**Movie Recommendation System Using ML**

**Project Overview**

I worked on a movie recommendation system to provide personalized movie suggestions for users based on their viewing history and ratings. This project aimed to improve user engagement and satisfaction by suggesting movies they would likely enjoy.

**My Role**

I was responsible for:

* **Loading and Preprocessing Data:** Collecting and cleaning user ratings and movie details.
* **Building and Training Models:** Using machine learning algorithms to create a recommendation engine.
* **Making Predictions:** Providing movie recommendations for users.

**Simplified Code Explanation**

**Load Data:**

import pandas as pd

from surprise import Dataset, Reader

**# Load datasets**

movies = pd.read\_csv(r"C:\Users\LENOVO\Downloads\ml-latest-small\movies.csv")

ratings = pd.read\_csv(r"C:\Users\LENOVO\Downloads\ml-latest-small\ratings.csv")

**# Prepare data for the Surprise library**

reader = Reader(rating\_scale=(0.5, 5.0))

data = Dataset.load\_from\_df(ratings[['userId', 'movieId', 'rating']], reader)

**Train the Model:**

from surprise import SVD

from surprise.model\_selection import train\_test\_split

**# Split data into training and test sets**

trainset, testset = train\_test\_split(data, test\_size=0.25)

**# Train the SVD algorithm**

algo = SVD()

algo.fit(trainset)

**Make Predictions:**

**# Test the algorithm on the test set**

predictions = algo.test(testset)

**# Function to get top N recommendations for each user**

from collections import defaultdict

def get\_top\_n\_recommendations(predictions, n=10):

top\_n = defaultdict(list)

for uid, iid, true\_r, est, \_ in predictions:

top\_n[uid].append((iid, est))

for uid, user\_ratings in top\_n.items():

user\_ratings.sort(key=lambda x: x[1], reverse=True)

top\_n[uid] = user\_ratings[:n]

return top\_n

top\_n\_recommendations = get\_top\_n\_recommendations(predictions, n=10)

**Output Recommendations:**

# Example user ID

user\_id = '1'

print(f"Top 10 recommendations for user {user\_id}:")

for movie\_id, rating in top\_n\_recommendations[user\_id]:

movie\_title = movies[movies['movieId'] == int(movie\_id)]['title'].values[0]

print(f"Movie: {movie\_title}, Predicted Rating: {rating}")

**Conclusion**

This simplified code demonstrates how to build a movie recommendation system:

* **Load and Prepare Data:** Import movie and rating data, and format it for the surprise library.
* **Train a Model:** Use the SVD algorithm to train a recommendation model.
* **Make Predictions:** Predict movie ratings for users and generate top N recommendations.

This project improved user engagement by providing personalized movie suggestions, making the streaming service more enjoyable and user-friendly.