ReactoData

MACHINE LEARNING ON TOP OF HADOOP AND OTHER FLAVORS OF BIG DATA

a.k.a.

MACHINE LEARNING SOBRE HADOOP Y OTROS "SABORES" DE BIG DATA

WHOAMI

Federico Leven

- @ ReactoData
- Big Data sobre Open Source desde 2012
- Coordinador meetups Big Data (http://www.iaar.site), etc.
- federico@reactodata.net
- Skype: @federicol
- Twitter: @reactodata
- Linkedin : https://www.linkedin.com/in/federicoleven/

ReactoData

..... plataformas Big Data sobre Cloud y desarrollo de soluciones sobre Hadoop, consultoria y soporte.

- Machine Learning
- UX/UI y aplicaciones móviles
- Big Data y Hadoop
- Cloud

Agenda

- La irrazonable efectividad de los datos
- De la laptop al cluster
- Hadoop y Spark
- Frameworks de Machine Learning sobre Big Data
- Casos de Uso
- Bibliografía y referencias
- Preguntas, sugerencias y quejas

BUZZCONF != BUZZWORD

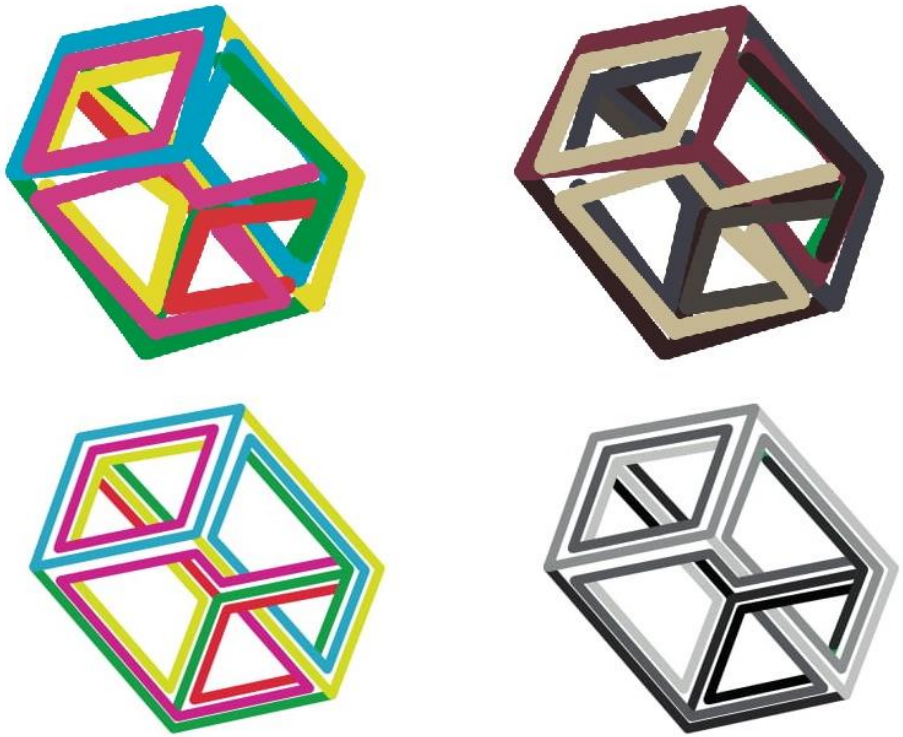
- Buzzword

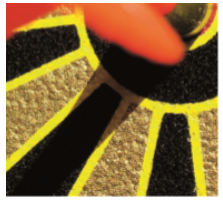
  Una palabra o frase altisonante, usualmente técnica, que no tiene mucho sentido o un sentido vago, usada para impresionar a los legos en el tema.

  - Machine Learning
  - Big Data
  - Cloud
  - Blockchain
  - …

- Usadas en un contexto apropiado, captan una idea de una tecnología o concepto tecnológico que muchas veces es demasiado complejo de expresar en términos simples.

La irracional efectividad de los datos





The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, *Google*

Eugene Wigner's article "The Unreasonable Effectiveness of Mathematics in the Natural Sciences"[1] examines why so much of physics can be neatly explained with simple mathematical formulas such as $f = ma$ or $e = mc^2$. Meanwhile, sciences that involve human beings rather than elementary particles have proven more resistant to elegant mathematics. Economists suffer from physics envy over their inability to neatly model human behavior. An informal, incomplete grammar of the English language runs over 1,700 pages.[2] Perhaps when it comes to natural language processing and related fields, we're doomed to complex theories that will behavior. So, this corpus could serve as the basis of a complete model for certain tasks—if only we knew how to extract the model from the data.

Learning from Text at Web Scale

The biggest successes in natural-language-related machine learning have been statistical speech recognition and statistical machine translation. The reason for these successes is not that these tasks are easier than other tasks; they are in fact much harder than tasks such as document classification that extract just a few bits of information from each document. The reason is that translation is a natural task routinely done every day for a real human need (think of the operations of the European Union or

"La irracional efectividad de los datos"
Año 2009
Norvig, Pereira, Halevy (Google)
"Reconocimiento de lenguaje natural"

Tres ideas principales

"Follow the data"

…with very large data sources, the data holds a lot of detail…

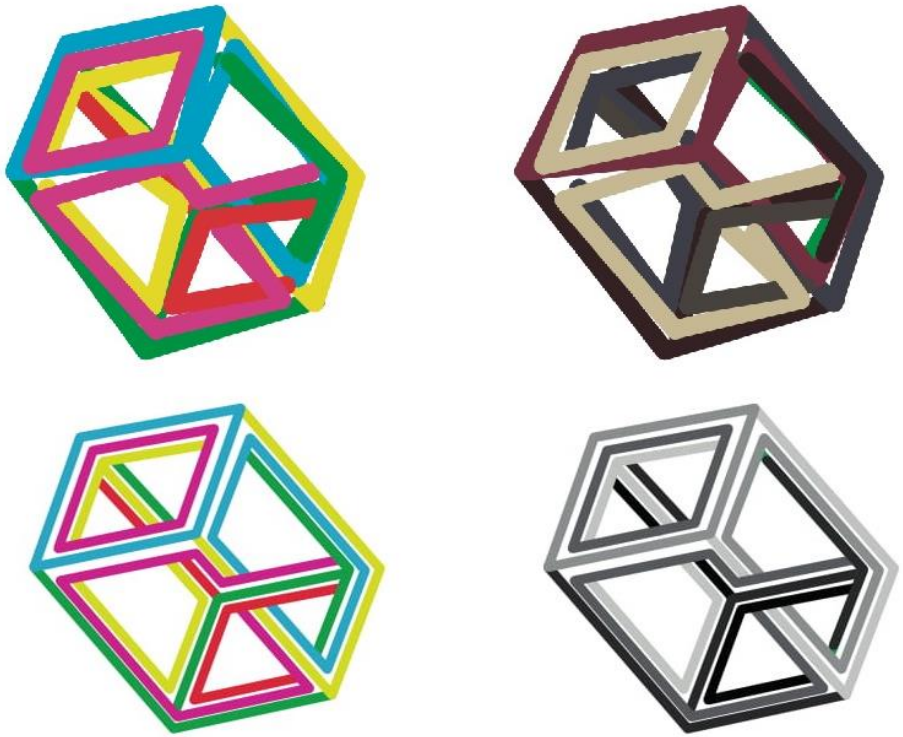
"Modelos complejos vs más datos"

…For those who were hoping that a small number of general rules could explain language…

"No supervisado + datos no etiquetados"

…Choose a representation that can use unsupervised learning on unlabeled data, which is so much more plentiful than labeled data…

De la laptop al cluster

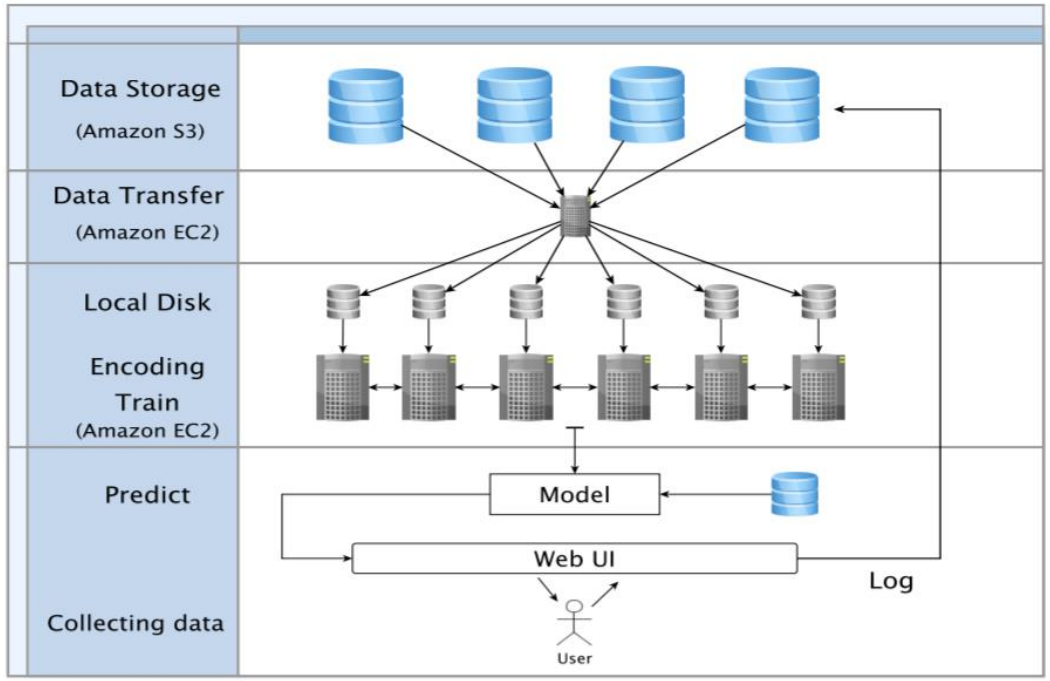


Procesamiento en un único nodo = mi laptop

- Es la forma habitual del inicio del trabajo de científico de datos
- No en todos los casos estamos frente a un problema de Big Data
- Es “simple” de llevar adelante con multiples IDEs y frameworks para esto
- Buen sampleo de datos, datos de volumen reducido, programación de procesamiento paralelo eficiente.

Procesamiento en un cluster

- Retomando la idea del artículo
  "Follow the data"
- Más datos = más memoria RAM
- Escalar verticalmente tiene un límite
- Algunos modelos requieren entrenarse en forma constante con SLA estrictos



- Single Node Read Time (1 TB) >> Single Node Read Time (100 GB) x 10 nodes

Ejemplo (Click-Through Rate)

- Uso en Online Advertising y Perfomance Advertising

- Definición de CTR:

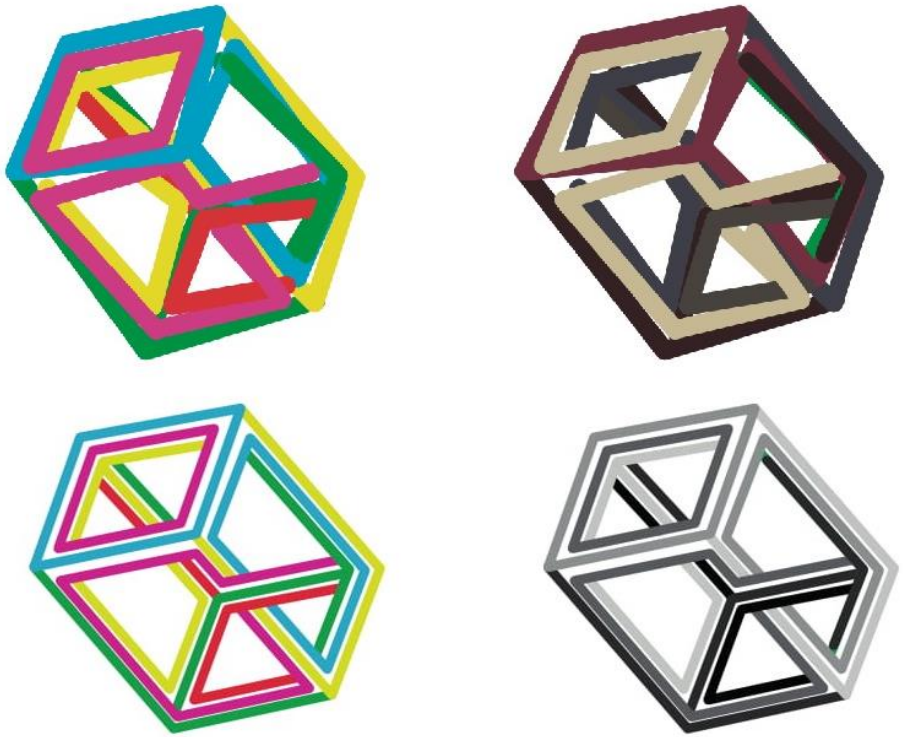
$$\text{CTR} = \frac{\text{\# clicks}}{\text{\# impressions.}}$$

- Input : Una secuencia de eventos

```
Not clicked + Atributos de usuario
Clicked + Atributos de usuario
Not clicked + Atributos de usuario
 . . . . . .
```

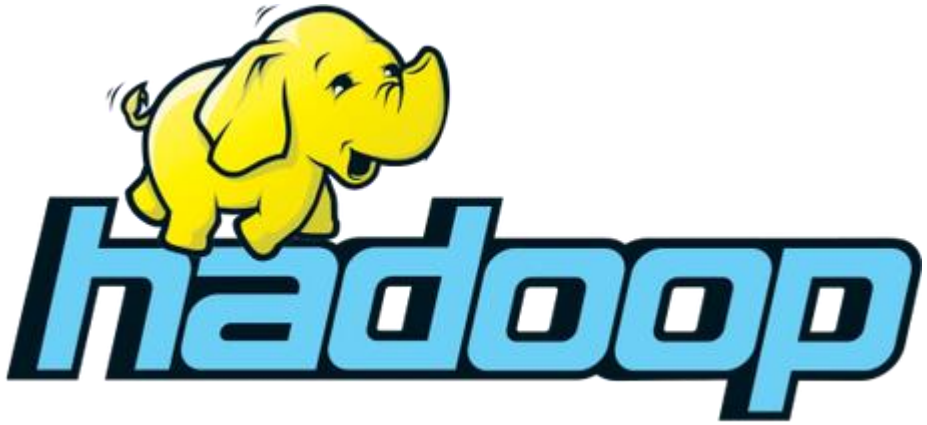
- Un problema de clasificación Binaria (usuario clickeará o no).

Hadoop y Spark



HADOOP

- Proyecto nacido en 2006 desde Yahoo



- Basado en 2 papers de Google:
  - GFS
  - MapReduce

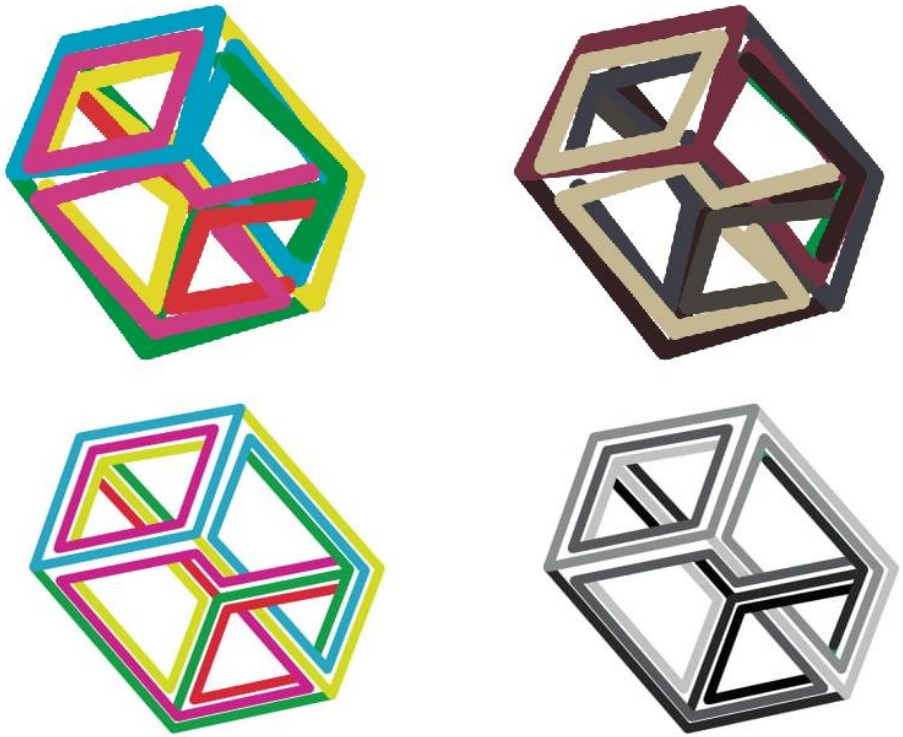
- "Framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models."

- Hadoop Core (hadoop.apache.org)
  - HDFS
  - YARN
  - MapReduce

SPARK

- Proyecto nacido en 2009
  UC Berkeley LAB
- Independiente de Hadoop pero con una integración completa.
- “Apache Spark™ is a unified analytics engine for large-scale data processing”
- Spark Core – SQL - Streaming – MLlib – GraphX (spark.apache.org)

Frameworks de Machine Learning sobre Big Data



TensorFlow sobre Hadoop (HDFS)

- Permite procesar archivos guardados en HDFS

- La configuración requiere :
  a) Configurar TensorFlow en modo distribuido
  b) Que los workers corren en los nodos de Hadoop
  c) Que TF tenga acceso a las variables de Hadoop (HDFS_HOME, etc) y a las librerias

```
filename_queue = tf.train.string_input_producer([
    "hdfs://namenode:8020/path/to/file1.csv",
    "hdfs://namenode:8020/path/to/file2.csv",
])
```

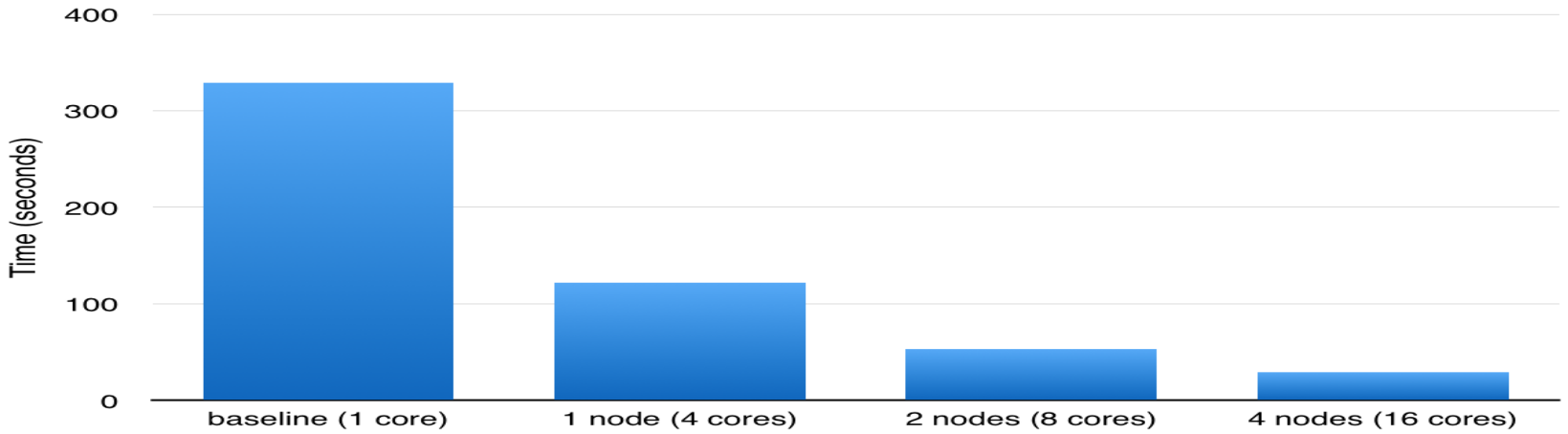
TensorFlow sobre Spark

- Proyecto de Yahoo para correr TensorFlow sobre Hadoop y Spark.
- Con pequeñas modificaciones se puede correr código distribuido de TF sobre Spark integrado con Hadoop para acceder a datos via HDFS
- Soporta CPU y GPU con una mejora de performance de 10x
- Se puede portar a Amazon EMR y tener clusters Spark con TensorFlow a demanda.
- Instalación y configuración todavia “poco amigable”

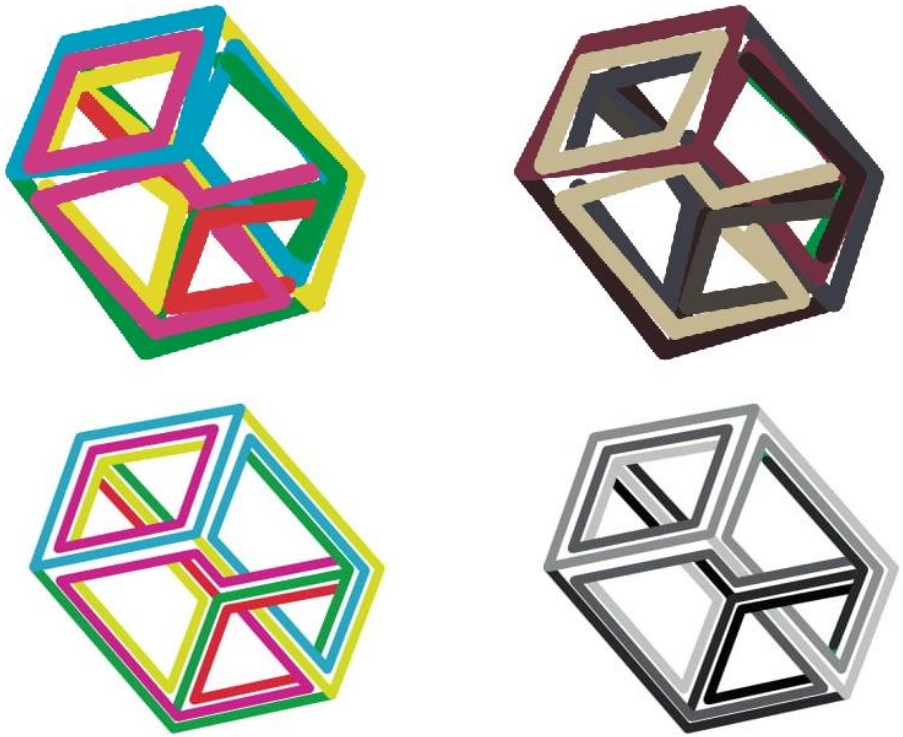
Python Scikit sobre Spark

- Corre como cualquier libreria sobre Spark.
- Se desarrollaron algunos paquetes adicionales para mejorar la integración.
- Databricks prove una integración que require cambios minimos entre el codigo original de scikit-learn y el codigo para Spark

  https://databricks.com/blog/2016/02/08/auto-scaling-scikit-learn-with-apache-spark.html



Casos de uso



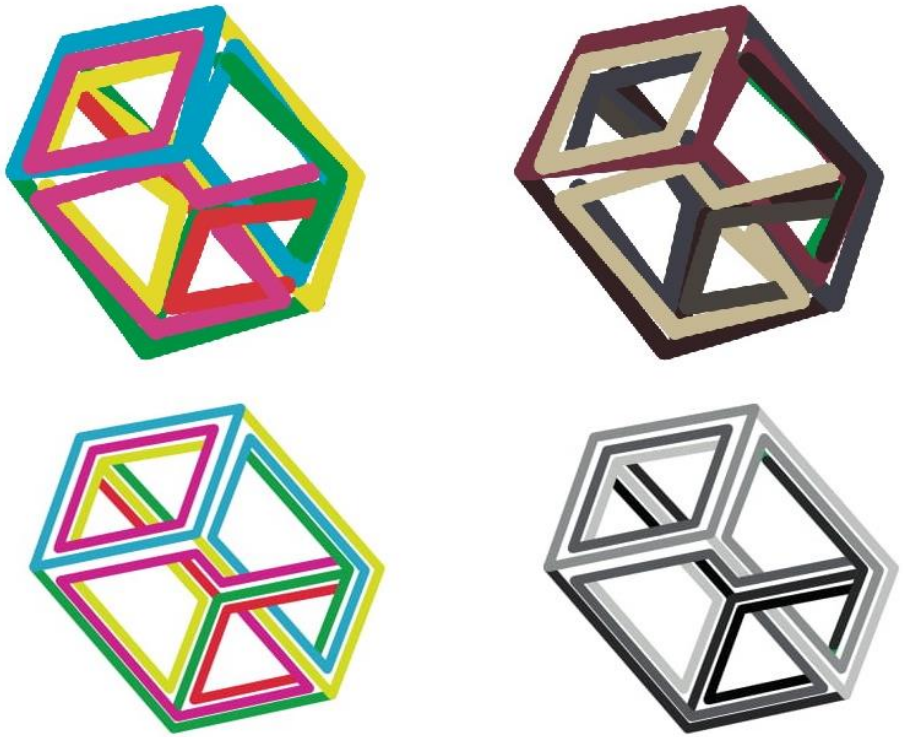
Algunos casos de uso que “ameritan” Big Data

- CTR
  - Predecir CTR
  - Historial de comportamiento de navegación = Volumenes muy grandes
  - Elegir aviso con mas alto CTR.
  - Reentrenar modelo en forma continua

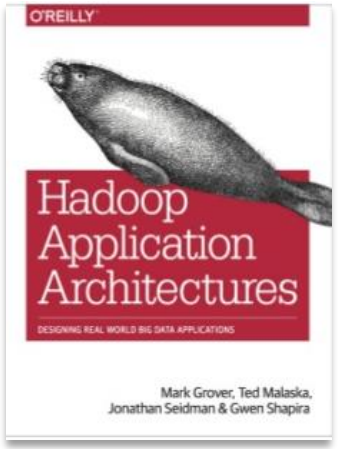
- Marketing Predictivo
  - Encuestas = Preguntas y respuestas.
  - Datos de alta dimensionalidad
  - Predecir respuestas de un entrevistado.

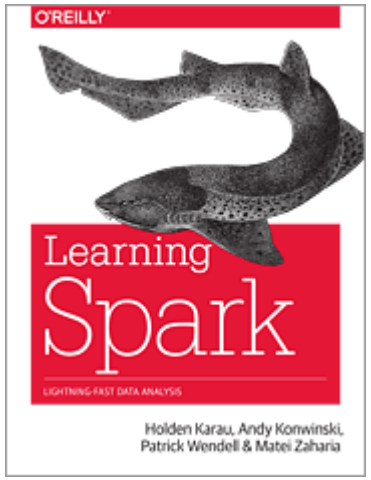
- Procesamiento de imagenes
  - Automatización de volcado de información de facturas y tickets
  - Disparidad de formatos y soportes físicos (factura, ticket, tipografia, impression, etc)
  - Tarea CPU intensiva e I/O intensiva.

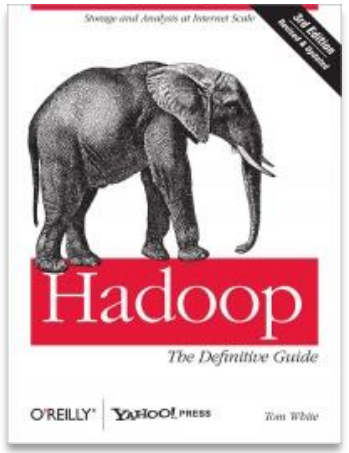
Bibliografia y referencias

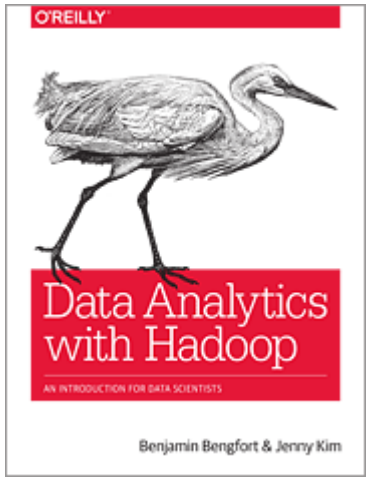


- https://www.tensorflow.org/deploy/hadoop
- https://www.tensorflow.org/deploy/distributed
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7906512
- https://static.googleusercontent.com/media/research.google.com/es//pubs/archive/35179.pdf
- https://databricks.com/blog/2016/02/08/auto-scaling-scikit-learn-with-apache-spark.html
- https://github.com/yahoo/TensorFlowOnSpark









Gracias por quedarse hasta el final !

¿ Preguntas, sugerencias o quejas ?