Phase-2 Submission Template

Student Name: K.SASI

Register Number: 510123106043

Institution: ADHIPARASAKTHI COLLEGE OF ENGINEERING

Department: BE-ELECTRONICS AND COMMUNICATION ENGINEERING

Date of Submission: 08.05.2025

PREDICTING CUSTOMER CHURN USING MACHINE LEARINGTO UNCOVER HIDDEN PATTERN

1.Problem Statement

Customer retention is a critical factor in the success and profitability of any business. However, identifying which customers are likely to stop using a service—known as customer churn—can be challenging due to the complexity and subtlety of behavioral patterns. This project aims to develop a machine learning-based system to predict customer churn by analyzing historical customer data and uncovering hidden patterns in customer behavior.

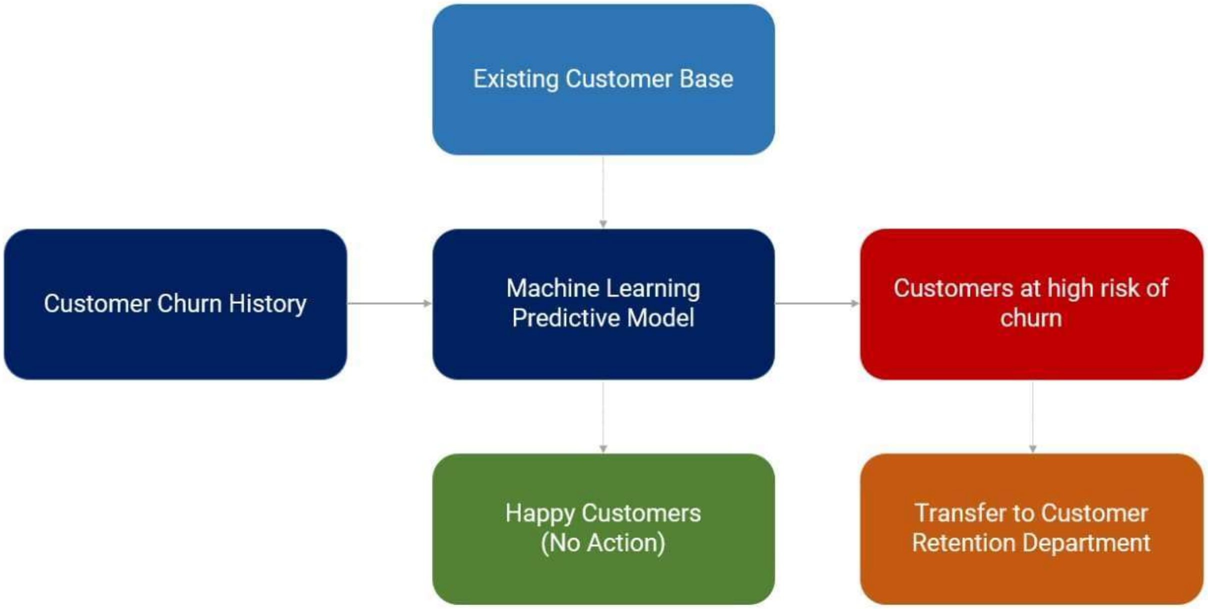
The objective is to leverage supervised learning algorithms to build a predictive model that can accurately classify customers as likely to stay or churn. By uncovering the underlying factors contributing to churn, businesses can proactively implement strategies to improve customer retention, reduce loss, and enhance customer satisfaction.

The model will be trained using customer attributes such as demographics, usage patterns, account information, service intractions, and other relevant features. The project will involve data preprocessing, feature engineering, model selection, evaluation, and interpretation of results.

2.Project Objectives

The objective of this project is to develop an accurate and efficient machine learning model to predict customer churn by analyzing historical customer data. The goal is to identify key factors that influence churn behavior, uncover hidden patterns in customer interactions, and provide actionable insights that help businesses proactively reduce churn and improve customer retention strategies.

3.Flowchart of the Project Workflow


Existing Customer Base
Customer Churn History
Machine Learning Predictive Model
Customers at high risk of churn
Happy Customers (No Action)
Transfer to Customer Retention Department

4.Data Description

Data Description

The dataset used for predicting customer churn typically includes historical customer data collected from a company’s customer base. This data helps identify patterns and behaviors that lead to customer churn. The dataset may include the following features:

1. Customer Demographics

CustomerID: Unique identifier for each customer.

Gender: Male or Female.

Age: Age of the customer.

Geography: Country or region (e.g., France, Spain, Germany).

2. Account Information

Tenure: Number of years the customer has been with the company.

Balance: Current account balance.

ProductsNumber: Number of products or services used by the customer.

HasCrCard: Whether the customer has a credit card (1 = Yes, 0 = No).

IsActiveMember: Indicates if the customer is actively using the service (1 = Active, 0 = Inactive).

3. Financial Metrics

EstimatedSalary: Estimated annual income of the customer.

CreditScore: Customer’s credit score.

4. Churn Label

This structured data allows machine learning algorithms to learn patterns that differentiate churned customers from retained ones, enabling the prediction of future churn with significant accuracy.

5.Data Preprocessing

1. Data Collection

Source data: CRM systems, transactional logs, user behavior, support ticket data, etc.

Common features: Customer ID, demographics, usage metrics, subscription type, payment history, service complaints, tenure, etc.

2. Data Cleaning

Handle missing values:

Drop rows/columns with too many missing values.

Use mean/median/mode for numerical fields and most frequent value for categorical ones.

Remove duplicates

Fix inconsistencies: Standardize categorical labels (e.g., "Yes", "YES", "yes" => "Yes")

3. Feature Engineering

Create new features:

Monthly Charges * Tenure → Total Revenue

Flag for Senior Citizen or Recent Activity

Binarize target variable:

Convert churn column into 0 (Not Churned) and 1 (Churned)

Convert dates into meaningful features (e.g., tenure, days since last login)

4. Encoding Categorical Variables

Label Encoding for binary categories (e.g., Yes/No)

One-Hot Encoding for nominal categories (e.g., internet type, payment method)

5. Feature Scaling

Apply Standard Scaler or Min Max Scaler for numerical features to normalize values

6. Outlier Detection and Treatment

Use IQR or Z-score methods to detect outliers in features like Monthly Charges, Tenure

Optionally remove or cap extreme outliers

7. Train-Test Split

Split the data: e.g., 70% training and 30% testing using train_test_split

8. Optional: Dimensionality Reduction

Use PCA or feature selection techniques if you have a large number of features

6.Exploratory Data Analysis (EDA)

1. Understand the Dataset

Shape of the dataset: Rows and columns

Data types: df.dtypes

Null values: df.isnull().sum()

Unique values per column: df.nunique()

2. Target Variable Analysis

Churn distribution:

```
sns.countplot(x='Churn', data=df)
plt.title("Churn Count")
```

Churn rate:

```
churn_rate = df['Churn'].value_counts(normalize=True) * 100
```

3. Univariate Analysis

Numerical Features:

Histograms: df['MonthlyCharges'].hist()

Boxplots to detect outliers: sns.boxplot(x=df['TotalCharges'])

Categorical Features:

Bar plots:

```
sns.countplot(x='Contract', data=df)
```

4. Bivariate Analysis

Churn vs Numerical Variables:

```
sns.boxplot(x='Churn', y='MonthlyCharges', data=df)

sns.boxplot(x='Churn', y='Tenure', data=df)
```

Churn vs Categorical Variables:

```
pd.crosstab(df['Churn'], df['InternetService']).plot(kind='bar', stacked=True)
```

5. Correlation Matrix

To identify relationships between numerical features:

```
corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap='coolwarm')
```

6. Insights to Look For

Do customers with longer tenure churn less?

Does contract type affect churn rate?

Are certain services (e.g., tech support, streaming TV) correlated with higher churn?

7. Optional: Grouped Summary Statistics

```
df.groupby('Churn')[['MonthlyCharges', 'TotalCharges', 'Tenure']].mean()
```

7.Feature Engineering

1. Creating New Features

These are additional features derived from existing ones to help uncover patterns:

Total Charges: Multiply a customer’s monthly charges by their tenure to estimate total revenue from them.

Average Monthly Spend: Divide total charges by the number of months the customer has been with the company.

Tenure Group: Categorize customers based on how long they've been subscribed (e.g., 0–1 year, 1–2 years, etc.).

Service Usage Flags: Create binary indicators (Yes/No converted to 1/0) for services like tech support, streaming, or multiple lines.

2. Converting Categorical Data

To make machine learning algorithms work properly:

Binary categories (e.g., Gender, Yes/No fields) are turned into 1s and 0s.

Multi-category fields (like Contract Type, Internet Service, or Payment Method) are converted into multiple columns—each representing one category (One-Hot Encoding).

3. Transforming Dates (if applicable)

If the dataset includes dates (like signup or last activity), you can:

Calculate how long the customer has been inactive.

Determine how long they've been a customer (customer age).

4. Data Normalization or Transformation

To deal with skewed or unbalanced data:
Apply scaling to numerical values like monthly or total charges.Transform values (e.g., using log scale) if there are large differences in ranges.

5. Feature Selection

Not all features help the model. You can:

Remove irrelevant features (like customer ID).

Use statistical methods or machine learning models to select the most important features (e.g., tenure, contract type, support usage).

8.Visualization of Results & Model Insights

1. Define the Problem

Type: Binary classification (Churn = Yes/No)

Goal: Accurately predict whether a customer will churn

2. Prepare the Dataset

Use preprocessed and feature-engineered data
Split into:

Training set (e.g., 70–80% of data)

Test set (e.g., 20–30% of data)

3. Choose Machine Learning Models

Common algorithms for churn prediction include:

Logistic Regression – Good for interpretability

Decision Tree – Easy to visualize decisions

Random Forest – Ensemble method, handles overfitting well

Gradient Boosting (XGBoost, LightGBM) – High accuracy, powerful for tabular data

Support Vector Machine (SVM) – Good for small and medium datasets

K-Nearest Neighbors (KNN) – Simple, good baseline

Neural Networks – Effective for complex patterns, but less interpretable

4. Train the Model

Feed the training data to the model

Adjust hyperparameters (e.g., depth of tree, learning rate) using techniques like:

Grid Search

Cross-validation

5. Evaluate the Model

Use the test set and metrics such as:

Accuracy – Overall correctness

Precision – How many predicted churns were correct
Recall – How many actual churns were correctly predicted

F1-score – Balance between precision and recall

AUC-ROC Curve – Measures classification performance

6. Select the Best Model

Compare metrics across models

Choose the one with best performance and generalization ability

7. Interpret the Model

Use tools like feature importance charts, SHAP values, or confusion matrix to understand:

What drives churn

Which customers are at risk

8. Make Predictions

Use the chosen model to predict churn for new/unseen customer data

Deploy the model in a system to alert or retain at-risk customers

9.Tools and Technologies Used

1. Churn Distribution

Purpose: Understand class imbalance Visualization:

Bar chart showing number of churned vs non-churned customer

2. Correlation Heatmap

Purpose: Show relationships between numerical features Visualization:

Heatmap to identify strongly correlated features (e.g., tenure, monthly charges)
3. Feature Importance

Purpose: Understand which features impact churn the most Visualization:

Bar plot of top features ranked by importance (from Random Forest, XGBoost, etc.)
Example features: Contract Type, Tech Support, Tenure, Monthly Charges

4. Churn vs Key Features

Purpose: Show how key variables affect churn Visualization
Examples:

Boxplots: Monthly Charges vs Churn

Bar plots: Contract Type vs Churn Rate

Stacked bar charts: Internet Service type vs Churn

5. Model Evaluation Metrics

Purpose: Show model performance Visualizations:

Confusion Matrix: To see true positives, false positives, etc.

ROC Curve: To show model's trade-off between sensitivity and specificity

Precision-Recall Curve: Useful when dealing with class imbalance

6. SHAP (SHapley Additive exPlanations) Summary Plot

Purpose: Explain how features affect each individual prediction Visualization:

Summary plot showing impact, magnitude, and direction of features on churn prediction

7. Customer Segmentation

Purpose: Identify patterns among churn-prone groups
Visualization:

Cluster plots or segment-based bar charts by age, service usage, etc.
8. Churn Risk ScorePurpose: Visualize risk for individual customersVisualization:
A risk heatmap or gauge chart per customer based on model prediction

10.Team Members and Contributions

1. Data Collection & Storage

Excel / CSV / SQL databases – For collecting and storing raw customer data

Google Sheets – For small-scale data entry and sharing

MySQL / PostgreSQL – For structured data storage and queries
2. Data Analysis & Preprocessing

Python – Most widely used language for data science

Pandas – For data manipulation and cleaning

NumPy – For numerical operations

Jupyter Notebook / Google Colab – For interactive development and analysis

3. Data Visualization

Matplotlib – Basic plotting library

Seaborn – High-level visualization based on Matplotlib

Plotly – Interactive and web-friendly visualizations

Tableau / Power BI (optional) – For dashboards and business reports

4. Machine Learning & Modeling

Scikit-learn – For classification models like Logistic Regression, Decision Trees, Random Forest

XGBoost / LightGBM – Advanced gradient boosting frameworks for high accuracy

Keras / TensorFlow / PyTorch (optional) – For deep learning models

MLflow – To track experiments and model versions

5. Model Evaluation

Scikit-learn metrics – Accuracy, Precision, Recall, F1-Score, AUC-ROC

SHAP / LIME – For model interpretability and explainability

Confusion Matrix Visualization – To understand prediction outcomes

6. Deployment (Optional / Advanced)

Flask / FastAPI – For building an API to serve your model

Streamlit / Dash – For creating a simple UI for predictions