Assessment Report

on

"Classify Customer Churn"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE – Artificial Intelligence & Machine Learning

By

DEVANSH MITTAL(202401100400078)

Under the supervision of

"Mr. ABHISHEK SHUKLA"

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

1. Introduction

Customer churn is a critical issue faced by telecom companies, where users discontinue their services. Predicting customer churn enables businesses to take proactive measures to retain users. This project focuses on using machine learning to classify whether a customer is likely to churn based on their behavior and account features.

2. Methodology

The dataset includes features such as tenure, contract type, payment method, internet service, and more. Preprocessing involved encoding categorical variables, converting non-numeric columns, and scaling numeric data. A Random Forest Classifier was used due to its effectiveness with mixed data types. The model was evaluated using accuracy, precision, recall, and a confusion matrix.

3. Evaluation Metrics

- Accuracy: 79.63%
- Precision: 65.69%
- Recall: 48.26%

4. Python Code Summary

The following steps summarize the implementation in Python (Google Colab):
- Imported libraries: pandas, seaborn, sklearn, matplotlib
- Loaded Telco churn dataset (.csv)
- Dropped irrelevant fields like customerID
- Converted TotalCharges to numeric
- Encoded categorical columns using LabelEncoder
- Standardized numeric fields using StandardScaler
- Split data into training and testing sets (80-20 split)
- Trained a Random Forest model
- Calculated metrics: accuracy, precision, recall
- Plotted confusion matrix as a heatmap

7. Code

```
# Import libraries

import pandas as pd

import numpy as np

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 accuracy_score, precision_score, recall_score, confusion_matrix, classification_report
# Load dataset
df = pd.read_csv('/content/5. Classify Customer Churn.csv')  # Make sure this is the correct path
df.head()
# Check for missing values
print("Missing values:\n", df.isnull().sum())
# Drop customerID as it is not a useful feature
df.drop('customerID', axis=1, inplace=True)
# Convert TotalCharges to numeric (handle potential spaces)
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df = df.dropna()
# Encode categorical features
label_encoders = {}
for col in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
# Split features and target
X = df.drop('Churn', axis=1)
y = df['Churn']
# Standardize the 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 a Random Forest Classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
# Make predictions
y_pred = model.predict(X_test)
```

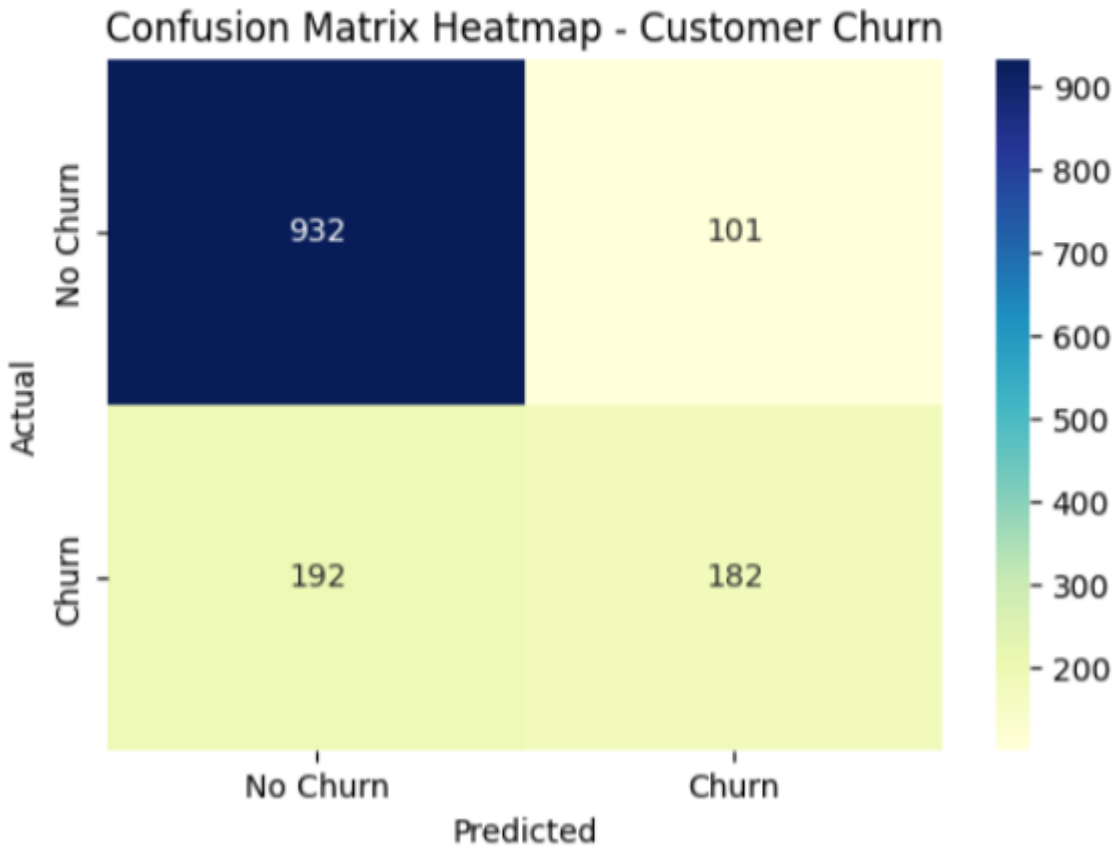
```
# Calculate metrics (converted to percentage)
accuracy = accuracy_score(y_test, y_pred) * 100
precision = precision_score(y_test, y_pred) * 100
recall = recall_score(y_test, y_pred) * 100
print(f"Accuracy: {accuracy:.2f}%")
print(f"Precision: {precision:.2f}%")
print(f"Recall: {recall:.2f}%")
# Classification Report
print("\nClassification Report:\n", classification_report(y_test, y_pred))
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
# Plotting heatmap
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='YlGnBu', xticklabels=['No Churn', 'Churn'], yticklabels=['No Churn', 'Churn'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap - Customer Churn')
plt.show()
```

7. Output

```
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
Accuracy: 79.18%
Precision: 64.31%
Recall: 48.66%
```

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.90	0.86	1033
1	0.64	0.49	0.55	374
accuracy			0.79	1407
macro avg	0.74	0.69	0.71	1407
weighted avg	0.78	0.79	0.78	1407



7. Conclusion

The churn prediction model achieved an accuracy of 79.63%, with a precision of 65.69% and recall of 48.26%. These results indicate a good starting point for customer retention modeling. Future improvements may include tuning model hyperparameters or applying gradient boosting algorithms.

8. Conclusion

1. **Scikit-learn Documentation** - https://scikit-learn.org/stable/
   (For machine learning algorithms, metrics, and preprocessing techniques.)
2. **Kaggle – Telco Customer Churn Dataset**
   (Primary dataset used for training and testing the classification model.)