**TMDB Movie Python Questions**

**Intro:**

TMDB is a production company that wants to analyze a movie dataset to identify what kinds of movies perform well in cinemas, which genres they belong to, and so on. It will help the company predict if a movie will be a commercial success, if the movie will be highly rated, etc.

Preliminary analysis:

1. **Load the movie dataset in the Python Notebook. Display the numbers of rows and columns in the dataset. Display the titles and genres of the first 50 movies from the dataset.**

I imported the data using pd.read\_csv

I used a shape function to get the total size of the dataset. It contained 4803 rows and 20 columns

I also used a for loop to print out the names of all the columns. An alternative way just to print data.columns to get an array of all the column names.

I filtered the data to print only the first 50 rows using slicing. i.e. [:50]

1. **Identify the columns that have null values and perform the null value treatment. (Choose the imputation method based on the type of data in the columns of interest.)**

I used the isnull().sum() to add up any null values in each column. 5 columns contained null values. The columns were ‘homepage with 3091 null values’, ‘overview with 3 null values’, ‘release\_date, with 1 null value’, ‘runtime with two null values’, and ‘tagline with 844 null values’.

In order to get the exact indices and location of where the null values were for the overview, release\_date and runtime columns, I used isnull()].index.tolist()) and it returns a list. After that I searched for those indices using the .loc function. I **went online and searched for the appropriate information and used the .fillna() function to insert this info**. For the homepage column which had thousands of null values. I filled it in with a generic website. I did a similar thing for the tagline column. I then ran the isnull().sum() function again to make sure there were no more null values.

Descriptive Analysis:

1. **Display the movie categories that have a budget greater than $220,000.**

Filtered the data to search the budget column where it was over 220,000. Pirates of the Caribbean and Avengers had the highest budgets

1. **Display the movie categories where the revenue is greater than $961,000,000.**

Filtered the data to search the budget column where it was over $961,000,000. Avatar and the Titanic had the highest revenues.

1. **In the dataset, there are some movies for which the budget and revenue columns have the value 0, which means unknown values. Remove the rows with value O from both the budget and revenue columns.**

I filtered the data for the values that did not contain the value zero and set it as the new data frame. By doing this, it drops all the rows that have a value 0.

1. **List the top 10 movies with the highest revenues and the top 10 movies with the least budget.**

I used the sort\_values() function and sorted by the revenue column in descending order. Avatar and Titanic have the highest revenue.

I used the sort\_values() function and sorted by the budget column in ascending order. Modern Times and a Farewell to Arms had the least budget but this information seemed to be inaccurate considering movies can’t have a budget of 1 or 4$. IM not sure if they meant 1 million and 4 million.

1. **How are the popularities of movies related to the movie budgets? Are they correlated or totally uncorrelated with each other? Write the interpretation of your analysis**.

After creating a subset of the data, I then created a scatter plot with both seaborn and matplotlib. After adding the trendline, you can notice that there is a positive correlation with the budget and movie popularity so the higher the budget , then the more popular the movie was. There were also a lot of outliers in the data.I also did some further analysis and checked to see if there was a correlation between budget and voting average and there actually was a slight negative correlation.

1. **Identify and display the names of all production companies along with the number of times they appear in the dataset.**

I used the str splitter function to separate the JSON formatted columns. The columns I separated were the production and genre columns, After applying this, I used the value counts function to count up the number of times they appeared in the columns. Paramount Pictures had the highest count and appeared in the Production column 48 times. Drama had the highest count with a total of 204 times in the Genres column.

1. **Display the names of the top 25 production companies based on the number of movies they have produced in descending order of the number of movies produced.**

To get the top 25 of the production companies, I used slicing again so I showed the top 25 for the production company with -> [:25]

I got the top 10 of the genre column by doing -> [:10]

1. **Sort the data in descending order based on revenue and filter the top 500 movies. Find the measures of central tendency for the following columns using the filtered data: Budget.Revenue.Runtime**

I used the sort values function in descending order as well as slicing[:500] ,to get the top 500.I sorted the data by the revenue column.

I then exported this new subset of data to csv format by doing to\_csv and naming it subset\_data. I opened a new python file and named it central tendencies.

Once I imported this new data with only 500 columns, I then filtered the data for it to just show me the budget, revenue and runtime columns. I also set the title as the index since I was only going to need columns with integers.

The means of central tendency are the mean, median, mode , range, standard deviation, coefficient of variation, 1st and 3rd quartile, inter quartile range, outliers, upper and lower fence.

To find all this, I created an empty list for every measure I was going to find and I used a for loop so it could iterate through every value in each of the 3 columns. I then appended the proper information to each of the lists, and then added converted each list to a combined data frame.

1. **Perform outlier analysis for the above three columns using box plots.**

Using matplotlib, I created one figure which would include a box plot for each of the 3 columns. I converted each column to a list by taking each column and using .tolist() function.

I then added the appropriate axes and created labels for the columns. After displaying the boxplots, the runtime didn’t show much information since all movies tend to be around the same time from 1 hour to almost 3 hours long. The revenue column showed a lot of outliers and the budget column only showed two outliers.

1. **Identify and display the names of the movies along with their run times for those movies that have above average runtime, using the data from the previous task**

Since I already had the average for the runtime column, I simply filtered the data to give me all the rows where the runtime was greater than 118.72(average). I then simply displayed this new data and all the runtime lengths were above the average. There were 234 rows out of the 500 that had an above average movie runtime.

Some questions of my interest were:

1. **Showing all the columns**

I used a for loop to iterate through the column names and print each one on a separate line

I also used a print statement to print al the column names in array format

I used data.columns

1. **Decoding the genre JSON column and** Extracting Info from the column and putting it into a new column

This took a lot of messing around with and I found it pretty interesting since it was new to me. I began by pulling the individual column and displaying it. In this case it was the genre column and of data type ‘object’. I then converted this to a list and gave it a variable name. I also created a new empty list. I started off by making a for loop that would iterate through each cell in the genre list. Inside the for loop, I imported a pd.read\_json function and inserted the iterable value inside the parameter. This function converts each value to json readable format. Since there were different amounts of genres in each cell, I had to use slicing but start backwards ,in order to be able to append every value of the cell to the empty list. After that was done, I simply appended this list as a new column in the existing data frame and named it ‘col’.

1. Movies titles and genres with the highest popularity score(see graphs below)
2. Movies titles and genres with the highest revenue(see graphs below)
3. Movies titles and genres with the highest voting average(see graphs below)

**A screen shot of a graph

Description automatically generated**

**A graph of a vote graph

Description automatically generated**

A graph of a number of movies

Description automatically generated

A diagram of a graph

Description automatically generated