**Movie Watch Pattern Clustering**

**Assessment Report**

**on**

**“Movie Watch Pattern Clustering”**

**submitted as partial fulfillment for the award of**

**BACHELOR OF TECHNOLOGY**

**DEGREE**

**SESSION 2024-25**

**in**

**CSE(AI) - C**

**By**

**Name : Rishabh Singh**

**Roll Number : 202401100300201**

**KIET Group of Institutions, Ghaziabad**

**May, 2025**

**Introduction**

The increasing availability of user interaction data on streaming platforms enables advanced analysis of viewing habits. By clustering users based on patterns like time of watching, genre preferences, and rating behavior, platforms can personalize content delivery, improve recommendations, and enhance user engagement.

**Problem Statement**

The challenge is to analyze a dataset of movie watchers and identify distinct clusters of users based on:

* Time of day they usually watch movies.
* Genre preferences (e.g., Action, Comedy, Drama).
* Rating behavior (e.g., average rating, consistency in rating).

**Objective**

* To segment users into meaningful clusters using unsupervised learning.
* To visualize these clusters using heat maps and interpret common viewing patterns.
* To extract actionable insights that could help in designing recommendation systems.

**Methodology**

**1. Data Preprocessing**

* **Clean the dataset (handle missing values, format timestamps).**
* **Feature engineering: extract hour of the day, encode genres, normalize ratings.**

**2. Clustering Technique**

* **Use K-Means Clustering as the primary algorithm.**
* **Evaluate cluster quality using:**
  + **Silhouette Score**
  + **Davies-Bouldin Index**

**Visualization**

* **Heat maps to visualize confusion matrix for cluster label vs actual label (if available).**
* **PCA or t-SNE for 2D visualization of clusters.**

Code (Python Example)

import pandas as pd

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

# Load the data

df = pd.read\_csv("//content/movie\_watch (1).csv")

# Define preprocessing for numerical and categorical columns

numeric\_features = ['watch\_time\_hour', 'avg\_rating\_given']

categorical\_features = ['genre\_preference']

# Create a column transformer

preprocessor = ColumnTransformer(

    transformers=[

        ('num', StandardScaler(), numeric\_features),

        ('cat', OneHotEncoder(), categorical\_features)

    ]

)

# Create a pipeline that first transforms the data then applies KMeans

pipeline = Pipeline([

    ('preprocessor', preprocessor),

    ('clusterer', KMeans(n\_clusters=3, random\_state=42))

])

# Fit the pipeline

pipeline.fit(df)

# Predict clusters

df['cluster'] = pipeline.predict(df)

# Display cluster assignment

print(df.head())

# Optional: Plotting (only 2D example using PCA)

from sklearn.decomposition import PCA

# Reduce dimensions for visualization

X\_transformed = pipeline.named\_steps['preprocessor'].transform(df)

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X\_transformed)

plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=df['cluster'], cmap='viridis')

plt.title("User Clusters based on Movie Watch Pattern")

plt.xlabel("PCA Component 1")

plt.ylabel("PCA Component 2")

plt.colorbar(label='Cluster')

plt.show()

**Output**

**watch\_time\_hour genre\_preference avg\_rating\_given cluster**

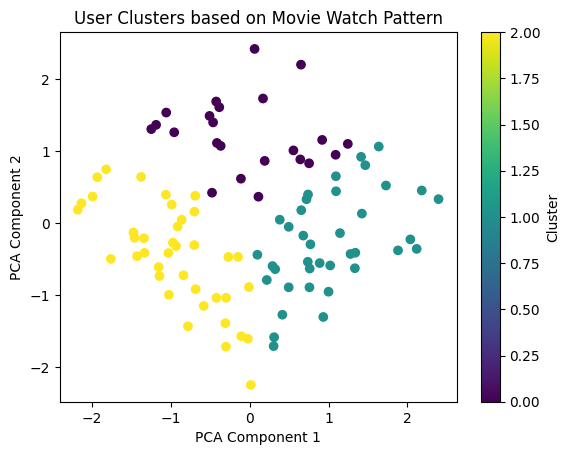
**0 13 action 2.037554 2**

**1 4 comedy 1.350365 2**

**2 15 thriller 1.359665 2**

**3 14 thriller 1.772998 2**

**4 14 comedy 1.202237 2**

****

**References / Credits**

Dataset Source: [IMDb / Kaggle MovieLens Dataset]

Libraries used: pandas, matplotlib, seaborn, scikit-learn

Inspired by academic works in recommendation systems and behavioral clustering