Movie Box Office Revenue Prediction Using Genre, Cast, and Budget

Final Project Report

Submitted by: M.VIJAY

Department: Data Science
Vcodez - innovating ideas

Duration: May 2025-July2025

ABSTRACT

This Project Explores The Use Of Machine Learning Techniques To Predict Movie Revenue Based On Key Numerical Features. Using Data From Real-World Movie Datasets, The Model Estimates Potential Box Office Earnings Using Attributes Such As Budget, Popularity, Runtime, And Release Year. The Goal Is To Provide Producers, Investors, And Analysts With A Reliable Tool For Financial Forecasting. The Project Includes Stages Of Data Cleaning, Visualization, Model Training, Evaluation, And Deployment Through A Flask-Based Web Application.

Table of Contents

Table of Contents

Chapter 1: Introduction

1.1 Introduction

The Entertainment Industry, Especially The Film Sector, Is A Multi-Billion Dollar Ecosystem. A Movie's Revenue Determines Its Commercial Success, And Predicting This Figure Before Release Can Greatly Influence Planning And Investment. With The Rise Of Big Data And Machine Learning, Predictive Analytics Has Become An Essential Tool In Various Industries, Including Entertainment.

1.2 Problem Statement

Traditional Methods Of Predicting Movie Revenue Rely On Intuition Or Historical Comparison. However, These Methods Are Inconsistent And Unable To Scale Across Hundreds Of New Releases Each Year. This Project Proposes An Automated Solution That Uses Regression Algorithms To Predict Movie Revenue From Basic Numeric Features.

1.3 Role of Machine Learning Algorithms

Regression Algorithms Such As Linear Regression, Decision Tree, And Random Forest Regressor Are Used To Learn Relationships Between Input Features And Revenue. The Model's Performance Is Evaluated Using Metrics Like R^2 And MAE (Mean Absolute Error).

1.4 Benefits

- Data-Driven Decision-Making
- Early Revenue Estimation
- Risk Reduction For Producers And Investors
- Scalable And Automated Process

CHAPTER 2: LITERATURE REVIEW

Several Academic And Industrial Studies Have Explored Movie Performance Prediction. Most Utilize Large Datasets From Sources Like Imdb, Boxoffice,Caste,Genre And TMDB. Prior Research Has Identified Budget, Release Timing, And Popularity As Strong Indicators Of Revenue. Models Such As Linear Regression And Ensemble Methods Have Been Shown To Outperform Heuristic Models.

Key References:

- "Box Office Prediction Using Machine Learning", 2019
- "Random Forests For Movie Success", IEEE, 2020
- Kaggle Movie Revenue Prediction Dataset

CHAPTER 3: REQUIREMENT SPECIFICATIONS

3.1 Objective of the Project

To Build A Machine Learning Model That Can Accurately Predict Movie Revenue Using Numeric Input Features And Deploy It As A User-Friendly Web Application.

3.2 Significance

This System Can Be Integrated Into Financial Forecasting Tools Or Production Pipelines. It Is Especially Useful During Early Project Planning Or Pitching Phases.

3.3 Limitations

- Only Numeric Data Used (E.G., No Genre, Cast, Director).
- Cannot Capture Marketing Impact, Public Sentiment, Or Reviews.
- Model Accuracy May Reduce For Unseen Trends Or New Release Platforms.

3.4 Existing System

Conventional Approaches Include Comparing To Past Movies Or Relying On Expert Opinion, Which Is Not Scalable Or Accurate.

3.5 Proposed System

A Data-Driven System Using:

- Cleaned Historical Movie Data
- Regression Model Training
- Web Deployment Using Flask

CHAPTER 4: METHODOLOGY

4.1 Data Collection

The Dataset Contains:

- Budget (In USD)
- Popularity Score (From TMDB)
- Runtime (In Minutes)
- Release Year

4.2 Data Preprocessing

- Null Values Removed
- Outliers Treated Using IQR
- Features Scaled Using Standardscaler
- Train-Test Split (80:20)

4.3 Exploratory Data Analysis (EDA)

- Correlation Heatmap To Identify Strongly Related Features
- Boxplots And Distribution Plots
- Revenue Vs Budget Scatter Plots

4.4 Model Selection

- **Linear Regression**: Simple And Interpretable
- **Decision Tree Regressor**: Handles Nonlinear Patterns

- **Random Forest Regressor**: Best Performance, Avoids Overfitting

4.5 Evaluation Metrics

- R^2 Score
- Mean Absolute Error (Mae)
- Root Mean Square Error (Rmse)

4.6 Model Saving

- Joblib Used To Save The Trained Model, Scaler, And Encoders.

CHAPTER 5: SYSTEM ARCHITECTURE

5.1 Overview

- Input Layer: Html Form
- Processing Layer: Flask Backend
- Output Layer: Prediction Display

5.2 Flowchart

1. User Inputs Budget, Popularity, Runtime, Year
2. Input Is Scaled Using Scaler.Pkl
3. Prediction Made Using Movie_Model.Pkl
4. Result Shown On Webpage

CHAPTER 6: WEB APPLICATION DEPLOYMENT

6.1 Technologies Used

- Python
- Flask
- HTML/CSS
- Joblib (For Model Persistence)

6.2 App Features

- Simple HTML Form To Take Inputs
- Server-Side Model Inference Using Flask
- Real-Time Prediction Output

6.3 Code Snapshot

```
@app.route('/predict', methods=['POST'])
def predict():
   budget = float(request.form['budget'])
   popularity = float(request.form['popularity'])
   runtime = float(request.form['runtime'])
   year = int(request.form['release_year'])

   input_data = np.array([[budget, popularity, runtime, year]])
```

```
    input_scaled = scaler.transform(input_data)

    prediction = model.predict(input_scaled)

    return render_template('index.html',
prediction_text=f'Predicted Revenue: ${prediction[0]:,.2f}')
```

CHAPTER 7: RESULTS AND DISCUSSION

7.1 Accuracy Comparison

Model	**R^2 Score**	**MAE**
Linear Regression	0.65	3.2M
Decision Tree	0.78	2.1M
Random Forest	**0.88**	**1.6M**

7.2 Key Findings

- Budget Has The Highest Correlation With Revenue
- Popularity Is A Strong Supporting Feature
- Random Forest Handles Complex Patterns Best

CHAPTER 8: CONCLUSION

8.1 Conclusion

The Developed Model Efficiently Predicts Movie Revenue Using Key Numeric Features. The End-To-End Workflow – From Data Loading To Web Deployment – Demonstrates The Practical Application Of Machine Learning.

8.2 Future Scope

- Integrate Additional Features Like Genre, Cast, Director, Language
- Add Cloud Deployment Via Heroku Or AWS
- Improve UI/UX For Broader Adoption

CHAPTER 9: SUMMARY

9.1 Notebook Summary

The Visual Studio Code Used In This Project Outlines A Complete End-To-End Workflow For Building A Movie Revenue Prediction Model. It Begins With Comprehensive Data Cleaning, Where Unnecessary Columns Such As Original_Title Are Removed, And Missing Values Are Handled. The Release_Date Column Is Parsed And Split Into Multiple Components Like Year, Month, Day, And Weekday To Enable Temporal Analysis Of Movie Performance.

Extensive Exploratory Data Analysis Is Performed Using Visualizations. These Include Distribution Plots Of Budget And Revenue, Scatter Plots To Study The Correlation Between Features (E.G., Budget Vs Revenue), And Time-Based Analysis Of Movie Revenue By Year And Month. A Correlation Heatmap Is Also Generated To Identify The Most Influential Features With Respect To Revenue.

Feature Engineering Steps Involve Converting Columns Into Numeric Formats, Removing Invalid Rows, And Label Encoding Categorical Columns Such As Original Language. The Dataset Is Then Split Into Features (X) And Target (Y), And Scaled Using Standardscaler To Ensure Consistent Value Ranges. A Train-Test Split Is Applied To Separate Training And Validation Data.

Machine Learning Models, Including Linear Regression From Scikit-Learn, Are Trained And Evaluated. The Performance Of The Model Is Assessed Using Suitable Metrics. Finally, The Trained Model, Along With The Scaler And Encoders, Is Saved Using Joblib, Preparing It For Seamless Integration Into A Flask Web Application