

9:00

9.00 KB/S VoLTE 5G 36%



Untitled9.ipynb...

rch.google.com



Commands + Code + Text ▶ Run all

RAM Disk



[18]

✓ 1m

```
from google.colab import files  
uploaded = files.upload()
```

Choose files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving IMDb Movies India.csv to IMDb Movies India (3).csv

[45]

✓ 0s

```
'IMDb_India_Movies.csv'  
'IMDb_India_Movies.csv'
```

[46]

⌚ 0s

```
df = pd.read_csv("IMDb_India_Movies.csv")  
df.head()
```

```
FileNotFoundError Traceback (most recent call last)  
/tmp/ipython-input-1959149819.py in <cell line: 0>()  
----> 1 df = pd.read_csv("IMDb_India_Movies.csv")  
      2 df.head()
```

```
-----  
/usr/local/lib/python3.12/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode,  
encoding, compression, memory_map, is_text, errors, storage_options)  
    871         if ioargs.encoding and "b" not in ioargs.mode:  
    872             # Encoding  
--> 873             handle = open(  
    874                 handle,  
    875                 ioargs.mode,
```

FileNotFoundError: [Errno 2] No such file or directory: 'IMDb_India_Movies.csv'

[47]

⌚ 0s

```
df = pd.read_csv("IMDb Movies India.csv")  
df.head()
```

```
UnicodeDecodeError Traceback (most recent call last)  
/tmp/ipython-input-3610933116.py in <cell line: 0>()  
----> 1 df = pd.read_csv("IMDb Movies India.csv")  
      2 df.head()
```

```
-----  
parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()
```

```
parsers.pyx in pandas._libs.parsers.TextReader._get_header()
```

```
parsers.pyx in pandas._libs.parsers.TextReader._tokenize_rows()
```

```
parsers.pyx in pandas._libs.parsers.TextReader._check_tokenize_status()
```

```
parsers.pyx in pandas._libs.parsers.raise_parser_error()
```

```
/usr/lib/python3.12/codecs.py in decode(self, input, final)
```

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe1 in position 76763: invalid continuation byte

[48]

✓ 0s

```
df = pd.read_csv("IMDb Movies India.csv", encoding='latin1')  
df.head()
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		Nan	Nan	Drama	Nan	Nan	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	#Homecoming	(2021)	90 min	Drama, Musical	Nan	Nan	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana
3	#Yaaram	(2019)	110 min	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor
4	...And Once Again	(2010)	105 min	Drama	Nan	Nan	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	Antara Mali

[49]

✓ 0s

```
df.info()  
df.isnull().sum()
```



```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 15509 entries, 0 to 15508  
Data columns (total 10 columns):  
 #   Column   Non-Null Count  Dtype     
 ---    
 0   Name     15509 non-null  object    
 1   Year     15509 non-null  int64     
 2   Duration  15509 non-null  float64  
 3   Genre    15509 non-null  object    
 4   Rating   15509 non-null  float64  
 5   Votes    15509 non-null  int64     
 6   Director 15509 non-null  object    
 7   Actor 1  15509 non-null  object    
 8   Actor 2  15509 non-null  object    
 9   Actor 3  15509 non-null  object
```



Untitled9.ipynb...

rch.google.com



Q

Commands

+ Code ▾ + Text ▾ ▶ Run all ▾

✓ RAM Disk ▾ ▾ ▾

≡

[49]
✓ 0s

```
df.info()  
df.isnull().sum()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 15509 entries, 0 to 15508  
Data columns (total 10 columns):  
 #   Column   Non-Null Count   Dtype     
 ---  -----   ----    
 0   Name     15509 non-null    object    
 1   Year     14981 non-null    object    
 2   Duration 7240 non-null    object    
 3   Genre    13632 non-null    object    
 4   Rating   7919 non-null    float64   
 5   Votes    7920 non-null    object    
 6   Director 14984 non-null    object    
 7   Actor 1  13892 non-null    object    
 8   Actor 2  13125 non-null    object    
 9   Actor 3  12365 non-null    object    
dtypes: float64(1), object(9)  
memory usage: 1.2+ MB
```

Q

[50]
✓ 0s

```
0  
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
```

<>

Key

File

Cell

[51]
✓ 0s

```
df.info()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 15509 entries, 0 to 15508  
Data columns (total 10 columns):  
 #   Column   Non-Null Count   Dtype     
 ---  -----   ----    
 0   Name     15509 non-null    object    
 1   Year     14981 non-null    object    
 2   Duration 7240 non-null    object    
 3   Genre    13632 non-null    object    
 4   Rating   7919 non-null    float64   
 5   Votes    7920 non-null    object    
 6   Director 14984 non-null    object    
 7   Actor 1  13892 non-null    object    
 8   Actor 2  13125 non-null    object    
 9   Actor 3  12365 non-null    object    
dtypes: float64(1), object(9)  
memory usage: 1.2+ MB
```

[52]
✓ 0s

```
df = df.dropna(subset=['Rating', 'Votes'])  
df.isnull().sum()  
df.shape
```

```
(7919, 10)
```

[53]
✓ 0s

```
features = ['Genre', 'Director', 'Actor 1', 'Votes', 'Year']
```

```
target = 'Rating'
```

[54]
✓ 0s

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder
```

```
# Select important columns
```

```
df = df[features + [target]]
```

```
# Encode Categorical columns
```

```
encoder = LabelEncoder()
```

```
for col in ['Genre', 'Director', 'Actor 1']:
```

```
    df[col] = encoder.fit_transform(df[col].astype(str))
```

```
# Splitting data
```

```
X = df[features]
```

```
y = df[target]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
df.head()
```

```
Genre  Director  Actor 1  Votes  Year  Rating
```

Variables

Terminal

Python 3



Untitled9.ipynb

rch.google.com



Commands + Code + Text ▶ Run all RAM Disk

	Genre	Director	Actor 1	Votes	Year	Rating
1	268	811	1782	8	(2019)	7.0
3	207	1749	1589	35	(2019)	4.4
5	177	2005	508	827	(1997)	4.7
6	331	2643	931	1,086	(2005)	7.4
8	367	174	2520	326	(2012)	5.6

[56]

✓ 0s

```
import pandas as pd

df = pd.read_csv("IMDb Movies India.csv", encoding='latin1')
df.head()
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		Nan	Nan	Drama	Nan	Nan	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	#Homecoming	(2021)	90 min	Drama, Musical	Nan	Nan	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana
3	#Yaaram	(2019)	110 min	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor
4	...And Once Again	(2010)	105 min	Drama	Nan	Nan	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	Antara Mali

[57]

✓ 1s

```
df.info()
df.describe()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15509 entries, 0 to 15508
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Name        15509 non-null   object  
 1   Year         14981 non-null   object  
 2   Duration     7240 non-null   object  
 3   Genre        13632 non-null   object  
 4   Rating       7919 non-null   float64 
 5   Votes        7920 non-null   object  
 6   Director     14984 non-null   object  
 7   Actor 1      13892 non-null   object  
 8   Actor 2      13125 non-null   object  
 9   Actor 3      12365 non-null   object  
dtypes: float64(1), object(9)
memory usage: 1.2+ MB
```

```
0
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
```

[58]

✓ 0s

```
df = df.dropna(subset=['Rating', 'Votes'])
df.shape
```

```
(7919, 10)
```

[62]

```
from sklearn.preprocessing import LabelEncoder
```

```
# LabelEncoder initialize
le = LabelEncoder()
```

```
# Categorical columns list
cat_cols = ['Genre', 'Director', 'Actor 1', 'Actor 2', 'Actor 3']
```

9:01

0.49 KB/S VoLTE 5G 35%



Untitled9.ipynb...

rch.google.com



Commands + Code + Text ▶ Run all

RAM Disk



```
ai.shape
(7919, 10)
```

```
[62]
from sklearn.preprocessing import LabelEncoder
# LabelEncoder initialize
le = LabelEncoder()

# Categorical columns list
cat_cols = ['Genre', 'Director', 'Actor 1', 'Actor 2', 'Actor 3']

for col in cat_cols:
    df[col] = df[col].astype(str) # Convert to string
    df[col] = le.fit_transform(df[col])
```

df.head()

```
-----
KeyError Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
    3804     try:
-> 3805         return self._engine.get_loc(casted_key)
    3806     except KeyError as err:
```

```
index.pyx in pandas._libs.index.IndexEngine.get_loc()
index.pyx in pandas._libs.index.IndexEngine.get_loc()
```

```
pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()
```

```
pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()
```

KeyError: 'Actor 2'

The above exception was the direct cause of the following exception:

```
-----
KeyError Traceback (most recent call last)
-> /usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
    3810     ):
    3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
    3813 except TypeError:
    3814     # If we have a listlike key, _check_indexing_error will raise
```

KeyError: 'Actor 2'

[63]

df.isnull().sum()

0

Genre 0

Director 0

Actor 1 0

Votes 0

Year 0

Rating 0

dtype: int64

[64]

```
from sklearn.model_selection import train_test_split
X = df[['Genre', 'Director', 'Actor 1', 'Actor 2', 'Actor 3', 'Duration', 'Year', 'Votes']]
y = df['Rating']
```

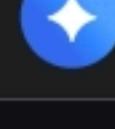
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```
-----
KeyError Traceback (most recent call last)
/tmp/ipython-input-3787267501.py in <cell line: 0>()
    1 from sklearn.model_selection import train_test_split
    2
----> 3 X = df[['Genre', 'Director', 'Actor 1', 'Actor 2', 'Actor 3', 'Duration', 'Year',
-> 4   'Votes']]
    5 y = df['Rating']
    5
```

```
----- 2 frames -----
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in
_raise_if_missing(self, key, indexer, axis_name)
    6250
    6251     not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 6252     raise KeyError(f"{not_found} not in index")
    6253
    6254     @overload
```

KeyError: "['Actor 2', 'Actor 3', 'Duration'] not in index"

{ Variables Terminal



Python 3



Untitled9.ipynb...

rch.google.com



Commands + Code + Text ▶ Run all ✓ RAM Disk

```
[64] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

-----  

KeyError Traceback (most recent call last)  

/tmp/ipython-input-3787267501.py in <cell line: 0>()  

    1 from sklearn.model_selection import train_test_split  

    2  

----> 3 X = df[['Genre', 'Director', 'Actor 1', 'Actor 2', 'Actor 3', 'Duration', 'Year',  

     'Votes']]  

    4 y = df['Rating']  

    5

----- 2 frames -----
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/base.py in  

_raise_if_missing(self, key, indexer, axis_name)  

6250  

6251         not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())  

-> 6252             raise KeyError(f"{not_found} not in index")  

6253  

6254     @overload

KeyError: "['Actor 2', 'Actor 3', 'Duration'] not in index"

[65] ✓ KeyError: "['Actor 2', 'Actor 3', 'Duration'] not in index"

[66] ✓ Os print(df.columns)
Index(['Genre', 'Director', 'Actor 1', 'Votes', 'Year', 'Rating'], dtype='object')

[68] ✓ # Example if there are extra spaces
df.columns = df.columns.str.strip()

[69] ✓ X = df[['Genre', 'Director', 'Actor 1', 'Year', 'Votes']] # remove missing ones
y = df['Rating']

[71] ✓ cols = ['Genre', 'Director', 'Actor 1', 'Actor 2', 'Actor 3', 'Duration', 'Year', 'Votes']
existing_cols = df.columns.intersection(cols)
X = df[existing_cols]
y = df['Rating']

[72] ✓ Os print(df.columns.tolist())
['Genre', 'Director', 'Actor 1', 'Votes', 'Year', 'Rating']

[73] ✓ from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Select features and target
X = df[['Genre', 'Director', 'Actor 1', 'Votes', 'Year']]
y = df['Rating']

# Encoding categorical columns
label_enc = LabelEncoder()
X['Genre'] = label_enc.fit_transform(X['Genre'])
X['Director'] = label_enc.fit_transform(X['Director'])
X['Actor 1'] = label_enc.fit_transform(X['Actor 1'])

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X.head()

/tmp/ipython-input-1148086992.py:10: SettingWithCopyWarning:  

A value is trying to be set on a copy of a slice from a DataFrame.  

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/  

X['Genre'] = label_enc.fit_transform(X['Genre'])  

/tmp/ipython-input-1148086992.py:11: SettingWithCopyWarning:  

A value is trying to be set on a copy of a slice from a DataFrame.  

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/  

X['Director'] = label_enc.fit_transform(X['Director'])  

/tmp/ipython-input-1148086992.py:12: SettingWithCopyWarning:  

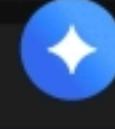
A value is trying to be set on a copy of a slice from a DataFrame.  

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/  

X['Actor 1'] = label_enc.fit_transform(X['Actor 1'])

Genre Director Actor 1 Votes Year
1 188 2932 871 8 (2019)
3 121 834 656 35 (2019)
5 87 1120 2007 827 (1997)
```



9:01

3.00 KB/S Vo 5G 35%



Untitled9.ipynb...

rch.google.com



Commands

+ Code ▾ + Text ▾ ▶ Run all ▾

... RAM Disk



```
[73] ✓ 0s
X = df[['Genre', 'Director', 'Actor 1', 'Votes', 'Year']]
y = df['Rating']

# Encoding categorical columns
label_enc = LabelEncoder()
X['Genre'] = label_enc.fit_transform(X['Genre'])
X['Director'] = label_enc.fit_transform(X['Director'])
X['Actor 1'] = label_enc.fit_transform(X['Actor 1'])

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X.head()
```

/tmp/ipython-input-1148086992.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
X['Genre'] = label_enc.fit_transform(X['Genre'])
/ttmp/ipython-input-1148086992.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
X['Director'] = label_enc.fit_transform(X['Director'])
/ttmp/ipython-input-1148086992.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
X['Actor 1'] = label_enc.fit_transform(X['Actor 1'])

	Genre	Director	Actor 1	Votes	Year
1	188	2932	871	8	(2019)
3	121	834	656	35	(2019)
5	87	1120	2007	827	(1997)
6	259	1828	2477	1,086	(2005)
8	298	824	1692	326	(2012)

```
[74] ✓ 0s
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Copy to avoid slice issues
X = df[['Genre', 'Director', 'Actor 1', 'Votes', 'Year']].copy()
y = df['Rating']

# Clean Votes: remove commas, convert to int
X['Votes'] = X['Votes'].str.replace(',', '').astype(int)

# Clean Year: remove parentheses, convert to int
X['Year'] = X['Year'].str.replace('(', '').str.replace(')', '').astype(int)

# Label encoding for categorical features
label_enc = LabelEncoder()
for col in ['Genre', 'Director', 'Actor 1']:
    X[col] = label_enc.fit_transform(X[col])

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

	Genre	Director	Actor 1	Votes	Year
1	188	2932	871	8	2019
3	121	834	656	35	2019
5	87	1120	2007	827	1997
6	259	1828	2477	1086	2005
8	298	824	1692	326	2012

Variables

Terminal

Python 3

9:01

4.00 KB/S VoLTE 5G 35%



Untitled9.ipynb...

rch.google.com



Commands + Code + Text ▶ Run all

RAM Disk



[75]

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error

# Initialize model
model = RandomForestRegressor(n_estimators=200, random_state=42)

# Train the model
model.fit(X_train, y_train)

# Predict on test set
y_pred = model.predict(X_test)

# Evaluate
print("R2 Score:", r2_score(y_test, y_pred))
print("MAE:", mean_absolute_error(y_test, y_pred))
```

```
R2 Score: 0.32286909787605667
MAE: 0.8422127525252525
```

[78]

! 56s

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error

# Features
X = df[['Genre', 'Director', 'Actor 1', 'Votes', 'Year']].copy()
y = df['Rating']

# Clean numeric fields
X['Votes'] = X['Votes'].astype(str).str.replace(',', '').astype(int)
X['Year'] = X['Year'].astype(str).str.replace('(', '').str.replace(')', '').astype(int)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Categorical columns
categorical_cols = ['Genre', 'Director', 'Actor 1']

# OneHotEncoder setup
preprocess = ColumnTransformer([
    ('encoder', OneHotEncoder(handle_unknown='ignore'), categorical_cols),
    ('remainder', 'passthrough'
])

# Model pipeline
model = Pipeline(steps=[
    ('preprocess', preprocess),
    ('regressor', RandomForestRegressor(n_estimators=300, random_state=42))
])

# Train model
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Scores
print("R2 Score:", r2_score(y_test, y_pred))
print("MAE:", mean_absolute_error(y_test, y_pred))
```

```
-----  
KeyboardInterrupt Traceback (most recent call last)  
/tmp/ipython-input-2779720886.py in <cell line: 0>()  
      33  
      34 # Train model  
---> 35 model.fit(X_train, y_train)  
      36  
      37 # Predict
```

```
----- 9 frames -----
```

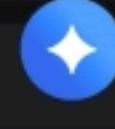
```
/usr/local/lib/python3.12/dist-packages/sklearn/tree/_classes.py in _fit(self, X, y,
sample_weight, check_input, missing_values_in_feature_mask)
    470
    471
--> 472         builder.build(self.tree_, X, y, sample_weight,
missing_values_in_feature_mask)
    473
    474         if self.n_outputs_ == 1 and is_classifier(self):
```

```
KeyboardInterrupt:
```

[79]

✓ 4s

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error
```





Untitled9.ipynb...

rch.google.com



Commands

37 # Predict

```
/usr/local/lib/python3.12/dist-packages/sklearn/tree/_classes.py in _fit(self, X, y,
sample_weight, check_input, missing_values_in_feature_mask)
    470
    471
--> 472     builder.build(self.tree_, X, y, sample_weight,
missing_values_in_feature_mask)
    473
    474     if self.n_outputs_ == 1 and is_classifier(self):
```

KeyboardInterrupt:

[79]

✓ 4s

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error

# Copy dataframe
X = df[['Genre', 'Director', 'Actor 1', 'Votes', 'Year']].copy()
y = df['Rating']

# Clean numeric fields
X['Votes'] = X['Votes'].astype(str).str.replace(',', '').astype(int)
X['Year'] = X['Year'].astype(str).str.replace('(', '').str.replace(')', '').astype(int)

# Encode categoricals (fast)
le = LabelEncoder()
for col in ['Genre', 'Director', 'Actor 1']:
    X[col] = le.fit_transform(X[col])

# Split data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Model (faster settings)
model = RandomForestRegressor(
    n_estimators=100,
    max_depth=15,
    random_state=42
)

model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

print("R2 Score:", r2_score(y_test, y_pred))
print("MAE:", mean_absolute_error(y_test, y_pred))
```

```
R2 Score: 0.32500939164430054
MAE: 0.8415694171973656
```

[80]

✓ 2s

```
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred)
plt.xlabel("Actual Ratings")
plt.ylabel("Predicted Ratings")
plt.title("Actual vs Predicted Movie Ratings")
plt.show()
```

