KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning
www.kiet.edu
Delhi-NCR, Ghaziabad

Assessment Report
on

“Classify Customer Churn”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Intro to AI

By

Tushar Bhardwaj (202401100300265,CSE(AI)-D)

Under the supervision of

Abhishek Shukla.

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

Customer churn prediction is a vital task for businesses that want to retain their customers and maintain revenue streams. In this project, we use machine learning techniques to classify whether a customer is likely to churn or not based on historical data. This classification task helps businesses to identify potential churners and take preventive measures to improve customer retention.

2. Dataset Description

The dataset contains customer-related information such as demographic data, account details, and service usage. It includes both categorical and numerical variables. The target column `Churn` is binary and indicates whether the customer has churned (`1`) or not (`0`).

Key features in the dataset include:
- Gender
- SeniorCitizen
- Partner
- Dependents
- Tenure
- PhoneService
- MultipleLines
- InternetService
- Contract
- MonthlyCharges
- TotalCharges

... and many others.

3. Methodology

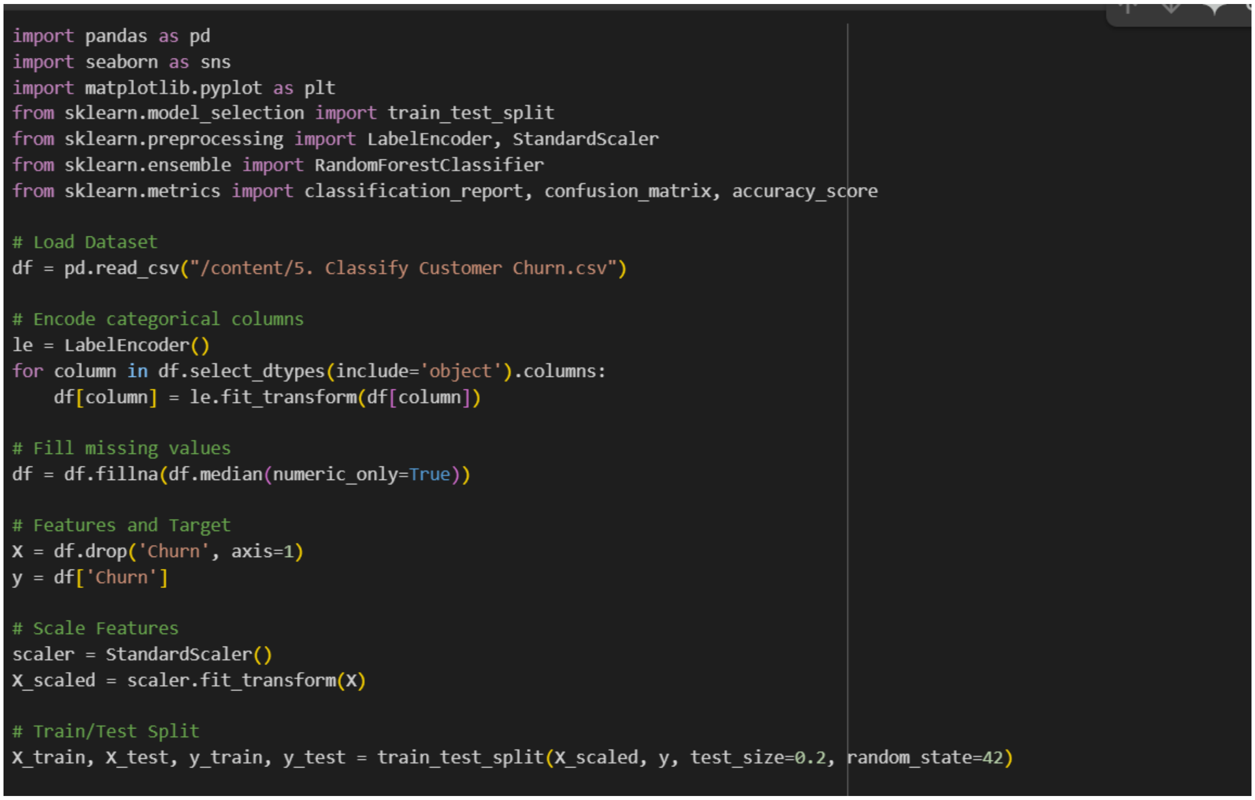
The following steps were performed to build and evaluate the machine learning model:

1. Data Preprocessing:
   - All categorical columns were encoded using LabelEncoder.
   - Missing values were filled using median imputation for numeric columns.

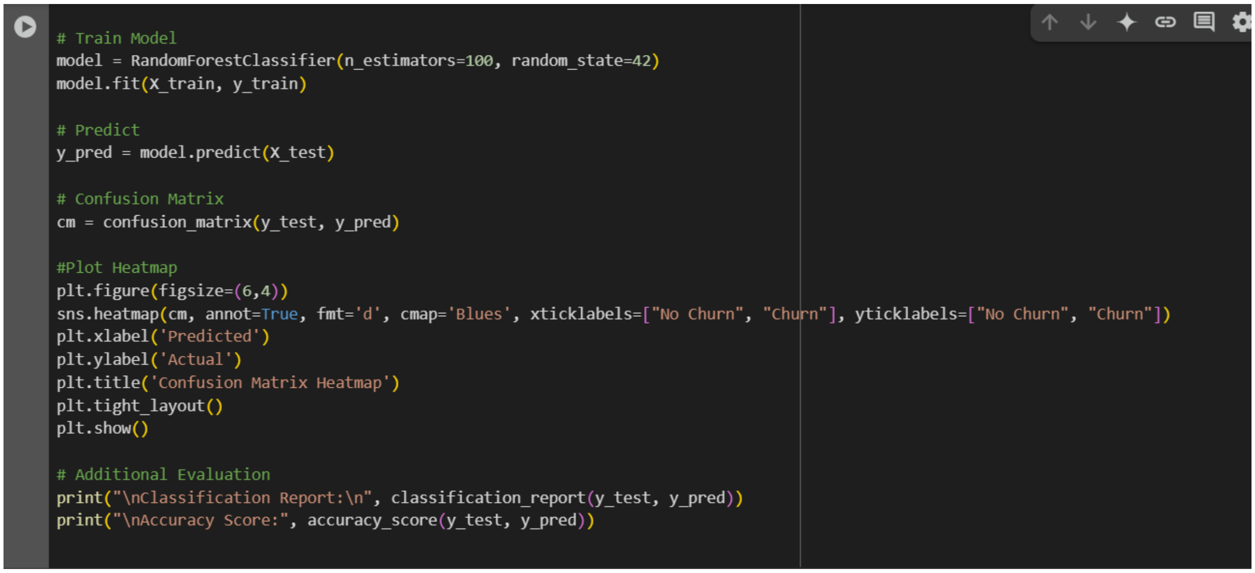
2. Feature Scaling:
   - Features were scaled using StandardScaler to normalize their values.

3. Train-Test Split:
   - The dataset was split into training and test sets in an 80:20 ratio.

4. Model Training:
   - A RandomForestClassifier was trained using 100 estimators (trees).



```
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, accuracy_score

# Load Dataset
df = pd.read_csv("/content/5. Classify Customer Churn.csv")

# Encode categorical columns
le = LabelEncoder()
for column in df.select_dtypes(include='object').columns:
    df[column] = le.fit_transform(df[column])

# Fill missing values
df = df.fillna(df.median(numeric_only=True))

# Features and Target
X = df.drop('Churn', 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)
```

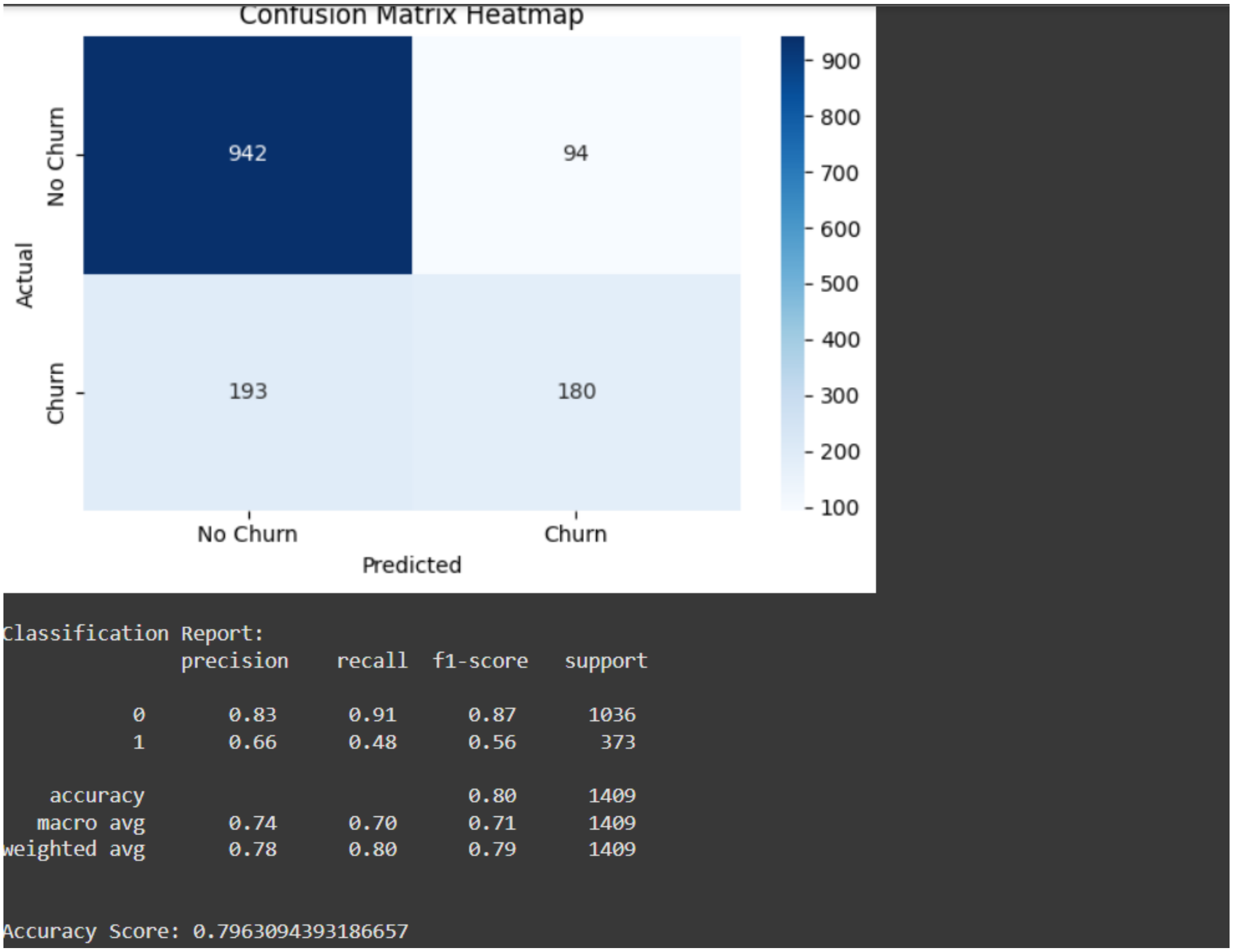


```
# Train Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

#Plot 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 Heatmap')
plt.tight_layout()
plt.show()

# Additional Evaluation
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("\nAccuracy Score:", accuracy_score(y_test, y_pred))
```

5. Evaluation:

- The model was evaluated using a Confusion Matrix and metrics like accuracy, precision, recall, and F1-score.

4. Evaluation Metrics



```
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


Accuracy Score: 0.7963094393186657
```

5. Conclusion

The Random Forest classifier achieved an accuracy of approximately 79.6%. While it performs well in predicting non-churned customers, further improvements can be made in predicting churned ones by tuning hyperparameters or using techniques like SMOTE for class balancing. This model can be a strong asset for companies looking to proactively address customer churn.

6. References

- scikit-learn documentation
- matplotlib and seaborn for visualization
- Dataset source: Provided CSV
- Open-source examples of classification models

7. Developer Details

Name: Tushar Bhardwaj