Speed Dating

Brief analysis of a speed dating event

Alessandro Cogollo, 10571078
Systems and Methods For Big and Unstructured Data - Politecnico di Milano

1. Introduction

This paper describes a brief analysis conducted over a dataset about a set of speed dating experiments; the analysis has been conducted for educational purposes, as the reported queries represent only an overview of the potential investigations that can be conducted on this dataset. However, the dataset is of interest, since, as will be shown in the subsequent sections of this paper, it is possible to analyze numerous aspects related to the perception, interests, and expectations of the participants, also based on aspects studied by social sciences. With a more in-depth analysis, more numerous datasets, and techniques that go beyond the topics of this course it would be possible and interesting to deepen the research using predictive and inference algorithms to predict possible matches, as well as identify patterns regarding the preferences of any further participants.

The chosen dataset collects 1,012,321 entries, and has been downloaded via Kaggle, at the following link. More specifically, it collects experimental data relating to speed dating events held between 2002 and 2004. The duration of the single date was set at four minutes, during which each participant had the opportunity to get to know and look for a possible "match" with (mostly) strangers of the opposite sex. The dataset contains different types of information, from information regarding the participant's lifestyle to interests and hobbies; more precise information on the structure of the dataset will be explained in the following sections.

The technology chosen is elasticsearch, mainly for two reasons; the first concerns the possibility of using it as a "suggester" of results, thanks to the possibility of "boosting" the results, obtaining answers more suited to specific preferences, even though, as explained later, the dataset didn't contain lot of text to be inspected with an analyser. The second concerns the possibility of exploiting Kibana to display interesting insights relating to the queries performed. Last but not least, Elasticsearch has been chosen due to a personal interest in exploring this technology.

2. Data Wrangling

The chosen dataset already had very high quality and usability, measured in terms of completeness, credibility, and compatibility (both usage license and format), therefore the data wrangling process required was minimal.

A first inspection of the dataset was performed using the Pandas and NumPy libraries, to evaluate the possible need to normalize the data. The inspection result immediately showed the presence of “b” characters at the beginning of some fields (`has_null`, `gender`, `match`, etc); although the dataset can also be used in these conditions, for reasons of consistency and quality of the dataset the interested columns have been cleaned with the python function shown below.

```
1 usage
def substr_b(x):
    return x[2:(-1)]

1 usage
def syntaxcorr(df):
    for col in df:
        if isinstance(df[col][0], str) and (df[col][0])[0:2] == "b'" and (df[col][0])[-1] == "'":
            df[col] = df[col].apply(lambda x: substr_b(x))
```

Further analysis involved exploring the statistics of the dataset, from which it was clear that the fields: `attractive_o` and `fun_o` must have been normalized in the range 0 to 10, as well as the fields `gaming`, and `reading`.

```
1 usage
def norm_met(df):
    df['met'] = df['met'].apply(lambda x: set_one_zero(x))

1 usage
def mm_scaler(df, field, ran):
    avg = df[field].mean()
    df[field].fillna(value=avg)
    scaler = MinMaxScaler()
    scaler.fit(np.array(df[field]).reshape(-1, 1))
    df[field] = scaler.transform(np.array(df[field]).reshape(-1, 1))
    df[field] = df[field] * ran
    df[field] = round(df[field], 0).astype('float32')
```

The “`met`” field should have been 0 or 1 (Boolean), but was between 0 and 8, thus it has been transformed using the following functions.

```
1 usage
def set_one_zero(x):
    y = 0
    if x > 0:
        y = 1
    return y


1 usage
def norm_met(df):
    df['met'] = df['met'].apply(lambda x: set_one_zero(x))
```

The first letter of each word in the “`field`” column has been transformed into all caps for consistency, even though term queries are case insensitive by default.

```
1 usage
def to_title(st):
    return st.title()


1 usage
def uniform_uppercase(df, field):
    df[field] = df[field].apply(lambda x: to_title(x))
```

Finally, derived values have been recomputed to be coherent with new values. The final dataset used has been attached to the delivery folder (dump subfolder) and named “`speeddating-v1`”; python code used for the data wrangling process has been attached as well.

3. Dataset

As mentioned in previous sections, the dataset contains 1,012,321 entries (excluding NULL values), in particular, the original CSV file is composed of 123 columns and 8378 rows.

The dataset gathers different information about the experiment, in particular, each entry contains information about a couple in a specific “wave” (round). For each entry, **demographic information** is collected, such as age, gender, "race", and field of study of both participants. Additional information covers six attributes: **attractiveness**, **sincerity**, **intelligence**, **fun**, **ambition**, and **shared interests**. The dataset also includes other questionnaire data gathered from participants at different points in the process. These fields include **dating habits**, **self-perception** across key attributes, **beliefs on what others find valuable in a mate** and **lifestyle information**. Finally, the dataset contains information regarding the result of the match, the interest in dating, etc. For all relevant fields, derived information was calculated, mainly related to clusters to which certain values can be mapped to, for example about interests etc.

The dataset has been distributed on two shards, since its dimensions are quite small, and for this analysis performed there wasn’t a strong requirement on speed.
Regarding the implementation, the mapping is quite standard, but modified with respect to the version suggested by the elasticsearch ingestion tool, as most of the numerical fields have been casted to integer values; the only values that have been left as double (float64) are preferences, and correlation fields. String values have been set as “keywords”, as most of them are about clusters (`d_funny_important`, `d_intelligence_partner`, etc), therefore need to be searchable exactly for their specific value. The only exception is made by the attribute `field` which needs to be searchable for its content, since entries have not been homogenized. Some attributes, such as met, or match, could have been boolean, but would have required further data wrangling and would have brought low advantages.

Below can be found a very small portion of the mapping, the rest will be attached in the delivery folder under the name of “`mapping.txt`”.

```
{
  "mappings": {
    "_meta": {
      "created_by": "file-data-visualizer"
    },
    "properties": {
      "age": {
        "type": "integer"
      },
      "age_o": {
        "type": "integer"
      },
      "ambition": {
        "type": "integer"
      },
      "ambition_partner": {
        "type": "integer"
```

4. Queries

The chosen queries represent a set of interesting requests both from an educational point of view and from social sciences point of view, with the aim of obtaining an insight into the preferences, demographics, and tendencies of the participants in the experiment. The queries were chosen with different degrees of complexity, and are shown below. All the results of the queries have been attached in the delivery folder.

a. Highest difference in age of a match

The query below returns first filter aggregates all entries for which the field "`match`" is set to 1 (TRUE) and returns the maximum value (`diff_age`) of these aggregates. The result is returned with the last line of the query and shown below.

```
GET /speeddating-v4/_search
{
  "size" : "1",
  "aggs": {
    "more-diff-match" : {
      "filter" : {
        "term": {"match": "1"}
      },
    "aggs" : {
      "diff_age": { "max": { "field": "d_age" } }
    }
  }},
  "_source": ["diff_age"]
}
```

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 8378,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "speeddating-v4",
        "_id": "bFq81IwBkPnrhWx6FobH",
        "_score": 1,
        "_source": {}
      }
    ]
  },
  "aggregations": {
    "more-diff-match": {
      "doc_count": 1380,
      "diff_age": {
        "value": 34
      }
    }
  }
}
```

The query could have been created also by using the range operator, as follows.

```
GET /speeddating-v4/_search
{
  "size" : "1",
  "aggs": {
    "more-diff-match" : {
      "filter" : {
        "range": {"match" :{"gt": "0"}}
      },
    "aggs" : {
      "diff_age": { "max": { "field": "d_age" } }
    }
  }
  },
  "_source": ["diff_age"]
}
```

b. Oldest participant to the speed dating experiment

This simple query returns the single element with the highest age in the dataset.

```
GET /speeddating-v4/_search
{
  "size": 1,
  "sort": [ {
  "age": {"order": "desc"}
  } ]
}
```


```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 8378,
      "relation": "eq"
    },
    "max_score": null,
    "hits": [
      {
        "_index": "speeddating-v4",
        "_id": "Clq81IwBkPnrhWx6N5bf",
        "_score": null,
        "_source": {
          "d_funny_important": "[21-100]",
          "d_reading": "[6-8]",
          "ambtition_important": 0,
          "d_age": 22,
          "intellicence_important": 25,
          "movies": 3,
          "music": 8,
          "d_tv": "[0-5]",
          "d_expected_num_interested_in_me": "[0-3]",
          "samerace": 0,
          "shopping": 1,
          "d_tvsports": "[0-5]",
          "d_attractive": "[6-8]",
          "d_ambition": "[0-5]",
          "dining": 8,
          "d_shared_interests_o": "[6-8]",
          "intelligence": 7,
          "d_funny": "[6-8]",
          "d_ambtition_important": "[0-15]",
          "d_sincere": "[6-8]",
          "field": "Soa -- Writing",
```

c. Correlation between races and matches?

This query has been formulated to try to understand if there's any correlation between races and matches. First an aggregation on different races is performed, then the number of matches per aggregate is counted.

```
GET /speeddating-v4/_search
{
  "size" : 0,
  "aggs" : {
    "race-count" : {
      "terms" : {"field" : "race"},
      "aggs" : {
        "num-matches" : {
          "terms" : {"field":"match"}
        }
        }
    }
  }
}
```


```
{
  "took": 0,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 8378,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "race-count": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "key": "European/Caucasian-American",
          "doc_count": 4727,
          "num-matches": {
            "doc_count_error_upper_bound": 0,
            "sum_other_doc_count": 0,
            "buckets": [
              {
                "key": 0,
                "doc_count": 3939
              },
              {
                "key": 1,
                "doc_count": 788
              }
            ]
          }
        },
        {
          "key": "Asian/Pacific Islander/Asian-American",
          "doc_count": 1982,
          "num-matches": {
            "doc_count_error_upper_bound": 0,
            "sum_other_doc_count": 0,
            "buckets": [
              {
```

Results shows the following numbers:

Race	Total	No Match	Match	Percentage
European/Caucasian-American	4727	3939	788	16.67%
Asian/Pacific Islander/Asian-American	1982	1715	267	13.47%
Latino/Hispanic American	664	541	123	18.52%
Black/African American	420	335	85	20.24%
Unknown	63	49	14	22.22%
Other	522	419	103	19.73%

d. Matches expectancies between men?

This query tries to understand what the expectancy in term of number of matches for the men participating in the experiment:

```
GET /speeddating-v4/_search
{
  "aggs": {
    "male_perception": {
      "filter": { "term": { "gender": "male" } },
      "aggs": {
        "avg_expected_matches": { "avg": { "field": "expected_num_matches" } }
      }
    }
  },
  "size" : "0"
}
```

The average is: 3.39, which can be rounded to 3. Executing the same query for the women, the result is slightly lower, and is 3.02, which can still be rounded to 3.

e. How much attractive people consider attractiveness important when looking for an attractive partner?

Find out the importance that attractive people are giving to finding an attractive partner.

```
GET /speeddating-v4/_search
{
  "query": {
    "range": {
      "attractive_o" : {"gt": 8}
    }
  },
  "aggs" : {
    "avg-attr" : {
      "avg" : {
        "field" : "attractive_important"
      }
    }
  },
  "size" : 0
}
```

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 869,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "avg-attr": {
      "value": 23.490058479532163
    }
  }
```

f. Return the IDs and field of entries about men, better if in the field of any science and with an age between 25 and 35

Find out the importance that attractive people are giving to finding an attractive partner.

```
GET /speeddating-v4/_search
{
  "query":
    { "bool":
      { "must": [
          { "match" : { "gender" : "male" } } ],
        "should": [
          { "match" : { "field" : "Science" } },
          { "range": { "age" : { "gt": 18, "lt" : 25 } } } ] }
  }, "_source" : ["_id", "field"]
}
```

g. Return the average value of all the possible interests, chosen from those participants that matched

This query is interesting as it permits to determine which is the interest leading to a higher number of matches. It could have been performed less “mechanically” by iterating over an array in which interests were specified.

```
GET /dating-final/_search
{
  "query":
    { "match":
      { "match": "b'1'" }

    }, "aggs": {
      "avg_sports": { "avg": { "field": "sports" } },
      "avg_tv_sports": { "avg": { "field": "tvsports" } },
      "avg_hiking": { "avg": { "field": "hiking" } },
      "avg_exercise": { "avg": { "field": "exercise" } },
      "avg_dining": { "avg": { "field": "dining" } },
      "avg_museums": { "avg": { "field": "museums" } },
      "avg_art": { "avg": { "field": "art" } },
      "avg_gaming": { "avg": { "field": "gaming" } },
      "avg_clubbing": { "avg": { "field": "clubbing" } },
      "avg_reading": { "avg": { "field": "reading" } },
      "avg_tv": { "avg": { "field": "tv" } },
      "avg_theater": { "avg": { "field": "theater" } },
      "avg_movies": { "avg": { "field": "movies" } },
      "avg_concerts": { "avg": { "field": "concerts" } },
      "avg_music": { "avg": { "field": "shopping" } },
      "avg_yoga": { "avg": { "field": "yoga" } }
    },
    "size": 0
}
```


```
{
  "took": 0,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 1380,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "avg_reading": {
      "value": 7.76996336996337
    },
    "avg_dining": {
      "value": 7.917948717948718
    },
    "avg_theater": {
      "value": 6.7765567765567765
    },
    "avg_art": {
      "value": 6.875457875457876
    },
    "avg_gaming": {
      "value": 3.9611721611721613
    },
    "avg_tv_sports": {
      "value": 4.547252747252747
    },
    "avg_movies": {
      "value": 7.831501831501831
    },
    "avg_sports": {
      "value": 6.553113553113553
    },
    "avg_concerts": {
      "value": 6.956776556776557
```

h. Who are the three “top rated” people in the dataset? (terms of attractiveness, intelligence and funny)

```
GET /dating-final/_search
{
  "size" : 3,
  "query": {
    "match_all": {}
  },
  "sort": [
    { "attractive_o": { "order": "desc" } },
    { "funny_o": { "order": "desc" } },
    { "intelligence_o": { "order": "desc" } } ]
}
```


```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 8378,
      "relation": "eq"
    },
    "max_score": null,
    "hits": [
      {
        "_index": "dating-final",
        "_id": "WALau4wBl3DzbDcOIbyk",
        "_score": null,
        "_source": {
          "d_funny_important": "b'[21-100]'",
          "d_reading": "b'[6-8]'",
          "ambtition_important": 3,
          "d_age": 4,
          "intellicence_important": 25,
          "movies": 8,
          "music": 8,
          "d_tv": "b'[0-5]'",
          "d_expected_num_interested_in_me": "b'[0-3]'",
          "samerace": "b'1'",
          "shopping": 6,
          "d_tvsports": "b'[0-5]'",
          "d_attractive": "b'[6-8]'",
          "d_ambition": "b'[6-8]'",
          "dining": 10,
          "d_shared_interests_o": "b'[0-5]'",
          "intelligence": 7,
          "d_funny": "b'[6-8]'",
          "d_ambtition_important": "b'[0-15]'",
          "d_sincere": "b'[9-10]'",
          "field": "b'Financial Engineering'",
          "guess_prob_liked": 9,
          "sincere": 9,
```


i. Find a person using certain values, and compute its number of expected matches, and the actual number of its matches

Values used in this query identify 10 entries regarding a single person, since no ID is available to identify a single person; with further data wrangling, thus with the addition of an ID, it would be possible (and easier) to perform the same query the entries about a specific person.

```
GET /speeddating-v4/_search
{
  "query": {
    "bool": {
      "filter": [
        { "term": { "gender": "female" }},
        { "term": { "age": 24 }},
        { "term": { "wave": 1 }}
      ]
    }
  },
  "aggs": {
    "count_matches": {
      "sum": {
        "field": "match"
      }
    }
  },
  "size": 1,
  "_source": ["id", "expected_num_matches", "count_matches"]
}
```


```
{
  "took": 0,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 10,
      "relation": "eq"
    },
    "max_score": 0,
    "hits": [
      {
        "_index": "speeddating-v4",
        "_id": "dlq81IwBkPnrhWx6FobH",
        "_score": 0,
        "_source": {
          "expected_num_matches": 3
        }
      }
    ]
  },
  "aggregations": {
    "count_matches": {
      "value": 2
    }
  }
}
```


j. Determine the number of matches in participants with a high interest correlation

```
GET /speeddating-v4/_search
{
  "query": {
    "range": {
      "interests_correlate": {"gt": 0.75}
    }
  },
  "aggs" : {
    "match-count" : {
      "terms" : {
        "field": "match"
      }
    }
  },
  "size" : 0
}
```

In the case of interest correlation greater than 0.75, returns a percentage of more or less: 18.18%, meanwhile for values between 0 and 0.75 it is: 16.73%, when between -0.75 and 0, it is: 15.35%.

```
{
  "took": 0,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 110,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "match-count": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 0,
      "buckets": [
        {
          "key": 0,
          "doc_count": 90
        },
        {
          "key": 1,
          "doc_count": 20
        }
      ]
    }
  }
}
```

5. Extra

As extra work, to further deepen the analysis of the dataset, a dashboard with Kibana has been chosen. In particular, an overview interface has been developed, which presents a general overview of the characteristics and statistics of the dataset. A preview is presented below.

Among the views developed there are:

- The **median of the age** of participants, of which the **number** is simply shown
- **Distribution of participants**, showing the percentage of entries involving males and females. Developed with a **donut diagram**, to show the balance of the slices.
- **Distribution of participants by "race"**. We chose to represent it with a **waffle graph** to transfer the percentages to a sample of 100 participants and make the result clearer.
- The **median of preferences**, represented with a **bar histogram**, to visualize the preferences in the character of potential partners.
- The **median of interests**, also represented with a **vertical bar graph**, to visualize the most "mainstream" interests, and those of which people are less "fond". “`gaming`” appears to be the least interesting for the participants, while “`dining`”, “`movies`”, “`dinner`”, and “`music`” appear to be the most interesting.
- **Matches per wave**, graph with **percentage area**, to show the total trend of match percentages (green) on the total participants per speed dating round.
- The **percentage of people who met previously**, represented with a **pie chart**, shows us that only just under 5% have already met before the experiment.

The developed dashboard presents an overview of various statistics, which can be clicked on to filter the results by aggregate categories.

For example, it is possible to filter the results by displaying only those relating to a male or female audience, from which, for example, it is possible to view a difference in the satisfaction of interests such as yoga (higher for females, second image), and gaming (higher for males, first image).