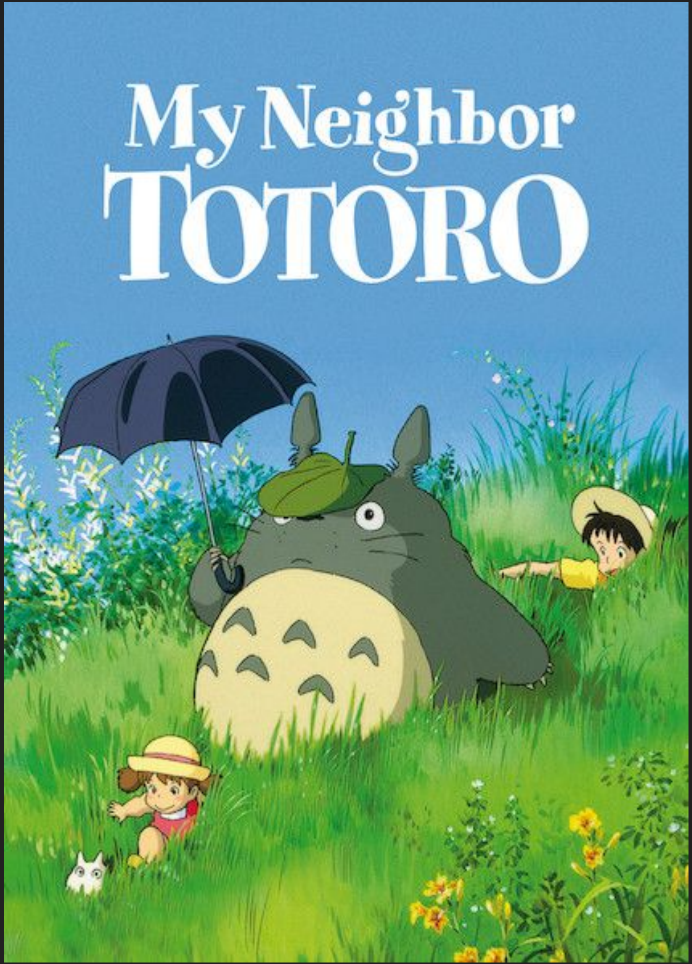
A Budding Data Scientist's First Modeling Journey: Japanese v. American Animated Films

Binh Hoang

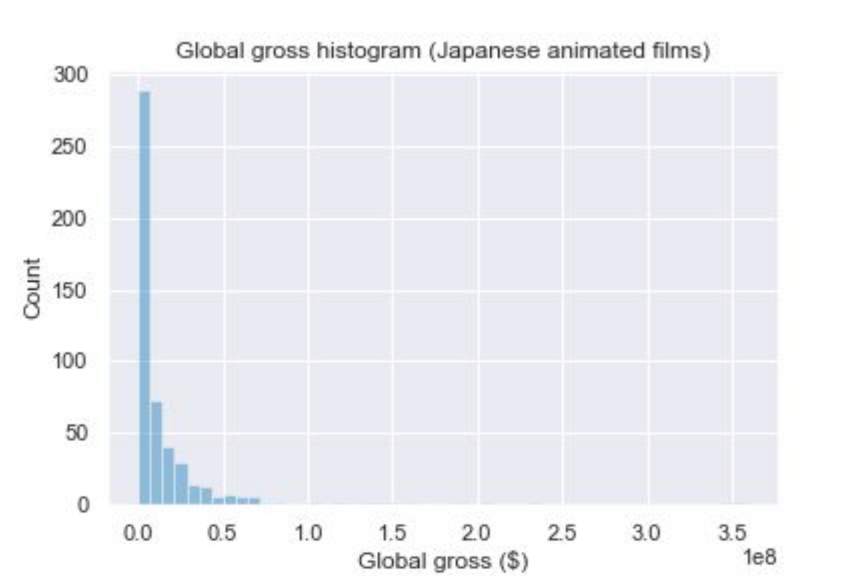
The challenge

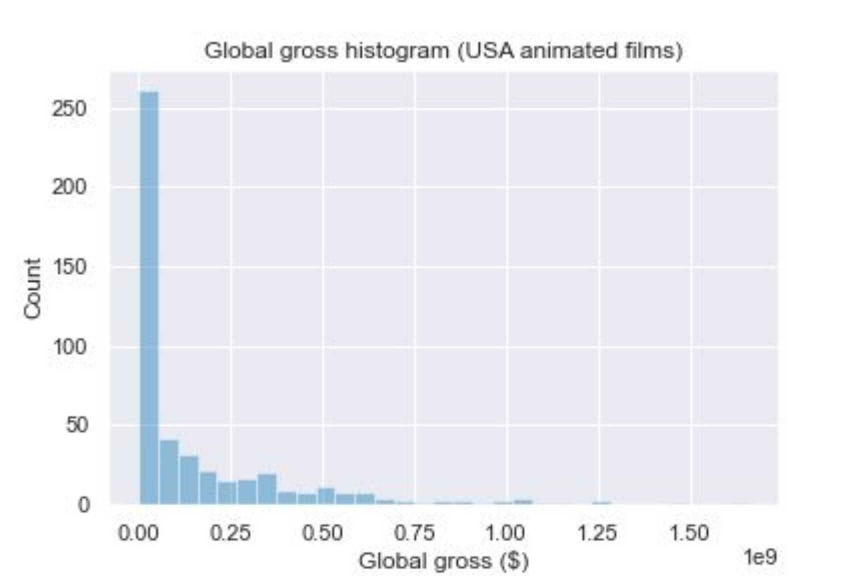
Can I predict the global box office gross for...?


My Neighbor
TOTORO


Disney
FROZEN
© 2019 Disney

Modeling difficulty: medium





Methodology

Scrape

Clean

Model

Scrape

imdb.com

- 2,846 animated films (almost evenly split between US/Japan)
- Packages: BeautifulSoup, requests

Clean

- Dropped 1,796 (63% of all data) data points b/c missing global gross (target)

- Dropped another 80 data points b/c films were produced in Japan and US

Final dataset

	American Films	Japanese Films
Data Points	474	496
Missing Budget Values	142	**451**

Final dataset

	American Films	Japanese Films
Data Points	474	496
Missing Budget Values	142	**451**

Important point! Will come back to this

Filled with median

Modeling approach

- OLS regression
- Two separate models (Japan model/US model)
- 5 k-fold cross-validation
- No regularization

US model

Feature coefficients:

budget	2.25
budget * is_summer_release	**0.99**
budget * is_xmas_release	**0.19**
oscar_wins	41,484,604.23
imdb_user_rating	10,066,891.63
imdb_user_rating_count	676.18
years_since_release	-760,422.17

US model

Training R^2:	0.695
Val R^2:	0.686
Test R^2:	0.661

Training MAE ($):	92,742,534.38
Test MAE ($):	73,133,756.76

Decent R^2, but high mean absolute error

Japan model

Feature coefficients:

imdb_user_rating_count	279.97
non_oscar_wins	3,596,523.12
years_since_release	-369,800.85
is_golden_week_release	-9,869,910.32
is_summer_release	6,034,622.74
is_xmas_release	2,255,158.46

No budget:

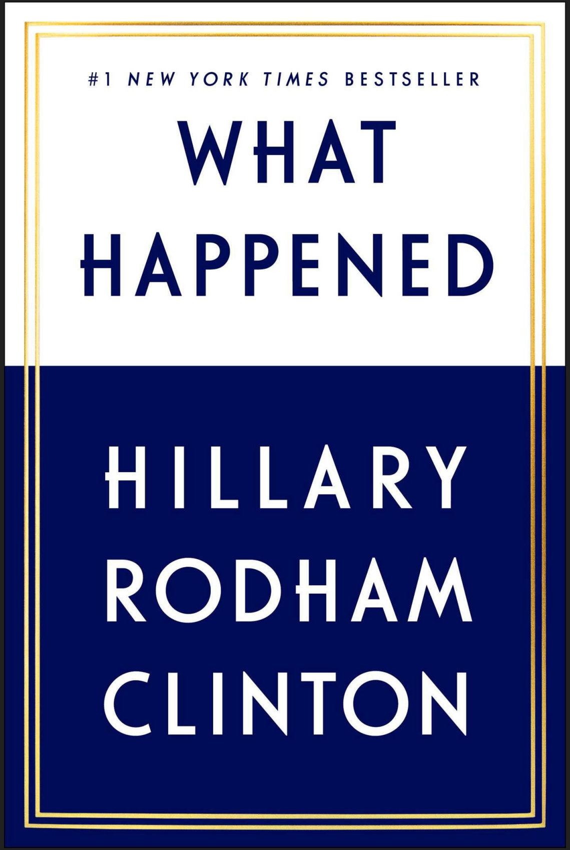
budget feature reduced validation R^2 by .01 (incomplete budget data caused issue)

Japan model

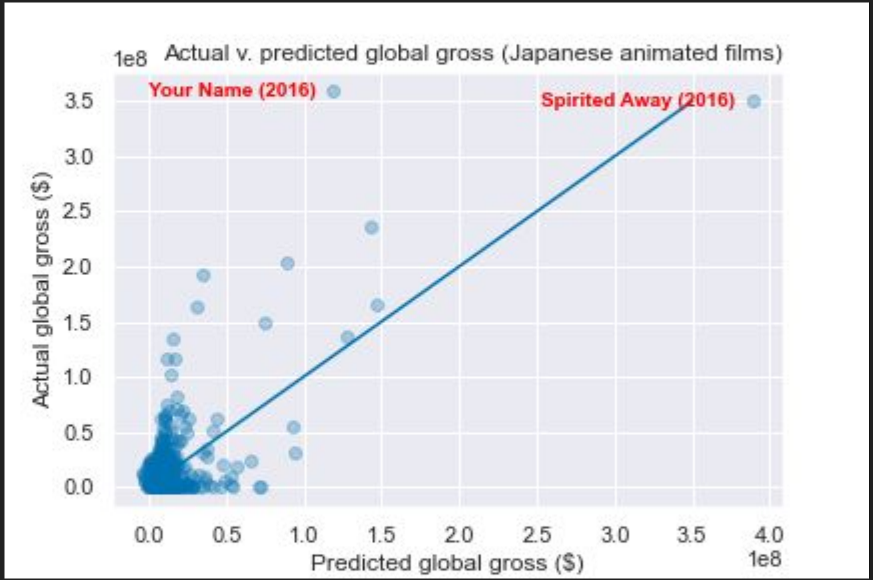
Training R^2:	0.491
Val R^2:	0.285
Test R^2:	0.059
Training MAE ($):	15,242,610.013
Test MAE ($):	15,286,223.419

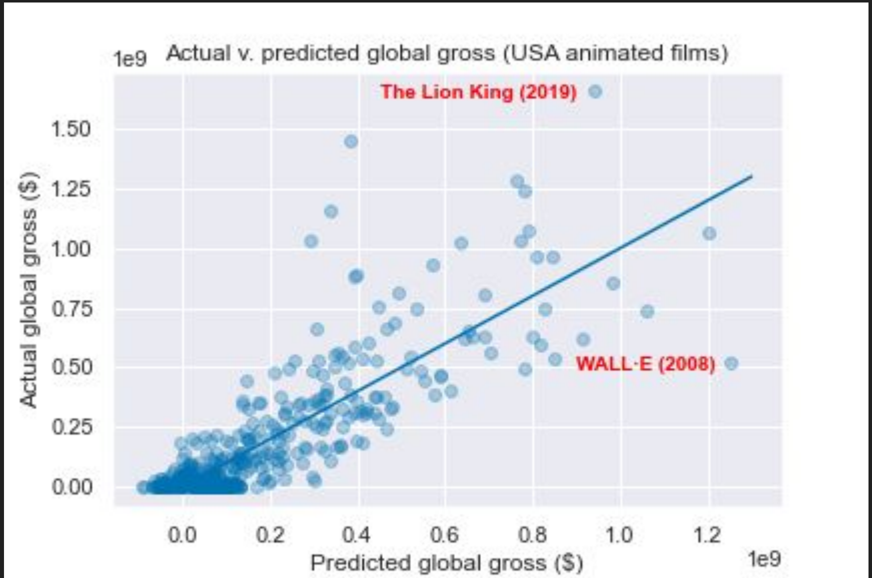
Two not so great models, with the Japan model almost having no predictive power

What happened?

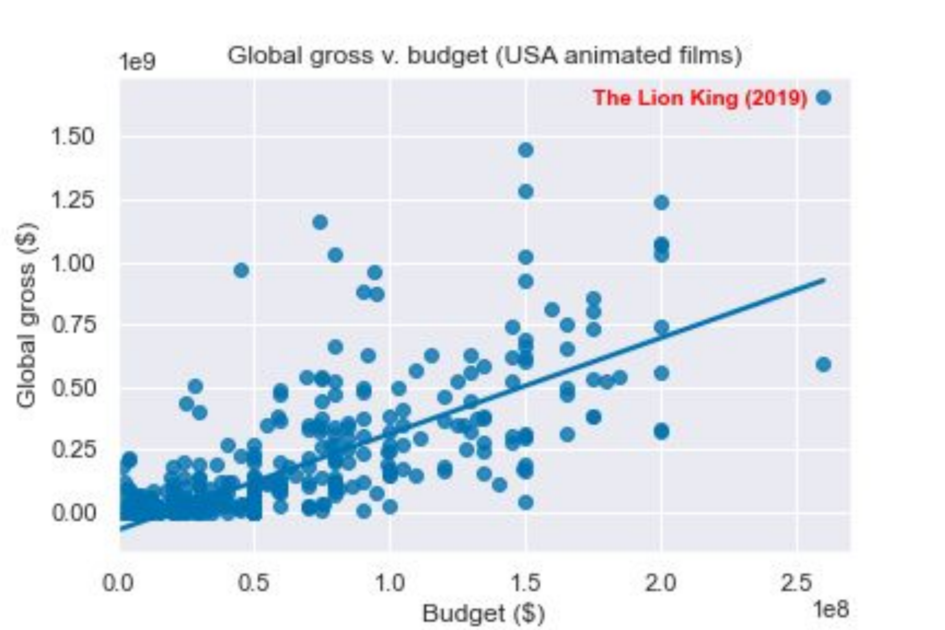


Prediction error increases for films with higher global gross (error heteroskedasticity)





Missing important budget feature in Japan model (potentially losing ~.5 in R^2)



Training R^2:	0.511
Test R^2:	0.476

Feature coefficients:

budget	3.84

Residual analysis (models mostly underpredicted)

US abs largest residuals:

1. Frozen II
2. Minions
3. Despicable Me 3
4. WALL·E
5. The Lion King (2019)

Underpredicted by as high as ***$1.07 bn***

Japan abs largest residuals:

1. Your Name
2. Weathering With You
3. Pokémon: The First Movie
4. Pokémon the Movie 2000
5. Ponyo

Underpredicted by as high as ***$241 mn***

Residual analysis (models mostly underpredicted)

US abs largest residuals:

1. Frozen II
2. Minions
3. Despicable Me 3
4. WALL·E
5. The Lion King (2019)

Underpredicted by as high as ***$1.07 bn***

Japan abs largest residuals:

1. Your Name
2. Weathering With You
3. Pokémon: The First Movie
4. Pokémon the Movie 2000
5. Ponyo

Underpredicted by as high as ***$241 mn***

Models missing an animation company feature!

What I learned/takeaways

- Poor data produces poor models (obvious, but learned this the hard way)
- Do residual analysis earlier
- Scape more than you need to
- An American website may not be the best data source for Japanese films

Future work

- Create two working models and compare/contrast them as way to gain business insights into Japanese v. American animated films (original project goal)


THANK YOU
FOR LISTENING TO MY
PRESENTATION
memegenerator.net

Appendix

Literature review:

- The determinants of box office performance in the film industry revisited (N.A. Pangarker and E.v.d.M. Smit)
- A study on box-office revenue: How user and expert ratings determine movie success (Sylvain Dingenouts)

Appendix

Scraping issues:

1. mpaa_rating was not scraped properly (missing certain ratings like TV-G) due to improper scraping logic
2. usa_release_date was not scraped properly (some release dates from other countries were pulled in) due to improper scraping logic

Appendix

Created a function to record my cross-validation scores for each feature engineering/model selection iteration.

Helps systematize workflow.

```python
def record_cv(mean_train_score, mean_val_score):
    cv_dict = {}
    model = input("Model: ")
    label = input("Iteration description: ")
    cv_dict['model'] = model
    cv_dict['label'] = label
    cv_dict['mean_train_score'] = mean_train_score
    cv_dict['mean_val_score'] = mean_val_score
    return cv_dict
```