Phase-2 Submission Template

Student Name: D.RAGUL

Register Number: 23uit039

Institution: AVS COLLEGE OF TECHNOLOGY

Department: INFORMATION TECHNOLOGY

Date of Submission: 03/05/2025

GithubRepositoryLink:

https://github.com/Raguldm012/Predicting-customer-churn-using-machine-learning-to-uncover-hidden-pattern

---

1. Problem Statement

- *In highly competitive markets, retaining customers is crucial for business success. However, many organizations struggle to identify which customers are likely to leave (churn) before it happens. This project aims to develop a machine learning model to predict customer churn by analyzing historical customer data and uncovering hidden patterns in behavior, usage, and engagement. By accurately identifying at-risk customers, businesses can take proactive measures to improve retention and optimize their customer relationship strategies.*

2. Project Objectives

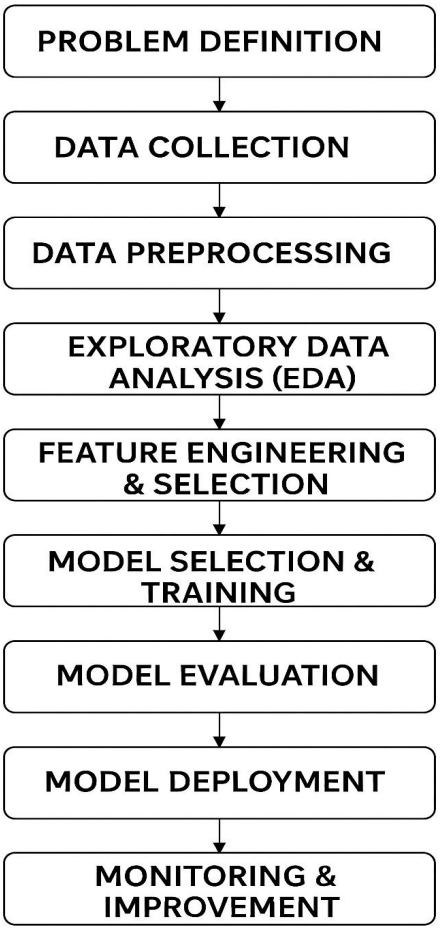
- *Collect and preprocess customer data from various sources to ensure quality and consistency for model training.*

- *Build and evaluate multiple machine learning models (e.g., logistic regression, decision trees, random forest, XGBoost) to predict the likelihood of churn.*

- *. Select the best-performing model based on metrics such as accuracy, precision, recall, and AUC-ROC.*

- *Deploy the predictive model to provide real-time churn risk scores for individual customers.*

3.Flowchart of the project workflow



4. Data Description

The dataset used for this project contains information about customers of a subscription-based service, aiming to identify patterns associated with customer churn (i.e., customers who stop using the service). Each row in the dataset represents a single customer and includes a variety of features related to demographics, account activity, and service usage.

Key Features:

- ***Gender:*** *Customer's gender (e.g., Male, Female).*
- ***Age:*** *Customer's age.*
- ***Tenure:*** *Number of months the customer has been with the company.*
- ***SubscriptionType:*** *Type of subscription plan (e.g., Basic, Premium).*
- ***MonthlyCharges:*** *The amount charged to the customer each month.*
- ***TotalCharges:*** *The total amount charged to the customer over the entire tenure.*
- ***ServiceUsageMetrics:*** *Aggregated usage statistics (e.g., hours watched, data used).*
- ***SupportCalls:*** *Number of support calls made by the customer.*
- ***PaymentMethod:*** *Method used for payment (e.g., Credit Card, Direct Debit).*
- ***ContractType:*** *Type of contract (e.g., Month-to-Month, One Year, Two Year).*

5. Data Preprocessing.

To predict customer churn using machine learning, data preprocessing is a critical first step. It ensures your model can learn effectively from the data by

cleaning and transforming it into a usable format. Here's a structured approach to data preprocessing for churn prediction**:**

1. Data Collection

- Gather historical customer data, including:Demographics (age, gender, location)
- Account details (tenure, subscription plan, billing method)
- Usage patterns (minutes used, data consumption)
- Customer service interactions (complaints, call history)

Churn label (Yes/No)

2. Data Cleaning

- Handle missing values
- Drop or impute with mean/median/mode
- Remove duplicates
- Ensure no duplicated records.

3. Feature Engineering

- Encode categorical variables:
- Use one-hot encoding or label encoding.

4. Feature Scaling

- Standardization or normalization:

- Especially important for models like Logistic Regression, SVM, KNN.

5. Handle Class Imbalance

- Churn datasets are often imbalanced (few customers churn vs. many stay).

Use techniques like:

- SMOTE (Synthetic Minority Over-sampling Technique)
- Random under-sampling/over-sampling

Class weights in algorithms

6. Split the Data

- Split into training and test sets (commonly 70/30 or 80/20).
- Optionally, use a validation set or cross-validation for better tuning

6. Exploratory Data Analysis (EDA)

- ***Univariate Analysis:***

Great! Here's a complete overview of Exploratory Data Analysis (EDA) for predicting customer churn using machine learning, with a focus on uncovering hidden patterns:

Exploratory Data Analysis (EDA) for Customer Churn Prediction

*1. **Understanding the Dataset***

```
import pandas as pd

df = pd.read_csv('your_dataset.csv')  # Replace with actual filename
```

```
print(df.shape)

print(df.columns)

print(df.info())

print(df.describe()
```

Goal: *Understand size, datatypes, and summary statistics.*

2.Univariate Analysis (Single Feature at a Time)

```
# Categorical Variables

categorical_cols = ['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService', 'Contract']

for col in categorical_cols:

sns.countplot(data=df, x=col, hue='Churn')

plt.title(f'{col} vs Churn')

plt.xticks(rotation=45)

plt.show()
```

3.Bivariate Analysis (Two Features Together)

```
Numerical Features vs Churn

for col in numeric_cols:

 plt.figure(figsize=(8, 4))

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

    plt.title(f'{col} by Churn')

    plt.show()
```

4.Correlation Matrix (Numeric Features Only)

```
corr_matrix = df.corr(numeric_only=True)
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

7. Feature Engineering

Excellent! Feature engineering is a crucial step in predicting customer churn because it helps the model uncover hidden patterns by creating meaningful inputs. Here's how you can approach it

Feature Engineering for Customer Churn Prediction

1. Convert Categorical to Numerical

a. Label Encoding (for binary variables):

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['gender'] = le.fit_transform(df['gender'])
df['Partner'] = le.fit_transform(df['Partner'])
df['Dependents'] = le.fit_transform(df['Dependents'])
df['PaperlessBilling'] = le.fit_transform(df['PaperlessBilling'])
df['Churn'] = le.fit_transform(df['Churn'])  # Target
```

b. One-Hot Encoding (for non-binary categorical variables):

df = pd.get_dummies(df, columns=['Contract', 'InternetService', 'PaymentMethod'], drop_first=True)

2. Create New Features

Tenure Grouping:

df['TenureGroup'] = pd.cut(df['tenure'], bins=[0, 12, 24, 48, 60, 72],

labels=['0-12', '12-24', '24-48', '48-60', '60-72'])

df = pd.get_dummies(df, columns=['TenureGroup'])

Average Charges per Tenure:

df['AvgChargesPerMonth'] = df['TotalCharges'] / (df['tenure'] + 1)

0, 'No phone service': 0})

df['TotalServices'] = df[services].sum(axis=1)

8. Model Building

Excellent! Now let's move to model building for predicting customer churn using machine learning.

Model Building for Customer Churn Prediction

1. Split the Data

Separate the features and target, then split into training and test sets:

```
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,
random_state=42, stratify=y)
```

2. Choose Algorithms

- Start with basic models:
- Logistic Regression
- Random Forest
- XGBoost (or Gradient Boosting)
- K-Nearest Neighbors
- Support Vector Machine (SVM)

Let’s try Logistic Regression and Random Forest as examples.

3. Train and Evaluate Models

1.Logistic Regression

```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

log_model = LogisticRegression(max_iter=1000)
```

```
log_model.fit(X_train, y_train)

y_pred_log = log_model.predict(X_test)

print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred_log))

print(confusion_matrix(y_test, y_pred_log))

print(classification_report(y_test, y_pred_log))
```

2.Random Forest

```
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

rf_model.fit(X_train, y_train)

y_pred_rf = rf_model.predict(X_test)

print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))

print(confusion_matrix(y_test, y_pred_rf))

print(classification_report(y_test, y_pred_rf))
```

9. Visualization of Results & Model Insights

```
from sklearn.metrics

import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred_rf)  # Replace with y_pred_log for
logistic regression
```

```
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["No Churn", "Churn"])

disp.plot(cmap='Blues')

plt.title("Confusion Matrix - Random Forest")

plt.show()
```

10. Tools and Technologies Used

[To successfully predict customer churn using machine learning, a combination of tools and technologies is used across the stages of data collection, processing, modeling, evaluation, and deployment. Here's a breakdown:

1.Programming Language

- ***Python:*** *Most popular choice due to its rich ecosystem for data science and machine learning.*
- ***R:*** *Alternative for statistical modeling and visualization.*

2. Data Processing & Analysis

- ***Pandas:*** *For data manipulation and analysis.*
- ***NumPy:*** *For numerical operations.*
- ***OpenPyXL / xlrd:*** *For reading Excel files if needed.*

3. ***Data Visualization***

- ***Matplotlib:*** *Basic plotting.*
- ***Seaborn:*** *Statistical visualizations (heatmaps, boxplots, distributions).*
- ***Plotly:*** *Interactive charts and dashboards (optional).*

4. Machine Learning Libraries

- ***Scikit-learn:*** *Core ML library (classification models, preprocessing, metrics, etc.).*

- ***XGBoost / LightGBM / CatBoost:*** *Advanced gradient boosting algorithms for high performance.*
- ***Imbalanced-learn:*** *Tools like SMOTE to handle imbalanced datasets.*

5. Model Evaluation & Tuning

- ***Scikit-learn*** *(metrics, cross-validation)*
- ***GridSearchCV / RandomizedSearchCV:*** *Hyperparameter tuning.*
- ***SHAP:*** *Interpretability and feature importance.*

6. Deployment (optional)

- ***Flask / FastAPI:*** *To deploy the model as an API.*
- ***Streamlit / Dash:*** *For creating interactive web apps.*
- ***Docker:*** *For containerizing the application.*
- ***AWS / Azure / GCP:*** *Cloud platforms to host the model.*

Workflow Summary

1. Data Collection: *CSV/Excel/SQL → Pandas*

2. EDA & Preprocessing: *Pandas, Seaborn, Scikit-learn*

3. Feature Engineering: *Pandas, Scikit-learn*

4. Model Training: *Scikit-learn, XGBoost*

5. Evaluation & Visualization: *Matplotlib, Seaborn, Scikit-learn*

6. Deployment (optional): *Flask, Docker, Cloud services*

11. Team Members and Contributions

TEAM LEADER: RAGUL D

RESEACHAR:AKASH E

DEVELOPER: HARISH S

DESIGNER:SUBASH C

TESTER:FRANKLIN M