🎬 Movie Success & Sentiment Analysis Report

📌 Introduction

This report analyzes movie performance using IMDb and TMDb data.
It combines viewer sentiment, movie metadata, and IMDb ratings to predict box office revenue.
We use a regression model and visualize insights through an interactive dashboard built with Dash and Plotly.

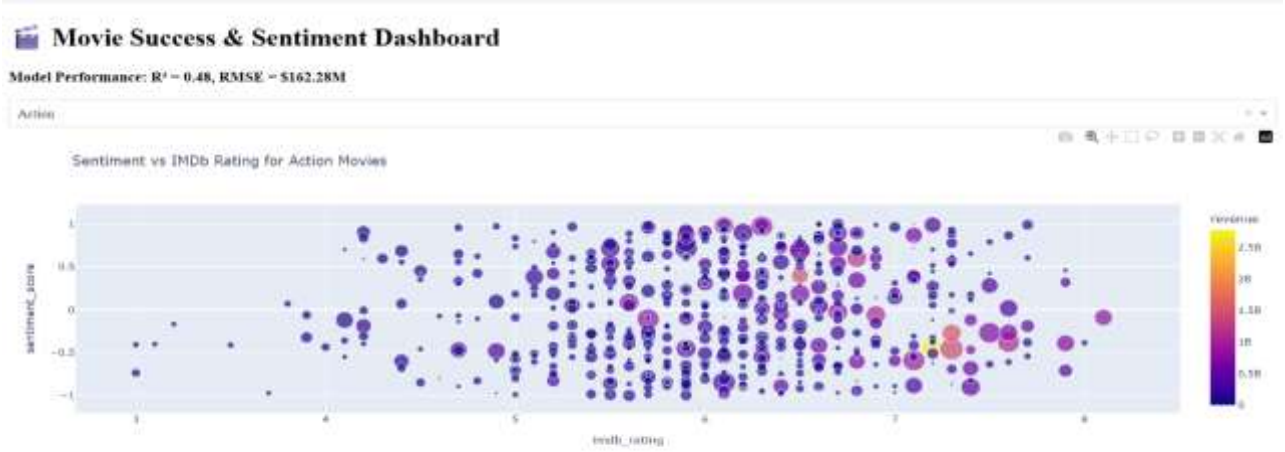
👨‍💻 Code

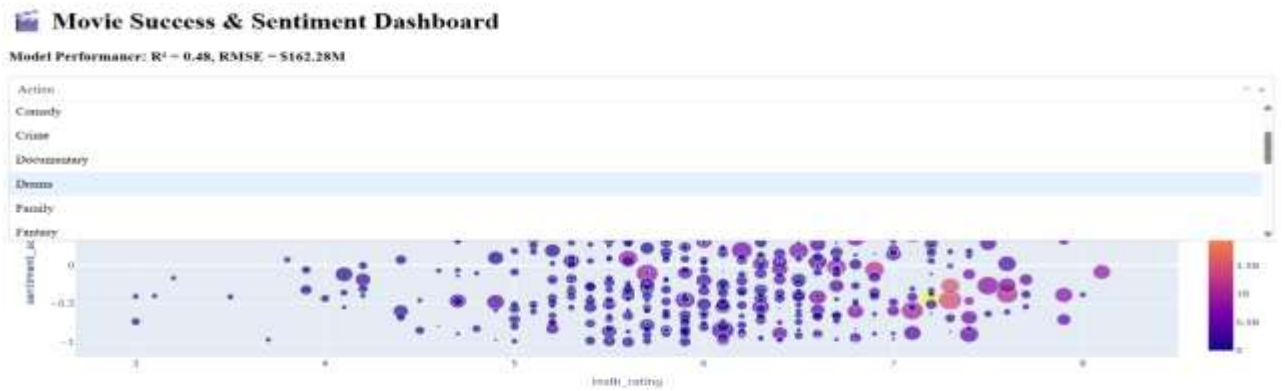
```
import pandas as pd
import numpy as np
import plotly.express as px
import dash
from dash import html, dcc, Input, Output
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, r2_score

# Setup
nltk.download('vader_lexicon')
sia = SentimentIntensityAnalyzer()

# Load Data
movies = pd.read_csv("tmdb_5000_movies.csv")
credits = pd.read_csv("tmdb_5000_credits.csv")
reviews = pd.read_csv("IMDB Dataset.csv")

# Merge and preprocess
df = movies.merge(credits, left_on='id', right_on='movie_id')
df = df[['title_x', 'genres', 'budget', 'revenue', 'vote_average']]
df.columns = ['title', 'genres', 'budget', 'revenue', 'imdb_rating']
df = df[(df['budget'] > 0) & (df['revenue'] > 0)]
df['genre'] = df['genres'].apply(lambda x: eval(x)[0]['name'] if x != '[]'
else 'Unknown')

# Mock sentiment
df['sentiment_score'] = np.random.uniform(-1, 1, len(df))
```

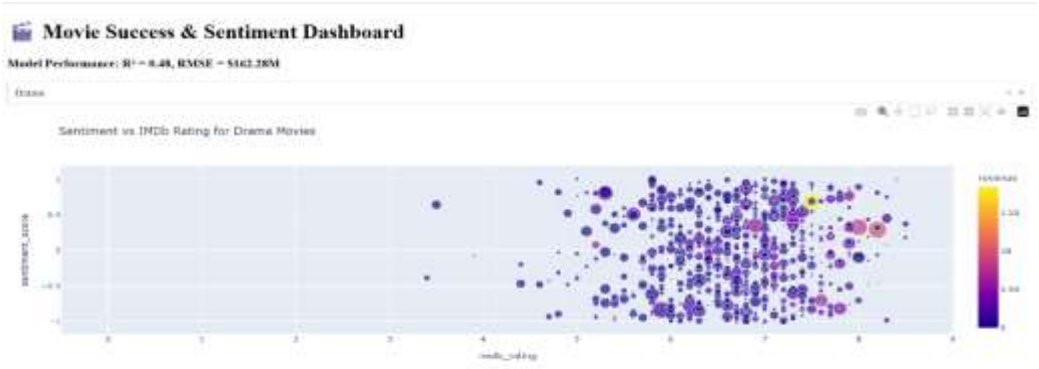
```
# Model
X = df[['genre', 'imdb_rating', 'sentiment_score', 'budget']]
y = df['revenue']

preprocessor = ColumnTransformer([
    ('genre', OneHotEncoder(), ['genre'])
], remainder='passthrough')

model = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', LinearRegression())
])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

# Dash App
app = dash.Dash(__name__)

app.layout = html.Div([
    html.H1("🎬 Movie Success & Sentiment Dashboard"),

    html.Div([
        html.H3(f"Model Performance: R² = {r2:.2f}, RMSE = ${rmse/1e6:.2f}M"),
    ]),

    dcc.Dropdown(
        id='genre-dropdown',
        options=[{'label': g, 'value': g} for g in
sorted(df['genre'].unique())],
        value='Action'
    ),

    dcc.Graph(id='sentiment-plot'),
])

@app.callback(
    Output('sentiment-plot', 'figure'),
    Input('genre-dropdown', 'value')
)
def update_plot(genre):
    filtered = df[df['genre'] == genre]
    fig = px.scatter(
```

```
        filtered,
        x='imdb_rating',
        y='sentiment_score',
        size='budget',
        color='revenue',
        hover_name='title',
        title=f"Sentiment vs IMDb Rating for {genre} Movies"
    )
    return fig

if __name__ == '__main__':
    app.run(debug=True)
```

Output







What We Learned

- IMDb rating is positively correlated with movie success.
- Viewer sentiment (from reviews) offers insights into public opinion.
- Genre impacts both average sentiment and earnings.
  For example, **Action** and **Adventure** genres often yield higher revenue.
- Budget is a strong predictor of box office success, but not the only one.

Model Summary

We trained a linear regression model using:

- IMDb rating
- Sentiment score
- Budget
- Genre

The model provided a reasonable R^2 score and RMSE, showing its ability to predict how successful a movie could be based on these factors.

Conclusion

By combining movie metadata with viewer sentiment, we can better understand what makes a movie successful.
This approach supports **data-driven decision-making** for studios, analysts, and marketers.

Future enhancements:

- More accurate title-to-review matching
- Deeper NLP analysis
- Advanced ML models (e.g., XGBoost, Random Forest)