**By Aditi Singh**

**Roll No: 202401100300014**

**Course: B.Tech CSE (AI)**

**Date: 11-03-2025**

Movie Watch Pattern Clustering

Project Report

**Introduction**

This project uses **clustering** to group users based on their **movie-watching behaviour**. The goal is to discover patterns by analyzing:

* **Watch Time** (the hour of the day the user watches movies),
* **Genre Preference** (e.g., Comedy, Drama, Action),
* **Rating Behaviour** (how the user rates movies).

The program uses the **K-Means Clustering** algorithm to group similar users together. This is useful for understanding user habits and potentially improving recommendation systems.

**Method**

**Data Loading**

1. The program reads data from a CSV file named movie\_watch.csv. Each row contains:
2. watch\_time\_hour: The hour (0–23) when the user watched a movie,
3. genre\_preference: The genre of the movie watched,
4. avg\_rating\_given: The rating the user gave the movie.

**Preprocessing the Data**

**Genre Encoding**: Since genres are words (like "comedy"), we convert them into numbers using LabelEncoder.

**Feature Scaling**: To make clustering work properly, all numeric values (hour, genre, rating) are scaled using StandardScaler. This ensures that no one feature dominates the others.

**Applying K-Means Clustering**

We use the **KMeans** algorithm from the sklearn library.

We choose n\_clusters = 5, which means we want to group users into 5 categories.

The model analyzes the scaled data and assigns a cluster label to each user.

**Visualizing the Results**

After clustering, the program counts how many users are in each cluster.

It displays this information using a **pie chart** with matplotlib, showing the percentage of users in each group.

This helps us visually understand the clustering result.

Code :

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import LabelEncoder, StandardScaler

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

df.rename(columns={

    'watch\_time\_hour': 'hour',

    'genre\_preference': 'genre',

    'avg\_rating\_given': 'rating'

}, inplace=True)

label\_encoder = LabelEncoder()

df['genre\_encoded'] = label\_encoder.fit\_transform(df['genre'])

features = df[['hour', 'genre\_encoded', 'rating']]

scaler = StandardScaler()

scaled\_features = scaler.fit\_transform(features)

kmeans = KMeans(n\_clusters=5, random\_state=42)

df['cluster'] = kmeans.fit\_predict(scaled\_features)

cluster\_counts = df['cluster'].value\_counts().sort\_index()

labels = [f"Cluster {i}" for i in cluster\_counts.index]

plt.pie(cluster\_counts, labels=labels, autopct='%1.1f%%')

plt.title("Movie Watch Pattern Clustering")

plt.show()

Result:

