# **Kaggle in class:**

# [**Master Data Science/MVA data competition 2017**](https://inclass.kaggle.com/c/master-data-science-mva-data-competition-2017)

**Report By : TeamX**

MEZZI NARJES & OUMOUSS EL MEHDI

Master 2 Data & Knowledge (Télécom Paristech)

<https://inclass.kaggle.com/c/master-data-science-mva-data-competition-2017/rules>

**Data Challenge 2017**

**Problem:** Email recipient recommendation

It was shown that at work, employees frequently forget to include one or more recipient(s) before sending a message. Conversely, it is common that some recipients of a given message were actually not intended to receive the message. To increase productivity and prevent information leakage, the needs for effective email recipient recommendation systems are thus pressing.

**Goal:** Based on the available data given, develop a system capable of recommending a list of 10 recipients ranked by decreasing order of relevance.

**DataSet:**

To implement a solution to this problem, we had to start from somewhere, and there is no better place to start from than the data at disposal. The fact that the domain word ‘enron’ came out in the majority of the emails, let us wonder what is this ‘enron’ about, to find out that behind this data there is a whole long story of fraud and bankruptcy of the Enron Corporation. The **Enron Email Corpus** is a large dataset, consisting of emails generated Enron’s employees.

**Why this dataset is very important ?** “This data is valuable; to my knowledge it is the only substantial collection of "real" email that is public” [3]

**Solution:**

After documenting about the Enron Email Corpus, we read several articles and papers. One of the papers that we appreciated the most was [1], thanks to its simplicity and pretty well explained. Next, we explain some of the approaches treated in the paper.

**Approach 1. TfIdf-Centroid**

As explained in [1], this technique is based on cosine similarity between two TF-IDF vector-based representations of the email messages.

* **Training (normally this must be called feature engineering):**

For every message in the training set sent by user(sender):

* Based on the mail’s body, we get its TF-IDF vector representation.
* Normalized the vector.

For every recipient in the Address Book |AB|of the user:

* Build a TF-IDF centroid vector, which represents the sum of the normalized TF-IDF vectors of all messages that were sent from the sender to recipient.
* **Testing:**

For each message in the test set:

* Get the TF-IDF vector representation of the message
* Computed the cosine similarity between this representation and the |AB(u)| Centroid vectors.
* Finally, Rank the |AB| recipients based on the cosine similarity scores .

**Approach 2. K-Nearest-Neighbors**

In this second approach, we applied the Knn-30 method presented in [1]. But due to computational constraints, we considered only the 20 nearest neighbors for each email. To implement the method we proceeded in the following way :

* Created for each sender a csv file which contains its emails in the training set.
* For each sender :
  + From the created file, build a corpus.
  + Retrieved the email we wish to predict its recipients from the test set.
  + Added the email from the test set to the corpus.
  + Calculated the TF-IDF vectors of the different emails.
  + Calculated the similarity between the test email and the training emails using the cosine distance.
  + Chose the 20 emails the most similar to the test email.
  + Pick all the recipients of the 20 chosen emails and calculate the similarity score of each one. (the similarity score is the sum of the cosine distance between the test email and the training emails to which the recipient belong)
  + Chose the 10 recipients having the greatest similarity score.
* Add all the result to a commun csv file.

**Propositions:**

We tried also to attack this problem from a machine learning perspective, by looking to the prediction of recipient as Multi-label classification problem.

We did follow the next steps, which included some data preparation, especially :

* Merging the two given datasets (training-set and training\_info based on the ‘mid’).
* Apply some transformation (use of dictionaries as mappings between the emails and given ids, in order to deal with numerical data rather than text)
* Use of tfidfVectorizer to get a numerical representation of the message body.
* Once the data was ready, we applied some machine learning algorithms

(OneVsRestClassifier + LinearSVC/SGDClassifier, KNN, RandomForest)

N.B: Constrained by limited computational power at disposal, the ML experiments were conducted on small subset of the data. Also, the results lacked a sense of order, which can be added ulteriorly to pick the top 10 predictions.

**Take offs:**

Working on this Data Challenge was of great importance to us, in the sense that we learnt a lot in matter of :

* Text processing (NLP).
* Different approaches to take on more or else the same problematic.
* How different people can use the same data, yet draw different insights and apply it on different applications.

**References:**

1. Vitor R. Carvalho, William W. Cohen, ‘Recommending Recipients in the Enron Email Corpus’ <http://www.cs.cmu.edu/~wcohen/postscript/cc-predict-submitted.pdf>
2. Scikit-learn <http://scikit-learn.org/stable/index.html>

(feature\_extraction, , ...)

1. Enron Email Dataset, <https://www.cs.cmu.edu/~./enron/>

**For more readings !**

* **Investigating Enron’s email corpus: The trail of Tim Belden (Neo4j)**

<https://linkurio.us/investigating-the-enron-email-dataset/>

Neo4j a graph database management system, applied to the Enron Email Corpus.

* **Intelligent Email: Aiding Users with AI**

<http://dirichlet.net/pdf/dredze08intelligent.pdf>

Artificial Intelligence application to Emails (summary keyword generation, **reply prediction** and attachment prediction)

* **Intelligent Email: Reply and Attachment Prediction**

<https://www.cs.jhu.edu/~mdredze/publications/dredze_intelligent_email_iui08.pdf>

* **EnronData - Research**

<https://enrondata.readthedocs.io/en/latest/references/research/>

Some research studies conducted on the Enron Email Dataset.

* **Network Analysis with the Enron Email Corpus**

<http://ww2.amstat.org/publications/jse/v23n2/hardin.pdf>

Study relationships in a network by applying centrality measures.

* **SNAP : Stanford Network Analysis Project** (Python)

<https://snap.stanford.edu/snappy/index.html>

<https://snap.stanford.edu/data/email-Enron.html>

A general purpose network analysis and graph mining library.

* **Towards Hierarchical Email Recipient Predictions**

<https://www.youtube.com/watch?v=r1wVAz3opSE>

Presentation of paper, 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (IEEE CollaborateCom 2012)