Machine Learning Engineer Nanodegree

Capstone Proposal

Raphael Roullet
October 30st, 2016

Proposal

(approx. 2-3 pages)

Domain Background

(approx. 1-2 paragraphs)

In this section, provide brief details on the background information of the domain from which the project is proposed. Historical information relevant to the project should be included. It should be clear how or why a problem in the domain can or should be solved. Related academic research should be appropriately cited in this section, including why that research is relevant. Additionally, a discussion of your personal motivation for investigating a particular problem in the domain is encouraged but not required.

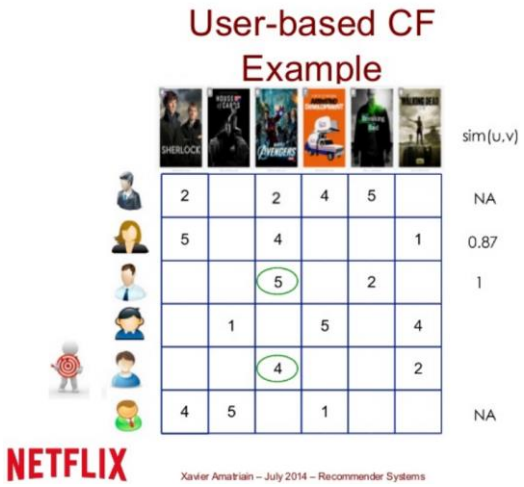
This project would be part of the domain of recommender (or recommendation) systems. These systems are used to suggest items to users that they will like based on different factors, increasing the chances that they perform an action with it (buying, watching…). With the increase of data on the web and 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 can take a hybrid approach. For instance, Netflix, which launched the famous Netflix Prize, opted for a recommender system that takes into account user similarity (collaborative filtering) and movie similarities (content-based).

Most of the recommendation systems are geared towards recommending items to users through collaborative filtering which starts by generating a matrix with n users and m items:



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.

My personal motivation for this project stems from a desire to turn this project into an API that could be used through a mobile app including messaging between users.

Problem Statement

(approx. 1 paragraph)

In this section, clearly describe the problem that is to be solved. The problem described should be well defined and should have at least one relevant potential solution. Additionally, describe the problem thoroughly such that it is clear that the problem is quantifiable (the problem can be expressed in mathematical or logical terms), measurable (the problem can be measured by some metric and clearly observed), and replicable (the problem can be reproduced and occurs more than once).

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.

Similarity is a word used in our daily life but for this project several meanings of similarity should be considered: http://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/

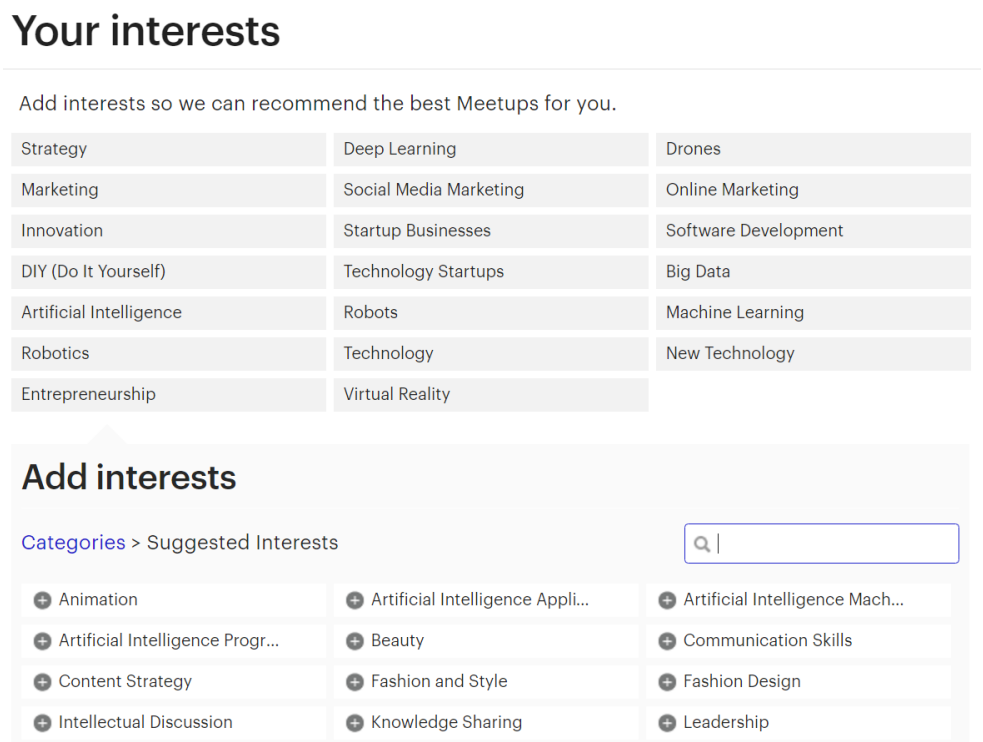
Datasets and Inputs

(approx. 2-3 paragraphs)

In this section, the dataset(s) and/or input(s) being considered for the project should be thoroughly described, such as how they relate to the problem and why they should be used. Information such as how the dataset or input is (was) obtained, and the characteristics of the dataset or input, should be included with relevant references and citations as necessary It should be clear how the dataset(s) or input(s) will be used in the project and whether their use is appropriate given the context of the problem.

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, I fetched 103,729 profiles of Meetup members in a 15-mile radius from Miami. Among these, 82,186 persons had at least one interest indicated on their profile for a total number of 21,892 unique interests. Here are the 10 most common interests with their respective count:

```
[(u'Outdoors', 27747),
 (u'Dining Out', 27091),
 (u'Fitness', 26581),
 (u'Live Music', 24432),
 (u'Art', 21946),
 (u'Travel', 21925),
 (u'Self-Improvement', 17139),
 (u'Fun Times', 16044),
 (u'Wine', 15550),
 (u'Watching Movies', 15172)]
```

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.

Solution Statement

(approx. 1 paragraph)

In this section, clearly describe a solution to the problem. The solution should be applicable to the project domain and appropriate for the dataset(s) or input(s) given. Additionally, describe the solution thoroughly such that it is clear that the solution is quantifiable (the solution can be expressed in mathematical or logical terms) , measurable (the solution can be measured by some metric and clearly observed), and replicable (the solution can be reproduced and occurs more than once).

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 after turning them into vectors.

Benchmark Model

(approximately 1-2 paragraphs)

In this section, provide the details for a benchmark model or result that relates to the domain, problem statement, and intended solution. Ideally, the benchmark model or result contextualizes existing methods or known information in the domain and problem given, which could then be objectively compared to the solution. Describe how the benchmark model or result is measurable (can be measured by some metric and clearly observed) with thorough detail.

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.

Evaluation Metrics

(approx. 1-2 paragraphs)

In this section, propose at least one evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model. The evaluation metric(s) you propose should be appropriate given the context of the data, the problem statement, and the intended solution. Describe how the evaluation metric(s) are derived and provide an example of their mathematical representations (if applicable). Complex evaluation metrics should be clearly defined and quantifiable (can be expressed in mathematical or logical terms).

The evaluation metric proposed is for each person (or vector), the difference of the cosine similarities of each of the others vectors between the ones produced by the benchmark model and the ones produced by the solution model.

Another metric that can be used to evaluate the performance of both models is to determine the number of interests that the queried person shares with the most matching persons returned.

Project Design

(approx. 1 page)

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

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.

Before submitting your proposal, ask yourself. . .

- Does the proposal you have written follow a well-organized structure similar to that of the project template?
- Is each section (particularly **Solution Statement** and **Project Design**) 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 proposal?
- Have you properly proofread your proposal to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?