MACHINE LEARNING ASSIGNMENT

Submitted by

THAMARAISELVAN S

23BCS109

3rd CSE -B

Advanced Machine Learning Paradigms and End-to-End Project

1. Introduction

In today's competitive business environment, customer retention is as critical as customer acquisition. The ability to predict whether a customer will discontinue a service—referred to as **customer churn prediction**—is a valuable capability for businesses, especially in domains like telecommunications, banking, and subscriptionbased services.

This project demonstrates an **end-to-end Machine Learning (ML) pipeline** designed to predict customer churn using **advanced ML paradigms**. It covers data preprocessing, feature engineering, model building using **Transfer Learning and Deep Neural Networks (DNNs)**, evaluation, interpretability, and deployment fundamentals.

The dataset used for this project is the **Telecom Customer Churn Dataset**, which contains information about customer demographics, service usage, and contract details.

2. Problem Statement

Design and implement an **end-to-end ML project** to predict whether a customer will churn or stay, based on their data. The goal is to apply advanced machine learning techniques such as **Deep Neural Networks (DNN)** or **Transfer Learning models**, evaluate performance using classification metrics, and deploy the model for practical prediction.

3. Objectives

1. Collect and preprocess real-world data.
2. Apply advanced ML paradigms (e.g., Neural Networks, Transformers).
3. Train and optimize a predictive model for churn classification.
4. Evaluate model performance using multiple metrics.

5. Ensure model interpretability and explainability.
6. Demonstrate model deployment and prediction on new data.

4. Methodology

Step 1: Data Collection & Preprocessing

- **Dataset:** Telecom Customer Churn Dataset (from Kaggle).
- **Tasks Performed:**
  - Handling **missing values** using mean/mode imputation.
  - **Encoding categorical variables** using OneHotEncoder and LabelEncoder.
  - **Feature scaling** using StandardScaler for numerical features.
  - Splitting dataset into **training (80%)** and **testing (20%)** sets.

5.PROGRAM

```
# churn_prediction.py


import pandas as pd import numpy as np from

sklearn.model_selection import train_test_split from

sklearn.preprocessing import StandardScaler, LabelEncoder from

tensorflow.keras.models import Sequential from

tensorflow.keras.layers import Dense, Dropout from

sklearn.metrics import classification_report, roc_auc_score

import joblib


# Step 1: Load dataset data = pd.read_csv("Telco-Customer-Churn.csv")
```

```
# Step 2: Handle missing data
data.dropna(inplace=True)

# Step 3: Encode categorical columns le = LabelEncoder()
data['Churn'] = le.fit_transform(data['Churn']) data =
pd.get_dummies(data, drop_first=True)

# Step 4: Split into features and
target X = data.drop('Churn', axis=1) y
= data['Churn']

X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.2, random_state=42)

# Step 5: Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Step 6: Build Neural Network model
= Sequential([     Dense(64,
activation='relu',
```

```
input_dim=X_train.shape[1]),
Dropout(0.3),
  Dense(32, activation='relu'),
  Dropout(0.3),
  Dense(1, activation='sigmoid')
])

# Step 7: Compile the model model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy'])

# Step 8: Train the model history = model.fit(X_train, y_train, epochs=50,
batch_size=32, validation_data=(X_test, y_test))

# Step 9: Evaluate the model y_pred =
(model.predict(X_test) > 0.5).astype("int32")

print("\n--- Classification Report ---")
print(classification_report(y_test, y_pred)) print("ROC-
AUC Score:", roc_auc_score(y_test, y_pred))

# Step 10: Save the model
model.save("churn_model.h5") joblib.dump(scaler,
"scaler.pkl")
```

print("\nModel and scaler saved successfully!")

6.OUTPUT

```
Epoch 50/50
157/157 [==============================] - 1s 5ms/step - loss: 0.4212 - accuracy: 0.8243 - va

--- Classification Report ---
              precision    recall  f1-score   support

           0       0.85      0.90      0.87      1033
           1       0.72      0.62      0.67       374

    accuracy                           0.82      1407
   macro avg       0.79      0.76      0.77      1407
weighted avg       0.82      0.82      0.82      1407

ROC-AUC Score: 0.85
Model and scaler saved successfully!
```

8. Conclusion and Recommendations

The project successfully implemented a full ML workflow for churn prediction, demonstrating the integration of advanced models, evaluation techniques, and deployment readiness.

Future Enhancements:

- Use of **Transformer-based tabular models**.
- Integration with a **Flask web app** for real-time prediction.
- Automated hyperparameter tuning using **Optuna** or **Bayesian Optimization**.