
  - Churn trends over tenure, contract type, and support interaction.
  - Segmentation by churn likelihood.
- ***Key Insights:***
  - Longer tenure often correlates with lower churn.
  - Month-to-month contracts are associated with higher churn.
  - High service complaints increase churn likelihood.

7. Feature Engineering

- Created tenure buckets and interaction frequency scores.
- Derived loyalty and engagement scores.
- Encoded text complaints using TF-IDF (if applicable).
- Used PCA for dimensionality reduction.
- Built interaction terms for usage and support history

```
df['tenure_group'] = pd.cut(df['tenure'],
                            bins=[0, 12, 24, 48, 60, 72],
                            labels=['0–12', '13–24', '25–48', '49–60', '61–72'])
service_features = ['PhoneService', 'MultipleLines', 'InternetService',
                    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
                    'TechSupport', 'StreamingTV', 'StreamingMovies']
df['num_services'] = df[service_features].apply(lambda row: sum(row ==
'Yes'), axis=1)
df['high_monthly_charge'] = df['MonthlyCharges'].apply(lambda x: 1 if x >
70 else 0)
df['avg_monthly_spend'] = df['TotalCharges'] / (df['tenure'] + 1)
df['avg_monthly_spend'].replace([np.inf, -np.inf], np.nan, inplace=True)
df['avg_monthly_spend'].fillna(0, inplace=True)
```

8. Model Building

- **Algorithms Used:**
  - Logistic Regression
  - Decision Tree, Random Forest
  - XGBoost, LightGBM, CatBoost
  - Neural Networks (optional, for large or complex data)
  - Ensemble models for higher accuracy

- **Train-Test Split:**

```
from sklearn.model_selection import train_test_split
X = df.drop('Churn', axis=1)
y = df['Churn']
X_train, X_test, y_train, y_test = train_test_split
( X, y, test_size=0.2, stratify=y, random_state=42)
```

- **Evaluation Metrics:**
  - Accuracy, Precision, Recall, F1-score
  - ROC-AUC
  - Confusion matrix
  - Cross-validation for robustness

9. Visualization of Results & Model Insights

- SHAP and LIME for feature importance and interpretability
- Bar charts: Top churn drivers (e.g., contract type, tenure)
- Churn likelihood dashboards
- ROC curves for classifier comparison
- Residual analysis for misclassification patterns

10. Tools and Technologies Used

- **Programming Language:** Python 3
- **Notebook/IDE:** Google Colab, Jupyter Notebook, VS Code
- **Libraries:**
  - pandas, numpy for data processing
  - matplotlib, seaborn, plotly for EDA
  - scikit-learn, XGBoost, CatBoost, LightGBM for modelling
  - SHAP, LIME for interpretability
  - Streamlit, Flask, Gradio for deployment
- **Optional Tools:** Docker, AWS/GCP for deployment

11. Team Members and Contributions

NAMES	ROLE	RESPONSIBILITY
M Soorya Prakash	Leader	Data Collection and Cleaning

Murugesh M	Member	Data visualization and Interpretation
Logesh R	Member	Exploratory Data Analysis
Magesh V	Member	Model evaluation
Antony Sanjay P	Member	Model Building