Netflix Content Analysis Using Python

Project Title	**Netflix Data: Cleaning, Analysis and Visualization**
Tools	Python, ML, SQL, Excel
Domain	Data Analyst & Data scientist
Project Difficulties level	intermediate

Dataset : Dataset is available in the given link. You can download it at your convenience.

About Dataset

Netflix is a popular streaming service that offers a vast catalog of movies, TV shows, and original contents. This dataset is a cleaned version of the original version which can be found here. The data consist of contents added to Netflix from 2008 to 2021. The oldest content is as old as 1925 and the newest as 2021. This dataset will be cleaned with PostgreSQL and visualized with Tableau. The purpose of this dataset is to test my data cleaning and visualization skills. The cleaned data can be found below and the Tableau dashboard can be found here .

Data Cleaning

We are going to:

1. Treat the Nulls
2. Treat the duplicates
3. Populate missing rows
4. Drop unneeded columns
5. Split columns

   Extra steps and more explanation on the process will be explained through the code comments

Example: You can get the basic idea how you can create a project from here

Netflix Data: Cleaning, Analysis, and Visualization (Beginner ML Project)

This project involves loading, cleaning, analyzing, and visualizing data from a Netflix dataset. We'll use Python libraries like Pandas, Matplotlib, and Seaborn to work through the project. The goal is to explore the dataset, derive insights, and prepare for potential machine learning tasks.

Step 1: Import Required Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
```

Step 2: Load the Dataset

Assume we have a dataset named netflix_titles.csv.

```
# Load the dataset
data = pd.read_csv('netflix_titles.csv')

# Display the first few rows of the dataset
print(data.head())
```

Step 3: Data Cleaning

Identify and handle missing data, correct data types, and drop duplicates.

```
# Check for missing values
print(data.isnull().sum())

# Drop duplicates if any
data.drop_duplicates(inplace=True)

# Drop rows with missing critical information
data.dropna(subset=['director', 'cast', 'country'],
inplace=True)

# Convert 'date_added' to datetime
data['date_added'] = pd.to_datetime(data['date_added'])
```

```
# Show data types to confirm changes
print(data.dtypes)
```

Step 4: Exploratory Data Analysis (EDA)

1. Content Type Distribution (Movies vs. TV Shows)

```
# Count the number of Movies and TV Shows
type_counts = data['type'].value_counts()

# Plot the distribution
plt.figure(figsize=(8, 6))
sns.barplot(x=type_counts.index, y=type_counts.values,
palette='Set2')
plt.title('Distribution of Content by Type')
plt.xlabel('Type')
plt.ylabel('Count')
plt.show()
```

2. Most Common Genres

```
# Split the 'listed_in' column and count genres
data['genres'] = data['listed_in'].apply(lambda x: x.split(',
'))
all_genres = sum(data['genres'], [])
genre_counts = pd.Series(all_genres).value_counts().head(10)
```

```
#    Plot    the    most    common    genres
plt.figure(figsize=(10,                 6))
sns.barplot(x=genre_counts.values,        y=genre_counts.index,
palette='Set3')   plt.title('Most   Common
Genres  on  Netflix')  plt.xlabel('Count')
plt.ylabel('Genre') plt.show()
```

3. Content Added Over Time

```
# Extract year and month from 'date_added'
data['year_added'] = data['date_added'].dt.year
data['month_added'] = data['date_added'].dt.month

# Plot content added over the years
plt.figure(figsize=(12, 6))
sns.countplot(x='year_added', data=data, palette='coolwarm')
plt.title('Content Added Over Time')
plt.xlabel('Year')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

4. Top 10 Directors with the Most Titles

```
# Count titles by director
top_directors = data['director'].value_counts().head(10)

# Plot top directors
plt.figure(figsize=(10, 6))
sns.barplot(x=top_directors.values, y=top_directors.index,
palette='Blues_d')
plt.title('Top 10 Directors with the Most Titles')
plt.xlabel('Number of Titles')
plt.ylabel('Director')
plt.show()
```

5. Word Cloud of Movie Titles

```
# Generate word cloud
movie_titles = data[data['type'] == 'Movie']['title']
wordcloud = WordCloud(width=800, height=400,
background_color='black').generate(' '.join(movie_titles))

# Plot word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Step 5: Conclusion and Insights

In this project, we:

1. **Cleaned the data** by handling missing values, removing duplicates, and converting data types.
2. **Explored the data** through various visualizations such as bar plots and word clouds.
3. **Analyzed content trends** over time, identified popular genres, and highlighted top directors.

Step 6: Next Steps

1. **Feature Engineering** :Createnewfeatures,suchascountingthenumberof genres per movie or extracting the duration in minutes.
2. **Machine Learning**: Use the cleaned and processed data to build models for recommendations or trend predictions.
3. **Advanced Visualization**: Use interactive plots or dashboards for more detailed analysis.

This project is a foundational exercise that introduces essential data analysis techniques, paving the way for more advanced projects.

📬 Conclusion

This project provides a basic but insightful look at Netflix's content using Python data analysis techniques. It can be extended to include recommendations, sentiment analysis, or integration with streaming APIs.