Problem Description

The following data was provided by the courtesy of yenta, which is a professional networking app launched by Japanese startup Atrae that uses artificial intelligence (AI) to optimize profile matching for its users.

Atrae strives to improve the AI it uses on its platform to enable yenta users to make new, valuable connections and expand their networks.

The goal of this competition is to optimize yenta's matching algorithm by predicting the compatibility of two app users.
This ensures that the app recommends the most relevant profiles to each user.

Our team has presented an approach towards improvising the recommendation algorithm of the same.

Matching

Swipe right on interesting profiles. Start chatting if they like your profile too.

Problem Statement

The task of this competition is to predict the level of compatibility of two given users to improve the profile recommendation algorithm for yenta. For this purpose, we classify the level of compatibility between user A and user B into 4 categories:

• **No Match = 0:** At least one of either user A or user B swiped left on the other, meaning there is no possibility of a match.

• **Match = 1:** Both user A and user B swiped right on each other and matched.

• **Matched and met but unfavorable review = 2:** Both user A and user B swiped right on each other and matched, then met. After the meeting, user A gave user B a review of 1-3 out of 5 (an “unfavorable” review).

• **Matched and met and favorable review = 3:** Both user A and user B swiped right on each other and matched, then met. After the meeting, user A gave user B a review of 4-5 out of 5 (a “favorable” review).

To build this model, we provide 2 different types of data subsets: user data and interaction data.
Note that all of the data is anonymized through the use of alias IDs and multi-step vectorization models to ensure that user privacy is protected. IDs with low frequency are grouped into a category labelled “other” with an ID of 999999.

I. User data: These files are connected through the user_id column (e.g. 41245)

- user_ages.csv:
  # user_id: user ID
  # age: user age (in years)
- user_educations.csv:
  # user_id: user ID
  # school_id: school ID
  # degree_id: degree ID (just for some users)
- user_works.csv:
  # user_id: user ID
  # company_id: company ID
  # industry_id: company's industry ID
  (please note: one company can have multiple values; also, this column is user-selected, so values are not

necessarily tied to company ID, which means that the same company ID can have different values for different
users)
over_1000_employees: variable indicating if the company has over 1000 employees or not
(please note: this column is user-selected, so values are not tied to company ID, which means the same company
ID can have different values for different users)

- user_skills.csv:
  # user_id: user ID
  # skilld_id: skill ID
- user_strengths.csv:
  # user_id: user ID
  # strength_id_x: the number of votes that the user has received as review from other users
- user_purposes.csv:
  # user_id: user ID
  # purpose_id_x: whether the user marked "x" as a reason for using the app or not
- user_self_intro_vectors_300dims.csv:
  # user_id: user ID
  # num_char: number of characters in the user's self-introduction text
  # num_url: number of URLs in the user's self-introduction text
  # num_emoji: number of emojis used in the user's self-introduction text
  # self_intro_x: value for dimension "x" of the user's vectorized self-introduction text (out of 300 dimensions)
- user_sessions.csv:
  # user_id: user ID
  # timestamp: session timestamp

II. Interaction data: These files are indexed by from-to user_id pairs (e.g. 12345-52462)

- interaction_review_comments_300dims.csv:
  # from-to: user ID of reviewer-user ID of reviewed
  # review_comment_x: value for dimension "x" of the review comment (out of 300 dimensions)
- interaction_swipes.csv:
  # from-to: user ID of swiper-user ID of target
  # timestamp: timestamp of the swipe event
  # swipe_status: result of the swipe (-1 = not interested, 1 = interested)
- interaction_review_strengths.csv:
  # from-to: user ID of reviewer-user ID of reviewed
  # strength_id: ID of the strength evaluated by the reviewer

III. Train and test files: In order to train the model, we provide a train.csv file with pairs of user IDs and their corresponding scores

- train.csv:
  # from-to: user ID of scorer-user ID of target
  # score: compatibility score ID (0-3)
- test.csv:
  # from-to: user ID of scorer-user ID of target (to be predicted)

The solution file to be provided should follow this format: the from-to IDs should be the same IDs contained in the test.csv file, and they must be in the same order.

IMPORTANT NOTE: Score values on the submission file should be formatted with at least one decimal (e.g. 0.0 instead of 0, 1.0 instead of 1) or the system will not be able to score it properly.

• Submission.csv
from-to, score
6280229-6293525, 1.0
670384-50085, 2.0
2271906-4685859, 1.0
...

NOTE: The maximum number of submissions that can be made per day are 3 submission.csv files.

Hypothesis Generation

1. Same age preference.
    a. Plot: Joint Kernel Density Plot
    b. StatTest: ...
2. Likeliness of peers from same educational institute (school) or organization.
    a. Plot: Mean of Ratio for peers having interaction between them and belonging to same school to the peer having no interaction and belonging to same school.
    b. Chi-Square test: of frequencies.
3. People have similar interest have higher chance of having a match.
    a. Purposes id matches.
4. The first level of mutual connection has higher chance of match making, which continues to decrease over subsequent levels.

5. Likeliness of people from same Skills background.
    a. Skill Id
6. Highly rated profile are likely to find a match.

7. Are there Levels of social hierarchy ?
    a. Cluster Formation
8. Complete different peers to have a match based on ones' activity?

9. Features that differentiate opposite polarity peers.

10. Are patterns across cluster similar ?

Variable Identification

Score (Target): Categorical

FILE	Variable	Description
train.csv	score	(target), Categorical
users_ages.csv	age	Continuous: Outliers (<= 4, >=73) Missing Values
user_educations.csv	school_id	categorical

	degree_id	Categorical missing values
user_strengths.csv	strength_id_<n> *8	Continuous
user_purposes.csv	purposes_id_<n> *15	Categorical
user_self_intro_vectors_300dims.csv	num_char	Categorical Needs Clustering
	num_url	Continuous (freq) 26 and 53 are outlier
	num_emoji	Continuous (freq) Missing Values 55 - rowise 68 - rowise
user_sessions.csv	timestamp	Datetime
user_skills.csv	skill_id	Categorical repeated
user_works.csv	comapny_id	Categorical
	industry_id	Categorical
	over_1000_employees	Boolean Missing Values
interaction_review_strengths.csv	strength	Categorical
interaction_swipes.csv	Timestamp	Datetime
	swipe _status	Boolean (-1 and 1) Needs preprocessing
interaction_review_comments_300dims.csv	review_comment_300	Continuous (No Nan value)

Univariate Analysis

User_ages.csv(Completed)
User_works.csv(Completed)
user_skills.csv(Completed)

User_educations.csv (Completed)
Interaction_swipes.csv (error with libraries to process it)
Interaction_review_comments.csv (Completed)

User_purposes.csv
User_strengths.csv (completed)

User_sessions.csv (Don't Do, Not Required at present)
Interaction_review_strength.csv (Completed)

Bivariate Analysis

Load the train.csv and your allotted .csv then merge according to your thought process as if how can the matchmaking be affected by that particular feature or any derived feature that you have thought of.

Use_ages.csv

User_works.csv

User_education.csv
User_skills.csv

User_purposes.csv

user_strenghts.csv

MODEL MAKING

Made a decision tree classifier

HYPER PARAMETER OPTIMIZATION

Using validation set, We have optimized the dept of tree.