Predicting Movie Ratings From IMDB

Christian Hansen

Mission Statement:

Scrape the hell out of IMDB and run regression to find features that predict the highest imdb ratings on the site.

My intention was to specifically look at independent movies but this became difficult as the term is disputed, meaning that what constitutes an independent movie is hard to accurately define as well as scrape.

Scrape as much as possible out of IMDB

Initially:
I started looking at the most voted movies that imdb sorted and selecting as much as I could from the source.

I took as much as I could from the 200 allowed pages to scrape with 50 titles per page. The maximum number of movies: ~50*200 = 10000.

Eventually:

I came to realize that some of this data was corrupted, so I threw it out and stuck to indie movies.

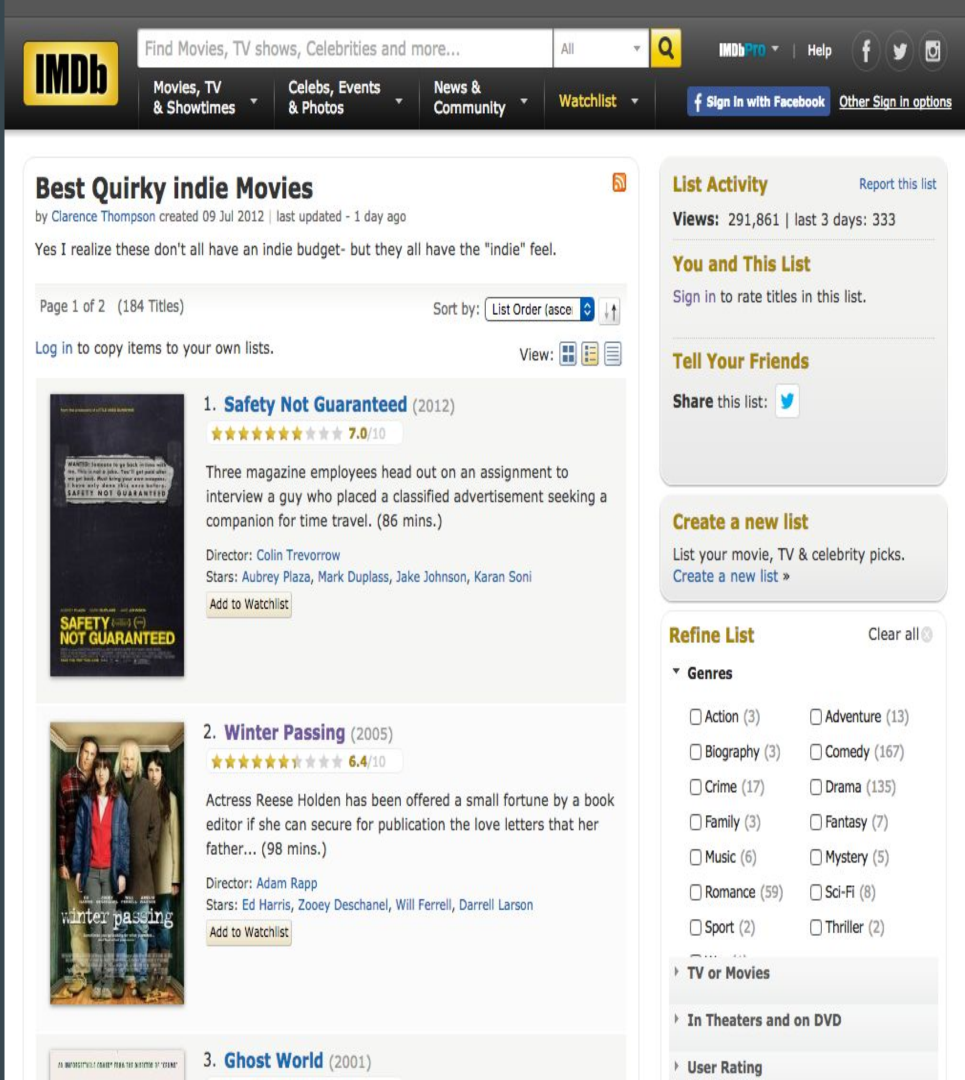
For independent movies I had a harder time finding a list to scrape and it came out to be a lot smaller of a dataset.

The initial Format for the site

I scraped two different pages for indie movies

Each Page had 100 or less movies.

www.imdb.com



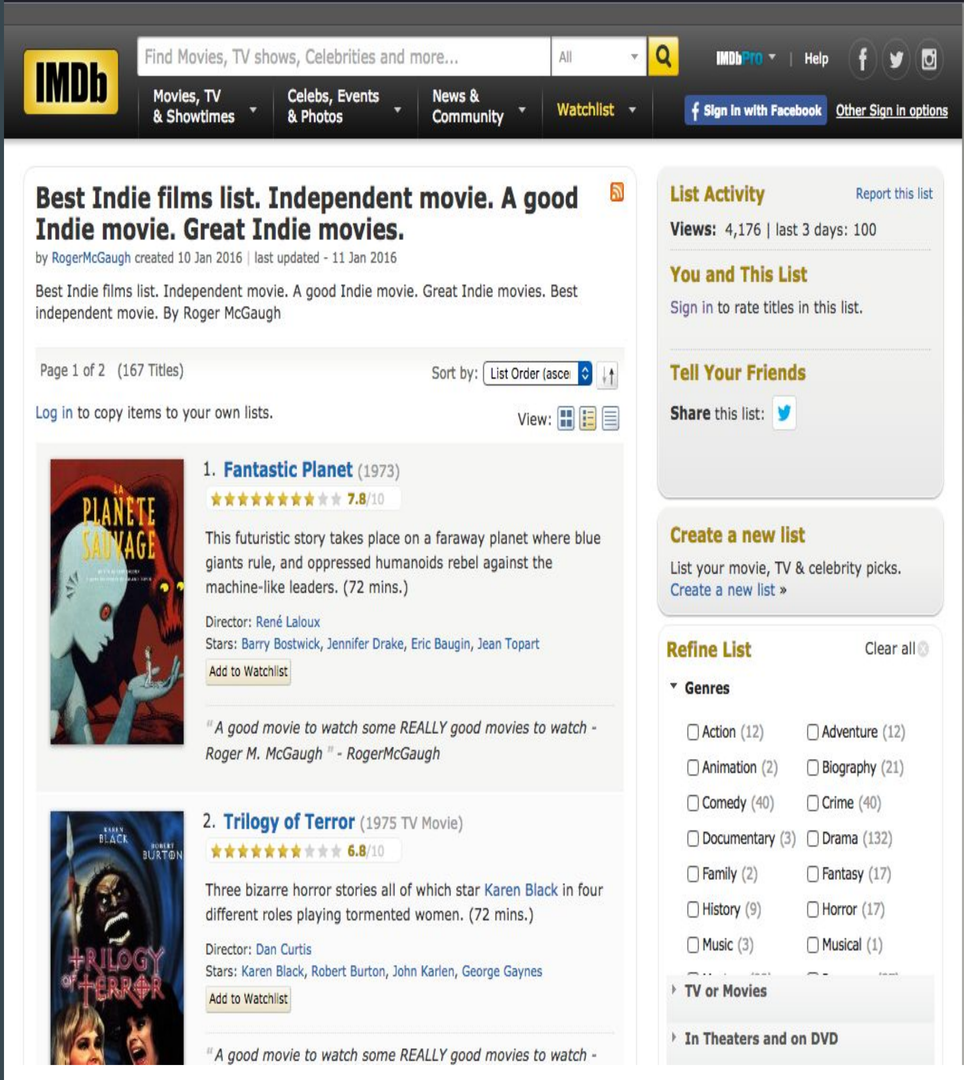
The initial Format for the site

I began scraping all base information off these lists for as many pages as I could.

This came out to around 250 movies.

Ideally.

Only around 200 survived the cleaning.



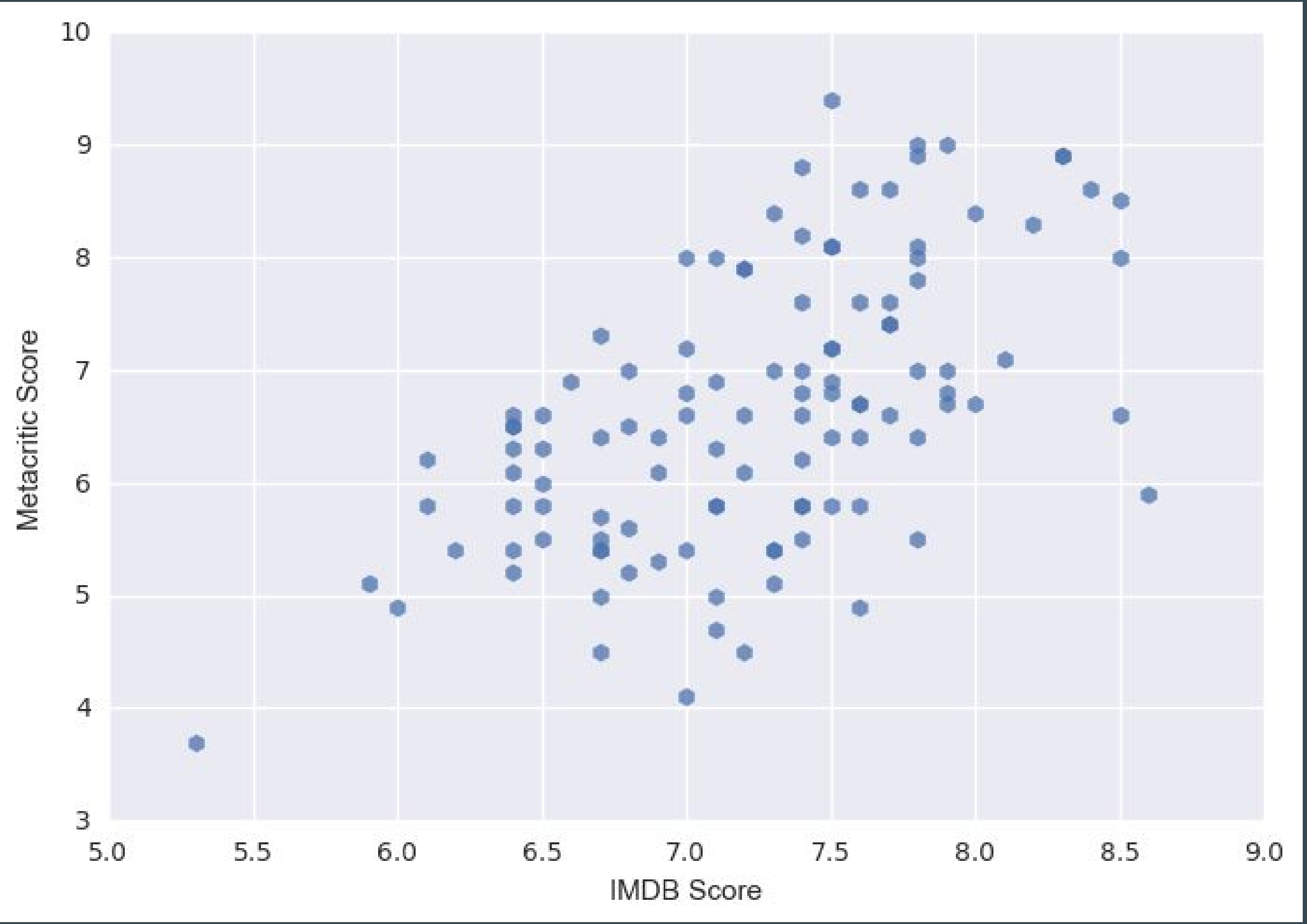
www.imdb.com

Tools used:

- Python (of course)
- BeautifulSoup
- Pandas
- Sklearn packages
- Patience

Metacritic versus IMDB

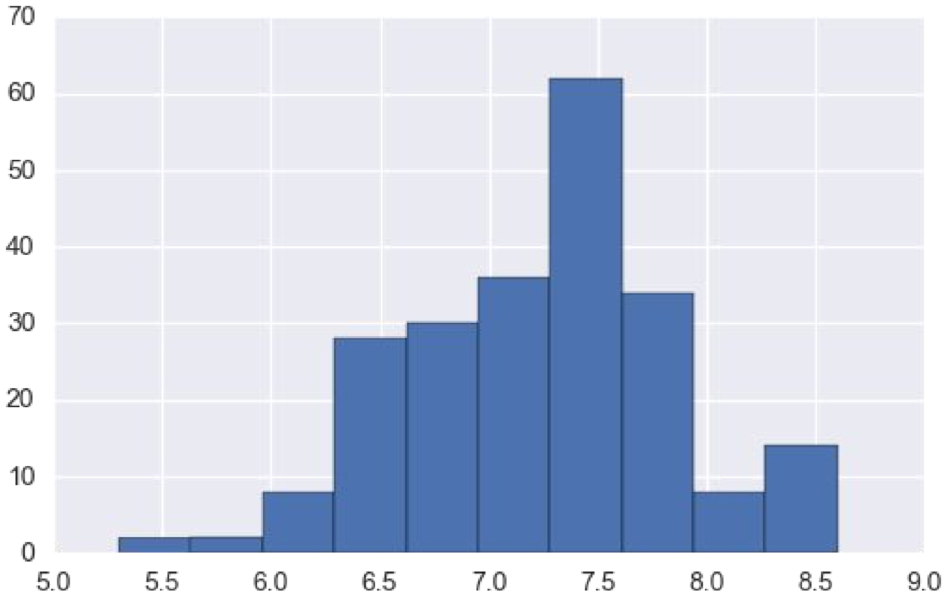
The metacritic score was scaled to match imdb.



EXPLORING THE DATA

IMDB Score Distribution

For independent movies

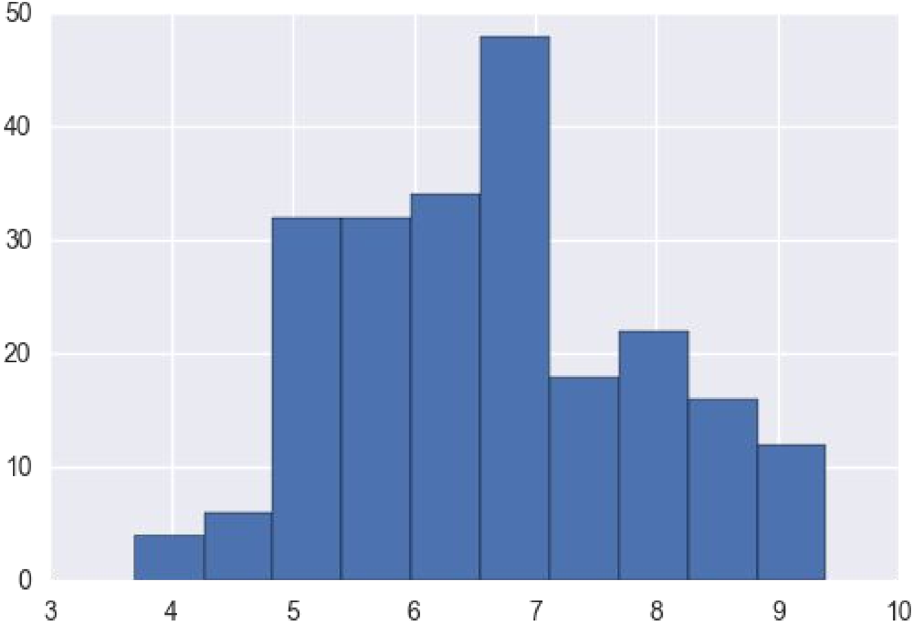


Histogram of the imdb scores

EXPLORING THE DATA

Metacritic Scores Distribution

For independent movies

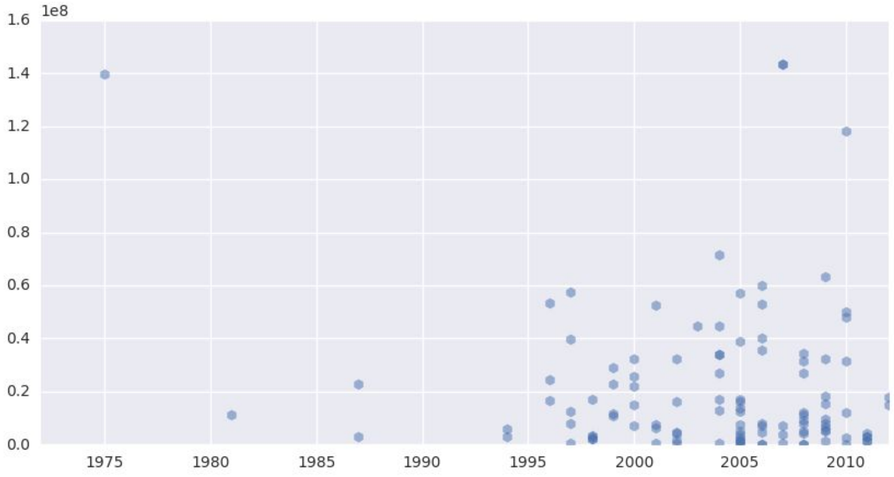


Histogram of Metacritic scores

EXPLORING THE DATA

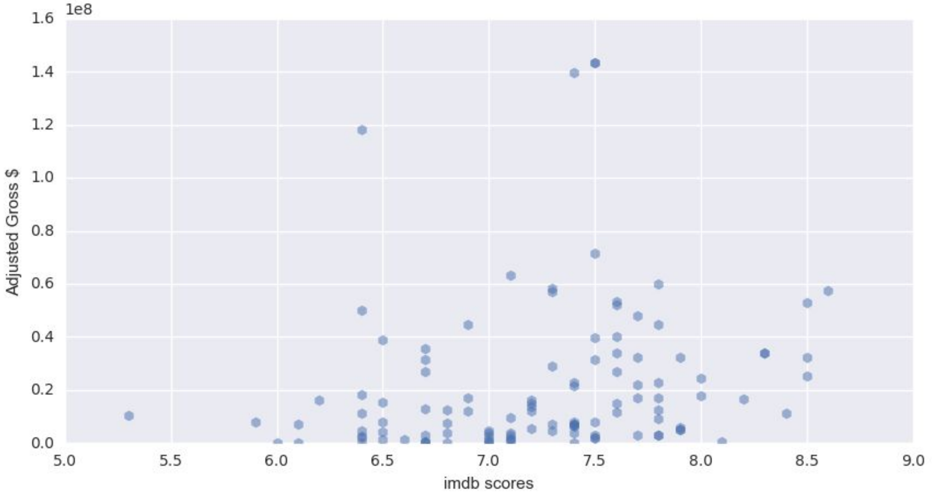
Adjusted gross versus years when the movies were released

More movie data was collected for years in the 90's and beyond, but not much below.



Adjusted Gross versus the known IMDB scores

Very few if any movies in this dataset were rated below a 5. IMDB tends to have a skewed rating system. This is interesting.

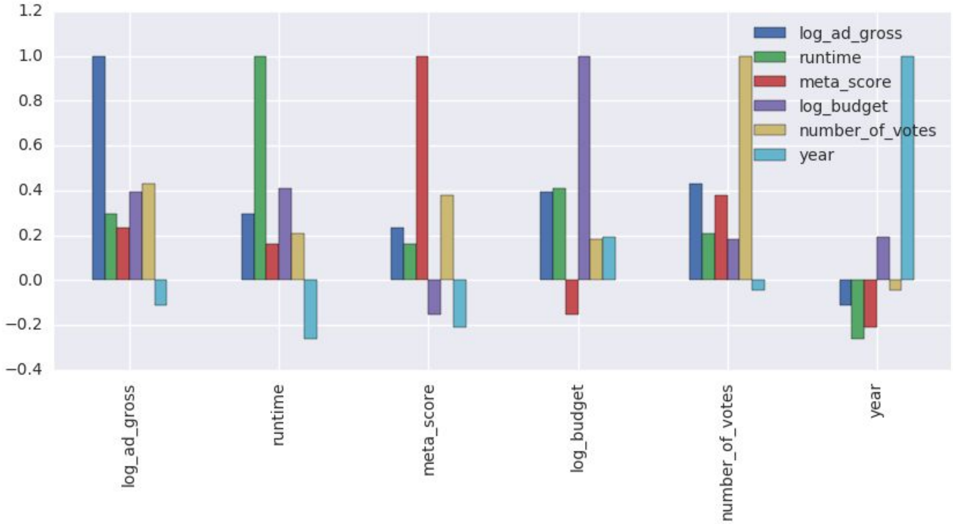


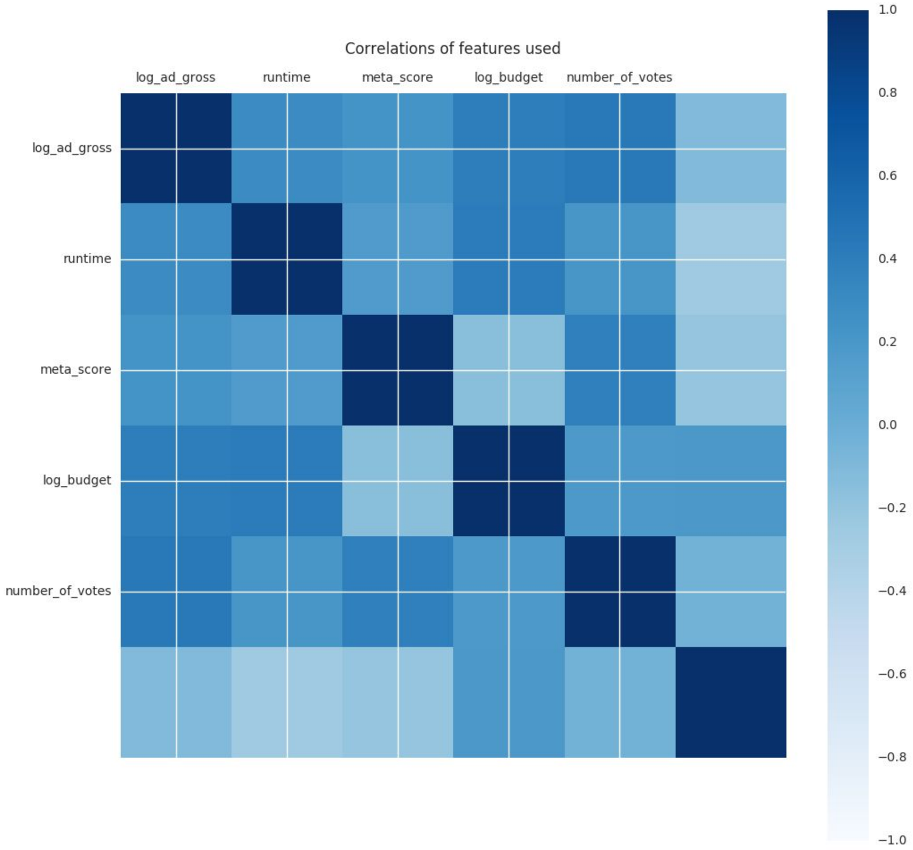
Okay, Features:

- Metacritic Scores
- Runtime in minutes
- Log10 of budget
- Log10 of adjusted gross income
- Number of votes on IMDB
- Estimated tickets
- year

Some features were highly correlated,
Such as the estimated tickets and Metacritic scores. Metacritic had a low correlation to the others for indie movies so I let it tag along with my model building

Correlations!





Model Metrics

All model values are pretty small and probably overfitting for all models. Something to look into in the future when we model.

Linear Regression:

Mean squared error: 0.17117544264192824

R-squared: 0.60954506696640465

Random forest:

Mean squared error: 0.12776655679724114
R-squared: 0.70856168613767989

Grid search random Forest:

Mean squared error: 0.0567824742567
R-squared: 0.991918963246

Gradient Boosted:

mean squared error: 0.13517333034433474
R -squared: 0.55631673556731542

Running various models

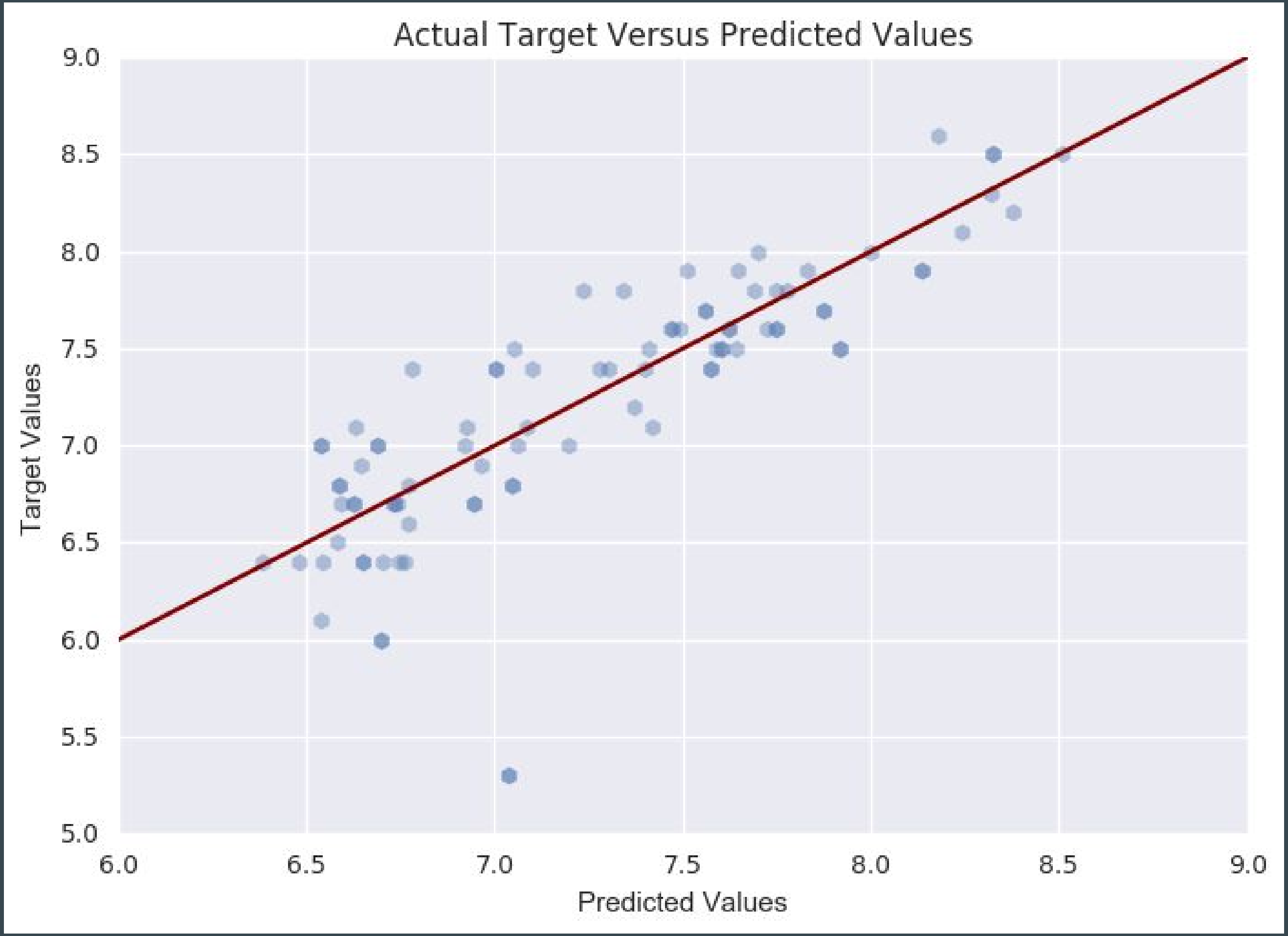
- Linear Regression
- Random Forest
- Grid search Random Forest
- Gradient Boosted Random Forest

The Best:
Gradient Boosted Random Forest

It maintained a decent MSE out of the models and the R^2.

Gradient Boost

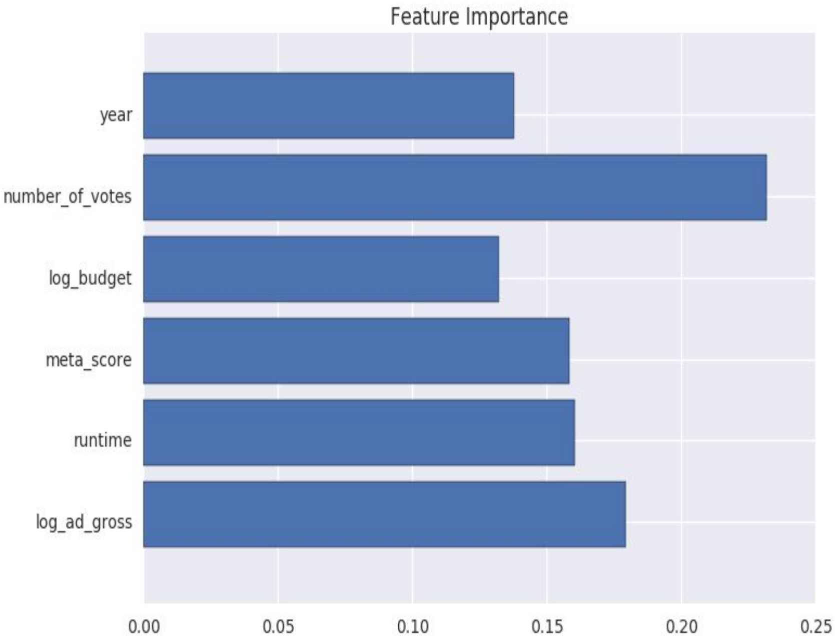
Mean squared error: 0.13517
R-squared: 0.55631



Looking at parameters from Gradient Boosted Random Forest

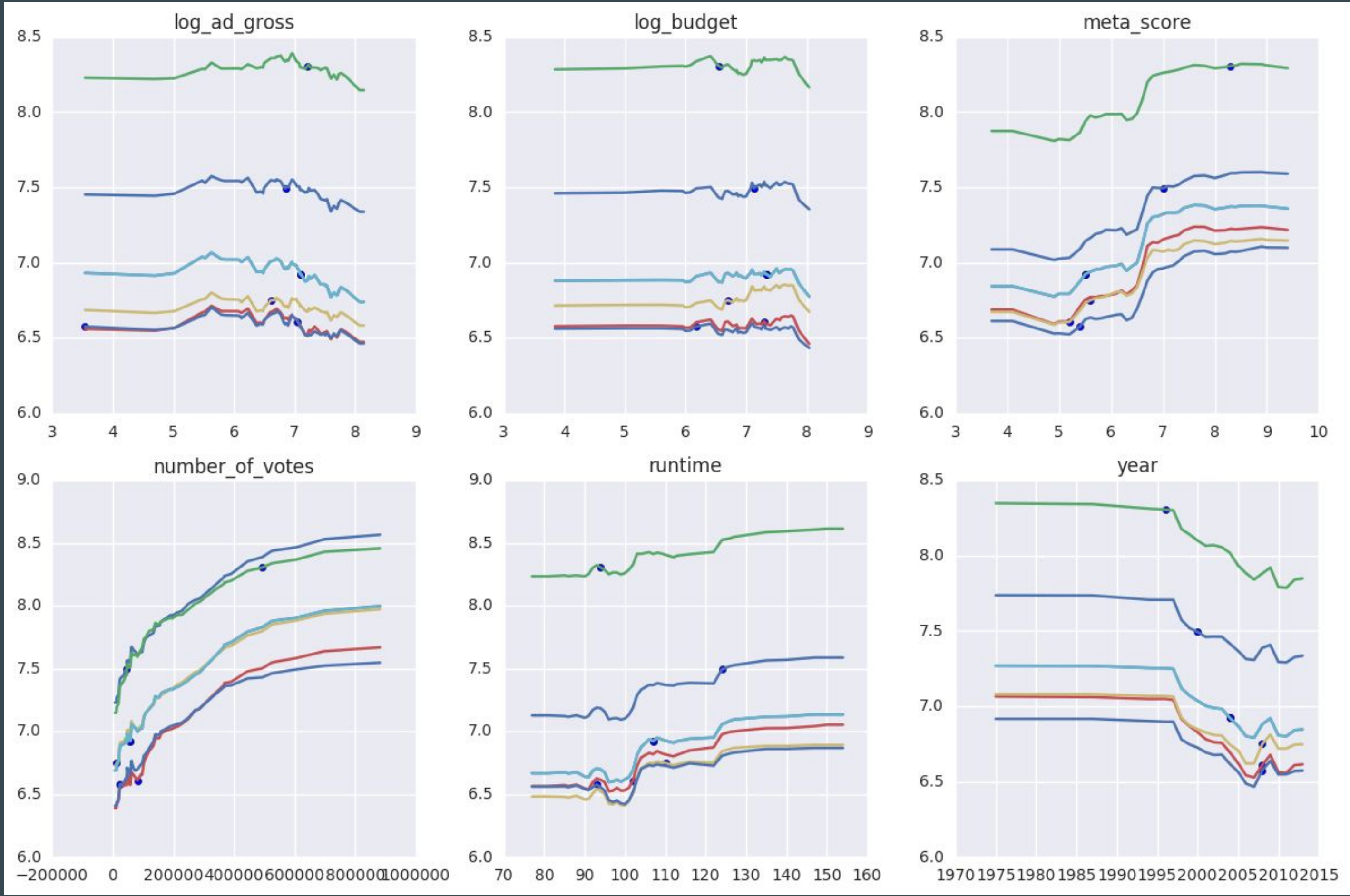
- Most influential coefficients are the number of votes per score and the log of the adjusted gross.
- Considering better grossing movies are ideally better movies as more people pay to see them.
- Reasoning: those who do like a movie or hate a movie could be more likely to vote, but given IMDB's scoring distributions, most seem to vote higher rather than lower. The money factor seems the most appropriate.

HOW IMPORTANT ARE THESE FEATURES

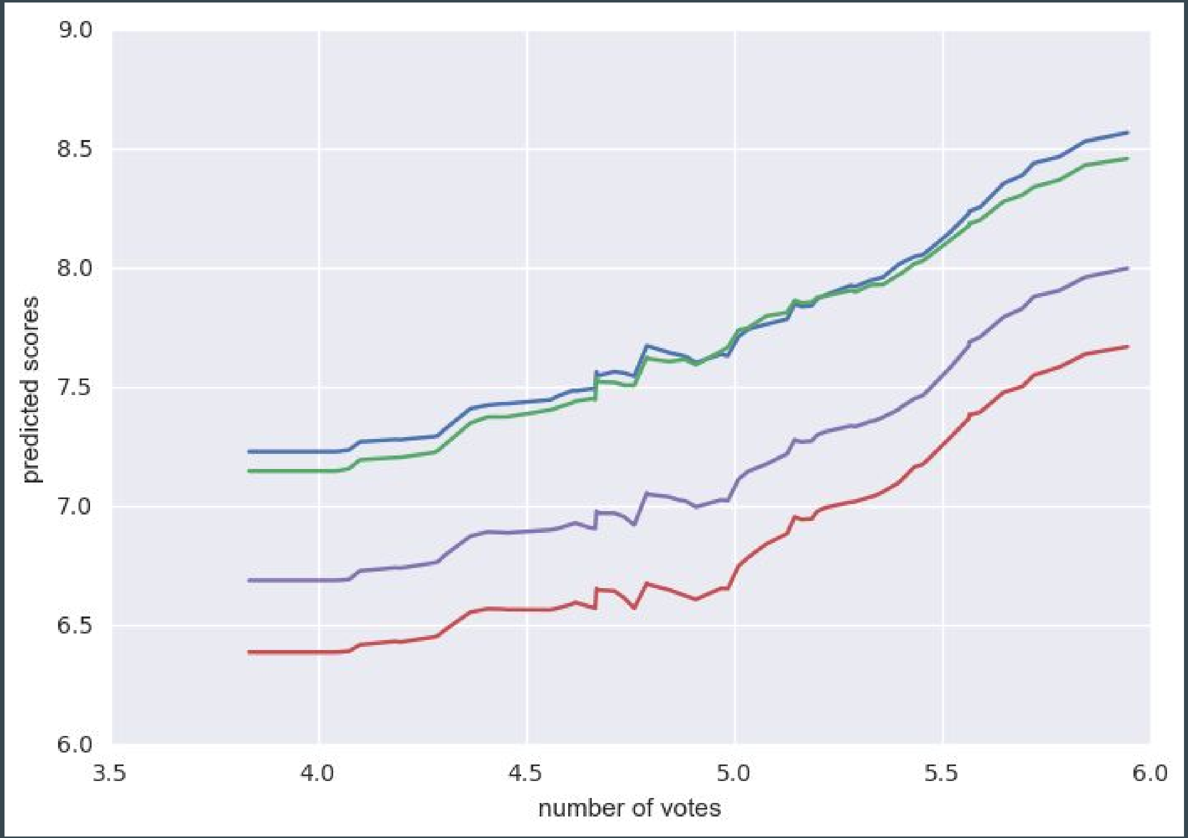


Both the log of the adjusted gross and number of vote
Indicate a high IMDB score

Analyzing the predicted values based on features individually:

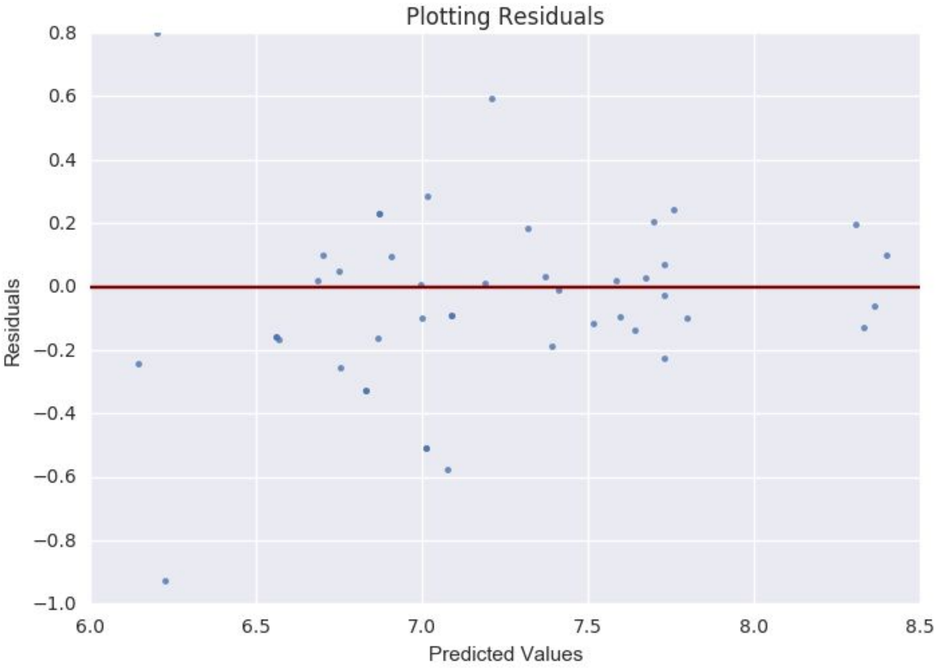


Looking further at predicted values versus the log of the number of votes:



Looking at the residuals of the Gradient Boosted Random Forest

- There are a few key outliers but that is not a problem, because that's more interesting. Let's look at a single case.



This movie was overrated by my movie:

Kick ass. It is overrated.

movie_title	Kick-Ass
meta_score	6.6
number_of_votes	448897
imdb_score	7.7
budget	3e+07
gross	48043505
runtime	117
year	2010
Adjusted_gross	4.80435e+07
Avg. Price	7.89
rate	0.912139
est_tickets	6.08916e+06
log_budget	7.47712
log_ad_gross	7.68163
sqrt_num_votes	669.998

This Movie was underrated by my model:

I agree with this too. Totally underrated. Totally confusing.

movie_title	Primer
meta_score	6.8
number_of_votes	76752
imdb_score	7
budget	7000
gross	424760
runtime	77
year	2004
Adjusted_gross	424760
Avg. Price	6.21
rate	0.717919
est_tickets	68399.4
log_budget	3.8451
log_ad_gross	5.62814
sqrt_num_votes	277.042

Conclusions

- Overall low numbers of values.
- Comparing this more directly to see how a model trained on a larger and more widespread dataset would react to the indie subset.
- Novel model correlations and agreeable results.