SCRIPTING FOR DATA ANALYSIS
HOMEWORK 2_SEMI STRUCTURED DATA

IST652_WINTER_2022

Contents

OVERVIEW

The main purpose of this document to provide a report as part of Homework 2, this one primarily provides the information how the data, have been processed for analysis and the kind of analysis done by using the learnings from the class. The computation part, have been done using basic python commands with the Jupyter note book .

I have chosen movies that were released in 2020 from the movie database, by creating an own API at the website and installing the same at the python environment

The main interest for me to choosing is that it has lot of elements like profit, revenue, budget, viewership, voting done by viewers, production companies. Etc...., which gives a great scope to experiment some of the learnings.

However, I would be focusing on minimum viable and decent set (less than 250 records) to perform analysis as I have been experiencing lot of issues while compiling in my previous assignments.

Intention is to utilize the concepts of semi structured data loading, data cleaning, identifying the columns to fit for usage of analysis and provide certain descriptive statistics and if possible provide few key graphical representations

ANALYSIS

For any kind of analysis there are few mandatory steps need to be followed, in our case the prerequisite to be semi structured which can be JSON, XML, HTML,

a) Converting raw format of the source data to machine understandable form

b) Create a data file

c) Cleanup the data (removing such as blank fields and renaming few fields for our better understanding)

d) The most important aspect is to find out the useful data by removing the unnecessary data, figure out the range of outliers (data frames)

e) Compute the descriptive statistics to have various numbers of the data.

f) Figure out the measurable parameters and define various functions to identify the pattern and trends

g) Experiment various measurable parameters with python functions by the principles of scripting for data analysis

Coming to the movie data, this is related to how many movies are available for a particular chosen year ,what were the parameters (based on the available columns) like movie

budget, revenue, profit, views, voting, popularity, which production company produced movie ..etc. but not about the ratings and reviews of the movie

DATA SOURCE & DATA CLEAN UP

Here, the data is collected from website www.themoviedb.org, this website provides a unique (free) API to access the movie data collection. This API provides a API Key, that can be stored as a txt file in the same folder where we are coding for this homework to avoid an errors while accessing/ import the data.

Also, we need to install the API in the python file as below, the command is something

```
"$ pip install tmdbv3api
```

```
(base) C:\Users\praso>pip install tmdbv3api
Collecting tmdbv3api
  Downloading tmdbv3api-1.7.6-py2.py3-none-any.whl (17 kB)
Requirement already satisfied: requests in c:\users\praso\anaconda3\lib\site-packages (from tmdbv3api) (2.26.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\praso\anaconda3\lib\site-packages (from requests->tmdbv3api) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\praso\anaconda3\lib\site-packages (from requests->tmdbv3api) (3.2)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\praso\anaconda3\lib\site-packages (from requests->tmdbv3api) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\praso\anaconda3\lib\site-packages (from requests->tmdbv3api) (2021.10.8)
Installing collected packages: tmdbv3api
Successfully installed tmdbv3api-1.7.6
```

Figure 1: Reference of tmdb api installation

Also we need to create tmdb api references in the python (attached as part of the files)

I have used "beautiful soup" package to provide the needful information while collecting the data from "movie db api", additionally installed the necessary packages for any python program that would require to do the numerical calculations and data frames to pandas and proper treatment of regular expressions and all.

As faced multiple errors in computing while trying to calculate the percentages of few parameters in the data , I have split my program into two parts

1) Graphical analysis of the data

2) To provide descriptive statistics and certain measures while analyzing data

Below are the key questions that I am targeting to have a meaningful results for the information that has been opted for analyzing

Steps followed to write the JSON data:

1) Imported the required libraries, packages

2) Created a separate api text file ad api data file to import to python program

3) We have selected the range from “1990 to 2021” reasoning being too much data will complicate the analysis with lot of if else and other functions and conditions. Listing data as arrays for better machine understandable input form

4) Opted to write the data as csv file (We’ve tried to get the data to Mongo db and retrieve the data from there db connection, it was a successful failure, one of the main blocker for delay in assignment submission)

5) Below are the key parameters captured for the analysis snapshot for the same for quick reference.

```
In [27]: #Getting summary statistics for our df
         movies.describe()
```

Out[27]:

	budget	revenue	profit	popularity	vote_average	vote_count	month	day	year
count	1.740000e+02	1.740000e+02	1.740000e+02	174.000000	174.000000	174.000000	174.000000	174.000000	174.000000
mean	4.661302e+07	1.296438e+08	8.303082e+07	45.985408	6.313218	2736.844828	6.919540	15.114943	2010.040230
std	5.261059e+07	1.926056e+08	1.537114e+08	44.628165	0.878713	3948.127812	3.476489	8.585409	0.197065
min	1.500000e+03	4.549000e+03	-1.110072e+08	1.243000	3.600000	1.000000	1.000000	1.000000	2010.000000
25%	1.000000e+07	1.451735e+07	-1.252039e+06	16.456500	5.900000	543.000000	4.000000	9.000000	2010.000000
50%	2.500000e+07	5.803693e+07	2.841555e+07	28.996500	6.300000	1466.000000	7.000000	15.000000	2010.000000
75%	6.375000e+07	1.563374e+08	9.612743e+07	60.867500	6.800000	3206.500000	10.000000	22.000000	2010.000000
max	2.600000e+08	1.066970e+09	8.669697e+08	225.462000	10.000000	31119.000000	12.000000	31.000000	2011.000000

Figure 2: Key Parameters Descriptive stats

6) We have explored few options for the better fitment to get movie ids, released year/date, movie name ..etc.

- Removed the columns with blank data
- Added a row as heading
- Over all data was 276 rows and ten columns
- The count have been brought down to “174 rows and 10 columns” as the useful data values
- Over all attributes that considered for the analysis are as below

```
try:
    title = jdata['title']
    budget = jdata['budget']
    genres = jdata['genres']
    production_companies = jdata['production_companies
    release_date = jdata['release_date']
    revenue = jdata['revenue']
    profit = revenue - budget
    popularity = jdata['popularity']
    vote_average = jdata['vote_average']
    vote_count = jdata['vote_count']
except KeyError:
    title = 'NA'
    budget = 'NA'
    genres = 'NA'
    production_companies = 'NA'
    release_date = 'NA'
    revenue = 'NA'
    profit = 'NA'
    popularity = 'NA'
    vote_average = 'NA'
    vote_count = 'NA'

movie_data = {
    'release_date': release_date,
    'title': title,
    'budget': budget,
    'genres': genres,
    'production_companies': production_companies,
    'revenue': revenue,
    'profit': profit,
    'popularity': popularity,
    'vote_average': vote_average,
    'vote_count': vote_count
```

Figure 3: Key Parameters

The main analysis part is that to get the aggregation of certain measurements and discretize (to make the series of values in the descriptive statistical analysis)

1) Budget, Revenue, Profit, Popularity, Vote Average, Vote Count, all the measures and math are clearly articulated with outputs in the python file.

2)As part of the typical analysis one shall categorize the data with certain boundaries or ranges here we are measuring as following

For Positive Math : {extremely_low' < 'low' < 'high' < 'extremely_high'}

3) The above will not be the case for the varchar and or float type data such as days of the week

```
Name: day, dtype: float64
```

```
#We are setting new categories for the day column by creating a new column for week
'''week_1 is the first 7 days of the month, week_2 is days 8 - 14, week_3 is days 15 - 21, and week_4 are the
rest of the days'''
categories = ["week_1", "week_2", "week_3", "week_4"]

movies_discretized_df["week"] = pd.cut(movies_discretized_df["day"], [0, 8, 15, 22, 32], labels = categories)
```

By doing all such and applying the descriptive and inferential statistics as the fundamental measures, we have t ried to get the results to certain questions.

Questions :

1) How are the amounts of percent_profits distributed across budget levels?
Below is the snippet from code file for quick reference as it is the base of the analysis

In [53]:
```
#Question 1:
    #How are the amounts of percent_profits distributed across budget levels?

'''We want to compare the budget category percentage make up for each percent_profit level. To do this we need to
get the count for each budget level, the count for each percent_profit level by budget level and then divide
the count of the percent_profit/count of budget level and multiply by 100. We have to do this for each
budget level and level of percent_profits. We think that we could potentially answer this question by group bys.'''
movies_discretized_count = movies_discretized_df.groupby(["budget", "percent_profit"])["budget"].count()
'''Taking the output from the line above and converting it to a data frame. We are using pandas, which we important as pd.
First, we call the package we are using then the function from that package and then what we want to run the function on.
pd.function(item to use). We are using the DataFrame function from the pandas package on the series created by our group by'''
movies_discretized_count_df = pd.DataFrame(movies_discretized_count)
#Checking to see what our df looks like.
movies_discretized_count_df
#Changing the column name from budget to counts
movies_discretized_count_df.columns = ["counts"]
#Checking to see what our df looks like.
movies_discretized_count_df
```

Out[53]:

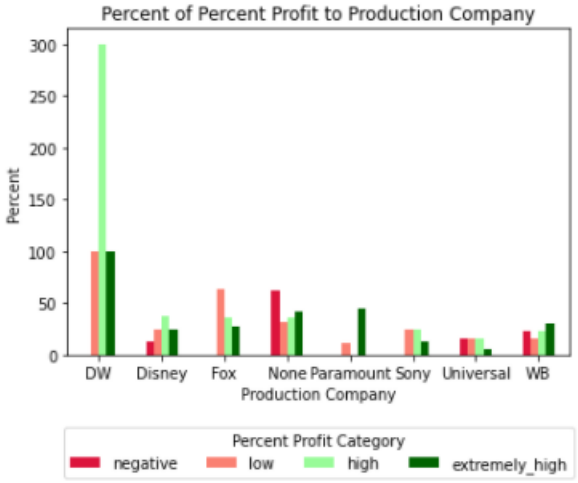
		counts
budget	percent_profit	
extremely_low	negative	30
	low	6
	high	4
	extremely_high	18
low	negative	8
	low	6
	high	9
	extremely_high	7
high	negative	6
	low	13
	high	13
	extremely_high	9
extremely_high	negative	4
	low	14
	high	16
	extremely_high	10

Like, wise the following questions, are addressed with the analysis

2) Do big name production companies impact the percent profit?

Yes, it has the impact in both positive and negative ways for all the production companies, below is the graph f or the same.

Attaching only few snippets and graphs for a quick reference in the document to provide an outline of the analysis, rest of the results are available in the code file.



Graphical Analysis of the Original Raw Data

3) Do "Good" Movies Make Money? -- We're defining "Good" as vote average?

Yes, this does happen however the range is favorable for most viable budget movies, however the bigger budget has bigger outlier, though its good it did not make much money in par with budget however it was not a loss venture though

4) Does Popularity = Profit?

No, this is true popularity cannot be turned out materialistic profits for all the production houses that are produced movies. Budget is the key there.

5) How does budget impact vote average?

This has a positive impact for the minimum viable budget , however the case with bigger budget movies are at the edge with no loss

6) How does budget impact popularity?

This question could not answer very well was not able to conclude it requires certain more data like how much viewership was attain at the initial days of release and how long it ran locally and is it telecasted in theaters for long or telecasted in other platforms and re- releases…etc,.

7) Is there a relationship between "Above Average Movies" and Budget/Price?

Yes, certainly this is most profitable venture for most of the production companies

Conclusions

All the questions are, answered with most useful insights toward future productions however the higher the budget the scale of releases across the world needs to consider for the profit analysis.

Note : In this homework assignment our concentration was only with location US and English language, for better and more accurate results for the profits needs to analyze the data in how many languages the movie was released along with English and across the world especially more crowd pulling countries and the vote average in that countries

Lessons Learnt during this process of analysis

1) Data type needs to be understood properly and convert them according to the output and math that is getting applied

2) Data reading to csv I have done mistakes like not splitting them to correctly beforehand even though tried to form an array couple of times I got the CSV file as empty

3) I was under the impression that tmdb api installation at the local machine in the python directory path would suffice while connect and load the data, however it was a wrong assumption. It needs a one liner text file with api key inside the file and call that file in the program to execute.

4) Like any other time for the graphs even though the packages are installed, typing mistakes made me rewrite few times.

5) Even though numpy as installed in the packages we have the following error while doing the divide operations for few aggregate measurements.

“TypeError: Object with dtype category cannot perform the numpy op true_divide”.

References

1) https://pypi.org/project/tmdbv3api/

2) https://notebook.community/M0nica/datalogues/content/posts/tmdb-api

3) https://www.themoviedb.org/settings/api

4) https://blog.jovian.ai/web-scraping-popular-movies-using-beautifulsoup-5bab0852fee4

5) https://www.crummy.com/software/BeautifulSoup/bs4/doc/

6) https://numpy.org/doc/stable/reference/generated/numpy.true_divide.html

7)https://www.theatlantic.com/sponsored/ibm-transformation-of-business/big-data-and-hollywood-a-love-story/277/

8) https://trace.tennessee.edu/cgi/viewcontent.cgi?article=3282&context=utk_chanhonoproj

All the Asynch materials and exercises