Projet Neo4j : le speed-dating

Marie JOIGNEAU
Laura PAPER

I - Introduction

a) Contexte

Intérêt du speed dating.

b) Notre jeu de données

Le jeu de données a été compilé par les professeurs Ray Fisman et Sheena Iyengar de l'école de Business de Columbia pour leur papier "Différence de genre dans la sélection de partenaire : mise en avant par une expérience de speed dating".
Les données ont été rassemblées dans des expériences de speed dating entre 2002 et 2004, divisées en plusieurs vagues ou sessions. Pour chaque évènement, les participants ont un premier rendez-vous de 4 minutes avec chacun des participants du sexe opposé. A la fin de la rencontre, il leur a été demandé s'ils veulent voir leur rendez-vous à nouveau, ainsi que de noter leur partenaire.

Les variables sont basées sur un questionnaire distribué aux participants. Tout d'abord l'individu a noté le partenaire rencontré sur 6 attributs : attractivité, sincérité, intelligence, fun, ambition et intérêts communs. Ensuite les autres domaines qui ont été mis en avant sont la démographie, les habitudes de rencontre, la perception personnelle à travers des attributs clés, les croyances sur ce que les autres trouvent important dans une relation, et des informations sur le style de vie.

De façon plus pratique, sur cette base de données, une ligne représente une rencontre entre un individu et un partenaire. Ainsi, si un individu rencontre 10 personnes, il y a 10 lignes qui représentent cet individu.
Ce data frame comprend 195 variables. Voici les différents groupes de variables du data frame que nous avons décidé de mettre en avant :

- Les caractéristiques de l'individu : identifiant unique du sujet (iid), genre (gender), lieu de vie (from), domaine de travail (field) et intérêt de l'individu pour certains domaines (sport, culture…), l'intérêt étant évalué sur une échelle de 0 à 10.
- Les caractéristiques des rencontres : nombre de personnes qui se sont rencontrées (round), la vague ou session de la rencontre (wave), s'il y a eu un match ou non (match = 1 ou 0)
- Les caractéristiques du partenaire : identifiant unique (pid), ses notes et préférences.

c) Notre problématique

Nous nous sommes intéressées à une représentation de type graphe de cette base de données grâce à Neo4j. Sur la base de ces individus, nous allons voir les chemins qui les relient les uns aux autres. Ainsi nous allons avoir une représentation visuelle de qui a rencontré qui, qui a matché avec qui, qui est intéressé par qui…et ainsi avoir une visualisation plus claire des relations entre les différents individus.

Ainsi, deux problématiques se dégagent. Tout d'abord, nous avons voulu savoir quels sont les individus qui ont eu le plus de match et qu'est ce qui les caractérisent, en séparant les individus par genre dans un second temps.
Ensuite, nous aimerions utiliser cette base de données pour permettre à l'utilisateur de trouver son partenaire idéal en lui laissant choisir certaines caractéristiques. Ceci permettrait de lui sélectionner les individus qui correspondent à ces critères, dans un but potentiel de réaliser un speed dating avec cet individu.

Traiter cette problématique à l'aide d'une base de données de type graphe est plus simple et intuitif qu'avec seulement un dataframe. Cela exploite bien les relations entre les individus. Elles peuvent être visualisées de façon plus intuitives.

Nous avons choisi ce jeu de données pour répondre à cette problématique car il est complet d'une part (7411 rencontres, 518 individus et 195 variables), et bien documenté d'autre part (un fichier word explique en détail la démarche et la signification de chaque variable). De plus, les variables sont pertinentes au vu de notre problématique (match ou pas, caractéristique des individus ect). Enfin, ce sujet est d'actualité et peut être intéressant pour toute personne désirant réaliser un speed dating ou souhaitant simplement en apprendre plus sur les clés de l'attirance entre deux individus.

I - Pré traitement

a) Lecture du dataframe

Tout d'abord nous avons ouvert le dataframe, ne gardant que les 20 premières vagues, soit 7411 rencontres sur 8378.

b) Ajout de l'identifiant aux variables iid et pid

Afin de ne pas avoir de confusion avec d'autres variables quantitatives, les variables iid (identifiant unique de l'individu) et pid (identifiant unique du partenaire) ont été modifiées, passant de iid = 1 à iid = id1, et pid = 10 à pid = id10. Pour cela, une boucle for a été nécessaire, ainsi que du repérage de NaN.
Une liste pid_nan a été créée pendant le processus, permettant de repérer les NaN de la colonne pid. Elle sera utile pour plus tard.

c) Création de la variable total_matches

Une partie de notre problématique tourne autour des individus qui ont effectué le plus de match. Ainsi, il a été nécessaire de créer une variable total_matches qui n'était pas présente dans le jeu de données. Ainsi, dans un premier temps, les NaN sont remplacés par des 0. Puis une boucle for tourne sur les variables match pour voir s'il y a des match, et sur iid pour additionner le nombre de match par individu, le tout enregistré sur un vecteur total_match. On a ensuite ajouté ce vecteur en tant que variable dans le data frame.

d) Création de la variable pmatches

Etant donné que dans chaque vague, il n'y a pas le même nombre d'individu qui se rencontre, il a été nécessaire de créer la variable pmatches qui calcule le pourcentage de matches réussi. Cela nous permet de comparer le taux de réussite entre les individus. Pour se faire, on réalise le calcul suivant :

pmatches = total_matches / round
Soit pourcentage de match = nombre total de match / nombre de personnes rencontrées

e) Préparation de variables pour les hobbies

Les variables liées aux hobbies ou activités ont été filtrés puis transformés en integer afin de pouvoir effectuer ultérieurement des graphes dessus plus facilement

f) Doublons dans field

Certains doublons dans la variable domaine (qui correspond au domaine de travail) ont été corrigés.

g) df_unique : 1 ligne 1 individu

Grâce à un subset sur la variable iid (identifiant unique de l'individu), on obtient un dataset comprenant 1 ligne pour un individu. On a 518 individus. Cela est utile pour créer le nœud Individu.

h) df_ville_unique : 1 ligne 1 ville

Grâce à un subset sur la variable from (ville d'origine de l'individu), on obtient un dataset comprenant 1 ligne pour une ville. On a 255 villes. Cela est utile pour créer le nœud Ville.

i) df_wave_unique : 1 ligne 1 vague

Grâce à un subset sur la variable wave (vague auquel l'individu appartient), on obtient un dataset comprenant 1 ligne pour une vague. On a 20 vagues. Cela est utile pour créer le nœud Vague.

j) df_meet : 1 ligne 1 rencontre

Grâce au vecteur pid_nan calculé précédemment, on ne garde que les lignes du data frame où la colonne pid n'est pas un NaN, on obtient un dataset comprenant 1 ligne pour une 1 rencontre. On a 7400 rencontres. Cela est utile pour créer la relation Rencontre.

k) df_interet : 1 ligne 1 intérêt positif

Grâce à la variable dec (décision), on trie et on ne garde que les lignes du data frame avec dec=1 (décision de revoir l'individu) et on obtient un dataset comprenant 1 ligne pour une 1 intérêt positif. Cela est utile pour créer la relation Interet.

l) df_match : 1 ligne 1 match

Grâce à la variable match, on trie et on ne garde que les lignes du data frame avec match=1 (l'individu a matché avec son partenaire) et on obtient un dataset comprenant 1 ligne pour une 1 match. Cela est utile pour créer la relation Match.

m) df_field_unique : 1 ligne 1 domaine de travail

Grâce à un subset sur la variable field (domaine de travail auquel l'individu appartient), on obtient un dataset comprenant 1 ligne pour un domaine. On a 243 domaines. Cela est utile pour créer le nœud Domaine.

Après ce travail de pré processing nous avons pu créer les différents nœuds.

II - Création des noeuds

a) Noeud Individu

Nous avons tout d'abord créé le noeud Individu grâce au data frame df_unique dans lequel chaque individu est présenté par une ligne unique (puisque dans le dataframe initial il y a autant de lignes que de partenaire que rencontre l'individu). Puis nous avons gardé dans ce nœud les caractéristiques propres de l'individu, soit environ 70 variables. Toutes ces variables ne sont pas utilisées dans nos requêtes mais sont néanmoins visibles dans les graphes que l'utilisateur pourra regarder afin qu'il ait accès au maximum d'information possible sur l'individu qu'il regarde.

b) Noeud Ville

Nous avons ensuite créé les nœuds Ville grâce au data frame créé précédemment df_ville_unique, avec une ville par ligne. Cela permet à l'utilisateur de plus facilement visualiser les relations entre les villes et les individus qui vont être créées par la suite, faisant ainsi des regroupements d'individus. Cela sera aussi utile pour des requêtes au sein de notre problématique.

c) Noeud Vague

Grâce au data frame df_wave_unique, nous avons pu créer les nœuds Vague avec les mêmes buts que les nœuds Ville.

d) Noeud Domaine

Grâce au data frame df_field_unique, nous avons pu créer les noeuds Domaine avec les mêmes buts que les nœuds Ville et Vague.

III - Création des relations

Nous avons ensuite créés les relations :

- VIT : entre un individu et une ville, pour voir où vit l'individu, et permet de voir les individus qui vivent dans une même ville
- VAGUE : pour voir dans quelle vague les individus se sont rencontrés
- MEET : pour voir quels individus se sont rencontrés
- INTERESSE : pour voir la relation entre une personne intéressée et la personne qui l'intéresse
- MATCH : pour relier 2 personnes qui ont matchés
- TRAVAILLE : pour voir les sujets qui travaillent dans le même domaine

IV - Création du graphe

Nous avons ensuite pu créer notre base de données sous Neo4j. Voici quelques exemples :

- Un premier graphe reliant les individus en vert labellisées par leur identifiant unique, et les villes en rouge, labellisées par leurs noms (voir figure 1).. Les flèches entre individus symbolisent, des rencontres des matches ou le fait qu'une personne soit intéressée par une autre.
- Un second graphe reliant les individus et les vagues (voir figure 2). On remarque bien les individus gravitant autour des noeuds Vague.

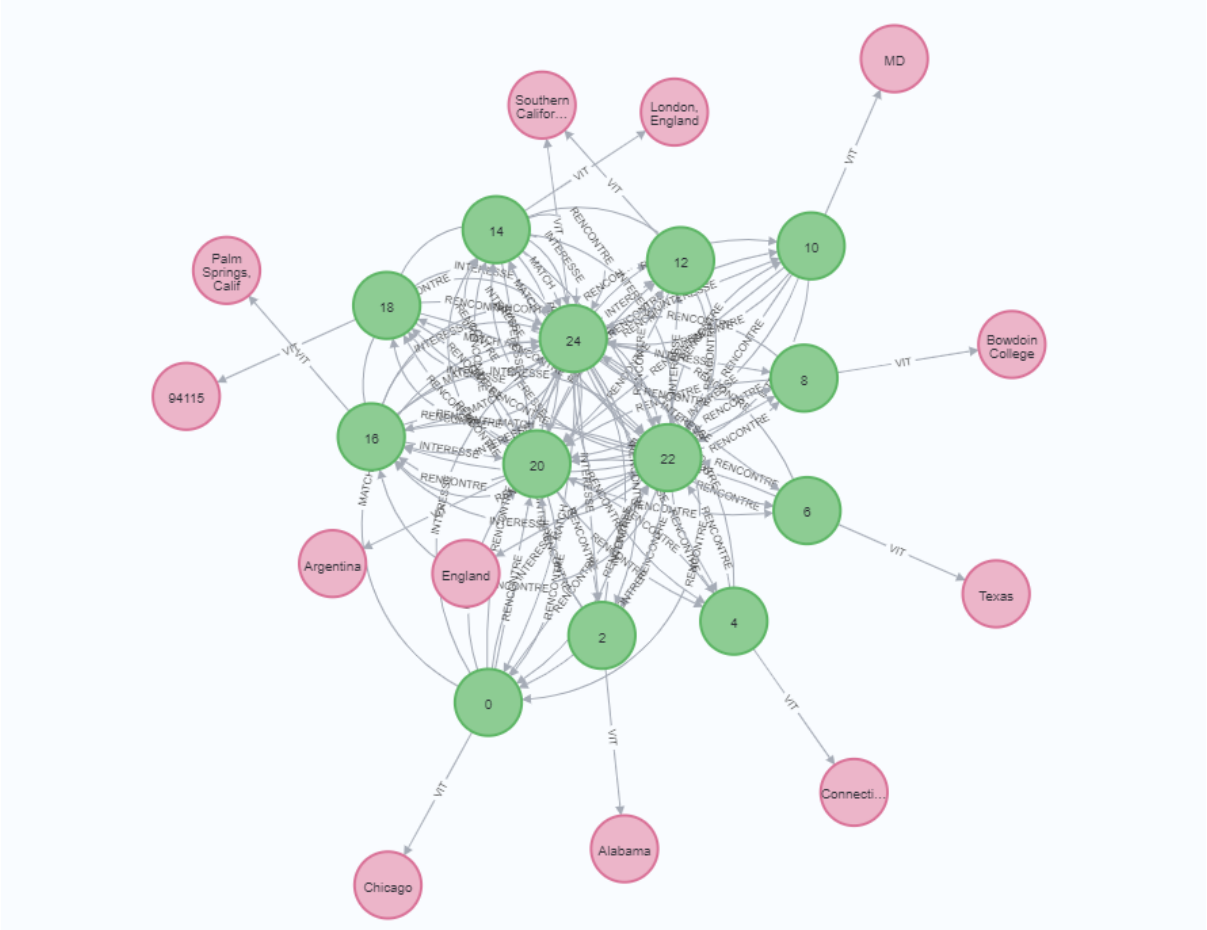

Figure 1 : Noeuds Individu (vert) et Ville (rouge)

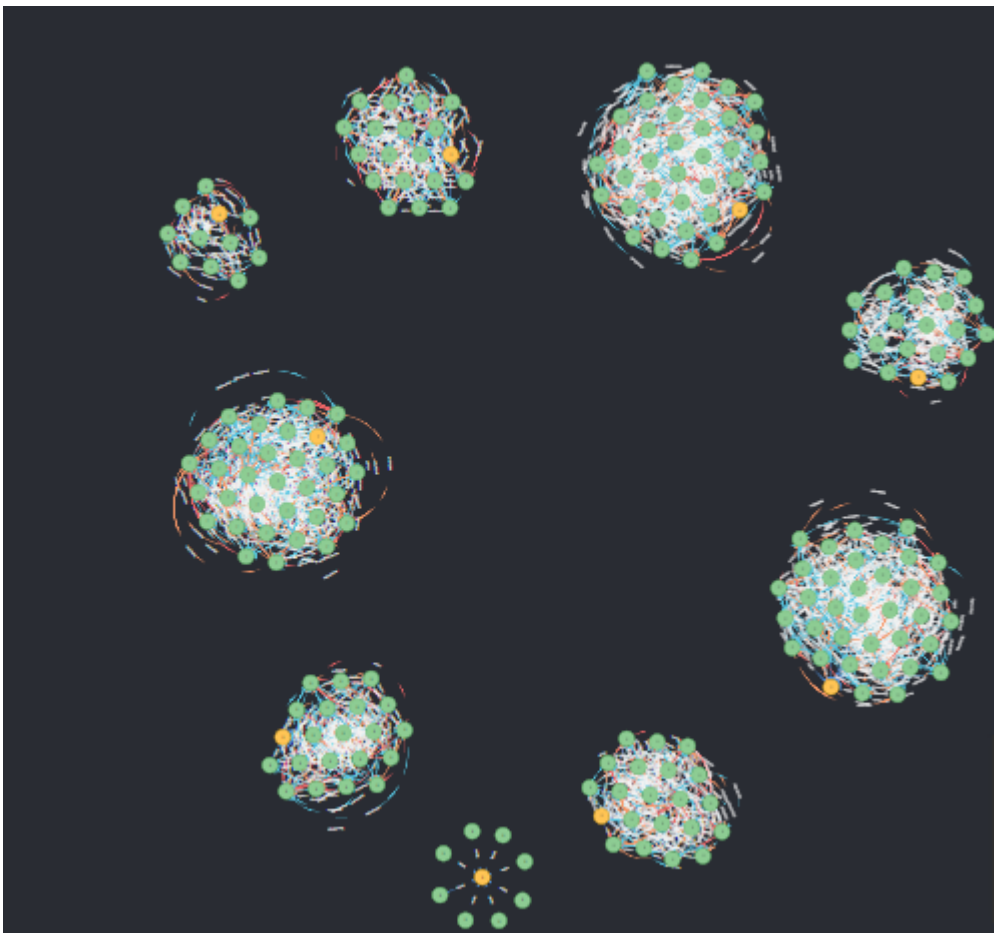

Figure 2 : Noeuds Vague (en orange) et Individu (en vert)

V - Requête Cypher

Une fois le graphe créé nous avons réalisé plusieurs requêtes afin de répondre à notre problématique. Notre partie requête est divisée en trois grandes parties :

- Les caractéristiques des individus ayant le plus de match
- Liste d'individus qui pourraient matcher selon des caractéristiques rentrées par l'utilisateur
- Voir des informations sur un individu

a) Caractéristiques des individus ayant le plus de match

Cette première partie de requête consiste à sélectionner les individus qui ont le plus de matchs en pourcentage du nombre de matchs totaux. Cette requête est faite en 3 temps, dans un premier temps pour tous sexes confondus, dans un second temps pour les hommes puis pour les femmes séparément.

Ci-dessous la première requête réalisée sur les 2 sexes :

```
rq1_1 = "MATCH (i:Individu) RETURN i.pourc_match AS pourc_match,
i.int_sport AS int_sport, i.int_tvsports as int_tvsports,
i.int_exercice AS int_exercice, i.int_diner AS int_diner,
i.int_musee AS int_musee, i.int_art AS int_art, i.int_hiking AS
int_hiking, i.int_clubbing AS int_clubbing, i.int_reading AS
```

```
int_reading, i.int_tv AS int_tv, i.int_theatre AS int_theatre,
i.int_film AS int_film, i.int_concert AS int_concert, i.int_music
AS int_music, i.int_shopping AS int_shopping, i.int_yoga AS
int_yoga, i.iid AS iid ORDER BY pourc_match DESC LIMIT 5"
```

MATCH (i:Individu) : on cherche sur les noeuds Individu
RETURN i.pourc_match AS pourc_match, i.int_sport AS int_sport … : on veut en sortie le pourcentage de match, l'intérêt en sport …
ORDER BY pourc_match DESC LIMIT 5 : on veut les 5 premiers qui ont le pourcentage de match le plus élevé

Deux graphes sont ensuite générés trois fois pour ces trois cas de figure. Le premier permet de visualiser les pourcentages de matchs les plus élevés des individus. Le choix ici du graphe s'est porté sur un barplot vertical, avec les identifiants des individus en index, et la variable sur le pourcentage de match représenté (voir figure 2).
Le second graphe permet de visualiser les caractéristiques de ces individus, ici concentrées sur leur activités ou hobbies. Il a été choisi un barplot horizontal afin d'aider à la visualisation, avec les identifiants des individus en index, et les variables sur les différents hobbies réalisés (de 0 à 10) représentés. Cela permet d'avoir une idée sur les caractéristiques de ces individus (voir figure 3).

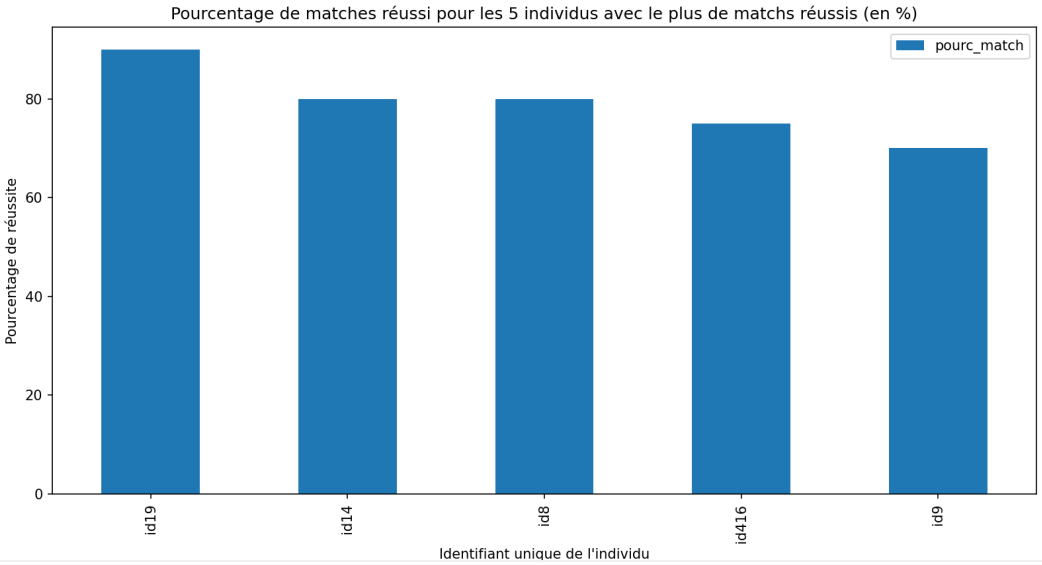

Figure 2 : Pourcentage de matches réussis pour les 5 individus avec le plus de matchs (en %)

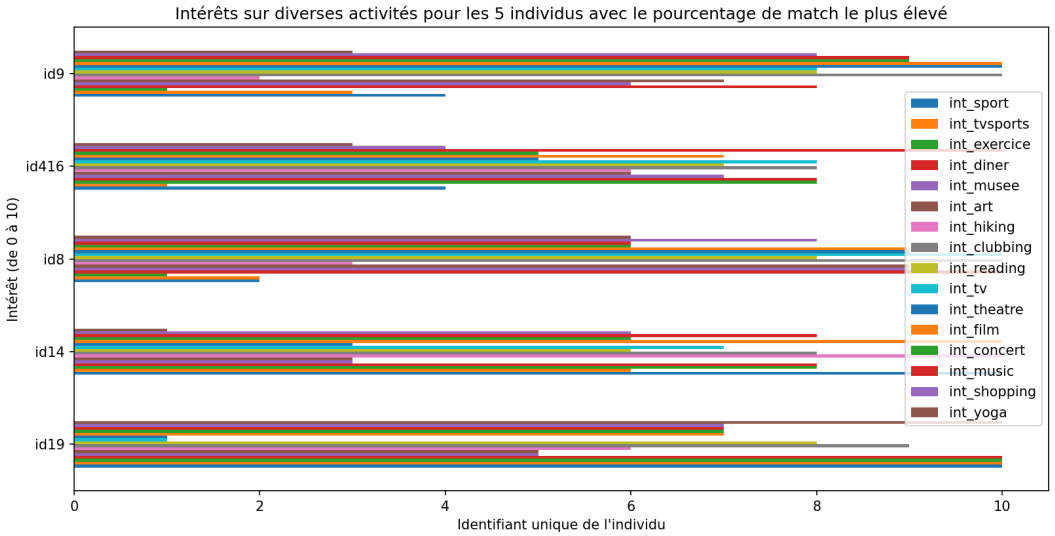

Figure 3 : Intérêt sur diverses activités pour les 5 individus avec le plus de matchs (en %)

De même, ci-dessous la requête appliquée uniquement aux hommes, qui est presque identique à la requête précédente :

```
rqh = "MATCH (i:Individu {genre : '1'}) RETURN i.pourc_match AS
pourc_match, i.int_sport AS int_sport, i.int_tvsports as
int_tvsports, i.int_exercice AS int_exercice, i.int_diner AS
int_diner, i.int_musee AS int_musee, i.int_art AS int_art,
i.int_hiking AS int_hiking, i.int_clubbing AS int_clubbing,
i.int_reading AS int_reading, i.int_tv AS int_tv, i.int_theatre AS
int_theatre, i.int_film AS int_film, i.int_concert AS int_concert,
i.int_music AS int_music, i.int_shopping AS int_shopping,
i.int_yoga AS int_yoga, i.iid AS iid ORDER BY pourc_match DESC
LIMIT 5"
```

MATCH (i:Individu {genre : ‘1’}) : on veut les nœuds Individus avec genre = 1.
La requête pour les femmes est la même, avec genre : ‘0’.

On peut ainsi voir les caractéristiques liées aux activités effectuées par les individus ayant le plus de succès, ainsi qu’au sein des genres. Cela permet de répondre à notre première problématique.

b) Liste d’individus qui pourraient matcher selon des caractéristiques rentrées par l’utilisateur

Dans un second temps l’utilisateur peut rentrer les caractéristiques souhaités pour son partenaire idéal grâce à un input. Nous avons ici sélectionné la ville, le domaine professionnel ainsi que le sexe du partenaire recherché. Le programme renvoie à l’utilisateur les individus correspondant à ces critères.

Ci-dessous la requête :

```
rq3 = "MATCH (i:Individu {domaine:'" + domaine + "', origine:'" +
ville + "', genre:'" + sexe + "'}) RETURN i.iid"
```

On y ajoute {domaine : ‘“ + domaine+ “‘, …} afin que la requête prenne en compte les input entrés par l’utilisateur.

c) Voir des informations sur un individu

Enfin, l'utilisateur peut choisir un individu par un input (celui qui a le plus de match ou ceux qui ont été sélectionnés selon ces critères par exemple) afin de voir davantage de caractéristiques sur les individus qu’il souhaite.
Il peut voir dans quelle ville vit cet individu, avec qui l’individu a matché, qui il a rencontré, par qui il est intéressé, la ville dans lesquelles vivent les participants avec qui il a matché, le type de travail des personnes avec qui il a matché ect. Ceci lui permet de voir quels sont les critères pour bien réussir un match, quel individu a des chances de séduire ect.

Ci dessous une requête sur la villes de l’individu choisi :

```
rq2_1 = "MATCH (i:Individu {iid: '" + votre_individu +
"'})-[:VIT]->(v:Ville) RETURN v.origine"
```

On utilise ici la notion de chemin entre le nœud Individu et le nœud Ville, reliés par la relation VIT.

Ci-dessous une requête sur les personnes qui ont matché avec cet individu :

```
rq2_2 = "MATCH (i:Individu {iid: '" + votre_individu +
"'})-[:MATCH]->(p:Individu) RETURN p.iid"
```

On utilise ici le chemin créé par la relation MATCH entre plusieurs nœuds Individu.

Ci-dessous une requête sur les personnes qui ont rencontré cet individu :

```
rq2_3 = "MATCH (i:Individu {iid: '" + votre_individu +
"'})-[:RENCONTRE]->(p:Individu) RETURN p.iid"
```

On utilise ici le chemin créé par la relation RENCONTRE entre plusieurs nœuds Individu.

Ci-dessous une requête sur les personnes qui ont intéressé cet individu :

```
rq2_4 = "MATCH (i:Individu {iid: '" + votre_individu +
"'})-[:INTERESSE]->(p:Individu) RETURN p.iid"
```

On utilise ici le chemin créé par la relation INTERESSE entre plusieurs nœuds Individu.

Ci-dessous une requête sur les personnes d’un certain domaine afin de voir avec quel autre domaine elles ont matché.

```
rq2_6 = "MATCH (i:Individu {domaine: '" + domaine +
"'})-[:MATCH]->(p:Individu) RETURN p.domaine "
```

On utilise ici le chemin créé par la relation MATCH entre plusieurs nœuds Individu.

VI - Conclusion et limites

Nos requêtes et notre base de données graphique nous ont permis de davantage visualiser les relations entre les différents individus de ce speed dating. Nous avons aussi pu mettre en avant les caractéristiques des individus ayant le pourcentage le plus élevé de matches, et permettre à l’utilisateur d’avoir accès aux informations qu’il désire sur un individu en particulier.

Certaines limites sont à soulignées dans notre projet.
Des requêtes auraient pu être faites sur bien plus de données, il y a en effet beaucoup de données dans la base (7411 rencontres, 518 individus et 195 variables). Nous ne nous sommes pas intéressées à toutes les variables par manque de temps.
Pour améliorer les requêtes et l'utilité de notre travail pour un utilisateur qui cherche son partenaire de vie de rêve, il faudrait ajouter davantage de caractéristiques que l'utilisateur peut choisir.
Nous aurions également pu tenter du machine learning sur les graphes pour tenter de prédire à partir des caractéristiques d'un nouvel individu, avec quels individus déjà présents dans la base il aurait pu matcher. Nous n'avons pas eu assez de temps pour le tester en raison du temps demandé pour comprendre les notions liées aux graphes et s'approprier le code lié à la création de graphe et aux requêtes propre à Neo4J.
Le projet peut donc encore être approfondi. Ceci permettrait une utilisation semblable à ce que l'on retrouve sur les applications de rencontre.