# The Economics Crime over Time and Space
# Theory, Practice and Applications
## *(A primer in) Text as Data with Python*

Arpita Ghosh

*22nd September 2022*

# Why do we care?

- We can obtain a lot of data from internet
- Most of which will be in texts, which are unstructured
- Examples: digitised text data from old records etc.
- Systematically extracting quantitative data from texts
- Redefining the field: machine learning - artificial intelligence

# Aim for this session

1. Take the data we scraped and make sense of it.
    - Some basic operations - standard unstructured data
2. First doing some run-of-the-mill text operations. All the steps not always relevant but good starting point.
    - Write a script to get a sense of the data, some bag of words, topic modelling etc.
3. Write a script to actually obtain the dates (regex) and locations (training to recognise the locations).

# Assumption/Goal/Declaration

## Goal

We want to look at bringing out only date and location of unsolved murders from unstructured text.

*Two ways to attempt: supervised or unsupervised learning. We explore mainly supervised learning.*

## Assumption

We are not looking to perfect this data.

*There will be cases where the training/regex do not exhaust the data and we will let that go.*

## Declaration

There are many approaches - this is one way of looking at it.

*Not building ML models: classic split, train, test, validation etc.*

# A few basic text preprocessing commands

- Lower cases; Remove white spaces; Punctuations; Count common words
- Tokenize words: split text in tokens - words, numbers, etc
- Stopwords: most common words like ‘the’, ‘a’, ‘on’ etc - do not carry important meaning and can be removed straight away. NLTK built-in.
- Frequent words: in cases counting/removing frequent words without contextual meaning might be important, e.g., target is to get cricket scores from newspaper archives, but having famous players’ names repeatedly might not be important.
- Lemmatization: process of grouping together the different inflected forms of a word so they can be analyzed as a single item, e.g., “been had done languages cities mice” to “be have do language city mouse”
- Stemming: reduce words to the core terms; like runner, ran, running to run. Generally either stemming or lemmatization.

# Bag of Words

- Text can be collection or bag of words
- Count of different words in a document after basic cleaning
- Doesn't account for order and structure of the words
- Total sum of each occurrence can be useful as well
- (not very useful for us, but can be especially in parliament speech analysis etc)

```
Name: clean_text, dtype: object
   abandoned  abbey  abdel  abdhou  abdi  ...  zhang  ziggy  zip  zoe  total_tokens
0          0      0      0       0     0  ...      0      0    0    0            22
1          0      0      0       0     0  ...      0      0    0    0            21
2          0      0      0       0     0  ...      0      0    0    0            22
3          0      0      0       0     0  ...      0      0    0    0            45
4          0      0      0       0     0  ...      0      0    0    0            29
```

# Topic Modelling

- Not just counting words but understand relevant key words in a document
- Term-Frequency Inverse Document-Frequency (TF-IDF) - identify common words in a document; taking in account how many documents contain that word
- Generate a TF-IDF score and then check keywords that occur together in documents to identify topics (with keywords weights) - assign topic scores to a document.
- Latent Dirichlet Allocation (LDA): unsupervised learning algorithm which finds blend of themes or topics in a set of documents. More details here: Link

# Named Entity Recognition

## Named Entity (NE)

A real-life object with proper identification and name. Named Entities can be person, date, place, organization, time, geographic entity etc.

- NER is the process of NLP that identifies and classifies NEs.
- Named entities are identified/segmented in predetermined classes.
- Common types - ORG (NASA, Google etc), DATE (23 January 2005), PERSON (Bill Gates, Elon Musk etc)
- Spacy: great open-source NLP library with built-in NER methods

```
>>> doc = NER(
...     u'Dean Tully, 37, was shot dead in Fraser House on the Haverfield Estate off Green Dragon Lane, Brentford, at around 9.15pm on 25 January 2007.
Police said two men burst into a two-bedroom flat and sprayed bullets from a Czech-made CZ25 sub-machine gun.')
... show_ents(doc)
...
Dean Tully-0-10-People, including fictional
37-12-14-Absolute or relative dates or periods
Fraser House-33-45-Companies, agencies, institutions, etc.
Green Dragon Lane-75-92-People, including fictional
Brentford-94-103-People, including fictional
9.15pm-115-121-Buildings, airports, highways, bridges, etc.
25 January 2007-125-140-Absolute or relative dates or periods
two-154-157-Numerals that do not fall under another type
two-175-178-Numerals that do not fall under another type
Czech-219-224-Nationalities or religious or political groups
```

# Extract Dates

- Can't use built-in NER as sometimes identifies age or time etc.
- But dates are stored only in some formats
- So we explore regular expressions instead of training
- I like https://regex101.com to build/test my regex
- Attempt in three stages - reasoning in code
- Five cases (out of 194) don't get captured from regex (like Boxing Day 2007, or New Years Eve 2009 etc.)
- One can write custom NER to capture those as well

# Extracting Locations

- Locations more difficult to extract
- One approach: train a custom NER model in Spacy to recognise locations
- Generate a training data from the raw data we have
- Essentially showing where the locations are by giving start and end points of the 'new' entity

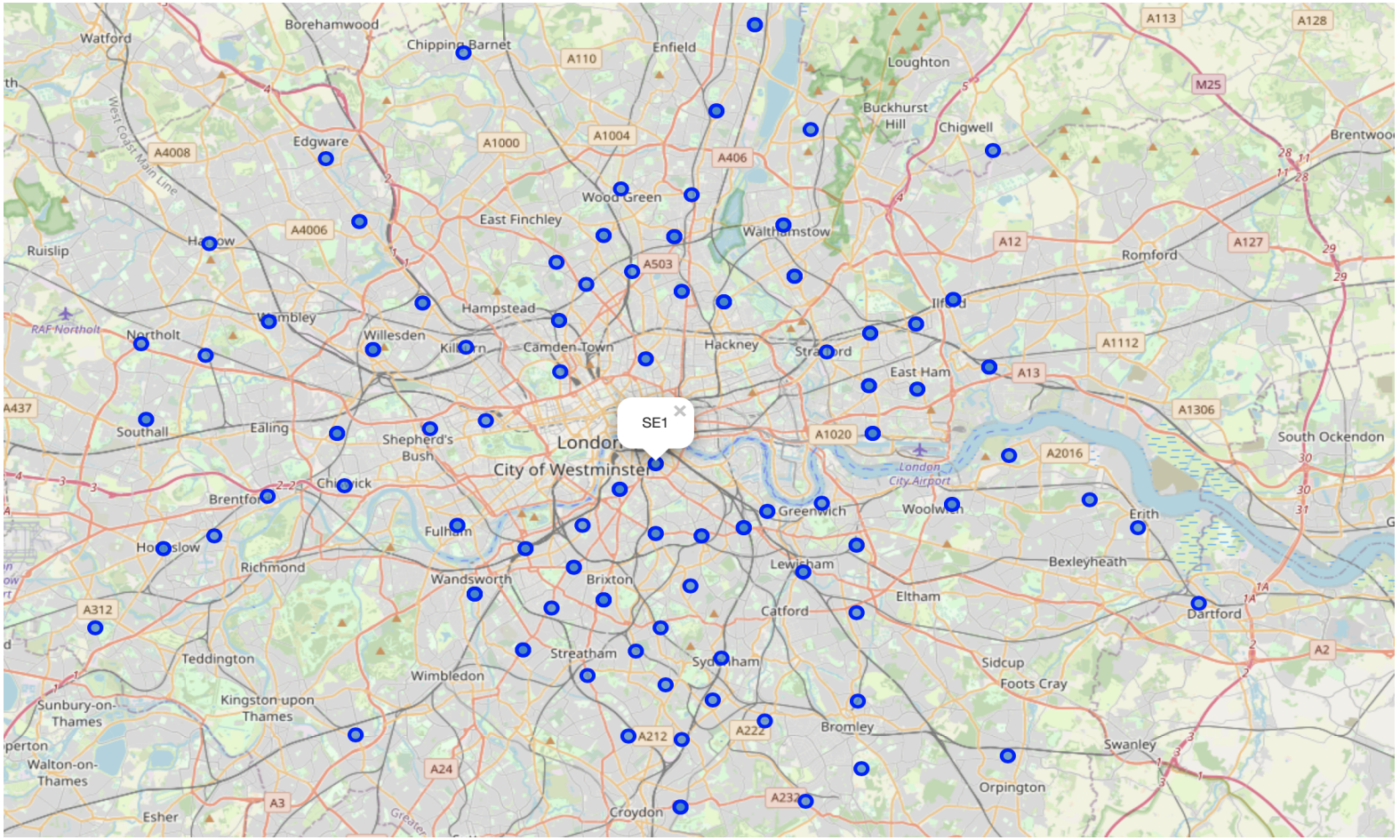
# Extracting Locations - the code

- Loading a pre-existing model and adding the new label of the entity we want
- Disable some of the pipes in the model that we don't want to change
- Randomly shuffle the order of training data for a better training
- Test in a sample text and apply in our data
- One can create a training model and save etc.

# Plot Locations of unsolved murders - the code

- Greedy in Location extractions - for missing use built-in NERs to get locations
- Get the postcode districts mapped based on locations in London and from that latitudes and longitudes
- Using Folium get a basic interactive map of unsolved murders (not accounting for years etc.)
- Will generate as an HTML map in the project folder (does not show interactively)

# Finally!




*Note: Can be modified/extended in many many ways. . . !*

# Wrapping up

- Mapped locations of unsolved murders in London from text data
- One can do clusters - geographic concentrations - interactive maps (progression by year) etc.
- To note:
  - training data in more effective ways with validations etc.
  - the location matching can be with ngram, Vectorization, TF-IDF, cosine similarity etc.
- Other common things: Parts of Speech tagging, Sentiment Analysis, Word embedding models, Semantic similarity & Arithmetic etc. (perhaps more relevant in other contexts)
- There are many complex things than can be done in NLP - like BERT built on recurrent neural networks (RNN) that can do contextual understanding of words

Thank you very much for your attention and if you have any comments/questions please do reach out to me.

arpitaghoshecon@gmail.com

# Sources for both talks


- Parashar, A, 2022, "Web Scraping Using Selenium Python" (https://medium.com/pythoneers/web-scraping-using-selenium-python-6c511258ab50)
- Nathan, P, 2019, "Natural Language Processing in Python using spaCy: An Introduction" (https://www.dominodatalab.com/blog/natural-language-in-python-using-spacy)
- Nishanth, N, 2020, "Training Custom NER" (https://towardsdatascience.com/train-ner-with-custom-training-data-using-spacy-525ce748fab7)
- Dyer, J, 2022, Text as Data Workshop. University of Exeter. (https://github.com/juliandyer/TextAsDataWorkshop)
- Davydova, O, 2018, Text Preprocessing in Python: Steps, Tools, and Examples. (https://medium.com/product-ai/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908)
- Gentzkow, M., Kelly, B. T., and Taddy, M., 2017, Text as Data. NBER Working Paper # 23276