```
import numpy as np
```

```
df=pd.read_csv('/content/disney_plus_titles.csv')
```

```
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	A Spark Story	Jason Sterman, Leanne Dare	Apthon Corbin, Louis Gonzales	NaN	September 24, 2021	2021	TV-PG	88 min	Documentary	Two Pixar filmmakers strive to bring their uni...
1	s2	Movie	Spooky Buddies	Robert Vince	Tucker Albrizzi, Diedrich Bader, Ameko Eks Mas...	United States, Canada	September 24, 2021	2011	G	93 min	Comedy, Fantasy, Kids	The puppies go on a spooky adventure through a...
2	s3	Movie	The Fault in Our Stars	Josh Boone	Shailene Woodley, Ansel Elgort, Laura Dern, Sa	United States	September 24, 2021	2014	PG-13	127 min	Coming of Age, Drama, Romance	Hazel and Gus share a love that sweeps them on...

```
df.shape
```

```
(1368, 12)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1368 entries, 0 to 1367
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       1368 non-null   object
 1   type          1368 non-null   object
 2   title         1368 non-null   object
 3   director      928 non-null    object
 4   cast          1194 non-null   object
 5   country       1193 non-null   object
 6   date_added    1365 non-null   object
 7   release_year  1368 non-null   int64 
 8   rating        1366 non-null   object
 9   duration      1368 non-null   object
 10  listed_in     1368 non-null   object
 11  description   1368 non-null   object
dtypes: int64(1), object(11)
memory usage: 128.4+ KB
```

```
df.columns
```

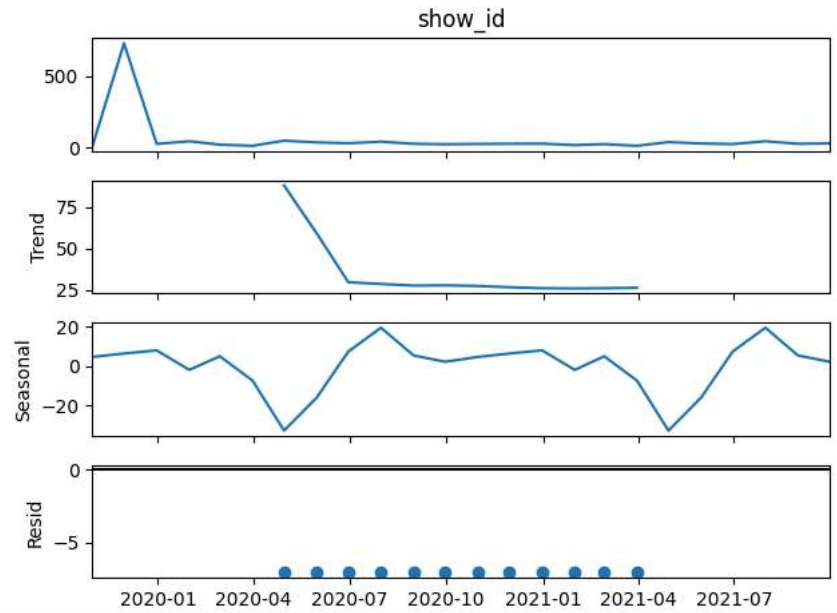
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

```
df['date_added'] = pd.to_datetime(df['date_added'])
df.set_index('date_added', inplace=True)
```

```
monthly_data = df['show_id'].resample('M').count()
```

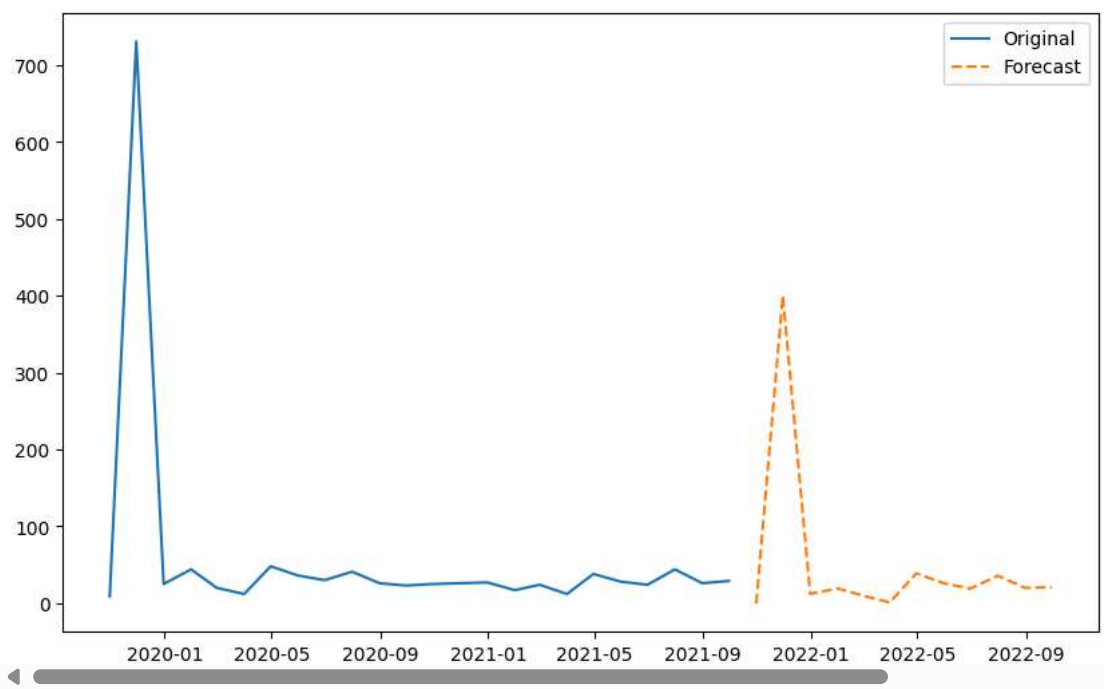
```
decomposition = seasonal_decompose(monthly_data, model='additive')
decomposition.plot()
plt.show()
```




```
model = ExponentialSmoothing(monthly_data, seasonal='add', seasonal_periods=12).fit()
forecast = model.forecast(12)
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency information was provided, so i
  self._init_dates(dates, freq)
```

```
plt.figure(figsize=(10, 6))
plt.plot(monthly_data, label='Original')
plt.plot(forecast, label='Forecast', linestyle='--')
plt.legend()
plt.show()
```



```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

nltk.download('vader_lexicon')

sid = SentimentIntensityAnalyzer()

def get_sentiment(description):
    return sid.polarity_scores(description)['compound']

df['sentiment_score'] = df['description'].apply(get_sentiment)

df[['description', 'sentiment_score']]
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

date_added	description	sentiment_score
2021-09-24	Two Pixar filmmakers strive to bring their uni...	0.2263
2021-09-24	The puppies go on a spooky adventure through a...	-0.2023
2021-09-24	Hazel and Gus share a love that sweeps them on...	0.7506
2021-09-22	Matt Beisner uses unique approaches to modifyi...	0.0000
2021-09-22	Spidey teams up with pals to become The Spidey...	0.0000
...	...	...
2019-10-01	Merlin trains a young orphan who's destined to...	0.0000
2019-10-01	A strong-willed family struggles to establish ...	-0.3612
NaT	Welcome to Kirby's world! It's rude and sketchy.	0.0000
NaT	Ryan discovers his ability to control a giant ...	0.3182
NaT	Rock out with the Imagination Movers, Disney J...	0.5093

1368 rows × 2 columns

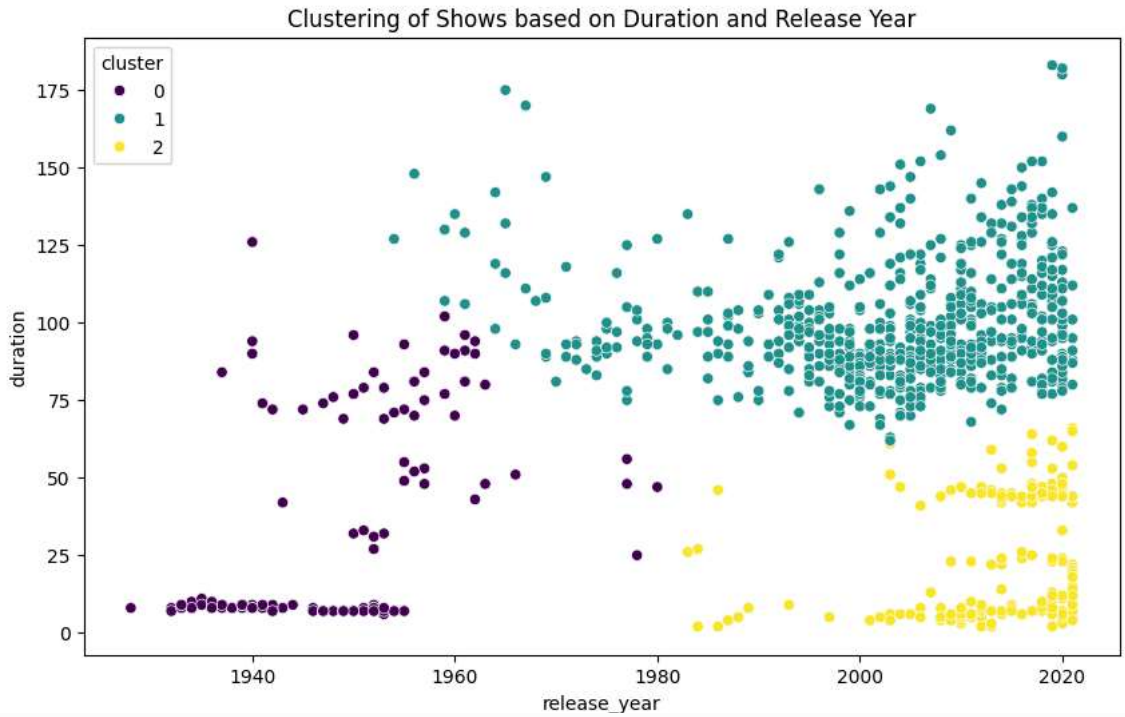
```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import seaborn as sns


df_data = df[df.duration.str.endswith("min")]
df_data['duration'] = df_data['duration'].str.replace(' min', '').astype(float)
data = df_data[['release_year', 'duration']].dropna()
```

```
<ipython-input-27-b58205cf03d6>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df_data['duration'] = df_data['duration'].str.replace(' min', '').astype(float)
```

```
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)


kmeans = KMeans(n_clusters=3, random_state=0)
data['cluster'] = kmeans.fit_predict(data_scaled)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10
  warnings.warn(
```

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='release_year', y='duration', hue='cluster', data=data, palette='viridis')
plt.title('Clustering of Shows based on Duration and Release Year')
plt.show()
```



Start coding or generate with AI.