Natural language processing with screenplays

Is it possible to predict the success of a movie based only on its script?

Pedro Argento

General Assembly - Data Science, 2016

The film and TV industry generated 287 Billion Dollars in 2016, and it is expected to generate 324B in 2020.

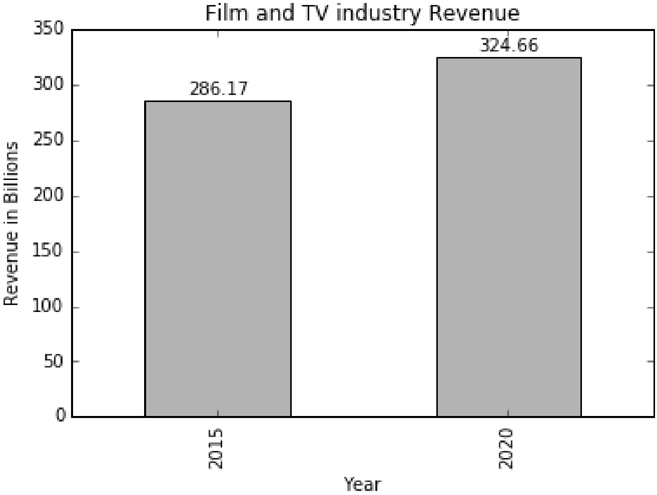


Figure: PWC expects a 13% growth in film revenue in the next 5 years.

Although film industry is strong and profitable, a lot of the movies produced fail to break even.

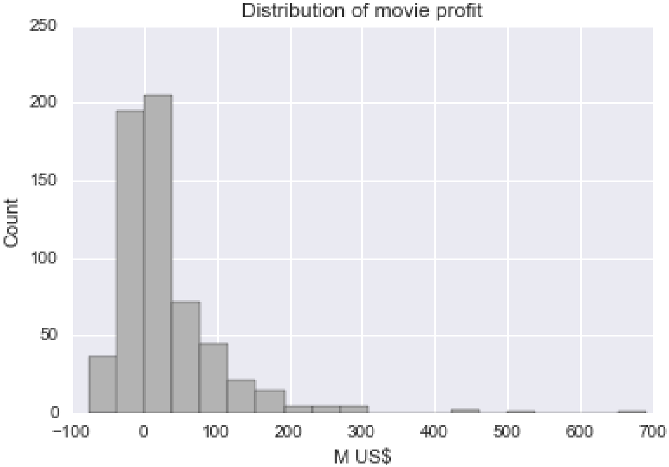


Figure: Almost one third of movies generates loss.

Machine Learning models can help humans make better decisions in three ways:

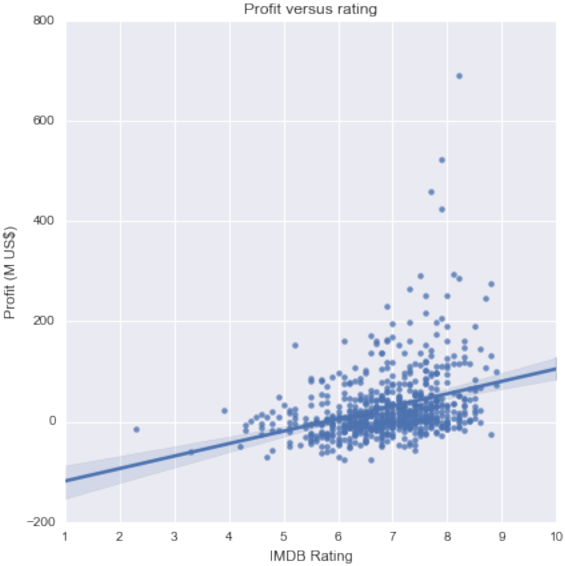
- Get an estimation of how good a movie made from the script can be, before committing to it.
- Help filtering and prioritizing which scripts should be read by a human: An average producer gets at least 10,000 scripts a year, most go unread.
- Help script writers make better scripts.

Success will be defined by the movie rating from IMDB, which we will try to predict.

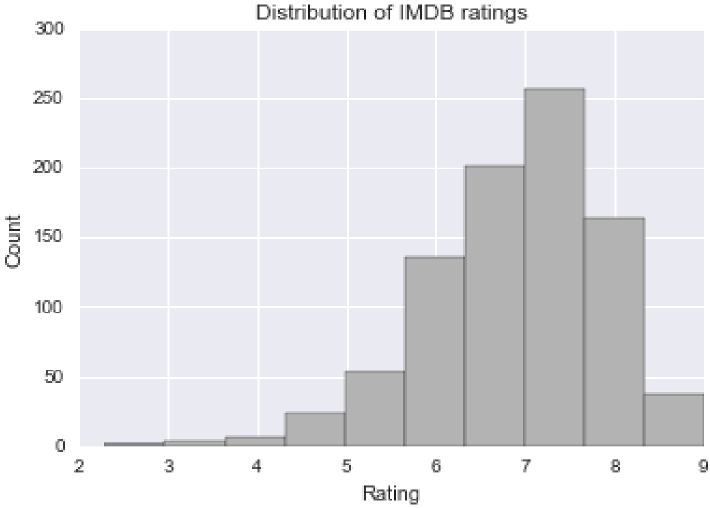
- It is important to use only features available before the movie is made, or the prediction become pointless.
- The sooner in the pre-production stage the model can be used, more useful it will be (eg. before actors are cast).
- As movie quality goes beyond only the script, we can expect somewhat big errors.

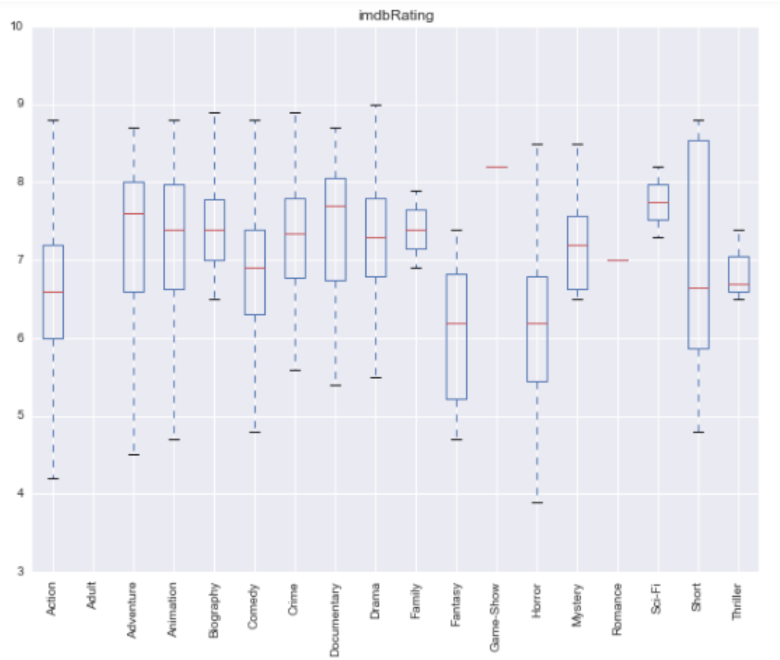
Before going into modeling, its important to understand the database with visualizations.

Movies profits are strongly related to its IMDB Rating

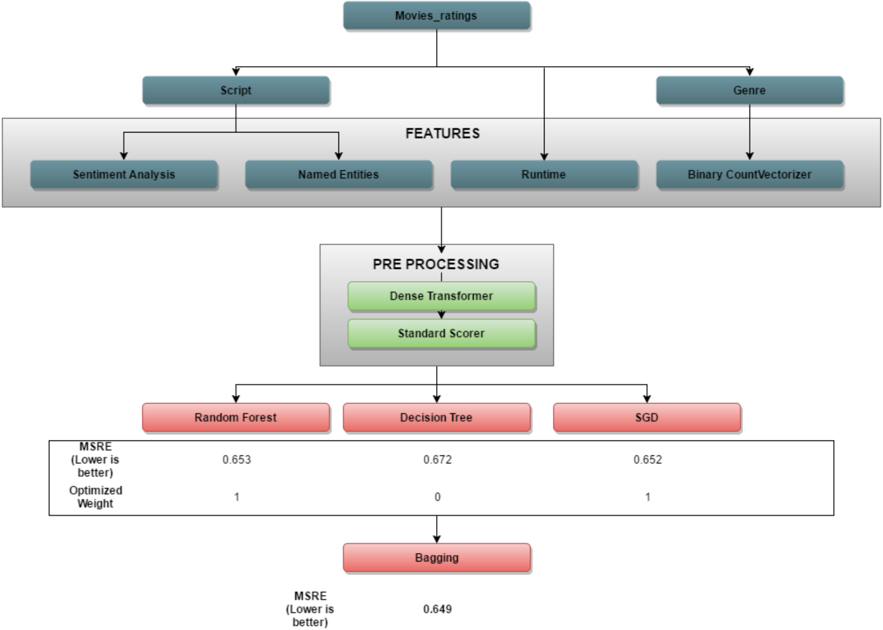


Movie goers are not very demanding, IMDB averages are high.




imdbRating
10
9
8
7
6
5
4
3
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
Game-Show
Horror
Mystery
Romance
Sci-Fi
Short
Thriller

Now its time to do some modeling.



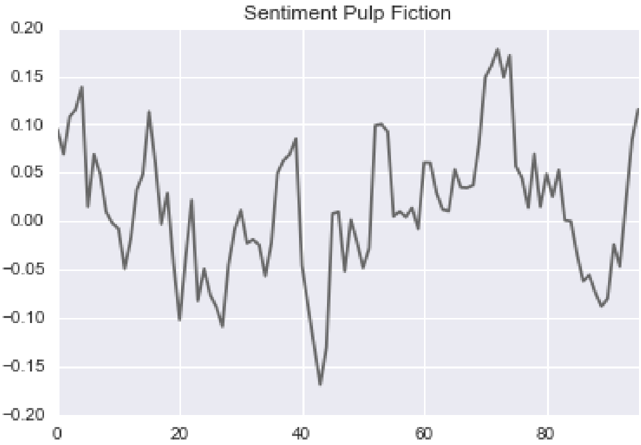
	Random Forest	Decision Tree	SGD
MSRE (Lower is better)	0.653	0.672	0.652
Optimized Weight	1	0	1

	Bagging
MSRE (Lower is better)	0.649

BOY
THAT ESCALATED QUICKLY
quickmeme.com

There are 4 sets of features from 3 different sources

- Script - Named entities: Number of characters, locations and important elements in the scripts (Substantives all in upper case)
- Runtime: How long the movie runs in minutes, it is equivalent to the number of script pages.
- Genre: The genre of the movie. A movie can have multiple genres, thus the CountVectorizer().
- Script - Sentiment Analysis: Metrics from the moving average sentiment of the script.



- Range, Average, Maximum, Minimum, Max/Min locations, Maximum change, Maximum change location, Max negative change, Max positive change, begin sentiment, end sentiment, plot twist index*

*Plot twist index: Defined as the range of sentiment in the last 10% of the movie diveded by the range in the first 90%.

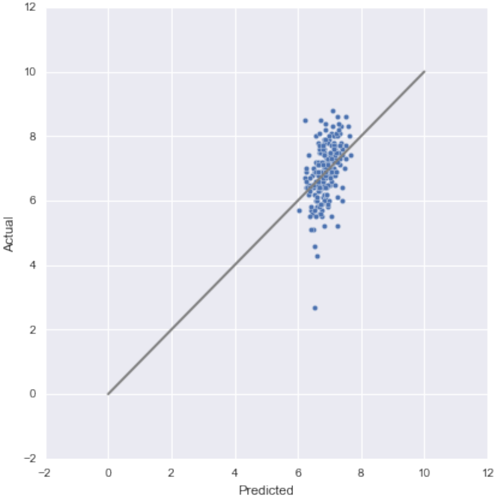
Three models were used and averaged to get predictions

Averaging the models together achieves to reduce the overall variance of the model.

- Random Forest Regressor
- Decision Tree: The worse of the three, was discarded by the optimized weights.
- Stochastic Gradient Descent: The best model, but it has big variance.

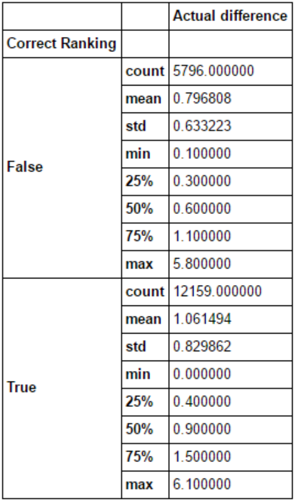
Some results: The model tends to overestimate bad movies and underestimate really good movies

The average error achieved was 0.648.



Given 2 movies, can the model predict witch one is going to do better?

For the 17955 combinations of two scripts, the model picked correctly 12000. And half of the pairs it got wrong had an actual difference of less than 0.6



		Actual difference
Correct Ranking		
False	count	5796.000000
	mean	0.796808
	std	0.633223
	min	0.100000
	25%	0.300000
	50%	0.600000
	75%	1.100000
	max	5.800000
True	count	12159.000000
	mean	1.061494
	std	0.829862
	min	0.000000
	25%	0.400000
	50%	0.900000
	75%	1.500000
	max	6.100000

Next steps...

- Implement a summarizer so a producer can have an idea of what the script is about.
- Create a model that given a script find similar movies that were already made. I have been trying with KDTree, but the results are strange.