COURSE CERTIFICATE

Sep 4, 2021

YOHANES SETIAWAN

has successfully completed

Exploratory Data Analysis for Machine Learning

an online non-credit course authorized by IBM and offered through Coursera

Joseph Santarcangelo
Senior Data Scientist
IBM

Svitlana Kramar
Data Science Content Developer
Skills Network

Verify at:
https://coursera.org/verify/PNTSVMHEU52U

Coursera has confirmed the identity of this individual and their participation in the course.

coursera

Yohanes Setiawan

Exploratory Data Analysis for Machine Learning

Analysed by Yohanes Setiawan

I. Dataset Description

The dataset was taken from Kaggle (https://www.kaggle.com/bharatnatrayn/movies-dataset-for-feature-extracion-prediction). The dataset has name "MOVIES DATASET FOR FEATURE EXTRACTION, PREDICTION". According to the owner, the data is scrapped from IMDB Top Netflix Movies and TV Shows. This dataset contains 9 columns with further explanation:

- MOVIES → the movies/TV-shows name
- YEAR → the year of the movies/TV shows
- GENRE → the genres of the movies/TV shows
- RATING → the movies/TV shows based on user's assessment
- ONE-LINE → a brief description about the movies/TV shows
- STARS → the casts of the movies/TV shows
- VOTES → total audiences who give the rating
- RUNTIME → duration of the movies/TV shows
- GROSS → total amount earned in the worldwide

II. Initial Plan

The initial plan I'd like to explore from the dataset:

- Checking the datatypes for each column. I have shown it in Fig. 1.

```
Total rows:  9999
Column names:  ['MOVIES', 'YEAR', 'GENRE', 'RATING', 'ONE-LINE', 'STARS', 'VOTES', 'RunTime', 'Gross']
Datatype of each column:
MOVIES       object
YEAR         object
GENRE        object
RATING      float64
ONE-LINE     object
STARS        object
VOTES        object
RunTime     float64
Gross        object
dtype: object
```

Figure 1. Data Description

- Identifying missing value(s) for each column. I have shown it in Fig. 2.

```
Missing Value from the Dataset for each column:
MOVIES         0
YEAR         644
GENRE         80
RATING      1820
ONE-LINE       0
STARS          0
VOTES       1820
RunTime     2958
Gross       9539
dtype: int64
```

Figure 2. Missing Value for each Column

- Computing the descriptive statistics

III. Data Cleaning and Feature Engineering

As I identified from the initial plan in Fig. 2, “Gross” has 9539/9999 = 95.4% missing values in their rows. Thus, I move the “Gross” column into another variable to keep it safe for the future use as shown in Fig. 3

```
#Saving not null "Gross" feature to another variable
mov_gross = movies[movies["Gross"].notnull()]
mov_gross.head()
```

	MOVIES	YEAR	GENRE	RATING	ONE-LINE	STARS	VOTES	RunTime	Gross
77	The Hitman's Bodyguard	(2017)	\nAction, Comedy, Crime	6.9	\nThe world's top bodyguard gets a new client,...	\n Director:\nPatrick Hughes\n| \n Stars...	205,979	118.0	$75.47M
85	Jurassic Park	(1993)	\nAction, Adventure, Sci-Fi	8.1	\nA pragmatic paleontologist visiting an almos...	\n Director:\nSteven Spielberg\n| \n Sta...	897,444	127.0	$402.45M
95	Don't Breathe	(2016)	\nCrime, Horror, Thriller	7.1	\nHoping to walk away with a massive fortune, ...	\n Director:\nFede Alvarez\n| \n Stars:\...	237,601	88.0	$89.22M
111	The Lord of the Rings: The Fellowship of the Ring	(2001)	\nAction, Adventure, Drama	8.8	\nA meek Hobbit from the Shire and eight compa...	\n Director:\nPeter Jackson\n| \n Stars:...	1,713,028	178.0	$315.54M
125	Escape Room	(I) (2019)	\nAction, Adventure, Horror	6.4	\nSix strangers find themselves in a maze of d...	\n Director:\nAdam Robitel\n| \n Stars:\...	99,351	99.0	$57.01M

Figure 3. Checking Not Null in Dataset

Then, I drop the column from the main dataframe. The head of the dataset after dropping “Gross” is shown in Fig. 4.

	MOVIES	YEAR	GENRE	RATING	ONE-LINE	STARS	VOTES	RunTime
0	Blood Red Sky	(2021)	\nAction, Horror, Thriller	6.1	\nA woman with a mysterious illness is forced ...	\n Director:\nPeter Thorwarth\n| \n Star...	21,062	121.0
1	Masters of the Universe: Revelation	(2021–)	\nAnimation, Action, Adventure	5.0	\nThe war for Eternia begins again in what may...	\n \n Stars:\nChris Wood, \nSara...	17,870	25.0
2	The Walking Dead	(2010– 2022)	\nDrama, Horror, Thriller	8.2	\nSheriff Deputy Rick Grimes wakes up from a c...	\n \n Stars:\nAndrew Lincoln, \n...	885,805	44.0
3	Rick and Morty	(2013–)	\nAnimation, Adventure, Comedy	9.2	\nAn animated series that follows the exploits...	\n \n Stars:\nJustin Roiland, \n...	414,849	23.0
4	Army of Thieves	(2021)	\nAction, Crime, Horror	NaN	\nA prequel, set before the events of Army of ...	\n Director:\nMatthias Schweighöfer\n| \n ...	NaN	NaN

Figure 4. Head of Dataset after Dropping “Gross”

I still have problems with missing values in “YEAR”, “RATING”, “VOTES”, and “RunTime”. For ‘YEAR”, I let the missing values for further feature engineering step (as it’s in “string”) and I fill it with “Unknown” to be processed later. So, I focus on handling “RATING”, “VOTES”, and “RunTime”. I can estimate the value by mean or their minimum or maximum value. However, as we knew that movie is an unexpected thing to be estimated, thus I choose to fill the missing values “0”. Then, all of the missing values from “RATING”, “VOTES”, and “RunTime” are to be zero as seen in Fig. 5.

```
#Fill the missing value in RATING with "0"
movies[["RATING","VOTES","RunTime"]] = movies[["RATING","VOTES","RunTime"]].fillna(0)
```

```
print("Updated Missing Value from the Dataset for each column: ")
print(movies.isnull().sum())
```

```
Updated Missing Value from the Dataset for each column:
MOVIES        0
YEAR        603
GENRE         0
RATING        0
ONE-LINE      0
STARS         0
VOTES         0
RunTime       0
dtype: int64
```

```
#Fill the missing value in RATING with "0"
movies["YEAR"] = movies["YEAR"].fillna("Unknown")
```

Figure 5. Updated Missing Value in Dataset

In brief, I have no other missing values in my main dataframe as seen in Fig. 6.

```
Updated Missing Value from the Dataset for each column:
MOVIES      0
YEAR        0
GENRE       0
RATING      0
ONE-LINE    0
STARS       0
VOTES       0
RunTime     0
dtype: int64
```

Figure 6. Updated Missing Value in Dataset

Next section, I tried to check whether there's duplicated data or not. I found duplicated data as shown in Fig. 7.

```
print("Duplicated data: ")
movies.duplicated().sum()
```

```
Duplicated data:

429
```

Figure 7. Checking Duplicated Data

Then, I drop them such that it will not disturb our future analysis. This shown in Fig. 8

```
movies.drop_duplicates(inplace = True)
```

```
print("Duplicated data: ")
movies.duplicated().sum()
```

```
Duplicated data:

0
```

Figure 8. Dropped Duplicated Data

After that, we move into the feature engineering. Firstly, I look at the "GENRE" feature as shown in Fig. 9:

```
GENRE
\nAction                                  36
\nAction, Adventure                        5
\nAction, Adventure, Biography             4
\nAction, Adventure, Comedy               75
\nAction, Adventure, Crime                56
                                          ..
\nTalk-Show, Sport                        2
\nThriller                               65
\nThriller, Mystery                       1
\nWar                                     1
\nWestern                                 6
Name: MOVIES, Length: 510, dtype: int64
```

Figure 9. Checking "GENRE" column

Then I removed the "\n" from all rows and turned all rows into the list of genres as shown in Fig. 10 below:

```
0            [Action, Horror, Thriller]
1       [Animation, Action, Adventure]
2            [Drama, Horror, Thriller]
3       [Animation, Adventure, Comedy]
4              [Action, Crime, Horror]
                     ...
9485               [Drama, Thriller]
9486    [Animation, Action, Adventure]
9487            [Documentary, Sport]
9488       [Adventure, Drama, Fantasy]
9489       [Adventure, Drama, Fantasy]
Name: GENRE, Length: 9490, dtype: object
```

Figure 10. Updated Data in "GENRE" column

After that, I do one-hot encoding for the "GENRE" for future genre classification project. Thus, I have multi-class genre for each row. Also, I have removed the "\n" from "ONE-LINE" and "STARS" too for better future NLP project as shown in Fig. 11.

IG	ONE-LINE	STARS	VOTES	RunTime	Action	Adventure	Animation	...	News	Reality-TV	Romance	Sci-Fi	Short	Sport	Talk-Show	Thriller	War	Western
.1	A woman with a mysterious illness is forced in...	Director:Peter Thorwarth| Stars:Peri B...	21,062	121.0	1	0	0	...	0	0	0	0	0	0	0	1	0	0
.0	The war for Eternia begins again in what may b...	Stars:Chris Wood, Sarah Michel...	17,870	25.0	1	1	1	...	0	0	0	0	0	0	0	0	0	0
.2	Sheriff Deputy Rick Grimes wakes up from a com...	Stars:Andrew Lincoln, Norman R...	885,805	44.0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
.2	An animated series that follows the exploits o...	Stars:Justin Roiland, Chris Pa...	414,849	23.0	0	1	1	...	0	0	0	0	0	0	0	0	0	0
.0	A prequel, set before the events of Army of th...	Director:Matthias Schweighöfer| Stars:...	0	0.0	1	0	0	...	0	0	0	0	0	0	0	0	0	0

Figure 11. Updated Missing Value in Dataset

Furthermore, as I promised before, I fixed the "YEAR" column. I changed it from string into integers, and I assigned the "Unknown" value as zero (0) as seen in Fig. 12.

MOVIES	YEAR	GENRE	RATING	ONE-LINE	STARS	VOTES	RunTime
Astérix	2023	[Animation, Action, Adventure]	0.0	Add a Plot		0	0.0
The Monkey King	2023	[Animation, Action, Adventure]	0.0	An animated version of the mythical Chinese hero.	Director:Anthony Stacchi| Stars:BD Won...	0	0.0
The Mother	2022	[Action, Thriller]	0.0	Female-led action thriller.	Director:Niki Caro| Star:Jennifer Lopez	0	0.0
Hiyama Kentarô no ninshin	2022	[Comedy, Drama, Romance]	0.0	The story takes place in a world where-in rare...	Stars:Takumi Saitoh, Juri Ueno	0	0.0
Bulbul Tarang	2022	[Comedy, Drama, Romance]	0.0	A bride tries to fight against the rules once ...	Director:Shree Narayan Singh| Stars:So...	0	0.0
...	...	...	...	...	...	...	...
The Formula	0	[Crime, Drama, Sport]	0.0	Follow a Formula One racing prodigy who is for...	Director:Gerard McMurray| Stars:Robert...	0	0.0
Family Leave	0	[Comedy]	0.0	The Brenners wake up to a full family body swi...	Star:Jennifer Garner	0	0.0
Carmen Sandiego	0	[Action, Adventure, Family]	0.0	A live-action feature film based on Carmen San...	Star:Gina Rodriguez	0	0.0
The Chronicles of Narnia: The Magician's Nephew	0	[Action, Adventure, Fantasy]	0.0	The next installment of C.S. Lewis's "Chronicl...		0	0.0
The Out-Law	0	[Action, Comedy]	0.0	A straight-laced bank manager about to marry t...	Director:Tyler Spindel| Stars:Pierce B...	0	0.0

Figure 12. Updated "YEAR" column in Dataset

Finally, we have got our data cleaned and ready to analyse. We moved into Exploratory Data Analysis (EDA) for further insights from the dataset.

IV. Key Findings and Insights from Exploratory Data Analysis

For EDA, I made a plot to identify what mostly available movies according to their genres. I set sum for each column that we have separated the genres into columns and I plot it into horizontal bar plot.

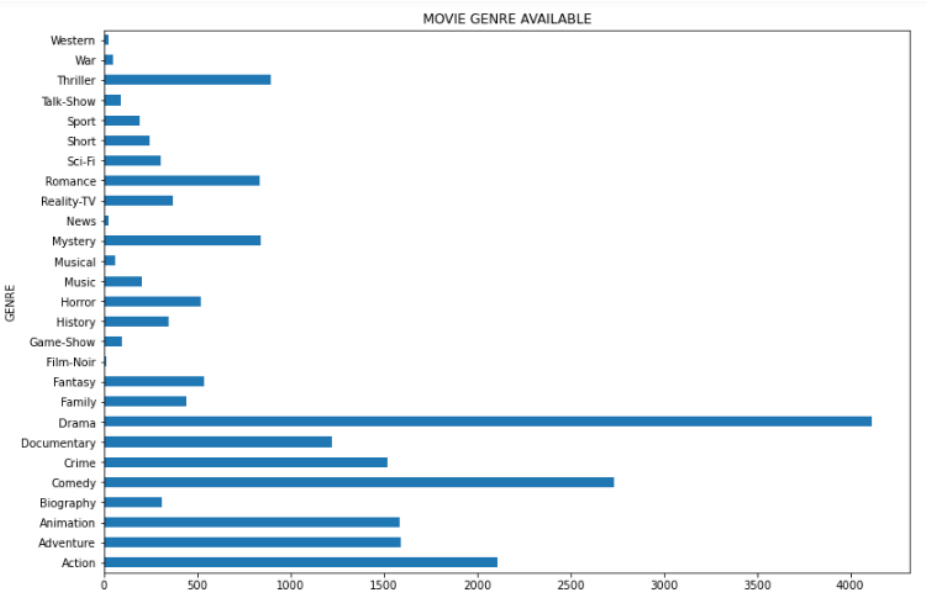

Figure 13. Horizontal Bar Plot from Sum of Genres

As we seen in Fig. 13, Film-Noir be the least movie genre available in the dataset and Drama be the most movie genre available. However, this plot does not represent movies which have multi genres. We just plot into single genre to see the pattern of each genre from the dataset.

Furthermore, I plot the average movie rating from 1990-2021 as seen in Fig. 14. As we see that the highest average movie rating occurs from 1992 movie. And as the year passed, we see

some significant up and down movie rating from 2000s, especially we have higher down rating from 2011-2021 movies

Figure 14. Line Plot of Average of Movie Rating

Finally, I compute the descriptive statistics for "RATING" and "RunTime". As we seen in Fig. 15, both of column have zero ("0") value which means they have missing values as we have discussed in previous section.

```
Descriptive Statistics of column 'RATING'

count    9490.000000
mean        5.948145
std         2.657227
min         0.000000
25%         5.500000
50%         6.800000
75%         7.600000
max         9.900000
Name: RATING, dtype: float64
```

```
Descriptive Statistics of column 'RunTime'

count    9490.000000
mean       50.781981
std        50.666680
min         0.000000
25%         0.000000
50%        43.000000
75%        87.000000
max       853.000000
Name: RunTime, dtype: float64
```

Figure 15. Descriptive Statistics of Column "RATING" AND "RunTime"

V. Hypothesis

From this dataset, I formulate 3 (three) hypotheses which can be used to check the population parameters:

- Hypothesis about comparing the mean of "RATING"
  H_0: The mean of movie rating from 2011-2021 $>=$ The mean of movie rating from 2000-2020
  H_1: The mean of movie rating from 2011-2021 $<$ The mean of movie rating from 2000-2020

- Hypothesis about identifying the mean of "RunTime"
  H_0: The mean of movie RunTime from 1900s-1990s $>=$ 120 min
  H_1: The mean of movie RunTime from 1900s-1990s $<$ 120 min

- Hypothesis about correlation between the "RATING" and "RunTime"
  H_0: "RATING" and "RUNTIME" are independent samples
  H_1: There is dependency between "RATING" and "RunTime"

VI. Hypothesis Test

From the hypotheses above, I conduct the formal hypothesis testing about comparing the mean of "RATING" according to the year of movie releases. As we know from The Central Limit Theorem, the dataset has normal distribution. Thus, I used the Z-Test to test whether the mean of movie rating from 2011-2021 is greater or equal than the mean of movie rating from 2000-2020. The result of hypothesis testing is shown in Fig. 16.

```
import pandas as pd
from scipy import stats
from statsmodels.stats import weightstats as stests
alpha = 0.05
mean_hipotesis = 7
# Default : Two Tailed
ztest , pval = stests.ztest(df.loc[(movies["YEAR"] >= 2011) & (movies["YEAR"] <= 2021)]["RATING"], movies.loc[(movies["YEAR"] >=
print("H0: The mean of movie rating from 2011-2021 >= The mean of movie rating from 2000-2010")
print("H1: The mean of movie rating from 2011-2021 < The mean of movie rating from 2000-2010")
print("The alpha is ", alpha)
print("The p-value is ", float(pval))
if pval < alpha:
    print("H0 is rejected, therefore H1 is accepted")
else:
    print("H0 is accepted")
```

```
H0: The mean of movie rating from 2011-2021 >= The mean of movie rating from 2000-2010
H1: The mean of movie rating from 2011-2021 < The mean of movie rating from 2000-2010
The alpha is  0.05
The p-value is  4.3776506128224235e-09
H0 is rejected, therefore H1 is accepted
```

Fig. 16. Hypothesis Testing for movie rating

As we seen from the result, H_1 is accepted therefore the mean of movie rating from 2011-2021 is less than the mean of movie rating from 2000-2020. We have seen the line plot of the rating in previous section. Therefore, the hypothesis testing result has strengthened our analysis about rating in year, especially the old one (2000-2010) and the new one (2011-2021).

VII. Suggestions

For further analysis, we can predict the genre as multiclass genre classification since we have separated the missed "GENRE" into another pandas dataframe. Also, we can train our "RATING" and "RunTime" in regression analysis to predict the future movie rating and runtime. For multiclass genre classification, we can use NLP concept because this dataset is more related to the text mining in NLP.

VIII. Conclusion

In brief, this "MOVIES DATASET FOR FEATURE EXTRACTION, PREDICTION" is very dirty in having missing values, duplicated data, and inconsistent datatype, especially in mixed datatype, such as number and string in a row of data. After some pre-processing, we have a cleaned and prepared dataset to be analysed in future projects in regression and classification. However, as we found before, we have set the missing values as zero ("0") values. In the future, they can be replaced by their true values for better analysis and prediction. Therefore, I hope that some additional data are required to replace that zero values can be added soon by extracting data from the available source (Netflix/IMDB).