Drake Song

Project 3 Technical Report

Project Description

Using classification and regression models of Logistic Regression, Decision Trees, and Support Vector Machines, predict rating (above 7.5 or not), rank (which 100s level), and revenue of a film using performance data of past movies.

Feature Selection Rationale

- num_words_title
    - According to an article titled "Top Ten Tips for Titling Your Movie"[1], the first tip is the shorter the title the better. Thus, instead of using Titles, number of words in the titles are used for analysis
- num_words_desript
    - Number of words in the description was initially used in hopes of expecting higher performance of a movie if the description is longer (also considered the case where more brief descriptions are more effective). However, that analysis did not work at all. Instead, since bag-of-words function from MatLab requires a Toolkit that one must pay for, a decision has been made to count number of word that exceed 7 letters (tested number of letters from 1 to 10 and 7 seemed to perform the best). By only counting long words, insignificant words such as "a", "the", "as", and so on can be ignored in the analysis.
- genre_num
    - This feature contains 3 columns: genre_1, genre_2, and genre_3. Movies in the database contain up to 3 genres with a minimum of 1. Using "Market Share for Each Genre 1995 – 2017"[2] and some educated guesses about genres not included in the chart, each genre was assigned a number; the higher the number, the more market share that genre has. For example, 'Adventure' has been assigned '20' because it has the highest market share of 27.71% with $57.5 trillion in Total Box Office. If a movie has only 1 genre, then genre_2 and genre_3 are assigned '0'.
    - Average of the genre scores have been tested but performed poorly
- top_actors_count
    - Using "Vulture's 100 Most Valuable Stars"[3] (after considering multiple "top actors" lists, this was the most sensible list that seemed to not have too much bias towards certain actors in particular genres of films), a list of top 100 actors was created. Using this list, the number of "top actors" was counted for each movie in the data.
- top_directors_count

1. http://www.chrisjonesblog.com/2011/07/top-ten-tips-for-titling-your-movie.html
2. http://www.the-numbers.com/market/genres
3. http://www.vulture.com/2012/07/most-valuable-movie-stars.html

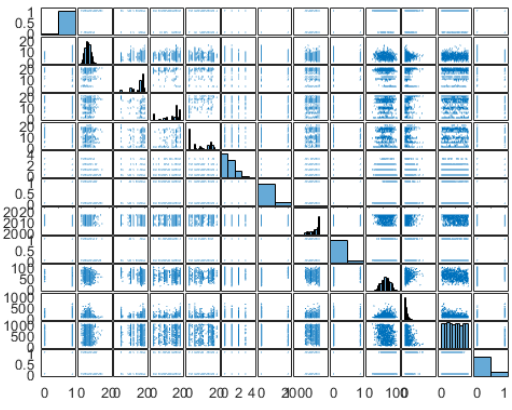
    - Similar to top_actors_count. Using "The 100 Best and Most Exciting Directors Working Today"[4], each film was given a '1' if the film was directed by one of the 100 and '0' if not. Certain precautions were taken because for some reason, the data's director field does not support special characters and there are many directors with special characters in their name.
- num_votes
    - Through trial and error, it was determined that separating movies with votes higher than 295,000 from those with lower than that number of votes provides an efficient model.
- run_time
    - Similarly, through trial and error, it was determined that separating movies with runtime higher than 140 minutes from those with lower provides an efficient model.

Models

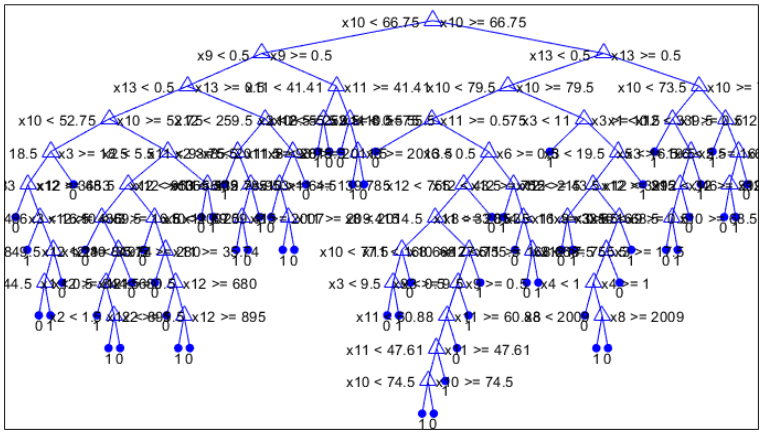
- Logistic regression
    - Logistic regression has 3 main assumptions: classes are linearly separable, log-odds of features are in a linear relationship with the classification, features are independent. For this particular project, the first two assumptions are presumed to be met. The third assumption, independence of features, are tested before logistic regression models are used.
- Decision trees
    - For decision trees, 3 different types of trees are tested: regular decision tree, tree bagger, and random forest. All 3 of these are tested and then the error produced by each are compared.
- Support vector machines
    - Parameters of support vectors machines are tweaked and tested such as box constraint to find the best fitting model

Analysis / Prospectus

This company boasts in its thoroughness with the models and in choosing the best model to use in different scenarios. When predicting the rating of a movie, all three models are used: logistic regression, decision trees, and support vector machine. Taking a look at logistic regression first, the algorithm first checks for independence within the features. The training data set being used has no linear correlation between the features. Then the algorithm initially creates a model using logistic regression. From there, p-values of the features are calculated to 95% confidence level and those who have p-values greater than 0.05 are removed until all features pass the test. Confidence intervals of

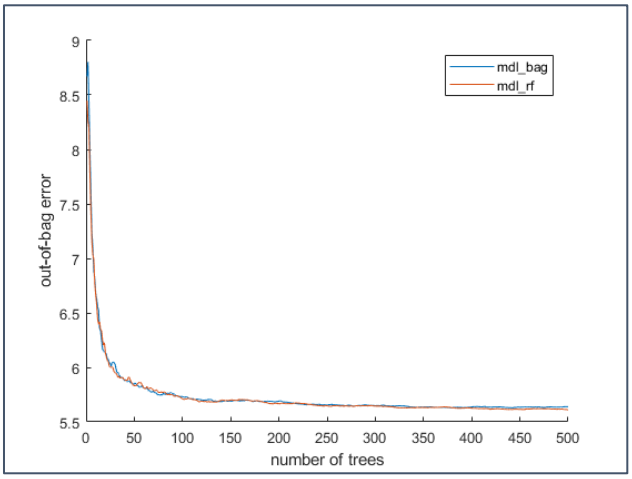


4. https://theplaylist.net/100-best-exciting-working-directors-20160926/#cb-content

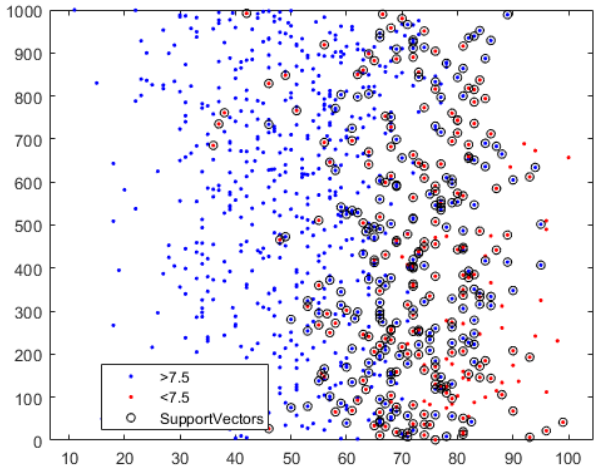


features are then created to make sure that all the remaining features are in fact 95% confident. Yhat are then calculated with the model and then the error of the model by using test data (100 movies) reserved before any analysis was performed. Confusion matrix is also calculated for further analysis of the error.

Decision trees are then used to create another set of models to predict ratings of movies. First a regular tree is created as seen on the left. Then two more trees are created: one that utilizes the tree bagger method and the other that uses random forest method. These two extra models are created to add to the thoroughness of analysis in making sure that the tree that is being created is of the highest quality. For tree bagger and random forest, out-of-bag error anlysis is performed to make sure that the most efficient number of trees is selected for the model. After these three tree models have been created, the algorithm then selects the best performing tree to be used for comparison at the end.



A support vector machine model is created as the last model in predicting ratings. After all three of the models have been created, the algorithm then determines the best model by comparing the error produced by each model. By doing so, the algorithm thoroughly utilizes the models in making sure that the most accurate model is used to preduct ratings of a movie.



In predicting the rank of a movie, the same steps are taken with slightly adjusted models. For predicting revenue of a movie, only decision trees and support vector machines are utilized as logistic regression model fails horribly in predicting revenue that is not categorized into classes. Also, in calculating errors for revenue, root-mean-squared-errors (RMSE) are used in order to better analyze the errors produced by the models.

Conclusion

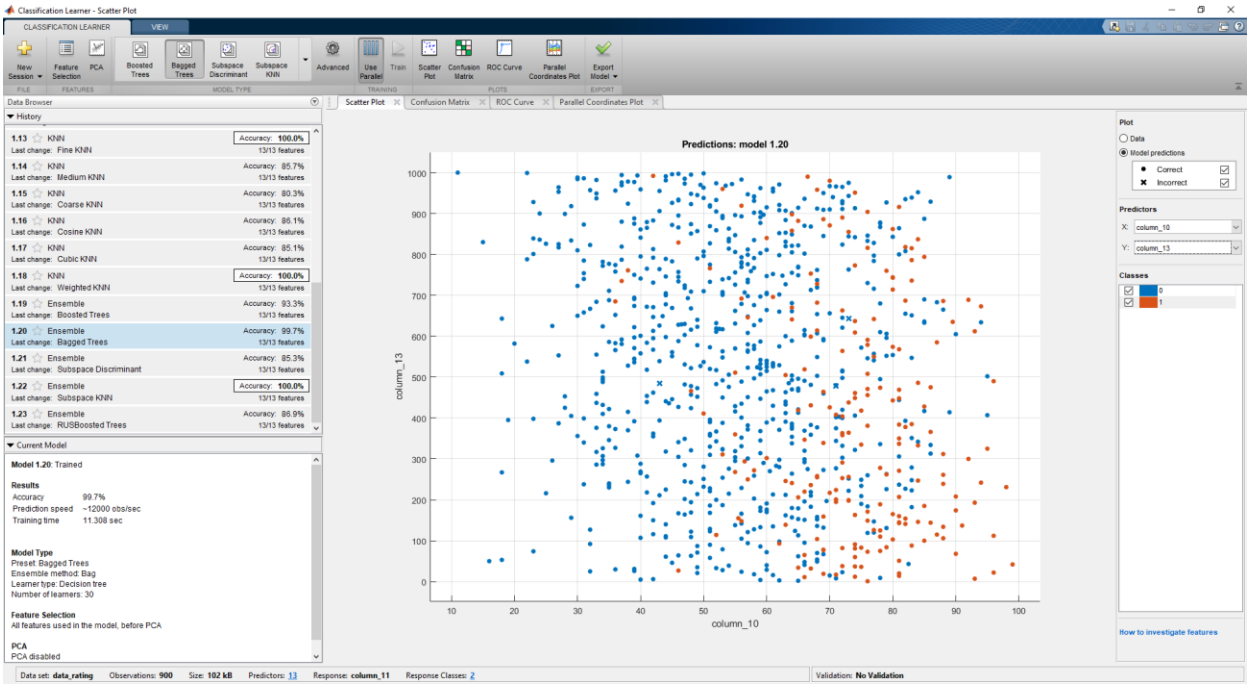
After a countless number of runs, the algorithm produces an error in the mid-10% in predicting ratings, mid-80% in predicting rankings, and low-100 ($100 mil) as RMSE in predicting

revenue. The error rate for predicting ratings can be acceptable, the error produced by predicting rankings and revenue cannot be accepted at all. In order to fix these high error rates, many methods have been used. The data, at first, were reshaped; from converting the features into a binary feature to even just not doing anything to it at all, the data set have been reshaped multiple times by utilizing reasonable logic in converting the data.

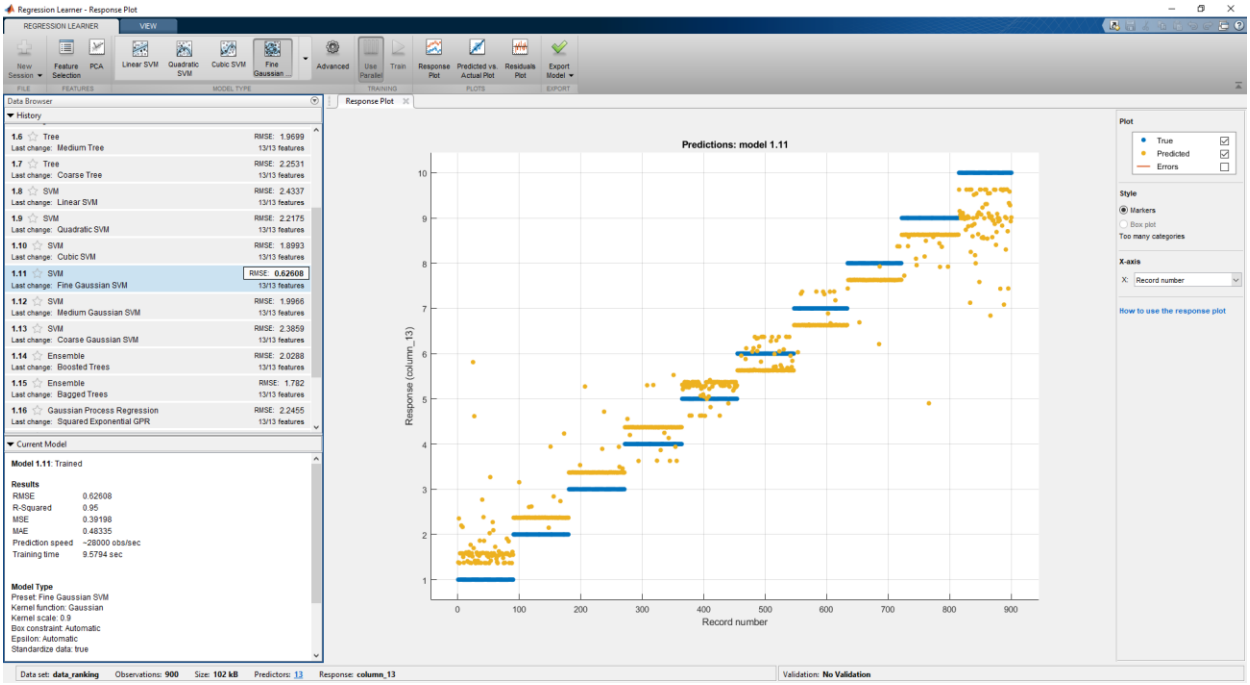
When changes within the data did not produce a big difference, the parameters within the models were changed. From changing the functions (mnrfit to fitcecoc) to trying out all the different options for the parameters that were able to be understood, a countless number of trials and errors was undergone in hopes of finding better parameters for the models. When all failed, cross validation was then utilized; however, cross validation only made the models worse. After spending an entire week of sleepless nights and after going through blocks of commented out codes, a conclusion has been made that the models are just not going to get any better. Thus, the best possible combinations of feature selections and model parameters have been selected as the final models.

For the sake of reducing errors, MatLab's Machine Learning Toolkit was utilized in creating models that are in the function *MLApp.m*. Using this tool, errors have significantly decreased: 1 – 12% error for predicting rating, 20 – 40% error for predicting rank, and approximately 40 - 60 RMSE. Because this toolkit also has the function to generate the code and export the models, attempts have been made to utilize the choices made in this toolkit in hopes of creating better models for myself. However, for some unexplainable reason, my models (even when the code is exactly copied) did not show any improvements whatsoever. This is probably due to some parameters used by the toolkit that I have no clue about. The models produced by the toolkit is attached below for demonstration.

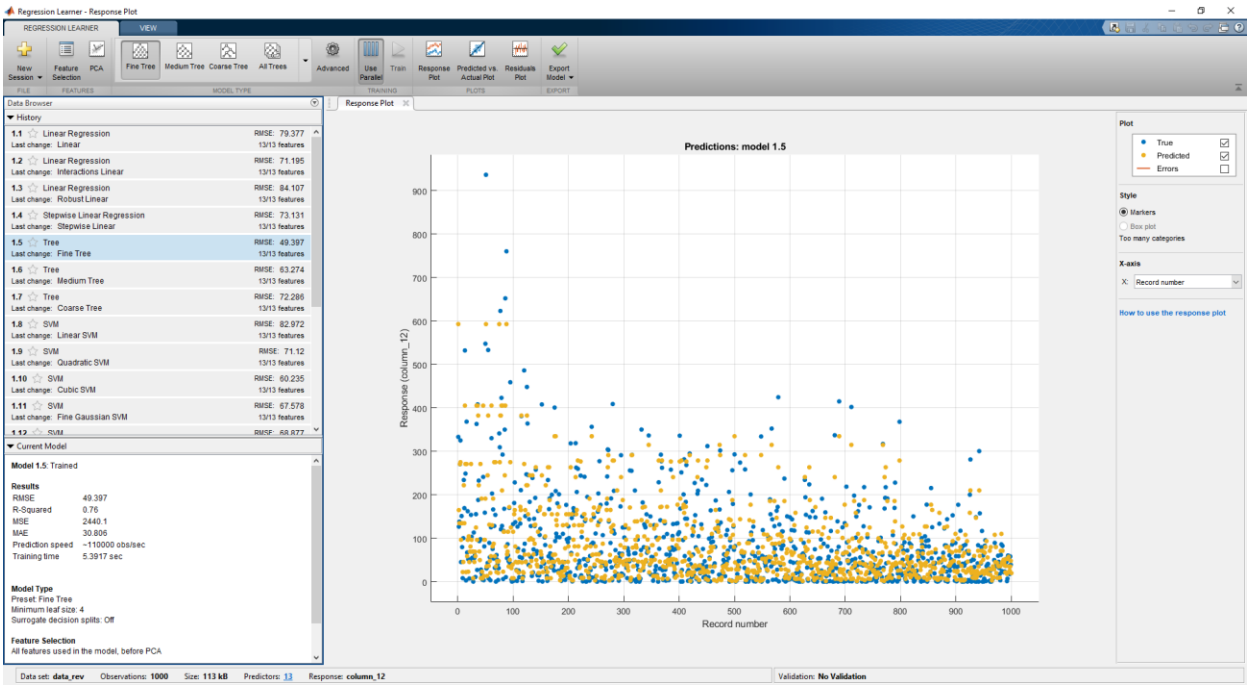
<u>Rating</u>



Rank



Revenue



In conclusion, the only model that would be useful is the one that predicts ratings of movies. The model that predicts rank should not even be looked at because of the atrocious error rate. The model that predicts revenue can be used as a reference but one should not place too much weight on the results of the model. If one does really desire predicting ratings, ranks, and revenue, the models created by MatLab's Machine Learning Toolkit should perform a lot more accurately.