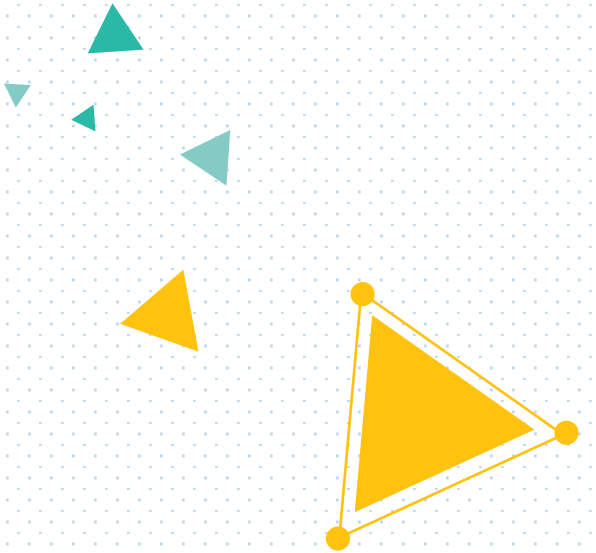
Movie Popularity Prediction

Group 8

Mengqi CHEN, Sitong CHEN, Xiyao CHEN, Xiaochen FAN,
Jike FANG, Ning FU, Tao HAN, Sijie JIN, Xianghong LUO,
Yan QIN, Yuhao TIE, Che XU, Siyang XUE, Yixiao ZHANG

CONTENTS

01 Part One *Overview*



Overview

- Purpose: Predict on movies' ratings.
- Model:
  - Linear Model
  - Nonlinear Model
  - Neural Network

Overview – Raw Data

- Over 4000 movies, 21 Variables

ID:

- id - The movie_id

Scalar Variable:

- budget - The budget in which the movie was made
- revenue - The worldwide revenue generated by the movie
- runtime - The running time of the movie in minutes
- vote_average - average ratings the movie received
- vote_count - the count of votes recieved

Other Variable:

- homepage - A link to the homepage of the movie
- keywords - The keywords or tags related to the movie
- overview - A brief description of the movie
- tagline - Movie's tagline
- title - Title of the movie

Categorical Variable

- genre - The genre of the movie, Action, Comedy, Thriller etc
- original_language - The language in which the movie was made
- original_title - The title of the movie before translation or adaptation
- spoken_language - The language in which the movie was released
- production_companies - The production house of the movie
- production_countries - The country in which it was produced
- release_date - The date on which it was released
- status - "Released" or "Rumored"
- cast – cast of the movie
- crew – crew of the movie

Overview – Raw Data

- 4802 movies, 21 Variables

ID:

- id - The movie_id

Scalar Variable:

- budget - The budget in which the movie was made
- revenue - The worldwide revenue generated by the movie
- **runtime - The running time of the movie in minutes**
- **vote_average - average ratings the movie received**
- **vote_count - the count of votes recieved**

Other Variable:

- homepage - A link to the homepage of the movie
- keywords - The keywords or tags related to the movie
- overview - A brief description of the movie
- tagline - Movie's tagline
- title - Title of the movie

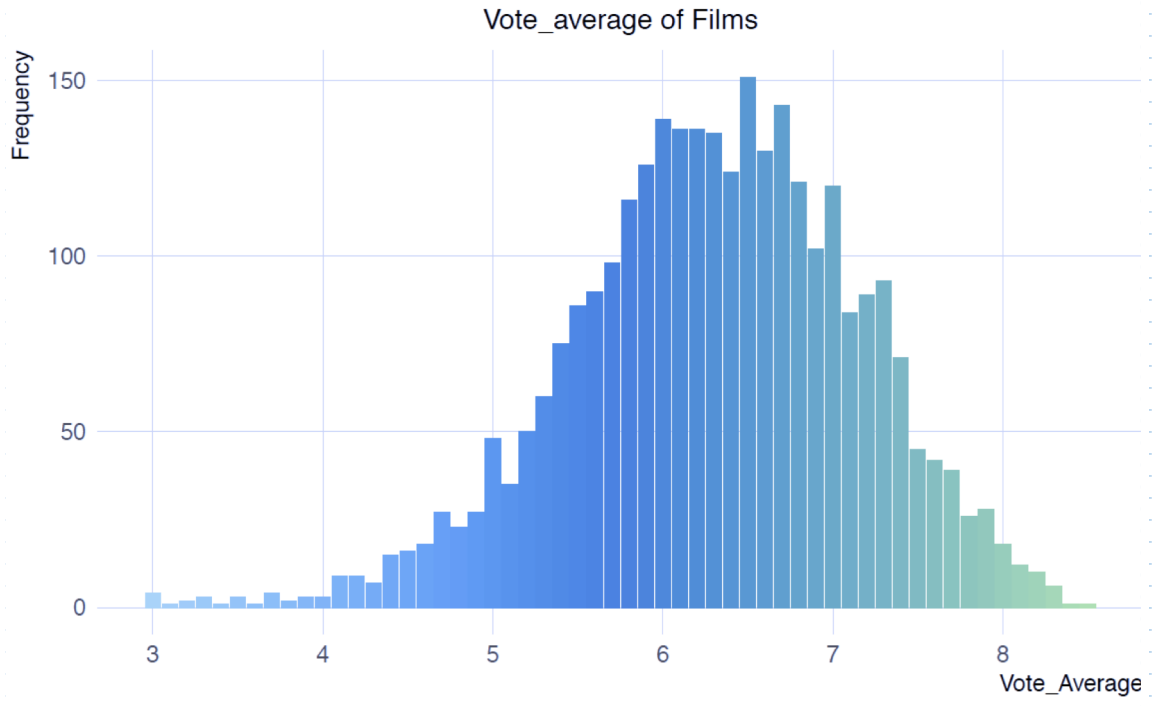
Categorical Variable

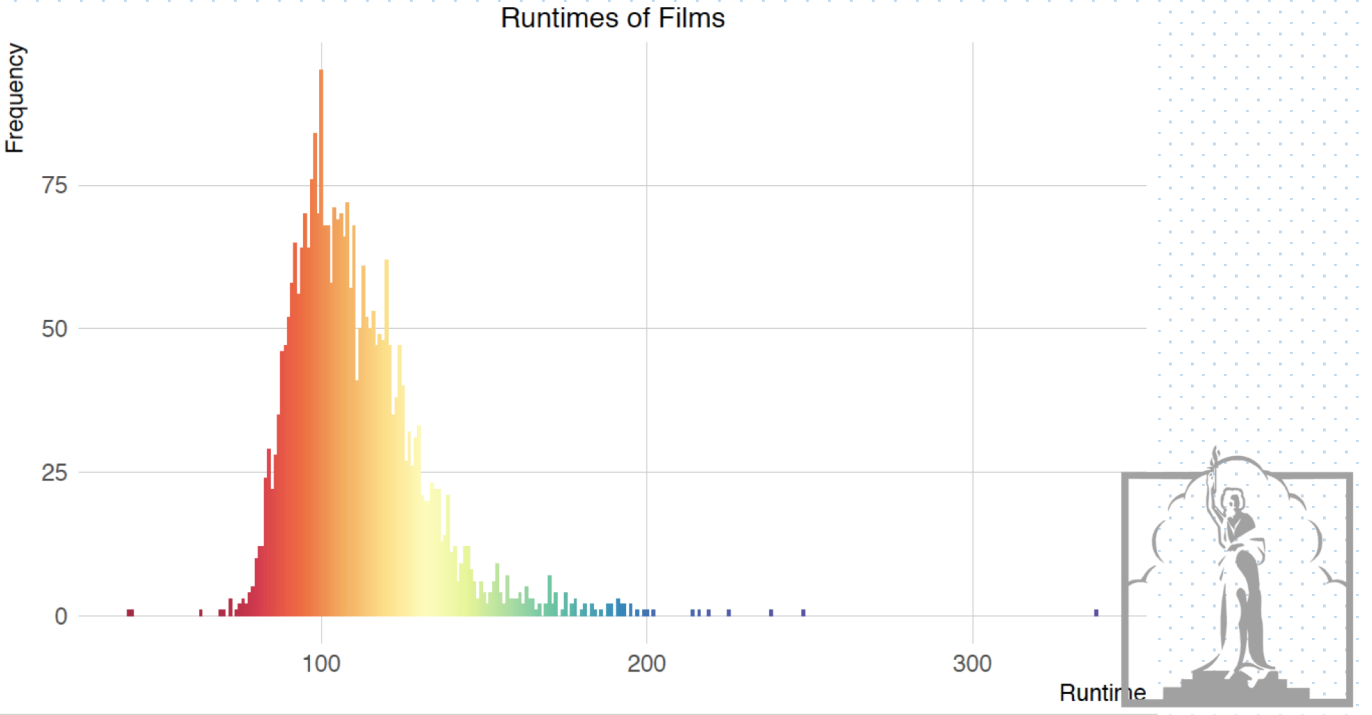
- **genre - The genre of the movie, Action, Comedy, Thriller etc**
- original_language - The language in which the movie was made
- original_title - The title of the movie before translation or adaptation
- **spoken_language - The language in which the movie was released**
- **production_companies - The production house of the movie**
- production_countries - The country in which it was produced
- **release_date - The date on which it was released**
- status - "Released" or "Rumored"
- cast – cast of the movie
- crew – crew of the movie

02 Part Two
Exploratory Data Analysis

EDA – Vote & Runtime

- **Vote_average: vote_count/number of people who liked**
- **Vote_count: the number of likes for this movie**
- **Runtime: the running time of the movie in minutes**

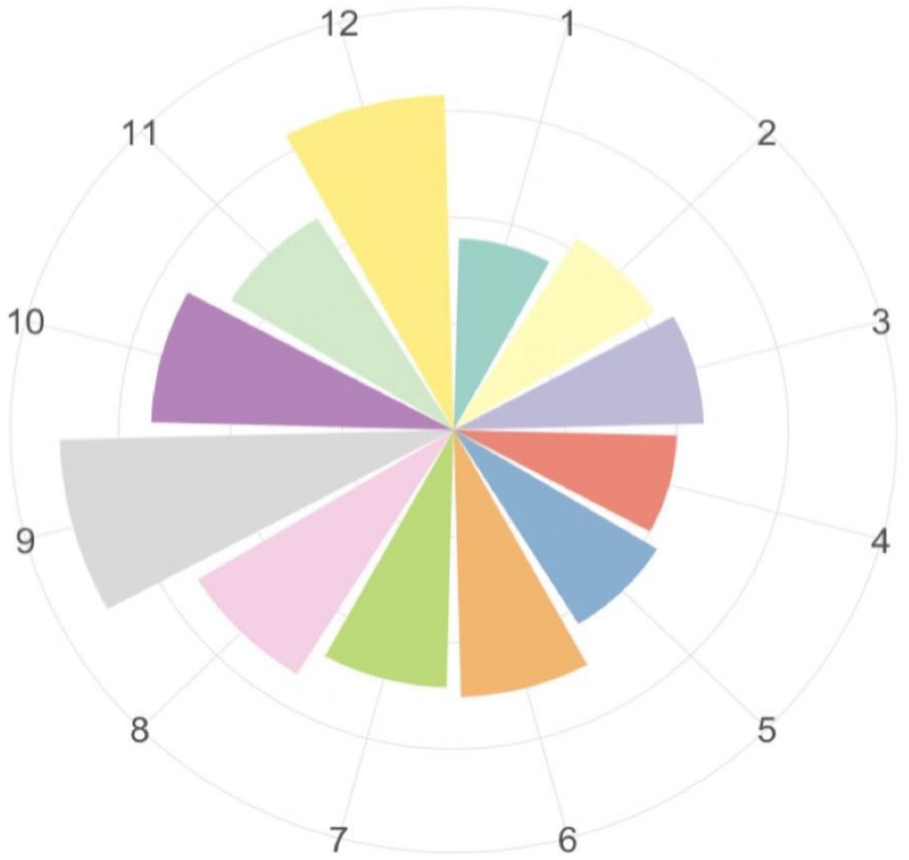


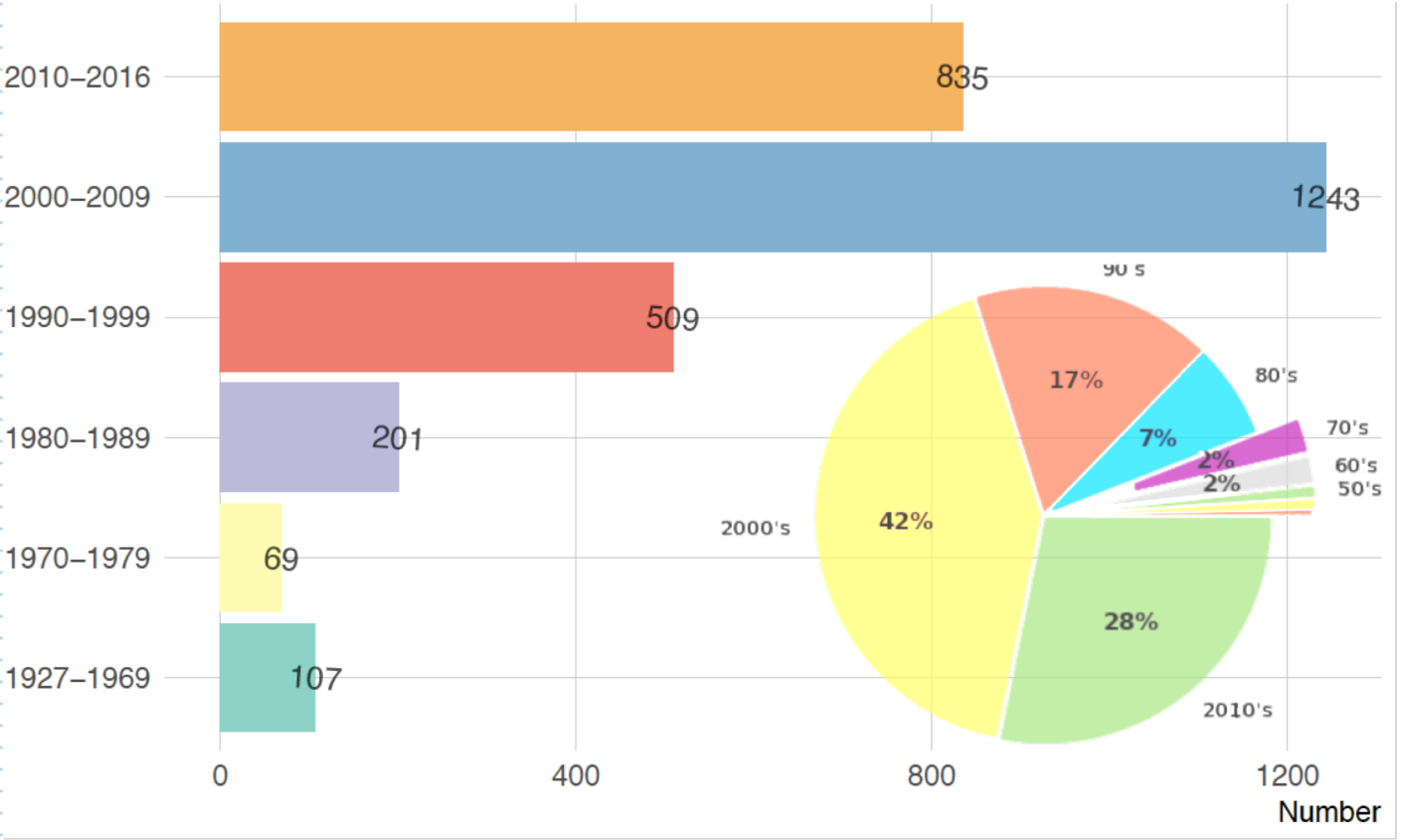


EDA – Time

- **Month**: convert to a factor variable;

  month 1 (January) is the reference group.

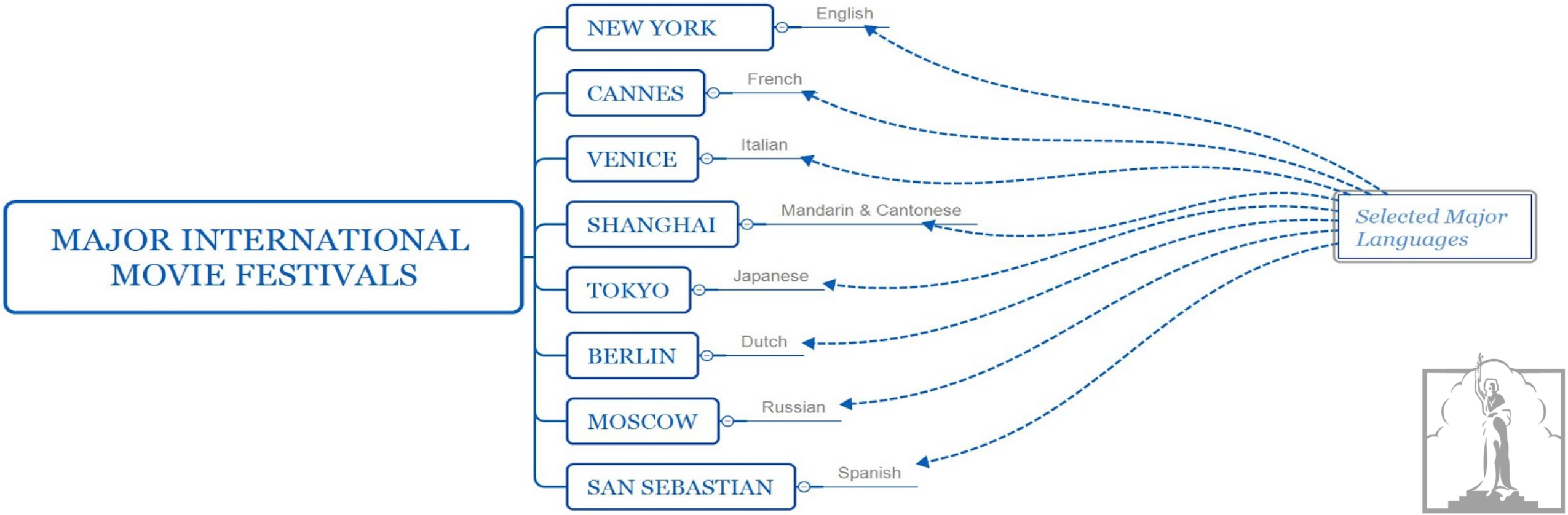




- **Year**: 1927-2016; spans 89 years; convert to a variable that has value 0 if year = 1927, has value 89 if year = 2016, etc.

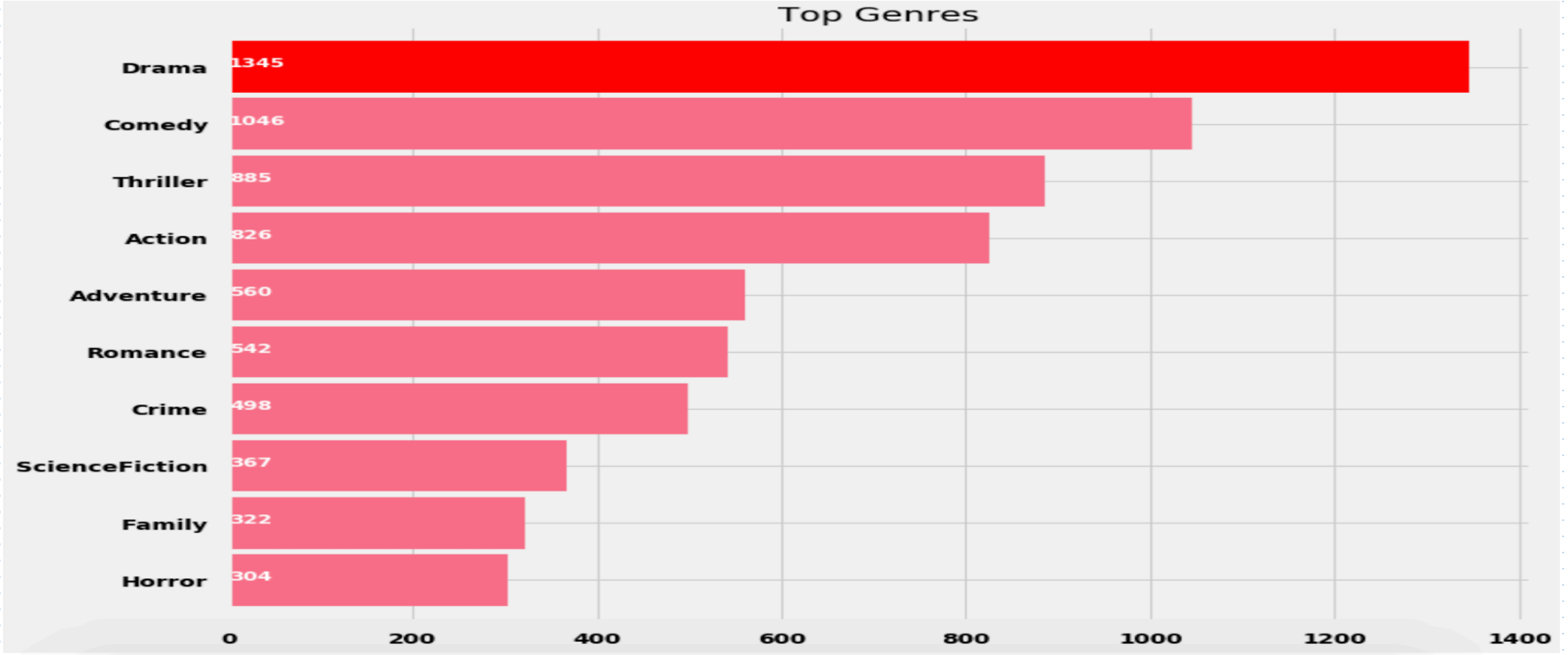
EDA – Spoken Language

- **Spoken Language**: 15 major language are encoded into multi-hot variables.
- **Min_language**: 40 minor languages are compressed in one variable



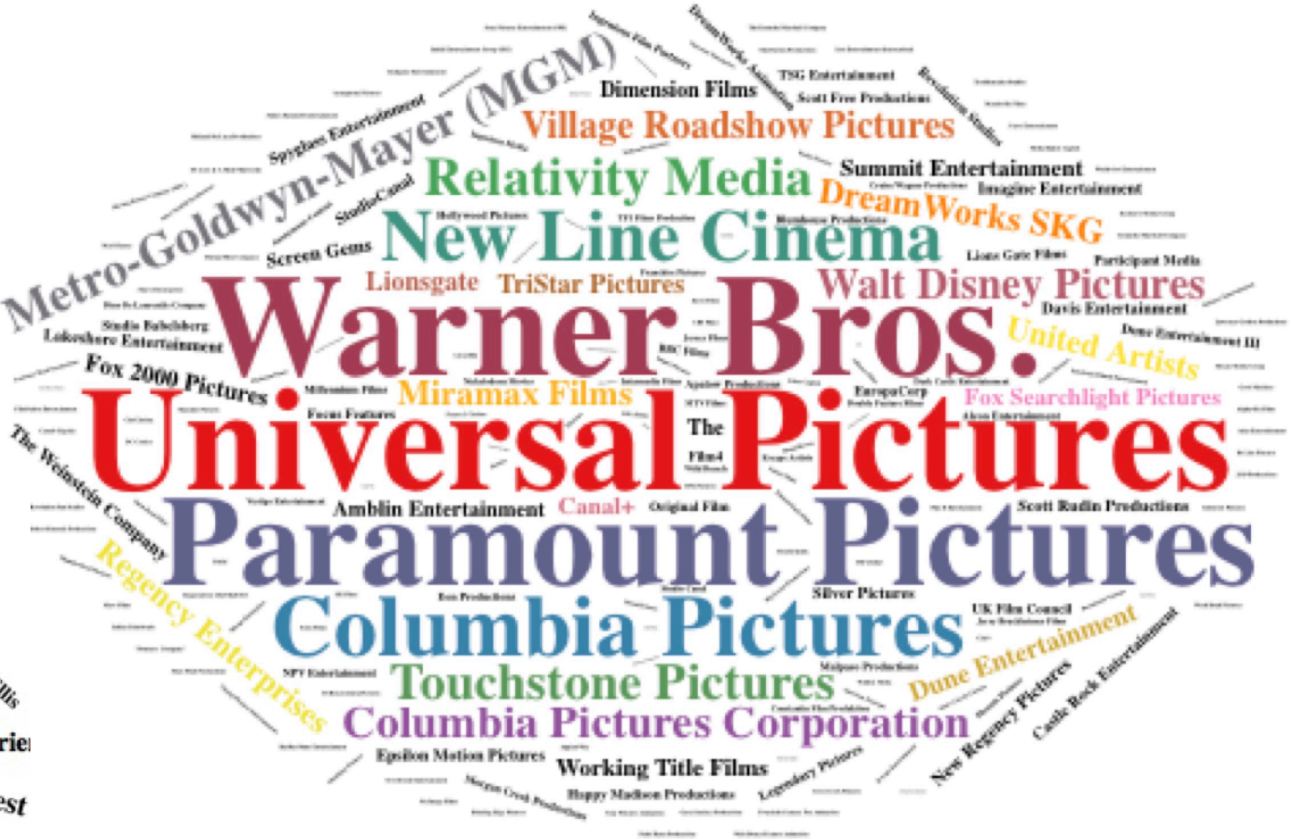
EDA – Genres

- Action, Comedy, Thriller etc.
- 19 genres are encoded into multi-hot variables



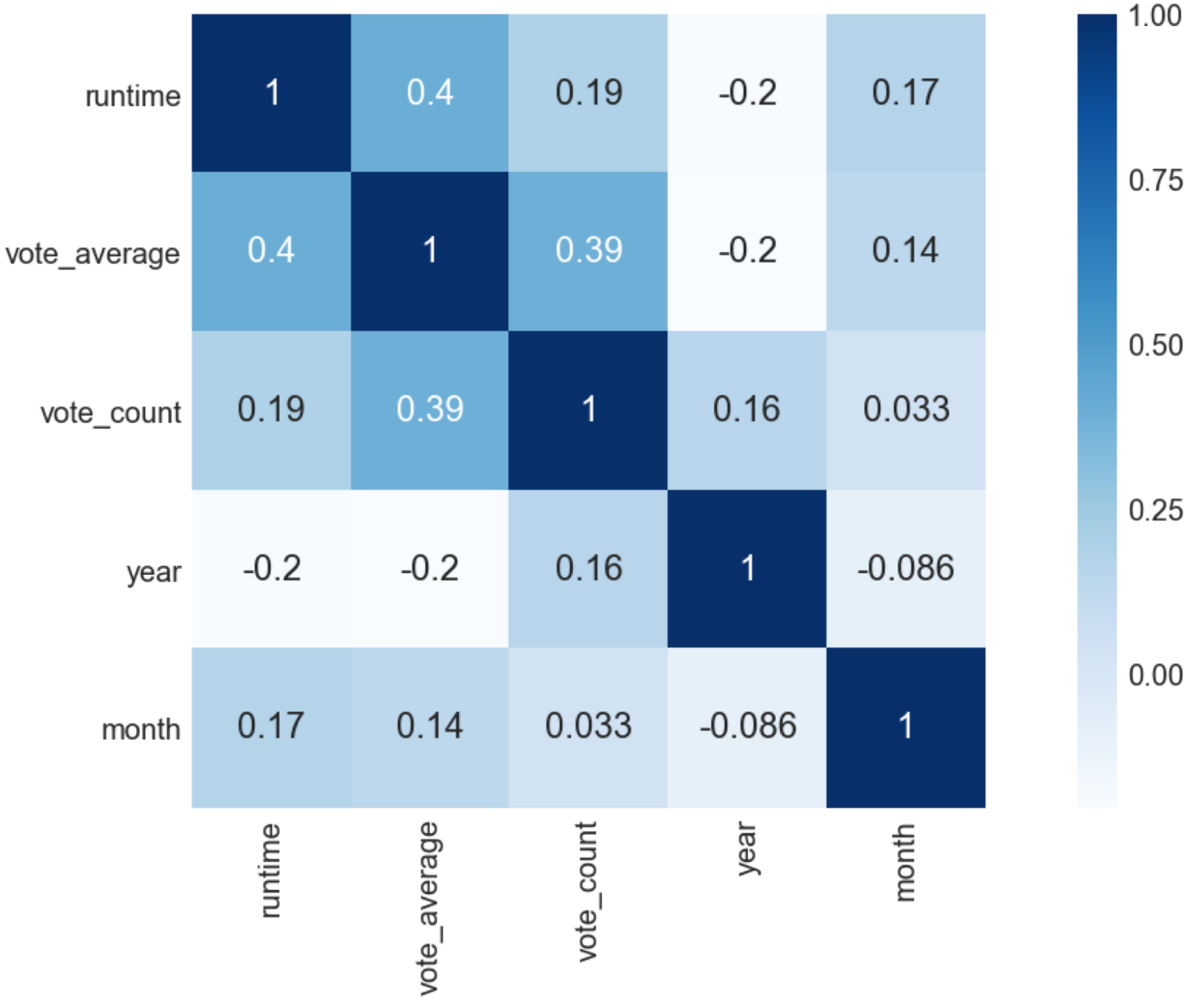
EDA – Director vs Studios

- **Director:** Draw ‘director’ feature from ‘crew’.



- **Studios**
- majority_studio: the number of top six production companies that involved in the movie.
- minority_studio: the number of other production companies that involved in the movie.

EDA – Correlation between Variables



03 Part Three
Linear Regression Model

Datasets

- VIFs of predictors in our dataset are very small and less than
  - no collinearity is detected
- Training data: include 75% of the cleaned data
- Testing data: include 25% of the cleaned data
- We train models on training data and assess their performances using testing data.

Linear Regression Model

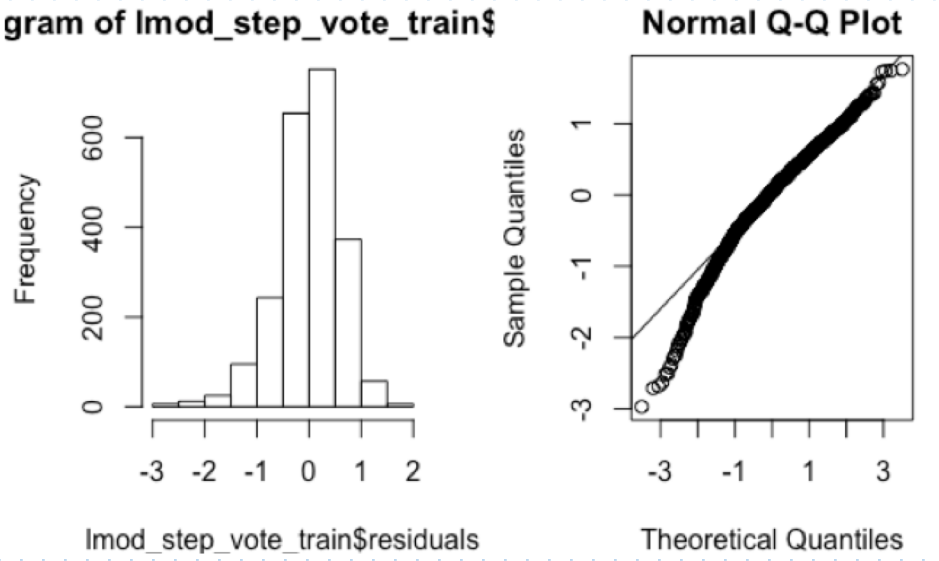
1. Model Selection
   - Stepwise selection using AIC
   - LASSO L1 regularization
     - 41 predictors, large p
2. Diagnosis of underlying assumptions
   - Linearity
   - Normality
   - Homoscedasticity:
   - Uncorrelated Variance
3. Corrective Measures
   - Transformation: Log, Box-cox
   - Build WLS
   - Ridge

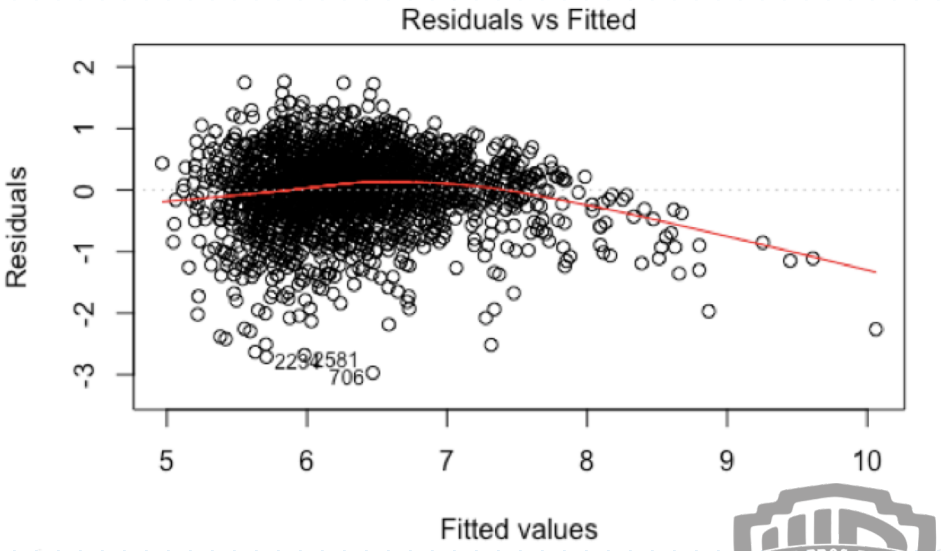
Linear Regression Model – Model Selection (Stepwise)

- **OLS model:**
  - The final model has the smallest AIC
- Vote_average ~ runtime + vote_count + year + month + Documentary + Crime + Foreign + Adventure + Action + Comedy + Science.Fiction + Fantasy + Drama + Animation + Family + Horror + L4 (Russian) + L9 (Dutch) + L15 (English) + L50 (Cantonese) + L53 (Thai) + min_language + majority_studios + minority_studios
- (24 predictors in total)

Linear Regression Model – Diagnostics

- Linearity & functional form
  - R-squared: 0.48
- Normality
- Homoscedasticity
  - Breusch-Pagan test against heteroskedasticity:
    p-value = 0.000627 < 0.05
  - Score Test for Non-Constant Error Variance:
    p-value = 0.010083 < 0.05
- Uncorrelated error
  - Durbin-Watson test: p-value = 0.626 > 0.05
- Outliers and influential points
  - Bonferroni Outlier Test： 706, 2234, 2581, 357 are detected
  - Plot Cook's distance: 2244, 2132, 182 are highly influential points.





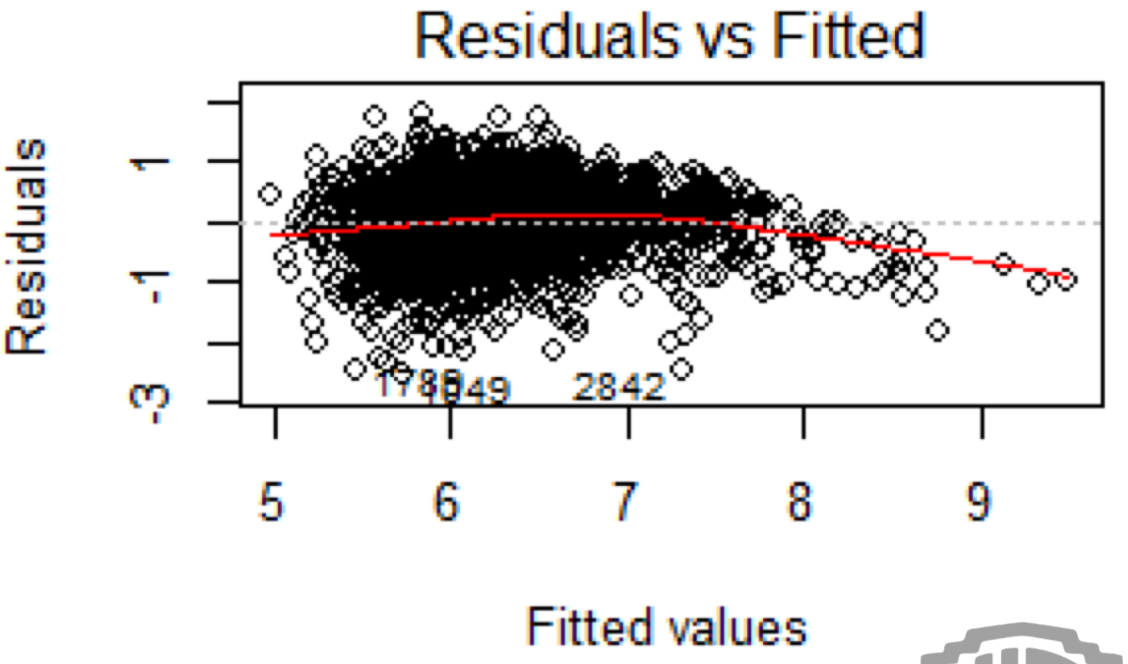
Most assumptions fail, except for uncorrelated error.
Need transformation!!

Linear Regression Model – Weighted Least Squares

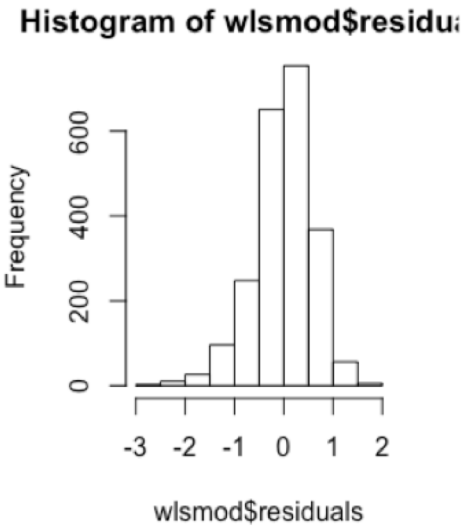
- The suggested model for heterskedasticity is:
  $\mathrm{Var}(\varepsilon_i) = \hat{\sigma}_i^2 = \hat{Y}_i^2$, where $\hat{Y}_i$ is the fitted value of OLS model.
- Solve for heteroscedasticity:
  1. Regress the absolute values of the residuals against the fitted values.
  2. Get new fitted values, denote them as $\hat{Y}_{new,i}$
  3. Let $\mathrm{weight}_i(W_i) = {}^{1}/_{\sigma_i^2} \approx {}^{1}/_{\hat{Y}_{new,i}^2}$, fit a linear mode

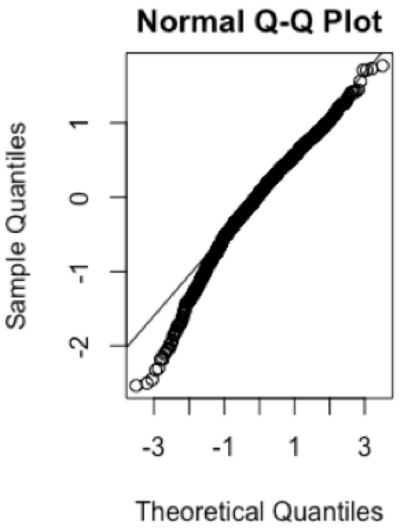
$$\hat{\beta}_{WLS} = \arg\min_{\beta} \sum_{i=1}^{n} \varepsilon_i^{*2} = (X^T W X)^{-1} X^T W Y$$

A "metaphone" shape, this indicates a possible solution for calculating weights in WLS.



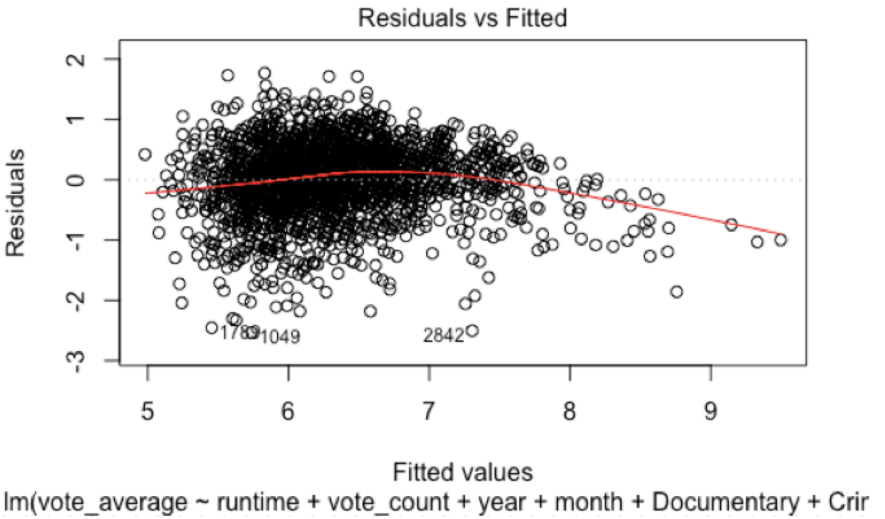
Linear Regression Model – Diagnostics





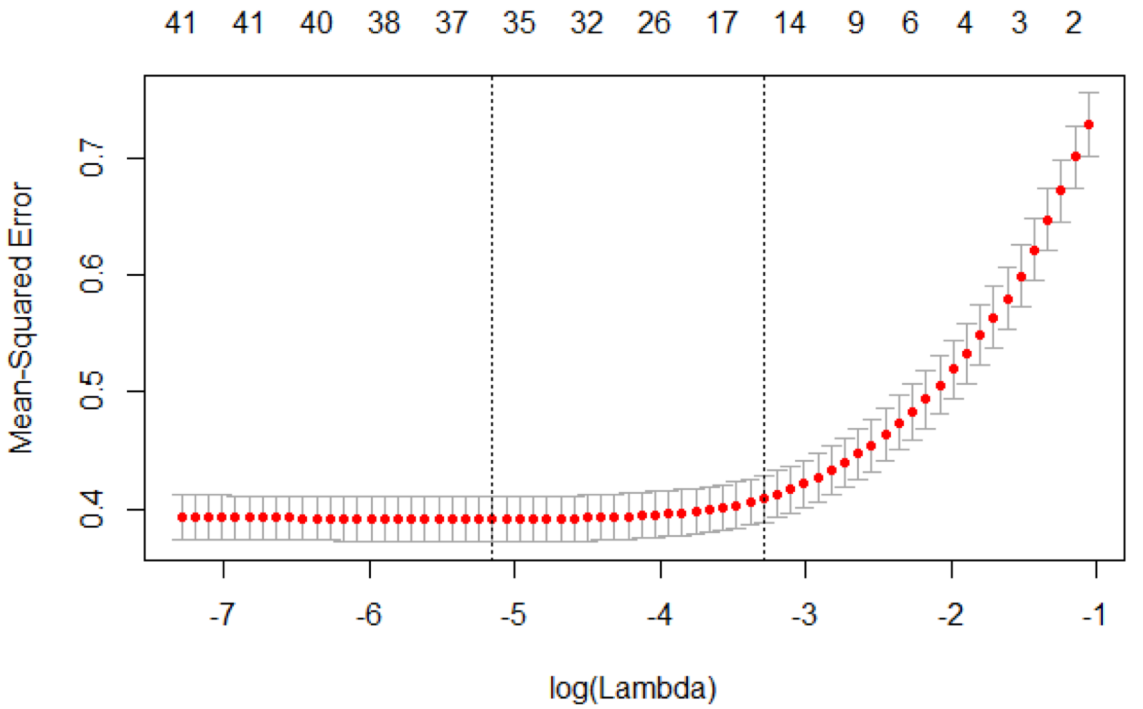
- Remove extreme points and fit WLS
- Diagnostics after building WLS
  - Linearity & functional form
    - R-squared: 0.5
  - Normality
  - Homoscedasticity
    - Score Test for Non-Constant Error Variance: p-value = 0.61733 > 0.05
    - Most error variances look constant from residuals vs fitted value plot.
  - Uncorrelated error
    - Durbin-Watson test: p-value = 0.982 > 0.05
  - Outliers and influential points
    - Bonferroni Outlier Test: Only “2842” is detected
    - Plot Cook’s distance: No influential points.

All Assumptions hold!



Linear Regression Model – Model Selection (LASSO)

- Selected Variables: runtime, vote_count, year, month, Documentary, Crime, Foreign, War, Adventure, Western, Music, Mystery, Action, Comedy, Science.Fiction, Romance, Fantasy, Drama, Animation, Family, Horror, L4, L6, L9, L10, L12, L13, L15, L33, L50, L52, L53, L54, min_language, majority_studios and minority_studios. (**36** predictors in total)

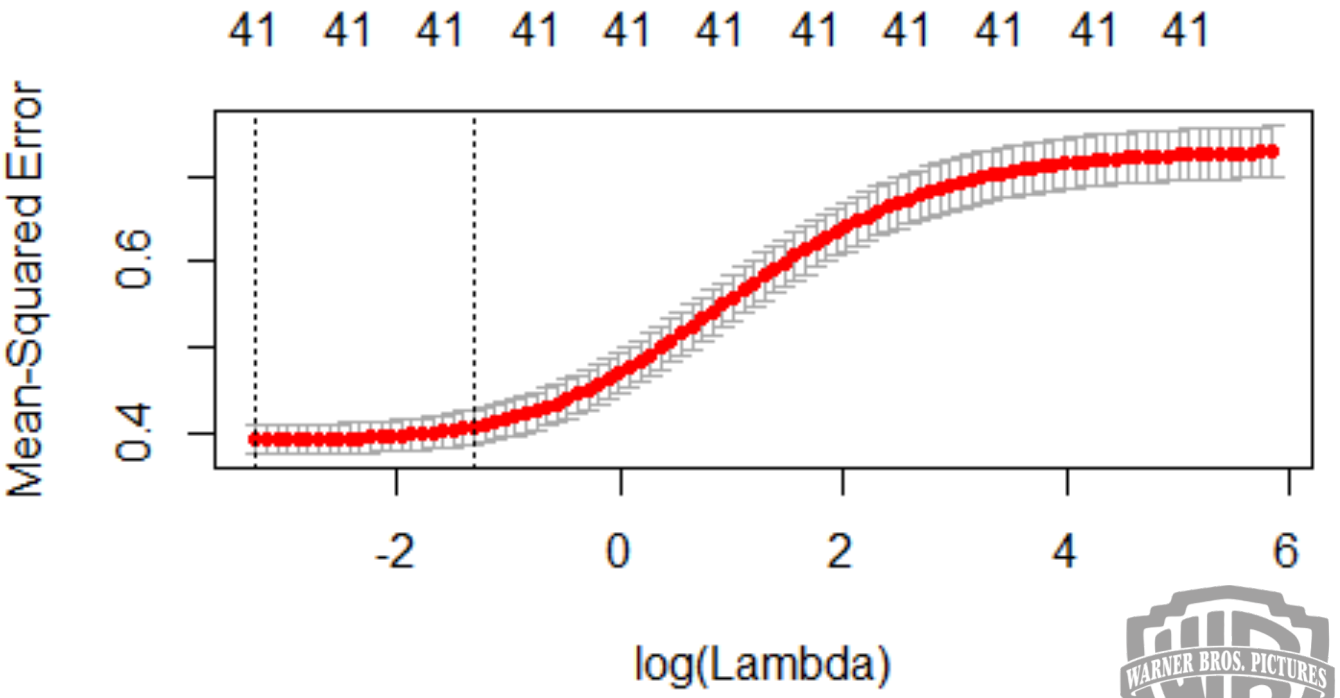


Linear Regression Model – Diagnostics & Weighted Least Squares

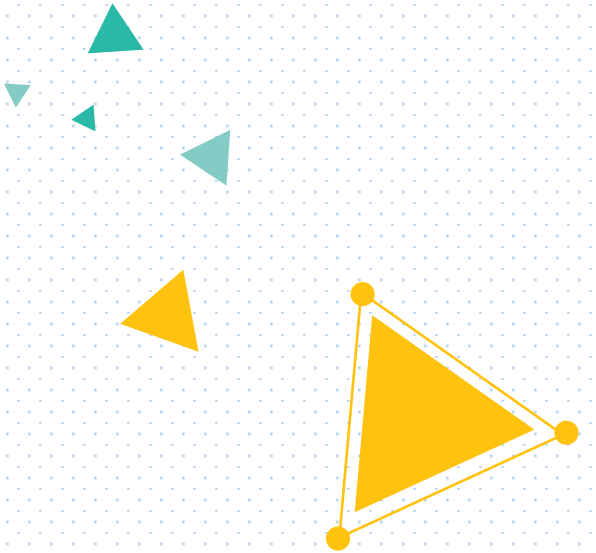
- Diagnostics
  - Linearity and Uncorrelated errors hold
  - Normality and Homoscedasticity are violated
  - Same extreme points are detected
- Transformation
  - Log, Box-cox can’t solve for unequal error variance
- WLS
  - All Assumptions hold!

Linear Regression Model – Ridge Regression

- Ridge regression is aimed to minimize: $\sum_i (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$
- Ridge regression does not have the function of variable selection
- Check of the assumptions
  - Linearity and Uncorrelated errors hold
  - Normality need not be assumed
  - Homoscedasticity are violated
- Use ridge on the predictors selected by LASSO



04 Part Four
Non-Linear Model

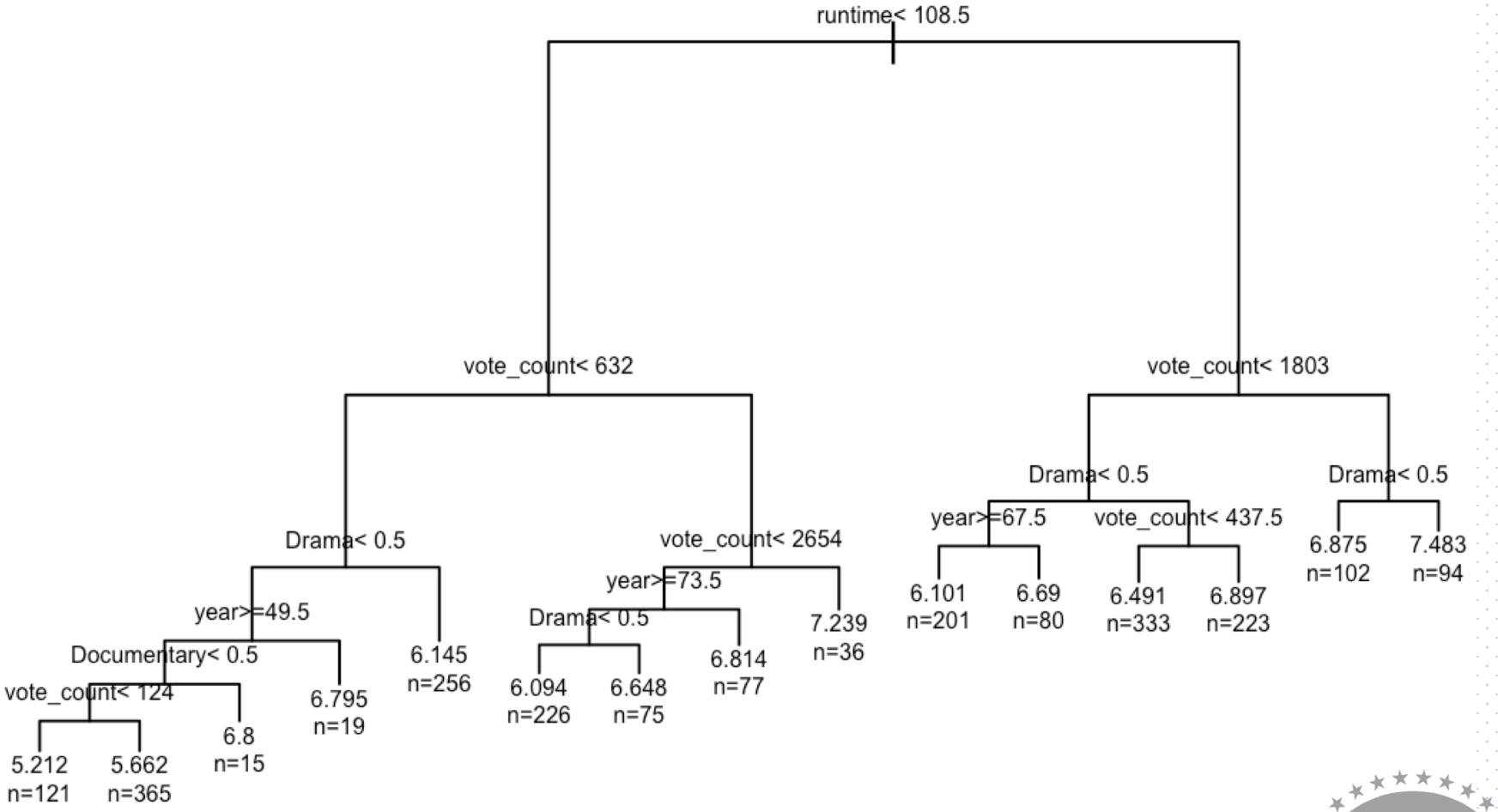


Nonlinear Model

1. Simple Tree Model
2. Random Forest
3. Xgboost
4. Neural Network

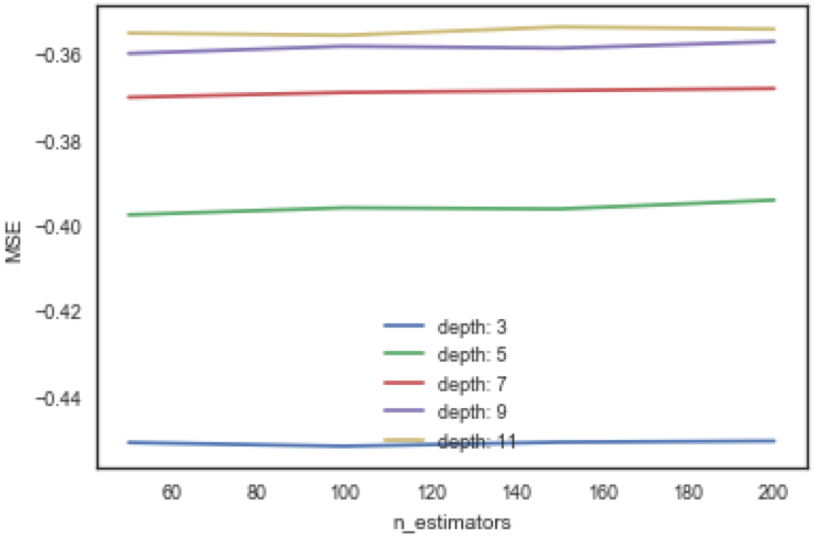
Nonlinear Model – Simple Tree Model

- Parameters:
  - Depth: 8
  - Split: 20
  - CP: 1%



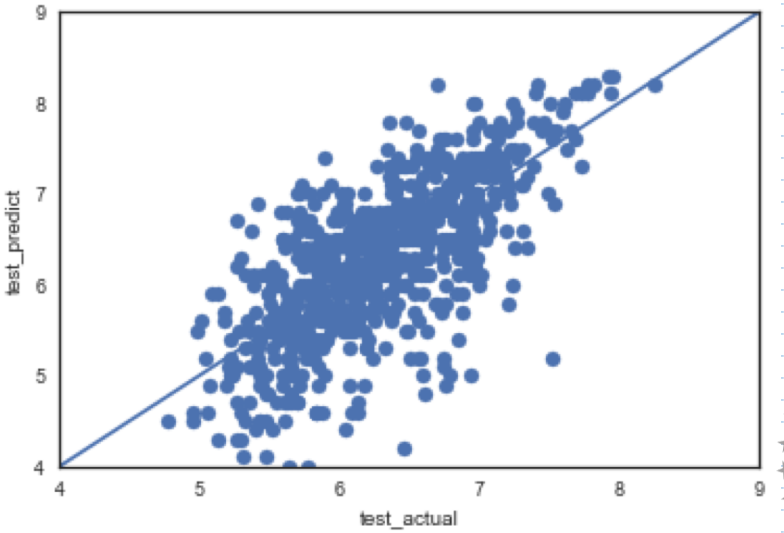
Nonlinear Model – Random Forest

- Model selection:
  - Max_depth = 11
  - N_estimators = 150



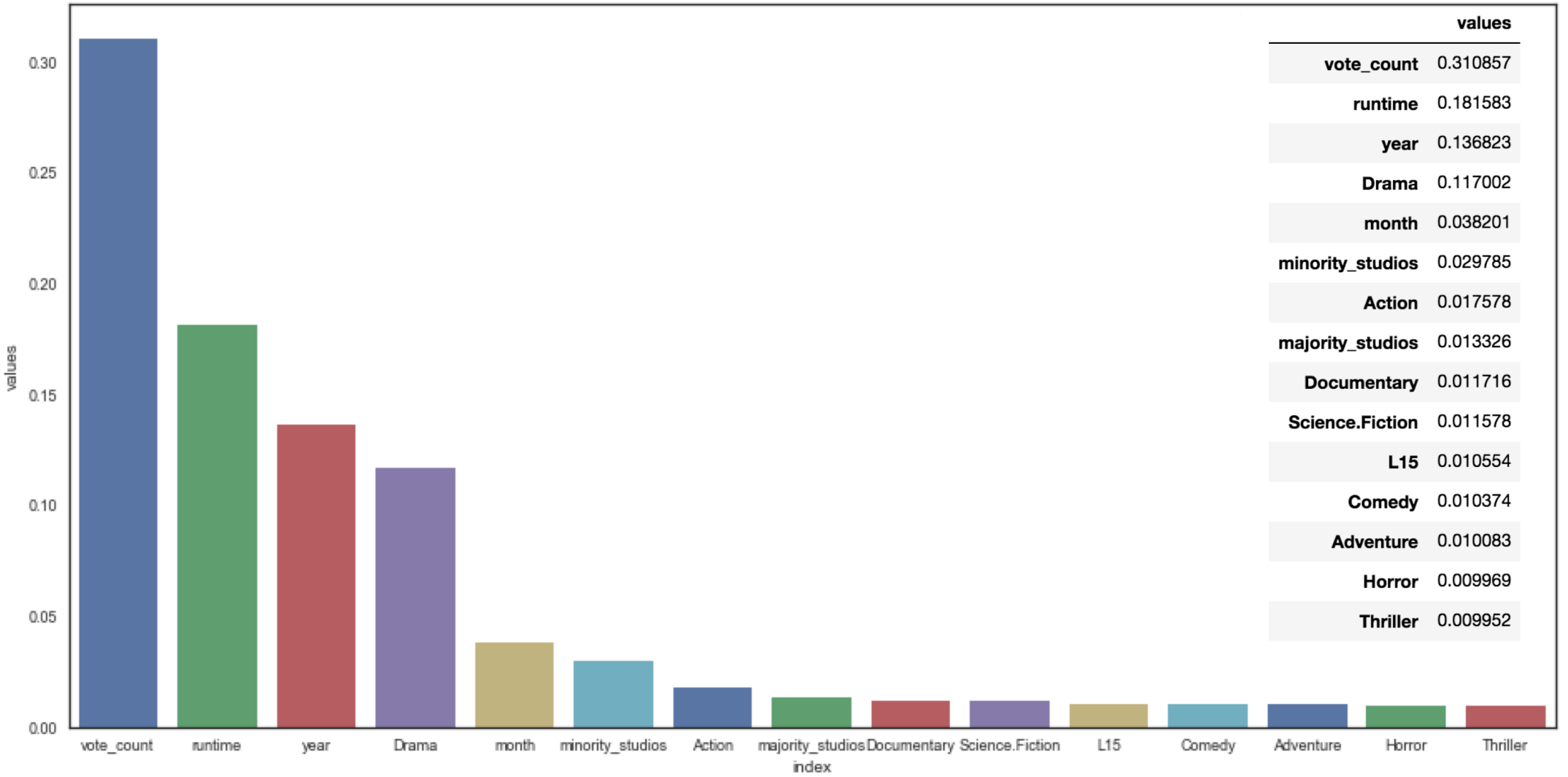

```
# R squared
best_model.score(X_vote_average_test, y_vote_average_test)
```

```
0.49996323304538326
```



Nonlinear Model – Random Forest

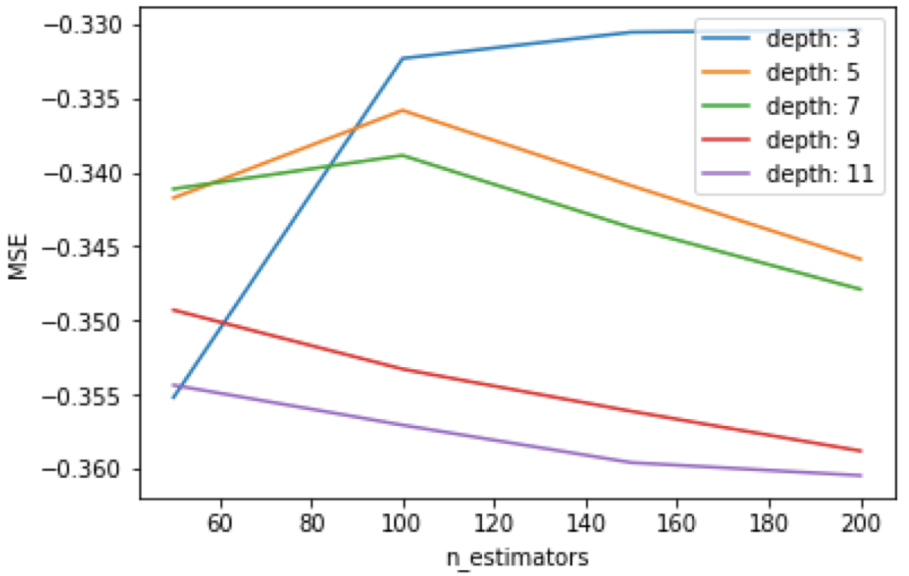
Feature Importance

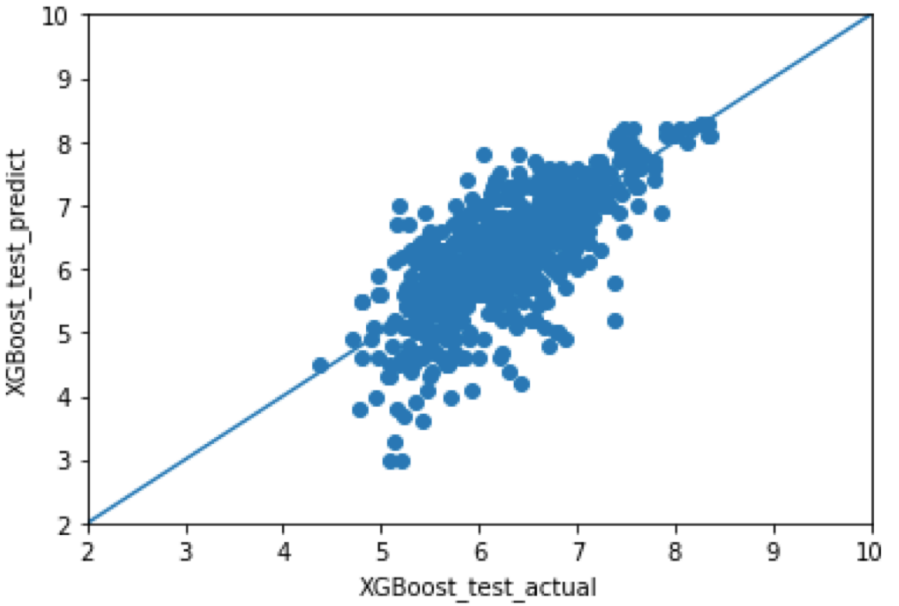


	values
vote_count	0.310857
runtime	0.181583
year	0.136823
Drama	0.117002
month	0.038201
minority_studios	0.029785
Action	0.017578
majority_studios	0.013326
Documentary	0.011716
Science.Fiction	0.011578
L15	0.010554
Comedy	0.010374
Adventure	0.010083
Horror	0.009969
Thriller	0.009952

Nonlinear Model – XGBoost

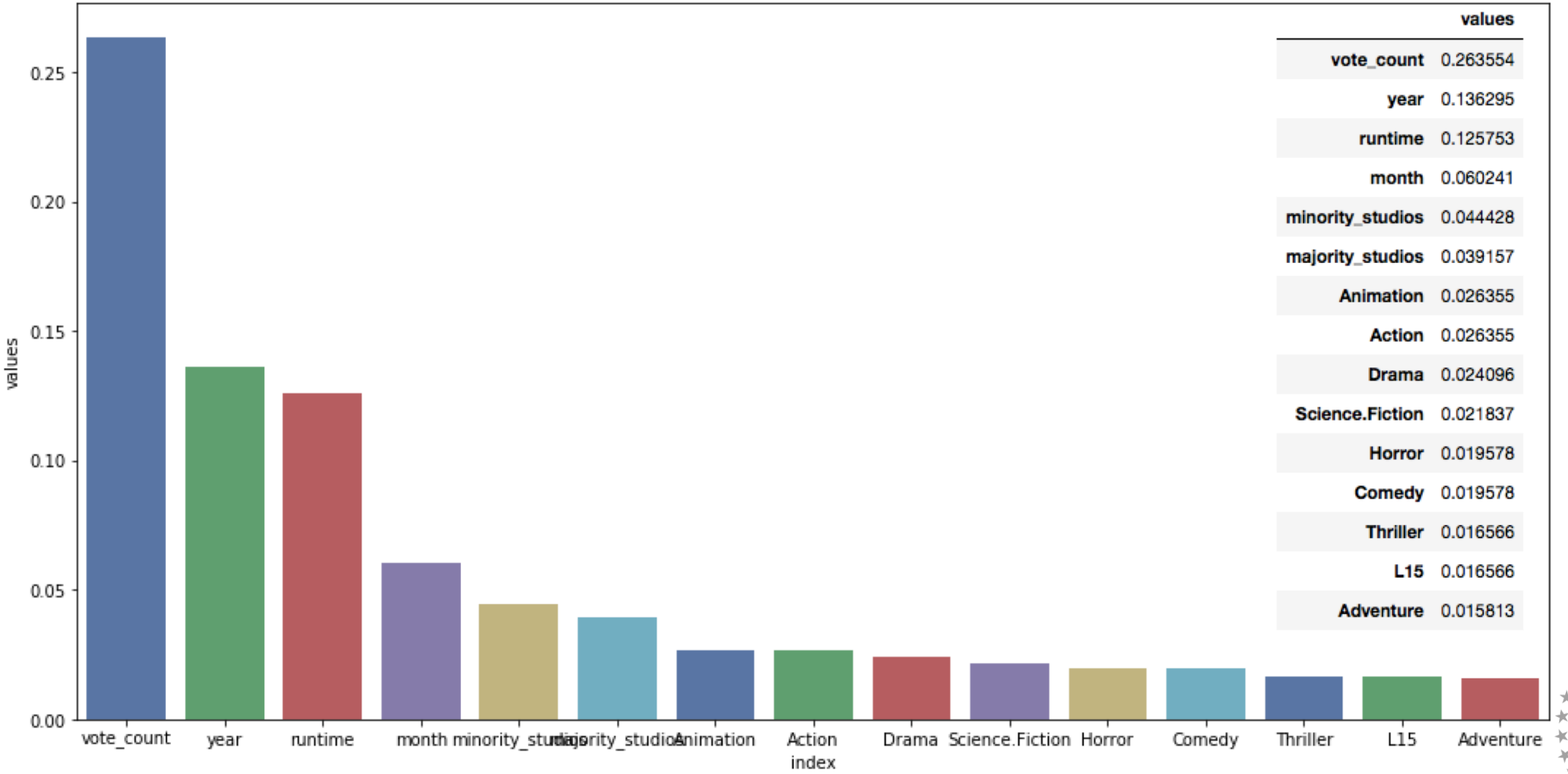
- Model selection:
  - Max_depth = 3
  - N_estimators = 200





Nonlinear Model – XGBoost

Feature Importance



	values
vote_count	0.263554
year	0.136295
runtime	0.125753
month	0.060241
minority_studios	0.044428
majority_studios	0.039157
Animation	0.026355
Action	0.026355
Drama	0.024096
Science.Fiction	0.021837
Horror	0.019578
Comedy	0.019578
Thriller	0.016566
L15	0.016566
Adventure	0.015813

Neural Network

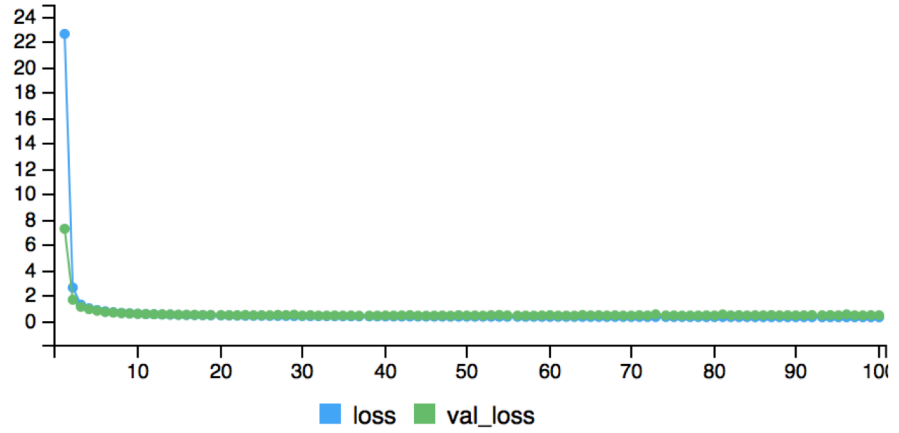
- Model specification
  - Two hidden dense layers followed by dropout layers
  - 16 neurons each

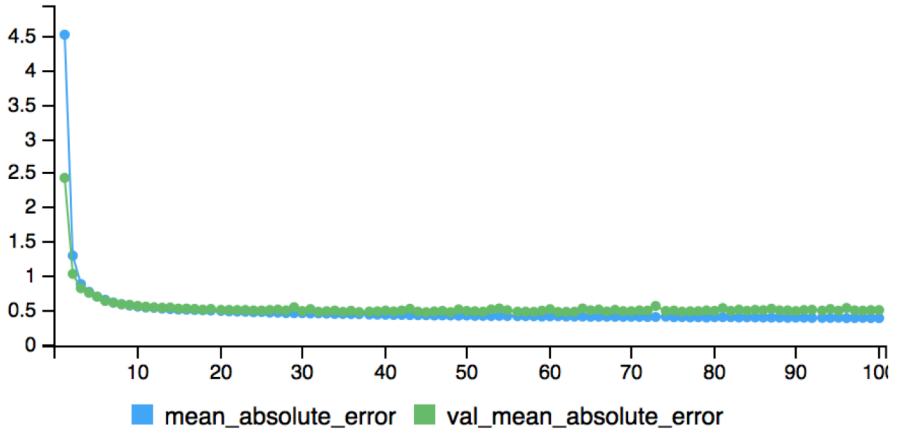
Layer (type)	Output Shape	Param #
dense_58 (Dense)	(None, 16)	672
dropout_34 (Dropout)	(None, 16)	0
dense_59 (Dense)	(None, 16)	272
dropout_35 (Dropout)	(None, 16)	0
dense_60 (Dense)	(None, 1)	17

Total params: 961
Trainable params: 961
Non-trainable params: 0

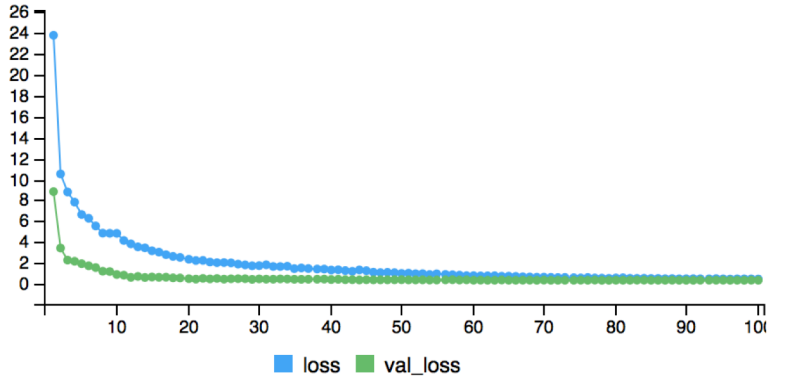
Neural Network

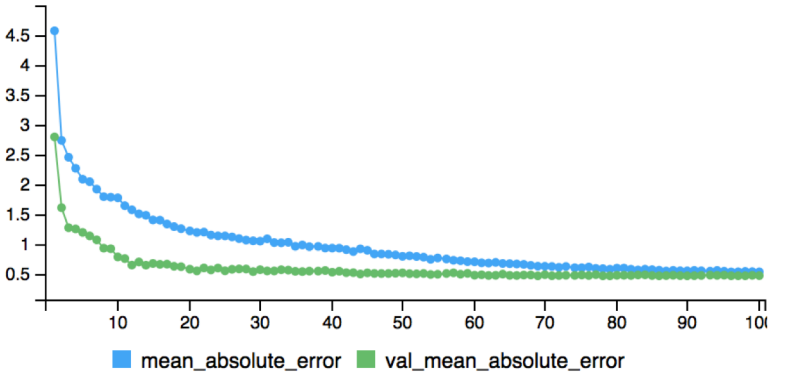
- Training without regularization
- Overfitting



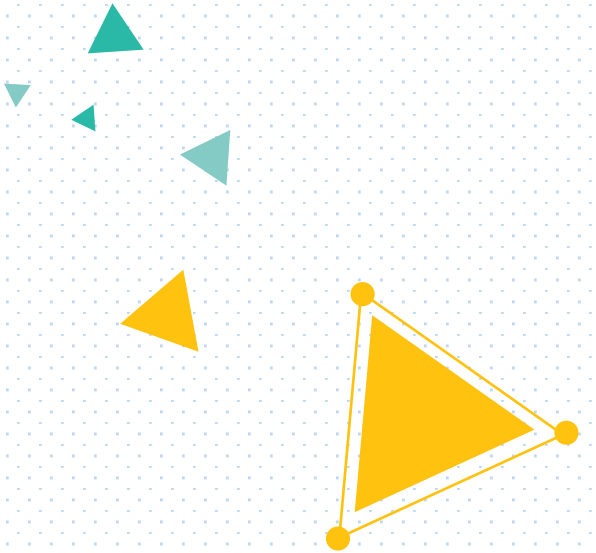


- Training with regularization (dropout)
- Small improvement
- Overfitting persists
- Test error similar to that of slm





05 Part Five
Conclusion



Model Comparison

- **Test Error (MAPE)**

Models	**Test Error (in %)**
Weighted Least Square (AIC)	8.37%
Ridge	8.38%
Weighted Least Square (LASSO)	13.43%
Tree Model	9.60%
Random Forest	8.21%
Xgboost	7.83%
Neural Network	8.21%

Conclusion

- Xgboost is the best model based on MAPE measurement
  - It can learn complex non-linear decision boundaries via boosting.
  - It is optimized for sparse input.
  - It has an additional custom regularization term in the objective function, which helps prevent overfitting.
- Further Study
  - Interpret models back to the real world
  - Get a different testing data
    - Predict upcoming movies' vote average, find prediction accuracies
      - Movies released in November

Thank You!

Q & A

Thank you

Add your text in here