**Machine Learning Engineer Nanodegree**

**Capstone Project**

Raphael Roullet  
December 1st, 2016

# I. Definition

(approx. 1-2 pages)

## Project Overview

In this section, look to provide a high-level overview of the project in layman’s terms. Questions to ask yourself when writing this section:

* Has an overview of the project been provided, such as the problem domain, project origin, and related datasets or input data?
* Has enough background information been given so that an uninformed reader would understand the problem domain and following problem statement?

This project would be part of the domain of recommender (or recommendation) systems.

Recommendation systems are used to suggest items to users that they will like based on different factors, increasing the chances that they perform an action with it (buying, watching…). With the increase of data on the web and the so-called long-tail, recommendation systems have become widely used to recommend different things: items on Amazon, news articles on Google News and music on Spotify…

Two main types of recommendation systems exist:

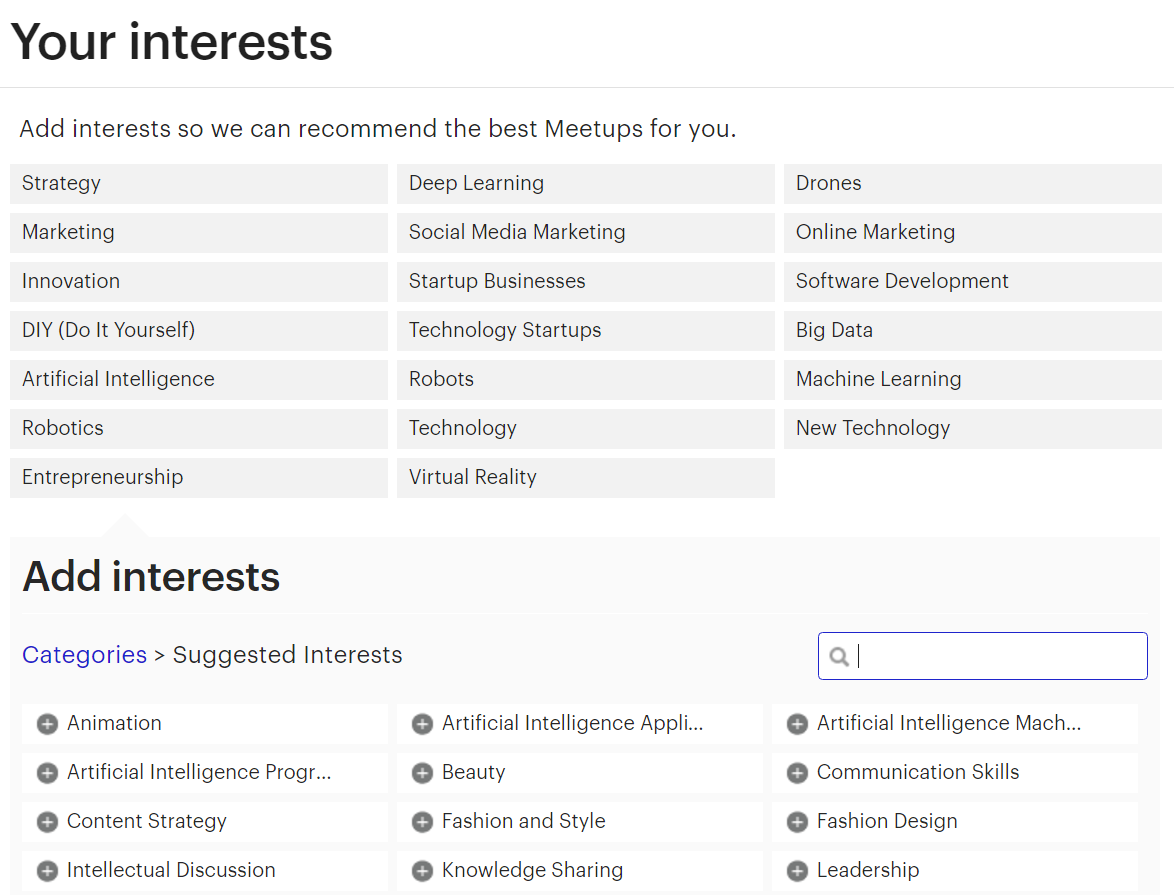
* Content-based: These algorithms use characteristics of items to make predictions
* Collaborative filtering: This type recommends items only based on the users past behavior. It is based on the principle that if someone with similar tastes than me has liked something, I’m more likely to like it as well.

Other types of recommendation systems exist and some, like Netflix’s, can take a hybrid approach.

This project however would take a different approach by *recommending users to other users* based on their similarity, a step that is part of collaborative filtering. Nevertheless, it would take a content-based approach by looking at the characteristics of the users, namely their interests. This project, indeed, aims at **recommending to users other users with similar interests, hobbies, passions…**

The ideal dataset would contain data about real people and their interests. We would need a list of users and for each one a list of their interests such that users would have interests in common. The bigger the dataset, the better since algorithms tend to perform better and better when given larger and larger datasets.

It turns out that this data can be found using the Meetup API. “Meetup is an online social networking portal that facilitates offline group meetings in various localities round the world.” On the platform, users can add interests to their profiles so that Meetup can recommend Meetup groups to join:



Using Meetup’s official API, 103,729 profiles of Meetup members were fetched in a 15-mile radius from Miami. Among these, 82,186 persons had at least one interest indicated in their profile.

This data seems to be perfect to work with in order to solve our problem since it meets all the requirements mentioned above. It might be needed to ignore persons that don’t have at least a minimum number of interests indicated. In addition, only a subset of the 82,186 might be used because of constraints in terms of computational resources.

## Problem Statement

In this section, you will want to clearly define the problem that you are trying to solve, including the strategy (outline of tasks) you will use to achieve the desired solution. You should also thoroughly discuss what the intended solution will be for this problem. Questions to ask yourself when writing this section:

* Is the problem statement clearly defined? Will the reader understand what you are expecting to solve?
* Have you thoroughly discussed how you will attempt to solve the problem?
* Is an anticipated solution clearly defined? Will the reader understand what results you are looking for?

Given a dataset of X individuals and for each individual a list of n discrete interests from a finite universe of N interests, the problem is the following:

For a given individual how can we order the list of all the others X-1 individuals in descending order by similarity of interests.

In order to recommend to users the 5, 10 or 20 users with whom they share the most interests we need to compute the similarity of their interests with those of all other users and then ranked users by descending similarity. By considering each list of users interests as documents the problem becomes one that appertains to the semantic analysis field which is well-documented (pun unintended).

The solution would then consist of using a similarity measure to compute the similarity between users’ interests (documents) after turning them into vectors.

## Metrics

In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These calculations and metrics should be justified based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section:

* Are the metrics you’ve chosen to measure the performance of your models clearly discussed and defined?
* Have you provided reasonable justification for the metrics chosen based on the problem and solution?

Similarity can be tricky to define as it can be considered somehow subjective. However, in order to have quantifiable way to evaluate our model, two metrics will be used:

* The solution model and the benchmark model are expected to produce a measure of similarity for any given pair of users (, ) as represented by a number between 0 and 1.
* The other metric used to evaluate the performance of both models will be determined by the number of interests that the queried person shares with the most matching persons returned.

# II. Analysis

(approx. 2-4 pages)

## Data Exploration

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

* If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?
* If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?
* If a dataset is **not** present for this problem, has discussion been made about the input space or input data for your problem?
* Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)

The dataset used for this project consists of profiles fetched from Meetup.com using Meetup’s API. First, it returned 1266 groups in a 15-mile radius from Miami. We then extracted 103,729 profiles, all members of these groups. Each data entry has the following features, selected among those provided by the API (<http://www.meetup.com/meetup_api/docs/2/members/>). Here’s an example for someone who really seems to like dogs.

{

'city': 'Miami',

'topics':

[

{‘name’: 'Pets', ‘urlkey’: 'pets-animals', ‘id’: 53052},

{‘name’: 'Dogs', ‘urlkey’: 'dogs', ‘id’: 15067},

{‘name’: 'Active Dogs', ‘urlkey’: 'activedogs', ‘id’: 9772},

{‘name’: 'Animals', ‘urlkey’: 'animals', ‘id’: 37663},

{‘name’: 'English Bulldog', ‘urlkey’: 'engbulldog', ‘id’: 560},

{‘name’: 'Off-Leash Dog Recreation', ‘urlkey’: 'offleash', ‘id’: 9753},

{‘name’: u'Pug', ‘urlkey’: 'pug', ‘id’: 591}

],

'link': 'http://www.meetup.com/members/111111111',

‘id’: 111111111,

‘name’: 'John Doe'

}

As seen above, each profile contains information about:

* **City**: Where the person is located
* **Topics**: or what we will call interests. For each interest we have the name, urlkey (used to access on Meetup.com) and unique id of the topic/interest.
* **Link**: towards the person’s profile on Meetup.com
* **Id**: Unique id of the person
* **Name**: Name of the person

Among all the information provided for each profile, only topics will actually be used to feed our algorithm. The rest is extra information that could be used for future reference.

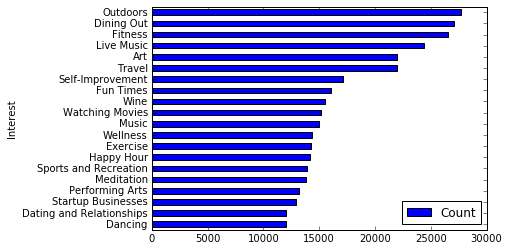
Among all the profiles, 82,186 had at least one interest indicated in their profile. Let’s explore more about those interests.

## Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

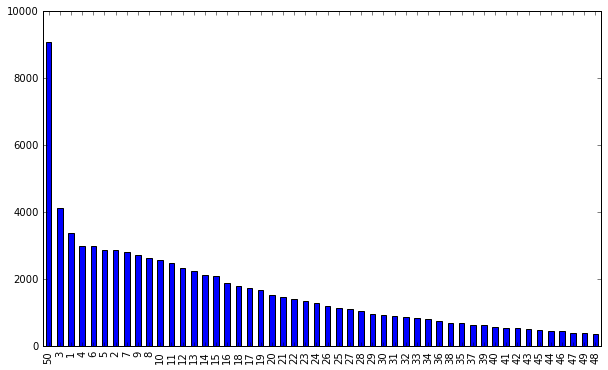
* Have you visualized a relevant characteristic or feature about the dataset or input data?
* Is the visualization thoroughly analyzed and discussed?
* If a plot is provided, are the axes, title, and datum clearly defined?

Here are the most common interests among our dataset with their respective count:



The top interests are not surprising: Outdoors, dining out, Music, Art, Travel, Fun times… are all very broad categories that are highly appreciated by most people. Fitness might be so high in the ranking because body perfection is particularly regarded in places like Miami Beach.

How many interests do people have indicated on their Meetup profile? The following chart provides a good overview of the breakdown:



50 is the maximum number of interests you can select on Meetup and by far the most common number among our dataset followed by small number 3,1,4,6… up to 48 in almost ordinal order.

## Algorithms and Techniques

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

* Are the algorithms you will use, including any default variables/parameters in the project clearly defined?
* Are the techniques to be used thoroughly discussed and justified?
* Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?

From the Meetup data as input, the approach taken would be that of treating each list of user interests as a bag of words (or “documents”): <http://scikit-learn.org/stable/modules/feature_extraction.html#the-bag-of-words-representation>

As opposed to some other documents, ours are already cleaned in the sense that there’s no need to tokenize documents, remove common words (a, the, and …) and punctuation as it is often the case in NLP and semantic analysis.

The goal is to vectorize our corpus: turn each individual list of interests into sparse vectors using each interest as a feature. Scipy.sparse matrices will be used because for each person, the number of non-selected interests is huge, resulting in many features whose values are zero. This will give us a matrix Y where rows are users and columns are interests. Given the size of the corpus, it might even be wise to perform the hashing trick: <http://scikit-learn.org/stable/modules/feature_extraction.html#vectorizing-a-large-text-corpus-with-the-hashing-trick>

In analyzing corpuses, it is very common to use tf-idf. However, in our case term frequency is not relevant because each interest appears only once in each document. Inverse document frequency seems to be more interesting to explore since it would give a lower weight to interests that most people share and that by definition are not helping us better match people with similar interests.

With 21,892 unique interests, it might be interesting to try to reduce the vector space into a space of lower dimensionality. This might reveal some interesting associations between interests. Some transformations to explore include Latent Dirichlet Allocation and Latent Semantic Indexing (or Latent Semantic Analysis): <http://scikit-learn.org/stable/modules/decomposition.html#truncated-singular-value-decomposition-and-latent-semantic-analysis>

Finally, given a sparse vector X representing the interests of one person, I’d have to find the most similar vector among all the others. To do this, we would use as a measure of similarity cosine similarity (<http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html>) or Euclidean distance (<http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.euclidean_distances.html>) to compute the similarity of each pair X and a vector of Y. Then results will be shown by descending order of similarity.

## Benchmark

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

* Has some result or value been provided that acts as a benchmark for measuring performance?
* Is it clear how this result or value was obtained (whether by data or by hypothesis)?

As a benchmark model we could take the results obtained by using “gensim”, an excellent Python library made by Radim Řehůřek, which was designed to automatically extract semantic topics from documents: <https://radimrehurek.com/gensim/index.html>

Gensim can also be used to compute document similarity and output a list of documents ordered by similarity if given a query *or another document as a query*: <https://radimrehurek.com/gensim/tut3.html>

It includes its own implementation of tfidf, LSA and cosine similarity. So in the end, we would give the exact same corpus (our dataset of users interests) as an input to both the gensim model and our model, pick randomly 50 users and compare what the two models return as the most matching users, i.e the vectors with the highest cosine similarities.

# III. Methodology

(approx. 3-5 pages)

## Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?
* Based on the **Data Exploration** section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?
* If no preprocessing is needed, has it been made clear why?

## Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?
* Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?
* Was there any part of the coding process (e.g., writing complicated functions) that should be documented?

## Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* Has an initial solution been found and clearly reported?
* Is the process of improvement clearly documented, such as what techniques were used?
* Are intermediate and final solutions clearly reported as the process is improved?

# IV. Results

(approx. 2-3 pages)

## Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?
* Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?
* Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?
* Can results found from the model be trusted?

## Justification

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* Are the final results found stronger than the benchmark result reported earlier?
* Have you thoroughly analyzed and discussed the final solution?
* Is the final solution significant enough to have solved the problem?

# V. Conclusion

(approx. 1-2 pages)

## Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* Have you visualized a relevant or important quality about the problem, dataset, input data, or results?
* Is the visualization thoroughly analyzed and discussed?
* If a plot is provided, are the axes, title, and datum clearly defined?

## Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* Have you thoroughly summarized the entire process you used for this project?
* Were there any interesting aspects of the project?
* Were there any difficult aspects of the project?
* Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?

## Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

* Are there further improvements that could be made on the algorithms or techniques you used in this project?
* Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?
* If you used your final solution as the new benchmark, do you think an even better solution exists?

Before submitting, ask yourself. . .

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?