Machine Learning Powered Compatible People Proposer Based On Personality Traits

Gaurav Goswami, Divyanshu Gaur, Eshani Agarwal, Enosh Kumar, Dr. Mukesh Rawat

Abstract: Today almost every task is carried out in groups/teams and the accuracy of the same depends on the compatibility among members. And throughput is the prime concern of every group task. So in order to get maximum throughput in team/group tasks, we need to group people with similar thinking, likes, dislikes, choices, behavior etc. To distinguish between people with similar and different personality traits, we're analyzing speeches, views, social updates (in textual form) using some Machine Learning models and getting 27 distinct personality traits scores (in numeric form). And since working on 27-dimensional data is very haptic, we reduced the data in 3-dimensions using PCA (Machine Learning) and then a 3D graph is plotted in which each person is depicted using a point and clusters are made of points having less than or equal to 0.7 Euclidean distance. These clusters are nothing but compatible groups.

Index Terms: NLU: Natural Language Understanding, PCA: Principal Component Analysis, API: Application Program Interface, 3D: 3-Dimensional, DB: Database

1 INTRODUCTION

It is the age of social media liberalism; everyone is free to express what he/she feels; they can say whatever they want to. But the question arises now is: What to do about people working together for same task having different thinking? It leads to poor communication, different working behavior, and different views, choices, likes and dislikes among team members and as a result, team heads to wrong direction. So, in order to avoid all this, we need to make teams (or groups) of people having similar thinking, views and behaviors, choices, likes and dislikes. However, it is not possible to find people with identical thinking, but we could go for similar thinking. It is obvious and easily understood that people with common thinking and personality traits tend to work efficiently with each other, which could reduce work fatigue and pressure they're carrying in current work culture. 

2 APPLICATION STRUCTURE

2.1 Overall Architecture

The application is divided into 4 tiers, frontend, backend, database and API tiers. Frontend just comprises of user interfaces for various operations, whereas backend comprises of algorithms of various operations, such as

- Gaurav Goswami is currently pursuing bachelors degree program in computer science engineering in Dr. APJ Abdul Kalam Technical University, Lucknow, India, PH: +917017925280. E-mail: gouravgoswami48@gmail.com
- Divyanshu Gaur is currently pursuing bachelors degree program in computer science engineering in Dr. APJ Abdul Kalam Technical University, Lucknow, India, PH: +918267883385. E-mail divyanshugaur14@gmail.com
- Eshani Agarwal is currently pursuing bachelors degree program in computer science engineering in Dr. APJ Abdul Kalam Technical University, Lucknow, India, PH:+917351555518. E-mail: eshaniaagarwal1923@gmail.com
- Enosh Kumar is currently pursuing bachelors degree program in computer science engineering in Dr. APJ Abdul Kalam Technical University, Lucknow, India, PH:+919917415599. E-mail: enosh.kumar.cs2016@miet.ac.in
- Dr. Mukesh Rawat received Ph.D (Computer Engg.) from Shobhit University, Meerut, Uttar Pradesh in 2015. Currently he is working as Professor in the Department of Computer Science and Engg., MIET, Meerut. His research interest includes Documents clustering & summarization and Hidden Web Exposure. PH: +918218938978, E-mail: mukesh.rawat@miet.ac.in

The application runs as follows: User logs in using his/her Twitter credentials and then user may enter any raw text manually or select any friend/follower or own self and backend will fetch all the recent updates to the selected person[1] and then clean the text data and forward it to IBM Watson NLU and Personality Insights services, and after getting response, backend filter out specific 27 personality traits with their corresponding scores. The backend generates cards for each selected person with those personality traits and then the dataset is reduced to 3-
dimensions using PCA algorithm and an overall 3D graph is generated which depicts clusters of compatible people (having <= 0.7 units of Euclidean distance). The backend finally generates a report showing what all groups or clusters can be made (if any).

3 MACHINE LEARNING MODELS

We've used two machine learning models which can analyze textual data and to predict how much they fall in defined categories. These models have variable accuracy, which is dependent on words count of textual data fed to it. Textual data is cleaned first before sending to NLU Model and Personality Insights Model.

3.1 NLU Model

Analyze text to extract metadata from content such as concepts, entities, keywords, categories, sentiment, emotion, relations, and semantic roles using natural language understanding [35].

NLU Model provides these traits:
- Joy
- Disgust
- Anger
- Fear
- Sadness

3.2 Personality Insights Model

Predict personality characteristics, needs, and values via written text. Understand customer habits and preferences on an individual level—at scale [2].

Personality Insights Model provides these traits:
- Openness
- Conscientiousness
- Extraversion
- Agreeableness
- Emotional range
- Challenge
- Closeness
- Curiosity
- Excitement
- Harmony
- Ideal
- Liberty
- Love
- Practicality
- Self-Expression
- Stability
- Structure
- Conservation
- Openness to change
- Hedonism
- Self-Enhancement
- Self-Transcendence

4 CUSTOM ALGORITHMS

4.1 Data Preprocessing Model
- Remove punctuation from words (e.g. 'what’s').
- Removing tokens that are just punctuation (e.g. ‘‘’).
- Removing tokens that contain numbers (e.g. ‘10/10’).
- Remove tokens that have one character (e.g. ‘a’).
- Remove tokens that don’t have much meaning (e.g. ‘and’).
- Remove any non-ASCII characters (e.g. ‘’).
- Encode whole text content to UTF-8, for simplicity.

For better understanding, let’s clean the following textual data:

"It’s all about one’s hard work. Luck is not anything."

After removing punctuation from words, we get:

"It all about one hard work. Luck is not anything."

After removing punctuation marks, we get:

"It all about one hard work Luck is not anything"

The data doesn’t have any numeric, non-ASCII, meaningless tokens, and single characters. So, we can say the data is cleaned now.

Final cleaned data is:

"It’s all about one’s hard work. Luck is not anything."

4.2 Data Dimensional Reduction Algorithm

- Find the mean vector.
- Assemble all the data samples in a mean adjusted matrix.
- Create the covariance matrix.
- Compute the Eigen vectors and Eigen values.
- Compute the basis vectors.
- Represent each sample as a linear combination of basis vectors.

For better understanding, let’s reduce the following data using PCA to 3-dimensional data [4]:

<table>
<thead>
<tr>
<th>#</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Mean vector (= [(1/n) * ΣXi]):

[2.5, 2.75, 4.75, 4]

Centered column dataset (= [Xi - µ])

<table>
<thead>
<tr>
<th>#</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.5</td>
<td>-1.75</td>
<td>2.25</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>-1.5</td>
<td>-0.75</td>
<td>-1.75</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
<td>2.25</td>
<td>2.25</td>
<td>-2</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.25</td>
<td>-2.75</td>
<td>0</td>
</tr>
</tbody>
</table>

Covariance Matrix (= [(1/(n-1)) * (Xi - µx)(Yi - µy)])
Eigen Decomposition of Covariance Matrix:

**Eigen Vectors:**

<table>
<thead>
<tr>
<th>#</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.66666667</td>
<td>1.83333333</td>
<td>1.66666667</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>1.83333333</td>
<td>2.91666667</td>
<td>0.58333333</td>
<td>-3</td>
</tr>
<tr>
<td>2</td>
<td>1.16666667</td>
<td>0.58333333</td>
<td>6.91666667</td>
<td>1.33333333</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>-3</td>
<td>1.33333333</td>
<td>4.66666667</td>
</tr>
</tbody>
</table>

**Eigen Values:**

<table>
<thead>
<tr>
<th>#</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1.1022302e-15</td>
</tr>
<tr>
<td>1</td>
<td>1.2007468e+00</td>
</tr>
<tr>
<td>2</td>
<td>7.2695236e+00</td>
</tr>
<tr>
<td>3</td>
<td>7.7770683e+00</td>
</tr>
</tbody>
</table>

Here, we see last two columns have high rank of all others, so taking those two columns only.

**Projected Dataset**

<table>
<thead>
<tr>
<th>#</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.44089210e-16</td>
<td>-1.30737110e-01</td>
<td>5.59026969e-01</td>
<td>4.12860711e+00</td>
</tr>
<tr>
<td>1</td>
<td>2.77555556e-16</td>
<td>-1.32274832e+00</td>
<td>1.79336750e+00</td>
<td>1.38172714e+00</td>
</tr>
<tr>
<td>2</td>
<td>2.22044605e-16</td>
<td>6.84523439e-02</td>
<td>3.96124776e+00</td>
<td>-8.23911213e+00</td>
</tr>
<tr>
<td>3</td>
<td>9.0066208e-16</td>
<td>-1.28046906e-00</td>
<td>1.60853300e+00</td>
<td>1.92296875e+00</td>
</tr>
</tbody>
</table>

Above experiment clearly shows that up to a maximum extent (above which accuracy could not be increased further) is directly proportional to the word count. Mathematically,

\[
A \propto n \quad (\text{up to an extent})
\]

\[
A = \text{accuracy of model} \quad \mid n = \text{word count}
\]

6 CONCLUSION

Hence, it can be concluded by above research work that we can form groups of compatible people by analyzing their speech, views, social media updates etc. and comparing their personality traits with those of others which could prove to be best and most efficient groups. And, accuracy of the groups’ formation can be increased by increasing the text content.

7 REFERENCES