BUAN 6341.003 – S24
Applied Machine Learning Project

TMDB Box Office Prediction

By-

Aditya Chaganti – vxc230016

Shobha Ojha – sxo210014

1. Introduction

1.1 Abstract:

In a world… where movies made an estimated $41.7 billion in 2018, the film industry is more popular than ever. But what movies make the most money at the box office? How much does a director matter? Or the budget? For some movies, it's "You had me at 'Hello.'" For others, the trailer falls short of expectations, and you think "What we have here is a failure to communicate."

In this project, we're presented with metadata on over 3000 past films from The Movie Database to try and predict their overall worldwide box office revenue. Data points provided include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries.

1.2 Objective:

The primary goal is to build a machine-learning model the prediction of revenue that a new movie can generate based on a few given input attributes such as budget, release dates, genres, production companies, production countries, etc. The modeling performance is evaluating based on the MSE and R2

2. Data Understanding

2.1 Data Description

The dataset presents a comprehensive collection of movie-related data, encapsulating 3,000 entries across various dimensions of cinematic analysis. It is structured into 23 variables that provide a wide lens through which the intricacies of film data can be explored. The dataset incorporates three variables of integer type (int64), including 'id', 'budget', and 'revenue', which offer numerical insights into the financial aspects of the film entries. Additionally, there are two variables, 'popularity' and 'runtime', represented as floating-point numbers (float64), reflecting the critical audience reception metrics and the duration of the films respectively. The remaining majority, 18 variables, are categorized as object data types, encompassing a spectrum of qualitative attributes such as 'belongs_to_collection', 'genres', 'homepage', 'imdb_id', 'original_language', 'original_title', 'overview', 'poster_path', 'production_companies', 'production_countries', 'release_date', 'spoken_languages', 'status', 'tagline', 'title', 'Keywords', 'cast', and 'crew'. These object type columns encode the categorical data that conveys descriptive details about each movie, from production backgrounds to linguistic and narrative elements, forming a dataset rich for exploratory and predictive analysis.

2.2 Data Dictionary:

Attribute Name	Data Type	Description
id	int64	Unique identifier for each movie
belongs_to_collection	object	Collection to which the movie belongs, if applicable
budget	int64	The budget allocated for the production of the movie
genres	object	List of genres associated with the movie
homepage	object	URL of the movie's official homepage
imdb_id	object	The IMDb identifier for the movie
original_language	object	The language in which the movie was originally produced
original_title	object	The original title of the movie
overview	object	A brief summary of the movie
popularity	float64	The popularity score of the movie on TMDB
poster_path	object	The path to the poster image for the movie
production_companies	object	List of production companies involved in the movie
production_countries	object	List of countries where the movie was produced
release_date	object	The release date of the movie
runtime	float64	The duration of the movie in minutes
spoken_languages	object	Languages spoken in the movie
status	object	The release status of the movie
tagline	object	The tagline of the movie
title	object	The title of the movie
Keywords	object	Keywords associated with the movie
cast	object	List of main cast members of the movie
crew	object	List of main crew members of the movie
revenue	int64	Box office revenue generated by the movie

2.3 Data Preparation:

- **Target Variable Definition:** The primary variable of interest, 'revenue' serves as the target variable for our predictive modeling. It is a continuous variable representing the movie's box office income, which we aim to predict based on the features present in the dataset.

- **Attribute Datatype Identification:** A thorough examination of the dataset reveals a mix of data types. While numerical variables like 'budget' and 'revenue' are straightforward, several categorical variables are encoded as object datatypes, such as 'genres' and 'production_countries'. These require careful extraction and encoding to be used effectively in prediction models.

- **Handling Categorical Variables:** The categorical variables in the dataset, which include 'original_language', 'status', 'genres', and others, were transformed to a suitable format for modeling. This involves extracting them from dictionary data and converting them to 'factor' or 'dummy' variables, which entails encoding the categorical data into a binary format that can be interpreted by machine learning algorithms.

- **Feature Engineering:** New features were derived from existing data to enhance the model's performance. For instance, the 'release_date' was parsed into 'release_year', 'release_month', and 'release_day', providing more granular temporal information. Logarithmic transformations were applied to 'budget' and 'revenue' to normalize their distributions and potentially improve the predictive quality of the models.

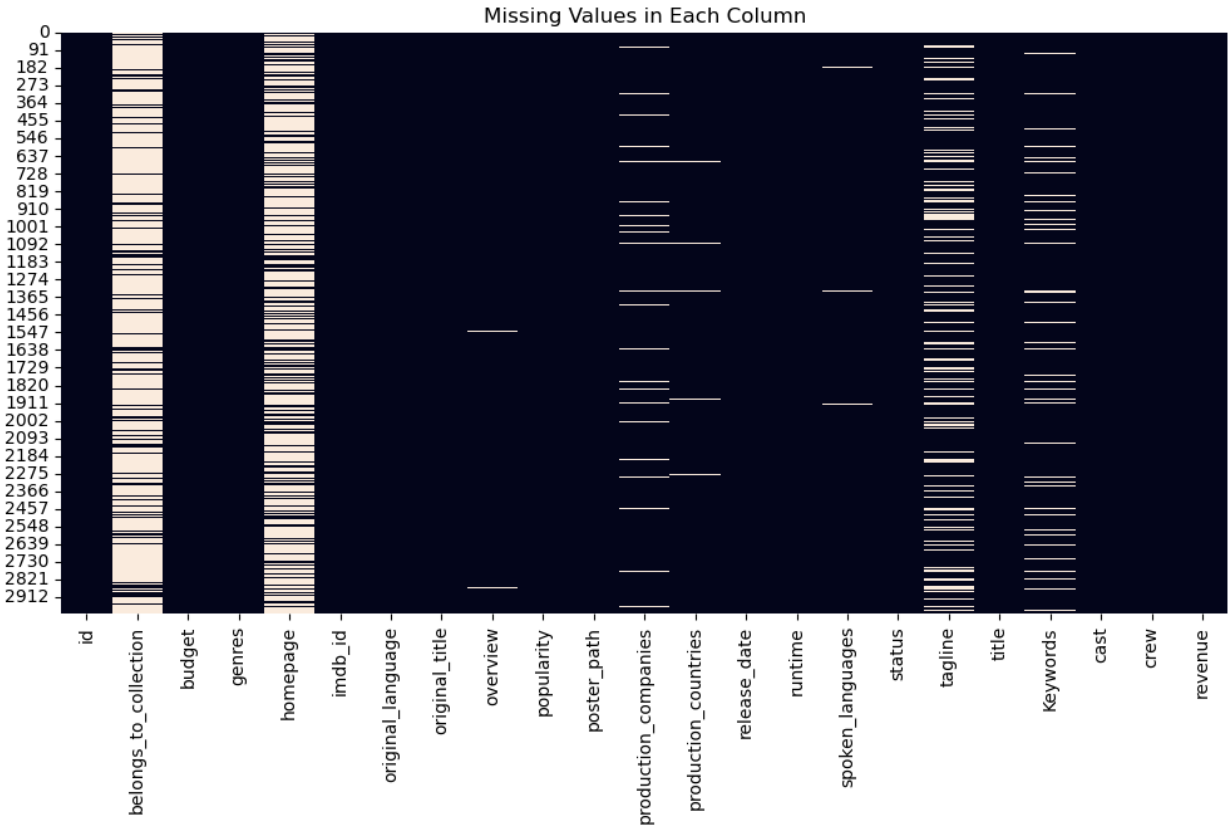
- **Scaling and Normalization:** Numerical features such as 'popularity' and 'runtime' were scaled to ensure that all features contribute equally to the model's prediction capability. Log transformation was done on ‘budget’ and ‘revenue’ for normalization.
  Through meticulous data preparation, the TMDB dataset was transformed into a clean, model-ready format that facilitates the development of accurate predictive models.

3. Data Cleaning

The data cleaning process is a critical phase in preparing the dataset for subsequent analyses and predictive modeling. In the TMDB Box Office dataset, specific steps were taken to address missing values and transform the data to a format with no null values suitable for machine learning algorithms.

3.1 Identification and Removal of Rows with NA:

Utilizing heatmaps for data visualization revealed extensive missing values in 'belongs_to_collection', 'homepage', and 'tagline' columns. Considering their high proportion of missing data and minor importance in our predictive model, we decided to remove these columns to ensure a more robust and focused dataset for analysis.



3.2 Handling Zero Values in the Budget Column:

Upon inspecting the 'budget' column, it was noted that while there were no null entries, a number of movies had a budget listed as zero. These zero values are not representative of actual movie production costs and could skew the analysis. To rectify this, these zero values were replaced with the mean budget of all movies in the dataset that had a non-zero budget.

3.3 Generating Dummy Variables for Categorical Data:

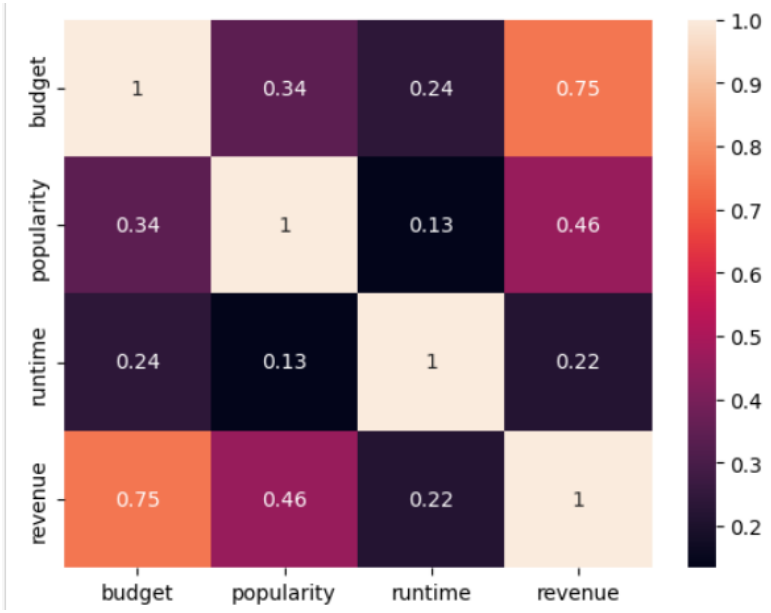
Using one-hot encoding, we transformed each category within a categorical variable into a new binary column. For instance, the 'original_language' column was transformed into multiple columns such as 'language_english', 'language_spanish', etc., with binary indicators representing the presence of each language.

4. Exploratory Data Analysis(EDA)

4.1 Correlation between Revenue and Budget, Popularity, Runtime

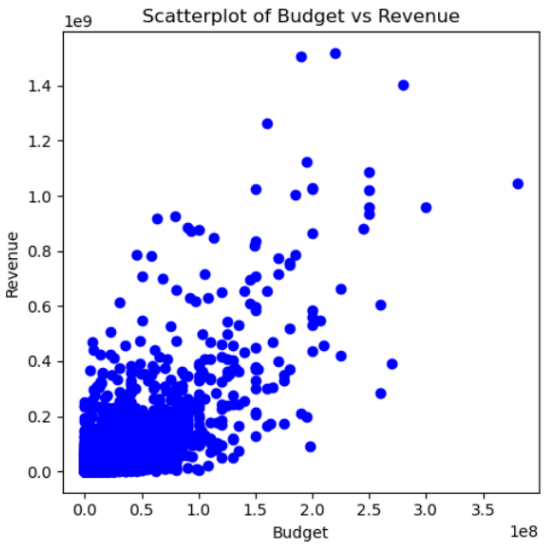
The correlation heatmap indicates a strong positive correlation between movie budget and revenue (0.75), suggesting higher budgets often lead to higher revenues. Popularity is moderately correlated

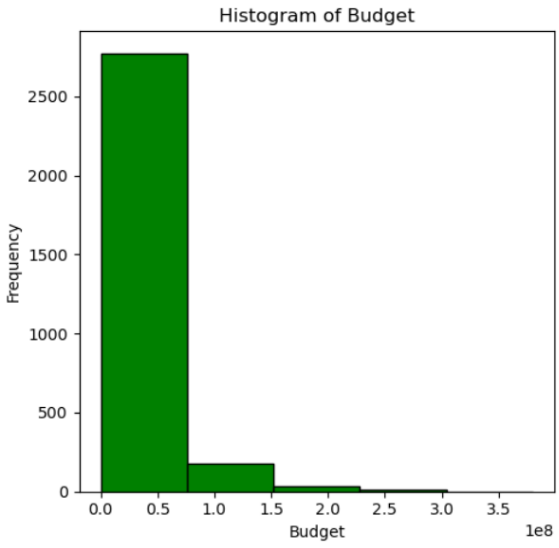
with revenue (0.46), while runtime shows a weak positive correlation with revenue (0.22), indicating less impact on earnings. These insights highlight budget as a potential key driver of a movie's financial success.



4.2 Budget vs Revenue

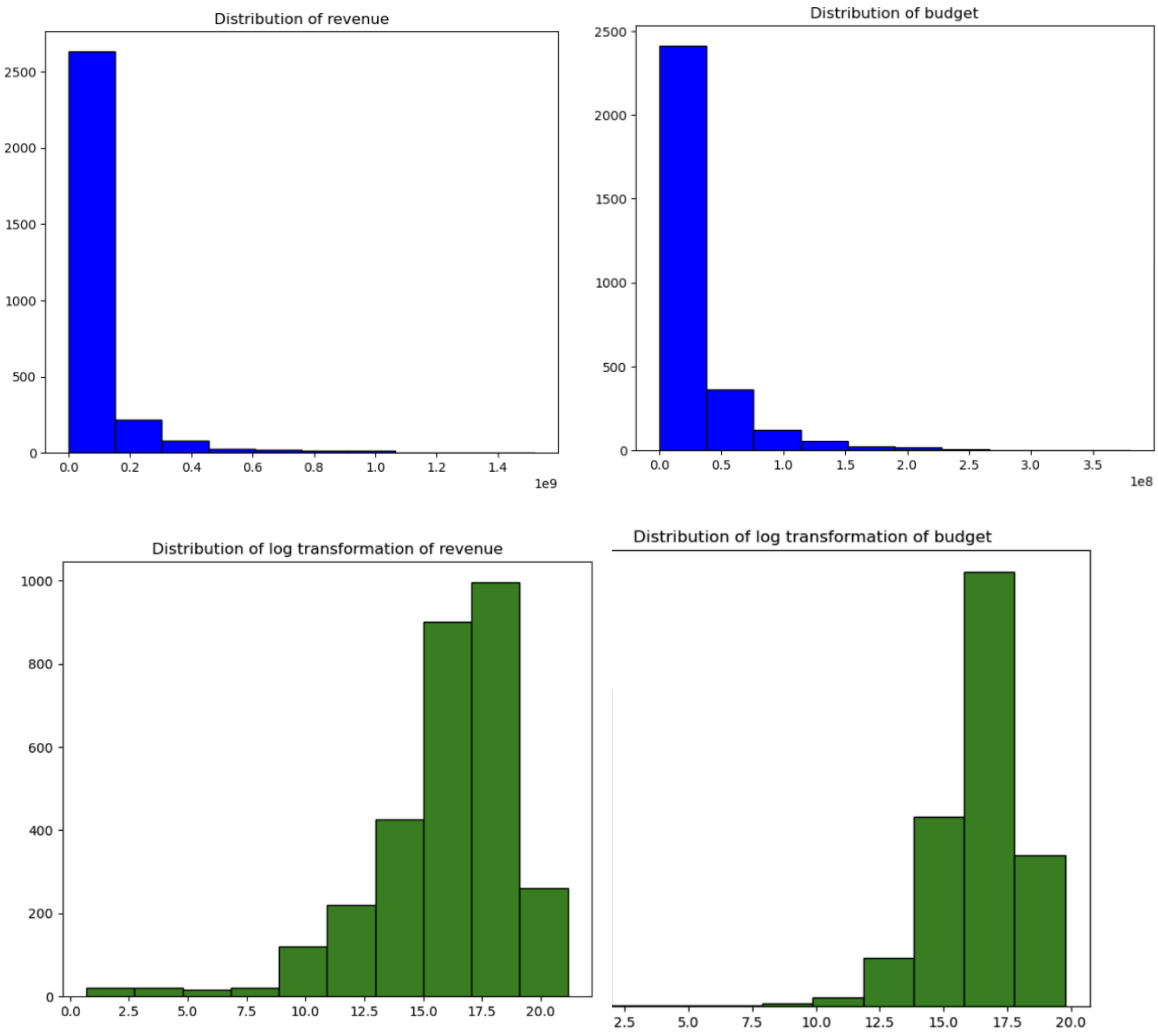
Both scatter plot as well as histogram shows, there is a strong relationship between revenue and budget. Revenue is highly dependent on the budget.





4.3 Transforming Budget into Log_Budget and Revenue into Log Revenue

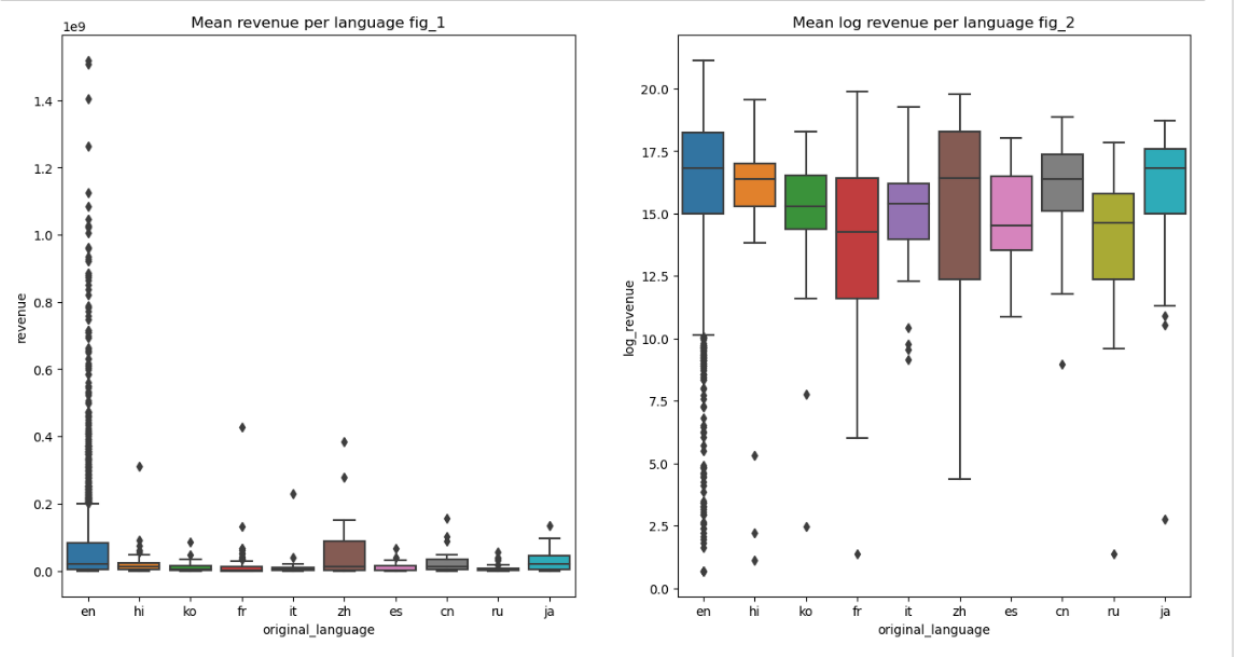
The budget and revenue data was transformed into a logarithmic scale to normalize the distribution and reduce skewness The original budget and revenue distribution is highly right-skewed (as we can see from the graph), indicating that most movies have a low budget, with a few outliers having very high budgets/revenue. After log transformation, the distribution appears more symmetrical and likely follows a normal distribution more closely. This transformation will improve the performance and validity of statistical analyses and regression models by meeting their assumptions of normality.



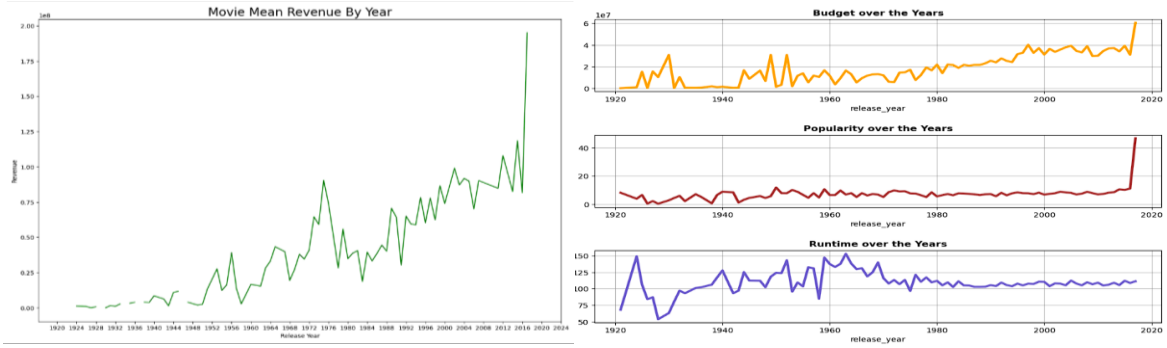
4.4 Movie revenues across different original languages

The boxplots compare the distribution of movie revenues across different original languages. The left boxplot shows a wide range of revenues with many outliers, especially in English-language films. The right boxplot, which presents log-transformed revenue, reveals a more normalized view, suggesting that while there are variations in revenue across languages. We can see that there are many outliers on the higher range for English-language films. At first we thought to remove those high magnitude outliers.

But then we considered the presence of those outliers reasonable, as their existence may be because of the fact that English speaking people all over the world are relatively more than other language speaking people. So to avoid any important information loss, we decided to keep those outliers as it is.



4.5 Movie revenue interpretation through time(in years)



The significant increase in mean movie revenue over time suggests a positive trend in the film industry's financial performance. This trend could be attributed to several factors, including inflation, advancements in technology leading to higher production values and ticket prices, globalization resulting in wider distribution channels and larger audience reach, and strategic shifts towards blockbuster franchises and high-budget productions driving higher box office returns.

Apart from Revenue, no other numerical variables show such strong correlation with release_year.

4.5 Data extraction for categorical variables

Several columns like ‘genre’, ‘production_companies’, ‘production_countries’, and keywords have data in dictionary format. Thus, it was important for us to extract the valuable categorical data from them and create dummy variables accordingly

```
df.dropna(subset=['genres'], inplace=True)
```

Before extracting the genre names from ‘genre’ column, we dropped the null values that were present as shown above in the code so that there would be no problem while parsing and extraction.

```
# Define a function to extract genre names
def extract_genre_names(genres_str):
    genres = ast.literal_eval(genres_str)  # Convert string representation to list of dictionaries
    return [genre['name'] for genre in genres]

# Apply the function to create a new column 'genres_names'
df['genres_names'] = df['genres'].apply(extract_genre_names)

# If you want to join the list of genre names into a single string
df['genres_names'] = df['genres_names'].apply(lambda x: ', '.join(x))

# Print the head of the DataFrame with the new column
df.head()
```

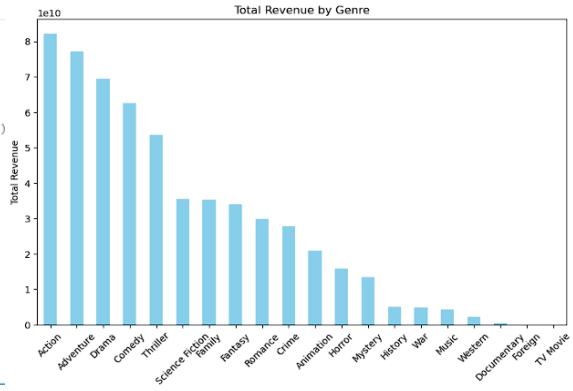
genres_names
Comedy
Comedy, Drama, Family, Romance
Drama

This code will create a new column named 'genres_names' with the extracted genre names for each row and then print the head of the dataframe with the new column included as shown above.

Same approach was used to extract valuable information from other dictionary datatype variables.

4.5 Creation of dummy variables

```
genres_df = df['genres_names'].str.split(', ', expand=True)

# Concatenate the 'revenue' column with the genres DataFrame
genres_df['revenue'] = df['revenue']

# Melt the DataFrame to have one row per genre per movie
melted_df = pd.melt(genres_df, id_vars=['revenue'], value_vars=[0, 1, 2, 3, 4], value_name='genre')

# Drop NaN values in the 'genre' column
melted_df.dropna(subset=['genre'], inplace=True)

# Group by genre and calculate the total revenue for each genre
genre_revenue = melted_df.groupby('genre')['revenue'].sum().sort_values(ascending=False)

# Plot the bar plot
plt.figure(figsize=(10, 6))
genre_revenue.plot(kind='bar', color='skyblue')
plt.title('Total Revenue by Genre')
plt.xlabel('Genre')
plt.ylabel('Total Revenue')
plt.xticks(rotation=45)
plt.show()
```



From the extracted genres_names we visualized top genres in a bar plot based on revenue generated by each of them.

Then we grouped the top revenue generating genre names and assigned them a value of ‘1’ and ‘0’ for all the remaining genre names.

en_or_not	Top_Genres	Top_prod_comp	Top_Countries	Top_months
1	1	1	1	0
1	1	1	1	0

Similar approach was used for other categorical variables to decide which were the top categorical variable values that significantly affected revenue.

After the required analysis we can observe that:

- Top values in genre are: 'Drama', 'Comedy', 'Thriller', 'Action', 'Adventure'
- Top values in production_companies are: 'Warner Bros.', 'Universal Pictures', 'Paramount Pictures', 'Twentieth Century Fox Film Corporation', 'Columbia Pictures','Metro-Goldwyn-Mayer (MGM)', 'New Line Cinema','Legendary Pictures','Walt Disney Pictures','Amblin Entertainment'
- Top values in production_countires are: 'United States of America','United Kingdom'
- Top values in release_month are: '5','6','7','11','12'
- Top values in keywords are: 'based on novel', 'sequel', 'superhero', 'marvel comic', 'based on comic', 'dystopia', 'saving the world', 'dc comics', 'magic', 'secret identity'

5. Data preparation

5.1 Data reduction

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2627 entries, 0 to 2999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   log_budget      2627 non-null   float64
 1   popularity      2627 non-null   float64
 2   runtime         2627 non-null   float64
 3   release_year    2627 non-null   int32
 4   en_or_not       2627 non-null   int32
 5   Top_Genres      2627 non-null   int64
 6   Top_prod_comp   2627 non-null   int64
 7   Top_Countries   2627 non-null   int64
 8   log_revenue     2627 non-null   float64
 9   Top_keywords    2627 non-null   int64
 10  Top_months      2627 non-null   int64
dtypes: float64(4), int32(2), int64(5)
memory usage: 225.8 KB
```

- There were total 41 variables in the data set after the required log transformation, extraction and dummy variable creation.
- Out of those 9 variables were selected as features for our final model

5.1 Train and test split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

The dataset is split into a training set (80%) and a testing set (20%) to evaluate model performance.

6. Modeling

6.1 Data Description

The refined dataset, derived from the TMDB Box Office information, consists of 3,000 movies, representing a rich diversity of cinematic attributes. For the purpose of this project, the 'revenue' variable has been designated as the target variable. It is continuous, indicating the box office earnings. The dataset encompasses 22 predictor variables, with 3 numerical variables ('budget', 'popularity', 'runtime') and the rest being categorical variables represented as object types after preprocessing.

6.2 Predictive Models

Given that the goal is to predict a continuous outcome (movie revenue), the problem is approached as a regression problem. The selected regression models are Linear Regression, Random Forest, Gradient Boosting, and XGBoost, which are suitable for handling both the numerical and categorical predictors after appropriate encoding.

6.3 Model Choice Rationale

Linear Regression:

A foundational modeling approach offering simplicity and interpretability.

It's particularly well-suited for establishing baseline performance.

Random Forest:

An ensemble method that is effective in reducing overfitting by averaging multiple decision trees.

It provides insights into feature importance and can handle a mix of numerical and categorical data without scaling.

Gradient Boosting:

Another ensemble technique that builds trees sequentially, each one correcting the errors of its predecessor.

It tends to deliver high performance, though it can be sensitive to overfitting and requires careful tuning.

XGBoost:

An optimized gradient boosting library that is renowned for its performance and speed.

It offers advanced regularization features, which helps in improving model's generalization.

6.4 Model Development

Model-1: Linear Regression

The dataset is split into a training set (80%) and a testing set (20%) to evaluate model performance.

An initial regression model is trained with all predictors, followed by feature selection to identify significant variables.

Model-2: Random Forest

Random Forest models are built using the same train-test split, and hyperparameters are tuned for optimal performance.

The model's feature importance metric is used to identify the most predictive attributes.

Model-3: Gradient Boosting

A Gradient Boosting model is trained, carefully controlling the learning rate and depth of trees to prevent overfitting.

Performance is evaluated using metrics such as R-squared and Mean Squared Error (MSE).

Model-4: XGBoost

The XGBoost model is implemented with cross-validation to optimize its parameters.

Evaluation metrics are documented, with a focus on the model's ability to generalize unseen data.

For each model, the performance of the training set is evaluated using appropriate metrics. The models are then validated against the test set to ensure they accurately predict revenue without overfitting the training data. Performance measures such as R-squared and MSE provide a quantitative basis for model comparison, and the best-performing model is selected for potential operational use.

Figures and tables illustrating the model's coefficients, feature importance, and performance metrics offer a clear visual and statistical understanding of the model's predictive capabilities and limitations.

7. Conclusion

- Based on the comparative analysis of the four predictive models presented in the graph, the Linear Regression model exhibits the least favorable performance metrics, registering the highest Mean Squared Error (MSE) at 4.32, coupled with the lowest R-squared (R^2) value of 0.402. These figures suggest a significant discrepancy between the predicted values and the actual values, alongside a suboptimal fit to the data variability.

- Conversely, the Gradient Boosting model emerges as the most proficient, characterized by the lowest MSE of 3.43, which implies more accurate predictions, and the highest R^2 of 0.67, indicating a superior fit to the data's variance.

- Considering this evidence, we have decided to proceed with the Gradient Boosting**.**

8. Potential improvements

- Clean data more thoroughly
- Create further dummy variables directors, actors etc
- Try to extract any kind of numerical values from the cast and crew column like the number of actors,etc.

9. References

- Data: Source (https://www.kaggle.com/zero92/tmdb-prediction/data)
- TMDB Box Office Prediction - Discussion -https://www.kaggle.com/c/tmdb-box-office-prediction/discussion
- Python (pandas, scikitlearn) for data cleaning and modeling
- For coding and ML-related questions- https://github.com/saphal/TMDB-Revenue-Prediction