**Analyzing Movie Ratings with Python**

PAVAN KUMAR / 27TH FEB,2024

We watch movies primarily for entertainment. Some viewers don't bother to rate them, while others consistently rate every movie they watch. These dedicated viewers contribute to rating movies for those who rely on reviews before choosing what to watch. If you're new to data science and eager to learn how to analyze movie ratings using Python, this article is tailored for you. I'll guide you through the process of analyzing movie ratings in Python.

**Analyzing Movie Ratings with Python**

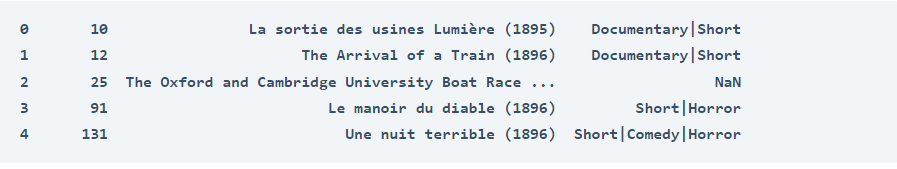
Examining viewer ratings for a movie assists numerous individuals in determining whether to watch it. Thus, for the Movie Rating Analysis assignment, acquiring a dataset containing data on the ratings provided by each viewer is essential. For this purpose, I've acquired a dataset from Kaggle comprising two files:

- One file comprises data about the movie ID, title, and genre.

- The other file contains user ID, movie ID, user ratings, and timestamps of the ratings.

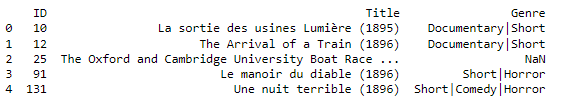
Let's commence the movie rating analysis task by importing the required Python libraries and datasets.

1. import numpy as np
2. import pandas as pd
3. movies = pd.read\_csv("movies.dat", delimiter='::')
4. print(movies.head())



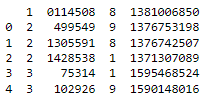
In the provided code snippet, I've solely imported the movies dataset without any associated column names. Now, let's establish the column names for clarity.

1. movies.columns = ["ID", "Title", "Genre"]
2. print(movies.head())



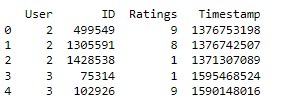
Next, we'll import the ratings dataset.

1. ratings = pd.read\_csv("ratings.dat", delimiter='::')
2. print(ratings.head())



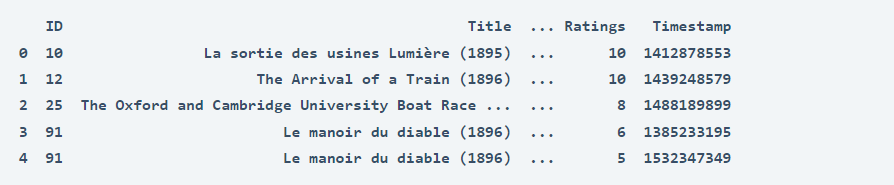
The rating dataset similarly lacks column names, so let's establish the column names for this data as well.

1. ratings.columns = ["User", "ID", "Ratings", "Timestamp"]
2. print(ratings.head())



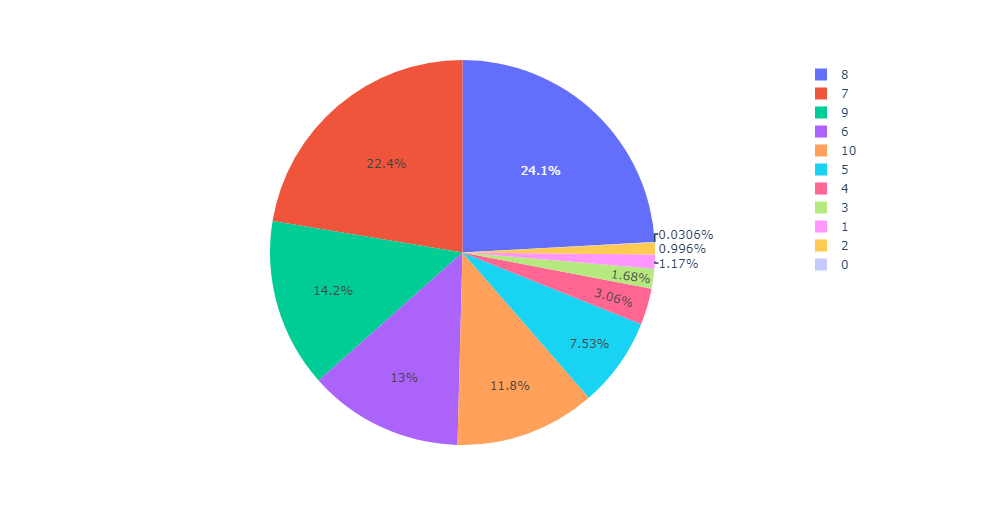
I'm about to merge these two datasets into one. They both share a common column labeled as ID, which contains movie IDs. Therefore, we can utilize this column as the shared identifier for merging the two datasets.

1. data = pd.merge(movies, ratings, on=["ID", "ID"])
2. print(data.head())



Since this task is at a beginner level, I'll begin by examining the distribution of ratings provided by viewers for all the movies.

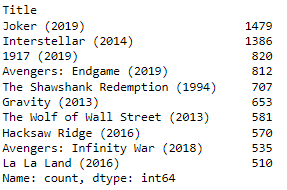
1. ratings = data["Ratings"].value\_counts()
2. numbers = ratings.index
3. quantity = ratings.values
4. import plotly.express as px
5. fig = px.pie(data, values=quantity, names=numbers)
6. fig.show()



According to the pie chart depicted above, the majority of movies receive an 8 rating from users. This suggests that most movies are positively rated based on the provided data.

Since 10 represents the highest rating given by viewers, let's explore the top 10 movies that have received a perfect 10 rating.

1. data2 = data.query("Ratings == 10")
2. print(data2["Title"].value\_counts().head(10))



According to this dataset, Joker (2019) received the most 10 ratings from viewers. This demonstrates a basic approach to analyzing movie ratings using Python for beginners in data science..

**Summary**

This demonstrates how one can conduct movie rating analysis as a novice in data science using the Python programming language. Evaluating viewer ratings of a movie assists many individuals in determining whether they should watch it or not.