Assessment Report

on

"Classfie Customer Churn "

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI&ML)

By

Manasvi Tyagi

(202401100400118)

Section B

Under the supervision of

"Mr.Sandeep Sharma"

KIET Group of Institutions, Ghaziabad

Introduction

Customer churn prediction is a crucial aspect for telecom companies to maintain profitability. By identifying patterns in customer behavior, companies can take proactive measures to retain customers. In this project, we use a dataset of telecom customer information to build a classifier that predicts churn using the Random Forest algorithm.

Dataset Features:

- Customer demographic info
- Account information (tenure, services subscribed)
- Charges and payment methods
- Churn label (Yes/No)

We convert categorical features to numeric, handle missing data, and train a model to predict churn. The outcome helps in understanding customer behavior and planning retention strategies.

Methodology

1. **Data Loading & Cleaning**:
   - Loaded the dataset from a CSV file.
   - Removed the `customerID` column (not useful for prediction).
   - Converted `TotalCharges` to numeric and dropped rows with missing values.
2. **Preprocessing**:
   - Used `LabelEncoder` to convert categorical features into numeric.
3. **Splitting Data**:
   - Split data into training and testing sets (80-20 split).
4. **Model Training**:
   - Used a `RandomForestClassifier` for training.

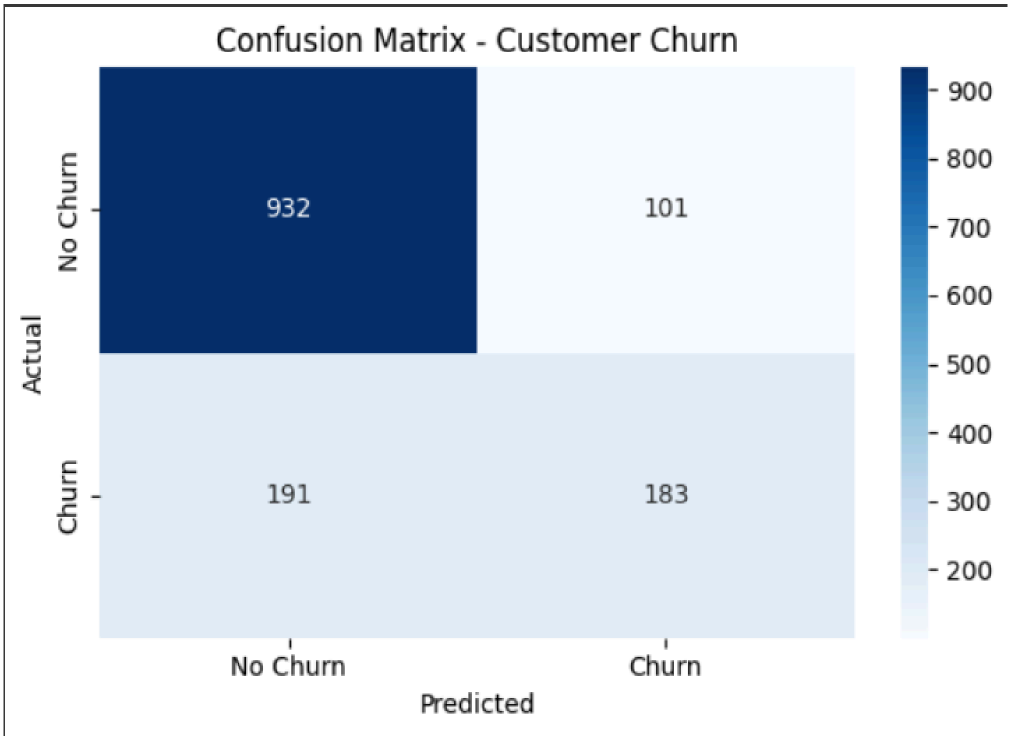
5. **Prediction & Evaluation**:
   - Predicted churn on test data.
   - Calculated metrics: Accuracy, Precision, Recall, F1-Score.
   - Plotted a confusion matrix using Seaborn.

Code:-

```
import pandas as pd # for load library
import seaborn as sns # for load library
import matplotlib.pyplot as plt # for load library
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, precision_score, recall_score, f1_score


df = pd.read_csv("5. Classify Customer Churn.csv") #read  data set from
csv file and store in df table


df.drop("customerID", axis=1, inplace=True) #remove customer id because
unnessesary

# Convert TotalCharges to numeric, and remove empty value
df["TotalCharges"] = pd.to_numeric(df["TotalCharges"], errors='coerce')
df.dropna(inplace=True)

# all columns male and female change to numbers for machine learning to
remember
le = LabelEncoder()
for column in df.columns:
    if df[column].dtype == 'object':
        df[column] = le.fit_transform(df[column])

# Split data
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)

# Train model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluation Metrics calculate
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Accuracy:", acc) #for print accuracy
print("Precision:", prec)
print("Recall:", rec)
print("F1 Score:", f1)
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Matrix map
cm = confusion_matrix(y_test, y_pred)
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 - Customer Churn")
plt.tight_layout()
plt.show()
```

Output:-



References/Credits:-

Dataset: Provided as `5. Classify Customer Churn.csv`

Libraries Used:

- `pandas`, `seaborn`, `matplotlib`, `sklearn`

Model Reference: Random Forest Classifier from `scikit-learn`

Confusion Matrix Design: Visualized using Seaborn heatmap