Predicting Movie Revenue Using Machine Learning

Keyi Jiang, Freya Wang, Rita Xu

CHICAGO BOOTH

Introduction

Motivation

- Growing Movie Market
- Revenue cases which violate our common sense
- A profitable business prospect to find a consistent predicting formula

Central question

- What would be the determining factors for a movie to succeed

Approach

- Apply text sentiment analysis on text variables
- Use variable selection methods to select important variables
- Explore machine learning methods to train different models

Dataset

Combination of four data base

- IMDB
- TMDB
- The Numbers database
- Kaggle

overview	popularity	production_companies
In the 22nd century, a paraplegic Marine is dispatche...	150.437577	[{"name": "Ingenious Film Partners", "id": 289}, {"name..
Captain Barbossa, long believed to be dead, has come...	139.082615	[{"name": "Walt Disney Pictures", "id": 2}, {"name": "Jerr..
A cryptic message from Bond's past sends him on a tr...	107.376788	[{"name": "Columbia Pictures", "id": 5}, {"name": "Danja..
Following the death of District Attorney Harvey Dent, ...	112.312950	[{"name": "Legendary Pictures", "id": 923}, {"name": "W..
John Carter is a war-weary, former military captain w...	43.926995	[{"name": "Walt Disney Pictures", "id": 2}]

```
'data.frame':   4944 obs. of
 $ color
 $ director_name
 $ duration
 $ director_facebook_likes
 $ actor_3_facebook_likes
 $ actor_2_name
...
 $ actor_1_facebook_likes
 $ genres
"Action|Adventure|Thriller"
 $ actor_1_name
 $ title
knight rises" ...
 $ cast_total_facebook_likes
 $ actor_3_name
```

IMBD and TMBD are two mainstream movie rating and review websites. These two datasets contain multiple facets of information about the top 5000 movies such as budget, issue company, crew information, genre, overview and reviews, etc.

The **Numbers** database and Kaggle data can be the supplements for the `revenue` variable.

Data Processing

- Remove useless variables & variables that we don't know before the movie is open for consumers
- standardize the variable format in four datasets
- Categorize `content_rating`
- Separate month of release
- Merge the two main data and use extra data to impute the missing revenue
- Compare and modify columns with similar meanings
- Split the text variables into separate words

Data Processing-Title standardization

Star Wars: Episode VII – The Force Awakens

X-Men: The Last Stand

→

star wars episode vii the force awakens

xmen the last stand

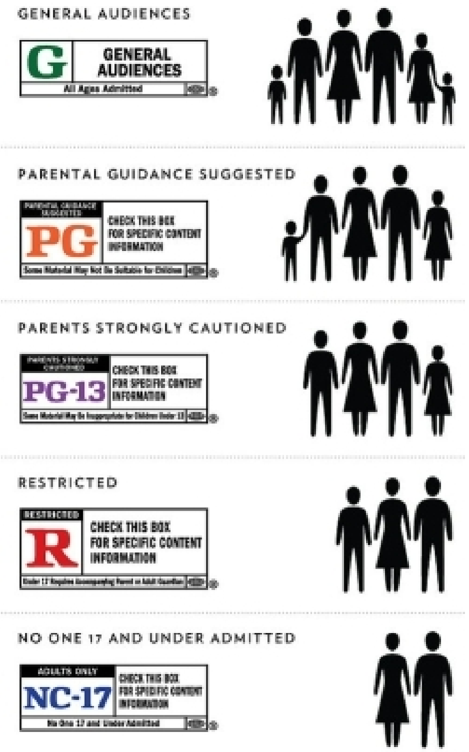
- Standardized the title as unique IDs
- join different datasets by *title* and *title_year*
    - to avoid movies with the same title
    - E.g. *The Three Musketeers(2011) &*
      The Three Musketeers(1993)

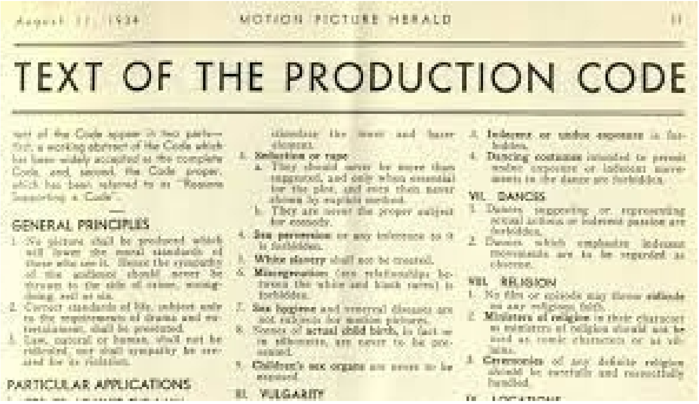
Data Processing-Categorize *`content_rating`*

- We are using dataset range from 1929 to 2016
- Movies industries adopted different criteria in the past decades

```
unique(df$content_rating)
```

```
"PG-13"       "R"          "G"
"Approved"    "Unrated"    "TV-G"
"M"           "TV-PG"      "NC-17"
"Not Rated"   ""           "PG"
"X"           "Passed"     "GP"
"TV-14"
```




MOTION PICTURE HERALD

TEXT OF THE PRODUCTION CODE

GENERAL PRINCIPLES

PARTICULAR APPLICATIONS

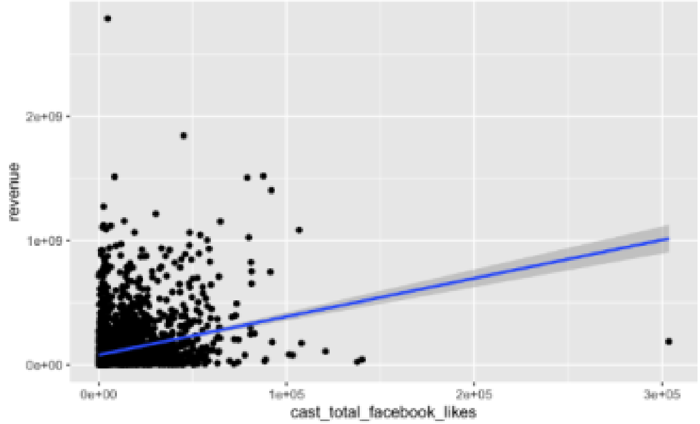
Hay’s Code

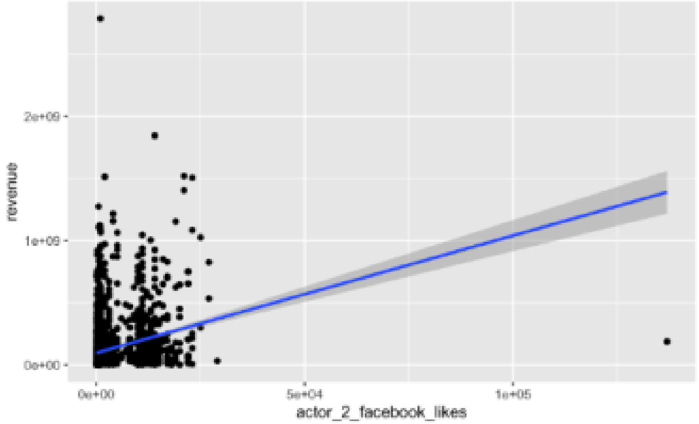
MOTION PICTURE ASSOCIATION OF AMERICA

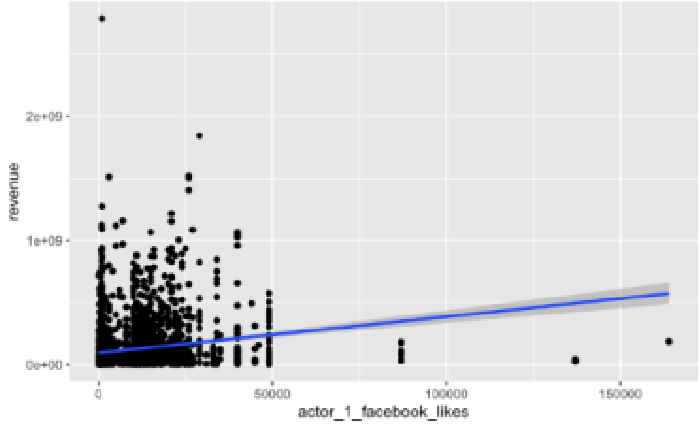
MPAA

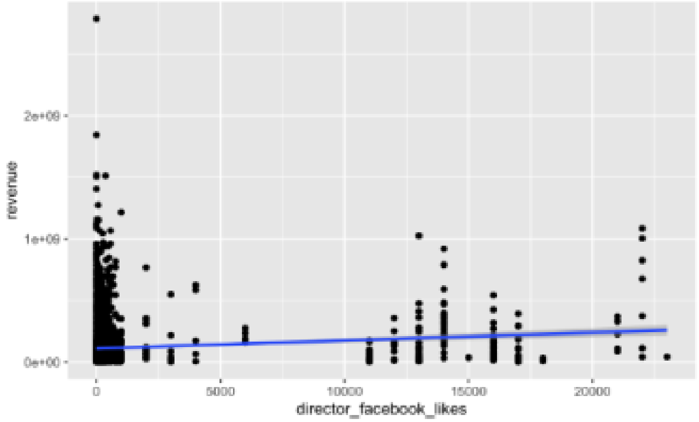
CHICAGO BOOTH

Data Processing- Data Explore







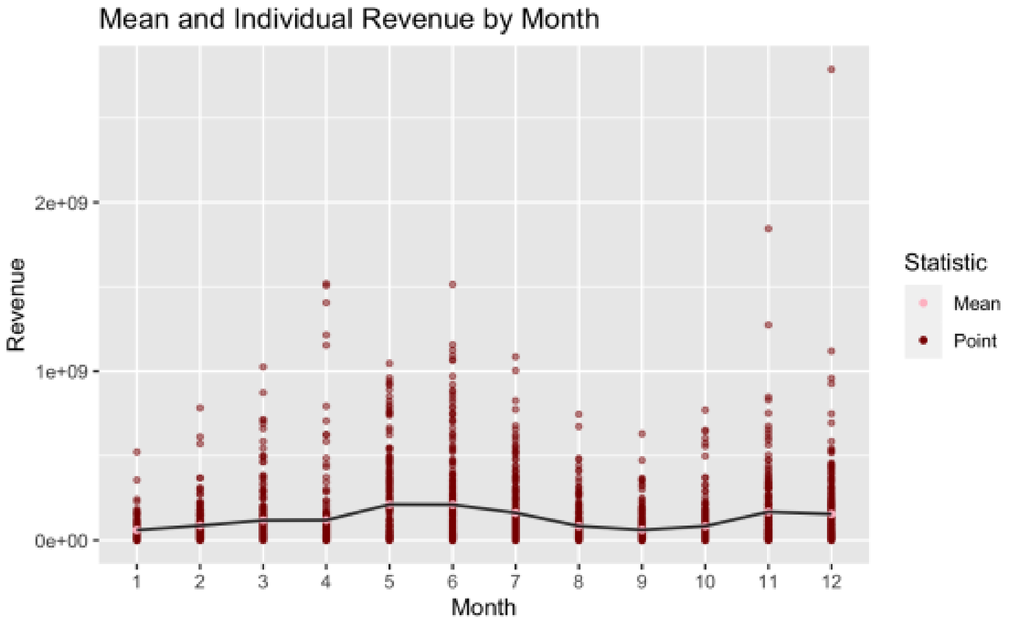


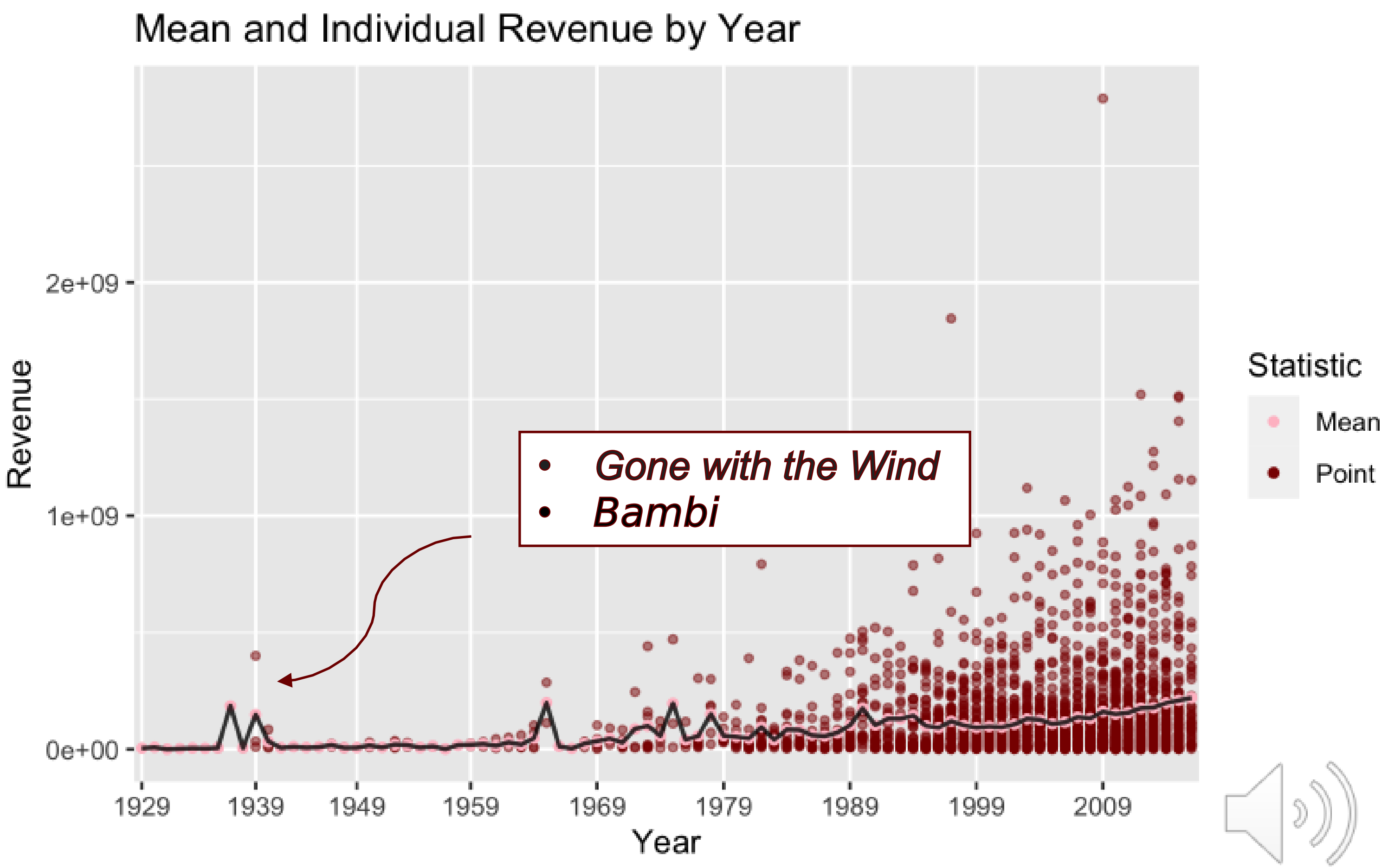
- Positive relation between Facebook like and revenue.
- No longer holds true when coming to director

- Highly depends on company and budget
- Need further examination

Data Processing- Data Explore

- Average revenue increases through out the year
- Summer and holiday are peak seasons for cinema
    - Abnormal peaks due to re-release

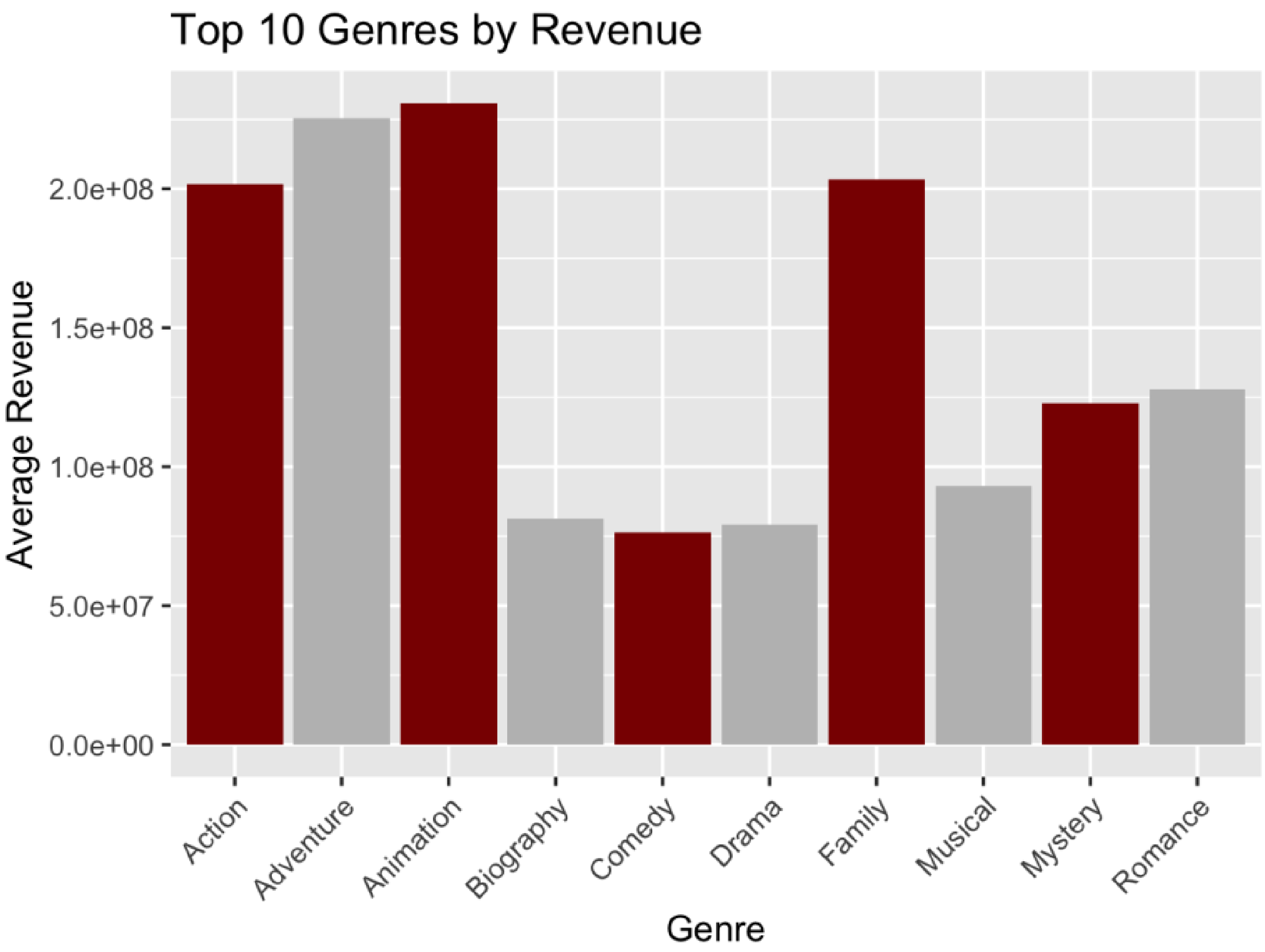




CHICAGO BOOTH

Data Processing- Data Explore

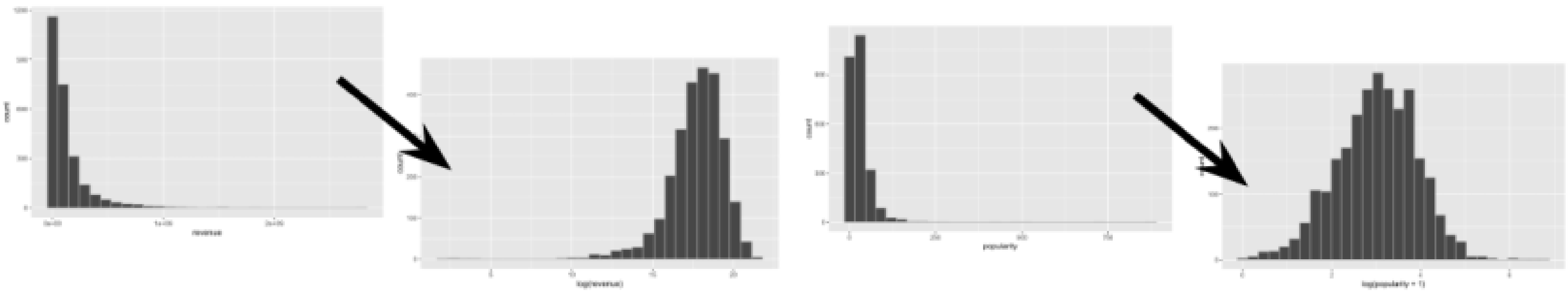
- Average Revenue by Genre
  - Animation
  - Adventure
  - Family
  - Action
- Most Shot Genre
  - Comedy 716
  - Action 698
  - Drama 410
  - Adventure 269



Data Processing-transformation

- Log transformation

Heavily skewed → better distributed



- Home page transformation

Hyperlink → dummy indicator

homepage
http://www.avatarmovie.com/
http://disney.go.com/disneypictures/pirates/
http://www.sonypictures.com/movies/spectre/
http://www.thedarkknightrises.com/
http://movies.disney.com/john-carter

homepage
0
0
1
0
0
0

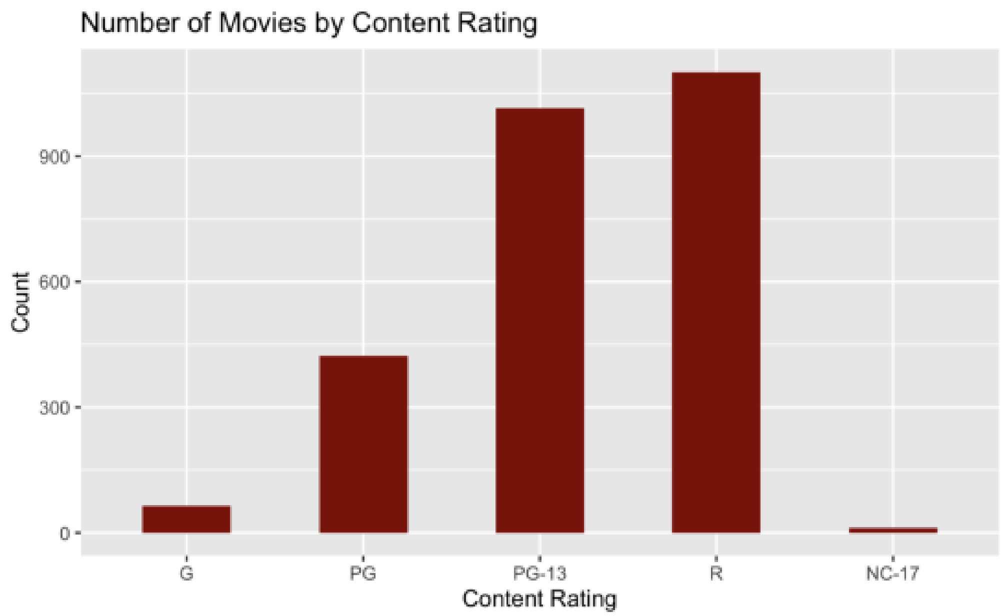
- Create *'main_genre'*
- Create '*main_company*'
- Split text information
  - Length of letter
  - Length of word
  - For **Text Analysis**

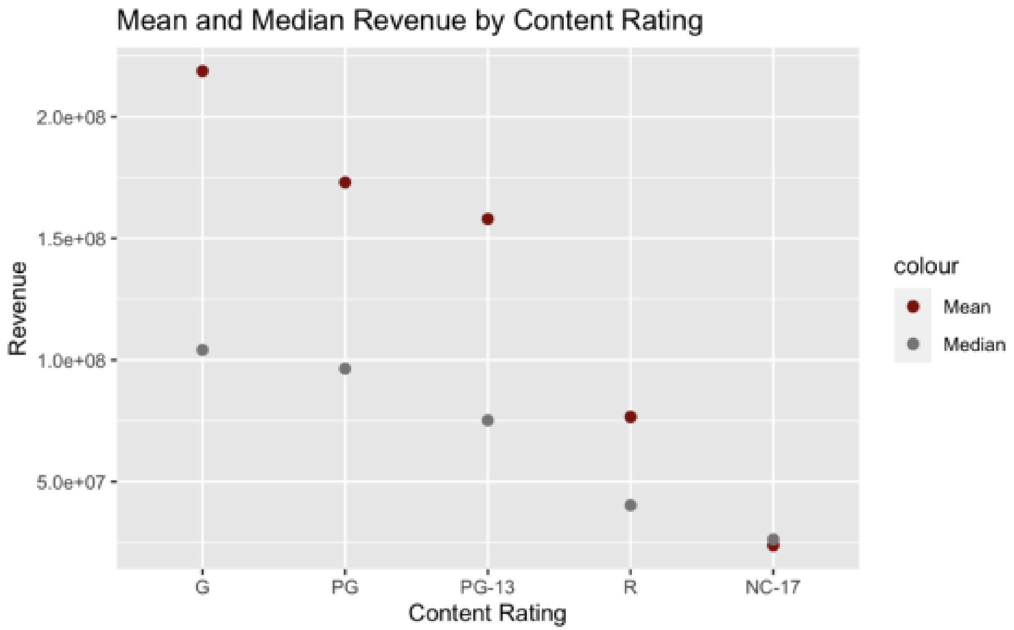
- Keep only US movies
- Deal with *NA*
  - OLS imputation
  - Mean
  - Manually add

Data Processing- Data Explore

Content Rating

- “R” level counts the largest proportion of movies
- “PG-13” ranked second
- “G” which means all age audience level generates the highest mean and median revenue

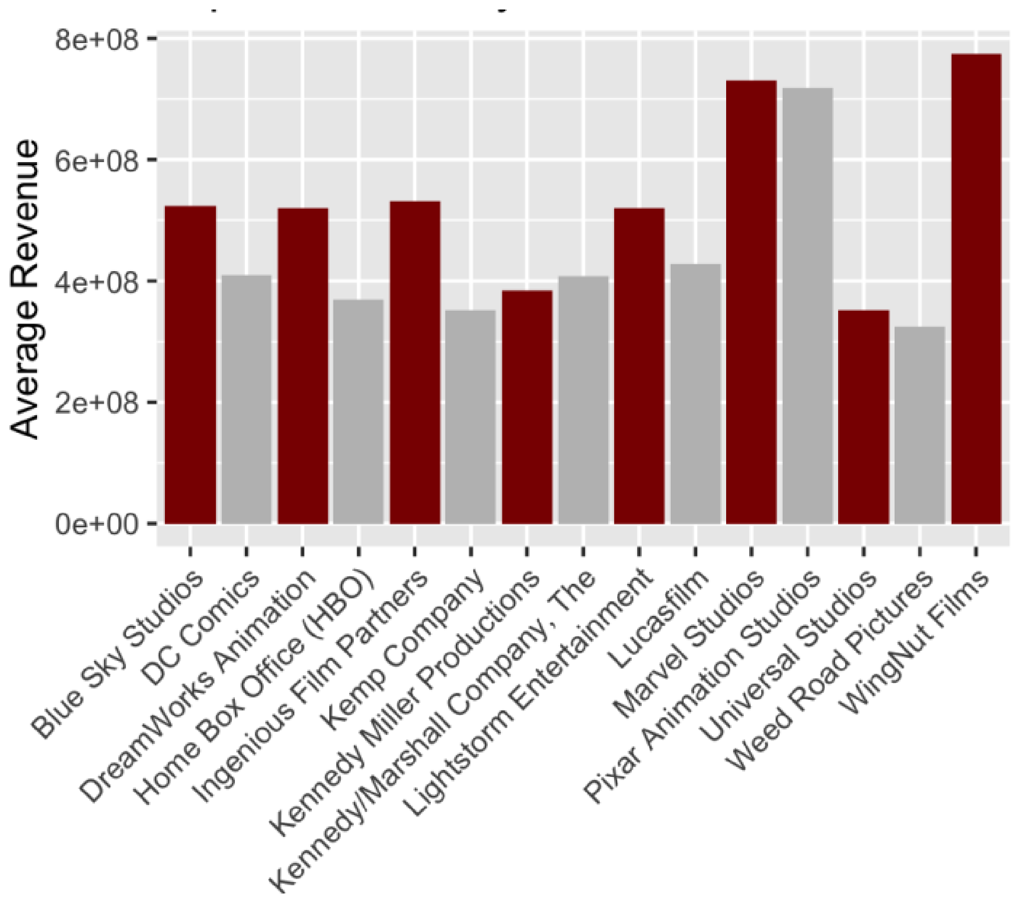




Data Processing- Data Explore

- Companies dominantly determine the budget and lots of unobserved factors for a movie.

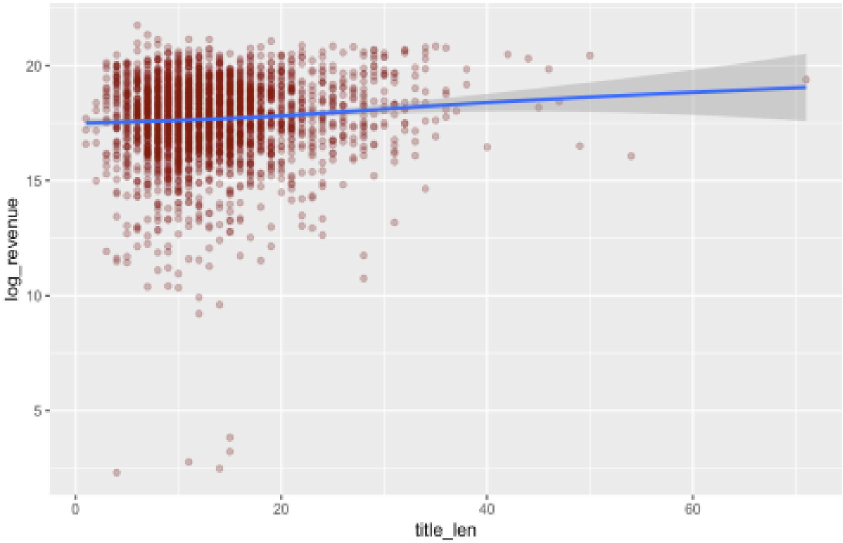
- Most famous Company may not genterate the largest average revenue:
    - *Warner Bros.* — ranks over 50th
    - *WingNut Films* tops the ranking
        - *Hobbit*
        - *The Lord of the Ring*

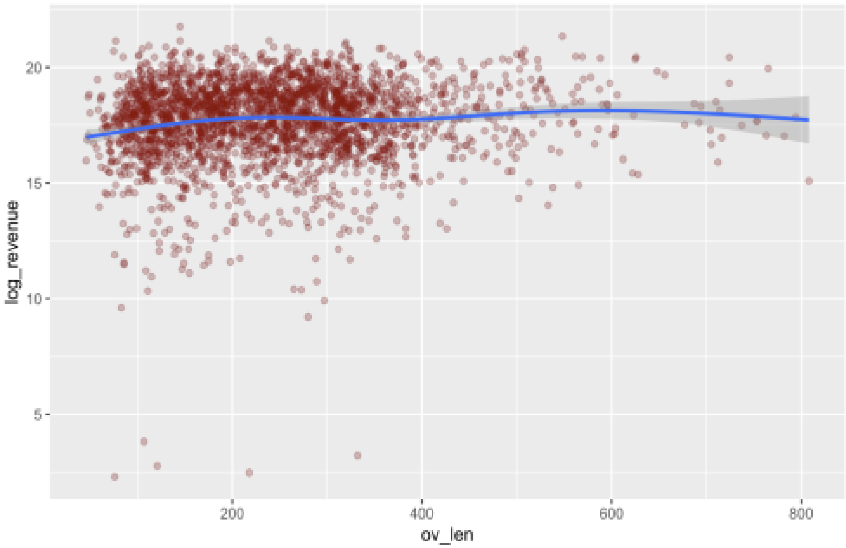


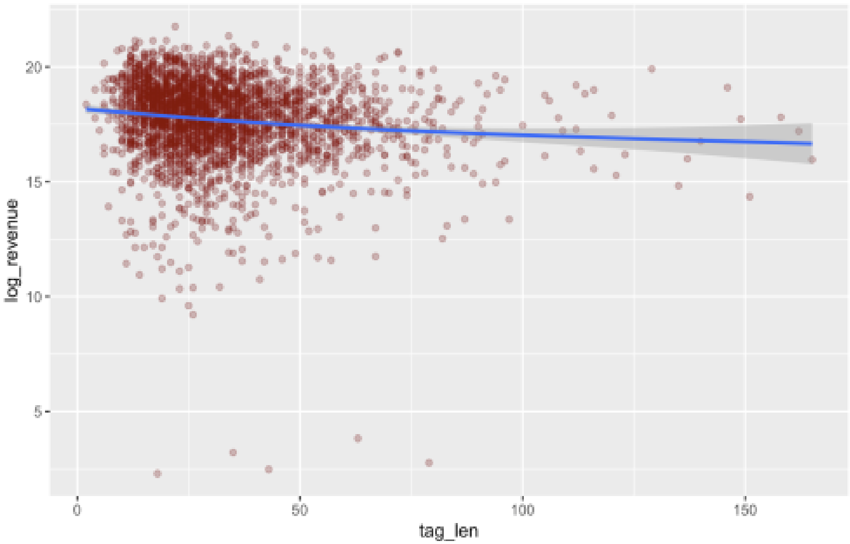
Data Processing- Data Explore

Length of the character variables

- The length of “title” has a slightly positive relationship with log_revenue
- The length of “overview” doesn’t have a clear linnear relationship with log_revenue
- The length of “tag” has a slightly negative relationship with log_revenue



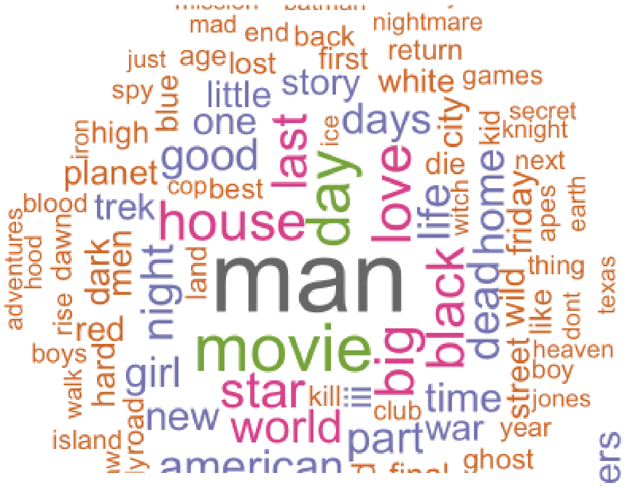




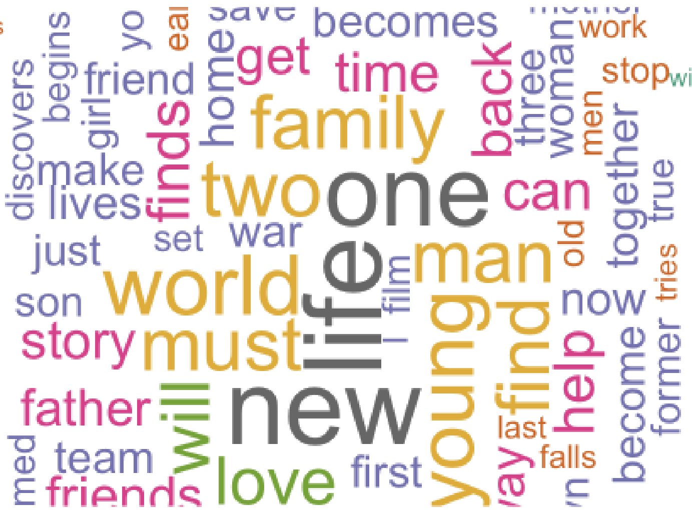
CHICAGO BOOTH

Text Analysis: Word Cloud

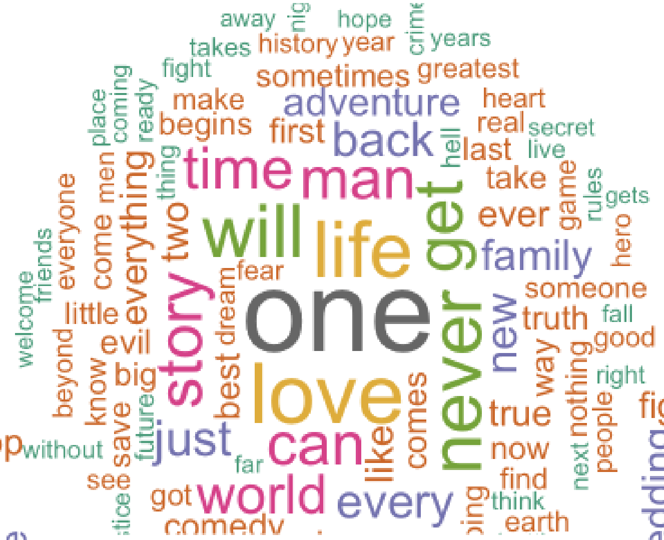
We found the most frequent words and drew the word cloud plots for 'title', 'overview', 'tagline', 'keywords'.



Title Frequency



Overview Frequency



Tag Frequency

Keywords Frequency

CHICAGO BOOTH

Text Analysis: Sentiment Analysis

We generated sentiment score variables for 'title', 'overview', 'tagline' and 'keywords' in the data frame.

A sample table below shows the summary statistics for the sentiment scores of 'title' variable.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.75000	0.00000	0.00000	-0.02573	0.00000	2.30000

However, we found the sentiment analysis inefficient at some points.

And one potential operation is TF-IDF Analysis. We can try to use the weights gained from TF-IDF to identify the most important and connotative words in the overview and then combine them with the sentiment score we captured in the previous section, so that we can avoid giving every word the same weight.

Text Analysis-Potential Improvement

- **Problem**: Sentiment is **not efficient** for overview, though it works well with review and twitter.

> 84 years later, a 101-year-old woman named Rose DeWitt Bukater tells the story to her granddaughter Lizzy Calvert, Brock Lovett, Lewis Bodine, Bobby Buell and Anatoly Mikailavich on the Keldysh about her life set in April 10th 1912, on a ship called Titanic when **young** Rose boards the departing ship with the **upper-class** passengers and her mother, Ruth DeWitt Bukater, and her **fiancé,** Caledon Hockley. Meanwhile, a drifter and artist named Jack Dawson and his best friend Fabrizio De Rossi **win** third-class tickets to the ship in a game. And she explains the whole story from departure until the **death** of Titanic on its first and last voyage April 15th, 1912 at 2:20 in the morning.

- Overview of **Titanic**
- Score: 1.8
- Positive Sentiment

- Only process single word, without interpretation
- Each word **weighs** the same in the final score calculation

Text Analysis-TF-IDF

- Term frequency-inverse document frequency:
- calculates a score based on frequency in the text and the inverse frequency
- Better capture the connotation of the text than single words analysis
- **Score = Weight*Sentiment_score**

> And she explains the whole story from departure until the **death** of Titanic on its first and last voyage April 15th, 1912 at 2:20 in the morning.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

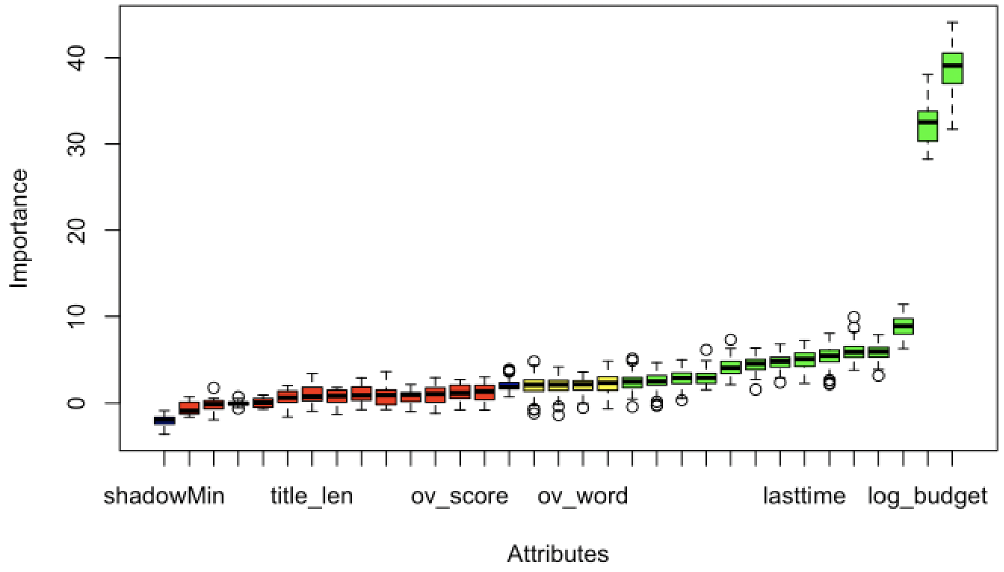
$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Feature Engineering

- Feature selection:
  - **Boruta**
  - Forward Stepwise
  - Random Forest

- *'log_popu'* and '*log_budget*' has the largest importance in Boruta selection

"title_year"
"actor_1_facebook_likes"
"content_rating"
"lasttime"
"tag_len"
"director_facebook_likes"
"main_genre"
"actor_2_facebook_likes"
"log_budget"
"tag_word"
"actor_3_facebook_likes"
"cast_total_facebook_likes
"homepage"
"log_popu"



Feature Engineering

- Feature selection:
  - Boruta
  - **Forward Stepwise**
  - Random Forest

- Only selected four variables
- Fit a line for each '*content_rating*' category

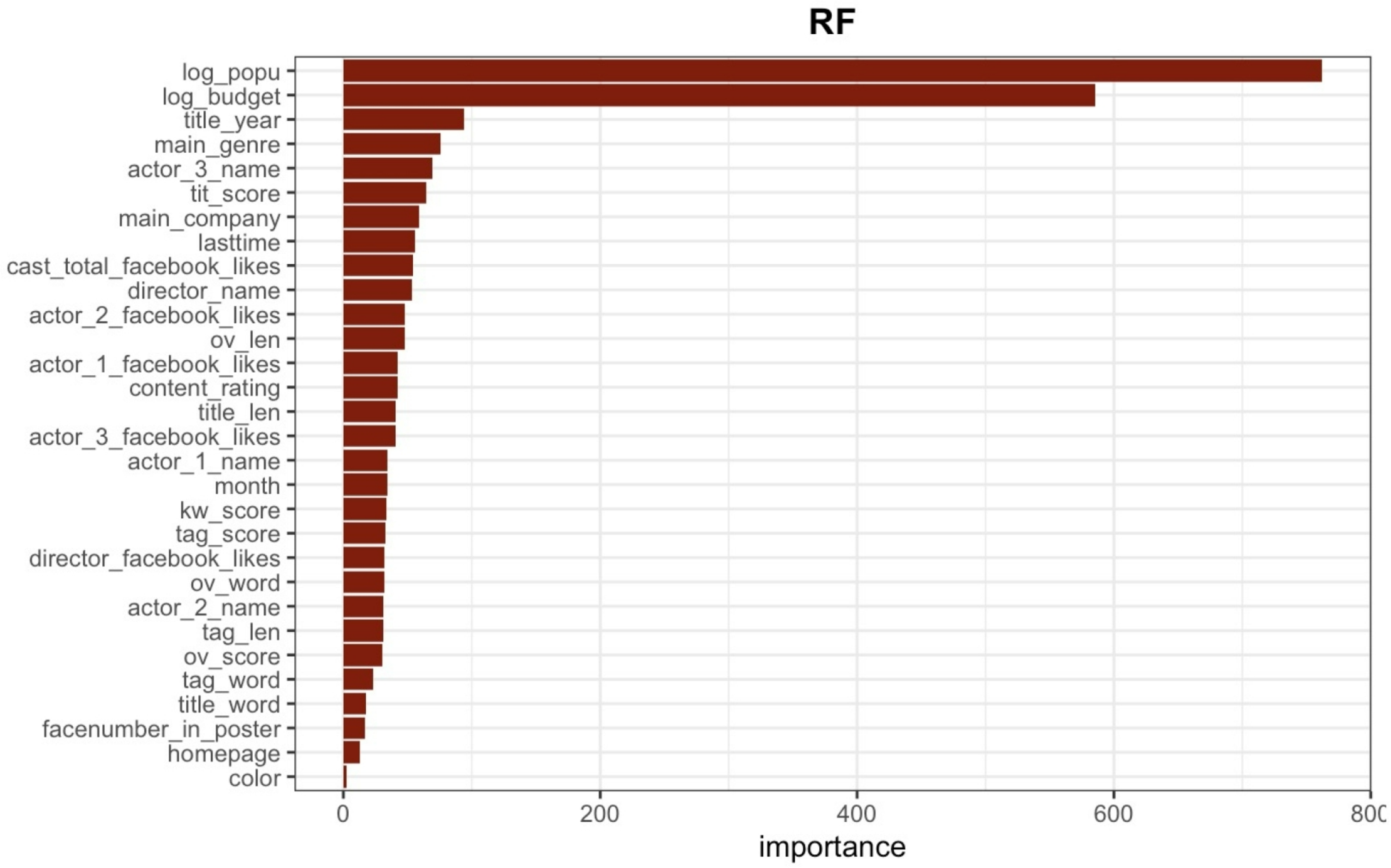
	log_revenue		
Predictors	*Estimates*	*CI*	*p*
(Intercept)	33.85	23.55 – 44.14	**<0.001**
log popu	0.95	0.88 – 1.02	**<0.001**
log budget	0.52	0.46 – 0.57	**<0.001**
content_ratingNC-17	-0.84	-1.78 – 0.11	0.082
content rating [PG]	-0.04	-0.43 – 0.35	0.856
content_ratingPG-13	-0.31	-0.68 – 0.07	0.110
content rating [R]	-0.60	-0.98 – -0.22	**0.002**
title year	-0.01	-0.02 – -0.01	**<0.001**
Observations	1568		
R^2	0.561		

- May not be suitable for models with too many factorized variables,
- Names of 2000+ directors,
- Computationally expensive
- Overfitting.

Feature Engineering

- Feature selection:
  - Boruta
  - Forward Stepwise
  - **Random Forest**

- Need further examination
- Compare RF with Boruta



Modeling-Random Forest

- Built 3 random forest models
  - use all variables
  - variables selected by Boruta
  - variables selected by variable importance

Models	Variable Selection Method	RMSE on train	RMSE on test
Random Forest	None	1.1684	1.1568
Random Forest	Boruta	1.1807	1.1333
Random Forest	Variable Importance	1.1785	1.1560
Linear Model	Forward Stepwise+BIC		1.1647

- Use out-of-sample RMSE to tuning
- Predicted on the testing set

- Random Forest with Boruta feature selection performs best

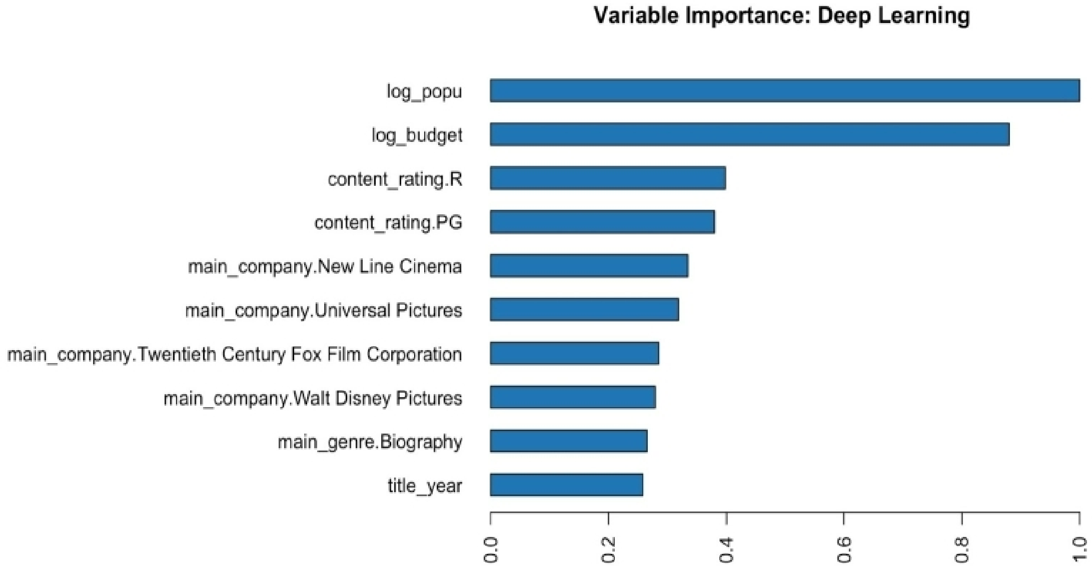
```
### rf+b selection
```{r}
rf.fit.b <- ranger(
    formula         = log_revenue ~ .,
    data            = train.red_dim,
    num.trees       = 500,
    mtry            = 10,
    min.node.size   = 5,
    sample.fraction = .8,
    importance      = 'impurity',
    seed = 1108
    )
```

```
### rf+rf selection
```{r}
rf.fit.rf <- ranger(
    formula         = log_revenue ~ .,
    data            = train.rf,
    num.trees       = 500,
    mtry            = 4,
    min.node.size   = 4,
    sample.fraction = .8,
    importance      = 'impurity',
    seed = 1108
    )
```

Modeling-Neural Networks

- Tried Deep Learning models with 2, 3, and 4 layers and found 2-layer with 50 features per layer achieved the best performanceuse
- Extracted deep features and used them to further build a random forest model and a boosting model
- The boosting model using deep features performs best; Neural networks performs better than regular machine learning models.
- **Business insights**:10 most important variables to predict movie revenue

Models	RMSE on train	RMSE on CV	RMSE on test
Deep Learning	1.076885	1.162771	1.138346
Random Forest with deep features	1.10308	1.062118	1.134001
Boosting with deep features	0.8060875	1.043528	1.115288



Conclusion and Implication for Business