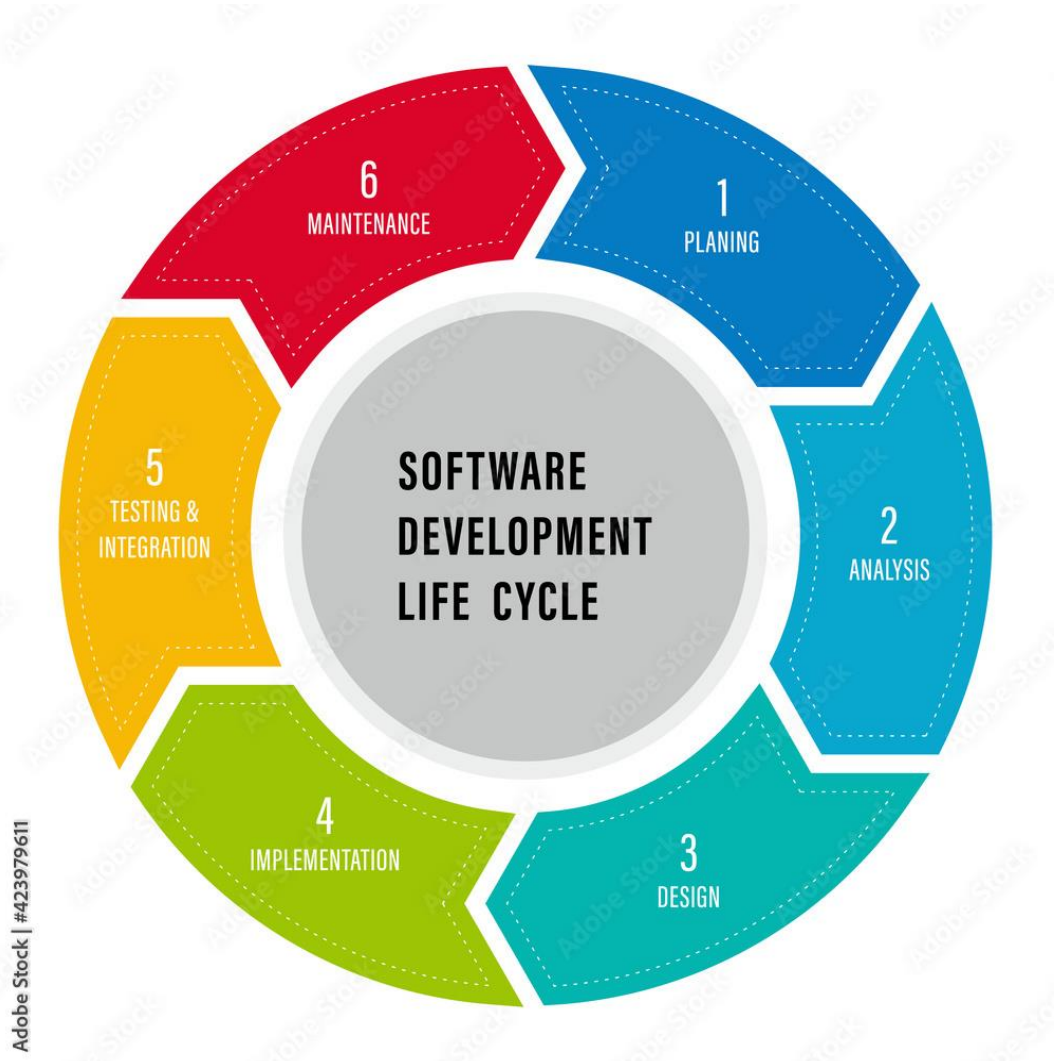
Project Development Guidelines

The software development life cycle (SDLC) is a step-by-step process that development teams use to create high-quality, cost-effective and secure software.

The SDLC breaks down software development into distinct, repeatable stages and provides a roadmap that helps organizations create software that meets stakeholder needs and customer expectations throughout the software's lifecycle.

Each phase of the SDLC has its own objectives and specific deliverables that help guide the next phase of software development.



✶ SDLC typically has various phases, many models condense it into six main phases. Here's an overview of the six-phase SDLC model:

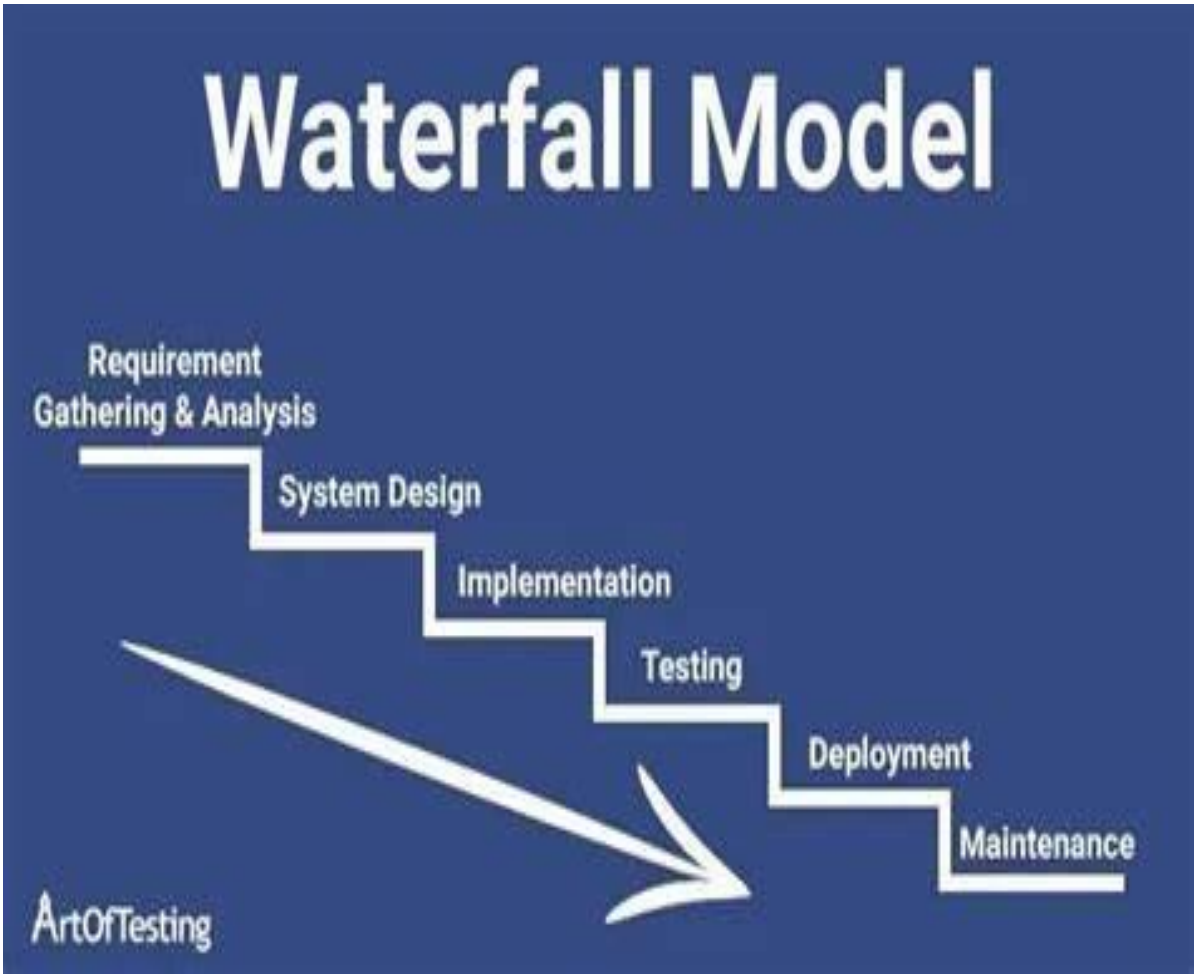
1. **Planning**: This phase involves defining the project's objectives, scope, resources, and timeline. It includes conducting feasibility studies, risk assessments, and resource allocation.
2. **Analysis**: During this phase, detailed requirements are gathered from stakeholders. This involves understanding user needs, documenting functional and non-functional requirements, and creating use case diagrams or other analysis models.
3. **Design**: The design phase translates the requirements into detailed specifications for system architecture, user interfaces, databases, and other components. This phase includes creating design documents, data models, and prototypes.
4. **Development**: In this phase, the actual coding and implementation of the system take place. Developers build the system based on the design documents, ensuring it meets the specified requirements.
5. **Testing**: Once the system is developed, it undergoes rigorous testing to identify and fix any defects or issues. This phase ensures that the system functions correctly and meets quality standards.
6. **Deployment and Maintenance**: The final phase involves deploying the system to the production environment and making it available to end-users. It also includes ongoing maintenance and support to ensure the system remains operational and performs optimally.

SDLC MODELS

1. Waterfall Model

The waterfall model is a software development model used in the context of large, complex projects, typically in the field of information technology. It is characterized by a structured, sequential approach to project management and software development.

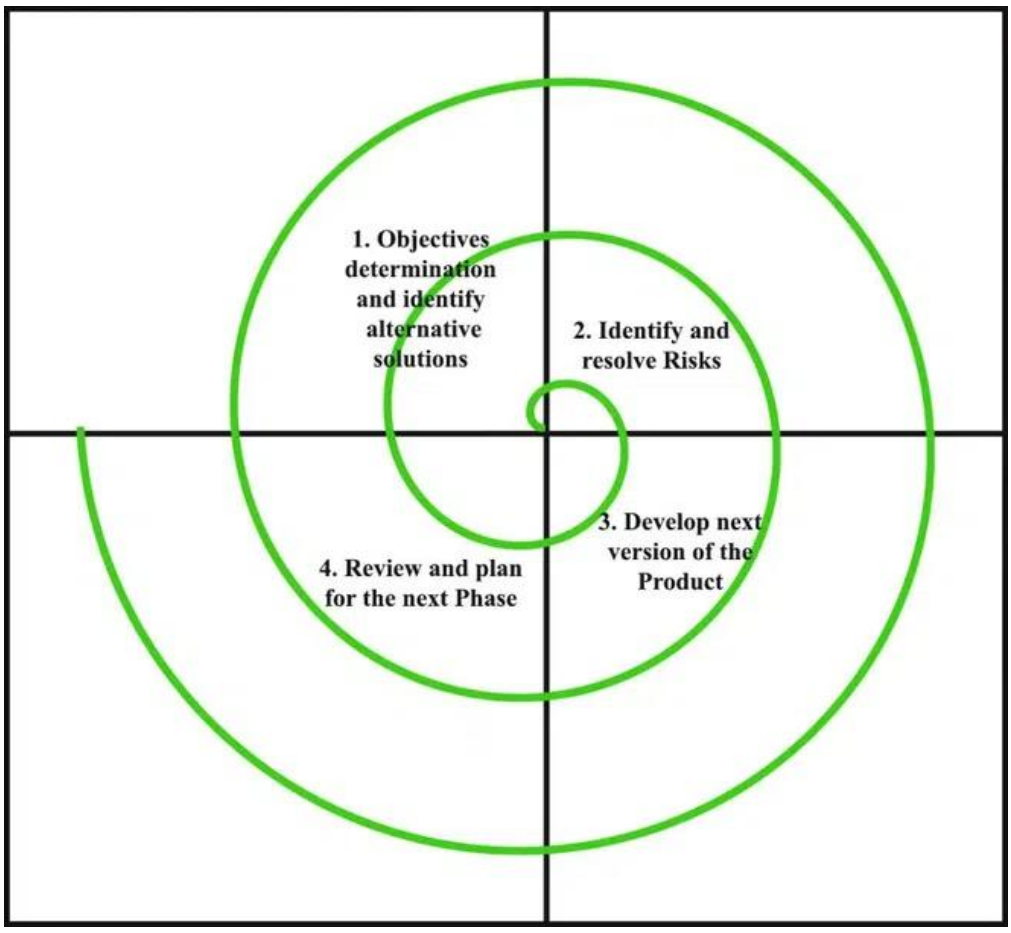
The waterfall model is useful in situations where the project requirements are well-defined and the project goals are clear. It is often used for large-scale projects with long timelines, where there is little room for error and the project stakeholders need to have a high level of confidence in the outcome.



2. Spiral Model

The Spiral Model is one of the most important Software Development Life Cycle models. The Spiral Model is a combination of the waterfall model and the iterative model. It provides support for Risk Handling. The Spiral Model was first proposed by Barry Boehm.

The Spiral Model provides a systematic and iterative approach to software development. In its diagrammatic representation, looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a phase of the software development process. The Spiral Model is often used for complex and large software development projects, as it allows for a more flexible and adaptable approach to software development. It is also well-suited to projects with significant uncertainty or high levels of risk.



Scenario

Netflix Movie Data Analysis using Python

Libraries used – Numpy, Pandas, Matplotlib and Seaborn

1. What is Netflix Movies Data Analysis?

- Netflix Movies Data Analysis involves the use of data analytics to understand viewer behaviour, preferences, and trends in movie consumption on the Netflix platform. This analysis is crucial for informing.

2. How to perform?

- To perform Netflix Movies Data Analysis, you can follow a structured approach that includes data collection, cleaning, exploratory data analysis (EDA), and visualization. Here's a detailed guide based on the information gathered:
- Step-by-Step Guide to Perform Netflix Movies Data Analysis

a. **Data Collection**

You can obtain the Netflix dataset from sources like Kaggle, which hosts various datasets related to movies and shows available on Netflix.

b. **Data Cleaning**

Once you have the dataset, you need to clean it to ensure accuracy and consistency. This involves:

- **Handling Missing Values**: Replace missing entries in columns like director, country, and cast with "Unknown" rather than using statistical methods like mean or mode.
- **Unnesting Data**: For columns like **cast**, which contain multiple names separated by commas, you need to split these into separate rows. Here's an example code snippet using Python:

c. **Exploratory Data Analysis (EDA)**

EDA is crucial for understanding the dataset. This involve

✶ **Generating Visualizations**: Use libraries such as Matplotlib and Seaborn to create various plots:

- **Histograms**: To visualize distributions of numerical variables.
- **Box Plots**: To identify outliers and understand data spread.
- **Count Plots**: To show counts of categorical variables.
- **Scatter Plots**: To examine relationships between two numerical variables.

✶ **Statistical Analysis**: Calculate descriptive statistics (mean, median, mode) to summarize data characteristics.

✶ **Correlation Analysis**: Use correlation matrices to explore relationships between different variables.

d. **Insights Generation**

After conducting EDA, derive insights that can inform decision-making:

- Identify popular genres or directors based on historical data.
- Analyze viewer demographics and preferences to tailor content offerings.
- Examine seasonal trends in viewership for strategic release planning.

e. Reporting Findings

Document your findings and insights in a clear and structured manner. Use visualizations to support your conclusions and provide actionable recommendations for content strategy.

❖ **Tools Used**

- **Programming Languages**: Python is commonly used for data analysis.
- **Libraries**:
  - Pandas for data manipulation.
  - Matplotlib and Seaborn for visualization.
- **Jupyter Notebook**: Ideal for documenting the analysis process interactively.

3. Purpose

The purpose of Netflix Movies Data Analysis is to use data to make better decisions about the movies and shows they offer. By looking at what viewers like and how they watch content, Netflix can figure out which genres are popular and what types of shows to create or buy. This analysis helps Netflix understand its audience better, allowing them to tailor their marketing strategies based on where viewers are located and when they watch. Additionally, it helps Netflix measure how well their movies and shows are doing based on ratings and viewership. Overall, this process helps Netflix stay ahead in the competitive world of streaming by ensuring they provide content that people want to watch.

4. Things to considering before doing analysis.

Before you start analyzing data, keep these simple things in mind:

1. **Know Your Goals**: Decide what you want to find out from the analysis.
2. **Understand Your Data**: Get to know the data you have and what it means.
3. **Check Data Quality**: Look for any missing information or mistakes in the data.
4. **Clean the Data**: Fix any errors and handle missing values to make sure the data is accurate.
5. **Choose Analysis Methods**: Pick the right ways to analyze the data based on what you want to learn.
6. **Plan for Problems**: Think about any challenges you might face during the analysis and how to deal with them.

7. **Keep Records**: Write down what you do and any important decisions you make during the analysis.

By considering these points, you'll be better prepared for a successful data analysis.

5. Objective

The primary objective of this project is to perform an in-depth analysis of Netflix movie data to uncover insights into content trends, user preferences, and potential areas for improvement. This involves exploring various aspects such as content diversity, release patterns, user ratings, and genre preference

- **Importing the libraries**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

- **# Load the dataset**

```
[11]: import pandas as pd
      df= pd.read_csv('mymoviedb.csv', lineterminator = '\n')
```

- **# Display the first few rows of the dataset**

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

[12]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	en	Action, Adventure, Science Fiction	https://image.tmdb.org/t/p/original/1g0dhYtq4i...
1	2022-03-01	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	8.1	en	Crime, Mystery, Thriller	https://image.tmdb.org/t/p/original/74xTEgt7R3...
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmdb.org/t/p/original/vDHsLnOWKl...
3	2021-11-24	Encanto	The tale of an extraordinary family, the Madri...	2402.201	5076	7.7	en	Animation, Comedy, Family, Fantasy	https://image.tmdb.org/t/p/original/4j0PNHkMr5...
4	2021-12-22	The King's Man	As a collection of history's worst tyrants and...	1895.511	1793	7.0	en	Action, Adventure, Thriller, War	https://image.tmdb.org/t/p/original/aq4Pwv5Xeu...

- **#The number of rows and columns**

```
[14]: df.shape
```

[14]: (9827, 9)

- **# Prints the information about the DataFrame**

```
[13]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9827 entries, 0 to 9826
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Release_Date       9827 non-null   object 
 1   Title              9827 non-null   object 
 2   Overview           9827 non-null   object 
 3   Popularity         9827 non-null   float64
 4   Vote_Count         9827 non-null   int64  
 5   Vote_Average       9827 non-null   float64
 6   Original_Language  9827 non-null   object 
 7   Genre              9827 non-null   object 
 8   Poster_Url         9827 non-null   object 
dtypes: float64(2), int64(1), object(6)
memory usage: 691.1+ KB
```

- **#Dropping Column**

Dropping columns

```
5]: import pandas as pd

# Sample DataFrame (assuming it's loaded with the correct data)
df= pd.read_csv('mymoviedb.csv', lineterminator = '\n')
# Verify column names


# Drop columns if they exist
cols = ['Overview', 'Original_Language', 'Poster_Url']
df.drop(cols, axis=1, inplace=True, errors='ignore')

# Check remaining columns
print(df.columns)
```

```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
       'Genre'],
      dtype='object')
```

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

[68]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021-12-15	Spider-Man: No Way Home	5083.954	8940	8.3	Action, Adventure, Science Fiction
1	2022-03-01	The Batman	3827.658	1151	8.1	Crime, Mystery, Thriller
2	2022-02-25	No Exit	2618.087	122	6.3	Thriller
3	2021-11-24	Encanto	2402.201	5076	7.7	Animation, Comedy, Family, Fantasy
4	2021-12-22	The King's Man	1895.511	1793	7.0	Action, Adventure, Thriller, War

- **# Display the Description of the data in DataFrame**

```
[20]: df.describe()
```

[20]:

	Popularity	Vote_Count	Vote_Average
count	9827.000000	9827.000000	9827.000000
mean	40.326088	1392.805536	6.439534
std	108.873998	2611.206907	1.129759
min	13.354000	0.000000	0.000000
25%	16.128500	146.000000	5.900000
50%	21.199000	444.000000	6.500000
75%	35.191500	1376.000000	7.100000
max	5083.954000	31077.000000	10.000000

✶ Data Visualization

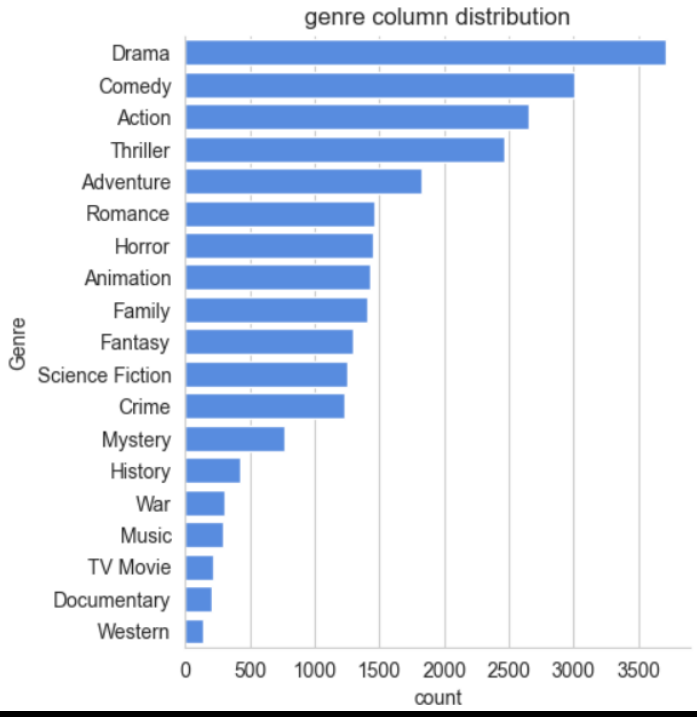
Q1: What is the most frequent genre in the dataset?

```
[97]: df['Genre'].describe()

[97]: count     25552
      unique       19
      top       Drama
      freq       3715
      Name: Genre, dtype: object
```

```
113]: import seaborn as sns
      import matplotlib.pyplot as plt
      sns.catplot(y = 'Genre', data = df, kind = 'count',
              order = df['Genre'].value_counts().index,
              color = '#4287f5')
      plt.title('genre column distribution')
      plt.show()
```

Output :-

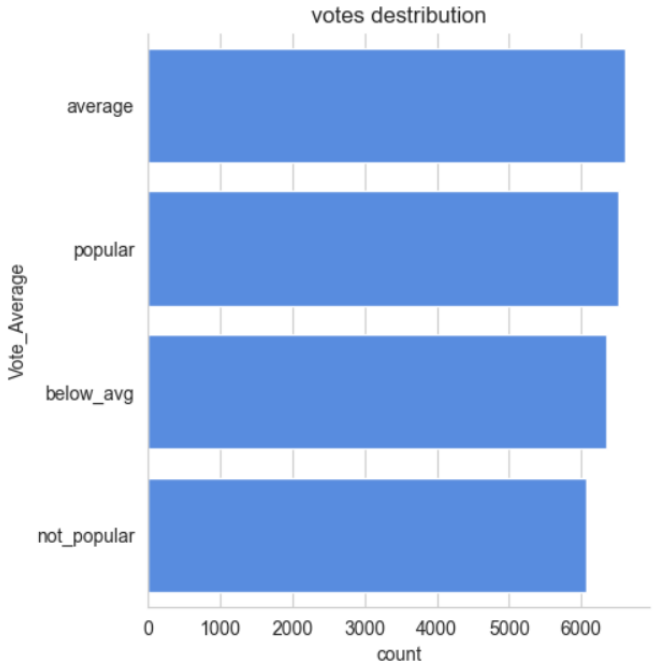


We can notice from the above visual that Drama genre is the most frequent genre in our dataset and has appeared more than 14% of the times among 19 other genres.

Q2: What genres has highest votes ?

```
[114]: # visualizing vote_average column
sns.catplot(y = 'Vote_Average', data = df, kind = 'count',
order = df['Vote_Average'].value_counts().index,
color = '#4287f5')
plt.title('votes destribution')
plt.show()
```

Output:-



Average Voting are Highest.

Q3: What movie got the highest genre?

[116]:
```
# checking max popularity in dataset
df[df['Popularity'] == df['Popularity'].max()]
```

Output:-

[6]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	en	Action, Adventure, Science Fiction

Conclusion :-

Spider-Man: No Way Home has the highest popularity rate in our dataset and it has genres of Action, Adventure and Science Fiction.

Q4: What movie got the lowest popularity? what's its genre?

[119]:
```
# checking max popularity in dataset
df[df['Popularity'] == df['Popularity'].min()]
```

Output:-

[7]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
9825	2021-03-31	The United States vs. Billie Holiday	Billie Holiday spent much of her career being ...	13.354	152	6.7	en	Music, Drama, History	https://image.tmdb.org/t/p/original/vEzkxuE2sJ...
9826	1984-09-23	Threads	Documentary style account of a nuclear holocau...	13.354	186	7.8	en	War, Drama, Science Fiction	https://image.tmdb.org/t/p/original/lBhU4U9Eeh...

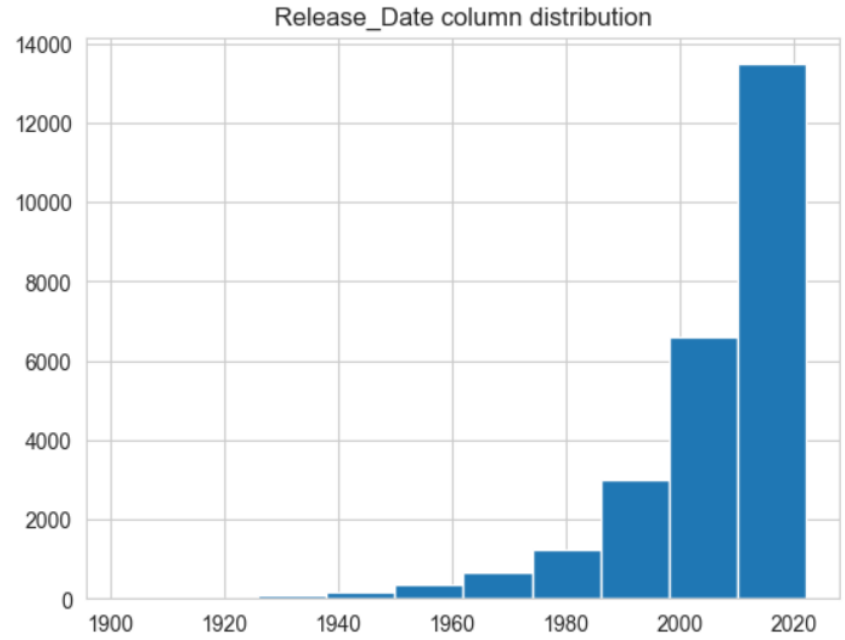
Conclusion:-

The united states, thread' has the highest lowest rate in our dataset and it has genres of music, drama, 'war', 'sci-fi' and history`.

Q5: Which year has the most filmmed movies?

```
[121]: df['Release_Date'].hist()
       plt.title('Release_Date column distribution')
       plt.show()
```

Output :-



Conclusion:-

Year 2020 has the highest filmming rate in dataset.