IMDB Movie Rating Score Classification

Using IMDB 5000 MOVIE Dataset from Kaggle

GROUP 2

Jingwen Zhong | Chenglu Xia

DSCC465 Intermediate Statistical methods – Final Report

Instructor: Yusuf Bilgic

May. 9th, 2021

Abstract: This report introduces the process of classifying IMDB rating scores of 5000 movies, spanning across 100 years in 66 countries, by implementing KNN, K-means and feature selection methods in modeling with R.

Keywords: High Dimensional; Classification; Supervised; Unsupervised; Feature selection; KNN; K-means; R

INTRODUCTION

Background

Most people enjoy kicking back and watching movies in their free time. With so many films out there and so little time to watch them, it is unwise to spend that precious time figuring out what to watch. The rating system on IMDB is one great resource that allows users to pick a movie quickly, in combination with it's massive user database as well as being one of the world's most popular and leading authorities on movie, television and celebrity content, higher ratings for a movie usually mean it is more popular and enjoyable.

Goal

To classify movies into suitable classes with optimal methods so that users can get a more precise view when selecting movies and find out the importance factors that related to the classification. The expected suitable classes contain labels bad, ok, good and excellent with the IMDB score <4, 4~6, 6~8, and 8~10 respectively.

Data Description

The IMDB movie dataset from Kaggle contains 28 variables and 5043 observations which were retrieved from 1916 to 2016 in 66 countries. The response variable in our project is “imdb_score” and the other variables are the possible predictors.

METHODOLOGT

1. Supervised Method: KNN

[1]K-Nearest Neighbors is a supervised learning statistical method for classification that can be used in both quantitative and qualitative response variables, it is preferred when attempting to estimate the conditional distribution of the response variables giving predictors, then it performs a classification by the highest estimated probability of the given observations to the class (James et al. 2013).

2. Unsupervised Method: K-Means Clustering

[1]K-means clustering is one of the most used unsupervised statistical learning methods which is a simple and elegant approach for partitioning a data set into k distinct groups, where k is the number of groups pre-specified by the analysis. In K-Means clustering, each cluster is represented by its center which corresponds to the mean of points assigned to the cluster (James et al. 2013).

3. Feature Selection: Boruta

[2] Boruta in R is a relevant feature selection wrapper algorithm using random forest for classification problems and outputs variable importance measure (VIM); [3]This method tries to capture all relevant features instead of minimal-optimal features, that it , it competes features with randomized version of them instead of competing among themselves. To eliminate the

variables, it runs a random forest classifier on the shuffled copies of all variables and finds the threshold score, then performs a test among the attributes. It then permanently removes the attributes with the importance that is significantly lower than the threshold. Repeat the process.

4. Improvement: Resampling & Oversampling

[1] Random resampling and oversampling could rebalance the imbalanced dataset. Random resampling methods such as Cross Validation draw samples repeatedly from the training dataset and fit a model to each new sample, and it also allows us to obtain additional information that we could not get from the original training set (James et al. 2013).

[4] Oversampling aims to modify an imbalanced dataset into balanced distribution. On the one hand, this method leads to no information loss. One the other hand, since oversampling simply adds replicated observations in the original dataset, it ends up adding multiple observations of several types, thus leading to overfitting.

DATA ANALYSIS

1. Data Preprocessing

- ***Fill in NAs***

There are 12 variables containing missing values in this dataset. We impute missing values of numeric variables with median and drop NAs for “title_year” and “aspect_ratio” columns.

- ***Remove Columns***

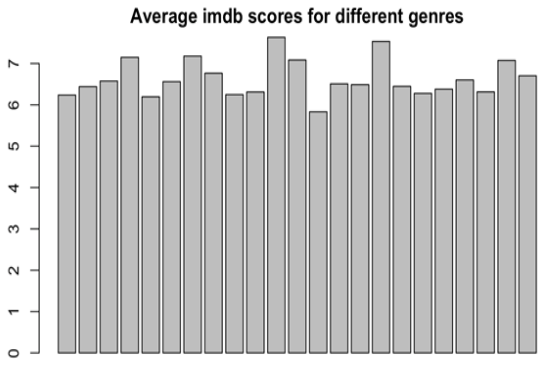

Figure 1: Average imdb_score by different genres

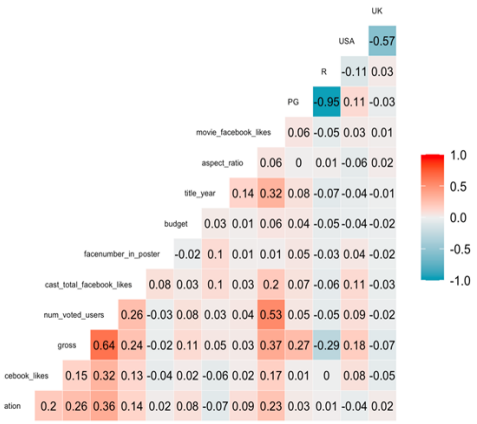

Figure 2: Correlation between numeric variables

1. We remove “color”, “language”, “genres” columns because around 96% movies are colored, and 95% movies are in English and showing from Figure 1 above which means they are nearly constant.

2. We remove “actor_1_facebook_likes”, “actor_2_facebook_ likes” and “actor_3_facebook_likes” since they are highly correlated variables to “cast_total_facebook_like”. (Figure 2)
3. We remove name variables such as “directro_name”, “actor_name” and “movie_imdb_link” since they have numerous unique values which is not helpful for the classification models.

- ***Combine Some Values***
  1. “content_rating”: Replaced ‘M’, ‘GP’, ‘PG-13’ with ‘PG’; replaced ‘X’, ‘Approved’, ‘Not Rated’, ‘Passed’, ‘Unrated’ with mode ‘R’; replaced others with ‘Others’.
  2. “country”: Replace other countries except ’USA’ and ‘UK’ with ‘Others’.
- ***Encode Categorical Variables***

  Encoded categorical variables using one-hot encoding method.

- ***Bin Response Variables***

  Binned the “imdb_score” into 4 buckets: less than 4, 4~6, 6~8 and 8~10, which represents bad, ok, good, excellent.

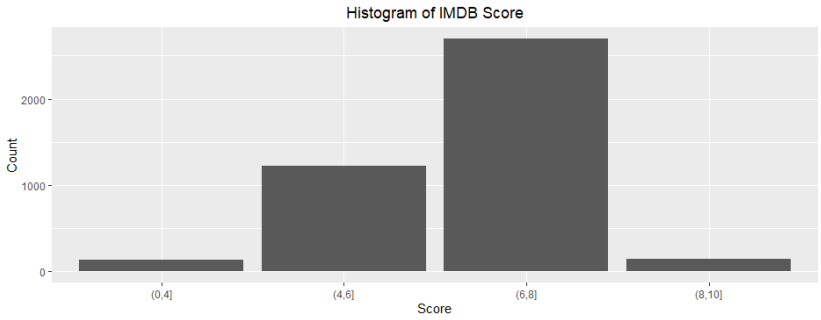

Figure 3: Binned Response Variables

The final date we use includes 4193 observations and 15 variables.

2. KNN

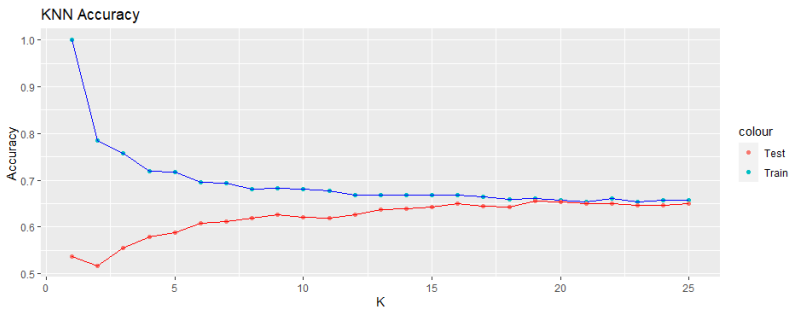

Figure 4: Accuracy Plot of KNN

For the KNN algorithm, first We take 80% of the data as a training set, 20% as a testing set. We then use the KNN algorithm in the class library of R, along with a grid search we

designed to find the best k using accuracy. Figure 4 above shows the changing of the accuracy of the KNN algorithm from k=1 to k=25. The best K we select is 19, with the highest accuracy 0.655 which still has rooms for improvement.

3. K-Means

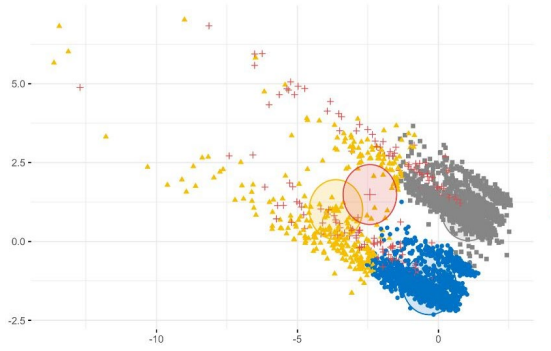

Figure 5: Center = 4 for K-Means

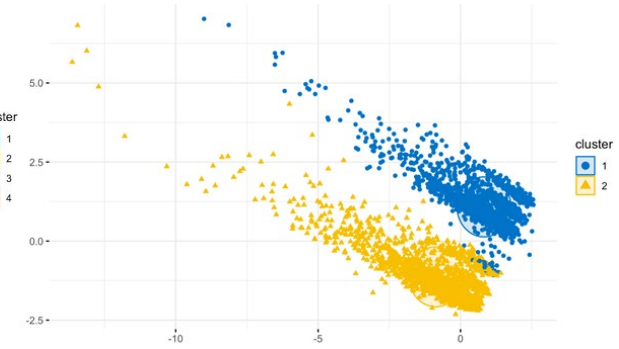

Figure 6: Center = 2 for K-Means

For the K-Means algorithm, we first standardized scale our data using scale function in R. Then we use the K-Means algorithm in R, along with a grid search we designed using accuracy to find the best number of centers. Figure 5 above shows 4 clusters with the highest accuracy 0.354. However, 'Accuracy' of unsupervised methods is not the best choice since no true labels are involved in these kinds of methods. Thus, we use the average silhouette approach in library factoextra with the function fviz_nbclust to measure the quality of clustering and determine the best number of clusters, and plot a graph showing clusters. Figure 6 above is the best cluster result of the algorithm. Clearly, clustering with 2 centers is better than the one with 4 centers.

4. Feature Selection - Boruta Package

After using Boruta algorithm in Boruta package, 13 attributes were confirmed important: *num_voted_users, duration, gross, budget, movie_facebook_likes, title_year, cast_total_facebook_likes, PG, R, USA, director_facebook_likes, UK, facenumber_in_poster* (listed in descending order). One variable *aspect_ratio* is confirmed unimportant. (Figure7)

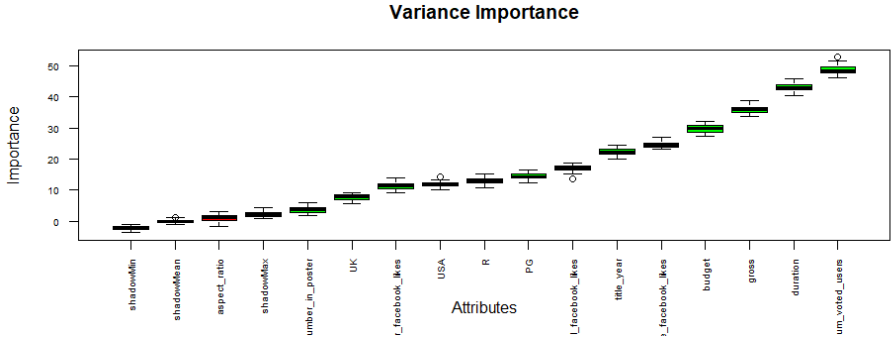

Figure 7: Variance Importance Plot

5. Improvement

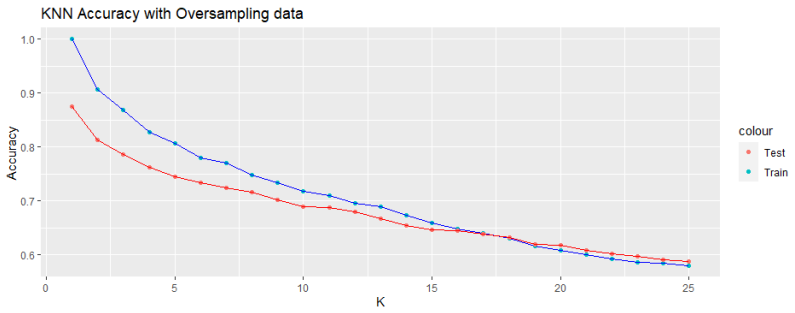

Figure 8: Accuracy Plot with Oversampling of KNN

Since the data is imbalanced (Figure3), we try both oversampling method with RandOverClassif function in library UBL, and cross-validation(k=10) method implemented by ourselves. As we can see from Figure 8 above, oversampling of our dataset creates a serious overfitting problem. Cross-validation on the other hand does not create an overfitting problem but the average accuracy of the test dataset for 10 times cv is only around 0.617 which is worse than the first case. Therefore, our improvement methods might not work on our dataset with KNN algorithm.

DISCUSSION

From the above, our methods for improvement do not work, but we still want to improve the classification accuracy. We can try other data preprocessing methods to deal with our original data or we can try other models to do this classification. In fact, we tried Linear Discriminant Analysis and the accuracy reached 0.69.

CONCLUSION

By using KNN algorithm, we obtained the highest accuracy 0.655 of classifying IMDB ranking score into 4 classes. By using the K-Means algorithm, we obtained the conclusion that 2 classes for our data are more suitable for 4 classes. By using the feature selection model, we select 13 features that are important to our classification. In the improvement step, we use the 13 features instead of 14 features and cross validation method for resampling the data, and we get the accuracy 0.64. We also tried the oversampling method, but it creates an overfitting problem. Thus, if we want to improve the results of our classification problem, we either can improve our dataset or try other models, such as Linear Discriminant Analysis model.

REFERENCE

Yueming Zhang: Predict IMDB Score with Data Mining Algorithms. Retrieved May. 8, 2021, from: https://www.kaggle.com/carolzhangdc/predict-imdb-score-with-data-mining-algorithms

[1]James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

[2]Feature selection with the Boruta algorithm. Retrieved May. 8, 2021, from: https://search.r-project.org/CRAN/refmans/Boruta/html/Boruta.html

[3]Aleksey Bilogur:Automated feature selection with boruta. Retrieved May. 8, 2021, from: https://www.kaggle.com/residentmario/automated-feature-selection-with-boruta

[4]Practical Guide to deal with Imbalanced Classification Problems in R. Retrieved May. 8, 2021, from: https://www.kaggle.com/residentmario/automated-feature-selection-with-boruta