Mini project

Analysis of a Speed Dating Experiment dataset with R, Gephi and Neo4j

colinleverger [at] gmail [dot] com

Colin Leverger - Roskilde Universitet, Informatics
ERASMUS Fall 2016

Report addressed to: Henrik Bulskov Styltsvig

November 24, 2016

Contents

1 Introduction

Since a few semesters; it is possible in Roskilde Universitet to learn about the *Big Data* Field, going from *Data Mining* to *Business Intelligence*. This BIBA module gives a good overview of the tools and theories behind these complex concepts. Students have the chance to study under both business and technical perspectives.

During the lectures, we became aware of the huge number of possibilities linked to *Big Data*, but also of the dangers that can be encountered: biased analysis, fake numbers… It is important to keep in mind that "the Big Data is a very good tool to gain knowledge about hidden patterns and behaviours, but it is certainly not a silver bullet".

In this context, we had to work on a mini-project, which consisted in analysing a dataset: a real-case study. We had to use at least one of the tools we had gained knowledge on, and to explain our approach. The goal was to link theoretical knowledge to experience, and to relate our work to the core areas of *Big Data*.

This report presents my work for this mini-project. It is divided in 3 parts. The first one explains

my work with *R*, the second one explicits my experiments with *Gephi* and the last one describes my experiments with *Neo4j*.

Note that you can find my project on GitHub at the following addresses:

- https://github.com/ColinLeverger/speed-dating-experiment-r: the git repos itself,
- https://colinleverger.github.io/speed-dating-experiment-r: the knitted "*.Rmd*" document produced at every build.

No code will be displayed on this report; everything can be found on the GitHub repository.

2 Speed Dating Experiment

The dataset used for this paper was gathered by an American Business School, during Speed Dating events from 2002 to 2004. The dates lasted four minutes, and the attendees were asked several questions about their date. They also had to provide a lot of personal data, regarding their lifestyle, their preference, etc. Most of the things they had to answer were awaiting boolean answers (yes/no) or notations on a 1 to 10 scale.

The dataset can be downloaded on the famous data science platform Kaggle. The dedicated page can be found at the following address: https://www.kaggle.com/annavictoria/speed-dating-experiment.

3 The Data

3.1 Summary

The data is composed of **8,378** observations and **195** variables.

It was gathered during **21** waves, a wave being an event. **552** persons in total have participated in the events. There are **277** men and **274** women in the dataset. The variables are very diversified, and concern, for each speed date:

- unique id,
- unique id within the wave,
- gender,
- id of the person met,
- gender of the person met,
- preferences (race preference, religious),
- goals in attending to the events,
- match with the person met,
- …

A Word document is also provided with the data to have a better understanding of the many different variables. They are well described and this doc can therefore be used as a "*Data Key*". Taking a glance at this document might be required to have a good overview of my work here.

3.2 Important variables: resumed Data Key

Here can be found 7 of the most important variables you might need to understand my work.

- *iid*: unique subject number, group(wave id gender)

- *id*: subject number within wave
- *gender*: Female=0, Male=1
- *wave*: wave number
- *partner*: partner's id number the night of event
- *pid*: partner's iid number
- *match*: 1=yes, 0=no

4 R Analysis

Having a good dataset and good tools to analyse is not enough: it is better to have a goal. I have chosen this dataset for two main reasons:

- I was curious about several things concerning people's way of dating—not only to tune my lover's skill, I swear…
- This dataset was also good for a graphs analysis. Indeed, there are connections between people. It can thus be analysed with *Gephi* or *Neo4j*.

Therefore, I questioned myself about things I wanted to investigate.

I focus on two questions on in this section—the others are elaborated in the online html document.

The questions I have selected are the following:

- What are the least desirable attributes in a male partner? Does this differ for female partners?
- Is a date between two people working in the same field more likely to create a match or not?

4.1 Get and clean the data

In data manipulation, and in data science in general, it is well known that if the dataset is messy, the analysis will not be the best; indeed, the GIGO[1] rule says that "if there is some garbage in, there will be some garbage out".

In order to limit the bias, I have cleaned the dataset before my analysis. First, I have loaded the data in *R*, using a *dataframe*. I have then created a *Data Quality Report*, which has been used to have an overview of the quality of the dataset and the overall missing values. I have thirdly taken decisions about the missing values, and their possible imputations.

4.1.1 Data Quality Report

Generating a *DQR* is really useful to gain insights concerning the dataset quality. By using an *R* package called *dataQualityR*, it is possible to create a summary of the missing continuous and categorical values, their quartiles, the maximums, minimums, etc. with only one line of code instead of hours of trials and errors by manipulating the raw dataset straight. This library is saving a lot of work to the data scientist: this preliminary job has to be done, anyway.

The *DQR* is generated using the original dataset, and is then written on the disk on two separated ".*csv*" files: one for the continuous variables and one for the categorical variables. To analyse it, we can simply load these two files on *R* as two *dataframes*.

[1] GIGO stands for "Garbage in, garbage out".

	X	non.missing	missing	missing.percent	unique	mean	min	p1	p5	p
1	iid	8378	0	0.00	551	283.68	1.00	9.00	34.00	
2	id	8377	1	0.01	23	8.96	1.00	1.00	1.00	
3	gender	8378	0	0.00	2	0.50	0.00	0.00	0.00	
4	idg	8378	0	0.00	44	17.33	1.00	1.00	2.00	
5	condtn	8378	0	0.00	2	1.83	1.00	1.00	1.00	
6	wave	8378	0	0.00	21	11.35	1.00	1.00	2.00	

Figure 1: First six continuous variables of the dataset, described in the DQR.

If we take a quick look to figure 1, we can for example notice that there is zero *iid* missing, but there is one *id* missing. Hence we can imagine that the missing value can easily be imputed, in that particular case (mostly because there is only one missing value, and also because the *iid* probably leads to the missing *id*). We also have additional information, concerning the total unique values, the mean of all the values, the first **1%** percentile, etc.

	X	non.missing	missing	missing.percent	unique	mean	min	p1	p5	p
172	fun4_3	2959	5419	64.68	16	14.28	0.00	0.00	6.00	
173	amb4_3	2959	5419	64.68	21	9.21	0.00	0.00	0.00	
174	shar4_3	2959	5419	64.68	17	11.25	0.00	0.00	1.00	

Figure 2: Three continuous variables with a lot of missing values, described in the DQR.

The figure 2 shows variables with a lot more missing values; for example, there are **5,419** missing values for the feature *fun4_3*, approximately **65%**, which is a lot. The question could then be: can we clean those missing values?

	X	n.non.miss	n.miss	n.miss.percent	n.unique	cat_1	freq_1	cat_2	fre
1	field	8315	63	0.75	260	Business	521	MBA	
2	undergra	4914	3464	41.35	242	UC Berkeley	107	Harvard	
3	from	8299	79	0.94	270	New York	522	New Jersey	
4	zipcode	7314	1064	12.70	410	0	355	10,021	

Figure 3: Four categorical values, described in the DQR.

The *DQR* can also be used to detect strange outliers. In the figure 3, we can see that there are **355** *zipcode* equal to **0**; it is not a valid *zipcode* and every zero can for this reason be changed to NA.

4.1.2 Choices about missing values

With a complete analysis of the *DQR*, here are the choices I have made concerning the cleaning of the data:

- Input the only missing *id* using the *iid* of the person,
- Input the ten missing *pid*s using the *partner* and *wave* features,
- Change zeros in *zipcode* to NAs
- Change male and female attributes (male is now “M” and female is “W”).

The dataset is very messy and half of the dataset has more than **20%** missing values. Even worst, **30%** of the dataset has more than **50%** missing values! This is due to a main reason: the participants had to fill a bunch of paper forms concerning their preferences, personal details, etc. The forms were very long, and there probably was some resilience to fill them up entirely—especially when they concerned personal details.

For most of them, the missing values concern personal data; it is consequently impossible to input those. Note that having a lot of missing values is not very problematic in our case, to the extent that no highly critical machine learning nor predictions will be performed.

4.2 Analyse the Data

When the data is clean enough, it is time for the data scientist to analyse the data. This step involves:

- Plotting graphs,
- Manipulating the data to extract value from it,
- Creating new columns, new categories, if necessary,
- Answering questions and gaining knowledge about the business process,
- Making the results of the analysis understandable to exploit them and create business value upon them.

Before answering critical questions, it is common to explore the dataset with simple graphs: gender repartition in waves, age repartition and extrema, careers of the people… The goal is to become familiar with the data, but this step can sometimes be useful to extract in a first hand some interesting patterns.

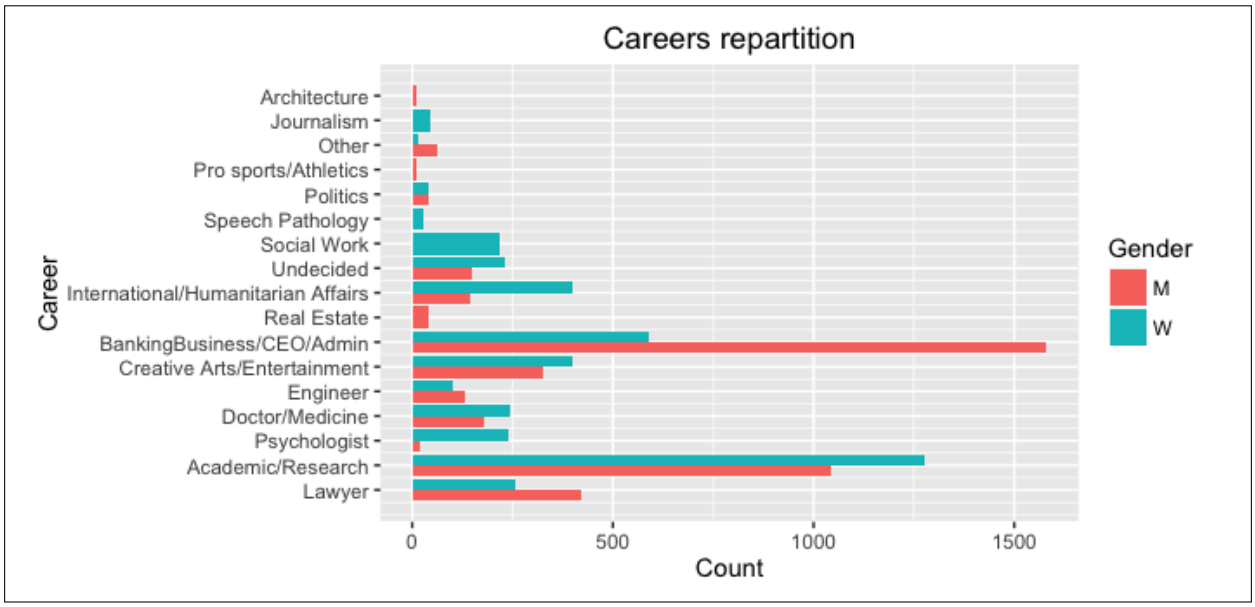

Figure 4: Career repartitions in the dataset.

For example, the figure 4 shows that there are a lot of men working as CEO/Admin, and also a lot of researchers. There are also a lot of women researchers. This observation could be useful to make a correlation between behaviours and carriers, for example.

In the figure 5, we can see that most of the people of this experiment are outgoing persons. This clue is really interesting, and we will use it afterwards to explain some observations made in section 4.3.

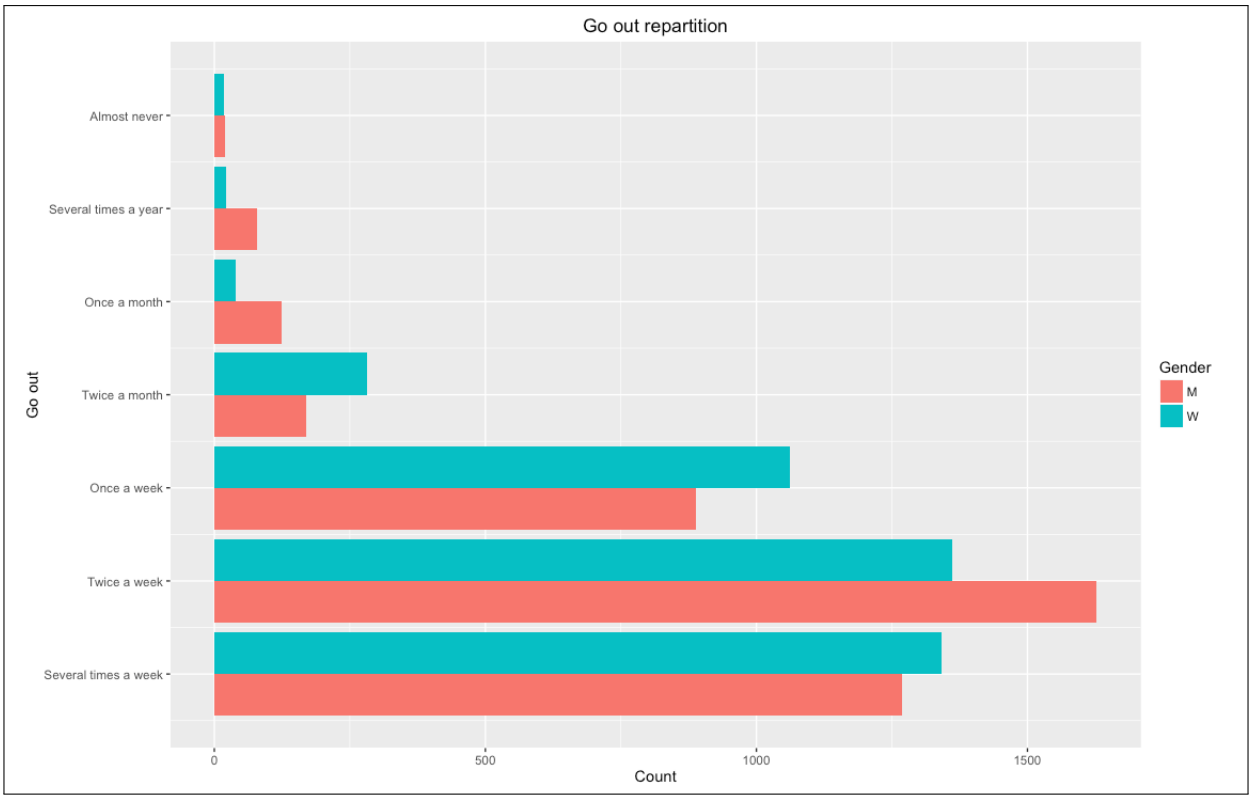

Figure 5: Go out repartition in the dataset.

4.3 Best and Worst Attributes to Have

How to get more matches during those events? What activities make a woman or a man more willing to please his/her date? To answer those questions, we are now going to analyse the good features and attributes to have.

For a quick recall, the participants had to note on a 1 to 10 scale their preferences concerning different activities, such as *yoga*, *tv*, *clubbing*, etc. Thus, we will see what activities are preferred by men/women in this dataset.

4.3.1 General Process

To find out what are the best attributes to have, we need to create and feed a model. The goal is to figure out which features are the most informative ones; basically, the process is:

1. Isolate from the original dataset:
   - The features we want to benchmark: the preferences and activities of a participant, such as *yoga*, *tv*, *clubbing* ...,
   - The target feature: the one that will be influenced by the preferences: *match*.
2. Feed a random forest model with the data, not forgetting to configure the output column: *match*,
3. Exploit the output of the model—we are especially interested in the *importance* table provided with the output,
4. Plot, graph, and conclude on the good features to have.

Indeed, a mathematical model usually measure the importance of variables; as a result, it is possible to predict the outcome/target feature knowing the general variable values of an unknown

subject. Here, the target feature is obviously *match*.

In other words, if a new case with a unique combination of preferences (*tv*, *clubbing*, ...) shows up in the process, it could be possible to say if it will result in a match or not. It is definitely the basis of any machine learning scripts.

Note that to have a proper analysis, it is always good to do an evaluation of the output afterwards. Without a proper evaluation, numbers and predictions are just meaningless and cannot consequently be used to gain insights. Because of a time issue, there will be no evaluation here.

4.3.2 For men

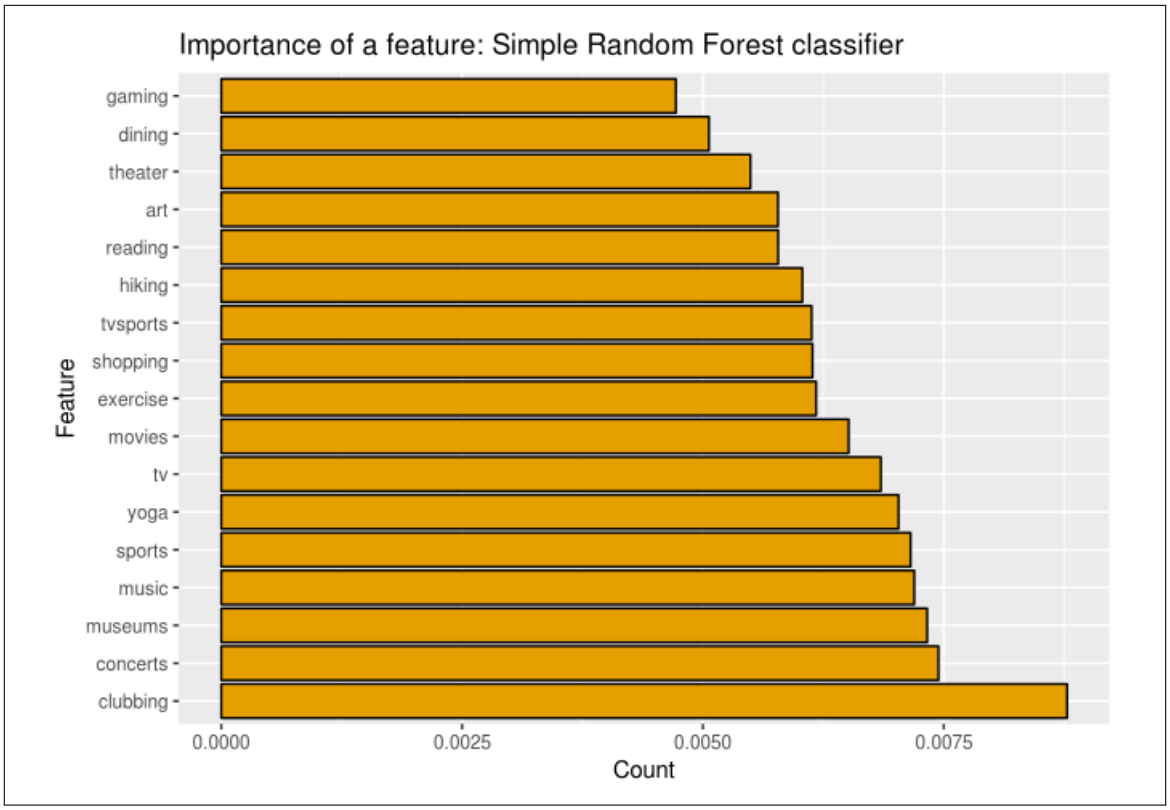

Figure 6: Overall importance of a feature for men.

If we take a closer look at the figure 6, we can see that *gaming* is definitely not a very good interest for a man to have... As opposition to *clubbing*. We can also notice that between *art*, *exercise* and *tv*, there is no big difference and the distribution is nearly flat.

This graph confirms the outgoingness of the people in the dataset, thing that we have already noticed above.

The figure 7 is also using a random forest model; but this time, there is a filter which selects every man with more than 5 matches. It is interesting to see that the interests to have are slightly different than before. Liking *theater* and *hiking* seem to be good assets in a speed dating; liking *sports* or *music* seem not.

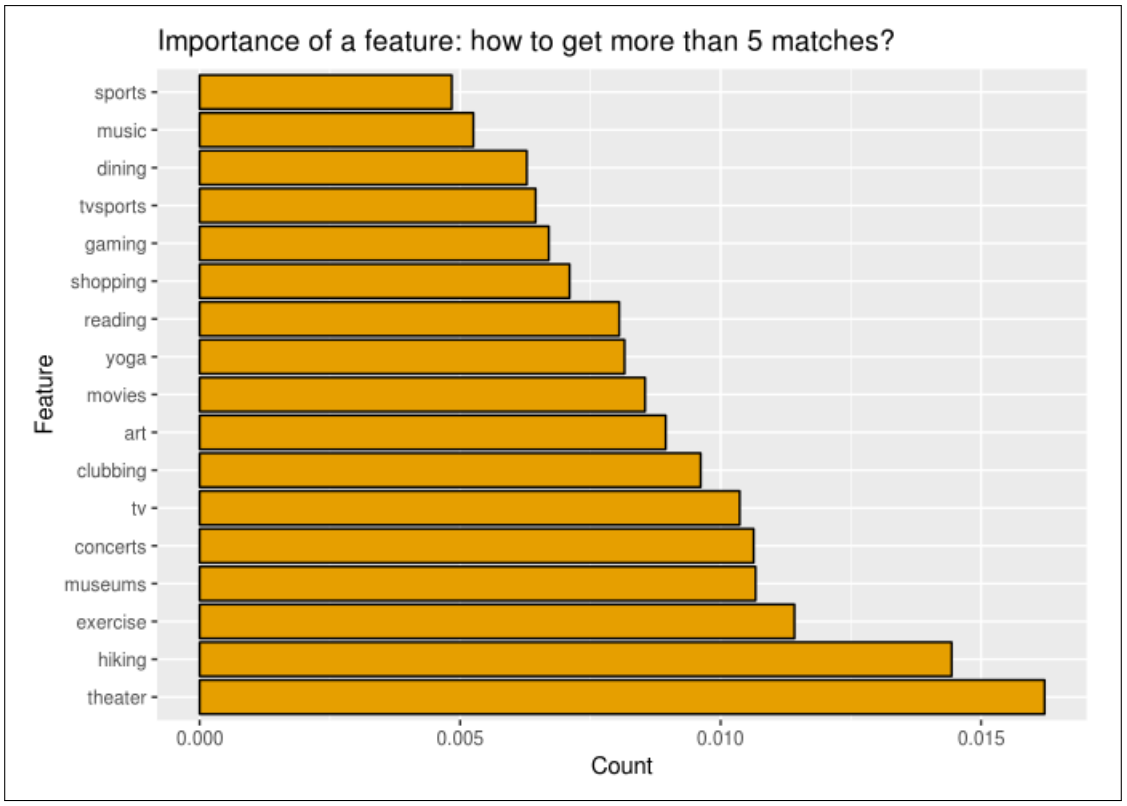

Figure 7: Importance of a feature for men: how to get more than 5 matches?

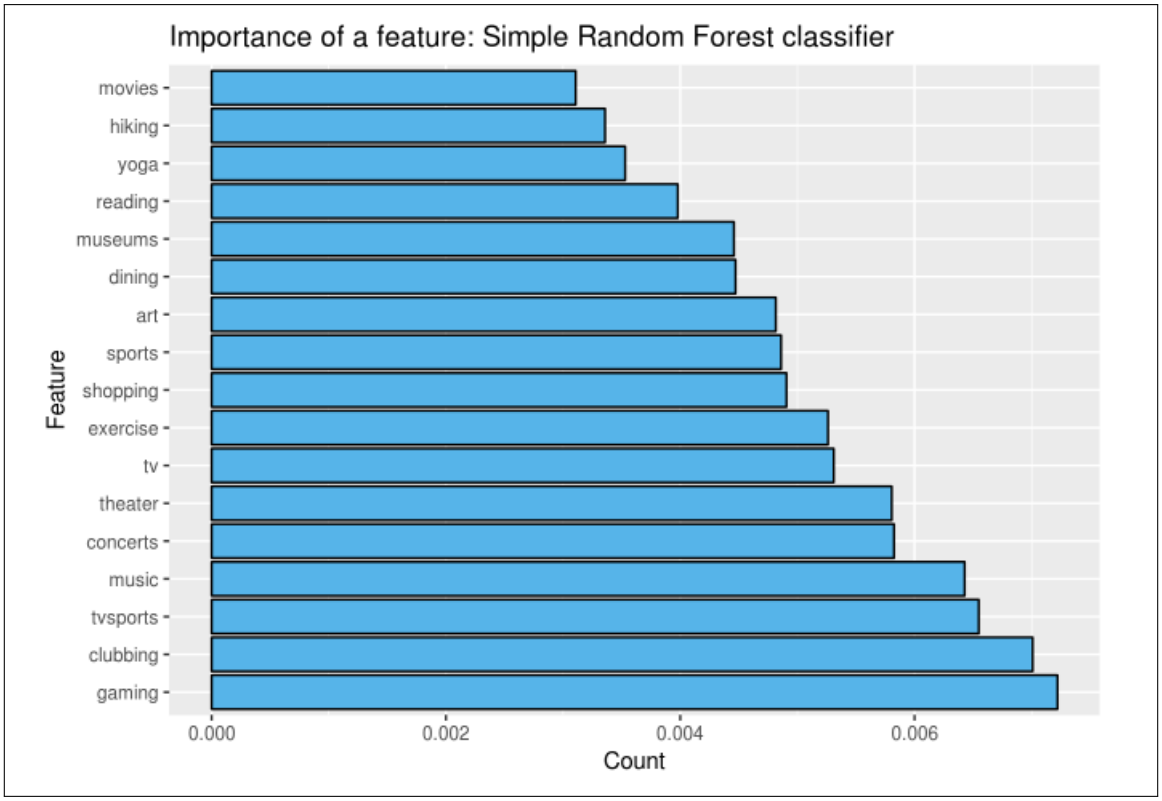

Figure 8: Overall importance of a feature for women.

4.3.3 For women

As we can see in figure 8, a woman who likes *gaming* or *clubbing* is more likely to have a match than the one who likes *movies* or *hiking*.

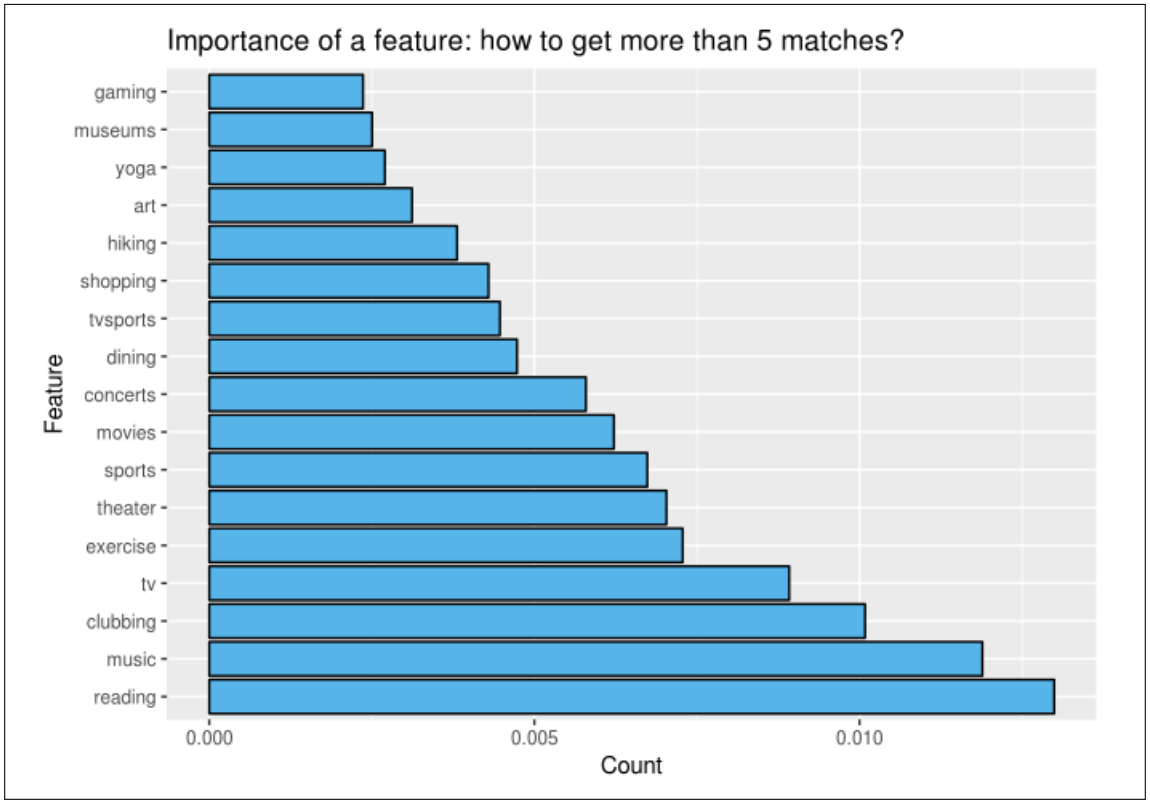

Figure 9: Importance of a feature for women: how to get more than 5 matches?

The figure 9 shows that for a woman to have more than 5 matches, it is good for her if she reads books while listening to music, preferably in a nightclub—if everything is put together, who knows, it might be even more efficient.

4.3.4 Conclusion on the best feature to have

I decided to check my results in feeding another type of model, known as the Extra-tree random forest classifier. The purpose was to check the tendencies and insure that there were not two totally different outputs. This simplistic test confirmed the first model results[2].

We can also see that there are differences between good attributes to have if you are a man or a woman. In other words, men and women are not exactly looking for the same things. But because the people in this dataset are really outgoing persons, we see that liking *clubbing* is still a valuable attribute anyway.

4.4 Dates and Field of Carrier

I was curious to know if people working in the same field were more compatible in a speed-dating event. Unfortunately, the field of the dates was not present in the dataset. I then had to create a new column to do the analysis: see section **4.12 of the code** for detailed script.

[2]Check the code to see the small variations between the two models fed.

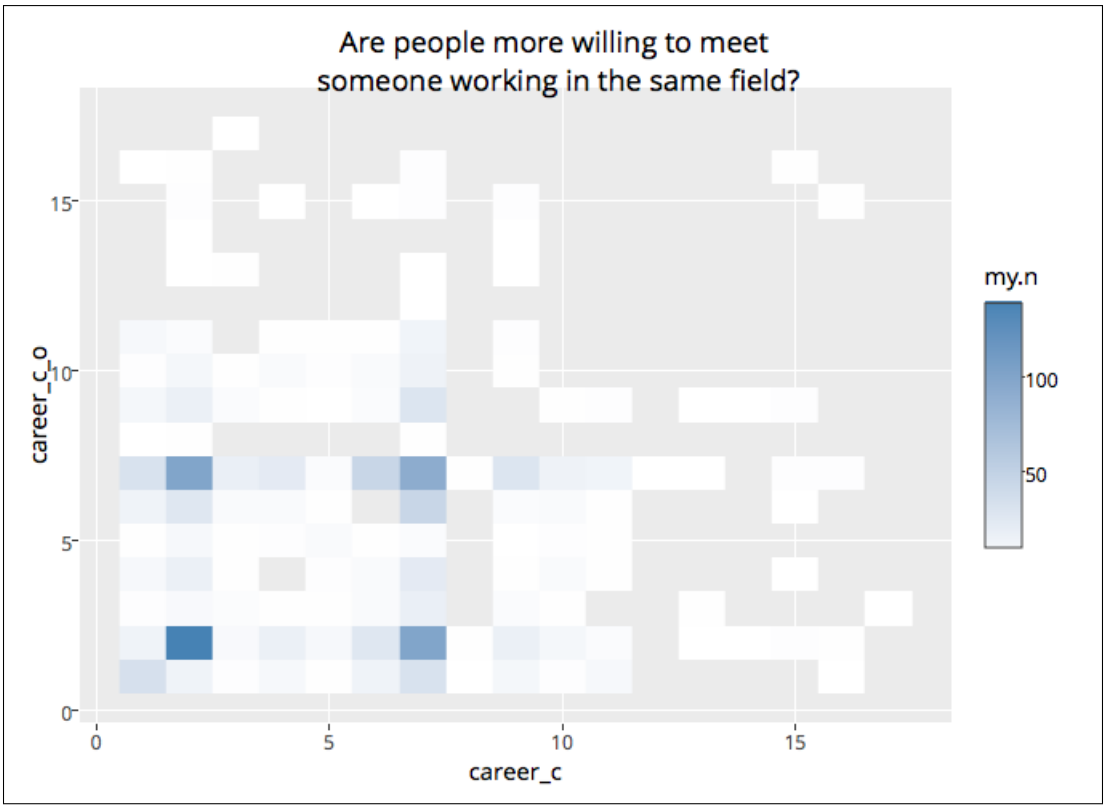

Figure 10: Are people more willing to meet someone working in the same field?

The figure 10 is a heat map of the *matches* versus *career* in this dataset. In the "x" axis are the coded careers of the persons, and in "y" axis the coded careers of the dates; the more there are matches between similar careers, the darker will be the blue colour of the square. We see that there are relatively more matches for the coded careers **2** and **7** (which are academic/Research and CEO/Banking), but this could mainly be due to the fact that there are more people in these categories (ref. figure 4). No interesting hidden behaviours here!

4.5 Create Data to Feed Gephi and Neo4j

To display interesting graphs in *Gephi*, we need to extract some columns from the original dataset and create a new lightened one. The original dataset is very large and we only need a couple of columns.

The first features we need are *iid* and *pid*; they will represent connection between nodes (people). It could also be wise to include the *match* feature, because it will help us a lot in analysing the interactions between nodes—as a matter of fact, it will be used as a label for the edges.

For *Neo4j*, it has been chosen to divide the dataset in two parts: one concerning the individuals (their personal information, ids, etc.) and one concerning the dates (the *wave* numbers, the issue of the date/*matches*, etc.). We will then link these two datasets on *Neo4j* using the *Cypher* language.

5 Gephi Analysis

Once the data has been created with *R*, a simple csv importation is possible to use *Gephi*. The graph which will be created is a *unimodal* graph, as the nodes are all of the same type: persons. To display the nodes with an interesting layout, several steps have been followed:

- Apply a Fruchterman Reingold layout[3] with an *Area* of 10,000.0,
- Customise the size of the *Nodes* using the *Ranking* tab with *In-Degree* (*Min size*: **10**, *Max size*: **100**)
- Colour each node with the *In-Degree Ranking*.

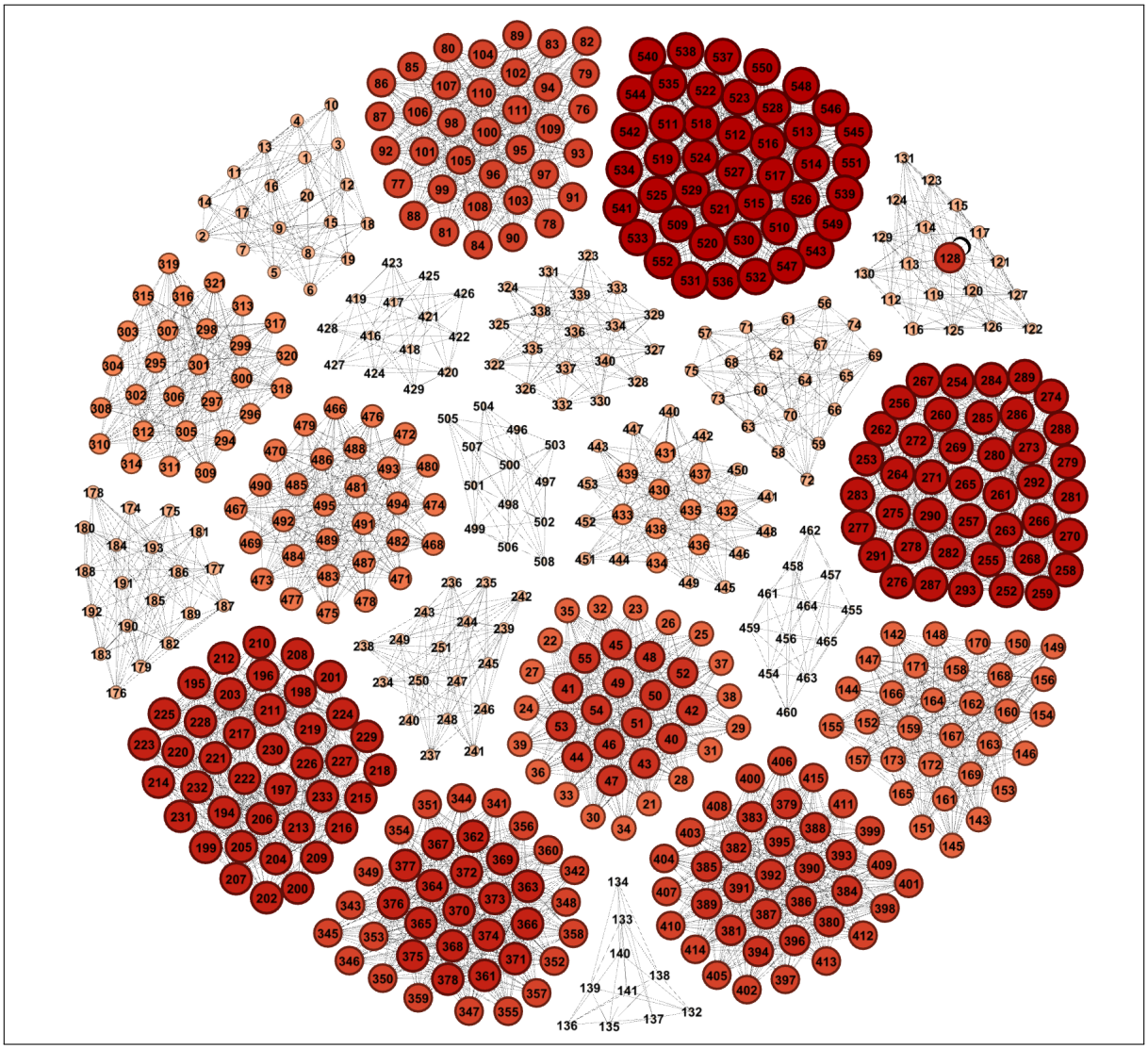

Figure 11: Graph visualisation using Gephi after several initialisation steps.

A first analysis of the figure 11 shows several interesting clues:

- There is no connection between waves.[4] The clusterisation is thus very clean.
- There is a self-loop with the individual **128**; he met himself during the event. Two explanations to this: either there is a problem in the dataset, or this person is schizophrenic…

[3] See: https://github.com/gephi/gephi/wiki/Fruchterman-Reingold

[4] Those type of connection could have been called *Bridges*.

– The size and colour of the nodes are really even; it suggests that in most of the waves, every person has met everyone!

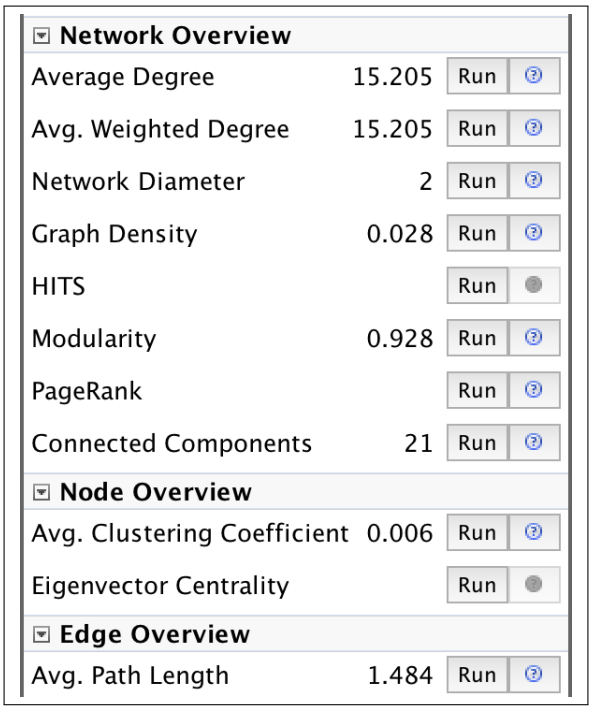

Figure 12: Statistics of the network, using Gephi.

To gain more knowledge about the graph, the *Gephi's Statistics* module is interesting. The insights which can be gained from the results of computation (see figure 12) are:

– The average degree is pretty high, which can be explained by the fact that everyone meets everyone in most of the waves.
– The graph density is very low; it is due to the absence of bridges.
– There are **21** connected components, for **21** waves.
– The modularity is close to one, which means that the separation between clusters is very clear.

In the figure 13, I have coloured the nodes with a *Degree Partition*. It is noticeable that in most of the waves, the degree of the participants is the same. Again, it confirms that in most of the waves, everyone has met everyone.

If we want to go further in our investigations, it could be useful to use some filters. I have decided to focus on the nodes with a match. To even narrow down the results, I have added another filter to only get the nodes with an *In Degree* equal to **20** (twenty connections to those persons). I have finally applied a *Force Atlas* layout for a better visualisation. The result of this nested filters can be seen on the figure 14.

The figure 14 shows that some nodes have no connections between them: it means that they are probably from another wave and thus have no relations with the central cluster considered.

6 Neo4j analysis

Neo4j is another software which can be used to analyse graphs. This freeware is providing a web interface to create and interact with a graph database. It is possible to query it with the *Cypher* language, which will also be used to import the csv raw data of this dataset.

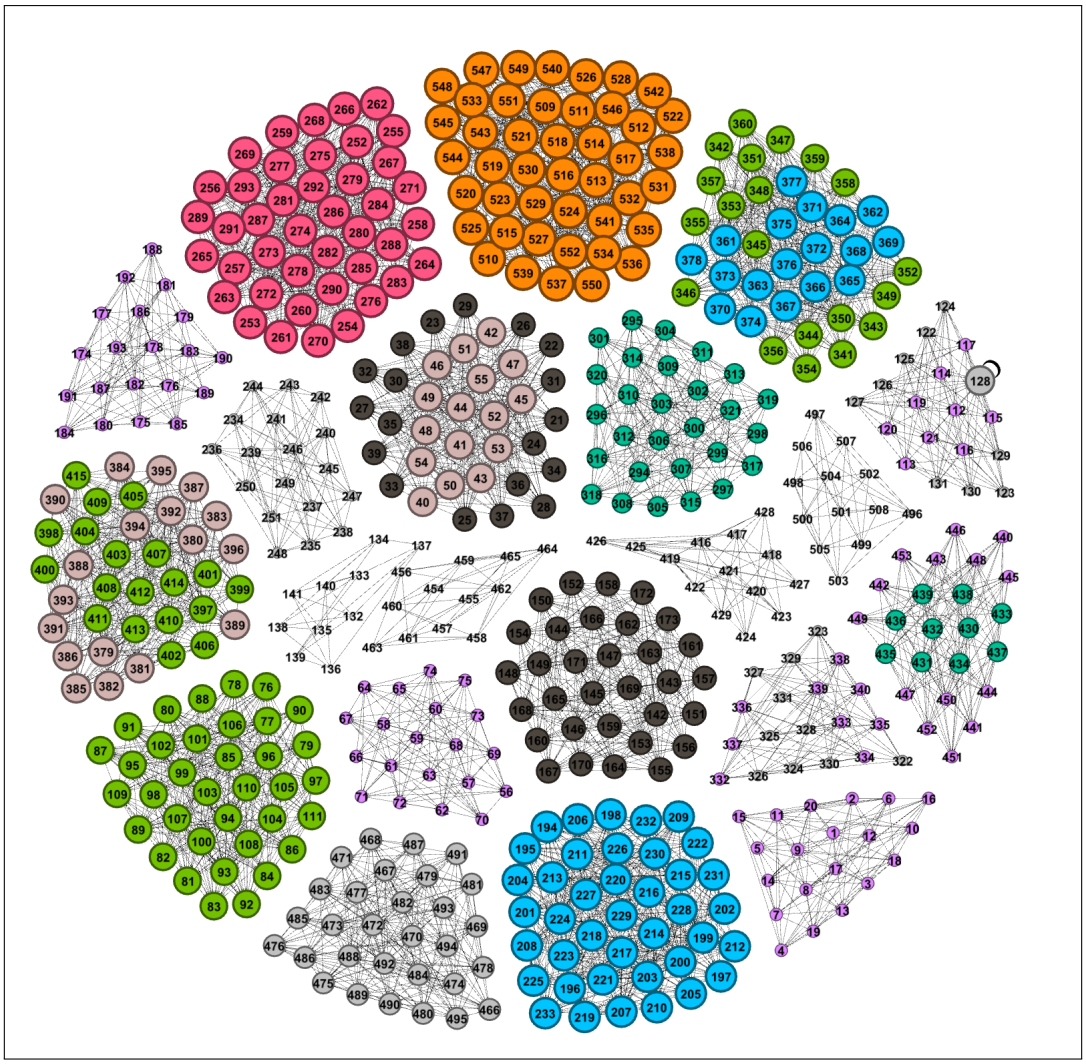

Figure 13: Network coloured with Degree, using Gephi.

The procedure followed to create and exploit the graph database is:

- Split our csv file in two at the end of our *R* script, in order to distinguish the *persons* and the *events* (a *person* being somebody with an *iid*, an *event* being somebody meeting somebody else),
- Think about the possible relations between people/dates/places, for example:
  - a **Person :LIVES** in a **City**,
  - a **Person :MET** other **Person**.
- Using the *Cypher* language
  - Import the data in the graph database,
  - Create the relations between people (**:MET**, **:LIVES**)
  - Query the graph to gain insights.

I have loaded data, created the database and experimented with a lot of queries; the latter won't be displayed here.[5]

Displaying **20** nodes with only the **:MET** relation can produce a messy result. If we take a look at the figure 15, no knowledge can be easily extracted from that graph. That is why I am focusing

[5]See: https://github.com/ColinLeverger/speed-dating-experiment-r/blob/master/cypher.cyp for more details about my *Cypher* script.

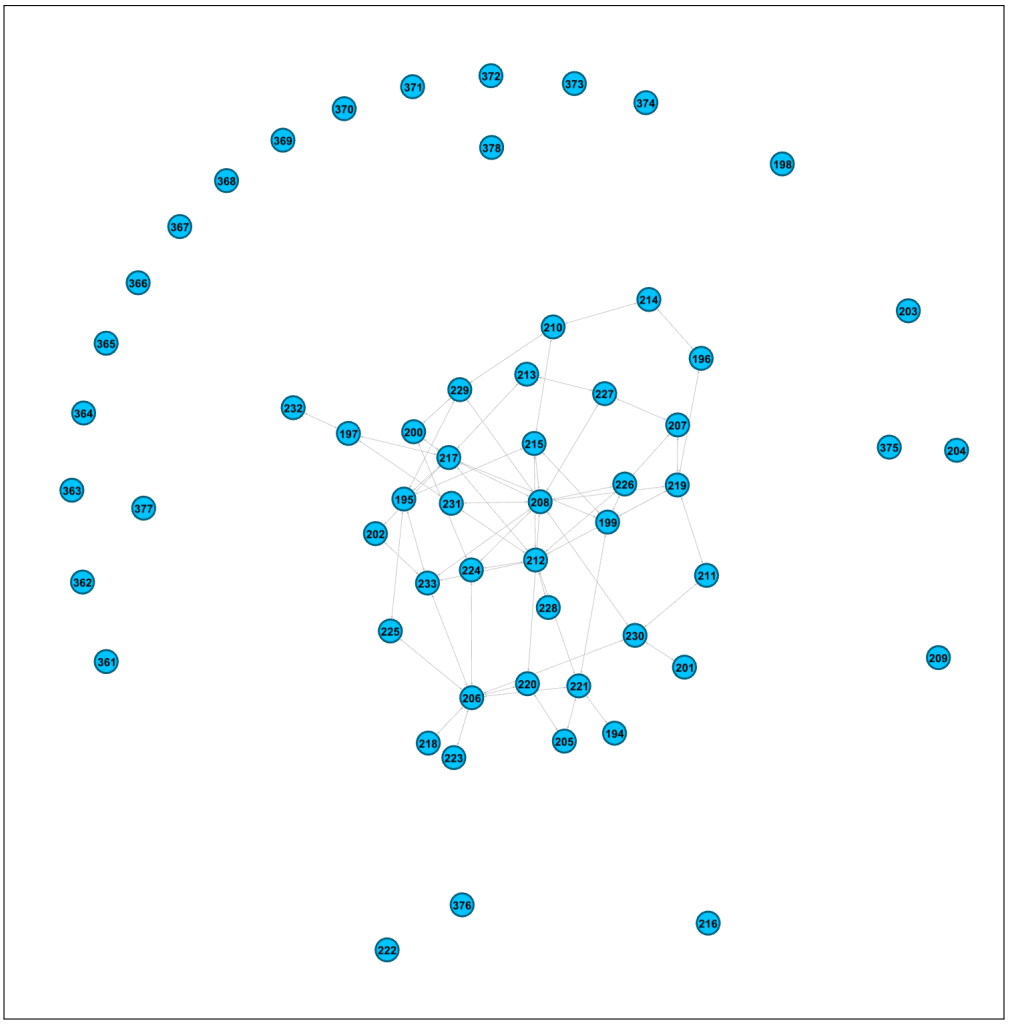

Figure 14: Filters on a network, using Gephi.

on two precise questions on this part.

6.1 Analysis of the People With the More Matches

Once my graph database ready, I was curious about the ten people with the more matches. I have decided to write a query for that.

In the figure 16, we can see that the person with the *iid* **524** is the winner of the match game: he got in total fourteen matches. Let take a closer look to the node in the figure 17: the attributes of this person can be observed here (*age* **25**, etc). It is also possible to double-click on the node to expand the neighbours/relations linked to it, and to have a sharper analysis of its environment.

The very same procedure can be followed to analyse the people with the less match (customising the match filter on the *Cypher* script should be enough).

6.2 Analysis of the cities with the more people living in

The figure 18 displays the ten cities with the more people living in. We can see that there is now some bridges between waves, which is something we never had before: the *Gephi* analysis showed clearly that there was no connection at all between the twenty-one waves.

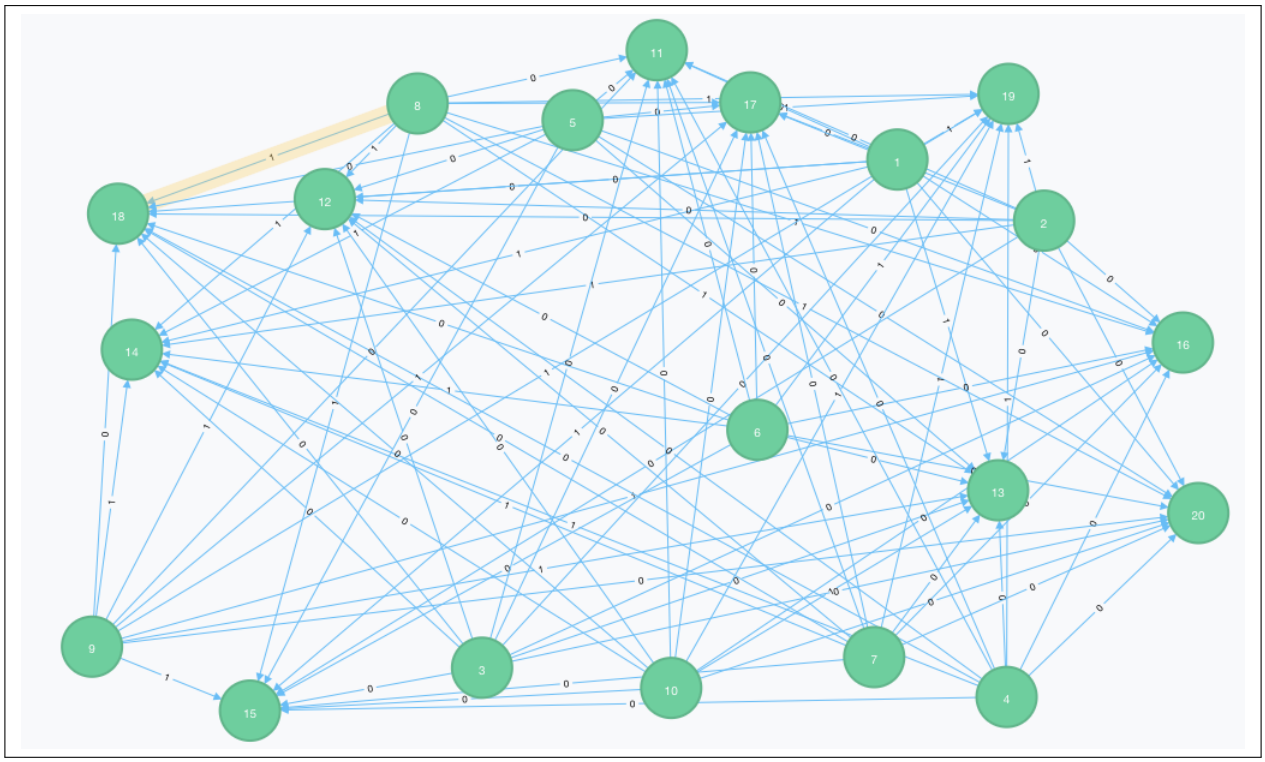

Figure 15: 20 random nodes with the **:MET** relation, using Neo4j.

```
$ MATCH (p:Person)-[:MET {match: '1'}]->(p2:Person) WITH p2, count(p) as degree RETURN p2.iid, degree ORDER BY degree DESC LIMIT 10
```

Rows | Text | Code

p2.iid	degree
524	14
107	11
208	11
366	10
268	10
404	9
212	9
19	9
489	8
99	8

Figure 16: Ten people with more matches, with the Cypher query.

Again, having a real exploration goal and a real problematic is really useful to extract real knowledge. I here had no other goal than learning how to use *Neo4j* and I did not extract any very interesting clues from this analysis. Moreover it is complicated to export the results of the analysis in an image, because *Neo4j* is a very dynamic software—a lot of non-static *JavaScript* graphs are produced on the fly.

7 Conclusion

Finding a dataset which was compatible with every software we studied during the BIBA module was for me a priority. I really wanted to have an overview of a real and complete case. Manipulating

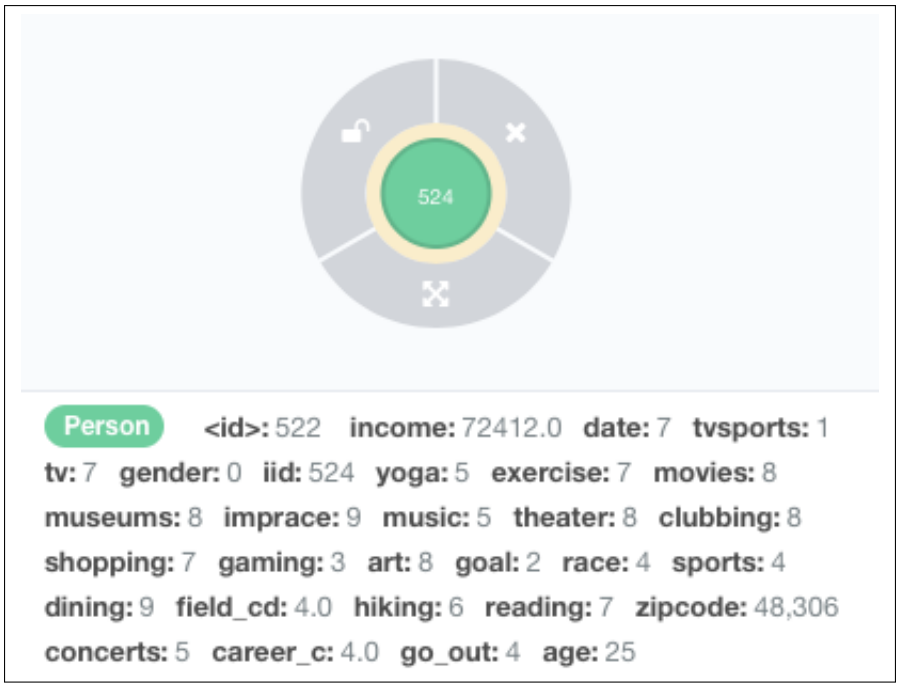

Figure 17: Individual n°**524**'s node.

data both with *R* and *Gephi/Neo4j* is really useful because having several directions to investigate gives a better view of the problems. The Speed Dating dataset was then an interesting choice, and I enjoyed analysing it.

I have spent a lot of time working with *R* and less time with graphs and networks. Indeed, making choices about the cleaning of the data and plotting graphs are time consuming, and I eventually ran out of time. But I am pretty happy with the graphs created with *Gephi*, and with the discovering of the *Neo4j Cipher* language and csv importation.

I think that graph databases and visualisations are absolutely critical in our world. Every single thing and object will soon be connected, and having knowledge about the main concepts and theories behind graphs is definitely a good asset for my curriculum. I unfortunately did not have enough time to experiment with *Hadoop*, but I will probably try to duplicate the dataset to distribute it over several nodes one day.

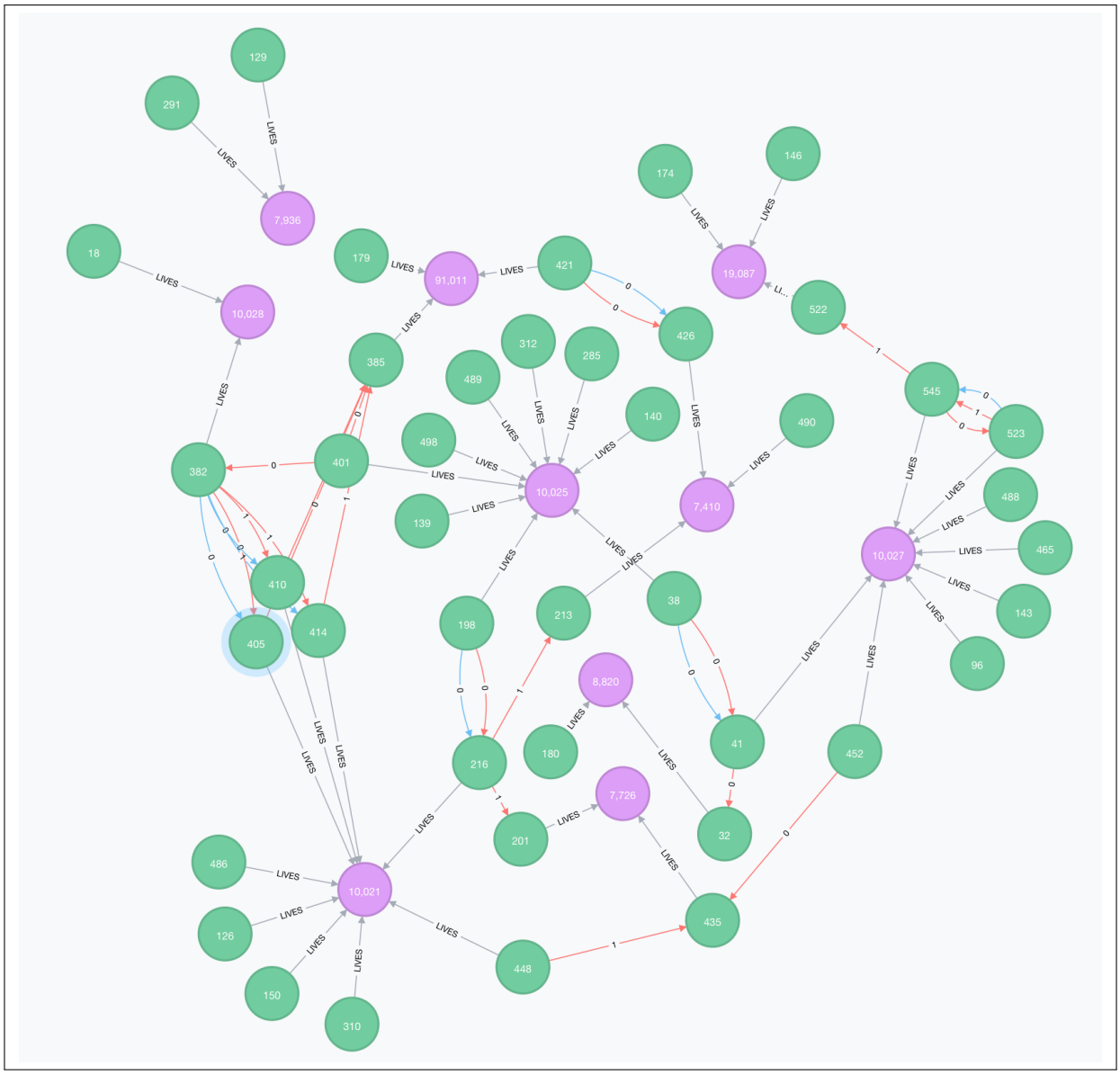

Figure 18: 10 cities with the most people living in, with Neo4j.

List of Figures