FRIEND AFFINITY FINDER
Team EDGE

What is the uniqueness/novelty added by you to the defined problem statement?

The problem statement is all about finding affinity score among friends. We need to find friends with common thinking and same properties, so that they can constitute best teams.

Our solution for the same is analyzing social activities (like Tweets) of all of them and getting basic properties (Personality Needs, Personality Big Five Properties and Sentiments etc.) of them enlisted. We then need to compare those properties of all people and those visualize it in a 3D graph, which will give an idea about affinity score among people. More distant points in the graph depict less affinity and closer points depict high affinity.

How is the proposed solution impacting the business? How are the business processes simplified or bringing value over the existing process?

The solution is surely impacting the business in very effective manner, because after implementation we can group up the people with common thinking and properties which would surely result in these:
- Better compatibility between people
- Better communication among team members
- Less chances of inter-group grievances
- Better output with team efforts
- And much more…

In existing process, we're grouping up people on the basis of either of these:
- Skill set
- Departments
- Etc.

But this method, most of the times fails because it's not obvious that people with same skill set or within same department become good friends. For having good friendship, people need to compatible and to have common thinking.
Our solution makes the work more easy and more accurate as compared to the existing process.

Architectural flow of the proposed solution, with the mention of technologies to be used in developing the solution.

Architectural flow of the solution is as follows:
- User logs in to Twitter Developer Account
- User selects which person to analyse (either any friend or follower, or own self, or raw text).
- User gets a list of people just analysed with their properties and scores.
- Dataset having large number of properties is reduced using PCA (ML Algorithm).
- User gets affinity 3D graph to estimate affinity among analysed people.

Technologies to be used:
- Python3 (for Backend Server and Machine Learning)
- JavaScript, HTML5, CSS3 (for Frontend and basic programming)
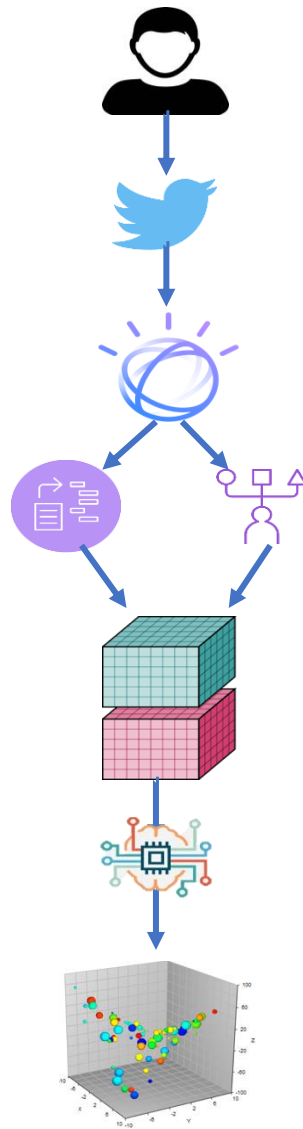
Define the scope of work to be implemented in the project with modules etc.

Module-1: Authentication Phase
This module authenticates user when user enters 4 keys from his/her Twitter Developer Account. If the user is authenticated, user is allowed to proceed to next module.

Module-2 : Fetch Tweets
This module allows user to select any friend/follower/own self on twitter or user can also feed raw text in this module. After fetching tweets of the selected person, it sends the textual data to next module.

Module-3 : Analyze
This module analyzes the textual data and mines various properties resembling the person's personality. These properties have specific scores with them. These scores are sent to next module.

Module-4 : Visualize
This module takes large dataset having various person's specific properties with score. This dataset is reduced using PCA algorithm of ML and then a 3D graph is rendered for the user which shows the affinity among the analyzed people (on the basis of their properties).