Modelo_preditivo

June 29, 2024

[21]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder
import pickle

# Load the dataset
df = pd.read_csv(r"C:\Users\gusta\Documents\desafio_LH\desafio_indicium_imdb.
 ↪csv")


# Drop the 'Unnamed: 0' column
df = df.drop(columns=['Unnamed: 0'])

# Convert 'Released_Year' to numeric
df['Released_Year'] = pd.to_numeric(df['Released_Year'], errors='coerce')

# Convert 'Runtime' to numeric
df['Runtime'] = df['Runtime'].str.replace(' min', '')
df['Runtime'] = pd.to_numeric(df['Runtime'], errors='coerce')

# Convert 'Gross' to numeric
df['Gross'] = df['Gross'].str.replace(',', '')
df['Gross'] = pd.to_numeric(df['Gross'], errors='coerce')

# Fill missing values
df['Certificate'] = df['Certificate'].fillna('Unrated')
df['Meta_score'] = df['Meta_score'].fillna(df['Meta_score'].mean())
df['Gross'] = df['Gross'].fillna(df['Gross'].mean())

numeric_cols = df.select_dtypes(include=[np.number]).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())

# Encode categorical variables
label_encoders = {}
```

```
categorical_columns = ['Certificate', 'Genre']

for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))
    label_encoders[col] = le

# Drop the specified columns
df = df.drop(['Director', 'Star1', 'Star2', 'Star3', 'Star4', 'Overview'],␣
 ↪axis=1)

# Split the data
X = df.drop(['IMDB_Rating', 'Series_Title'], axis=1)
y = df['IMDB_Rating']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Train the model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

# Features for predicting 'The Shawshank Redemption'
film_features = {
    'Released_Year': 1994,
    'Certificate': label_encoders['Certificate'].transform(['A'])[0],
    'Runtime': 142,
    'Genre': label_encoders['Genre'].transform(['Drama'])[0],
    'Meta_score': 80,
    'No_of_Votes': 2343110,
    'Gross': 28341469
}

film_df = pd.DataFrame([film_features])

imdb_rating_prediction = model.predict(film_df)
print('Predicted IMDB rating:', imdb_rating_prediction[0])
```

```
Predicted IMDB rating: 8.773999999999996
```

[22]:

```
with open('modelo_preditivo_imdb.pkl', 'wb') as file:
    pickle.dump(model, file)
```

[ ]:

```
```