Predicting the Financial Success of Will Ferrell Movies

Nicholas Mankowski

December 13, 2023

In the competitive and unpredictable world of film, Will Ferrell has managed to build a vast portfolio of movies across many genres, creating multiple box-office hits in the process. But what really makes a Will Ferrell movie successful? In this paper, we will be predicting the success of a Will Ferrell movie measured in final box office sales. We will use IMDb ratings, Rotten Tomatoes Scores, Will Ferrell's age, as well as the production cost of the movie to predict this success. Parsing data from the Rotten Tomatoes site dedicated to Will Ferrell, we will randomly select 10 distinct movies to pull data from. Using the corresponding data from these 10 movies, we will build 5 models to predict the total box office sales of future movies casting Will Ferrell. Comparing the outcomes predicted by these models to actual data, we will also analyze their effective at predicting financial success of movies casting Will Ferrell.

1 Will Ferrell Movie Data

1.1 Means of Data Collection

The movies were picked from a list of Will Ferrell's movies in order of their Rotten Tomatoes score. A Python script was developed to extract movie data—including Movie Title, Year, and Rotten Tomatoes Score—from the HTML of the corresponding website. From this list of movies, the script picks 10 distinct and random movies that were released in or after 2000. From here, the production cost of the movie, IMDb rating, as well as final box office sales were found from the IMDb website. If no data was found, the movie was substituted for another randomly pulled movie pulled from the list that does have the aforementioned data included. The dollar amounts listed were then adjusted for inflation and listed in millions and the age of Will Ferrell at the time of release was calculated.

1.2 Raw Will Ferrell Movie Data

Table 1: Rotten Tomatoes Score, IMDb Rating, and Age of Will Ferrell at Time of Movie Release.

Title	Rotten Tomatoes Score	IMDb Rating	Will Ferrell Age
Daddy's Home 2	21%	6	50
Holmes & Watson	10%	3.9	51
Bewitched	24%	4.8	38
The LEGO Movie 2: The Second Part	84%	6.6	52
The Other Guys	79%	6.6	43
Anchorman: The Legend of Ron Burgundy	66%	7.1	37
Zoolander	65%	6.5	34
Kicking & Screaming	41%	5.6	38
Semi-Pro	23%	5.8	41
Get Hard	28%	6	48

Table 2: Production Cost and Final Box Office Sales in Millions and Adjusted for Inflation

Title	**Production Cost(nillions)**	**Final Box Office Sales(millions)**
Daddy's Home 2	\$86.05	\$225.24
Holmes & Watson	\$51.13	\$49.26
Bewitched	\$133.05	\$205.71
The LEGO Movie 2: The Second Part	\$118.38	\$238.67
The Other Guys	\$140.19	\$238.97
Anchorman: The Legend of Ron Burgundy	\$42.07	\$146.79
Zoolander	\$45.31	\$98.36
Kicking & Screaming	\$70.44	\$87.76
Semi-Pro	\$78.09	\$62.48
Get Hard	\$51.59	\$144.08

2 Modeling Movie Data

2.1 One Variable Linear Model

2.1.1 Description of Mathematics

For our one variable linear model, we are building a linear regression model of the form

$$F = a_1 + a_2 B\,,$$

where F is the final box office sales of a movie, B is the production cost of said movie, and a_1, a_2 are coefficients that minimize the error of our regression line. We can describe the error of our model at each point by looking at the difference in expected result vs. actual result, which yields

$$E_i = F_i - a_1 - a_2 B_i\,,$$

where F_i is the final box office sale of an individual movie, B_i is the budget of said movie, and E_i is the error in our prediction.

Furthermore, we can sum the squared difference of each of the n movie results to produce the ordinary least squares method of error calculation, which we can write as a function of our coefficients a_1 & a_2:

$$E(a_1, a_2) = \sum_{i=1}^{n} (F_i - a_1 - a_2 B_i)^2\,.$$

Notice that if we want to find the coefficients that minimize our value of E, we can find the partial derivatives of E with respect to a_1 as well as a_2 and find where these derivatives are equal to zero. This will give us a system of equations that we could use to solve for the a_1, a_2 that minimize E. Applying this, we get

$$\frac{\partial E(a_1, a_2)}{\partial a_1} = 0 \implies \sum_{i=1}^{n} -2(F_i - a_1 - a_2 B_i) = 0\,,$$
$$\frac{\partial E(a_1, a_2)}{\partial a_2} = 0 \implies \sum_{i=1}^{n} -2B_i(F_i - a_1 - a_2 B_i) = 0\,.$$

We can rewrite this system of equations as the following:

$$\sum_{i=1}^{n} F_i = n a_1 + a_2 \sum_{i=1}^{n} B_i$$
$$\sum_{i=1}^{n} B_i F_i = a_1 \sum_{i=1}^{n} B_i + a_2 \sum_{i=1}^{n} B_i^2\,.$$

Expressed as a matrix, we have the system of equations to be

$$\begin{bmatrix}\sum_{i=1}^n F_i \\ \sum_{i=1}^n B_iF_i\end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^n B_i \\ \sum_{i=1}^n B_i & \sum_{i=1}^n B_i^2\end{bmatrix}\begin{bmatrix}a_1 \\ a_2\end{bmatrix}.$$

Thus, we can solve for a_1 and a_2 in the above to get the coefficients that minimize the least squares error in our linear regression model.

2.1.2 Application in Code

We can follow the above steps to find the values of a_1, a_2 to minimize our least squares error using the `polyfit` function in MATLAB.

```matlab
% Get our coeficients a_1, a_2
coef = polyfit(cost, sales, 1);
```

Looking at the results of `polyfit` specifying degree `1`, we get the values

$$a_1 = 30.750329, a_2 = 1.457573\,.$$

Note: `polyfit` returns the values in order of the highest order to lowest order, so the vector `coef` will look like `[a_2, a_1]`.

Hence, we can see that we now have this explicit equation for our model:

$$F = 30.750329 + 1.457573B\,.$$

Now, we can evaluate our model by using the `polyval` function at cost b as follows:

```matlab
% Assemble our model for a given cost b
model = @(b) polyval(coef, b);
```

To build a visualization of our model, we can simulate a continuous expression of time using a small step size, and then evaluate the model at each time step.

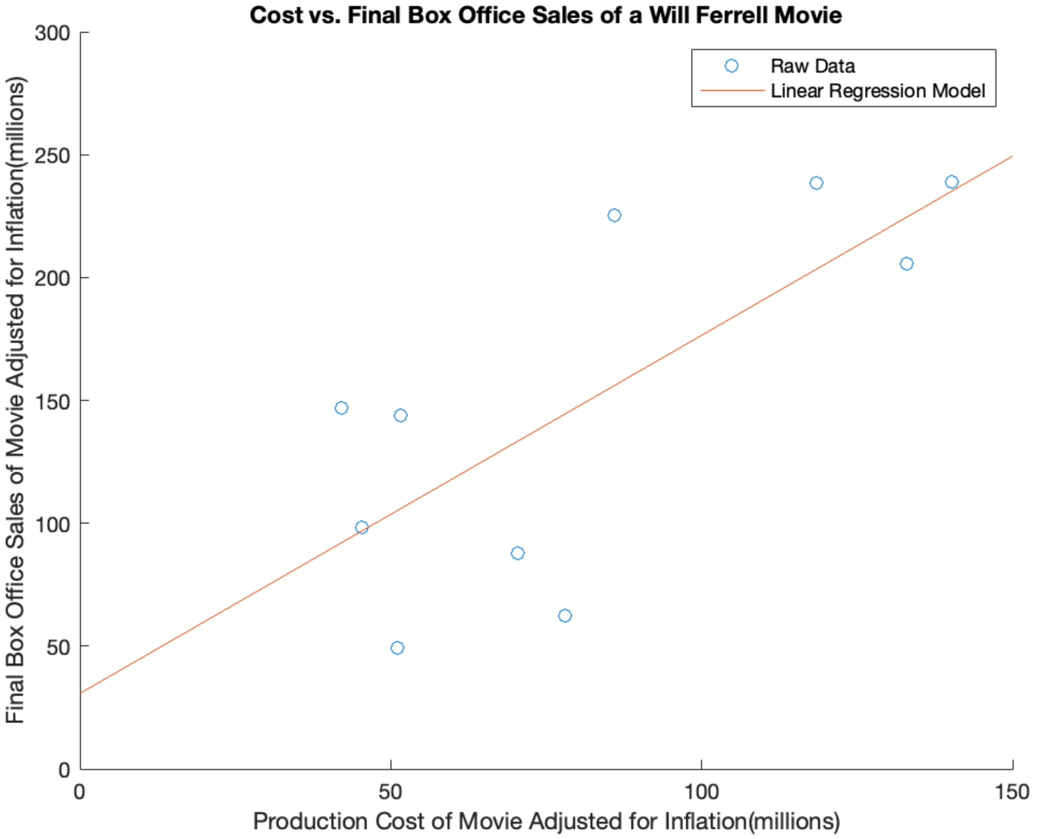
```matlab
% Create a domain for the models
timeCts = 0:0.1:150;
% Create the models
cost_visualization = model(timeCts);
```

Furthermore, we can plot our model using the following MATLAB code:

```matlab
hold on;
% Plot the raw data
scatter(cost, sales, 'DisplayName', 'Raw Data');
% Plot our model
plot(timeCts, cost_visualization, 'DisplayName', 'Linear Regression Model');
% Set limits to properly showcase our graph
xlim([0, 150]);
ylim([0, 300]);
% Label and title the graph accordingly
xlabel('Production Cost of Movie Adjusted for Inflation(millions)');
ylabel('Final Box Office Sales of Movie Adjusted for Inflation(millions)');
title('Cost vs. Final Box Office Sales of a Will Ferrell Movie');
% Show the legend
legend;
hold off;
```

2.1.3 Model Analysis

As a result of the model visualization presented in 2.1.2, we obtain the following result:



As seen above, the model seems to accurately describe the relationship between production cost and final box office sales. If we were to evaluate the error as described in 2.1.1, we can see that $E(30.750329, 1.457573) = 22808.960460$. Despite the error seeming rather high when quantified and also perhaps visually, when we look at the scale of values involved in this situation, it actually appears to be rather good. The model seems to fit our data well and trend in the correct direction. Looking further at the model, we can see that it makes intuitive sense. Our model suggests a positive correlation between production cost and final box office sales of a Will Ferrell movie. If the producers of a Will Ferrell movie invest more in quality production, they are likely to see higher box office returns.

Another interesting question is whether or not the relationship between production cost and box office sales is linear. When we think of investing in the terms of every day life, there is often an expectation of returns being compounded. This, of course, is often spread over time, so it makes intuitive and mathematical sense that our returns are non linear. This idea of investing may effect our intuition and incline us to believe the more accurate model for these data points is nonlinear, but is that really the case? This question will be explored further at a later point in this paper.

2.2 One Variable Quadratic Model

2.2.1 Description of Mathematics

Similarly, we can extend the ideas discussed in 2.1.1 to a one variable quadratic model of the form

$$F = a_1 + a_2B + a_2B^2\,.$$

We can see that the least squares error of this will now be the following function of a_1, a_2, a_3:

$$E(a_1, a_2, a_3) = \sum_{i=1}^{n}(F_i - a_1 - a_2B_i - a_3B_i^2)^2\,.$$

Hence, we can apply the ideas explored in 2.1.1 to find the $a_1, a_2,$ & a_3 that minimize E. Deriving with respect to a_1, a_2, a_3 and setting each to zero, we get

$$\begin{aligned}
\frac{\partial E(a_1,a_2,a_3)}{\partial a_1} = 0 &\implies \sum_{i=1}^{n} -2(F_i - a_1 - a_2B_i - a_3B_i^2) = 0,\\
\frac{\partial E(a_1,a_2,a_3)}{\partial a_2} = 0 &\implies \sum_{i=1}^{n} -2B_i(F_i - a_1 - a_2B_i - a_3B_i^2) = 0,\\
\frac{\partial E(a_1,a_2,a_3)}{\partial a_3} = 0 &\implies \sum_{i=1}^{n} -2B_i^2(F_i - a_1 - a_2B_i - a_3B_i^2) = 0,\ .
\end{aligned}$$

Rewriting as a matrix equation as we did in 2.1.1, we get

$$\begin{bmatrix} \sum_{i=1}^{n} F_i \\ \sum_{i=1}^{n} B_iF_i \\ \sum_{i=1}^{n} B_i^2F_i \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^{n} B_i & \sum_{i=1}^{n} \\ \sum_{i=1}^{n} B_i & \sum_{i=1}^{n} B_i^2 & \sum_{i=1}^{n} B_i^3 \\ \sum_{i=1}^{n} B_i^2 & \sum_{i=1}^{n} B_i^3 & \sum_{i=1}^{n} B_i^4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}\,.$$

Thus, solving for $a_1, a_2,$ & a_3 will find the minimum error for our quadratic regression model.

2.2.2 Application in Code

Similarly to how we solved for our regression coefficients in 2.1.2, we can use the `polyfit` function in MATLAB to find $a_1, a_2,$ & a_3 by specifying degree 2.

```
% Get our coeficients a_1, a_2, a_3
coef = polyfit(cost, sales, 2);
```

Inspecting the results of `polyfit`, we can see that we have the values

$$a_1 = 101.013325,\ a_2 = -0.388936,\ a_3 = 0.010194\,.$$

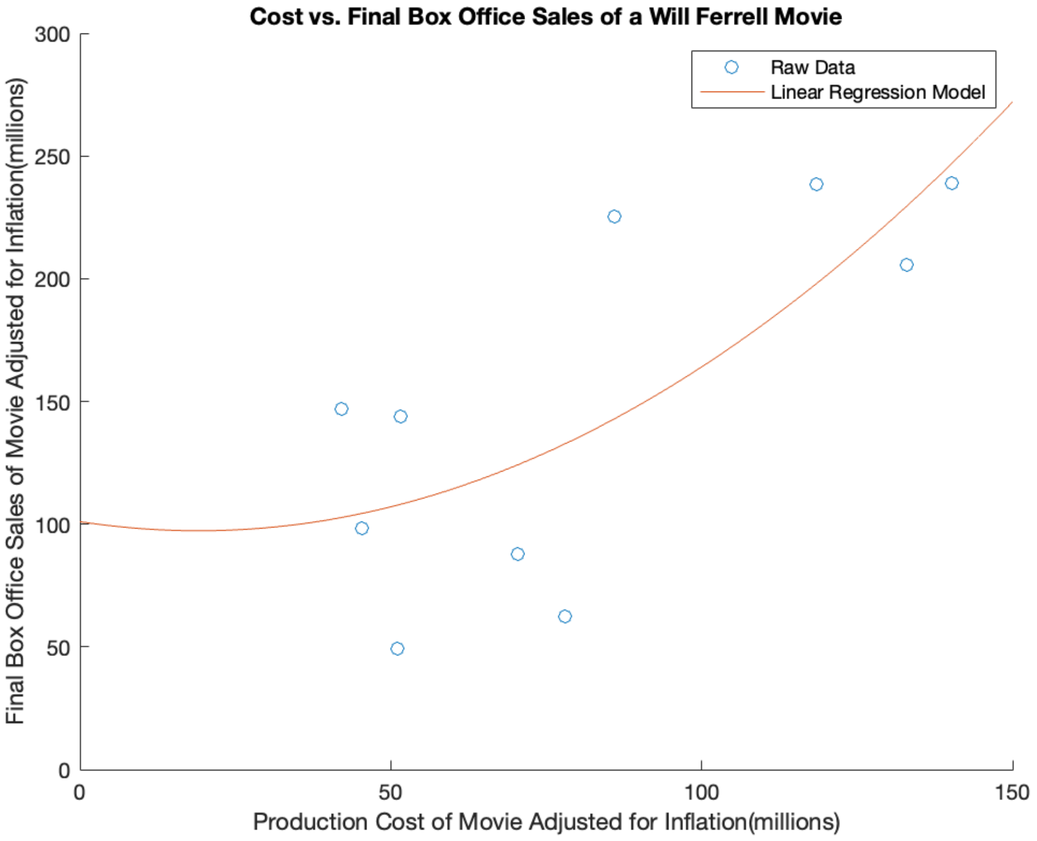
Hence, we have our explicit model to be

$$F = 101.013325 + -0.388936B + 0.010194B^2\,.$$

Now that we have our regression coefficients and our explicit model, we can assemble our model and the visualization following what we did directly in 2.1.2.

2.2.3 Model Analysis

We obtain the following model from the code described in 2.2.2.



This model seems to fit the data decently well, and if we analyze our least squares error we can see that $E(101.013325, -0.388936, 0.010194) = 22038.897241$. This error is clearly slightly less than the error presented for our linear model in 2.1.3, however, there is more to be said. If we look at the general trend of the model, there appers to be a minimum value that is greater than \$0, which does not make intuitive sense. If we are thinking about our model in terms of what it means in real life, it clearly does not make sense that a movie with a \$0 budget would have a higher box office revenue than a movie with a budget greater than \$0.

Other than this issue, our model seems to fit the data well and goes along well with the intuition discussed in 2.1.3, where we said that people may be inclined to believe that the relationship between investment and return is not a linear one. This model seems to showcase this idea fairly nicely, but it is unclear if it has a non-negligable difference in error from our linear model. Because of this and the issue with our minimum box office sales being greater than \$0, we can come to the conclusion that our linear regression model is likely a better fit for this situation than our quadratic one.

2.3 Two Variable Linear Model

2.3.1 Description of Mathematics

By solving a linear system of two variables, we are trying to build a model of the form

$$F = a_1 + a_2T + a_3B\,,$$

where F is our final box office sales prediction, T is our Rotten Tomatoes score, and B is the production cost of the film.

In order to find our regression coefficients to minimize the error of our system, we want to solve the system

$$\begin{aligned} F_1 &= a_1 + a_2T_1 + a_3B_1 \\ F_1 &= a_1 + a_2T_1 + a_3B_1 \\ &\vdots \\ F_n &= a_1 + a_2T_n + a_3B_n\,, \end{aligned}$$

where we are solving for the regression coefficients that optimize the model error for n distinct movies. Notice we can express this system as a matrix equation of the form

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & T_1 & B_1 \\ 1 & T_2 & B_2 \\ & \vdots & \\ 1 & T_n & B_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} .$$

However, notice that we may have that our matrix may not necessarily be solvable. Namely, if $n \neq 3$. So, we can left multiply both equations by the transpose of A, A^T, so that we are multiplying by a square matrix. This gives us

$$\begin{bmatrix} 1 & T_1 & B_1 \\ 1 & T_2 & B_2 \\ & \vdots & \\ 1 & T_n & B_n \end{bmatrix}^T \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & T_1 & B_1 \\ 1 & T_2 & B_2 \\ & \vdots & \\ 1 & T_n & B_n \end{bmatrix}^T \begin{bmatrix} 1 & T_1 & B_1 \\ 1 & T_2 & B_2 \\ & \vdots & \\ 1 & T_n & B_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} ,$$

which we can see is solvable, since A^TA is a square matrix. Hence, if we consider this matrix equation to be of the form $\overrightarrow{b} = A'\overrightarrow{x}$ where $A' = A^TA$, we can solve for $\overrightarrow{x}$ by doing the following:

$$\begin{aligned} \overrightarrow{b} = A'\overrightarrow{x} \implies (A')^{-1}\overrightarrow{b} &= (A')^{-1}A'\overrightarrow{x} \\ &= \mathbb{1}\overrightarrow{x} \\ &= \overrightarrow{x}\,. \end{aligned}$$

So, we have

$$\overrightarrow{x} = A^{-1}\overrightarrow{b}\,.$$

Hence, if we calculate the value of $(A')^{-1}\overrightarrow{b}$, we will have the values of our coefficients $a_1, a_2,$ & a_3.

2.3.2 Applications in Code

We are able to solve for our coefficients $a_1, a_2,$ & a_3 in MATLAB using the methods described in 2.3.1. First, we set up our A matrix and $\overrightarrow{b}$ vector using the following

```
% Set up A matrix and b vector
A = [ones(size(tomato_score)) tomato_score cost];
b = sales;
```

Now, we are able to solve for x by using the \ operation in MATLAB. This will automatically left multiply the transpose of A in order to get a solvable matrix.

```
% Solve Ax = b for x
x = A\b;
```

Inspecting the results of this operation, we have

$$a_1 = 8.723659,\ a_2 = 0.778640,\ a_3 = 1.306754\,.$$

Hence, we have our explicit model to be

$$F = 8.723659 + 0.778640T + 1.306754B\,.$$

We can assemble our model in code by doing the following:

```
% Create our model
model = @(t, b) (x(1) + x(2)*t + x(3)*b);
```

Now that we have created our model, we can build our visualization similarly to the steps taken in 2.1.2 with the only difference is that instead of creating a ”continuous” set of points, we have to build a 2-D mesh that acts this same way. We can do this using the `meshgrid` function in MATLAB.

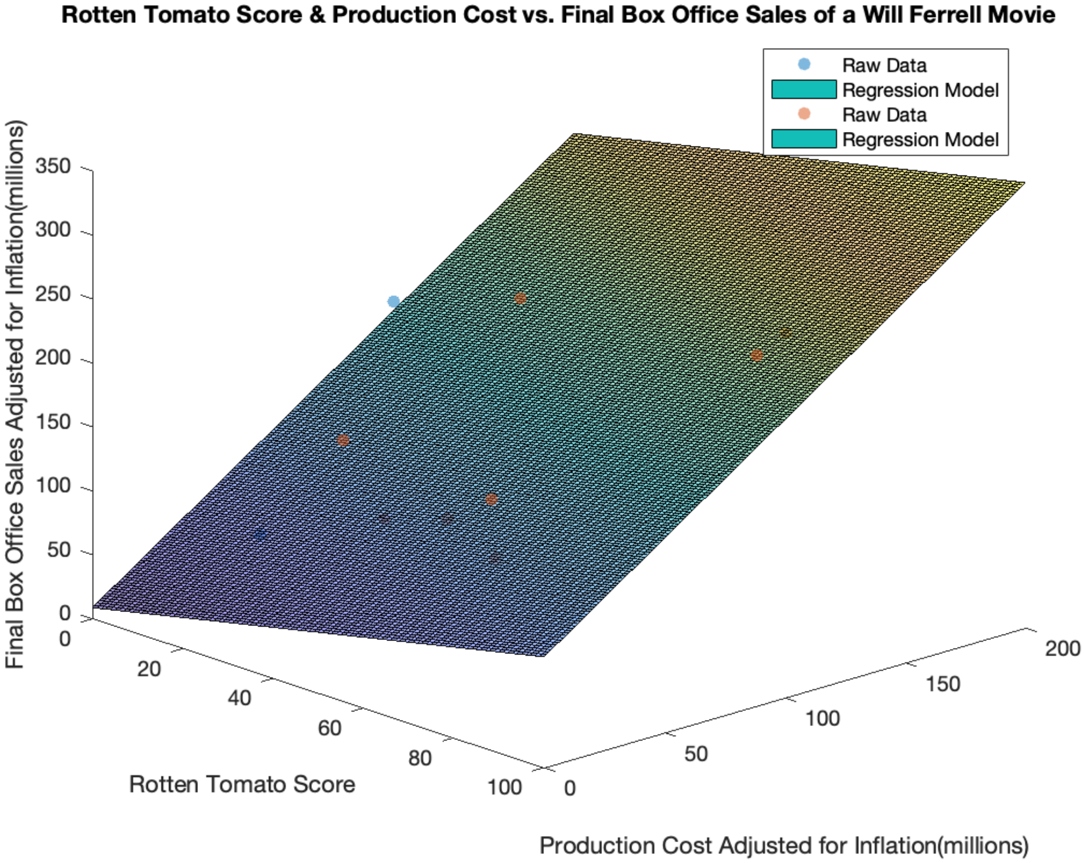
```
% Build our model visualization
[X, Y] = meshgrid(0:1:100, 0:1:200);
model_visualization = model(X, Y);
```

Similarly to what we did to plot in 2.1.2, we can plot our visualization of the model. We can use the `scatter surf` functions in MATLAB to show the model and our raw data as follows:

```
scatter3(tomato_score, cost, sales, 'filled', 'DisplayName', 'Raw Data');
surf(X, Y, model_visualization, 'DisplayName', 'Regression Model');
```

2.3.3 Model Analysis

We obtain the following model from the code in 2.3.2.



Looking at the visualization of our model, it appears to be a good fit for the presented data. The model also seems to make intuitive sense. We can see as the Rotten Tomatoes score increases, so does the anticipated final box office sales. Similarly, as the production cost increases, so does the final box office sales. We can also see from the model that the production cost seems to effect the projected final box office sales in a way that is stronger than the Rotten Tomatoes score. This is an interesting observation, as it seems to indicate that production cost is a more important factor in determing box office success than reviews. However, this may not be as good for predicting the results of an upcoming movie since there will not be a Rotten Tomatoes score for such a movie, whereas the production cost may be available.

We can evaluate the least squares error of our model to see that $E(8.723659, 0.778640, 1.306754) = 19123.077237$. Given the scale of numbers we are working with, this seems to be a fairly reasonable error. Given that our model makes clear intuitive sense and has a reasonably low least squares error, we can come to the conclusion that this model is good for predicting the final box office sales given the production cost and Rotten Tomatoes score of a Will Ferrell movie.

2.4 Two Variable Quadratic Model

2.4.1 Description of Mathematics

In creating a two variable quadratic model, we would like to build a model of the form

$$F = a_1 + a_2T + a_3B + a_4T^2 + a_5B^2 + a_6TB\,,$$

where F is our final box office sales prediction, T is our Rotten Tomatoes score, and B is the production cost of the film. We can extend the ideas discussed in 2.3.1 to set up a matrix equation that we can use to solve for our regression coefficients $a_1, a_2, a_3, a_4, a_5,$ & a_6. Thus, we have

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & T_1 & B_1 & T_1^2 & B_1^2 & T_1B_1 \\ 1 & T_2 & B_2 & T_1^2 & B_2^2 & T_2B_2 \\ & & & \vdots & & \\ 1 & T_n & B_n & T_n^2 & B_n^2 & T_nB_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} .$$

Similarly as discussed previously, if $n \neq 6$, then we see that we can solve the matrix by left multiplying the LHS and RHS by the transpose of our A matrix. Thus, we have a way to find our regression coefficients $a_1, a_2, a_3, a_4, a_5,$ & a_6.

2.4.2 Applications in Code

Similarly to what was discussed in 2.3.2, we can extend our A matrix in MATLAB to be representative of what we have in 2.4.1. Thus, in code we have the following:

```
% Set up A matrix
A = [ones(size(tomato_score)) tomato_score cost (tomato_score.^2) (cost.^2) ...
    (tomato_score.*cost)];
```

Solving $A^TA\overrightarrow{x} = A^T\overrightarrow{b}$ using the \ operator, we can see that we have

$$a_1 = -27.495717, a_2 = 0.386540, a_3 = 2.179780, a_4 = 0.016820, a_5 = -0.001396, a_6 = -0.013145 .$$

Hence, we have our explicit model to be

$$F = -27.495717 + 0.386540T + 2.179780B + 0.016820T^2 + -0.001396B^2 + -0.013145TB .$$

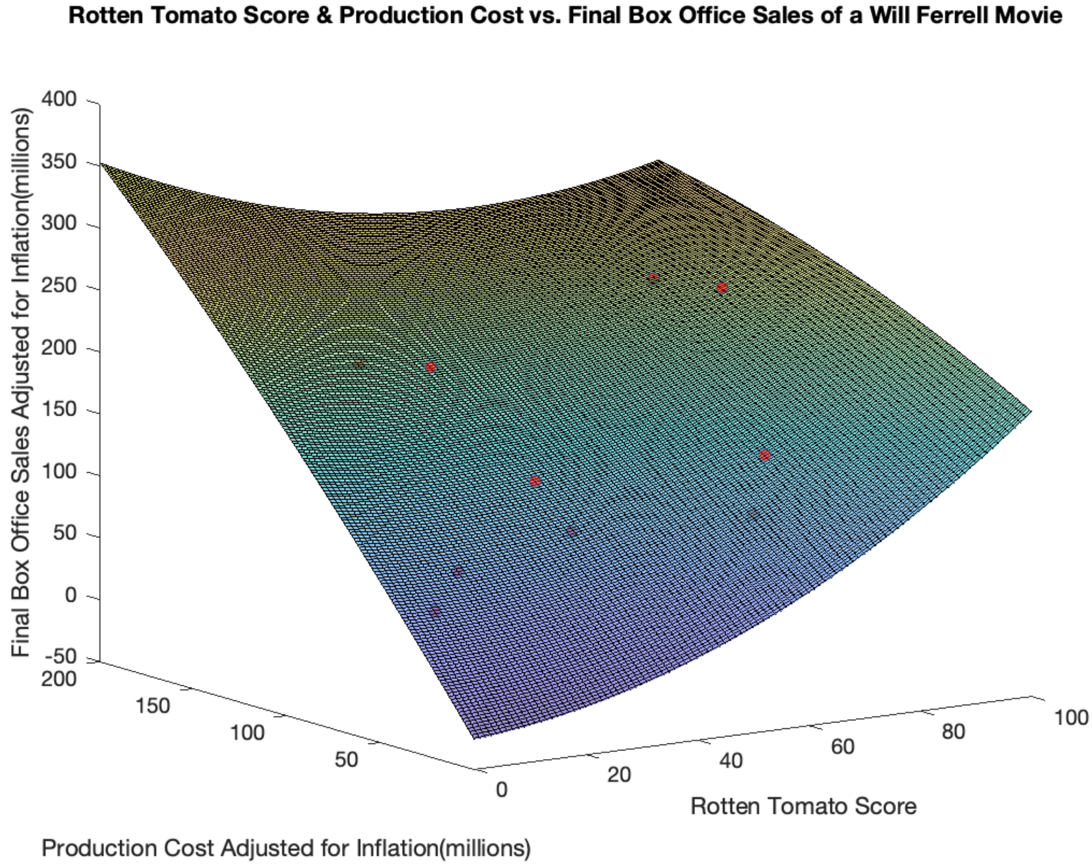
Thus, we can build our model similarly to what we had done for the 2 variable linear model.

```
% Build our model
model = @(t, b)(x(1) + x(2)*t + x(3)*b + x(4)*(t.^2) + x(5)*(b.^2) + ...
    x(6)*(t.*b));
```

We are then able to build and plot our visualization the same way we had done in 2.3.2.

2.4.3 Model Analysis

We obtain the following model from 2.4.2.



We can see that this model seems to fit the data fairly well, however it does not necessarily make intuitive sense upon further inspection. For example, if we fix a production cost of \$200 million, we see that our model predicts a higher box office sales at a Rotten Tomatoes score of 0 than a Rotten Tomatoes score of 20. If we evaluate the least squares error of our model, we can see that we have an error of 18409.715164. Despite this error being fairly low given the context of our dataset, the fact that this model does not make intuitive sense means that it is likely not the greatest representation of the overall relationship between Rotten Tomatoes score, Production Cost, and Final Box Office sales of a Will Ferrell movie.

2.5 Three Variable Linear Model

2.5.1 Description of Mathematics

In building a 3 variable linear model, we want to build an equation of the form

$$F = a_1 + a_2T + a_3W + a_4B\,,$$

where F is our final box office sales prediction, W is the age of Will Ferrell at the time of release, T is our Rotten Tomatoes score, and B is the production cost of the film.

The process for solving for our regression coefficients is quite similar to what was presented in 2.3.1. We can extend these ideas to build a matrix equation of the form $A\overrightarrow{x} = \overrightarrow{b}$. Hence, we have our matrix equation to be

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & T_1 & W_1 & B_1 \\ 1 & T_2 & W_2 & B_2 \\ & & \vdots & \\ 1 & T_n & W_n & B_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} .$$

Which we can solve for our optimal coefficients by left multiplying the transpose of our A matrix as discussed previously. Hence, we have a way to solve for our regression coefficients.

2.5.2 Applications in Code

The code used to build our model is very similar to what we had presented in 2.3.2, with the exception of two minor changes. The first change is expanding our A matrix to account for our new variable.

```
% Set up A matrix
A = [ones(size(tomato_score)) tomato_score age cost];
```

The second change is when we actually build our model. We need our equation to reflect that presented in 2.5.1. Solving our matrix equation for $\overrightarrow{x}$, we have

$$a_1 = -109.707546, a_2 = 0.953030, a_3 = 2.771033, a_4 = 1.196892\,.$$

Thus, we have our explicit model to be

$$F = -109.707546 + 0.953030T + 2.771033W + 1.196892B\,.$$

The representation of this model in code is as follows:

```
model = @(t, a, b)(x(1) + x(2)*t + x(3)*a + x(4)*b);
```

Note: We are unable to plot such a model, since it is a model of 3 dimensions and we cannot plot a 4th dimension representing the output.

2.5.3 Model Analysis

Despite not having a visual representation of our model, we can still draw some interesting conclusions regarding it. This model has a relatively high least squares error of 565319.514600. This may be due to the introduction of age. Intuitively, it does not make sense for the relationship between Will Ferrell's age and the final box office sales of his movies to be related in a linear fashion. If any intuitive conclusion was to be made, it would likely be that his box office sales would be related in a more quadratic fashion to his age. This is because you may expect an actor to "peak" at a certain age, where they end up producing the most commercially successful movies. Though it is worth noting that this is most definitely a massive oversimplification of this relationship. Due to the clear unintuitive nature and high error of this model, it is likely not a great one to predict the success of a Will Ferrell movie.

3 Summary and Conclusions

Following our discussion and analysis of the 5 presented models, we can conclude that the best model is likely the two variable linear model. This model, despite not having the lowest error, made the most intuitive sense among the models presented. As discussed in our analysis, the model depended strongly on the production cost of said movie. While the Rotten Tomatoes score did not play as important of a role as production cost, it had an important effect on our overall model and the regression coefficients for these followed general intuition.

We can use our model to predict the success of the movie *Strays*(2023). Knowing that *Strays* had a budget of \$46 million dollars and with a Rotten Tomatoes score of 54%, we can plug these values into the model,

$$F = 8.723659 + 0.778640T + 1.306754B\,,$$

to see a predicted final box office result of \$110.88 million USD.