Step 1 : Loading and Exploring the Dataset ¶

In [1]:

```
import pandas as pd
# Loading the dataset
df = pd.read_csv(r"C:\Users\Ritik\Downloads\IMDB.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2199 entries, 0 to 2198
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  2199 non-null   int64 
 1   movie_id    2199 non-null   object
 2   movie_name  2199 non-null   object
 3   year        2134 non-null   object
 4   genre       2199 non-null   object
 5   overview    2199 non-null   object
 6   director    2199 non-null   object
 7   cast        2199 non-null   object
dtypes: int64(1), object(7)
memory usage: 137.6+ KB
```

In [2]:
```
df.head(11)
```

Out[2]:

	Unnamed: 0	movie_id	movie_name	year	genre	overview	director	cast
0	0	tt15354916	Jawan	2023	Action, Thriller	A high-octane action thriller which outlines t...	Atlee	Shah Rukh Khan, Nayanthara, Vijay Sethupathi, ...
1	1	tt15748830	Jaane Jaan	2023	Crime, Drama, Mystery	A single mother and her daughter who commit a ...	Sujoy Ghosh	Kareena Kapoor, Jaideep Ahlawat, Vijay Varma, ...
2	2	tt11663228	Jailer	2023	Action, Comedy, Crime	A retired jailer goes on a manhunt to find his...	Nelson Dilipkumar	Rajinikanth, Mohanlal, Shivarajkumar, Jackie S...
3	3	tt14993250	Rocky Aur Rani Kii Prem Kahaani	2023	Comedy, Drama, Family	Flamboyant Punjabi Rocky and intellectual Beng...	Karan Johar	Ranveer Singh, Alia Bhatt, Dharmendra, Shabana...
4	4	tt15732324	OMG 2	2023	Comedy, Drama	An unhappy civilian asks the court to mandate ...	Amit Rai	Pankaj Tripathi, Akshay Kumar, Yami Gautam, Pa...
5	5	tt18266472	Sukhee	2023	Drama	Much to the dismay of her husband, a middle-cl...	Sonal Joshi	Shilpa Shetty Kundra, Amit Sadh, Chaitannya Ch...
6	6	tt18561736	The Great Indian Family	2023	Family	Ved Vyas Tripathi, aka Bhajan Kumar, is a devo...	Vijay Krishna Acharya	Alka Amin, Bhuvan Arora, Manushi Chhillar, Sri...
7	7	tt3691740	The BFG	2016	Adventure, Family, Fantasy	An orphan little girl befriends a benevolent g...	Steven Spielberg	Mark Rylance, Ruby Barnhill, Penelope Wilton, ...
8	8	tt12844910	Pathaan	2023	Action, Adventure, Thriller	An Indian agent races against a doomsday clock...	Siddharth Anand	Shah Rukh Khan, Deepika Padukone, John Abraham...
9	9	tt15464390	Mastaney	2023	Action, Drama, History	Set in 1739, Nadar Shah's undefeated army was ...	Sharan Art	Tarsem Jassar, Simi Chahal, Gurpreet Ghuggi, K...
10	10	tt1187043	3 Idiots	2009	Comedy, Drama	Two friends are searching for their long lost ...	Rajkumar Hirani	Aamir Khan, Madhavan, Mona Singh, Sharman Joshi

Step 2 : Text Preprocessing : Tokenization, Lemmatizing

In [3]:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
#from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

# Creating a lemmatizer
lemmatizer = WordNetLemmatizer()

# Function for lemmatization
def lemmatize_text(text):
    return ' '.join([lemmatizer.lemmatize(word) for word in text.split()])

# Applying lemmatization to the 'overview' column
df['Processed_Plot'] = df['overview'].apply(lemmatize_text)
```

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

In [4]:
```
#displaying the Processed_Plot
df.head(6)
```

Out[4]:

	Unnamed: 0	movie_id	movie_name	year	genre	overview	director	cast	Processed_Plot
0	0	tt15354916	Jawan	2023	Action, Thriller	A high-octane action thriller which outlines t...	Atlee	Shah Rukh Khan, Nayanthara, Vijay Sethupathi, ...	A high-octane action thriller which outline th...
1	1	tt15748830	Jaane Jaan	2023	Crime, Drama, Mystery	A single mother and her daughter who commit a ...	Sujoy Ghosh	Kareena Kapoor, Jaideep Ahlawat, Vijay Varma, ...	A single mother and her daughter who commit a ...
2	2	tt11663228	Jailer	2023	Action, Comedy, Crime	A retired jailer goes on a manhunt to find his...	Nelson Dilipkumar	Rajinikanth, Mohanlal, Shivarajkumar, Jackie S...	A retired jailer go on a manhunt to find his s...
3	3	tt14993250	Rocky Aur Rani Kii Prem Kahaani	2023	Comedy, Drama, Family	Flamboyant Punjabi Rocky and intellectual Beng...	Karan Johar	Ranveer Singh, Alia Bhatt, Dharmendra, Shabana...	Flamboyant Punjabi Rocky and intellectual Beng...
4	4	tt15732324	OMG 2	2023	Comedy, Drama	An unhappy civilian asks the court to mandate ...	Amit Rai	Pankaj Tripathi, Akshay Kumar, Yami Gautam, Pa...	An unhappy civilian asks the court to mandate ...
5	5	tt18266472	Sukhee	2023	Drama	Much to the dismay of her husband, a middle-cl...	Sonal Joshi	Shilpa Shetty Kundra, Amit Sadh, Chaitannya Ch...	Much to the dismay of her husband, a middle-cl...

Step 3 : Vectorize Text Data

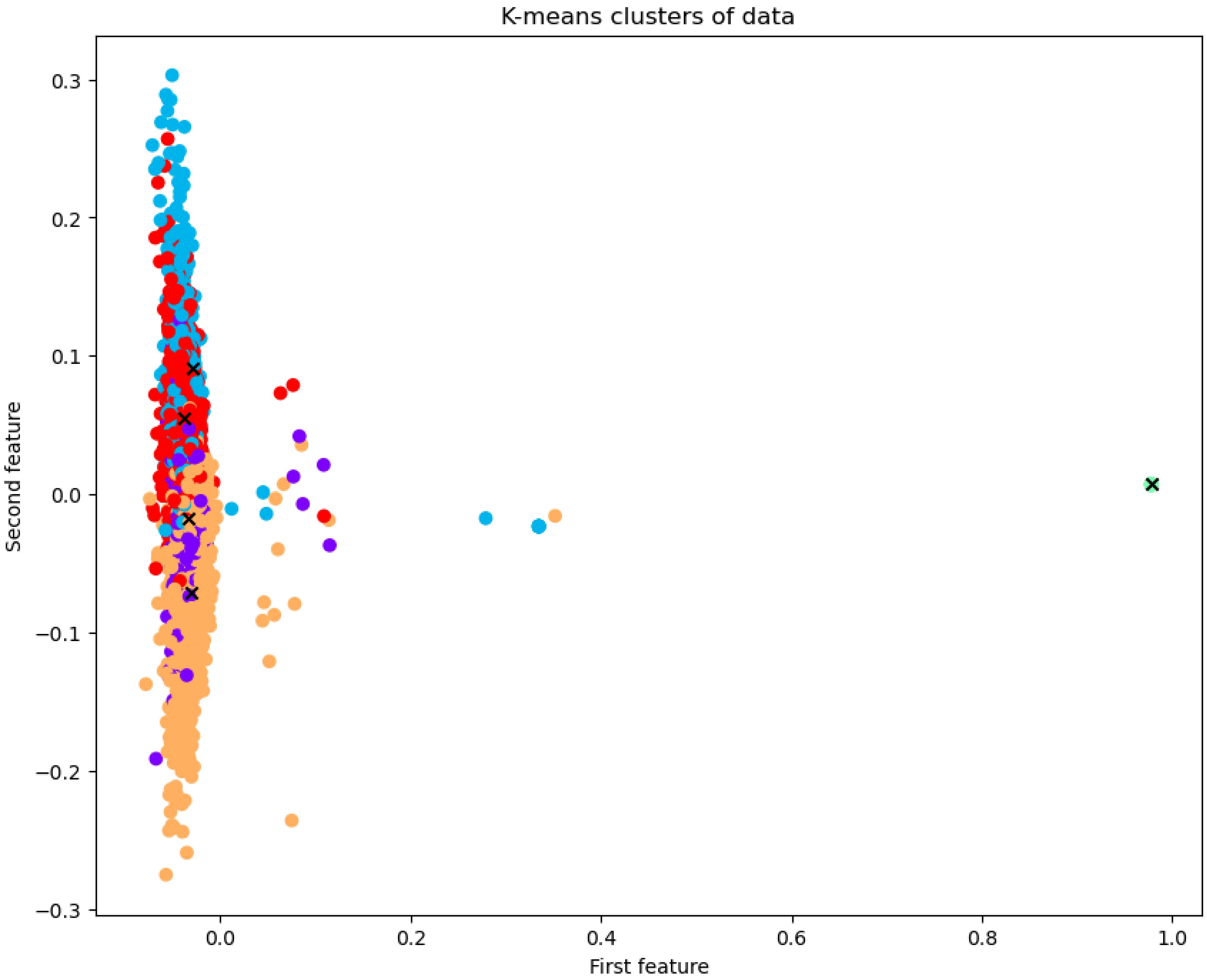
In [5]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer(max_features=30000)
tfidf_matrix = tfidf_vectorizer.fit_transform(df['Processed_Plot'])
```

Step 4 : Importing KMeans & Creating Clusters to plot it

In [6]:

```python
from sklearn.cluster import KMeans
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

kmeans = KMeans(n_clusters=5, random_state=92)
kmeans.fit(tfidf_matrix)

y_pred = kmeans.predict(tfidf_matrix)
centers_pred = kmeans.cluster_centers_

pca = PCA(n_components=2, random_state=92)
tfidf_matrix_reduced = pca.fit_transform(tfidf_matrix.toarray())
centers_reduced = pca.transform(centers_pred)

plt.figure(figsize=(10,8))
plt.scatter(tfidf_matrix_reduced[:,0], tfidf_matrix_reduced[:,1], c=y_pred, cmap='rainbow')
plt.scatter(centers_reduced[:,0], centers_reduced[:,1], c='black', marker='x')
plt.xlabel('First feature')
plt.ylabel('Second feature')
plt.title('K-means clusters of data')
plt.show()
```

```
C:\Users\Ritik\anaconda3\ane\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```


K-means clusters of data
Second feature
First feature
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.2
0.4
0.6
0.8
1.0

Step 5 : Compute Similarity

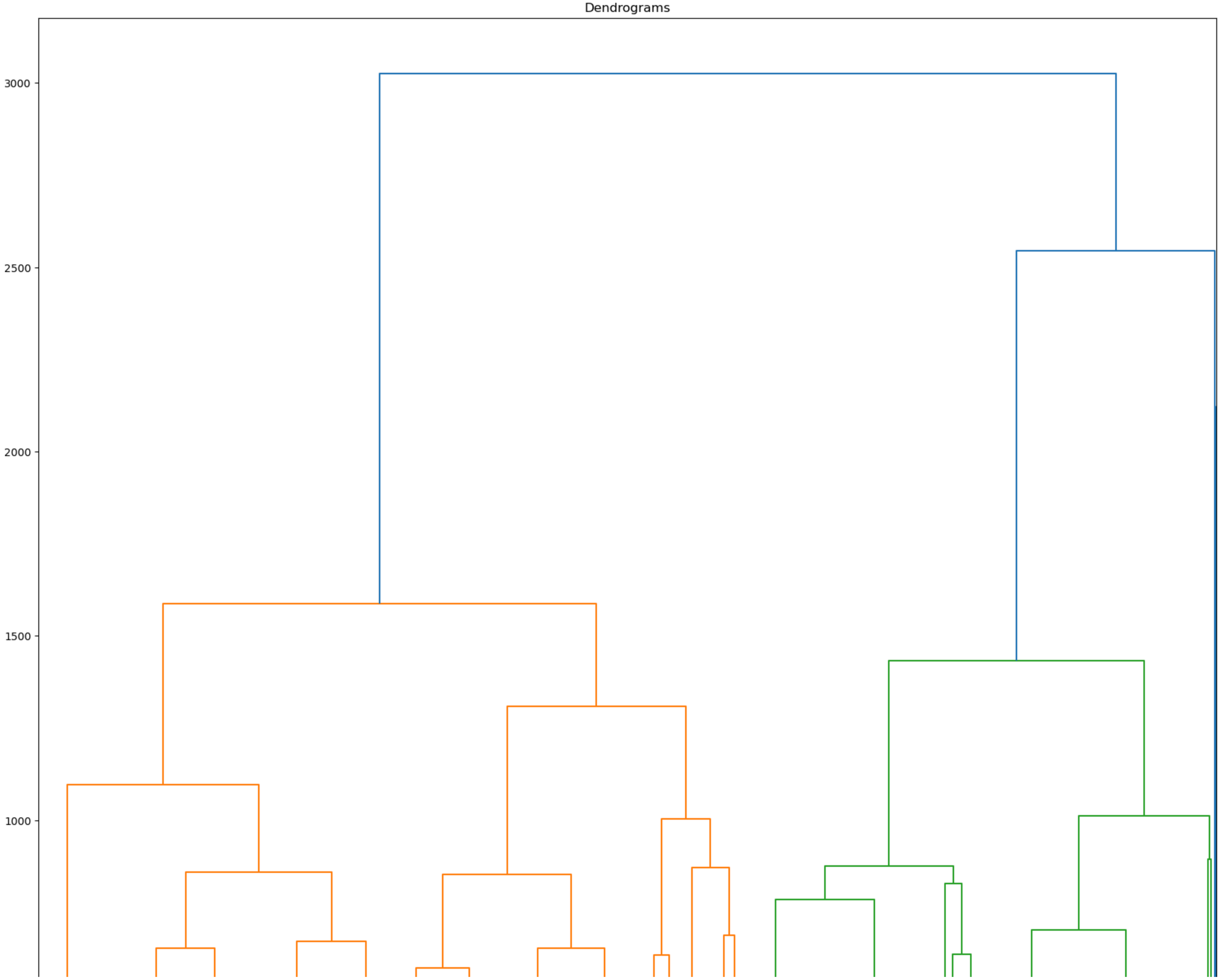
In [7]:

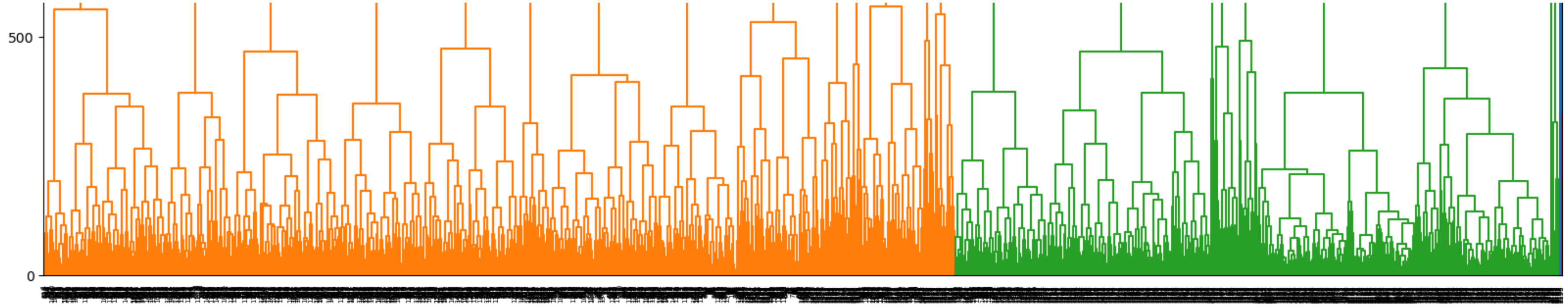
```
from sklearn.metrics.pairwise import cosine_similarity
similarity_matrix = cosine_similarity(tfidf_matrix, tfidf_matrix)

# Convert the similarity matrix to a DataFrame
similarity_df = pd.DataFrame(similarity_matrix, index=df['movie_name'], columns=df['movie_name'])
```

Step 6 : Importing Matplot, Linkage

In [8]:

```
from scipy.cluster import hierarchy
from scipy.cluster.hierarchy import linkage, dendrogram
import numpy as np
import matplotlib.pyplot as plt

# Creating mergings matrix
mergings = linkage(similarity_matrix, method='complete')

# Creating Dendrogram for our data
Z = hierarchy.linkage(mergings, method='average')

plt.figure()
fig = plt.gcf()
fig.set_size_inches(20, 20)
plt.title("Dendrograms")
# Dendrogram plotting using linkage matrix
dendrogram = hierarchy.dendrogram(Z)
```


Dendrograms
3000
2500
2000
1500
1000



Step 7 : Finding Similar Movies

For checking similarity user we will enter the movie name of different types to check we get results or not. Below given names can be tested as all are of different type :-

Copy past the names below in textbox for output

K.G.F: Chapter 1

Ramayana: The Legend of Prince Rama

M.S. Dhoni: The Untold Story

Jab Harry Met Sejal

In [11]:

```
#function for checking the movies similar
def find_similar_movies(movie_title, similarity_matrix, num_recommendations=10):
    similar_movies = similarity_matrix[movie_title].sort_values(ascending=False)[1:num_recommendations+1]
    return similar_movies

# Find similar movies for a given title
#Note : Entering the correct movie title is important so can use the above given movie list
movie_title = input("Enter the movie name : ")
similar_movies = find_similar_movies(movie_title, similarity_df)

# Displaying the result
print(f"Movies similar to '{movie_title}':\n{similar_movies}")
```

```
Enter the movie name : K.G.F: Chapter 1
Movies similar to 'K.G.F: Chapter 1':
movie_name
Thangalaan                     0.235664
Mission Majnu                  0.183996
Once Upon a Time in Mumbaai    0.162990
Trishna                        0.157698
C U at 9                       0.137888
Haré Rama Haré Krishna         0.136515
Baazi                          0.135631
Angaaray                       0.128823
Kalicharan                     0.128096
Thugs                          0.123480
Name: K.G.F: Chapter 1, dtype: float64
```

In [ ]: