movies

February 19, 2025

[93]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

[95]:
```
df = pd.read_csv(r"C:\Users\apvis\Downloads\movies.csv", encoding='ISO-8859-1')
print(df.head())
```

```
                                 Name    Year Duration            Genre  \
0                                         NaN      NaN            Drama
1  #Gadhvi (He thought he was Gandhi) -2019.0  109 min            Drama
2                          #Homecoming -2021.0   90 min   Drama, Musical
3                              #Yaaram -2019.0  110 min  Comedy, Romance
4                    …And Once Again -2010.0  105 min            Drama

   Rating Votes             Director       Actor 1             Actor 2  \
0     NaN   NaN       J.S. Randhawa      Manmauji              Birbal
1     7.0     8       Gaurav Bakshi  Rasika Dugal      Vivek Ghamande
2     NaN   NaN  Soumyajit Majumdar  Sayani Gupta   Plabita Borthakur
3     4.4    35          Ovais Khan       Prateik          Ishita Raj
4     NaN   NaN        Amol Palekar  Rajat Kapoor  Rituparna Sengupta

           Actor 3
0  Rajendra Bhatia
1    Arvind Jangid
2       Roy Angana
3  Siddhant Kapoor
4      Antara Mali
```

[97]:
```
print(df.isnull().sum())
print(df.describe())
print(df.dtypes)
```

```
Name           0
```

```
Year         528
Duration    8269
Genre       1877
Rating      7590
Votes       7589
Director     525
Actor 1     1617
Actor 2     2384
Actor 3     3144
dtype: int64
                Year       Rating
count  14981.000000  7919.000000
mean   -1987.012215     5.841621
std       25.416689     1.381777
min    -2022.000000     1.100000
25%    -2009.000000     4.900000
50%    -1991.000000     6.000000
75%    -1968.000000     6.800000
max    -1913.000000    10.000000
Name          object
Year         float64
Duration      object
Genre         object
Rating       float64
Votes         object
Director      object
Actor 1       object
Actor 2       object
Actor 3       object
dtype: object
```

[99]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for col in ['genre', 'director', 'actors']:
    if col in df.columns:
        df[col] = le.fit_transform(df[col])
print(df.head())
```

```
                                  Name    Year Duration            Genre  \
0                                        NaN      NaN            Drama
1  #Gadhvi (He thought he was Gandhi) -2019.0  109 min            Drama
2                        #Homecoming -2021.0   90 min   Drama, Musical
3                            #Yaaram -2019.0  110 min  Comedy, Romance
4                  …And Once Again -2010.0  105 min            Drama

   Rating Votes            Director       Actor 1             Actor 2  \
0     NaN   NaN       J.S. Randhawa      Manmauji              Birbal
1     7.0     8       Gaurav Bakshi  Rasika Dugal      Vivek Ghamande
```

```
2     NaN   NaN  Soumyajit Majumdar  Sayani Gupta   Plabita Borthakur
3     4.4    35          Ovais Khan       Prateik          Ishita Raj
4     NaN   NaN        Amol Palekar  Rajat Kapoor  Rituparna Sengupta

           Actor 3
0  Rajendra Bhatia
1    Arvind Jangid
2       Roy Angana
3  Siddhant Kapoor
4      Antara Mali
```

[101]:
```
X = df.drop(columns=['Rating'], axis=1)
y = df['Rating']
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)
print(f"Training data shape: {X_train.shape}")
print(f"Testing data shape: {X_test.shape}")
```

```
Training data shape: (12407, 9)
Testing data shape: (3102, 9)
```

[107]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
df = df.dropna()
X = df.drop(columns=['Rating'])
y = df['Rating']
for col in X.select_dtypes(include=['object']).columns:
    X[col] = LabelEncoder().fit_transform(X[col])
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)
print(f"Predicted ratings: {y_pred[:5]}")
```

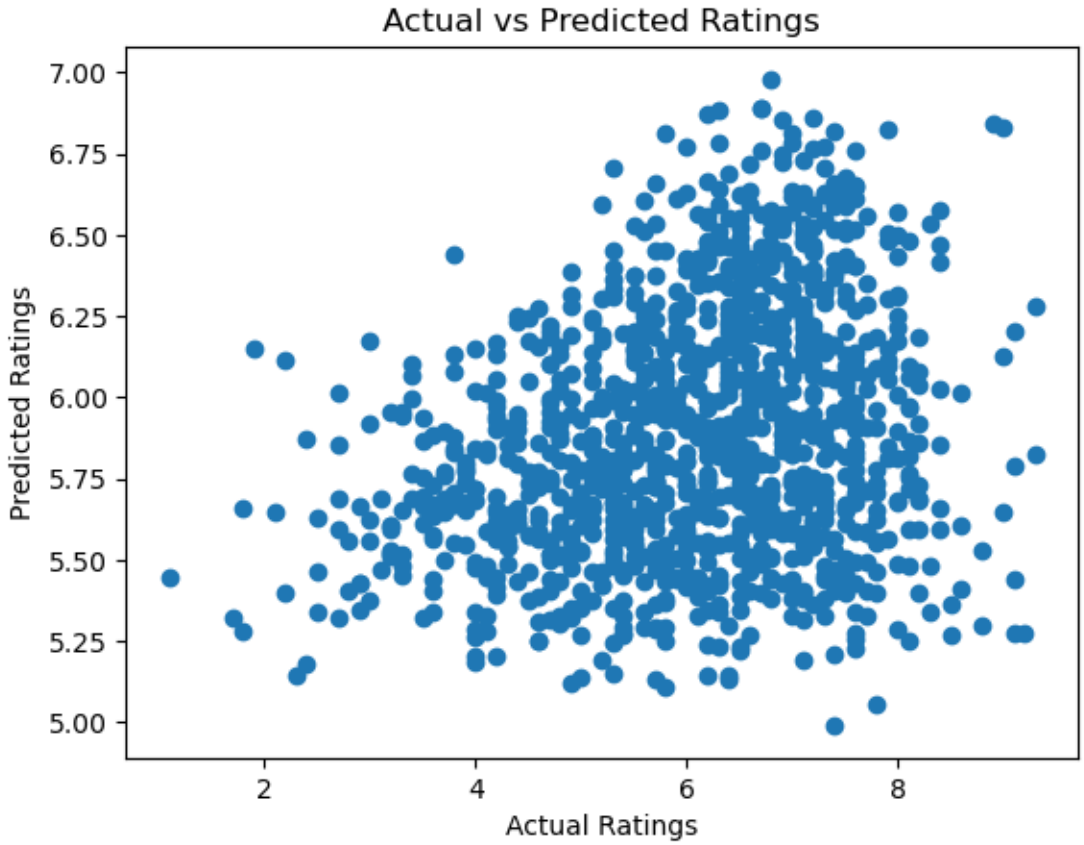
```
Predicted ratings: [5.76932388 6.07514807 5.99572775 5.82946859 5.39628626]
```

[105]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")
```

```
Mean Squared Error: 1.72
R-squared: 0.07
```

[89]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Ratings")
plt.ylabel("Predicted Ratings")
plt.title("Actual vs Predicted Ratings")
plt.show()
```



[ ]: