**GURU GOBIND SINGH COLLEGE OF ENGINEERING & RESEARCH CENTRE, NASHIK**

**MINI PROJECT REPORT**

**Academic year: 2024-25**

**“Movie Ticket Management”**

**Bachelor of Engineering (Computer Engineering)**

**Course**: Laboratory Practice-III (ML)

**Course code: 410246**

**By**

**Name(s): Aman H. Bagga**

Under the Guidance of

**Mrs. Pradnya K. Bachhav**

**Guru Gobind Singh College of Engineering & Research Centre, Nashik**

**Mini-Project Report**

**Name of Programme**: Computer Engineering **Academic Year:** 2024-25

**Semester:** BECO-Sem 1 C**ourse code:** 410246

**Name of Course**: Laboratory Practice-III (ML)

-------------------------------------------------------------------------------------------------------------------------------

**Title of Mini-Project:** “Movie Recommendation System”

**1.0 Rationale:**

In this mini project we have developed a movie recommendation system. Movie recommendation systems have become an integral part of modern entertainment platforms, providing personalized suggestions to users based on their viewing history and preferences. Building such a system is a valuable learning experience for anyone interested in machine learning (ML).

**2.0 Aim /Benefits of Mini-Project:**

The mini project on a movie recommendation system aims to build a personalized system that suggests movies based on user preferences using machine learning techniques. This project provides a hands-on opportunity to learn data science skills, understand real-world ML applications, and develop problem-solving abilities.

**3.0 Course Outcomes achieved (COs):**

1. Apply preprocessing techniques on datasets (CO1)
2. Implement and evaluate linear regression and random forest regression models (CO2)
3. Apply and evaluate classification and clustering techniques (CO3)

**4.0 Literature Review:** -

Movie recommendation systems have become an integral part of online streaming platforms, enhancing user experience by suggesting personalized content. These systems leverage various machine learning techniques to analyse users’ behaviour and item attributes to provide tailored recommendations. In traditional systems, the user has to go through all the movie description to find out whether it is of the genre he/she likes and then decide whether to watch it or not. This was a tedious and time-consuming work, but with the innovation of movie recommendation system, the user just has to enter his favourite genre and he/she gets the list of movies of his/her liking.

**5.0 Actual Methodology followed:**

The user has to enter the movie he/she liked watching. If the movie name is in the dataset, then it will compare that movie with other movies. Depending upon the similarity and relation between the movies, it will suggest movies with the most similarity with the entered movie. The number of movies suggested, can be altered depending upon the user.

**1.Algorithm:**

1. Start
2. Import Pandas, NumPy and Sklearn packages.
3. Import the cosine\_similarity and CountVectorizer from sklearn package
4. Read the dataset of various movies with its features
5. Extract 4 feature from the main dataset (keyword, cast, genre, director)
6. Combine those extracted feature into a new dataframe
7. Apply CountVectorizer() method on this new dataframe and store data into variable
8. Apply cosine\_similarity on this variable and store the answer in another variable.
9. Create 2 methods: first that takes index as input and returns title and second that takes title as input and returns index
10. Take input from user about the movie he/she likes and find its index
11. Use this index to find similarity between this movie and rest all movies and store the cosine value in a list
12. Sort this list in descending order, meaning the maximum cosine value will be at 0 position indicating high similarity.
13. Select only top 11 elements and keep it in the list and delete the rest
14. Find out the original index of each element in the list and using that index find the title of the movie and display it.
15. Stop

**6.0 Actual Code of Program:**

import pandas as pd

import numpy as np

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

from google.colab import files

uploaded = files.upload()

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

features = ['keywords', 'cast', 'genres', 'director']

df['keywords'] = df['keywords'].astype(str)

df['cast'] = df['cast'].astype(str)

df['genres'] = df['genres'].astype(str)

df['director'] = df['director'].astype(str)

def combine\_features(row):

    return row['keywords'] + " " + row['cast'] + " " + row["genres"] + " " + row["director"]

df['combined\_features'] = df.apply(combine\_features, axis=1)

df['combined\_features'] = df['combined\_features'].fillna('')

vectorizer = CountVectorizer()

count\_matrix = vectorizer.fit\_transform(df['combined\_features'])

cosine\_sim = cosine\_similarity(count\_matrix)

def get\_title\_from\_index(index):

return df[df.index == index]['title'].values[0]

def get\_index\_from\_title(title):

return df[df.title == title].index.values[0]

movie\_index = get\_index\_from\_title('Avatar')

sim\_scores = list(enumerate(cosine\_sim[movie\_index]))

sim\_scores = sorted(sim\_scores, key=lambda x: x[1], reverse=True)

sim\_scores = sim\_scores[1:11]

for i, score in sim\_scores:

print(get\_title\_from\_index(i))

**7.0 Actual Resources used:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Name of Resource/material** | **Specifications** | **Qty** | **Remarks** |
| 1 | Computer System | Windows OS,i3 processor ,2GB RAM | 01 | - |
| 2 | Software | Tomcat-8.5.24, MS Office word, netbeans 8.1 | 01 | - |

**8.0 Skill Developed / Learning outcome from this Mini-Project:**

1. Understanding the concept of Matrix Factorization

2. How content-based filtering works

**9.0 Applications of Mini Project:**

1. To recommend movies depending upon what movie you like

**Evaluated by: Mrs. Pradnya K. Bachhav**

**Date: Name & Signature of Guide**