

```
In [22]: import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
plt.style.use('fivethirtyeight')

#Modules for ML(Recommendation)
from sklearn.preprocessing import MinMaxScaler
from sklearn.neighbors import NearestNeighbors
from sklearn.metrics.pairwise import cosine_similarity

%matplotlib inline

In [2]: df = pd.read_csv('top100_kdrama.csv')
df.shape

Out[2]: (100, 14)

In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   Name                 100 non-null   object
 1   Year of release      100 non-null   int64
 2   Aired Date           100 non-null   object
 3   Aired On             100 non-null   object
 4   Number of Episode    100 non-null   int64
 5   Network              100 non-null   object
 6   Duration              100 non-null   object
 7   Content Rating       100 non-null   object
 8   Synopsis             100 non-null   object
 9   Cast                 100 non-null   object
10   Genre                100 non-null   object
11   Tags                 100 non-null   object
12   Rank                 100 non-null   object
13   Rating               100 non-null   float64
dtypes: float64(1), int64(2), object(11)
memory usage: 11.1+ KB

In [5]: df.head()

Out[5]:
```

	Name	Year of release	Aired Date	Aired On	Number of Episode	Network	Duration	Content Rating	Synopsis	Cast	Genre	Tag
0	Move to Heaven	2021	May 14, 2021	Friday	10	Netflix	52 min.	18+ Restricted (violence & profanity)	Geu Roo is a young autistic man. He works for ...	Lee Je Hoon, Tang Jun Sang, Hong Seung Hee, Ju...	Life, Drama, Family	Autism Uncle Nephew Relationship Death Sava...
1	Hospital Playlist	2020	Mar 12, 2020 - May 28, 2020	Thursday	12	Netflix, tvN	1 hr. 30 min.	15+ - Teens 15 or older	The stories of people going through their days...	Jo Jung Suk, Yoo Yeon Seok, Jung Kyung Ho, Kim...	Friendship, Romance, Life, Medical	Friendship Multiple Mains, Bes Friend.
2	Flower of Evil	2020	Jul 29, 2020 - Sep 23, 2020	Wednesday, Thursday	16	tvN	1 hr. 10 min.	15+ - Teens 15 or older	Although Baek Hee Sung is hiding a dark secret...	Lee Joon Gi, Moon Chae Won, Wang Hae Jin, Seo ...	Thriller, Romance, Crime, Melodrama	Married Couple Deception Suspense Family Se...
3	Hospital Playlist 2	2021	Jun 17, 2021 - Jan	Thursday	12	Netflix, tvN	1 hr. 40 min	15+ - Teens 15 or older	Everyday is extraordinary for live	Jo Jung Suk, Yoo Yeon Seok	Friendship, Romance,	Workplace Story Friendship Rec

```
In [6]: kdrama_names = df[['Name']]
kdrama_names.head()

Out[6]:
```

	Name
0	Move to Heaven
1	Hospital Playlist
2	Flower of Evil
3	Hospital Playlist 2
4	My Mister

```
In [7]: cols_for_recommend = ['Year of release', 'Number of Episode', 'Network', 'Duration', 'Content Rating', 'Rating']
df = df[cols_for_recommend]
df.head()

Out[7]:
```

	Year of release	Number of Episode	Network	Duration	Content Rating	Rating
0	2021	10	Netflix	52 min.	18+ Restricted (violence & profanity)	9.2
1	2020	12	Netflix, tvN	1 hr. 30 min.	15+ - Teens 15 or older	9.1
2	2020	16	tvN	1 hr. 10 min.	15+ - Teens 15 or older	9.1
3	2021	12	Netflix, tvN	1 hr. 40 min.	15+ - Teens 15 or older	9.1
4	2018	16	tvN	1 hr. 17 min.	15+ - Teens 15 or older	9.1

```
In [8]: networks = []
[networks.append(list(set(network.replace(' ','').split(','))))[0]] for network in df['Network']

Out[8]: ['Netflix', 'tvN', 'tvN', 'tvN', 'tvN']

In [9]: df['Network'] = networks
df['Network'].unique()

Out[9]: array(['Netflix', 'tvN', 'jTBC', 'KBS2', 'OCN', 'SBS', 'MBC'],
      dtype=object)

In [10]: plt.figure(figsize=(7,7))
df['Network'].value_counts().plot(kind='barh')

plt.gca().invert_yaxis()
plt.title("Networks of Kdramas.")
plt.xlabel('Frequency')
plt.show()

df['Network'].value_counts()
```

```
Out[10]: tvN      35
SBS       19
KBS2      11
jTBC       11
MBC         9
OCN         8
Netflix     7
Name: Network, dtype: int64

In [11]: df['Network'].replace(['OCN', 'jTBC'], ['Others', 'Others'], inplace=True)

In [12]: plt.figure(figsize=(7,7))
df['Network'].value_counts().plot(kind='barh')

plt.gca().invert_yaxis()
plt.title("Networks of Kdramas.")
plt.xlabel('Frequency')
plt.ylabel('Network')
plt.show()

df['Network'].value_counts()
```

```
Out[12]: tvN      35
SBS       19
KBS2      11
jTBC       11
MBC         9
Others      8
Netflix     7
Name: Network, dtype: int64

In [13]: df['Duration'] = df['Duration'].str.replace('[A-Za-z]', '', regex=True)
df['Duration'].head()

Out[13]:
```

	Duration
0	52
1	1 30
2	1 10
3	1 40
4	1 17

Name: Duration, dtype: object

```
In [14]: df['Duration'] = df['Duration'].str.replace(' ', '', regex=True)
df['Duration'] = pd.to_numeric(df['Duration'])
df['Duration'].head()

Out[14]:
```

	Duration
0	52
1	130
2	110
3	140
4	117

Name: Duration, dtype: int64

```
In [17]: plt.figure(figsize=(7,7))
df['Content Rating'].value_counts().plot(kind='pie', autopct='%2f%%')
plt.title("Content Rating")
plt.show()
```

```
In [18]: df['Content Rating'].value_counts()

Out[18]: 15+ - Teens 15 or older      88
18+ Restricted (violence & profanity)  10
13+ - Teens 13 or older              2
Name: Content Rating, dtype: int64

In [27]: df[['Rating']].describe()

Out[27]:
```

	Rating
count	100.000000
mean	8.723000
std	0.174573
min	8.500000
25%	8.600000
50%	8.700000
75%	8.800000
max	9.200000

```
In [28]: df.head()

Out[28]:
```

	Rating
count	100.000000
mean	8.723000
std	0.174573
min	8.500000
25%	8.600000
50%	8.700000
75%	8.800000
max	9.200000

```
In [29]: cols_to_encode = ['Network', 'Content Rating']
dummies = pd.get_dummies(df[cols_to_encode], drop_first=True)
dummies.head()

Out[29]:
```

	Network_MBC	Network_Netflix	Network_Others	Network_SBS	Network_jTBC	Network_tvN	Content Rating_15+ - Teens 15 or older	Content Rating_18+ Restricted (violence & profanity)
0	0	1	0	0	0	0	0	1
1	0	0	0	0	0	1	1	0
2	0	0	0	0	0	1	1	0
3	0	0	0	0	0	1	1	0

```
In [30]: df.drop(cols_to_encode, axis=1, inplace=True)
df.head()

Out[30]:
```

	Year of release	Number of Episode	Duration	Rating
0	2021	10	52	9.2
1	2020	12	130	9.1
2	2020	16	110	9.1
3	2021	12	140	9.1
4	2018	16	117	9.1

```
In [31]: scale = MinMaxScaler()
scaled = scale.fit_transform(df)

In [32]: i=0
for col in df.columns:
    df[col] = scaled[:,i]
    i += 1

In [33]: df.head()

Out[33]:
```

	Year of release	Number of Episode	Duration	Rating
0	1.000000	0.042553	0.312500	1.000000
1	0.944444	0.063830	0.921875	0.857143
2	0.944444	0.106383	0.765625	0.857143
3	1.000000	0.063830	1.000000	0.857143
4	0.833333	0.106383	0.820312	0.857143

```
In [34]: new_df = pd.concat([df, dummies], axis=1)
new_df.shape

Out[34]: (100, 12)

In [35]: new_df.head()

Out[35]:
```

	Year of release	Number of Episode	Duration	Rating	Network_MBC	Network_Netflix	Network_Others	Network_SBS	Network_jTBC	Net
0	1.000000	0.042553	0.312500	1.000000	0	1	0	0	0	0
1	0.944444	0.063830	0.921875	0.857143	0	0	0	0	0	0
2	0.944444	0.106383	0.765625	0.857143	0	0	0	0	0	0
3	1.000000	0.063830	1.000000	0.857143	0	0	0	0	0	0
4	0.833333	0.106383	0.820312	0.857143	0	0	0	0	0	0

```
In [36]: kdrama_names['Name'].loc[23]='Kingdom'

In [38]: new_df.head()

Out[38]:
```

	Year of release	Number of Episode	Duration	Rating	Network_MBC	Network_Netflix	Network_Others	Network_SBS	Network_jTBC
Move to Heaven	1.000000	0.042553	0.312500	1.000000	0	1	0	0	0
Hospital Playlist	0.944444	0.063830	0.921875	0.857143	0	0	0	0	0
Flower of Evil	0.944444	0.106383	0.765625	0.857143	0	0	0	0	0
Hospital Playlist 2	1.000000	0.063830	1.000000	0.857143	0	0	0	0	0
My Mister	0.833333	0.106383	0.820312	0.857143	0	0	0	0	0

```
In [41]: def getRecommendation_dramas_for(drama_name, no_of_recommend=5, get_similarity_rate=False):
    kn = NearestNeighbors(n_neighbors=no_of_recommend+1, metric='manhattan')
    kn.fit(new_df)

    distances, indices = kn.kneighbors(new_df.loc[drama_name])

    print(f'Similar K-Dramas for "{drama_name[0]}"')
    nearest_dramas = [kdrama_names.loc[i][0] for i in indices.flatten()[1:]]
    if not get_similarity_rate:
        return nearest_dramas
    sim_rates = []
    for drama in nearest_dramas:
        synopsis = nearest_dramas.loc[drama][8]
        sim = cosine_similarity(new_df.loc[drama_name], [new_df.loc[drama]]).flatten()
        sim_rates.append(sim[0])
    recommended_dramas = pd.DataFrame({'Recommended Drama': nearest_dramas, 'Similarity': sim_rates, 'Synopsis': synopsis})
    recommended_dramas.sort_values(by='Similarity', ascending=True)
    return recommended_dramas

In [43]: kdrama = kdrama_names.loc[0]
kdrama

Out[43]: Name      Move to Heaven
      dtype: object

In [48]: getRecommendation_dramas_for(kdrama, no_of_recommend=5)

Out[48]: ['Kingdom', 'Kingdom', 'My Name', 'Sweet Home', 'Squid Game']

In [46]: rd2 = kdrama_names.loc[10]
rd2

Out[46]: Name      Signal
      dtype: object

In [49]: getRecommendation_dramas_for(rd2, no_of_recommend=5)

Out[49]: ['It's Okay to Not Be Okay',
      'Stranger',
      'Crash Landing on You',
      'My Mister',
      'Reply 1988']

In [ ]:
```