Women cloth reviews prediction

Objective The project involves creating a predictive model for women's clothing reviews using the Multinomial Naïve Bayes algorithm, a probabilistic learning approach commonly employed in Natural Language Processing and ideal for text classification with discrete features.

Importing Libraries

IMPORT DEPENDENCIES

```
!pip3 install -q numpy pandas matplotlib plotly wordcloud scikit-learn
```

```
import string
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
from wordcloud import WordCloud
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np
from scipy.sparse import save_npz
import warnings
warnings.filterwarnings('ignore')
```

New Section

```
netflix_data = pd.read_csv("netflix.csv")
netflix_data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

Next steps: View recommended plots

DATA PRE-PROCESSING AND EDA

```
netflix_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
netflix_data.isnull().sum()
```

```
show_id           0
type              0
title             0
director       2634
cast            825
country         831
date_added       10
release_year      0
rating            4
duration          3
listed_in         0
description       0
dtype: int64
```

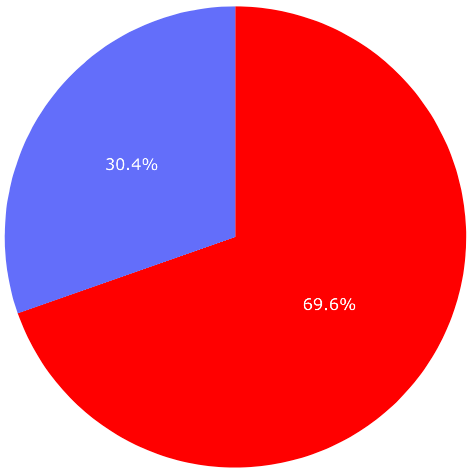
```
netflix_data.fillna('', inplace=True)
```

```
netflix_data.describe(include='all').T
```

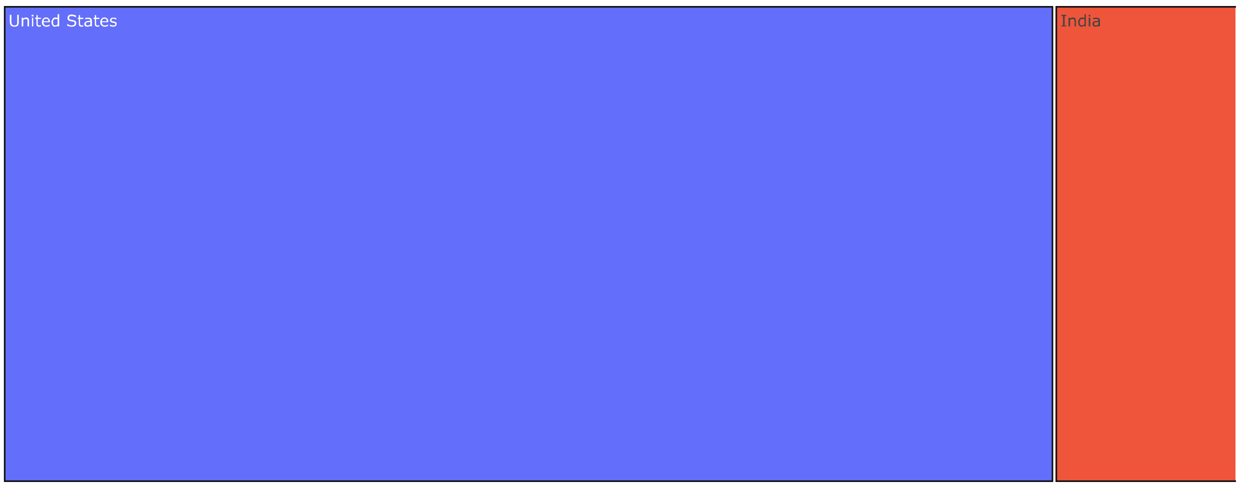
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
show_id	8807	8807	s1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
type	8807	2	Movie	6131	NaN	NaN	NaN	NaN	NaN	NaN	NaN
title	8807	8804	15-Aug	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
director	8807	4529		2634	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cast	8807	7693		825	NaN	NaN	NaN	NaN	NaN	NaN	NaN
country	8807	749	United States	2818	NaN	NaN	NaN	NaN	NaN	NaN	NaN
date_added	8807	1768	January 1, 2020	109	NaN	NaN	NaN	NaN	NaN	NaN	NaN
release_year	8807.0	NaN	NaN	NaN	2014.180198	8.819312	1925.0	2013.0	2017.0	2019.0	2021.0
rating	8807	18	TV-MA	3207	NaN	NaN	NaN	NaN	NaN	NaN	NaN
duration	8807	221	1 Season	1793	NaN	NaN	NaN	NaN	NaN	NaN	NaN
listed_in	8807	514	Dramas, International Movies	362	NaN	NaN	NaN	NaN	NaN	NaN	NaN
description	8807	8775	Paranormal activity at a lush, abandoned prope...	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
movie_counts = netflix_data['release_year'].value_counts().sort_index()
fig = go.Figure(data=go.Bar(x=movie_counts.index, y=movie_counts.values))
fig.update_layout(
    plot_bgcolor='rgb(17, 17, 17)',
    paper_bgcolor='rgb(17, 17, 17)',
    font_color='white',
    title='Number of Movies Released Each Year',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Number of Movies')
)
fig.update_traces(marker_color='red')
fig.show()
```

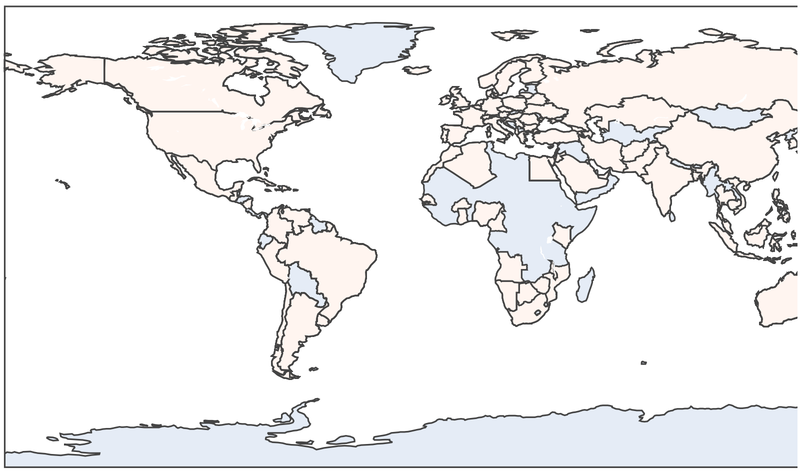
```
movie_type_counts = netflix_data['type'].value_counts()

fig = go.Figure(data=go.Pie(labels=movie_type_counts.index, values=movie_type_counts.values))

fig.update_layout(
    plot_bgcolor='rgb(17, 17, 17)',
    paper_bgcolor='rgb(17, 17, 17)',
    font_color='white',
    title='Distribution of C. Types',
)
fig.update_traces(marker=dict(colors=['red']))
fig.show()
```



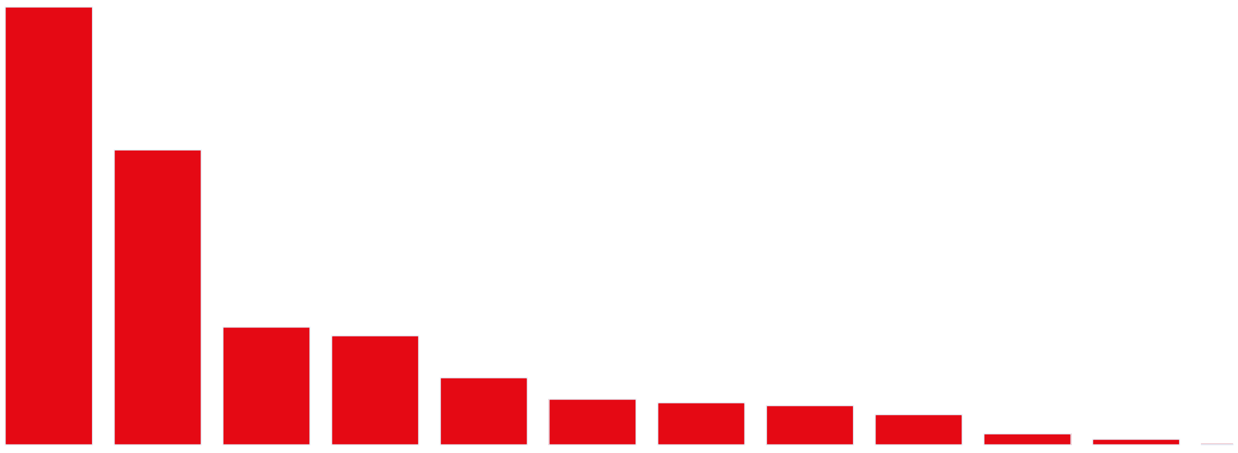
```
top_countries = netflix_data['country'].value_counts().head(10)

fig = px.treemap(names=top_countries.index, parents=["" for _ in top_countries.index], values=top_countries.values)

fig.update_layout(
    plot_bgcolor='rgb(17, 17, 17)',
    paper_bgcolor='rgb(17, 17, 17)',
    font_color='white',
    title='Top Countries with Highest Number of Movies',
)
fig.show()
```



```
country_movie_counts = netflix_data['country'].value_counts()

data = pd.DataFrame({'Country': country_movie_counts.index, 'Movie Count': country_movie_counts.values})

fig = px.choropleth(data_frame=data, locations='Country', locationmode='country names',
                    color='Movie Count', title='Number of Movies Released By Country',
                    color_continuous_scale='Reds', range_color=(0, data['Movie Count'].max()),
                    labels={'Movie Count': 'Number of Movies'})

fig.update_layout(
    plot_bgcolor='rgb(17, 17, 17)',
    paper_bgcolor='rgb(17, 17, 17)',
    font_color='white'
)
fig.show()
```

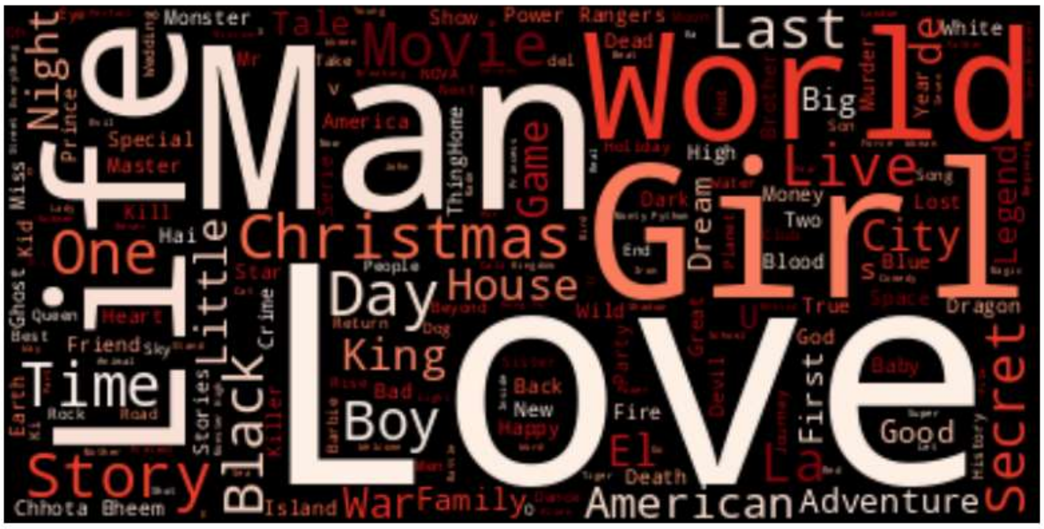


```
ratings       = list(netflix_data['rating'].value_counts().index)
rating_counts = list(netflix_data['rating'].value_counts().values)

fig = go.Figure(data=[go.Bar(
    x=ratings,
    y=rating_counts,
    marker_color='#E50914'
)])

fig.update_layout(
    title='Movie Ratings Distribution',
    xaxis_title='Rating',
    yaxis_title='Count',
    plot_bgcolor='rgba(0, 0, 0, 0)',
    paper_bgcolor='rgba(0, 0, 0, 0.7)',
    font=dict(
        color='white'
    )
)

fig.show()
```

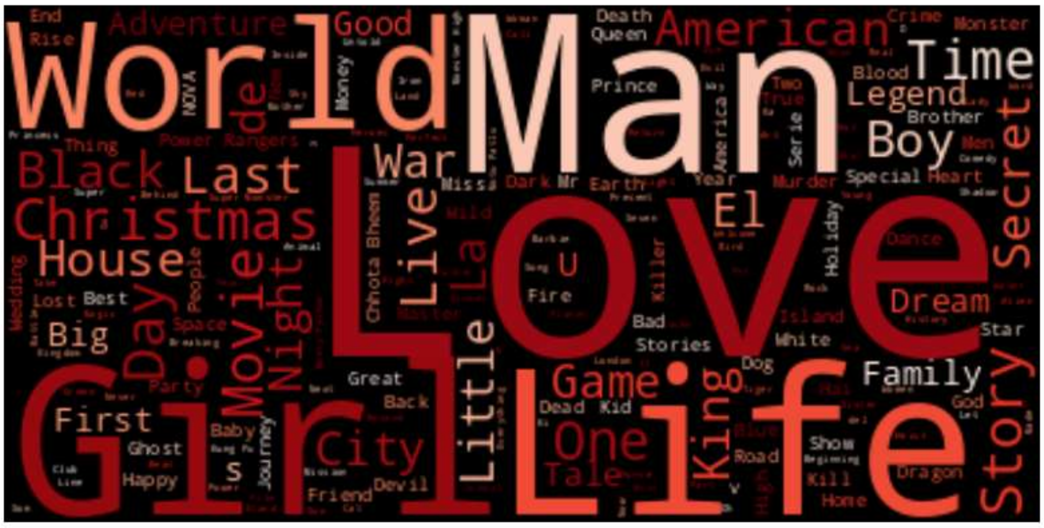


```
ratings       = list(netflix_data['duration'].value_counts().index)
rating_counts = list(netflix_data['duration'].value_counts().values)

fig = go.Figure(data=[go.Bar(
    x=ratings,
    y=rating_counts,
    marker_color='#E50914'
)])

fig.update_layout(
    title='Movie Durations Distribution',
    xaxis_title='Rating',
    yaxis_title='Count',
    plot_bgcolor='rgba(0, 0, 0, 0)',
    paper_bgcolor='rgba(0, 0, 0, 0.7)',
    font=dict(
        color='white'
    )
)

fig.show()
```

```
titles = netflix_data['title'].values

text = ' '.join(titles)

wordcloud = WordCloud(background_color='black', colormap='Reds').generate(text)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Words in Netflix Titles', color='white')
plt.show()
```



```
titles = netflix_data['title'].values

text = ' '.join(titles)

wordcloud = WordCloud(background_color='black', colormap='Reds').generate(text)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Words in Netflix Titles', color='white')
plt.show()
```



```
titles = netflix_data['listed_in'].values

text = ' '.join(titles)

wordcloud = WordCloud(background_color='black', colormap='Reds').generate(text)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Words in Netflix Descriptions', color='white')
plt.show()
```

Movies Action
Movies Dramas
Children Family
Movies Comedies
Independent Movies
International Movies
Comedies Dramas
Music Musicals
Family Movies
Romantic Movies
Movies Thrillers
TV Shows
Horror Movies
Movies International
TV Dramas
TV Comedies
Shows International
Action Adventure
Fi Fantasy
Crime TV
Dramas International
Dramas Independent
International TV
Romantic TV

```
netflix_data
```

	show_id	type	title	director	cast	country	date_added	release_y
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...		September 24, 2021	2
3	s4	TV Show	Jailbirds New Orleans				September 24, 2021	2
4	s5	TV Show	Kota Factory		Mayur More, Jitendra Kumar, Ranjan	India	September 24, 2021	2

Next steps: View recommended plots

FEATURE ENGINEERING

```
new_data = netflix_data[['title', 'type', 'director', 'cast', 'rating', 'listed_in', 'description']]
new_data.set_index('title', inplace=True)
```

```
new_data.head()
```

	type	director	cast	rating	listed_in	description
title						
Dick Johnson Is Dead	Movie	Kirsten Johnson		PG-13	Documentaries	As her father nears the end of his life, filmm...
Blood & Water	TV Show		Ama Qamata, Khosi Ngema, Gail	TV-MA	International TV Shows, TV Dramas, TV	After crossing paths at a party, a Cape

Next steps: View recommended plots

```
class TextCleaner:
    def separate_text(self, texts):
        unique_texts = set()
        for text in texts.split(','):
            unique_texts.add(text.strip().lower())
        return ' '.join(unique_texts)

    def remove_space(self, texts):
        return texts.replace(' ', '').lower()

    def remove_punc(self, texts):
        texts = texts.lower()
        texts = texts.translate(str.maketrans('', '', string.punctuation))
        return ' '.join(texts.split())

    def clean_text(self, texts):
        texts = self.separate_text(texts)
        texts = self.remove_space(texts)
        texts = self.remove_punc(texts)
        return texts
```

```
cleaner = TextCleaner()
```

```
new_data['type']        = new_data['type'].apply(cleaner.remove_space)
new_data['director']    = new_data['director'].apply(cleaner.separate_text)
new_data['cast']        = new_data['cast'].apply(cleaner.separate_text)
new_data['rating']      = new_data['rating'].apply(cleaner.remove_space)
new_data['listed_in']   = new_data['listed_in'].apply(cleaner.separate_text)
new_data['description'] = new_data['description'].apply(cleaner.remove_punc)
```

```
new_data.head()
```

	type	director	cast	rating	listed_in	description
title						
Dick Johnson Is Dead	movie	kirsten johnson		pg-13	documentaries	as her father nears the end of his life filmma...
Blood & Water	tvshow		khosi ngema mekaila mathys ryle de morny	tv-ma	tv dramas tv mysteries international tv shows	after crossing paths at a party a cape town te...

Next steps: View recommended plots

```
new_data['BoW'] = new_data.apply(lambda row: ' '.join(row.dropna().values), axis=1)
new_data.drop(new_data.columns[:-1], axis=1, inplace=True)
```

```
new_data.head()
```

	BoW
title	
Dick Johnson Is Dead	movie kirsten johnson pg-13 documentaries as ...
Blood & Water	tvshow khosi ngema mekaila mathys ryle de mor...
Ganglands	tvshow julien leclercq sami bouajila noureddin...
Jailbirds New Orleans	tvshow tv-ma reality tv docuseries feuds fli...
Kota Factory	tvshow jitendra kumar ahsaas channa alam khan...

Next steps: View recommended plots

```
tfid = TfidfVectorizer()
tfid_matrix = tfid.fit_transform(new_data['BoW'])
```

```
cosine_sim = cosine_similarity(tfid_matrix, tfid_matrix)
cosine_sim
```

```
array([[1.        , 0.00504833, 0.02011193, ..., 0.01065369, 0.02109898,
        0.03048859],
       [0.00504833, 1.        , 0.01714561, ..., 0.00103121, 0.        ,
        0.00481712],
       [0.02011193, 0.01714561, 1.        , ..., 0.00560911, 0.01042642,
        0.0333502 ],
       ...,
       [0.01065369, 0.00103121, 0.00560911, ..., 1.        , 0.05649084,
        0.00600011],
       [0.02109898, 0.        , 0.01042642, ..., 0.05649084, 1.        ,
        0.01046521],
       [0.03048859, 0.00481712, 0.0333502 , ..., 0.00600011, 0.01046521,
```

```
cosine_sim
```

```
array([[1.        , 0.00504833, 0.02011193, ..., 0.01065369, 0.02109898,
        0.03048859],
       [0.00504833, 1.        , 0.01714561, ..., 0.00103121, 0.        ,
        0.00481712],
       [0.02011193, 0.01714561, 1.        , ..., 0.00560911, 0.01042642,
        0.0333502 ],
       ...,
       [0.01065369, 0.00103121, 0.00560911, ..., 1.        , 0.05649084,
        0.00600011],
       [0.02109898, 0.        , 0.01042642, ..., 0.05649084, 1.        ,
        0.01046521],
       [0.03048859, 0.00481712, 0.0333502 , ..., 0.00600011, 0.01046521,
        1.        ]])
```

```
np.save('tfidf_matrix.npy', tfid_matrix)
np.save('cosine_sim_matrix.npy', cosine_sim)
```

```
with open('tfidf_vectorizer.pkl', 'wb') as f:
    pickle.dump(tfid, f)
```

```
final_data = netflix_data[['title', 'type']]
```

```
final_data.head()
```

	title	type
0	Dick Johnson Is Dead	Movie
1	Blood & Water	TV Show
2	Ganglands	TV Show
3	Jailbirds New Orleans	TV Show
4	Kota Factory	TV Show

Next steps: View recommended plots

```
final_data.to_csv('movie_data.csv',index=False)
```

_ Movie Recommendation System 🎬 (FLIX-HUB) _

```
import re
class FlixHub:
    def __init__(self, df, cosine_sim):
        self.df = df
        self.cosine_sim = cosine_sim
    def recommendation(self, title, total_result=5, threshold=0.5):
        idx = self.find_id(title)
        self.df['similarity'] = self.cosine_sim[idx]
        sort_df = self.df.sort_values(by='similarity', ascending=False)[1:total_result+1]

        movies = sort_df['title'][sort_df['type'] == 'Movie']
        tv_shows = sort_df['title'][sort_df['type'] == 'TV Show']

        similar_movies = []
        similar_tv_shows = []
```