Table of contents

1 - Introduction

The goal of this case study is to provide a comprehensive quantitative overview on science's current topics in order to give insights to our company, which is aiming to use those insights and knowledge to perhaps, direct its activity towards a more academic and research direction.

To achieve this, a dataset is provided, to which I, as the data scientist at this company, have access to. The dataset is available on Kaggle at https://www.kaggle.com/Cornell-University/arxiv.

This dataset contains around 2 million papers on different science subjects and areas published between 1991 and now. This work intends to focus only on current topics in science so it might be needed to filter the dataset in this aspect.

The data has been downloaded on the 10th of July 2022, this statement is important because there are weekly updates and at the time of reading this written assignment the dataset might have changed slightly. This change will probably have very little influence in the analysis, but it is nevertheless important noticing.

This analysis described in the next chapters was of course iterative and some other approaches were tried with less or no success.

I will try either way to show some rationale that was perhaps interesting but turned out to be irrelevant for this work, hard to follow or get usable information.

The software used besides python, was Jupiter notebooks and its 'derivative' Atom's Hydrogen which also allows for some similar functionalities. Apart from this numerous data science relevant python libraries like spacy, sklearn, pandas, numpy were used besides some 'native' basic python libraries like datetime and string.

The hardware used was a 2014 Macbook Pro which was as expected, a much slower experience than it would be with a more modern computer.

We will then start with the data loading, cleaning and pre-processing and aim to give a clear understanding to the reader of what was performed and why, either through an explanation or through screenshots and we will go from there.

2 - Data loading and raw feature selection

After a quick look at the data and its documentation we observe that the data is in the json format. In order to load it and work with it, it is a good idea to start by loading some libraries like json and dask. Json is useful because the downloaded dataset is in json format and dask because since I am working on a laptop my resources are limited, so this library can make it faster to load a large dataset like this one.

```
import pandas as pd
import dask.bag as db
import json
```

```
data = db.read_text('/Volumes/未命名/machine_uns_case_study/science_topic.json').map(json.loads)
```

```
data.count().compute()
```

```
2090016
```

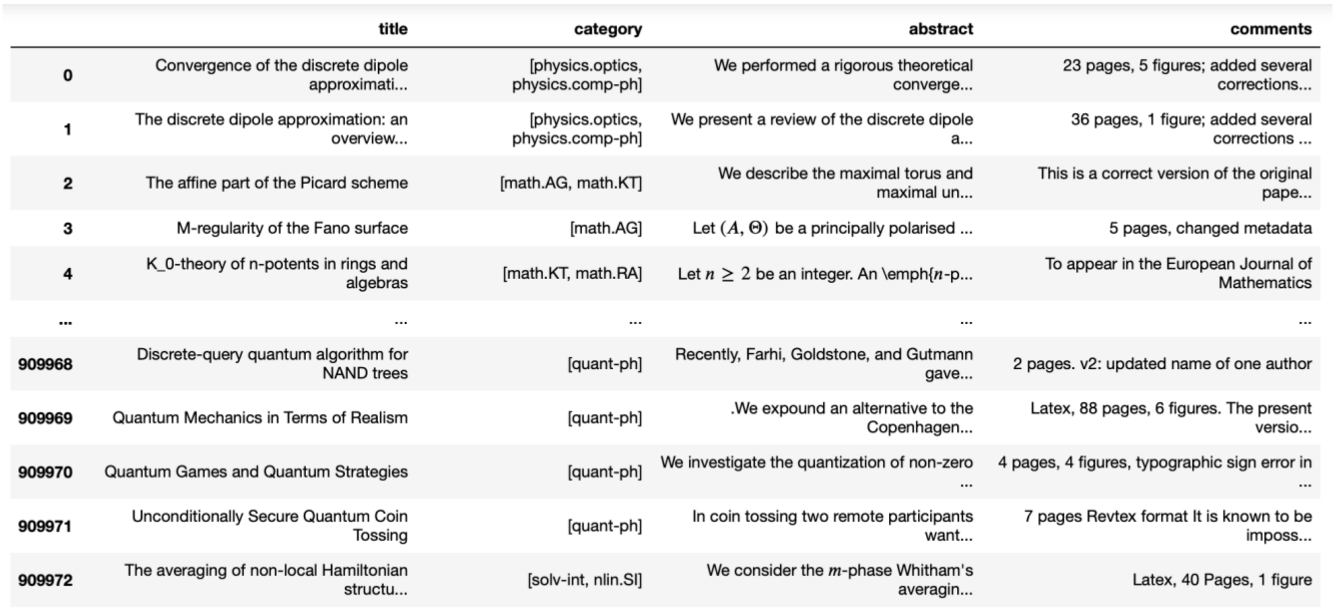
As we see there are over 2 million data records. Later, perhaps it would be a good idea to first analyze a subset of the data due to computing power constraints.

```
({'id': '0704.0001',
  'submitter': 'Pavel Nadolsky',
  'authors': "C. Bal\\'azs, E. L. Berger, P. M. Nadolsky, C.-P. Yuan",
  'title': 'Calculation of prompt diphoton production cross sections at Tevatron and\n  LHC energies',
  'comments': '37 pages, 15 figures; published version',
  'journal-ref': 'Phys.Rev.D76:013009,2007',
  'doi': '10.1103/PhysRevD.76.013009',
  'report-no': 'ANL-HEP-PR-07-12',
  'categories': 'hep-ph',
  'license': None,
  'abstract': '  A fully differential calculation in perturbative quantum chromodynamics is\npresented for the prod
uction of massive photon pairs at hadron colliders. All\nnext-to-leading order perturbative contributions from quar
k-antiquark,\ngluon-(anti)quark, and gluon-gluon subprocesses are included, as well as\nall-orders resummation of i
nitial-state gluon radiation valid at\nnext-to-next-to-leading logarithmic accuracy. The region of phase space is\n
specified in which the calculation is most reliable. Good agreement is\ndemonstrated with data from the Fermilab Te
vatron, and predictions are made for\nmore detailed tests with CDF and DO data. Predictions are shown for\ndistribu
tions of diphoton pairs produced at the energy of the Large Hadron\nCollider (LHC). Distributions of the diphoton p
airs from the decay of a Higgs\nboson are contrasted with those produced from QCD processes at the LHC, showing\nth
at enhanced sensitivity to the signal can be obtained with judicious\nselection of events.\n',
  'versions': [{'version': 'v1', 'created': 'Mon, 2 Apr 2007 19:18:42 GMT'},
   {'version': 'v2', 'created': 'Tue, 24 Jul 2007 20:10:27 GMT'}],
  'update_date': '2008-11-26',
  'authors_parsed': [['Balázs', 'C.', ''],
   ['Berger', 'E. L.', ''],
   ['Nadolsky', 'P. M.', ''],
   ['Yuan', 'C. -P.', '']]},)
```

Afterwards after analyzing a data record, we can already identify some interesting columns that might come in handy in our analysis and some that we can discard for the moment. Columns like 'Id','submitter','journal-ref','doi','report-no' and 'license' I will discard them for the moment because the rest of the columns in principle should offer more relevant information for our analysis. I will also discard the columns 'authors' and 'authors_parsed' for now.

Like I mentioned before this work should focus on current topics in science. Therefore, filtering the dataset for records after 2016, or from 2017, which gives us around 6 years of data, is a must to better understand current trends and not having past trends 'cloud' our analysis. This was achieved by filtering through the second key 'created' of the of the 'versions' feature.

In previous iterations there were other columns kept like 'authors_parsed' and 'submitter', but both did not add any further information to the current trends in science. If we notice in the feature comments in the beginning the number of pages is stated so we will keep that feature because the number of pages could provide us an insight on the current trend of the length of papers.

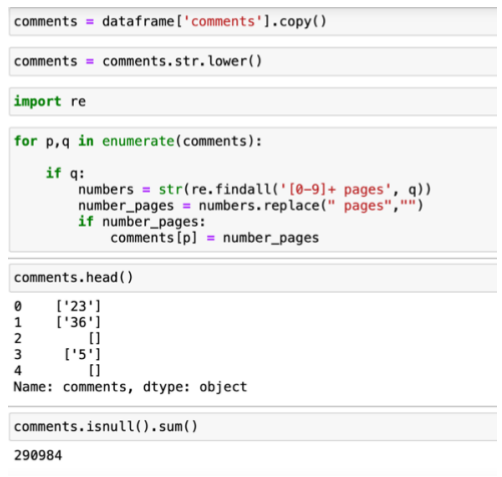


	title	category	abstract	comments
0	Convergence of the discrete dipole approximati...	[physics.optics, physics.comp-ph]	We performed a rigorous theoretical converge...	23 pages, 5 figures; added several corrections...
1	The discrete dipole approximation: an overview...	[physics.optics, physics.comp-ph]	We present a review of the discrete dipole a...	36 pages, 1 figure; added several corrections ...
2	The affine part of the Picard scheme	[math.AG, math.KT]	We describe the maximal torus and maximal un...	This is a correct version of the original pape...
3	M-regularity of the Fano surface	[math.AG]	Let (A, Θ) be a principally polarised ...	5 pages, changed metadata
4	K_0-theory of n-potents in rings and algebras	[math.KT, math.RA]	Let $n \geq 2$ be an integer. An \emph{n-p...	To appear in the European Journal of Mathematics
...	...	...	...	...
909968	Discrete-query quantum algorithm for NAND trees	[quant-ph]	Recently, Farhi, Goldstone, and Gutmann gave...	2 pages. v2: updated name of one author
909969	Quantum Mechanics in Terms of Realism	[quant-ph]	.We expound an alternative to the Copenhagen...	Latex, 88 pages, 6 figures. The present versio...
909970	Quantum Games and Quantum Strategies	[quant-ph]	We investigate the quantization of non-zero ...	4 pages, 4 figures, typographic sign error in ...
909971	Unconditionally Secure Quantum Coin Tossing	[quant-ph]	In coin tossing two remote participants want...	7 pages Revtex format It is known to be imposs...
909972	The averaging of non-local Hamiltonian structu...	[solv-int, nlin.SI]	We consider the m-phase Whitham's averagin...	Latex, 40 Pages, 1 figure

The dataset after being loaded and filtered for dates after 2016. There are about 900 thousand rows left. The next step is processing the text features and feature engineering.

3 - Processing text features and feature engineering

Let's start by the feature comments. Which might contain some interesting information regarding the number of pages. We will clean it with Regex and transform it into a new feature called 'number_pages'.



```
comments = dataframe['comments'].copy()
```

```
comments = comments.str.lower()
```

```
import re
```

```
for p,q in enumerate(comments):

    if q:
        numbers = str(re.findall('[0-9]+ pages', q))
        number_pages = numbers.replace(" pages","")
        if number_pages:
            comments[p] = number_pages
```

```
comments.head()
```

```
0    ['23']
1    ['36']
2        []
3     ['5']
4        []
Name: comments, dtype: object
```

```
comments.isnull().sum()
```

```
290984
```

As you can see in the picture after extracting the number of pages with Regular expressions, we observe that there are around 290 thousand null values which is about a third. This might mean that this feature contains too many null values and might be unusable as in most cases 30% null values is kind of of a threshold although in some fields like social sciences it might exceed that and still be usable. We could try to fill the null values with the average or the most frequent value for example. We will go with the most frequent value in this case to keep values as integers.

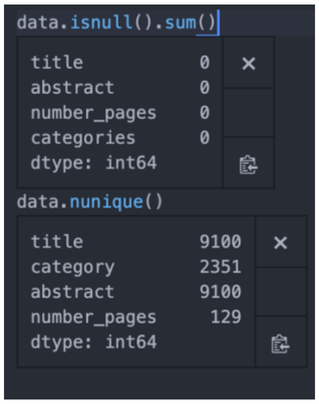
After some more cleaning, to make sure all values are in fact convertible to integer, we fill the null values with the mode of the number of pages.

	title	category	abstract	number_pages
0	Convergence of the discrete dipole approximati...	[physics.optics, physics.comp-ph]	We performed a rigorous theoretical converge...	23
1	The discrete dipole approximation: an overview...	[physics.optics, physics.comp-ph]	We present a review of the discrete dipole a...	36
2	The affine part of the Picard scheme	[math.AG, math.KT]	We describe the maximal torus and maximal un...	10
3	M-regularity of the Fano surface	[math.AG]	Let (A, Θ) be a principally polarised ...	5
4	K_0-theory of n-potents in rings and algebras	[math.KT, math.RA]	Let $n \geq 2$ be an integer. An \emph{n-p...	10
...	...	...	...	...
909968	Discrete-query quantum algorithm for NAND trees	[quant-ph]	Recently, Farhi, Goldstone, and Gutmann gave...	2
909969	Quantum Mechanics in Terms of Realism	[quant-ph]	.We expound an alternative to the Copenhagen...	88
909970	Quantum Games and Quantum Strategies	[quant-ph]	We investigate the quantization of non-zero ...	4
909971	Unconditionally Secure Quantum Coin Tossing	[quant-ph]	In coin tossing two remote participants want...	7
909972	The averaging of non-local Hamiltonian structu...	[solv-int, nlin.SI]	We consider the m-phase Whitham's averagin...	40

The next steps will likely involve more computational power so we should work with a subset of the dataset to avoid 8/9 hours waiting times for some processes that follow. Let us keep 1% of the data which is around 10000 data points.

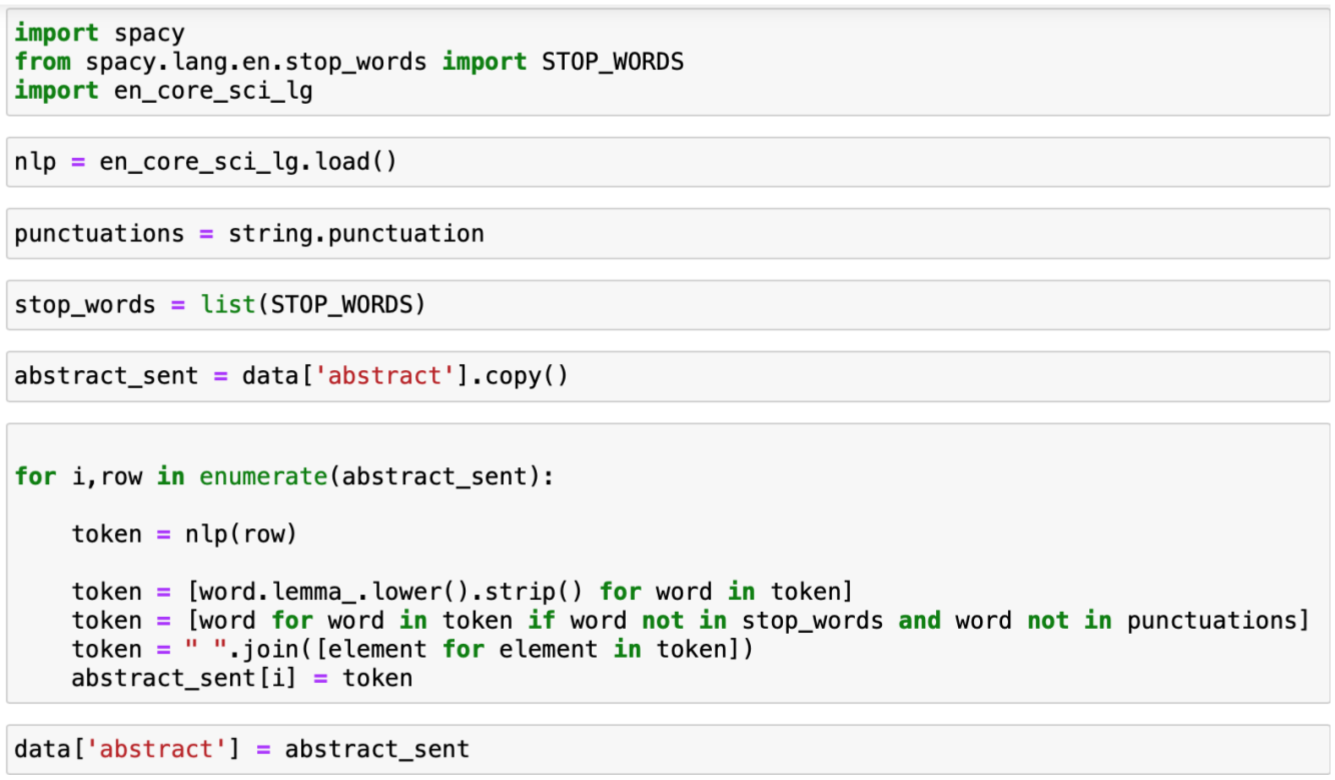
Simple analytics like the number of null values and unique values should be performed at this point to get an idea of what we are dealing with.

As we can see from the image below there no null-values in the chosen features and at least in this sample of the dataset there are also almost a 100% unique abstracts and titles so no duplicate cleaning is needed.



```
data.isnull().sum()
title          0
abstract       0
number_pages   0
categories     0
dtype: int64
data.nunique()
title          9100
category       2351
abstract       9100
number_pages    129
dtype: int64
```

We will continue cleaning the data and transforming it into useful features. Since we are working with text features, using nlp (Natural Language Processing) is a good idea, to turn the abstract and the category columns into usable features for the machine learning models we will use later. A python library that allows us to do that is spacy, which can help us make sense of text features by tokenizing them.



```
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
import en_core_sci_lg
```

```
nlp = en_core_sci_lg.load()
```

```
punctuations = string.punctuation
```

```
stop_words = list(STOP_WORDS)
```

```
abstract_sent = data['abstract'].copy()
```

```

for i,row in enumerate(abstract_sent):

    token = nlp(row)

    token = [word.lemma_.lower().strip() for word in token]
    token = [word for word in token if word not in stop_words and word not in punctuations]
    token = " ".join([element for element in token])
    abstract_sent[i] = token
```

```
data['abstract'] = abstract_sent
```

As we see in the picture above we import the libraries, load the model (nlp), which we use the one specific for scientific texts (en_core_sci_lg), we import stop words and string punctuations to remove from the abstracts, and in the last cell we proceed to do that after tokenizing and lemmatizing each abstract.

After this we can vectorize the abstracts. It should be noted that this point it is probably a better decision to analyze each pertinent columns individually with respect to current trends.

There are some choices for vectorizing the "cleaned" abstracts but a good one is Term Frequency-Inverse Document Frequency, which can help us give less importance to common words that might appear in several abstracts.

In the first try without setting a maximum features limit, there were over 78000 features created. So, we need to set a more reasonable number to avoid overfitting and improving performance of the model we will use. Maybe for this case 5000 is a more reasonable number. So, we will go with it.

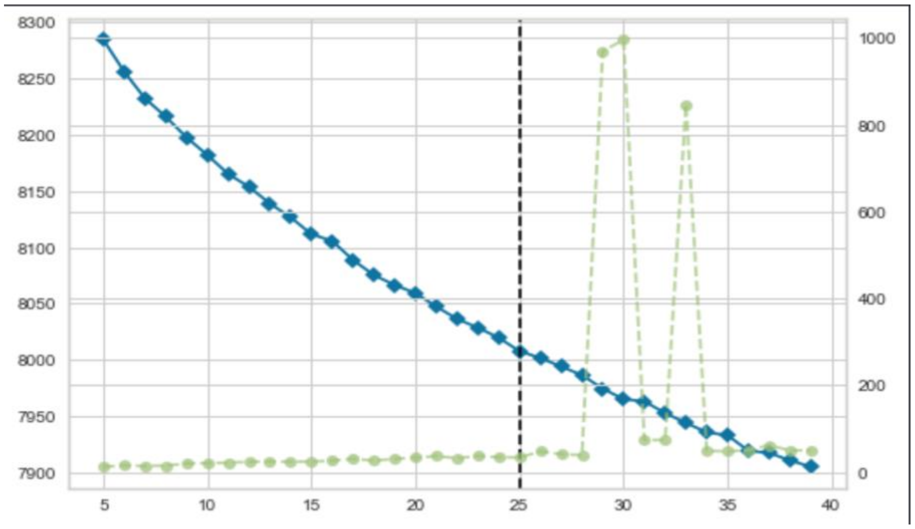
We will do the same for the column 'categories', and for the moment let's discard the column 'title' because it is likely to contain similar information to 'abstracts' and not add anything new.

This column contains abbreviatures for each science field and therefore we can probably use a different approach Bag-of-words. We do not need to set stop words and just a small cleaning of symbols like [or , are needed in order to separate the terms and apply this feature extraction or vectorization. Regex was used in order to keep the terminology consistent and Math.AG for example to be considered a word by the vectorizer. The number of features created was less than the one for abstracts due to the method used. There were only 157 features created.

4 - Dimensionality Reduction and clustering

For dimensionality reduction we have several options, (according to the course script) we have principal component analysis, multi-dimensional scaling and local linear embedding for example. We will start with the vectorized abstracts. Since we are dealing with a large sub-set of the dataset, we can probably exclude MDS since it takes more computational power and time than this machine can handle. LLE seems like a good idea because this is a large dataset and it requires less computational power, but we are still not sure if we have a lot of outliers/noise or not, which affects LLE's performance. We are also not sure about the shape of the data and are aiming just at reducing the number of features. So, we will try PCA first. After applying it with a 95% variance to the vectorized abstracts we get 2702 features left. We can now ponder upon what model to use for the abstracts. After performing the KElbow method with kmeans we get a possible value for the number of clusters, which is 25. as we can see in the figure below.

The 'elbow' is not very clear in the graph, but it is our best estimate so far. So, we will have to work with it, as 25 is probably a reasonable number on science's current topics. Also, we have to be realistic and expect some overlap in the topics so some clusters will likely not be very clear and sub-clusters might appear.

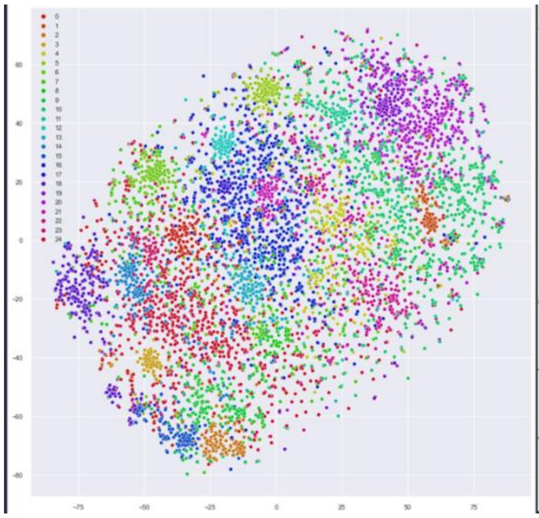
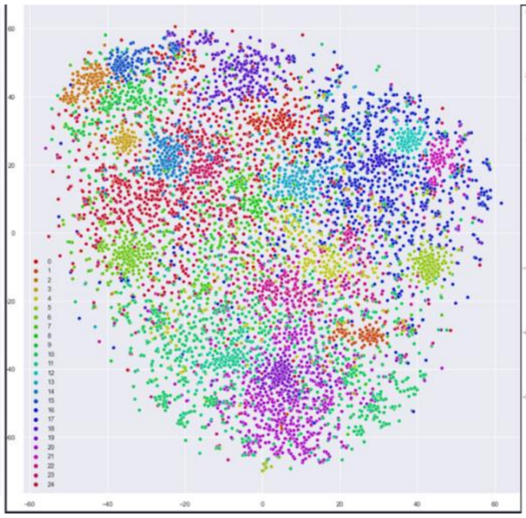
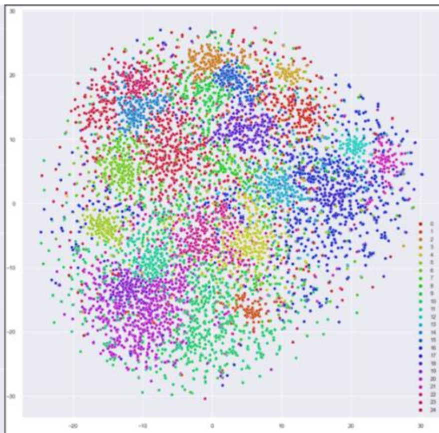


The next step is then using the model to predict the labels of the abstracts.

In order to be able to visualize this multidimensional data into a lower dimension we need further dimensionality reduction. There are some options. We already used PCA. We could try MDS,T-SNE or UMAP for example. Let's try both T-SNE and UMAP as MDS is just too expensive, computing wise.

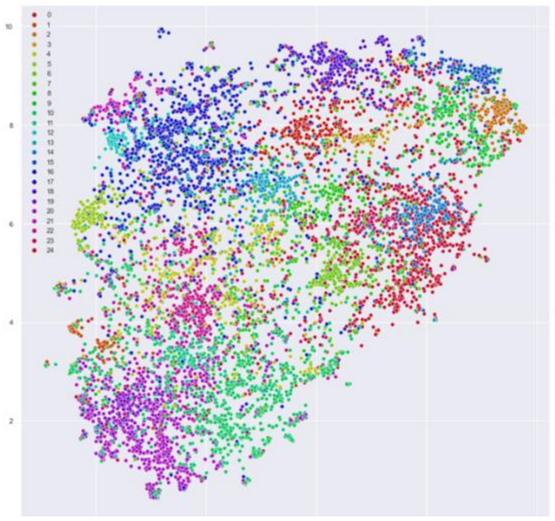
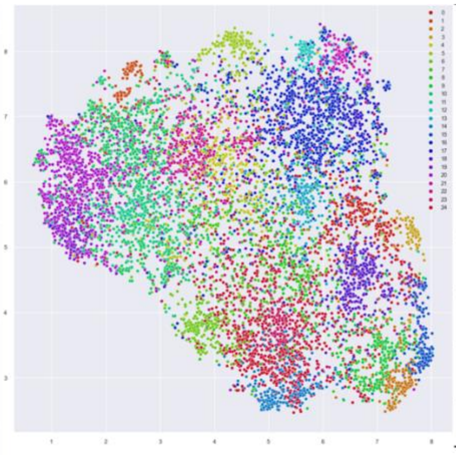
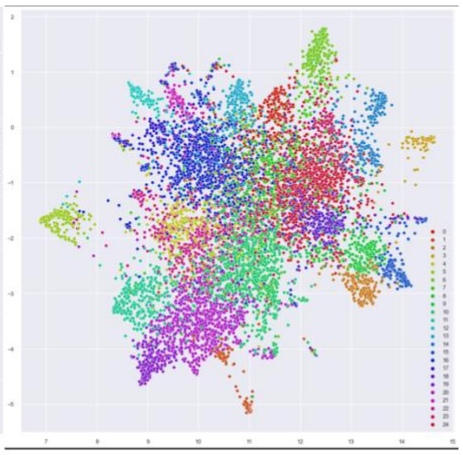
We will try to map the data into a 2-dimensional form, although a 3-dimensional form would also be possible. We will start with T_SNE, try different perplexity values and different learning rates and then do the same to UMAP. Leaving the learning rate automatic and perplexity 40 we get the first plot.

We can see some well-defined clusters and some more dispersed, but it seems that the result is not that bad, and we can have an initial view on the data. We can try to tune in the hyperparameters and try to get a clearer result in the result of a more homogeneous view. Increasing the learning rate from 200 to 1000 and the perplexity to 100 and then to 400, we get the second graph and the third respectively.

We can see the difference and now it is time to compare it to UMAP.

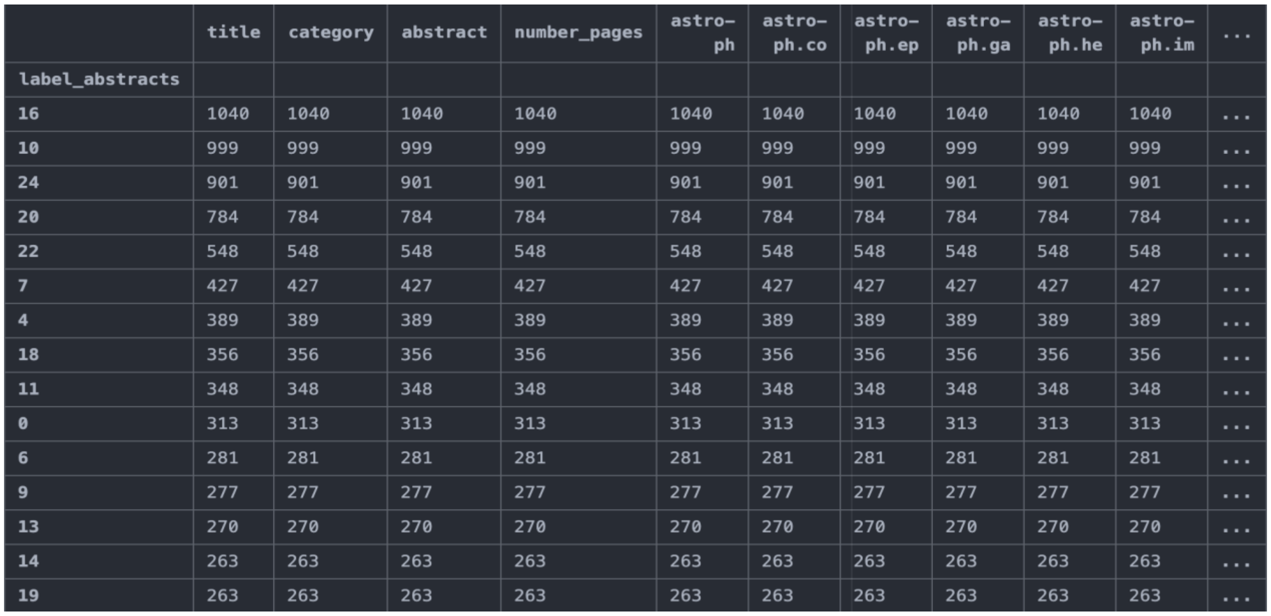
We try UMAP with default parameters first, n_neighbors=100 and min_dist = 0.2 second as well as the third, which has chebyshev's distance as the metric used and provides an extra interesting view.

UMAP provides us with more homogeneous clusters and a clearer view of the data than T-SNE.

5- Analysis

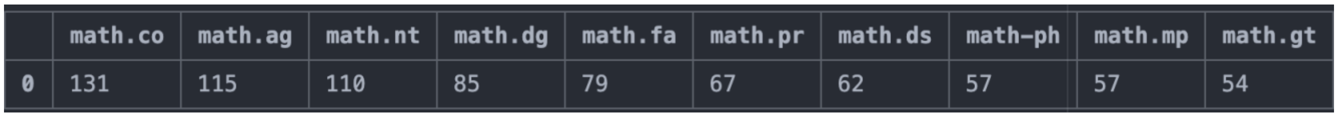
After getting this cluster information and visualization it is clear that there are some clusters bigger than others. We should not forget that the goal of this written work is to provide an analysis and recommendation to the company on the current topic trends in science, so we should focus on the "bigger" clusters. In order to do this, grouping the data points by cluster and checking the number of data points in each one is necessary and afterwards check what are the similarities between the data points that belong to those clusters and therefore use the information we got from categories to "translate" them.



	title	category	abstract	number_pages	astro-ph	astro-ph.co	astro-ph.ep	astro-ph.ga	astro-ph.he	astro-ph.im	...
label_abstracts											
16	1040	1040	1040	1040	1040	1040	1040	1040	1040	1040	...
10	999	999	999	999	999	999	999	999	999	999	...
24	901	901	901	901	901	901	901	901	901	901	...
20	784	784	784	784	784	784	784	784	784	784	...
22	548	548	548	548	548	548	548	548	548	548	...
7	427	427	427	427	427	427	427	427	427	427	...
4	389	389	389	389	389	389	389	389	389	389	...
18	356	356	356	356	356	356	356	356	356	356	...
11	348	348	348	348	348	348	348	348	348	348	...
0	313	313	313	313	313	313	313	313	313	313	...
6	281	281	281	281	281	281	281	281	281	281	...
9	277	277	277	277	277	277	277	277	277	277	...
13	270	270	270	270	270	270	270	270	270	270	...
14	263	263	263	263	263	263	263	263	263	263	...
19	263	263	263	263	263	263	263	263	263	263	...

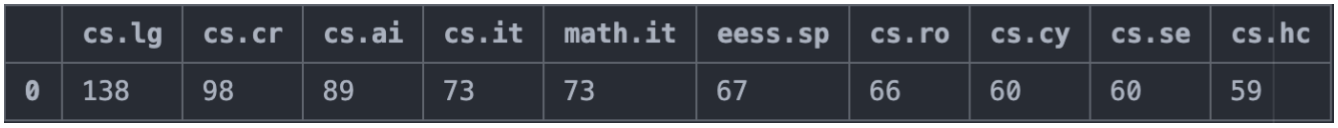
We can see from the picture above that clusters with labels 16,10,24,20 and 22 contain more datapoints associated with them than the rest. So, we should focus on these ones. It should be noted that we are working with a subset of a 900 thousand datapoints so if we would have access to a more powerful machine and analyze all the datapoints, these numbers could vary. Nevertheless, we took a good sample and there is a clear difference in the amount of datapoints associated with each label.

We will examine each of these 5 clusters and see what insights we can take from them. The biggest cluster has the label 16. There are 1040 datapoints in this cluster. If we check the number of values (True or 1) in each of the bag of words we got from analyzing categories, we get this figure.



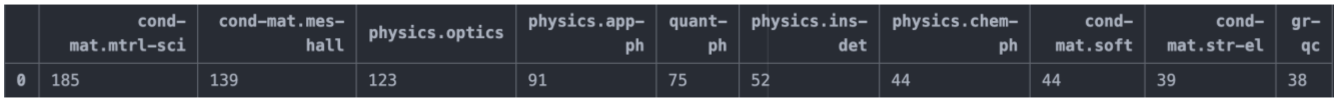
	math.co	math.ag	math.nt	math.dg	math.fa	math.pr	math.ds	math-ph	math.mp	math.gt
0	131	115	110	85	79	67	62	57	57	54

With the help of the "dictionary" for this abbreviatures on https://arxiv.org/archive/, we can see that various branches of math are prevalent in this first cluster. Namely Combinatorics, algebraic geometry, number theory and differential geometry as the first 4. Down the line we can see that there are some papers that overlap with High Energy Physics - Experiment or Computer science logic. The maximum number of pages in this cluster is 238 and minimum 2. Being the average around 20 pages.



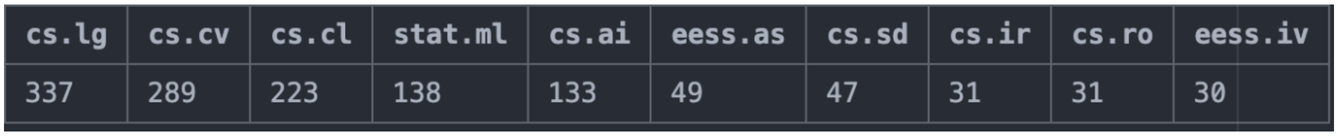
	cs.lg	cs.cr	cs.ai	cs.it	math.it	eess.sp	cs.ro	cs.cy	cs.se	cs.hc
0	138	98	89	73	73	67	66	60	60	59

In the second cluster, which has label 10, we can see that computer science is predominant, subsets like Machine Learning, Cryptography and Security, Artificial Intelligence and Information Theory. Math is also present in the information theory field or Signal processing in around 130 papers. Statistics Machine Learning and Statistics applications also overlap with computer science in this cluster. Both this cluster and the previous one, if we look at the abstracts graph in the last chapter, also overlap with nearby clusters which is interesting so this could explain some of this other fields overlapping in our analysis at the moment. This second cluster has an average page length of 15.6 and max/min of 224/2.



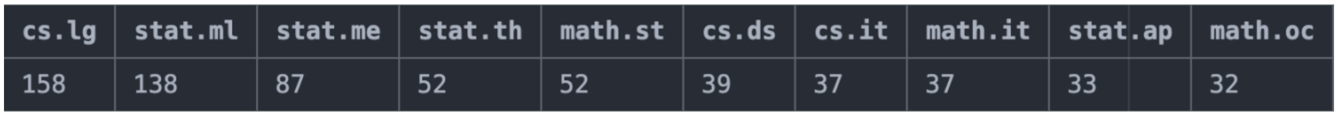
	cond-mat.mtrl-sci	cond-mat.mes-hall	physics.optics	physics.app-ph	quant-ph	physics.ins-det	physics.chem-ph	cond-mat.soft	cond-mat.str-el	gr-qc
0	185	139	123	91	75	52	44	44	39	38

In this third cluster, which has label 24, we have condensed matter and physics as its main categories. Materials Science,Mesoscale and Nanoscale physics as the most common followed by optics, applied physics and quantum physics. So, there are physics topics and Condensed Matter topics overlapping and "put" into the same cluster by the abstract analysis and the model. We have an average of 15 pages per paper, maximum 127 and minimum 2.



cs.lg	cs.cv	cs.cl	stat.ml	cs.ai	eess.as	cs.sd	cs.ir	cs.ro	eess.iv
337	289	223	138	133	49	47	31	31	30

Fourth biggest cluster label 20, as we see from the graphs on the previous chapter and the most common categories in this cluster, overlaps with cluster with label 10. The most common related categories are subsets of computer science like Machine learning, computer vision and pattern recognition, computation and language as well as artificial intelligence. There is also the statistics aspect of machine learning and Electrical Engineering and System Science Audio and speech processing. As we can see this cluster is heavily directed towards artificial intelligence and the components that make it possible, so it is easy to understand why there is an overlap and relationship with other areas of computer science present in the label 10 cluster. It also allows us to better understand the graphs in the previous chapter. Page length mean is 13, 90 maximum and 2 minimum.



cs.lg	stat.ml	stat.me	stat.th	math.st	cs.ds	cs.it	math.it	stat.ap	math.oc
158	138	87	52	52	39	37	37	33	32

Machine learning comes again as the most common topic in this cluster (22) which is natural as this cluster overlaps with clusters 20 and 10 in the graph. The differences lie in the fact that this cluster's topics are more directed at the field of statistics. We have machine learning, methodology, statistics theory and applications for example. There is also Statistics theory in Math as well as Computer Science's Data structures and algorithms and Information theory present. Average page length of 18, maximum of 272 and minimum of 3.

Based on the information we obtained, it should be clear to the decision makers of the company, that, if the company, indeed, intends to direct its activity towards a more academic and research direction the main topics that should be look at should be topics and areas present in the 5 clusters described in this chapter as they comprise nearly 50% of all current topics in science based on our analysis. A deeper analysis should be made depending on the other goals of the company, but this analysis should form a pertinent and clear starting point for a discussion and eventual decision on the path to take.

6 - Conclusion

In order to analyze the data and reach the insights discussed in the last chapter, some assumptions and compromises had to be made, so every insight although informative and worthy of consideration should always be taken with a pinch of salt and wary of its limitations. These limitations were mostly hardware related, which "forced" me to use a subset of the data instead of the data as a whole. Also, when it comes to the number of pages, about 1/3 of the values were filled with the most common occurrence. So, when considering this metric, all conclusions and discussions should not dismiss this fact. Even though we had some limitations which might have had an influence on what was accomplished, still some insights and views were able to be transmitted in an understandable way to the reader and hopefully will be of some help to the company when it comes to discussing this objective and eventual decision making.

I - Bibliography

Cornell University, Tricot, J., Devrishi, & Maltzan, B. (n.d.). *arXiv Dataset*. Kaggle. Retrieved July 10, 2022, from https://www.kaggle.com/datasets/Cornell-University/arxiv

Dy, J. G., & Brodley, C. E. (2004). Feature selection for unsupervised learning. Journal of Machine Learning Research, 5 (Aug), 845—889.

Kaufman, L., & Rousseeuw, P. J. (2009). Finding groups in data: An introduction to cluster analysis (Vol. 344). John Wiley & Sons.

Harris, C. R., et al (2020). Array programming with NumPy. *Array Programming with NumPy*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/mcse.2007.55

Müller, A. C., & Guido, S. (2016). Introduction to machine learning with Python: A guide for data scientists. O'Reilly Media.

Pedregosa Et Al. (2011). Scikit-learn Machine Learning in Python. *JMLR*, *12*, 2825–2830. Sainburg, T., McInnes, L., & Gentner, T. Q. (2021). Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, *33*(11), 2881–2907. https://doi.org/10.1162/neco_a_01434

The Pandas Development Team. (2020). *pandas-dev/pandas: Pandas* (latest) [Software]. Zenodo. https://doi.org/10.5281/zenodo.3509134

Waskom, M. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60). https://doi.org/10.21105/joss.03021