CRUDE OIL PRICES PREDICTIONS USING NLP
MAXIME GENIN, ARTHUR MORLOT, SOPHIE TRASTOUR

PROBLEM
In our project, we focused on the correlations that could exist between the commodity prices of crude oil and external factors highlighted in newspapers. Thanks to machine learning and NLP (natural language processing) techniques, more and more documents can be processed in a semi-automated way. We used topic modeling (Latent Dirichlet Allocation) to extract the main topics from the articles in newspapers such as New-York Times, Reuters and the Associated Press so as to predict the movement of the stock oil price.

DATA
We used the NY Times API to get automatically all articles from 1986 to 2015 containing the words “oil price”, which corresponds to around 30,000 articles. We were able to extract the headline, an abstract and a snippet. We used several API, the nytimesarticle package, and the python function time.sleep() to avoid the limitations of 5 articles per second and 1,000 articles per day. We used the stock oil price data from the EIA website. Finally, in order to improve the quality of our predictions, we adjusted the stock oil prices with the US inflation from 1974.

PREPROCESSING
There are packages available to do topic modeling in python like gensim and pyLDAvis [3]. We preprocessed the newspapers articles by:
- setting to lower case
- removing the punctuation
- removing stop words (like “I”, “my”, “their”) which does not carry meaning
- lemmatizing the words (so that “like” and “liked” would be treated the same way)
- removing the numbers

The list of stop words and the data for lemmatizing words were extracted from the nltk package in python. We first set the number of topics to 10. The topics found were:

NLP method: Latent Dirichlet Allocation (LDA)
We decided to use a LDA technique because of its capacity to capture multiple topics within a document (a complete description can be found in [1]). The intuition behind the LDA is the following: each document is a “mixture” of different topics. A document may be 90% about “oil” and 10% about “cars” (see figure 1). We now make an assumption: if a document is composed of 90% about “oil” and 10% about “cars”, then it is constructed by randomly sampling 90% of its words from a distribution about “oil”, and 10% from a distribution about “cars” (the ordering of the words does not matter to the algorithm). We end up with three hidden variables to explain our corpus:
- The topics, that is to say the words distribution inside a topic
- Per-document topic assignments
- Per-document per-word topic assignments
Each of them can be set to a specific prior before we run the algorithm, to encode information known by humans about the subject.

The LDA then uses inference algorithms to compute the posterior on these distributions and infer the more likely ones.

The topics found were:
- Commodity prices
- Financial news
- Oil crisis
- Economic news
- Geopolitical news
- Weather news
- Renewable energy
- Oil production
- Oil consumption
- Oil prices

RESULTS
Linear regression with 10 topics:

<table>
<thead>
<tr>
<th>Features</th>
<th>Outputs</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily proportions for each topic</td>
<td>Daily returns</td>
<td>Too volatile</td>
</tr>
<tr>
<td>Monthly proportions for each topic</td>
<td>Monthly returns</td>
<td>Better but still volatile</td>
</tr>
</tbody>
</table>

So as to select the best model, we used a 10-fold cross validation strategy adjusting the following parameters:
- Number of topics: 10, 40, 60 and 100
- Polynomial features to introduce non-linearity
- Different LDA models (stopwords list, topics)
- Type of regression: Thell-Sen, least-square, ridged least-square, locally weighted regression

Best model found: locally weighted regression with 60 topics and no polynomial features \( R^2 = 0.70 \)

REFERENCES