Data.dλy

Ángel Alvarado

Data Engineer TalkingPoints

Co-Founder MOLANCO Data Engineering

Co-Founder:
getbeany.com
AgTech

Mentor Techqueria

#DataDays

Data.dλy

TalkingPoints

Messaging Apps that connects millions of teachers and families in 145 languages

- Two-way messaging translation.
- Video translation captioning

Unidos compartiendo y aprendiendo

#DataDays

Demystifying NLP with a use case:
from unigrams, vectors and embeddings to BERT models, HuggingFace and OpenAI

Slides: https://bit.ly/DataDaysNLP

Ángel Alvarado

angel@molanco.com

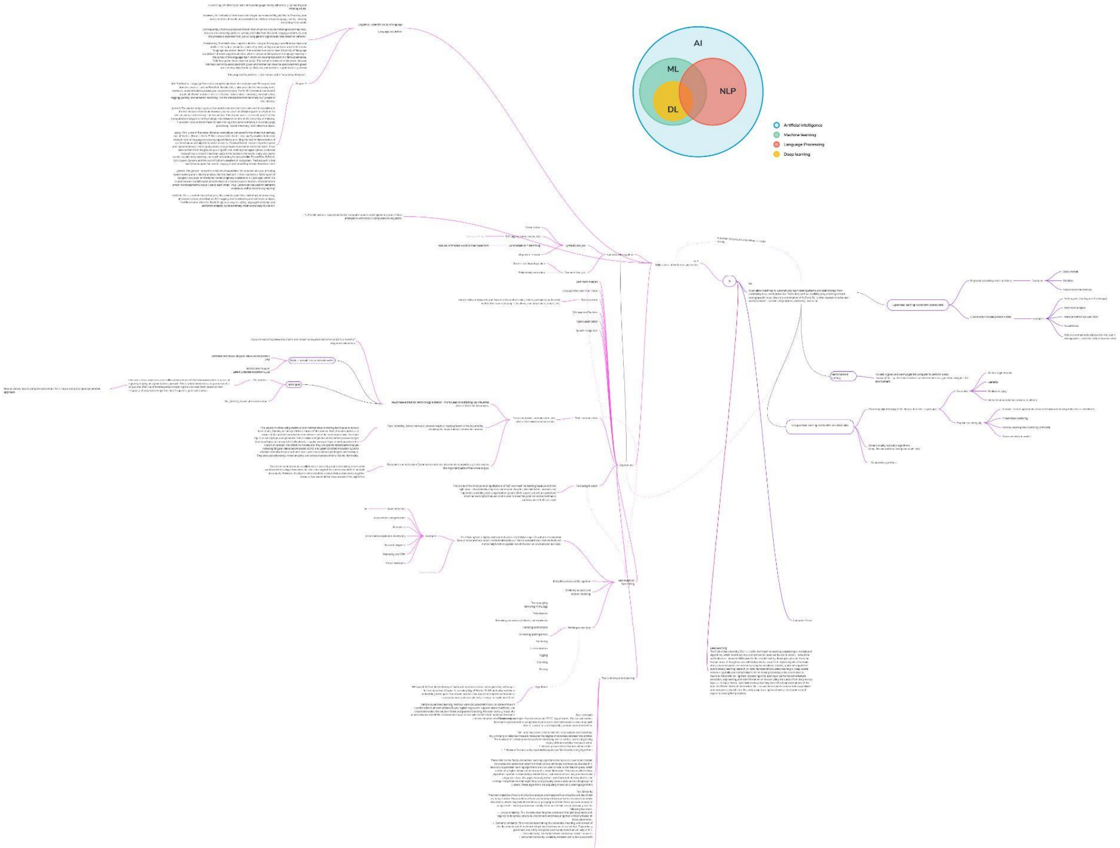
Agenda

- NLP basics
  - Big picture
  - Bag of words
  - Unigrams, Bigrams & N-grams
  - Vectors, Embeddings
  - Clustering
- Using HuggingFace and OpenAI
  - Understanding text analysis with HuggingFace’s models
  - Understanding OpenAI’s API
- Deploying models: a fast and quick approach

NLP BASICS

Big picture:
Where does NLP fall within AI?

https://bit.ly/AI-NLP-ML

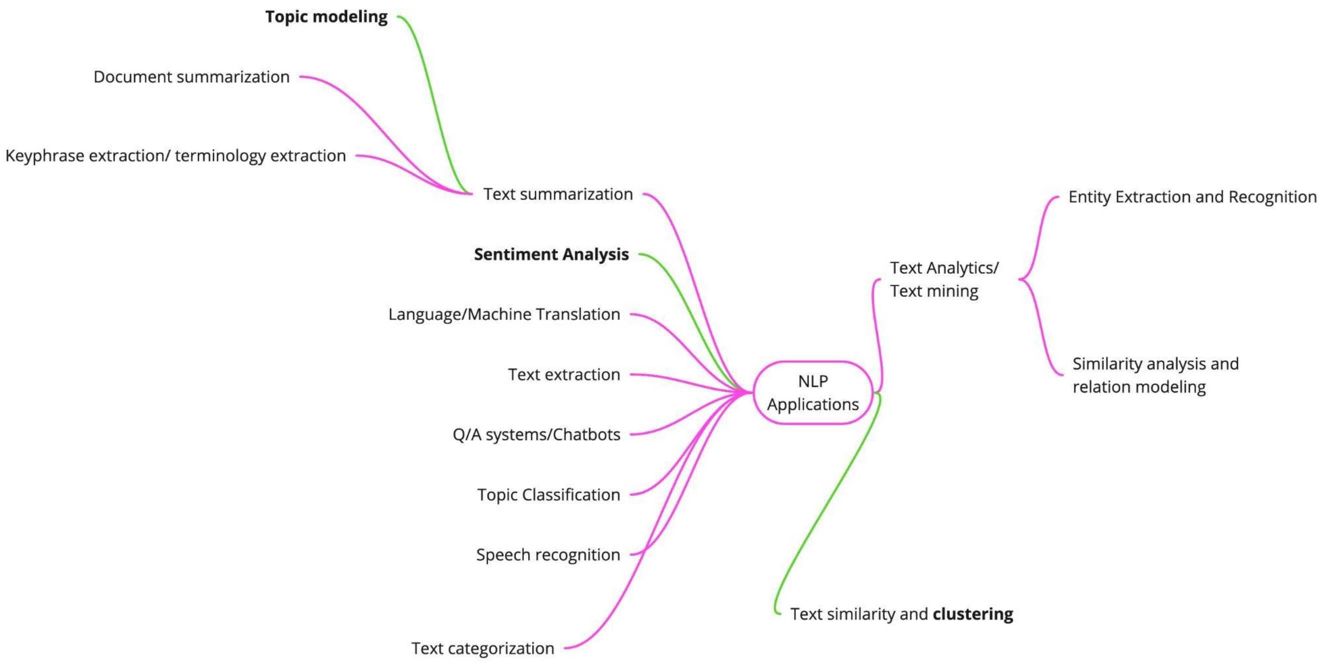


NLP

Makes sense of written or spoken text.

NLP is defined as a specialized field of computer science and engineering and artificial intelligence with roots in computational linguistics.

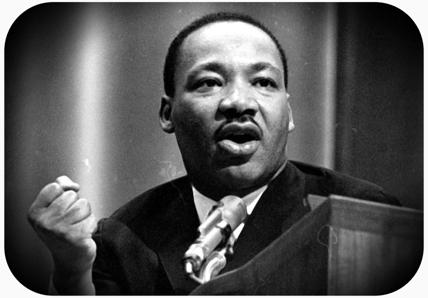
Today’s use case



Document A:

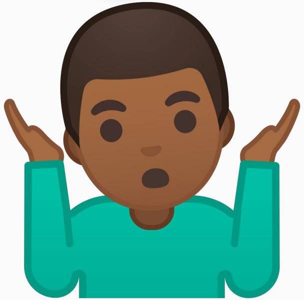
"We've got some difficult days ahead. But it really doesn't matter with me now because I've been to the mountaintop. And I don't mind. Like anybody I would like to live a long life. Longevity has its place, but I'm not concerned about that now. I just want to do God's will, and He's allowed me to go up to the mountain, and I've looked over and I've seen the Promised Land. I may not get there with you, but I want you to know tonight that we as a people will get to the Promised Land. So I'm happy tonight, I'm not worried about anything, I'm not fearing any man. Mine eyes have seen the glory of the coming of the Lord."

Memphis, Tennessee, April 3, 1968.
Martin Luther King, Jr.



Document B:

"This is a second phrase and I'm a worried very worried fearing man"



Why Machine Learning?

ML algorithms applied to text cleary have a speed advantage

Analysing 100 docs/survey responses of "Artisanal data"*

Suppose we are interested in classifying **100 open-ended question answers**. We want to assign every text to one and only one category of documents.

Bell Number for **2** docs {A,B} is **2**: AB and A,B

Bell Number for **3** docs {A,B,C} is **5**: ABC; A,BC; AB,C; A,BC

Bell Number for **100** docs: **10 ^ 111.68**

Scientist estimate there are approx 10^80 atoms in the known universe.

∴ Computational methods can help us explore a massive space of possible organizations.

*Artisanal data (per Hanna Wallach): A reference to a dataset's smaller size and careful curation of text collections

Augmenting humans reading ability

Text Analysis serve a **qualitative** task: extract understanding and analysis from collections of text.

Automated methods are useful because their statistical and algorithmic foundations help humans read, organize and analyze documents BUT it would be a mistake to replace the need for careful and close readings of text or otherwise obviating the need for human analysis. Rather, computer-assisted text analysis augments our reading ability. **New text analysis methods help us read differently, not avoid reading at atll. This amplification of human effort improves the analyst's ability to discover interesting organizations, measure key qualities of interest, estimate causal effects and make predictions**

TEXT AS DATA: GRIMMER | ROBERTS | STEWARTc

Augmenting humans reading ability

“ Combining data, statistical algorithms, and substantive knowledge lead to deeper and richer theoretical insights - contrast with recent pronouncements that big data would change how social scientist operate and obviate the need for theory development. Writers have proclaimed that big data and ML algorithms would lead to ‘The end of Theory: The Data Deluge Makes Specific Methods Obsolete’ (Anderson, 2008). **The argument was that large datasets would eliminate the need for theoretical thinking because we could replace any work that theorizing does in a project with more data. These pronouncements are *wrong*, because the overstate what any algorithm can provide. There is an amount of interpretive work that is essential to the functioning of these approaches that simple cannot be automated.”**

TEXT AS DATA: GRIMMER | ROBERTS | STEWARTc

Bag of words

Unigrams, Bigrams & N-grams

Vectors, Embeddings,

Clustering

High dimensional data

"We've got some difficult days ahead. But it really doesn't matter with me now because **I've been to the mountaintop**. And I don't mind. Like anybody I would like to live a long life. Longevity has its place, but I'm not concerned about that now. I just want to do God's will, and He's allowed me to go up to the mountain, and I've looked over and I've seen the Promised Land. I may not get there with you, but I want you to know tonight that we as a people will get to the Promised Land. So I'm happy tonight, I'm not worried about anything, I'm not fearing any man. Mine eyes have seen the glory of the coming of the Lord."

Memphis, Tennessee, April 3, 1968. Martin Luther King, Jr.

Computers are good at understanding low-dimensional data

Bag of words

A bag of words is a representation of text that describes the occurrence of words within a document

- Order is not important
- Just count occurences
- ML models work with numerical data not text

Document B:

- Worried; 2 times

Document A:

- Worried; 1 time

Bag of words: Step 1 - Tokenize

Each individual word is a token and the process a document into its constituent words is Tokenization.

BoW n-grams: Step 2 - order set of n-words as our ***features***

Unigrams a set of one word

difficult	doesn	...	second	seen	sick	talk	threats	tonight	ve	want	white	worried
1	1	...	0	2	1	1	2	2	4	2	1	1
0	0	...	1	0	0	0	0	0	0	0	0	2

Bigrams a set of two words

	ahead really	allowed mountain	anybody like	began say	brothers don	coming lord	concerned just	days ahead	difficult days	doesn matter	...	tonight worried	ve got	ve looked	ve mountaintop	ve seen	want god	want know	white brothers	worried fearing	worried worried
0	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	1

Trigrams a set of three words

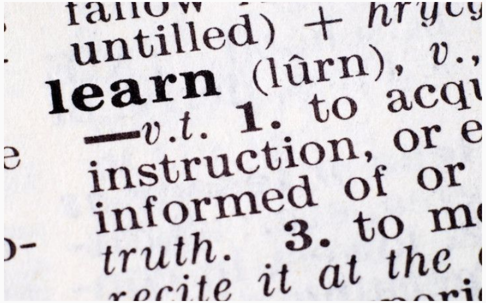
	ahead really doesn	allowed mountain ve	anybody like live	began say threats	brothers don know	concerned just want	days ahead really	difficult days ahead	doesn matter ve	don know happen	...	tonight worried fearing	ve got difficult	ve looked ve	ve mountaintop don	ve seen promised	want god allowed	want know tonight	white brothers don	worried fearing man	worried worried fearing
0	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	1

Bag Of words: Step 3 - Reduce complexity

Lowercase
Remove punctuation
Remove stop words: and, the, that

tonight worried	ve got	ve looked	ve mountaintop
1	1	1	1
0	0	0	0

Lemmatize: Family, families, family’s -> family



Stemming: family, families, family’s -> famili

Bag of words: Document-feature Matrix

	ahead	allowed	anybody	began	brothers	coming	concerned	days	difficult	doesn	...	second	seen	sick	talk	threats	tonight	ve	want	white	worried
0	1	1	1	1	1	1	1	1	1	1	...	0	2	1	1	2	2	4	2	1	1
1	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	2

Generally it turns out a sparse matrix (mostly zeroes)

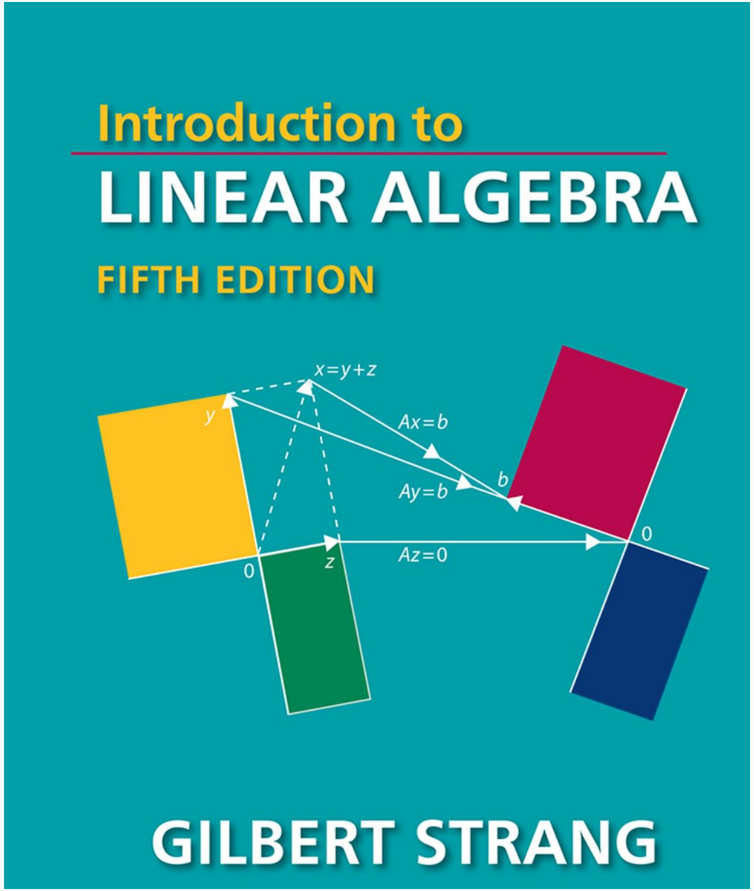
How do we use this?

- For now you can create simple word clouds
- Could attempt to draw a common topic (i.e. topic modeling) but you can have a hard time.

Bag of words: Step 1 - Step 2 - Step 3

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

sentence_1="And then I got into Memphis, and some began to say the threats, or tal
sentence_2="This is a second phrase and I'm a worried very worried fearing man"
CountVec = CountVectorizer(ngram_range=(2,2), # to use bigrams ngram_range=(2,2)
                           stop_words='english')
#transform
Count_data = CountVec.fit_transform([sentence_1, sentence_2])
```


Introduction to
LINEAR ALGEBRA
FIFTH EDITION
x=y+z
y
Ax=b
b
Ay=b
0
z
Az=0
0
GILBERT STRANG

Vector Space Model

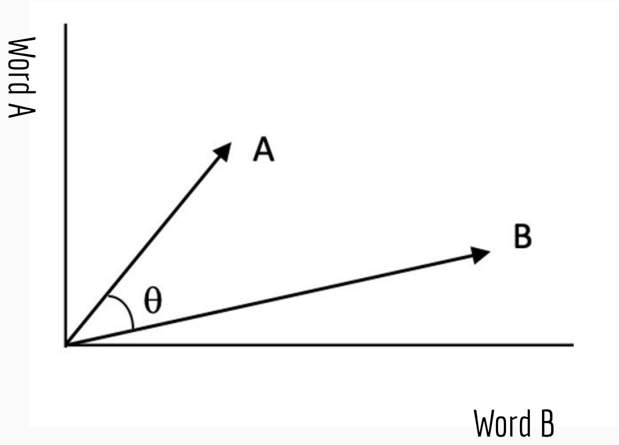
...	second	seen	sick	talk	threats	tonight	ve	want	white	worried
...	0	2	1	1	2	2	4	2	1	1
...	1	0	0	0	0	0	0	0	0	2

Each row is a vector - linear algebra:
∴ We can measure similarity, distances and way more

Cosine Similarities: how close are document A and B?

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

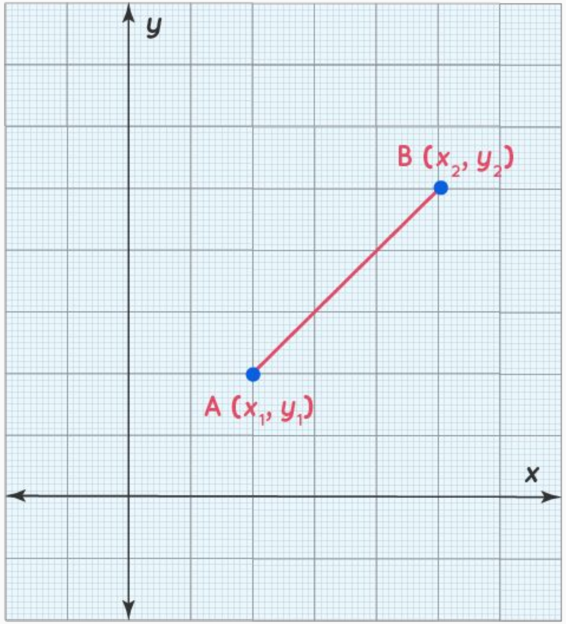
Cosine Similarities: how close are document A and B



Document A and B's cosine similarity: **0.08**
(in a scale -1,1)

```
from sklearn.metrics import pairwise_distances
from scipy.spatial.distance import cosine
pairwise_similarity = 1 - pairwise_distances(Count_data, metric = 'cosine')
```

Euclidean distance: If document A and B were points in space



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

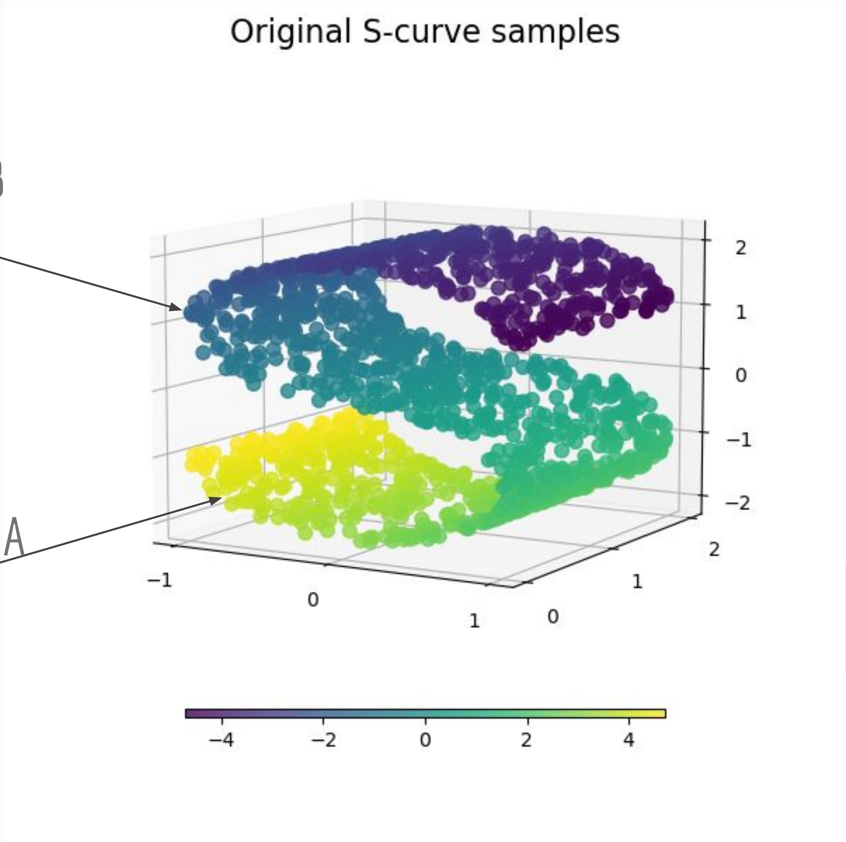
Tf-Idf (term frequency-inverse document frequency). Weighing / Dummy Variables

Let's look at our two documents, which ones are likely not to bring any substance to the document? Which ones appear too many times and don't seem that important?

TF-IDF is intended to reflect how relevant a term is in a given document.

Clustering

Document B



Document A

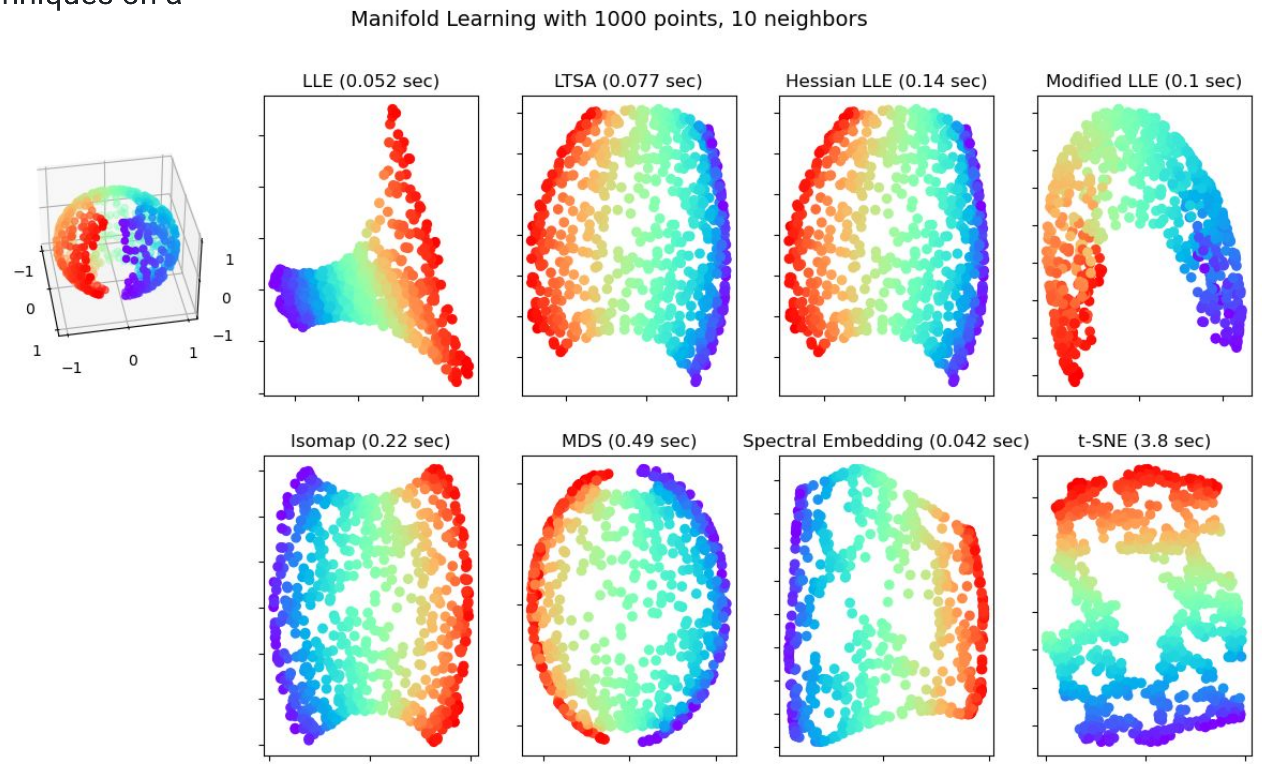
https://scikit-learn.org/stable/modules/clustering.htm

- K-Means
- Affinity propagation
- Mean-shift
- Spectral clustering
- …

Hyper-parameter tuning

Topic Modeling

Manifold learning techniques on a spherical data-set



https://scikit-learn.org/

t-SNE is a tool to visualize high-dimensional data.

Dense vectors

Embeddings

What's BERT and GPT3 models?

OpenAI's GPT3

Models (DaVinci, etc)

Outpainting

From one-hot-encoding vectors to dense vectors

Dog = (0, 0, 0, 1) - One-hot encoding vector. (What we've done with ngrams so far, basically)

Dog = (0.3, 0,5, 0, 1.....) - Word embedding (dense vector in a low-dimensional space) using Word2vec

- **But how are these numbers/low-dimensional vectors calculated?**

Word Embeddings

WIKIPEDIA
The Free Encyclopedia

- What's it about?
- What does the word mean ?
- Which texts are similar to each other based on that word?

Banana Vector

BANANA.VECTOR

```
array([2.02280000e-01,  -7.66180009e-02,   3.70319992e-01,
        3.28450017e-02,  -4.19569999e-01,   7.20689967e-02,
       -3.74760002e-01,   5.74599989e-02,  -1.24009997e-02,
        5.29489994e-01,  -5.23800015e-01,  -1.97710007e-01,
       -3.41470003e-01,   5.33169985e-01,  -2.53309999e-02,
        1.73800007e-01,   1.67720005e-01,   8.39839995e-01,
        5.51070012e-02,   1.05470002e-01,   3.78719985e-01,
        2.42750004e-01,   1.47449998e-02,   5.59509993e-01,
        1.25210002e-01,  -6.75960004e-01,   3.58420014e-01,
        # ... and so on ...
        3.66849989e-01,   2.52470002e-03,  -6.40089989e-01,
       -2.97650009e-01,   7.89430022e-01,   3.31680000e-01,
       -1.19659996e+00,  -4.71559986e-02,   5.31750023e-01], dtype=float32)
```

Pretrained (Word) Embeddings

- Google Word2Vec (spaCy has a nice implementation)
- Stanford NLP Group - GloVe
- Facebook - fastText

These are trained with ***hundreds of millions*** of ***tokens***. Words are **independent** of their context

Word2Vec Embeddings

Word2Vec : Let's look at an embedding using Spacy's Word2Vec: https://spacy.io/usage/spacy-101#vectors-similarity

Word vectors and similarity NEEDS MODEL

Similarity is determined by comparing **word vectors** or "word embeddings", multi-dimensional meaning representations of a word. Word vectors can be generated using an algorithm like word2vec and usually look like this:

BANANA.VECTOR

```
array([2.02280000e-01,  -7.66180009e-02,   3.70319992e-01,
       3.28450017e-02,  -4.19569999e-01,   7.20689967e-02,
      -3.74760002e-01,   5.74599989e-02,  -1.24009997e-02,
       5.29489994e-01,  -5.23800015e-01,  -1.97710007e-01,
      -3.41470003e-01,   5.33169985e-01,  -2.53309999e-02,
       1.73800007e-01,   1.67720005e-01,   8.39839995e-01,
       5.51070012e-02,   1.05470002e-01,   3.78719985e-01,
       2.42750004e-01,   1.47449998e-02,   5.59509993e-01,
       1.25210002e-01,  -6.75960004e-01,   3.58420014e-01,
       # ... and so on ...
       3.66849989e-01,   2.52470002e-03,  -6.40089989e-01,
      -2.97650009e-01,   7.89430022e-01,   3.31680000e-01,
      -1.19659996e+00,  -4.71559986e-02,   5.31750023e-01], dtype=float32)
```

Word2Vec - who was it trained?

spacy.io/models/en#en_core_web_lg

Out now: spaCy v3.4 USAGE MODELS

INES

_sm
_md
_lg
_trf

nål

English pipeline optimized for CPU. Components: tok2vec, tagger, parser, senter, ner, attribute_ruler, lemmatizer.

LANGUAGE	EN English
TYPE	CORE Vocabulary, syntax, entities, vectors
GENRE	WEB written text (blogs, news, comments)
SIZE	LG 560 MB
COMPONENTS ?	`tok2vec`, `tagger`, `parser`, `senter`, `attribute_ruler`, `lemmatizer`, `ner`
PIPELINE ?	`tok2vec`, `tagger`, `parser`, `attribute_ruler`, `lemmatizer`, `ner`
VECTORS ?	514k keys, 514k unique vectors (300 dimensions)
DOWNLOAD LINK ?	en_core_web_lg-3.4.1-py3-none-any.whl
SOURCES ?	OntoNotes 5 (Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, Ann Houston) ClearNLP Constituent-to-Dependency Conversion </> (Emory University) WordNet 3.0 (Princeton University) Explosion Vectors (OSCAR 2109 + Wikipedia + OpenSubtitles + WMT News Crawl) </> (Explosion)
AUTHOR	Explosion
LICENSE	MIT

SOTA (State of the Art) / Large Language Models

Contextualized word embeddings

- Google - BERT (Bidirectional Encoder Representations from Transformers)
- OpenAI - GPT3 (Generative Pre-trained Transformer 3)

The AI community building the future.

- Deep Learning to learn embeddings
- **Context-specific** embeddings (know relation to other words around it)
- Based on Recurrent Neural Networks (**RNN**) or **Transformer** architectures
- Trained with **billions** of parameters

OpenAI

SOTA/LL Models - BERT

HuggingFace's Sentence-Transformers library

- You can use this framework to compute sentence / **text embeddings for more than 100 languages.** These **embeddings can then be compared e.g. with cosine-similarit**y to find sentences with a similar meaning. This can be useful for semantic textual similar, semantic search, or paraphrase mining.

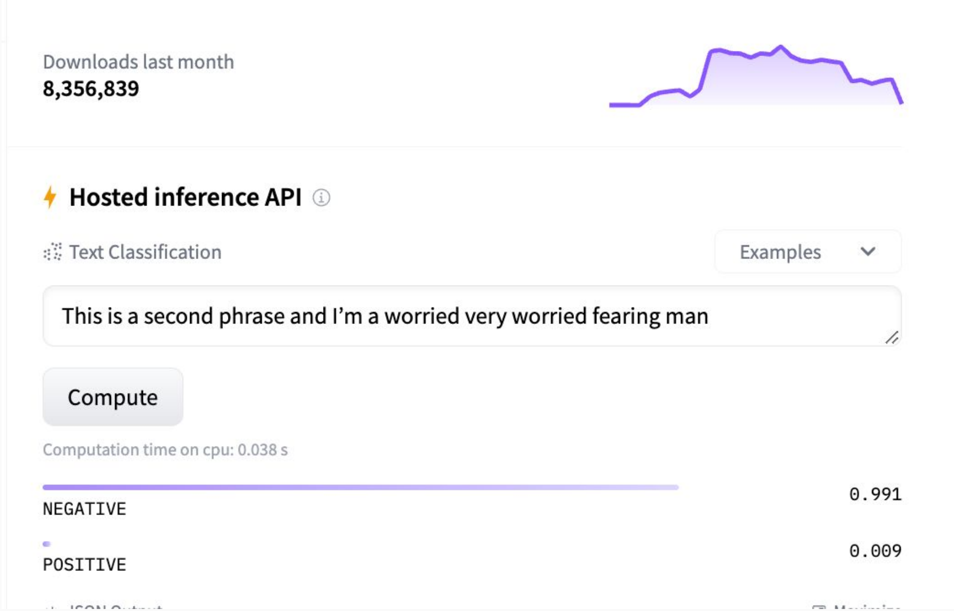
SOTA/LL Models - BERT

Has 100s of [Pretrained Models]

SOTA/LL Models - BERT

Text Classification Model (Sentiment):

https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english?text=This+is+a+second+phrase+and+I%E2%80%99m+a+worried+very+worried+fearing+man



SOTA (State of the Art)/LL Models - GPT3

OpenAI

Embeddings https://beta.openai.com/docs/guides/embeddings

To see embeddings in action, check out our code samples

- Classification
- Topic clustering
- Search
- Recommendations

Browse Samples

Types of embedding models

Currently we offer three families of embedding models for different functionalities: text search, text similarity and code search. Each family includes up to four models on a spectrum of capability:

- Ada (1024 dimensions),
- Babbage (2048 dimensions),
- Curie (4096 dimensions),
- Davinci (12288 dimensions).

SOTA (State of the Art) Models - GPT3

Dall-e https://labs.openai.com/s/R89aVa05GNpYFjaoW6CHWfBv

Try it: https://labs.openai.com/

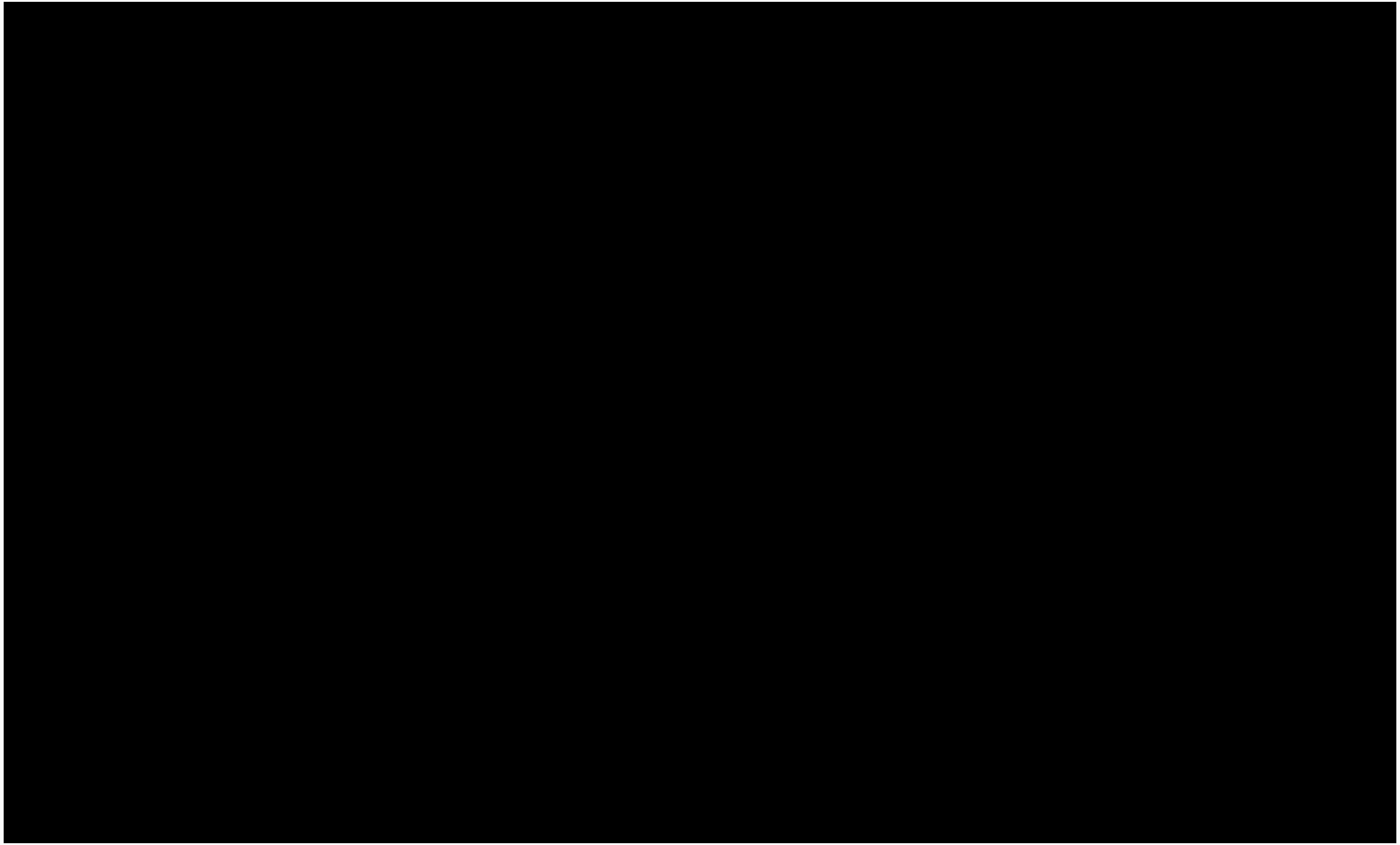
Outpainting (if time permits)
https://labs.openai.com/editor

Created with DALL·E, an AI system by OpenAI

“a painting of Goku working at KFC in the style of Salvador Dalí”

A Angel × DALL·E
Human & AI

ORB Amsterdam



SOTA Models - Training

When you all learned all Wikipedia you were trained

Training a BERT model costs about 7k dlls.. And days or months of GPU

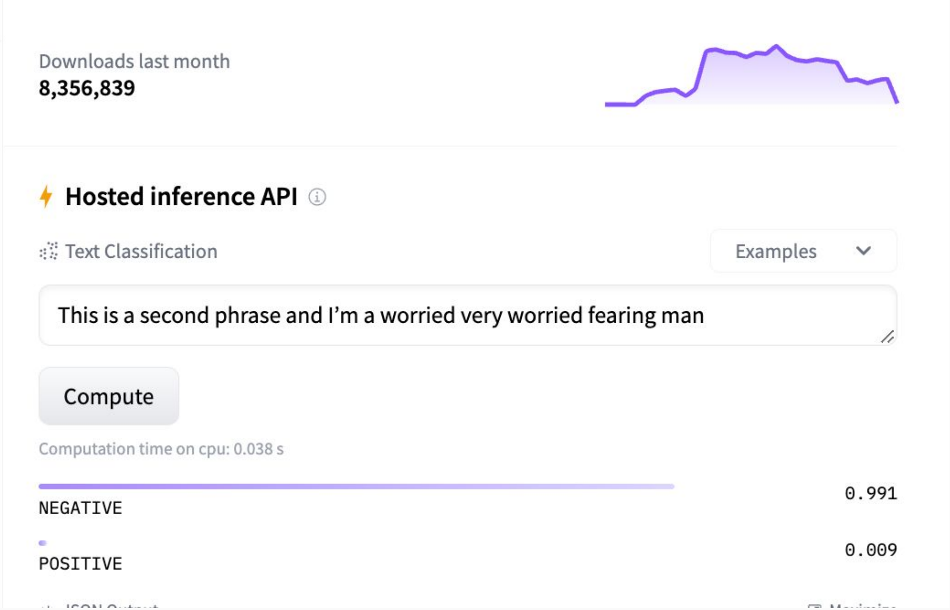
SOTA Models - Fine-tuning

Let's say go back to our Wiki example… and you are ready to learn more…

Confusion Matrix (For Classification Models)

Let's say you have to ensure a model's output is valid for Text Classification..

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

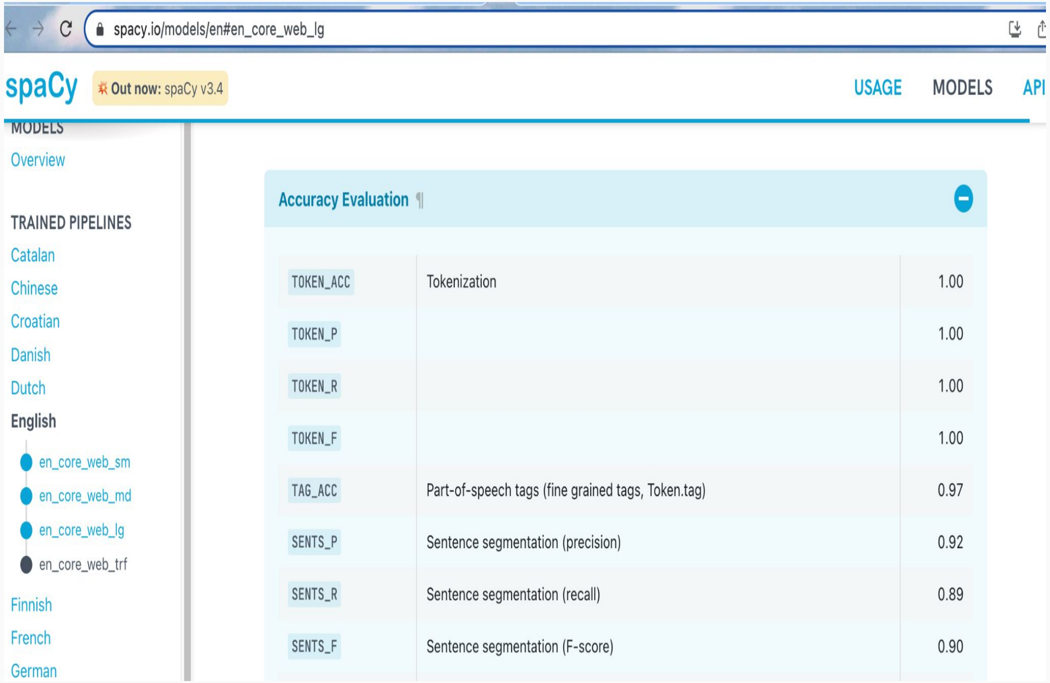


- Error rate
- Accuracy
- Precision
- Recall

Confusion Matrix (For Classification Models)

Let’s say you have to ensure a model’s output is valid for Text Classification..

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	**True positive** (TP)	**False negative** (FN)
	Negative (N)	**False positive** (FP)	**True negative** (TN)



- Error rate
- Accuracy
- Precision
- Recall

You know the basics now. Where to go from here?

huggingface.co/scjnugacj/jurisbert

Hugging Face   Search models, datasets, users...

scjnugacj/**jurisbert**   like 5

Fill-Mask | PyTorch | Transformers | Spanish | roberta | AutoTrain Compatible | License: other

Model card | Files and versions | Community

JurisBert

JurisBert, es una iniciativa de la **Suprema Corte de Justicia de la Nación (SCJN) de México**, nace en agosto del 2020, a propuesta de la **Unidad General de Administración del Conocimiento Jurídico (UGACJ)**, para entrenar un Modelo del Lenguaje contextualizado al ámbito jurídico. Su principal objetivo es generar aplicaciones de **Procesamiento del Lenguaje Natural (PLN)** que coadyuven a la labor jurisdiccional del Alto Tribunal mediante el aprovechamiento del conocimiento de la SCJN plasmado en documentos no estructurados que generan las áreas jurisdiccionales.

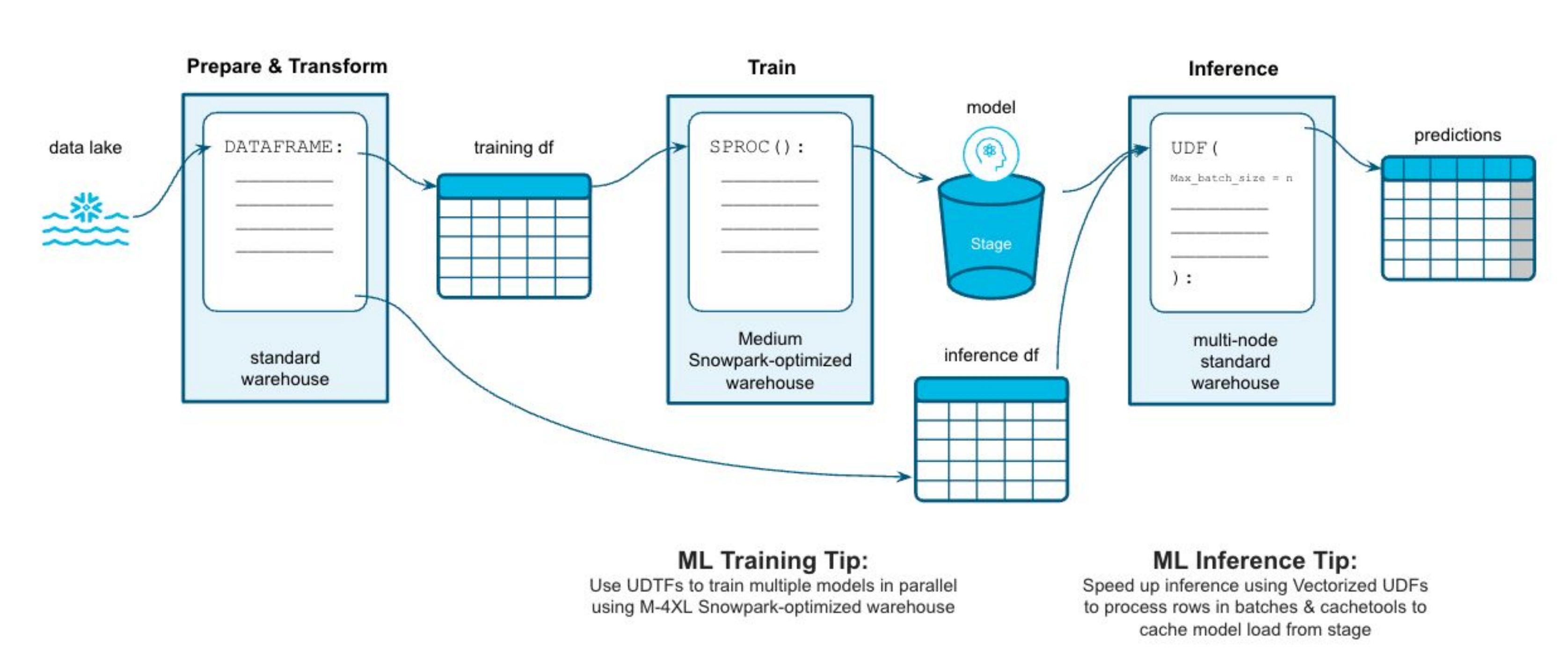
En 2021, esta iniciativa tomó mayor relevancia con la llegada de la Reforma Judicial y el inicio de la undécima época del SJF, puesto que la creación de JurisBert tiene como objetivos principales la ayuda a la identificación del precedente y la creación de Plataformas de Recuperación de Información.

Como parte de la Transformación Digital impulsada por la SCJN, en razón de generar un esquema de “Gobierno Abierto” mediante la Colaboración e Innovación y en el contexto de la operación remota

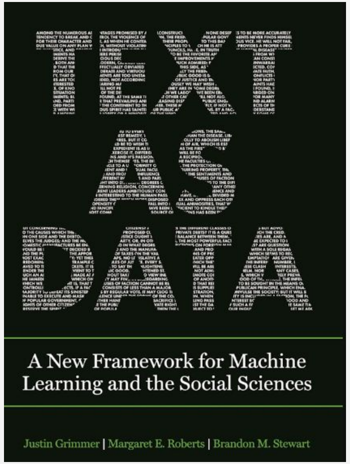
Deploying models

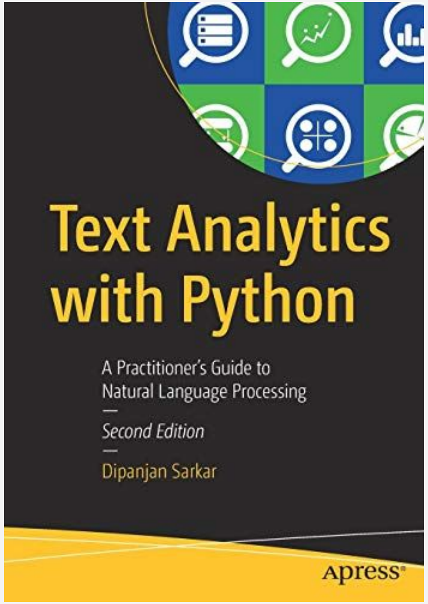
- Kubernetes, MLFlow, Airflow, Sagemaker
- Snowflake/Snowpark

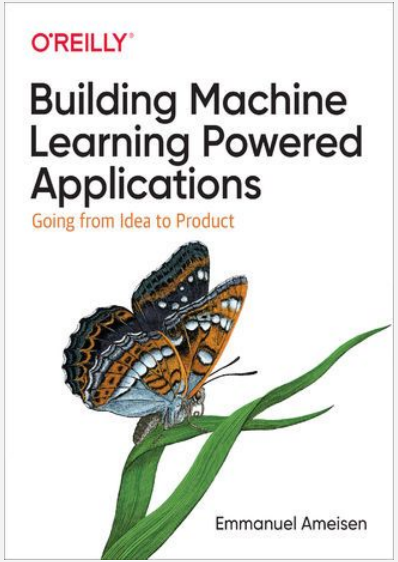
Snowflake - Snowpark

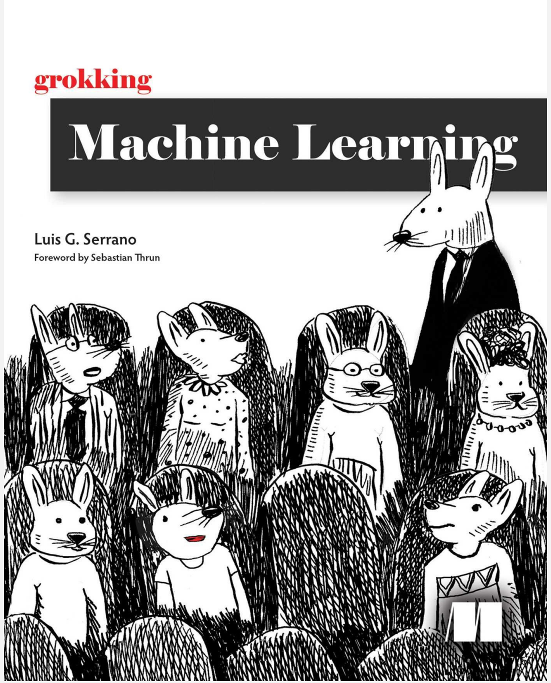


Recommended readings









angel@molanco.com
angel@talkingpts.org
linkedin.com/in/angel-alvarado-robledo/

Slides

https://bit.ly/DataDaysNLP

Qs?