Comunicación de Datos Utilizando Herramientas Interactivas: Historia de la Crisis Económica (2000 – 2016)

Máster en Data Science – Universidad Rey Juan Carlos

Ignacio Arias Barra
Raúl Sánchez Martín

1 Introducción

El objetivo de la presente práctica es realizar una visualización interactiva de la evolución de la economía durante los años. En concreto, se ha analizado tal aspecto desde el año 2000 hasta el año 2016 en cuatro países diferentes: España, EEUU, Reino Unido y Japón. En cada uno de ellos, se ha querido analizar aspectos tanto macro- como microeconómicos. Para ello, se ha analizado la evolución de diferentes mercados bursátiles de dichos países, que representa la macroeconomía, como los niveles de desempleo, que puede relacionarse con la microeconomía. Para la visualización de este estudio, se ha combinado Bokeh, Kafka y de manera opcional MongoDB. En concreto, se han preparado dos configuraciones diferentes, que serán explicadas en secciones posteriores detallando las ventajas e inconvenientes de cada modalidad. A continuación, se describen los archivos entregados, el proceso de obtención y preparación de datos, y posteriormente, las dos versiones de visualización preparadas. Finalmente, se hará una recopilación de los principales retos que ha conllevado la realización de la presente práctica, especialmente relacionados con el hecho de combinar diferentes tecnologías de Big Data.

2 Archivos entregados

En el presente trabajo, se han entregado los siguientes archivos

- *Memoria_IgnacioArias_RaulSanchez.pdf*: corresponde a la presente memoria
- *visualizacion_v1*: carpeta donde se incluyen todos los archivos necesarios para la ejecución de la versión 1 de la visualización propuesta. Su contenido será detallado en posteriores secciones.
- *visualizacion_v2*: carpeta donde se incluyen todos los archivos necesarios para la ejecución de la versión 2 de la visualización propuesta. Su contenido será detallado en posteriores secciones.

3 Preparación de datos

La preparación de los datos ha sido una de los elementos que más complicaciones ha ocasionado a la hora de realizar el presente trabajo. En concreto, para cada uno de los 4 países analizados (España, EEUU, UK y Japón), se deseaba obtener los dos siguientes tipos de datos, desde el año 2000 al 2016, con frecuencia mensual:

- Nivel de paro
- Valor bursátil de una bolsa importante del país

Dado que no fue posible encontrar un dataset público que tuviera tal conjunto de datos, hubo que realizar una búsqueda exhaustiva de los mismos en diferentes portales de Internet, y posteriormente se tuvo que realizar una labor de homogeneización de los datos. A continuación se mencionan las diferentes fuentes utilizadas para cada país:

País	Paro	Bolsa
España	[1]	IBEX 35 [5]
EEUU	[2]	DJI [6]
UK	[3]	LSE [7]
Japón	[4]	N255 [8]

Una vez descargados los datos, se normalizaron en cada caso los datos mensuales. Dado que para cada una de las versiones de visualizaciones preparadas, se van a utilizar una versión de datos ligeramente diferente (mismo contenido pero estructurados en diferentes ficheros), se procederá a describir en cada caso concreto el formato del conjunto de datos utilizado.

4 Visualización interactiva (versión 1)

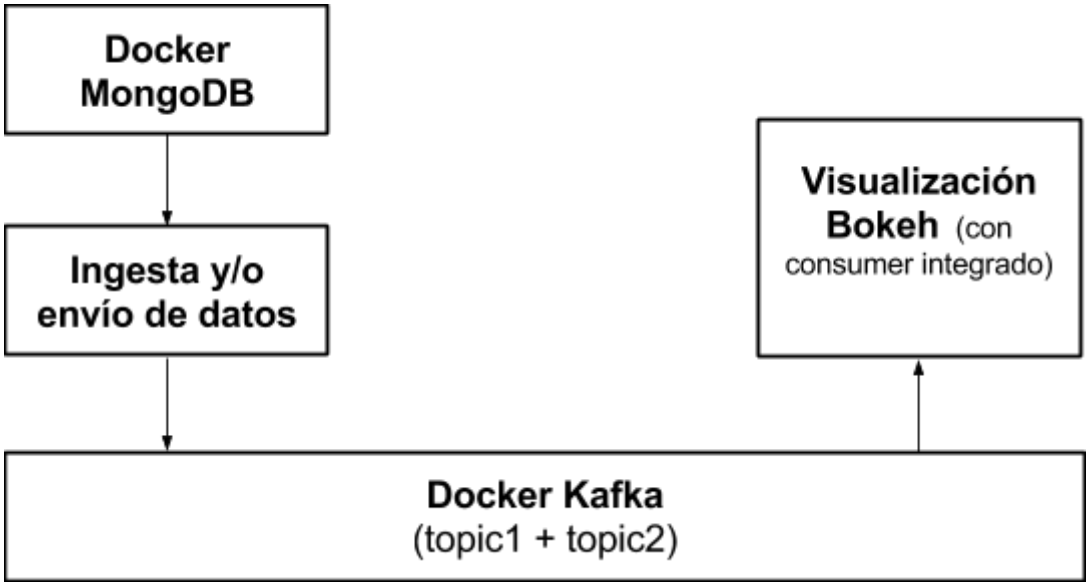
4.1 Archivos incluidos

Todos los archivos necesarios para la ejecución de la presente versión de visualización se encuentran incluidos en la carpeta “visualizacion_v1”. En concreto, podemos encontrar los siguientes:

- data
  - datos_bolsa
    1. *^IBEX.csv*
    2. *^DJI.csv*
    3. *^LSE.csv*
    4. *^SSE.csv*
    5. *^N255.csv*
    6. Processed
       - ✓ *csv_stockExchange*_mixed: fichero de datos de bolsa procesados
- datos_paro
  - london
    1. *series-040617.csv*: fichero con datos de paro de Londres
    2. *readme.txt*
  - japon
    1. *jma_data.csv*: fichero con datos de paro de Japón
    2. *readme.txt*
  - espana
    1. *4247.csv*: fichero con datos de paro de España
    2. *readme.txt*
  - eeuu
    1. *SeriesReport-20170604132227_e518f3.csv*: fichero con datos de paro de EEUU
    2. *readme.txt*

    - processed
        1. *csv_Unem_mixed*: fichero de datos de paro procesados
    - final_data
        1. *csv_stockExchange_mixed*: fichero con los datos de bolsa procesados
        2. *csv_Unem_mixed*: fichero con los datos de paro procesados
        3. *final_data.csv*: fichero con todos los datos procesados
- docker
    - MongoDB: docker de la base de datos mongoDB
    - Spotify_Kafka: docker kafka
- src
    - Ingest_and_sendDATA
        1. *DDBBingest_sendData.py*: script que además de insertar o eliminar ficheros de la base de datos, los envía para el streaming
    - KafkaConnection
        1. *kafka_connection.py*: script de clase para la conexión con Kafka
    - mongoDBclass
        1. *mongoDBclass.py*: script de clase para las operaciones con la base de datos
    - process_data
        1. *process_data.py*: script de clases para el preprocesado de los datos para conseguir un formato común
    - visualization
        1. *bokeh_visualization.py*: script de visualizacion en bokeh

4.2 Arquitectura



En primer lugar tenemos que tener en cuenta si tenemos los datos cargados en la base de datos o no. En caso de no tenerlos, usamos el paso de ingesta de datos para, además de realizar el envío de datos a Kafka, realizar la ingesta de los mismo en la base de datos.
Una vez introducidos en la base de datos, éstos se envían al Docker Kafka. Éste sirve de conexión con la visualización. Se enviarán los datos en dos topics, uno para los datos de desempleo y otro para los datos de valores bursátiles. En el script de visualización se

declaran dos consumer, uno para cada topic. Se junta la información proveniente de ambos y se muestra en las gráficas conforme van llegando. Se ha de recalcar que el intervalo de tiempo entre recepción de datos se define en la parte de ingesta de datos (parámetro “time to query”)

4.3 Requisitos

Para la ejecución de este módulo, habría que utilizar un sistema operativo Linux junto con Python 2.7, que ha de incluir las siguientes librerías:

1. bokeh
2. pykafka
3. pandas
4. pymongo

4.4 Instrucciones

La ejecución del presente módulo es muy sencilla. Simplemente habría que abrir una terminal en la localización del archivo DDBBingest_sendData.py y ejecutar el siguiente comando:

```
python DDBBingest_sendData.py ingest
```

Esto provocará, de una sola vez, la ingesta de datos al Docker MongoDB, configuración de un Docker Kafka, envío de mensajes a Kafka y posteriormente visualización interactiva de los mismos. Como resultado final, se deberá de abrir el navegador, produciéndose la visualización interactiva.

Este paso sólo se requiere en la primera ejecución, ya que necesitaremos que los datos procesados se inserten en la base de datos. Otros dos modos de ejecución son:

```
python DDBBingest_sendData.py remove
```

Elimina los datos y termina el programa.

```
python DDBBingest_sendData.py
```

Si los datos ya se encuentran insertados, realiza las mismas acciones que la ejecución “python DDBBingest_sendData.py ingest” pero sin la ingesta de datos en la base de datos.

En caso de encontrar cualquier error, se recomienda ejecutar la versión 2 de la visualización (siguiente sección), que es una versión simplificada y los requerimientos técnicos son más sencillos.

5 Visualización interactiva (versión 2)

La segunda versión de visualización es equivalente a la primera pero se ha simplificado. En concreto, se ha eliminado el uso de dockers y MongoDB.

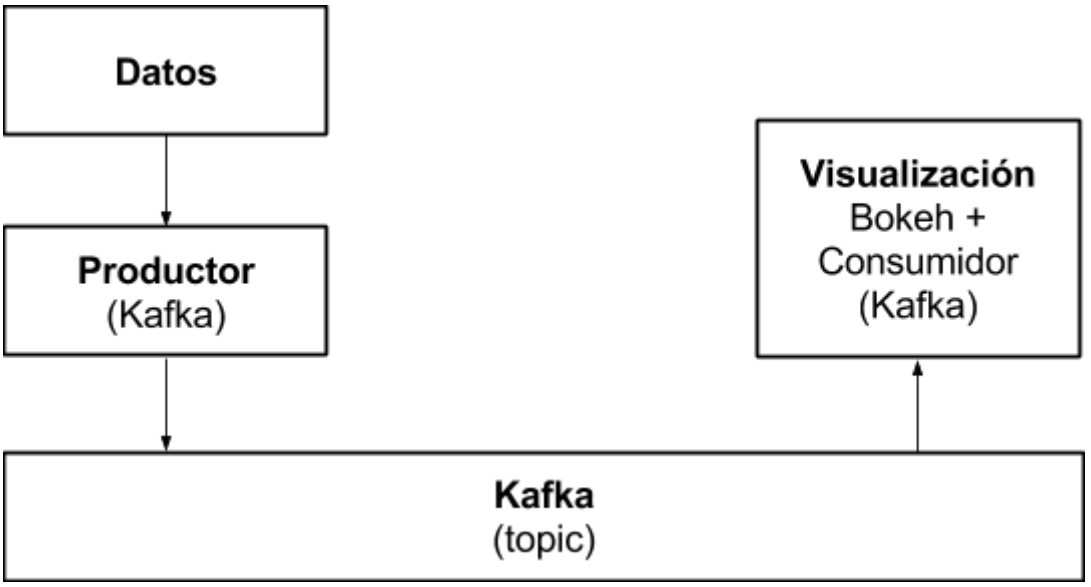
5.1 Archivos incluidos

Todos los archivos necesarios para la ejecución de la presente versión de visualización se encuentran incluidos en la carpeta “visualizacion_v2”. En concreto, podemos encontrar los siguientes:

- *final_data.csv*: archivo csv donde están almacenados todos los datos de todos los países, normalizados convenientemente . En concreto, tiene tantas filas como meses analizados, y 8 columnas cuyo significado, en orden, es el siguiente:
    - Fecha
    - Índice bursátil del DJI (EEUU)
    - Índice bursátil del LSE (UK)
    - Índice bursátil del IBEX 35 (España)
    - Índice bursátil de N225 (Japón)
    - Nivel de desempleo de EEUU
    - Nivel de desempleo de UK
    - Nivel de desempleo de España
    - Nivel de desempleo de Japón
- *bokeh_visualization.py*: script de Python con el que se realizará la visualización propiamente dicha
- *kafka_producer_p2.py*: script de Python utilizado para el envio de mensajes a Kafka
- *KafkaConnection*: carpeta que incluye un módulo para realizar la conexión entre Python y Kafka
- *instrucciones.txt*: archivo donde se incluyen los principales comandos necesarios para la ejecución de este módulo. Serán explicados en detalle en posteriores secciones.

5.2 Arquitectura

En el siguiente esquema se describe la arquitectura utilizada:



En concreto, en primer lugar se utiliza un productor de mensajes destinados a Kafka (*kafka_producer_p2.py*), que se basa en datos leídos desde un archivo csv (*final_data.csv*). Esos mensajes son dirigidos a Kafka, a un topic en particular. Posteriormente, dichos mensajes son leídos por el módulo de visualización (*bokeh_visualization.py*), que además realizará la visualización interactiva. A continuación, se describen, por un lado, los requisitos necesarios para la ejecución del código incluido, y por otro lado, todos y cada uno de los pasos a seguir.

5.3 Requisitos

Es necesario que nuestra entorno de trabajo de Python (versión 2.7) tenga instaladas las siguientes librerías:

5. bokeh
6. pykafka

También será necesario tener descargado Kafka en nuestro ordenador. Además, la localización de la carpeta principal de dicho software deberá de estar guarda como la siguiente variable de entorno: *KAFKA_HOME.* En cuanto al sistema operativo, en principio la ejecución del presente programa está pensada para Linux (Ubuntu 16.04), si bien es cierto que en principio también se debería de poder ejecutar sin problemas en Mac y Windows.

5.4 Instrucciones

En primer lugar, habrá que configurar e inicial Kafka. Para ello, se han de ejecutar los dos siguientes comandos (en terminal):

1. $KAFKA_HOME/bin/zookeeper-server-start.sh $KAFKA_HOME/config/zookeeper.properties

2. $KAFKA_HOME/bin/kafka-server-start.sh $KAFKA_HOME/config/server.properties

Una vez que que Kafka está configurado, crearemos un nuevo *topic* que utilizaremos en la visualización.

3. $KAFKA_HOME/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic <topic_name>

Para comprobar que dicho topic ha sido creado de manera satisfactoria, ejecutaremos el siguiente comando:

4. $KAFKA_HOME/bin/kafka-topics.sh --list --zookeeper localhost:2181

Una vez que Kafka está configurado, incluido el topic, podemos empezar a lanzar mensajes a dicho topic. Para ello, utilizaremos el script de Python *kafka_producer_p2.py,* el cual se ejecuta de la siguiente manera:

python kafka_producer_p2.py <campo1> <campo2> <campo3> <campo4> <campo5>

Donde el <campo1> es el nombre del topic donde queremos lanzar (<topic_name>), <campo2> es del archivo desde donde queremos leer los datos (final_data.csv), <campo3> es el tiempo de espera entre cada mensaje (lo que nos permitirá controlar la velocidad de generación de mensajes, en segundos), <campo4> es la fecha de inicio y <campo5> la fecha de fin de los mensajes que queremos lanzar (formado YYYY-MM)

Para comprobar que todo funcional correctamente, lanzamos un primer mensaje que contiene tan sólo los datos del primer mes:

5. python kafka_producer_p2.py <topic_name> final_data.csv 0.5 1999-12 2000-02

El siguiente paso sería ejecutar el módulo de visualización *bokeh_visualization.py*. **En dicho archivo, habría que modificar la variable *topic_name*, línea 27, e igualarla al valor del <topic_name> creado**. Posteriormente, habría que ejecutar el siguiente comando.

6. bokeh serve --show bokeh_visualization.py

Si todo es correcto, se debería de abrir en el navegador una nueva pestaña con 6 gráficas vacias. Como último paso, habría que lanzar más mensajes al topic de Kafka de tal manera que fueran recogidos por el módulo de y posteriormente graficados. Dado que tenemos la oportunidad de controlar cuántos mensajes son enviados y a qué velocidad, proponemos hacerlo en 4 pasos:

7. python kafka_producer_p2.py <topic_name> final_data.csv 0.25 2000-02 2007-01
8. python kafka_producer_p2.py <topic_name> final_data.csv 0.5 2007-01 2009-01
9. python kafka_producer_p2.py <topic_name> final_data.csv 0.5 2009-01 2013-01
10. python kafka_producer_p2.py <topic_name> final_data.csv 0.5 2013-01 2016-12

Estos cuatro pasos (7, 8, 9, 10) corresponden a 4 etapas diferenciadas del ciclo económico analizado. En la primera de ellas, se puede observar una gran bonanza económica (paro pequeño y bolsas en niveles altos en todos los países). En la segunda etapa, explota la crisis de manera muy notable (el paro sube y las bolsas bajan de manera dramática). En la tercera etapa, EEUU y Japón empiezan a recuperarse. Por otro lado, UK sufre cierto estancamiento. Sin embargo, en España continúan los problemas, ya que el paro sigue subiendo con fuerza y la bolsa no termina de recuperarse. Finalmente, en la 4 etapa, los 4 países sufren una mejora de la economía, si bien es cierto que España es el único de ellos que no logra alcanzar los niveles del año 2000. En comparación con la versión 1 de la visualización, en este caso, nos basamos en que todos los datos provienen de un mismo productor. En aquel caso, datos de diferentes fuentes son volcados en el mismo topic, ganando flexibilidad y grado de generalización. Sin embargo, se ha preparado este segundo ejemplo dado que los requerimientos del sistema a para su ejecución son menos complicados (no se usa ningún docker ni MongoDB), y además nos permite mandar los mensajes por etapas, controlando su velocidad.

6 Principales retos superados

A continuación, se enumeran diversos retos que han surgido a la hora de realizar la presente práctica:

1. Descarga de datos de diferentes fuentes y su normalización
2. Uso de dockers para la universalización de la arquitectura
3. Conexión entre Bokeh y Kafka.
4. Visualización de datos temporales en Bokeh (eje x de carácter temporal)
5. Doble eje y en Bokeh

7 Referencias

[1] http://www.ine.es/jaxiT3/Tabla.htm?t=4247

[2] https://data.bls.gov/timeseries/LNS14000000

[3] https://www.ons.gov.uk/employmentandlabourmarket/peoplenotinwork/unemployment/timeseries/mgsx/lms

[4] https://www.japanmacroadvisors.com/page/category/economic-indicators/labor-markets/unemployment-rate/

[5] https://finance.yahoo.com/quote/%5EIBEX/history?p=%5EIBEX

[6] https://finance.yahoo.com/quote/%5EDJI/history?p=%5EDJI

[7] https://finance.yahoo.com/quote/LSE.L/history?period1=978303600&period2=1483138800&interval=1d&filter=history&frequency=1d

[8] https://finance.yahoo.com/quote/%5EN225/history?period1=946681200&period2=1483138800&interval=1mo&filter=history&frequency=1mo