Big Data Analysis in Python

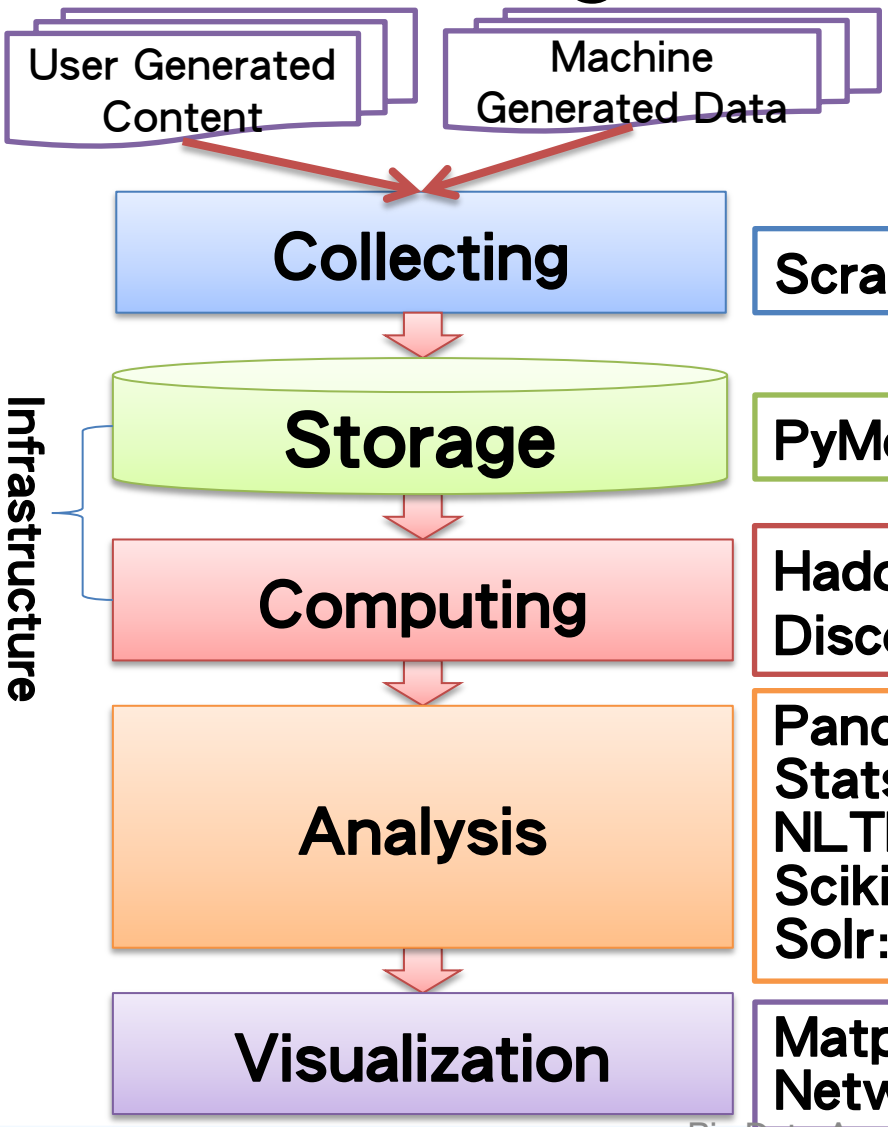
Jimmy Lai
r97922028 [at] ntu.edu.tw
http://tw.linkedin.com/pub/jimmy-lai/27/4a/536
2013/05/26

This slides: http://www.slideshare.net/jimmy_lai/big-data-analysis-in-python

Outlines

1. Overview
2. Big Data Analysis Example
3. Scrapy
4. Mongodb
5. Solr
6. Scikit-learn

When Big Data meet Python

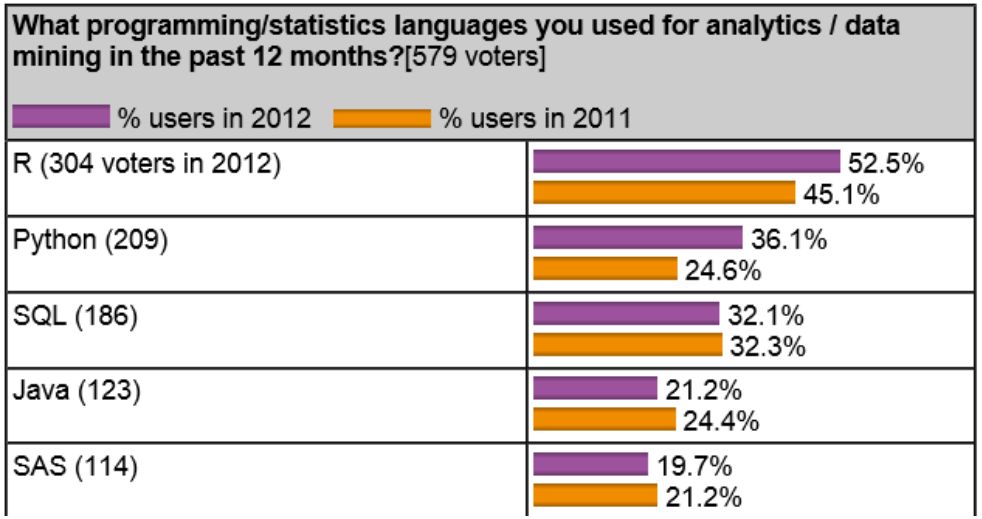


http://www.slideshare.net/jimmy_lai/when-big-data-meet-python

Why Python?

- Good code readability for fast development.
- Scripting language: the less code, the more productivity.
- Application:
  - Web
  - GUI
  - OS
  - Science

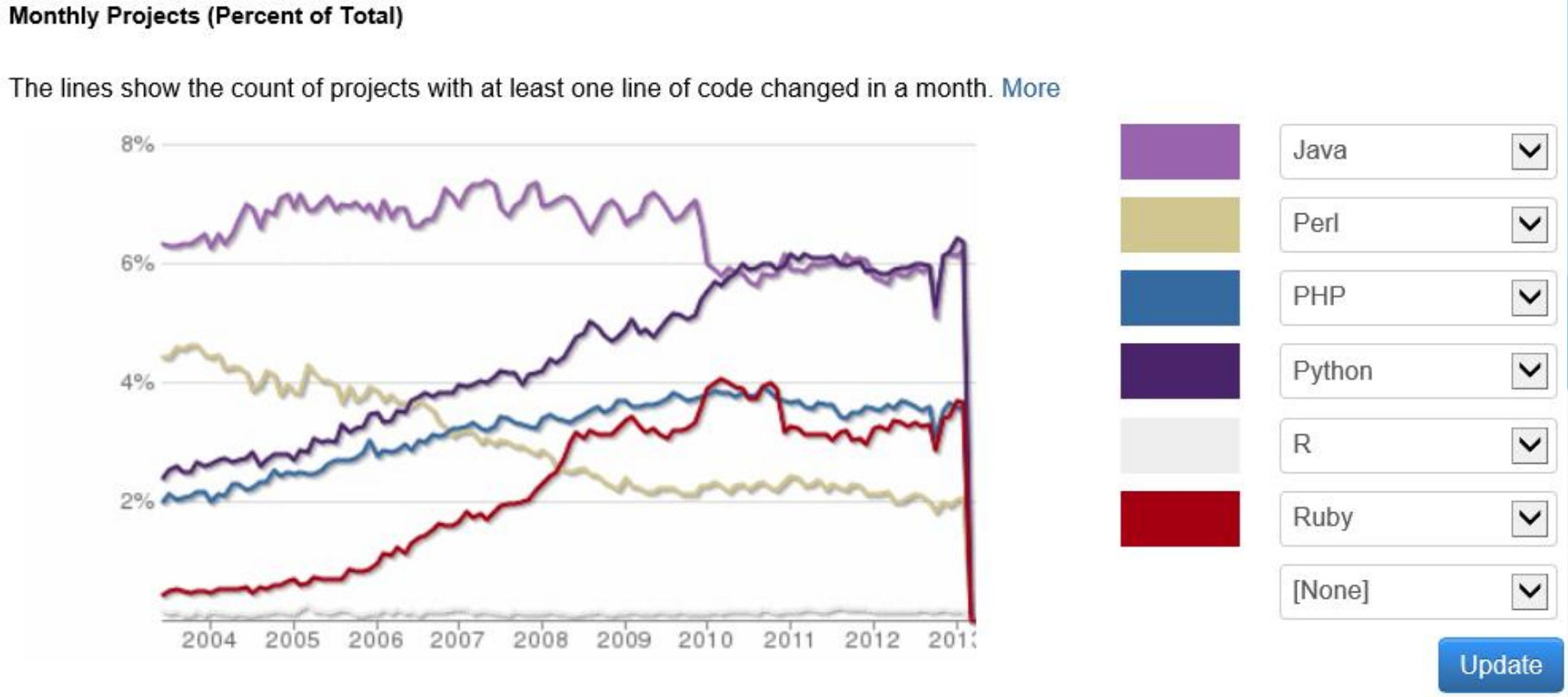
- Fast growing among open source communities.
  - Commits statistics from ohloh.net



http://www.kdnuggets.com/2012/08/poll-analytics-data-mining-programming-languages.html

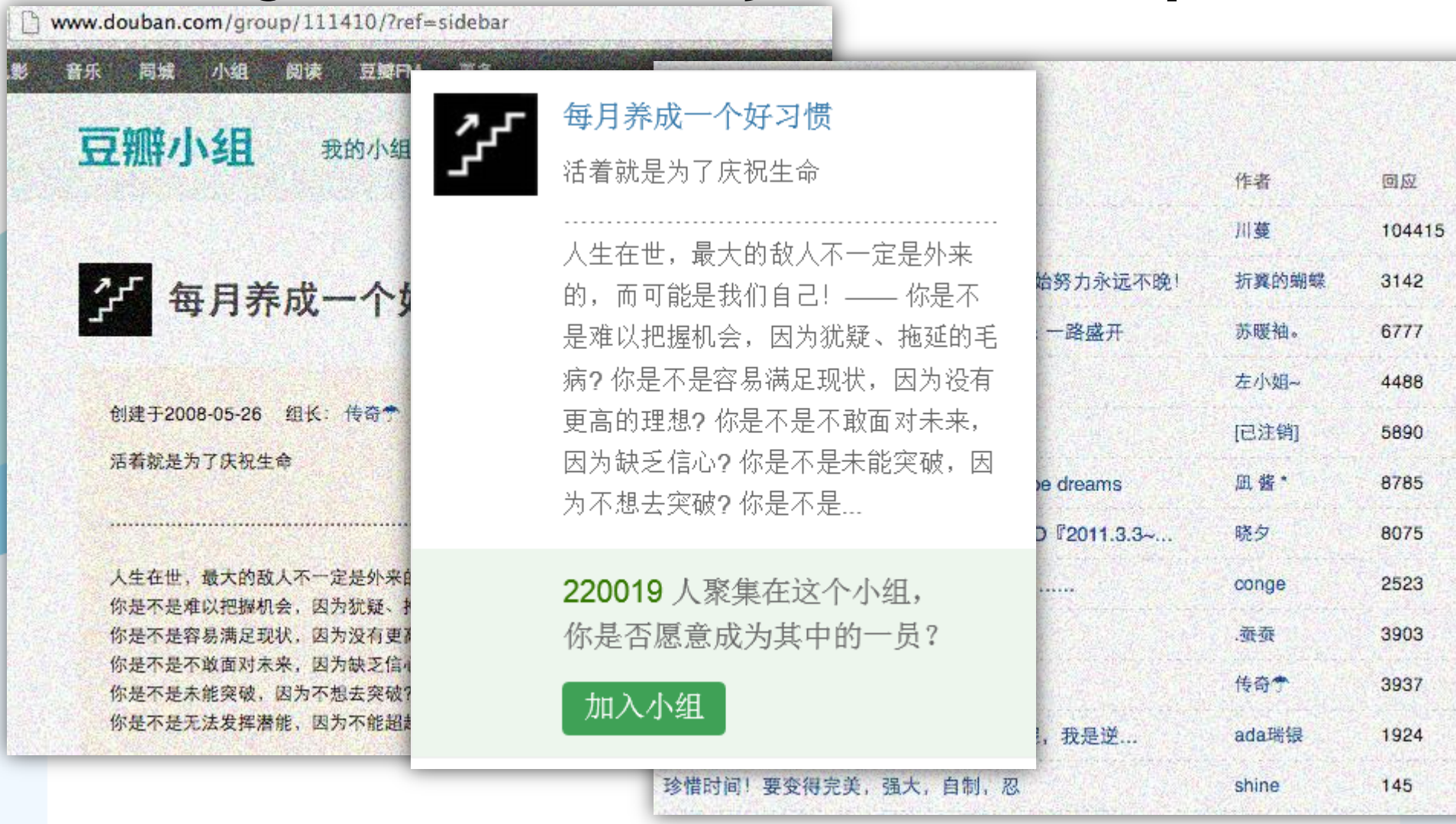
Why Python?

- The support of open source projects



http://www.ohloh.net/languages/compare?measure=projects&percent=true&l0=java&l1=perl&l2=php&l3=python&l4=r&l5=ruby&l6=-1&l7=-1&commit=Update

Big Data Analysis Example



Big Data Analysis Example

Scrapy web scraping framework

http://scrapy.org/

- pip install scrapy
- Parse field by XPath
- Spider
  - Request
  - Response
  - Parse_function
  - Pipeline
- For **Big Data**:
  - Parallel crawling

Traverse web pages

Scrapy web scraping framework

http://scrapy.org/

Scrapy

```
class GroupSpider(BaseSpider):
    name = "group"
    allowed_domains = ["douban.com"]
    start_urls = ["http://www.douban.com/group/111410/discussion?start=%d" % i
for i in range(10000, 20000, 25)]

    def __init__(self):
        self.coll = get_coll()

    def parse(self, response):
        hxs = HtmlXPathSelector(response)
        for tr in hxs.select('//tr'):
            title, url, author, author_url = None, None, None, None
            result = tr.select('td[@class="title"]')
            if len(result) == 1:
                url = result.select('a/@href').extract()[0]
                if self.coll.find_one({'url': url}) is not None:
                    self.log('duplicated url: %s' % (url))
                else:
                    yield Request(url, callback=self.parse_article)
```

Traverse web pages

Extract field by XPath

Document based NoSQL database

http://www.mongodb.org/

- Python client: pip install pymongo
- Json style document
- For **Big Data**:
  - Mongodb cluster with shard and replica
- article:
  - author
  - author_url
  - title
  - content
  - comments
  - url

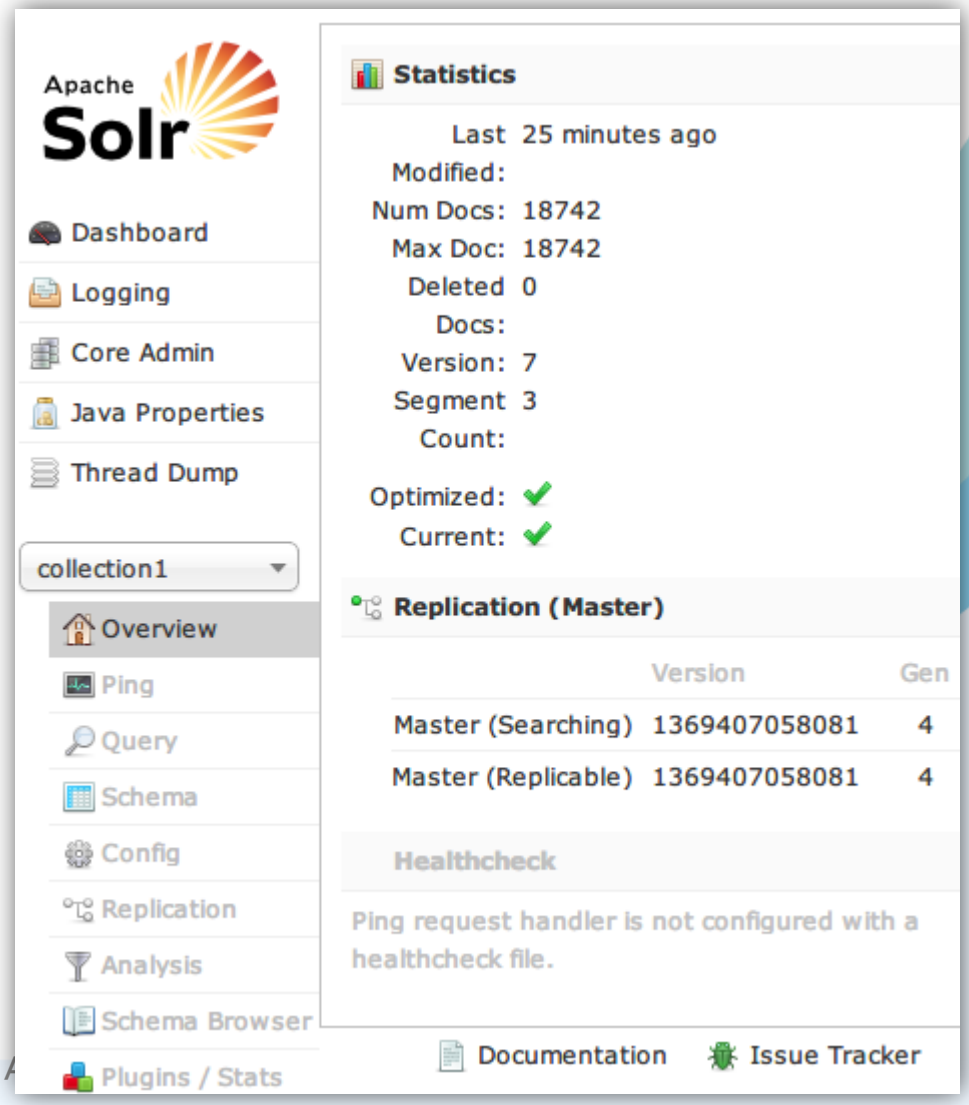
```
import pymongo

conn = pymongo.Connection()
coll = conn.database.collection
coll.ensure_index('url', unique=True)
article = {'author': u'小明', 'author_url':
'http://host/path1', 'comments': [],
        'title': u'天天向上', 'content': '', 'url':
'http://host/path2'}
coll.insert(article)
results = [i for i in coll.find({'title': u'天天
向上'})]
```

Apache Solr Full-text search engine

http://lucene.apache.org/solr/

- Customize full-text indexing and searching by xml config
- RESTful API for select/update index
- Spatial Search
- For **Big Data**:
  - SolrCloud: distributed indexs



schema.xml

```
<field name="url" type="text_general" indexed="true" stored="true"/>
<field name="title" type="text_general" indexed="true" stored="true"/>
<field name="content" type="text_general" indexed="true" stored="true"/>
<field name="comments" type="text_general" indexed="true" stored="true"
multiValued="true"/>
<field name="author" type="text_general" indexed="true" stored="true"/>
<field name="author_url" type="text_general" indexed="true" stored="true"/>
```

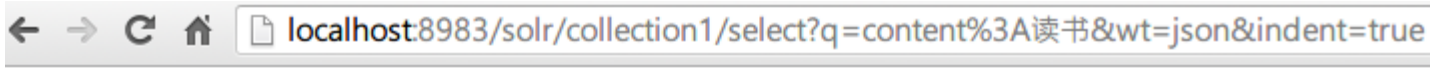
Define fileds and types

```
<copyField source="title" dest="text"/>
<copyField source="author" dest="text"/>
<copyField source="content" dest="text"/>
<copyField source="comments" dest="text"/>
```

Define index process

```
<fieldType name="text_smart_chinese" class="solr.TextField"
positionIncrementGap="100">
        <tokenizer class="solr.SmartChineseSentenceTokenizerFactory"/>
        <filter class="solr.SmartChineseWordTokenFilterFactory"/>
        <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt"
enablePositionIncrements="true" />
        <filter class="solr.LowerCaseFilterFactory"/>
        <filter class="solr.PorterStemFilterFactory"/>
</fieldType>
```


```
<uniqueKey>url</uniqueKey>
```




```
{
  + responseHeader: {…},
  - response: {
      numFound: 7834,
      start: 0,
    - docs: [
        - {
            author: "追筑",
            url: "http://www.douban.com/group/topic/28792422/",
            title: " 【好好生活】读书笔记 ",
            content: "每天坚持读书，坚持思考。",
            author_url: "http://www.douban.com/people/1393829/",
            _version_: 1435927368971583500
          },
        - {
            author: "Wise海綿",
            url: "http://www.douban.com/group/topic/26795128/",
            title: " 要早睡早起，坚持贴 ",
            content: "早睡早起，每日读书",
            author_url: "http://www.douban.com/people/NinaWang_/",
            _version_: 1435927371653841000
          },
        - {
            author: "糖糖",
            url: "http://www.douban.com/group/topic/27373910/",
            title: " 【送书】地址给我，我把书免费邮给你 （关于外公的） ",
            content: "读书会第二期：《外公与我》试读 点链接 ",
            author_url: "http://www.douban.com/people/57500408/",
            _version_: 1435927372410912800
          },
```

Popular Book Ranking List

By regex search in Mongodb	By full-text search in Solr
少有人走的路	那些年我们一起追的女孩
百年孤独	了不起的盖茨比
遇见未知的自己	我不要你死于一事无成
平凡的世界	考拉小巫的英语学习日记
送你一颗子弹	高效能人士的7个习惯
小王子	哪来的天才
拖延心理学	被嫌弃的松子的一生
活着	与众不同的心理学
苏菲的世界	奇特的一生
红楼梦	普罗旺斯的一年
目送	接纳不完美的自己
如何阅读一本书	我就是想停下来，看看这个世界
围城	这些都是你给我的爱

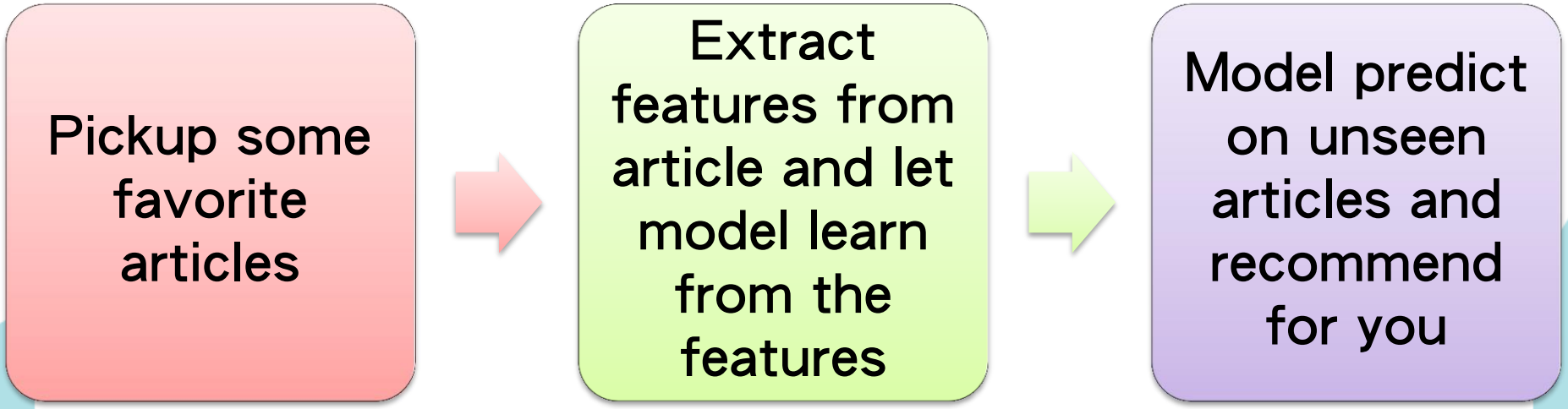
Search “《[^》]+》” pattern

Machine Learning

http://scikit-learn.org/

- Machine learn algorithms
  - Supervised learning
  - Unsupervised learning
- Processing
  - Model selection
  - Pipeline
- Application: Article Recommendation
  - Let computer figures out your interests, and then recommend articles for you.
  - Learning by OneClassSVM

Article Recommendation




```
articles = [coll.find_one({'url': url})['content'] for url in urls]
tfidf = TfidfVectorizer(tokenizer=my_tokenizer,
ngram_range=(1, 3))
svm = OneClassSVM(kernel="linear", nu=0.3)
train_vectors = tfidf.fit_transform(articles)
svm.fit(train_vectors)
svm.predict(test_vectors)
```

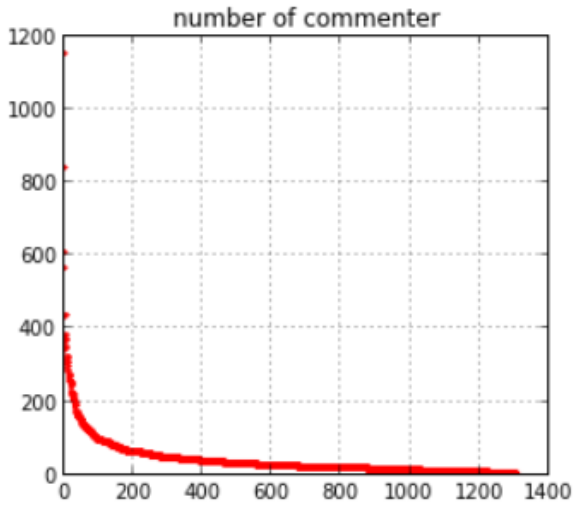
- **Result: recommend 390 from 8000 articles, and the recommended articles are good to me.**

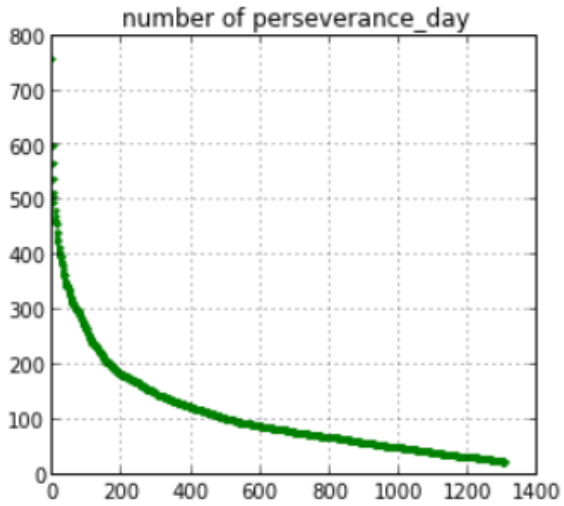
Numerical Data Analysis

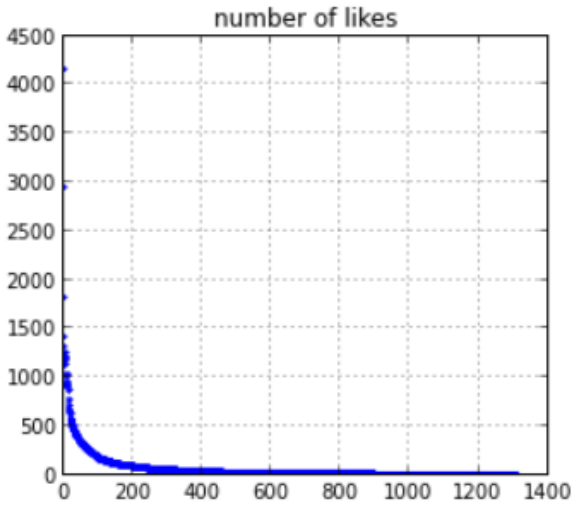
http://pandas.pydata.org/

- Data Structure:
  - Series
  - Frame
- Operations: slicing, indexing, subsetting

```
from pandas import Series
import matplotlib.pyplot as plt
sa = Series(index=range(len(commenters)),
data=commenters)
fig, axs = plt.subplots(1, 3)
sa.plot(ax=axs[0], title='number of
commenter', style='r.')
```







Reference (1/2)

- Book:
  - Wes McKinney, “Python for Data Analysis”, O´Reilly, 2012
  - Toby Segaran, “Programming Collective Intelligence: Building Smart Web 2.0 Applications”, O´Reilly, 2008
  - Philipp K. Janert, “Data Analysis with Open Source Tools”, O´Reilly, 2010
- Coursera:
  - Web Intelligence and Big Data, url
  - Introduction to Data Science, url
  - Data Analysis, url

Reference (2/2)

- Conference:
  - PyData 2013, http://pydata.org/abstracts/
  - PyData 2012, http://marakana.com/s/post/1090/2012_pydata_workshop
- My former shares:
  - When Big Data Meet Python, COSCUP 2012, slides
  - NLTK: natural language toolkit overview and application, PyCon.tw 2012, slides