PREDICTING IMDb SCORES
PHASE-3

Objective:

The objective of the project is to build the IMDb score prediction model. In this phase we will import the given movie dataset and preprocess the dataset and analyse it.

The given dataset is:https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores

Data Cleaning:

Data cleaning, also referred to as data cleansing and data scrubbing, is one of the most important steps

Data cleansing is a subprocess of the data science process that focuses on removing errors in the data so that the data becomes a true and consistent representation of the processes it originates from.

The first type is the interpretation error, such as incorrect use of terminologies.

The second type of error points to inconsistencies between data sources or against your company's standardized values.

Data cleaning is the process of fixing or removing

- Corrupted data
- Incorrectly formatted
- Duplicate
- Incomplete data

If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset.

Joining Tables:

Joining tables is used to combine the information of one observation found in one table with the information that was found in another table.

Appending Tables:

Appending or stacking tables is effectively adding observations from one table to another table.

Transforming Data:

Certain models require their data to be in a certain shape. So the datas can be changed according to our need. Data transformation is the process of converting data from one format or structure into another. Transformation processes can also be referred to as data wrangling, or data munging, transforming and mapping data from one "raw" data form into another format for warehousing and analyzing

Let us see the steps done in phase-1 of project development:

Step 1: The required libraries are imported.

Step2: The given dataset is loaded in the program.

Step 3: To check whether the uploaded dataset is correct, a sample data is displayed in the output.

Step 4: The information from the dataset is summarized for better understanding.

Step 5: Then preprocessing is done, in which missing data, duplicate data are checked.

Step 6: If any modify the data or print False.

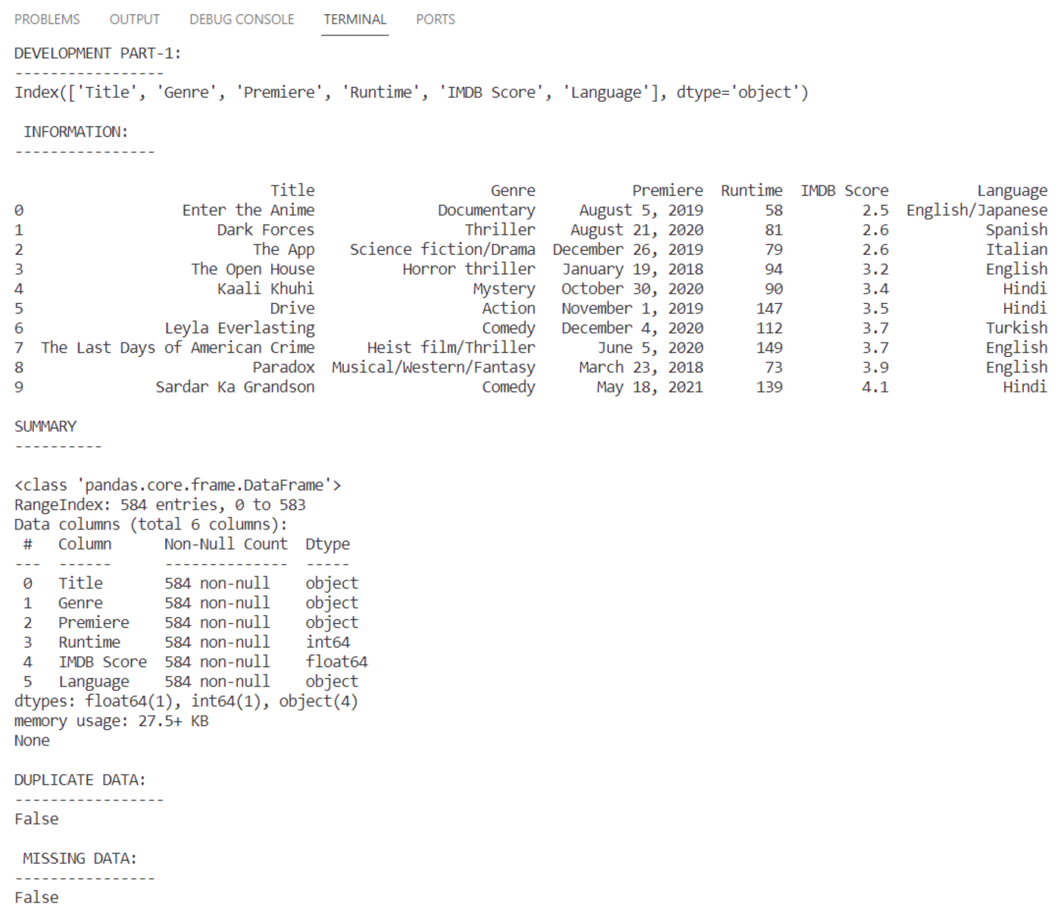
When the dataset is summarized range index, total number of columns, datatypes, column heading and memory usage are viewed.

CODE:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.ensemble import GradientBoostingRegressor

# loading the data set
data = pd.read_csv("NetflixOriginals.csv",encoding='latin-1')
print('\n')

print('DEVELOPMENT PART-1:')
print('------------------')
print(data.columns)
print('\n INFORMATION:')
print('----------------\n')

print(data.head(10))
# getting the information about the data for bettter understanding
print('\nSUMMARY')
print('---------- \n')
print(data.info())
print('\nDUPLICATE DATA:')
print('-----------------')
# checking for any duplicate data in the data set
print(data.duplicated().any())
# encoding the categorical features
label_encoder = LabelEncoder()
data['Genre'] = label_encoder.fit_transform(data['Genre'])
data['Language'] = label_encoder.fit_transform(data['Language'])
# encoding the premiere data since they contain month date, year and month
date. year
data['Premiere_comma'] = pd.to_datetime(data['Premiere'], format='%B %d, %Y',
errors='coerce')
data['Premiere_period'] = pd.to_datetime(data['Premiere'], format='%B %d. %Y',
errors='coerce')

# Merge the two columns into a single datetime column
data['Premiere'] =
data['Premiere_comma'].combine_first(data['Premiere_period'])

# Drop the temporary columns
data.drop(['Premiere_comma', 'Premiere_period'], axis=1, inplace=True)

# Convert valid datetime values to timestamps and handle NaT values
data['Premiere'] = data['Premiere'].apply(lambda x: x.timestamp() if not
pd.isna(x) else None)
```

```python
data.head()
print('\n MISSING DATA:')
print('----------------')
# checking for any missing data in the premiere
print(data.isnull().values.any())

print('\n')
```

OUTPUT:



```
DEVELOPMENT PART-1:
-----------------
Index(['Title', 'Genre', 'Premiere', 'Runtime', 'IMDB Score', 'Language'], dtype='object')

 INFORMATION:
----------------

                             Title                    Genre           Premiere  Runtime  IMDB Score          Language
0                  Enter the Anime              Documentary     August 5, 2019       58         2.5  English/Japanese
1                      Dark Forces                 Thriller    August 21, 2020       81         2.6           Spanish
2                          The App    Science fiction/Drama  December 26, 2019       79         2.6           Italian
3                   The Open House          Horror thriller   January 19, 2018       94         3.2           English
4                      Kaali Khuhi                  Mystery   October 30, 2020       90         3.4             Hindi
5                            Drive                   Action   November 1, 2019      147         3.5             Hindi
6                Leyla Everlasting                   Comedy   December 4, 2020      112         3.7           Turkish
7  The Last Days of American Crime      Heist film/Thriller       June 5, 2020      149         3.7           English
8                          Paradox  Musical/Western/Fantasy     March 23, 2018       73         3.9           English
9               Sardar Ka Grandson                   Comedy       May 18, 2021      139         4.1             Hindi

SUMMARY
----------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 584 entries, 0 to 583
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Title       584 non-null    object
 1   Genre       584 non-null    object
 2   Premiere    584 non-null    object
 3   Runtime     584 non-null    int64
 4   IMDB Score  584 non-null    float64
 5   Language    584 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 27.5+ KB
None

DUPLICATE DATA:
-----------------
False

 MISSING DATA:
----------------
False
```