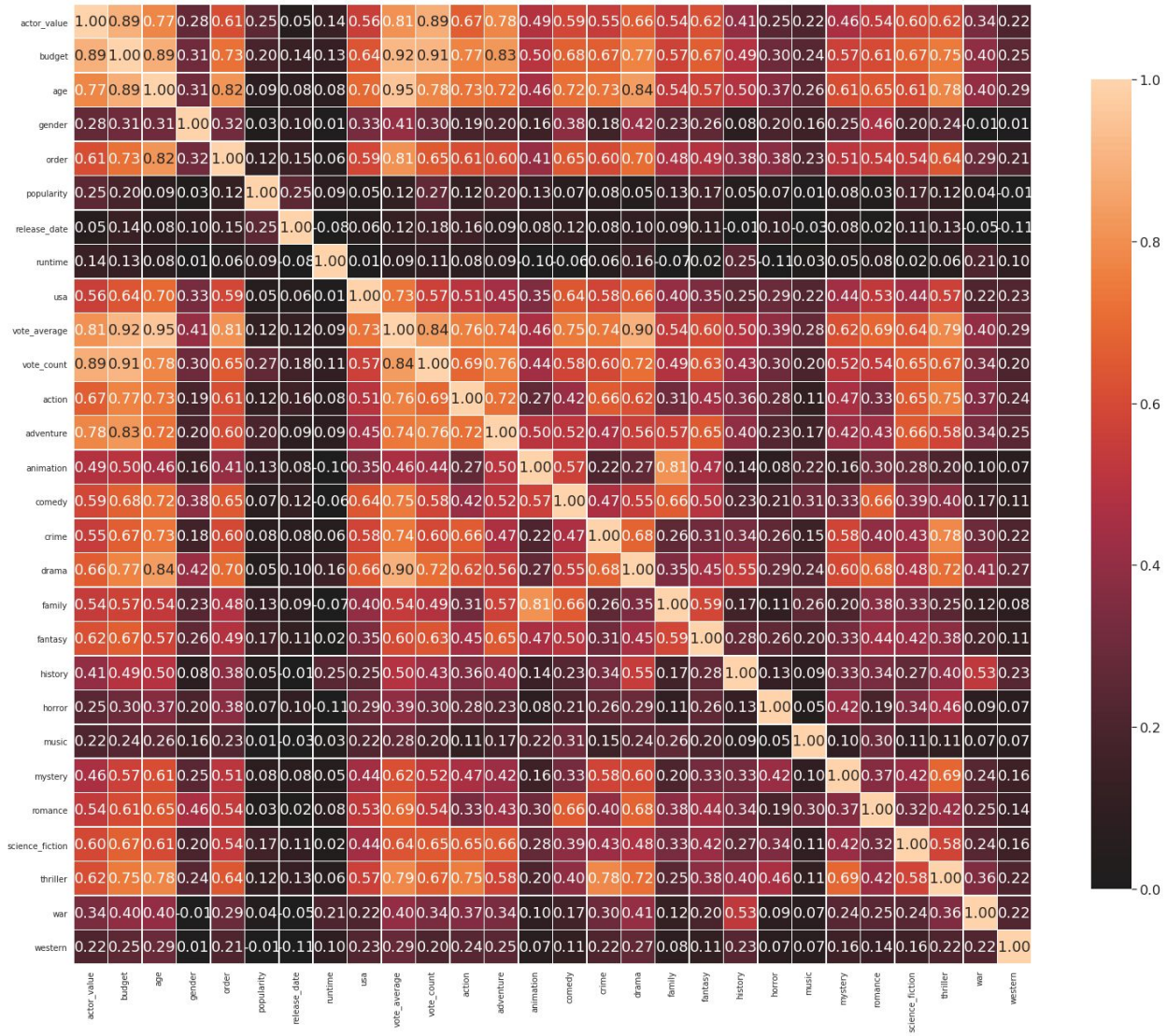
Capstone 1: In-Depth Analysis

While most studies of cinema data have been movie focused, with this project, I attempted to investigate the same information with an actor centric view. Having requested the data for this project through the API of a movie database, most of the effort expended for building an accurate and generalizable regressor model was put into data extraction and munging and feature engineering. Several features were created from the transformation and amalgamation of the existing data, as it was received from the TMDb database. This was necessary, because most cinematic metadata available described attributes of a movie, not of an actor. For example, each movie was categorized with a combination of different genres. I sifted out the individual genres for each movie and grouped them by the actors who played in those movies. An actor feature was created for each of the 17 movie genres. The values for these features represented the sums of an actor's appearances under each genre. This would allow algorithms to select similar actors based on their appearances in common styles of film. As for the movie reviews data, average rating and total number of ratings, I converted these values into actor features in a similar manner. In addition, I used the average runtimes for the actors' films as attributes of the actors themselves. Also, the actor birthdates were aligned with the movie release dates to give values for the ages of the actors on the days the movies came out. Finally, monetary features were created using the revenue and budget numbers of the movies. The resulting features represented the total dollar amounts that were associated with the films which comprised each actor's lifetime work. The monetary feature associated with the movies' revenues became the target variable. Needless to say, this type of feature engineering came with a bit of uncertainty compared to the simple extraction of toy datasets that could have been easily obtained from movie data repositories such as Kaggle, IMDb, TMDb, Rotten Tomatoes, and Movie Lens, among others. I was determined to deliver a new spin on the old movie recommender system trope. I felt that acquiring fresh data with a unique focus that required novel features to obtain success would be well received and greatly appreciated.

The decision of limiting my available tools to only using a linear regression model to make predictions on this dataset defined the goal for this analysis. I had to find the best way to represent my data in order to optimize the chosen model. Linear regression models tended to be less robust to messy data than other regressors, such as random forests and gradient boosters. To choose linear regression over other regression models required diving deeper into the dataset to understand the relation of the target variable to each predictor.

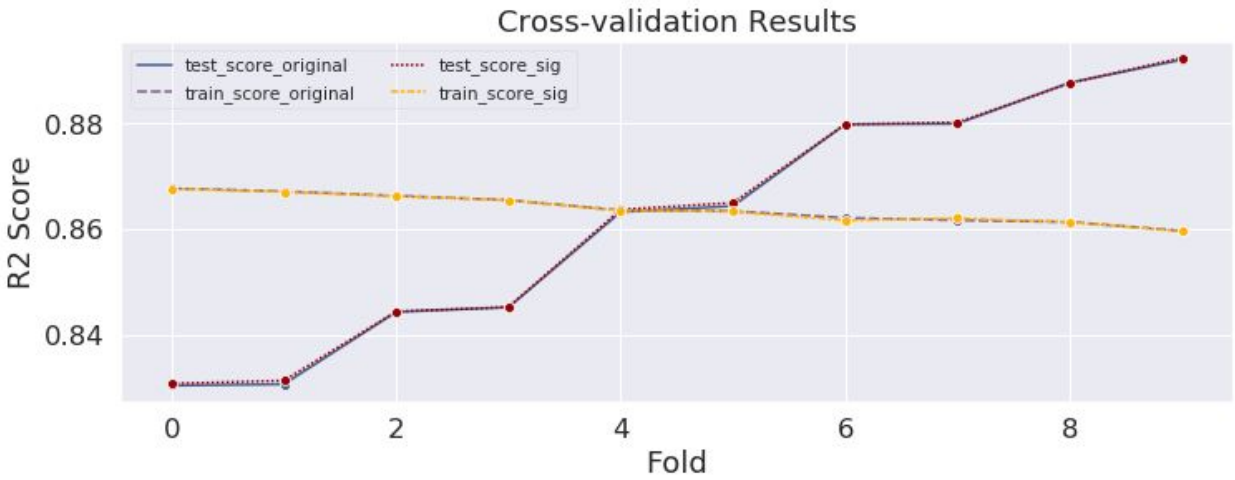
First, I observed the Pearson correlation coefficients between the dependent and each independent variable. I felt that this would give me a good starting point on how to proceed with any data manipulation that may have been required to optimize the model for more accurate predictions.



After observing the many strong correlations in the plot, I felt good about my chances of success with this model. Budget, age, vote average, and vote count all had very strong correlation coefficients with respect to the dependent variable. I decided to proceed with predicting on an unmodified dataset and make any further decisions after analyzing the results.

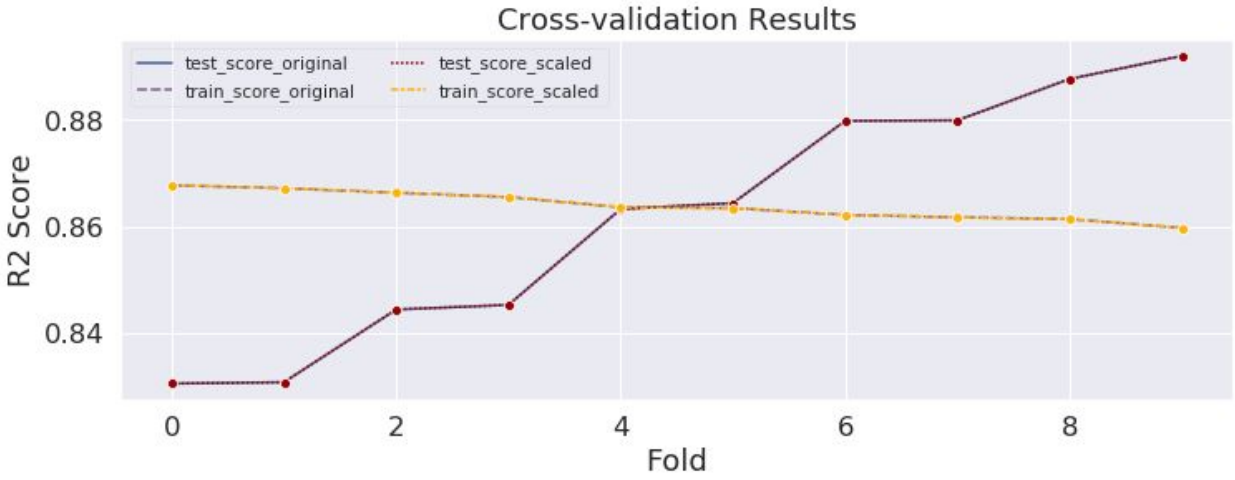
First, I split the data into training and final validation sets to guard against data leakage contributing to model overfitting. To my great delight, the model scored 0.864 for its coefficient of determination on the training data. To be reassured that this score wasn't merely the result of a favorable random split, I ran the model through a 10 fold cross-validation analysis to observe the range of scores that would result from many different random splits. The scores ranged from 0.8304 to 0.8921, with an average score of 0.8618. It appeared that the result from the first split was not extreme.

To see if I could improve on this score, I decided to look for features that the Statsmodels OLS summary report indicated were not significant as predictors of the target variable. These features were all movie genres. There were four of them: action, horror, music, and war. After removing these features, I ran the model through another cross-validation comparison. The scores from this model ranged from 0.8308 to 0.8924, with an average of 0.8621. There was a slight improvement, but not by very much. I plotted both the training and test scores from the cross-validation results on the original data and on the data with the subset of features to observe any differences.

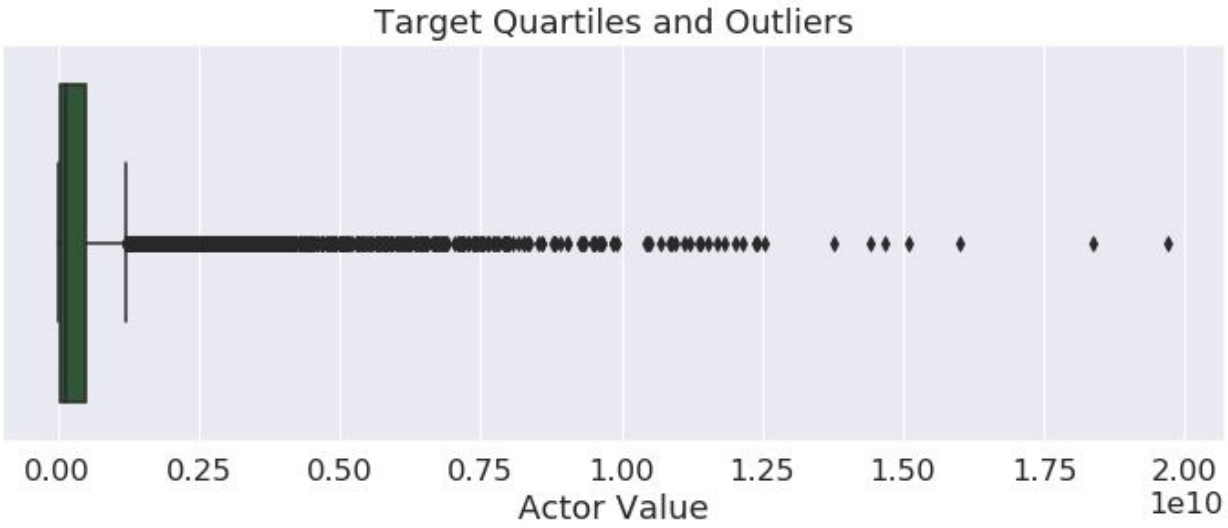


The results from each dataset overlapped so well that it was difficult to distinguish between the two. There was a slight difference that was observable at the 2nd fold.

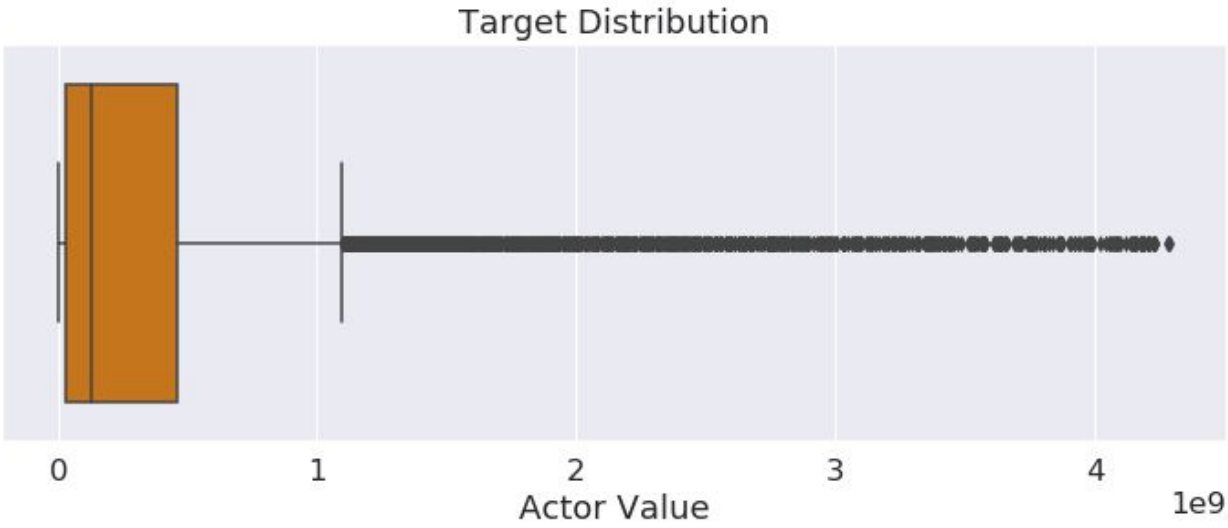
Another method that I knew was commonly used to increase linear regression model accuracy was the scaling of the predictor variables. This involved setting the mean of each feature to zero and giving each a variance of one. The results were exactly the same as they were on the original dataset. Nothing had been gained by scaling the features.



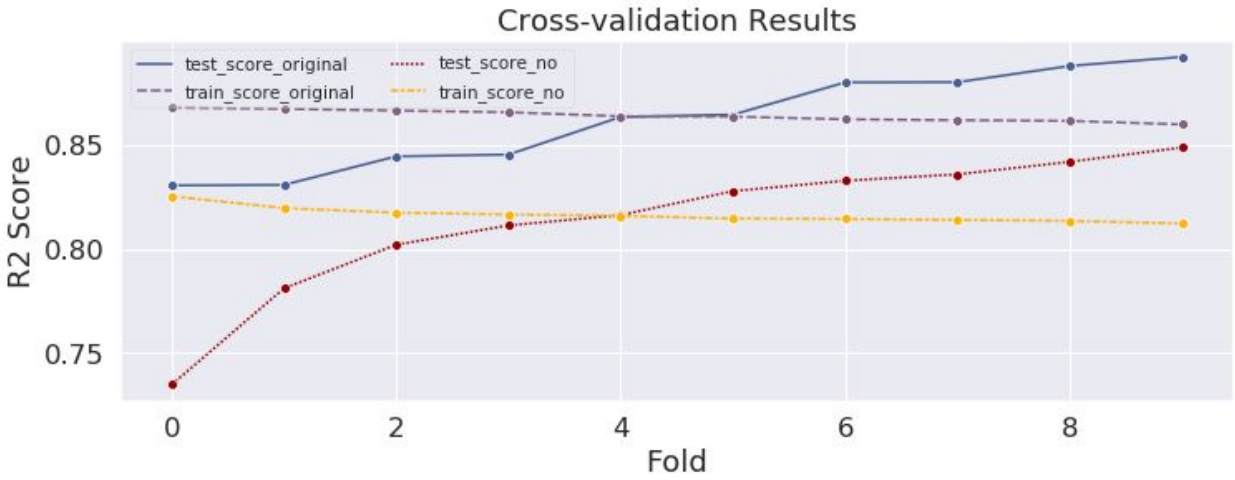
Next, I decided to take into account the possible sensitivity of this particular regressor to the many outliers of the target variable, due to the vast range of the film revenues.



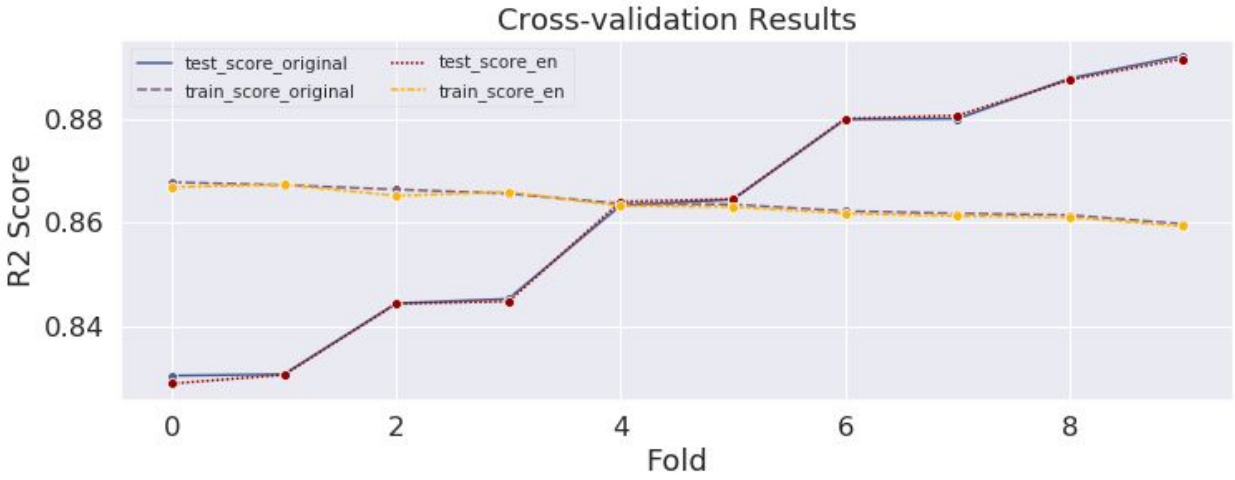
The outliers were very easy to observe in the boxplot of the target variable. I generated another boxplot after removing outliers from the target that had a z score greater than 3.



Actors who had lifetime movie revenue totals slightly over $4 billion were removed. The scores returned were not helpful in improving the model. In fact, the changes made the model less accurate. The cross-validation scores ranged from 0.7349 to 0.8486, with an average of 0.8132.

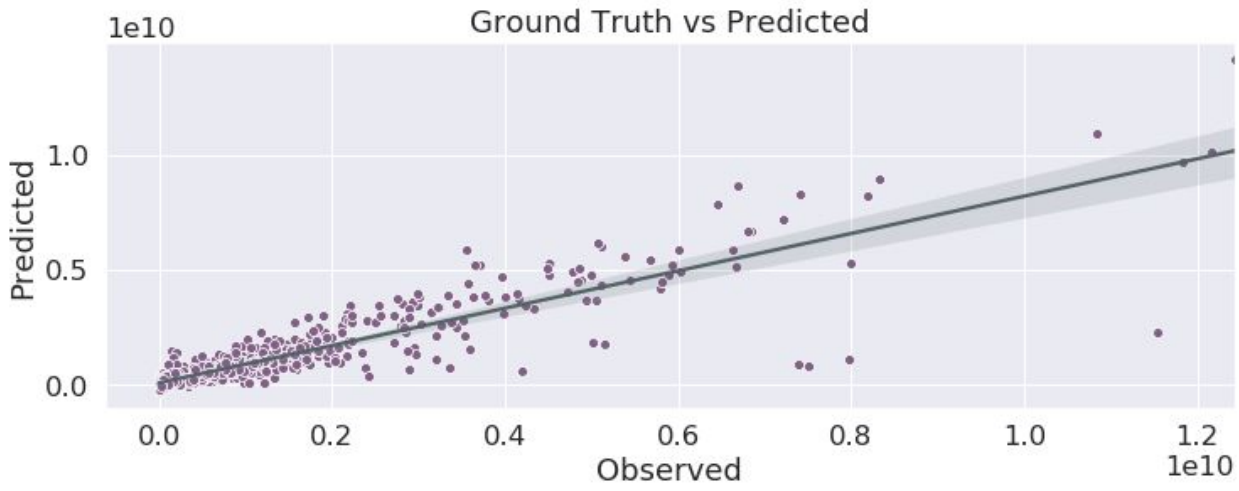


Lastly, I wanted to try using a regularization method on the model. Ridge regularization would penalize the features with large linear regression coefficients. Instead of removing them, as I did earlier, it would keep them small (L2 penalty). This would decrease model complexity, but eliminate the need to remove features. Similarly, Lasso regularization would still enforce the penalty for non-zero coefficients, but would allow for some coefficients to be exactly zeroed out (L1 penalty). I used Elastic Net, a method that combined both Ridge and Lasso regularization techniques. This method had two hyperparameters that could be tuned for model optimization, the L1 ratio and the alpha parameter. The L1 ratio would be tuned to decide how much the regularization performed like Ridge and how much it performed like Lasso. The alpha parameter was used to determine the severity of the penalization. I chose a range of 8 values to try for each hyperparameter. I used an exhaustive 10 fold cross-validation grid search on all 64 model candidates, which resulted in running 640 total fits. The best performing model's hyperparameters were an L1 ratio of 0.5 and an alpha of 0.1. The average score of the best Elastic Net model was 0.8620. It was just slightly lower than the average score of the model that had its features removed through inspection.



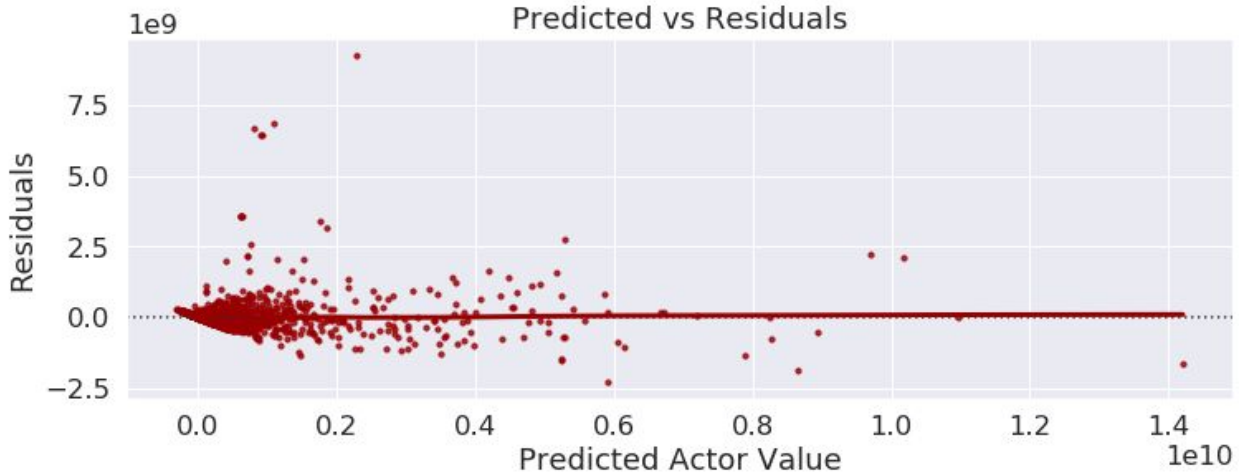
I decided to use the model which had the subset of predictors that I had chosen for the final evaluation on the test data, as it performed the best on the training data.

Before predicting on the holdout data, I wanted to observe the residuals of the best model. The training data were split once more into additional training and test sets. The residuals plotted were the errors calculated from predicting on the new training data and comparing the predictions to the new test data. The first plot I made was of the ground truth against the predictions.



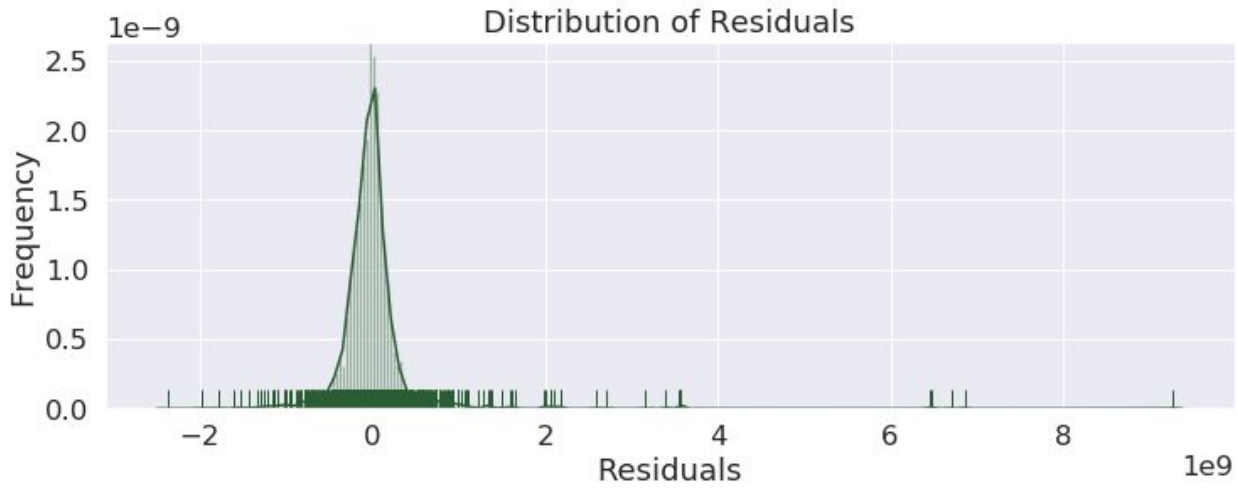
The model performed best when it was not predicting on the outliers. There were a few errors above $7 billion that pulled the fitted line away from the optimal angle of 45 degrees.

Next, I plotted the predictions against the residuals to look for any deviation away from a horizontal line.



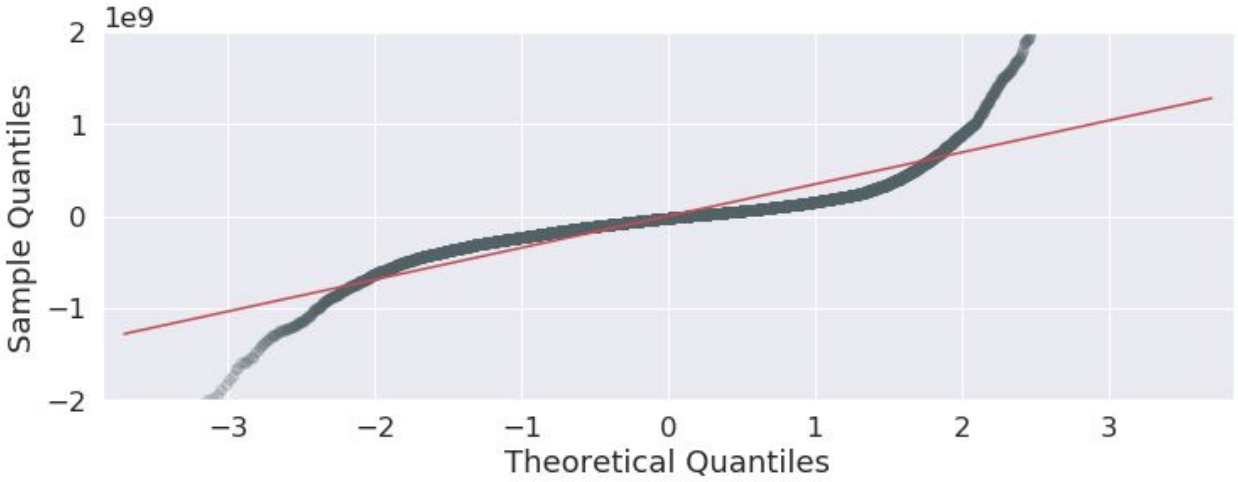
That plot looked good, as well. The fitted line was completely horizontal. This indicated that there was a strong linear relationship between the predator variables and the target variable.

Then, I plotted the distribution of the residuals to observe its form.



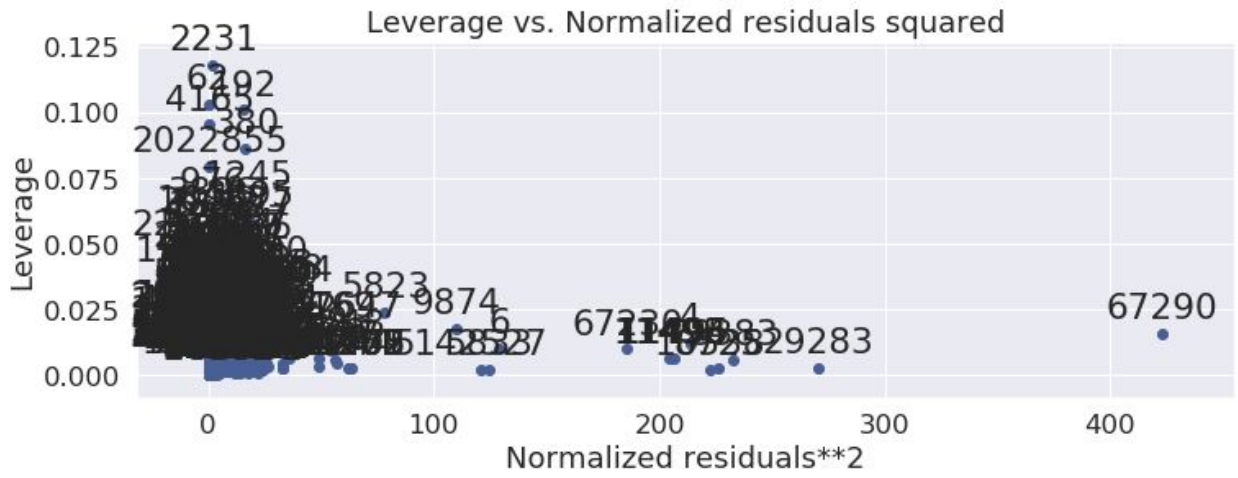
The residuals appeared normally distributed about the origin. A few large errors could be observed in the lower right corner of the plot, as well.

Next, I observed the quantile-quantile plot to look for where the residuals of the model started to deviate from a normal distribution.



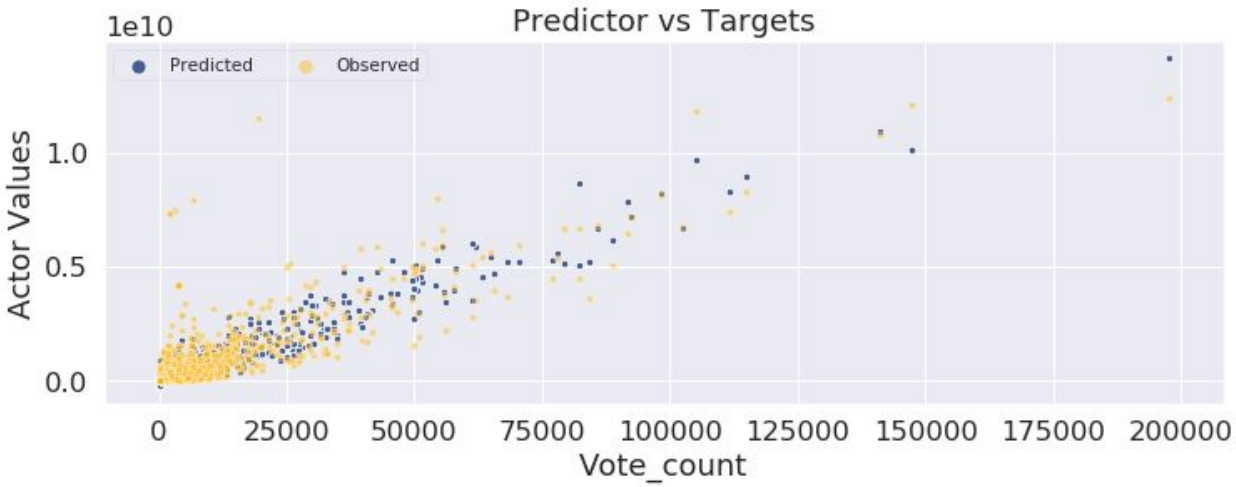
The residuals pulled away from the desired fitting just before hitting $1 billion. These errors accounted for less than 3% of the total number of residuals.

Finally, I created a leverage plot to observe any data points that had strong influence over the predictions of the model.



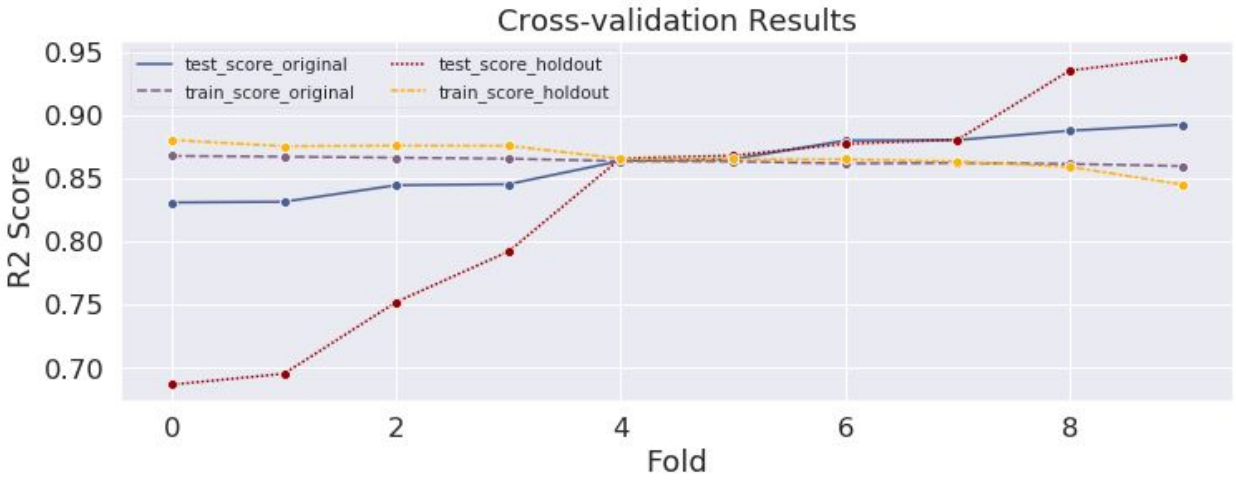
Fortunately, there were no observations that had both a large residual and a high leverage value. Those points would have appeared in the upper right corner of the plot. The actors with the highest leverages were still found near the population cluster. There was one actor who had a noticeably high residual, as could be observed in the lower right corner of the plot. After I referred to my reference dataset, I discovered that this person was Verna Felton. She was the voice of many characters in Disney movies throughout the 50's and 60's. Some of her characteristics that may have thrown off the model included her higher than average age of 66 years. This would have been especially true given her gender, as it was observed earlier that female actors tended to be over 6 years younger than their male colleagues. Another interesting note about the data related to her is that her roles were spread fairly evenly among the billing orders of her movies. Her billing order ranged from 3rd to 8th in 5 films that had over $2 billion in collective revenue. Also, she averaged as the 6th actor in the credits. Despite performing in many high revenue films, her casting did not indicate that she had the star power that was a characteristic of all of the other actors who held the largest career revenue totals.

Before I turned to the final model validation, I wanted to observe the performance of some of the strongest predictors. I plotted a handful of them against their target values as well as the predictions they made on those targets. I will only show the plot for the strongest predictor for the sake of brevity.



The predictions aligned quite nicely with the ground truth under this predictor. The next three strongest predictors showed similar performances.

Finally, after being allowed to predict on the validation data, the average cross-validation score obtained by the best model found was 0.8298.



The linear regression model that performed the best in this dataset had a training score of 0.86 and a test score of 0.83. The model accurately predicted career movie revenue of actors, and its performance was shown to be generalizable to unseen data.