Batch: A2 **Roll No.: 16010122041**

Experiment 02

Grade: AA / AB / BB / BC / CC / CD /DD

Title: Dataset preparing/ pre-processing

Objective:

1. To learn how to prepare the dataset

2. To learn various steps in Data -Preprocessing

Course Outcome:

CO1 : Learn how to locate and download datasets, extract insights from that data and present their findings in a variety of different formats.

Books/ Journals/ Websites referred:

https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset

https://pandas.pydata.org/

Resources used:

1. Kaggle

Theory (About Data Preprocessing):

Different steps in Data Preprocessing:

1. **Finding missing, null values:**

    - df. isnull() : This will return boolean value for every column in the data frame, i.e. if the vale is null it returns True, and False values are other than null.
    - df. isnull(). sum() : This code will give you total number of null values in each features in the data frame.

- df. isnull(). any() : This will return Boolean value for every column, True if column has null values, False if column doesn't have null values.
- isnull(). values. any() : This will check if missing values are present or not, will give single line Boolean answer
- isnull(). sum(). sum() : This will return the total count of missing values

2. Replacing missing, null values with statistical parameters.

- df['language'].fillna(0,inplace=True)

  Replace missing values with '0'

3. Encoding categorical data

The categorical data must be encoded to numbers before we can use it to fit and evaluate a model.

4. Normalization

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Platform used by the student: Python

Working (Put the code and Output for each Data Preprocessing task):

- **Loading dataset**

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

# loading dataset
df = pd.read_csv('Datasets/update.csv', delimiter=',' )
```

- **Check for missing values in each individual column**

  Code:

```
#Check for missing values in each individual columns
print (df['title'].isnull())
print (df['year'].isnull())
print (df['genre'].isnull())
print (df['duration'].isnull())
print (df['country'].isnull())
print (df['language'].isnull())
print (df['director'].isnull())
print (df['actors'].isnull())
print (df['description'].isnull())
print (df['avg_vote'].isnull())
print (df['reviews_from_users'].isnull())
print (df['reviews_from_critics'].isnull())
```

Output:

```
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
Name: title, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
Name: year, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
```

```
Name: genre, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
Name: duration, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
Name: country, Length: 85855, dtype: bool
0         True
1         True
2         True
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
```

```
Name: language, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
Name: director, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
Name: actors, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852     True
85853     True
85854    False
```

```
Name: description, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852    False
85853    False
85854    False
Name: avg_vote, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850     True
85851    False
85852     True
85853     True
85854     True
Name: reviews_from_users, Length: 85855, dtype: bool
0        False
1        False
2        False
3        False
4        False
         ...
85850    False
85851    False
85852     True
85853     True
85854    False
Name: reviews_from_critics, Length: 85855, dtype: bool
```

- **Check for missing values in columns**

Code:

```
print(df.isnull().any())
print(df.isnull().sum())
```

Output:

```
title                  False
year                   False
genre                  False
duration               False
country                 True
language                True
director                True
actors                  True
description             True
avg_vote               False
reviews_from_users      True
reviews_from_critics    True
dtype: bool
title                     0
year                      0
genre                     0
duration                  0
country                  64
language                901
director                 87
actors                   69
description            2115
avg_vote                  0
reviews_from_users     7597
reviews_from_critics    11797
dtype: int64
```

Code:

```
print(df.isnull().values.any()) #check if missing values a
re present or not, single line answer
print(df.isnull().sum().sum()) #total count of missing val
ues
```

Output:

```
True
22630
```

- **Replacing missing values in a column**

  Code:

```
df['language'].fillna(0,inplace=True)
print(df['language'])
```

  Output:

```
0                   0
1                   0
2                   0
3             English
4             Italian
             ...
85850          French
85851    German, Dutch
85852        Malayalam
85853          Turkish
85854          Catalan
Name: language, Length: 85855, dtype: object
```

- **Encoding the dataset**

  Code:

```
col_list = ["genre"]
# Encoding
genre = {'Romance':1,
             'Drama':2,
             'Crime':3,
             'Action':4,
             'Adventure':5,
             'Comedy':6,
             'Horror':7,
             'Thriller':8,
             'Fantasy':9,
             'Mystery' : 10,
             'Animation' : 11,
             'Family' : 12,
             'Sci-Fi' : 13,
             'Biography' : 14,
             'Sport' : 15,
             'Musical' : 16,
             'History' : 17,
             'War' : 18,
```

```
            }
df['genre'] = df.genre.map(genre)
print(df['genre'])
```

Output:

```
0        1.0
1        NaN
2        2.0
3        NaN
4        NaN
        ...
85850    6.0
85851    NaN
85852    2.0
85853    NaN
85854    2.0
Name: genre, Length: 85855, dtype: float64
```

- **Normalization**

Code:

```
# normalization of columns
df['avg_vote'].plot(kind = 'bar')

df_min_max_scaled = df.copy()

# apply normalization techniques to the Rating column
column = 'avg_vote'
df_min_max_scaled[column] = (df_min_max_scaled[column] -
 df_min_max_scaled[column].min()) / (df_min_max_scaled[col
umn].max() - df_min_max_scaled[column].min())

# view normalized data
print(df_min_max_scaled)
```

Output:

```
                          title  year                  genre  ... avg_vote reviews_from_users reviews_from_critics
0                    Miss Jerry  1894                Romance  ... 0.550562               1.0                2.0
1      The Story of the Kelly Gang  1906   Biography, Crime, Drama  ... 0.573034               7.0                7.0
2                  Den sorte drøm  1911                  Drama  ... 0.539326               5.0                2.0
3                      Cleopatra  1912          Drama, History  ... 0.471910              25.0                3.0
4                     L'Inferno  1911  Adventure, Drama, Fantasy  ... 0.674157              31.0               14.0
...                         ...   ...                     ...  ...      ...               ...                ...
85850                   Le lion  2020                 Comedy  ... 0.483146               NaN                4.0
85851  De Beentjes van Sint-Hildegard  2020           Comedy, Drama  ... 0.752809               6.0                4.0
85852      Padmavyuhathile Abhimanyu  2019                  Drama  ... 0.775281               NaN                NaN
85853            Sokagin Çocuklari  2019           Drama, Family  ... 0.606742               NaN                NaN
85854     La vida sense la Sara Amat  2019                  Drama  ... 0.640449               NaN                2.0

[85855 rows x 12 columns]
```

Platform used by the student: R

Working (Put the code and Output for each Data Preprocessing task):

- **Loading and finding the summary of the dataset**

  Code:

```
dataset = read.csv("update.csv")


# finding the summary of the dataset
summary(dataset)
```

  Output:

```
    title               year               genre              duration     
 Length:85855       Length:85855       Length:85855       Min.   : 41.0  
 Class :character   Class :character   Class :character   1st Qu.: 88.0  
 Mode  :character   Mode  :character   Mode  :character   Median : 96.0  
                                                          Mean   :100.4  
                                                          3rd Qu.:108.0  
                                                          Max.   :808.0  

   country            language           director           actors         
 Length:85855       Length:85855       Length:85855       Length:85855      
 Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character  



 description           avg_vote     reviews_from_users reviews_from_critics
 Length:85855       Min.   :1.000   Min.   :    1.00   Min.   :  1.00     
 Class :character   1st Qu.:5.200   1st Qu.:    4.00   1st Qu.:  3.00     
 Mode  :character   Median :6.100   Median :    9.00   Median :  8.00     
                    Mean   :5.899   Mean   :   46.04   Mean   : 27.48     
                    3rd Qu.:6.800   3rd Qu.:   27.00   3rd Qu.: 23.00     
                    Max.   :9.900   Max.   :10472.00   Max.   :999.00     
                                    NA's   :7597       NA's   :11797      
```

- **Encoding the dataset**

  Code:

```
dataset = read.csv("update.csv")

#Encoding the data
dataset$genre = factor(dataset$genre, levels = c('Romance',
            'Drama',
```

```
            'Crime',
            'Action',
            'Adventure',
            'Comedy',
            'Horror',
            'Thriller',
            'Fantasy',
            'Mystery',
            'Animation',
            'Family',
            'Sci-Fi',
            'Biography',
            'Sport',
            'Musical',
            'History',
            'War'
            ),labels=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,1
8))

print(dataset$genre)
```

Output:

```
71723] <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> 2    <NA> <NA> 4    <NA> 3
71737] <NA> 2    <NA> <NA> <NA> <NA> 2    <NA> <NA> <NA> <NA> 6    <NA> <NA>
71751] <NA> 2    7    <NA> <NA> <NA> <NA> <NA> <NA> 2    2    <NA> 8    <NA>
71765] <NA> 2    <NA> <NA> <NA> 7    <NA> 2    <NA> 1    <NA> <NA> <NA> <NA>
71779] 2    <NA> <NA> 8    <NA> 7    <NA> <NA> 6    <NA> <NA> <NA> <NA> <NA>
71793] <NA> 2    <NA> 2    <NA> <NA> <NA> 7    <NA> 7    <NA> <NA> <NA> <NA>
71807] <NA> 11   <NA> <NA> <NA> <NA> 8    <NA> <NA> <NA> <NA> <NA> <NA> <NA>
71821] 4    <NA> 2    12   6    2    <NA> <NA> <NA> 4    7    <NA> <NA> <NA>
71835] <NA> <NA> <NA> 6    <NA> <NA> <NA> <NA> 2    <NA> <NA> 2    <NA> <NA>
71849] <NA> <NA> <NA> 2    <NA> 7    <NA> 2    <NA> 2    <NA> 8    <NA> <NA>
71863] <NA> <NA> <NA> <NA> <NA> 2    <NA> 16   <NA> <NA> <NA> <NA> <NA> <NA>
71877] <NA> <NA> <NA> 4    <NA> <NA> <NA> 4    <NA> <NA> 2    2    <NA> <NA>
71891] <NA> 7    <NA> 2    <NA> <NA> <NA> 6    6    <NA> <NA> 4    <NA> <NA>
71905] <NA> 13   <NA> 2    <NA> <NA> 6    <NA> <NA> <NA> 8    <NA> <NA> <NA>
71919] <NA> <NA> <NA> <NA> 2    <NA> <NA> 7    <NA> <NA> <NA> <NA> <NA> <NA>
71933] <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> 2    2    <NA> <NA> <NA> 6
71947] 6    <NA> <NA> <NA> <NA> 13   <NA> <NA> 2    <NA> 4    <NA> 4    <NA>
71961] 2    <NA> 7    <NA> <NA> <NA> <NA> <NA> 7    <NA> <NA> <NA> 6    <NA>
71975] <NA> <NA> 2    <NA> <NA> <NA> 6    <NA> 2    <NA> <NA> <NA> <NA> <NA>
71989] <NA> 2    2    13   <NA> <NA> 2    <NA> 2    <NA> <NA> <NA> <NA> 2
72003] 2    <NA> <NA> 7    12   <NA> <NA> 7    <NA> <NA> <NA> <NA> 2    <NA>
```

Conclusion

Thus, we learnt how to locate and download datasets, and process the data, extract insights from that data and present their findings in a variety of different formats.

Post Lab Question:

1. **Write the importance of Data Preprocessing in Software System Designing**

   The importance of Data Preprocessing in Software System Designing is:

- It reduces overall development cycle
- Makes the data process easier to maintain (no matter which programming language or data preparation tool is used)
- Make the system more open and easier to operate
- Ensure data quality from the beginning