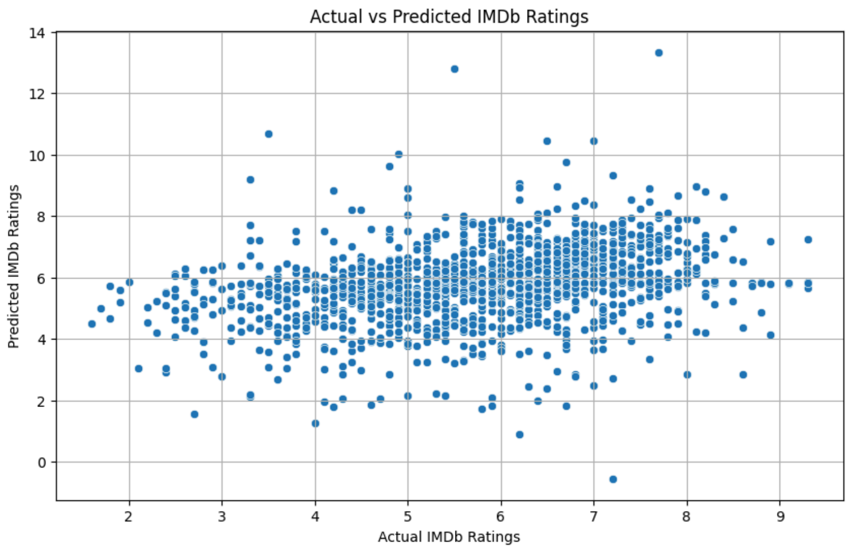
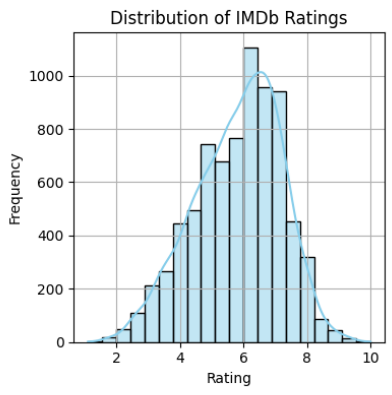
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
df = pd.read_csv('/content/movie rating.zip', encoding='ISO-8859-1')
df.dropna(subset=['Rating', 'Genre', 'Director', 'Actor 1'], inplace=True)
df['Votes'] = df['Votes'].str.replace(',', '', regex=True)
df['Votes'] = pd.to_numeric(df['Votes'], errors='coerce')
df['Votes'].fillna(df['Votes'].median(), inplace=True)
features = ['Genre', 'Director', 'Actor 1']
X = df[features]
y = df['Rating']
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), features)
    ]
)
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('model', LinearRegression())
])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)
print("R² Score:", r2_score(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.xlabel("Actual IMDb Ratings")
plt.ylabel("Predicted IMDb Ratings")
plt.title("Actual vs Predicted IMDb Ratings")
plt.grid(True)
plt.show()
```

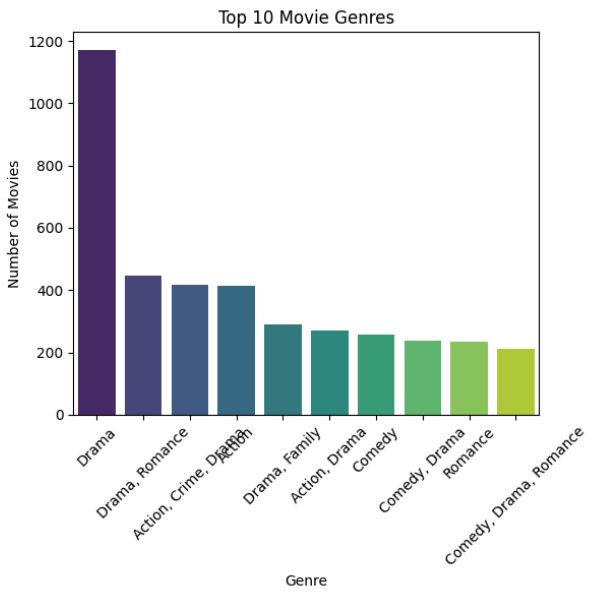
```
<ipython-input-1-725bf9553913>:15: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the ope


  df['Votes'].fillna(df['Votes'].median(), inplace=True)
R² Score: -0.1881077082661322
MSE: 2.292893597497809
```



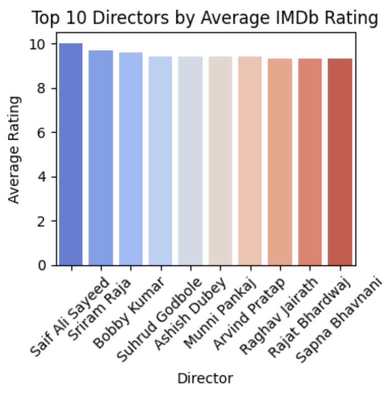
```
plt.figure(figsize=(4,4))
sns.histplot(df['Rating'], bins=20, kde=True, color='skyblue')
plt.title("Distribution of IMDb Ratings")
plt.xlabel("Rating")
plt.ylabel("Frequency")
plt.grid(True)
plt.tight_layout()
plt.show()
plt.figure(figsize=(6,6))
top_genres = df['Genre'].value_counts().head(10)
sns.barplot(x=top_genres.index, y=top_genres.values, palette='viridis')
plt.title("Top 10 Movie Genres")
plt.ylabel("Number of Movies")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
top_directors = df.groupby('Director')['Rating'].mean().sort_values(ascending=False).head(10)
plt.figure(figsize=(4, 4))
sns.barplot(x=top_directors.index, y=top_directors.values, palette='coolwarm')
plt.title("Top 10 Directors by Average IMDb Rating")
plt.ylabel("Average Rating")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
<ipython-input-9-049fa7306358>:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_genres.index, y=top_genres.values, palette='viridis')
```



```
<ipython-input-9-049fa7306358>:21: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_directors.index, y=top_directors.values, palette='coolwarm')
```


Top 10 Directors by Average IMDb Rating
Average Rating
Director
Saif Ali Sayeed
Sriram Raja
Bobby Kumar
Suhrud Godbole
Ashish Dubey
Munni Pankaj
Arvind Pratap
Raghav Jairath
Rajat Bhardwaj
Sapna Bhavnani