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

Do we write a Management summary?
This paper represents a documentation for the data mining project ”Mining the Success for Movies”. Information in this paper refers to the (sample) dataset and python scripts handed in for a classification problem. The structure of this paper is as follows: Chapter 1 provides an overview of the problem the project is based on. Goals of the project and the theoretical framework will be discussed as well. 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. Chapter 3 explains which preprocessing steps had to be taken in order to cleanse the dataset and to prepare it for the data mining step. 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. 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. *Do we need a source? koennte noch ausgefuehrt werden.*

The benefit 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 given data of already released movies, 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.

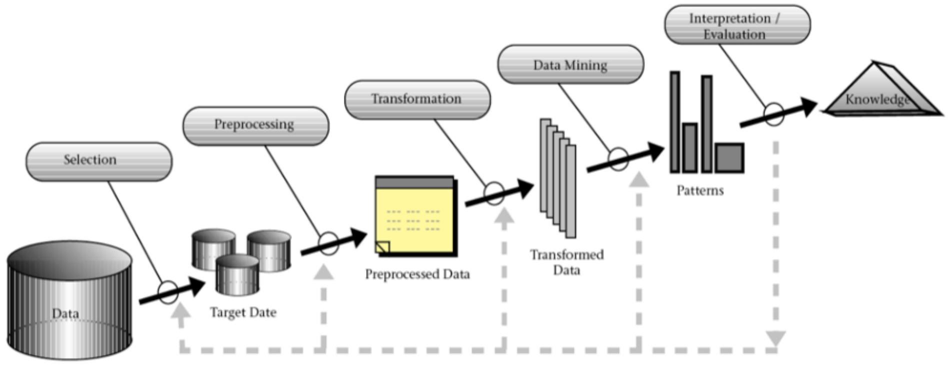
1.2 Goals

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 Python. 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 [1].

- Problem Statement and idea behind the project
- General introduction similar to Project outline

1.3 Theoretical framework

- keep it small
- roughly 1 Page

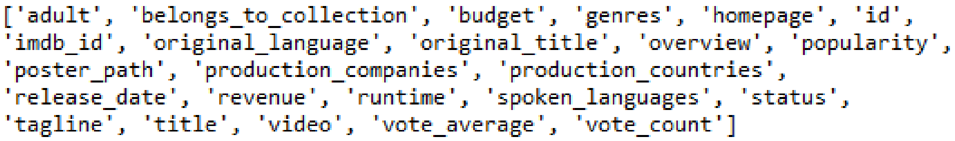


[1] Further information on applied techniques and evaluation methods is provided in chapter 4

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 csv-file has 24 columns.



```
['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

unvollstaendige daten erlaeutern, grundlegende Graphen zeichnen lassen

- In slides named: ”structure and size of data”
- min. 1 Page
- Selection:
  - What data is available?
  - What do I know about the provenance of the data?
  - What do I know about the quality of the data?
- Exploration
  - Get an initial understanding of the data

[1] 2017 Kaggle Inc

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

  - Calculate basic summarization statistics
  - Visualize the data
  - Identify data problems such as outliers, missing values, duplicate records

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**

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 dataset is very unprecise, an binary classifier was created. In the following results for the multiclass prediction as well as for the binary prediction will be presented. 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.

4.1.1 Decision Tree

SOMETHING ABOUT MULTICLASS HERE. Regarding the binary classifier, predicting if a movie is going to be a success, the dataset is unbalanced with a proximate ratio of 75% ”succesfull” and 25% ”unsuccessfull”. Most of the classifier give bad results with unbalanced datasets. But especially tree-structures can handle unbalanced datasets well, since the hierarchical structure allows them to learn signals from both classes. The parameters for the Gridsearch tested the split criterion Entropy and Gini, the max-depth of the tree and the minimum of samples to split. With this setting the decision tree achieves an F1 score of 56% (macro) and 75.5% (micro). Since scoring the F1-micro only predicts successfull, downsampling was used, to improve the prediction. Tuples of the majority class were removed to have a balance of 50/50. With downsampling the tree scores 61.9% (macro) and 63.0% (micro). The downsampling improved the macro score, so the tree does not only predict ”successfull”. Since the downsampled dataset is probably too small to learn a good classifier, the next step to improve the algorithm is to upsample the dataset. WRITE REST FOR UPSAMPLING HERE

4.1.2 Random Forest

Random Forest builds like the decsion tree a hierarchical structure, which can handle unbalanced datasets 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 wheather bootstrap samples are used or not are evaluated. With the best parametersetting the algorithm scores in the ten-fold cross-validation 58.7% (macro) and 76.0% (micro). With a downsampled dataset the algroithm scores 80.6% (macro) and 81.5% (micro). 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