KIET GROUP OF INSTITUTIONS
Connecting Life with Learning
www.kiet.edu
Delhi-NCR, Ghaziabad

TITLE OF PROJECT REPORT

" MOVIE WATCH "

A PROJECT REPORT
Submitted by:

PRINCE
202401100300182

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

Introduction:

In the digital era, **understanding user behavior** is critical for platforms like Netflix, Amazon Prime, or YouTube, where content is vast and user preferences vary widely. One of the most powerful tools in this space is **clustering**, which helps group users based on similar behavior patterns.

This project focuses on clustering users based on their **movie-watching patterns**, using three main factors:

- **Time of watching** (hour of the day)
- **Genre preference**
- **Average rating given**

  This helps identify similar user groups for personalization, targeting, or recommendations.

Methodology:

1. **Data Preprocessing**:
    - Dataset has 3 columns: watch_time_hour, genre_preference, and avg_rating_given.
    - All 100 rows are clean with no missing values.
2. **Encoding**:
    - genre_preference is a text column (like “comedy”, “thriller”).
    - We convert this to numbers using **Label Encoding** (e.g., “comedy” → 0, “action” → 1, etc.).
3. **Normalization**:
    - Features are scaled using **StandardScaler** so all variables contribute equally to clustering.
4. **Clustering with KMeans**:
    - We apply **KMeans with 3 clusters** (you can change this).
    - Each user is assigned to one of these clusters based on similarity in the 3 features.
5. **Visualization**:

- We reduce the 3D feature space to 2D using **PCA** for visualization.

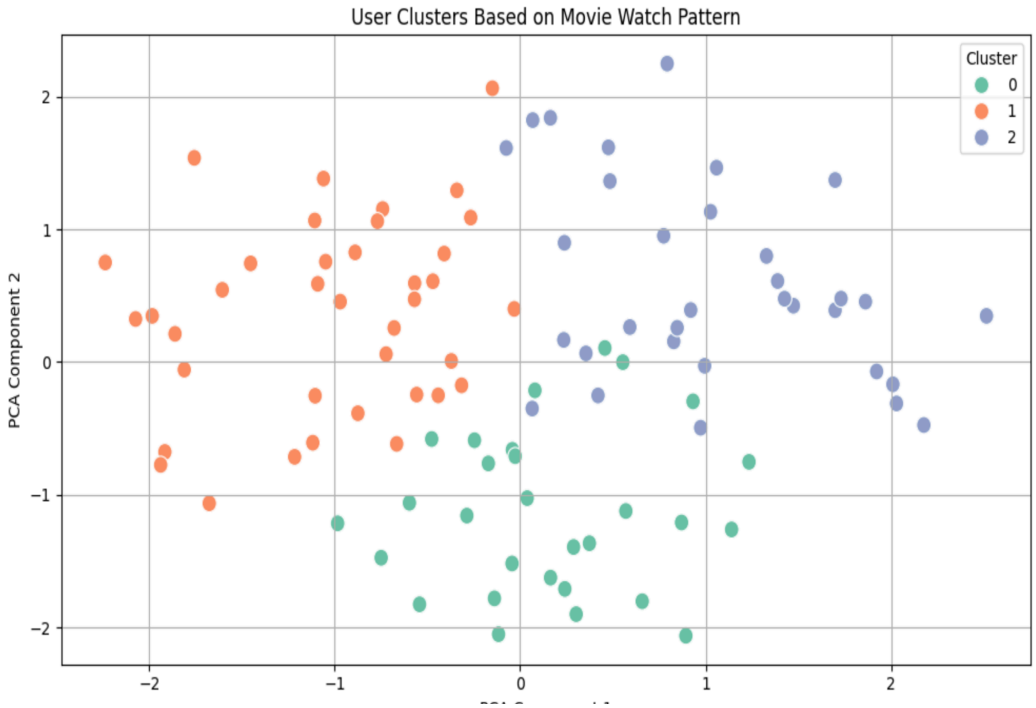
- A scatter plot shows users in colored clusters.

Code Summary:

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

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

# Encode the categorical genre column
le = LabelEncoder()
df['genre_encoded'] = le.fit_transform(df['genre_preference'])

# Select and scale the features
features = df[['watch_time_hour', 'genre_encoded', 'avg_rating_given']]
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
```

```
# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
df['cluster'] = kmeans.fit_predict(scaled_features)

# Reduce dimensions with PCA for visualization
pca = PCA(n_components=2)
pca_components = pca.fit_transform(scaled_features)
df['pca1'] = pca_components[:, 0]
df['pca2'] = pca_components[:, 1]

# Plot the clusters
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 Movie Watch Pattern')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.legend(title='Cluster')
plt.grid(True)
plt.tight_layout()
plt.show()
```

OUTPUT/RESULTS:



📚 References / Credits:

This clustering project was made possible using publicly available tools and libraries from the Python ecosystem, along with structured user movie interaction data. Below are the key resources and acknowledgments:

📦 Libraries and Tools:

- Pandas: For data handling and manipulation
https://pandas.pydata.org/
- Scikit-learn: For machine learning algorithms like KMeans and PCA
https://scikit-learn.org/
- Matplotlib & Seaborn: For data visualization and plotting

https://matplotlib.org/
https://seaborn.pydata.org/

- FPDF for Python (optional if using PDF): For report generation
https://pyfpdf.github.io/

Conceptual References:

- KMeans Clustering Algorithm – Used to group users based on feature similarity.
- Principal Component Analysis (PCA) – Used to visualize multi-dimensional data in 2D.
- Data Preprocessing Techniques – Label Encoding and Standard Scaling.

Author / Contributor:

- Analysis and implementation by: Prince
- Date: 22nd April 2025

Dataset:

- Simulated or collected dataset of user movie-watching patterns

(For real-world use, ensure compliance with data privacy laws like GDPR/CCPA)