

Machine Learning & Personality:

Minimizing the number of questions required to identify Personality Traits

1. Introduction

This project is a continuation of a current large scale project that I am working on. We are building an application that allows users to discover professional opportunities and connections through a quiz based system. For the project, we wanted to see if it was reasonable to take a 300 question personality assessment and turn it into a 20 question assessment with similar results. This was an exploration to identify possibilities of implementation as an added feature to the application. Post research, the results from this project will be utilized within the application to help connect users with better opportunities. All research was done on the Big 5 Personality Assessment (IPIP-NEO 300) Personality assessment & Narcissism Personality Index (NPI) but applications could be used with any set of questions that use percentile to classify users.

2. EDA & Latent Feature Identification

Principle Component Analysis (PCA): To begin this project, it is interesting to see the questions within both IPIP-NEO and (NPI) that separate users the most. This implies that by using PCA, I can easily understand the questions that users have the largest spread of answers. Directly, this implies these questions *split* users the most. This is very interesting as we can use these questions to match users. Here are a couple of the questions that split users the most from the IPIP-NEO assessment were I split out users based on gender.

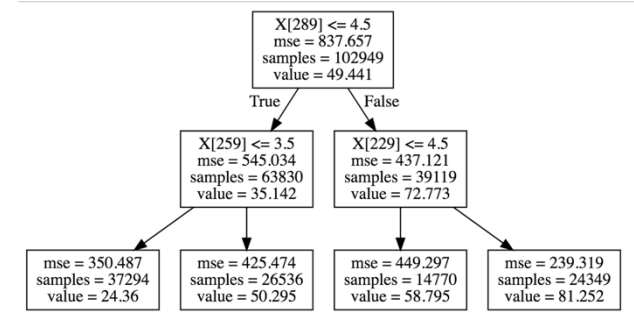
Correlation Between Personality Facets:

Another interesting aspect of this dataset is that the 5 personality traits all contain 5 facets within each. By looking at the 6 questions that make up each facet, it is possible to understand if specific facets are correlated with each other outside of their trait category. By applying this analysis, I immediately recognized there exists signal overlap between different categories. This lead me to discover the possibility of implementing a Decision Tree using these features to predict a users' percentile rank.

Male Top Dividing Questions	Female Top Dividing Questions
Believe that there is no absolute right or wrong.	Boast about my virtues.
Tend to vote for liberal political candidates.	Believe that I am better than others.
Believe that I am better than others.	Love to eat.
Boast about my virtues.	Believe that we coddle criminals too much.
Act wild and crazy.	Believe people should fend for themselves.
Don't understand people who get emotional.	Enjoy thinking about things.
Know how to get around the rules.	Value cooperation over competition.
Believe that criminals should receive help rather than punishment.	Believe that criminals should receive help rather than punishment.
Am not bothered by disorder.	Know how to get around the rules.
Do crazy things.	Do crazy things.
Like music.	Have a vivid imagination.
Dislike loud music.	Put people under pressure.
Hate to seem pushy.	Can't stand weak people.
Seldom toot my own horn.	Never splurge.
Use flattery to get ahead.	Use flattery to get ahead.

3. Building & Traversing a Decision Tree to Identify Big 5 Personality Traits

By building a decision tree for each of the personality traits, I was able to identify the personality percentile that user falls into with mean abs error < 15% for each trait. This was done with only 4 questions per trait, and error can be minimized further by adding more questions. After building the decision tree, I implemented OOP approach to create a system that allowed the user to traverse the decision tree by answering at each split and thereby having a unique path through the questions. All users start at the same node and progress through the splits in the tree.



4. Implementing Recommender System to Predict Facets within each Trait

After identifying the 5 Big Personality Traits, the user has already answered a minimum number of questions. This allows me to use other users answers to create a recommender system that can predict where the test-taker will fall on the spectrum for a multitude of facets that exist under the Big 5 Personality assessment. By assessing error, I was able to identify an optimum recommender system.

5. Online Website using Flask Server and JS

Once finished with the exploration and modeling, I was able to implement the tree traversal into an online web application using Flask and JavaScript. This is available online by accessing my online digital portfolio.

6. Future Work

I would like to use this system within our application and am currently working on developing with React-Native to implement this system within our application.

