



prime video

Analysis By
Rizwan Shah



www.linkedin.com/in/shahrizwan819



rizwansh5211@gmail.com



https://github.com/ShahRizwan007/Supermarket_sales_python_project

amazon_prime_data

September 25, 2024

```
[1]: import opendatasets as od
```

```
[17]: link=r'https://www.kaggle.com/datasets/shivamb/amazon-prime-movies-and-tv-shows/'  
      ↪amazon_prime_titles.csv'  
      od.download(link)
```

Please provide your Kaggle credentials to download this dataset. Learn more:

<http://bit.ly/kaggle-creds>

Your Kaggle username:

shahrizwan52

Your Kaggle Key:

.....

Dataset URL: <https://www.kaggle.com/datasets/shivamb/amazon-prime-movies-and-tv-shows>

Downloading amazon-prime-movies-and-tv-shows.zip to .\amazon-prime-movies-and-tv-shows

100%|

| 1.61M/1.61M [00:01<00:00, 1.01MB/s]

```
[1]: import pandas as pd  
      path=r'C:\Users\Rizwan\Downloads\Shah Rizwan\Python data analyst\  
      ↪project\Streaming_app\amazon-prime-movies-and-tv-shows\amazon_prime_titles.  
      ↪csv'  
      df=pd.read_csv(path)  
      #df = pd.DataFrame() # This will reset the DataFrame to an empty one
```

```
[2]: df.head()
```

```
[2]:  show_id  type  title  director \  
0      s1  Movie  The Grand Seduction  Don McKellar  
1      s2  Movie  Take Care Good Night  Girish Joshi  
2      s3  Movie  Secrets of Deception  Josh Webber  
3      s4  Movie  Pink: Staying True  Sonia Anderson
```

```
4      s5  Movie      Monster Maker      Giles Foster
```

```

                                cast      country \
0      Brendan Gleeson, Taylor Kitsch, Gordon Pinsent      Canada
1      Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar      India
2      Tom Sizemore, Lorenzo Lamas, Robert LaSardo, R...  United States
3      Interviews with: Pink, Adele, Beyoncé, Britney...  United States
4      Harry Dean Stanton, Kieran O'Brien, George Cos...  United Kingdom

```

```

      date_added  release_year  rating  duration      listed_in \
0  March 30, 2021      2014      NaN  113 min      Comedy, Drama
1  March 30, 2021      2018     13+  110 min      Drama, International
2  March 30, 2021      2017      NaN   74 min  Action, Drama, Suspense
3  March 30, 2021      2014      NaN   69 min      Documentary
4  March 30, 2021      1989      NaN   45 min      Drama, Fantasy

```

```

                                description
0  A small fishing village must procure a local d...
1  A Metro Family decides to fight a Cyber Crimin...
2  After a man discovers his wife is cheating on ...
3  Pink breaks the mold once again, bringing her ...
4  Teenage Matt Banting wants to work with a famo...

```

```
[3]: df.dtypes
```

```

[3]: show_id      object
      type        object
      title       object
      director    object
      cast        object
      country     object
      date_added  object
      release_year int64
      rating      object
      duration    object
      listed_in   object
      description  object
      dtype: object

```

```
[5]: max(df['description'].dropna().str.len())
```

```
[5]: 1099
```

```
[5]: from sqlalchemy import create_engine
```

```

engine=create_engine('postgresql+psycpg2://postgres:Rizwanpostsq1@localhost:
↳5432/amazon_data')

```

```
conn=engine.connect()

df.to_sql('amazon_raw',con=conn,index=False,if_exists='append')
conn.close()
```

[]:

```
create database amazon_data
create table amazon_raw(
show_id  varchar(10),
"type"   varchar(10),
title    varchar(120),
director varchar(1100),
"cast"   varchar(1150),
country  varchar(70),
date_added date,
release_year int,
rating   varchar(10),
duration varchar(10),
listed_in varchar(85),
description varchar(1200)
);
```

```
select * from amazon_raw;
```

```
select column_name,data_type,is_nullable
from information_schema.columns
where table_name = 'amazon_raw'
```

```
--checking duplicates for show_id
```

```
select show_id,count(*)
from amazon_raw
group by show_id
having count(*) >1;
```

```
alter table amazon_raw
add constraint pk_show_id primary key (show_id);
```



```

select distinct(type) from amazon_raw
--creating director table
create table amazon_director as
select show_id,trim(unnest(string_to_array(director','',''))) as director
from amazon_raw

--select * from amazon_director
--creating cast table
create table amazon_cast as
select show_id,trim(unnest(string_to_array(ar.cast','',''))) as "cast"
from amazon_raw as ar

select * from amazon_cast
--creating country table
create table amazon_country as
select show_id,trim(unnest(string_to_array(country','',''))) as country
from amazon_raw

select * from amazon_country

--creating genre table
create table amazon_genre as
select show_id,trim(unnest(string_to_array(listed_in','',''))) as genre
from amazon_raw

select * from amazon_genre

select distinct(regex_replace(duration,'[0-9]','','g')) as duration
from amazon_raw
--[" Season"," min"," Seasons"]

```

```
select distinct(rating) from amazon_raw
select * from amazon_raw;
--checking duplicate value for title column
select upper(title),upper(type)
from amazon_raw as ar
group by upper(title),upper(type)
having count(*) > 1
order by upper(title);
```

```
select count(*) from amazon_raw where
upper(title) in (
select upper(title)
from amazon_raw as ar
group by upper(title)
having count(*)>1)
```

```
select show_id,upper(title),count(*)
from amazon_raw as ar
group by upper(title),show_id
having count(*)>1
order by upper(title);
```

```
select count(*) from amazon_raw
where concat(UPPER(title),upper(type)) in (
SELECT concat(UPPER(title),upper(type))
FROM amazon_raw AS ar
GROUP BY concat(UPPER(title),upper(type))
HAVING COUNT(*) > 1)
ORDER BY UPPER(title);
```

```

with cte as (
select
show_id,type,title,date_added,release_year,
rating,cast(REGEXP_REPLACE(duration,'[a-zA-Z]', '', 'g') as int ) as duration,description
from amazon_raw)
,cte2 as (
select *,
ROW_NUMBER() OVER (Partition by Upper(title),type order by show_id) as rn
from cte

create table amazon_cleaned as (
with cte as (
select *,
ROW_NUMBER() OVER (Partition by upper(regex_replace(title,'^0-9a-zA-Z','', 'g')),type order by show_id) as rn
from amazon_raw)
select show_id,type,regex_replace(cte.title,'^0-9a-zA-Z','', 'g') as title,
case when date_added is not null then cast(date_added as date)
else null end as date_added
,release_year,rating,cast(REGEXP_REPLACE(duration,'[a-zA-Z]', '', 'g') as int ) as duration,description
from cte
where rn=1
order by show_id);

select * from amazon_cleaned
where concat(UPPER(title),upper(type)) in (
SELECT concat(UPPER(title),upper(type))
FROM amazon_cleaned AS ar
GROUP BY concat(UPPER(title),upper(type))
HAVING COUNT(*) > 1)
ORDER BY UPPER(title);

```



```

SELECT upper(title)
FROM amazon_cleaned
GROUP BY title
HAVING COUNT(*) > 1;
--addind values into date_added column
update amazon_cleaned
set date_added=case
    when release_year='2021' then '2021-01-01'::date
    else '2020-01-01'::date
end
where date_added is null

select * from amazon_cleaned where show_id is null

delete from amazon_cleaned where show_id is null

--adding value for missing country data
insert into amazon_country
select ar.show_id,dr_cr.country
from amazon_raw as ar
join
(
select ad.director,ac.country
from amazon_director ad
join amazon_country ac
on ad.show_id=ac.show_id
group by ad.director,ac.country
order by ad.director) as dr_cr on ar.director=dr_cr.director
where ar.country is null

```

```
--1. find all unique show types from the amazon_raw table?
select distinct(type) from amazon_cleaned

--2. Write a query to identify shows that have the same title but are of different types (e.g., movie vs. series).

select * from amazon_cleaned

select * from amazon_cleaned
where upper(title) in(
select upper(title)
from amazon_cleaned
group by upper(title)
having count(*)>1)
order by upper(title)

--3. list all shows that have been added to Amazon in the year 2021?
select * from amazon_cleaned

with cte as(
select show_id,title,type, extract(year from date_added) as release_year
from amazon_cleaned)
select count(*) from cte
where cte.release_year=2021
```

--Q.Which genres are most popular based on the number of shows/movies listed in each category on Amazon?

--select * from amazon_cleaned;

--select * from amazon_genre;

```
select ag.genre,count(ac.show_id) as total_show
from amazon_cleaned as ac
join amazon_genre as ag
on ac.show_id=ag.show_id
group by ag.genre
order by total_show desc;
```

--Q1.How many shows are there in total on Amazon across all types (movies, series, etc.)?

```
select distinct(type) from amazon_cleaned
```

```
select
sum(case when type='Movie' then 1 end) as Total_Movies,
sum(case when type='TV Show' then 1 end) as Total_TV_show
```

```
from amazon_cleaned as ac
```

--Q2.What is the distribution of shows by country? Which countries have produced the most content?

--select * from amazon_cleaned

```
select ac.country,
sum(case when type='Movie' then 1 end) as Total_Movies,
sum(case when type='TV Show' then 1 end) as Total_TV_show
from amazon_cleaned as a
join amazon_country as ac
on a.show_id=ac.show_id
group by ac.country
having sum(case when type='Movie' then 1 end) > 0
and sum(case when type='TV Show' then 1 end) > 0
order by sum(case when type='Movie' then 1 end) desc,
sum(case when type='TV Show' then 1 end)
```

--part2.

```
limit 1
```

--Q2.What is the distribution of shows by country? Which countries have produced the most content?

```
--select * from amazon_cleaned
select ac.country,
sum(case when type='Movie' then 1 end) as Total_Movies,
sum(case when type='TV Show' then 1 end) as Total_TV_show
from amazon_cleaned as a
join amazon_country as ac
on a.show_id=ac.show_id
group by ac.country
having sum(case when type='Movie' then 1 end) > 0
and sum(case when type='TV Show' then 1 end) > 0
order by sum(case when type='Movie' then 1 end) desc,
sum(case when type='TV Show' then 1 end)
--part2.
limit 1
```

```
select count(distinct a.show_id) as total_cleaned,
       count(distinct ac.show_id) as total_with_country
from amazon_cleaned a
left join amazon_country ac
on a.show_id = ac.show_id;
--Q3.What are the most common genres available on Amazon?
--select distinct(genre) from amazon_genre
select genre,count(ac.show_id) as Total_show
from amazon_genre as ag
join amazon_cleaned as ac
on ag.show_id=ac.show_id
group by genre
order by Total_show desc
```

```

--Q4.How many shows have been added to the platform in the last year?
--select * from amazon_cleaned

with cte as(
select show_id,type,
case when date_added is not null then extract(year from date_added)
else case when date_added is null and release_year=2021 then 2021
else 2020 end
end as added_year
from amazon_cleaned
)
select added_year,sum(case when type='Movie' then 1 end) as Total_Movies,
sum(case when type='TV Show' then 1 end) as Total_TV_show,count(show_id) as total_shows
from cte
group by added_year
order by added_year

--Q5.What are the most popular ratings for the shows on Amazon?
--select * from amazon_cleaned
select rating,count(ac.show_id) as no_of_show
from amazon_cleaned as ac
group by rating
order by no_of_show desc

--Q6.How many shows or movies were released in each year?
--select * from amazon_cleaned
select release_year,count(show_id) as total_shows
from amazon_cleaned
group by release_year
order by release_year

```


--Q7. Which directors have contributed to the most shows on Amazon, and in what genres?

--select * from amazon_cleaned

```
select director, count(ac.show_id) as total_shows
from amazon_cleaned as ac
join amazon_director as ad
on ac.show_id=ad.show_id
group by director
order by total_shows desc
```

```
select director, count(ac.show_id) as total_shows,
string_agg(distinct ag.genre, ',') as genres
from amazon_cleaned as ac
join amazon_director as ad
on ac.show_id=ad.show_id
join amazon_genre as ag
on ac.show_id=ag.show_id
group by director
order by total_shows desc
```

--Q8. How does the distribution of content by genre vary across different countries?

--select count(distinct(genre)) from amazon_genre

--select distinct(country) from amazon_country

```
select country, genre, count(a.show_id)
from amazon_cleaned as a
join amazon_genre as ag
on a.show_id=ag.show_id
join amazon_country as ac
on a.show_id=ac.show_id
group by country, genre
order by country
```

```

--Q9. What is the average duration of a show by genre, and which genres tend to have the longest/shortest shows?
select genre,round(avg(duration),0) as average_duration_in_mins
from amazon_cleaned as ac
join amazon_genre as ag
on ac.show_id=ag.show_id
group by genre
order by average_duration_in_mins desc

--Q10. Which actors appear most frequently across multiple shows or movies, and in what types of content?
--select * from amazon_cast
select ac.cast,ag.genre,count(a.show_id) as total_shows
from amazon_cleaned as a
join amazon_cast as ac
on a.show_id=ac.show_id
join amazon_genre as ag
on a.show_id=ag.show_id
group by ac.cast,ag.genre
order by total_shows desc,ac.cast;

--Q11. Which countries have the most shows of a particular genre (e.g., Drama, Comedy, etc.)?
with cte as (
select country,genre,count(a.show_id) as total_show
from amazon_cleaned as a
join amazon_country as ac
on a.show_id=ac.show_id
join amazon_genre as ag
on ag.show_id=ac.show_id
group by country,genre
),cte2 as (
select *,
Row_number() over (partition by country order by total_show desc) as rn
from cte )
select * from cte2 where rn=1;

```

--Q11. Which countries have the most shows of a particular genre (e.g., Drama, Comedy, etc.)?

```
with cte as (  
select country,genre,count(a.show_id) as total_show  
from amazon_cleaned as a  
join amazon_country as ac  
on a.show_id=ac.show_id  
join amazon_genre as ag  
on ag.show_id=ac.show_id  
group by country,genre  
,cte2 as (  
select *,  
Row_number() over (partition by country order by total_show desc) as rn  
from cte )  
select * from cte2 where rn=1;
```

```
with cte as (  
select country,genre,count(a.show_id) as total_show  
from amazon_cleaned as a  
join amazon_country as ac  
on a.show_id=ac.show_id  
join amazon_genre as ag  
on ag.show_id=ac.show_id  
group by country,genre  
,cte2 as (  
select *,  
Row_number() over (partition by genre order by total_show desc) as rn  
from cte )  
select * from cte2 where rn=1;
```

```
--Q12. Which countries have shown an upward trend in content production over the years,  
--and how can this be leveraged for marketing or partnerships?  
select * from amazon_cleaned as a  
  
with cte as (  
select country,release_year,count(a.show_id) as total_shows,  
lag(count(a.show_id)) over (partition by country order by release_year ) as previous_year  
from amazon_cleaned as a  
join amazon_country as ac  
on a.show_id=ac.show_id  
group by country,release_year  
--order by country  
,cte2 as (  
select *  
from cte  
where previous_year is not null  
and total_shows > previous_year  
)  
select country,release_year,cte2.total_shows  
from cte2  
order by cte2.total_shows desc,country,release_year;
```



Total Shows
10K

average duration
74
mins

ADDED YEAR

01-01-2020

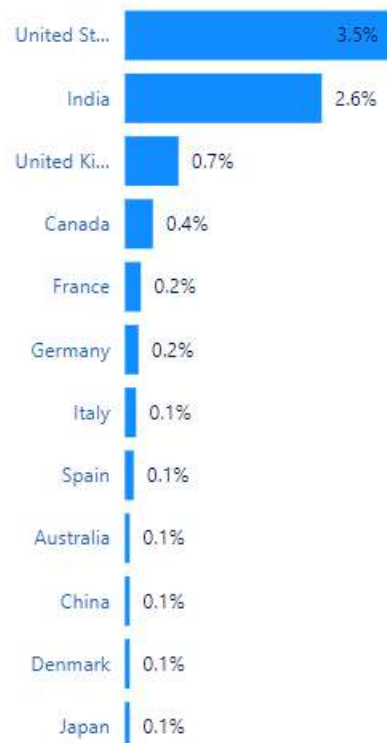
10-10-2021

TYPE

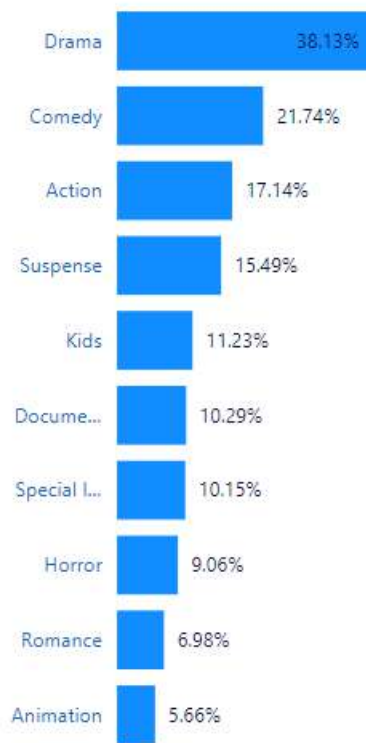
Movie

TV Show

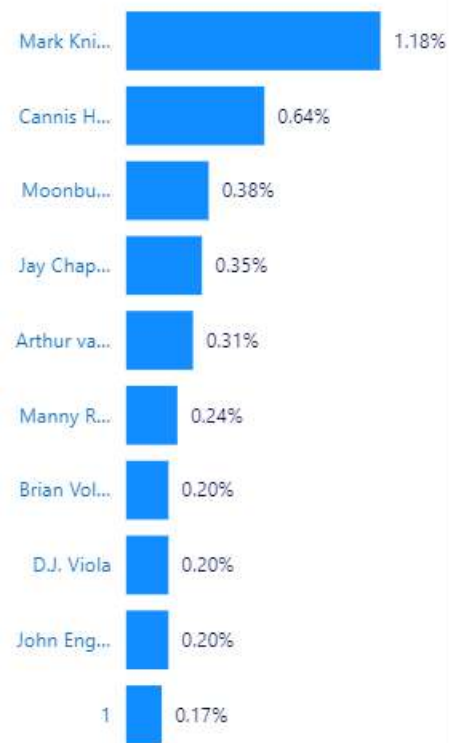
Top Contributing Country



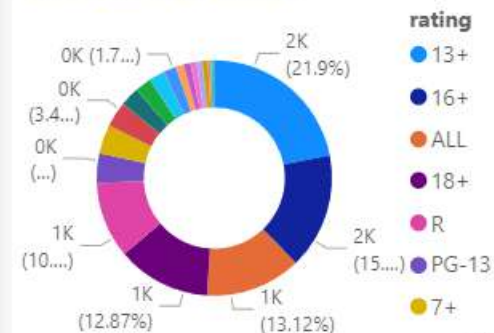
Top Genre



Top Contributing Director



Contribution by Rating



Contribution By Genre

