A model-based Cluster Analysis approach to anxiety problem behaviors in young and adult outcomes

Hanna Carucci Viterbi, Olimpia Sannucci

February 6, 2022

Contents

Hanna Carucci Viterbi,Olimpia Sannucci
Bonus Project

1 Introduction

This paper reviews the use of machine learning tools to describe the participants included in a study. It identifies and groups similar data points in the larger data set without concern for the specific outcome and discusses the role that a healthy lifestyle plays on anxiety. Data from 53 people has been collected. The sample of the analysis was selected specifically based on age, so a comparison between two generations was carried out (the first group has an average age of 22 years-old while the second one is 55 years old). After having divided the data into different groups and have observed the characteristics of the gotten data set, a cluster analysis has been carried out to stress the dissimilarities between the two groups and K-nearest neighbours algorithm will be applied to find out the anxiety level in each group, after having computed a logistic regression on the complete data set. The aim of the analysis is precisely to observe the answers of the two groups and to analyze the different behaviours. The target variable is Anxiety - since it was detected that ,on average, adults do not suffer from anxiety while young people do, that's why a logistic regression has been implemented with the intention to determine which factors could have an impact on it.

2 Methods

2.1 Data Collection

To collect the necessary information, two questionnaires were created and submitted to a sample of students attending the course of Data Science and Management and a more adult sample, i.e. over 50 years old. The questionnaires were developed using Google Form and multiple-choice and open-ended questions were included. Each of the two samples had to answer the same questions. The answers obtained from the Form were taken to Excel documents and imported into Python enviroment as Data frames. An additional dataset has been created by joining the two obtained. The final dataset is composed of 53 observations and 35 variables (12 numeric and 23 categorical). The questions are:

1. **Gender** - Female or Male
2. **Have you got Insomnia problems?** - Yes or No
3. **How many days you think you can stay without a smartphone?** - Multiple Choice Question
4. **Have you ever looked at your phone to find something but got distracted?** - Yes or No
5. **Book or Movie** - book or movie
6. **How many books do you read in a year** - Multiple Choice Question
7. **What is your Source of Daily News?** - Multiple Choice Question
8. **Text or Call** - Text or Call
9. **Are you Bored?** - Yes or No
10. **Are you at the phone when you are around people?** - Yes or No
11. **How many cigarettes do you smoke a day?** - Multiple Choice Answer
12. **Are you happy?** - Yes or No

13. **Are your parents entrepreneur?** - Yes or No
14. **What is your parents' degree?** - Multiple Choice Question
15. **How many times do you drink carbonated drinks a week?** - Multiple Choice Answer
16. **Do you monitor how many calories do you eat?** - Yes or No
17. **Do your parents smoke?** - Yes or No
18. **Do your friends smoke?** - Yes or No
19. **Are you anxious?** - Yes or No
20. **What is your Age?**
21. **Sport frequency**
22. **How many languages do you Speak?**
23. **Do you have any siblings?**
24. **How much coffee do you drink a day?**
25. **How much do you use your phone in a day?**
26. **What is your sleep average?**
27. **How much time do you dedicate to hobbies in a day?**
28. **How much are you stressed?** - On a scale from 1 to 5
29. **What's your level of weekly Alcohol assumption?** - On a scale from 1 to 5
30. **How much do you care about nutrition?**
31. **How much do you care about diet?** - On a scale from 1 to 5
32. **What Uni do/did you go to?** - Open Question
33. **Which Sport do you practice?** - Open Question
34. **What is your Bachelor Degree?** - Open Question
35. **What is the thing you value the most in life?** - Open Question
36. **How Would You describe yourself in three Words?** - Open Question
37. **What Would you say is your best quality?** - Open Question

The inquiries have been renamed to more concise keywords to facilitate the analysis using the .rename() command.

2.2 Data Cleaning

To prevent omitting all the missing values, Phone usage's, sleep average's and hobbies' time have been substituted with their mean. To make sure the data type of each column is mapped properly, it has been inspected their type manually using data4.info() command. All the columns had the right data type. There are 22 categorical variables in the dataset. The *Uni*, *Sport*, *Valuable Things* and *Bachelor Degree* variables will be removed from cluster analysis as They have a lot of unique values. With data4.describe() have been inspected the 8 numerical variables. The sleep average is approximately 7 hours per night while most of the people dedicate averagely 1.7 hours per day to hobbies. The sport frequency is 2 hours per week and on a scale from 1 to 5 people are averagely stressed 2.

2.3 Data Visualization

A Correlation Matrix has been created to summarize the data. Each cell in the table shows the correlation coefficients between the majority of the variables from the data set. (Appendix 7.4) For the open-ended questions, individuals were asked to describe themselves in three words ('self') and what they considered to be their best quality. Word Cloud, which is a data visualization technique was used for representing text data in which the size of each word indicates its frequency or importance, applied to the variables 'self' and $'bestquality'$ for each group. (Appendix 7.1 and Appendix 7.2)

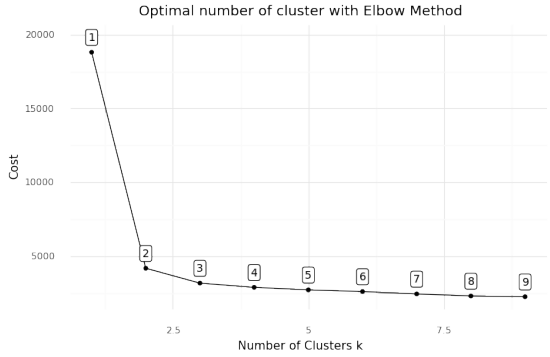
2.4 Features

Data exploration for descriptive analysis is important before doing cluster analysis. It is designed to find an interesting point that can be used to report generation and capture phenomena in the data. The hypothesis is that the anxiety has a strong linear correlation to the age and a healthy lifestyle. To conclude this, descriptive data analysis has been chosen. From the results, it is possible to conclude that Female suffer from anxiety more than men, averagely younger people are more anxious and a healthy lifestyle contributes to the reduction of anxiety.

2.5 Algorithm

2.5.1 Logistic Regression Logistic regression is a powerful supervised Machine Learning algorithm used for binary classification problems (when target is categorical). The dataset has been imported to the Python environment to carry out a logistic regression to find out which independent variables had an impact on the target variable (anxiety). At first the regression could not be computed because of the singularities occurring between several variables, so some of them had to be dropped. The remaining 16 were the most significant and they have been kept for the following analysis. Those variables are: age, sport frequency, phone usage, coffee, alcohol, nutrition care, sleep average, diet, hobbies time, happiness, smoking friends, gender, insomnia, bored and cigarettes.

2.5.2 K-prototypes algorithm K-means algorithm is one of the traditional clustering methods typically used in clustering technology and effectively used for big data. However, its approach is not good and works well for data comprising categorical variables. This problem occurs when calculating the cost function in K-means using Euclidean distance, which is only applicable to numerical data. And K-mode is only suitable for categorical data, not for mixed data types. Faced with these problems, the K-Prototype algorithm was developed to handle clustering algorithms with mixed data types (numeric and categorical variables). K-prototype is a clustering method based on partitioning. Its algorithm is an improvement over the K-means and K-mode clustering algorithms for handling clusters of mixed data types. The Elbow method has been used to select the optimal number of clusters. Instead of computing Sum of Squared Error (WSSE) using Euclidean distance, K-Prototype provides a cost function that is computed combining numerical and categorical variables. Looking at the Elbow, the optimal number of clusters seems is 2.



CC0

The first group counts 27 people, mostly men, and the average age is 22.8 years old. The main differences between the other group have been found in different fields: the members of this group practice on average fewer sports activities but speak more languages, the alcohol consumption is a little bit higher as well as the time dedicated to hobbies. A wanted result is that the time spent on average using the phone is considerably more (5.39 hours for students versus 2.87 hours for adults), they have more friends who smoke and their parents' degree level is Master's Degree; while an unexpected result is that students consume less coffee than the adults and they suffer more from anxiety.

CC1

The second group counts 26 people, mostly female, and the average age is 55.5 years old. Their source of daily news is television (and not Social Networks), their parents' degree level is mainly High School Diploma, they care more about their diet and they read a lot more than students (More than 10 books per year versus minimum 1 and maximum 5).

2.5.3 K-Nearest Neighbours

Finally, the KNN Classification algorithm was implemented to each cluster to identify which most common class among its k nearest neighbors the target variable has been assigned to. Since this algorithm relies on distance for classification and the units come in different scales, normalization will be applied to the data to improve its accuracy.

3 Code Description

The pandas library is imported and the $read_excel()$ command executed to import the files into the Python environment and transform them into Data Frames.The columns have been renamed using the $rename()$ command to more concise keywords to facilitate the analysis. To check if there are missing values in the merged data frame (called data3) the $isna().sum()$ command is used and the latter replaced with their mean.

```
mean_phone = data3["phone_usage"].mean()
data3["phone_usage"] = data3["phone_usage"].replace(np.nan, mean_phone)

mean_sleep = data3["sleep_average"].mean()
data3["sleep_average"] = data3["sleep_average"].replace(np.nan, mean_sleep)

mean_hobbies = data3["hobbies_time"].mean()
data3["hobbies_time"] = data3["hobbies_time"].replace(np.nan, mean_hobbies)
```

The data frame has been copied with the *copy()* command and a new data frame (data4) has been obtained that will be used for this part of the analysis. From the latter, we use the *drop()* function to remove the variables that have a lot of unique values.

```
data4 = data3.copy()
data4.drop('self_description', axis=1, inplace=True)
data4.drop('covid_life', axis=1, inplace=True)
data4.drop('bed_time', axis=1, inplace=True)
data4.drop('hobbies', axis=1, inplace=True)
data4.drop('best_quality', axis=1, inplace=True)
data4.drop('personal_crisis', axis=1, inplace=True)
data4.drop('keyword', axis=1, inplace=True)
data4.drop('evening', axis=1, inplace=True)
data4.drop('bachelor_degree', axis=1, inplace=True)
data4.drop('valuable_things', axis=1, inplace=True)
data4.drop('sport', axis=1, inplace=True)
data4.drop('uni', axis=1, inplace=True)
data4.drop('Do you consider yourself a stressed person? Answer from 1 to 7 (1 minimum level, 7
    maximum level)', axis = 1, inplace = True)
data4.info()
```

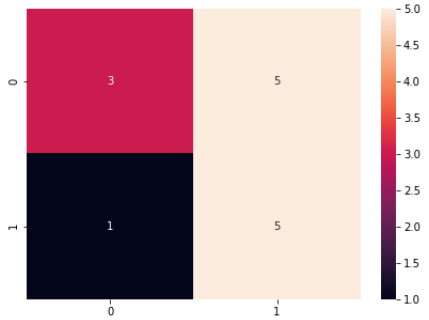
3.1 Logistic Regression

From *sklearn.linearmodel* library *LogisticRegression* and *traintestsplit* were imported. By importing *traintestsplit* command the database is divided randomly (thanks to the *randomstate* command) the database into its first 75% part, for model training, and the remaining 25% for testing. The model is then instantiated using the default parameters and then fit with the data.

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train,y_train)

y_pred=logreg.predict(X_test)
```

A Confusion Matrix, which is a technique for summarizing the performance of a classification algorithm, has been developed to get a better idea about what the classification model is getting right and what types of errors it is making, indeed it has been used to find the accuracy of the model.



Measurement of the accuracy, precision and recall of predictions:

```

Accuracy: 0.5714285714285714
Precision: 0.5
Recall: 0.8333333333333334
```

3.2 Cluster Algorithm

The K-Prototype algorithm is developed to handle clustering algorithms with mixed data types (numeric and categorical variables).

The k-prototypes algorithm is applied and combines the means of the numerical variables and the modes of the categorical ones, building a new hybrid Cluster Center "prototype".

```
data44 = data4.copy()
cost = []
for cluster in range(1, 10):
    try:
        kprototype = KPrototypes(n_jobs = -1, n_clusters = cluster, init = 'Huang', random_state =
    0)
        kprototype.fit_predict(data44, categorical =
    [0,2,4,5,6,8,10,12,14,15,16,17,22,23,24,25,26,28,29,30,31,32,33])
        cost.append(kprototype.cost_)
        print('Cluster initiation: {}'.format(cluster))
    except:
        break
```

To select the optimal number of cluster the Elbow method is used. Plotnine is imported and a ggplot graph is created to have a better visualization of the cost function produced by different values of k. The optimal number of clusters seems to be 2.

```
data44_cost = pd.DataFrame({'Cluster':range(1, 10), 'Cost':cost})
data44_cost.head()

pip install plotnine
from plotnine import *
import plotnine

plotnine.options.figure_size = (8, 4.8)
(
    ggplot(data = data44_cost)+
    geom_line(aes(x = 'Cluster',
                  y = 'Cost'))+
    geom_point(aes(x = 'Cluster',
                   y = 'Cost'))+
    geom_label(aes(x = 'Cluster',
                   y = 'Cost',
                   label = 'Cluster'),
               size = 10,
               nudge_y = 1000) +
    labs(title = 'Optimal number of cluster with Elbow Method')+
    xlab('Number of Clusters k')+
    ylab('Cost')+
    theme_minimal()
)
```

The number of clusters is specified with the command $nclusters = 2$ and $init =$ "$Huang$" for Huang initialization. The function $kprototype.fitpredict()$ is used to fit the centroids and assign points to the closest cluster; the categorical variables have been specified. The cluster centroids are identified by $kprototype.cluster_centroids$; then the iteration and the cost of the clusters created are checked by $kprototype.n_iter$ and $kprototype.cost$. The clusters are then added to the copied dataframe (data44) and placed in order as First and Second. Now it is easier to identify which cluster each observation belongs to.

```
kprototype = KPrototypes(n_jobs = -1, n_clusters = 3, init = 'Huang', random_state = 0)
kprototype.fit_predict(data44, categorical =
    [0,3,4,6,8,10,12,13,14,15,20,21,22,23,24,25,26,27,28,29,30])
# Cluster centorid
kprototype.cluster_centroids_
# Check the iteration of the clusters created
kprototype.n_iter_
# Check the cost of the clusters created
kprototype.cost_
# Add the cluster to the dataframe
data44['Cluster Labels'] = kprototype.labels_
data44['Segment'] = data44['Cluster Labels'].map({0:'First', 1:'Second'})
# Order the cluster
data44['Segment'] = data44['Segment'].astype('category')
data44['Segment'] = data44['Segment'].cat.reorder_categories(['First','Second'])
data44.info()
data44.rename(columns = {'Cluster Labels':'Total'}, inplace = True)
```

With the groupby() function the observations were grouped together according to the cluster they belonged to and for each of them the mean and the mode were calculated and put in a new dataframe (data5). While cc0 and cc1 were created (a dataframe containing a single cluster).

```
data5 = data44.groupby('Segment').agg(
    {
        'Total':'count',
        'gender': lambda x: x.value_counts().index[0],
        'insomnia': lambda x: x.value_counts().index[0],
        'days_without_phone':lambda x: x.value_counts().index[0],
        'phone_distraction':lambda x: x.value_counts().index[0],
        'book_movie':lambda x: x.value_counts().index[0],
        'books_year':lambda x: x.value_counts().index[0],
        'daily_news':lambda x: x.value_counts().index[0],
        'text_call':lambda x: x.value_counts().index[0],
        'bored':lambda x: x.value_counts().index[0],
        'phone_with_people':lambda x: x.value_counts().index[0],
        'cigarettes':lambda x: x.value_counts().index[0],
        'happiness':lambda x: x.value_counts().index[0],
        'parents_ent':lambda x: x.value_counts().index[0],
        'parents_degree':lambda x: x.value_counts().index[0],
        'carbonated_drinks':lambda x: x.value_counts().index[0],
        'calories':lambda x: x.value_counts().index[0],
        'smoking_parents':lambda x: x.value_counts().index[0],
        'smoking_friends':lambda x: x.value_counts().index[0],
        'anxiety':lambda x: x.value_counts().index[0],
        'age': 'mean',
        'sport_frequency': 'mean',
        'languages': 'mean',
        'bro_sis': 'mean',
        'coffee': 'mean',
        'phone_usage':'mean',
        'sleep_average':'mean',
        'hobbies_time':'mean',
        'stress':'mean',
        'alcohol':'mean',
        'nutrition_care':'mean',
        'diet':'mean'

```

```
    }
).reset_index()
data5.info()
cc0 = data44[data44['Segment']== 'First']
cc1 = data44[data44['Segment']== 'Second']
```

3.3 K-Nearest Neighbour

A K-Nearest Neighbours analysis has been carried out for each cluster. Firstly, all the variables that were not useful for the analysis were removed with the *drop()* command.

```
cc0.drop('days_without_phone', axis=1, inplace=True)
cc0.drop('phone_distraction', axis=1, inplace=True)
cc0.drop('languages', axis=1, inplace=True)
cc0.drop('book_movie', axis=1, inplace=True)
cc0.drop('bro_sis', axis=1, inplace=True)
cc0.drop('daily_news', axis=1, inplace=True)
cc0.drop('text_call', axis=1, inplace=True)
cc0.drop('parents_ent', axis=1, inplace=True)
cc0.drop('parents_degree', axis=1, inplace=True)
cc0.drop('carbonated_drinks', axis=1, inplace=True)
cc0.drop('calories', axis=1, inplace=True)
cc0.drop('smoking_parents', axis=1, inplace=True)
cc0.drop('phone_with_people', axis=1, inplace=True)
cc0.drop('books_year', axis=1, inplace=True)
cc0.drop('Total', axis = 1, inplace = True)
cc0.drop('Segment', axis = 1, inplace = True)
```

To apply the algorithm is necessary to dummy code any factor or categorical variables. A new dataframe has been created and for each categorical variable, a category has been dropped: the value left out can be thought of as the reference value and the fit values of the remaining categories represent the change from this reference. ($happiness'$, $'smokingfriends'$, $'gender'$, $'insomnia'$, $'bored'$, $'cigarettes'$), by the $pd.getdummies()$ function. Those dummies were so added to the cc0 data frame with the pd.concat() function.

```
Y_dummy = pd.get_dummies(cc0['anxiety'])
dummy_vars = ['happiness','smoking_friends','gender','insomnia','bored','cigarettes']
other_dummies = pd.get_dummies(cc0[dummy_vars])
dummy_cc0 = pd.concat([Y_dummy,other_dummies], axis = 1)
dummy_cc0.head()

cc0 = pd.concat([cc0,dummy_cc0], axis = 1)
cc0.columns
cc0.drop(dummy_vars,axis=1,inplace=True)
cc0.info()

cc0.info()
cc0 = cc0.drop(['happiness_No','insomnia_No','gender_Female','bored_No','cigarettes_1 - 4','
    cigarettes_5 - 9','smoking_friends_No'],axis = 1)
cc0.drop(['anxiety','Yes'],axis=1,inplace=True)
cc0.info()
cc0.drop(['Total','Segment'],axis = 1, inplace = True)
```

From the sklearn.neighbors library, KNeighborsClassifier was imported to create a KNN model. With the iloc() command the predictors were seleced by their indexes and the target variable Y, which represents the anxiety.

The $traintestsplit$ command was imported from the $sklearn.modelselection$ library and before fitting the model data has been split through a proportion of 75% for training data and the rest for the testing set.

```
cc0.info()
X = cc0.iloc[:, [0,1,2,3,4,5,6,7,8,9,12,14,15,16,17]].values
y = cc0.iloc[:, 10].values
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Finally, the standardScaler command was imported from the sklearn.preprocessing library to standardize features by removing the mean and scaling to unit variance.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

The KNN algorithm assumes that similar things exist in close proximity. The algorithm has been run several times with different values of K, and $nneigbours = 2$ has been chosen because it reduces the number of errors while maintaining the algorithm's ability to make predictions.

```
len(X_train)
len(X_test)
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=2)
knn.fit(X_train, y_train.astype('int'))
KNeighborsClassifier(n_neighbors=2)
knn.score(X_test, y_test)

0.714285714285714
```

the command $knn.score$ indicates the accuracy of the model.

```

from sklearn.metrics import confusion_matrix
y_pred = knn.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
cm
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(7,5))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

We repeat the same for cc1 dataframe, from which results that the accuracy is 0.57142.

```
knn.score = 0.5714285714285714
```

4 Experimental Design

4.1 Evaluation Metrics

For all of the three algorithms implemented in the analysis, the accuracy of each model was evaluated with confusion matrices.

5 Results

5.1 Main Findings

The major conclusions from the study are that there are several variables (phone usage, sleep average, alcohol and coffee assumption, source of daily news, anxiety, stress level, books read in a year,) that highlight the discrepancies between a sample of young people and one of older ones. Phone usage among older people is low and this is substantiated that their main source of daily news is either Television or Newspaper, and this consequence is driven also by the outcome that the average number of books read in a year is more than 10. The parents' degree level of this sample is mainly High School Diploma, and the languages spoken are on average considerably less. These results are highly contrasting for what concerns the younger participants. Finally, regarding the conduction of a "healthy lifestyle", older people drink more coffee than younger, but at the same time, they practice more sport activities and smoke less. What stands out from the analysis is that younger people suffer more from anxiety than adults. After having compared the two clusters, the variables that mainly influence the level of anxiety were found to be: age, sport frequency, phone usage, coffee, alcohol, nutrition care, sleep average, hobbies time, happiness, smoking, insomnia. Indeed, the K-nearest neighbours model confirmed that, despite the similar social context, there is still a generation gap where different habits have a considering impact on the way of living, and, consequently, the level of anxiety.

6 Conclusions

6.1 Considerations

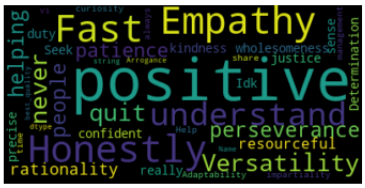
Although youngest generation has many more tools to accomplish general wellness that would simplify their lifestyle compared to previous generations, ironically they found themselves trapped in the comfort that tools like social media, the internet, phone usage, and easier access to university provide causing them a higher level of anxiety.

6.2 Thinking Bigger

Some variables that have an impact on the level of anxiety have been identified, but to make the analysis more accurate, also other variables should have been taken into account (e.g. how COVID-19 impacted social lives). The number of observations was considerably low and did not permit a more complete analysis.

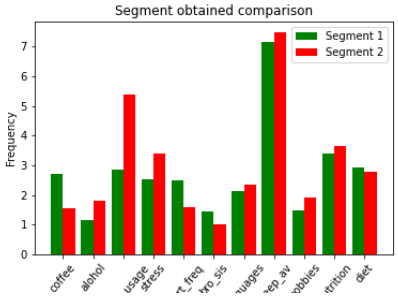
7 Appendix

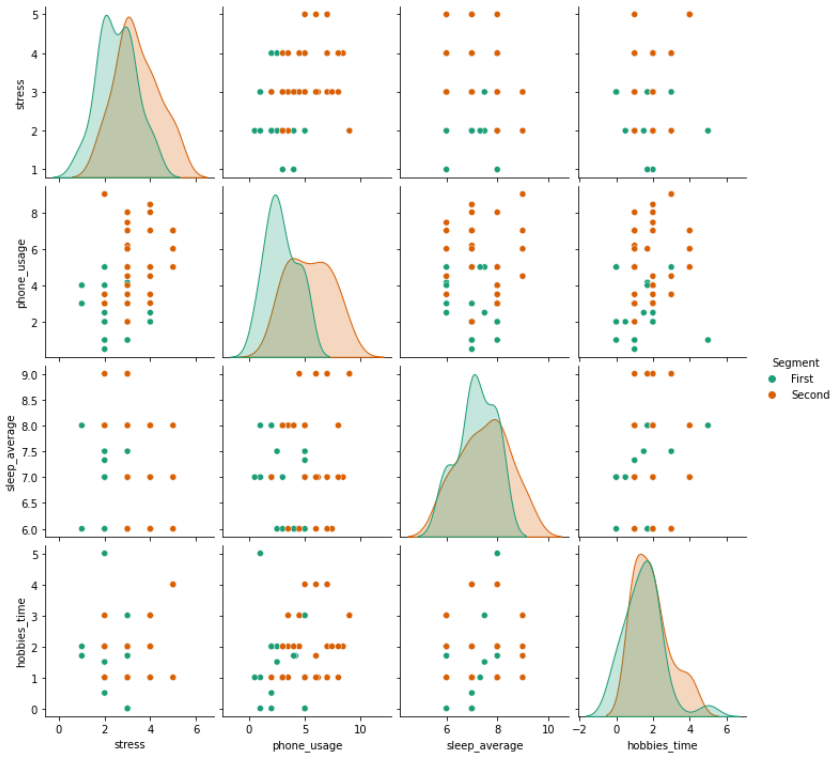
7.1 Best Quality for each Cluster



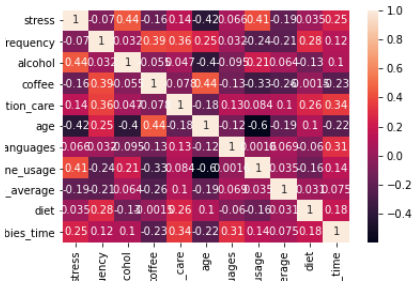
7.2 Self description for each Cluster

7.3 Clusters' comparison visualization for quantitative variables





7.4 Correlation Matrix for quantitative variables



7.5 Confusion Matrix for CC0 and CC1