Predicting Success of Movies and Video Games

Creating Lasso Regression and Random Forest models for movie and video game datasets

Daniel Gomes, John Gomes, Blake Simmons Dartmouth, MA
CIS490 - Machine Learning dgomes5@umassd.edu
University of Massachusetts Dartmouth jgomes15@umassd.edu
bsimmons@umassd.edu

***Abstract*—There are many machine learning methods that can be used to make predictions based on existing datasets. Two examples of supervised learning methods include Lasso Regression and Random Forest regression trees. We use these two methods to explore their performance when applied to two datasets. The first attempts to predict movie revenue based on various movie attributes and the second attempts to predict video game global sales based on several video game attributes. The results are mixed and it was concluded that while lasso regression and random forest perform to similar accuracy within the movies database, random forest far outperformed lasso regression when using the video games database. We also concluded that the datasets used were not ideal for these purposes and were difficult to predict.**

Keywords—lasso regression, random forest, movies, video games, budget, global sales, revenue, RMSE, R-squared, lambda

I. Introduction

Our Motivation for conducting this project on predicting success of movies and videogames is just our own interests in the movie and video game industries. We want to see what qualities can make or break the success of a movie or video game. For example looking into how much of an impact do qualities such as lead role, budget, and production company have on a movie or qualities like platform, publisher, developer, and genre may have on a video game. We are predicting the success of movies based on the revenue produced from the movies and the success of video games based on their Global sales. For the purposes of this experiment we are comparing only two machine learning methods: Lasso Regression and Random Forest regression tree. We will compare the performance of both methods between application to both datasets.

II. Literature Review/ Methods

A. Datasets

Our first dataset is a "TMDB 5000 Movie Dataset"[1] which is a Metadata on 5,000 movies from the website TMDb off kaggle. It included attributes like genre and budget as well a revenue attribute which is what we will be predicting. A full list of all the movies dataset variables are below.
Movies Dataset Variables:

budget	Money spent creating movie
genres	Type of movie (Horror, Comedy)
homepage	Movies internet home address
id	Identification of movie
keywords	Iterated words throughout movie
original_language	Language movie was created
original_title	Name given to movie in production
overview	Summary of movie
popularity	General Public interest in movie
production_companies	Companies involved in filming
production_countries	Countries movie was filmed in
release_date	MM/DD/YYYY movie came out
revenue	How much the movie made ($US)
runtime	How long the move is
spoken_languages	Languages in the movie
status	In theatres, dvd, digital copy, etc.
tagline	Catchphrase/Slogan of movie
title	Name of the movie
vote_average	Average vote score
vote_count	Number of votes for the movie

Our second dataset "Video Game Sales with Ratings"[2] is a Metadata on video game sales from a website called Vgchartz and their corresponding ratings from another website called Metacritic well known for reviews. It includes attributes like genre, platform, developer, and rating with which we will be predicting the success of these titles through its Global sales attribute.

Video Game Dataset Variables:

Name	Name of the game
Platform	Console on which the game is running
Year_of_Release	Year of the game released
Genre	Game's category
Publisher	Publisher
NA_Sales	Game sales in North America (in millions)
EU_Sales	Game sales in the EU (in millions)
JP_Sales	Game sales in Japan (in millions)
Other_Sales	Game sales in the rest of the world, i.e. Africa, Asia excluding Japan, Australia, Europe excluding the E.U. and South America (in millions)
Global_Sales	Total sales in the world (in millions)
Critic_Score	Aggregate score compiled by Metacritic staff
Critic_Count	The number of critics used in coming up with the Critic_score

User_Score	Score by Metacritic's subscribers
User_Count	Number of users who gave the user_score
Developer	Party responsible for creating the game
Rating	The ESRB ratings (E.g. Everyone, Teen, Adults Only..etc)

B. Methods

For our project we decided that it would be best to go with Supervised Learning. We felt this was best for the predictions that we are going to perform. We are trying to predict the best possible outcome based on other variables. As mentioned we will be predicting financial success with these variables in the form of revenue for movies and global sales for video games.

The first method we performed on our dataset is Lasso Regression. We decided to use lasso because we found out a very easy and simple way to evaluate variables that are correlated to each other. With Lasso Regression in order to evaluate its performance we decided to go with RSME (Root-Mean-Square-Error) so that we could analyze the best fit of our model. We will also be using Cross validation to measure the accuracy of our predictions and to determine the best lambda.

The Second method we choose to was Regression Trees (Random Forests). We felt this method would be useful as its simple to understand a tree model and we can add the possibility of fine tuning to obtain an even more accurate prediction. Once again we will be using RSME (Root-Mean-Square-Error) to analyze our tree model and Cross validation to judge the tree size for our evaluation methods.

For both methods and both datasets, we first cleaned the data, removing any rows with empty data, regularizing variables, removing unnecessary columns such as movie and game titles, and reformatting attributes into meaningful types, such as changing certain character strings (‘genre’, ‘platform’) into factors. We also scaled revenue from dollars to millions of dollars to make the values/scale more manageable.

We then split the data into train and test partitions with our training data comprising 80% of the dataset and our testing data comprising 20%. We created the regression formulas for each dataset based on numeric variables, and factor variables with levels less than 53 (as this is the maximum number of factor levels accepted by the randomForest functions). Since our goal was to compare the performance of both machine learning methods, both the train/test partitions and the formulas needed to be identical within the application onto each dataset.

The formulas applied for the movies and video games datasets, respectively were:

revenue ~ budget + original_language + popularity + production_countries + release_date + runtime + vote_average + vote_count + genre

and

GL_S ~ PLAT + YOR + GEN + RATE + CR_S + CR_C + UR_S + UR_C

with the predicted value for each being total revenue, and global sales, respectively.

It is important to note that the video game dataset included sales numbers for different regions such as North America and Europe. Since these numbers essentially added up to the global sales, we decided to eliminate them and use other attributes to attempt to predict the global sales.

We performed Lasso regression on each dataset, using 10-fold cross validation to determine the best lambda. This algorithm eliminated any attributes which were not predictive. The pseudocode representing the model and cross validation portion of our code for lasso regression is as follows:

```
model <- glmnet(X training data, Y training
data)
plot(model, xvar="lambda")
cross validation <- cv.glmnet(X, Y, alpha=1,
nfolds = 10)
plot(cross validation)
```

We then created a Random Forest regression tree model using the same training data. We also evaluated the best mtry value to use in our model. The pseudocode representing the model and tree building is as follows:

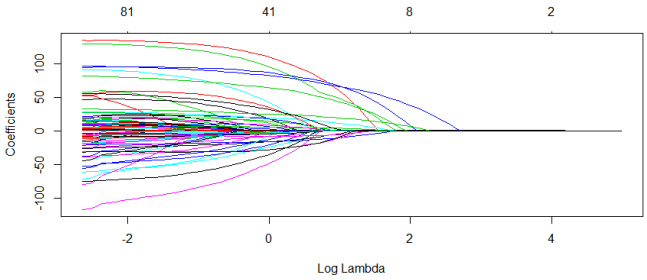
```
model <- randomForest(formula, data=training,
mtry=round(square root(number of
columns(training) -1)), importance=TRUE)
```

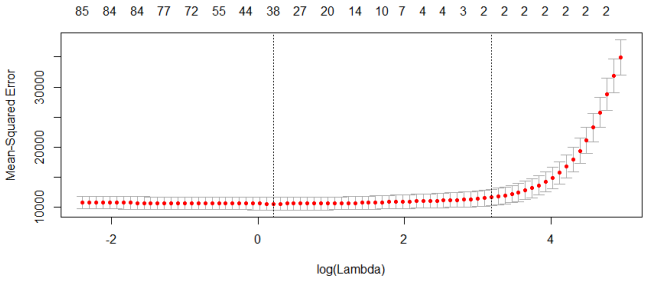
Once both methods were used, we plotted the predicted values against the true values in order to observe the accuracy of each model visually. We then calculated the individual RMSE and R-squared values for each model and compared them.

III. RESULTS

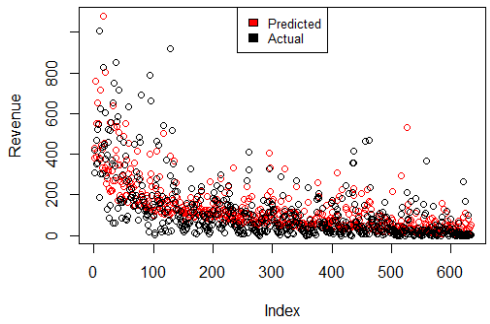
A. Movies Dataset

Lasso Regression
Best Lambda: 24.13829





Besides variables budget and vote_count, lasso regression removed every other variables from the movies dataset.



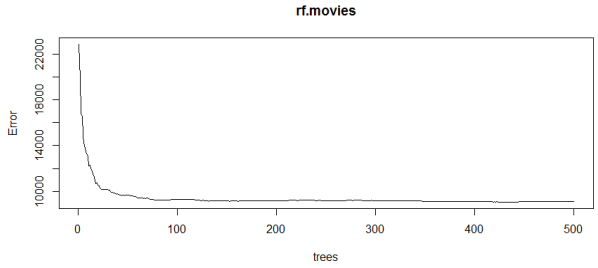
RMSE:
96.5230200967387

R-squared: 0.6987305

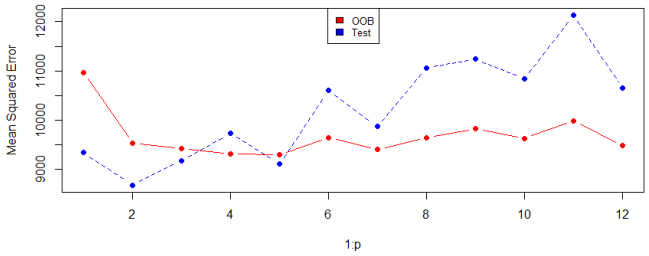
Final Formula: Revenue = 6.79905830 + (Budget * 1.411) + (Vote Count * 0.058)

Given the large range of the movies dataset, while an RMSE of ~96 is relatively large, it is not a terrible value. This is confirmed by the R-squared value of ~0.7 and by observing the plotted comparison above.

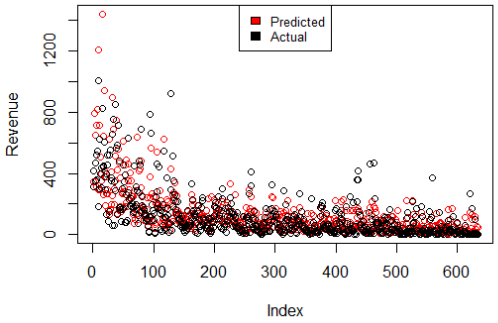
Random Forest



At about 125 trees, Random Forest shows that the error rate begins to stabilize.



Based on our tests, we chose mtry of 2 for our final random forest model, given the low error rate for both test and OOB data.

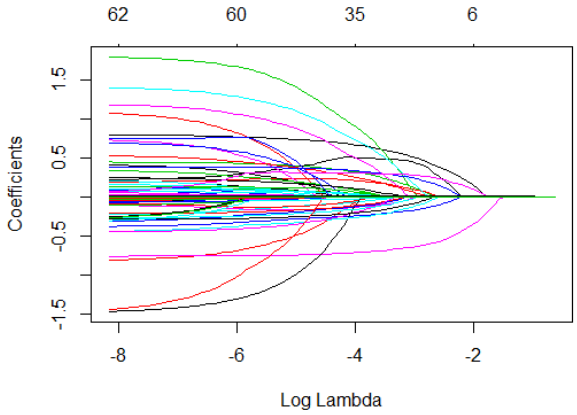


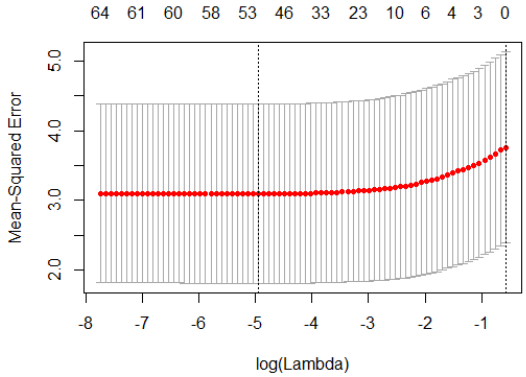
RMSE: 94.6004570140387
R-squared: 0.6660517

It is surprising to see an RMSE value so close to that of the Lasso Regression, but it does appear slightly better. Curiously, we get a lower R-squared value then that of the Lasso model. The visual does suggest a similarly accurate prediction.

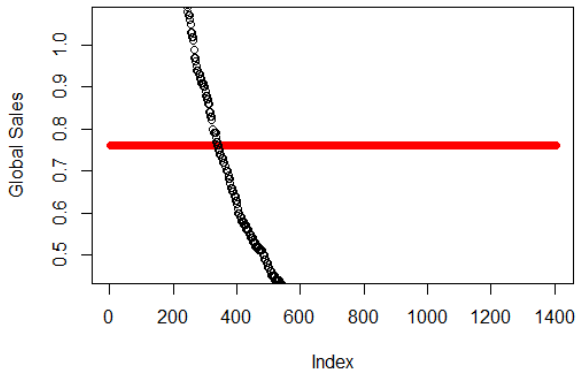
B. Games Dataset

Lasso Regression
Best Lambda: 0.5612952





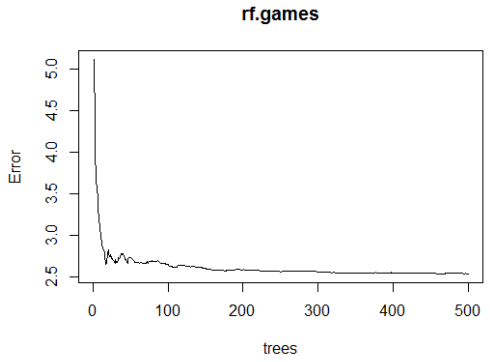
Besides the CR_C (critic count), lasso regression moved every other variables from the movies dataset. Since the coefficient for the only remaining attribute included in the model was so small, the prediction line essentially reflected only the intercept of ~7.5 indicating an absolute failure of the model to predict anything meaningful.



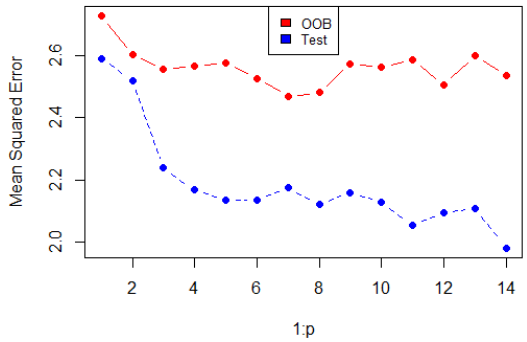
RMSE: 2.02496867599784
R-squared: 0.1774034
Final Formula: Global Sales = 0.7615562719 + (Critic Count * 0.0001908167)

The low R-squared value confirms very low performance of this model.

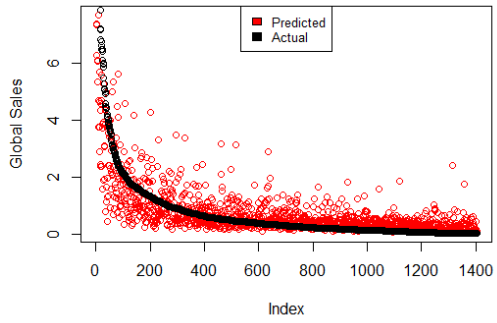
Random Forest



We chose to use 175 trees, as the error begins to stabilize around this point.



Based on our testing we chose an mtry of 12, given a relatively low error rate for both OOB and test data.



RMSE:1.44290201087766
R-squared: 0.4921552

Clearly, Random Forest yielded a much more accurate model than Lasso Regression with this dataset. While an R-squared value of ~0.49 is not exceptional, given the low accuracy of the Lasso model, this accuracy level is relatively good.

IV. Conclusion/Discussion

In conclusion, we discovered for the Movies dataset both Lasso Regression and Random Forests performed well in predicting movie revenue but used only a few attributes from the entire dataset. For instance in our prediction of revenue formula the only two attributes included were budget and vote_count. With these attributes and our Intercept (6.79905830) we can use these components to create our revenue prediction formula.

However for the video games dataset Lasso regression was not very efficient at predicting Global game sales based on its attributes other than singular continental game sales. Random Forests was however much better for predicting global game sales.

A limitation on the games dataset was that there was no revenue attribute on the total sales of video games and we felt revenue was the best measure of financial success. As a result we had to use the Global sales attribute in the video games dataset for our predictor which was only well correlated to other sales attributes i.e. (NA Sales, EU Sales, JP Sales). We could not find any relationship from this variable to attributes like platform, genre, rating, critic score, user score, etc.

In the end, these datasets were not ideal for the purposes in which we were conducting our experiment. These datasets could still serve as a solid basis for collecting data in the future but we would need to add more workable data attributes to our datasets. Given more time, possible continued work on these datasets in particular might be using dummy variables to consolidate factors of high levels so that they can be used.

V. References

[1] (TMDb), The Movie Database. "TMDB 5000 Movie Dataset." Kaggle, 28 Sept. 2017, www.kaggle.com/tmdb/tmdb-movie-metadata.

[2] Kirubi, Rush. "Video Game Sales with Ratings." Kaggle, 30 Dec. 2016, www.kaggle.com/rush4ratio/video-game-sales-with-ratings.

[3] "RandomForest." R Documentation, www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest.

[4] "Ridge Regression and the Lasso." R-Bloggers, 23 May 2017, www.r-bloggers.com/ridge-regression-and-the-lasso/.

VI. Team Statement

All group members contributed equally.