Intro to Data Science

DS-UA 112, Summer 2022

Pascal Wallish

Candice Yao

N11895239

Capstone Project

a) Introduction: Handling of dimension reduction, data cleaning, and data transformation.

i. Dimension reduction:

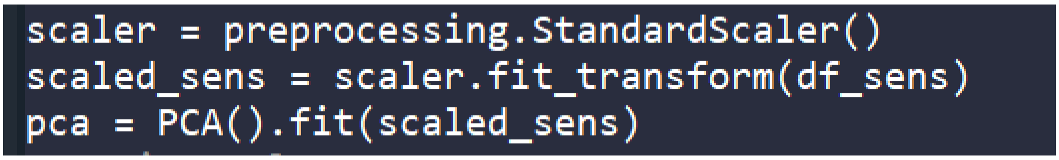
In this project, I handled dimension reduction using PCA (principal component analysis) when a visible degree of collinearity is displayed in the correlation between the variables of interests. I only conducted PCA whenever there is this precondition is fulfilled and there is productive purpose to do so for the following data processing. For example, I implemented PCA for investigation of group-wise correlation and K-means clustering for the first two questions. However, I did not conduct PCA but instead keep the raw data when I construct random forests to classify movie ratings for the last three questions.

ii. Data cleaning

I handled NaNs or null values in the dataset differently, catering to the different purposes of each question. For question like Q1 and Q2, since there are a large number of n, row-wise (participant wise) dropping of NaNs will not decrease the size of the sample. Also, since many questions exhibit opposite psychological connotations (e.g. "is outgoing/sociable" vs. "Is sometimes rude to others", it is unreasonable to compute the mean and do imputations with the mean values. On the other hands, when handling missing movie ratings, I imputed the null data with the mean or the median of the specific movie whose rating is missing. By doing so, I avoided dropping row(participant)-wise data, which is going to largely decrease the amount of data; and also element wise dropping, which is going to render the size of the matrix uneven and the information incomparable between participants.

iii. Data transformation

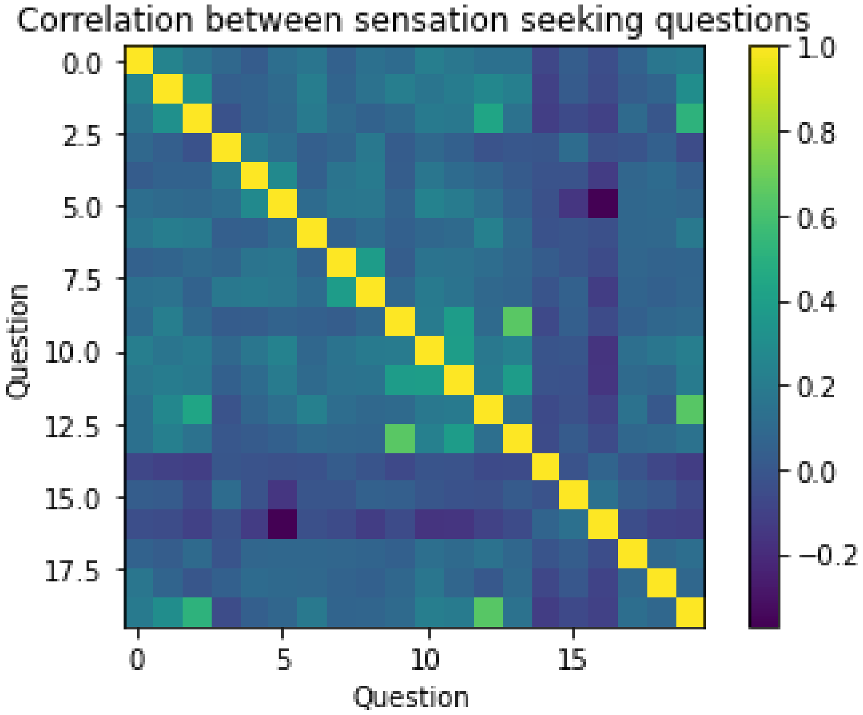
Throughout the project, I preprocessed the data to let them have a center at zero and standard deviation of 1 using sklearn.preprocessing.StandardScaler. I used this as a convenient equivalent of stats.zscore to handle DataFrames.

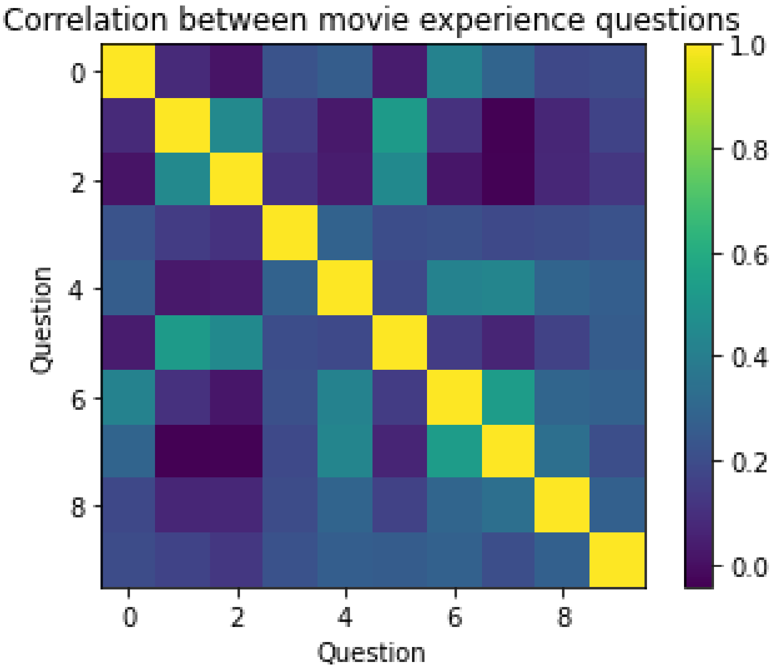


```
scaler = preprocessing.StandardScaler()
scaled_sens = scaler.fit_transform(df_sens)
pca = PCA().fit(scaled_sens)
```

In question 8-10, I chose to use random forest classification as a prediction model, which requires the output values have to be discrete. I discretize the ratings by multiplying all ratings with 100 for these questions in order to create the tree.

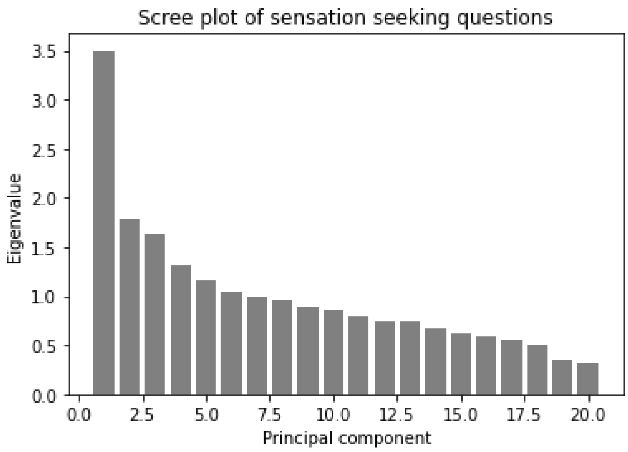
Question 1: What is the relationship between sensation seeking and movie experience?

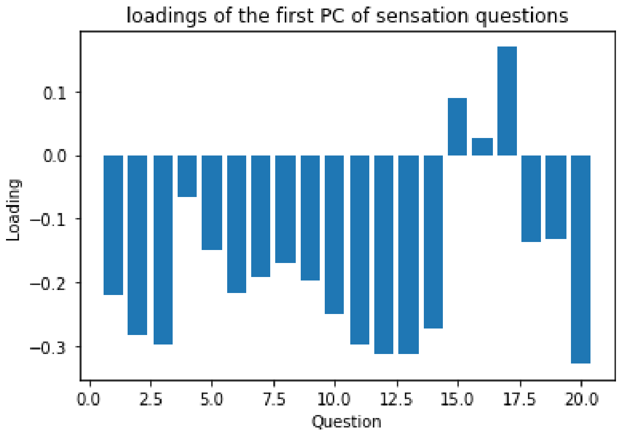


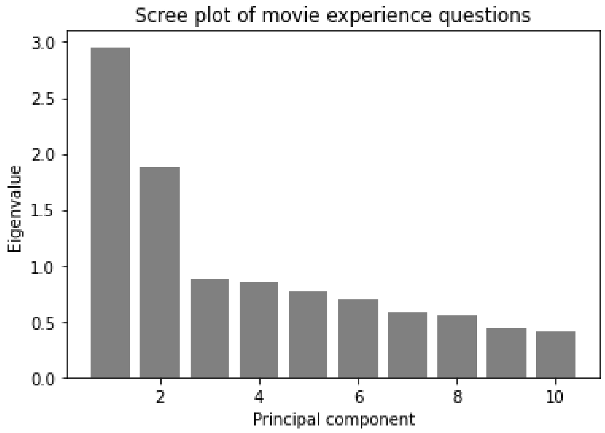


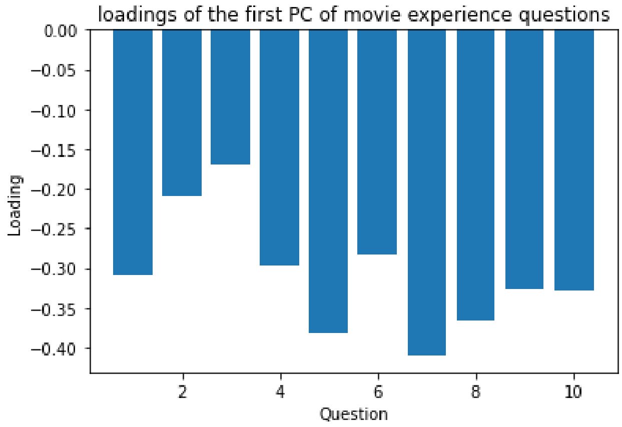
By investigating and visualizing the correlation within the group of sensation seeking questions and movie experience questions, I discovered that there are a visible degree of in-group correlation, as displayed on the light green blocs on both of the correlation heat graphs.

Thus, a PCA procedure for both groups of question is indicated.

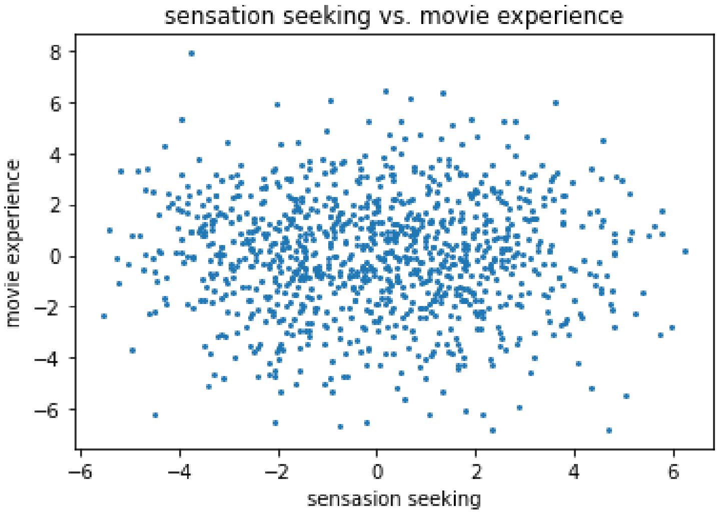

Scree plot of sensation seeking questions

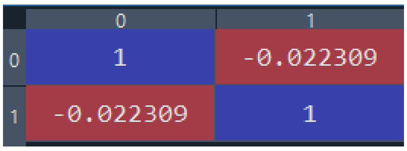

loadings of the first PC of sensation questions


Scree plot of movie experience questions


loadings of the first PC of movie experience questions

By investigating the questions alone with the loading and the loading plots of both groups, we can confirm that the first principal component of sensation seeking questions is affected by most questions in the group towards the same directions except for three questions, so we can reasonably reduce the questions to the first PC since it captures the most important variances in the data. As of movie experience questions, the loading for every question in this category points to the same direction, so we can also confirm that this group can be reduced to its first PC.

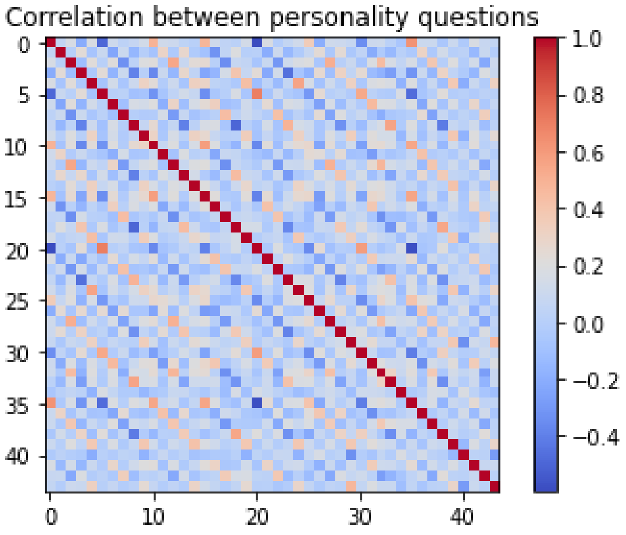




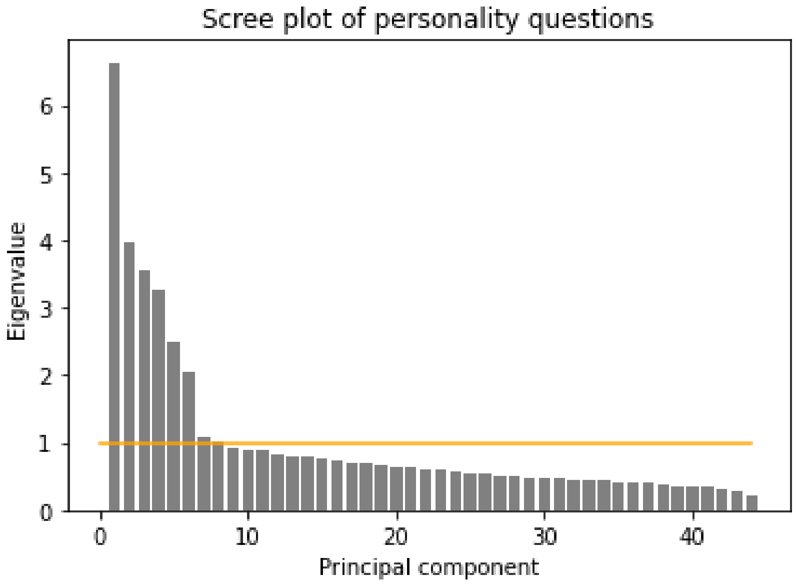
	0	1
0	1	-0.022309
1	-0.022309	1

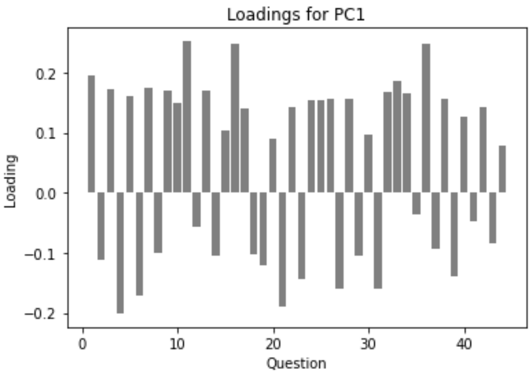
The rotated data is plotted on the new coordinate with PC1 for sensation seeking on the x-axis and PC1 for movie experience on the y-axis. The data seems to display a broad clustering structure at the lower half of the new coordinate, but the pattern is not strong. Computing the correlation matrix outputs a correlation coefficient of -0.022 between sensation seeking PC1 and movie experience PC1. This shows us that there is only a mildly negative correlation between the two categories of questions.

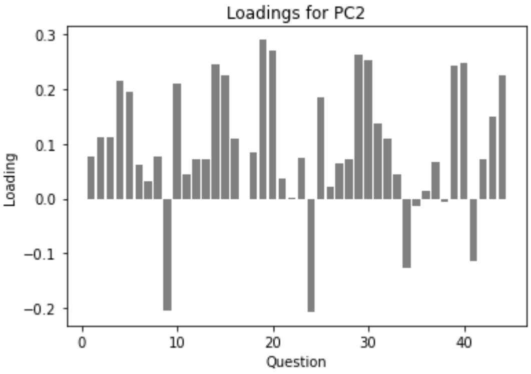
Question2 :

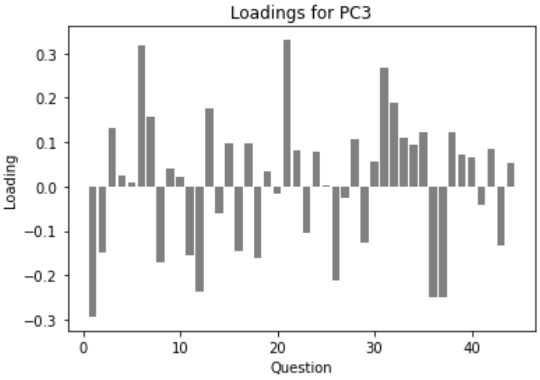


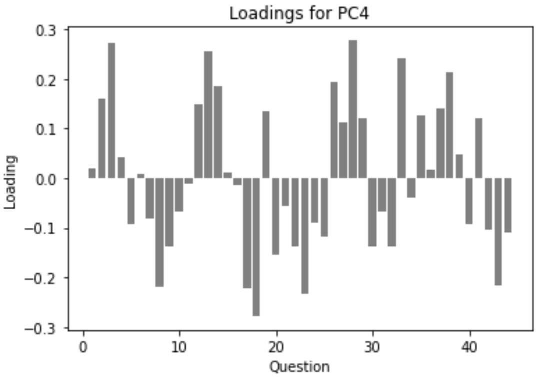
The correlation graph between the personality questions indicates that there isn’t no in-group correlation between the personality questions. Therefore, a PCA is indicated to reduce the dimensionality.

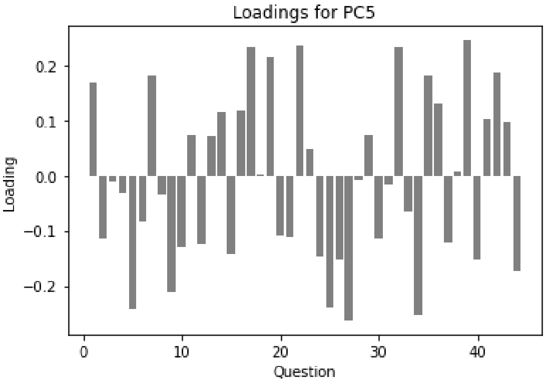

Scree plot of personality questions
Eigenvalue
0
1
2
3
4
5
6
Principal component
0
10
20
30
40

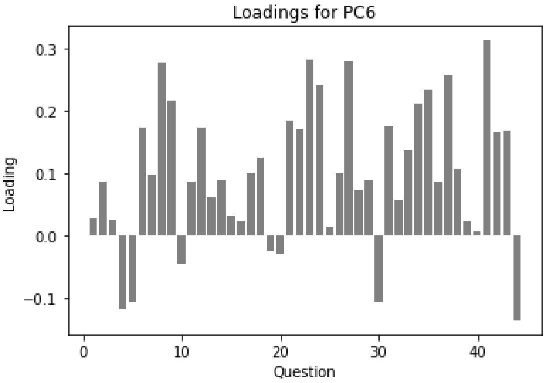

Loadings for PC1
Loading
0.2
0.1
0.0
−0.1
−0.2
Question
0
10
20
30
40

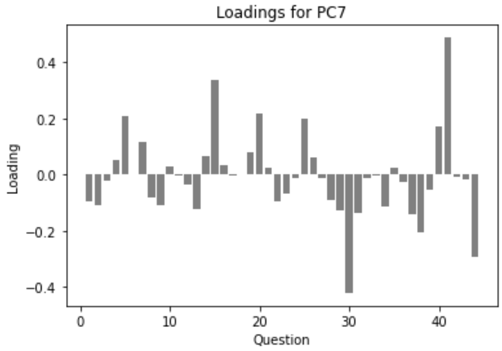

Loadings for PC2
Loading
0.3
0.2
0.1
0.0
−0.1
−0.2
Question
0
10
20
30
40

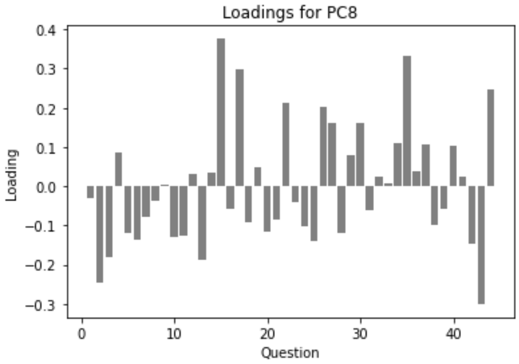

Loadings for PC3
Loading
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Question
0
10
20
30
40


Loadings for PC4
Loading
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Question
0
10
20
30
40


Loadings for PC5
Loading
0.2
0.1
0.0
−0.1
−0.2
Question
0
10
20
30
40

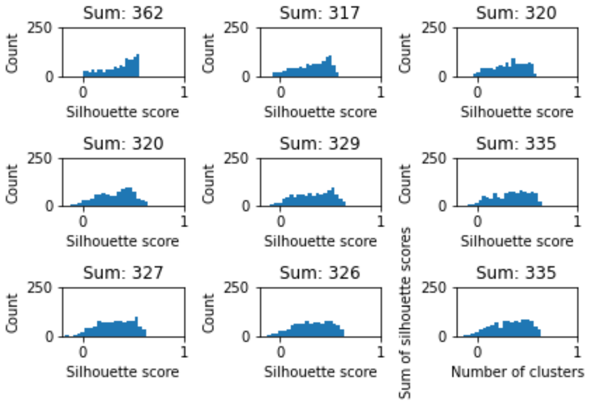

Loadings for PC6
Loading
0.3
0.2
0.1
0.0
−0.1
Question
0
10
20
30
40

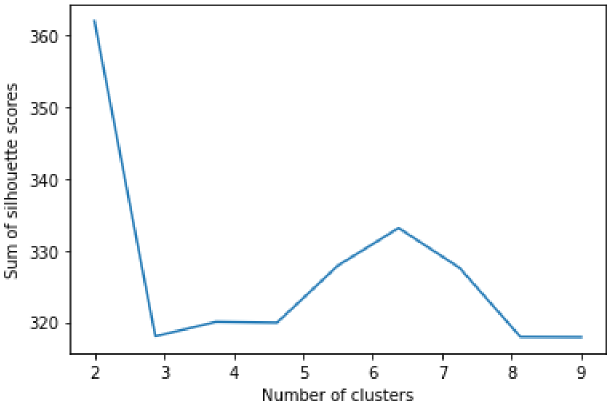


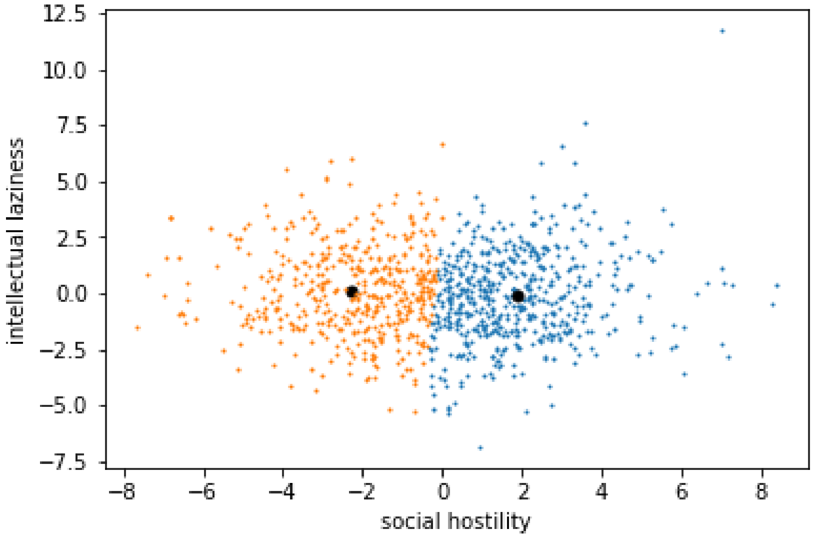


According to the Kaiser rule, I determine that the number of PCs should be kept is 8, and the loadings for each is plotted as above. However, for simpler visualization on a 2-d dimensional space, I only picked the first 2 PCs as coordinates to plot the rotated data. By analyzing the loadings of the first two PCs, since the first PC is strongly and positively associated with questions about “starting quarrel, being rude”, the first PC is named as “social hostility”; since the second PC is linked negatively to question about “curiosity, inventive, and cooperation”, the second PC is named as “intellectual laziness”.

On top of that, I performed K-means clustering. The Silhouette method is used to find the optimal number of k from 2 to 10. The peak of the sum of silhouette scores is at k=2.

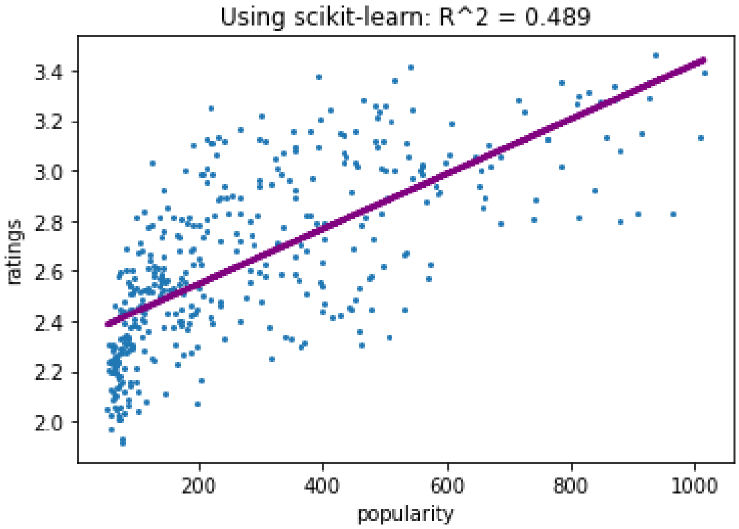


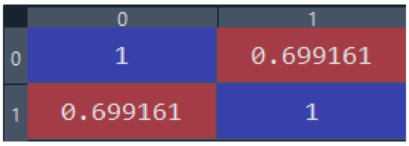




According to the K-mean clustering of the data on the rotated coordinate, we can visually see that the participants are placed into 2 clusters that has an approximate split in the middle. There's a cluster based on low social hostility (social hostility level is negative) and another cluster based on high higher social hostility (social hostility level is positive), while the two clusters display similar, scattered pattern regarding intellectual laziness.

Q3:

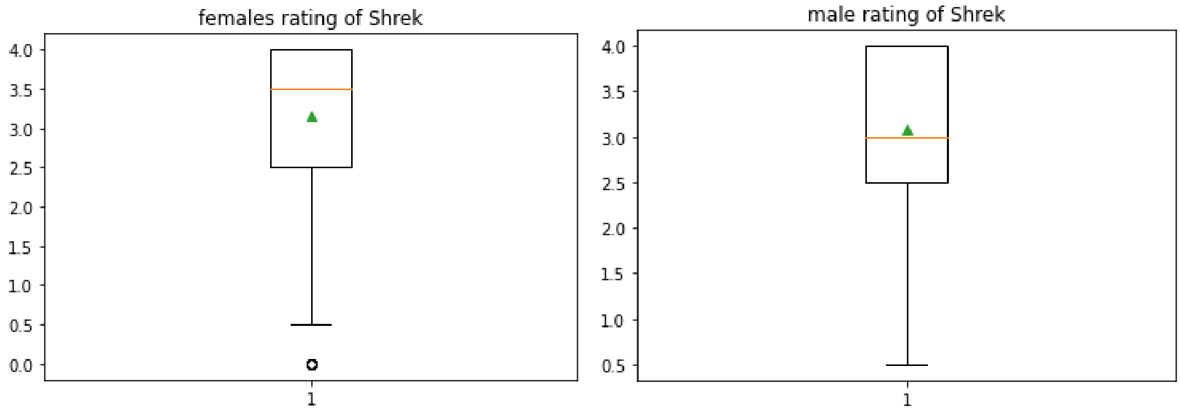


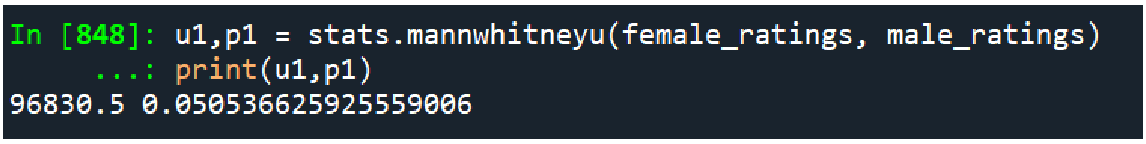


	0	1
0	1	0.699161
1	0.699161	1

We could not conclude that movies that are more popular are necessarily rated higher than movies are less popular from the data. The correlation coefficient between the two variables is closed to 0.70, implying that there is a positive correlation. However, fitting a linear regression line to movie's popularity and ratings result in a model with r-squared = 0.489, which means that only about 48.9% of variance in ratings can be explained by a movie's popularity. There is a considerable amount of scattered data beyond the linear regression line, which indicates that their popularity fails to account for their high ratings, so these are pointing to the groups of movies that have relatively lower popularities and higher ratings. There are also a few data points below the line, indicating that for some movies, their popularity fails to predict their low ratings.

Q4:



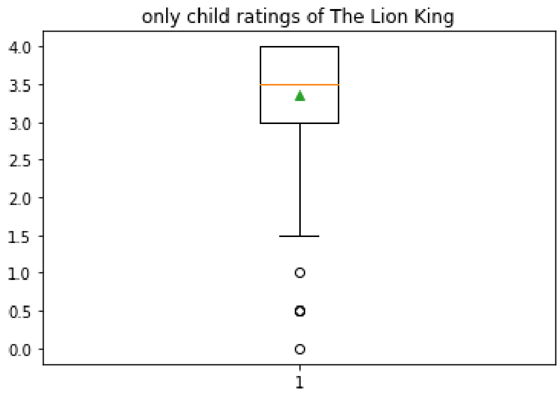


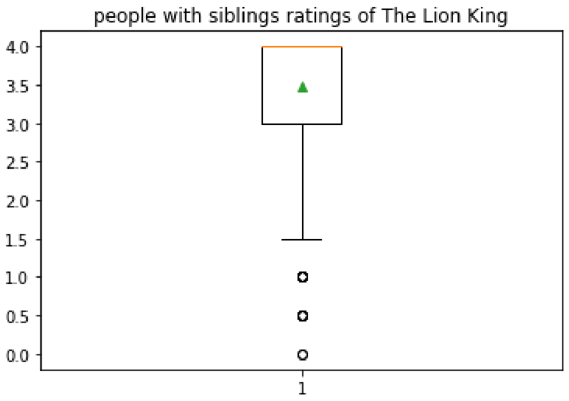
```
In [848]: u1,p1 = stats.mannwhitneyu(female_ratings, male_ratings)
     ...: print(u1,p1)
96830.5 0.050536625925559006
```

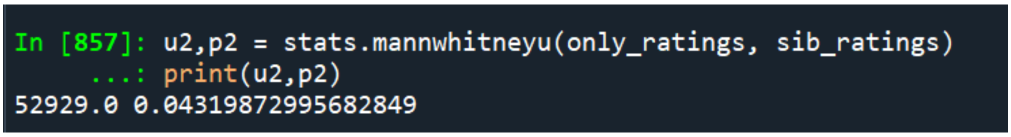
We fail to conclude that the ratings of "Shrek" are gendered. Firstly, the visual exploration of females' and males' rating of "Shrek" allow us to see that the range and the mean of the scores of the two samples seems to be similar, but females' ratings' median is slightly higher than males'. To test whether the two samples are indeed systemically different, a Mann Whitney U-test is performed since we are not sure about equal variances between the score of ratings. The null hypothesis is that there is no difference between males' and females' ratings of Shrek, while the alternative hypothesis is that there is actually difference between males' and females' ratings of "Shrek".

According to the test statistics and the resulting p-value(0.051), I conclude that at we fail to reject the null hypothesis at alpha level = 0.05 since 0.051 > 0.05. The difference between males' and females' ratings of Shrek is not statistically significant, so we don't have evidence to conclude that the ratings of Shrek is gendered.

Q5:



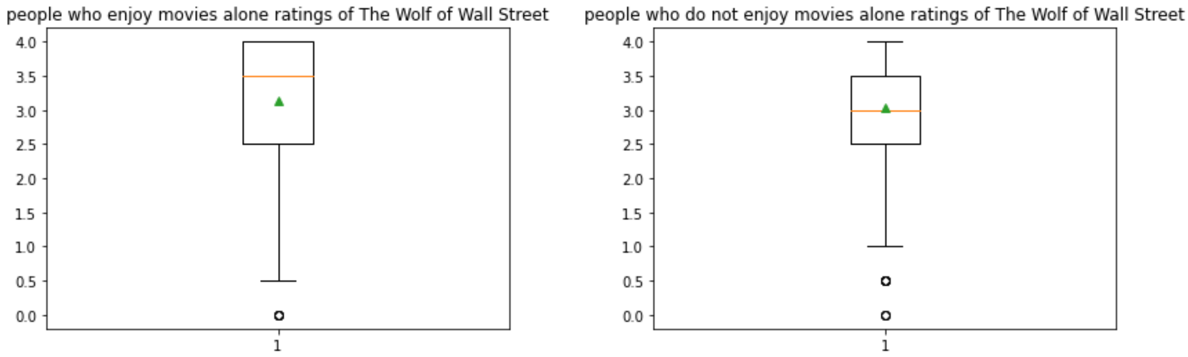


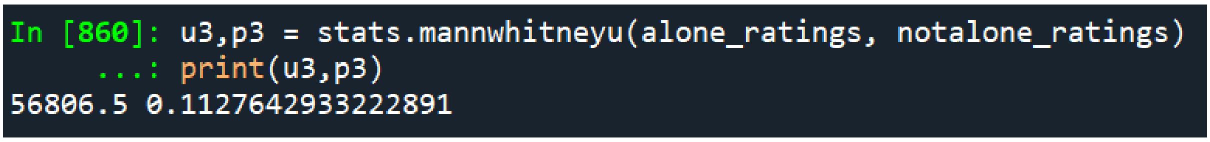


```
In [857]: u2,p2 = stats.mannwhitneyu(only_ratings, sib_ratings)
     ...: print(u2,p2)
52929.0 0.04319872995682849
```

We find statistically significant evidence that people who have siblings and people who do not rate "The Lion King" differently. Firstly, the visual exploration of only childs' and people with siblings' rating of The Lion King exhibits that both groups have similar range of ratings but the average and median ratings of people with siblings is slightly higher. To test whether the two samples are indeed systemically different, a Mann Whitney U-test is performed since we are not sure about equal variances between the score of ratings. The null hypothesis is that there is no difference between people without siblings' and people who have siblings' ratings of "The Lion King", while the alternative hypothesis is that there is actually difference between people without siblings' and people who have siblings' ratings of The Lion King. The resulting p-value(0.043) given by the resulting test statistics provides us evidence to choose to reject the null hypothesis, at alpha level = 0.05, since 0.043 < 0.05. The difference between people without siblings' and people who have siblings' ratings of "The Lion King" is statistically significant, which means that it is unlikely that the two samples are from the same population. Therefore, I made a choice to conclude that the people who have siblings rate "The Lion King" higher than people who don't have siblings.

Q6:



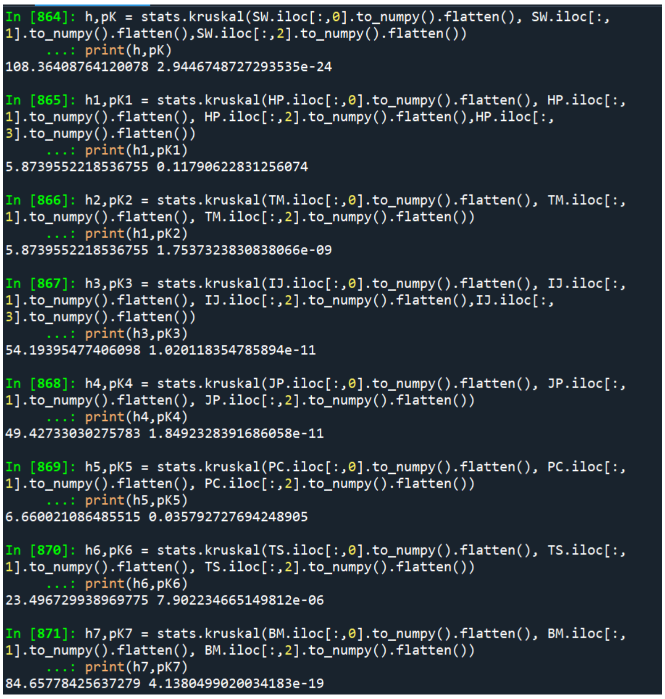


```
In [860]: u3,p3 = stats.mannwhitneyu(alone_ratings, notalone_ratings)
     ...: print(u3,p3)
56806.5 0.1127642933222891
```

We fail to conclude that people who like to watch movies socially enjoy "The Wolf of Wall Street (2013)" more than those who prefer to watch them alone. Initially, plotting the scores from the two samples respectively show us that the median ratings of people who enjoy movies alone is higher than that of people who do not enjoy movies alone. To check for actual difference, the null hypothesis is that there is no difference between the ratings of "The Wolf of Wall Street (2013)" from people who enjoy movies alone and people who don't enjoy movies alone, while the alternative hypothesis is that there is actually difference between the ratings of "The Wolf of Wall Street (2013)" from people who enjoy movies alone and people who don’t enjoy movies alone.

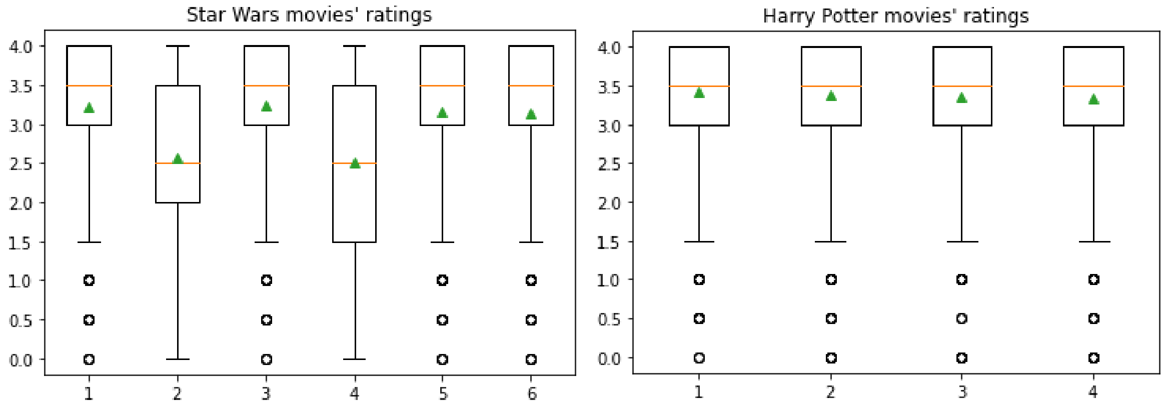
Applying a Mann Whitney U-test (since we are not sure about equal psychological distances between the ratings) amounts to a p-value of 0.112. At alpha = 0.05, since 0.112 > 0.05, we fail to find statistical significance in the difference between the two groups, and we fail to conclude that there is actually difference between the ratings of “The Wolf of Wall Street (2013)” from people who enjoy movies alone and people who don’t enjoy movies alone.

Q7: Since all of the movie series in this question contain three or more movies, and I could not be sure that it is appropriate to reduce movie ratings to mean because of inequal psychological distances, I call forth a nonparametric test, the Kruskal-Wallis test, to determine that if the movies are of inconsistent quality. The null hypothesis is that every movie in the series is of consistent quality, while the alternative hypothesis is that the ratings of each movie in one series is systemically inconsistent.



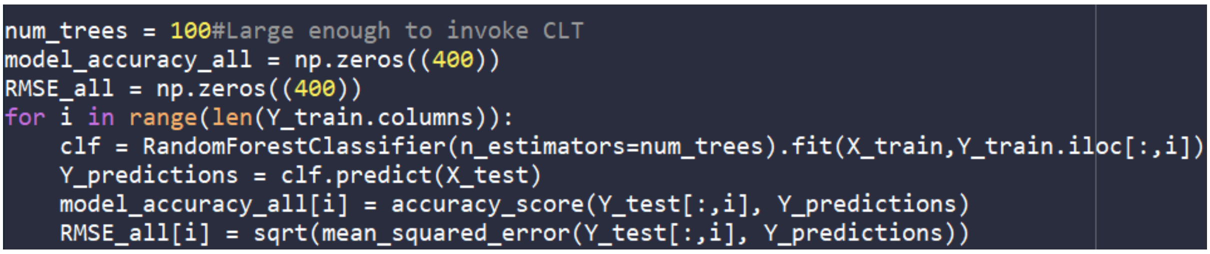
```
In [864]: h,pK = stats.kruskal(SW.iloc[:,0].to_numpy().flatten(), SW.iloc[:,
1].to_numpy().flatten(),SW.iloc[:,2].to_numpy().flatten())
     ...: print(h,pK)
108.36408764120078 2.9446748727293535e-24

In [865]: h1,pK1 = stats.kruskal(HP.iloc[:,0].to_numpy().flatten(), HP.iloc[:,
1].to_numpy().flatten(), HP.iloc[:,2].to_numpy().flatten(),HP.iloc[:,
3].to_numpy().flatten())
     ...: print(h1,pK1)
5.8739552218536755 0.11790622831256074

In [866]: h2,pK2 = stats.kruskal(TM.iloc[:,0].to_numpy().flatten(), TM.iloc[:,
1].to_numpy().flatten(), TM.iloc[:,2].to_numpy().flatten())
     ...: print(h1,pK2)
5.8739552218536755 1.7537323830838066e-09

In [867]: h3,pK3 = stats.kruskal(IJ.iloc[:,0].to_numpy().flatten(), IJ.iloc[:,
1].to_numpy().flatten(), IJ.iloc[:,2].to_numpy().flatten(),IJ.iloc[:,
3].to_numpy().flatten())
     ...: print(h3,pK3)
54.19395477406098 1.020118354785894e-11

In [868]: h4,pK4 = stats.kruskal(JP.iloc[:,0].to_numpy().flatten(), JP.iloc[:,
1].to_numpy().flatten(), JP.iloc[:,2].to_numpy().flatten())
     ...: print(h4,pK4)
49.42733030275783 1.8492328391686058e-11

In [869]: h5,pK5 = stats.kruskal(PC.iloc[:,0].to_numpy().flatten(), PC.iloc[:,
1].to_numpy().flatten(), PC.iloc[:,2].to_numpy().flatten())
     ...: print(h5,pK5)
6.660021086485515 0.035792727694248905

In [870]: h6,pK6 = stats.kruskal(TS.iloc[:,0].to_numpy().flatten(), TS.iloc[:,
1].to_numpy().flatten(), TS.iloc[:,2].to_numpy().flatten())
     ...: print(h6,pK6)
23.496729938969775 7.902234665149812e-06

In [871]: h7,pK7 = stats.kruskal(BM.iloc[:,0].to_numpy().flatten(), BM.iloc[:,
1].to_numpy().flatten(), BM.iloc[:,2].to_numpy().flatten())
     ...: print(h7,pK7)
84.65778425637279 4.1380499020034183e-19
```

According to the results of the 8 Kruskal-Wallis tests performed, the p-value for movies in the series of ‘Star Wars’, ‘The Matrix’, ‘Indiana Jones’, ‘Jurassic Park’, ‘Pirates of the Caribbean’, ‘Toy Story’, and ‘Batman’ all have a smaller value than alpha = 0.05. Meanwhile, ‘Harry Potter’ has a p-value at 0.11 which is larger than alpha = 0.05. Overall, we detect statistically significance for the difference between movies in every movie series other than ‘Harry Potter’ in this question. Therefore, I conclude that ‘Star Wars’, ‘The Matrix’, ‘Indiana Jones’, ‘Jurassic Park’, ‘Pirates of the Caribbean’, ‘Toy Story’, and ‘Batman’ series are all of inconsistent qualities, while ‘Harry Potter’ movies are of a fairly consistent quality.



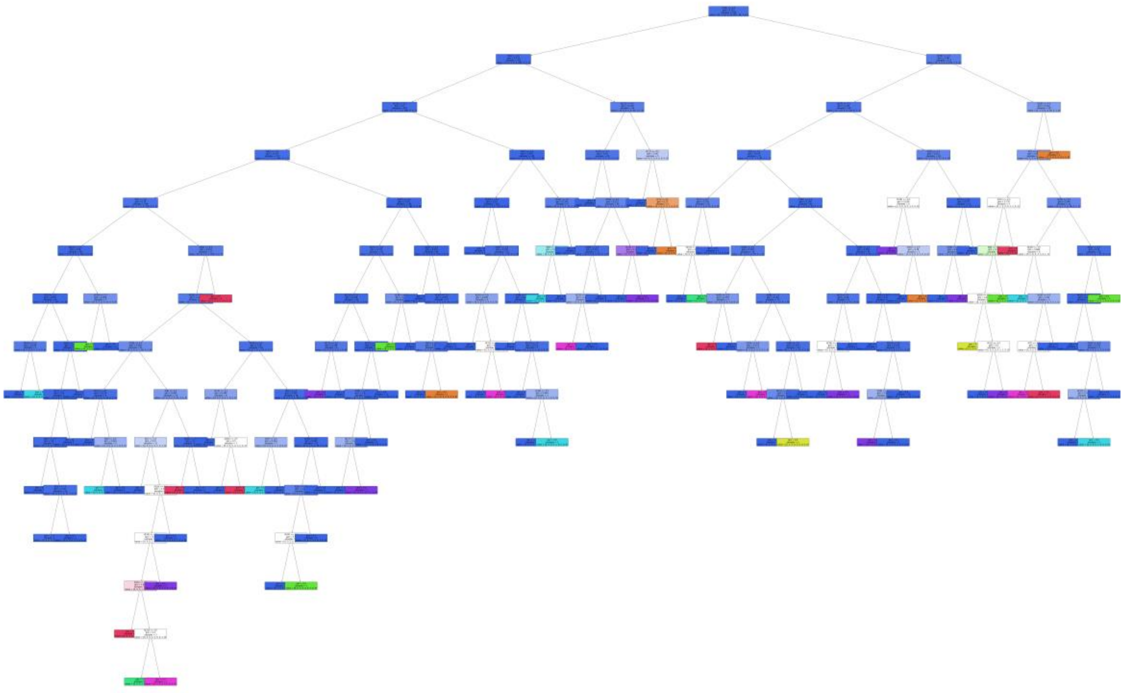
Visualizing the movie ratings of “Star Wars” series and the “Harry Potter” series also lays out a pattern that conforms to the results of the tests. As the overall range and central tendencies of “Star Wars” movies fluctuate clearly between movies, they are basically the same throughout “Harry Potter” movies.

Q8: For this question, I build a random forest to predict movie ratings (for all 400 movies) from personality factors only.

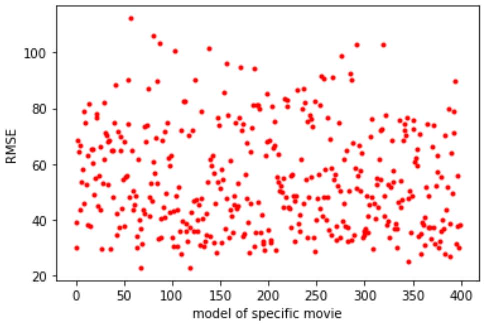


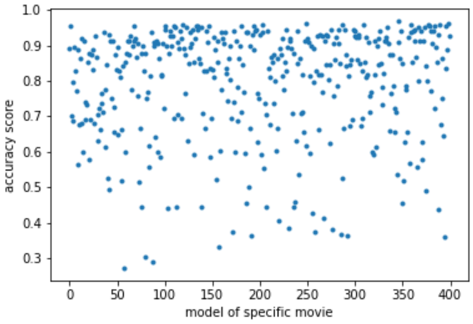
```
num_trees = 100#Large enough to invoke CLT
model_accuracy_all = np.zeros((400))
RMSE_all = np.zeros((400))
for i in range(len(Y_train.columns)):
    clf = RandomForestClassifier(n_estimators=num_trees).fit(X_train,Y_train.iloc[:,i])
    Y_predictions = clf.predict(X_test)
    model_accuracy_all[i] = accuracy_score(Y_test[:,i], Y_predictions)
    RMSE_all[i] = sqrt(mean_squared_error(Y_test[:,i], Y_predictions))
```

All of the questions in the personality categories and the ratings of every movie across all the participants are used to build one random forest for each respective movie. The number of repeat is assigned to be 100 so it is large enough to invoke the Central Limit Theorem. In total, 400 random forests are constructed using the X_train data. Cross validation is achieved by splitting the test and train sets before running the model(I used a 0.8 vs. 0.2 train-test split). After the random forest is fitted using the training sets, I inputted the test set into the model and stored the prediction values. By comparing the predictions and the actual targets of the test set, I was able to compute model accuracy for the random forest for each movie. In every iteration, the model accuracy and RMSE is calculated and stored. As a sample, the first decision tree out of the total one-hundred trees is plotted below:

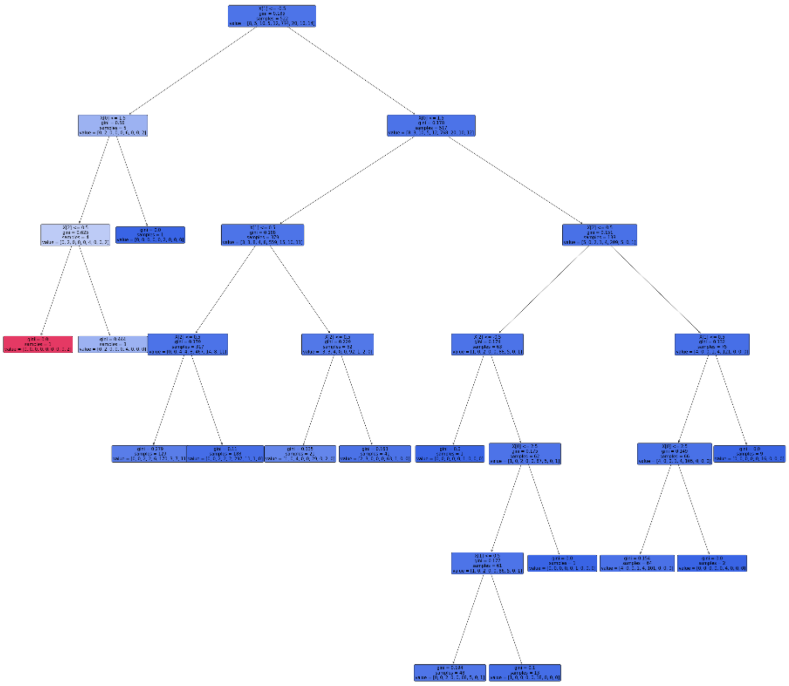


The average accuracy scores across the 400 models for 400 movies is 0.791, implying that when we generalize the model to make predictions with the test sets(X_test), the predictions correctly match the actual test targets(Y_test) are correct at 79.1 percent of the times. The average RMSE is 54.520. Plotting the RMSE and accuracy score gives us an idea that most accuracy scores cluster between 0.9 to 1.0 and spread downwards, while most RMSE clusters between 20 to 40 and spread upwards.

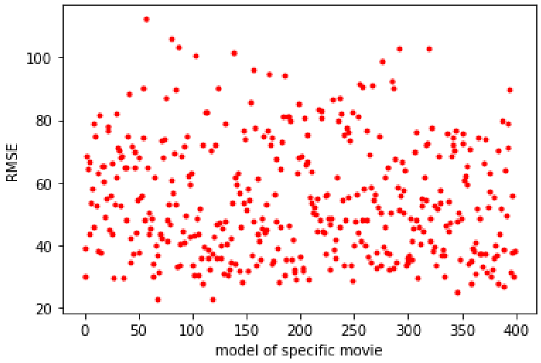


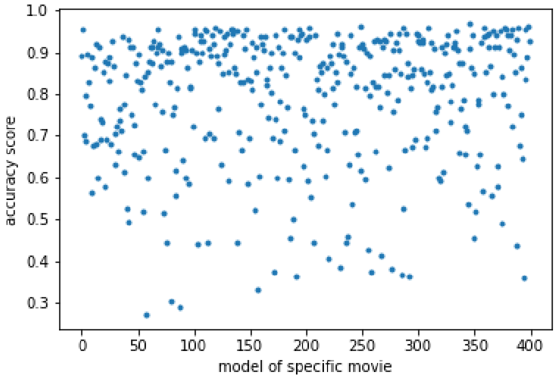


Q9: For this question, I build a random forest to predict movie ratings (for all 400 movies) from gender identity, sibship status and social viewing preferences (columns 475-477) only. In order to cross-validate the data and avoid overfitting, I split the train and test sets using a 0.8 to 0.2 ratio. I fitted 400 random forests, each ran for a specific movie. After the forests are constructed using all of the gender identity, sibship status and social viewing preferences information from the training set, I used the model to predict output of the testing sets and computed the accuracy scores and RMSE for each of the 400 random forests. As a sample, the first decision tree out of the total one-hundred trees is plotted below:



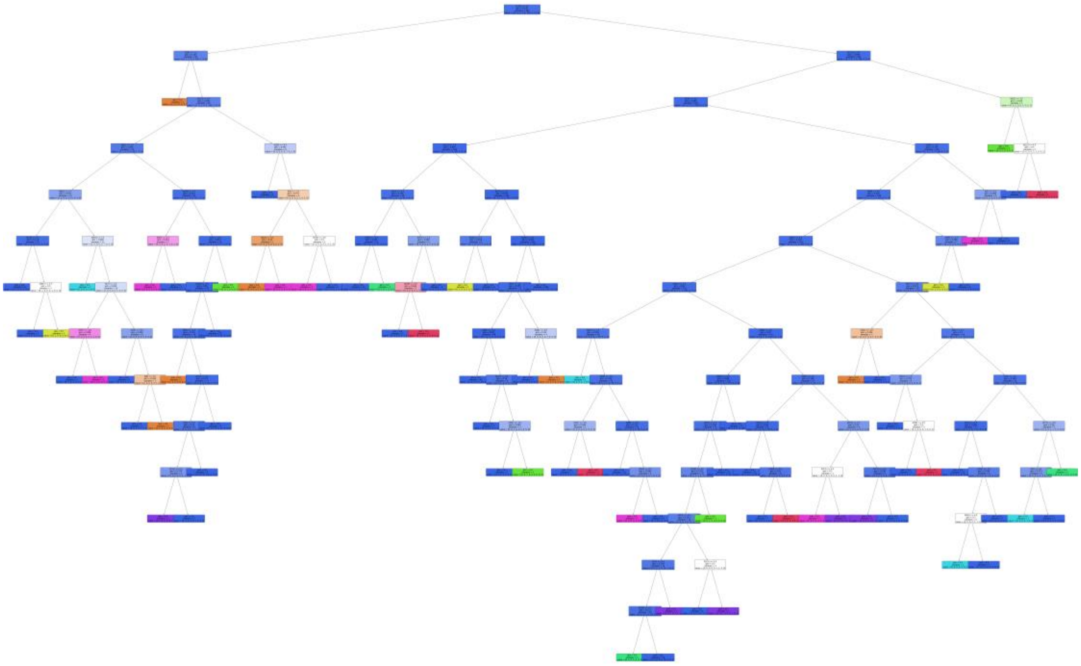
The average accuracy scores across the 400 models for each movies is 0.788, implying that when we generalize the model to make predictions from the test sets(X_test), given the new data of gender identity, sibship status and social viewing preferences, the random forests, in average, correctly predicts 78.8 percent of the actual test targets(Y_test). The average RMSE is 52.034. Plotting the RMSE and accuracy score gives us an idea that most accuracy scores cluster between 0.9 to 1.0 and spread downwards, while most RMSE clusters between 20 to 40 and spread upwards. Plotting the RMSE and accuracy score gives us an idea that many accuracy scores clusters between 0.85 to 0.95 and spread downwards, while most RMSE scatter between 30 to 60.



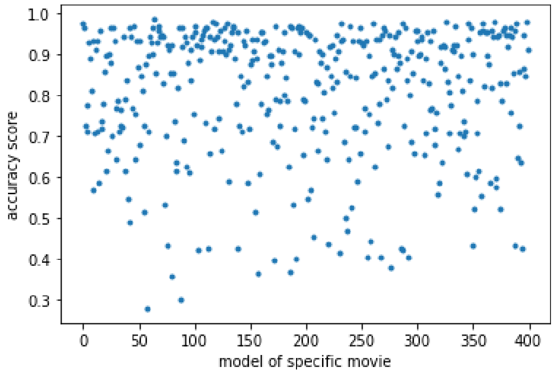


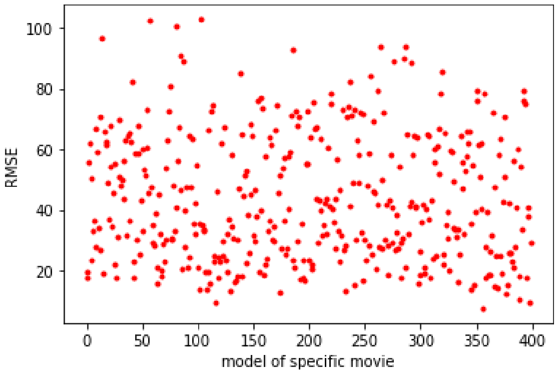
Compared to the random forests using only personality data from the previous question, we can infer that building random forests using personality data generates higher correct predictions if we only consider the accuracy score(0.788 < 0.791). However, the RMSE could lead to different interpretation (52.034 < 52.520).

Q10: For this question, I build a random forest to predict movie ratings (for all 400 movies) from all columns except from the movie ratings. For cross-validation and making sure that there is no overfitting into the test set, I split 80% of the data into the training set and 20% into the test set and fitted 400 random forests using the training set. After the forests are constructed, I used the model to predict outputs with the testing sets and computed the accuracy scores and RMSE for each of the 400 random forests. As a sample, the first decision tree out of the total one-hundred trees is plotted below:



The average accuracy scores across the 400 models for each movies is 0.806, implying that when we generalize the model to make predictions from the test sets(X_test), given the new data of all the information from the columns other than the movie ratings, the random forests in average correctly predicts 80.6 percent of the actual ratings(Y_test). The average RMSE is 52.035. Plotting the RMSE and accuracy score gives us an idea that most accuracy scores clusters between 0.9 to 1.0 and spread downwards, while most RMSE scatters from 20 to 60 and spread upwards.





So far, the prediction accuracy(based on the average accuracy score) of the random forests using all the columns of information besides movie ratings is the best – 80.6% of the results are correctly predicted using the testing set.