Memoria de la práctica Análisis de sentimientos con Hadoop

Sistemas Distribuidos de Procesamiento de Datos

Adrián Romero Hernández

Enrique Macip Belmonte

Maria Ruiz Teixidor

Índice

Descripción del código: incluyendo descripción de las principales funciones implementadas.

1. afinn_dictionary

Creamos un diccionario con las palabras y su valor.
Luego lo usaremos a la hora de cuantificar las palabras.

2. twitterstream

El objetivo de este fichero es obtener los datos de Twitter que queremos analizar. Para ello, primero definimos las variables de acceso. Después, a través de la URL "https://stream.twitter.com/1.1/statuses/sample.json" obtenemos los Words, sin seleccionar aquellos que han sido borrados “delete” o “status_withheld” y los almacenamos en un fichero Words.json.

3. tweet_map

Para crear el map, antes que nada cargamos el diccionario y lo llamamos ‘scores’. Para cada tweet del fichero .json se realizan los siguientes pasos:

- Filtramos por aquéllos que sí tengan el campo de ubicación (‘place’) informado y por los que en el campo de país ('country') indique United States o bien que el código del país sea US.

- Seleccionamos el campo ‘full_name’, dónde vienen la ciudad y el estado separados por una coma (ej. "full_name":"Manhattan, NY"). Los separamos.
  Presentan errores los siguientes estados: "full_name": "Michigan, USA" "Virginia, USA" ya que USA no es un estado.

- Si vemos que efectivamente constan dos términos asignamos a la variable location el estado.

- Finalmente creamos la clave-valor.

  Creamos una variable ‘score’ igual a cero. Separamos el tweet por palabras.

  - Si la palabra **aparece en el diccionario ‘scores’**: sumamos el valor que indica el diccionario para esa palabra en la variable score. Si no aparece, sumamos cero.

  - Si la palabra **empieza por #**: definimos como clave dicha palabra y como valor el estado. Si no empieza por #, definimos como clave el estado y como valor el score obtenido.

De esta manera creamos un diccionario **clave-valor** con dos tipos de clave-valor:

1. Clave: **estado** de Estados Unidos. Valor: **suma de las correspondientes puntuaciones de las palabras** que aparecen en el tweet.

2. Clave: hashtag. Valor: estado.

Ejemplo:
CT -5
KY 1
DE -1
VT 5
#Thanksgiving ID
#Tabú FL
#Portland NY

4. tweet_reduce

Con los datos obtenidos del map, creamos dos diccionarios vacíos (uno para los estados *dict* y otro para los hashtags *hashtag*) y separamos los clave-valor.

- Si la clave empieza con un # y no está en el diccionario, le asignamos valor 1, si ya está en el diccionario *hashtag* le sumamos 1. Los escribimos en formato clave-valor.

- Si no se trata de un hashtag:
  Contamos el número de veces que aparece la palabra, lo asignamos a la variable 'Total', y sumamos el score acumulado de las palabras del tweet en 'score'. Lo guardamos en el diccionario *dict como valores.*

Para el diccionario *dict,* para cada clave, indicamos que presente los siguientes valores: 'score', 'total', y 'media'. Siendo la media el score entre el total. Hemos aplicado el filtro de no contar palabras sin una puntuación asignada por calcular bien la media.

Ordenamos el diccionario *hashtag* y seleccionamos los 10 primeros clave-valor. Los escribimos en formato clave-valor.

De forma que obtenemos finalmente un diccionario con un único valor de cada estado con la suma de sus scores, el total de palabras con puntuaciones de sentimientos y la media del score, así como los 10 hashtags más usados con su número de menciones.

Los resultados obtenidos se pueden consultar en el anexo, mostramos las conclusiones más significativas. En total hemos analizado **3746** tweets.

La puntuación de las palabras del diccionario va desde -5 hasta 5, representando desde menos felices hasta más felices.

Estado con la media de Words más felices
El estado con Words más felices es Delaware (DE), con una media de 5 en sus Words.

Estado con la media de Words menos felices
Los estados con Words de media menos felices son Virginia Occidental (WV), Nebraska (NE), Rhode Island (RI) con una media de -3.

Top 10 trending topics obtenidos

#NEvsHOU	8
#JoeBurrow!	5
#90DayFiance	5
#Watchmen	5
#WeAreTexans	5
#pdx911	5
#NewProfilePic	4
#Texans	4
#Patriots	4
#SNFonNBC	3

Vemos que se comentó mucho el partido de los Patriots (New England) contra los Chieffs (Houston) de fútbol americano. En particular, también se mencionó mucho el jugador de fútbol americano Joe Burrow. Otros hashtags comentados se deben al tv reality de 90DayFiance o la serie Watchmen.

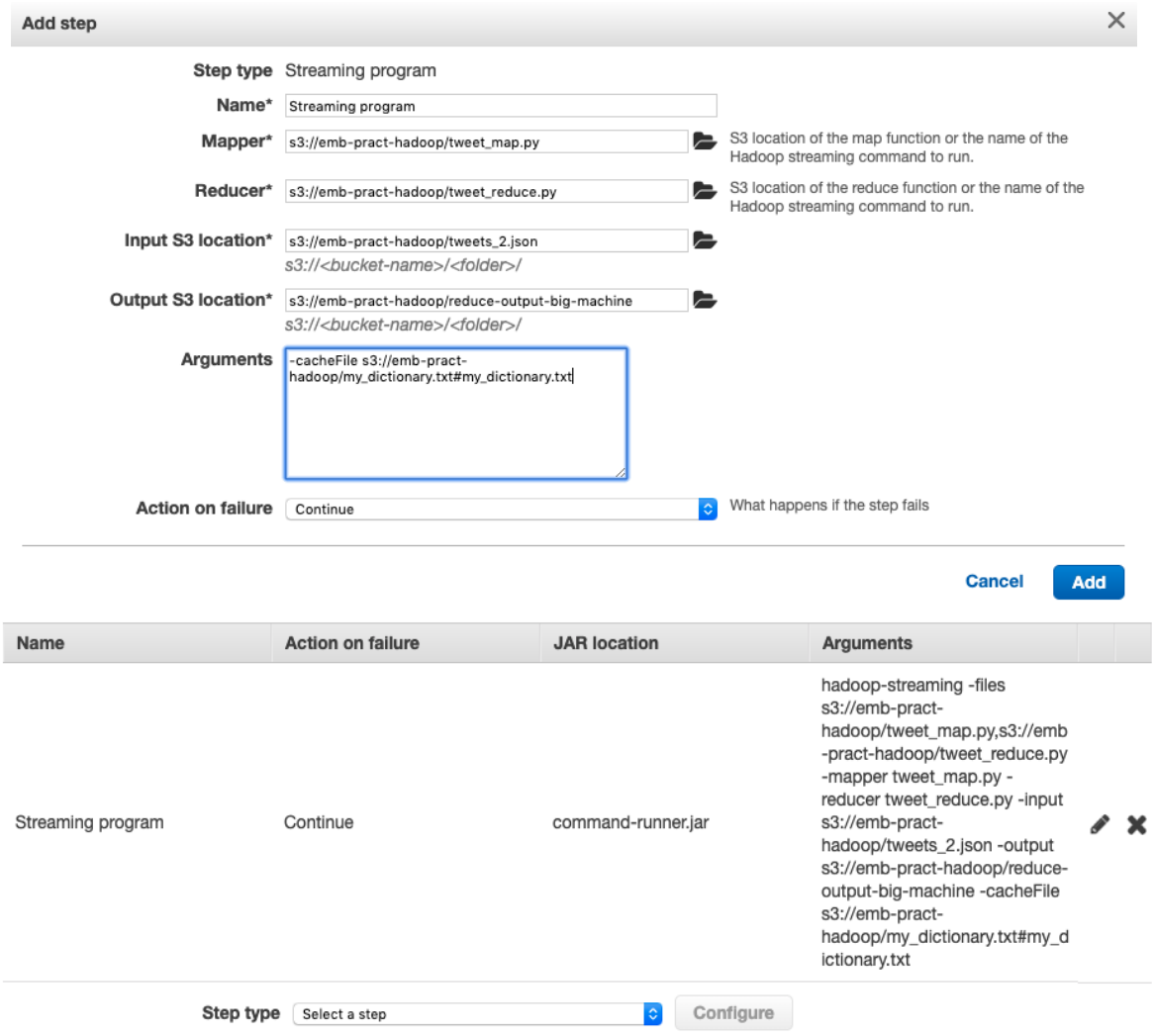
Diferentes formas de ejecución realizadas incluyendo los parámetros utilizados.

Amazon Web Services

La principal forma que hemos utilizado para ejecutar el map y el reduce ha sido a través de Amazon Web Services. Hemos subido los ficheros en S3 y posteriormente hemos creado en EMR un clúster con los siguientes parámetros:

- **Modo de lanzamiento:** ejecución de pasos.
- **Tipo de paso:** programa de retransmisión.
- **Configuración del tipo de paso:** hemos indicado las rutas al map, reduce, fichero de entrada y carpeta de salida. Hemos incluido como argumentos el diccionario.
- **Configuración del software:** emr-5.28.0.
- **Tipo de instancia:** m5.xlarge.
- **Número de instancias:** 3.
- **Permisos**: predeterminado.

Adicionalmente hemos realizado pruebas con diferentes parámetros que detallamos en el siguiente apartado. En las siguientes imágenes mostramos el ajuste de parámetros.



MRJOB

Con el MrJob utilizamos el mapper y el reducer en un mismo archivo, importamos mrjob.job y mrjob.step para poder trabajar con la librería. La clase MRWordFrequencyCount consta de un mapper, un reducer y los steps que ejecuta ambos.

En el mapper se leen los Words por líneas que vienen del fichero que hemos mandado a la función mediante los parámetros de ejecución. Se hace un filtro por país en el que se seleccionan sólo aquellos que pertenecen a estados unidos. Después seleccionamos nuestra clave y valor, en este caso utilizaremos como clave el código del estado y como valor la suma de la puntuación de las palabras que están en el texto del tweet.

Además se ha utilizado este mapper para hacer los trending topics en los que cogemos las palabras del tweet que empiezan por # y las apuntamos como clave para después contarlas en el reducer. Para poder procesar todas las palabras que están en el diccionario de sentimientos y poder recoger los hashtags como clave recorremos el texto del tweet con un bucle de palabra por palabra.

En el reducer recorremos lo obtenido en el mapper para hacer las cuentas de los datos que hemos obtenido, aquí separamos en dos tipos de claves, las que no tienen # que seran con las que hagamos el análisis de sentimiento y las que tienen el # con las que haremos un contador para ver el trending topic.

Para poder lanzar el mr job en local utilizaremos en la consola:
`python mr_job.py -r inline Words.json`
y obtendremos:

- Hashtags:

  "#THWGT" {"Cont": 1}
  "#Tab\u00fa" {"Cont": 3}
  "#Thanksgiving" {"Cont": 4}
  "#Thanksgiving?" {"Cont": 1}
  "#Timberwolves" {"Cont": 1}

- Sentimiento por estado:

  "KY" {"Total": 6, "Media": 1.2}
  "LA" {"Total": 37, "Media": 1.4230769230769231}
  "MA" {"Total": -2, "Media": -0.10526315789473684}
  "MD" {"Total": 3, "Media": 0.1875}
  "MI" {"Total": -6, "Media": -0.42857142857142855}

Evaluación en la medida de los posible de escalabilidad y elasticidad de la solución propuesta.

Para analizar la escalabilidad y elasticidad hemos generado tres ficheros de Words de tamaños diferentes:

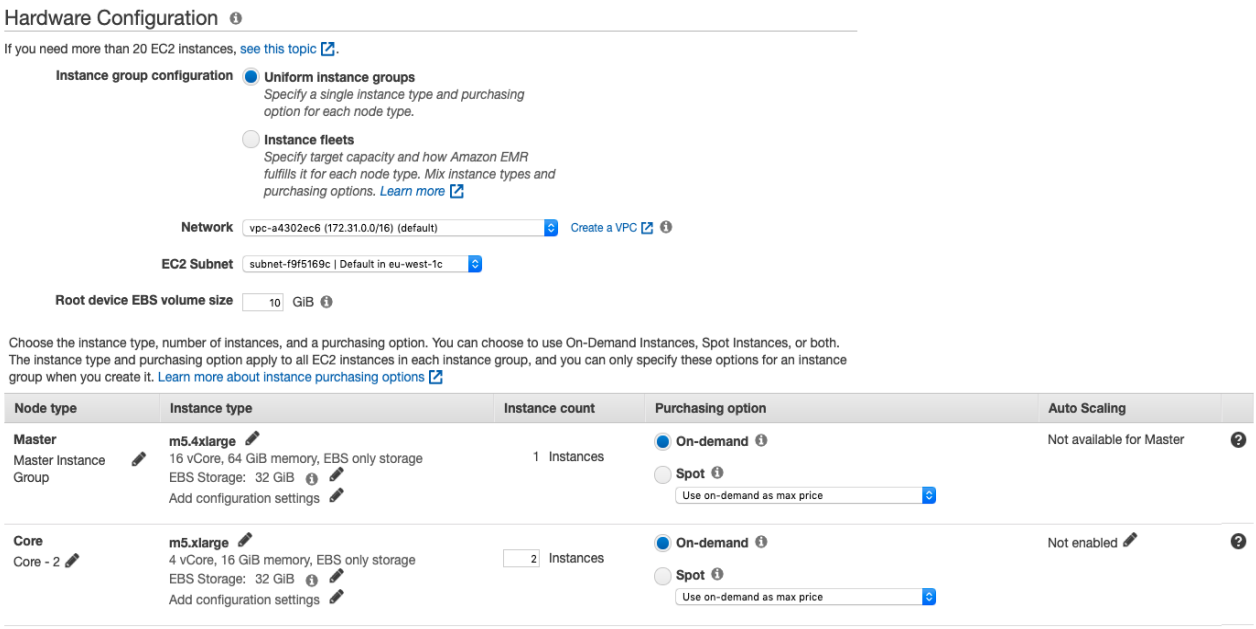
A continuación, hemos ejecutado cada uno de ellos con diferentes máquinas: por un lado hemos utilizado una máquina estándar como es la m1.large con un nodo y por otro lado hemos querido ver los resultados con una máquina más grande como es la m5.xlarge para los nodos y m5.4xlarge para el máster.

Los resultados obtenidos son los siguientes:

Fichero	Tipo de instancia	Número de instancias	Tiempo de ejecución (min)
small size	m1.large	1	2
small size	m5.4xlarge (máster) m5.xlarge (core)	3	46s
medium size	m1.large	1	7
medium size	m5.4xlarge (máster) m5.xlarge (core)	3	1
large size	m1.large	1	31
large size	m5.4xlarge (máster) m5.xlarge (core)	3	6

Podemos observar que a mayores nodos, más rápido se ejecuta el proceso y que efectivamente con máquinas más grandes el tiempo de ejecución también se reduce mucho. Por ejemplo, con el archivo más grande de Words vemos que podemos pasar de 31 minutos a 6 minutos.

Mostramos las características de las máquinas usadas para instancias más grandes.



Hardware Configuration

If you need more than 20 EC2 instances, see this topic.

Instance group configuration

- **Uniform instance groups** — *Specify a single instance type and purchasing option for each node type.*
- **Instance fleets** — *Specify target capacity and how Amazon EMR fulfills it for each node type. Mix instance types and purchasing options. Learn more*

Network vpc-a4302ec6 (172.31.0.0/16) (default) Create a VPC

EC2 Subnet subnet-f9f5169c | Default in eu-west-1c

Root device EBS volume size 10 GiB

Choose the instance type, number of instances, and a purchasing option. You can choose to use On-Demand Instances, Spot Instances, or both. The instance type and purchasing option apply to all EC2 instances in each instance group, and you can only specify these options for an instance group when you create it. Learn more about instance purchasing options

Node type	Instance type	Instance count	Purchasing option	Auto Scaling
Master Master Instance Group	**m5.4xlarge** 16 vCore, 64 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	**On-demand** **Spot** Use on-demand as max price	Not available for Master
Core Core - 2	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	2 Instances	**On-demand** **Spot** Use on-demand as max price	Not enabled

Captura de los tiempos de ejecución.

▸	MR Tweets MK2 Large	j-1RNE6QOY16T7H	Terminated All steps completed	2020-01-11 16:06 (UTC+0)	10 minutes	48
▸	MR Tweets MK2 medium	j-9FXTRMKK20Q5	Terminated All steps completed	2020-01-11 16:04 (UTC+0)	6 minutes	32
▸	MR Tweet MK2 small size	j-3UM6IW5LBNWY7	Terminated All steps completed	2020-01-11 16:01 (UTC+0)	5 minutes	24
▸	MR Tweets mk2 Large	j-1Y2Y1OJQXN70K	Terminated All steps completed	2020-01-11 13:09 (UTC+0)	37 minutes	4
▸	MR Tweets mk2 medium	j-3LF9SJ7ZSXS8P	Terminated All steps completed	2020-01-11 13:09 (UTC+0)	13 minutes	4
▸	MR Tweet mk2 small size	j-1RTK78ED6D4IN	Terminated All steps completed	2020-01-11 13:08 (UTC+0)	9 minutes	4

Comentarios personales

Incluyendo problemas encontrados, críticas constructiva, propuesta de mejoras y evaluación del tiempo dedicado.

- **Problemas encontrados en parsear los Words:**

Fue necesario convertir el texto en formato UTF-8 ya que los emoticonos y caracteres extraños estaban en formato ASCII.
En las primeras iteraciones decidimos únicamente seleccionar texto y location, para evitar el problema del encoder con otros atributos. Una vez codificado el json a UTF-8 pudimos agregar todos los atributos del tweet.

- **Problemas encontrados en MrJob:**

Ha sido imposible poder seleccionar los trending topics con más contadores en el reducer.

Hemos tratado de utilizar Multi-step jobs para poder hacer una selección pero no hemos sido capaces. ya que no nos devolvía nada más que un Process finished with exit code 1.

Por como funciona nuestro código de Mrjob y al tener en un mismo mapper ambas claves no hemos podido filtrar bien los hashtags por contador.

Como esta parte no ha funcionado hemos decidido utilizar el AWS con el map y el reduce que nos ha quedado mucho más visual y simplificado.

- **Problemas encontrados en AWS:**

Aunque el map y reduce se ejecutaban bien en local, al subir el mismo código en AWS se produce un error debido a Words que contienen fragmentos en código ASCII, por ejemplo ‘/u206’, por lo que lo hemos tenido que formatear a encoding UTF-8.

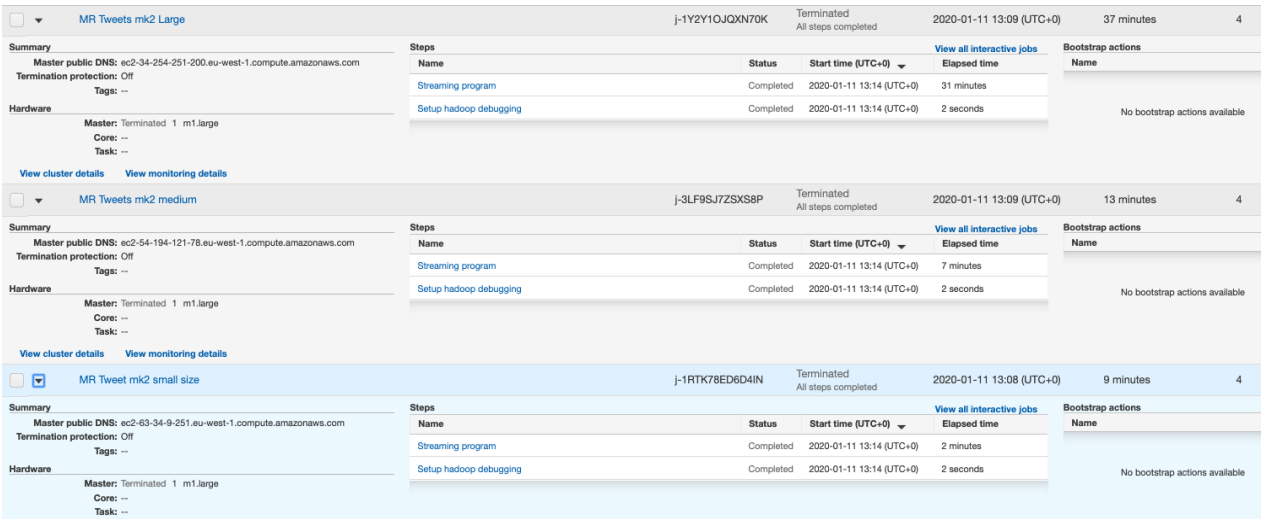
Al trabajar con los hashtags y el UTF-8 han surgido algunas complicaciones, como por ejemplo aparecen los caracteres \MD que dificultan el procesado.
Otro problema que descubrimos fue que AWS utiliza una versión de python 2 y nosotros usamos una python 3 en local. Tuvimos que ajustar las librerías para usar python 2.

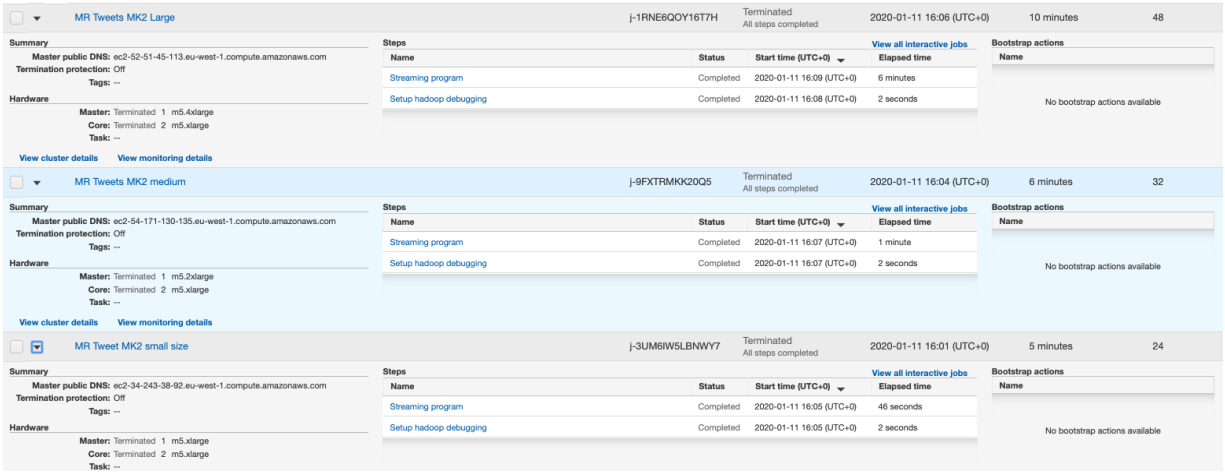
Anexo: resultados obtenidos

ME Happines Score : 2 Total Words: 41 Total Words Value : 4 Words Mean : 0
NJ Happines Score : 32 Total Words: 785 Total Words Value : 41 Words Mean : 0
MN Happines Score : 15 Total Words: 373 Total Words Value : 23 Words Mean : 0
CO Happines Score : -16 Total Words: 353 Total Words Value : 23 Words Mean : -1
NM Happines Score : 1 Total Words: 221 Total Words Value : 16 Words Mean : 0
WA Happines Score : 19 Total Words: 672 Total Words Value : 32 Words Mean : 0
NY Happines Score : 37 Total Words: 2683 Total Words Value : 119 Words Mean : 0
WV Happines Score : -6 Total Words: 43 Total Words Value : 3 Words Mean : -2
AK Happines Score : 4 Total Words: 91 Total Words Value : 2 Words Mean : 2
DE Happines Score : 6 Total Words: 45 Total Words Value : 2 Words Mean : 3
ND Happines Score : 0 Total Words: 21 Total Words Value : 0 Words Mean : 0
RI Happines Score : -5 Total Words: 87 Total Words Value : 9 Words Mean : -1
KS Happines Score : -3 Total Words: 189 Total Words Value : 13 Words Mean : -1
MT Happines Score : 0 Total Words: 4 Total Words Value : 0 Words Mean : 0
IL Happines Score : 44 Total Words: 1164 Total Words Value : 61 Words Mean : 0
KY Happines Score : 6 Total Words: 301 Total Words Value : 13 Words Mean : 0
AZ Happines Score : 29 Total Words: 802 Total Words Value : 45 Words Mean : 0
FL Happines Score : 16 Total Words: 1669 Total Words Value : 72 Words Mean : 0
OR Happines Score : 10 Total Words: 473 Total Words Value : 20 Words Mean : 0
NV Happines Score : 11 Total Words: 805 Total Words Value : 38 Words Mean : 0
WY Happines Score : 1 Total Words: 39 Total Words Value : 1 Words Mean : 1
MD Happines Score : 13 Total Words: 694 Total Words Value : 32 Words Mean : 0
MA Happines Score : -23 Total Words: 823 Total Words Value : 40 Words Mean : -1
MS Happines Score : -11 Total Words: 288 Total Words Value : 22 Words Mean : -1
TX Happines Score : 49 Total Words: 5170 Total Words Value : 333 Words Mean : 0
OH Happines Score : 29 Total Words: 916 Total Words Value : 42 Words Mean : 0
UT Happines Score : 43 Total Words: 411 Total Words Value : 25 Words Mean : 1
PA Happines Score : 15 Total Words: 1105 Total Words Value : 66 Words Mean : 0
NC Happines Score : 47 Total Words: 929 Total Words Value : 44 Words Mean : 1
WI Happines Score : 10 Total Words: 253 Total Words Value : 11 Words Mean : 0
VA Happines Score : 0 Total Words: 988 Total Words Value : 64 Words Mean : 0
HI Happines Score : 6 Total Words: 172 Total Words Value : 10 Words Mean : 0
GA Happines Score : -52 Total Words: 1058 Total Words Value : 47 Words Mean : -2
IN Happines Score : 17 Total Words: 516 Total Words Value : 28 Words Mean : 0
OK Happines Score : 6 Total Words: 285 Total Words Value : 16 Words Mean : 0
SD Happines Score : 3 Total Words: 39 Total Words Value : 2 Words Mean : 1
CT Happines Score : 29 Total Words: 386 Total Words Value : 25 Words Mean : 1
NH Happines Score : 4 Total Words: 59 Total Words Value : 3 Words Mean : 1
NE Happines Score : -9 Total Words: 97 Total Words Value : 8 Words Mean : -2
LA Happines Score : -8 Total Words: 1124 Total Words Value : 65 Words Mean : -1
CA Happines Score : 102 Total Words: 7255 Total Words Value : 376 Words Mean : 0
MI Happines Score : -10 Total Words: 721 Total Words Value : 42 Words Mean : -1
AL Happines Score : 0 Total Words: 500 Total Words Value : 22 Words Mean : 0

DC Happines Score : 2 Total Words: 302 Total Words Value : 10 Words Mean : 0
TN Happines Score : -2 Total Words: 692 Total Words Value : 38 Words Mean : -1
SC Happines Score : 6 Total Words: 368 Total Words Value : 19 Words Mean : 0
AR Happines Score : 8 Total Words: 160 Total Words Value : 11 Words Mean : 0
IA Happines Score : 24 Total Words: 293 Total Words Value : 13 Words Mean : 1
MO Happines Score : 11 Total Words: 434 Total Words Value : 19 Words Mean : 0
ID Happines Score : 7 Total Words: 109 Total Words Value : 3 Words Mean : 2

#NEvsHOU	8
#JoeBurrow!	5
#90DayFiance	5
#Watchmen	5
#WeAreTexans	5
#pdx911	5
#NewProfilePic	4
#Texans	4
#Patriots	4
#SNFonNBC	3
#MyTwitterAnniversary	3
#MonarcasMorelia	3
#RHOA	3
#WWEStarrcade	3
#TwitterMomentsOfTheDecade	3
#NoTimeToDie	3
#Christmas	2
#SundayThoughts	2
#Blessed	2
#NFL	3
count_tweets	3746





MR Tweets MK2 Large | j-1RNE6QOY16T7H | Terminated, All steps completed | 2020-01-11 16:06 (UTC+0) | 10 minutes | 48

Summary
Master public DNS: ec2-52-51-45-113.eu-west-1.compute.amazonaws.com
Termination protection: Off
Tags: --

Hardware
Master: Terminated 1 m5.4xlarge
Core: Terminated 2 m5.xlarge
Task: --

View cluster details | View monitoring details

Steps — View all interactive jobs

Name	Status	Start time (UTC+0)	Elapsed time
Streaming program	Completed	2020-01-11 16:09 (UTC+0)	6 minutes
Setup hadoop debugging	Completed	2020-01-11 16:08 (UTC+0)	2 seconds

Bootstrap actions

Name

No bootstrap actions available

MR Tweets MK2 medium | j-9FXTRMKK20Q5 | Terminated, All steps completed | 2020-01-11 16:04 (UTC+0) | 6 minutes | 32

Summary
Master public DNS: ec2-54-171-130-135.eu-west-1.compute.amazonaws.com
Termination protection: Off
Tags: --

Hardware
Master: Terminated 1 m5.2xlarge
Core: Terminated 2 m5.xlarge
Task: --

View cluster details | View monitoring details

Steps — View all interactive jobs

Name	Status	Start time (UTC+0)	Elapsed time
Streaming program	Completed	2020-01-11 16:07 (UTC+0)	1 minute
Setup hadoop debugging	Completed	2020-01-11 16:07 (UTC+0)	2 seconds

Bootstrap actions

Name

No bootstrap actions available

MR Tweet MK2 small size | j-3UM6IW5LBNWY7 | Terminated, All steps completed | 2020-01-11 16:01 (UTC+0) | 5 minutes | 24

Summary
Master public DNS: ec2-34-243-38-92.eu-west-1.compute.amazonaws.com
Termination protection: Off
Tags: --

Hardware
Master: Terminated 1 m5.xlarge
Core: Terminated 2 m5.xlarge
Task: --

Steps — View all interactive jobs

Name	Status	Start time (UTC+0)	Elapsed time
Streaming program	Completed	2020-01-11 16:05 (UTC+0)	46 seconds
Setup hadoop debugging	Completed	2020-01-11 16:05 (UTC+0)	2 seconds

Bootstrap actions

Name

No bootstrap actions available