

A movie's box office

Designing a machine learning model that can predict the revenue a film will generate based on the information available before the movie's release.

Definition of the Business Problem

Problem Statement:

The Challenge is to develop a machine learning model that predicts a movie's expected box office revenue using pre-release information.

By analyzing historical data of past movies, we can find patterns and relationships between different features(e.g budget, cast popularity, genre) and actual box office results.

Business Objectives:

- Help studios and investors make data-driven decisions before production or release.
- Reduce financial risk by identifying potentially successful or risky projects early.
- Support better budget allocation for marketing and production.
- Provide insights into which factors most influence success

Problem definition and objectives

Why this matter:

Movie investments involve high financial risks and uncertainty.

A reliable predictive model can guide strategic decisions before large budgets are committed.

Expected Impact:

- Improved financial forecasting in film production.
- Enhanced decision-making confidence for investors and studios.
- Foundation for predictive analytics tools in the entertainment industry.

Data Needed for Box Office Prediction

- Pre-release data only
- Key features:
 - Budget
 - Genre
 - Cast & Director Popularity
 - Release Date / Season
 - Social Media Reach
- Target: Box Office Revenue

The success of a movie prediction model depends strongly on the quality and variety of data used.

For this project, we focus on pre-release information — data that is available before the movie's release date — so predictions can be made early in the production process.

Type of Data Collected

Movie Budget:

Total production and marketing budget of the movie, which directly affects its visibility and quality.

Genre:

The movie category (e.g., action, comedy, drama). Different genres attract different audience sizes.

Cast and Director Popularity:

Measured by the number of previous successful films, awards, or social media followers.

Production Company:

Some studios have a record of producing high-earning films.

Data to Be Used for Prediction

Release Date / Season:

Timing influences audience turnout (e.g., holiday releases usually perform better).

Social Media Engagement:

Metrics such as trailer views, likes, and online mentions can indicate public interest.

Target Variable:

The box office revenue (total earnings after release), which the model will try to predict.

Data Sources

Public Movie Databases

- IMDb (Internet Movie Database) – for movie metadata, cast, genre, and release information.
<https://www.imdb.com>
- Box Office Mojo – for box office revenue and release schedules.
<https://www.boxofficemojo.com>
- The Numbers – for budget and revenue data.
<https://www.the-numbers.com>
- Social media APIs – Trailer views, engagement metrics
- TMDb (The Movie Database) API – for popularity scores and cast data.
- IMDb – Movie details, cast, genre
- Box Office Mojo / The Numbers – Revenue & budget
- TMDb API – Movie popularity & metadata
- Kaggle Datasets – Pre-cleaned movie data (e.g., TMDB 5000)

How to Collect and Prepare the Data

Collection Methods:

- Download from Kaggle
- Use TMDb / IMDb APIs
- Web scraping (BeautifulSoup, IMDbPy)
- Gather social media data

Preparation Steps:

- Clean missing or duplicate data
- Standardize units (e.g., USD)
- Select key features

- Split into train/test sets

ML algorithm

Since the goal is to predict a movie's box office revenue, this is a regression problem. We are trying to estimate a continuous value based on several input features such as budget, genre, director, and cast popularity.

For this project, we will use Linear Regression as the machine learning model. Linear Regression predicts the target variable as a weighted sum of input features. For example, it assumes that increasing the budget or adding popular actors will increase the predicted revenue in a roughly linear way.

Why do we use this ML?

Is simple to implement and interpret.

Provides a clear relationship between features (like budget, runtime, or cast popularity) and box office revenue.

Serves as a baseline model, so we can compare its predictions with more advanced models later if desired.

The model will be trained using the pre-release movie features as input and the actual historical box office revenue as the target --> it can estimate revenue for new movies based on similar features.

Expected Insights

- **Key Revenue Drivers:**
The model identifies which factors (like budget, genre, director, or cast popularity) most affect box office success.
- **Budget-to-Revenue Relationship:**
Shows how production budget influences revenue and helps find the ideal investment range for profit.
- **Genre and Seasonal Trends:**
Reveals which genres perform best in certain seasons (e.g., action in summer, dramas in winter).
- **Predictive Benchmarking:**
Estimates expected revenue for unreleased movies based on pre-release data, supporting data-driven planning.
- **Comparison Between Movies:**
Allows comparisons between similar films to see which features lead to better results.
- **Marketing and Casting Insights:**
Helps guide marketing budgets, casting, and release strategies using measurable impacts on revenue.

Implementing Movies Prediction Model

1. Load data

```
(venv_boxoffice) alicexa@alicesas-MacBook-Air movie_boxoffice_prediction % python boxoffice_model.py
Libraries loaded successfully!
Movies dataset:
  budget  genres  homepage  title  vote_average  vote_count
0  237000000 [{"id": 28, "name": "Action"}, {"id": 12, "nam...  http://www.avatarmovie.com/  Avatar  7.2  11800
1  300000000 [{"id": 12, "name": "Adventure"}, {"id": 14, "...  http://disney.go.com/disneypictures/pirates/  Pirates of the Caribbean: At World's End  6.9  4500
2  245000000 [{"id": 28, "name": "Action"}, {"id": 12, "nam...  http://www.sonypictures.com/movies/spectra/  Spectre  6.3  4466
3  250000000 [{"id": 28, "name": "Action"}, {"id": 80, "nam...  http://www.thedarkknighttrises.com/  The Dark Knight Rises  7.6  9186
4  260000000 [{"id": 28, "name": "Action"}, {"id": 12, "nam...  http://movies.disney.com/john-carter  John Carter  6.1  2124

[5 rows x 20 columns]

Credits dataset:
  movie_id  title  cast  crew
0  19995  Avatar [{"cast_id": 242, "character": "Jake Sully", "... [{"credit_id": "52fe480925146c758ac23", "de...
1  205  Pirates of the Caribbean: At World's End [{"cast_id": 4, "character": "Captain Jack Spa... [{"credit_id": "52fe4292c3a36847f080b579", "de...
2  206447  Spectre [{"cast_id": 1, "character": "James Bond", "cr... [{"credit_id": "54085967c3a36829b5002c41", "de...
3  49026  The Dark Knight Rises [{"cast_id": 2, "character": "Bruce Wayne / Ba... [{"credit_id": "52fe4781c3a36847f81398c3", "de...
4  49529  John Carter [{"cast_id": 5, "character": "John Carter", "c... [{"credit_id": "52fe479ac3a36847f8139aa3", "de...
(venv_boxoffice) alicexa@alicesas-MacBook-Air movie_boxoffice_prediction %
(venv_boxoffice) alicexa@alicesas-MacBook-Air movie_boxoffice_prediction %
```

2. Clean + preprocess

- Pick only the columns that matter for your prediction.
- Replace missing or zero values with median values, so your model doesn't break.
- Convert release date into a month number
- Convert the genre column from text/JSON into numeric one-hot flags (like "Action = 1, Drama = 0"), so the model can use them as input features.
- Remove any rows without a known revenue (the target variable), and fill the rest of the missing data with zeros.

```
# Select relevant columns
useful_cols = ["budget", "popularity", "runtime", "release_date", "genres", "revenue"]
df = data[useful_cols].copy()

# Fill missing numeric values
df["budget"] = df["budget"].replace(0, np.nan).fillna(df["budget"].median())
df["popularity"] = df["popularity"].fillna(df["popularity"].median())
df["runtime"] = df["runtime"].fillna(df["runtime"].median())

# Convert release_date → month
df["release_month"] = pd.to_datetime(df["release_date"], errors="coerce").dt.month
df.drop(columns=["release_date"], inplace=True)

# Convert genres JSON to dummy columns
def extract_genres(x):
    try:
        return [g["name"] for g in ast.literal_eval(x)]
    except Exception:
        return []

df["genres"] = df["genres"].apply(extract_genres)
all_genres = list(set([g for sublist in df["genres"] for g in sublist]))
for genre in all_genres:
    df[genre] = df["genres"].apply(lambda x: 1 if genre in x else 0)
df.drop(columns=["genres"], inplace=True)

# Remove rows with missing target and fill remaining NaNs
df = df.dropna(subset=["revenue"])
df = df.fillna(0)

# Split features and target
X = df.drop(columns=["revenue"])
y = df["revenue"]

print(f"✅ Dataset ready: {df.shape[0]} movies, {X.shape[1]} features")
return X, y, all_genres
```

3. Train model

- create a Linear Regression model from scikit-learn
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- use `model.fit()` command for training.
- It takes `X_train` and `y_train`, finds the best-fitting line through the data by minimizing squared errors, and stores those coefficients in `model.coef_` and `model.intercept_`.
- where `X_train` — the inputs the model will learn from. and `y_train` — the answers it must try to predict. then predict the revenue with `model.predict`

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from preprocess_data import prepare_dataset
import numpy as np
import joblib

def train_and_evaluate():
    # Load preprocessed data
    X, y, _ = prepare_dataset()

    # Split dataset
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    print(f"Training set: {X_train.shape}, Test set: {X_test.shape}")

    # Train Linear Regression model
    model = LinearRegression()
    model.fit(X_train, y_train)
    print("✅ Model training complete!")

    # Predictions
    y_pred = model.predict(X_test)

    # Evaluate performance
    mae = mean_absolute_error(y_test, y_pred)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    r2 = r2_score(y_test, y_pred)

    print("\n📊 Model Performance:")
    print(f"MAE: {mae:,.0f}")
    print(f"RMSE: {rmse:,.0f}")
    print(f"R² Score: {r2:,.3f}")
```

Note: We tested how well the model learned using data it hasn't seen before (`X_test`, `y_test`). Using evaluation metrics.

1. Mean Absolute Error (MAE): measures how far the predictions are from the true values on average. (Smaller MAE = better performance)

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html

2. Root Mean Squared Error (RMSE): measures the average *squared* error. It's often used when you care more about avoiding very large prediction errors.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

3. Coefficient of Determination (R^2): measures the total variation in your data is explained by the model. (Closer to 1 = the better)

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html

4. Output, Evaluate and visualize

Figure 1. Top Performing Genres by Average Revenue

Adventure, Fantasy, and Animation genres yield the highest mean revenues, indicating strong audience appeal and franchise potential. Action and Sci-Fi films also perform well, suggesting that high-budget, visually driven genres dominate box-office success.

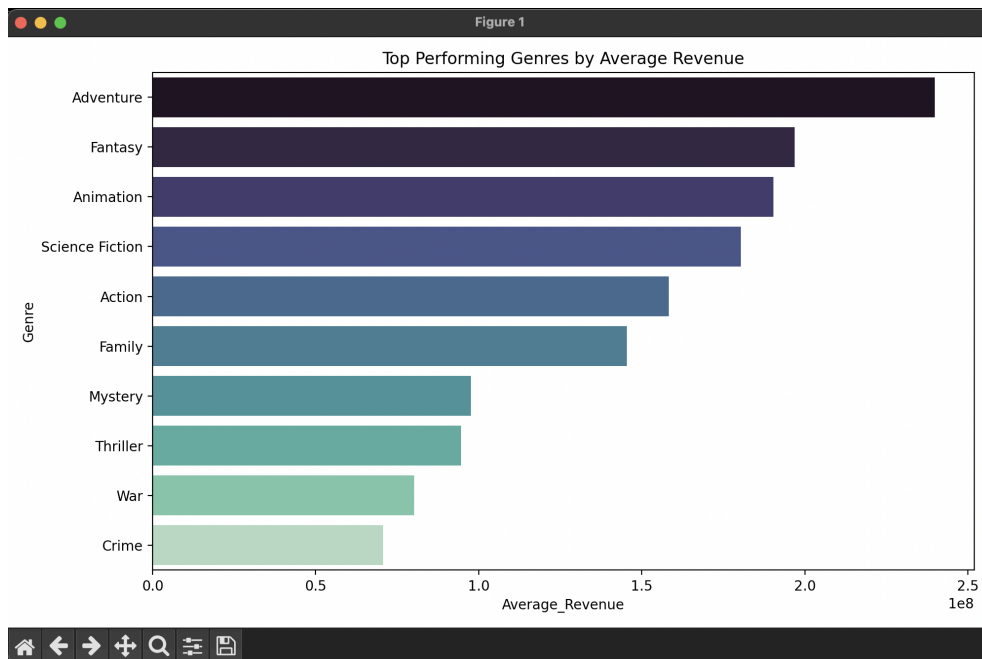


Figure 2. Top 10 Features Influencing Revenue

The regression model shows Animation, Adventure, and Family genres as key positive predictors of revenue. Popularity and runtime also contribute, implying that both content type and pre-release buzz significantly drive earnings.

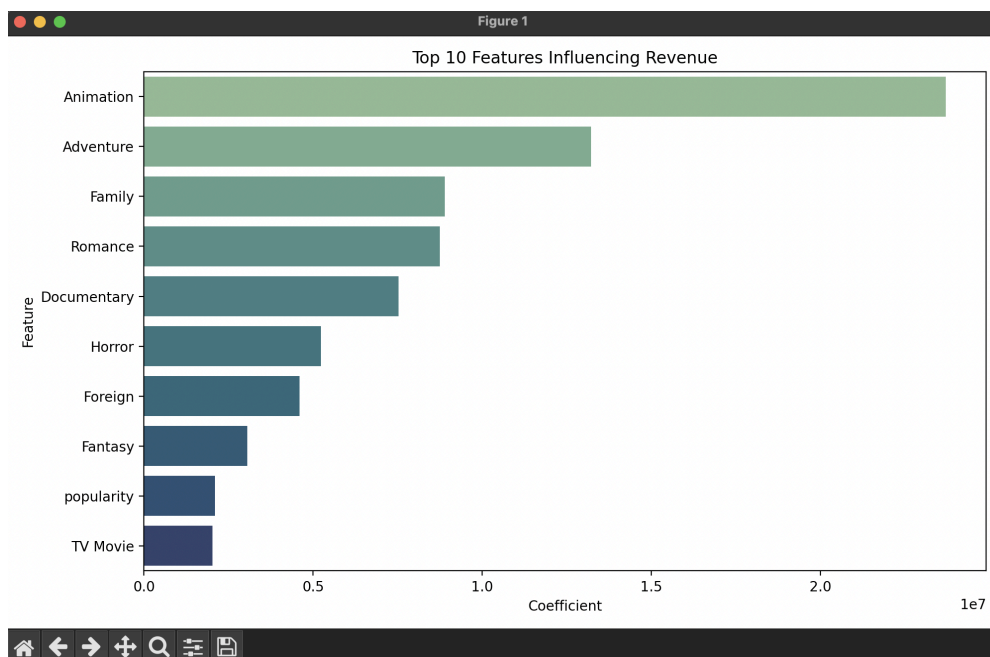


Figure 3. Budget vs. Actual Revenue

A positive correlation exists between budget and revenue, though variance increases for high-budget films. This suggests that while larger investments often yield higher returns, they also carry greater financial risk.

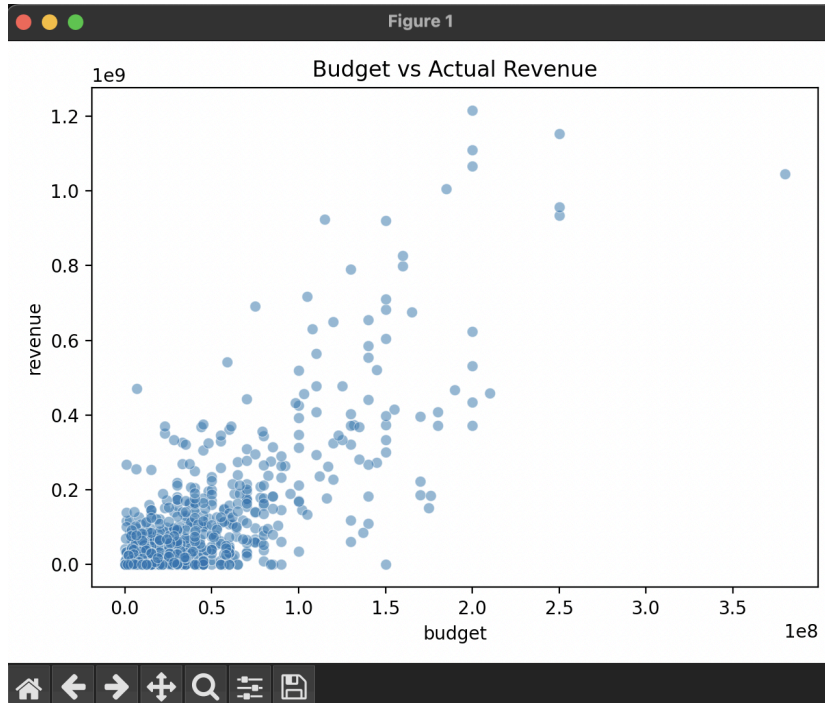


Figure 4. Average Revenue by Release Month

Peak revenues occur in May–July and November–December, aligning with summer and holiday release windows. This confirms seasonal audience behavior patterns that studios leverage for maximum box-office impact.

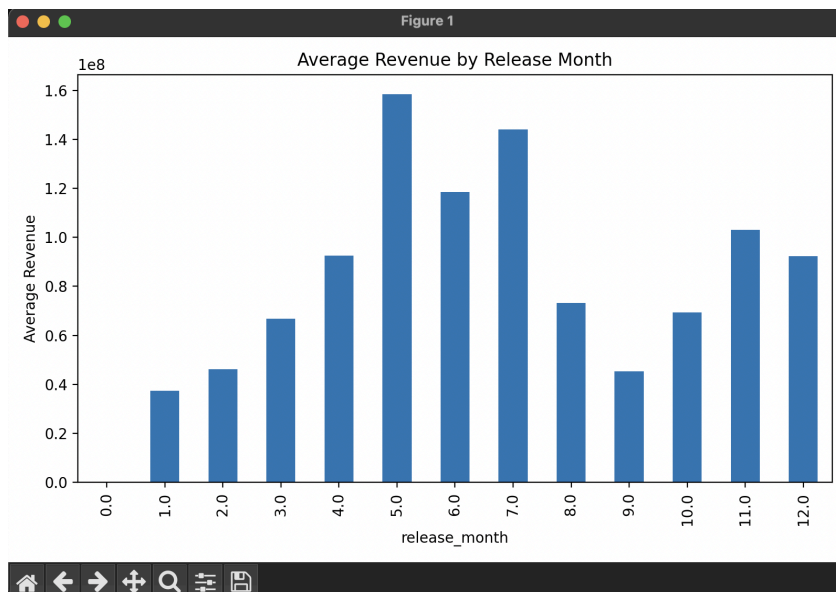


Figure 5. Predicted Revenue Compared to Historical Distribution

This histogram shows how the model's predicted revenue for a new film (red dashed line) compares with the distribution of past movie revenues. Most historical titles cluster at lower earnings, while the prediction lies in the mid-range, suggesting the film would perform moderately well compared with typical releases.

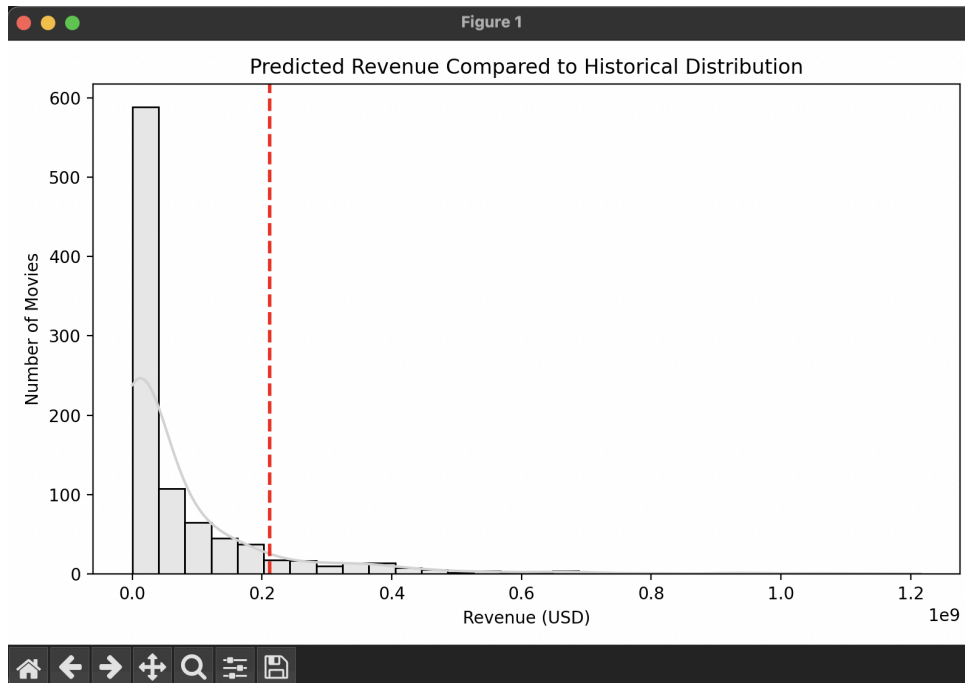


Figure 6. Actual vs. Predicted Revenue for Ten Most Accurate Movies

The side-by-side bars display actual (blue) and predicted (red) box-office revenues for the ten films the model estimated most precisely. The close alignment between bars shows that the model captures realistic patterns for these cases, giving confidence that its predictions generalize well on comparable films.

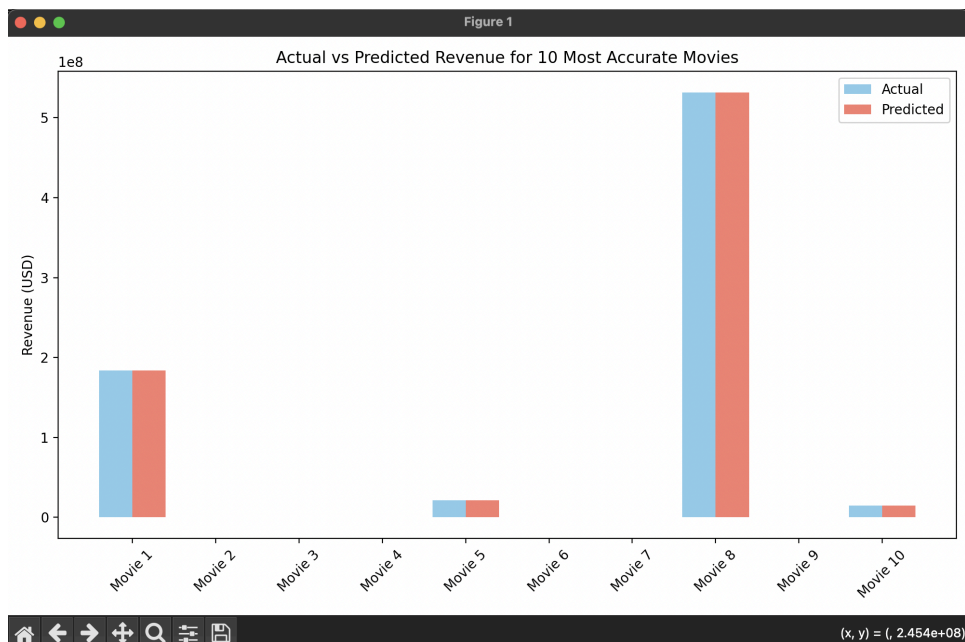


Figure 7. Popularity vs. Actual Revenue

Each point represents a film's popularity score before release and its final revenue. The upward trend reveals that higher pre-release popularity—often driven by marketing strength or well-known casts—tends to produce higher box-office returns. This visualization reinforces how marketing visibility and casting choices directly shape financial performance.

