Recap • February 26, 2014


BINGO
text analytics
big data
data cloud
Cassandra
data volume
realtime analytics
visualization
Whirr
ZooKeeper
unstructured data
NoSQL
Kafka
FREE
columnar database
big data science
Sqoop
in-memory analytics
BASE
Map/Reduce
geospatial analysis
small data
in-memory database
yottabyte
HBase
data velocity
The game draws a new Bingo! card when loaded. Click RELOAD to make a new card. Enable your browser's JavaScript feature to see the card. This page may not work with old versions Microsoft Internet Explorer; please upgrade.

Brian Granger (Cal Poly SLO)

"Compose and share reproducible stories that involve code and data"

IPy IPython Dashboard | IPy spectrogram

127.0.0.1:8888/a5222740-848b-4ac1-b212-d732c9f8f78b

IP[y]: Notebook spectrogram Last saved: Mar 07 11:14 PM

File Edit View Insert Cell Kernel Help

Markdown

Simple spectral analysis

An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

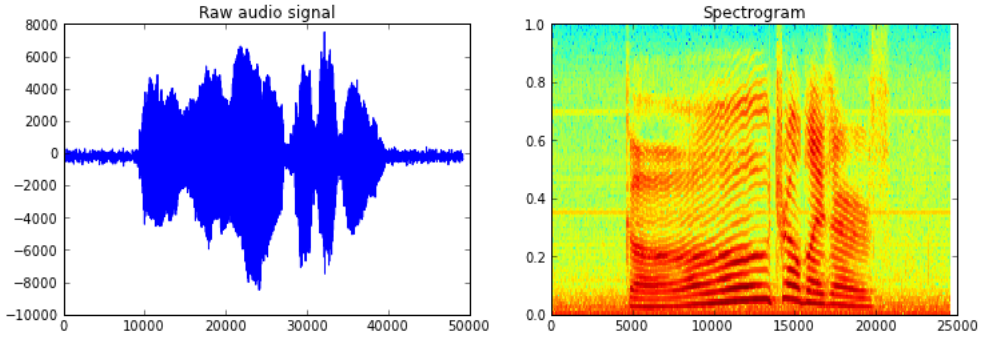
using windowing, to reveal the frequency content of a sound signal.

We begin by loading a datafile using SciPy's audio file support:

In [1]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

In [2]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```



Interactive visualization

coming in IPython 2.0

- Vincent/**Vega**/**d3**
- **mpld3**
- **plotly**

Architecture

- js in browser
- asynchronous, bi-directional JSON messaging (WebSockets/ZeroMQ)
- kernel (Python interpretter)

Multilanguage support

coming soon, a drop down menu for what kernel language you'd like to use

- Julia
- R
- Matlab / Octave
- Scala

- **Brian Granger's slides** are an IPython notebook
- Lots of examples at **nbviewer.ipython.org**

Scikit-learn

Olivier Grisel (Parietal, INRIA)

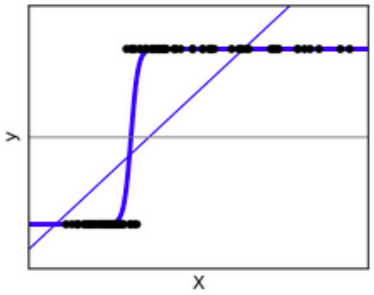
Machine learning in Python

from INRIA

Classification

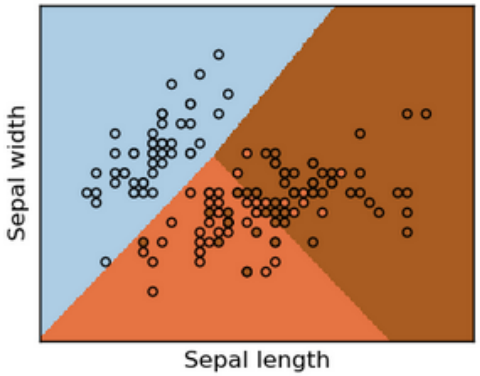
For classification, as in the labeling iris task, linear regression is not the right approach as it will give too much weight to data far from the decision frontier. A linear approach is to fit a sigmoid function or **logistic** function:

$$y = \text{sigmoid}(X\beta - \text{offset}) + \epsilon = \frac{1}{1 + \exp(-X\beta + \text{offset})} + \epsilon$$




```
>>> logistic = linear_model.LogisticRegression(C=1e5)
>>> logistic.fit(iris_X_train, iris_y_train)
LogisticRegression(C=100000.0, class_weight=None, dual=False,
          fit_intercept=True, intercept_scaling=1, penalty='l2
',
          random_state=None, tol=0.0001)
```

This is known as **`LogisticRegression`**.



dplyr and ggvis

Hadley Wickham (RStudio)

More awesome R packages from Hadley.

- data manipulation
- interactive visualization

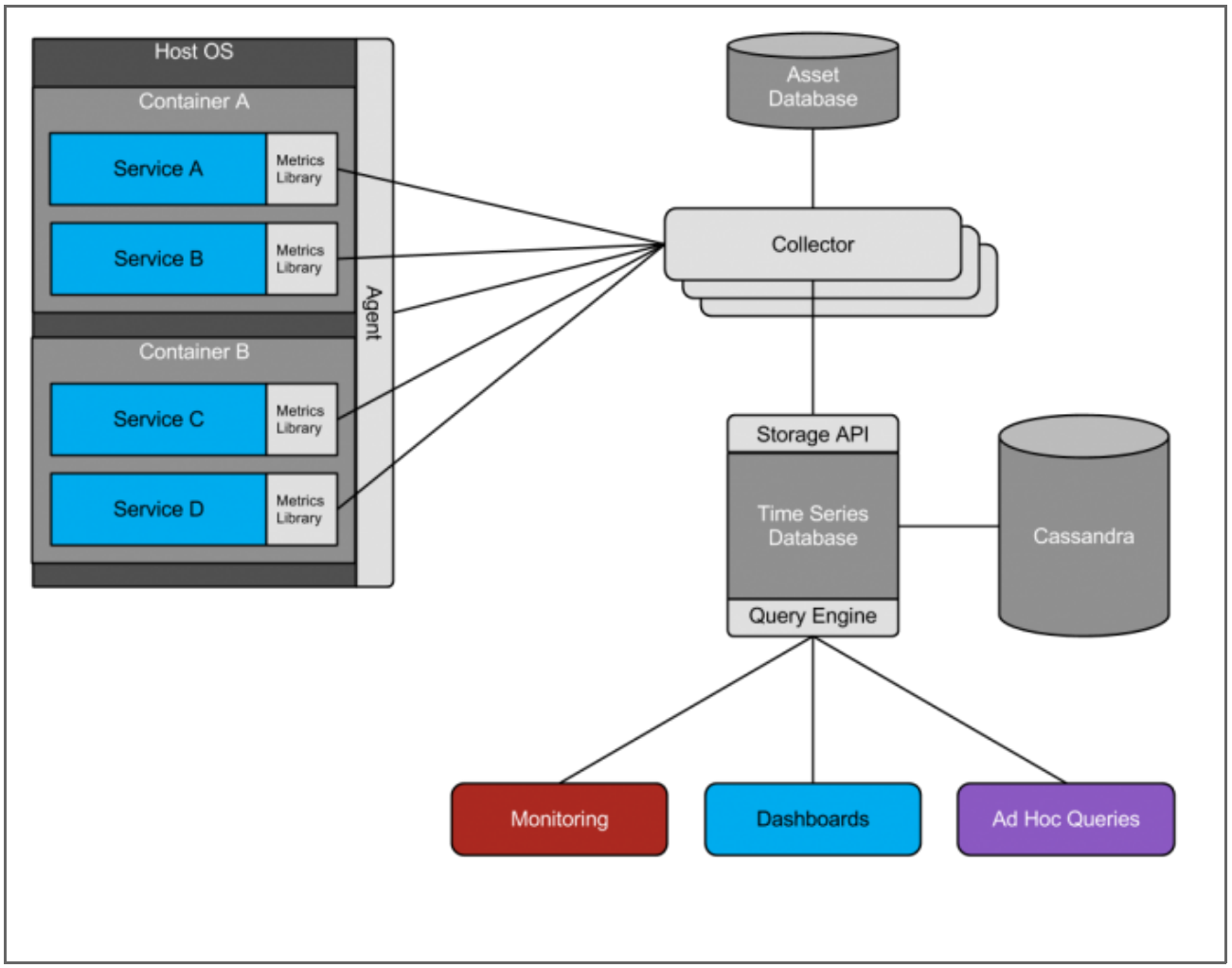
Twitter's Observability stack

Yann Ramin (Twitter)

- “Moving beyond logging” (logging and parsing are expesive)
- Instrument applications to collect runtime statistics

~

Just like you design for testability, design for monitoring


Host OS
Container A
Service A
Metrics Library
Service B
Metrics Library
Container B
Service C
Metrics Library
Service D
Metrics Library
Agent
Asset Database
Collector
Storage API
Time Series Database
Query Engine
Cassandra
Monitoring
Dashboards
Ad Hoc Queries

Stack

- instrumentation (**Ostrich**, **Finagle**)
- storage (Cassandra[†])
- query engine (Cuckoo)
- vizualization
- analytics

† cassandra-like key/value store, fronted by an API over caches, queues, storage

Ostrich

Increment a counter

```
Stats.incr("some_important_counter")
```

Ostrich

Store the value of a variable at a particular time

```
Stats.addGauge("current_temperature") {
  myThermometer.temperature
}
```

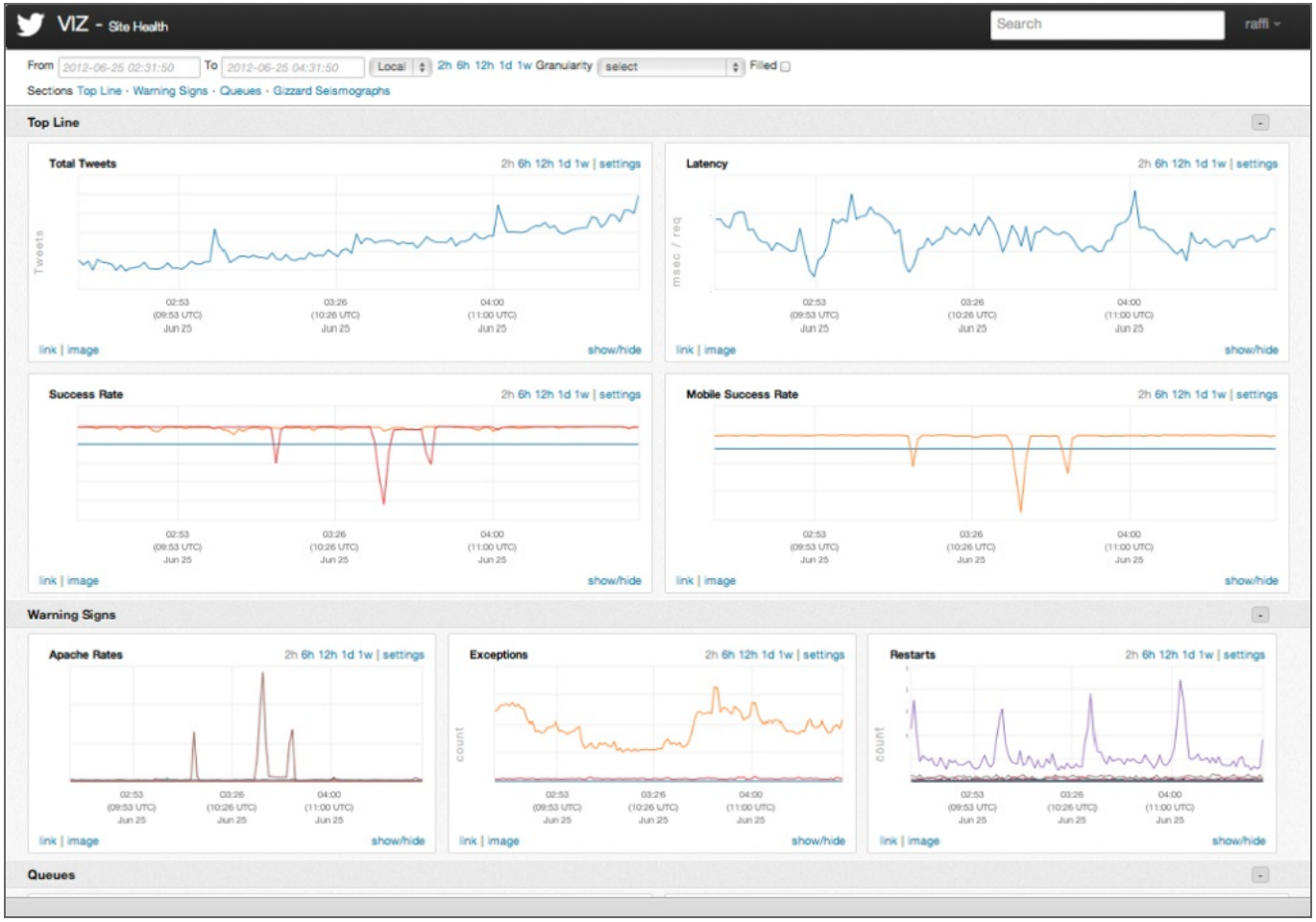
Ostrich

Record running time

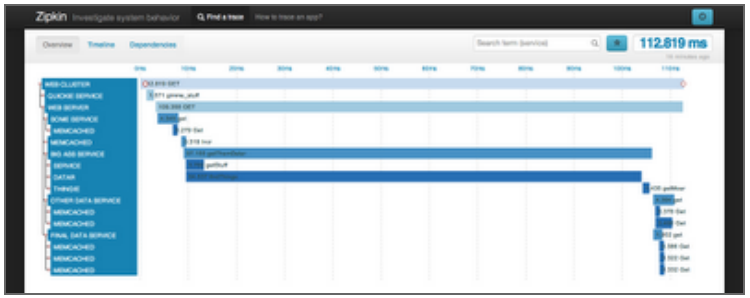
```
Stats.time("translation") {
  document.translate("de", "en")
}
```

Keys

(service,source(host,process),metric)


VIZ - Site Health
Search
raffi
From 2012-06-25 02:31:50 To 2012-06-25 04:31:50 Local 2h 6h 12h 1d 1w Granularity select Filled
Sections Top Line · Warning Signs · Queues · Gizzard Seismographs
Top Line
Total Tweets
2h 6h 12h 1d 1w | settings
Tweets
02:53 (09:53 UTC) Jun 25
03:26 (10:26 UTC) Jun 25
04:00 (11:00 UTC) Jun 25
link | image
show/hide
Latency
msec / req
Success Rate
Mobile Success Rate
Warning Signs
Apache Rates
Exceptions
count
Restarts
Queues

Zipkin tracing for distributed systems (modeled after Google Dapper)



Trained systems

Chris Ré (Stanford)

Examples of trained systems:

- IBM's Watson
- Google's Knowlege Graph
- Netflix recommender engine

- RDBMS + machine learning, stats, optimization
- combining structured and unstructured data

DeepDive

Extracting structured information from unstructured sources such as raw text

DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference

Feature engineering

"Good features allow a simple model to beat a complex model"

Brainwash: A Data System for Feature Engineering

~

† similar to what we observe in challenges

Industry strategy: copy Google

Google	**Open**
MapReduce	Hadoop
GFS	HDFS
BigTable	HBase
Dremel	Impala
Pregel	Giraph

Giraph

Avery Ching (Facebook)

"iterative graph processing system"

Model of Computation

Two phases

- compute on each vertex
- send messages to other vertices

~

...iterate until halting criterea reached

Examples

- Page rank
- friends of friends

Matei Zaharia (MIT, DataBricks)

Spark is a next-generation Hadoop, in-memory computing

- Written in Scala
- APIs for Scala, Java and Python
- Shark, distributed SQL query engine for Spark (like Hive)
- can read from HDFS, HBase or Cassandra
- Runs on top of Hadoop YARN, Mesos or EC2

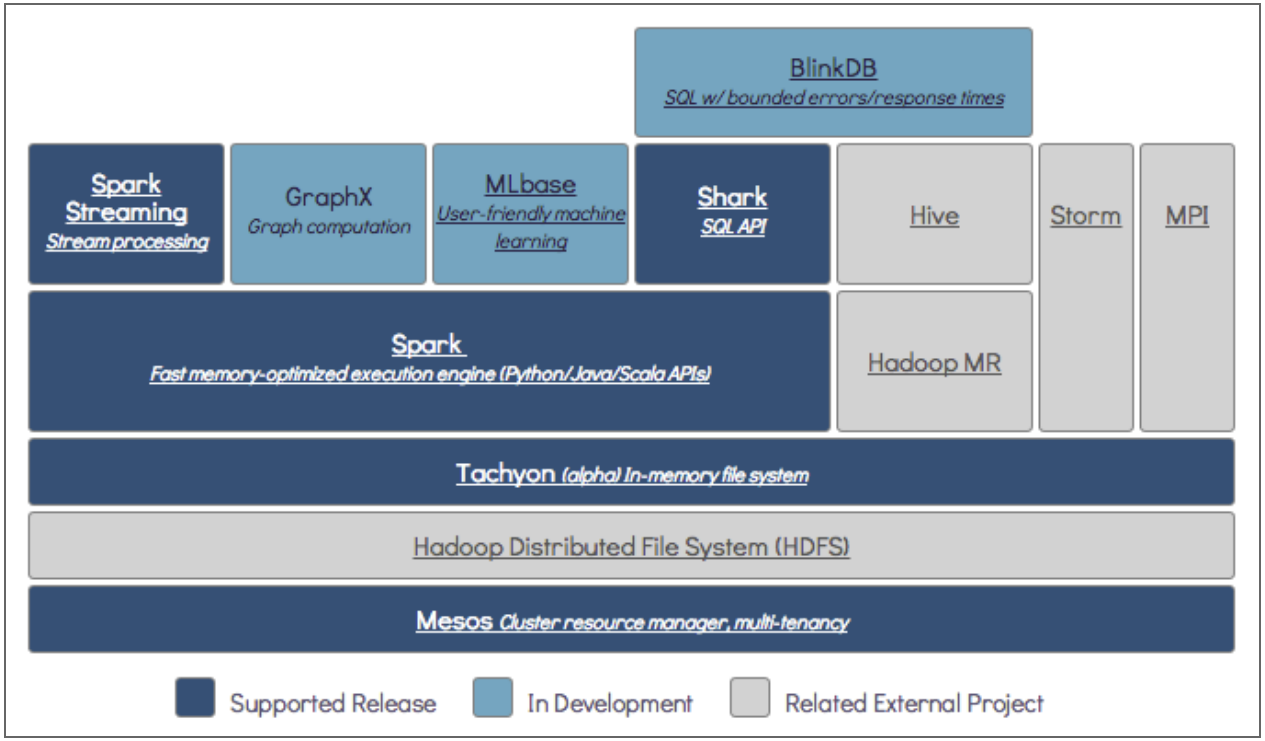
Berkeley Data Analytics Stack

Ameet Talwalkar (UC Berkeley)

BDAS is a suite of software created by the **AMPLab**

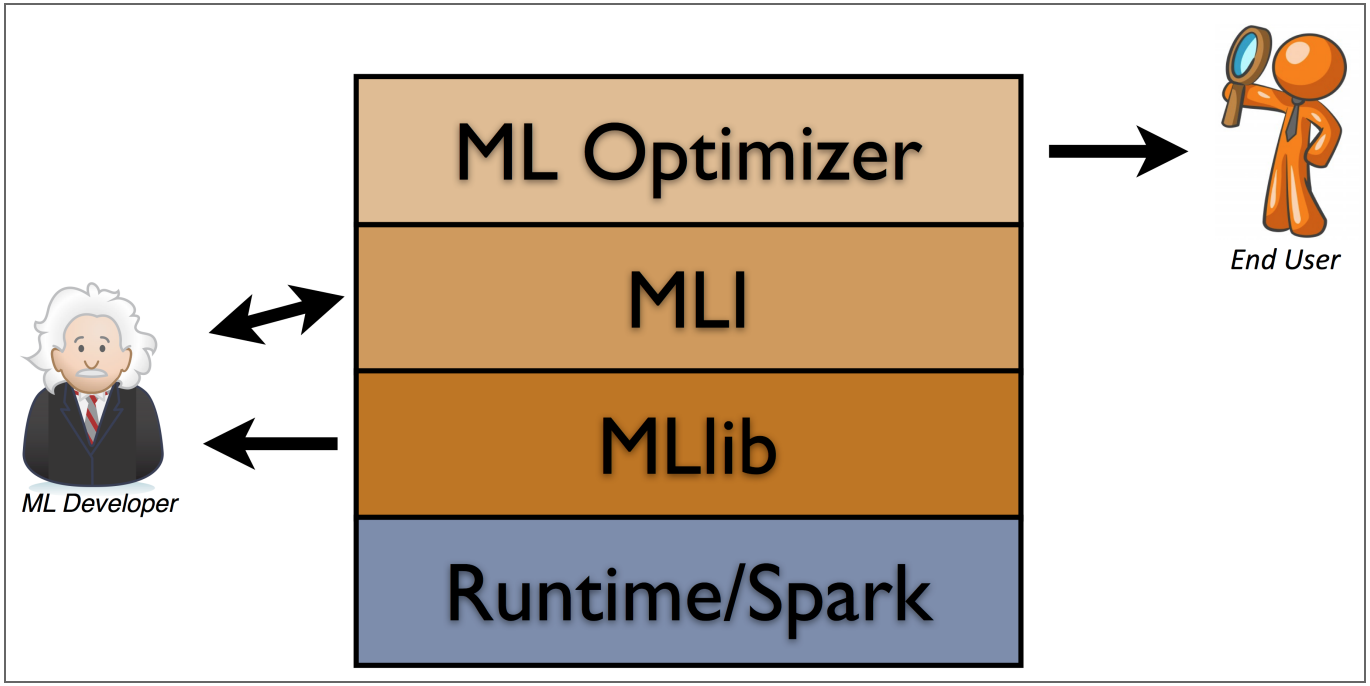
Data-intensive distributed computing stack

- Queries / Machine Learning / Graph processing
- Task scheduler
- Distributed storage
- Resource management


BlinkDB
SQL w/ bounded errors/response times
Spark Streaming
Stream processing
GraphX
Graph computation
MLbase
User-friendly machine learning
Shark
SQL API
Hive
Storm
MPI
Spark
Fast memory-optimized execution engine (Python/Java/Scala APIs)
Hadoop MR
Tachyon (alpha) In-memory file system
Hadoop Distributed File System (HDFS)
Mesos Cluster resource manager, multi-tenancy
Supported Release
In Development
Related External Project

MLBase

MLBase is a machine learning libraries that run over Spark (like Mahout/Hadoop)

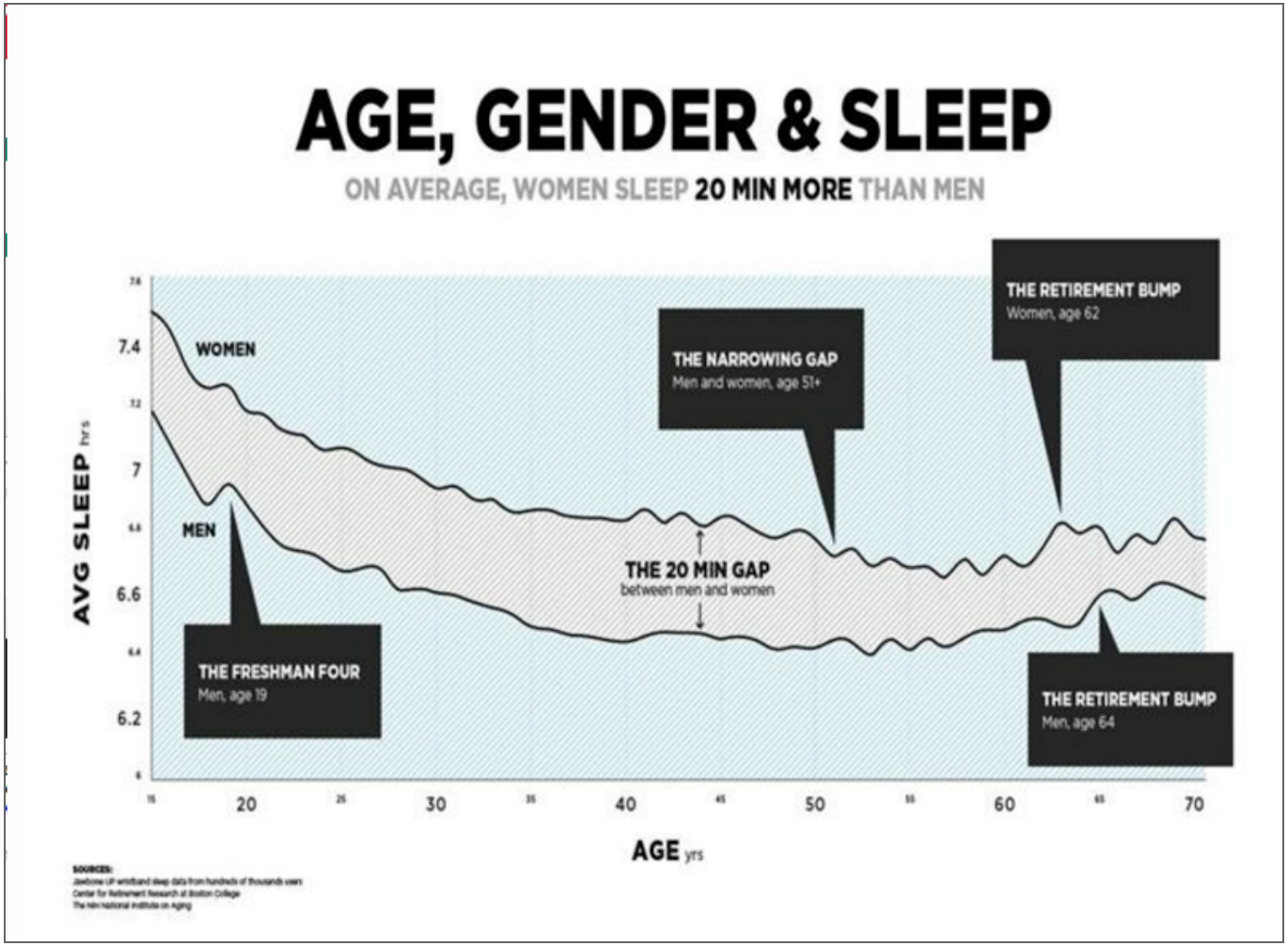


Jawbone UP

Monica Rogati (Jawbone)

Quantified Self

- 50 million nights of sleep
- 500 billion steps


AGE, GENDER & SLEEP
ON AVERAGE, WOMEN SLEEP 20 MIN MORE THAN MEN
AVG SLEEP hrs
7.4
7
6.6
6.2
WOMEN
MEN
THE NARROWING GAP
Men and women, age 51+
THE RETIREMENT BUMP
Women, age 62
THE 20 MIN GAP
between men and women
THE FRESHMAN FOUR
Men, age 19
THE RETIREMENT BUMP
Men, age 64
20
30
40
50
60
70
AGE yrs
SOURCES:

THE END


DO WE HAVE ANY ACTIONABLE ANALYTICS FROM OUR BIG DATA IN THE CLOUD?