Mining the Success for Movies

Student Project Data Mining HWS17
Team 6

Presented by

Steffen Jung
Adrian Kochsiek
Martin Koller
Marvin Messenzehl
Daniel Szymkowiak

Submitted to the
Data and Web Science Group
Prof. Dr. Heiko Paulheim
University of Mannheim

October - December 2017

Contents

Chapter 1

Application Area and Goals

This paper represents a documentation for the data mining project *"Mining the Success for Movies"*[1]. The structure of this paper follows the classical data mining process. Chapter 1 provides an overview of the problem the project is based on and is complemented by the goals and objectives of this project. Afterwards, chapter 2 deals with the structure and size of the data. Here, a closer look will be taken at the dataset at hand. Questions that had to be answered were for example which information were provided in the original dataset or which problems were identified concerning for instance outliers or missing values. Upon that, chapter 3 explains which preprocessing steps had to be taken in order to cleanse the dataset to prepare it for the data mining step and model learning. Chapter 4 describes which data mining techniques regarding algorithms and parameters were used to learn an expedient model in respect of the goals set in Chapter 1. Finally, chapter 5 closes this paper by describing which insights could be won for the problem at hand. Here, a critical reflection is delineated how the model could be improved further in order to provide even more precise results.

1.1 Problem Statement

Already well before new movies are being produced, every stakeholder certainly is interested in the monetary success of the given movie. In order to predict the success costly methods are being applied, such as market investigations or deep analysis and .

The benefit which Data Mining brings to the analysis of large datasets, can also be transferred onto the stated problem of predicting a movie's success. Based on

[1] Information in this paper refers to the (sample) dataset and python scripts handed in for a classification problem

given data of already released movies and successful movies in the past, a model is being learned which shall be applied on upcoming or planned movies. In order to learn and apply the model various pieces of information are taken into account. Just a few are budget, revenues, runtime, genre and information on the release. Information on the dataset and on all preprocessing methods which were applied will be provided in chapter 2.

1.2 Goals

The goal of this project is to learn a model which will predict how successful a not yet released movie will be. This is done by using common data mining techniques in the Python programming language. As a main objective the question *"Based on revenue, will the movie be popular or will it be a flop?"* shall be answered for all possible combinations of information on a new movie as precisely as possible. In order to be as precisely as possible, not only different algorithms are being tested, but also parameter tuning is being applied with different performance measures [2].

[2] Further information on applied techniques and evaluation methods is provided in chapter 4

Chapter 2

Data Selection

The selected dataset onto which a classification model shall be learned is provided by Kaggle [1]. It is named *The movies Dataset*[2] and contains metadata on approximately 45,000 movies in its raw format. It is provided and updated by Rounik Banik. The complete dataset consits out of several files in the *.csv* format containing specific info about movie casts, and external score. For the outcome of this project the central file, used for further preprocessing is named *movies-metadata.csv*. This csv-file holds 24 columns in total, which can be expected in the graphic below.

```
['adult', 'belongs_to_collection', 'budget', 'genres', 'homepage', 'id',
'imdb_id', 'original_language', 'original_title', 'overview', 'popularity',
'poster_path', 'production_companies', 'production_countries',
'release_date', 'revenue', 'runtime', 'spoken_languages', 'status',
'tagline', 'title', 'video', 'vote_average', 'vote_count']
```

Untertitel bla bla

2.1 Structure and size of data

2.2 Basic data exploration

- In slides named: ”structure and size of data”
- min. 1 Page
- Selection:

[1] 2017 Kaggle Inc

[2] Link to the dataset: https://www.kaggle.com/rounakbanik/the-movies-dataset

  - What data is available?
  - What do I know about the provenance of the data?
  - What do I know about the quality of the data?
  - Wrong values, lot of null values

- Exploration
  - Get an initial understanding of the data
  - Calculate basic summarization statistics
  - Visualize the data
  - Identify data problems such as outliers, missing values, duplicate records
  - problem with number scales, data formats, truth of contents

Main issue quality of data. Further explained in preprocessing...

Chapter 3

Preprocessing and Transformation

In order to pick out columns that have a significant impact on forecasting a movie's success, the following assumptions concerning the importance of information in the metadata were made:

- The budget and revenue are crucial numbers.
- The release year has an impact on numbers such as budget and revenue.
- The genre, the production company and the runtime are important information.
- Transform data into a representation that is suitable for the chosen data mining methods
  - number of dimensions
  - scales of attributes (nominal, ordinal, numeric)
  - amount of data (determines hardware requirements)
- Methods
  - Aggregation, sampling
  - Dimensionality reduction / feature subset selection
  - Attribute transformation / text to term vector
  - Discretization and binarization
- Good data preparation is key to producing valid and reliable models

- Data preparation estimated to take 70-80% of the time and effort of a data mining project!

3.1 Preprocessing steps according to Python script

3.2 A list of problems we encountered

1. **list further problems we had and solved!**
2. Prod. Comp.: Same prod. company named differently -¿ using Regex to solve (Steffen)
3. dataset: 5 datasets have duplicates

Chapter 4

Data Mining

- Input: Preprocessed Data
- Output: Model / Patterns

1. Apply data mining method
2. Evaluate resulting model / patterns (using P, R, F1, not accuracy)
3. Iterate:
    - Experiment with different parameter settings
    - Experiment with different alternative methods Improve preprocessing and feature generation Combine different methods

4.1 Algorithms

To predict the success of a movie different Algorithms were used:

- K-Nearest Neighbor
- Naive Bayes
- Support Vector Classifier
- Neural Net
- **Decision Tree**
- **Random Forest**

Algorithm	F1 Macro	F1 Micro
Decision Tree	36.2%	38.6%
Random Forest	40.4%	43.4%
Support Vector Classifier	37.5%	39.1%

Table 4.1: Multi Label Classifier Results

Algorithm	F1 Macro	F1 Micro	Downsampled Macro	Downsampled Micro
Decision Tree	56.5%	75.5%	61.9%	63.0%
Random Forest	58.7%	76.0%	80.6%	81.5%
Support Vector Classifier	56.5%	75.0%	60.7%	60.8%

Table 4.2: Binary Classifier Results

The following analysis concentrates on the three algorithms with the best results: Random Forest, Decision Tree and Support Vector Classifier. The first goal of the analysis was to predict the success in five different classes. Since an analysis in that detail with the given data set is very unprecise as seen in table 4.1, an binary classifier was created. To find the best parameter setting , a gridsearch in combination with a ten-fold cross-validation, scoring the highest F1-score, was applied for each of the classifier. The achieved F1-scores of the three binary classifier are listed in table 4.2. At first the classifier were scoring the micro F1-score. Even though the results look promising at the beginning, the classifier were mostly guessing the majority class in this setting. For that reason every classifier was also scored on the macro F1-score, which led to a worse score.

4.1.1 Decision Tree

Regarding the binary classifier, predicting if a movie is going to be a success, the data set is unbalanced with a proximate ratio of 75% ”successful” and 25% ”unsuccessful”. Most of the classifier give bad results with unbalanced data sets. But especially tree-structures can handle unbalanced data sets well, since the hierarchical structure allows them to learn signals from both classes. But even with a hierarchical structure the algorithm was still just predicting the majority class,

while scoring an micro F1 score, as seen in table 4.2. Since the macro F1 score does not that promising, downsampling was used in the next step to improve the prediction. Tuples of the majority class were removed to have a balance of 50/50. Since the downsampled data set is probably too small to learn a good classifier, the next step to improve the algorithm is to upsample the data set. WRITE REST FOR UPSAMPLING HERE

4.1.2 Random Forest

Random Forest builds like the decision tree a hierarchical structure, which can handle unbalanced data sets better than other algorithms. In addition it corrects the overfitting habit of a decision tree. In the gridsearch hyperparameters like the split-criterion, number of features, the minimum sample to split and whether bootstrap samples are used or not are evaluated. With an downsampled data set the algorithm scored an F1 score micro of 80.6%, which is an improvement of more than 18% compared to the decision tree. SOMETHING ABOUT UPSAMPLING HERE:

4.1.3 Support Vector Classifier

4.2 Three best performing algorithms

- Pick best three algos
- GridSearch
- Why does each classifier perform how it performs (unausgeglichene Klassen, ...)?

Chapter 5

Interpretation / Evaluation

Maybe also prospect?

- Output of Data Mining
  - Patterns
  - Models
- In the end, we want to derive value from that, e.g.,
  - gain knowledge
  - make better decisions
  - increase revenue