Use Case #2 *Mentor-mentee recommender system*

Connecting people with people is a central task in an important class of applications. Searching for mentors and matching mentors with mentees is a good example of such application. For this assignment, we'd like you to design and implement a prototype of a recommender system for mentor-mentee matching. Such a recommender system should match mentors with mentees by considering their attributes and preferences. For the mentors the system should store their inferred attributes from the data corpora, specifically their expertise on a particular topic. For the mentees the system should capture their preferences, which come from a certain ontology of the topics.

As an input data set to be used to retrieve the mentors, infer the topics and measure mentor's expertise level, please use one of the two datasets:

- the Enron Email dataset available here: http://www.cs.cmu.edu/~./enron/enron_mail_20150507.tgz, where mentors will be the email senders and topics will be inferred from the email message body
- the DBLP Computer Science Bibliography dataset, available here: http://dblp.uni-trier.de/xml/, where mentors will be the authors and topics will be inferred from the titles of their publications.

Deliverables:

In the final presentation you'll have to justify and explain:

- design choices behind the prototype of your system
- methods and libraries used for topic extraction
- proposed metric used for measuring the level of expertise of a mentor
- proposed recommendation method
- proposed method for quantifying the performance of your recommender system

Together with your presentation (in PDF) you'll need to provide us with your source code written in Python – available on GitHub latest one day before the on-site meeting at PMI.

Use Case #3 *What drives the sales?*

The ultimate goal of each company is to achieve the highest revenue. In order to do that, a company should maximize sales. There are many ways to achieve that. One of them is to maximize sales in physical point of sales. How to do that? Possible solution could be to place the product in the most convenient location for consumers. In this task, we will ask you to check what surroundings are leading to top point of sales performance.

For performing this analysis, please use following data sources:

Based on dataset *sales_granular.csv* create a target variable, that you will use for modelling.

It is up to you to decide:

- how you design the target variable
- what timeframe you use to calculate the target variable
- if you want to treat it as a classification or regression problem.

In the dataset *Surroundings.json* you will find information about 90 different amenities like restaurants, shops, beauty salons etc. that are in the surroundings of each point of sales.

All datasets required for this use case are available here:
https://drive.google.com/drive/folders/1YR2n9Leh98s2TbX2Ugo5bku6pYzEb-W9?usp=sharing

Based on this dataset please prepare a table with exploratory variables you will use for modeling.

Please create and evaluate a model that identifies important attributes in the surroundings that impact sales.

Deliverables:

Please prepare the presentation with your results. In the final presentation you'll have to justify and explain:

- How you created a target variable, please be ready to explain all the assumptions that you made
- How you split datasets into training and testing,
- Techniques that you used for modelling,
- Metric that you used for quantifying the performance of your model

Together with your presentation (in PDF) you'll need to provide us with your source code written in Python – available on GitHub latest one day before the on-site meeting at PMI.

Use Case #4 *Aspect Term Extractor*

Companies that provides services or products to their customers would like to know which **aspects** of their products preoccupy the most the customers. This knowledge will help to improve their products in the future. Are people talking the most about the **quality** of the food, the **price** of an item or the **battery life** of a computer? All those terms are what we call aspect term.

You are asked to build an *Aspect Term Extractor.* An ATE is a model that extracts the aspect terms from a review, i.e. for each word of one review, your model should predict if the word is an aspect term or not.

The input data use the following format: review → list of aspect terms, e.g. "The battery life is really good and its size is reasonable" → “battery life”, “size”. We recommand to change it and use the BIO format instead. Example:

The	**battery**	**life**	**is**	**really**	**good**	**and**	**its**	**size**	**is**	**reasonable**
O	B	I	O	O	O	O	O	B	O	O

With this format, we can see that the role of an ATE is to assign to each word one of the 3 possible class:

- O = not an aspect (Outside)
- B = first word of an aspect (Beginning)
- I = second, third, ... word of an espect (Inside)

All datasets required for this use case are available here:
https://drive.google.com/drive/folders/1YR2n9Leh98s2TbX2Ugo5bku6pYzEb-W9?usp=sharing

Use *Laptops_Train_v2.xml* to train your model and then evaluate it using *Laptops_Test_Gold.xml*

N.B: 2 aspects can exist in the same sentence and that one aspect term can be composed of more than one token (Batery life). In such case it should be clear that it forms one aspect and not 2 aspects.

Deliverables:

Please prepare the presentation with your results. In the final presentation you'll have to justify and explain:

1. The features extraction process: what is the intuition behind your method, which features are the most important.
2. The algorithm you selected (we ask you to explore **at least 2** different algorithms)
3. The metric that you used for quantifying the performance of your model

N.B. You model must be **interpretable**, i.e. you are **not allowed** to use any kind of deep learning method (LSTM, CNN, …)

Together with your presentation (in PDF) you'll need to provide us with your source code written in Python – available on GitHub latest one day before the on-site meeting at PMI.