Project Report

Giacomo Bais 851364

Contents

1 Introduction **2**

2 Dataset Exploration **2**
2.1 Revenue . . . 2
2.2 Belongs to Collection . . . 2
2.3 Budget . . . 2
2.4 Genres . . . 2
2.5 Homepage . . . 3
2.6 IMDB ID . . . 3
2.7 Original language . . . 3
2.8 Original title . . . 3
2.9 Overview . . . 3
2.10 Popularity . . . 3
2.11 Poster path . . . 3
2.12 Production companies . . . 3
2.13 Production countries . . . 4
2.14 Release date . . . 4
2.15 Runtime . . . 4
2.16 Spoken language . . . 4
2.17 Status . . . 4
2.18 Tagline . . . 4
2.19 Title . . . 4
2.20 Keywords . . . 4
2.21 Cast . . . 5
2.22 Crew . . . 5
2.23 Conclusions . . . 5

3 Creating Predictive Models **5**
3.1 Baseline . . . 5
3.2 K-nearest-neighbors . . . 5
3.3 Linear Regression . . . 5
3.4 Decision Tree . . . 6
3.5 Random Forest . . . 6
3.5.1 AdaBoost . . . 6

4 Conclusions **6**

1 Introduction

The following document is intended to provide a more accurate and comprehensive documentation of only the code required for the Data & Web Mining exam. The code should always be accompanied by consulting this document for a complete understanding of the analysis process. The project under examination requires the analysis and construction of predictive models based on a dataset of movies, available at this link on Kaggle. Specifically, the task is to build a model for predicting a film's revenue. The document will be divided into two main sections:

- Dataset exploration and data transformation
- Predictive model creation

In conclusion, the document includes a collection of graphs, which can also be accessed from the notebook through the attachment for convenience.

2 Dataset Exploration

The dataset consists of 3000 movies, each with 22 features. To explore each feature and transform it into a numerical value, the preliminary dataset is divided into train and test sets to produce meaningful graphs that are not biased by the test set itself. The procedures followed for each feature and a brief explanation will be presented below.

2.1 Revenue

Revenue is the feature we want to predict. Figure 1 shows the distribution graph of revenue, which is heavily left-skewed. To correct this issue, I try analyzing the logarithm of revenue, as shown in Figure 2. The skewness of the distribution is greatly improved, and if there is a need for more normally distributed data for comparison with other features, the logarithm of revenue can be used. Additionally, there appear to be several seemingly anomalous revenue values below a certain threshold considered by us.

2.2 Belongs to Collection

This feature indicates whether the movie is part of a connected series. Intuitively, we can model the notion of being part of a series using a binary variable, "hasCollection." Figure 3 shows the graph of "hasCollection" against revenue. There doesn't seem to be a strong correlation between the two variables. However, it can be observed that movies with higher revenue tend to be more frequently part of a series. Intuitively, this feature seemed relevant for studying revenue, so it is added to the dataset.

2.3 Budget

This feature indicates the budget value in dollars for the movie. This feature is intuitively very important in studying revenue. Immediately, it is noticeable that there are many anomalous values for this feature, i.e., values lower than a threshold considered by us. As a strategy to correct these values, the average budget is calculated, and the anomalous value is replaced with this average. Furthermore, through another binary feature added by us, "isBudgetDirty," I take into account movies with anomalous budgets, in case this anomaly plays a role in explaining revenue. The graph of budget against revenue is shown in Figure 4, and a strong, possibly linear, correlation with our target variable is immediately noticeable. Therefore, the budget is added to the dataset.

2.4 Genres

This feature lists the various genres that characterize the movie. To model this feature numerically, I attempted to count the number of genres to which each movie belongs, generating the new feature "genreCount." Figure 5 shows the

graph of "genreCount" against revenue. There appears to be some normality in the distribution, suggesting it may explain some of the variance and be useful for certain predictive models. Therefore, the variable is added to the model.

2.5 Homepage

This feature indicates the link to the movie's homepage. Intuitively, we can model this feature with a binary variable, "hasHomepage," indicating the presence or absence of the link. Figure 6 shows the graph of this new feature against revenue. Like in the case of "hasCollection," the correlation is debatable, but there seems to be a non-random distribution. Thus, the variable is added to the dataset.

2.6 IMDB ID

This feature indicates the IMDb ID code for a given movie on the famous movie information website IMDb. Intuitively, this feature is not useful for explaining a movie's revenue and is immediately removed from the dataset.

2.7 Original language

This feature indicates the original language of the film. Exploring the various languages in the dataset, it is noticed that the vast majority are films in English. For this reason, it is natural to model the feature with a binary variable, isEnglish, which indicates whether the film is in English or not. Figure 7 shows the graph of isEnglish against revenue. There is a strong correlation between the two variables, and this new feature is therefore added to the dataset.

2.8 Original title

This feature indicates the original title of the film. Intuitively, this feature is not useful for explaining the film's revenue and is immediately removed from the dataset.

2.9 Overview

This feature describes a brief summary of the film. Intuitively, this feature is not useful for explaining the film's revenue and is immediately removed from the dataset.

2.10 Popularity

This feature is a numeric index of the film's popularity. Figure 8 shows the graph of popularity against revenue, and it is indeed observed that less popular films have very low revenue. However, the behavior of highly popular films, which have variable revenue, is ambiguous. Figure 9 shows the same graph using the logarithm of popularity, which does not seem to adjust its behavior. Popularity is added to the dataset.

2.11 Poster path

This feature provides the link to the film's poster. Intuitively, this feature is not useful for explaining the film's revenue and is immediately removed from the dataset.

2.12 Production companies

This feature indicates the companies involved in the production of the film. Since there are many companies, I try to model the feature with a binary variable that indicates the presence or absence of a company from the top 20 most frequent in the dataset. The variable companiesCount, which counts the number of companies involved for each film, is also generated. Figure 10 shows the graph of this variable against revenue. There is a certain normality

with a left skew, and the feature is therefore added to the dataset. The code also includes graphs for each company in the top 20 against revenue. There is not always a clear correlation, but the feature that distinguishes the presence or absence of an "important" company is added to the dataset.

2.13 Production countries

This feature indicates the country of production of the film. With a quick exploration, it is noticed that this feature is already modeled by the isEnglish variable and is therefore removed from the dataset.

2.14 Release date

This feature indicates the day, month, and year of the film's release. After a quick separation of the three pieces of information, the graphs against revenue are shown in Figures 11, 12, and 13. The year variable may be relevant when using decision trees and is therefore added. The other two variables do not seem useful, but for now, they are still added.

2.15 Runtime

This feature indicates the duration of a film in minutes. The graph shown in Figure 14 reveals a normal distribution, and it is therefore added to the dataset. Additionally, since there are only two missing values, the correct values for these films have been manually added through a quick internet search.

2.16 Spoken language

This feature indicates the languages spoken in the film. Intuitively, this feature is not useful for explaining the film's revenue and is immediately removed from the dataset.

2.17 Status

This feature indicates the release status of the film. After a quick exploration, it seems that all films except two are in the released state. The feature is therefore removed from the dataset as it is not useful.

2.18 Tagline

This feature indicates the slogan of a film. We can model the presence or absence of a slogan using a binary variable, hasTagline. Figure 15 shows the graph of this feature with revenue, and there seems to be a correlation. It is therefore added to the dataset.

2.19 Title

This feature indicates the title of the film. Intuitively, this feature is not useful for explaining the film's revenue and is immediately removed from the dataset.

2.20 Keywords

This feature indicates a series of keywords that characterize the film. Similar to the production companies feature, a ranking of the 50 most frequent keywords is created, and a binary variable is generated to indicate the presence or absence of one of these keywords for a certain film. Figure 16 shows the graph with revenue, which does not seem to have a strong correlation. The feature is therefore removed from the dataset. The count of keywords for each film is also attempted, and the graph is shown in Figure 17. There seems to be some normality or correlation, so the feature is added to the dataset.

2.21 Cast

This feature indicates the actors in the film. A similar approach to the previous feature is used, creating a ranking of the top 100 most frequent actors. The number of actors present in each film is also counted. Figure 18 shows the graph for this latter feature, where a correlation is observed. It is therefore added to the dataset. The code includes graphs for the presence or absence of the "top" actors in the dataset, but their individual contribution seems ambiguous and is ignored.

2.22 Crew

This feature indicates the personnel who worked on the film. To model this feature, the number of people involved in the film is counted, and the graph is shown in Figure 20. There doesn't seem to be much correlation, but the variable is added for now.

2.23 Conclusions

After exploring the dataset, I obtained 38 features that I will use to create predictive models, as described in the next section.

3 Creating Predictive Models

In this section, various algorithms will be used to create predictive models for film revenue using the 38 features obtained in the previous section. First, a baseline will be presented, which is a very simple model that will serve as a starting point. Then, an exploration of algorithms will be carried out, and the chosen algorithms will be presented. The critical choices that led to these algorithms will also be discussed.

3.1 Baseline

As a baseline, I chose a linear regressor that uses the two features that seem most relevant in explaining revenue: budget and popularity. Figure 21 shows the graph of the linear regressor, which appears to moderately explain the variance. The R2 score of this model is approximately 0.548, which will be the starting point for the subsequent algorithms and the threshold for a successful model.

3.2 K-nearest-neighbors

The K-nearest-neighbors algorithm tries to explain revenue by finding similarities among the k nearest points to each point in the dataset. In our case, I initially tried the algorithm with an arbitrary k value (10), and then performed validation and cross-validation to tune the parameter. The use of scaling to rescale the data seemed detrimental, so it was not used for validation or cross-validation. Figures 22, 23, and 24 show the performance of the models in the various stages of parameter tuning. The R2 score reached through cross-validation reaches a maximum of 0.567, already surpassing our baseline.

3.3 Linear Regression

A perhaps more natural choice for the next model is linear regression. We had already observed a certain level of precision with the baseline using only two features, so I tried to create a model that uses all the features. Figure 25 shows the impact of each feature in the model, and once again it is noticeable that the first two features, budget and popularity, are predominant in explaining the revenue. The R2 score of the model is 0.644, which is a significant improvement compared to the baseline.

3.4 Decision Tree

The choice of a decision tree aims to change the approach to predictive modeling. Decision trees use features in a very different way compared to the previously used models and can capture nuances that other models might miss. Additionally, decision trees can easily analyze the importance of features, which is useful for understanding the contribution of each variable. After arbitrarily setting a maximum number of leaves to 10, I performed validation and cross-validation for parameter tuning, as shown in Figures 26 and 28. Figures 27 and 29 display the importance of features, highlighting a strong imbalance. Trying to remove the least useful features does not significantly improve the model, as shown. This is primarily due to the poor use of model parameters. The maximum R2 score achieved after cross-validation is 0.524, comparable to the baseline.

3.5 Random Forest

Using Random Forest can be seen as a simple improvement over using a decision tree. This algorithm involves using multiple decision trees, and the revenue is predicted by averaging the predictions of each tree. Even with an arbitrary number of trees (50), the precision increases significantly, as shown in the histogram of Figure 30. Further efforts to improve the forest's performance through validation, cross-validation, and parameter tuning show that the histograms reduce variance and slightly improve, as demonstrated in Figures 32 and 35. Figures 31 and 34 illustrate the parameter tuning process in the validation and cross-validation phases, while Figures 33 and 36 show the importance of features in these two phases. Features seem to be better utilized in this model, improving its R2 score, which reaches an average value of 0.7, making it the best model so far. Exploring instances with less accurate predictions, as shown in Figures 37 and 38, it is noticed that these instances, on average, are distributed towards the left of the revenue values in the dataset. Therefore, I attempt to use AdaBoost to improve the precision in these "difficult" instances.

3.5.1 AdaBoost

The use of AdaBoost seems to worsen the variance that had stabilized through cross-validation. This outcome is perplexing, and I cannot explain the reason for it. The model's precision slightly deteriorates, as shown in Figure 39, and the "difficult" instances do not change significantly, as shown in Figures 40 and 41.

4 Conclusions

As expected, the best model is the Random Forest. It effectively utilizes the available features and achieves a good level of precision compared to the baseline. Clearly, the feature selection can be improved, and there is likely potential to achieve higher levels of precision with a more accurate choice. Nevertheless, the results obtained using the algorithms are satisfactory.