Personality System using K-Means Clustering Algorithm

A MINI PROJECT REPORT SUBMITTED BY

Disha Shetty
4NM20CS066
VI Semester, B Section

Jayashree
4NM20CS079
VI Semester, B section

UNDER THE GUIDANCE OF

Ms. Vaishali Bangera
Assistant professor GD I
Department of Computer Science and Engineering

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Bachelor of Engineering in Computer Science & Engineering

from

Visvesvaraya Technological University, Belagavi

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum)
AICTE approved, (ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC NITTE -574 110, Udupi District, KARNATAKA.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

May 2023

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

"Personality System using K-Means Clustering Algorithm" is a bonafide work carried out by Disha Shetty (4NM20CS066) and Jayashree (4NM20CS079) in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2022-2023.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide

Signature of HOD

ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to making this project successful.

First and foremost, our sincere thanks to our beloved principal, Dr. Niranjan N. Chiplunkar for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We sincerely thank Dr. K.R. Udaya Kumar Reddy, Head of the Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

We express our deep sense of gratitude and indebtedness to our guide Mrs. Divya Jennifer D'Souza, Assistant Professor GD I, Department of Computer Science and Engineering, for her inspiring guidance, constant encouragement, support, and suggestions for improvement during the course of our project.

We thank all the teaching and non-teaching staff members of the Computer Science and Engineering Department and our parents and friends for their honest opinions and suggestions throughout the course of our project.

Finally, we thank all those who have supported us directly or indirectly throughout the project and made it a grand success.

Disha Shetty
(4NM20CS066)

Jayashree
(4NM20CS079)

ABSTRACT

Predicting personality is crucial for understanding individuals and their unique characteristics. It allows us to uncover the underlying drivers of behaviour and make accurate predictions about preferences, tendencies, and reactions. This knowledge is valuable in psychology and research, where personality prediction helps study the interactions between traits and their contributions to outcomes like mental health, relationships, and job performance. It also has practical applications, such as personalized recommendation systems, where tailored suggestions enhance user experience, and in human resources, where matching candidates to job roles based on personality traits improves job satisfaction and reduces turnover.

Additionally, personality prediction plays a pivotal role in marketing and advertising. By predicting personality and segmenting the target audience accordingly, marketers can develop targeted strategies and personalized campaigns, increasing the effectiveness of their marketing efforts and connecting with the right audience. On an individual level, predicting personality contributes to personal development and well-being. It provides insights into traits, aiding decision-making, self-awareness, interpersonal relationships, and personal growth. Overall, predicting personality improves our understanding of individuals, enables personalized experiences, and informs decision-making processes in various domains, from psychology and research to marketing and personal development.

Table of Contents

Introduction

Personality prediction is a fascinating area of study that aims to understand and categorize individual differences in personality traits. With the advancements in machine learning and data analysis, it has become possible to predict personality using algorithms and techniques that leverage large datasets. One such technique is K-means clustering, an unsupervised machine-learning algorithm that groups similar data points into clusters based on their feature similarities. In this project, we will explore the use of K-means clustering for personality prediction, specifically focusing on the "Big Five" personality traits: extroversion, openness, conscientiousness, agreeableness, and neuroticism. The project involves several key steps. First, we gather a dataset that contains relevant personality-related features or traits. This dataset can be collected through surveys, questionnaires, or other assessment tools designed to measure personality traits. Next, we pre-process the data to ensure its quality and consistency, handling missing values, normalizing data, and removing outliers if necessary.

Once the data is prepared, we select the most informative features for personality prediction. These features should capture the essence of the personality traits and have a significant impact on clustering results. Then, we apply the K-means clustering algorithm, specifying the desired number of clusters (K). The algorithm iteratively assigns data points to different clusters based on their proximity to the cluster centers, minimizing the sum of squared distances. After running the K-means algorithm, we analyse the resulting clusters to understand the different personality profiles or groups present in the data. This analysis may involve examining the centroid characteristics and interpreting the traits associated with each cluster. Furthermore, we can use the trained clustering model to predict the personality traits of new individuals by assigning them to the appropriate cluster based on their feature similarities.

This project has practical applications in various domains. For example, it can be used to personalize recommendation systems, where individuals are provided with tailored suggestions based on their personality traits. It can also assist in human resources by matching candidates to job roles that align with their personality characteristics, leading to better job satisfaction and performance. Additionally, marketers can utilize personality prediction to segment their target audience and develop targeted marketing strategies.

By leveraging K-means clustering for personality prediction, we can gain valuable insights into individual differences in personality traits. The project aims to showcase the power of machine learning algorithms in understanding and categorizing personality profiles, enabling personalized experiences and informed decision-making in fields such as psychology, marketing, and human resources.

Literature Survey

"Predicting personality from social media text" by Golbeck et al. (2011): This study explores the relationship between personality traits and the language used in social media posts. The authors use a machine learning approach to predict the Big Five personality traits (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) from users' Facebook status updates.

"Predicting personality traits from content using IBM Watson" by Dhingra et al. (2017): This study uses IBM Watson's Personality Insights API to predict personality traits from text data. The authors compare the accuracy of the API's predictions to human ratings of personality traits and find that the API performs well.

"Deep learning for predicting personality from text" by Li et al. (2019): This study explores the use of deep learning models to predict personality traits from text data. The authors use a convolutional neural network (CNN) and a long short-term memory (LSTM) network to predict the Big Five personality traits from Twitter data.

"Personality prediction on Twitter using convolutional neural networks and linguistic features" by Gómez-Adorno et al. (2020): This study uses a combination of convolutional neural networks (CNNs) and linguistic features to predict personality traits from Twitter data. The authors compare the performance of their model to other machine learning approaches and find that their model performs well.

"Predicting personality traits from speech using deep learning" by Schuller et al. (2021): This study explores the use of deep learning models to predict personality traits from speech data. The authors use a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to predict the Big Five personality traits from speech recordings.

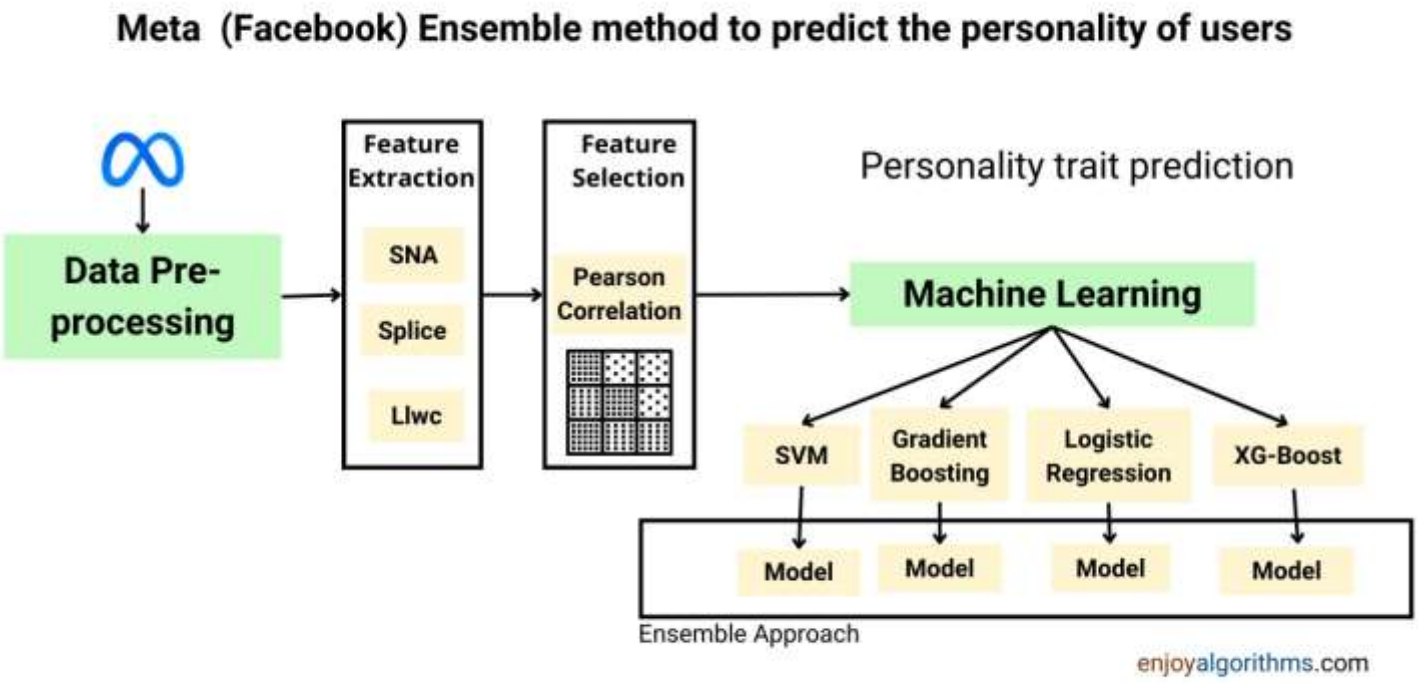
Software requirements

- **Operating System:** Windows
- **Programming Language**: Python.
- **Integrated Development Environment** (IDE): Jupyter Notebook, PyCharm, or Spyder.
- **Libraries:**
  - **NumPy**: For numerical computations and array operations.
  - **Pandas**: For data manipulation and analysis.
  - **Scikit-learn**: For machine learning algorithms, including K-means clustering.
  - **Matplotlib** and Seaborn: For data visualization and plotting.

Hardware requirements

- **Processor**: A multi-core processor (e.g., Intel Core i3 or higher)
- **Memory** (RAM): At least 8 GB of RAM is recommended
- **Storage**: Sufficient storage space to store the dataset, libraries, and project files.
- **GPU** (optional)

Design and analysis



Design:

Data Collection: Gather a dataset containing personality-related features through surveys or questionnaires. Ensure the dataset includes a sufficient number of samples to provide meaningful insights.

Data Pre-processing: Clean the dataset by handling missing values and removing outliers if necessary. Normalize the data to bring all features to a common scale, ensuring fair representation of different features during clustering.

Feature Selection: Identify the most relevant features that capture the essence of the personality traits. This can be done through domain expertise, statistical analysis, or feature importance techniques.

K-means Clustering: Apply the K-means clustering algorithm to the preprocessed dataset. Specify the desired number of clusters (K) based on the nature of the dataset and prior knowledge.

Conduct multiple iterations of the algorithm to mitigate initialization biases and improve the stability of the clustering.

Cluster Analysis: Analyze the resulting clusters to understand the distinct personality profiles. Examine the centroid characteristics of each cluster and interpret the personality traits associated with them. Visualize the clusters and their characteristics using plots or charts to gain further insights.

Personality Prediction: Utilize the trained clustering model to predict the personality traits of new individuals. Assign new data points to the closest cluster based on their feature similarities, providing an estimate of their personality profile.

Analysis:

Cluster Evaluation: Assess the quality of the clustering results using appropriate evaluation metrics such as silhouette score or within-cluster sum of squares. These metrics measure the compactness and separation of the clusters, providing insights into the effectiveness of the clustering algorithm.

Interpretation of Clusters: Interpret the personality traits associated with each cluster by analysing the centroid characteristics. Understand the unique characteristics and tendencies of each personality profile identified by the clustering algorithm.

Validation and Generalization: Validate the performance of the personality prediction model by comparing the predicted personality traits with ground truth labels if available. Assess the generalizability of the model by testing it on new and unseen data to ensure its effectiveness beyond the training dataset.

Fine-tuning and Optimization: Explore techniques to improve the clustering results, such as adjusting the number of clusters (K) or trying alternative clustering algorithms. Fine-tune the model parameters, feature selection, or pre-processing techniques to enhance the accuracy and stability of the personality prediction.

Implementation

In this section, we have command and queries used in the project implementation. Initial commands are for setting up big data tools and later we have queries of creation of schema, table, loading data and some operation on data.

- Open Jupyter Notebook.
- Create a new notebook or open an existing one.
- Import the required libraries:

```
import pandas as pd

import numpy as np
from sklearn.cluster import MiniBatchKMeans
```

- Load the dataset using the read_csv() function from Pandas library:

```
df = pd.read_csv('data-final.csv', delimiter="\t")
```

- Perform visualization of the dataset using the print() function, get the columns of the dataset using the columns attribute, and print them using a for loop, Get the feature matrix X by selecting the first 50 columns of the dataset using the iloc function and fill the NaN values with 0 using the fillna() function:
- Train the MiniBatchKMeans clustering model with 10 clusters and other hyperparameters using the **MiniBatchKMeans()** function from the **sklearn.cluster** module

```
kmeans = MiniBatchKMeans(n_clusters=10, random_state=0, batch_size=100, max_iter=100).fit(X)
```

- Get the cluster centers of the trained model and store them in variables one, two, three, four, five, six, seven, eight, nine, and ten:

```
one = kmeans.cluster_centers_[0]
two = kmeans.cluster_centers_[1]
three = kmeans.cluster_centers_[2]
```

```
four = kmeans.cluster_centers_[3]
        five = kmeans.cluster_centers_[4]
        six = kmeans.cluster_centers_[5]
        seven = kmeans.cluster_centers_[6]
        eight = kmeans.cluster_centers_[7]
        nine = kmeans.cluster_centers_[8]
        ten = kmeans.cluster_centers_[9]
```

- Calculate the scores of the Big Five personality traits for each cluster and store them in dictionaries one_scores, two_scores, three_scores, four_scores, five_scores, six_scores, seven_scores, eight_scores, nine_scores, and ten_scores:

```
one_scores = {}
one_scores['extroversion_score'] =  one[0] - one[1] + one[2] - one[3] + one[4] - one[5] + one[6] - one[7] + one[8] - one[9]
one_scores['neuroticism_score'] =  one[0] - one[1] + one[2] - one[3] + one[4] + one[5] + one[6] + one[7] + one[8] + one[9]
one_scores['agreeableness_score'] =  -one[0] + one[1] - one[2] + one[3] - one[4] - one[5] + one[6] - one[7] + one[8] + one[9]
one_scores['conscientiousness_score'] =  one[0] - one[1] + one[2] - one[3] + one[4] - one[5] + one[6] - one[7] + one[8] + one[
```

Result

- Obtain the uploaded dataset using the print() function

```
In [83]: print(df)

1015337  2018-11-08 12:07:18   1920.0   1080.0          3.0       122.0
1015338  2018-11-08 12:07:49   1920.0   1080.0          2.0       135.0
1015339  2018-11-08 12:08:34   1920.0   1080.0          6.0       212.0
1015340  2018-11-08 12:08:45   1920.0   1080.0          3.0       176.0

         endelapse  IPC country lat_appx_lots_of_err long_appx_lots_of_err
0                6    1      GB               51.5448                0.1991
1               11    1      MY                3.1698              101.706
2                7    1      GB               54.9119               -1.3833
3                7    1      GB                 51.75                 -1.25
4               17    2      KE                   1.0                  38.0
...            ...  ...     ...                   ...                   ...
1015336         10    2      US               39.9883              -75.2208
1015337          7    1      US                  38.0                 -97.0
1015338         12    6      US               36.1473                -86.777
1015339          8    1      US               34.1067              -117.8067
1015340          9    1      US                  38.0                 -97.0

[1015341 rows x 110 columns]

In [85]: columns = df.columns

In [86]: for column in columns:
             print(column)

EXT1
```

- Get the columns of the dataset using the columns attribute, and print them using a for loop, Get the feature matrix X by selecting the first 50 columns of the dataset using the iloc function and fill the NaN values with 0 using the fillna() function:

```
In [88]: X = df[df.columns[0:50]]
         pd.set_option('display.max_columns', None)
         X
```

Out[88]:

	EXT1	EXT2	EXT3	EXT4	EXT5	EXT6	EXT7	EXT8	EXT9	EXT10	EST1	EST2	EST3	EST4	EST5	EST6	EST7	EST8	EST9	EST10	AGR1	AGR
0	4.0	1.0	5.0	2.0	5.0	1.0	5.0	2.0	4.0	1.0	1.0	4.0	4.0	2.0	2.0	2.0	2.0	2.0	3.0	2.0	2.0	5.
1	3.0	5.0	3.0	4.0	3.0	3.0	2.0	5.0	1.0	5.0	2.0	3.0	4.0	1.0	3.0	1.0	2.0	1.0	3.0	1.0	1.0	4.
2	2.0	3.0	4.0	4.0	3.0	2.0	1.0	3.0	2.0	5.0	4.0	4.0	4.0	2.0	2.0	2.0	2.0	2.0	1.0	3.0	1.0	4.
3	2.0	2.0	2.0	3.0	4.0	2.0	2.0	4.0	1.0	4.0	3.0	3.0	3.0	2.0	3.0	2.0	2.0	2.0	4.0	3.0	2.0	4.
4	3.0	3.0	3.0	3.0	5.0	3.0	3.0	5.0	3.0	4.0	1.0	5.0	5.0	3.0	1.0	1.0	1.0	1.0	3.0	2.0	1.0	5.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1015336	4.0	2.0	4.0	3.0	4.0	3.0	3.0	3.0	3.0	3.0	4.0	3.0	3.0	3.0	4.0	3.0	4.0	3.0	3.0	3.0	5.0	4.
1015337	4.0	3.0	4.0	3.0	3.0	3.0	4.0	4.0	3.0	3.0	4.0	3.0	5.0	1.0	5.0	5.0	4.0	4.0	4.0	5.0	2.0	4.
1015338	4.0	2.0	4.0	3.0	5.0	1.0	4.0	2.0	4.0	4.0	3.0	2.0	4.0	3.0	2.0	2.0	4.0	2.0	4.0	1.0	3.0	5.
1015339	2.0	4.0	3.0	4.0	2.0	2.0	1.0	4.0	2.0	4.0	4.0	3.0	4.0	2.0	4.0	4.0	2.0	2.0	4.0	4.0	2.0	3.
1015340	4.0	2.0	4.0	2.0	4.0	1.0	4.0	2.0	4.0	4.0	4.0	3.0	4.0	3.0	2.0	3.0	3.0	1.0	4.0	2.0	1.0	5.

1015341 rows × 50 columns

```
In [89]: X = X.fillna(0)
         from sklearn.cluster import MiniBatchKMeans
```

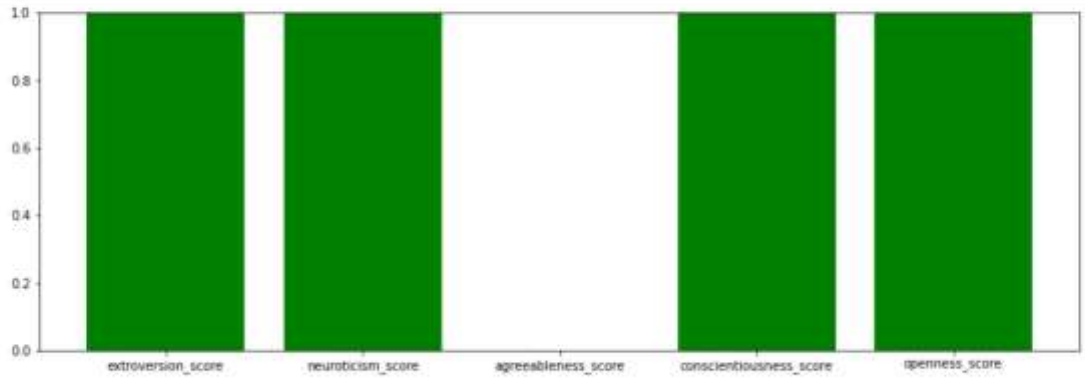
- Get the cluster centres 7of the trained model and store them in variables one, two, three, four, five, six, seven, eight, nine, and ten, Calculate the scores of the Big Five personality traits for each cluster and store them in dictionaries

```
In [90]: len(kmeans.cluster_centers_)

Out[90]: 10

In [91]: one = kmeans.cluster_centers_[0]
         two = kmeans.cluster_centers_[1]
         three =kmeans.cluster_centers_[2]
         four = kmeans.cluster_centers_[3]
         five =kmeans.cluster_centers_[4]
         six = kmeans.cluster_centers_[5]
         seven = kmeans.cluster_centers_[6]
         eight = kmeans.cluster_centers_[7]
         nine= kmeans.cluster_centers_[8]
         ten = kmeans.cluster_centers_[9]
         one

Out[91]: array([3.51875306, 1.8936703 , 4.45234737, 2.11922512, 4.35885553,
                1.56325379, 3.98647132, 2.76422671, 3.81405256, 2.3716646 ,
                2.05604329, 3.97875709, 2.92127457, 3.4824271 , 1.8897844 ,
                1.67228923, 1.87329092, 1.47554762, 1.86143174, 1.5434501 ,
                1.79666676, 4.49912207, 1.70769408, 4.29999136, 1.7910538 ,
                3.85187531, 1.50729685, 4.16565441, 4.14265565, 4.31282922,
                3.87432716, 2.42716099, 4.31288679, 1.68691172, 3.30911603,
                2.07210501, 3.97875709, 1.76252842, 3.69278374, 4.00077718,
                4.0497395 , 1.58055324, 4.18154342, 1.64966466, 4.2821738 ,
                1.53360583, 4.44892202, 3.33657638, 4.18433551, 4.42462796])

In [92]: one_scores = {}
```

```
In [92]: one_scores = {}

         one_scores['extroversion_score'] =  one[0] - one[1] + one[2] - one[3] + one[4] - one[5] + one[6] - one[7] + one[8] - one[9]
         one_scores['neuroticism_score'] =  one[0] - one[1] + one[2] - one[3] + one[4] + one[5] + one[6] + one[7] + one[8] + one[9]
         one_scores['agreeableness_score'] =  -one[0] + one[1] - one[2] + one[3] - one[4] - one[5] + one[6] - one[7] + one[8] + one[9]
         one_scores['conscientiousness_score'] =  one[0] - one[1] + one[2] - one[3] + one[4] - one[5] + one[6] - one[7] + one[8] + one[9]
         one_scores['openness_score'] =  one[0] - one[1] + one[2] - one[3] + one[4] - one[5] + one[6] + one[7] + one[8] + one[9]
         one_scores

Out[92]: {'extroversion_score': 9.418439308022236,
          'neuroticism_score': 22.81672951267956,
          'agreeableness_score': -2.472352551740031,
          'conscientiousness_score': 14.16176851558679,
          'openness_score': 19.690221927981366}

In [93]: all_types = {'one':one, 'two': two, 'three' :three, 'four':four, 'five':five, 'six': six, 'seven': seven, 'eight': eight,
                      'nine': nine, 'ten': ten}

         all_types_scores ={}

         for name, personality_type in all_types.items():
             personality_trait = {}

             personality_trait['extroversion_score'] =  personality_type[0] - personality_type[1] +personality_type[2] - personality_type[
             personality_trait['neuroticism_score'] =  personality_type[0] - personality_type[1] + personality_type[2] -personality_type[3
             personality_trait['agreeableness_score'] =  -personality_type[0] +personality_type[1] - personality_type[2] + personality_typ
             personality_trait['conscientiousness_score'] = personality_type[0] - personality_type[1] + personality_type[2] -personality_t
             personality_trait['openness_score'] =  personality_type[0] -personality_type[1] + personality_type[2] - personality_type[3] +
```

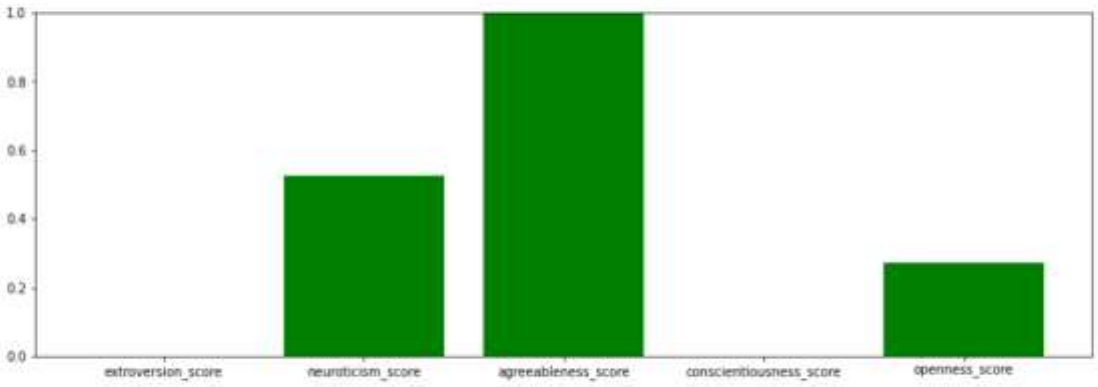
- Resultant personality graphs obtained

```
In [97]: import numpy as np
         import matplotlib.pyplot as plt

         plt.figure(figsize=(15,5))
         plt.ylim(0, 1)
         plt.bar(list(normalized_all_types_scores['one'].keys()), normalized_all_types_scores['one'].values(), color='g')
         plt.show()
```



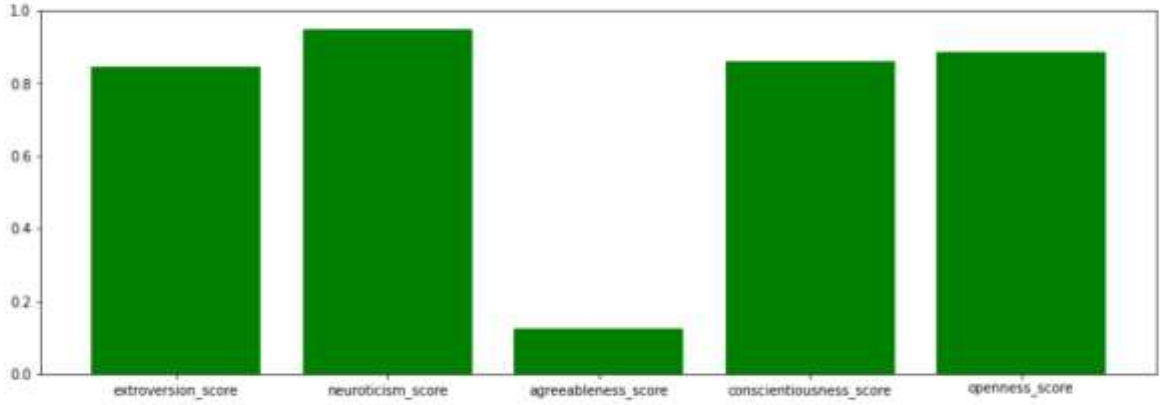

```
In [98]: plt.figure(figsize=(15,5))
         plt.ylim(0, 1)
```

```
In [98]: plt.figure(figsize=(15,5))
         plt.ylim(0, 1)
         plt.bar(list(normalized_all_types_scores['two'].keys()), normalized_all_types_scores['two'].values(), color='g')
         plt.show()
```



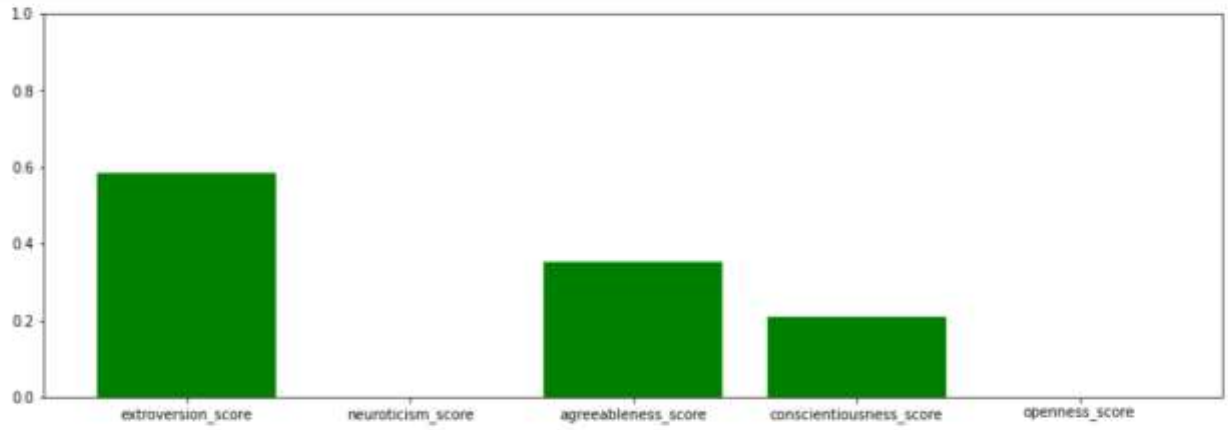

```
In [99]: plt.figure(figsize=(15,5))
         plt.ylim(0, 1)
         plt.bar(list(normalized_all_types_scores['three'].keys()), normalized_all_types_scores['three'].values(), color='g')
         plt.show()
```

In [100]:
```
plt.figure(figsize=(15,5))
plt.ylim(0, 1)
plt.bar(list(normalized_all_types_scores['four'].keys()), normalized_all_types_scores['four'].values(), color='g')
plt.show()
```



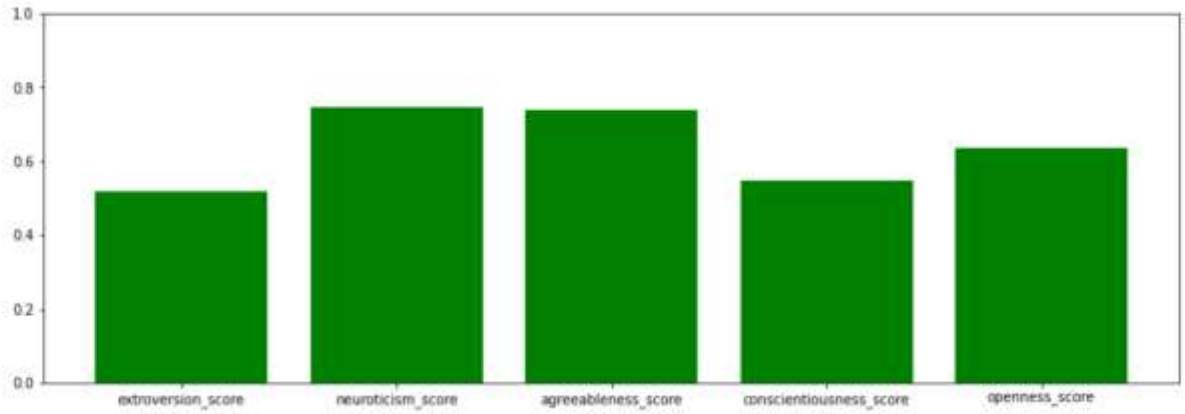
In [101]:
```
plt.figure(figsize=(15,5))
plt.ylim(0, 1)
plt.bar(list(normalized_all_types_scores['five'].keys()), normalized_all_types_scores['five'].values(), color='g')
```

In [99]:
```
plt.figure(figsize=(15,5))
plt.ylim(0, 1)
plt.bar(list(normalized_all_types_scores['three'].keys()), normalized_all_types_scores['three'].values(), color='g')
plt.show()
```



In [100]:
```
plt.figure(figsize=(15,5))
plt.ylim(0, 1)
plt.bar(list(normalized_all_types_scores['four'].keys()), normalized_all_types_scores['four'].values(), color='g')
plt.show()
```

In [106]:
```
plt.figure(figsize=(15,5))
plt.ylim(0, 1)
plt.bar(list(normalized_all_types_scores['ten'].keys()), normalized_all_types_scores['ten'].values(), color='g')
plt.show()
```



Conclusion

In conclusion, this project explored the use of K-means clustering for personality prediction and showcased the power of machine learning algorithms in understanding and categorizing individual differences in personality traits. By gathering a dataset containing relevant personality-related features, pre-processing the data, selecting informative features, and applying the K-means clustering algorithm, we were able to identify distinct personality profiles and predict the personality traits of new individuals.

The project demonstrated the practical applications of personality prediction in various domains, such as personal development, psychology, marketing, and human resources. It also highlighted the importance of data quality and feature selection in achieving accurate and meaningful clustering results.

Moving forward, the project can be extended by exploring alternative clustering algorithms or combining multiple algorithms to improve the clustering accuracy. Additionally, further research can be conducted to assess the generalizability of the model and its performance on different populations or cultural contexts.

Overall, the project contributes to the growing body of research on personality prediction and provides insights into the underlying drivers of behaviour, enabling personalized experiences, and informed decision-making in various fields.

References

- k-means clustering :https://en.wikipedia.org/wiki/K-means_clustering

- Jupyter Notebook http://jupyter.org/

- Kaggle https://www.kaggle.com/