

# The Elephant Man in The Room: An analysis of good and bad movies

[Project Assignment B in 02806: Social Data Analysis and Visualizations] \*

Søren Howe Gersager  
s094557  
syrelyre@gmail.com

## ABSTRACT

This report and the associated website[1] is the result of the final project in DTU Course 02806: Social Data Analysis and Visualizations. The course subject was both gathering data and analyzing data using visualizations, machine learning, sentiment analysis and network theory.

This final project investigates potential differences in movies with high IMDB ratings and low IMDB ratings.

## 1. MOTIVATION

Our dataset is created from the top and bottom 1000 movies sorted by IMDB rating[2][3]. The motivation for the analysis stems from a personal interest in movies, and being curious on what makes a movie good or bad and perhaps eliminating prejudice especially regarding bad movies and their characteristics. An example is the comedy genre, it's our belief that people often have the notion that a lot of bad comedies and romantic comedies exist. By doing the analysis we can investigate whether that hypothesis holds true or not. It is our goal to either confirm or reject the end-users belief in what makes a movie good or bad.

## 2. IMPLEMENTATION

Outlined below is details specific to implementations and practical tools used.

### 2.1 Website

For creating the website we simply used a IPython Notebook for doing the calculations and visualizations, because we liked its simplicity and features. We realize the website is meant for end-users and they might not understand the code being written, however we supply the code with descriptions and text throughout to supply the visualizations and analyzes.

---

\*A website showcasing the project is available at <https://goo.gl/IIlRT8>

## 2.2 Webscraping

For creating the dataset used in the analysis, we used BeautifulSoup[4] to scrape the top 1000 movies and bottom 1000 movie pages of IMDB[6]. Afterwards we used the ID's of the movies extracted from IMDB to get more data about each movie from The Open Movie Database API[5].

The data exposed through the API is: title, release date, MPAA-rating, runtime, genre, director, writers, names of first-billed actors, language spoken, country of origin, short description of plot, awards received, poster image url, Metascore, IMDB rating and number of imdb votes.

## 3. THEORY

We have used several of the tools learned in the course for the website and report, below is a list of tools we have used.

### 3.1 Visualization

We have used matplotlib and basemap for visualizing differences between good and bad movies.

### 3.2 Machine learning

We have used machine learning with cross-validation to try to classify good movies from bad movies. We tried this with several datasets: One dataset with basic features of all 1389 movies, one with a bag of words representations of 100 movie subtitles and one with the ANEW scores of the bag of words representations. We used several models: Decision Tree, Multinomial Naive Bayes and Logistic Regression. We used K-fold cross-validation with  $k = 10$  to validate the models.

### 3.3 Natural Language Processing

We have tried to use sentiment analysis to compare the two groups. We did this by calculating the ANEW scores[8] for the bag-of-words representations of subtitles for a random selection of 50 good and 50 bad movies. From this we tried to visualize the average scores to find differences, however we did not find any significant differences between the two groups.

We also tried to use the ANEW dataset for prediction, however cross-validation yielded a test-error of 50% so this was not better than simply predicting all samples to be the most commonly found class.

### 3.4 Python and relevant libraries

For performing the analysis we have relied on Python, its standard library and the external libraries: matplotlib, scikit-learn, requests, beautifulsoup, basemap.

## 4. DISCUSSION

In this section we reflect on and discuss the analysis we have done and what could be done in future work.

Initially we wanted to use the Rotten Tomatoes API[7] for retrieving the movie data. This was to not be burdened by just a maximum of 2000 movies from the IMDB top and bottom pages, and would also allow us to select more criteria for what makes a movie good or bad, like smaller or bigger vote-count or ratings. However after requesting an API key we got a reply back several days later that we were not granted access because they only offered access to domestic students. Future work could be spent on creating a bigger dataset of movies using either the Rotten Tomatoes API or scraping IMDB.

Furthermore if we had more time we would like to scrape subtitles for every movie and perhaps combine them with our feature dataset to get a larger sample size and more successfully predict good from bad.

## 5. REFERENCES

- [1] <https://goo.gl/III1RT8> *Website showcase*
- [2] [http://www.imdb.com/search/title?=&groups=top\\_1000&sort=user\\_rating,desc&view=simple](http://www.imdb.com/search/title?=&groups=top_1000&sort=user_rating,desc&view=simple) *IMDB Top 1000 movies*
- [3] [http://www.imdb.com/search/title?=&groups=bottom\\_1000&sort=user\\_rating,asc&start=301&view=simple](http://www.imdb.com/search/title?=&groups=bottom_1000&sort=user_rating,asc&start=301&view=simple) *IMDB Bottom 1000 movies*
- [4] <http://www.crummy.com/software/BeautifulSoup/BeautifulSoup>
- [5] <http://www.omdbapi.com/> *The Open Movie Database API*
- [6] <http://www.imdb.com/> *The Internet Movie Database*
- [7] <http://developer.rottentomatoes.com/docs> *Rotten Tomatoes API*
- [8] [http://crr.ugent.be/papers/Ratings\\_Warriner\\_et\\_al.csv](http://crr.ugent.be/papers/Ratings_Warriner_et_al.csv) *ANEW data*