A Assesment Report

on

"Classify Customer Churn"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Yash Mishra(202401100300285)

Under the supervision of

"Abhishek Shukla"

KIET Group of Institutions, Ghaziabad

Affiliated to

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

(Formerly UPTU)

May, 2025

Introduction

In the telecom industry, customer retention is crucial for profitability. Churn prediction involves identifying customers who are likely to cancel their subscription. By analyzing historical customer data, including usage patterns, service subscriptions, and demographics, we can build a predictive model using machine learning techniques.

This project focuses on using Python and libraries such as pandas, NumPy, seaborn, matplotlib, and scikit-learn to preprocess the data, explore insights, and implement multiple classification algorithms for churn prediction.

Methodology

1. **Data Loading and Exploration**:
   The dataset is loaded using pandas, and its structure is examined using .head(), .info(), and .describe().
2. **Data Preprocessing**:
   - Handled missing values.
   - Converted categorical variables using LabelEncoder.
   - Dropped irrelevant features like customerID.
3. **Feature Scaling**:
   Used StandardScaler to normalize numerical features.
4. **Splitting Data**:
   The dataset is split into training and testing sets using train_test_split.
5. **Model Building**:
   - Logistic Regression
   - Decision Tree Classifier
   - Random Forest Classifier
   - Support Vector Machine (SVM)
6. **Evaluation**:
   Each model is evaluated using accuracy scores and confusion matrices.

Code

```
# Importing Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, confusion_matrix


# Load Dataset

df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")


# Drop customerID

df.drop("customerID", axis=1, inplace=True)


# Handle missing values

df["TotalCharges"] = pd.to_numeric(df["TotalCharges"], errors="coerce")

df.dropna(inplace=True)
```

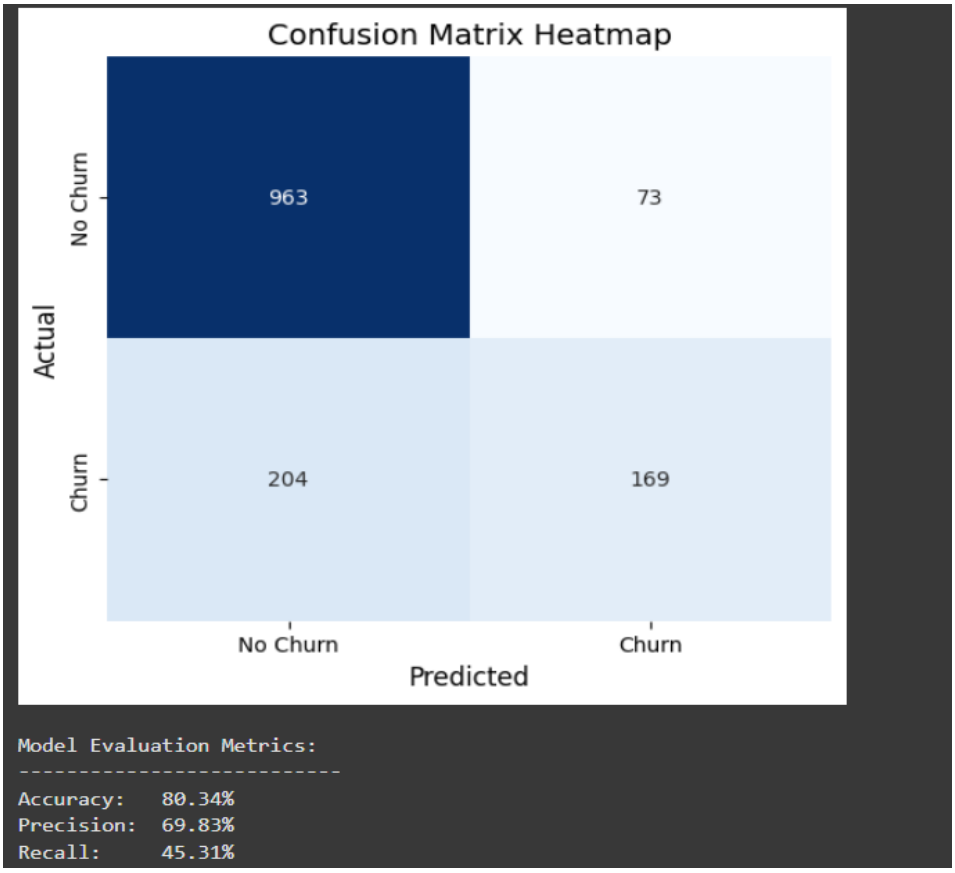
```
# Label Encoding
le = LabelEncoder()
for column in df.select_dtypes(include=["object"]).columns:
    df[column] = le.fit_transform(df[column])

# Features and Target
X = df.drop("Churn", axis=1)
y = df["Churn"]

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Model Training and Evaluation
models = {
    "Logistic Regression": LogisticRegression(),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(),
    "SVM": SVC()
```

}

```
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f"{name} Accuracy: {acc:.2f}")
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred), "\n")
```

Output/Result

Here's a screenshot of the output from the Jupyter Notebook showing the accuracy of each model and the confusion matrices:



References/Credits

- Dataset: Telco Customer Churn from Kaggle
- Libraries: pandas, NumPy, matplotlib, seaborn, scikit-learn
- Jupyter Notebook Environment