Assessment Report

on

"Customer Churn Classification Report"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Name of discipline

By

Hemang Singh (202401100400098, CSE(AI&ML)-B)

Under the supervision of

"Name of mentor/guide"

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction:

\In today's highly competitive telecom industry, customer retention is a major priority. One of the key challenges companies face is understanding which customers are likely to discontinue their services—a phenomenon known as **customer churn**. If companies can predict which users are at risk of churning, they can take timely action such as offering incentives, improving service quality, or providing personalized support to retain those customers.

This project focuses on building a **classification model** to predict customer churn using historical telecom customer data. The dataset contains a variety of features related to customer behavior and demographics, such as contract type, payment method, internet usage, tenure, and service subscriptions. By analyzing these features, the goal is to train a machine learning model that accurately identifies customers who are likely to leave the company.

We chose a **Random Forest Classifier** for this task due to its ability to handle both numerical and categorical data, its robustness to overfitting, and its generally strong performance on classification problems.

Once the model is trained, we assess its performance using several standard **evaluation metrics**, including:

- **Accuracy** – the overall correctness of the model.
- **Precision** – the proportion of correctly identified positive churns among all predicted churns.
- **Recall** – the proportion of actual churns that were correctly identified.

To visualize the model's performance, we also generate a **confusion matrix heatmap**, which helps in understanding the distribution of true positives, false positives, true negatives, and false negatives.

The insights from this project can support data-driven strategies in customer relationship management (CRM) and improve business decision-making in the telecom sector.

Methodology:

Steps Followed:

1. **Data Upload & Loading:**
   - Loaded the "5. Classify Customer Churn.csv" dataset.
2. **Data Preprocessing:**
   - Dropped unnecessary columns like customerID.
   - Converted TotalCharges to numeric and handled missing values.
   - Encoded categorical variables using LabelEncoder.
   - Normalized the features using StandardScaler.
3. **Train-Test Split:**
   - Data was split into 80% training and 20% testing sets using train_test_split.
4. **Model Training:**
   - Trained a **Random Forest Classifier** using the training set.
5. **Model Evaluation:**
   - Predicted churn using the test set.
   - Computed evaluation metrics: **accuracy**, **precision**, and **recall**.
   - Generated and visualized a **confusion matrix heatmap**.

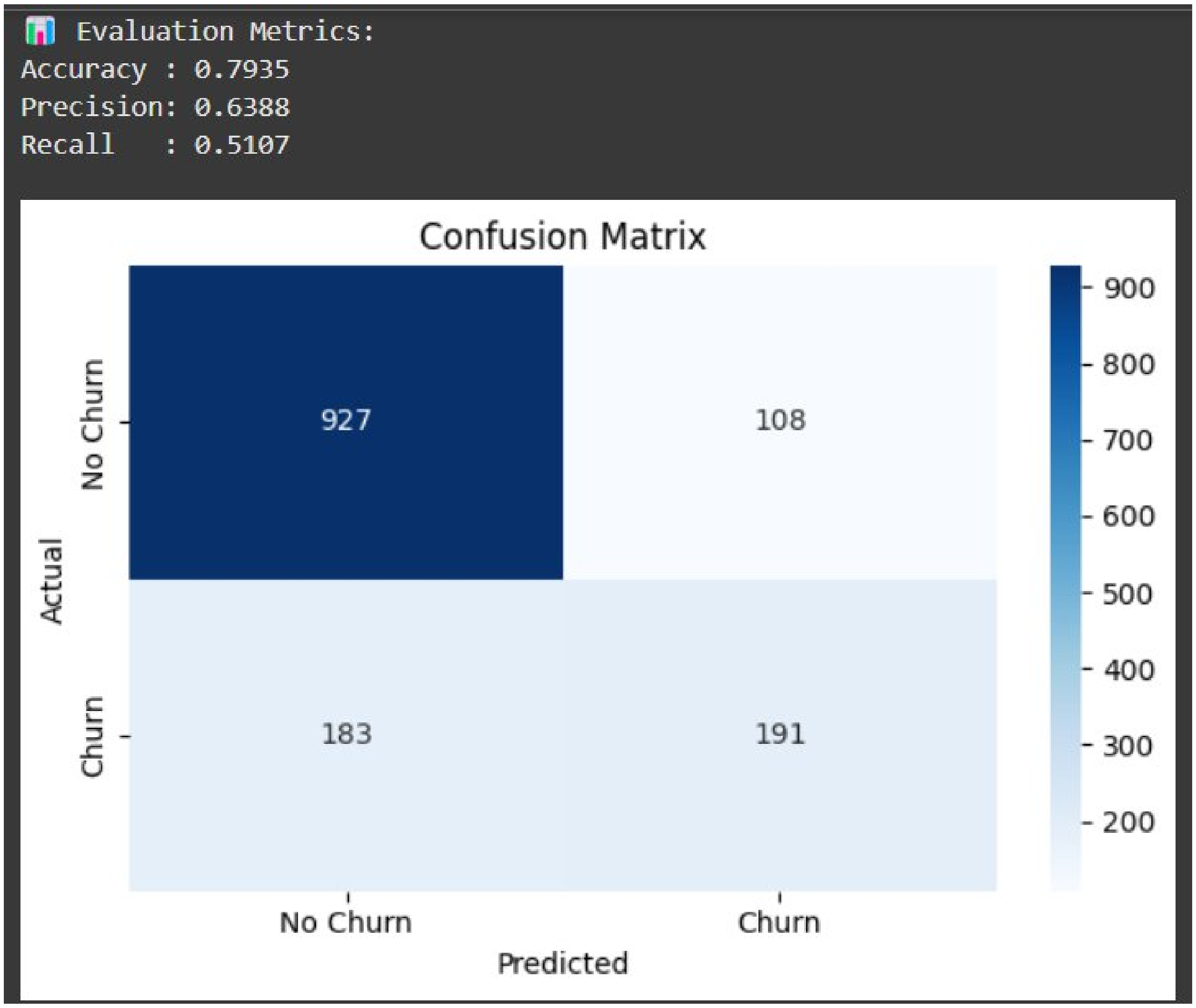
Code:

```
# Step 1: Upload the dataset
from google.colab import files
uploaded = files.upload()

# Step 2: Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score

# Step 3: Load the dataset
df = pd.read_csv("5. Classify Customer Churn.csv")

# Step 4: Data preprocessing
# Drop customerID as it is not a useful feature
df.drop("customerID", axis=1, inplace=True)

# Convert 'TotalCharges' to numeric (handle missing or bad values)
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df['TotalCharges'].fillna(0, inplace=True)

# Encode categorical variables
categorical_cols = df.select_dtypes(include=['object']).columns.tolist()
categorical_cols.remove("Churn")  # We'll encode target separately

le = LabelEncoder()
for col in categorical_cols:
    df[col] = le.fit_transform(df[col])

# Encode the target column
df['Churn'] = df['Churn'].map({'No': 0, 'Yes': 1})

# Separate features and target
X = df.drop("Churn", axis=1)
```

```
y = df["Churn"]

# Normalize the features (optional)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 5: Train/test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, random_state=42, stratify=y)

# Step 6: Train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 7: Predictions and Evaluation
y_pred = model.predict(X_test)

# Calculate Evaluation Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
```

```
# Display the metrics as text
print("📊 Evaluation Metrics:")
print(f"Accuracy : {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall   : {recall:.4f}\n")

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['No Churn', 'Churn'],
            yticklabels=['No Churn', 'Churn'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.tight_layout()
plt.show()
```

Result:



```
📊 Evaluation Metrics:
Accuracy : 0.7935
Precision: 0.6388
Recall   : 0.5107
```

References:

Dataset: Provided dataset titled *“5. Classify Customer Churn.csv”* used for model training and evaluation.

Libraries & Tools Used:

- Pandas – for data manipulation and analysis
- NumPy – for numerical operations
- Matplotlib & Seaborn – for data visualization and plotting the heatmap
- Scikit learn – for preprocessing, model building, and evaluation

Development Environment: Code implemented using Google Colab