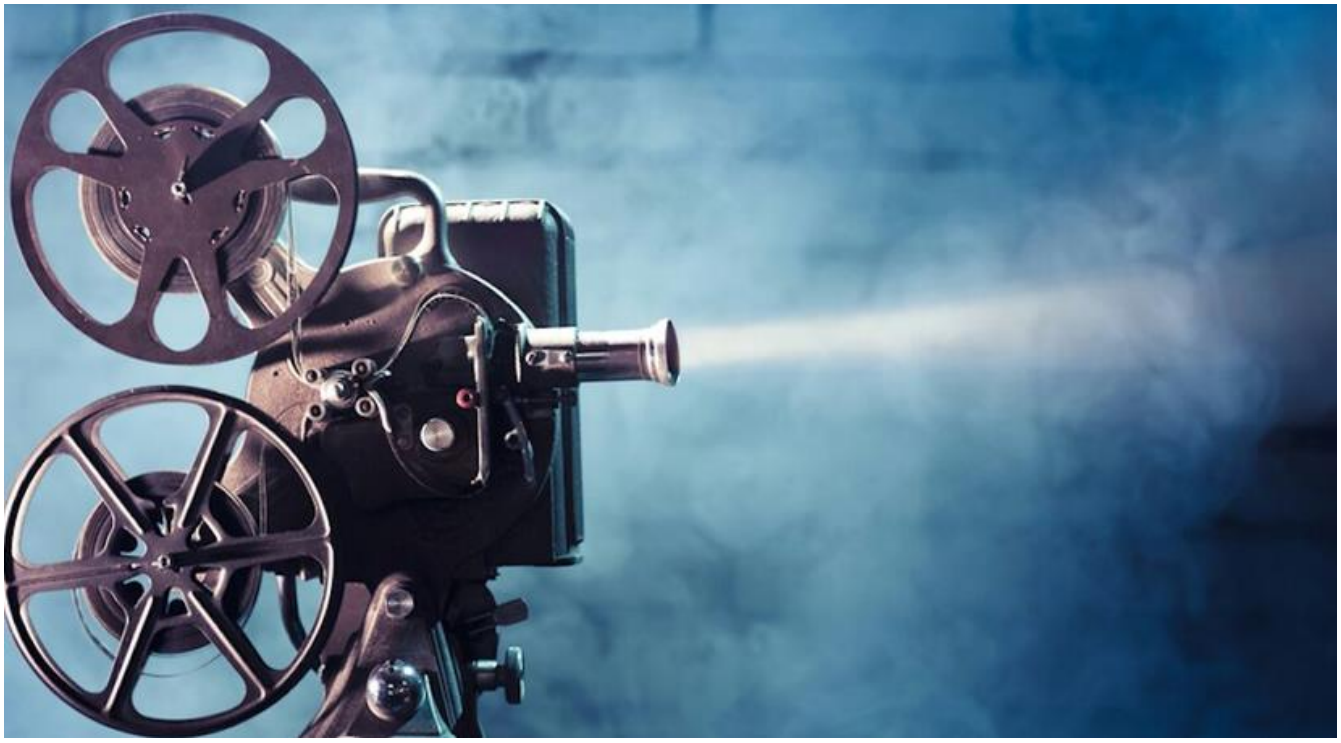
The business of the Seventh Art: predicting movies revenue.

Sebastiano Denegri

January 31st, 2021

Oh how Shakespeare would have loved cinema!
— Derek Jarman, Dancing Ledge.



Supervised Learning: Regression – IBM / Coursera.

Table of contents.

1. Executive Summary

In this project I've analyzed a dataset about movies to better understand what features have the strongest impact on the box office revenue. I've developed a **Linear Regression** model with main focus on **interpretation**.

Target market of the project are Movie Production Companies interested in knowing more about the relationship between the exploitable features (for instance: Budget) and the target (Revenue), and also what the actual impact of the available variables when it comes to generating Revenue is.

Database has been cleaned (by removing duplicates, missing values, and outliers), analyzed, and transformed, using Polynomial Features and nominal One-Hot-Encoding types of transformation.

Attributes which were either not relevant, redundant, or not available at the time of attempting a prediction were dropped.

Selected model: **LASSO (Least Absolute Shrinkage and Selection Operator) Regression - L1 penalty.**

Model performance was measured through cross-validated out-of-sample testing. Metrics used: Coefficient of Determination (R^2), and Mean Squared Error.

Model's results are as follows:

- Coefficient of Determination (Cross-Validated): 0.6656741789026847
- Mean Squared Error (Cross-Validated): 1.2415715607436606e+16
- Best hyperparameter (Cross-Validated): 1,000,000.0
- Number of coefficients equal to zero: 232

Top 10 most relevant features:

	Feature	Type of Correlation
0	Budget*Runtime	Positive
1	Collection	Positive
2	Samuel L. Jackson Dan O'Connell	Positive
3	Animation Family	Positive
4	Steven Spielberg	Positive
5	Samuel L. Jackson John T. Cucci	Negative
6	Adventure Fantasy	Positive
7	Samuel L. Jackson Hans Bjerno	Positive
8	Adventure Family Fantasy	Negative
9	Frank Welker Barbara Harris Hans Zimmer	Positive

2. Introduction.

2.1. Scope of the project.

In this project I've analyzed a dataset about movies to better understand what features have the most impact on the box office revenue. I've developed a Linear Regression model with main focus on **interpretation**.

The model has been trained only on features available at the time of attempting a prediction. A feature might have a strong explanatory value over the target, but if it's not available when making a prediction, then it cannot be used.

The goal of this project is highlighting, for business analysis purposes, what usable features are most impactful when it comes to generating movie's box office revenue, rather than making target predictions.

Target market of the project are Movie Production Companies interested in knowing more about the relationship between the exploitable features (for instance: Budget) and the target (Revenue), and also what the actual impact of the available variables when it comes to generating Revenue is.

In order to deliver a model able to explain the impact of the available variables over the target I've used a **machine-learning driven approach, and built a predictive Regression Model, not only capable of making predictions, but also of classifying/selecting features in order of importance.**

For the scope of this analysis, I've used a dataset from Kaggle.com, containing 2 files of popular movies across the globe.

Source of data: The Movie Database (TMDb).

2.2. Data Understanding.

The dataset contained 2 files with data on popular movies across the globe, from 1902 until nowadays. First file contained 7,101 movies, and second file contained 10,000 films. Each file contained the same 22 attributes.

After concatenating the 2 files, the movie dataset contained **17,101 observations with 22 attributes: 15 object-types and 7 numeric-types (5 floats and 2 integers).**

Attributes, data types, and other considerations about the dataset are as follows:

- TMDb_Id - Unique movie id as stored in TMDb's Database. Data type: integer. Range: 2 - 688131. **Not Relevant**.
- IMDb_Id - Unique movie id as stored in IMDb's Database (Internet Movie Database). Data type: object. 10,578 categories. **Not Relevant.**

- Title - English Title of the Movie. Data type: object. 10,307 categories. **Not Relevant.**
- Original_Title - Title of the Movie in the Original Language. Data type: object. 10,375 categories. **Not Relevant.**
- Overview - A summary of the Movie. Data type: object. 10,600 categories. **Not Relevant.**
- Genres - Genres the movie belongs to. Data type: object. 1,534 categories. **Feature selected for further analysis.**
- Cast - The Cast of the Movie. Data type: object. 10,589 categories. **Feature selected for further analysis.**
- Crew - The Crew of the Movie. Data type: object. 10,613 categories. **Feature selected for further analysis.**
- Collection - The Series the movie is part of or related to. If the movie does not belong to any series it is mentioned as "Single". Data type: object. 1,049 categories. **Feature selected for further analysis.**
- Release_Date - The day the Movie was released. Data type: object. 6,183 categories. **Feature selected for further analysis.**
- Release_Status - Is the Movie released. Data type: object. 5 categories. **Feature selected for further analysis.**
- Original_Language - The Original Language the Movie was made in. Data type: object. 52 categories. **Feature selected for further analysis.**
- Languages_Spoken - Languages Spoken in the Movie. Data type: object. 1,026 categories. **Feature selected for further analysis.**
- Runtime - Total runtime of the movie in minutes. Data type: float. Range: 0.0 - 400.0. **Feature selected for further analysis.**
- Tagline - Tagline of the movie. Data type: object. 7,814 categories. **Not Relevant.**
- Popularity - Popularity of the Movie. Data type: float. Range: 0.6 - 463.48699999999997. **Feature not available at the time of attempting a prediction.**
- Rating_average - Average rating of the Movie. Data type: float. Range: 0.0 - 10.0. **Feature not available at the time of attempting a prediction.**
- Rating_Count - Number of ratings the movie has received. Data type: integer. Range: 0 - 25,159. **Feature not available at the time of attempting a prediction.**
- Production_Companies - The Companies that made the Movie. Data type: object. 8,567 categories. **Feature selected for further analysis.**
- Country_of_Origin - The country the Movie was made in. Data type: object. 915 categories. **Feature selected for further analysis.**
- Budget - Total Budget of the Movie. Data type: float. Range: 0.0 - 387,000,000.0. **Feature selected for further analysis.**
- Revenue - Net amount that the Movie grossed for. Data type: float. Range: 0.0 - 2,797,800,564.0. **Target.**

3. Methodology.

In order to build an efficient Linear Regression model, with focus on **interpretation,** I followed the following methodology:

- Define the scope of the project: to develop a Linear Regression Model to predict movies revenue, with main focus on interpretation.
- Select an analytic machine-learning driven approach.
- Data understanding (content and format analysis).
- Data Preparation:
    - Data Cleaning
    - Data Mining (Exploratory Data Analysis, Variable Transformation, Feature Selection)
- Model Development:
    - Plain Linear Regression
    - Polynomial Transformation
    - Linear Regression with Regularization techniques
- Model Evaluation: compare performance of different models, and select the best for the project purpose.

3.1. Data Cleaning.

Duplicates.

The concatenated database had 6,215 duplicates, that were dropped. I then checked for duplicates in the attributes:

- TMDb_Id
- IMDb_Id
- Title
- Original_Title

I found and dropped 237 movies with the same TMDb_Id. Some of the observations had the same Title, or Original_Title, but they were not duplicate.

After removing the duplicates, the dataset had 10,649 observations. I also dropped some of the non-relevant attributes. **After the cleaning, the dataset had 14 attributes** (13 features, and the target).

Missing values.

The attribute Revenue had quite a big number of values (4,676) equal to 0, as well as the attribute Budget (4,651). Some movies (189) had a Runtime of 0 minutes. I converted the values equal to 0 to Nan values (the python's default missing value marker), and I dropped all

missing values in each attribute. **After dropping the Null Values, the dataset had 4,893 observations.**

Outliers.

The attributes Budget and Revenue had some very low values, which seemed to be mistakes:

- the Revenue lowest value was 5 USD, which seemed to be such an incredible poor result even for a very low-performing movie.
- the Budget lowest value was 1 USD; again, we don't need to be experienced movie producers to realize that this cannot be possible.

The number of movies made with a budget lower than 100,000 USD, in the dataset, were just 47 (less than 1% of total dataset). In the movie history, there have been movies made with a budget lower than this: *The Blair Witch Project* had a budget of around 60,000 USD, *Following* (by Christopher Nolan) was made on a budget of 6,000 USD, *Paranormal Activity* had a budget of 15,000 USD, *Clerks* was written and directed by Kevin Smith for about 27,000 USD... (for more information on low-budget movies, check this link: low-budget film). However, in the dataset, I found **19 observations** whose budget was less than 1,000 USD: these data points seemed to be mistakes, beyond any possible doubt.

I dropped the movies which had a budget lower or equal to 1,000 USD.

The number of movies with Revenue lower or equal to 100,000 USD, in the dataset, were only 75. Again, in the movie history, there have been cases of films who grossed even less than 10,000 USD (even less than 1,000 in few cases). In the dataset, however, I found **3 observations** who grossed less than 100 USD: these data points seemed to be mistakes, beyond any possible doubt.

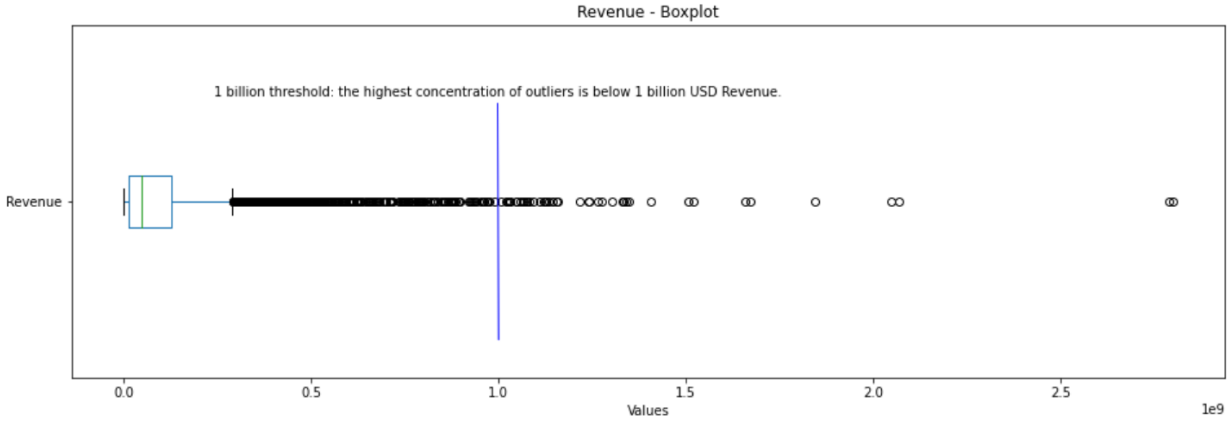
I dropped the movies, in dataset, which grossed less than 100 USD.

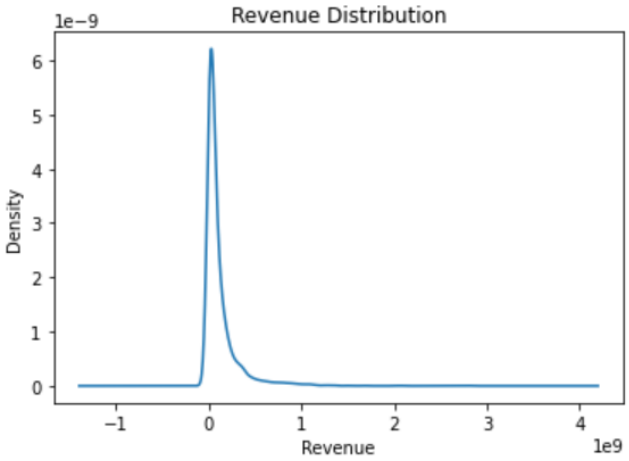
After removing the outliers (that were considered mistakes), **the movie dataset had 4,871 data points.**

After the cleaning, the lowest-grossing movie, in the dataset, was *The Room* (Revenue: 1,800 USD), and the lowest-budget movie was *Following*, by Christopher Nolan (Budget: 6,000 USD). Both data points were correct (see this link and this link).

After dropping the Revenue and Budget outliers, which were mistakes beyond any possible doubts, **I continued with the Outlier Analysis**.

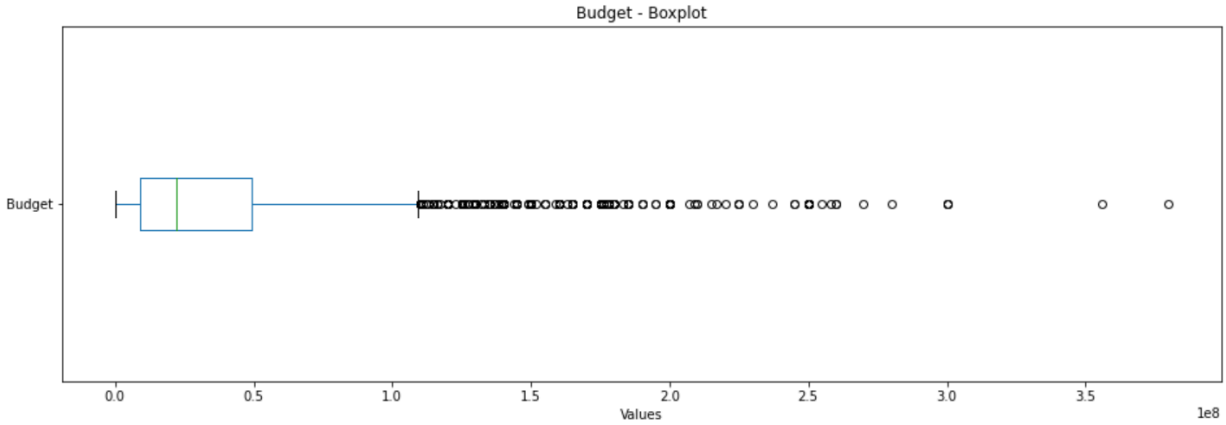
Revenue.

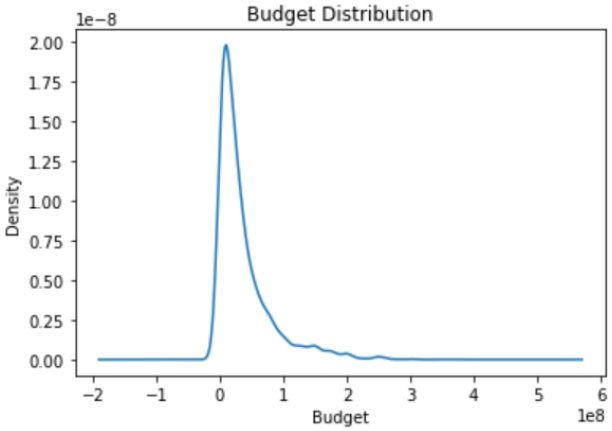




- The Target variable, Revenue, was not normally distributed, and it was actually very right skewed: all outliers were beyond the max limit of the no-outlier region (ca 290 mil).
- The number of outliers was quite substantial: 486. Almost 10% of observations grossed more than 290 million.
- 90% of outliers were below the 1-billion threshold.
- The number of observations above the 1-billion threshold was not substantial: 44 (less than 1% of total dataset).
- The extreme outliers belonged to very successful movie franchise (The Lord of Rings, The Dark Night, The Avengers, Harry Potter, Pirates of Caribbean, Star Wars...) or, in few cases, they were very successful stand-alone movies: Joker, Finding Dory, Titanic...
- These outliers didn't seem to be mistakes or data aberrations, but a normal part of the data distribution, due to its natural variation.

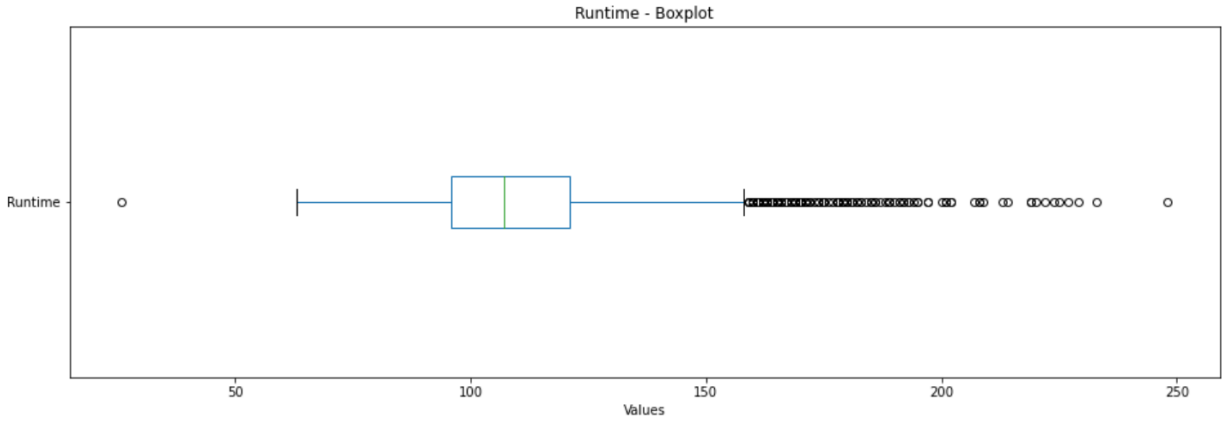
Budget.



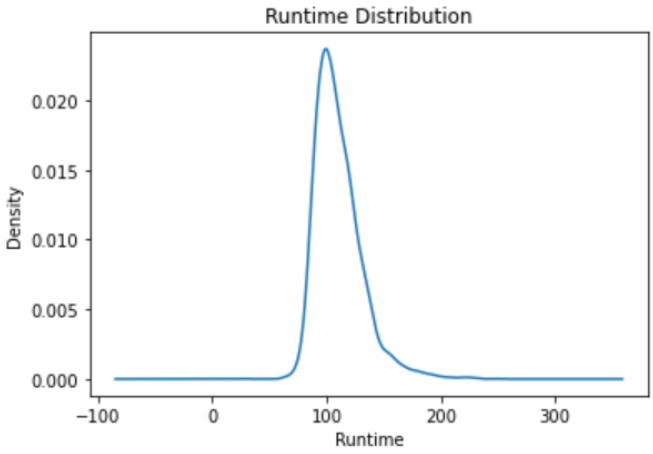


- The Budget distribution was, again, quite right skewed: all outliers were beyond the max limit of the no-outlier region (ca 110 mil).
- The number of outliers was again quite substantial: 357 (7.3% of total dataset).
- The outliers were famous movie which, probably, required some extra efforts from a production budget perspective: Inception, The Dark Knight, Avengers, Interstellar, Fantastic Four, Batman & Robin...
- The Budget outliers didn't seem to be mistakes or data aberrations, but a normal part of the data distribution, due to its natural variation.

Runtime.



- Runtime data distribution was right skewed.
- There was 1 outlier below the Min limit, a 26 minute-long movie: Mickey's Christmas Carol, which is quite a short movie. The observation was, therefore, correct.
- All other outliers were beyond the Max limit, that is 159 minutes (a little bit more than 2 and 40 mins). These are quite famous long movie such as: *Cleopatra, Gone with The Wind, Once Upon a Time in America…*

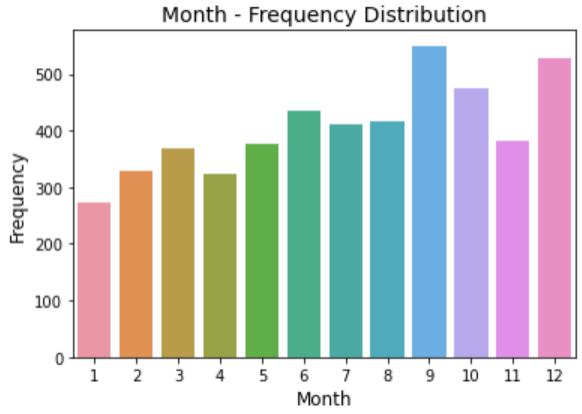


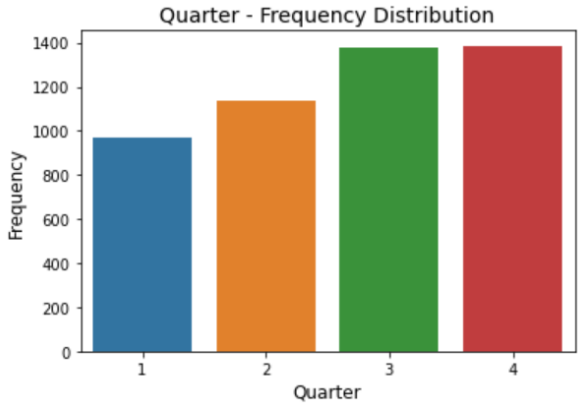
- 97% of movies were in the no-outlier region: between 1 hour (58.5 min) and 2 and 40 mins (158.5 min).
- Outliers accounted for 3% of the data (147 observations).
- These outliers didn't seem to be mistakes or data aberrations, but a normal part of the data distribution, due to its natural variation.

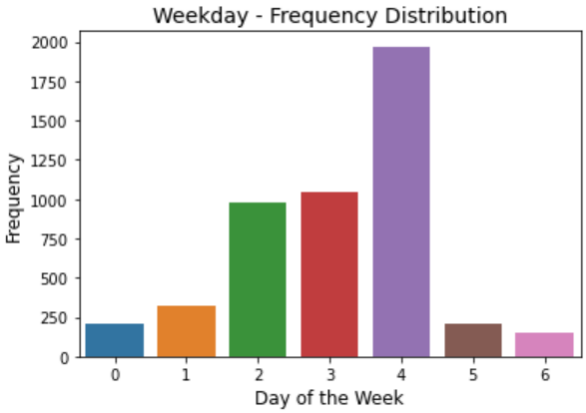
Categorical Variables.

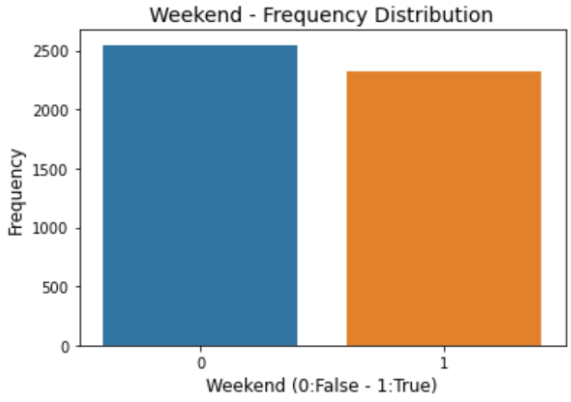
- I dropped the attribute Release_Status, since it had only 1 value for all observations: "Released".
- Some columns (like Cast and Crew) were JSON lists of 'values', that is they had a list (stored as a single value) for a single observation. For example, the 2005 *King Kong* had as Cast: "Christian Bale | Amy Adams | Bradley Cooper | ...", stored as a single value. I, therefore, created a function to count the number of "real" unique values per attribute (for instance, how many actors/actresses there are in the dataset). Results are below:
    - Genres: 19 categories.
    - Cast: 81,034 categories (actors/actresses).
    - Crew: 104,338 categories (directors, producers…). It's probably unlikely that the audience picks a movie because of a particular crew member, although a particular director can definitely have an impact on the box office: directors such as Woody Allen, Steven Spielberg, Martin Scorsese, just to name a few, have definitely a consistent number of fans, who won't miss their next movie. Furthermore, even other crew members can play a role: a certain executive producer, for instance, might have developed strong skills for producing box office successes, even though he's unknown to the public. Therefore, I consider Crew as an appropriate feature to predict movies' revenue.
    - Languages_Spoken: 62 categories.

    - Production_Companies: 5,120 categories. Generally, I wouldn't have considered a Production Company as a relevant feature for the audience on whether deciding to watch a movie or not, but this could actually be the case for very famous production companies such as Disney or Pixar, for instance. Therefore, I considered Production_Companies as a useful feature.
    - Country_of_Origin: 81 categories.
- Some attributes had a very high number of unique categories. Considering hardware and software limitations, I had to include in this analysis only the most frequent categories. More details about the selected categories are in the **"Feature Selection and Variable Transformations"** section.
- I changed Collection values into True and False categories: instead of having the name of the collection (Star Wars, Lord of the Rings…), or "single" for stand alone movies, the feature just showed whether a movie was part of a collection or not.
- I created new columns: "Quarter", "Month", "Weekday", and whether a movie was released over the weekend (Friday-Sunday), from the attribute Release_Date, and analyzed the frequency distribution.









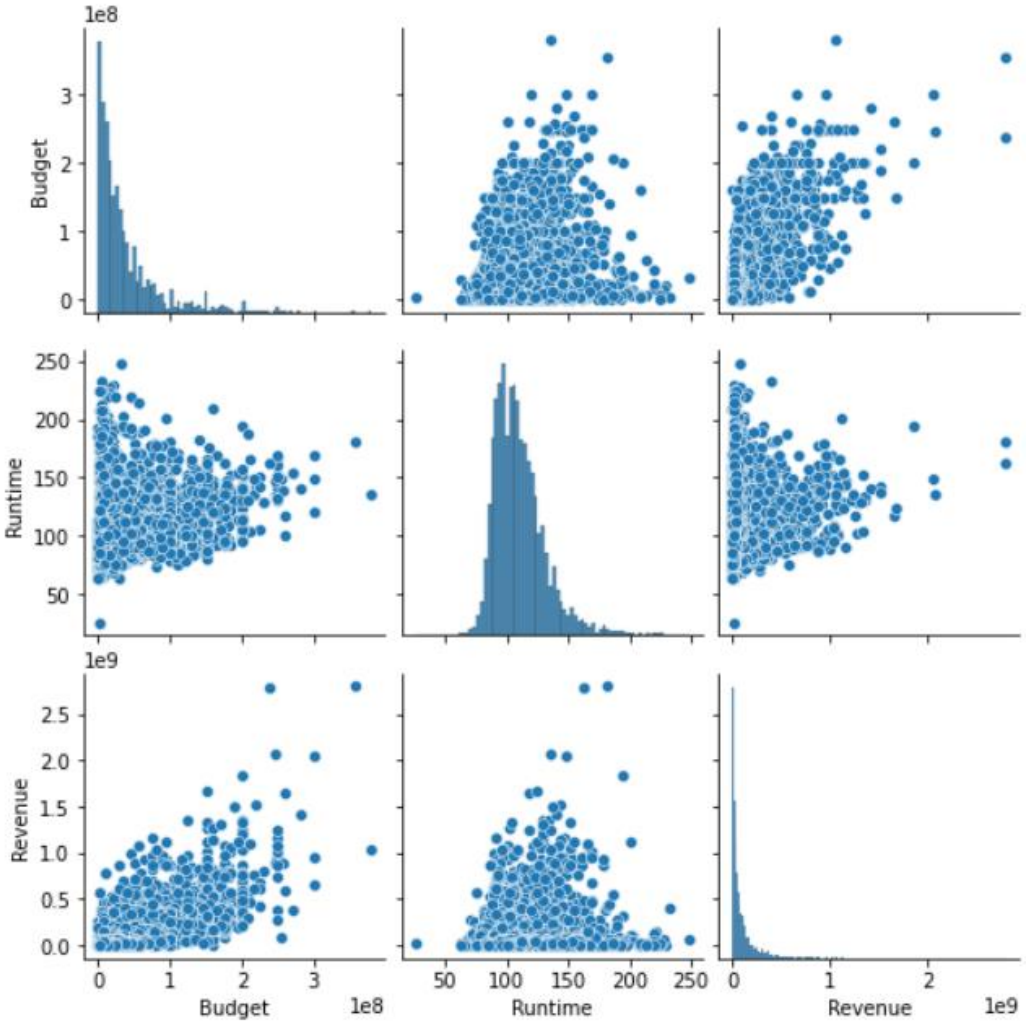
- Besides an imbalanced distribution for the attribute Weekday, (40% of movies had been released on Friday), no mistakes, aberrations, or problems were found in any of the attributes. I dropped the attribute Release_Date.
- Original_Language attribute: English was the original language for 4,488 (92%) movies in the dataset. Considering the greatly imbalanced distribution, I transformed the attribute into "English_Language", with True and False as values.

3.2. Exploratory Data Analysis.

Numeric-Type Data.

I performed either visual (Scatter plots) and statistical analysis (Pearson Correlation Coefficient) on numeric-type variables. I considered linearly correlated variables, only those ones with a Correlation Coefficient greater than 0.7.

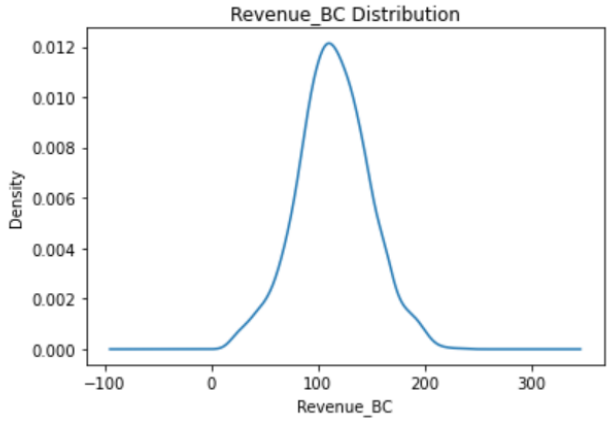
Pair Scatter plots of numeric-type variables



```
Pearson's correlation coefficient of Budget and Revenue:0.7245438708191911
Variables show a positive linear relationship.
Strong certainty in the calculated correlation coefficient.

Pearson's correlation coefficient of Runtime and Revenue:0.218589598786866
Variables don't show a linear tendency.
Strong certainty in the calculated correlation coefficient.
```

Budget and Revenue had a positive linear relationship, whilst Runtime and Revenue didn't show any linear tendency. We've already observed that the variable Revenue is not normally distributed. In Linear Regression, the Target Variable doesn't need to be normally distributed, but a Normal distribution of the Target might be helpful. I, therefore, applied Box-Cox transformation on the target variable to check whether the transformation helped "normalize" the target distribution, and strengthen the correlations with the independent variables.

```
Revenue_BC distribution skewness: 0.003598832066076301
Variable distribution is approximately Normal
```

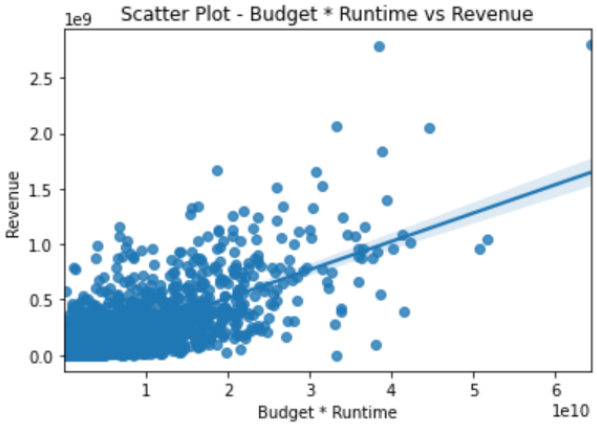



```
Pearson's correlation coefficient of Budget and Revenue_BC:0.657501396396.
Variables don't show a linear tendency.
Strong certainty in the calculated correlation coefficient.

Pearson's correlation coefficient of Budget_BC and Revenue_BC:0.687512081.
Variables don't show a linear tendency.
Strong certainty in the calculated correlation coefficient.

Pearson's correlation coefficient of Runtime and Revenue_BC:0.21247763260.
Variables don't show a linear tendency.
Strong certainty in the calculated correlation coefficient.

Pearson's correlation coefficient of Runtime_BC and Revenue_BC:0.22298861.
Variables don't show a linear tendency.
Strong certainty in the calculated correlation coefficient.
```

BoxCox succeeded in making the target variable distribution normal, but failed to strengthen any linear relationship between Runtime and Revenue, and the linear tendency between

Budget and Revenue is actually less evident after the transformation (I've tried with either BoxCox-transforming or not transforming the independent variables). Considering this, and the fact that a Normal distribution of the target variable is not a strict requirement for Linear Regression models, **I decided not to transform the target.**

Feature interactions: Budget * Runtime.



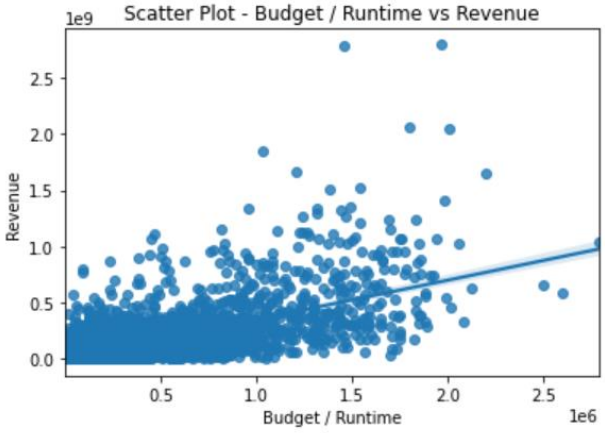
Pearson's correlation coefficient of Budget*Runtime and Revenue:0.7415612.

Variables show a positive linear relationship.

Strong certainty in the calculated correlation coefficient.

The feature interaction "Budget times Runtime" definitely showed a positive linear tendency with the target Revenue, even higher than Budget alone. This meant that, although Runtime alone was not enough to explain the Revenue, when multiplied by the Budget, Runtime also can have an explanatory value with regard to the revenue: in other words, a long movie, which also has a high budget, has more chances of being a box office success.

Feature interactions: Budget / Runtime.



Pearson's correlation coefficient of Budget/Runtime and Revenue:0.6705686.

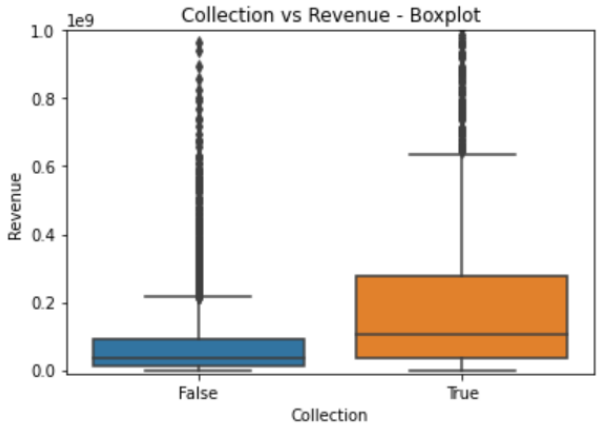
Variables don't show a linear tendency.

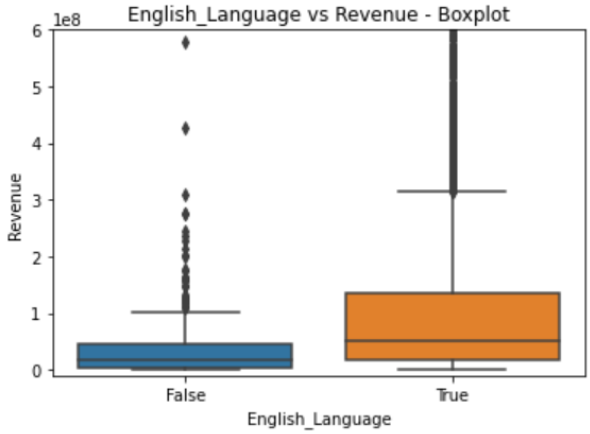
Strong certainty in the calculated correlation coefficient.

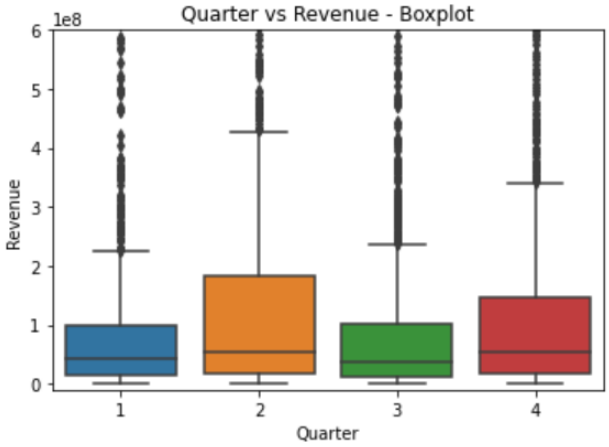
The Correlation coefficient is not great enough to state that there's a Linear Correlation with the target.

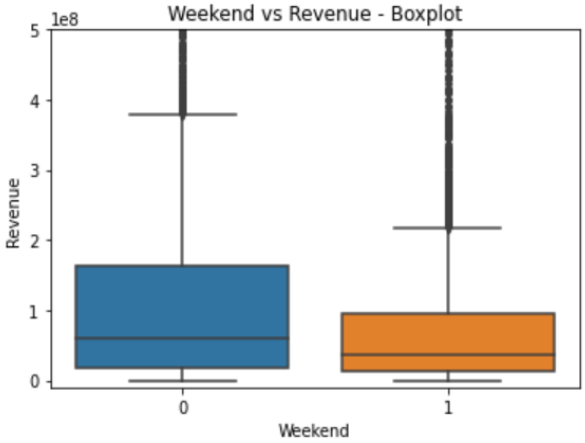
Categorical Variables.

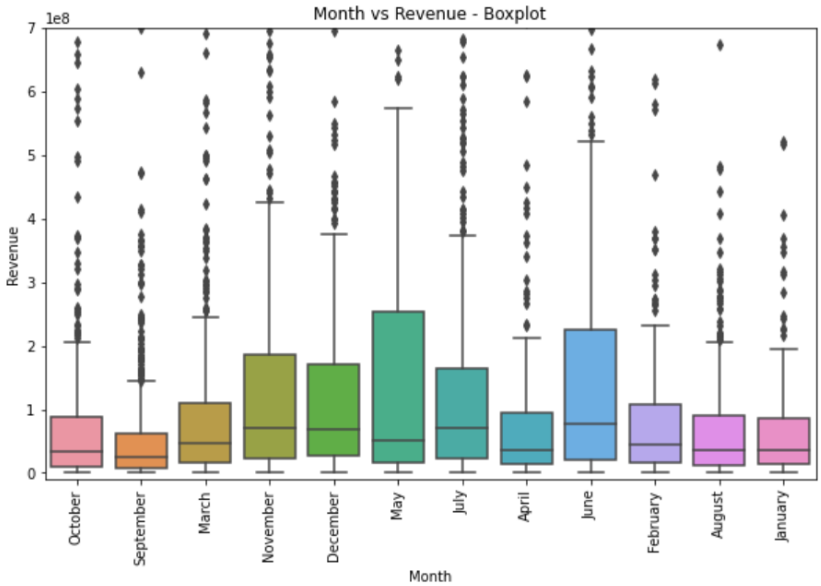
I performed either visual (Boxplots) and statistical analysis (ANOVA – Analysis of Variance) on categorical variables.

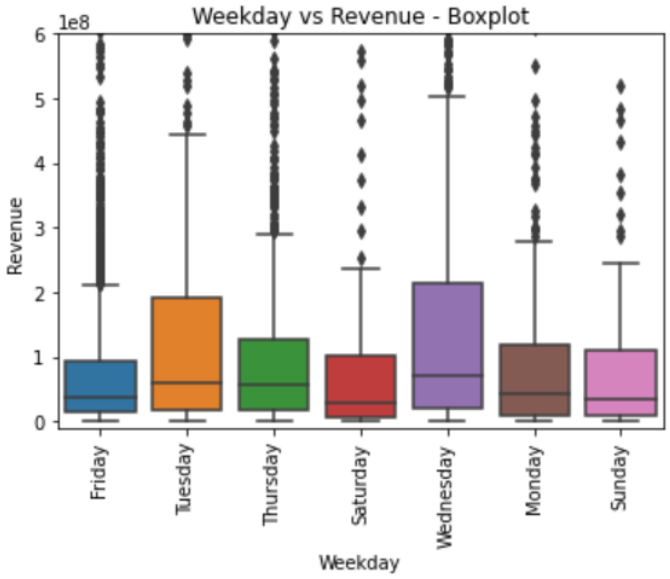













```
F_test scores by ANOVA:
{'Collection': 534.9807788403397,
   'Weekend': 130.61065415125591,
 'English_Language': 59.71092971,
    'Quarter': 36.85812987940133,
   'Weekday': 33.389122394658976,
     'Month': 20.887922223737462}
```

Based on both boxplots and ANOVA results, I drew the below conclusions:

- “Collection” (whether a movie is part of a collection or not) can be a good predictor for the revenue: a movie that is part of a collection has more chances of being a box-office success than a movie which is not.
- “Weekend” seemed to be a suitable predictor as well: movies released not during the weekends have grossed better than movies released over weekends.
- English_Language, Quarter, Month, and Weekday showed some overlapping between categories, and got quite low F-test scores by ANOVA: they may not be good explanatory variables for the target Revenue.

E.D.A. results:

- "Budget * Runtime" attribute was selected as feature to train a Linear Regression Model.
- Budget was not selected as feature to avoid multi-collinearity issues with the variable "Budget * Runtime" (the 2 features were, obviously, highly correlated, therefore they were not really independent variables).
- Runtime was not selected as feature to train the Model.
- "Budget/Runtime" was not selected as feature to train the Model.
- Collection and Weekend had the highest explanatory value, amongst the categorical variables, over the target Revenue, therefore were selected as features to train the Model.
- English_Language, Quarter, Month, and Weekday were not selected as features to train the Model.

3.3. Feature Selection and Variable Transformations.

I've analyzed the columns that were JSON lists of 'values' (Cast, Crew, Production_Companies, Genres, Languages_spoken, Country), selected the most frequent values in each feature, checked for correlations with the target, and, in case the features were relevant, applied one-hot encoding transformation.

Cast.

	Cast_Member	Num_Movies
1	Frank Welker	81
2	Samuel L. Jackson	78
3	Robert De Niro	64
4	Bruce Willis	62
5	Morgan Freeman	57
6	Matt Damon	55
7	Liam Neeson	54
8	Nicolas Cage	53
9	Steve Buscemi	50
10	J.K. Simmons	50
11	Johnny Depp	50
12	Willem Dafoe	47
13	Sylvester Stallone	47
14	John Goodman	46
15	Brad Pitt	45

For this kind of task, I've followed the below steps:

- Set a threshold at 45 movies

- Kept only the actors/actresses who had starred in at least 45 movies (the 15 most frequent cast members)

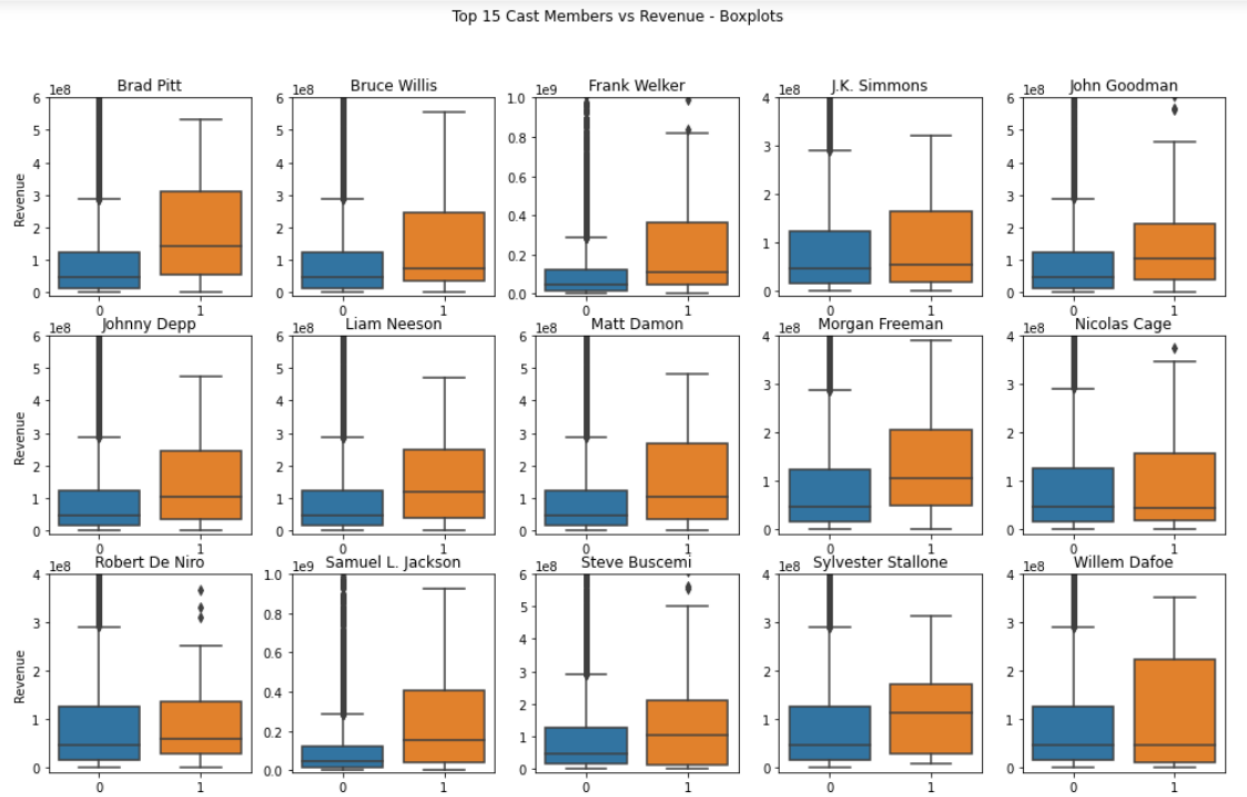
- Applied One-Hot-Encoding transformation for nominal categories.

- Checked for correlations between the top 15 cast members and the revenue (see Boxplot graph below).

- Selecting only the actors who seemed to be capable of making a difference when it comes to the movie's revenue: **Brad Pitt, Bruce Willis, Frank Welker, John Goodman, Johnny Depp, Liam Neeson, Matt Damon, Morgan Freeman, Samuel L. Jackson, Steve Buscemi**.

- Applied Polynomial Transformation on the encoded dataset to create interactions between the cast members. For this purpose, I created a function, poly_transform_dummies*(degree, encoded_dataset*), able the perform the following steps:

- Use PolynomialFeatures by Scikit-Learn, to poly-transform the encoded dataset.
- Assigned the correct name to each attribute.
- Eliminate the columns with only null values (that could be the case if cast members never had starred in a movie together, for instance).
- Remove duplicates (like squared, cubed,… features): obviously, if you squared or powered to a higher index "Brad Pitt", for instance, you'd just get a column which is exactly the same as the original, since the values were only 0 and 1 (besides having a feature like "Brad Pitt^2" wouldn't have made any sense).



After poly-transforming and encoding, I got 3 datasets with the top relevant cast members and their interactions:

- First dataset -> unique features only (10 features).
- Second dataset -> second degree feature interactions: couple of actors (35 features).
- Third dataset -> third degree feature interactions: combination of 3 actors (36 features).

Crew.

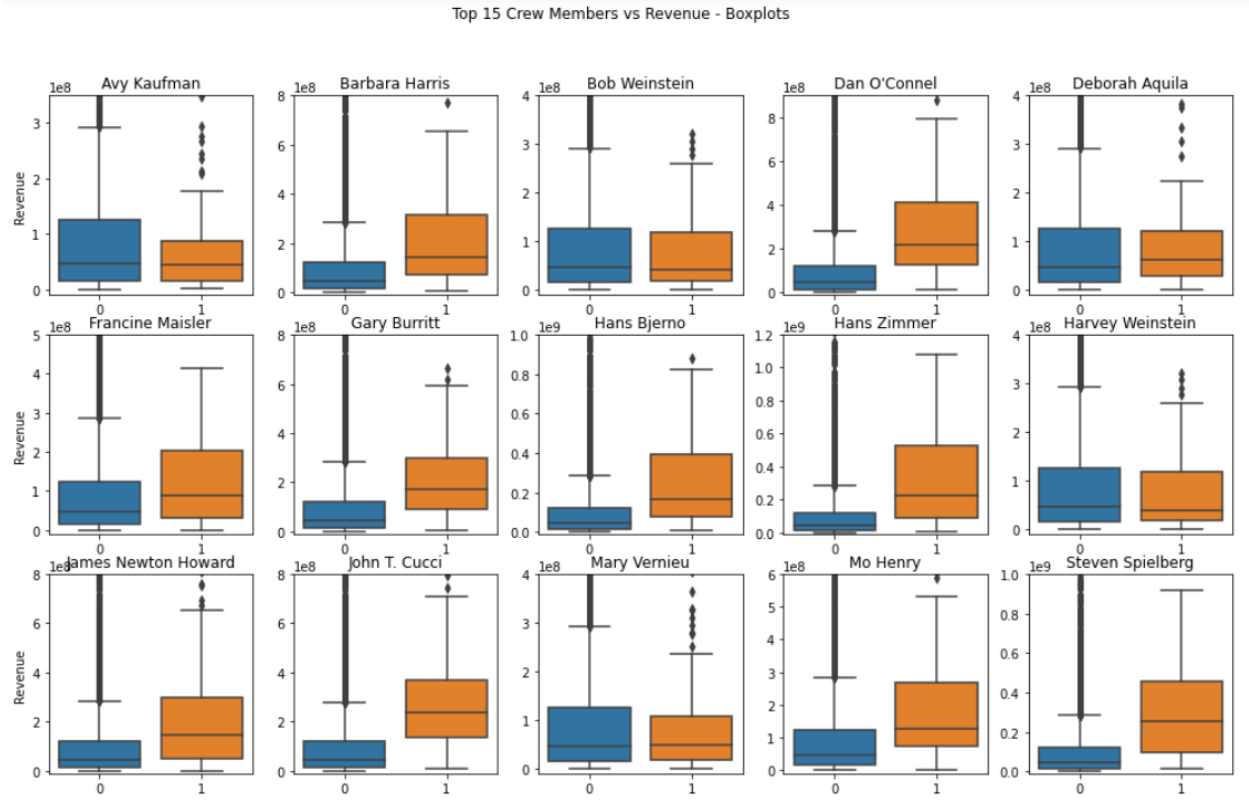
	Crew_Member	Num_Movies
1	Harvey Weinstein	139
2	Bob Weinstein	135
3	Mary Vernieu	116
4	John T. Cucci	105
5	Dan O'Connell	100
6	Mo Henry	96
7	Gary Burritt	95
8	Hans Zimmer	94
9	Barbara Harris	93
10	Avy Kaufman	90
11	Francine Maisler	84
12	Deborah Aquila	83
13	Steven Spielberg	83
14	Hans Bjerno	81
15	James Newton Howard	79

- Set a threshold at 79 movies.

- Kept only the crew members who had worked in at least 79 movies (the 15 most frequent crew members).

- Applied One-Hot-Encoding transformation, and checked for correlations with the revenue (see Boxplots below).

- Selecting only the crew members who seemed to be capable of making a difference when it comes to the movie's revenue: **Barbara Harris, Dan O'Connell, Francine Maisler, Gary Burritt, Hans Bjerno, Hans Zimmer, James Newton Howard, John T. Cucci, Mo Henry, Steven Spielberg.**



- Applied Polynomial Transformation on the encoded dataset to create interactions between the crew members, using the function I created: poly_transform_dummies().

After poly-transforming and encoding, I got 3 datasets with the top relevant crew members, and their interactions:

- First dataset -> unique features only (10 features)
- Second dataset -> second degree feature interactions: combination of 2 crew members (50 features).
- Third dataset -> third degree feature interactions: combination of 3 crew members (74 features).

Genres.

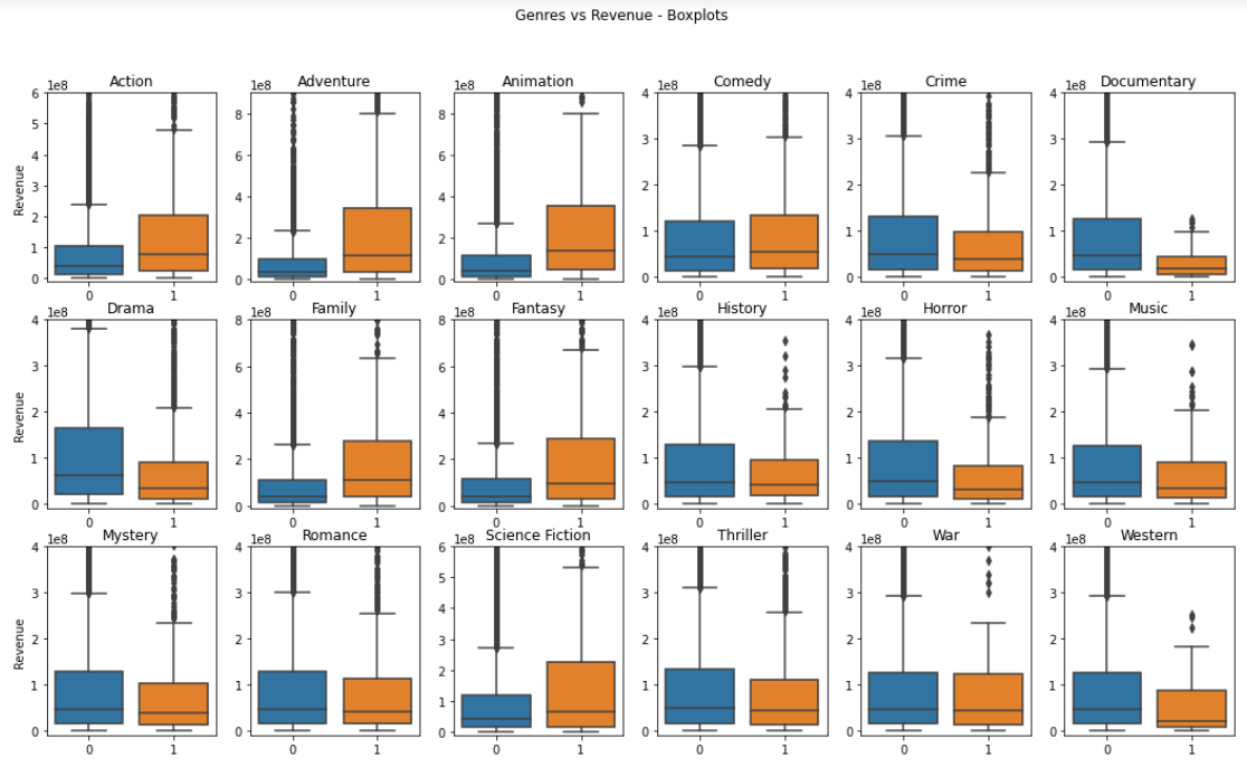
	Genres	Num_Movies
1	Drama	2175
2	Comedy	1686
3	Thriller	1410
4	Action	1320
5	Adventure	957
6	Romance	823
7	Crime	781
8	Science Fiction	641
9	Horror	606
10	Fantasy	556
11	Family	555
12	Mystery	444
13	Animation	310
14	History	231
15	War	181
16	Music	150
17	Western	79
18	Documentary	30
19	TV Movie	2

- Since the number of Genres was only 19, I didn't perform any category-frequency analysis. I just dropped "TV Movie" since it appeared in only 2 movies.

- Performed One-Hot-Encoding transformation, and box plot analysis on all genres, checking for correlations with the revenue.

- Selecting only the genres who seemed to be capable of making a difference when it comes to the movie's revenue: **Action, Adventure, Animation, Documentary, Drama, Family, Fantasy, Horror, Science Fiction, Western.**

- Applied Polynomial Transformation on the encoded dataset to create interactions between the genres, using poly_transform_dummies().



After poly-transforming and encoding, I got 5 datasets with the relevant genres and their possible interactions:

- First dataset -> unique features only (10 features)
- Second dataset -> second degree feature interactions: combination of 2 Genres (47 features).
- Third dataset -> third degree feature interactions: combination of 3 genres (102 features).
- Fourth dataset -> fourth degree feature interactions: combination of 4 genres (136 features).
- Fifth dataset -> fifth degree feature interactions: combination of 5 genres (141 features).

Production_Companies.

	Production_Companies	Num_Movies
1	Warner Bros. Pictures	442
2	Universal Pictures	426
3	Paramount	342
4	Columbia Pictures	327
5	20th Century Fox	316
6	New Line Cinema	181
7	Walt Disney Pictures	164
8	Metro-Goldwyn-Mayer	142
9	Sony Pictures	128
10	Relativity Media	118
11	Touchstone Pictures	114
12	Canal+	111
13	DreamWorks Pictures	105
14	United Artists	97
15	Miramax	92
16	Village Roadshow Pictures	90

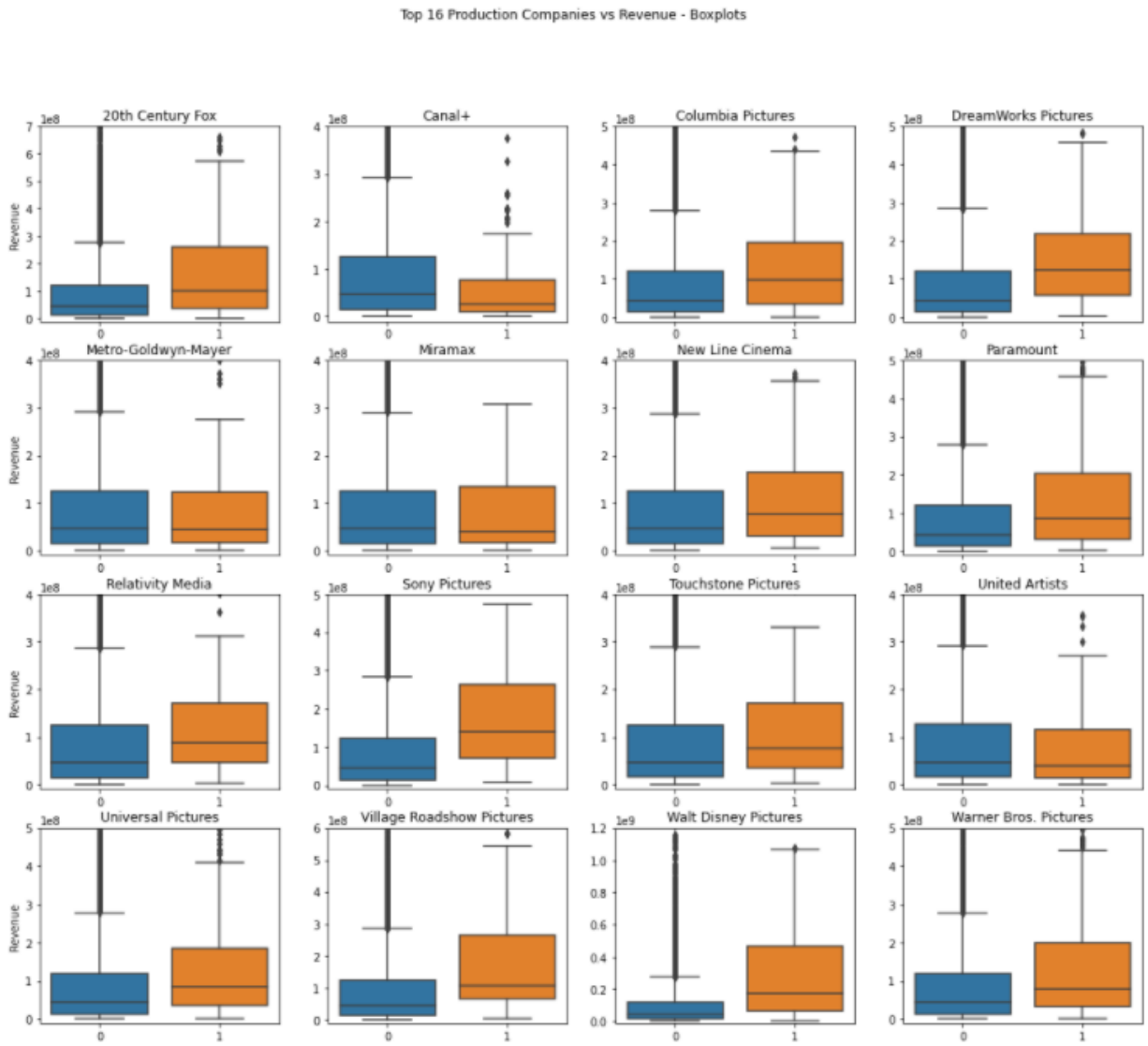
- Set a threshold at 90 movies.

- Kept only the Production Companies who had produced at least 90 movies (the 16 most frequent Production Companies).

- Applied One-Hot-Encoding transformation, and checked for correlations with the revenue (see Boxplots).

- Selecting only the Production Companies who seemed to be capable of making a difference when it comes the movie's revenue: **20th Century Fox, Canal+, Columbia Pictures, DreamWorks Pictures, Paramount, Sony Pictures, Universal Pictures, Village Roadshow Pictures, Walt Disney Pictures, Warner Bros. Pictures.**

Applied Polynomial Transformation on the encoded dataset to create interactions between the Production Companies, using poly_transform_dummies().



After poly-transforming and encoding, I got 3 datasets with the most-frequent relevant Production Companies, and their interactions:

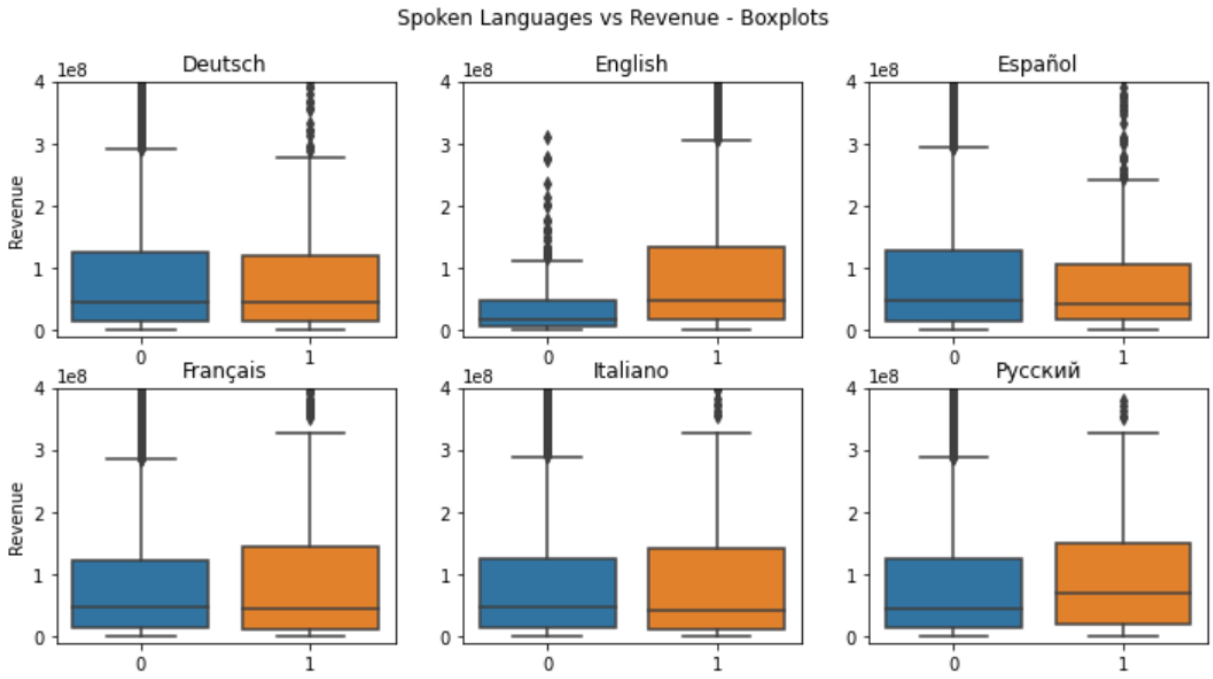
- First dataset -> unique features only (10 features)
- Second dataset -> second degree feature interactions: combination of 2 Production Companies (38 features).
- Third dataset -> third degree feature interactions: combination of 3 Production Companies (41 features).

Languages_Spoken.

	Languages_Spoken	Num_Movies
1	English	4595
2	Français	470
3	Español	465
4	Deutsch	272
5	Italiano	229
6	Русский	208

- Not surprisingly, almost 95% of movies (4,595) had English as one of the Spoken Languages.

- For further category selection, I set a threshold at 200 movies: I kept only the Languages spoken in at least 200 movies, that is the most frequent 6 spoken languages.



English seemed to be the Language with the strongest correlation to the movie's revenue. The other spoken languages seemed to have no, or little, explanatory value over the target.

Country_of_Origin.

	Country	Num_Movies
1	United States of America	4212
2	United Kingdom	678
3	France	372
4	Germany	321
5	Canada	246
6	Australia	110
7	Italy	104
8	Spain	91
9	Japan	86
10	China	77

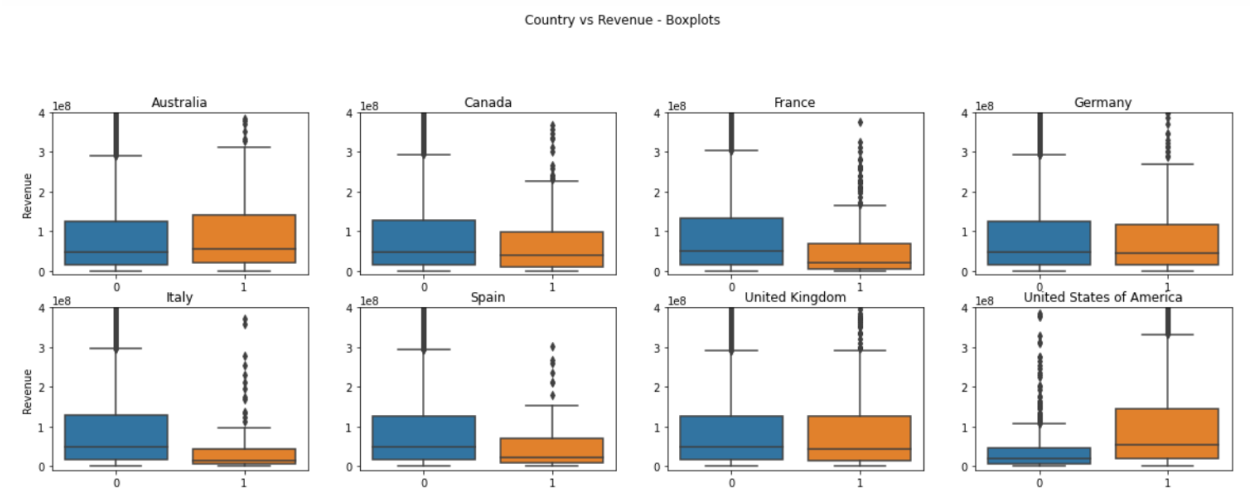
- Again not surprisingly, the vast majority of movies (86%, 4,212) were made in the United States.

- For category selection, I set a threshold at 90 movies: I kept only the countries where at least 90 movies had been made, that is the top 8 countries.

- Applied One-Hot-Encoding transformation, and checked for correlations with the revenue.

- Selected only the countries who seemed to be relevant when it comes the movie's revenue: **France, Italy, Spain, United States of America.**

Applied Polynomial Transformation on the encoded dataset to create interactions between the Countries, using poly_transform_dummies().



After poly-transforming and encoding, I got 4 datasets with the most-frequent relevant Countries, and their interactions:

- First dataset -> unique features only (4 features).
- Second dataset -> second degree feature interactions: combination of 2 countries (10 features).
- Third dataset -> third degree feature interactions: combination of 3 countries (14 features).
- Fourth dataset -> fourth degree feature interactions: combination of 4 countries (15 features).

Eventually, following features have been selected for model development:

- Budget * Runtime
- Collection. Feature has been encoded, and first column dropped to avoid multicollinearity issues.
- Weekend. Feature has been encoded, and first column dropped to avoid multicollinearity issues.
- Cast – 10 encoded features: Brad Pitt, Bruce Willis, Frank Welker, John Goodman, Johnny Depp, Liam Neeson, Matt Damon, Morgan Freeman, Samuel L. Jackson, Steve Buscemi.
- Crew - 10 encoded features: Barbara Harris, Dan O'Connell, Francine Maisler, Gary Burritt, Hans Bjerno, Hans Zimmer, James Newton Howard, John T. Cucci, Mo Henry, Steven Spielberg.
- Production Companies - 10 encoded features: 20th Century Fox, Canal+, Columbia Pictures, DreamWorks Pictures, Paramount, Sony Pictures, Universal Pictures, Village Roadshow Pictures, Walt Disney Pictures, Warner Bros. Pictures.
- Genres – 10 encoded features: Action, Adventure, Animation, Documentary, Drama, Family, Fantasy, Horror, Science Fiction, Western.
- Spoken Languages - 1 encoded feature: English.
- Countries of production - 4 encoded features: France, Italy, Spain, United States of America.

I, then, created the feature-set, the target, and split into train and test sets (test set = 25%).

```
Features - Train Set: (3653, 48)
Target - Train Set: (3653,)
Feature - Test Set: (1218, 48)
Target - Test Set: (1218,)
```

The train set contained 3,653 data points, with 48 features. The test set contained 1,218 observations.

3.4. Model Development.

3.4.1. Linear Regression.

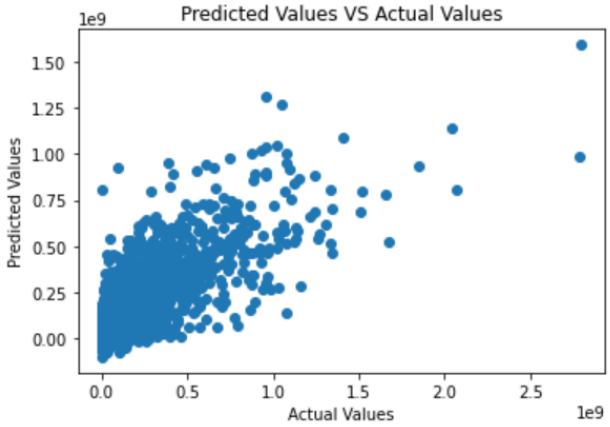
I trained a Linear Regression model on the train set, and used the test set to validate the model and calculate the model error. To asses and compare model's performance, I used 2 error metrics: **Coefficient of Determination (R^2),** and **Mean Squared Error.**

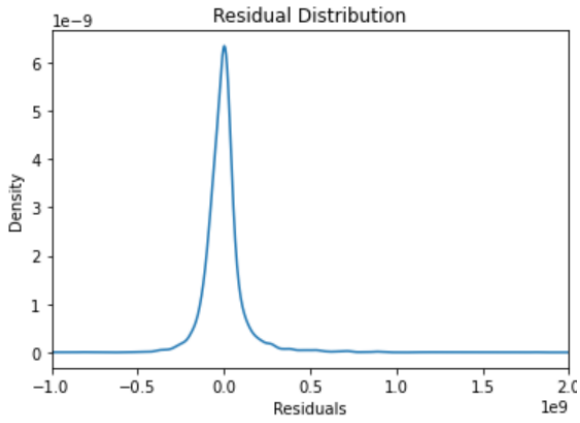
I, then, validated the model performance by cross-validation (shuffle = True, cv=4, seed=6789), and compared the In-Sample error (fitting the model on the whole dataset) with the Out-of-Sample error (cross-validation).

```
Linear Regression - In-Sample Coefficient of Determination: 0.615583804948
```

```
Linear Regression - In-Sample Mean Squared Error: 1.427590048831951e+16

Linear Regression with Cross Validation –
Out-of-Sample Coefficient of Determination: 0.5972917186347961
Linear Regression with Cross Validation –
Out-of-Sample Mean Squared Error: 1.4955205905961304e+16
```

As expected, both error metrics indicated that the model performed better when fitted and tested on the whole dataset, but the difference is actually very small: the model was **under-fitting the data.**

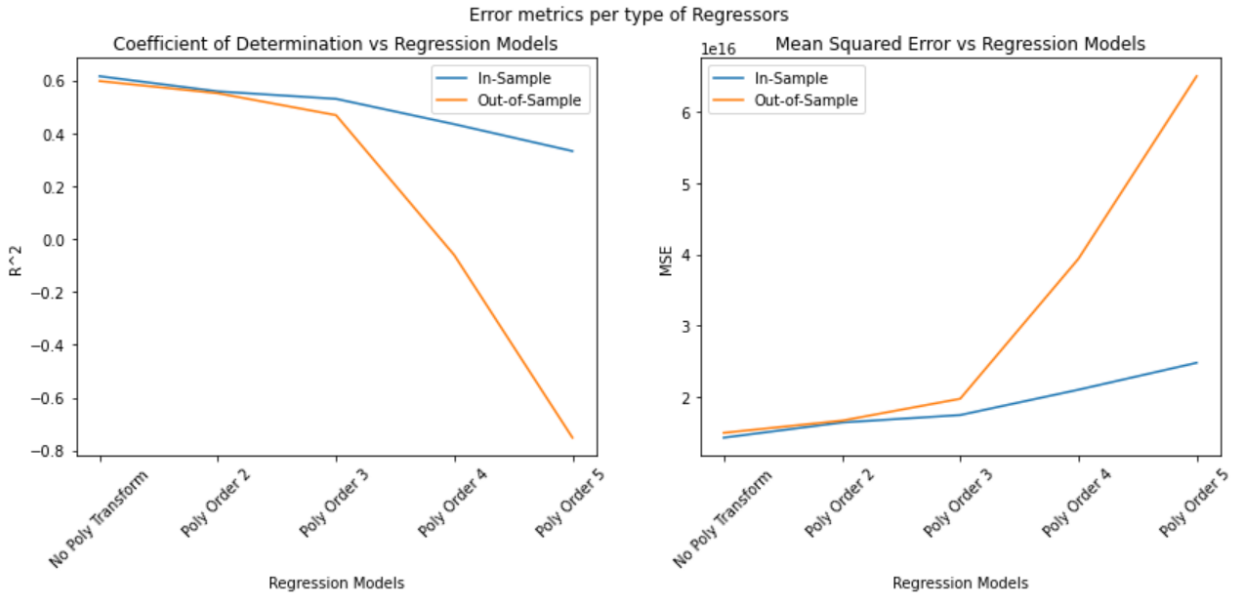




Residuals followed a (normal) bell-curve distribution, although right-skewed, (almost) centered around 0. Furthermore, predictions seemed to have a linear correlation with the actuals. The out-of-sample Coefficient of Determination was around 0.6, that is 60% of the target variation was explained by our model: the linear model worked just fine, but **under-fitting the data.**
I, therefore, tried to increase the complexity of the model, by adding Polynomial effects and checked if I managed to decrease the error.

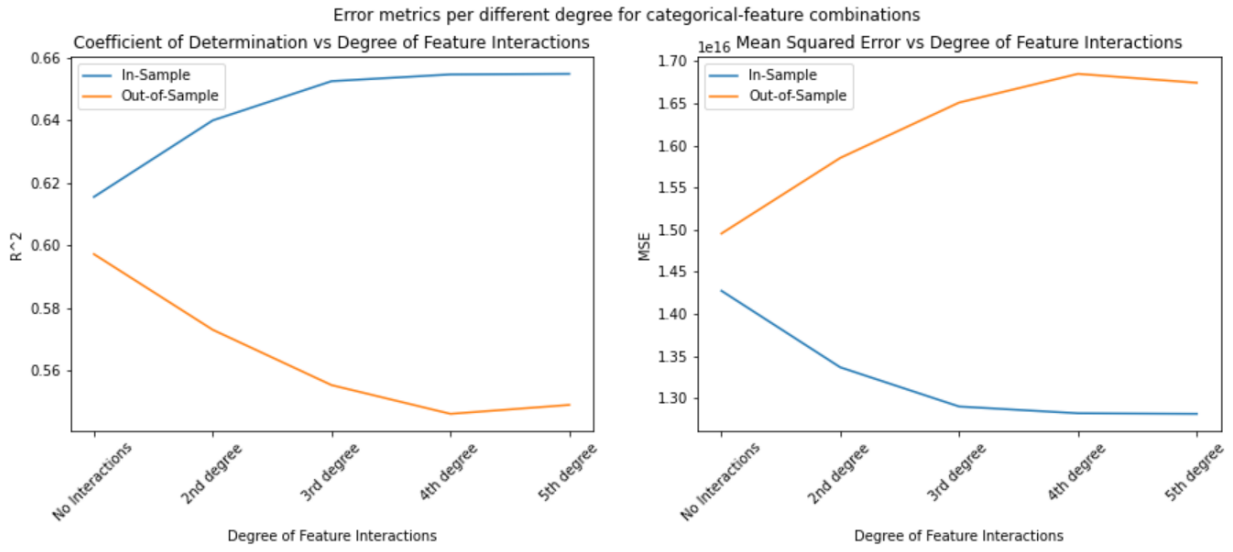
3.4.2. Polynomial Transformation.

I trained the model using different Polynomial Transformation orders (from 2 to 5) on the only numerical feature, and added the categorical feature interactions previously created (combination of cast members, crew…), following the polynomial degree of transformation; I used the same cross validation method as per the Linear Regression model, and compared the error metrics.



Polynomial Transformations didn't fix the under-fitting problem: the in-sample error was higher than before (the higher the degree, the higher the error). Considering that both In-Sample and Out-of-Sample errors increased with higher polynomial degrees, poly-transforming the features just increased the model Variance without decreasing the Bias.

I then tried to add the categorical feature interactions only, without poly-transform the numeric-type features. I kept the same cross validation method.



The in-sample error actually decreased, whilst the out-of-sample error increased. This meant that categorical-feature interactions improved the fitting of the model (in terms of under-fitting), but the model generalized worse: predictions on unseen data were actually less

accurate. Adding categorical feature interactions improved the under-fitting problem but increased the chance of over-fitting the data: I managed to decrease the model Bias, but I have increased the Variance as well. I, therefore, decided to move forward with the highest degree of feature interactions, and use regularization techniques to decrease the Variance to improve model accuracy on unseen data.

Before moving forward with Regularization, I tried to improve model performance by creating 3rd-order feature interactions between Cast and Crew categories (combinations of actors with crew members). Results are as follows:

```
Cast&Crew Interactions (Order 3) –
In-Sample Coefficient of Determination : 0.6795577666568725
Cast&Crew Interactions (Order 3) –
In-sample Mean Squared Error: 1.1900126722913352e+16

Cast&Crew Interactions (Order 3) –
Cross-Validated Coefficient of Determination : 0.5397407078225276
Cast&Crew Interactions (Order 3) –
Cross-Validated Mean Squared Error: 1.7092453279856632e+16

3rd-order interactions between different variables (Cast and Crew)
decreased the in-sample error.
The fit of the model has improved.

3rd-order interactions between different variables (Cast and Crew)
increased the out-of-sample error.
The model is over-fitting the data.
```

Interactions between Cast and Crew categories further decreased the model Bias, and increased the Variance as well. However, the model is better fitting the data, when trained and tested on the same dataset. **The in-sample R^2, without feature interactions was 0.62; with the created feature interactions became 0.68.**

The final dataset had 410 features (1 numeric, and 409 encoded categorical (using nominal one-hot encoding)):

- Budget * Runtime
- Collection
- Weekend
- English Language
- 3rd order interactions of cast members.
- 3rd order interactions of crew members.
- 3rd order interactions of cast and crew members.
- 3rd order interactions of production companies.
- 4th order interactions of countries.

- 5^{th} order interactions of movie genres

I used, then, Regularized Regressions to help the linear model generalize better and avoid over-fitting.

3.4.3. Linear Regression with Regularization.

Regularization algorithms add a "penalty" in the Cost Function based on the learned model parameters; since the parameters, obviously, depend on the selected features, it's important that all features are on the same scale; furthermore, considering the project focus on **interpretation,** features must be on the same value scale to assess their relative importance.

I, therefore, scaled the features using Standard Scaler (reducing the mean to 0, and scaling to unit variance), and compare the results from different Regularization approaches.

To come up with the out-of-sample error, I used the same Cross-Validation method, with GridSearchCV approach (LASSOCV, RidgeCV, ElasticNetCV) to find out to best hyperparameter values.

LASSO (Least Absolute Shrinkage and Selection Operator) Regression - L1 penalty

```
LASSO Regression –
Coefficient of Determination (Cross-Validated): 0.6656741789026847

LASSO Regression –
Mean Squared Error (Cross-Validated): 1.2415715607436606e+16

LASSO Regression - Best hyperparameter (cross validation): 1,000,000.0


Number of coefficients equal to zero: 232
```

Ridge Regression - L2 penalty

```
Ridge Regression –
Coefficient of Determination (Cross-Validated): 0.6783412801071025

Ridge Regression - Mean Squared Error (Cross-Validated):
1.194530286573308e+16

Ridge Regression - Best hyperparameter (cross validation): 100.0

Number of coefficients equal to zero: 0
```

Elastic Net - L1 + L2 penalties

```
Elastic Net –
Coefficient of Determination (Cross-Validated): 0.6720789177360258

Elastic Net - Mean Squared Error (Cross-Validated): 1.2177865549568886e+16

Elastic Net - Best hyperparameter (cross validation): 0.1

Elastic Net - Best L1/L2 ratio (cross validated): 0.2

Number of coefficients equal to zero: 0
```

4. Results.

```
Cross-validated out-of-sample error - Coefficient of determination:
{'Ridge Regression': 0.6783412801071025,
 'Elastic Net': 0.6720789177360258,
 'LASSO Regression': 0.6656741789026847,
 'Linear Regression': 0.5972917186347961,
 'Feature Interactions': 0.5397407078225276}


Cross-validated out-of-sample error - Mean Squared Error:
{'Ridge Regression': 1.194530286573308e+16,
 'Elastic Net': 1.2177865549568886e+16,
 'LASSO Regression': 1.2415715607436606e+16,
 'Linear Regression': 1.4955205905961304e+16,
 'Feature Interactions': 1.7092453279856632e+16}
```

As already observed, adding Feature Interactions, through Polynomial Transformation, decreased the model Bias (lower in-sample error), but increased the model Variance, and consequently, the chance of over-fitting (higher out-of-sample error).

I managed to decrease the model Variance by using regularization techniques, and obtain a smaller out-of-sample error.

Ridge Regression is the Regularization technique with the greatest Coefficient of Determination, and smallest Mean Squared Error, although the difference between different regularizers' performance is not big.

> Ridge Regression - Best Hyperparameters (cross validated):
> Alpha = 100

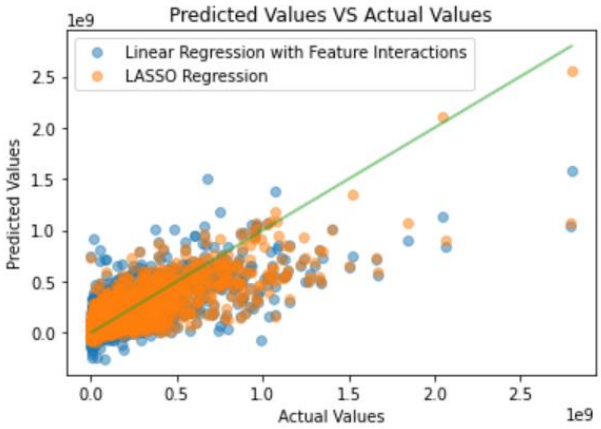
Elastic Net - Best Hyperparameters (cross validated):
Alpha = 0.1
Emphasis of L1 versus L2 penalties: 0.2 (stronger emphasis on L2 over L1)

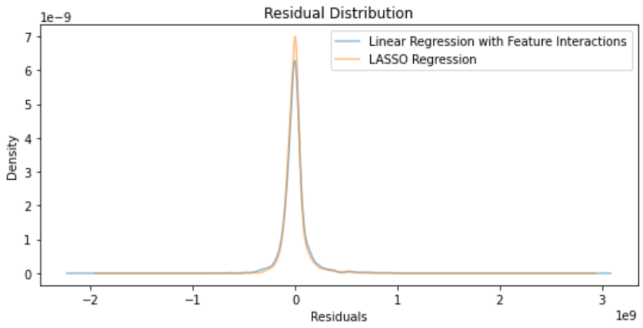
LASSO Regression - Best Hyperparameters (cross validated):
Alpha = 1,000,000

LASSO Regression best hyperparameter was way higher than the other 2: this meant that LASSO model had a way higher bias (and a way lower variance) than the other 2 regularized models. LASSO regression model was, in fact, way less complex than the others: **232 features have been zeroed out using LASSO regression.**

LASSO's error, considering both the Coefficient of Determination and the Mean Squared Error, was the greatest amongst the 3 Regularized Regressions, although the difference was not big: LASSO managed to greatly reduce the complexity of the model, without increasing the model error significantly.

Considering that the main focus of the project was on **model interpretation**, and that the 3 regularization error metrics were not very different, **I selected LASSO Regression model as the model that best suits the project purpose.**





Analyzing the graphs above, we can see how adding Feature Interactions and reducing complexity with regularization improved the model precision: LASSO predicted values were closer to the actuals, and there were more Residuals equal to 0. However, the main benefit of using LASSO regularization, vs Ridge or Elastic Net, was that we have now a way simpler model much easier to interpret.

Below some considerations about the features:

```
List of no significant features:
Index(['20th Century Fox Warner Bros. Pictures', 'Canal+ Columbia Pictures',
```

```
       'Canal+ Walt Disney Pictures', 'Canal+ Warner Bros. Pictures',
       'Columbia Pictures Paramount',
       'Columbia Pictures Village Roadshow Pictures',
       'DreamWorks Pictures Paramount', 'Paramount Sony Pictures',
       'Paramount Universal Pictures', 'Paramount Village Roadshow Pictures',
       ...
       'Morgan Freeman John T. Cucci Mo Henry',
       'Barbara Harris Dan O'Connell Hans Bjerno',
       'Barbara Harris Gary Burritt Hans Bjerno',
       'Barbara Harris Gary Burritt John T. Cucci',
       'Barbara Harris Mo Henry Steven Spielberg',
       'Dan O'Connell Gary Burritt Hans Zimmer',
       'Dan O'Connell Hans Bjerno James Newton Howard',
       'Dan O'Connell James Newton Howard John T. Cucci',
       'Francine Maisler James Newton Howard Mo Henry',
       'Hans Zimmer John T. Cucci Mo Henry'],
      dtype='object', length=232)

Number of positively related features: 78

10 most positively correlated Features:
[Index(['Budget*Runtime'], dtype='object'),
 Index(['Collection'], dtype='object'),
 Index(['Samuel L. Jackson Dan O'Connell'], dtype='object'),
 Index(['Animation Family'], dtype='object'),
 Index(['Steven Spielberg'], dtype='object'),
 Index(['Adventure Fantasy'], dtype='object'),
 Index(['Samuel L. Jackson Hans Bjerno'], dtype='object'),
 Index(['Frank Welker Barbara Harris Hans Zimmer'], dtype='object'),
 Index(['Action Adventure Science Fiction'], dtype='object'),
 Index(['Bruce Willis James Newton Howard Mo Henry'], dtype='object')]


Number of negatively related features: 100

10 most negatively correlated Features:
[Index(['Samuel L. Jackson John T. Cucci'], dtype='object'),
 Index(['Adventure Family Fantasy'], dtype='object'),
 Index(['Action'], dtype='object'),
 Index(['Weekend'], dtype='object'),
 Index(['Johnny Depp Hans Zimmer'], dtype='object'),
 Index(['Samuel L. Jackson Hans Bjerno James Newton Howard'], dtype='object')
 Index(['Science Fiction'], dtype='object'),
 Index(['Gary Burritt John T. Cucci'], dtype='object'),
```

```
 Index(['Barbara Harris Hans Bjerno'], dtype='object'),
 Index(['Francine Maisler Hans Zimmer'], dtype='object')]
```

```
Top 10 most relevant features:
```

	Feature	Type of Correlation
0	Budget*Runtime	Positive
1	Collection	Positive
2	Samuel L. Jackson Dan O'Connell	Positive
3	Animation Family	Positive
4	Steven Spielberg	Positive
5	Samuel L. Jackson John T. Cucci	Negative
6	Adventure Fantasy	Positive
7	Samuel L. Jackson Hans Bjerno	Positive
8	Adventure Family Fantasy	Negative
9	Frank Welker Barbara Harris Hans Zimmer	Positive

5. Discussion.

After adding several Feature Interactions (using Polynomial Transformation) and using LASSO regularization, I managed to increase the model performance, measured on out-of-sample sets with cross validation techniques (to avoid dependency on a particular train/test split), from 0.597 to 0.665 (R^2). **The model is now able to explain 66.5% of the target variable variation.**

Furthermore, I am able now to assess the relative importance of the model features, that is I can draw some considerations on what features are more impactful on movies' revenue, and in what way.

- "**Budget*Runtime**" is the feature with the greatest (positive) impact on the revenue: a long high-budget movie has lots of chances to become a box-office success.
- "**Collection**" is the second most important features, which affects the revenue in a positive way: movies that belong to a franchise (Marvel super-heroes, Star Wars, Harry Potter, Lord of the Rings…) have more chances to be high-grossing films.
- Regarding cast and crew, **Samuel L. Jackson** seems to have a knack for making successful movies, although in combination with some crew members: the combinations **Samuel L. Jackson** – **Dan O'Connel** (a foley artist) and **Samuel L. Jackson** – **Hans Bjerno** (a director of photography) seem to work very well for making a movie a commercial success.
- **Steven Spielberg**, a big name in Hollywood, is definitely a main driver for high-grossing movie.
- The combination **Frank Welker** (actor)**, Barbara Harris** (casting director)**, and Hans Zimmer** (a film score composer) seems to be quite successful as well.

- **Samuel L. Jackson** and **John T. Cucci** (a foley artist), together, are actually negatively correlated with the movie's revenue. This seems a little bit weird to me, but there may be an underlying reason for this.
- When it comes to genres, movie-goers seem to love **Animation/Family** and **Adventure/Fantasy** movies, whilst they seem not to appreciate much **Family/Adventure/Fantasy** films.
- 232 features have been zeroed out by LASSO. Amongst those ones, I found **Brad Pitt, Johnny Depp, and Steve Buscemi** as the actors who seem not to affect the commercial outcome of a movie, either in a positive or negative way.
- Crew members who don't have an impact on movie's revenue, either positive or negative, are: **Barbara Harris** (casting director), **Francine Maisler** (casting director), **James Newton Howard** (film score composer), and **John T. Cucci** (foley artist).
- Genres which don't affect the revenue outcome, either in a positive or negative way, are: **Adventure, Documentary, Drama, Family, Fantasy.** As we already observed, combination of different genres may have a more impactful role on the movies' revenue. Quite interestingly, Documentaries, which I would have expected being negatively correlated with the target, don't actually affect the revenue in a negative way.
- Not surprisingly, movies made in **France, Italy, and Spain** have fewer chances of being high-grossing movies, than movies made in the US.
- No **Production Companies** are in the 10 most impactful features, although they can still be relevant.
- Amongst positively correlated features, I found: **Action/Adventure/Science Fiction**, and the combination of following actor/crew members: **Bruce Willis** (actor), **James Newton Howard** (film score composer), and **Mo Henry** (film editor).
- Amongst negatively correlated features, I found: **Action**, and **Science Fiction** movies, **Weekend** (movies released not from Friday to Sunday have more chances to be commercially successful), and the combination of **Johnny Depp** (actor) and **Hans Zimmer** (film score composer).

6. Conclusion.

In this project I built a Linear Regression model to predict movie's revenue. The main focus was on **interpretation**, therefore my main purpose was to build a predictive model able to give insights on what the main drivers that generate movie's revenue are. I've adopted a machine-learning-driven approach, to uncover relationships between the variables, and find insights on the revenue-generating features.

Quality of data was not great: I found lots of duplicates, missing values, outliers (identified as mistakes), that required some data cleaning actions: from a dataset of **17,101 observations, I kept 4,871 data points only.**

I defined a new function poly_transform_dummies*(degree, encoded_dataset*), as already described, to create feature interactions between categorical features, using Polynomial Features, and deleting the unnecessary, not relevant features. Feature Interactions increased the complexity of the model, decreased the Bias, and helped the model better fit the data.

I, then, used Regularization Regression to decrease the model Variance, and help the model to generalize better, and reduce over-fitting. LASSO Regression was the selected regularization algorithm because of **LASSO capability of greatly reducing the model complexity by zeroing out several coefficients, and allow a better understanding of the model's driver and the revenue-generating features.**

The built model is able to explain 66.5% of the Revenue variation (R^2 = 0.665). Therefore, the model accuracy can be still improved. For this purpose, **additional data is needed to train the model on a larger dataset, and capture the still-not-seen underlying relationships between variables.**

Furthermore, model's performance can improve by increasing the number of features; for attributes like Cast, Crew, and Production Companies I had to select only a limited number of categories, due to the limitation of my laptop's hardware. I selected only the most frequent categories (and then further reduced the categories' number by selecting the most relevant ones, through a boxplot analysis). This approach managed to bring interesting results and insights, but other relevant features (actors/actresses, production companies, directors…) had been probably left out in the process. **It would be very interesting to see the model's results with a larger feature selection, which includes more actors/actresses, directors, and production companies.**

A point to note on the Model's results: LASSO Regression selected, as relevant features, some feature interactions between actors and technical crew members (foley artists, casting directors, score-producers…) who are, most likely, unknown to the public. It sounds odd that this kind of interactions were considered relevant by the LASSO algorithm, but there may be an underlying reason for this: some actors and crew members, for instance, may have a special alchemy, when working together, who is reflected into the movie quality, and gets rewarded at the box-office (or the opposite). However, **for more accurate results, specifications on crew members role would have been helpful, in order to select only the most prominent ones, like directors or producers, and compare the results from the 2 different models.**

Last point to note: the feature classification performed by LASSO model is related to the movie's Revenue, and it doesn't have any kind of relevancy in regard to the movie's artistic value, or the quality of work of the movie professionals mentioned in this analysis.

7. Appendix.

Reference Notebook Link - GitHub:

https://github.com/SebastianoDenegri/predicting_movies_revenue/blob/main/Predicting_Movie_Revenue-updated.ipynb

Reference Notebook Link - NBviewer:

https://nbviewer.jupyter.org/github/SebastianoDenegri/predicting_movies_revenue/blob/main/Predicting_Movie_Revenue-updated.ipynb

Linkedin Article:

https://www.linkedin.com/pulse/business-seventh-art-predicting-movies-revenue-sebastiano-denegri