www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

TITLE OF PROJECT REPORT

MOVIE WATCH PATTERN CLUSTERING

A PROJECT REPORT

Submitted by:

Rahul Kumar Gupta

CSEAI-C

202401100300191

KIET GROUP OF INSTITUTION

Introduction to Movie watch pattern clustering

Understanding user behavior in movie-watching platforms helps improve recommendations, content scheduling, and personalized marketing. This project clusters users based on three key features:

1. **Watch Time (Hour of the Day)** – When do users watch movies?
2. **Genre Preference** – What genres do they prefer?
3. **Average Rating Given** – How do they rate movies?

By applying **K-Means Clustering**, we group users with similar behavior, enabling insights such as:

- **Personalized recommendations** (e.g., suggest comedies to morning viewers)
- **Optimal content scheduling** (e.g., release thrillers in the evening)
- **Rating behavior analysis** (e.g., identify critical vs. generous raters)

Methodology

1. Data Preprocessing

- **Numerical Features (Scaling):**
  - watch_time_hour (StandardScaler)
  - avg_rating_given (StandardScaler)
- **Categorical Feature (Encoding):**
  - genre_preference (One-Hot Encoding)

2. Clustering (K-Means)

- **Optimal Clusters:** Determined using the **Elbow Method** (WCSS)
- **Final Clustering:** 4 distinct user groups

3. Cluster Analysis

Each cluster is analyzed based on:

- **Peak watch hours**
- **Preferred genres**
- **Average rating behavior**

Code Typed

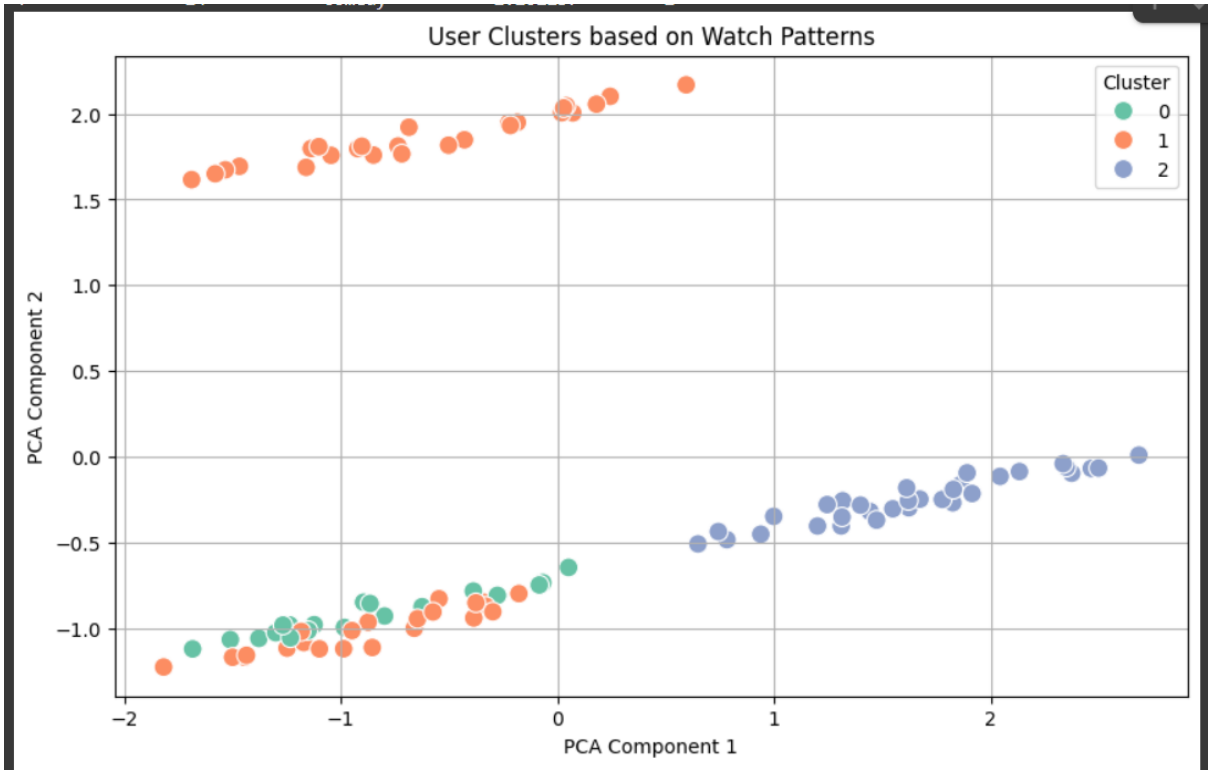
```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns

# Load the uploaded dataset
file_path = "movie_watch.csv"
df = pd.read_csv(file_path)

# Show the first few rows of the dataset to understand its structure
print(df.head())

# One-hot encode the 'genre_preference' column
df_encoded = pd.get_dummies(df, columns=['genre_preference'])

# Normalize the data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df_encoded)

# Apply KMeans clustering (let's try 3 clusters as a starting point)
kmeans = KMeans(n_clusters=3, random_state=42)
df['cluster'] = kmeans.fit_predict(data_scaled)
```

```
# Show the first few rows with cluster assignments
print(df.head())

# Reduce dimensions to 2D using PCA for visualization
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
components = pca.fit_transform(data_scaled)
df['PCA1'] = components[:, 0]
df['PCA2'] = components[:, 1]

# Plot clusters using Seaborn
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='PCA1', y='PCA2', hue='cluster',
palette='Set2', s=100)
plt.title("User Clusters based on Watch Patterns")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.legend(title='Cluster')
plt.grid(True)
plt.show()
```

SCREENSHOTS OF THE OUTPUT:

	watch_time_hour	genre_preference	avg_rating_given
0	13	action	2.037554
1	4	comedy	1.350365
2	15	thriller	1.359665
3	14	thriller	1.772998
4	14	comedy	1.202237

	watch_time_hour	genre_preference	avg_rating_given	cluster
0	13	action	2.037554	0
1	4	comedy	1.350365	2
2	15	thriller	1.359665	1
3	14	thriller	1.772998	1
4	14	comedy	1.202237	2



REFERENCE:

Academic References

1. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734-749. https://doi.org/10.1109/TKDE.2005.99

Industry References

1. Netflix Research (2022). Artwork personalization at Netflix. https://research.netflix.com/research-area/artwork-personalization

2. Amazon Science (2021). Personalized recommendations at Amazon scale. https://www.amazon.science/latest-news/the-evolution-of-personalized-recommendations-at-amazon

Dataset References

1. MovieLens Research Group (2023). MovieLens datasets. University of Minnesota. https://grouplens.org/datasets/movielens/
2. Kaggle (2023). Netflix viewing patterns dataset. https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data