TMDB Box Office Prediction
Team Hollywood

Shiyang Cheng
Sam Cherinet
May, 2019

Background

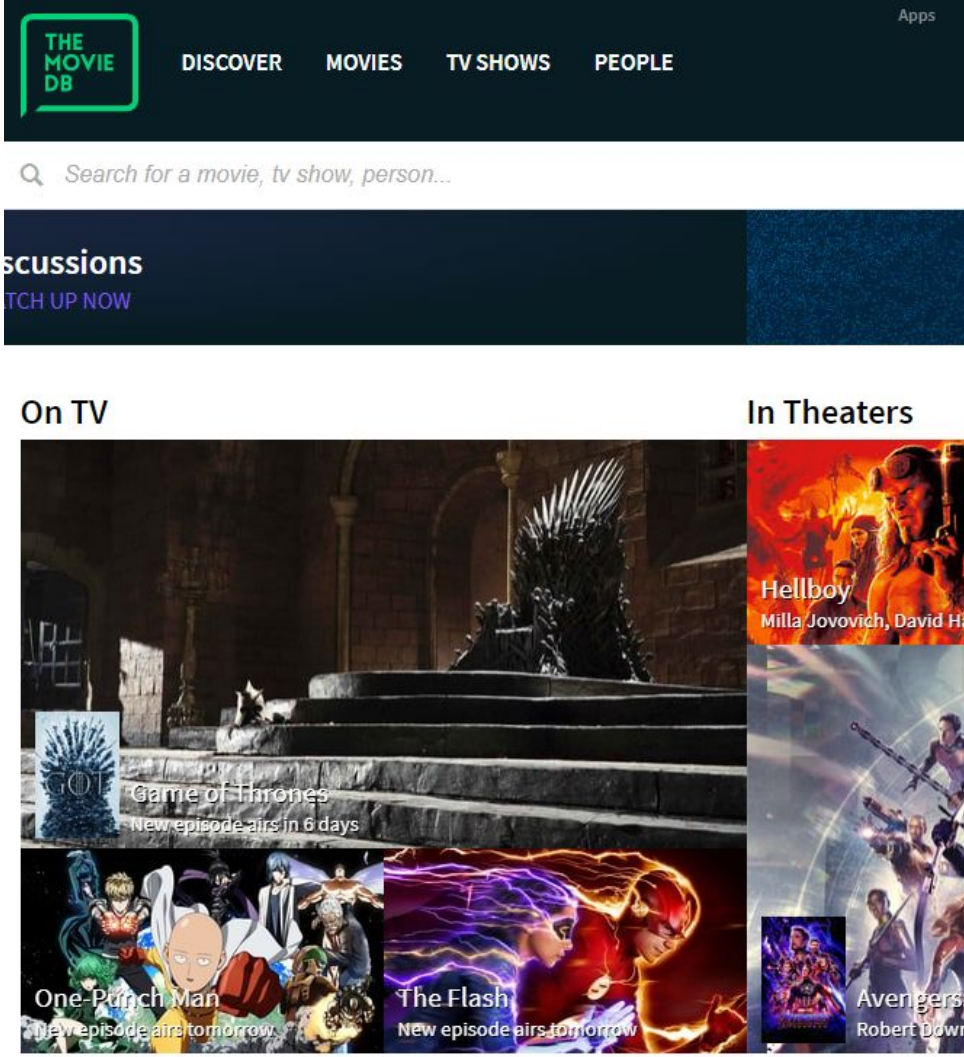
- $40 billion + dollar per year industry
- Usually large scale
- Will a movie make money?

TMDB Box Office Prediction

https://www.kaggle.com/c/tmdb-box-office-prediction

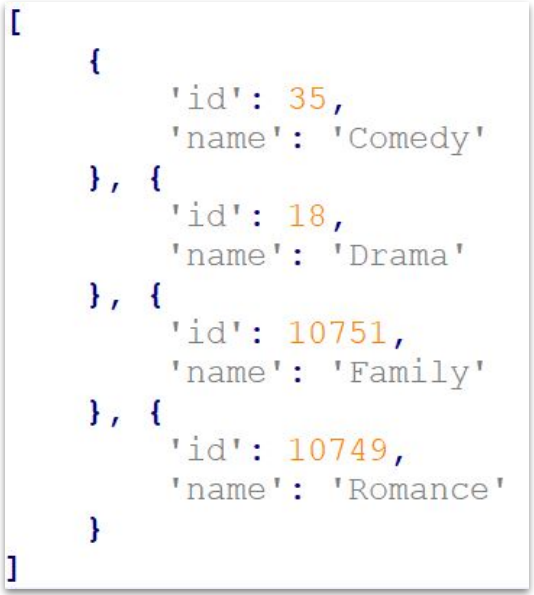
Data

- Data from the open TMDB
  - 3K movies
- 23 features out of the box
  - Title, release date, cast etc



Pre processing

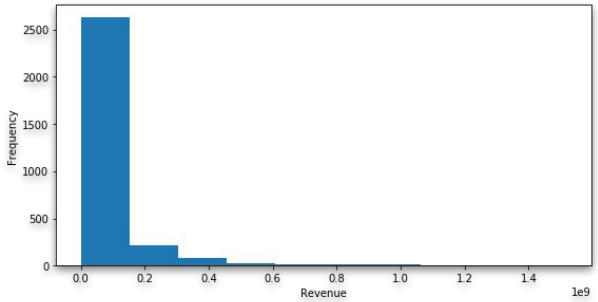
- JSON data flattening
  - belongs_to_collection
  - genres
  - production_companies
  - production_contries
  - spoken_languages
  - keywords
  - cast
  - crew

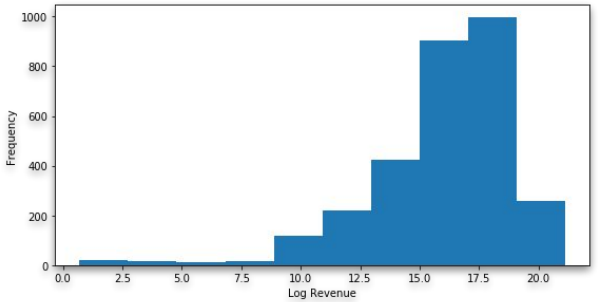


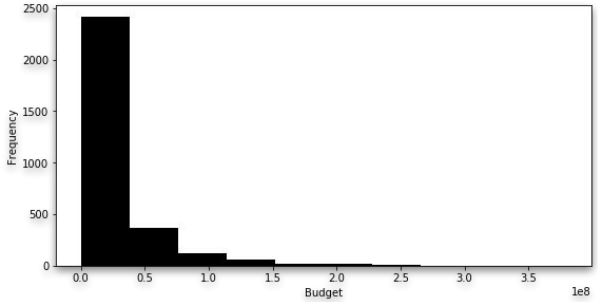
Comedy, Drama, Family, Romance

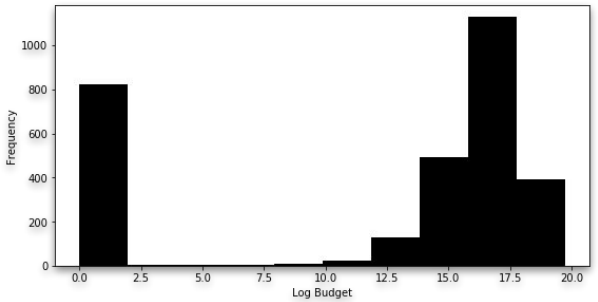
Skewed Data

- Skewed Data
  - Revenue
  - Budget



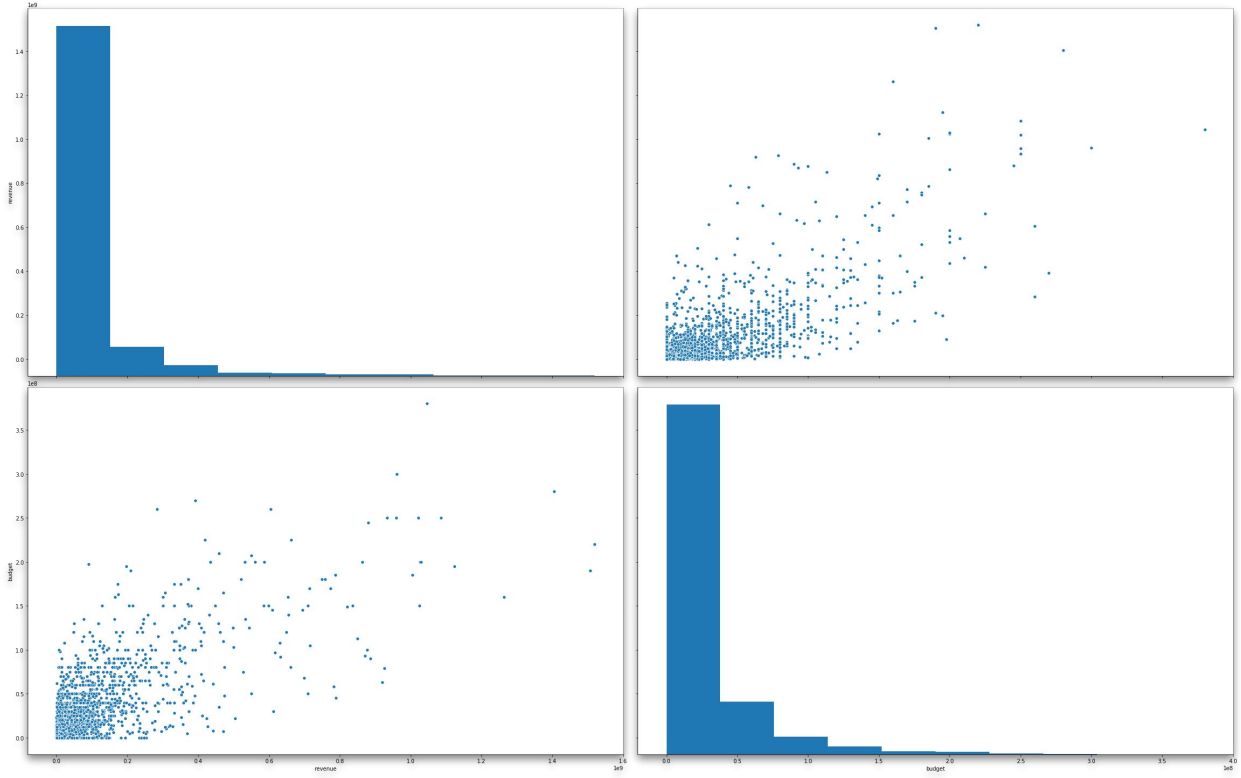




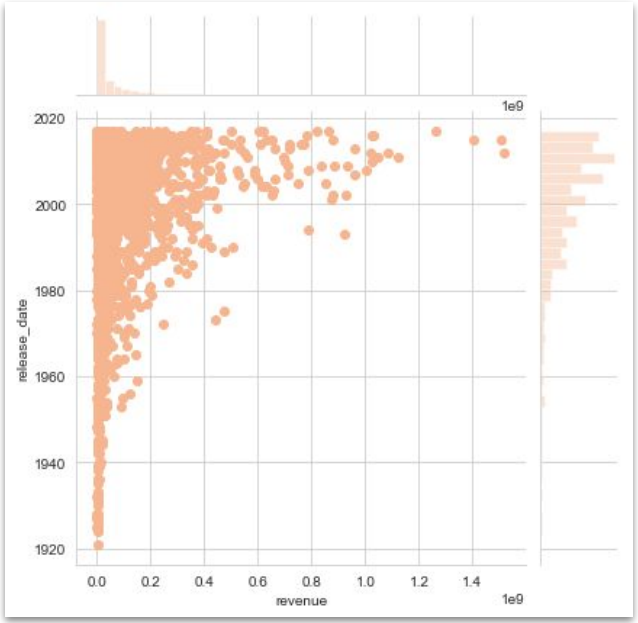


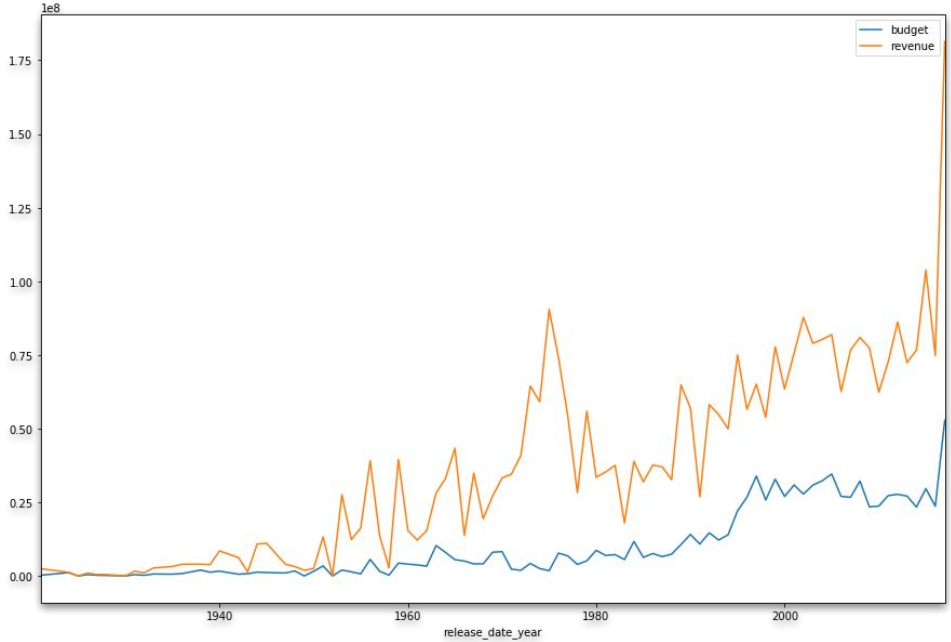
Data Exploration

Budget
Vs
Revenue

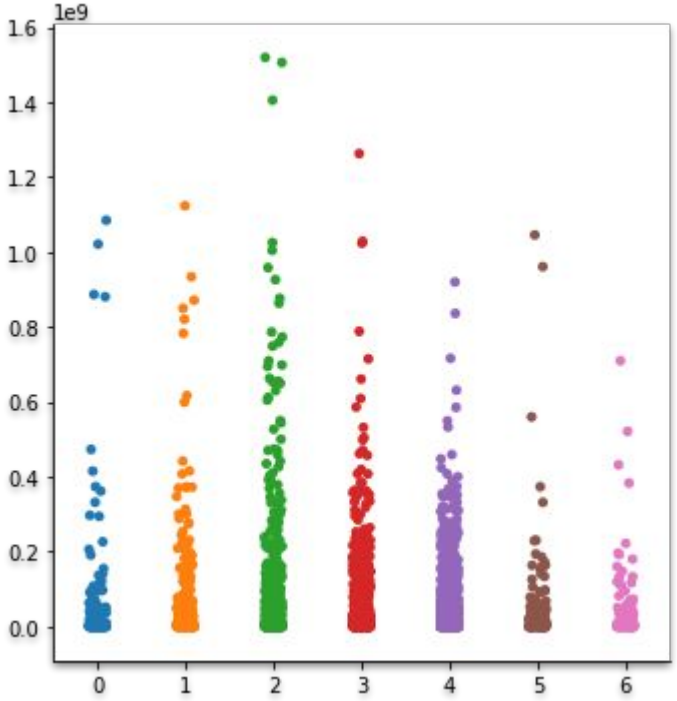


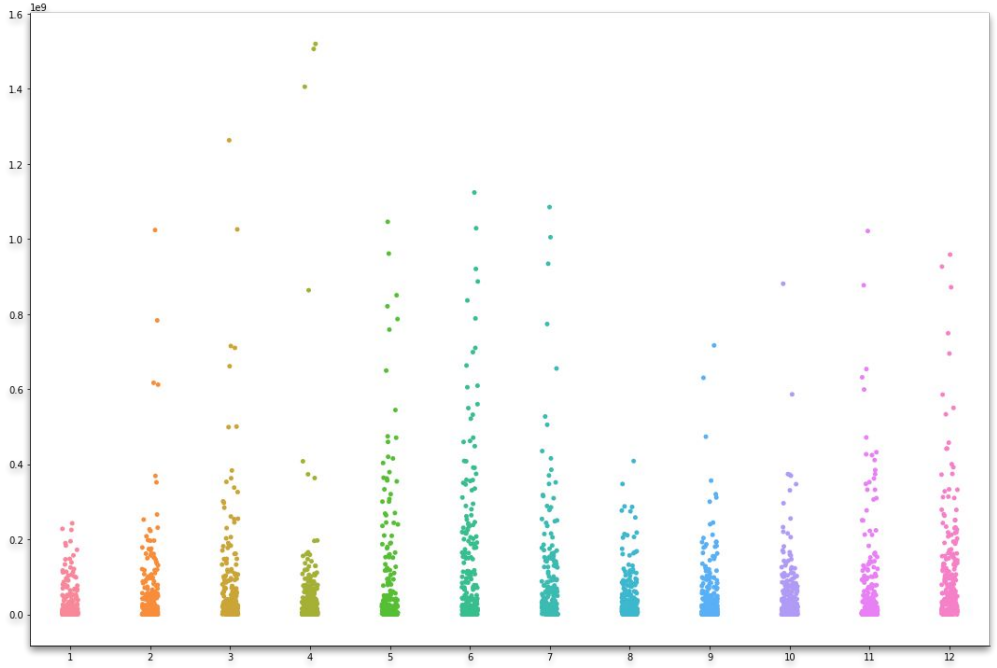
Release Year vs Budget & Revenue





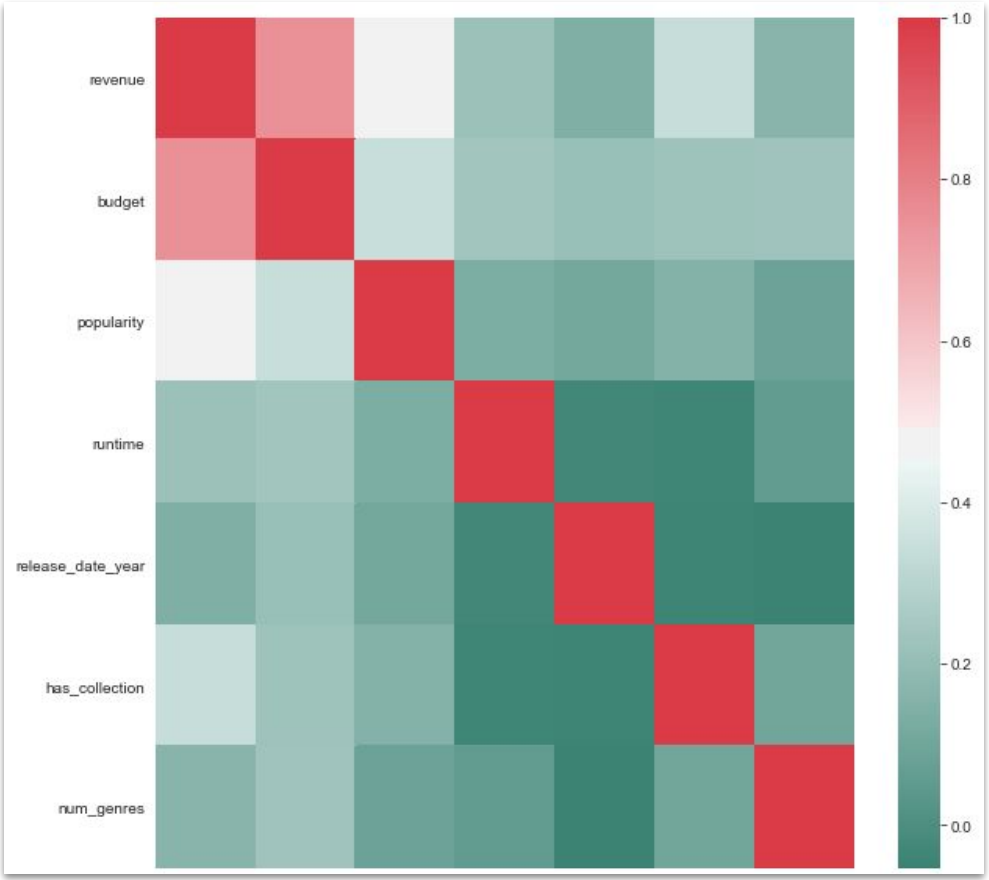
Release Month and Week of Day





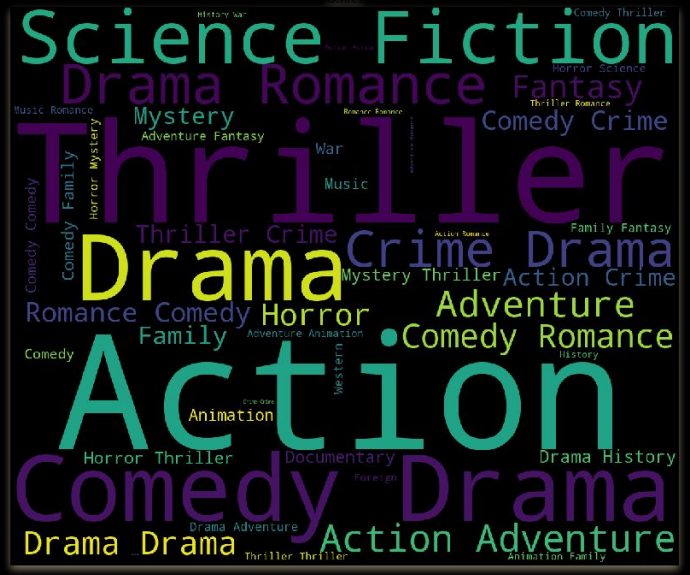
Notable Correlations

- Budget
- Popularity
- Sequel (has_collection)

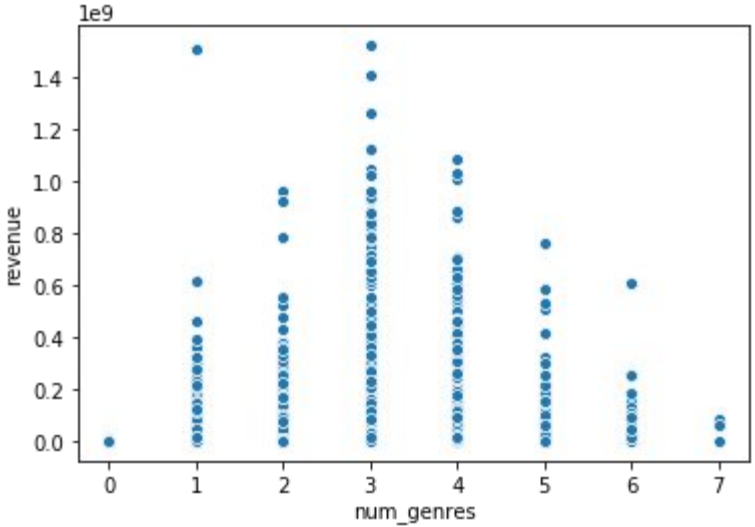


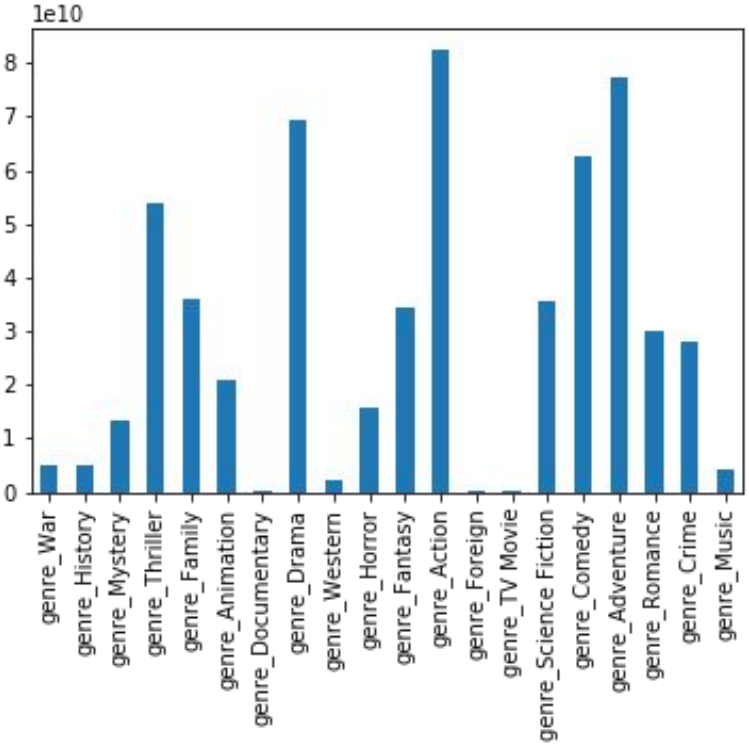
Genres (Word Clouds)

Used words clouds to visually see term prominence for text based features

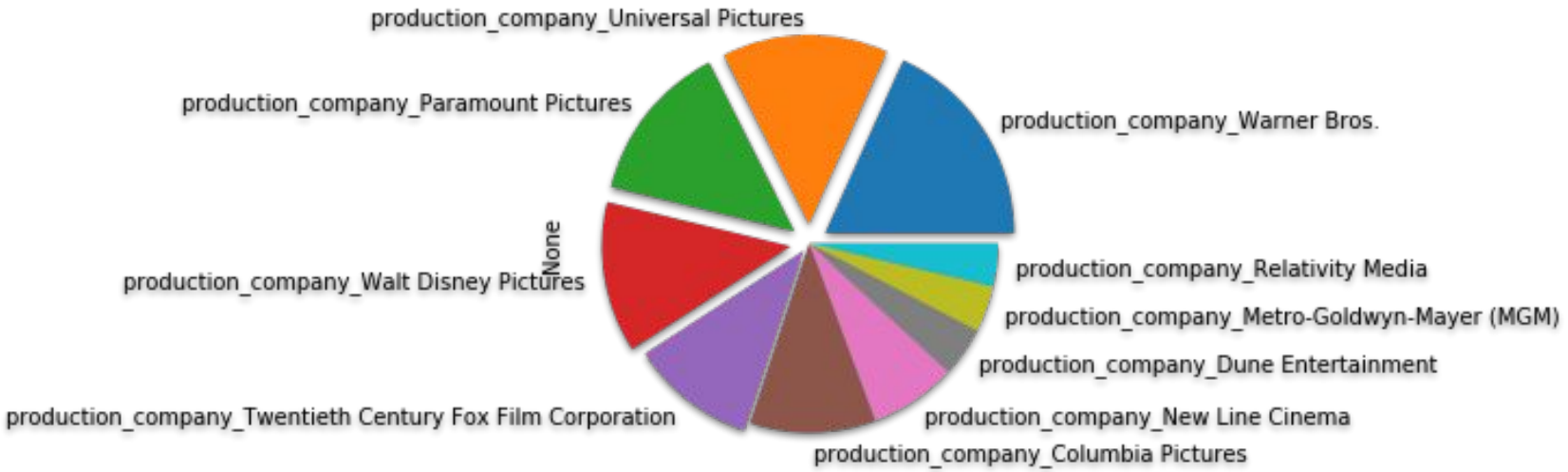


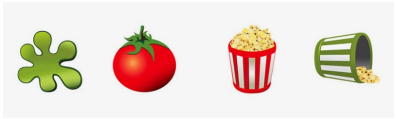
Genres vs Revenue





Production Companies - All Time Revenue (top 10)

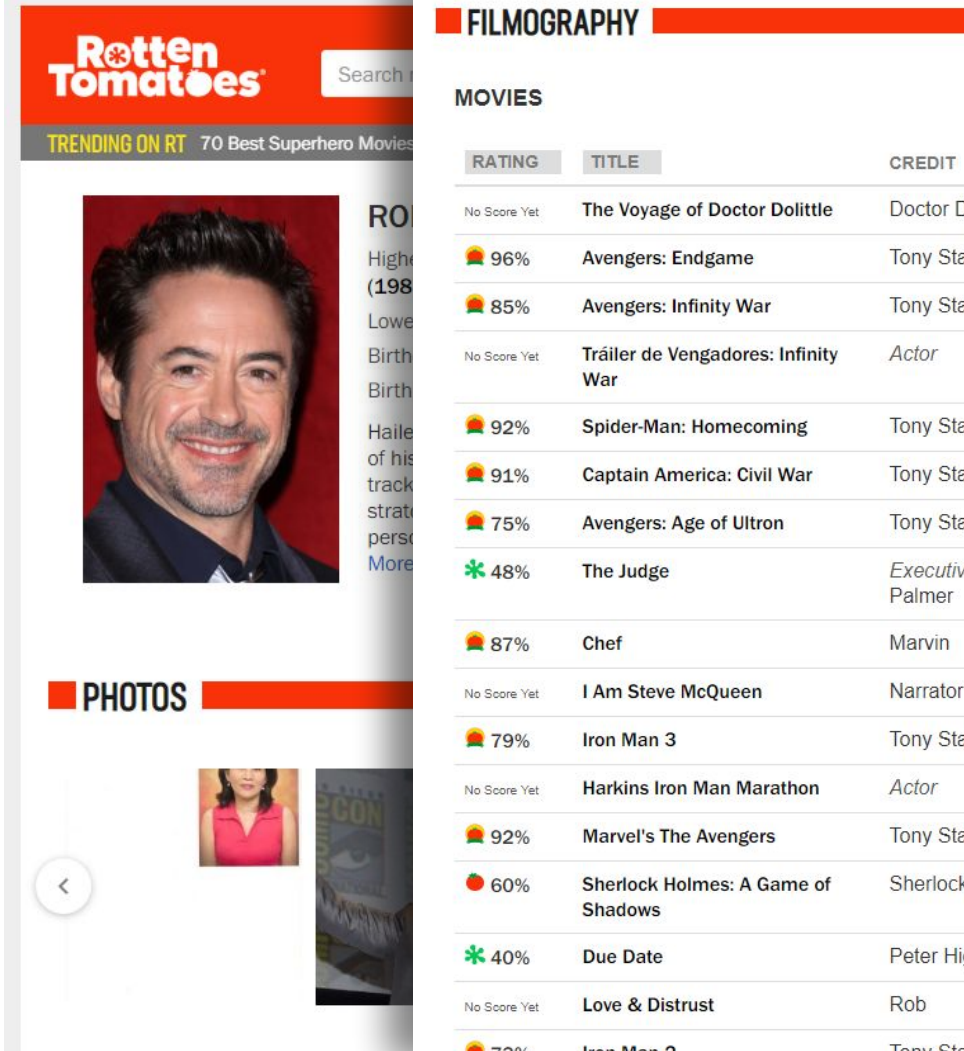




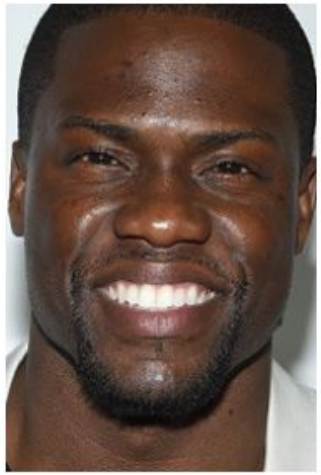
External Data RottenTomatoes

Director and Cast Scores

With little bit of BeautifulSoup and UrlLib



Sample Ratings



67.15

Kevin Hart

70.55

Lucy Liu

79.34

Jennifer Lawrence

91.6

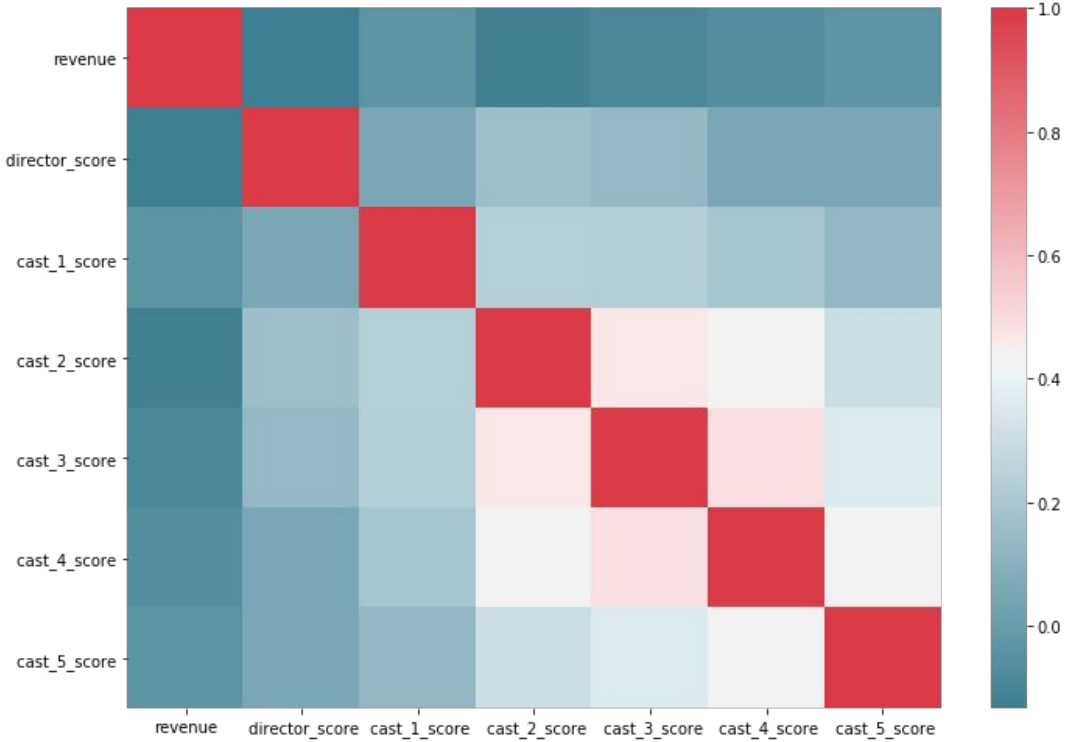
M Night Shyamalan

68.74

Robert Downey Jr

Revenue / Crew heatmap

- Cast Score has some impact on revenue
- Cast Scores for the first 5 does go together



Feature Engineering

- One Hot Encoding
  - Genres
  - Production Companies
  - Languages
- Create New Features
  - #genres
  - #production companies
  - #languages
  - TfidfVectorizer

Modeling Plan

Want to try:

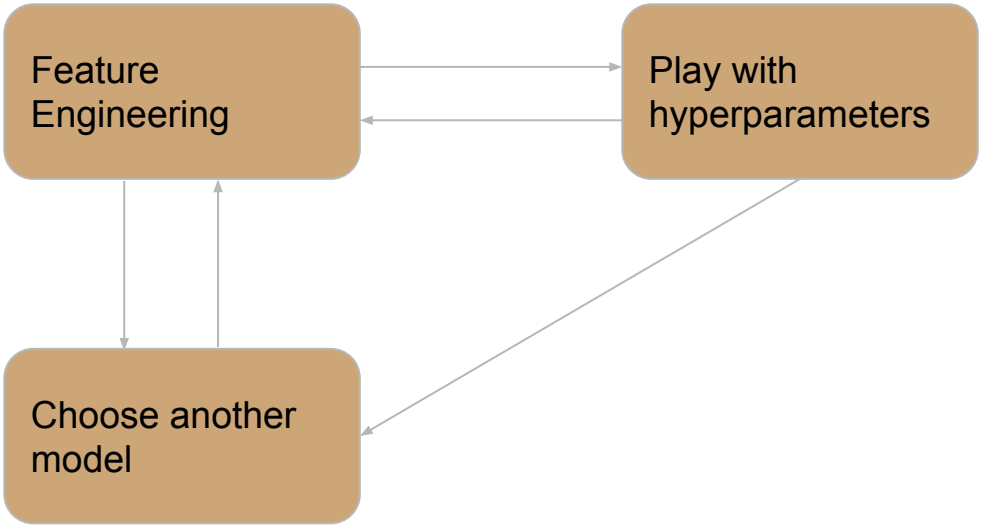
- Linear Regression
- Tree Based Model
  - Random Forest
  - Gradient Descent Boosting Tree
- Combined Models
- Deep Learning?
  - Poster?
  - Description?
  - All of the features?

Linear Regression

- No doing well
  - Data column is not linear but skewed
  - Target value cannot be negative in “negative mean squared log error”
  - Do not want to exhaustively try to generate all kinds of feature based on original features

Next one!

Expected Workflow (Experience from Kaggle Contest)



Feature Engineering

- Try to search other data (Rotten tomatoes gets reviews for crew and staff)
- Try to process poster as imagines using popular Deep Learning Algorithms?
  - Do you think the poster really made customer buying tickets? (I do not think so...)
  - Maybe trailer is much better
- Try to search all kinds of Natural Language Processing models
  - Seems a large part of implemented Algorithms is used to extract the topic from text
  - Other seems to generate text
  - Can we start with something simple? -> TF-IDF vector

Model Selection - Hyperparameter

- Grid Search
  - Prepare all manuals for each model you would like to try
  - Prepare the hyperparameter array for each model
  - Cross validation

Tree based Models

- Random forest
  - Random forest is not linear model
  - It works well on different kind of data
  - It gives you the feature importance
- Gradient Descent Boosting Tree
  - Non-linear model
  - Well implemented
  - Good reputation (From Kaggle contest and other kernels posted on Kaggle)
- Try to combine them together
  - Using feature stacking from result of Random Forest and XGBoosting

Random Forest

- It is OK and it give us 2.14
- It give feature importance, which could reduce overfitting in XGBoost

Gradient Descent Boosting - XGBoost

- With all features it give 2.07
- Using the most important features from random forest, we get 2.05 with top 60 features

What's next?

- Try to convert some features to other format, such as convert release_date to month_of_year, day_of_year, epoch time
- Try to combine some important features such as, budget, popularity, director average review

Does Additional Feature Engineering Work

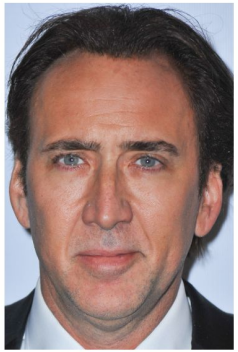
- It made random forest given score as 0.7 in training data, but still higher than 2.05 in testing data
- It made XGBoost given 1.1 in training data, but still higher than 2.05 in testing data
- Are we only overfitting the training set?

What Else?

- We try to use less features based on feature importance
- No such grid search lib (At least, I did not find one)
- Run it manually
- Still cannot beat 2.05

Results

- Our best is 2.05 which is 346 in 928 teams
- It may relate to the domain knowledge to create extra features
  - IMDB (Maybe other's model just proof that IMDB gets more accurate data)
  - Movie industry
- Misleading data
  - Time does matter, review score is not stable



NICOLAS CAGE

Highest Rated: 100% Love, Antosha (2019)

Lowest Rated: 0% A Thousand Words (2012)

Birthday: Jan 7, 1964

Birthplace: Long Beach, California

Actor Nicolas Cage has always strived to make a name for himself based on his work, rather than on his lineage. As the nephew of filmmaker Francis Ford Coppola, Cage altered his last name to avoid accusations of nepotism. (He chose "Cage" both out of admiration for avant-garde musician John Cage and en homage to comic book hero... More

61%	**Vampire's Kiss**	Peter Loew	—	1989
No Score Yet	**Never on Tuesday**	Man in Red Sports Car (uncredited)	—	1988
93%	**Moonstruck**	Ronny Cammareri	—	1987
91%	**Raising Arizona**	H.I. McDonnough	—	1987
85%	**Peggy Sue Got Married**	Charlie Bodell	—	1986
No Score Yet	**The Boy in Blue**	Ned Hanlan	—	1986
85%	**Birdy**	Al Columbato	—	1984
75%	**The Cotton Club**	Vincent Dwyer	—	1984
60%	**Racing With the Moon**	Nicky / Bud	—	1984
70%	**Rumble Fish**	Smokey	—	1983
82%	**Valley Girl**	Randy	—	1983
78%	**Fast Times At Ridgemont High**	Brad's Bud	—	1982

Future Works

- Collect more information
- Domain knowledge
- Try to filtered out the outliers
- Try to use more NLP technologies
- Maybe the review of staff and crew also related to some specific time period
- Maybe different country, get totally different patterns
- Maybe posters do made people buy tickets
- Economy - Inflation

Lesson Learned

- Feature Engineering is important - based on how you use them
- Domain knowledge is important - produce more meaningful features
- Try to gather more meaningful information
- Keep trying a lot and do remember to stop and thinking

Thanks!