Title: movie success rate ***prediction***

Student Name: V Rishi Shankar Sai Kapil\_2303a51083

Vishnu morthala \_2303a510j2

Jadala teja\_2303a51894

S Swasthik virinchi\_2303a51166

Raviteja\_2303a51164

Course: Applications of Data Mining

Instructor: Rama Krishna sir

Submission Date: 24-04-2025

🎬 Movie Success Rate Prediction

# Introduction

The success of a movie can often be influenced by key individuals like the hero, heroine, director, and comedian. This project uses a **rule-based approach** by assigning scores to prominent film personalities to estimate the likelihood of a movie being a hit.

# Objective

To create a **score-based system** that predicts the potential success of upcoming movies based on cast and crew reputations, without using machine learning algorithms.

# Dataset Description

* **Source:** Manually created Excel dataset – movie\_success\_rated\_dataset.xlsx
* **Data Includes:**
* Hero
* Heroine
* Comedian
* Villain
* Director

# Tools & Libraries

. Python 3.x

. Pandas

. NumPy

# Data Preprocessing

import pandas as pd

import numpy as np

# Load dataset

df = pd.read\_excel("movie\_success\_rated\_dataset.xlsx")

# Celebrity score mapping

celebrity\_scores = {

'Hero': {'ram charan': 9, 'nani': 7, 'vijay': 8, 'prabhas': 10, 'allu arjun': 9},

'Heroine': {'samantha': 9, 'rashmika': 8, 'kajal': 7},

'Comedian': {'brahmanandam': 9, 'vennela kishore': 8, 'ali': 7},

'Villain': {'jagapathi babu': 9, 'prakash raj': 8, 'sonu sood': 7},

'Director': {'rajamouli': 10, 'trivikram': 9, 'sukumar': 9}

}

# Convert names to scores

for col in celebrity\_scores:

df[col] = df[col].fillna("None").apply(lambda x: celebrity\_scores[col].get(str(x).strip().lower(), 0))

# Display processed data

print(df.head())

# Exploratory Data Analysis

import matplotlib.pyplot as plt

import seaborn as sns

# Set up visual style

sns.set(style="whitegrid")

plt.figure(figsize=(10, 6))

# Count plot of movie success

sns.countplot(x='Success', data=df, palette='viridis')

plt.title('Distribution of Movie Success')

plt.xticks([0, 1], ['Flop', 'Hit'])

plt.ylabel('Number of Movies')

plt.show()

# Box plots of score distributions by success

features = ['Hero', 'Heroine', 'Comedian', 'Villain', 'Director']

for feature in features:

plt.figure(figsize=(8, 5))

sns.boxplot(x='Success', y=feature, data=df, palette='pastel')

plt.title(f'{feature} Score by Movie Success')

plt.xticks([0, 1], ['Flop', 'Hit'])

plt.show()

# Correlation heatmap

plt.figure(figsize=(10, 6))

sns.heatmap(df[features + ['Success']].corr(), annot=True, cmap="coolwarm", linewidths=0.5)

plt.title("Feature Correlation with Success")

plt.show()

# Model Building

# Define a simple rule-based function to calculate total score

def calculate\_total\_score(row):

return row['Hero'] + row['Heroine'] + row['Comedian'] + row['Villain'] + row['Director']

# Apply the scoring logic

df['TotalScore'] = df.apply(calculate\_total\_score, axis=1)

# Define a threshold for success (customizable)

threshold = 40 # You can adjust this based on EDA

# Predict success based on threshold

df['Predicted'] = df['TotalScore'].apply(lambda x: 1 if x >= threshold else 0)

# Display results

print(df[['Hero', 'Heroine', 'Comedian', 'Villain', 'Director', 'TotalScore', 'Predicted']].head())

# Evaluation

# from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# if 'Success' in df.columns:

# accuracy = accuracy\_score(df['Success'], df['Predicted'])

# print(f"✅ Accuracy: {accuracy:.2f}")

# print("\n📊 Confusion Matrix:")

# print(confusion\_matrix(df['Success'], df['Predicted']))

# print("\n📋 Classification Report:")

# print(classification\_report(df['Success'], df['Predicted']))

# else:

# print("⚠️ 'Success' column not found in the dataset. Cannot evaluate model performance.")

# Conclusion

* This project implemented a **rule-based approach** to predict the success of movies using key roles such as Hero, Heroine, Comedian, Villain, and Director.
* By assigning **predefined scores** to celebrities and summing these scores, we developed a simple model to classify movies as **"Hit"** or **"Flop"** based on a score threshold.
* The model demonstrated **reasonable accuracy** when evaluated against real outcomes from the dataset.
* Key insights from the evaluation showed that roles like **Hero** and **Director** play a significant role in determining movie success.

 Although not as flexible or adaptive as machine learning models, this method is **transparent, fast, and easy to interpret**, making it ideal for early-stage predictions or when limited data is available.

# Future Enhancements

**🔮 Future Enhancements (Very Short Key Points)**

* **Dynamic Thresholding**: Adjust for genre, budget, or audience.
* **More Features**: Add budget, season, reviews, genre.
* **Machine Learning**: Use Random Forest or XGBoost.
* **Tuning**: Optimize model parameters.
* **Sentiment Analysis**: Use reviews or social media data.

# Appendix

- Full code in `diabetes\_prediction.py`

- Dataset file: `diabetes.csv`

- GitHub Repository: [Your GitHub URL]