ISE 540: Text Analytics

Mayank Kejriwal

Research Assistant Professor/Research Lead

Department of Industrial and Systems Engineering

Information Sciences Institute

USC Viterbi School of Engineering

kejriwal@isi.edu

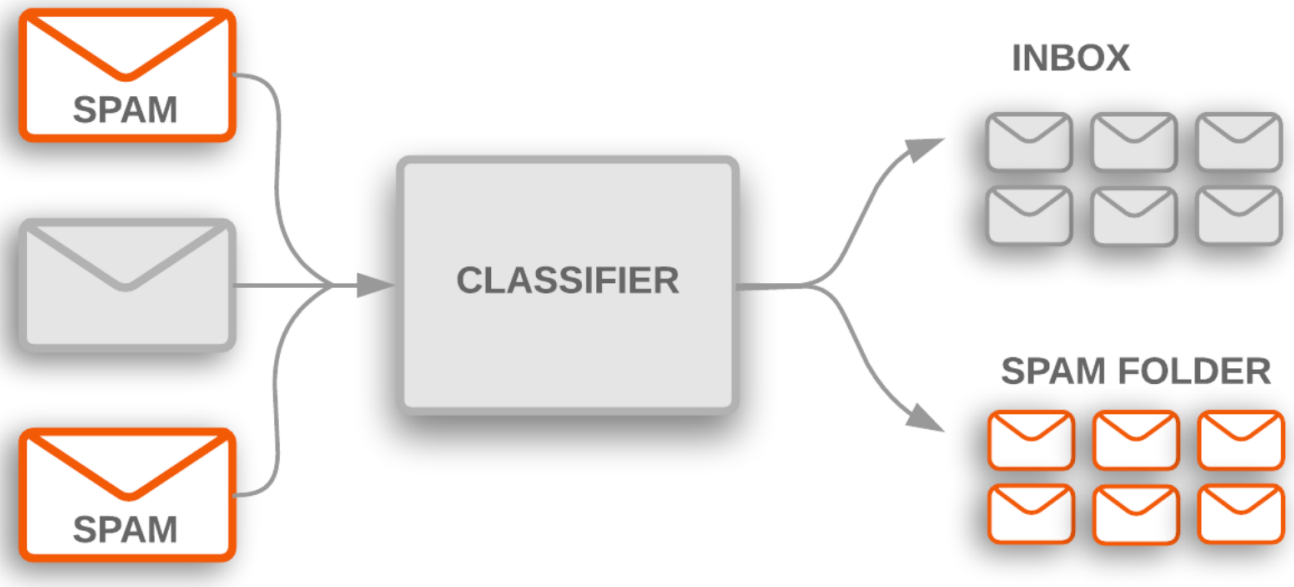
Text classification

Problem definition

- Simple: Given a piece of text, categorize (’classify’) it into one of several pre-defined categories (‘labels’)
- Many versions:
  - Change ‘one of…’ to ‘one or more…’ (also, have the ability to pick NA or NULL as a ‘label’)
  - What if the labels are not pre-defined? (zero-shot learning)
  - Online vs. offline: are the texts coming in one at a time, or is the full corpus available and waiting to be labeled in batch mode?
  - What kinds of ‘background’ resources can we use?

Applications

We've already seen one!

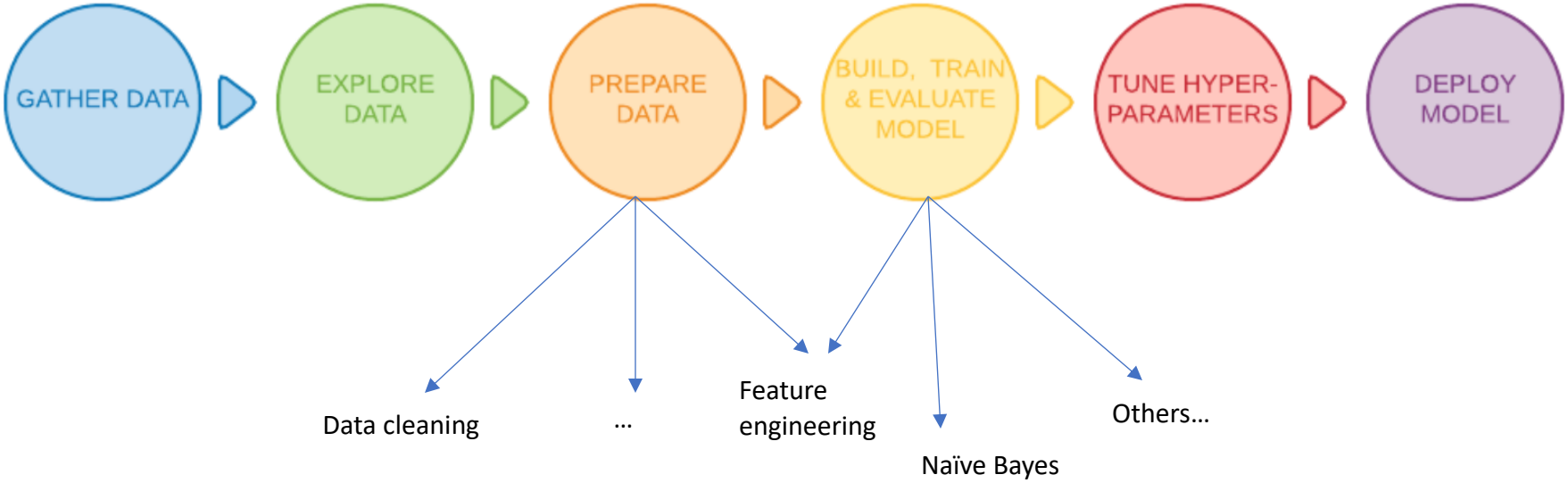


Source: https://developers.google.com/machine-learning/guides/text-classification

Applications: Sentiment Analysis

Loves the German bakeries in Sydney. Together with my imported honey it feels like home	Positive
@VivaLaLauren Mine is broken too! I miss my sidekick	Negative
Finished fixing my twitter...I had to unfollow and follow everyone again	Negative
@DinahLady I too, liked the movie! I want to buy the DVD when it comes out	Positive
@frugaldougal So sad to hear about @OscarTheCat	Negative
@Mofette briliant! May the fourth be with you #starwarsday #starwars	Positive
Good morning thespians a bright and sunny day in UK, Spring at last	Positive
@DowneyisDOWNEY Me neither! My laptop's new, has dvd burning/ripping software but I just can't copy the files somehow!	Negative

Workflow


GATHER DATA
EXPLORE DATA
PREPARE DATA
BUILD, TRAIN & EVALUATE MODEL
TUNE HYPER-PARAMETERS
DEPLOY MODEL
Data cleaning
…
Feature engineering
Naïve Bayes
Others…

Vector space models

Basic intuition

- Encode each ‘document’ in document-set D (‘corpus) as a vector
  - Many questions:
  - How to encode documents into vectors?
  - How to measure quality of vectors?
  - How to use the vectors? (this should be easy)

Incidence Matrix

Each column is a 'document' (in this case, a play)

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	…
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
…							

Each row is a word

Question: given a vocabulary (number of unique words in corpus) of size V and D documents, what is the number of elements in the matrix?

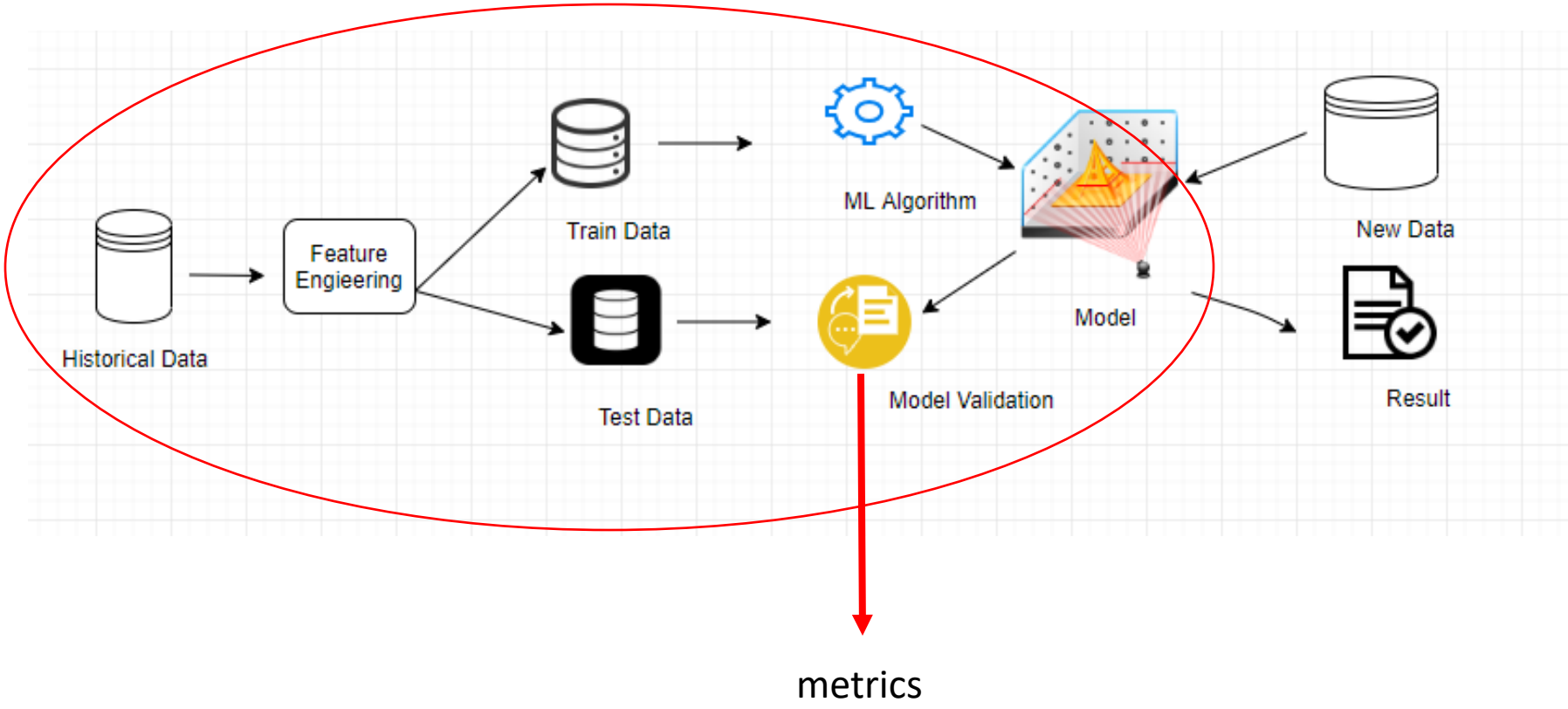
Simplest way to construct vectors…

- Incidence matrix
  - Each column of the incidence matrix is a vector of 1's and 0's encoding the document in terms of words that occur in it
- Why wouldn't this work well?
  - Hint: is every word equally important?

How do we use these vectors?

- Suppose we want to classify each Shakespearan play as a ‘tragedy’, ‘comedy’ or ‘neither’ (3-label problem)
- First step might be to preprocess/clean the data
- Second step is to ‘vectorize’ the data e.g., by using incidence matrix
- Third step is to train a model (and possibly tune it)
  - How? You’ll get a lot of practice through HW2 in deriving features and training models

Recall from last time…



Evaluating performance

- Suppose you've trained a model and are using it to for the Shakespearan 3-label problem
- How do you evaluate it?
- One option: accuracy on 'withheld' or *test* data
  - What fraction of the input did you label correctly?
- Other options: precision, recall…(to be covered later)

Input	Actual label	Model prediction
Macbeth	Tragedy	Tragedy
Julius Caesar	Tragedy	Comedy
Much ado about nothing	Comedy	Tragedy

Beyond ordinary classification

- Vectors are also useful in clustering (unsupervised learning) and other tasks like information retrieval (search engines and recommendation)
- Essential to understand and become comfortable with them to do anything useful with text
- If there is one thing you should remember after this class, it is the set of vector space models we will covering this week (and referring back to, over and over again)

Let’s return to getting the vectors in the first place

- One popular model is ‘bag of words’, commonly held to be the ‘tf-idf’ model
  - Tf-idf stands for term frequency-inverse document frequency
  - The tf term encodes the intuition that the more times a word occurs in a document, the more important it is to the document
  - The idf term encodes the intuition that, all else equal, a word that occurs more often throughout the corpus is less important to each document that it occurs in, than another word that is more rare (‘the’ vs. ‘apple’)
  - Tf is ‘local’ (need only a document), idf is ‘global’ (we need the full corpus to compute idf)…where can this go wrong?
- A bag is a ‘multi-set’; to compute ‘tf’ we need to know how many times the word occurs in the document!