TINDER 2.0

**ABOUT:**

We have created Tinder 2.0 with the aim of matching potential partners with better precision than the current application. Users will be asked to fill out a questionnaire to determine their preferred characteristics in a potential suiter, before seeing any possible matches. This then pre-scans all other Tinder 2.0 members, and only shows those who match the users list of preferred characteristics. Members will also be asked to fill out their own details/ characteristics, so the potential matches the view will also like them back. Users will be given multiple suiters, in which they could choose from.

To begin producing this new version of Tinder, we researched different possible theorems and algorithms that could help us in creating the right code on Matlab. We took inspiration from the Gale Shapley algorithm and constructed a variant of it ourselves, in order to help create Tinder 2.0, which has a similar concept to the stable marriage problem.

The Gale Shapley algorithm is one way of solving the stable marriage problem, where n men need to be matched with n women to produce n married couples. However, there can be no two people who would prefer to be with each other more than their current partners. To ensure this doesn’t happen, each male and female must rank their prospective partners in order of preference so they can be married with their most suitable, available male/female. In order for the algorithm to work, we must assume that all couples are heterosexual, and there are no same sex couples. Once everyone has made their internal compatibility ranking of the opposite sex, either the male group or the female group are selected to carry out the proposals. Research shows that men are 30% more likely than women to initiate conversation on dating apps, therefore we shall assign the men as being the ones who make the proposals. To demonstrate this algorithm, we will use an example.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MALES | RANKINGS | | | | |
| Adam | O | M | N | L | P |
| Ben | P | N | M | L | O |
| Callum | M | P | L | O | N |
| Dave | P | M | O | N | L |
| Eric | O | L | M | N | P |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FEMALES | RANKINGS | | | | |
| Lucy | D | B | E | C | A |
| Mel | B | A | D | C | E |
| Nicole | A | C | E | D | B |
| Olivia | D | A | C | B | E |
| Polly | B | E | A | C | D |

PROPOSERS

Adam will be the first male to propose to his highest ranked female, Olivia. As she has not previously rejected Adam and has not yet had any other marriage offers, she accepts. Ben will then propose to his most favoured partner Polly, and as she has neither rejected him or received any other offers, she accepts. Callum is next and proposes to Mel, she is under the same circumstances as Olivia and Polly, so accepts. Dave now proposes to Polly, however she is already engaged to Ben. As Ben is Polly’s most preferred match, she will not leave Ben for any other male, therefore Dave is rejected, and looks to propose to his next best match, Mel. Mel leaves Callum for Dave, as her ranking states that Dave is more preferable. Callum is now single, but it is Eric next in line to propose. He proposes to Olivia, but she is happy and prefers Adam. Eric’s next best option is to propose to Lucy, and she accepts. Callum is now the only single male and tries to propose to Polly, Lucy and Olivia, but they all prefer their current partners to Callum. He finally proposes to Nicole who accepts.

To conclude this example, we have 5 stable marriages, Adam and Olivia, Ben and Polly, Callum and Nicole, Dave and Mel, Eric and Lucy. All couples are with their most preferred partner possible.

**MOTIVATION:**

Tinder was launched in 2012 with the hope in changing people’s views on being single. “It was designed to celebrate that being single is a journey and isn’t a thing you do unhappily before settling down.”[[1]](#footnote-1) With users being able to link their Facebook, Instagram and Spotify accounts to their Tinder profile, it connects people with similar interests who wouldn’t have otherwise met in real life. Since its launch in 2012, Tinder is now estimated to have 50 billion users in 169 countries, made accessible by the 40 different language options. These 50 billion are responsible for the 1.6 billion swipes made on Tinder every day. Anyone of age 13 and upwards can create a Tinder profile, with privacy restrictions for those in the 13-17 age group. Even with the app being available to everyone, roughly 80% of its members are millennials, forever changing the young adult’s world of dating. Users can tailor their search for a match by selecting the age range, gender and distance from their current location.

However, a study found that 32.9% of Tinder users uninstall the app within 24 hours of downloading it, but why? “On average, users spend up to 35 minutes on the app per session.” [[2]](#footnote-2)If unsuccessful in finding a suitable match, it’s easy to imagine the frustration and loss of confidence experienced by thousands of people, possibly leading to them deleting the app all together. With Tinder 2.0, we aim to change that.

The main motivation behind creating Tinder 2.0 was to save users time. By communicating the type of person they’re looking for through answering the initial questionnaire, users will no longer have to sift through endless profiles which are mostly not what they’re looking for. With Tinder 2.0, users will see fewer potential partners, but each will be better suited and more likely to be a match. With this more refined search the 35 minutes spent per session has a stronger likelihood of being useful in the quest for a match. US adults spent an average of 3 hours and 35 minutes a day on their smartphones in 2018, with 90% of this time being dedicated to apps. Most of this is accounted for by small 5 -10 minute smartphone sessions when either in the back of an uber, on the train, or waiting on another person. If Tinder was prepped ready with a list of suitable dates for the user, members could quickly check the app when they have a few minutes to spare, instead of setting aside a longer amount of time to swipe left or right on the masses.

It also saves users time in not having to read each potential partner’s bio. By filling in the questionnaire, users will automatically know that the members visible fit the criteria that they themselves have set. This again makes the app more suitable to fit in with busy modern-day life, only needing a few minutes to swipe on each profile, with no need to read their full bio. Whilst a profile photo is requested when setting up a Tinder account, filling in the small bio box isn’t. With no additional information about the potential partner apart from their photo, it can be hard to judge whether or not they’re what you’re looking for. Even if they have got a bio, it can often not include the information you’re looking for. Tinder 2.0 eliminates the uncertainty you might have about any of the members. From the questionnaire you’ll be able to know details that might not be visible in the profile photo or that are missing from the bio. For example, you will have pre-selected the height range in which you are looking for a potential match to be in, as well as their income range and whether or not they smoke.

In addition to saving users time, Tinder 2.0 is also designed to make dating and using the app a more enjoyable experience. With all members filling out a questionnaire about their own characteristics as well as what they’re looking for in a potential partner, this will give members piece of mind knowing that the matches they’re viewing are looking for some of the characteristics they hold, making them more likely to swipe right on each other and get a match. With a more select group of potential dates, users will feel more confident in knowing that the profiles they’re viewing all have the characteristics they require. With this confidence, new members will find the process of dating via Tinder 2.0 less daunting, knowing that there are plenty of potential matches lined up and waiting for them. For those less confident members, choosing the characteristics and knowing a bit about your potential partners prior to matching provides multiple conversation starters/ topics to discuss, which can be the most challenging part. With this new version, users receive a personal experience tailored to their own requirements. This further builds the trust and belief in the app from the members, and ensures them that Tinder 2.0 is invested in helping each and every member find their suitable match. With this said, customer satisfaction will increase, seeing the 32.9% of Tinder users who uninstall the app within 24 hours decrease.

**METHOD:**

DATA CREATION:

To begin creating Tinder 2.0, we first used matlab to create our data set. We set a sample of 1000 people, and then split them equally into 500 males and 500 females. As well as this, we incorporated the fact that some members of the sample will be looking for same sex relationships, so made 333 homosexual, and 667 heterosexual. To assign different members of the sample with either homosexual or heterosexual preferences, we applied an if condition of ==0 meant that the member was heterosexual and ==1 meant that the member was homosexual. We assigned gender to sample members in the same way. However, with sexuality, we were required to come up with a condition where if for example a female said they were homosexual, then only other homosexual females came up as possible matches. In order to solve this condition later on in the coding, we assigned females ==0 and males ==1.

To assign members with names, we randomly generated 1000 forenames and surnames, which were withdrawn from an excel document and then we converted them into vectors.

We next started to give our sample of members own individual characteristics, starting with hair colour, location and ethnicity. We used 4 different hair colour categories, brunette, black, blonde and red haired, 3 ethnicity types, black, white and Asian and 5 different location catagories, south England, north England, midlands, Wales and Scotland. We created the distribution for hair colour, location and ethnicity in the same way. Each category was assigned a different value, e.g. brunette =1, black =2, blonde =3 and red =4 then black =1, white =2, and Asian =3. After assigning each category a value, we used the randi function on matlab to generate a vector with length 1000 to randomly generate a combination of the different hair colours, locations and ethnicities (3 separate functions). This condition resulted in each number being assigned to a hair colour, location and ethnicity.

The next characteristics we introduced were the persons income, height and age. To assign each sample member with an income, age and height, we used the randi function to randomly distribute values. Values between £20,000 and £120,000 were assigned for income per annum, a height between 120cm and 180cm and applied an age range of 18 to 80 years old. When selecting the preferences of your potential partners, we created bands so you the member could select a group of heights they would like their matches to be e.g. 160cm– 180”. We created these characteristics in a similar way to the ethnicity and hair colour, however the nature of this data was that each input already had a numerical value assigned e.g. each person in the sample already has an age, height and income, so each category didn’t need to be assigned a value. In this case we just used the randi function to create a 1000 length vector, providing us with 1000 samples.

The final characteristics we applied were whether or not the members were smokers/ non-smokers and whether they were parents/ non-parents. To apply these characteristics randomly to our sample, we generated a code where half are assigned the value of ==0 and the other half assigned the value of ==1. In both characteristics, we assigned ==0 to the half who were either non-smokers or non-parents. We used a random permutation function to randomly assign 500 of the data set with ==1 (500 of the sample will be smokers/ parents) and assign the other 500 of the data set with ==0 (500 of the sample are non-smokers / non-parents).

CODE:

Once we created the database, we started to create Tinder 2.0 on matlab. When the user fills in the questionnaire to find a match with the preferred characteristics, we implemented a find function on matlab. For example, if the user is looking for a match between the age of 18 and 25, the find function will pick out everyone in the sample whose age is in this bracket. We used this method of finding the right people from the sample for every different category of preferences, e.g. hair colour, location etc, until we had vectors filled with numbers which correlate to the column that the matched individual is in.

We then needed to find the intersection of 2 variables (categories) where the intersection represents the potential matches that fit the preferences the user selected at the beginning. We first found the intersection of gender and age, calling this intersection ‘Userlike’. We then found the intersection of Userlike and ethinicity, calling this ‘Userlike’ aswell, and overriding the first userlike we created, so it now includes the intersection of gender, age and ethnicity. We repeated this process until we ended up with a vector filled with individuals who matched every single preference stated by the user. A visual of the method we used to find the vector can be seen below.

Userlike = intersect(Gender,age)

Userlike = intersect(Userlike,ethnicity)

Userlike = intersect(Userlike,haircolour)

Userlike = intersect(Userlike,income)

Userlike = intersect(Userlike,height)

Userlike = intersect(Userlike,smoke)

Userlike = intersect(Userlike,parent)

Userlike = intersect(Userlike,location)

We then repeated this process to match each user’s individual personal information to the database preferences, to configure a vector filled with individuals who likes the user based on their own personal details. We called each intersection ‘Databaselike’, and called each category ‘prefcatagory’ to differentiate between an individual’s own characteristics and their preferred charactersitcs. This will help us later on to match pairings who like each other. This process is shown below.

Databaselike = intersect(prefGender,prefage)

Databaselike = intersect(Databaselike,prefethnicity)

Databaselike = intersect(Databaselike,prefhaircolour)

Databaselike = intersect(Databaselike,prefincome)

Databaselike = intersect(Databaselike,prefheight)

Databaselike = intersect(Databaselike,prefsmoke)

Databaselike = intersect(Databaselike,prefparent)

Databaselike = intersect(Databaselike,preflocation)

We then found the intersection of ‘Userlike’ and ‘Databaselike’ to find a set of potential matches who all like each other, and called this intersection Prefmatch.

Our next challenge was to differentiate between those in the sample who were looking for same sex couplings, and those who were looking for opposite sex couplings. as previously explained, males were assigned ==1, whilst females were assigned ==0. In a separate vector, homosexuals were assigned ==1, whilst heterosexuals were assigned ==0. We needed to add these 2 vectors together, which is shown in the table below.

|  |  |  |
| --- | --- | --- |
|  | Male | Female |
| Homo | **2** | **1** |
| Hetro | **1** | **0** |

* If someone was male(==1) and homosexual(==1), adding them together would equal 2.
* If someone was female(==0) and heterosexual(==0), adding them together would equal 0.
* However, if someone was heterosexual and male, or female and homosexual, they would both equal 1.

In the case where the vectors are added and equal 1, we had to insert an if function, where if both the heterosexual and male buttons were clicked we would use a find function to locate the individuals that suit this criteria. Following this, we would find where gender =1 and homosexual =0, and where gender =0 and homosexual =1 and call the intersections of the gender and sexuality preferences ‘homomatch’.

After this has been configured, we found the intersection of ‘prefmatch’ and ‘homomatch’ to ensure that the match is the right gender and has the correct preferences. Finally, we mapped the names back onto the elements of the final matches vector, and display them in the GUI. This results in the user viewing a pool of potential matches who are the correct gender, with the desired characteristics and who like the preferences of the user.

**RESULTS:**

Our aim was to re-design the Tinder application in order to better match users to people with desired characteristics as well as the matched liking the user back. We thing we have managed to achieve this. The code we have created on matlab successfully displays a set of suitable matches to the user.

However, even though we managed to create the application, we still faced problems throughout the process. We came across difficulties when trying to match the code we wrote on callback buttons on GUI to the function that displays the algorithm to a stable match. It was a process of trial and error which was time consuming, and could have been done more efficiently if we discussed and worked though the difficulties we were facing as a group before doing our own individual research into the problem. We also spent time over trying to create the algorithm which randomly assigned one preferred match to another, as this was a more complex part of the code. In hindsight this could have been overcome by using the help function on matlab, or looking on the mathworks web page for inspiration from previous similar codes.

Overall, we believe we have been successful in what we set out to achieve, and hope this application has bettered the quality and quantity of matches of the original Tinder application.

<https://www.youtube.com/watch?v=0m_YW1zVs-Q>

<https://www.geeksforgeeks.org/stable-marriage-problem/>

https://www.algorithm-archive.org/contents/stable\_marriage\_problem/stable\_marriage\_problem.html

<https://www.gotinder.com/jobs?locale=en-GB>

https://expandedramblings.com/index.php/tinder-statistics/

https://www.factinate.com/things/24-fiery-facts-tinder/

<https://www.datingsitesreviews.com/article.php?story=study-reveals-which-dating-apps-are-most-popular--and-which-get-deleted-first>

<https://www.emarketer.com/content/mobile-time-spent-2018>

https://www.independent.co.uk/life-style/love-sex/gender-roles-male-female-who-messages-first-dating-apps-study-bumble-tinder-eharmony-oxford-a8557646.html

1. <https://www.gotinder.com/jobs?locale=en-GB> [↑](#footnote-ref-1)
2. https://www.factinate.com/things/24-fiery-facts-tinder/ [↑](#footnote-ref-2)