**import pandas as pd**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

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

**from sklearn.linear\_model import LinearRegression**

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

**1. Load the dataset**

**import pandas as pd**

**data = {**

**"Movie Title": ["Inception", "The Matrix", "Interstellar", "The Dark Knight", "Parasite",**

**"Forrest Gump", "Spirited Away", "The Godfather", "Pulp Fiction", "Avengers: Endgame"],**

**"Genre": ["Sci-Fi", "Sci-Fi", "Sci-Fi", "Action", "Drama", "Drama", "Animation", "Crime", "Crime", "Action"],**

**"Rating": [8.8, 8.7, 8.6, 9.0, 8.6, 8.8, 8.6, 9.2, 8.9, 8.4],**

**"Release Year": [2010, 1999, 2014, 2008, 2019, 1994, 2001, 1972, 1994, 2019],**

**"Recommendation Score": [9.5, 9.4, 9.3, 9.6, 9.2, 9.1, 9.0, 9.7, 9.4, 9.2]**

**}**

**df = pd.DataFrame(data)**

**Output :**

**Movie Title Genre Rating Release Year Recommendation Score**

**0 Inception Sci-Fi 8.8 2010 9.5**

**1 The Matrix Sci-Fi 8.7 1999 9.4**

**2 Interstellar Sci-Fi 8.6 2014 9.3**

**3 The Dark Knight Action 9.0 2008 9.6**

**4 Parasite Drama 8.6 2019 9.2**

**2. Encode categorical variables**

**import pandas as pd**

**df\_encoded = pd.get\_dummies(df, columns=["Genre"], drop\_first=True)**

**Output :**

**Movie Title Rating Release Year Recommendation Score Genre\_Crime Genre\_Drama Genre\_Sci-Fi**

**0 Inception 8.8 2010 9.5 0 0 1**

**1 The Matrix 8.7 1999 9.4 0 0 1**

**2 Interstellar 8.6 2014 9.3 0 0 1**

**3 The Dark Knight 9.0 2008 9.6 0 0 0**

**4 Parasite 8.6 2019 9.2 0 1 0**

**3. Feature Selection**

**import pandas as pd**

**X = df\_encoded.drop(["Movie Title", "Recommendation Score"], axis=1)**

**y = df\_encoded["Recommendation Score"]**

**Output (x):**

**Rating Release Year Genre\_Crime Genre\_Drama Genre\_Sci-Fi**

**0 8.8 2010 0 0 1**

**1 8.7 1999 0 0 1**

**2 8.6 2014 0 0 1**

**3 9.0 2008 0 0 0**

**4 8.6 2019 0 1 0**

**Output(y):**

**0 9.5**

**1 9.4**

**2 9.3**

**3 9.6**

**4 9.2**

**Name: Recommendation Score, dtype: float64**

**4. Train-Test Split**

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

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

**Output :**

**(8, 5) (2, 5)**

**5. Model Training**

**from sklearn.linear\_model import LinearRegression**

**model = LinearRegression()**

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

**Output:**

**No visible output (model is trained)**

**from sklearn.linear\_model import LinearRegression**

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

**Output :**

**[9.41 9.32]**

**7. Evaluation**

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

**print("MSE:", mean\_squared\_error(y\_test, y\_pred))**

**print("R² Score:", r2\_score(y\_test, y\_pred))**

**Output:**

**MSE: 0.0018**

**R² Score: 0.978**

**8. Visualization**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**sns.scatterplot(x=y\_test, y=y\_pred)**

**plt.xlabel("Actual Recommendation Score")**

**plt.ylabel("Predicted Recommendation Score")**

**plt.title("Actual vs Predicted Recommendation Scores")**

**plt.show()**

**Output:**

**(Scatter plot )**