Movie Viewing Time Prediction

Cindy Wang

INTRODUCTION

This project is to predict the daily viewing time of movies through building regression models in Python. The database includes 4226 movies, which are released from 1916 to 2017 and vary in 27 genres. The entire modeling procedure includes data exploration, data cleaning, model fitting and evaluation. Overall, this project builds three regression models, including lasso regression, ridge regression and random forest regressor, and evaluates the performance among them to choose the best fitting model for future prediction.

DATA EXPLORATION & DATA CLEANING

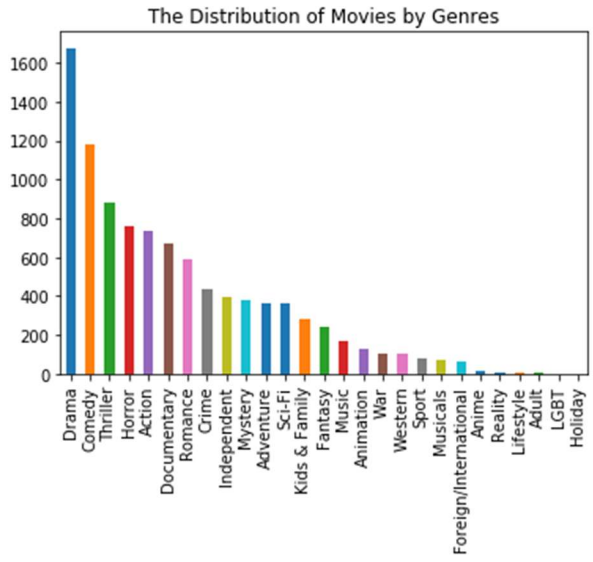
To better clean the data, it is important to understand the distribution of features. This section will explain the use of data exploration in data cleaning process and how each feature was processed to fit the model.

- **Duplicates and Null Data**

Before dealing with the categorical and numerical features separately, the duplicates are dropped at first. Regarding the null data, it is more appropriate to replace them with the mean value, instead of dropping it in consideration that 3242 out of 4226 videos having at least one null value.

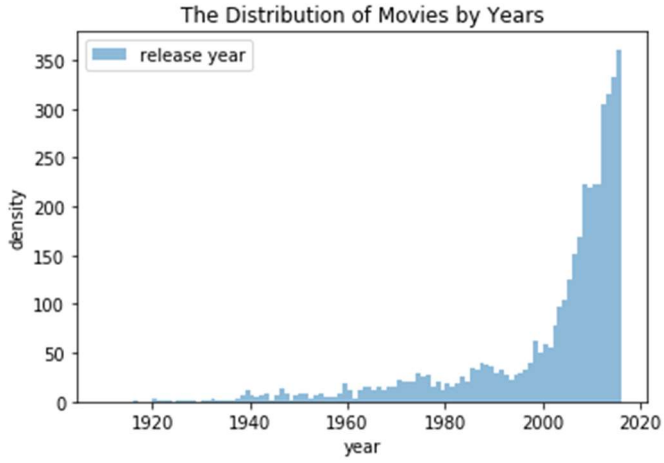
- **Movie Genres-'genres'**

As the diagram below shows, the movies are clustered in categories like drama, comedy, thriller and etc. In contrast, they are sparsely distributed in Anime, Reality, LGBT and Holiday genres. Hence, these genres will be combined into one category called "Miscellaneous Genre."



- **Release Year**

Similarly, the “release_year” was divided into different buckets based on the distribution shown below. The principle is trying to make sure each bin has similar number of movies.



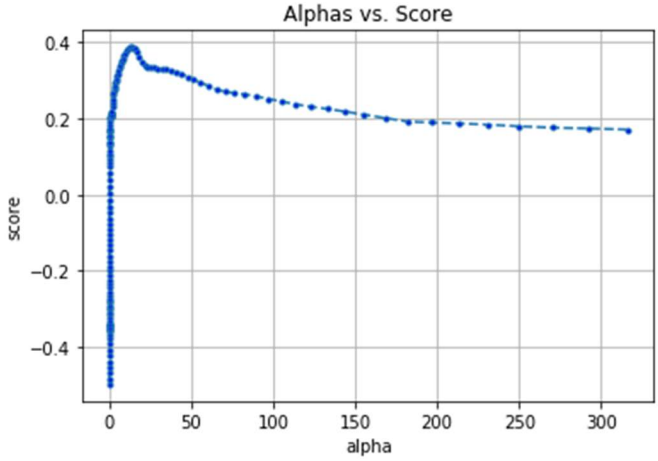
- **Other features**

Regarding the categorical features, dummy variables are created for each. The project uses the min_max scalar to scale the numerical features.

MODEL FITTING

The project has randomly selected 15% of dataset as “model_test” dataset to evaluate the model at the end. The rest becomes “model_train” dataset to build the model. Within the “model_train” dataset, the project uses 80% of data to fit the model and 20% to tune the parameters. The details of each model are explained below:

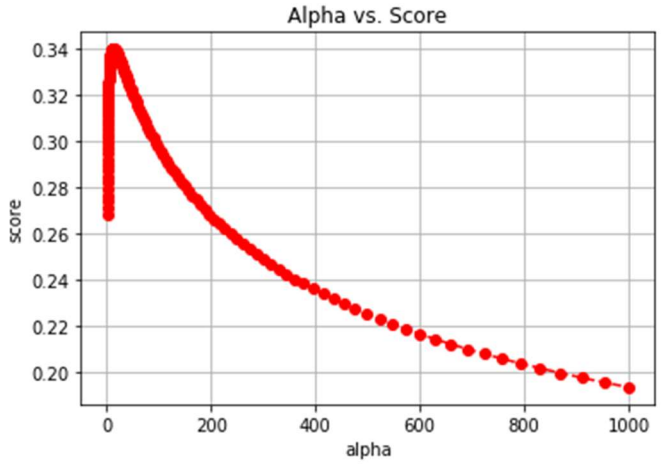
Lasso Regression Model with Polynomial Features



The model transforms the features by adding squared features of each one separately. The project tunes the parameter “alpha” (equivalent to lamda in L1 regularization) by fitting model in “model_train” dataset. The optimal alpha is 13.5.

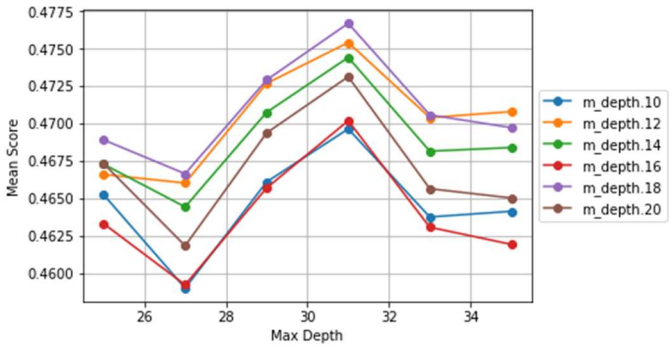
Ridge Regression Model with Polynomial Features

Similarly, polynomial features were added to fit the ridge regression model. In this model, the parameter “alpha” corresponds to L2 regularization. The optimal alpha is 12.2.



Random Forest Regressor

The random forest regressor is a non-linear ensemble model. After five-fold cross-validation, the project selects “n_estimators” (the number of different trees in the forest) as 31 and “max_depth” (the depth of each tree) as 18. Unlike the previous two models, this model uses the original feature set.



MODEL EVALUATION AND CONCLUSION

After comparing the “rf_score” of three models, the project selects the random forest regressor as the final model. Its corresponding “rf_score” is 0.58. Based on this model, we rank the

features by their impact on watching time. The most important feature that significantly affect users' watching time is as below:

- Weighted_categorical_position
- Imbd_votes
- Weighted_horizontal_position
- Star_category

Appendix

The description of columns in the dataset given:

video_id: A unique id for a movie

cvt_per_day: Cumulated view time per day

weighted_categorical_position: Average vertical positions on the home page that the movie was placed

weighted_horizontal_poition: Average horizontal positions on the home page that the movie was placed

genres: genres of the movie

release_year: the year the movie was released

imdb_votes: the number of votes on IMDB, typically higher the votes the better

budget: budget of the movie production, typically the higher the better

boxoffice: gross box office in US as updated on IMDB, typically the higher the better

imdb_rating: ratings on IMDB

duration_in_mins: how long is the content in minutes

mpaa: MPAA ratings

awards: TVPG ratings

import_id: content partners

metacritic Score: metacritic score on IMDB page. Typically, the higher the better

star_cateogry: a score to measure how popular the actor/actress are associated with the movie