Predict Movie Ratings

Ryan Tran, 03/20/18

Introduction

Proposal

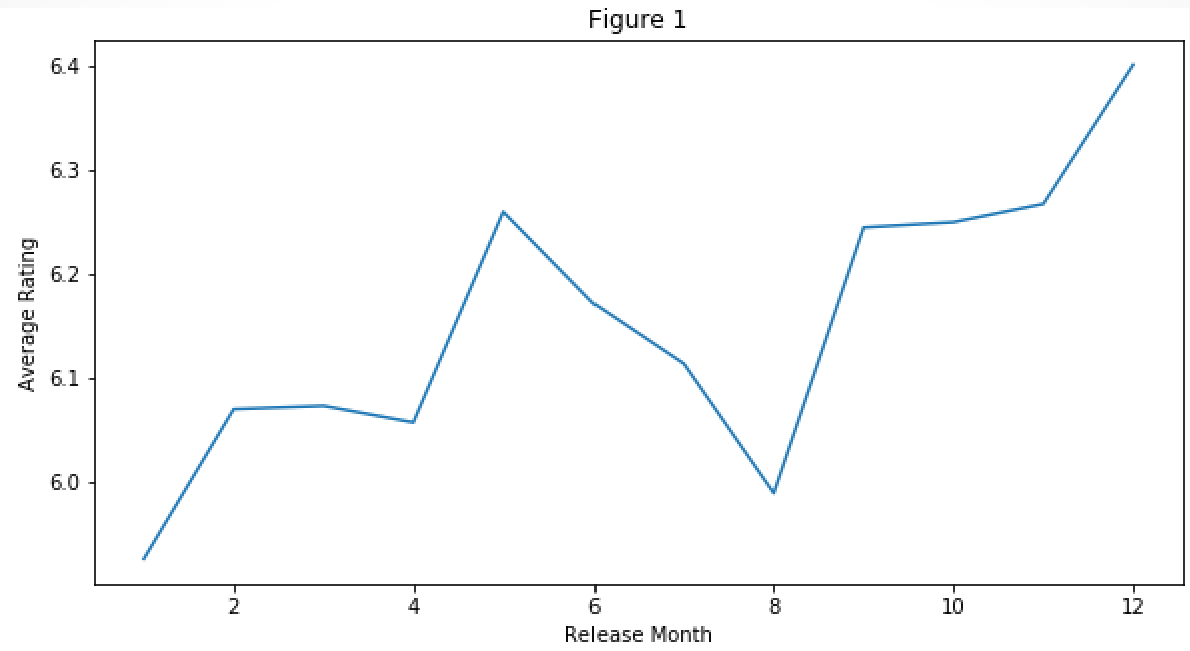
- Client: Movie Industry
- Problem:
  - The movie industry spends millions of dollars making movies
  - This can be very risky since they have no idea how well a movie will do
- Goal:
  - Predict the average rating of a movie before it has been released
  - Machine Learning

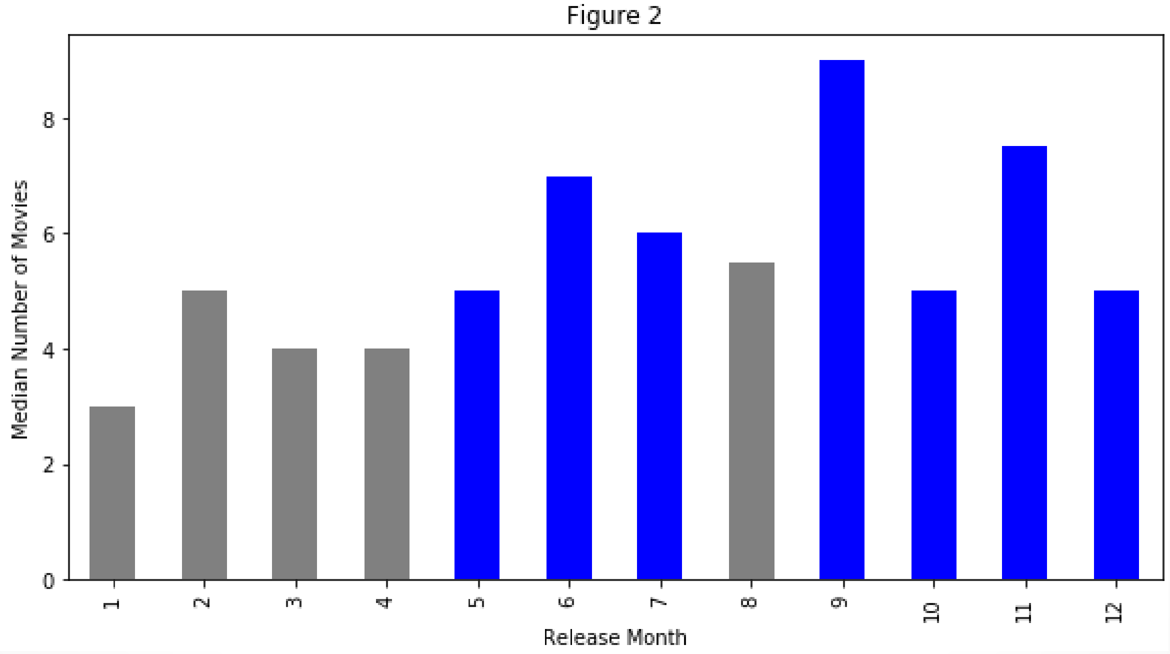
Exploratory Data Analysis

Look for features that the model can learn from

Obvious Correlations

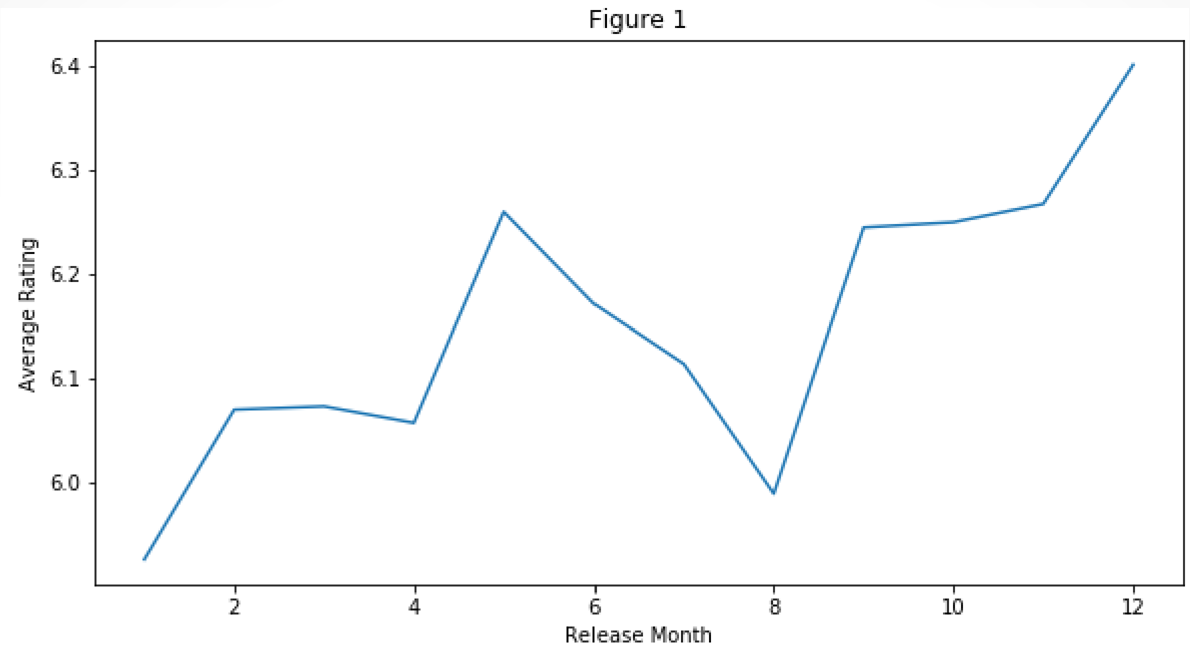
- Are there any obvious correlations?
- Release Month
  - Plot average ratings per month
  - Plot the median number of movies each month
- Budget
  - Average ratings by budget plot is too volatile
    - Plot average budget per month instead
  - Compare with average ratings per month

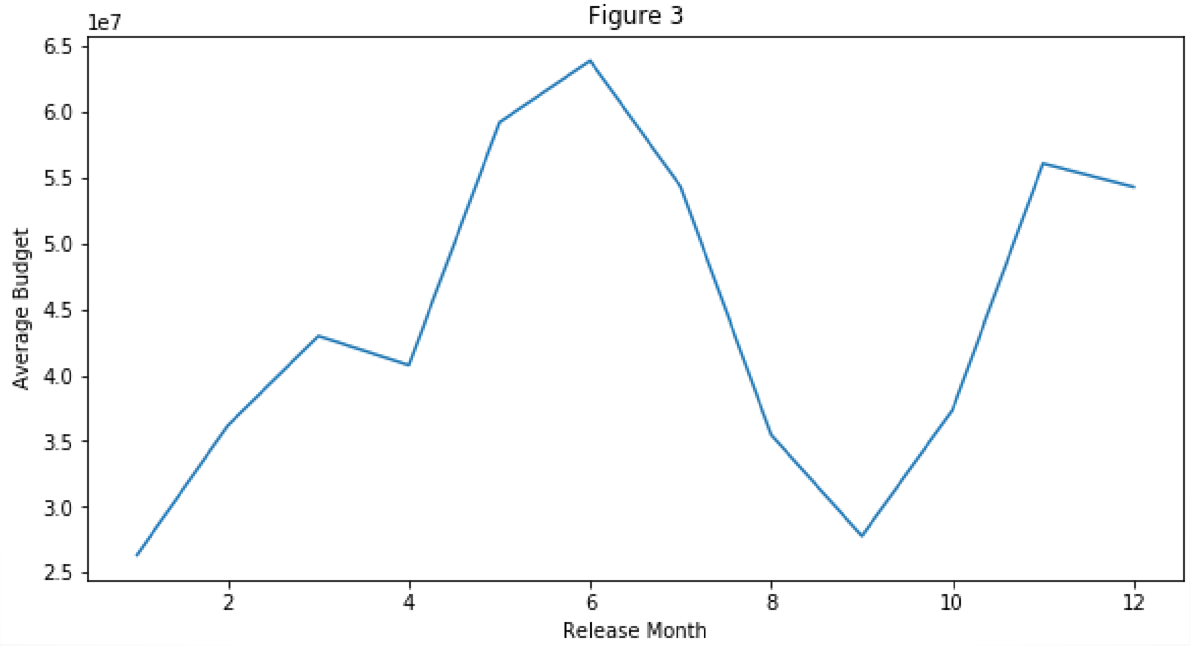




Release Month

- Certain months have higher average ratings.
  - Corresponds with higher number of new releases
- Possible correlation between rating and month
- Will use release month as a feature





Budget

- May, June, July, November, and December seem to be correlated
  - But Not for September and October
- Possible correlation between budget and rating
- Will use budget as a feature

Best actors / directors / production companies

- Actors, directors, and production companies are important features to include
  - How?
- Model should learn who the best actors, directors, and production companies are
- Start by finding the top 10 actors / directors / production companies by genre

Modeling

Build linear regression model to predict ratings

Preprocessing

Build the feature columns:

- Actors, Directors, and Production Companies
  - Count the number of times each movie has a feature that appears at least once in the top 10 lists
  - Compute using the movie's genres
- Genres
  - Boolean column for each genre
- Release Month
  - Numerical representation of month
- Budget
  - Unchanged

Preprocessing (2)

Actors, Directors, and Production Companies (Example)

- Movie categorized under the genres of Action and Adventure
- Actors Feature Column:
  - Count the number of times the following are true:
  - Movie has at least one actor from the top 10 action list
  - Movie has at least one actor from the top 10 adventure list
- Repeat process for Directors and Production Companies

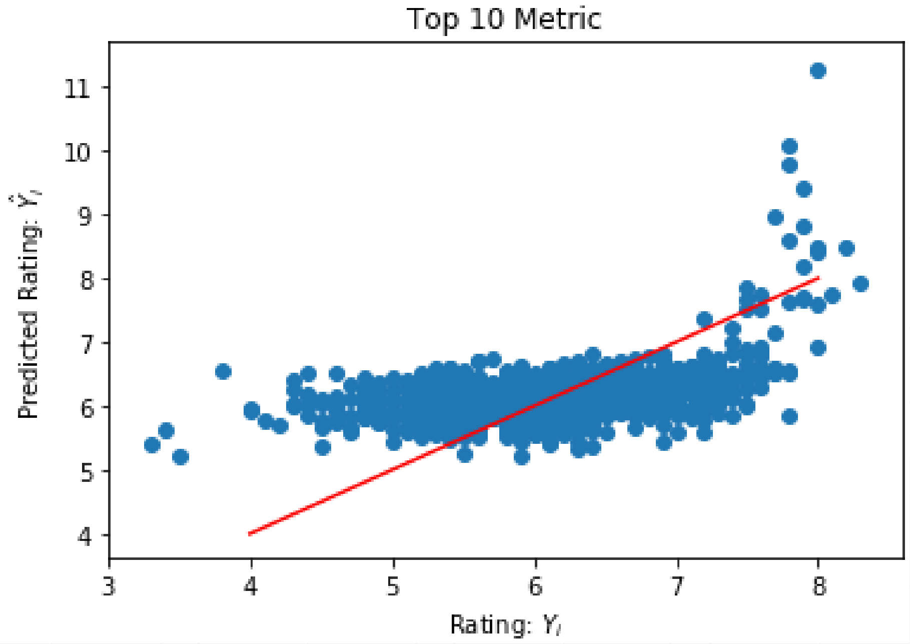
Fitting the Linear Regression Model

- Split data into training and test sets
- Build model using features of the training set
  - Ordinary Least Squares
- Predict the movie ratings of the test set

Fitting the Linear Regression Model (2)

- Extremely poor performance
- Most likely due to the scarcity of the main three features
  - Actor, Directors, Production Companies
- Must find ways to alter these features

```
R² (Training): 0.2552991953573587
R² (Test): 0.21130813333466592
MSE: 0.4738747447476443
```



Explore Possible Improvements

Look for ways to improve performance

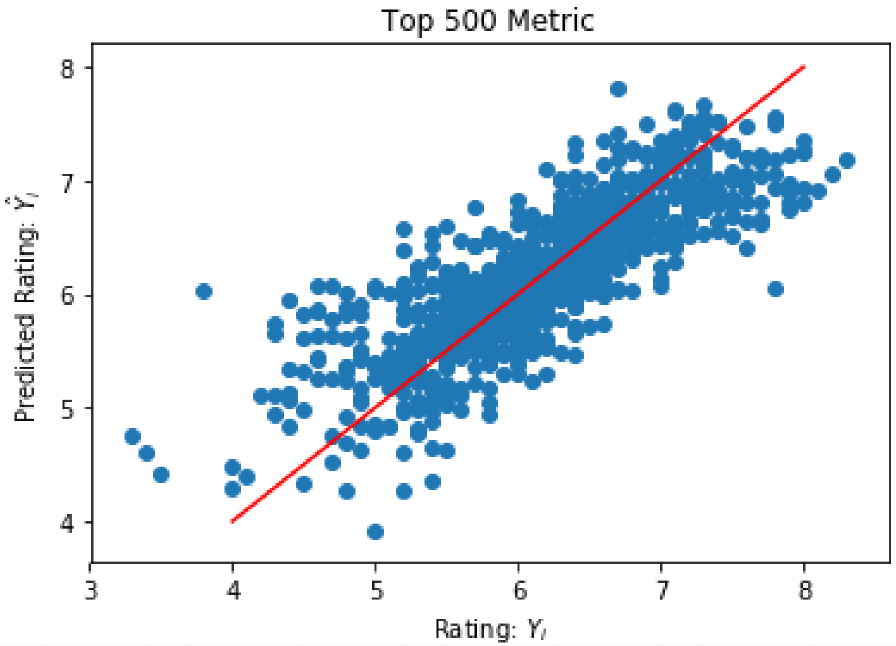
Increase # actors / directors / production companies

- Using the top 10 to produces sparse feature columns
  - Many zeros
- Increase non-zero values
  - Find the top 500 actors / directors / production companies by genre
- Check performance

Increase # actors / directors / production companies (2)

- Much better performance!
- Increasing the #, increases performance
- BUT, this is an odd metric
  - No reason to use such a large number
- Must find a more reasonable way to increase the number of
  - Actors / Directors / Production Companies

```
R² (Training): 0.6601872747752968
R² (Test): 0.6617422074423811
MSE: 0.20323757842828108
```



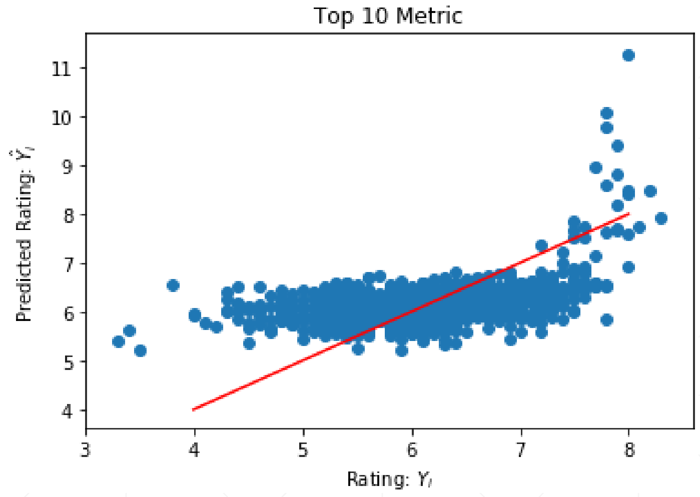
Preprocessing Changes

- Low values for Actors / Directors / Production Companies feature columns
  - Maximum value per movie limited to the number of its genres
- Increase numerical values of feature columns
- Create new preprocessing procedure
  - Count the total number of features listed for each movie by genre
- Check performance
  - Top 10
  - Top 500

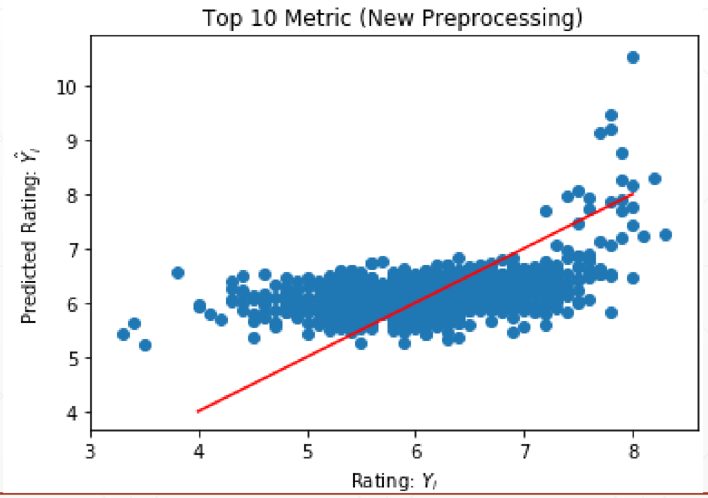
Preprocessing Changes (2)

- Comparison to Top 10 metric
- Little to no difference in performance
- BUT may prove useful for new models

```
R² (Training): 0.2552991953573587
R² (Test): 0.21130813333466592
MSE: 0.4738747447476443
```



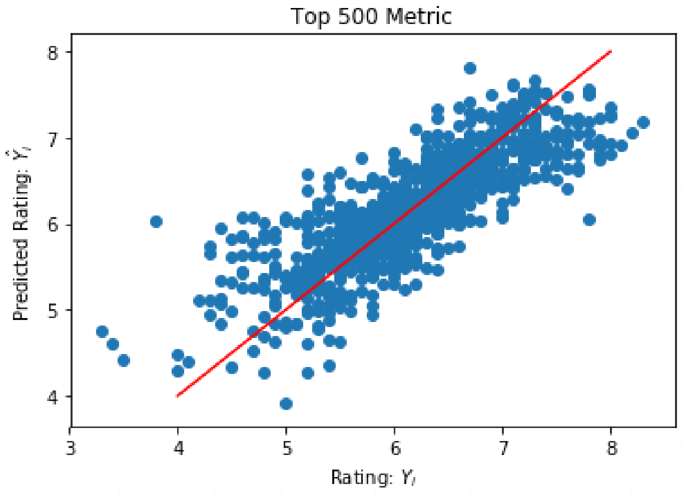

```
R² (Training): 0.24792179100483147
R² (Test): 0.22042639919728424
MSE: 0.4683961591417616
```



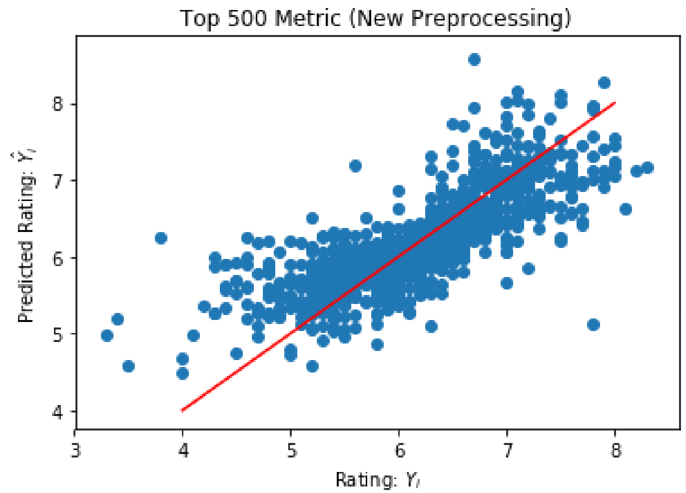
Preprocessing Changes (3)

- Comparison to Top 500 metric
- Slightly worse performance
- BUT may prove useful for new models

```
R² (Training): 0.6601872747752968
R² (Test): 0.6617422074423811
MSE: 0.20323757842828108
```




```
R² (Training): 0.6052314759134532
R² (Test): 0.587876427040588
MSE: 0.24761882453074427
```



New Models

Attempt to build new model(s) with better performance

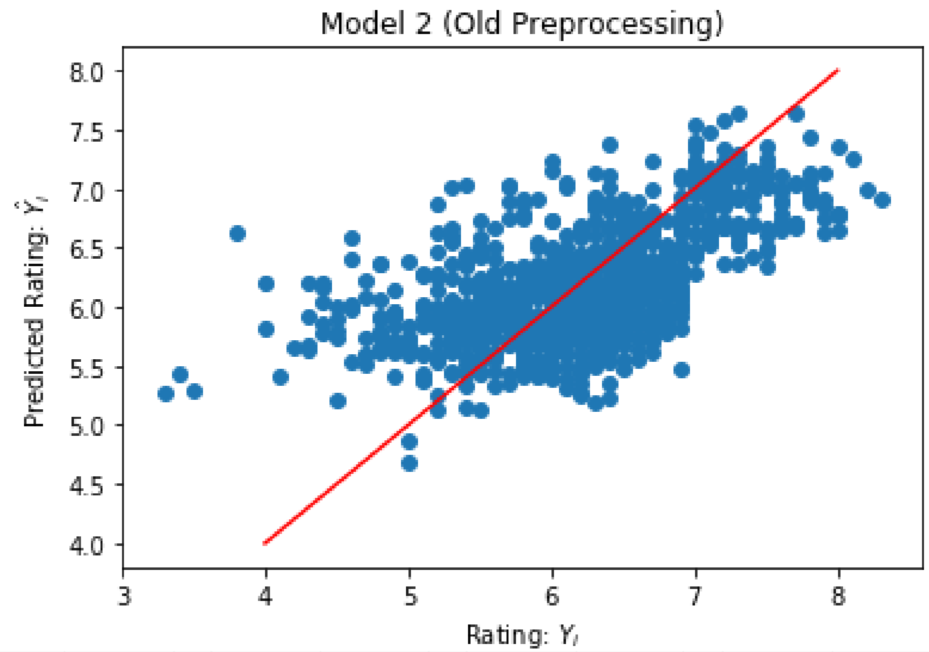
Model 2

- Controlled method of increasing # of Actors / Directors / Production Companies
- Find the BEST Actors / Directors / Production Companies
  - BEST: Ratings >= 7
- Check performance
  - Old Preprocessing
  - New Preprocessing

Model 2 (2)

- Old Preprocessing
- Better than Model 1: Top 10 metric
- Worse than Model 1: Top 500 metric

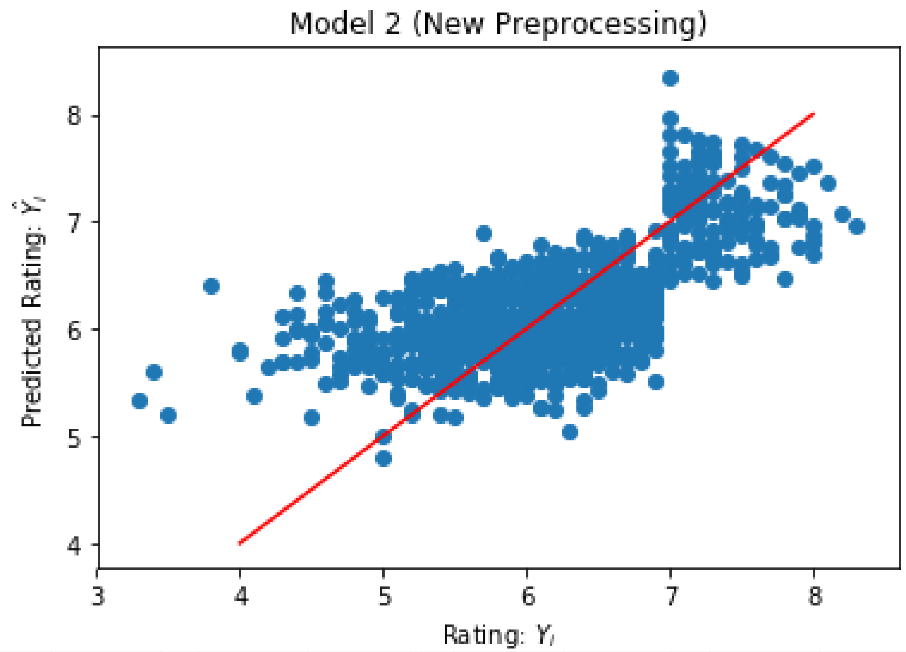
```
R² (Training): 0.3836005424729261
R² (Test): 0.355842205758068
MSE: 0.3870334197995831
```



Model 2 (2)

- New Preprocessing
- Better than Model 1: Top 10 Metric
- Worse than Model 1: Top 500 Metric
- Model 2 performs better using the New Processing process
    - Continue using New Preprocessing

```
R² (Training): 0.42407397272083824
R² (Test): 0.40357773139743336
MSE: 0.3583521806695396
```



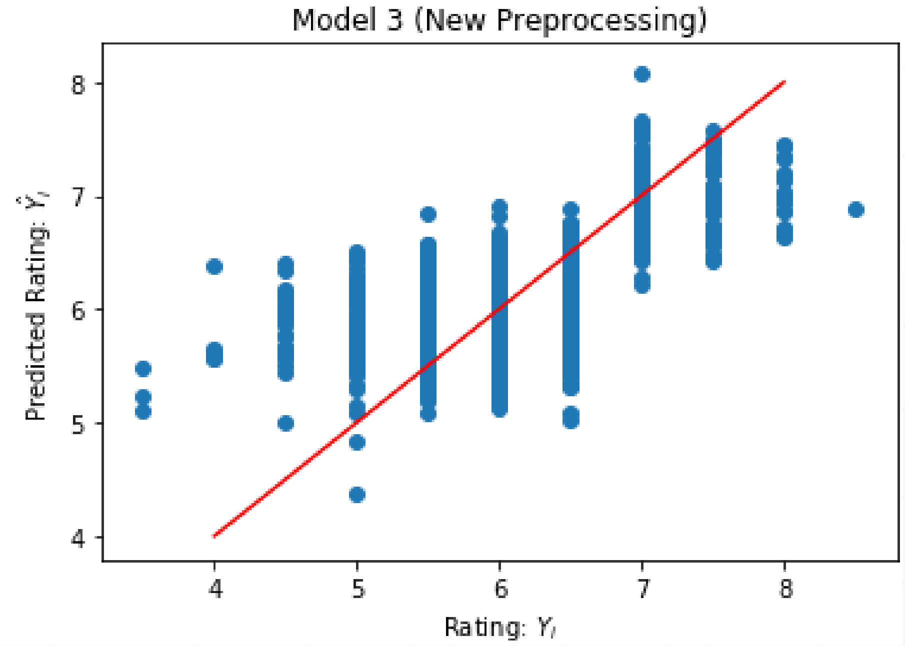
Model 3

- Adjust Model 2 to round ratings to nearest half step
  - (5.8 - 6.2 => 6.0), (6.3 - 6.7 => 6.5), (6.8 - 7.2 => 7.0), etc.
  - Reduces number of possible ratings
- Find the BEST Actors / Directors / Production Companies
  - BEST: Ratings >= 7
- Check performance

Model 3 (2)

- Slightly better than Model 2
- Worse than Model 1: Top 500 Metric

```
R² (Training): 0.46484232997218944
R² (Test): 0.4313814685423739
MSE: 0.3462865612685848
```



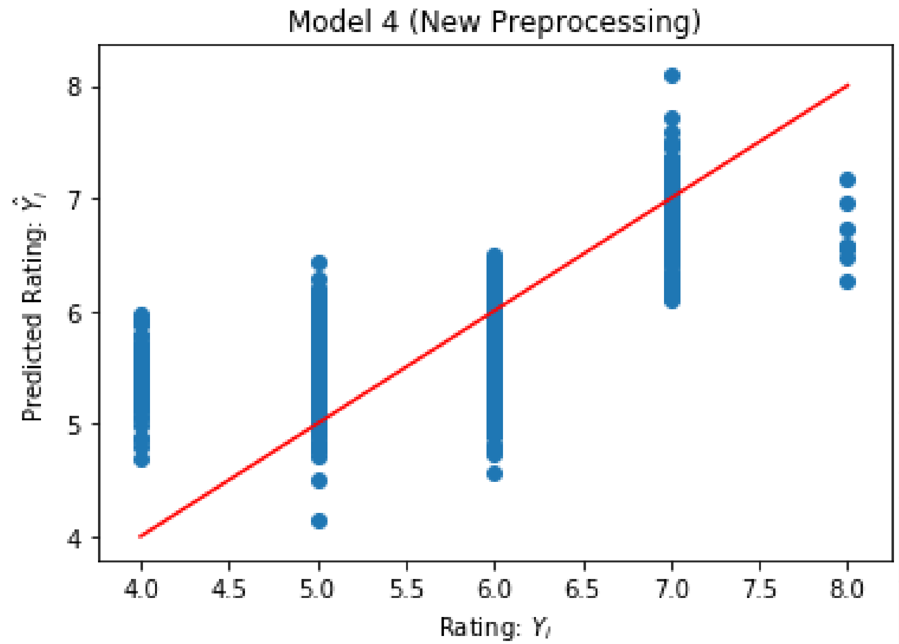
Model 4

- Adjust Model 3 to bin ratings instead of rounding
  - (0 – 4.9 => 5.0), (5.0 – 5.9 => 5.0), (6.0 – 6.9 => 6.0), etc.
  - Further reduces number of possible ratings
- Find the BEST Actors / Directors / Production Companies
  - BEST: Ratings >= 7
- Check performance

Model 4 (2)

- Slightly better than Model 2
- Slightly worse than Model 3
- Worse than Model 1: Top 500

```
R² (Training): 0.45101181710641347
R² (Test): 0.41665319122511096
MSE: 0.3884578272687888
```



Model 5

- Models 2, 3, and 4 used the same metric to build the main feature columns:
  - Actors / Directors / Production Companies
- Final preprocessing adjustment
  - Take the idea from Model 4 to bin the ratings
  - Create list of Actors / Directors / Production Companies for each genre and bin
  - Create one feature column per bin

Model 5

Feature Columns (Example)

- Movie categorized under the genres of Action and Adventure
- Rating Bins: 4, 5, 6, 7, 8
  - 5 total bins => 5 columns per feature
- Actors Feature Columns:
  - Count the number of actors listed for the Action genre for each bin
    - Two in Bin 5, one in Bin 6
  - Count the number of actors listed for the Adventure genre for each bin
    - Three in Bin 6
  - Sum bins

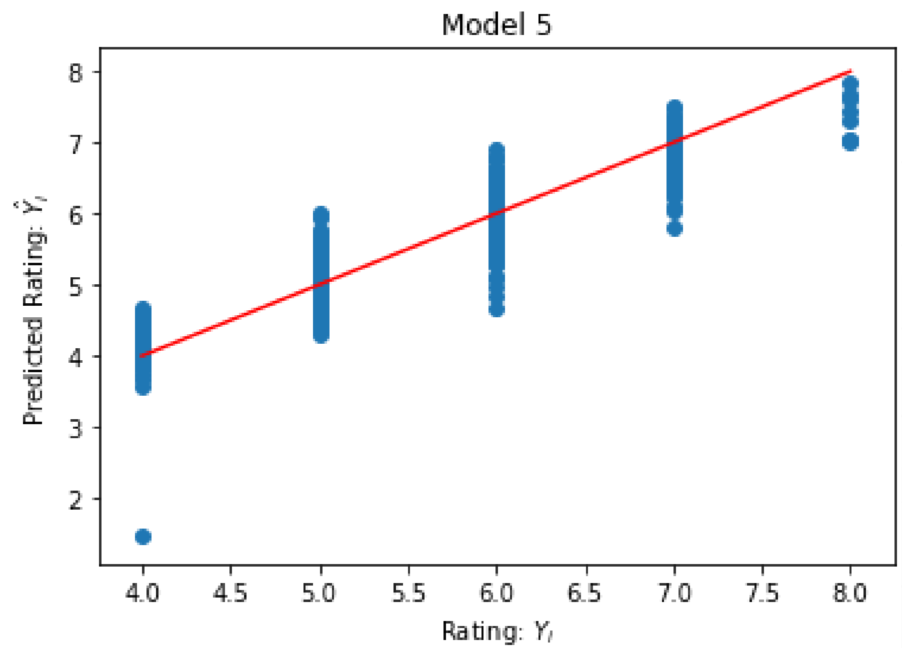
Bin 4: 0	**Bin 5: 2**	**Bin 6: 3**	**Bin 7: 0**	**Bin 8: 0**

- Repeat process for Directors and Production Companies

Model 5 (3)

- Great performance!
- Poor predictions for ratings of 8
- Error for each rating ~ 1
- May perform better as classifier

```
R² (Training): 0.8629151408785175
R² (Test): 0.8423997321430676
MSE: 0.10494796012899413
```



Classification

Convert model to a classifier

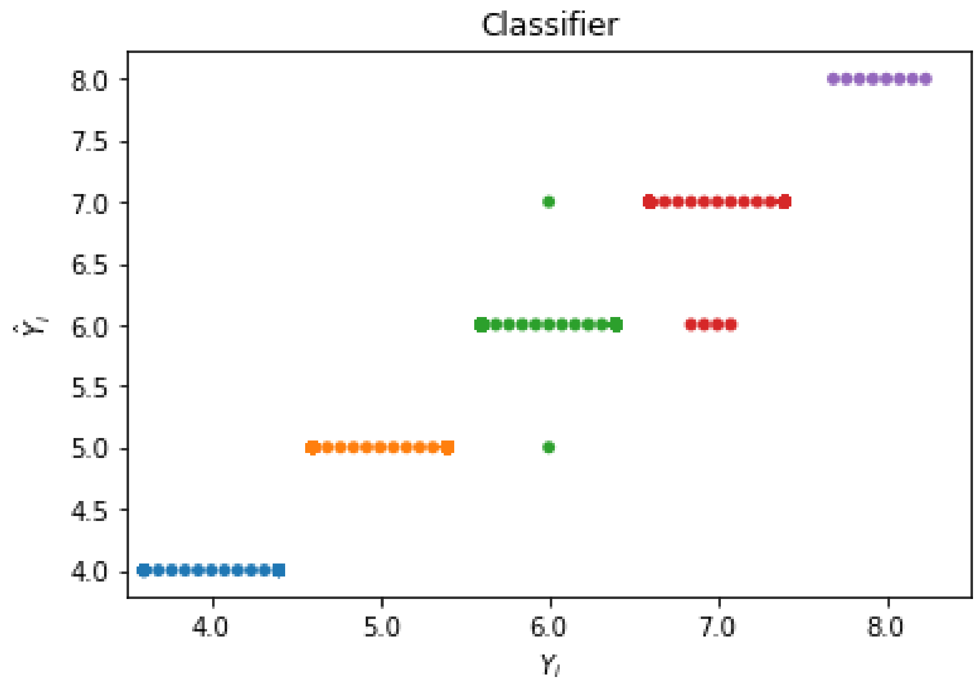
Classifier

- Turn Model 5 into a classifier
  - Use the same features
- Actors / Directors / Production Companies lists are binned by ratings
  - Classification may perform better

Classifier

- Increased performance!
- Can properly classify ratings of 8
- Much better accuracy
- Discrete predictions

```
Accuracy (Training): 0.9996494917630564
Accuracy (Test): 0.9936974789915967
```



Conclusions

Obvious Correlations

- Two finalized models
  - Model 5
  - Classifier
- Model 5 is less accurate
  - BUT continuous predictions (more flexible)
  - Good for less accurate, fine predictions, within a range
- Classifier more accurate
  - BUT discrete predictions (more strict)
  - Good for more accurate gradings