Wikipedia,
Dead Authors,
Naive Bayes
& Python

Outline

- Dead Authors : The Problem
- Wikipedia : The Resource
- Naive Bayes : The Solution
- Python : The Medium
    - NLTK
    - Scikits.learn

Authors, Books & Copyrights

Authors write books.
Books are published.
Authors earn royalty.

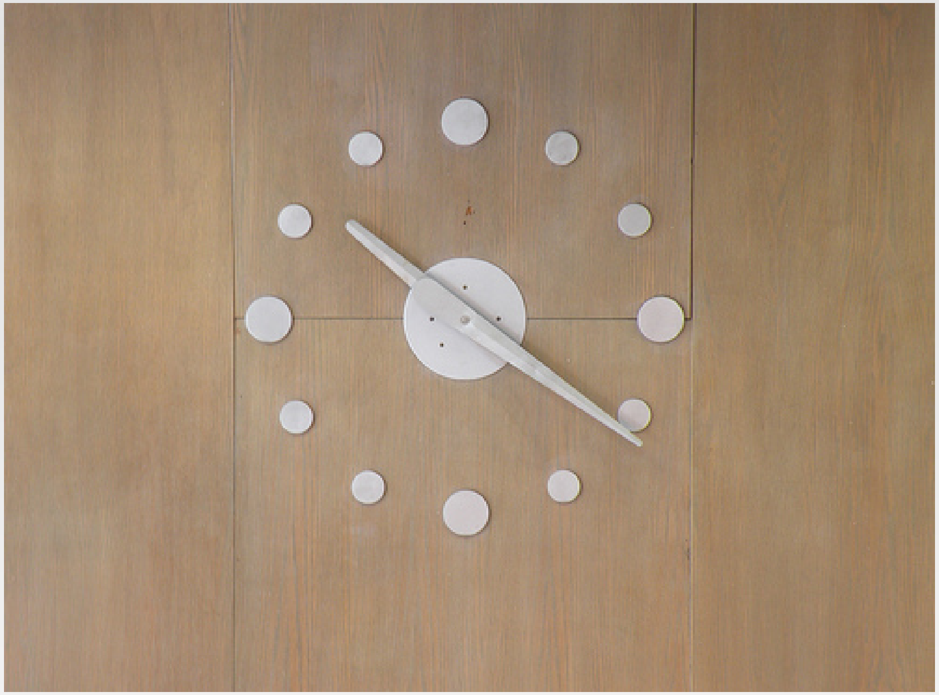
Authors hold copyright of their works which prevents unauthorized use.

As time goes by..

Then one day authors die...

Image from: http://www.flickr.com/photos/adamcrowe/4071096483/

The copyright clock



Once they die, a clock starts ticking!

Image from: http://www.flickr.com/photos/31332713@N04/3086719615/

After 60 years..

60 years after the death of an author, his works enter the public domain.

Image from: http://www.flickr.com/photos/magdav/5399905776/

What does that mean?

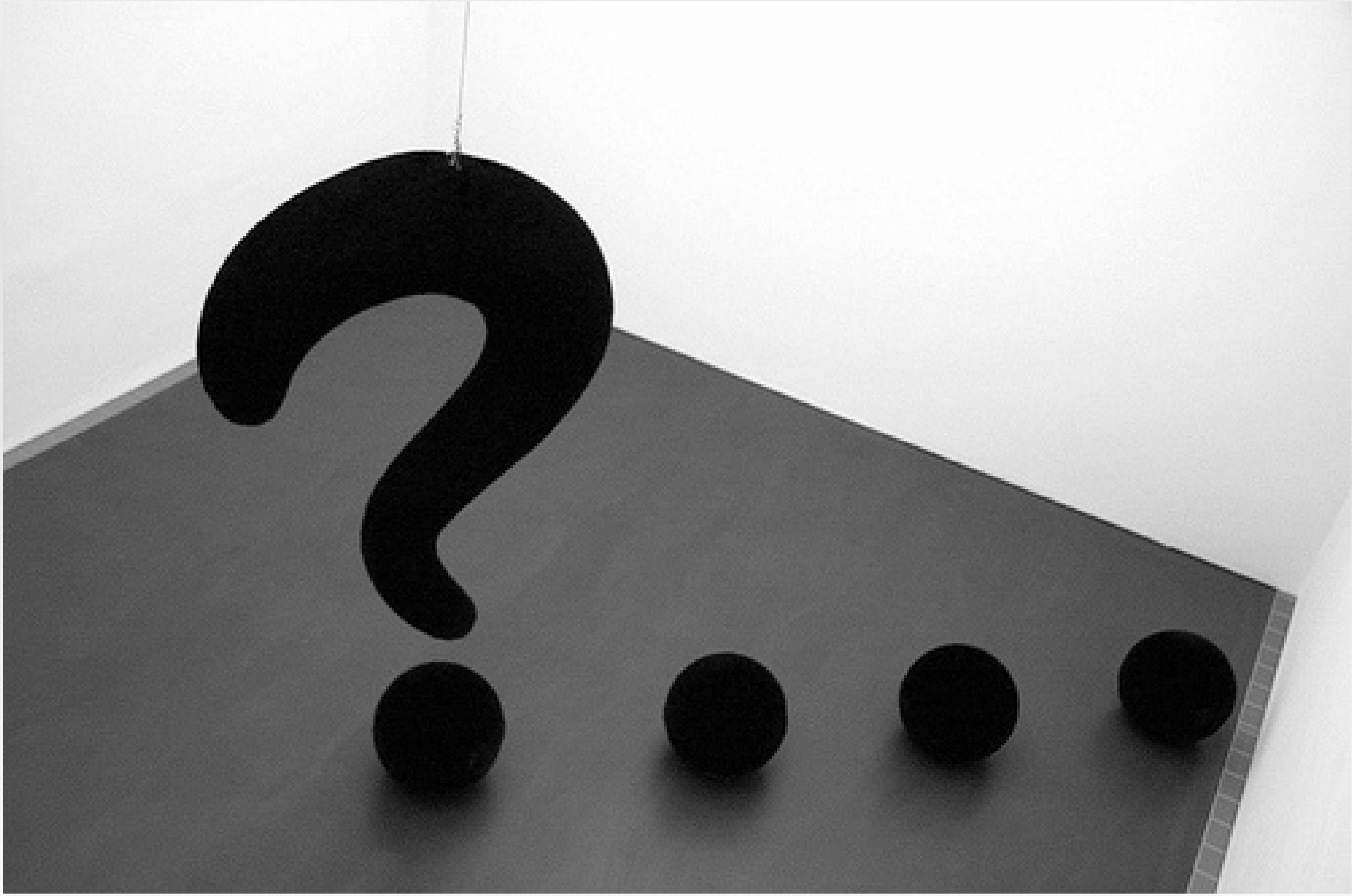


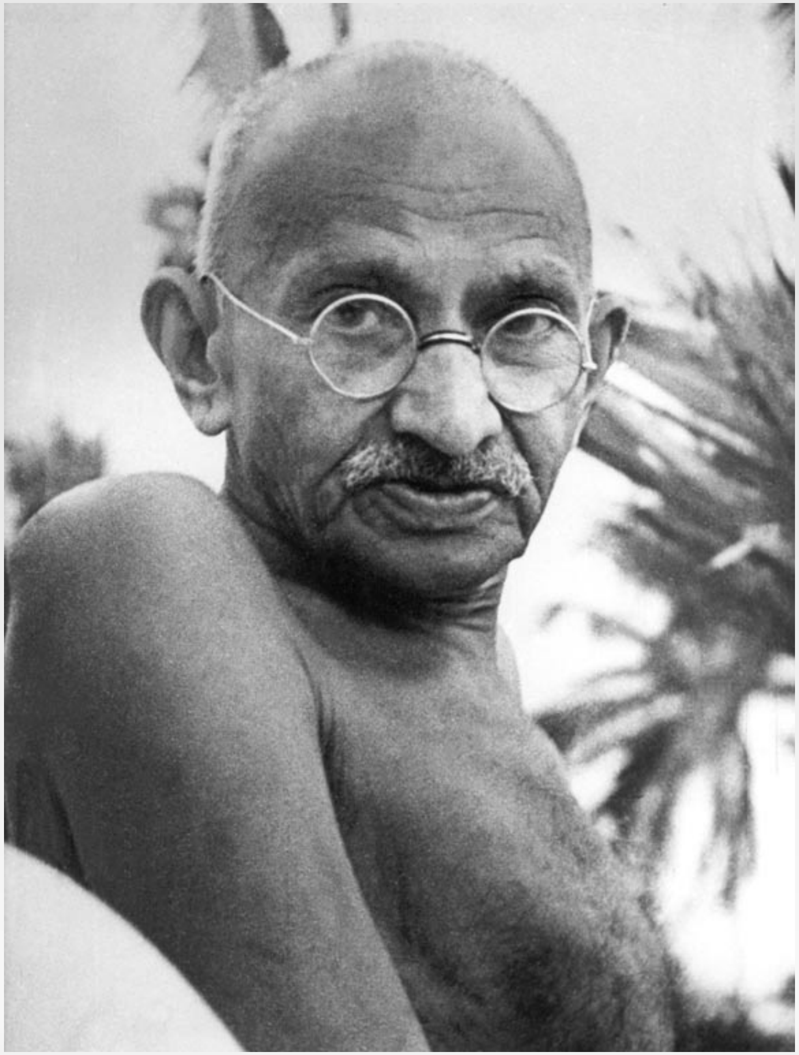
Image from: http://www.flickr.com/photos/-bast-/349497988/

Public Domain works

This means that anyone is free to

- Translate them
- Digitize them
- Record Audio books
- Create derivative works
- Publish cheaper editions
- Anything else you can think of!

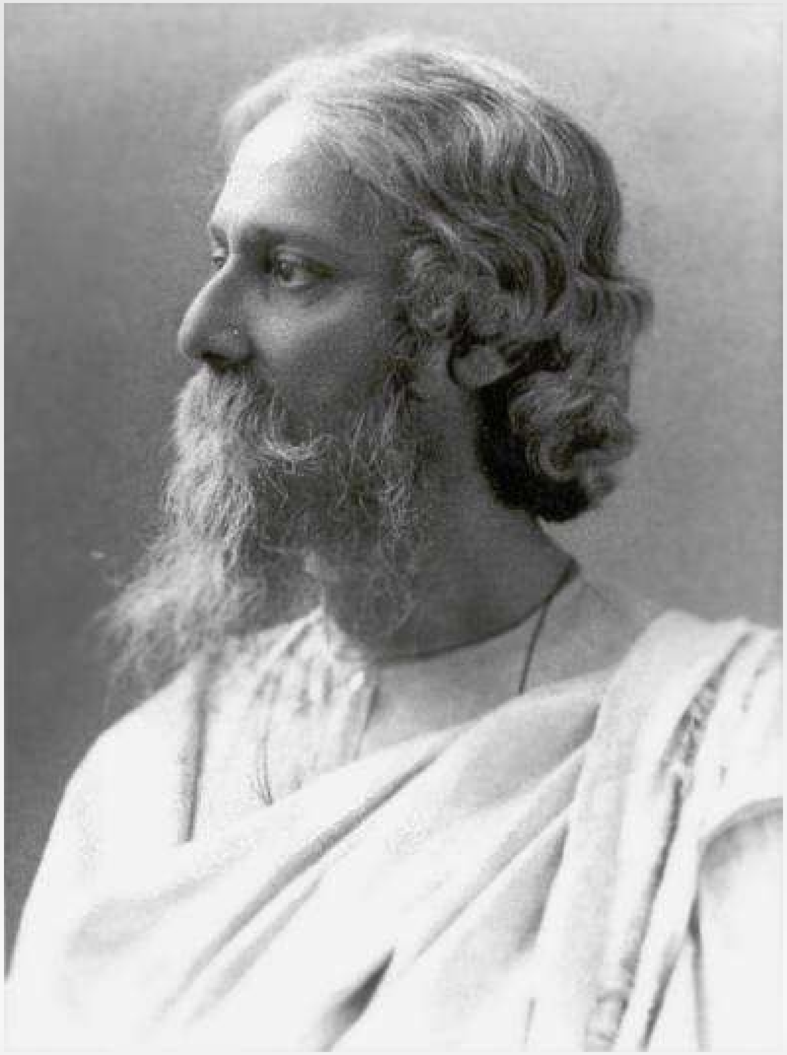
Mahatma Gandhi



- My Experiments with Truths (Gujarati version)
- Hind Swaraj

Rabindranath Tagore

- Geetanjali
- Autobiography (My Reminiscences)
- Gora



Munshi Premchand

- Godan
- Nirmala
- Other novels and stories

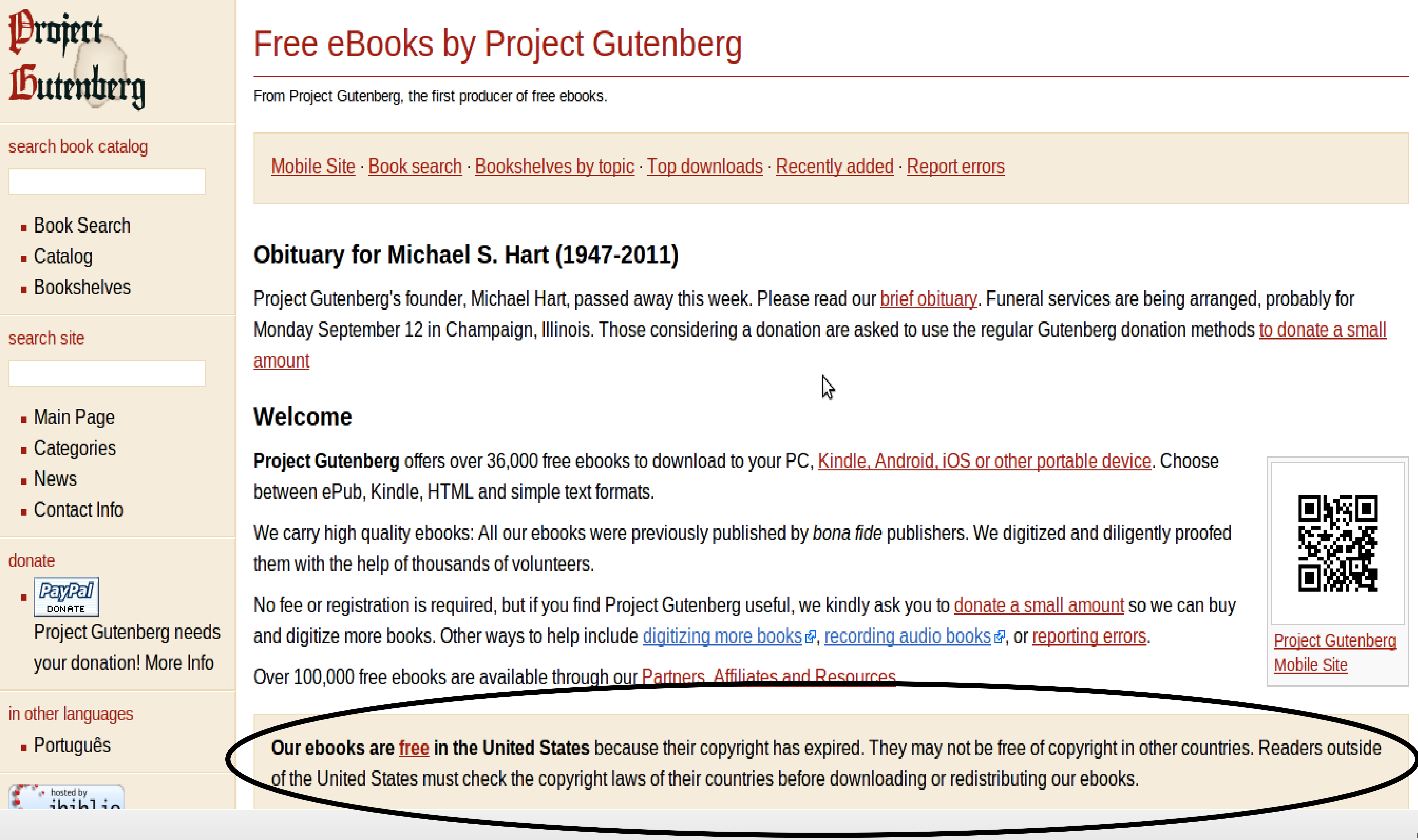
Sarojini Naidu

- Poetess
- The Golden Threshold

Jai Shankar Prasad

- Kamayani
- Wrote many historical plays

Project Gutenberg



Project Gutenberg needs your donation! More Info

Free eBooks by Project Gutenberg

From Project Gutenberg, the first producer of free ebooks.

Obituary for Michael S. Hart (1947-2011)

Project Gutenberg's founder, Michael Hart, passed away this week. Please read our brief obituary. Funeral services are being arranged, probably for Monday September 12 in Champaign, Illinois. Those considering a donation are asked to use the regular Gutenberg donation methods to donate a small amount

Welcome

Project Gutenberg offers over 36,000 free ebooks to download to your PC, Kindle, Android, iOS or other portable device. Choose between ePub, Kindle, HTML and simple text formats.

We carry high quality ebooks: All our ebooks were previously published by *bona fide* publishers. We digitized and diligently proofed them with the help of thousands of volunteers.

No fee or registration is required, but if you find Project Gutenberg useful, we kindly ask you to donate a small amount so we can buy and digitize more books. Other ways to help include digitizing more books, recording audio books, or reporting errors.

Over 100,000 free ebooks are available through our Partners, Affiliates and Resources.

Our ebooks are free in the United States because their copyright has expired. They may not be free of copyright in other countries. Readers outside of the United States must check the copyright laws of their countries before downloading or redistributing our ebooks.

Project Gutenberg

- Project Gutenberg (PG) is a big repository of digitized out of copyright books.

  (http://www.gutenberg.org/)
- No such resource exists for India.
- PG follows US copyright policy => only books published before 1923 can be added.
- In India, if the author died before 1950, his works are in public domain.

Why?

But why should we,
as hackers and geeks,
care?

Reasons

Hackers like to fix things that are broken!

Reasons

OCR

Transliteration

Machine Translation

Spell Checkrs

Information Retrieval

Large datasets are the necessary ingredients for building Machine Learning based NLP tools

Reasons

Hackers love free things!

Building an Indian PG

- First step is to identify the Indian authors who are out of copyright.
- But how do we find out when did an author die?
- There are websites that maintain author lists by year of death but coverage of Indian authors is low

  (http://www.authorandbookinfo.com/)
- So what do we do?

Wikipedia!

Wikipedia Solution

WP has a category for Indian Writers!

+

WP has categories by the death years!

Voila! we can just look at pages belonging to both the categories!

But..

- WP categories are not comprehensive. Many author pages are not tagged.
- Also, we are looking for everyone who wrote a book. Even if he may not be a full time writer.
- So gleaning all the yearwise death categories is required.

Wikipedia Solution cont

- Some stats
  - Typically 1800-2000 entries for each year
  - Around 25-30 Indians
  - Around 10-12 Indian authors
- Also WP is a work in progress. Information is continually updated.
- So we may want to look again every few months

In search of better solution

- This is a time consuming, tedious and hence an error prone task for humans.
- Can we do something better?

Being Naive!

Aren't we Naive?

- This is a classic document classification problem.
- Given all the pages listed in <year>_deaths category, classify them as Indian authors or not.

Document Classification

- Document classification and text classification are well studied problems.
- **Naive Bayes (NB)** is a simple Machine Learning Model that is known to perform nicely on this problem.

Naive Bayes : An Intro

<< Interactive >>

NB: Continued

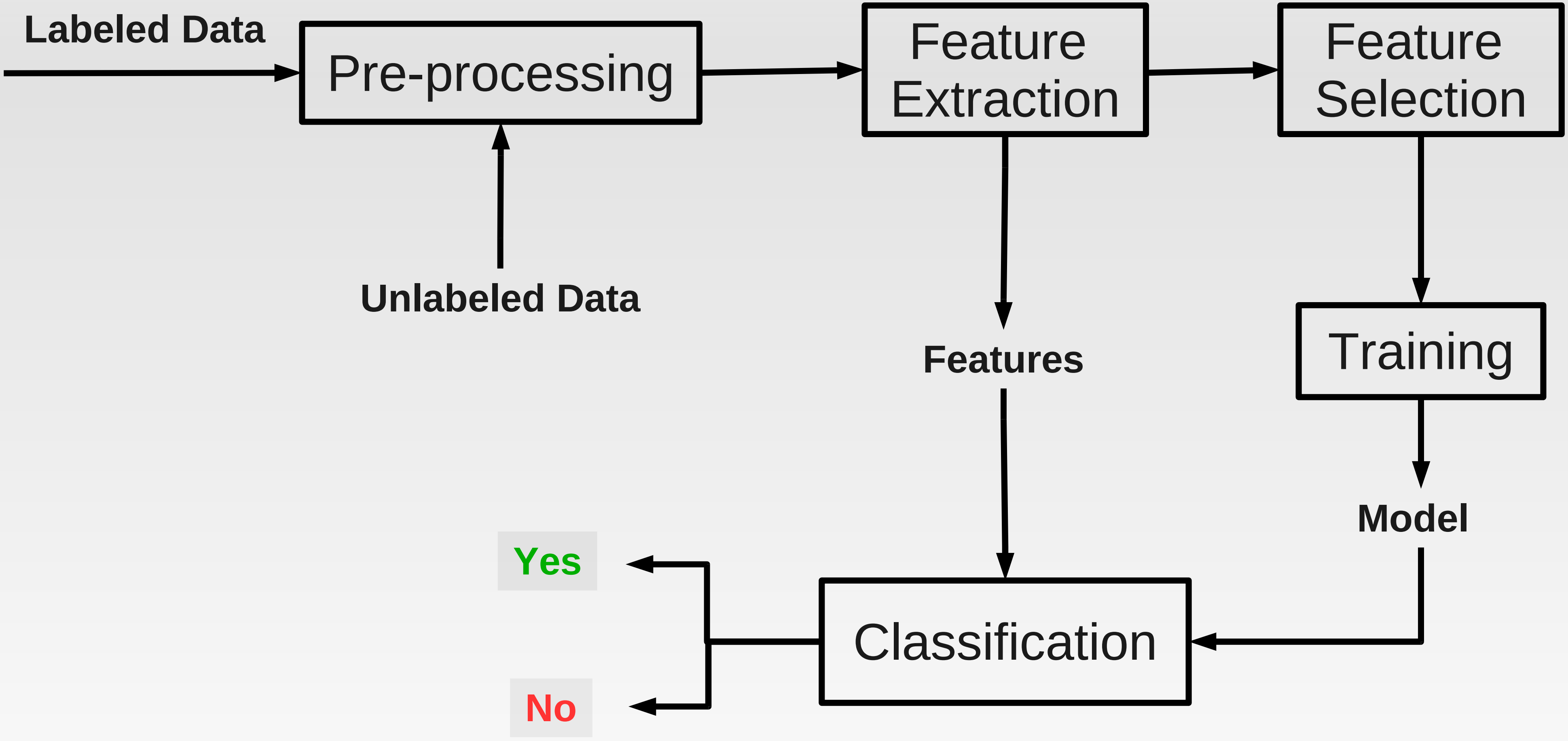
- The ”Naive” in NB refers to the assumption that all the features being used are independent
- In real life datasets, not easy to find completely independent features
  - Words in a document are not independent of each other!
- NB works well even when the features are not independent.

Python

One ring to bind them all

Image from: http://www.flickr.com/photos/thecaucas/2232897539/

Overview of supervised learning



Preprocessing

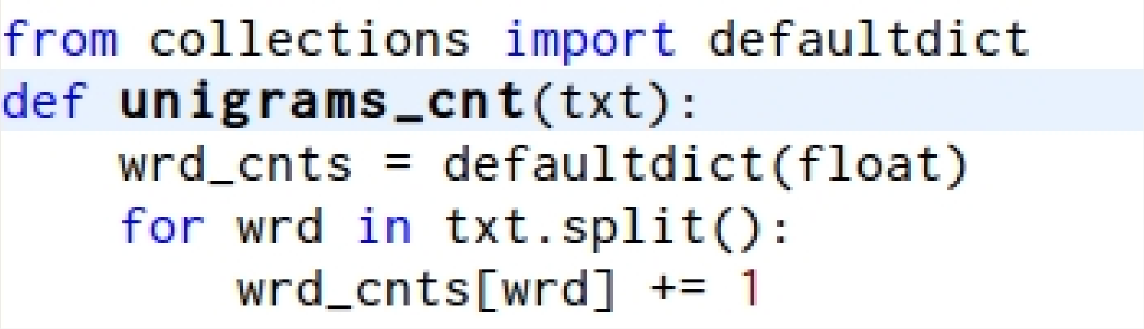
- Input text often needs cleaning before feature extraction
  - Stripping out markup, Tokenization, Decoding entities
- Regular expressions are your friends.
- Build a library of functions each doing exactly one transformation.
  - Allows for quickly putting together different preprocessing schemes and evaluating them.

Feature Extraction

- Typical features employed in NLP
  - Words and phrases (Unigram, bi-gram)
  - Part of speech tags
  - Dictionary Features. Ex: if a word is present in a list of place names
- Features need to be numerical.
  - You can either collect counts or have a boolean feature indicating presence or absence.

Feature Extraction Cont

- Iterators make is super easy to extract features from text.
- Combine them with defaultdict and itertools to make life even simpler.
- Unigram Counts



```
from collections import defaultdict
def unigrams_cnt(txt):
    wrd_cnts = defaultdict(float)
    for wrd in txt.split():
        wrd_cnts[wrd] += 1
```

NLTK

- NLTK is Natural Language Toolkit written in Python. (http://www.nltk.org)
- An excellent library with
  - Implementations of wide variety of NLP algorithms for tagging, parsing, stemming etc
  - Various trained models for Part of Speech tagger, sentence splitter etc
  - Wrappers for various ML libraries. Ex: Weka
- NLTK Book (http://www.nltk.org/book) is a good place to start

Naive Bayes in NLTK

- NLTK has an implementation of NB classifier.
- Very easy to use

```
fsets = [(unigrams(txt),lbl) for (txt, lbl) in
trdata]

clsfr = nltk.NaiveBayesClassifier.train(fsets)

print nltk.classify.accuracy(clsfr, fsets)
```

- Although the implementation doesn't look correct. :-(

Scikits.learn

Python module integrating classic machine learning algorithms in the tightly-knit world of scientific Python packages (numpy, scipy, matplotlib)

- Actively developed and has good documentation. (http://scikit-learn.sourceforge.net/stable/)
- If I had discovered it earlier, would have implemented in this framework

Scikits.learn

- Easy to use though slightly different interface as compared to NLTK
- Assuming X contains the feature sets and Y, the corresponding labels

```
from scikits.learn.naive_bayes import BernoulliNB

clsfr = BernoulliNB()

clsfr.fit(X,Y)

print clsfr.score(X,Y)
```

Scikits.learn

- X & Y need to be numpy arrays. Assuming we are using 5000 features:

```
X = np.zeros((len(fsets), 5000), dtype =
'float64')
  for docidx, fset in enumerate(fsets):
    for fname, fval in fset.iteritems():
      if fname in featd:
        X[docidx][featd[fname]] = fval
```

- featd is a map of feature_name to feature_id

Scikits.learn

- Learning curve is steeper
  - Talks in terms of Estimators, likelihoods and other technical terms
  - Needs familiarity with basic numpy concepts. (Totally worth it if you are planning to do any serious numerical work in Python.)
  - You need to have some level of familiarity with Linear Algebra to peek inside and optimize or to implement your own classifiers.

Our Experiment

- Preprocessing
  - Using Wikipedia API through wikitools (http://code.google.com/p/python-wikitools/)
  - Convert link markups, strip out the reference markings, decode html entities
- Features
  - Binary unigram occurance features
  - Section headings

Demo

Demo