Customer Churn Prediction Using Artificial Neural Network (ANN)

Author: Uru Onyemaobi

Introduction

Customer churn prediction is crucial for businesses to understand customer attrition and develop strategies for retention. This report focused on predicting customer churn in the telecom industry using an Artificial Neural Network (ANN). Here, I build a deep learning model to predict churn and evaluate its performance using metrics like accuracy, precision, recall, and F1-score.

As a proprietary project for an employer, I further took the intuition gained from this project and reverse-engineered this process to solve a reverse-churn problem.

1. Load and Preprocess the Data

Load the Data

We begin by loading the data into a Pandas DataFrame.

Python

```python
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

```
%matplotlib inline

df = pd.read_csv("customer_churn.csv")
df.sample(5)
```

Drop Unnecessary Columns

We can remove the `customerID` column as it doesn't contribute to our analysis.

Python

```
df.drop('customerID', axis='columns', inplace=True)
```

Data Type Conversion

We ensure the `TotalCharges` column is of type float, as it might be incorrectly stored as an object.

Python

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

Handle Missing Values

We can remove rows with missing values in the `TotalCharges` column.

Python

```
df = df[df['TotalCharges'].notnull()]
df['TotalCharges'] = df['TotalCharges'].astype(float)
```

2. Data Visualization

Data visualization helps understand the distribution and relationships between features.

Unique Values in Object Columns

We can print unique values in object columns to explore the data.

Python

```python
for column in df.columns:
  if df[column].dtype == object:
    print(f'{column}: {df[column].unique()}')
```

Replace Categorical Values

We can replace "No internet service" and "No phone service" with a simpler "No".

Python

```python
df.replace('No internet service', 'No', inplace=True)
df.replace('No phone service', 'No', inplace=True)
```

Convert Yes/No to 1/0

We can convert "Yes" and "No" to 1 and 0, respectively.

Python

```python
df['Churn'] = df['Churn'].apply(lambda x: 1 if x == 'Yes' else 0)
```

One-Hot Encoding

We can perform one-hot encoding to represent categorical columns effectively.

Python

```
df = pd.get_dummies(df, columns=['InternetService', 'Contract',
'PaymentMethod'])
```

3. Scaling

Scaling numerical features brings them to a similar range, improving model performance.

Python

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df[['tenure', 'MonthlyCharges', 'TotalCharges']] =
scaler.fit_transform(df[['tenure', 'MonthlyCharges', 'TotalCharges']])
```

4. Train-Test Split

We split the data into training and testing sets for model evaluation.

Python

```
from sklearn.model_selection import train_test_split

X = df.drop('Churn', axis='columns')  # Features
y = df['Churn']  # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=5)
```

5. Build the Model

We will build an ANN using TensorFlow and Keras.

Python

```
import tensorflow as tf
from tensorflow import keras

model = keras.Sequential([
  keras.layers.Dense(26, input_shape=(26,), activation='relu'),  # First
hidden layer
  keras.layers.Dense(15, activation='relu'),                    # Second
hidden layer
  keras.layers.Dense(1, activation='sigmoid')                    # Output
layer
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

model.fit(X_train, y_train, epochs=100)
```

6. Evaluate the Model

Accuracy

We evaluate the model's performance on the test data to determine its accuracy.

Python

```
model.evaluate(X_test, y_test)
```

Predictions

We make predictions on the test data to assess the model's ability to generalize.

Python

```python
yp = model.predict(X_test)
y_pred = [1 if x > 0.
```

Classification Report

We generate a classification report to evaluate precision, recall, and F1-score.

Python

```python
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

Confusion Matrix

We plot a confusion matrix to visualize the performance of the model.

Python

```python
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(10,7))
sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Truth')
plt.show()
```

Conclusion

This report demonstrates the process of building and evaluating a deep-learning model for customer churn prediction. The artificial neural network (ANN) effectively predicts customer churn with significant accuracy, and the evaluation metrics provide insights into the model's performance.

By following these steps, we can predict customer churn in various industries, helping businesses retain customers and improve their services.