Building a Regression Model on Movies Dataset to Predict Profitability

Aaron Luo, Abhinav Rana, Adam Casper, & Rongrong Liu

COGS 109: Data Modelling and Analysis

Abstract:

Recently, theater attendance rate in the US has been dropping at a worrying rate, with 2017 box office returns hitting around $11 billion. This has been the lowest since 1995 (adjusted for inflation). Ticket rates have been higher than ever in an effort to make up for this disparity. Because of this, we are looking for ways to reinvigorate American interest in the theater by using linear regression to find the biggest contributing factors to box office sales.

Introduction:

For this project, we used the UCI movie dataset tracking films released between 2014 and 2015. With this dataset, we are looking to answer questions such as: "Is there a relationship between budget and online attention (likes, dislikes, comments)", or "Was 2015 a more profitable year for movies than 2014?". Ultimately, we are attempting to answer the question "Which individual attributes or combination of attributes have the greatest correlation with a movie's gross box office return rate?"

Data Set Information:

The original dataset contains 231 samples and 12 attributes. For the purposes of this project, we only used 9 of the attributes provided in the dataset. We discarded the sentiment, sequel, and genre attributes because they were either impossible to interpret or provided little actual information.

Attribute Information: (you include the attributes you used for building up the model and the variable you used in the code or in the model)

Attribute	Explanation	Variable used

Ratings	User reviews and rating taken from IMDB	R
Budget	Amount spent on movie	B
Views	# of views on movie related content (IMDB, Youtube, Twitter)	V
Likes	# of likes on movie related content (Youtube, Twitter)	L
Dislikes	# of views on movie related content (Youtube, Twitter)	D
Aggregate followers	# of followers on movie related content (Youtube, Twitter)	AF
Comments	# of comments on movie related content (IMDB, Youtube, Twitter)	C
Year	Year movie was released	Yr
Screens	# of	Sc

Research Question:

What individual attribute or combination of attributes would best predict movie's gross box office return rates?

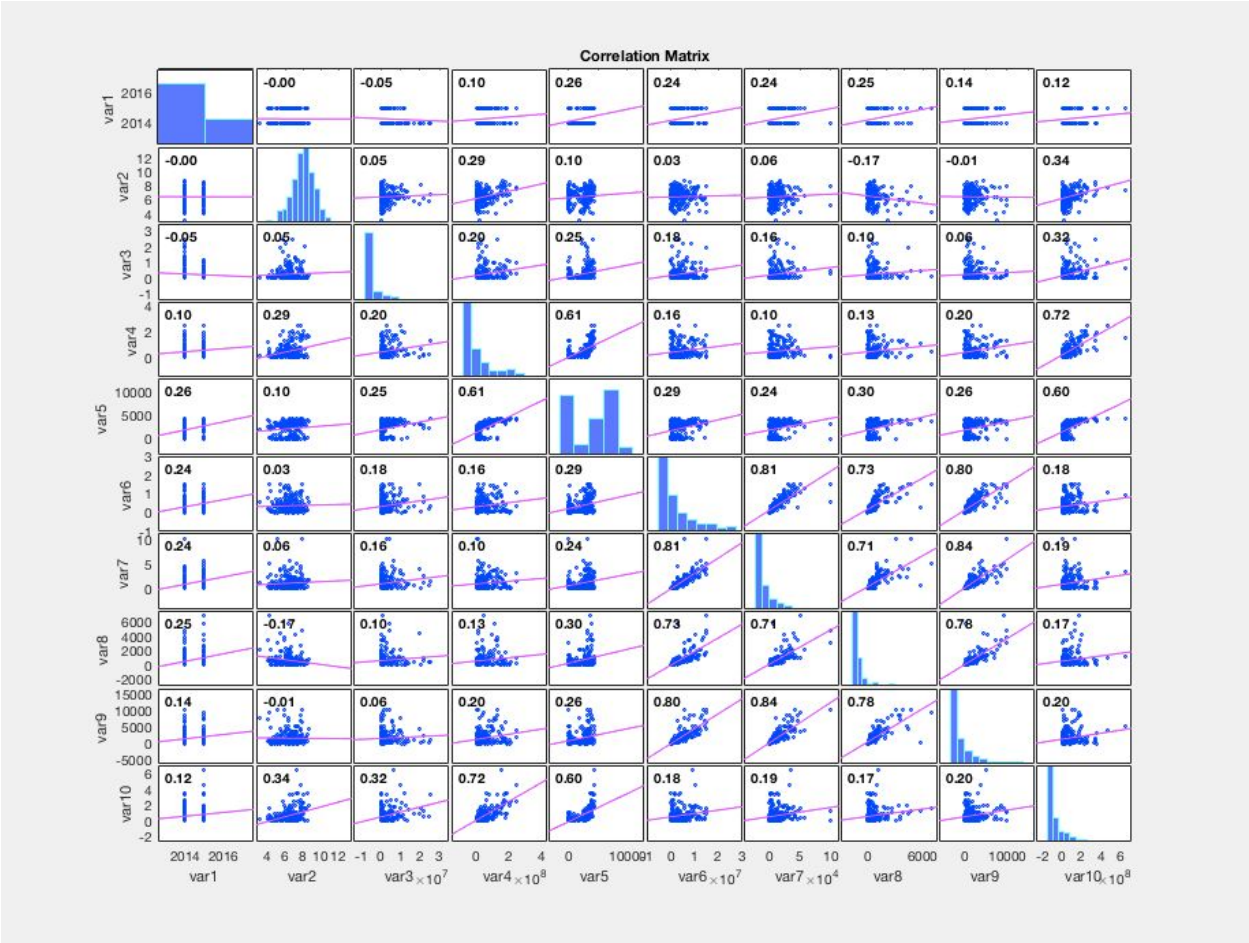
Methodology:

We used a linear regression model with our detailed procedure, as follows:

1. Clean the data by discarding attributes that we decided not to use, namely, sentiment, genre, and sequel.
2. Separate the X and Y by moving the gross column to the end of the movie data matrix.
3. Normalize the X and Y matrices using range normalization so that all the data points are weighted equally.
4. Shuffle both datasets to eliminate potential year-biased cross validation.
5. Split the datasets using the ratio 80:20 for Training:Testing data sets for cross validation.
6. **Train 5 models with 10 iterations of randomly selected datasets.**

7. Plot the average real vs prediction value of testing samples.
8. Calculate average SSE for all 10 iterations.

Regression Models:



(How we analyzed the correlation of gross and other attributes)

Before performing testing, we did pre-assessment of model. It was tested that linear models that utilize all 9 attributes had smaller SSE compared with models using part of the attributes.

Then we trained 5 models below on our dataset to observe and compare their performances. We expected that budget(B) has the greatest influence on the gross according to our graph of correlation matrix. In that case, we expected model 4 or 5 has the best performance on our testing data.

M1: Y = YrW0 + RW1 + AFW2 + BW3 + ScW4 + VW5 + LW6 + DW7 + CW8

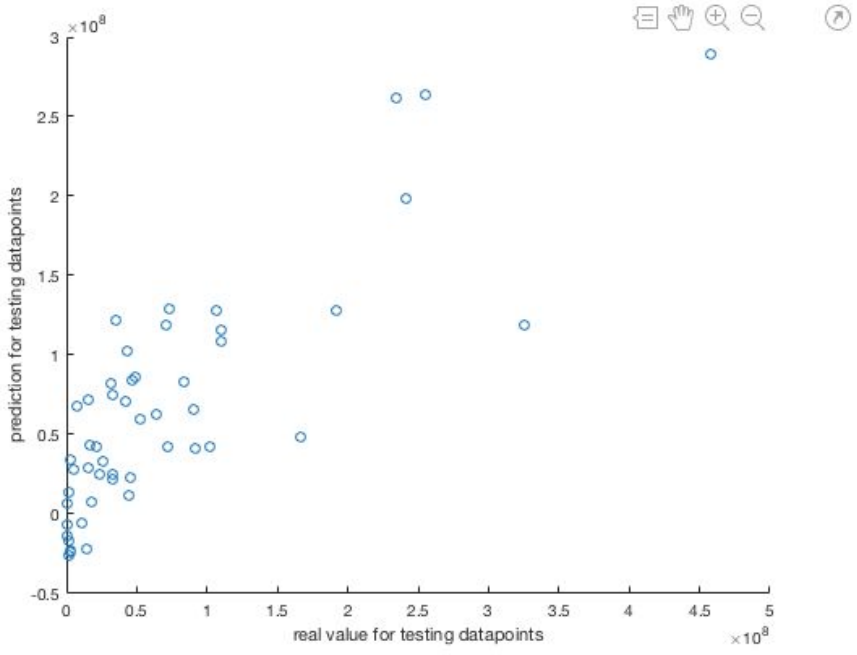
M2: Y = YrW0 + R^2 W1 + AFW2 + BW3 + ScW4 + VW5 + LW6 + DW7 + CW8

M3: Y = YrW0 + R^2 W1 + AF^3 W2 + BW3 + ScW4 + VW5 + LW6 + DW7 + CW8

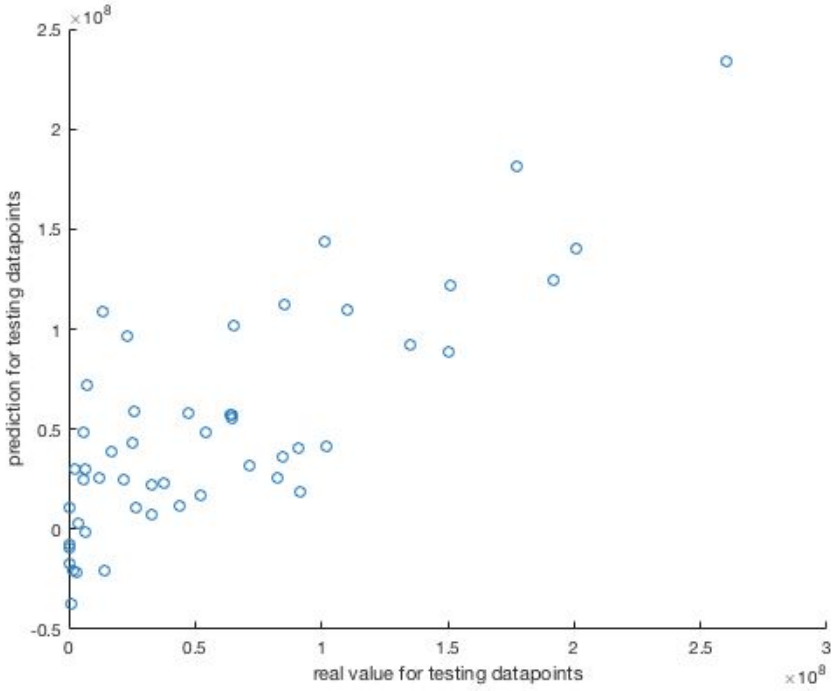
M4: Y = YrW0 + R^2 W1 + AF^3 W2 + B^4 W3 + ScW4 + VW5 + LW6 + DW7 + CW8

M5: Y = YrW0 + R^2 W1 + AF^3 W2 + B^4 W3 + Sc W4 + VW5 + LW6 + DW7 + CW8

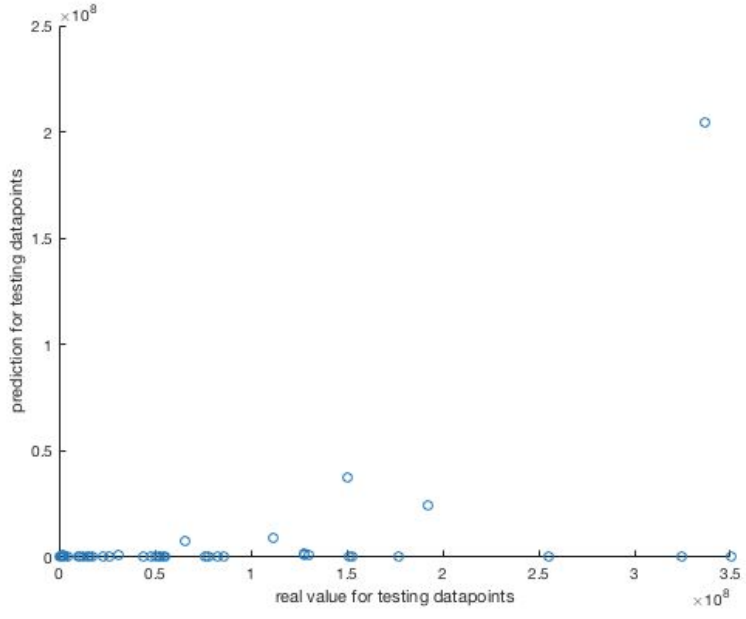
Error/Results: (SSE for the models and cross-validation for Linear Regression)



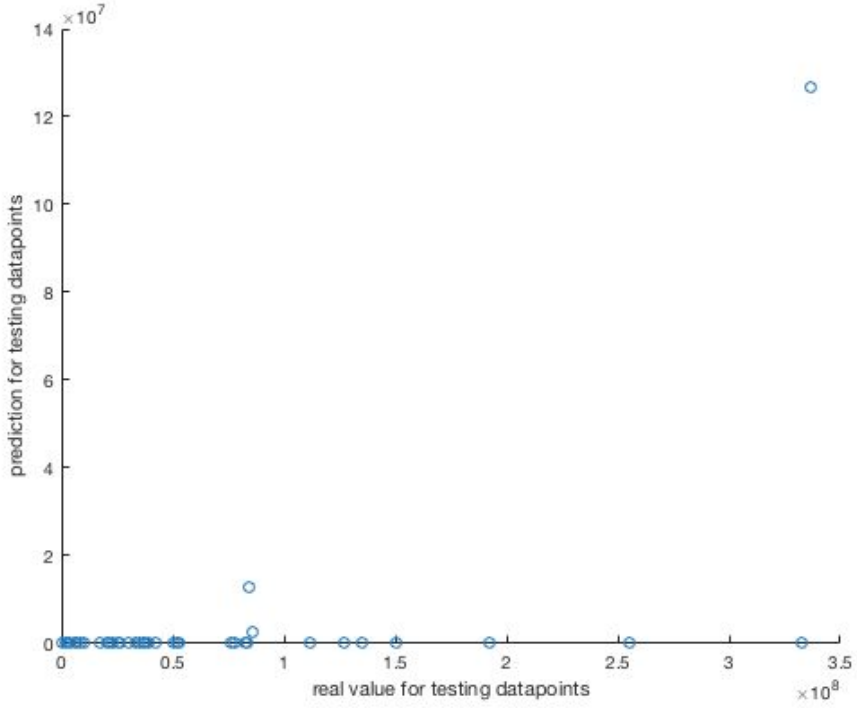

```
error1 = 1.5070e+19
```



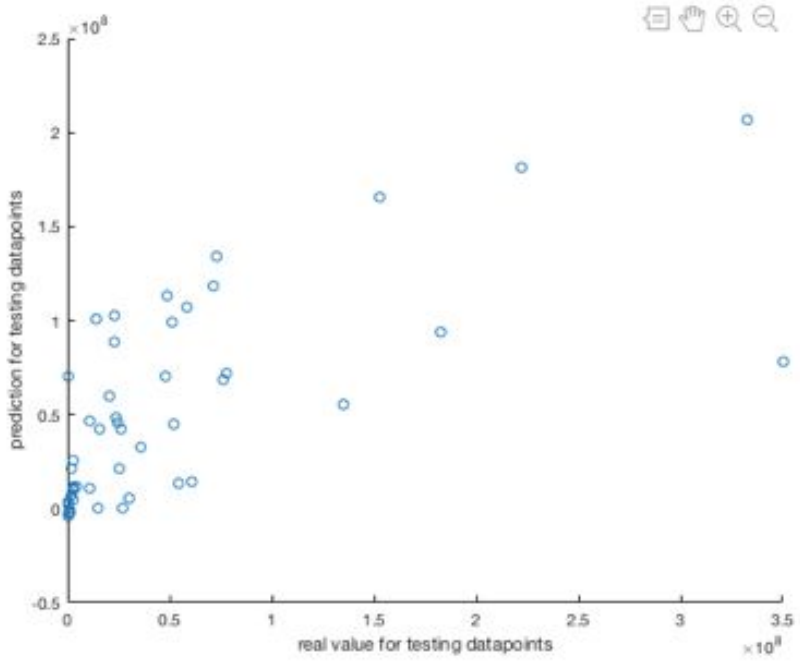

```
error2 = 1.1769e+18
```



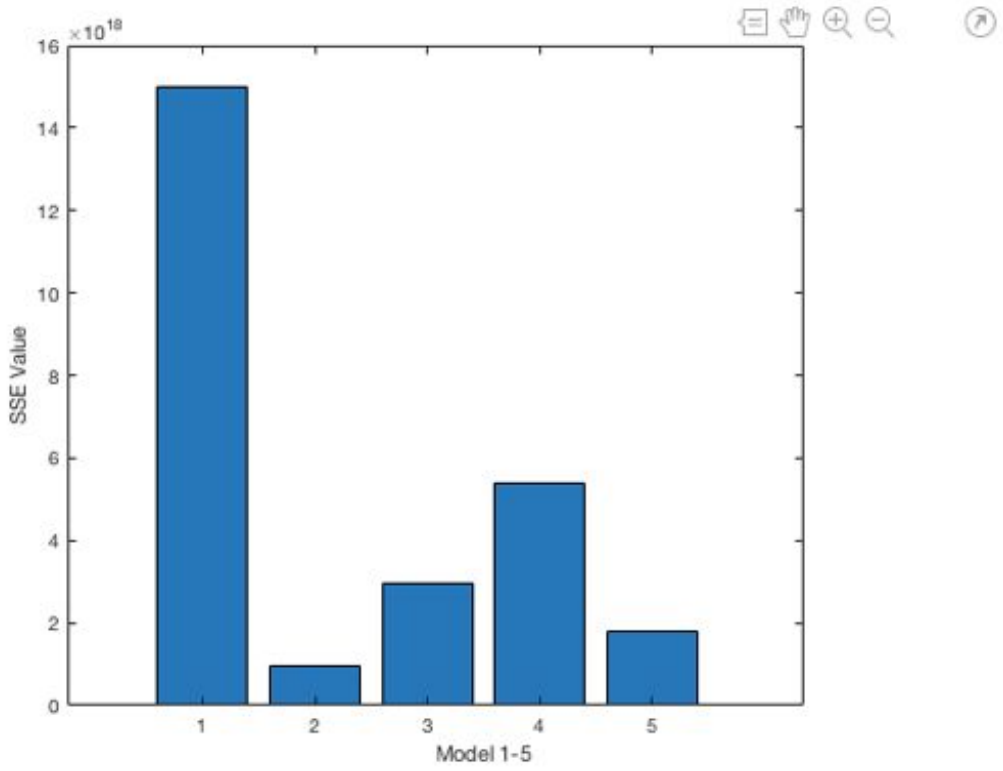

```
error3 = 4.9637e+18
```




```
error4 = 6.2380e+18
```




```
error5 = 1.7752e+18
```



After we calculated SSE, we found that model 1 had the highest SSE for testing data points. From the visualization of results of model 3 and 4, they barely did any prediction. It could be the case that the power is so high and over-fit the training datasets. Model 2 gave the best performance; however, it works poorly for testing data points which have low real value. We didn't expect model 1 to have such a high SSE; however, from the graph, it works better than model 3 and 4. It could be the case that model 1 didn't work for data points that have larger real value; therefore, the model was punished greatly.

Conclusion: (Was the research question answered? How well the model was constructed and any real life application of the model that can be used)

Our research question is answered. With all nine attributes combined and squaring the ratings gave out the best prediction of gross box office.

Based on our data, the correlation matrix found that ratings, budget, and screens had the strongest relationship with our gross output. However, upon further verification we found that a separate set of attributes, namely rating, followers, likes, and dislikes had a smaller SSE that of

the “correlated” data. However, the combination of all nine attributes always gave out the best performance.

The reason why the followers, likes, and dislikes attributes seem to rather accurately predict gross box office returns is that these three are a general measurement of how much attention the movie is getting even before it gets released, and therefore can be extrapolated to be an indication of the movie’s advertising and marketability, both of which are factors that directly affect the profitability of a feature film. The high relevance of the ratings attribute can largely attributed to the growing audience reliance on review aggregator Rotten Tomatoes, a website that has grown so influential that most movies will feature their positive Rotten Tomato score in their advertising. Apart from this, reviews also directly affect the marketability of a movie, as poor ratings can turn off many potential audiences while good ones can spark interest in audiences that otherwise would not have existed.

References:

https://archive.ics.uci.edu/ml/datasets/CSM+%28Conventional+and+Social+Media+Movies%29+Dataset+2014+and+2015

Appendix: [Commented Code]

```
%read in excel file to matrix

moviedata = xlsread('movie_data.xlsx');

%express so that is not in scientific notation

%format bank


%move 'gross' to the end and removing top index row

moviedata=moviedata(2:end,[1,2,10,4,5,6,7,8,9,3]);

%save to file

%save('moviedata.mat','moviedata');
```

```
%normalize data

%data = normalize(moviedata,'range');

data = moviedata;


%corrplot(data,'varNames',{'Year','Ratings','Followers','Budget','Screens','Views','Likes','Dislikes
','Comments','Gross'})

data = data(:,[1,2,3,4,5,6,7,8,9,10]);

%model1 >> do nothing to attributes

model1 = data;

%moded2 >> square second attribute

model2 = data;

model2(:,2) = model2(:,2).^2;


%model3 >> square second attribute and cube third

model3 = data;

model3(:,2) = model3(:,2).^2;

model3(:,3) = model3(:,3).^3;

%model4

model4 = data;

model4(:,2) = model4(:,2).^2;

model4(:,3) = model4(:,3).^3;

model4(:,4) = model4(:,3).^4;

%model5

model5 = data;
```

```
model5(:,2) = model5(:,2).^2;
model5(:,3) = model5(:,3).^3;
model5(:,4) = model5(:,4).^4;
model5(:,5) = model5(:,5).^5;



[error1] = model_test(model1,100,0.80)
[error2] = model_test(model2,10,0.80)
[error3] = model_test(model3,10,0.80)
[error4] = model_test(model4,10,0.80)
[error5] = model_test(model5,10,0.80)
bar([error1;error2;error3;error4;error5])

function [tot_SSE,avg_acc,squared_error,acc] = model_test(data,n_iter,train_percent)
% what percentage of data do you want to train



train_idx = round(size(data,1).*train_percent);
pred_results = zeros((size(data,1)-train_idx).*n_iter,1);
test_results = zeros((size(data,1)-train_idx).*n_iter,1);
error_results = zeros((size(data,1)-train_idx).*n_iter,1);
squared_error = zeros((size(data,1)-train_idx).*n_iter,1);
acc = zeros((size(data,1)-train_idx).*n_iter,1);
error = 0;
```

```
%adding ones column to begining of matrix and then separating into X and Y arrays
onecol = ones(size(data,1),1);
X = [onecol data(:,1:end-1)];
Y = data(:,end);
j=1;

for i=1:n_iter
% randomize rows in training data
% -> needed to convert 1 x 230 to 230 x 1
rand_idx = (randperm(size(data, 1))).';

Xrand = X(rand_idx,:);
Yrand = Y(rand_idx,:);

%take the top 80 percent of your randomized 'X' and 'Y' dataset
Xtrain = Xrand(1:train_idx,:);
Ytrain = Yrand(1:train_idx,:);
Xtest = Xrand(train_idx+1:end,:);
Ytest = Yrand(train_idx+1:end,:);
w = Xtrain\Ytrain;

   for i = 1:size(Xtest)

   % calculate predicted gross and save into column of pred_results matrix
```

```
    pred = Xtest(i,:)*w;
    pred_results(j) = pred;
    % stores the actual test results that will match the columns of pred_results
    test = Ytest(i);
    test_results(j) = test;
    %calulate error
    error = (test - pred).^2;
    squared_error(j) = error;
    sqrt_err = sqrt(error);
    difference = abs(test-pred);%changed sqrt_error into pred
    acc(j) = difference/test;% change matrix division into number division
    error_results(j) = sqrt_err;


    j = j+1;
    end
end


tot_SSE = sum(squared_error);
err = sqrt(tot_SSE);
tot_test = sum(test_results);
per_acc = err./tot_test;
acc1 = 1-per_acc;
avg_acc = mean(acc);
```

```
scatter(test_results(1:50),pred_results(1:50))
ylabel("prediction of testing data")
xlabel('real value of testing data')



end
```