

Ciclo de Vida Analítico del Dato
Persistencia, Buscadores y Arquitectura
Daniel Pérez Efremova

Índice

1. Instrumentos y métodos..... 4

 1.1 Conjunto de datos..... 4

 1.2 Hipótesis y métodos..... 4

 1.2 Hipótesis y métodos..... 4

 1.3 Mapa de aplicaciones..... 5

2. Resultados..... 6

3. Conclusiones..... 8

4. Visión general y lecciones aprendidas..... 8

INTRODUCCIÓN

En esta memoria se recoge el trabajo realizado con buscadores, almacenamiento y procesamiento persistente, y arquitectura específica de Big Data. Aunque no se utiliza un gran volumen de datos, el diseño permite escalar la aplicación a grandes dimensiones sin mucho esfuerzo, especialmente en cuanto a número de columnas.

Hasta hoy, se han conducido análisis sobre el mercado de valores con gran variedad de técnicas matemáticas, desde las más clásicas basadas en reglas de decisión o regresión lineal, hasta las más avanzadas, con redes neuronales o sistemas caóticos. Todos estos modelos tienen en común que incorporan alguna medida de sentimiento del mercado para modelizar la volatilidad de un activo financiero en un momento concreto.

El objetivo es comprobar la hipótesis de que las variables que se proponen en [1] como índices de sentimiento del mercado de valores, guardan relación con los precios de un activo famoso por su volatilidad: Bitcoin.

Para comprobar la hipótesis se ha desarrollado una aplicación que se conecta al repositorio de datos de Yahoo Finance, calcula los índices de sentimiento y la matriz de correlaciones, y devuelve tanto los datos recolectados como la matriz de correlaciones.

Las aplicaciones involucradas son:

- [Elasticsearch y Kibana](#) como tecnologías de descubrimiento
- [Pyspark](#) como herramienta de consulta y cómputo eficiente¹
- [HDFS](#) como sistema de almacenamiento
- [Parquet](#) como formato de fichero persistente orientado a columnas
- [Json](#) como formato de integración entre la aplicación y buscadores

El desarrollo se ha realizado de manera local, en un SO Ubuntu 20.04, con Procesador Intel(R) Core(TM) i5-8265U CPU con 1.60GHz y 8 GB de RAM.

Palabras Clave: Pyspark, Buscadores, Persistencia, Análisis de Datos, Mercado Financiero

¹ En la práctica se propone Hive como herramienta de consulta. Sin embargo, se ha decidido usar Pyspark porque la API de Hive (en HQL) no ofrece buen soporte para los cálculos requeridos.

1. Instrumentos y métodos

En esta sección se detallan los componentes de la aplicación y cómo se relacionan con los objetivos del trabajo. Primero, se presenta la estructura de los datos que se pueden obtener con la aplicación. Después, se detallan las hipótesis y las técnicas matemáticas para comprobarla. Finalmente, se da una visión general del mapa de aplicaciones que permite llevar a cabo el experimento.

1.1 Conjunto de datos

El conjunto de datos se extrae de manera automática mediante el script *prep.py* de python. Se encarga de llamar a la API pública de *Yahoo Finance* y extraer, para el índice de Bitcoin (BTC)², una tabla con las siguientes columnas:

- **Close.** Precio al cierre del mercado
- **Open.** Precio a la apertura del mercado
- **High.** Precio más alto durante la jornada
- **Low.** Precio más bajo durante la jornada
- **Volumne.** Número de transacciones
- **Adj Close.** Precio ajustado por efectos fuera de la jornada: partición de acciones, reparto de dividendos, comparaciones temporales, etc.
- **Date.** Fecha de la observación. En este caso se observa diariamente de manera agregada.

En este trabajo, por sencillez, se toma como referencia del precio la variable Close.

1.2 Hipótesis y métodos

En [1] se proponen, entre otros, dos índices como medidas de sentimiento del mercado. Llamando t a un instante de tiempo cualquiera y V al valor de un índice (en este caso se toma Close), se definen:

1. **PSY (Psychological Index Terms)** es la proporción de instantes de tiempo que el activo se ha devaluado respecto a una ventana de tiempo w (en este caso de $w=14$ días). Es un índice adimensional y formalmente se define:

$$PSY_t^w = \frac{\sum_{i=0}^{w-1} I\{V_{t-i-1} > V_{t-i}\}}{w}$$

Este índice mide la presión psicológica a la que están sometidos los inversores al ver su activo devaluado. Cuántos más periodos de la ventana permanezca devaluado, mayor es la presión que siente el inversor. Esto, debería traducirse en mayor volumen de transacciones para salvar la posición de riesgo o aprovechar precios bajistas y por tanto, debería estar correlacionado con Close y Volume.

2. **RSI (Relative Strength Index)** mide la magnitud de cambios recientes en el precio de un activo para evaluar si está depreciado o apreciado. Es un índice adimensional y formalmente se define:

$$RSI_t^w = 100 \times \frac{\bar{L}_t^w}{\bar{L}_t^w + \bar{G}_t^w}$$

donde

² En este caso solo se usa Bitcoin por sencillez, pero puede hacerse para cualquier conjunto de índices del mercado que esté en su catálogo.

$$\bar{L}_t^w = \frac{1}{w} \sum_{i=0}^{w-1} I\{V_{t-i-1} > V_{t-i}\} \left| \frac{V_{t-i} - V_{t-i-1}}{V_{t-i-1}} \right|$$

$$\bar{G}_t^w = \frac{1}{w} \sum_{i=0}^{w-1} I\{V_{t-i-1} < V_{t-i}\} \frac{V_{t-i} - V_{t-i-1}}{V_{t-i-1}}$$

son los promedios de pérdidas y ganancias. Se toma el valor absoluto de las pérdidas para que sean comparables a las ganancias. Si el índice es alto quiere decir que el activo está muy demandado y que por tanto puede estar sobrevalorado mientras que un valor bajo indica que está poco demandado, es decir, infravalorado. Por tanto, debería estar correlacionado con Volumen y Cierre.

Para comparar los efectos que pueden tener estos índices sobre el mercado, **se calcula el coeficiente de correlación de Pearson** entre éstos y distintas variables que captan la volatilidad del mercado desde el punto de vista de transacciones y de precio:

1. **PriceVar.** Es la variación del precio con respecto al día anterior

$$\frac{V_{t-i} - V_{t-i-1}}{V_{t-i-1}}$$

2. **VolumeVar.** Es la variación del volumen de transacciones con respecto al día anterior

$$\frac{N_{t-i} - N_{t-i-1}}{N_{t-i-1}}$$

3. **Volatility.** Es la desviación típica de Cierre en la ventana de tiempo (14 días)

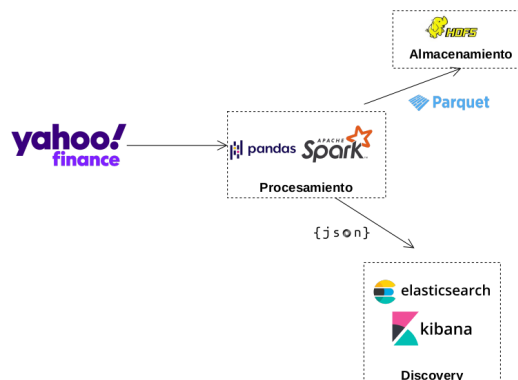
Los coeficientes se expresan en forma matricial (matriz de correlaciones).

1.3 Mapa de aplicaciones

La aplicación consta de distintos módulos. En primer lugar, **la fuente de datos es la API de Yahoo Finance**.

Para hacer las llamadas al conjunto de datos se utiliza un programa escrito en Python (prep.py) que recibe y prepara los datos. En este proceso, también se calculan los índices y variables descritos anteriormente. Una vez terminados los cálculos, se almacenan los datos en formato parquet en un sistema que esté preparado con HDFS (en este caso en local). Además, se produce una salida adicional del conjunto de datos recibido en formato .json para facilitar la integración con el buscador Elasticsearch. Este fichero .json se carga en Kibana para realizar las tareas de descubrimiento en los datos.

El sistema HDFS se escoge para dar persistencia a la arquitectura y el formato parquet para aprovechar la eficiencia del formato orientado a columnas, dado que las consultas a este conjunto de datos se hacen buscando solo unas pocas columnas. Además, en caso de añadir más activos, lo que aumentaría serían las columnas, no las filas (son fijas, un registro por día), por lo que es adecuado orientar el almacenamiento a columnas.



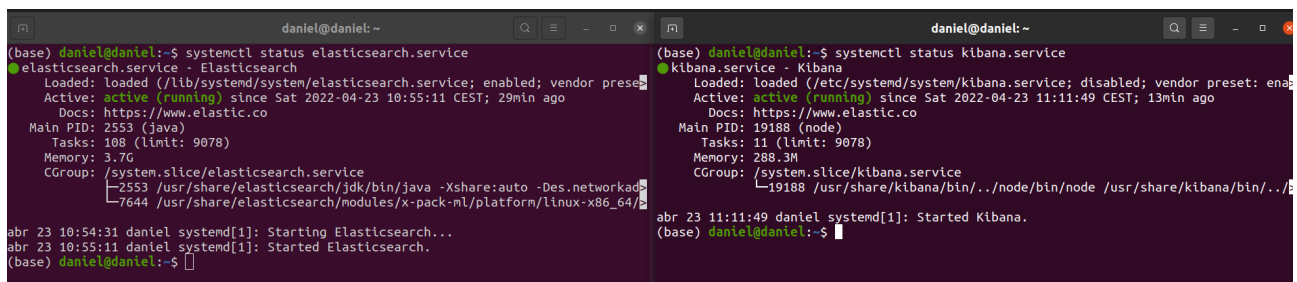
El fichero .json se carga manualmente mediante la interfaz de Kibana.

Tras ejecutar la preparación, se realiza el análisis. Esta parte se realiza con un script de Python (proc.py) que levanta una aplicación de Spark y calcula la matriz de correlaciones. Al terminar, se muestra por consola y la almacena en el sistema HDFS en formato parquet.

2. Resultados

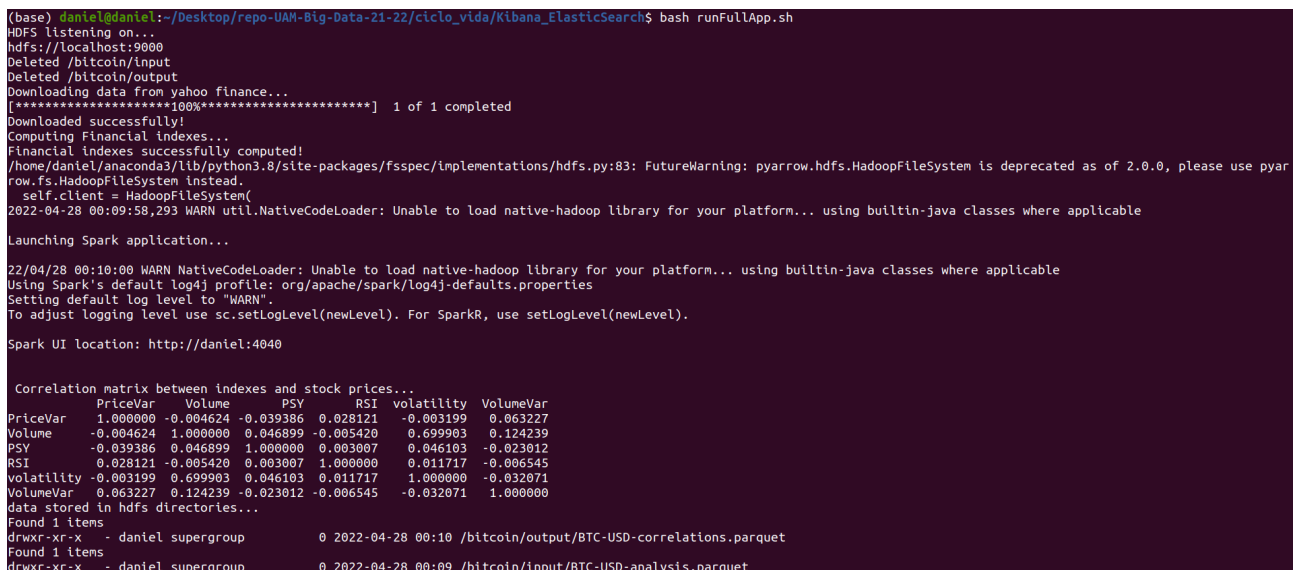
En esta sección se describen los resultados obtenidos en los distintos módulos de la aplicación. Para obtenerlos, basta ejecutar el script runFullApp.sh.

En primer lugar, tras instalar los servicios y levantarlos, se debería ver por consola algo similar a la siguiente imagen.



The image shows two terminal windows. The left window displays the command `systemctl status elasticsearch.service` and its output, indicating that Elasticsearch is active and running. The right window displays the command `systemctl status kibana.service` and its output, indicating that Kibana is also active and running.

Después, al ejecutar el script runFullApp.sh se debería ver por consola algo similar a la siguiente imagen.



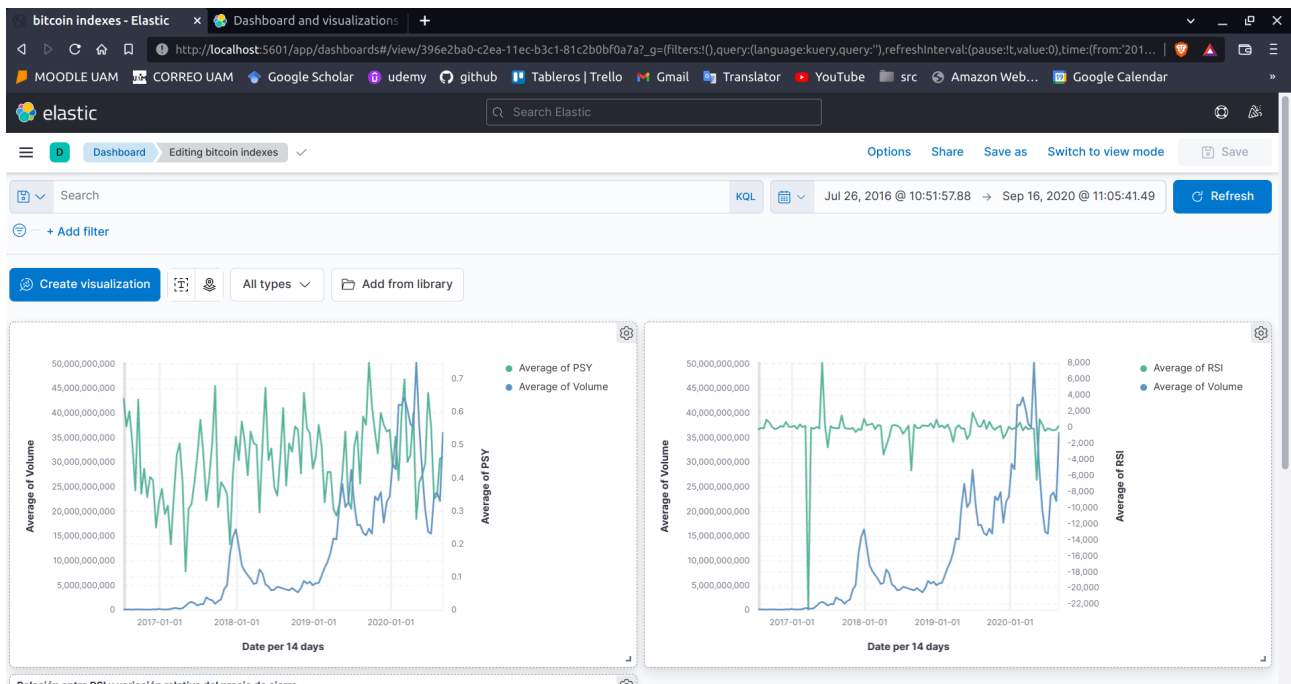
The image shows a terminal window with the output of the `runFullApp.sh` script. The script performs several tasks: it sets up HDFS, downloads data from Yahoo Finance, computes financial indexes, launches a Spark application, and calculates a correlation matrix between stock prices and various indicators. The output includes a detailed correlation matrix and information about the data stored in HDFS.

```
Correlation matrix between indexes and stock prices...
PriceVar   Volume   PSY      RSI      volatility   VolumeVar
PriceVar   1.000000  -0.004624 -0.039386  0.028121    -0.003199  0.063227
Volume     -0.004624  1.000000  0.046899  -0.005420    0.699903  0.124239
PSY        -0.039386  0.046899  1.000000  0.003007    0.046103  -0.023012
RSI         0.028121 -0.005420  0.003007  1.000000    0.011717  -0.006545
volatility  -0.003199  0.699903  0.046103  0.011717    1.000000  -0.032071
VolumeVar   0.063227  0.124239 -0.023012 -0.006545   -0.032071  1.000000

data stored in hdfs directories...
Found 1 items
drwxr-xr-x  - daniel supergroup          0 2022-04-28 00:10 /bitcoin/output/BTC-USD-correlations.parquet
Found 1 items
drwxr-xr-x  - daniel supergroup          0 2022-04-28 00:09 /bitcoin/input/BTC-USD-analysis.parquet
```

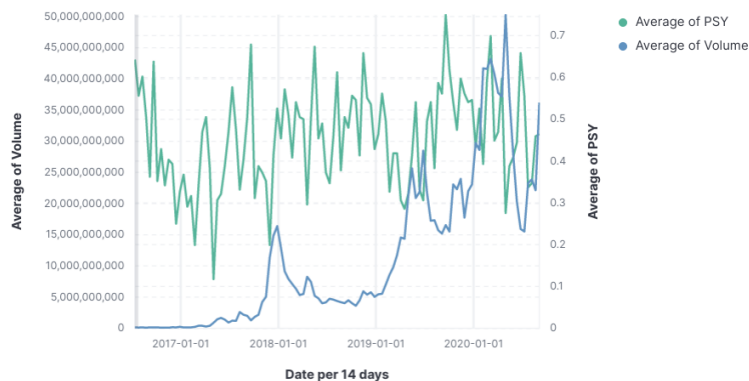
Primero, se informa del puerto donde está escuchando el sistema HDFS y se hace una limpieza de directorios (se trata de una versión de testing, de aquí que se elimine todo en cada nueva ejecución). Después, se descargan los datos, se calculan los índices y variables, y se estandarizan normalmente las variables. Finalmente, se levanta la aplicación Spark y se calcula la matriz de correlaciones. Además, dentro del directorio donde se ejecuta la aplicación, se debería crear el directorio data/elasticData que tenga el fichero .json a cargar en Kibana.

Tras cargar el fichero en Kibana, se construyen varias visualizaciones para series temporales como se ve en la siguiente imagen.



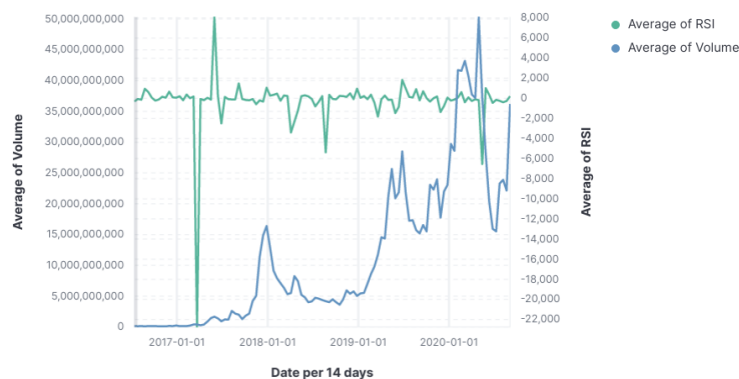
Más en detalle, se construyen tres visualizaciones.

La primera, muestra la serie de PSY y la serie Volume. Si la hipótesis de este trabajo es cierta, se debería observar cierta relación directa o indirecta entre las series.



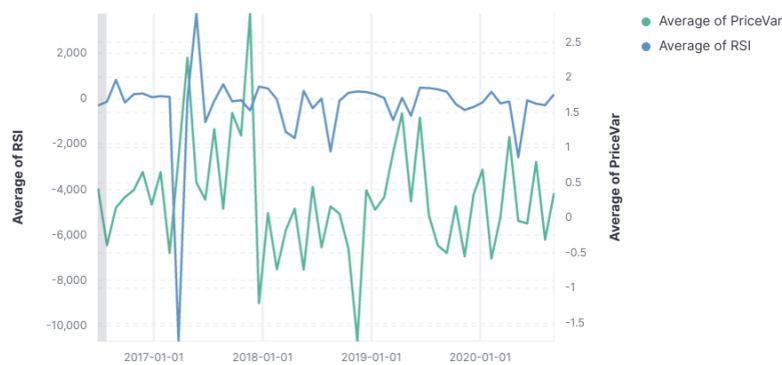
En el gráfico parece que los valles de la serie PSY vienen seguidos de un pico relativo de Volume y también con los picos. Así que es razonable comprobar la correlación indirecta entre los valores.

La segunda, muestra la serie de RSI y la serie Volume. Si la hipótesis de este trabajo es cierta, se debería observar cierta relación directa o indirecta entre las series.



En el gráfico se observa que los valles de Volume vienen precedidos de un pico en RSI, y también con los picos. Así que es razonable comprobar la correlación indirecta entre los valores.

La tercera, muestra la serie de RSI y la serie PriceVar. Si la hipótesis de este trabajo es cierta, se debería observar cierta relación directa o indirecta entre las series.



En el gráfico se observa que los valles de RSI vienen precedidos de un valle en PriceVar, y también con los picos. Así que es razonable comprobar la correlación indirecta entre los valores.

La matriz de correlaciones que se obtiene es la siguiente.

	PSY	RSI	PriceVar	Volume	VolumeVar	volatility
PSY	1.000	0.003	-0.040	0.048	-0.023	0.046
RSI	0.003	1.000	0.028	-0.005	-0.007	0.012
PriceVar	-0.040	0.028	1.000	-0.005	0.064	-0.003
Volume	0.048	-0.005	-0.005	1.000	0.124	0.700
VolumeVar	-0.023	-0.007	0.064	0.124	1.000	-0.032
volatility	0.046	0.012	-0.003	0.700	-0.032	1.000

3. Conclusiones

Se observa que ninguno de los índices (RSI y PSY) correlaciona significativamente con las variables escogidas para describir la volatilidad del mercado (PriceVar, VolumeVar, Volatility y Volume). Por otro lado, la única correlación significativa es Volume con Volatility, algo esperable ya que las variaciones en el precio siempre están relacionadas con un alto volumen de transacciones.

Se concluye que, con la metodología propuesta, estos índices no guardan relación ni con el comportamiento diario de los precios ni el volumen de transacciones. Se hace necesario un análisis más profundo para poder encontrar alguna relación, ya que los métodos propuestos son muy básicos.

4. Visión general y lecciones aprendidas

En esta práctica se ha visto cómo levantar un servicio de buscador con visualización. También, cómo sacar provecho de los formatos de persistencia y sistemas de almacenamiento distribuido. Además, se ha visto cómo integrar distintos servicios para aprovechar las bondades de cada herramienta y no limitarse a una concreta.

Se ha tratado de utilizar todas las herramientas propuestas de manera sencilla en un caso práctico con un mínimo de realismo, aunque no se ha conseguido encajar todas las herramientas en una única aplicación. Por ejemplo, se ha decidido usar Spark en lugar de Hive (Hadoop) para las consultas por ser una herramienta muy limitada técnicamente. Tampoco se ha utilizado Hbase al tratarse de datos relacionales y cuyo número de columnas puede escalar enormemente.

al añadir muchos productos financieros, algo que complica la gestión de los schemas de Hbase. Queda pendiente también cómo integrar de manera automática la ingesta de datos al buscador ya que en esta aplicación el .json se debe provisionar a mano.

En líneas generales se ha podido poner en práctica muchas de las herramientas y técnicas aprendidas durante el curso, tanto en la vertiente de análisis como de infraestructura y tecnologías para Big Data.

REFERENCIAS

[1] Xu, Qifa, Liukai Wang, Cuixia Jiang, and Yezheng Liu. "A novel (U) MIDAS-SVR model with multi-source market sentiment for forecasting stock returns." *Neural Computing and Applications* 32, no. 10 (2020): 5875-5888.