```
pip install pandas numpy scikit-learn matplotlib

Requirement already satisfied: pandas in c:\users\vasav\appdata\local\
programs\python\python311\lib\site-packages (2.2.2)
Requirement already satisfied: numpy in c:\users\vasav\appdata\local\
programs\python\python311\lib\site-packages (2.0.0)
Requirement already satisfied: scikit-learn in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (1.5.0)
Requirement already satisfied: matplotlib in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (3.9.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
vasav\appdata\local\programs\python\python311\lib\site-packages (from
pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (from pandas)
(2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
pandas) (2024.1)
Requirement already satisfied: scipy>=1.6.0 in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (from scikit-learn)
(1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (from matplotlib)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (4.53.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (from matplotlib)
(10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (from python-
dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip is available: 23.2.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\vasav\appdata\local\
programs\python\python311\lib\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\vasav\appdata\local\
programs\python\python311\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\vasav\appdata\local\
programs\python\python311\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\vasav\
appdata\local\programs\python\python311\lib\site-packages (from nltk)
(2024.7.24)
Requirement already satisfied: tqdm in c:\users\vasav\appdata\local\
programs\python\python311\lib\site-packages (from nltk) (4.66.4)
Requirement already satisfied: colorama in c:\users\vasav\appdata\
local\programs\python\python311\lib\site-packages (from click->nltk)
(0.4.6)


[notice] A new release of pip is available: 23.2.1 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```python
import numpy as np
import pandas as pd
import nltk

# Set seed for reproducibility
np.random.seed(5)

# Read in IMDb and Wikipedia movie data (both in same file)
movies_df = pd.read_csv("movies.csv")

print("Number of movies loaded: %s " % (len(movies_df)))

# Display the data
movies_df
```

```
Number of movies loaded: 100
```

```
    rank                     title  \
0      0             The Godfather
1      1  The Shawshank Redemption
2      2          Schindler's List
3      3               Raging Bull
```

```
4      4                Casablanca
..   ...                       ...
95    95     Rebel Without a Cause
96    96               Rear Window
97    97             The Third Man
98    98        North by Northwest
99    99       Yankee Doodle Dandy

                                             genre  \
0                       [u' Crime', u' Drama']
1                       [u' Crime', u' Drama']
2      [u' Biography', u' Drama', u' History']
3        [u' Biography', u' Drama', u' Sport']
4            [u' Drama', u' Romance', u' War']
..                                         ...
95                                 [u' Drama']
96                  [u' Mystery', u' Thriller']
97  [u' Film-Noir', u' Mystery', u' Thriller']
98                  [u' Mystery', u' Thriller']
99     [u' Biography', u' Drama', u' Musical']

                                            wiki_plot  \
0   On the day of his only daughter's wedding, Vit...
1   In 1947, banker Andy Dufresne is convicted of ...
2   In 1939, the Germans move Polish Jews into the...
3   In a brief scene in 1964, an aging, overweight...
4   It is early December 1941. American expatriate...
..                                                ...
95  \r\n\r\n\r\n\r\nJim Stark is in police custody...
96  \r\n\r\n\r\n\r\nJames Stewart as L.B. Jefferie...
97  \r\n\r\n\r\n\r\nSocial network mapping all maj...
98  Advertising executive Roger O. Thornhill is mi...
99   \r\n  In the early days of World War II, Coha...

                                            imdb_plot
0   In late summer 1945, guests are gathered for t...
1   In 1947, Andy Dufresne (Tim Robbins), a banker...
2   The relocation of Polish Jews from surrounding...
3   The film opens in 1964, where an older and fat...
4   In the early years of World War II, December 1...
..                                                ...
95  Shortly after moving to Los Angeles with his p...
96  L.B. "Jeff" Jeffries (James Stewart) recuperat...
97  Sights of Vienna, Austria, flash across the sc...
98  At the end of an ordinary work day, advertisin...
99                                                NaN

[100 rows x 5 columns]
```

```
movies_df["plot"] = movies_df["wiki_plot"].astype(str) + "\n" + \
                    movies_df["imdb_plot"].astype(str)

# Inspect the new DataFrame
movies_df.head()
```

```
   rank                     title
genre  \
0     0             The Godfather                   [u' Crime', u'
Drama']
1     1  The Shawshank Redemption                   [u' Crime', u'
Drama']
2     2          Schindler's List  [u' Biography', u' Drama', u'
History']
3     3               Raging Bull    [u' Biography', u' Drama', u'
Sport']
4     4                Casablanca        [u' Drama', u' Romance', u'
War']

                                           wiki_plot  \
0  On the day of his only daughter's wedding, Vit...
1  In 1947, banker Andy Dufresne is convicted of ...
2  In 1939, the Germans move Polish Jews into the...
3  In a brief scene in 1964, an aging, overweight...
4  It is early December 1941. American expatriate...

                                           imdb_plot  \
0  In late summer 1945, guests are gathered for t...
1  In 1947, Andy Dufresne (Tim Robbins), a banker...
2  The relocation of Polish Jews from surrounding...
3  The film opens in 1964, where an older and fat...
4  In the early years of World War II, December 1...

                                                plot
0  On the day of his only daughter's wedding, Vit...
1  In 1947, banker Andy Dufresne is convicted of ...
2  In 1939, the Germans move Polish Jews into the...
3  In a brief scene in 1964, an aging, overweight...
4  It is early December 1941. American expatriate...
```

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\vasav\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!

True
```

```
sent_tokenized = [sent for sent in nltk.sent_tokenize("""
                        Today (May 19, 2016) is his only daughter's
wedding.
                        Vito Corleone is the Godfather.
                        """)]

# Word Tokenize first sentence from sent_tokenized, save as
words_tokenized
words_tokenized = [word for word in
nltk.word_tokenize(sent_tokenized[0])]

# Remove tokens that do not contain any letters from words_tokenized
import re

filtered = [word for word in words_tokenized if re.search('[a-zA-Z]',
word)]

# Display filtered words to observe words after tokenization
filtered
```

```
['Today', 'May', 'is', 'his', 'only', 'daughter', "'s", 'wedding']
```

```
from nltk.stem.snowball import SnowballStemmer

# Create an English language SnowballStemmer object
stemmer = SnowballStemmer("english")

# Print filtered to observe words without stemming
print("Without stemming: ", filtered)

# Stem the words from filtered and store in stemmed_words
stemmed_words = [stemmer.stem(t) for t in filtered]

# Print the stemmed_words to observe words after stemming
print("After stemming:   ", stemmed_words)
```

```
Without stemming:  ['Today', 'May', 'is', 'his', 'only', 'daughter',
"'s", 'wedding']
After stemming:    ['today', 'may', 'is', 'his', 'onli', 'daughter',
"'s", 'wed']
```

```
def tokenize_and_stem(text):

    # Tokenize by sentence, then by word
    tokens = [word for sent in nltk.sent_tokenize(text) for word in
nltk.word_tokenize(sent)]

    # Filter out raw tokens to remove noise
    filtered_tokens = [token for token in tokens if re.search('[a-zA-
Z]', token)]

    # Stem the filtered_tokens
    stems = [stemmer.stem(t) for t in filtered_tokens]
    return stems

words_stemmed = tokenize_and_stem("Today (May 19, 2016) is his only
daughter's wedding.")
print(words_stemmed)
```

```
['today', 'may', 'is', 'his', 'onli', 'daughter', "'s", 'wed']
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Instantiate TfidfVectorizer object with stopwords and tokenizer
# parameters for efficient processing of text
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=200000,
                                 min_df=0.2, stop_words='english',
                                 use_idf=True,
tokenizer=tokenize_and_stem,
                                 ngram_range=(1,3))

tfidf_matrix = tfidf_vectorizer.fit_transform([x for x in
movies_df["plot"]])

print(tfidf_matrix.shape)
```

```
C:\Users\vasav\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\feature_extraction\text.py:523: UserWarning: The
parameter 'token_pattern' will not be used since 'tokenizer' is not
None'
  warnings.warn(
C:\Users\vasav\AppData\Local\Programs\Python\Python311\Lib\site-
packages\sklearn\feature_extraction\text.py:408: UserWarning: Your
stop_words may be inconsistent with your preprocessing. Tokenizing the
stop words generated tokens ['abov', 'afterward', 'alon', 'alreadi',
'alway', 'ani', 'anoth', 'anyon', 'anyth', 'anywher', 'becam',
'becaus', 'becom', 'befor', 'besid', 'cri', 'describ', 'dure', 'els',
'elsewher', 'empti', 'everi', 'everyon', 'everyth', 'everywher',
'fifti', 'forti', 'henc', 'hereaft', 'herebi', 'howev', 'hundr',
'inde', 'mani', 'meanwhil', 'moreov', 'nobodi', 'noon', 'noth',
'nowher', 'onc', 'onli', 'otherwis', 'ourselv', 'perhap', 'pleas',
'sever', 'sinc', 'sincer', 'sixti', 'someon', 'someth', 'sometim',
'somewher', 'themselv', 'thenc', 'thereaft', 'therebi', 'therefor',
'togeth', 'twelv', 'twenti', 'veri', 'whatev', 'whenc', 'whenev',
'wherea', 'whereaft', 'wherebi', 'wherev', 'whi', 'yourselv'] not in
stop_words.
  warnings.warn(
```

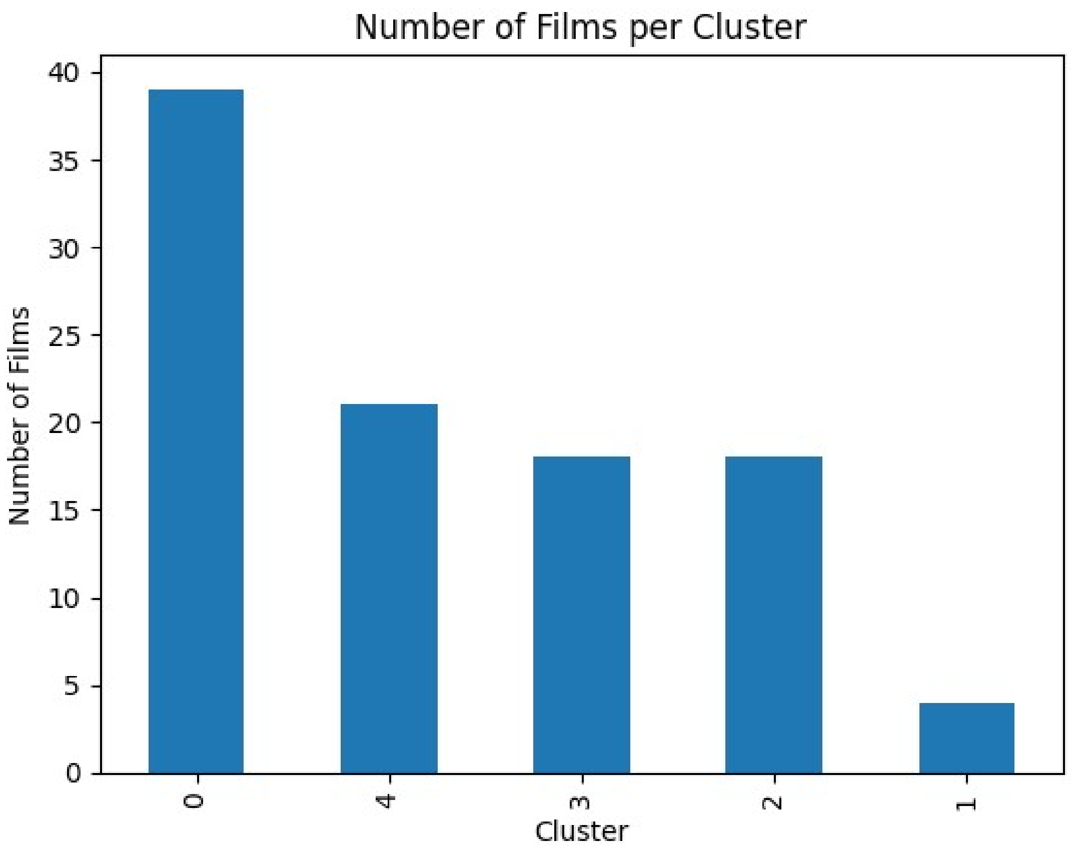
```
(100, 564)
```

```
# Import necessary libraries
from sklearn.cluster import KMeans
import pandas as pd
```

```
# Assuming tfidf_matrix and movies_df are already defined

# Create a KMeans object with 5 clusters and fit the model
km = KMeans(n_clusters=5)
km.fit(tfidf_matrix)

# Get the cluster labels for each movie
clusters = km.labels_.tolist()

# Add the cluster information to the DataFrame
movies_df["cluster"] = clusters

# Display the number of films per cluster
cluster_counts = movies_df['cluster'].value_counts()
print(cluster_counts)

# Plot the counts for better visualization
import matplotlib.pyplot as plt

cluster_counts.plot(kind='bar')
plt.title('Number of Films per Cluster')
plt.xlabel('Cluster')
plt.ylabel('Number of Films')
plt.show()

# Function to print movies in each cluster
def print_movies_in_clusters(df, cluster_label_column='cluster',
title_column='title'):
    for cluster in sorted(df[cluster_label_column].unique()):
        print(f"\nCluster {cluster}:")
        movies_in_cluster = df[df[cluster_label_column] == cluster]
[title_column].tolist()
        for movie in movies_in_cluster:
            print(movie)

# Print the names of the movies in each cluster
print_movies_in_clusters(movies_df)

# Function to find similar movies in the same cluster
def find_similar_movies(movie_title, df,
cluster_label_column='cluster', title_column='title'):
    # Check if the movie exists in the DataFrame
    if movie_title not in df[title_column].values:
        return f"Movie '{movie_title}' not found in the dataset."

    # Get the cluster label of the given movie
    cluster_label = df[df[title_column] == movie_title]
[cluster_label_column].values[0]
```

```
    # Get all movies in the same cluster
    similar_movies = df[df[cluster_label_column] == cluster_label]
[title_column].tolist()

    # Remove the given movie from the list
    if movie_title in similar_movies:
        similar_movies.remove(movie_title)

    return similar_movies

    # Example usage: Prompt user for a movie title and print similar
movies
movie_title = input("Enter a movie title: ")  # User input for movie
title
similar_movies = find_similar_movies(movie_title, movies_df)

if isinstance(similar_movies, list):
    print(f"\nMovies similar to '{movie_title}':")
    for movie in similar_movies:
        print(movie)
else:
    print(similar_movies)
```

```
cluster
0    39
4    21
3    18
2    18
1     4
Name: count, dtype: int64
```



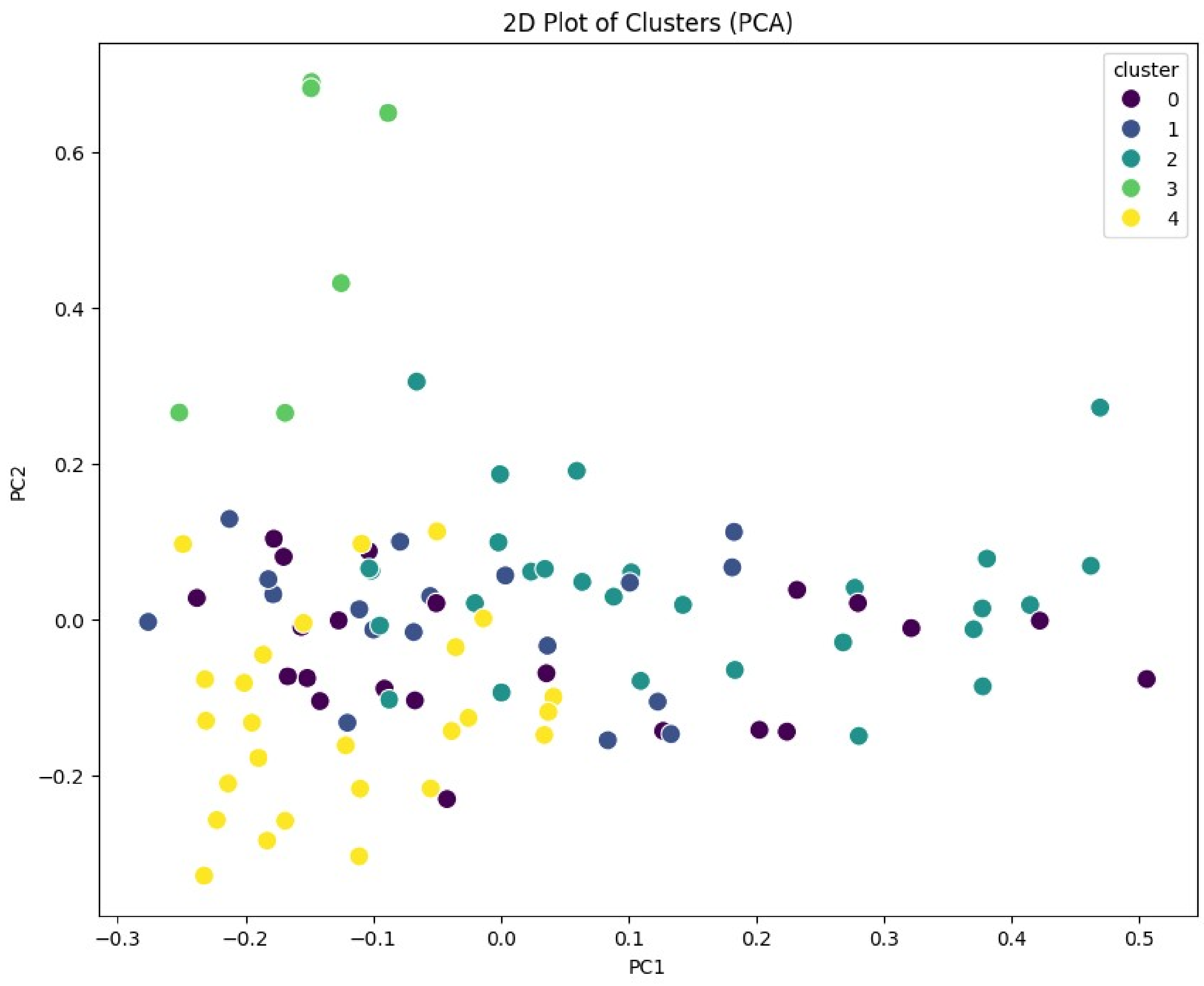
```
Cluster 0:
The Godfather
Psycho
Sunset Blvd.
Vertigo
On the Waterfront
West Side Story
The Silence of the Lambs
Some Like It Hot
Unforgiven
Rocky
A Streetcar Named Desire
An American in Paris
Butch Cassidy and the Sundance Kid
The Treasure of the Sierra Madre
The Apartment
High Noon
Goodfellas
The French Connection
Rain Man
Out of Africa
Good Will Hunting
```

```
Fargo
Giant
The Grapes of Wrath
Shane
The Green Mile
Close Encounters of the Third Kind
Network
American Graffiti
Pulp Fiction
Stagecoach
The Maltese Falcon
A Clockwork Orange
Taxi Driver
Double Indemnity
Rebel Without a Cause
Rear Window
The Third Man
North by Northwest

Cluster 1:
It's a Wonderful Life
The Philadelphia Story
The King's Speech
A Place in the Sun

Cluster 2:
One Flew Over the Cuckoo's Nest
Gone with the Wind
The Wizard of Oz
Titanic
Forrest Gump
E.T. the Extra-Terrestrial
2001: A Space Odyssey
Chinatown
12 Angry Men
To Kill a Mockingbird
My Fair Lady
Ben-Hur
Doctor Zhivago
Braveheart
The Exorcist
City Lights
It Happened One Night
Yankee Doodle Dandy

Cluster 3:
Raging Bull
Citizen Kane
The Godfather: Part II
The Sound of Music
```

```
Singin' in the Rain
Amadeus
Gandhi
The Best Years of Our Lives
The Good, the Bad and the Ugly
Midnight Cowboy
Mr. Smith Goes to Washington
Annie Hall
Terms of Endearment
Tootsie
Nashville
The Graduate
The African Queen
Wuthering Heights

Cluster 4:
The Shawshank Redemption
Schindler's List
Casablanca
Lawrence of Arabia
Star Wars
The Bridge on the River Kwai
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb
Apocalypse Now
The Lord of the Rings: The Return of the King
Gladiator
From Here to Eternity
Saving Private Ryan
Raiders of the Lost Ark
Patton
Jaws
Platoon
Dances with Wolves
The Pianist
The Deer Hunter
All Quiet on the Western Front
Mutiny on the Bounty

Enter a movie title:  Braveheart


Movies similar to 'Braveheart':
One Flew Over the Cuckoo's Nest
Gone with the Wind
The Wizard of Oz
Titanic
Forrest Gump
E.T. the Extra-Terrestrial
2001: A Space Odyssey
Chinatown
```

```
12 Angry Men
To Kill a Mockingbird
My Fair Lady
Ben-Hur
Doctor Zhivago
The Exorcist
City Lights
It Happened One Night
Yankee Doodle Dandy
```

```python
from sklearn.decomposition import PCA
import numpy as np

# Reduce dimensionality to 2D using PCA
pca = PCA(n_components=2)
reduced_matrix = pca.fit_transform(tfidf_matrix.toarray())

# Create a DataFrame for the reduced data
reduced_df = pd.DataFrame(reduced_matrix, columns=['PC1', 'PC2'])
reduced_df['cluster'] = clusters

# Plot the 2D clusters
plt.figure(figsize=(10, 8))
sns.scatterplot(x='PC1', y='PC2', hue='cluster', data=reduced_df,
palette='viridis', s=100)
plt.title('2D Plot of Clusters (PCA)')
plt.show()
```



2D Plot of Clusters (PCA)
PC2
PC1
cluster
0
1
2
3
4
0.6
0.4
0.2
0.0
−0.2
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5