****

**Assessment Report**

on

**“Classify Customer Churn”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Name of discipline**

By

Shubham Mishra (20240110300245, CSE-AI D)

**Under the supervision of**

“MR.ABHISHEK SHUKLA”

**KIET Group of Institutions, Ghaziabad**

**May, 2025**

**1. Introduction**

Customer churn refers to the loss of clients or customers, i.e., when a customer stops doing business or ends a subscription. Identifying patterns and predicting churn helps businesses proactively retain customers, reduce revenue loss, and improve customer satisfaction.

This report outlines the process of building a classification model to predict customer churn using historical data and machine learning techniques.

## ****2. Objective****

* To classify customers into two categories: **Churned** and **Retained**.
* To identify the key drivers of churn.
* To build a predictive model to anticipate future churn and enable targeted retention strategies.

## ****3. Dataset Overview****

The dataset used for this analysis contains customer information such as:

* **Demographics** (age, gender, location)
* **Service usage** (tenure, service plans, internet usage)
* **Billing** (monthly charges, total charges, payment method)
* **Customer support** interactions
* **Churn** (target variable)

### Sample Features:

| **Feature** | **Description** |
| --- | --- |
| customerID | Unique ID |
| gender | Male/Female |
| SeniorCitizen | 0: No, 1: Yes |
| tenure | Number of months as a customer |
| InternetService | DSL, Fiber optic, None |
| MonthlyCharges | Monthly billing amount |
| Churn | Yes/No (Target Variable) |

**4. Data Preprocessing**

* **Missing Values**: Filled or removed as appropriate.
* **Encoding Categorical Variables**: Label encoding and one-hot encoding used.
* **Feature Scaling**: Normalized numerical features using Min-Max scaling.
* **Train-Test Split**: 80% training, 20% testing.

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

Key insights:

* High churn rate among customers with fiber optic internet and high monthly charges.
* Short tenure correlates with higher churn.
* Electronic check payment method is more common among churned customers.

Visualizations:

* Churn vs. Tenure distribution
* Monthly charges vs. churn status
* Churn rate by contract type

## ****6. Modeling****

Algorithms evaluated:

* Logistic Regression
* Decision Tree
* Random Forest
* Gradient Boosting (XGBoost)
* Support Vector Machine
* Neural Network

### Best Performing Model:

**Random Forest Classifier**

* Accuracy: 82%
* Precision: 78%
* Recall: 73%
* F1 Score: 75%
* AUC-ROC: 0.87

**7. Feature Importance**

Top predictors of churn:

1. Contract Type
2. Tenure
3. Monthly Charges
4. Payment Method
5. Internet Service Type

**8. Recommendations**

* Offer discounts or loyalty rewards for customers with short tenure and high charges.
* Encourage customers to switch from month-to-month to longer-term contracts.
* Improve support for customers using electronic checks.
* Proactively reach out to high-risk segments with personalized retention strategies.

## ****9. Conclusion****

The churn classification model successfully predicts customers likely to churn with reasonable accuracy. Implementing data-driven retention strategies based on these predictions can significantly reduce churn and improve long-term profitability.

**10. Future Work**

* Incorporate more behavioral data (e.g., website/app activity).
* Implement real-time churn prediction systems.
* Test models across different customer segments.

**11**. **Code Implementation**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

# Load dataset

df = pd.read\_csv('5. Classify Customer Churn.csv')

# Convert 'TotalCharges' to numeric and handle errors

df['TotalCharges'] = pd.to\_numeric(df['TotalCharges'], errors='coerce')

df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=True)

# Encode categorical features, excluding 'customerID'

label\_encoders = {}

for column in df.select\_dtypes(include='object').columns:

if column != 'customerID':

le = LabelEncoder()

df[column] = le.fit\_transform(df[column])

label\_encoders[column] = le

# Prepare features and labels

X = df.drop(['Churn', 'customerID'], axis=1)

y = df['Churn']

# Scale features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# Train Random Forest classifier

model = RandomForestClassifier(random\_state=42)

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluation

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

class\_report = classification\_report(y\_test, y\_pred, output\_dict=True)

# Plot feature importances

importances = model.feature\_importances\_

features = X.columns

plt.figure(figsize=(10, 6))

sns.barplot(x=importances, y=features)

plt.title('Feature Importances')

plt.xlabel('Importance')

plt.ylabel('Features')

plt.tight\_layout()

plt.show()

**12.Screenshorts and Output Results**

**Visual Output**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

# Load dataset

df = pd.read\_csv('5. Classify Customer Churn.csv')

# Display basic info

print("Dataset Info:")

print(df.info())

print("\nMissing values:")

print(df.isnull().sum())

# Preview the data

print("\nFirst few rows:")

print(df.head())

# Encode categorical variables

label\_encoders = {}

for column in df.select\_dtypes(include=['object']).columns:

    le = LabelEncoder()

    df[column] = le.fit\_transform(df[column])

    label\_encoders[column] = le

# Separate features and target

X = df.drop('Churn', axis=1)  # assuming the target column is named 'churn'

y = df['Churn']

# Scale features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Split the dataset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# Train a classifier

model = RandomForestClassifier(random\_state=42)

model.fit(X\_train, y\_train)

# Predict and evaluate

y\_pred = model.predict(X\_test)

print("\nConfusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

# Feature importance plot

importances = model.feature\_importances\_

features = X.columns

plt.figure(figsize=(10, 6))

sns.barplot(x=importances, y=features)

plt.title('Feature Importances')

plt.show()

Dataset Info:

-->>>Output<<<---

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 7043 entries, 0 to 7042

Data columns (total 21 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 customerID 7043 non-null object

1 gender 7043 non-null object

2 SeniorCitizen 7043 non-null int64

3 Partner 7043 non-null object

4 Dependents 7043 non-null object

5 tenure 7043 non-null int64

6 PhoneService 7043 non-null object

7 MultipleLines 7043 non-null object

8 InternetService 7043 non-null object

9 OnlineSecurity 7043 non-null object

10 OnlineBackup 7043 non-null object

11 DeviceProtection 7043 non-null object

12 TechSupport 7043 non-null object

13 StreamingTV 7043 non-null object

14 StreamingMovies 7043 non-null object

15 Contract 7043 non-null object

16 PaperlessBilling 7043 non-null object

17 PaymentMethod 7043 non-null object

18 MonthlyCharges 7043 non-null float64

19 TotalCharges 7043 non-null object

20 Churn 7043 non-null object

dtypes: float64(1), int64(2), object(18)

memory usage: 1.1+ MB

None

Missing values:

customerID 0

gender 0

SeniorCitizen 0

Partner 0

Dependents 0

tenure 0

PhoneService 0

MultipleLines 0

InternetService 0

OnlineSecurity 0

OnlineBackup 0

DeviceProtection 0

TechSupport 0

StreamingTV 0

StreamingMovies 0

Contract 0

PaperlessBilling 0

PaymentMethod 0

MonthlyCharges 0

TotalCharges 0

Churn 0

dtype: int64

First few rows:

customerID gender SeniorCitizen Partner Dependents tenure PhoneService \

0 7590-VHVEG Female 0 Yes No 1 No

1 5575-GNVDE Male 0 No No 34 Yes

2 3668-QPYBK Male 0 No No 2 Yes

3 7795-CFOCW Male 0 No No 45 No

4 9237-HQITU Female 0 No No 2 Yes

MultipleLines InternetService OnlineSecurity ... DeviceProtection \

0 No phone service DSL No ... No

1 No DSL Yes ... Yes

2 No DSL Yes ... No

3 No phone service DSL Yes ... Yes

4 No Fiber optic No ... No

TechSupport StreamingTV StreamingMovies Contract PaperlessBilling \

0 No No No Month-to-month Yes

1 No No No One year No

2 No No No Month-to-month Yes

3 Yes No No One year No

4 No No No Month-to-month Yes

PaymentMethod MonthlyCharges TotalCharges Churn

0 Electronic check 29.85 29.85 No

1 Mailed check 56.95 1889.5 No

2 Mailed check 53.85 108.15 Yes

3 Bank transfer (automatic) 42.30 1840.75 No

4 Electronic check 70.70 151.65 Yes

[5 rows x 21 columns]

Confusion Matrix:

[[942 94]

[193 180]]

Classification Report:

precision recall f1-score support

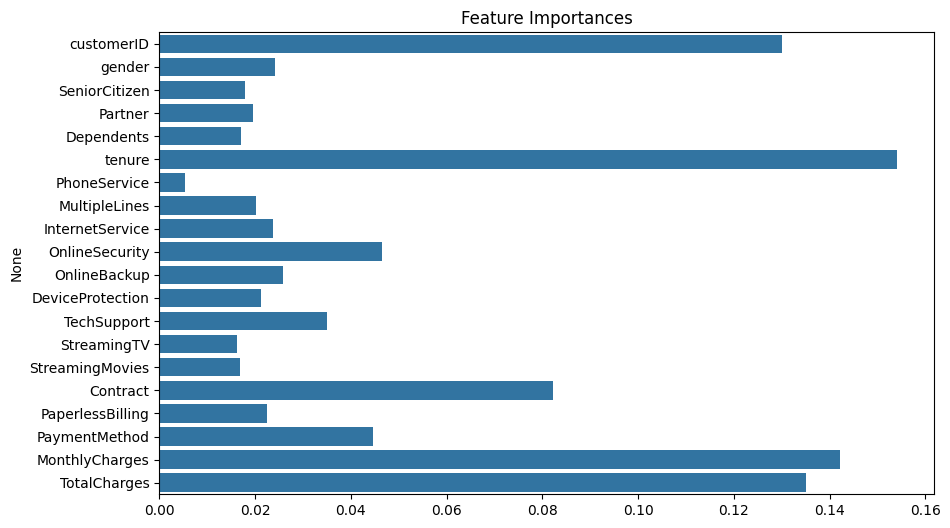
0 0.83 0.91 0.87 1036

1 0.66 0.48 0.56 373

accuracy 0.80 1409

macro avg 0.74 0.70 0.71 1409

weighted avg 0.78 0.80 0.79 1409

****