```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import warnings  # Correct way to import the warnings module

# Suppress warnings
warnings.filterwarnings("ignore")

data = pd.read_csv(r"C:\Users\Neha\Downloads\Netflix (2).csv")

print(data.head())

print(data.info())
print(data.isnull().sum())
```

```
                 Title                                              Genre  \
0     Lets Fight Ghost  Crime, Drama, Fantasy, Horror, Romance
1  HOW TO BUILD A GIRL                                             Comedy
2           Centigrade                                    Drama, Thriller
3                ANNE+                                              Drama
4                Moxie                          Animation, Short, Drama

          Languages Series or Movie  Hidden Gem Score  \
0  Swedish, Spanish          Series               4.3
1           English           Movie               7.0
2           English           Movie               6.4
3           Turkish          Series               7.7
4           English           Movie               8.1

                                Country Availability        Runtime  \
0                                           Thailand  < 30 minutes
1                                             Canada      1-2 hour
2                                             Canada      1-2 hour
3                               Belgium,Netherlands  < 30 minutes
4  Lithuania,Poland,France,Iceland,Italy,Spain,Gr...      1-2 hour

          Director                        Writer  \
0  Tomas Alfredson         John Ajvide Lindqvist
1    Coky Giedroyc                 Caitlin Moran
2    Brendan Walsh  Brendan Walsh, Daley Nixon
3              NaN                           NaN
4    Stephen Irwin                           NaN

                                              Actors View Rating  IMDb
```

```
Score  \
0  Kåre Hedebrant, Per Ragnar, Lina Leandersson, ...           R
7.9
1  Paddy Considine, Cleo, Beanie Feldstein, Dónal...           R
5.8
2                 Genesis Rodriguez, Vincent Piazza     Unrated
4.3
3  Vahide Perçin, Gonca Vuslateri, Cansu Dere, Be...         NaN
6.5
4                                      Ragga Gudrun         NaN
6.3

   Rotten Tomatoes Score  Metacritic Score  Awards Nominated For
Boxoffice  \
0                   98.0              82.0                  57.0
$2,122,065
1                   79.0              69.0                   NaN
$70,632
2                    NaN              46.0                   NaN
$16,263
3                    NaN               NaN                   NaN
NaN
4                    NaN               NaN                   4.0
NaN

  Release Date Netflix Release Date                            Netflix
Link  \
0    12-Dec-08           04-03-2021
https://www.netflix.com/watch/81415947
1    08-May-20           04-03-2021
https://www.netflix.com/watch/81041267
2    28-Aug-20           04-03-2021
https://www.netflix.com/watch/81305978
3    01-Oct-16           04-03-2021
https://www.netflix.com/watch/81336456
4    22-Sep-11           04-03-2021
https://www.netflix.com/watch/81078393

   IMDb Votes
0    205926.0
1      2838.0
2      1720.0
3      1147.0
4        63.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15483 entries, 0 to 15482
Data columns (total 20 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Title                  9227 non-null   object
 1   Genre                  7886 non-null   object 
 2   Languages              7678 non-null   object 
 3   Series or Movie        9227 non-null   object 
 4   Hidden Gem Score       7605 non-null   float64
 5   Country Availability   9216 non-null   object 
 6   Runtime                9227 non-null   object 
 7   Director               6076 non-null   object 
 8   Writer                 6127 non-null   object 
 9   Actors                 7735 non-null   object 
 10  View Rating            3999 non-null   object 
 11  IMDb Score             7746 non-null   float64
 12  Rotten Tomatoes Score  2771 non-null   float64
 13  Metacritic Score       1658 non-null   float64
 14  Awards Nominated For   3770 non-null   float64
 15  Boxoffice              1434 non-null   object 
 16  Release Date           7796 non-null   object 
 17  Netflix Release Date   9368 non-null   object 
 18  Netflix Link           9368 non-null   object 
 19  IMDb Votes             7745 non-null   float64
dtypes: float64(6), object(14)
memory usage: 2.4+ MB
None
Title                     6256
Genre                     7597
Languages                 7805
Series or Movie           6256
Hidden Gem Score          7878
Country Availability      6267
Runtime                   6256
Director                  9407
Writer                    9356
Actors                    7748
View Rating              11484
IMDb Score                7737
Rotten Tomatoes Score    12712
Metacritic Score         13825
Awards Nominated For     11713
Boxoffice                14049
Release Date              7687
Netflix Release Date      6115
Netflix Link              6115
IMDb Votes                7738
dtype: int64

numerical_cols = data.select_dtypes(include=['float64',
'int64']).columns
data[numerical_cols] =
data[numerical_cols].fillna(data[numerical_cols].mean())

categorical_cols = data.select_dtypes(include=['object']).columns
```

```python
data[categorical_cols] = data[categorical_cols].fillna('Unknown')

label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

target = 'Series or Movie'
features = data.drop(columns=[target])

X = features
y = data[target]

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

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

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

```
RandomForestClassifier(random_state=42)
```

```python
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()
```

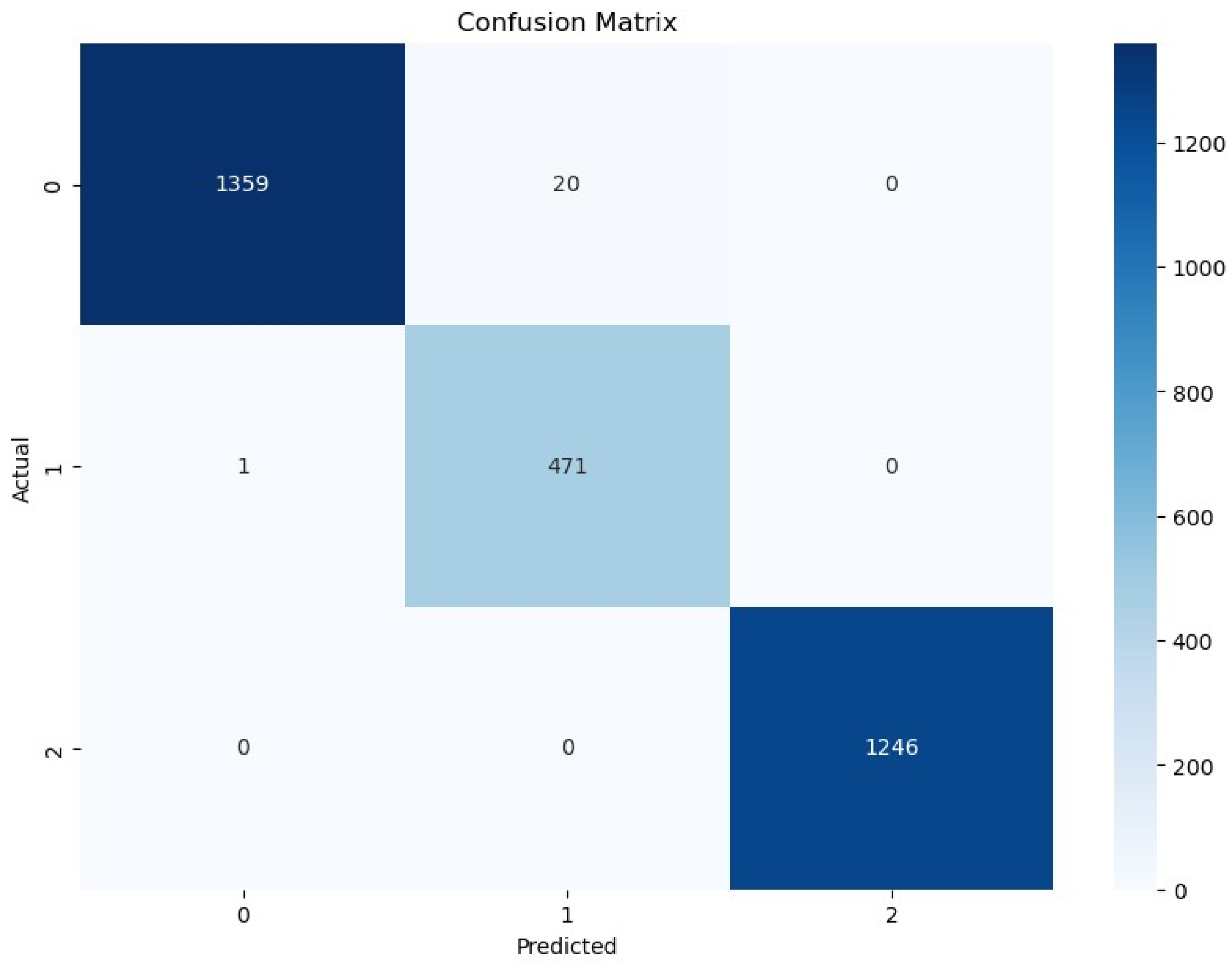
```
Accuracy: 0.9932192444300937

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      1379
           1       0.96      1.00      0.98       472
           2       1.00      1.00      1.00      1246

    accuracy                           0.99      3097
```

```
   macro avg       0.99      0.99      0.99      3097
weighted avg       0.99      0.99      0.99      3097
```



```python
import joblib

joblib.dump(model, 'random_forest_model.pkl')
joblib.dump(scaler, 'scaler.pkl')
```

```
['scaler.pkl']
```

```python
loaded_model = joblib.load('random_forest_model.pkl')
loaded_scaler = joblib.load('scaler.pkl')


sample_data = X_test[0].reshape(1, -1)
scaled_data = loaded_scaler.transform(sample_data)
prediction = loaded_model.predict(scaled_data)

print(f"Predicted Class: {prediction}")
```

```
Predicted Class: [0]
```

```python
import pandas as pd

# Load the dataset
data = pd.read_csv(r"C:\Users\Neha\Downloads\Netflix (2).csv")

# Display the first few rows to confirm loading
print(data.head())
```

```
                 Title                                        Genre  \
0     Lets Fight Ghost  Crime, Drama, Fantasy, Horror, Romance
1  HOW TO BUILD A GIRL                                       Comedy
2           Centigrade                              Drama, Thriller
3                ANNE+                                        Drama
4                Moxie                      Animation, Short, Drama

          Languages Series or Movie  Hidden Gem Score  \
0  Swedish, Spanish          Series               4.3
1           English           Movie               7.0
2           English           Movie               6.4
3           Turkish          Series               7.7
4           English           Movie               8.1

                                Country Availability      Runtime  \
0                                           Thailand  < 30 minutes
1                                             Canada      1-2 hour
2                                             Canada      1-2 hour
3                               Belgium,Netherlands  < 30 minutes
4  Lithuania,Poland,France,Iceland,Italy,Spain,Gr...      1-2 hour

          Director                       Writer  \
0  Tomas Alfredson         John Ajvide Lindqvist
1     Coky Giedroyc                Caitlin Moran
2     Brendan Walsh  Brendan Walsh, Daley Nixon
3               NaN                          NaN
4     Stephen Irwin                          NaN

                                              Actors View Rating  IMDb
Score  \
0  Kåre Hedebrant, Per Ragnar, Lina Leandersson, ...           R
7.9
1  Paddy Considine, Cleo, Beanie Feldstein, Dónal...           R
5.8
2                Genesis Rodriguez, Vincent Piazza     Unrated
4.3
3  Vahide Perçin, Gonca Vuslateri, Cansu Dere, Be...         NaN
6.5
4                                        Ragga Gudrun         NaN
6.3
```

```
   Rotten Tomatoes Score  Metacritic Score  Awards Nominated For
Boxoffice  \
0                   98.0              82.0                  57.0
$2,122,065
1                   79.0              69.0                   NaN
$70,632
2                    NaN              46.0                   NaN
$16,263
3                    NaN               NaN                   NaN
NaN
4                    NaN               NaN                   4.0
NaN

  Release Date Netflix Release Date                            Netflix
Link  \
0    12-Dec-08           04-03-2021
https://www.netflix.com/watch/81415947
1    08-May-20           04-03-2021
https://www.netflix.com/watch/81041267
2    28-Aug-20           04-03-2021
https://www.netflix.com/watch/81305978
3    01-Oct-16           04-03-2021
https://www.netflix.com/watch/81336456
4    22-Sep-11           04-03-2021
https://www.netflix.com/watch/81078393

   IMDb Votes
0    205926.0
1      2838.0
2      1720.0
3      1147.0
4        63.0
```

```
# Create a dictionary with column names as keys and their data types
as values
column_data_types = {col: data[col].dtype for col in data.columns}
print(column_data_types)
```

```
{'Title': dtype('O'), 'Genre': dtype('O'), 'Languages': dtype('O'),
'Series or Movie': dtype('O'), 'Hidden Gem Score': dtype('float64'),
'Country Availability': dtype('O'), 'Runtime': dtype('O'), 'Director':
dtype('O'), 'Writer': dtype('O'), 'Actors': dtype('O'), 'View Rating':
dtype('O'), 'IMDb Score': dtype('float64'), 'Rotten Tomatoes Score':
dtype('float64'), 'Metacritic Score': dtype('float64'), 'Awards
Nominated For': dtype('float64'), 'Boxoffice': dtype('O'), 'Release
Date': dtype('O'), 'Netflix Release Date': dtype('O'), 'Netflix Link':
dtype('O'), 'IMDb Votes': dtype('float64')}
```

```
# Find the number of duplicate rows
duplicates_count = data.duplicated().sum()
print(f"Number of duplicate rows: {duplicates_count}")

# Drop duplicate rows
data_deduplicated = data.drop_duplicates()
print(f"Data after dropping duplicates has {len(data_deduplicated)}
rows.")
```

```
Number of duplicate rows: 6123
Data after dropping duplicates has 9360 rows.
```

```
# Check if all film titles have a unique release date
# Group by 'Title' and take the earliest release date if there are
multiple dates for the same title
data_deduplicated['Release Date'] =
pd.to_datetime(data_deduplicated['Release Date'], errors='coerce')
unique_release_dates = data_deduplicated.groupby('Title')['Release
Date'].min().reset_index()

# Merge the oldest release dates back to the original data
data_deduplicated = pd.merge(data_deduplicated.drop('Release Date',
axis=1), unique_release_dates, on='Title', how='left')

print(f"Number of unique film titles with oldest release dates:
{data_deduplicated['Title'].nunique()}")
```

```
Number of unique film titles with oldest release dates: 9092
```

```
# Calculate the percentage of missing values for each column
missing_percentages = data_deduplicated.isnull().mean() * 100
print("Missing value percentages per column:")
print(missing_percentages)
```

```
Missing value percentages per column:
Title                     1.517094
Genre                    15.844017
Languages                18.023504
Series or Movie           1.517094
Hidden Gem Score         18.846154
Country Availability      1.634615
Runtime                   1.517094
Director                 35.181624
Writer                   34.626068
Actors                   17.457265
View Rating              57.350427
IMDb Score               17.339744
Rotten Tomatoes Score    70.416667
Metacritic Score         82.307692
Awards Nominated For     59.754274
Boxoffice                84.679487
```

```
Netflix Release Date      0.010684
Netflix Link              0.010684
IMDb Votes               17.350427
Release Date             18.173077
dtype: float64
```