Using Data Mining to Predict Popularity and Market Success of Movies

Harpreet Singh, Xinhai Zhou, Zelan Xiang

Department of Computer Science, University of Victoria

Victoria, BC, Canada V8P 5C2

***Abstract* - This document is our proposal to start a research project on applying data mining techniques and methods to predict popularity and market success of a specific movie with some given features such as production company, genre, cast, vote count etc.**

Introduction

Even though movies are an artistic endeavour and not purely a way to make money, people involved in production of movies dream of a solution that can help them predict whether a movie is going to be successful before they spend a lot of effort and money making it.

We propose a project using algorithms and techniques of data mining to predict different measures of a movie's success before its production. As part of this project we would analyze performance of several models on our dataset, and then write the best performing model ourselves from scratch. We would also build a website with a UI to pick different characteristics of a movie like Director, Budget, Genre, etc and then provide a prediction on the different measures of success i.e. Imdb rating, popularity and gross profit.

We all are students from the Department of Computer Science at the University of Victoria. There are three members in our team:

- Harpreet Singh (Grad., Email: hsingh91@uvic.ca),
- Xinhai Zhou (Undergrad., Email: zhouxinh@uvic.ca),
- Zelan Xiang (Undergrad., Email: zelan@uvic.ca)

Related Work

There has been a substantial amount of research on this topic. MIAS [1] provides early predictions of movie profitability. Ramesh Sharda uses neural networks to predict financial performance of a movie but considers the problem to be a classification problem rather than forecasting a point estimate [2]. Márton Mestyán and Taha Yasseri measured and analyzed the activity level of editors and viewers of the corresponding entry to the movie in Wikipedia and built a multivariate linear regression model for predicting box office revenue [3]. Kyuhan Lee and Jinsoo Park use a non conventional approach in which they add a new feature that incorporates transmedia storytelling and apply an ensemble approach. As a result of this they propose a new model which outperforms models from previous studies [4].

Data Source

Instead of scraping the TMDB website to obtain the dataset, we will use the existing kaggle TMDB 5000 movies dataset [5]. This dataset was generated from The Movie Database API by kaggle, but is not endorsed or certified by TMDB. The dataset include two data, movie metadata and movie credits data. The movie metadata attributes include: budget, genres, homepage, id, keywords, original-language, original-title, overview, popularity, production_companies, production_countries, release_date, revenue, runtime, spoken_languages, status, tagline, title, vote_average, and vote_count; the movie credits data attributes include: movie_id, title, cast, and crew.

Methodology

We will first collect movie data as our dataset from kaggle, and we will separate the set into training set and testing set according to the knowledge learned from the data mining course. Then we apply different existing classification and regression models in Scikit-learn [6] with our training data set. After comparing the results from different models in Scikit-learn, we will choose the one with the best performance, and then try to implement the algorithm in Python from scratch. Finally, we will build a UI that allow the user to input features of a movie and get the predicted result.

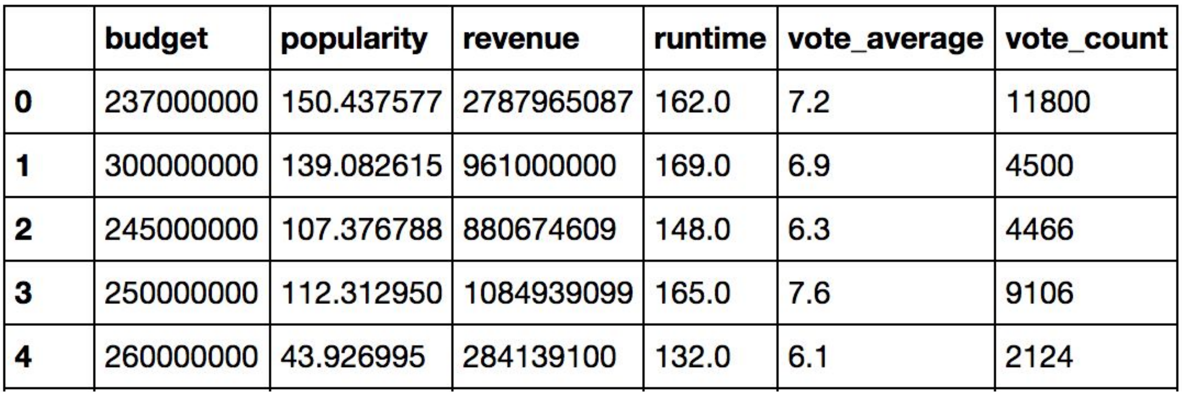
Overall, in this project we will use tools such as Python, Scikit-learn, Flask for backend and Angular [7] for the webapp.

Progress Report

Work Have done

Currently, we have decided our group members, and have reassigned tasks to each member: Harpreet collected and processed data, and then apply them with existing classifiers, Zelan will work on the classifier prototype, Xinhai will work on documentation and UI, and the rest of work will be done together.

The data has been collected through the existing kaggle TMDB 5000 movies dataset. We filtered numeric data for our current classifiers, and have dropped some useless columns from the original dataset. Here is the part of our dataset:



	budget	popularity	revenue	runtime	vote_average	vote_count
0	237000000	150.437577	2787965087	162.0	7.2	11800
1	300000000	139.082615	961000000	169.0	6.9	4500
2	245000000	107.376788	880674609	148.0	6.3	4466
3	250000000	112.312950	1084939099	165.0	7.6	9106
4	260000000	43.926995	284139100	132.0	6.1	2124

After that, we removed NaN values from the dataset, and then generated a heatmap to show the correlation. Then, we initialized three classifiers (decision tree classifier, SVC and naive bayes classifier) with our dataset and used k-fold validation to compare the performance. According to the incomplete dataset, the accuracy of classifiers are quite low (decision tree classifier: about 36%, SVC: about 42%, naive bayes: about 40%). We expect the accuracy will become more acceptable after we improve our current dataset.

We are currently working on improving our dataset with a python library called ‘tmdbsimple’[8] by pulling information from TMDB about popularity, imdb rating for each director or cast, and movies they have worked on. After collecting the data, we will add it into our dataset for our classifier prototype.

Difficulties

We do have encountered some difficulties when doing our work. Firstly, we lost one team member since he dropped the course, so we had to reassign tasks to the rest of members. We will still stick to the original plan, but the prototype may be delayed because we now have only three people inside the team. We will apply various existing classifiers (decision tree classifier, SVC and naive bayes classifier), and then choose the classifier with the best performance to build from scratch. Secondly, the original dataset missed some information we want, so the performances for each classifiers are bad. The dataset does not include the columns about ratings of directors or cast and the columns about facebook likes of directors or cast (how many likes a director or a cast got on the social media), and we need to find a way to obtain the data and add them into the dataset.

Updated Time Line

Our project starts on Oct 9th, 2017 and is estimated to finish by Nov 28th, 2017. Here is the updated timeline of our project:

Oct 13th, 2017: Decide the group members and the topic of the project (done)
Oct 18th, 2017: Detailed tasks assigned to each team member (done)
Oct 24th, 2017: Finish data collection and pre-processing (done)
Nov 8th, 2017: Apply various existing classifiers with our dataset (done)
Nov 11th, 2017: Improve dataset according to the performance of the classifiers (working)
Nov 11th, 2017: Come up a classifier prototype (working)
Nov 20th, 2017: Finish the data improvement and prototype
Nov 22th, 2017: Finish project UI
Nov 23th, 2017: Done with data analysis
Nov 26th, 2017: Finish the final report
Nov 28th, 2017: Final presentation

References

[1] M. Lash and K. Zhao, "Early Predictions of Movie Success: The Who, What, and When of Profitability", Journal of Management Information Systems, vol. 33, no. 3, pp. 874-903, 2016.

[2] R. SHARDA and D. DELEN, "Predicting box-office success of motion pictures with neural networks", Expert Systems with Applications, vol. 30, no. 2, pp. 243-254, 2006.

[3] M. Mestyán, T. Yasseri and J. Kertész, "Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data", PLoS ONE, vol. 8, no. 8, p. e71226, 2013.

[4] K. Lee, J. Park, I. Kim and Y. Choi, "Predicting movie success with machine learning techniques: ways to improve accuracy", Information Systems Frontiers, 2016.

[5] "TMDB 5000 Movie Dataset | Kaggle", Kaggle.com, 2017. [Online]. Available: https://www.kaggle.com/tmdb/tmdb-movie-metadata. [Accessed: 14- Oct- 2017].

[6] P. Fabian, et al. “Scikit-learn: Machine learning in Python.” The Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[7] "Angular Docs", Angular.io, 2017. [Online]. Available: https://angular.io/about?group=Angular. [Accessed: 14- Oct- 2017].

[8]"celiao/tmdbsimple", *GitHub*, 2017. [Online]. Available: https://github.com/celiao/tmdbsimple/. [Accessed: 14- Nov- 2017].