data product architecture

maestría en ciencia de datos

mayo 2016, cdmx

social data discovery

proyecto final

- ricardo lastra cuevas
- laura vargas hernández
- adrián vázquez páez
- ana paula alonzo fuentes

objetivo

- diseñar un producto de datos para la obtención de información en streaming, desde la plataforma twitter,
- clasificar los tweets de los usuarios a partir de sentimientos asociados a un tema específico,
- aplicar analítica sobre los datos procesados,
- presentar resultados del análisis y su interpretación.

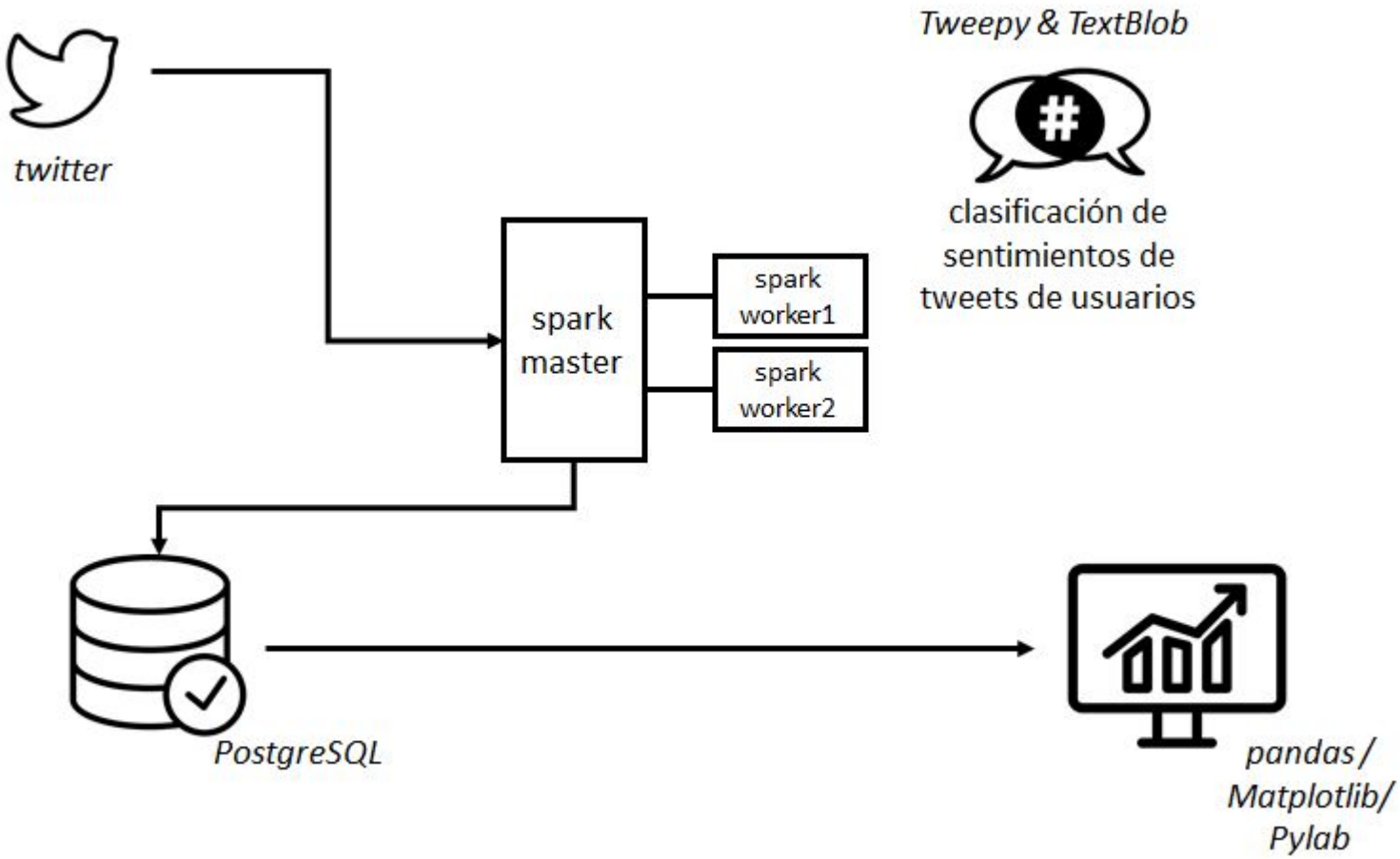
antecedentes

- El caso práctico está centrado en la obtención de Tweets en streaming para el análisis de sentimientos,
  - usando referencias de #HashTags de un Trending Topic - TT - sobre eventos específicos.

Tema seleccionado	• Terrorismo, aviones y situaciones alarmantes mundiales
TT analizado	• **#EgyptAir**, cuyo flujo en la red, se asume vigente a la fecha de entrega del presente proyecto
Evento detonante	• Avión de EgyptAir que viajaba de París a El Cairo con 66 pasajeros, desaparece del radar en pleno viaje, la noche del 19 de mayo

desarrollo técnico

pipeline


twitter
Tweepy & TextBlob
clasificación de sentimientos de tweets de usuarios
spark master
spark worker1
spark worker2
PostgreSQL
pandas / Matplotlib/ Pylab

producto

1. Docker
2. Docker Compose
3. Postgres
4. Datalake

ingesta

- La ingesta óptima depende de factores como:
  - Velocidad de los discos
  - Tipo del archivo de entrada
  - Base de datos relacional
  - Streaming
- Se incorporaron herramientas de análisis Open Source.

aplicación

Se desarrolló el WorkFlow para hacer el llamado de los datos referentes a **#EgyptAir**.

Se obtuvieron los datos con herramientas Open Source, se transformaron a HDFS y guardaron para su llamado posterior.

Si bien hoy las herramientas más poderosas para recolectar, agregar y mover grandes cantidades de datos, desde diferentes fuentes a un data store centralizado, son Apache Flume y Apache Kafka, la naturaleza del proyecto permitió aplicar otras alternativas.

- Tales como:
  - APIs de twitter + Python
- Se buscó que la forma de recolectar, agregar y mover los datos se hiciera de manera óptima.

aplicación...

- Una API -Application Programming Interface-, es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación OO),
    - que permite a través de una biblioteca, ser utilizada por otro software como capa de abstracción.

Se utilizaron las siguientes APIs:

1. tweepy.Stream
2. tweepy.StreamListener
3. tweepy.OAuthHandler

aplicación...

Tweepy.Stream descarga los mensajes de Twitter en tiempo real; genera un alto volumen de tweets, o crea una transmisión en vivo con una corriente de sitio o corriente de usuario.

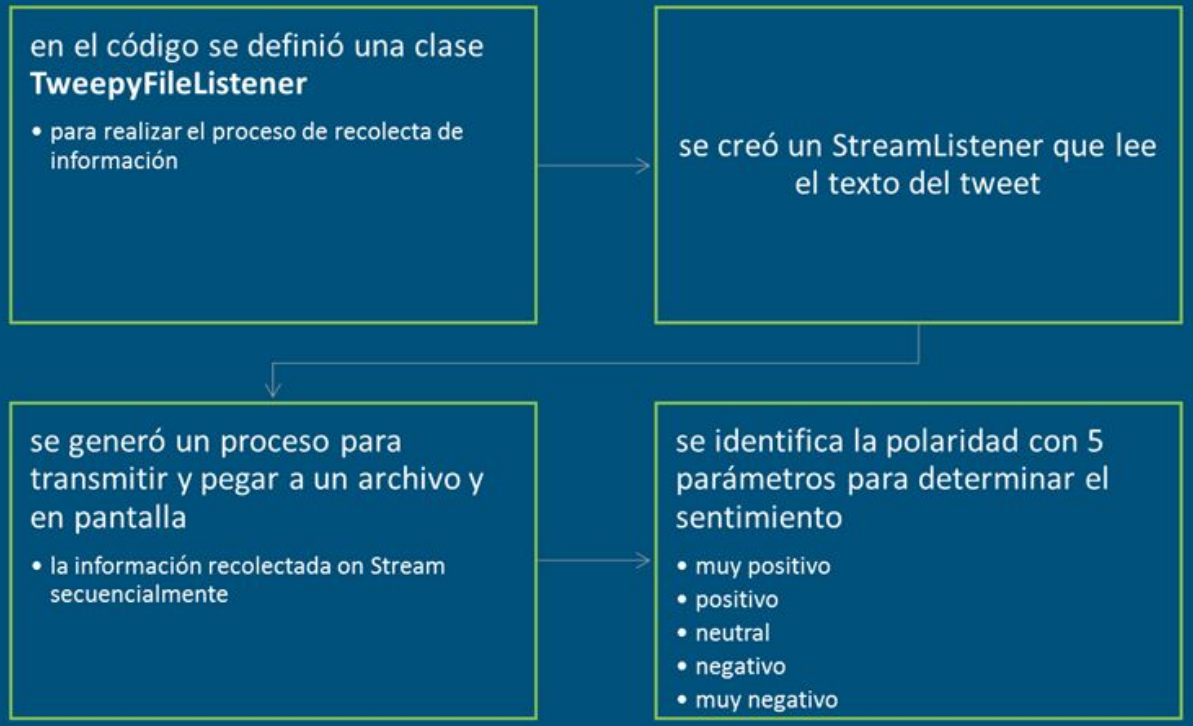
Es muy diferente de la API REST que se utiliza para tirar datos de Twitter, puesto que Tweepy. Stream empuja mensajes a una sesión persistente, permitiendo descargar más datos en tiempo real.

Una instancia de Tweepy.Stream establece una sesión de transmisión y enruta los mensajes a StreamListener. El método *on_data* de un oyente corriente, recibe todos los mensajes y llama a funciones según el tipo de mensaje.

StreamListener lee y clasifica el contenido de los mensajes que son generados por los usuarios.

Para el análisis del texto del tweet se usó la librería de Python TextBlob.

aplicación...


en el código se definió una clase **TweepyFileListener**
• para realizar el proceso de recolecta de información
se creó un StreamListener que lee el texto del tweet
se generó un proceso para transmitir y pegar a un archivo y en pantalla
• la información recolectada on Stream secuencialmente
se identifica la polaridad con 5 parámetros para determinar el sentimiento
• muy positivo
• positivo
• neutral
• negativo
• muy negativo

aplicación...

variables	'userScreen'	nombre de la pantalla o alias con el que el usuario se identifica. Los screen_names son únicos, normalmente de 15 caracteres máximo, aunque algunos relatos históricos pueden tener nombres más largos.
	'Sentiment'	descripción según la polaridad del sentimiento
	'tweetText'	String de texto generado por el usuario, conocido como tweet
	'tweetCreated'	fecha y hora UTC de creación del tweet en twitter
	'userCreateDt'	fecha y hora UTC de creación el tweet en la cuenta del usuario
	'userLocation'	ubicación definida por el usuario en el perfil de su cuenta. No necesariamente es una ubicación no analizable. En ocasiones es interpretado por el servicio de búsqueda
	'userTimezone'	cadena que describe la zona horaria del usuario que se declara a si mismo en el interior

herramientas de análisis

1. Json
2. PySpark / Spark SQL
3. TextBlob

Json

Formato para bases de datos.

- Los datos extraídos vía streaming de Twitter llegan en formato Json,
  - se guardan en el archivo EgyptAirJsons.txt.

PySpark / Spark Sql

API de Spark que permite hacer uso del modelo de programación de Spark en Python. El contexto de SQL posibilita utilizar código SQL en DataFrames de Spark.

- La información de los tweets es almacenada a través de PySpark y Spark SQL
  - se lee la información y se almacena,
  - a través de SQLContext se registra como una tabla temporal en PySpark,
  - se limpia con algunas ejecuciones de querys de SQL
  - y se transforma en formato Pandas para posteriormente realizar análisis de los datos y visualizar los resultados.

TextBlob

Librería de Python para procesar y analizar textos. Cuenta con una API para realizar tareas de procesamiento de lenguaje natural, en especial permite realizar análisis de sentimientos.

- Se crea un objeto TextBlob con el atributo *sentiment* que consiste en una tupla:
  - (polarity, subjectivity).

- **polarity**: número entre -1 y 1 que representa la polaridad de sentimiento del texto:
  - < 0 negativo
  - = 0 neutral
  - > 0 positivo
- **subjectitity**: número entre 0 y 1 con referencia a qué tan subjetiva es la polaridad asignada al texto:
  - 0 muy objetiva
  - 1 muy subjetiva

TextBlob...

- Realiza el análisis de texto basándose en los adjetivos presentes y en una base de datos propia de la librería,
  - cada uno con una puntuación de polaridad;
  - cada palabra también tiene puntuaciones de subjetividad e intensidad.
- La intensidad es una forma de modificar las puntuaciones relativas a la polaridad y subjetividad de una palabra,
  - de este modo TextBlob toma en cuenta las relaciones entre los distintos adjetivos y sus modificadores dentro del texto a analizar.
- La polaridad del sentimiento de una frase, se obtiene calculando el promedio de las puntuaciones asociadas a cada palabra del texto.

visualización

- Pandas
  - librería de Python que permite la visualización, manipulación y análisis de datos
- Matplotlib/Pylab
  - librería para la generación de gráficos a partir de datos en listas o arrays en Python y su extensión matemática NumPy. Proporciona una API pylab

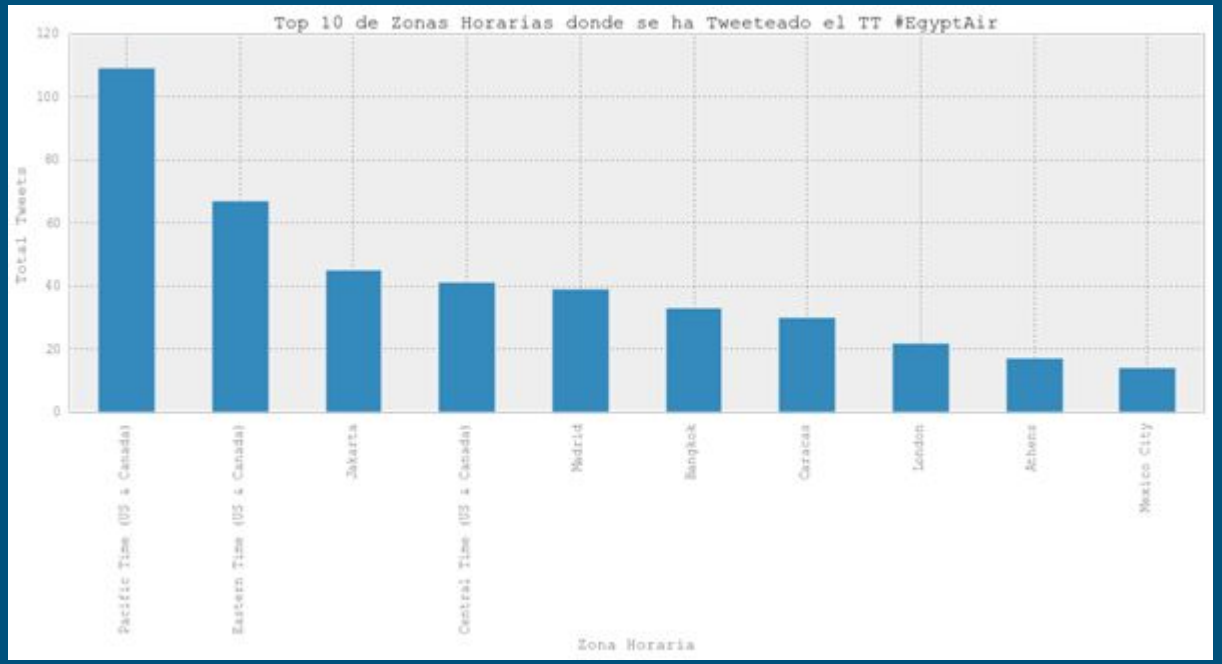
algunos hallazgos

ejemplos...

zona horaria

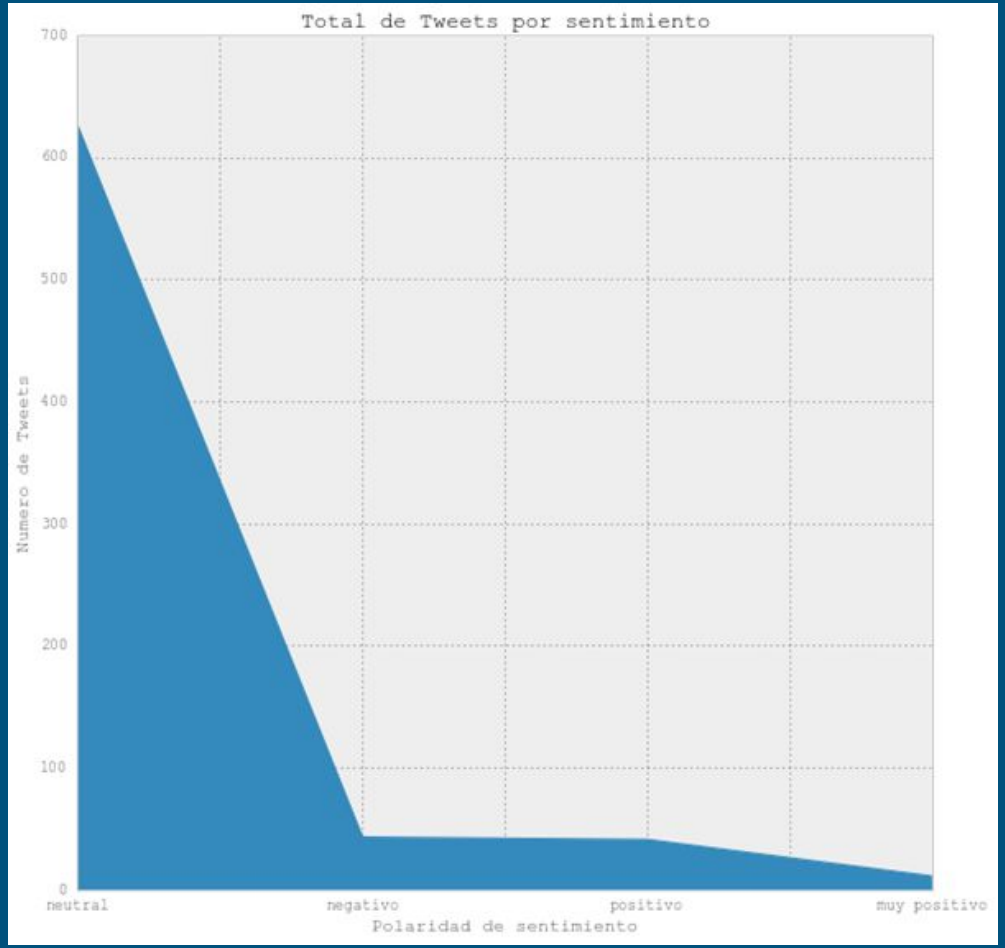
Ranking de tweets acerca del TT #EgyptAir según la zona horaria.

- zona norte de America (Pacific Time) con mayor presencia de tweets relacionados.
- la misma zona muestra interés sobre nota del avión encontrado recientemente cerca de las costas de Egipto, en Alejandría.
    - zona en américa muestra interés en el medio oriente.



sentimiento

- gran número de tweets con sentimientos neutrales
    - corresponden a notas relacionadas al tema y no a comentarios de la gente;
- los comentarios positivos se asocian a los avances en la búsqueda;
- en las últimas horas la distribución por positivo, negativo y neutral, se ha mantenido;
- las notas generan polémica y provocan movimientos en los sentimientos de la gente.
- desde su inicio, el TT se ha mostrado orientado a la polaridad de los sentimientos de neutral a negativo.



principales dificultades durante el diseño del producto

- Se intentó realizar el stream de los datos con Flume, pero no se logró solventar el error arrojado por falta de una librería
  - com.cloudera.flume.source.TwitterSource
- El probrema se solventó remplanzando el streaming con la API de Tweepy en conjunto con Python.

perspectiva

El producto diseñado ofrece la posibilidad de ingresar cualquier #HashTag sobre el que se decida hacer análisis, realizando la extracción en streaming de los tweets asociados e incorporándolos a todo el proceso hasta la visualización de los resultados.

Las aplicaciones pueden variar entre enfoques sociales, políticos, comerciales y de mercado, económicos, etc., según se requiera.

Si bien es mejorable, puede valorarse su funcionalidad en el actual punto de su desarrollo.

referencias

http://docs.tweepy.org/en/v3.5.0/api.html

https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

https://dev.twitter.com/overview/api/users

http://www.horamundial.com/husos.php

http://www.tweepy.org/

http://www.elfinanciero.com.mx/mundo/avion-de-egyptair-desaparece-en-el-mediterraneo.html