Python Program (imdb\_rating\_prediction.py)

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import mean\_squared\_error, r2\_score

df = pd.read\_csv('imdb\_movies.csv')

df.fillna(df.mean(numeric\_only=True), inplace=True)

df.fillna(method='ffill', inplace=True)

label\_encoders = {}

for col in ['Genre', 'Director']:

le = LabelEncoder()

df[col] = le.fit\_transform(df[col])

label\_encoders[col] = le

X = df.drop('Rating', axis=1)

y = df['Rating']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse:.2f}")

print(f"R² Score: {r2:.2f}")

def predict\_imdb\_rating(genre, director, runtime, votes, revenue):

genre\_encoded = label\_encoders['Genre'].transform([genre])[0]

director\_encoded = label\_encoders['Director'].transform([director])[0]

input\_data = pd.DataFrame([[genre\_encoded, director\_encoded, runtime, votes, revenue]],

columns=['Genre', 'Director', 'Runtime', 'Votes', 'Revenue (Millions)'])

prediction = model.predict(input\_data)[0]

return round(prediction, 2)

new\_rating = predict\_imdb\_rating('Comedy', 'Taika Waititi', 115, 200000, 120.0)

print(f"Predicted IMDb Rating: {new\_rating}")