Unit 3 Capstone – Predicting movie IMDb score success

Jeff Biehle

May 16, 2019

A little bit about me. . .

- Live in Dripping Springs, TX
  - Outside Austin, “just west of ‘Weird’ “
- 30+ years in high-tech
  - Primarily business development/strategic alliances
- Decided I needed a career switch
- Have been intrigued with data science for several years
- Joined Thinkful in February

Overview

- **Why movies?**
- **The question**
- **Data Fun!**
- **Model Selection/ Tuning**
- **Challenges**

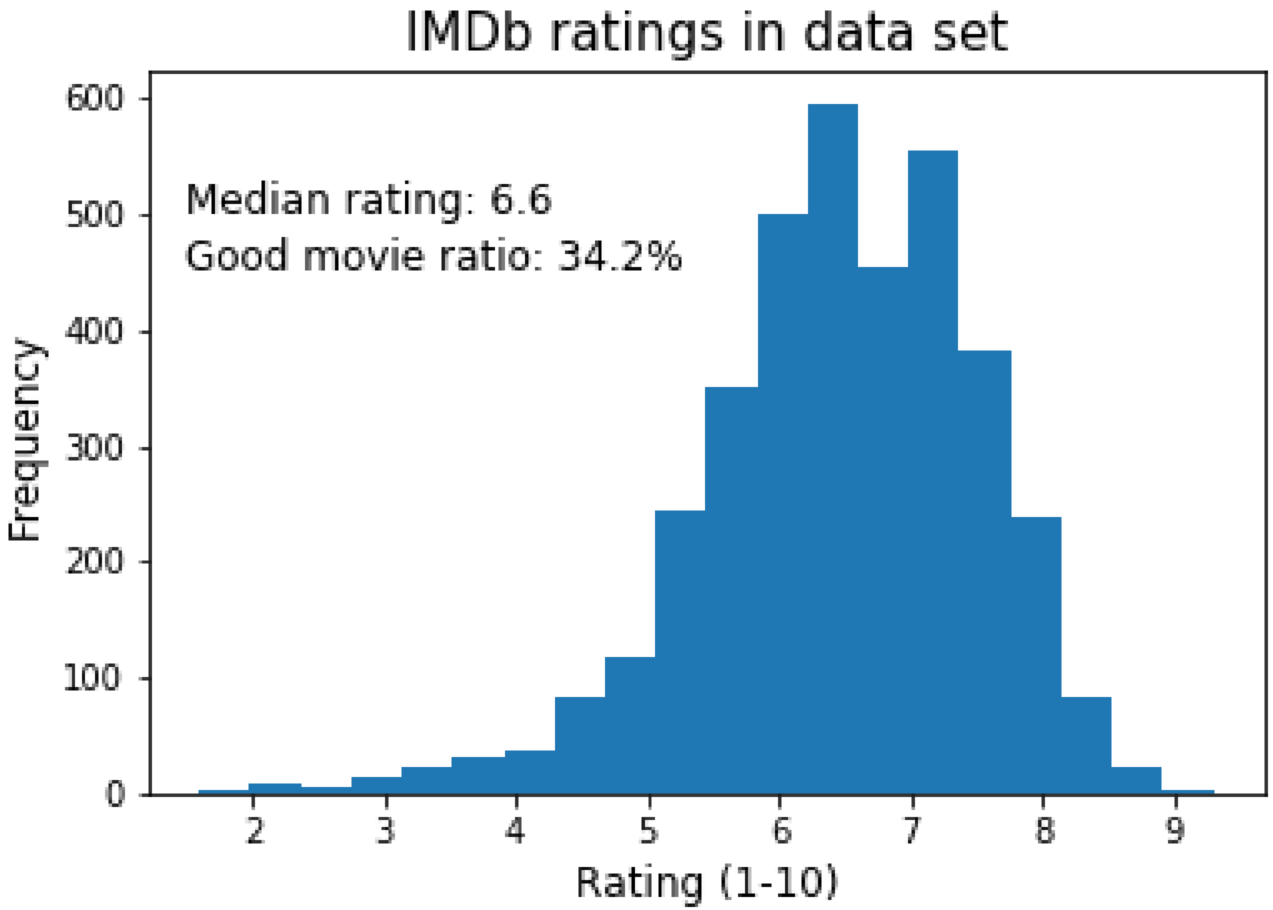
Why movies?

- Everybody loves the movies!
- Popularity (and business use) of sites like IMDb, Rotten Tomatoes, and boxofficemojo make it difficult to escape movie stats and ratings
- IMDb is a very influential site for people to determine what movies they might want to watch
  - A good score is essential to garner more viewers

The question

What movies will garner a good IMDb score?

- Generally defined as >= 7.0 (on a scale of 10)



The data set

Found on Kaggle

https://bit.ly/2J9XjLt

Over 5,000 movies from across the globe

65 countries/47 languages

US represents ~75% of all movies

100 years of data

1916 – 2016

The variables

Basic info

movie_title	country	duration	aspect_ratio
title_year	language	color	plot-keywords
genres	content_rating	imdb_link	

Financials

budget	gross

People

director	actor_2_name	facenumber_in_poster
actor_1_name	actor_3_name	

Facebook likes

actor_1	cast
actor_2	movie
actor_3	director

Ratings info

num_critics	num_users_for_review
num_voted_users	

Data set snapshot

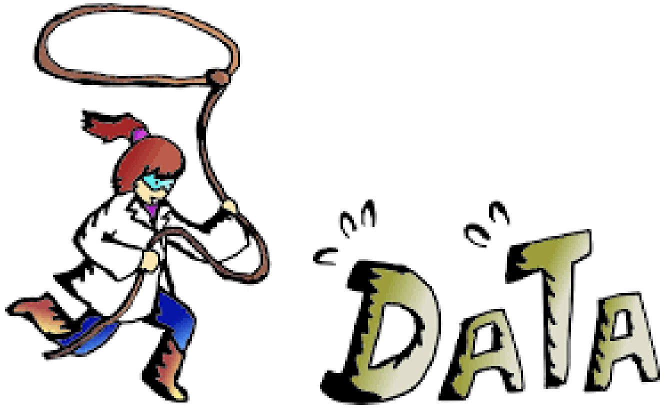
	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross	genres	actor_1_name	movie_title	num_voted_users
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.0	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.0	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.0	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	448130642.0	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	NaN	Documentary	Doug Walker	Star Wars: Episode VII - The Force Awakens ...	8
5	Color	Andrew Stanton	462.0	132.0	475.0	530.0	Samantha Morton	640.0	73058679.0	Action|Adventure|Sci-Fi	Daryl Sabara	John Carter	212204
6	Color	Sam Raimi	392.0	156.0	0.0	4000.0	James Franco	24000.0	336530303.0	Action|Adventure|Romance	J.K. Simmons	Spider-Man 3	383056

Data "quirks and peccadilloes"

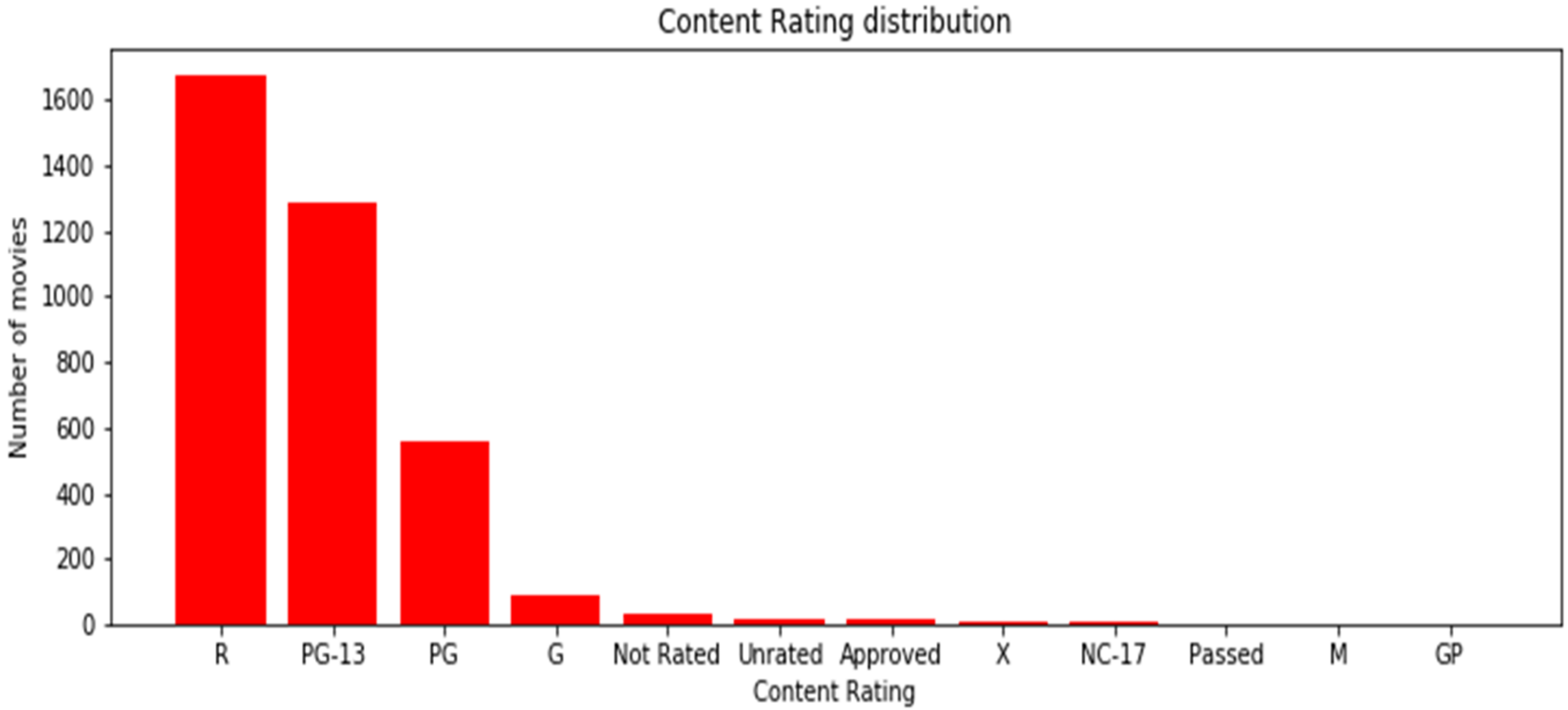
- Lots of NaN's in important variables
- Financial numbers are inconsistent
  - Gross numbers appear to be in $US, but budget is in local currency for many countries
- Facebook data is crucial but skewed toward later movies
- Movies are listed under multiple genres

Data fun!

- Clear the NAN’s
  - Nearly 25% lost due to lack of key variables such as gross & budget
- Remove text variables
  - Actor/director/movie names, etc.
- Split genres and create new genre categorical variables
- Create other categorical variables
  - Content, USA, decade
- Rectify inconsistent financials
- Scale variables using logs to more normalize them

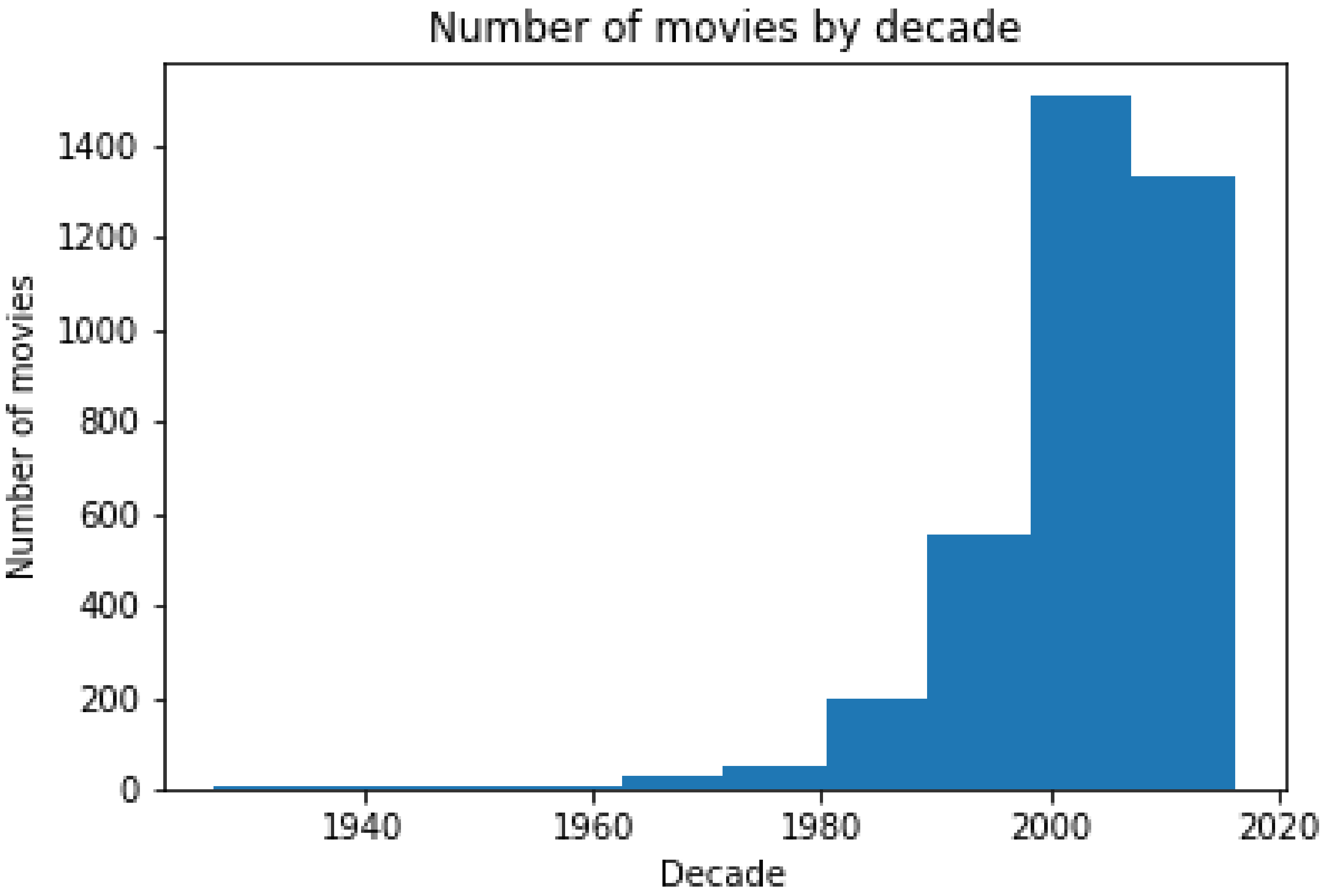


Sample variable distributions – Content rating



- Made 3 new categorical variables for ‘R’, ‘PG-13’ and ‘PG’ and dropped the content_rating variable

Sample variable distributions – Movie year



- Since most movies in dataset were 1980 and later, created categorical “post-1980” and dropped the year column

Sample variable distributions -- Countries

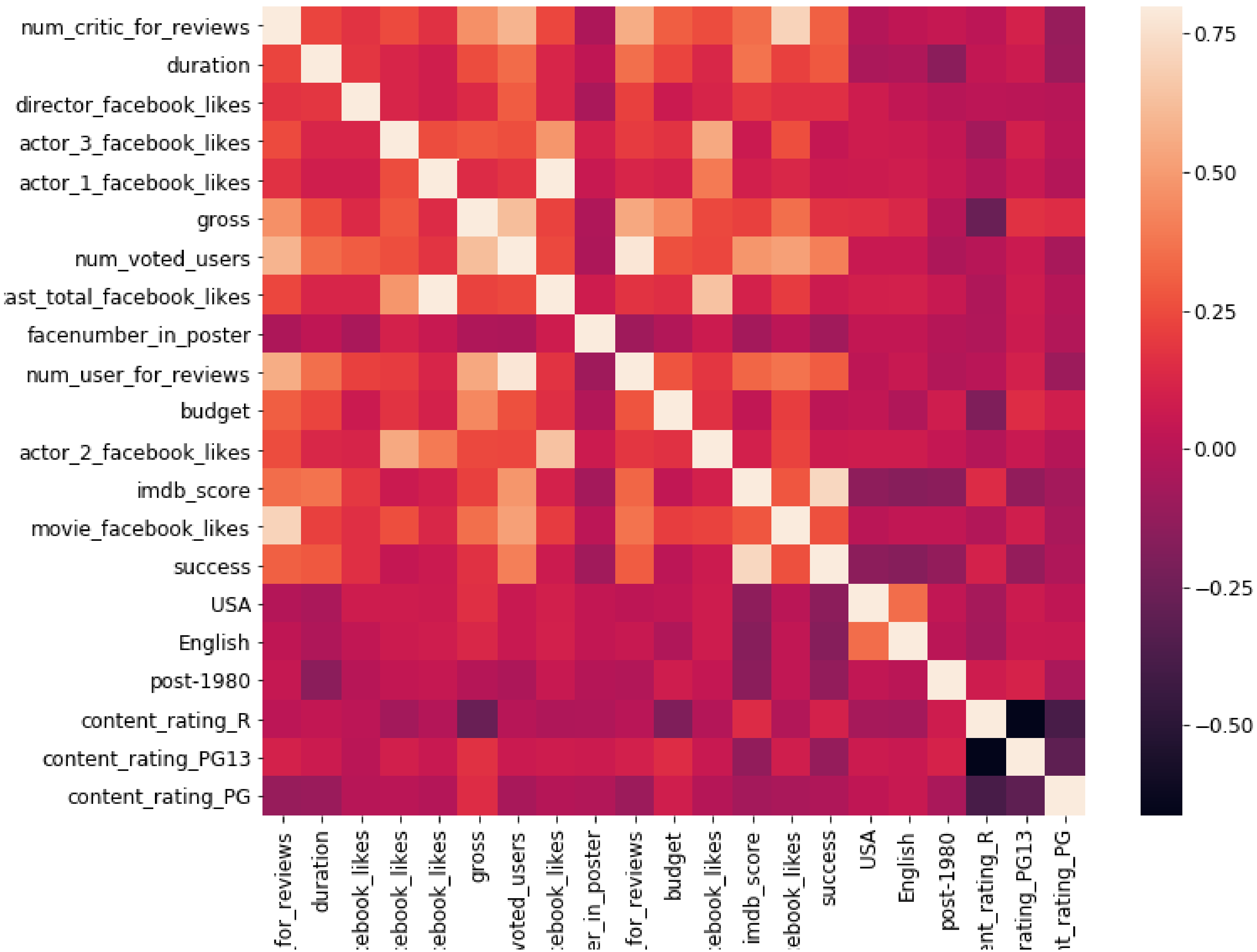
```
In [110]: df.country.value_counts()
```

```
Out[110]: USA               2961
          UK                 313
          France             101
          Germany             79
          Canada              59
          Australia           39
          Spain               21
          Hong Kong           13
          China               12
          Italy               11
          New Zealand         11
          Denmark              8
          Ireland              7
          Mexico               6
          Brazil               5
          India                5
          Iran                 4
          Norway               4
```

```
Norway              4
Netherlands         3
Czech Republic      3
South Africa        3
Argentina           3
Russia              3
Romania             2
Hungary             2
Taiwan              2
Chile               1
Official site       1
Georgia             1
Afghanistan         1
West Germany        1
Indonesia           1
Israel              1
Finland             1
Iceland             1
```

- Created ‘USA’ categorical variable (as vast majority are US movies) and dropped ‘country’ variable

Variable correlation



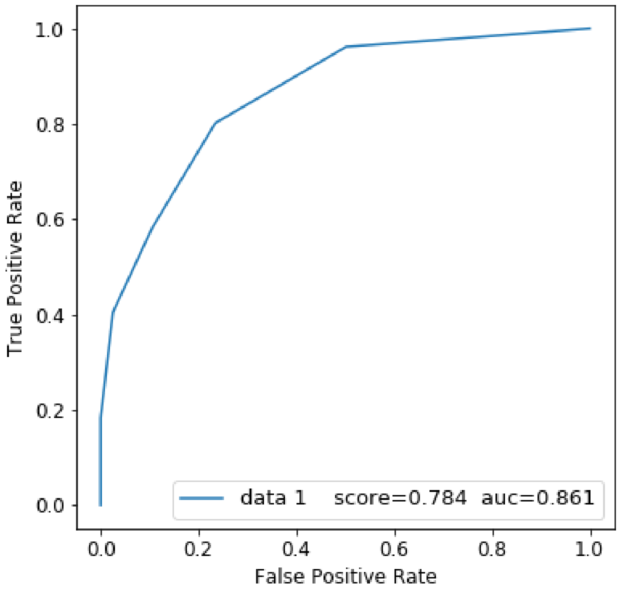
- Facebook “actor_” and “num_” variables can be combined

Run it!

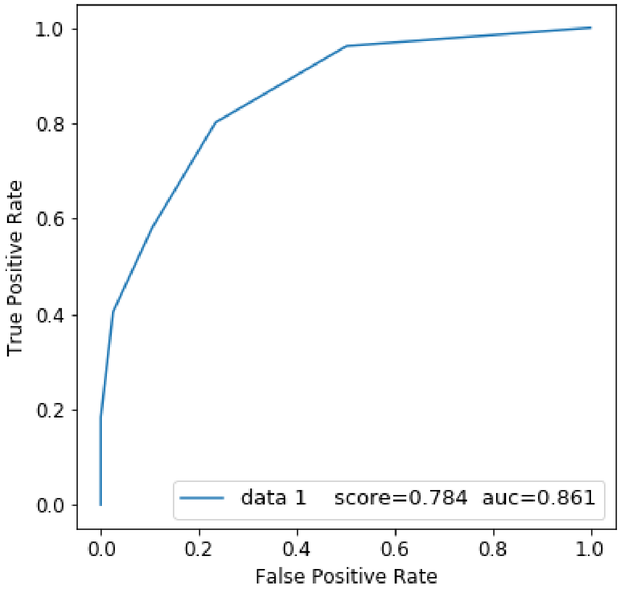
- Applied multiple models to the question
  - KNN
  - Decision Tree
  - Random Forest
  - Ridge Logistic Regression
  - Lasso Logistic Regression
  - Gradient Boosting
- Gives me an idea of how each performs and then I can choose & tweak

Initial results

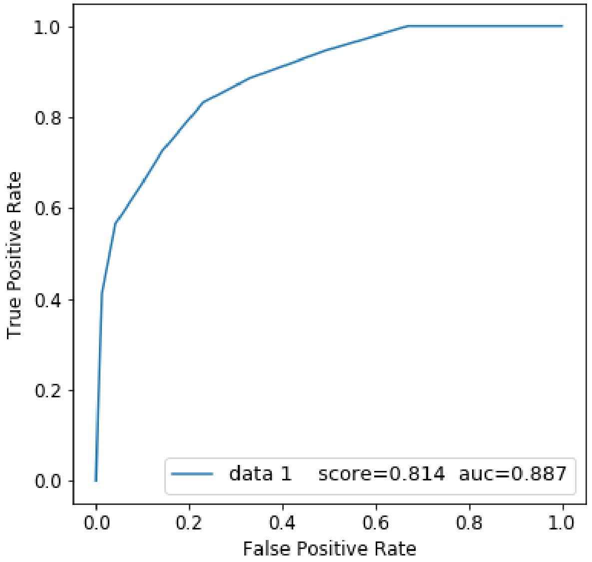
KNN: score = 0.784



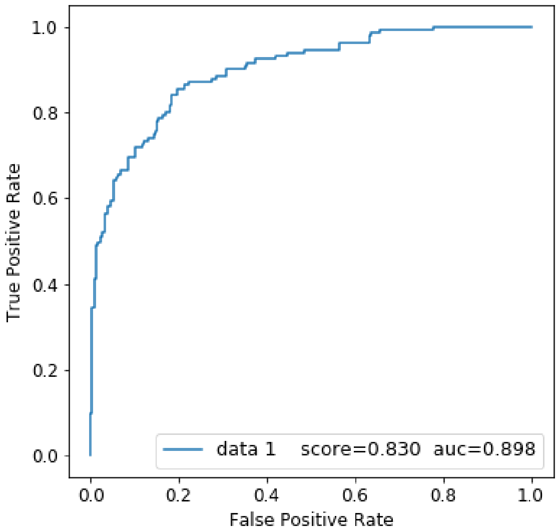
Decision Tree: score = 0.784



Random Forest: score = 0.814

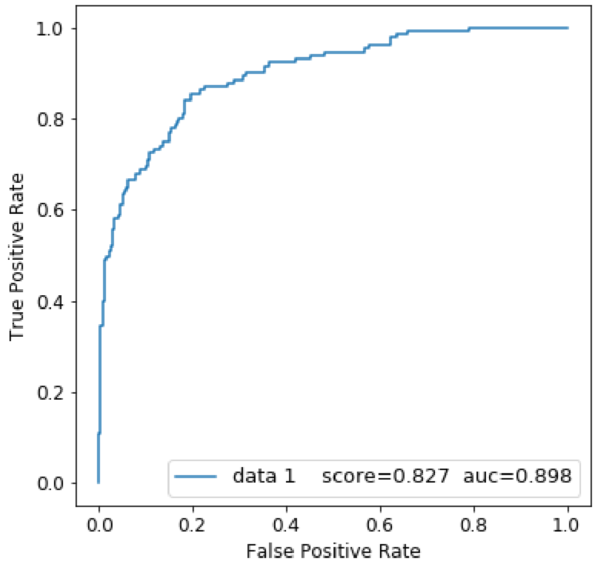


Ridge: score = 0.830

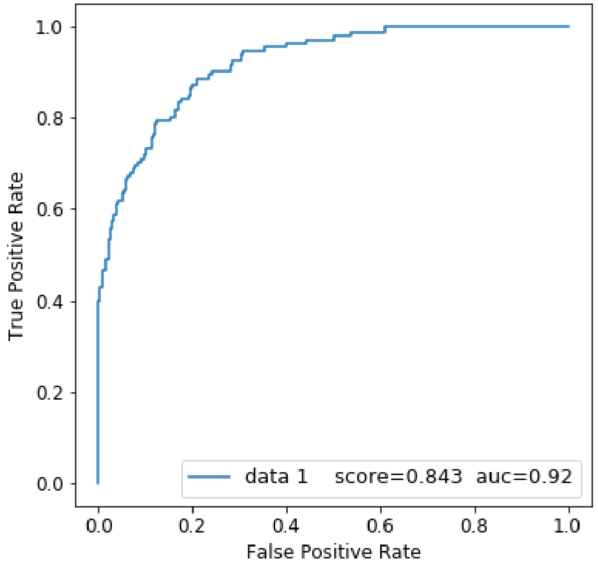


Initial results (cont'd)

Lasso: score = 0.827



Gradient Boosting: score = 0.843



The winner

- Gradient Boosting
  - Score: .843
  - AUC: .92

Model tuning

Tune hyperparameters using random parameter selection

```
{'bootstrap': [True, False],
 'max_depth': [5,10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [50, 100, 200, 300, 400, 600, 800, 1000, 1200, 1400, 1600,
1800, 2000]}

rf_random = RandomizedSearchCV(estimator = rf, param_distributions =
rf_random_grid,  n_iter = 100, cv = 3, verbose=2, random_state=42, n_jobs = -1)
```

Model tuning

Re-run with best_params_

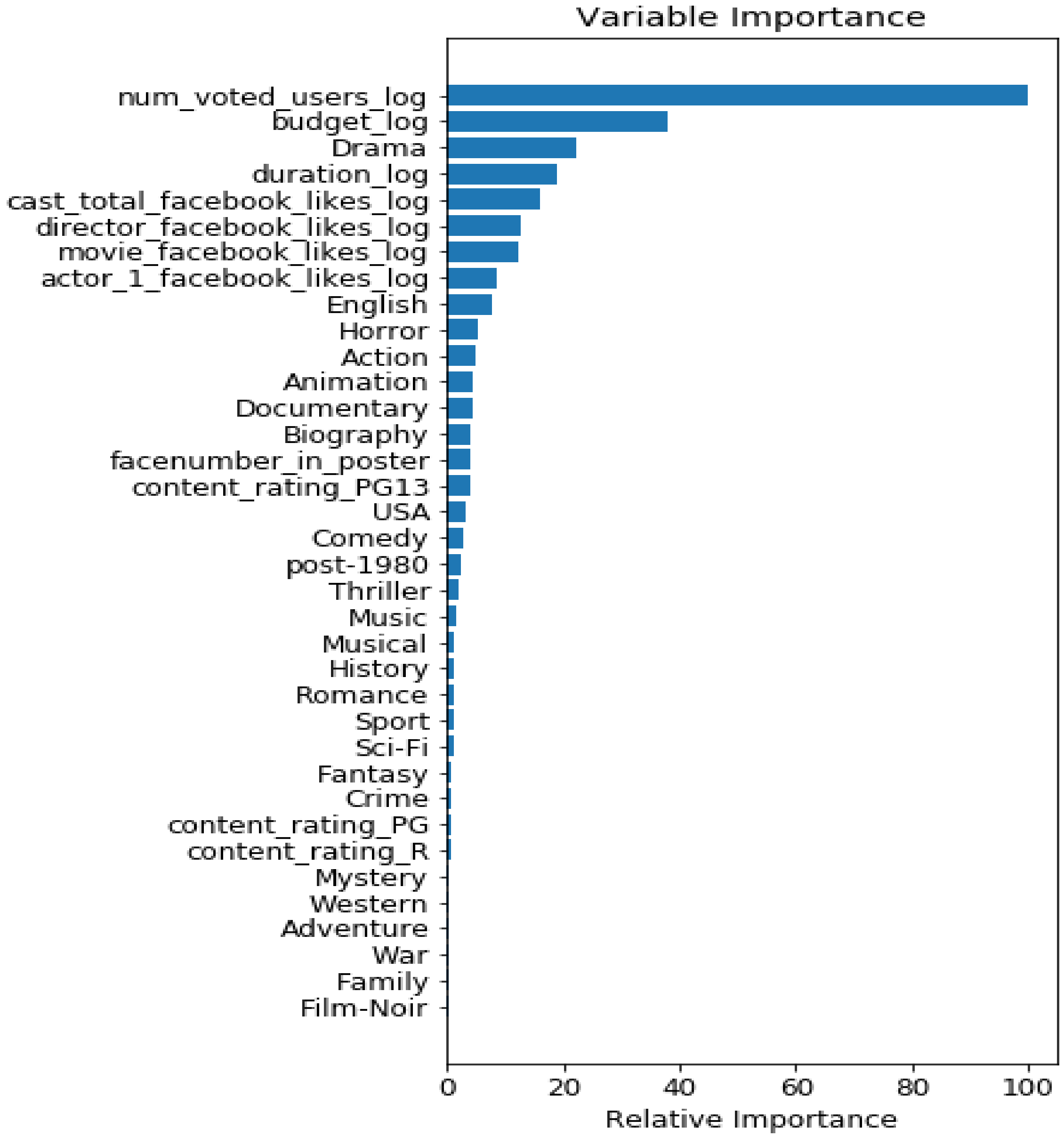
```
score = 0.854, auc = 0.922
```

```
In [39]: gb_random.best_params_

Out[39]: {'n_estimators': 800,
          'min_samples_split': 5,
          'min_samples_leaf': 2,
          'max_features': 'sqrt',
          'max_depth': 10,
          'loss': 'exponential'}
```

```
Mean cv score =  0.802
 [0.78378378 0.84864865 0.85675676 0.84054054 0.81351351 0.84864865
 0.78378378 0.73513514 0.76358696 0.75      ]
```

Important Features



Challenges/ Lessons Learned

- Not a tremendous amount of movies
  - But still got decent results
- Had to drop a lot of rows due to missing info
  - Lost about 23% of the data set
- There is definite bias in the dataset due to “Facebook likes”
  - Facebook has only been around 10 years and so will be heavily weighted to later movies
- You must familiarize yourself with every aspect of the data!
  - It’s very tempting to get the data set and jump into the analysis, but this can take you on a misleading path(s)
    - Understand “Quirks and peccadilloes”, as suggested in the curriculum
    - Took me awhile before I noticed the budget discrepancy

Thanks for coming to the show!