3.Task (RECOMMENDATION SYSTEM)

# SECTION 1: IMPORTS

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

# SECTION 2: LOAD SAMPLE MOVIE DATA

data = {

'movie\_id': [1, 2, 3, 4, 5],

'title': ['The Matrix', 'John Wick', 'The Godfather', 'The Dark Knight', 'Pulp Fiction'],

'description': [

'A hacker discovers the reality is a simulation and fights AI.',

'An ex-hitman goes on a revenge mission.',

'A mafia family faces betrayal and revenge.',

'A man in a bat suit fights crime in Gotham.',

'Non-linear story about crime, hitmen, and redemption.'

]

}

df = pd.DataFrame(data)

# SECTION 3: TF-IDF VECTORIZATION

tfidf = TfidfVectorizer(stop\_words='english')

tfidf\_matrix = tfidf.fit\_transform(df['description'])

# SECTION 4: COMPUTE COSINE SIMILARITY

cosine\_sim = cosine\_similarity(tfidf\_matrix, tfidf\_matrix)

# SECTION 5: MOVIE INDEX MAPPER

indices = pd.Series(df.index, index=df['title'])

# SECTION 6: RECOMMENDER FUNCTION

def recommend\_movie(title, num\_recommendations=3):

idx = indices[title]

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

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

movie\_indices = [i[0] for i in sim\_scores]

return df['title'].iloc[movie\_indices]

# EXAMPLE USAGE

print("Recommended for 'The Matrix':")

print(recommend\_movie('The Matrix'))