Predict Satisfaction of Movie Recommendation System based on Users' Personalities

Haoda Song

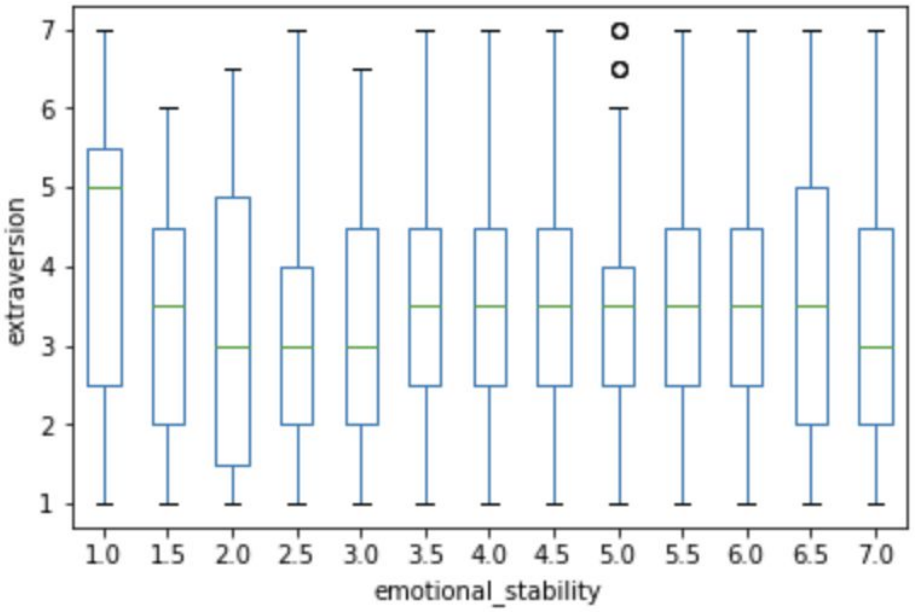
BrownUniversity

Introduction:

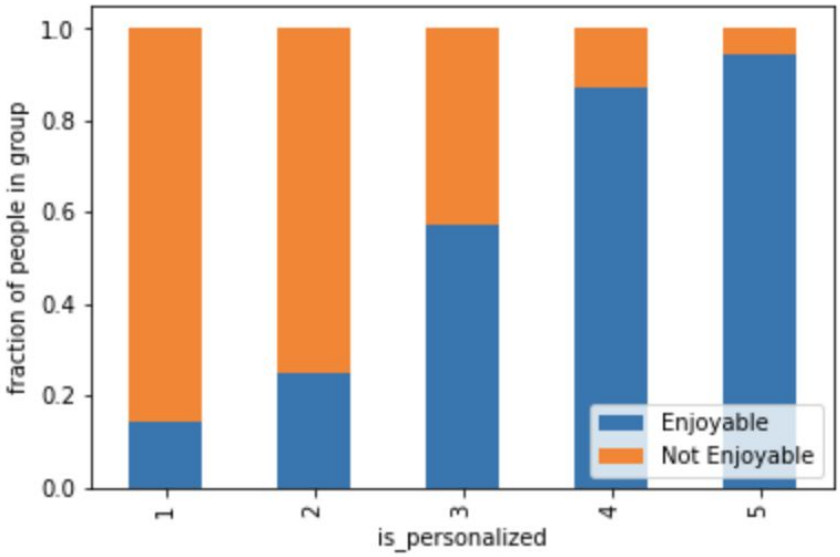
In this project, I am going to classify if the movies on the recommendation list are enjoyable based on the personalities of users. In the future, once we integrate the personality features, the movie recommendation system would show more accurate recommendations. Therefore, obviously the target variable is the "enjoy_watching" variable in the "personality" dataset. This Kaggle dataset, "2018-personality-data", provided by Arslan Ali contains 1834 observations with 34 features and no missing values. The data description can be easily found on the homepage of the link. However, 24 columns are the 12 movie IDs corresponding to the 12 rating scores. Therefore, in order to show the overall performance of the ratings, I took the average of them to get the new feature, "avg_ratings". Finally, I have 9 features to predict the target variable and distinguish the target variable into two groups (enjoyable/not enjoyable) by the scores that greater or equal than/ less than 4, which makes sense because people rated 3 may not enjoy the list of movies too much. Beside the target variable, in the 9 features, 8 of them are categorical features representing the survey scores and 1 of them is the continuous variable representing the average ratings of the movies.

In Kaggle, completed public projects are rarely shown about this data. The data was used for the project "Let's Understand & Implement KNN". The author's goal is to estimate which users rate movies in which categories the most by the K-nearest-neighbor regression method. Finally, he chose the K=50 or 30 by KDTree but didn't provide any evaluation metrics. Also, another project, called "Ocean features EDA and Clusters", is related to the data explorations. His/her goal may be about unsupervised learning while trying to get some information from the data. The author used pairwise scatter plots in EDA and implemented PCA and Clustering with 3D visualization to explore the data features.

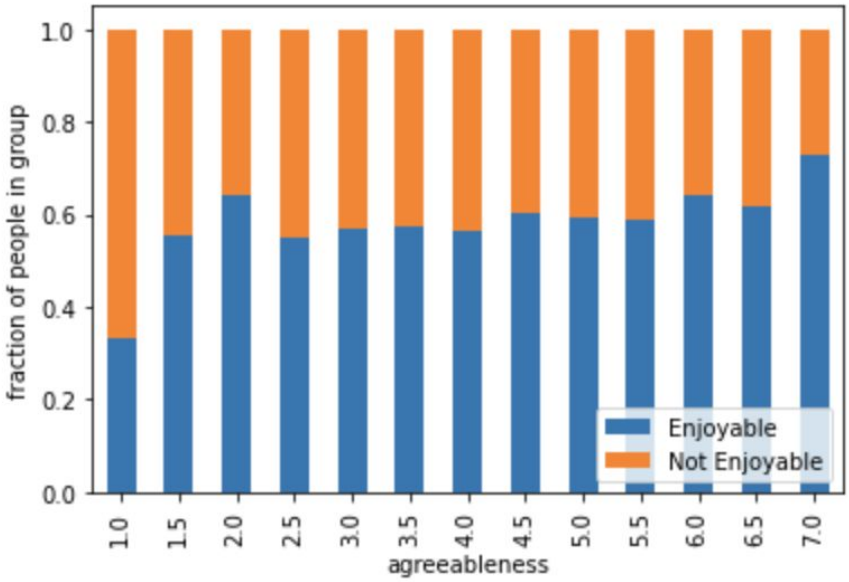
Exploratory data analysis:



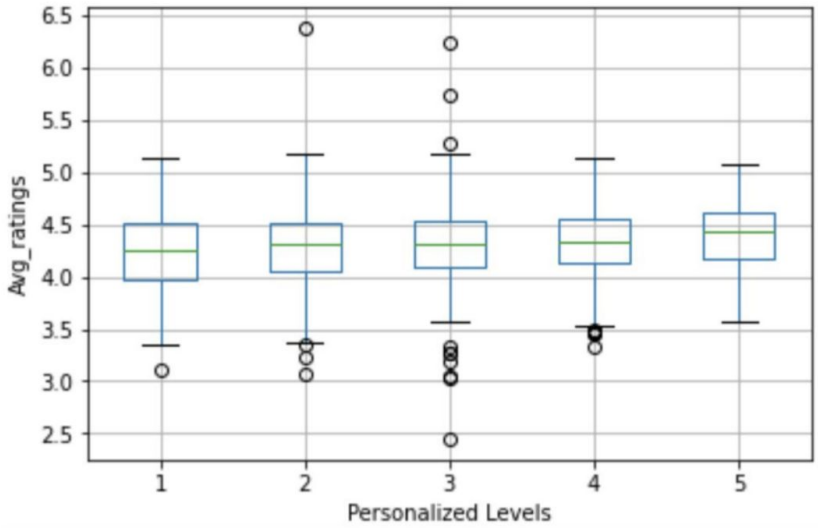
The above boxplot shows the relationship between two personalities, emotional_stability and extraversion. There are some interesting findings: If we take a look at the mean in the emotional stability groups from 3.5 to 6.5, these people tend to have constant extraversion proportion, around 3.5. However, the people who have the least emotional stability rated themselves' extraversion with the average 5 score which is well above any other groups.



The above figure shows the relationship between the response of the user to the question "The list is personalized for me" and the fraction of the people in the group which feel the movies in the list are enjoyable/not enjoyable. It makes sense because typically people who believe this list is personalized for them usually enjoy watching the movies in the list.



The above figure shows the relationship between the assessment scores of agreeableness (1-NOT have such a tendency 7-have such a tendency) and the fraction of the people in the group which feel the movies in the list are enjoyable/not enjoyable. The interesting thing is that the personalities of agreeableness seems to have very weak correlation with our target variable. As I expected, the more agreeableness, the larger proportion of that they believe the list is enjoyable. However, Almost each group, besides agreeableness-1 and 7, have pretty similar levels/proportions of views of the movies.



Here is the finding that surprised me. I believe that people who think the list is personalized for them would also rate the list of recommended movies at relatively high scores. The truth is, however, there is no obvious correlation between these two

variables. They may have a very weak correlation since the range between 25% and 75% percentiles becomes narrower as the personalized level increases.

Methods:

Data Splitting:

I splitted the data into three parts: train, validation and test. The data contains 1834 observations which can be considered as a small dataset. First, I splitted the dataset into two sections. 20% of the original dataset goes to the test set. I applied the StratifiedKFold method to the rest of the dataset in order to keep similarity and stable proportions in each fold we splitted.

K Fold

```
train balance:
Enjoyable        0.5763
Not Enjoyable    0.4237
Name: enjoy_watching, dtype: float64
val balance:
Enjoyable        0.588435
Not Enjoyable    0.411565
Name: enjoy_watching, dtype: float64
train balance:
Enjoyable        0.568627
Not Enjoyable    0.431373
Name: enjoy_watching, dtype: float64
val balance:
Enjoyable        0.619048
Not Enjoyable    0.380952
Name: enjoy_watching, dtype: float64
```

Stratified K Fold

```
train balance:
Enjoyable        0.578858
Not Enjoyable    0.421142
Name: enjoy_watching, dtype: float64
val balance:
Enjoyable        0.578231
Not Enjoyable    0.421769
Name: enjoy_watching, dtype: float64
train balance:
Enjoyable        0.578858
Not Enjoyable    0.421142
Name: enjoy_watching, dtype: float64
val balance:
Enjoyable        0.578231
Not Enjoyable    0.421769
Name: enjoy_watching, dtype: float64
```

Although the data should not be considered as the imbalance data (Enjoyable : Not Enjoyable = 58% : 42%), the StratifiedKFold still improves our balance in cross validation. For example, the differences among each subgroup in K Fold occur at the hundredth, whereas the differences in Stratified K Fold occur at the thousandth. Also, Based on the author's research paper page 6, the features should not come with any bias and grouped structure. Therefore, I have confidence that this method is reasonable.

Data Preprocessing:

In this dataset, 8 features which are used as the predictors are categorical variables and 1 feature is a continuous variable. Therefore, the Ordinal Encoder was used in seven out of eight variables, "openness, agreeableness , emotional_stability , conscientiousness , extraversion, is_personalized, assigned condition" since, besides "assigned condition" (from

low to high), they ranged from one to seven and in the survey 1 represents “Not have such a tendency” and 7 represents “have such a tendency”. Thus, these seven categorical variables should be considered as ranked categorical variables. Then, “assigned metrics” with the levels of “serendipity, popularity, diversity, default” is obviously an unordered categorical variable so that OneHotEncoder fits the situation. Finally, the MinMax method is applied on the continuous variable, “avg_ratings” since the ratings are bounded reasonably by the range from 0 to 7.

Machine Learning Pipeline:

Once data preprocessing completed, I applied four machine learning algorithms, logistic regression, random forest, super vector machine and k-nearest neighbor, on this classification problem. Also, for reproducibility and certainties of the models, 10 random states are chosen to make sure the results by each model are reliable. For each run, I used the GridSearchCV method to tune the parameters to find the model with the highest accuracy score of predictions we could get.

- **Logistic regression**: The parameters I chose to tune are “solver ('newton-cg', 'lbfgs' and 'liblinear')”, “penalty (l1 and l2)” and “C (100, 10, 1.0, 0.1, 0.01)”. The GridsearchCV finding the best combination of these parameters shows the model with C = 0.01, penalty = l2 and solver = newton-cg has the highest accuracy score.

- **Random Forest Classifier**: The parameters I chose to tune are “number of estimators (1,10,30,50,100,150)”, “max of depth (1,2.5,5,7.5,10)”, “min of samples split (2,5,10)”. The GridsearchCV shows the model with max_depth = 10, min_samples_split = 5, n_estimators = 100 has the highest accuracy score.

- **Super Vector Machine Classifier:** The parameters I chose to tune are the types of the kernels (linear kernel, polynomial kernel , radial basis kernel, sigmoid kernel and precomputed kernel) and the C values (100,10,1,0.1,0.01). The 10 best models generated by GridsearchCv are not quite similar since the chosen kernels and C values are different in each run. So, I have a high uncertainty of the results from 10 super vector machine models.

- **K-Nearest Neighbors Classifier:** The parameters I chose to tune are the number of neighbors (1,10,30,50,100), the leaf size (5,10,20,30,40), the p score (1 or 2) and the weights methods (uniform or distance). The best model generated by the GridsearchCV method is the model with leaf size = 5, n_neighbors = 100, p = 1 and weights = distance has the highest accuracy score.

As you can see, the “Accuracy” is used as my evaluation metric since our goal is a classification problem as well as the accuracy score is an evaluation metric which can be

understanded, interpreted and explained easily. Although my data is slightly imbalanced (55/45), f-1 scores from different models are quite the same. Therefore, accuracy would be an alternative common choice.

The uncertainty of the logistic regression model is mainly from the linear regression assumptions so that it cannot handle the non-linear problems. However, we don't know the linearity of our data before we fit the model. Although the other three models don't have this issue, they are easily overfitting. To avoid this issue, we always tune reasonable parameters manually. Although we tried our best to optimize the scores, it's impossible that we could always reach the optimum manually for each run because the parameters we offered to the machine are limited. Therefore, in order to minimize the uncertainty for each model, Grid Search Cross Validation and ten random states were used in the process.

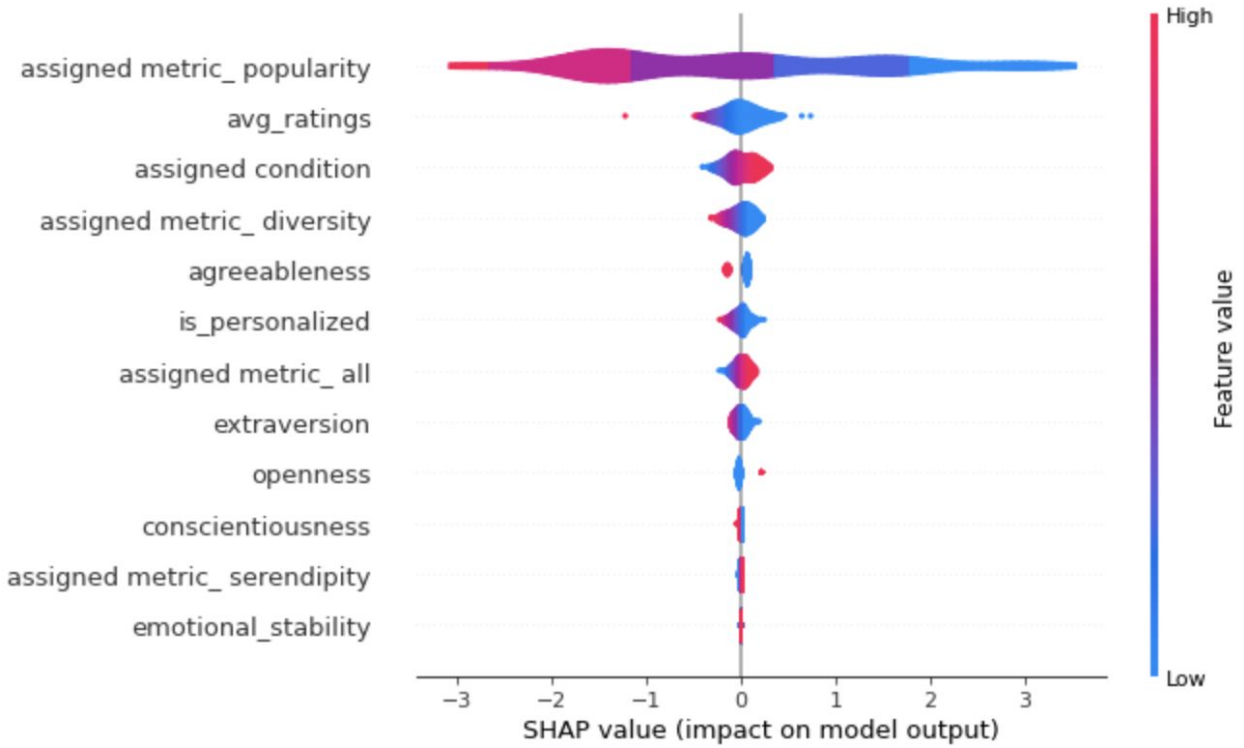
Results:

```
Logistic Accuracy: 0.7667574931880109 STD: 0.01515469835757618
RFC Accuracy: 0.7675749318801091 STD: 0.021186150057390937
SVC Accuracy: 0.7547683923705722 STD: 0.01935265735936651
KNNC Accuracy: 0.7370572207084468 STD: 0.024615495427552484
```

The test accuracies with the corresponding standard deviations are shown above. We could see there are quite closed test accuracy scores between logistic regression and random forest. However, the two corresponding standard deviations tell us that we have more certainty that the accuracy score from logistic regression would be more stable than the accuracy from the random forest algorithm. Therefore, I would like to choose the logistic regression model with C = 0.01, penalty = l2 and solver = newton-cg as the most accurate model.

Therefore, the baseline accuracy is 0.613. Compared to the accuracy score that we got from the best logistic regression, the model accuracy was improved by 25% above the baseline accuracy. In order to find how the best model returns the improvement, I tried to know the most/least important features in our dataset.

As the graph below shows, the "assigned metric_popularity" is the most important feature and also extremely more important than other features to predict if people enjoy watching the recommendation list. These two variables are correlated since most people would enjoy watching the "popular" movies. In other words, one of the measurements of "popularity" is that people like to watch movies. Also, the least important feature is the emotional stability. It means the emotional stability does not quite affect the enjoyability of the movies.



Based on the global feature importance above, the personalities don't have pretty much influence on the enjoyability of the recommended movies because the personalities features are all at the middle of the ranked importances. This also means one unexpected thing that I would get a pretty bad or very different prediction if I drop the assigned metric or assign the metric again.

Outlook:

In order to improve the accuracy of our data, I believe many different movies and features (both personalities and movie features) should be added in the recommendation list of the movies to do the survey because the sample of 12 movies seem not enough to represent the population. In the aspect of the model improvement, we could tune the parameters in detail and try different new models, e.g. XGBoost, Naive Bayes Classifier and Neural Network, to see if they would give us a better prediction. Also, trying different evaluation metrics may provide different decisions to us.

However, based on this data, I am likely to believe that the subject influence may affect the quality of the data. Many people who took the survey tended to evaluate their personalities subjectively but their real personalities are not like what they evaluated in the survey. For example, many people are very likely to believe they are optimistic in public. Another guess is that there may not exist obvious relationships between personalities and the enjoyability of the recommended list of movies.

Reference:

Kaggle dataset: Top Personality:
https://www.kaggle.com/arslanali4343/top-personality-dataset

Scatterplots: http://www.cs.umd.edu/hcil/trs/2016-10/2016-10.pdf

Seaborn Plots: https://seaborn.pydata.org/generated/seaborn.boxplot.html

Github:

https://github.com/Haoda1860/Movie-Satisfaction-Prediction