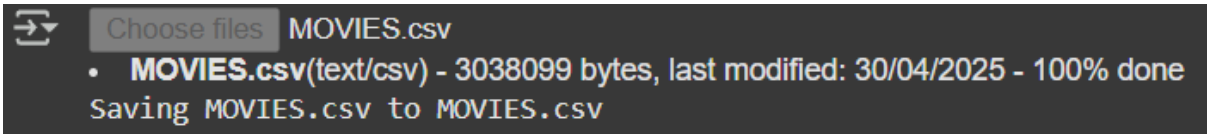
1.uploading files:

```
from google.colab import files

uploaded = files.upload()
```



2. Load and Display Basic Info:

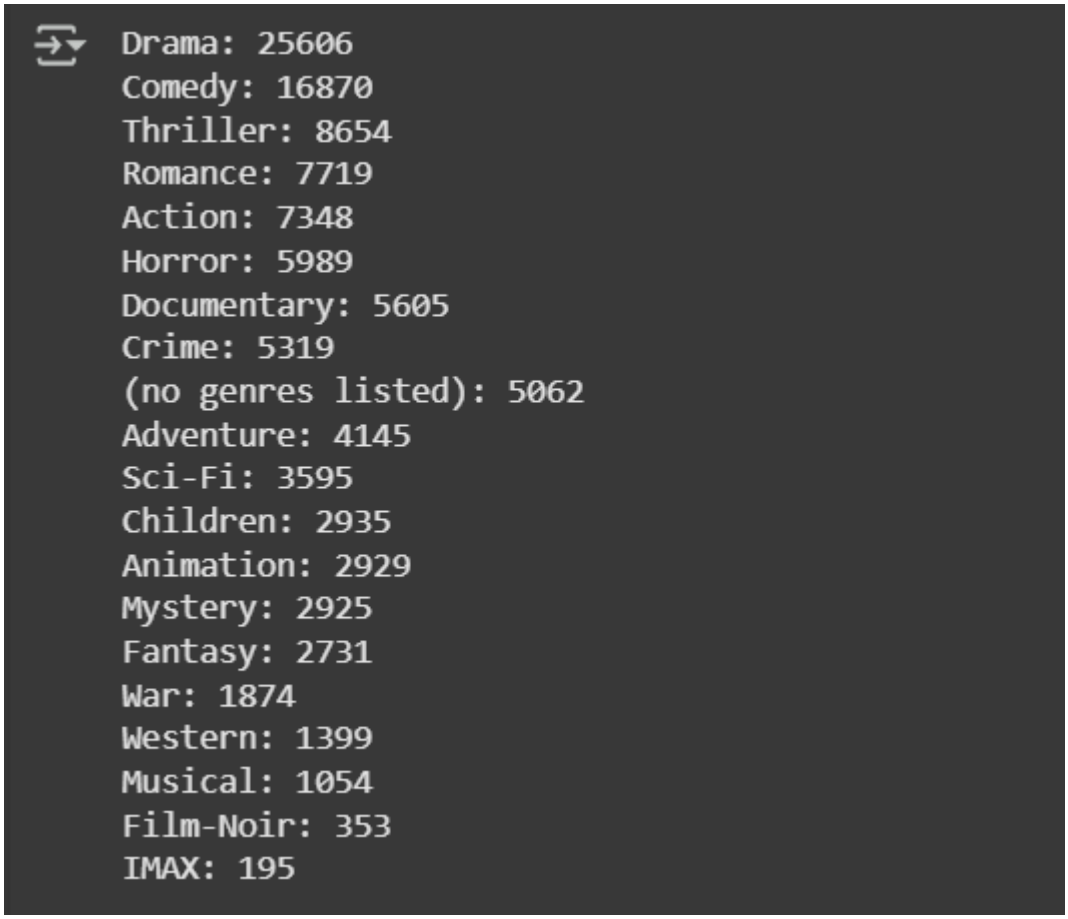
```
import pandas as pd

# Load the dataset

df = pd.read_csv("MOVIES.csv")

# Display basic info

print("Data Types:")

print(df.dtypes)

# Display first 10 rows

print("\nFirst 10 Rows:")

print(df.head(10))
```

```
Data Types:
movieId     int64
title      object
genres     object
dtype: object

First 10 Rows:
   movieId                               title  \
0        1                    Toy Story (1995)
1        2                      Jumanji (1995)
2        3             Grumpier Old Men (1995)
3        4            Waiting to Exhale (1995)
4        5  Father of the Bride Part II (1995)
5        6                           Heat (1995)
6        7                        Sabrina (1995)
7        8                  Tom and Huck (1995)
8        9                  Sudden Death (1995)
9       10                     GoldenEye (1995)

                                        genres
0  Adventure|Animation|Children|Comedy|Fantasy
1                   Adventure|Children|Fantasy
2                              Comedy|Romance
3                        Comedy|Drama|Romance
4                                      Comedy
5                       Action|Crime|Thriller
6                              Comedy|Romance
7                         Adventure|Children
8                                      Action
9                     Action|Adventure|Thriller
```

3. Genre Frequency Analysis:

```
import pandas as pd

df = pd.read_csv("MOVIES.csv")

# Split genres and count occurrences

genre_counts = {}

for genre_list in df['genres']:
```

```
    for genre in genre_list.split('|'):
        if genre in genre_counts:
            genre_counts[genre] += 1
        else:
            genre_counts[genre] = 1
# Sort and display top genres
sorted_genres = sorted(genre_counts.items(), key=lambda x: x[1], reverse=True)
for genre, count in sorted_genres:
    print(f"{genre}: {count}")
```



4. Filter Movies by Genre:

```
import pandas as pd
df = pd.read_csv("MOVIES.csv")
# Specify genre to filter
target_genre = "Comedy"
# Filter movies by genre
filtered_movies = df[df['genres'].str.contains(target_genre, na=False)]
# Display the result
print(f"Movies in Genre: {target_genre}")
print(filtered_movies.head(20))
```

```
Movies in Genre: Comedy
    movieId                                        title  \
0         1                             Toy Story (1995)
2         3                      Grumpier Old Men (1995)
3         4                     Waiting to Exhale (1995)
4         5           Father of the Bride Part II (1995)
6         7                               Sabrina (1995)
10       11                American President, The (1995)
11       12           Dracula: Dead and Loving It (1995)
17       18                           Four Rooms (1995)
18       19       Ace Ventura: When Nature Calls (1995)
19       20                          Money Train (1995)
20       21                            Get Shorty (1995)
37       38                          It Takes Two (1995)
38       39                              Clueless (1995)
44       45                           To Die For (1995)
51       52                      Mighty Aphrodite (1995)
53       54                        Big Green, The (1995)
55       56              Kids of the Round Table (1995)
57       58            Postman, The (Postino, Il) (1994)
62       63  Don't Be a Menace to South Central While Drink...
63       64                           Two if by Sea (1996)
```

```
                                                genres
0   Adventure|Animation|Children|Comedy|Fantasy
2                                Comedy|Romance
3                          Comedy|Drama|Romance
4                                       Comedy
6                                Comedy|Romance
10                         Comedy|Drama|Romance
11                                Comedy|Horror
17                                      Comedy
18                                      Comedy
19          Action|Comedy|Crime|Drama|Thriller
20                        Comedy|Crime|Thriller
37                              Children|Comedy
38                               Comedy|Romance
44                        Comedy|Drama|Thriller
51                         Comedy|Drama|Romance
53                              Children|Comedy
55           Adventure|Children|Comedy|Fantasy
57                         Comedy|Drama|Romance
62                                Comedy|Crime
63                               Comedy|Romance
```

5. Movies Count Per Year:

```
import pandas as pd

import re

df = pd.read_csv("MOVIES.csv")

# Extract year from title

df['year'] = df['title'].str.extract(r'\((\d{4})\)', expand=False)

year_counts = df['year'].value_counts().sort_index()

# Display the counts

print("Movies released per year:")

print(year_counts)
```

```
Movies released per year:
year
1874       1
1878       1
1880       1
1883       1
1887       1
          ...
2015    2513
2016    2488
2017    2374
2018    2034
2019     994
Name: count, Length: 135, dtype: int64
```

6. . Most Frequent Movie Titles:

```
import pandas as pd

df = pd.read_csv("MOVIES.csv")

# Count duplicate titles

title_counts = df['title'].value_counts()

# Display top 10 most frequent titles

print("Top 10 most frequent movie titles:")

print(title_counts.head(10))
```

```
Top 10 most frequent movie titles:
title
Code cell output actions
                                                                                2
Blockbuster (2017)                                                              2
Hostage (2005)                                                                  2
Delirium (2018)                                                                 2
Free Fall (2014)                                                                2
Grace (2014)                                                                    2
9 (2009)                                                                        2
Believer (2018)                                                                 2
The Lonely Island Presents: The Unauthorized Bash Brothers Experience (2019)    2
Escape Room (2017)                                                              2
Name: count, dtype: int64
```

7.cleaning of csv file:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('MOVIES.csv')

# Show original structure
print("Original dataset shape:", df.shape)
print("Sample records:")
print(df.head())

# Drop rows with missing values (if any)
df.dropna(inplace=True)

# Extract year from title and remove from title
import re
def extract_year(title):
    match = re.search(r'\((\d{4})\)', title)
    return int(match.group(1)) if match else None

def clean_title(title):
    return re.sub(r'\(\d{4}\)', '', title).strip()

df['year'] = df['title'].apply(extract_year)
df['clean_title'] = df['title'].apply(clean_title)

# Split genres into lists
df['genre_list'] = df['genres'].apply(lambda x: x.split('|') if isinstance(x, str) else [])

# Display the most common genres
```

```
from collections import Counter
genre_counts = Counter([genre for sublist in df['genre_list'] for genre in sublist])
print("\nTop 10 genres:")
for genre, count in genre_counts.most_common(10):
    print(f"{genre}: {count}")


# Save cleaned version
df.to_csv('cleaned_movies.csv', index=False)
print("\nCleaned data saved as 'cleaned_movies.csv'")
```

```
Original dataset shape: (62423, 3)
Sample records:
   movieId                                title  \
0        1                    Toy Story (1995)
1        2                      Jumanji (1995)
2        3             Grumpier Old Men (1995)
3        4            Waiting to Exhale (1995)
4        5  Father of the Bride Part II (1995)

                                        genres
0  Adventure|Animation|Children|Comedy|Fantasy
1                   Adventure|Children|Fantasy
2                              Comedy|Romance
3                        Comedy|Drama|Romance
4                                      Comedy

Top 10 genres:
Drama: 25606
Comedy: 16870
Thriller: 8654
Romance: 7719
Action: 7348
Horror: 5989
Documentary: 5605
Crime: 5319
(no genres listed): 5062
Adventure: 4145

Cleaned data saved as 'cleaned_movies.csv'
```

8.recommendation of users favourite movie:

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Load movie data
movies = pd.read_csv("MOVIES.csv")

# Fill missing genre values and preprocess
def preprocess_genres(df):
    df['genres'] = df['genres'].fillna('')
    df['genres'] = df['genres'].str.replace('|', ' ', regex=False)
    return df

movies = preprocess_genres(movies)

# TF-IDF Vectorization of genres
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(movies['genres'])

# Function to get recommendations based on genre preference
def recommend_movies(preferred_genres, top_n=10):
    user_profile = vectorizer.transform([preferred_genres.replace('|', ' ')])
    similarity_scores = cosine_similarity(user_profile, tfidf_matrix)
    scores = list(enumerate(similarity_scores[0]))
    scores = sorted(scores, key=lambda x: x[1], reverse=True)
    recommended_indices = [i for i, _ in scores[:top_n]]
```

```
    return movies.iloc[recommended_indices][['title', 'genres']]


# Example usage
if __name__ == "__main__":
    print("Welcome to the AI-Driven Movie Recommender!")
    user_input = input("Enter your favorite genres (e.g., Action|Comedy|Drama): ")
    recommendations = recommend_movies(user_input)
    print("\nTop Movie Recommendations for You:")
    print(recommendations.to_string(index=False))
```