

**UNIVERSIDAD EAFIT**

**MAESTRÍA EN CIENCIA DE DATOS Y ANALÍTICA**

**ST1612 SISTEMAS INTENSIVOS EN DATOS**

**PROFESOR:**

**EDWIN MONTOYA**

**ALUMNOS:**

Cindy Paola Guerra Medina

Carlos Alberto Murillo Martínez

Luz Stella Flórez Salazar

**Proyecto Final: Análisis de sentimiento de tweets en tiempo real sobre la  
cuenta de Daniel Quintero, alcalde de la ciudad de Medellín.**

Fecha de entrega: 10 de noviembre de 2021

## **Introducción:**

La red social Twitter se ha convertido en una importante fuente para el análisis de datos, debido a su gran volumen de usuarios, los cuales aportan diversas opiniones en cada una de sus publicaciones. Sobre estas opiniones, se puede realizar análisis de sentimientos, es el estudio computacional de los sentimientos, emociones y actitudes de las personas (Ligthart et al., 2021) que se pueden extraer de las opiniones incluidas en las comunicaciones textuales, como reseñas de películas o productos (Wyeld et al., 2021). Una opinión de una persona puede tener una valoración positiva, neutral o negativa acerca de un producto, servicio, organización, persona o temas que se esté tratando.

Twitter es uno de los mayores servicios de información en tiempo real que dispone una API para extraer datos y así poder realizar análisis de sentimientos sobre personas importantes como Daniel Quintero, alcalde de la ciudad de Medellín. Diversos usuarios publican mensajes de texto con sus opiniones, inquietudes, quejas, reclamos, felicitaciones, etc.

## **Objetivo general**

El objetivo de este proyecto es, implementar en una arquitectura de AWS, un análisis de sentimientos de tweets en tiempo real sobre la cuenta de Daniel Quintero, alcalde de la ciudad de Medellín.

### **Objetivos específicos**

- Identificar y desplegar la arquitectura Serverless que ejecute el análisis de sentimientos en tiempo real.
- Determinar el sentimiento (numérico) y categorizarlo como positivo, neutral o negativo, de cada uno de los Tweets que ingresen.
- Visualizar los resultados a través de consultas SQL
- Visualizar gráficamente de los resultados del análisis de sentimientos.

### **Descripción del Caso**

A través de los servicios de AWS estudiados durante la clase de sistemas intensivos en datos y herramientas opensource, queremos implementar un análisis en tiempo real sobre la cuenta de Daniel Quintero Calle (@QuinteroCalle). Se extraerán los datos a través de la API disponible de la red social Twitter, para la integración y descargue de información, de esta forma lograremos en una arquitectura lambda, tener dos caminos de solución:

1. Tener información batch para análisis.
2. Análisis del sentimiento en tiempo real.

Para el desarrollo del trabajo, usaremos varios servicios tecnológicos:

- Kibana
- Comprehend
- Elastic Search
- S3
- Glue
- Lambda
- Athena
- DynamoDB
- Kinesis Data Streams
- EC2
- Kinesis Firehose
- Logstash

## Metodología de trabajo

Se emplea la metodología CRISP-DM, la cual denota el “Proceso estándar de la industria cruzada para la minería de datos”. El modelo del ciclo de vida consta de seis fases: conocimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación y despliegue. La secuencia de las fases no es estricta. De hecho, la mayoría de los proyectos avanzan y retroceden entre fases según sea necesario (IBM, 2021). Este modelo es flexible y se puede personalizar fácilmente de acuerdo con las necesidades particulares, lo que lo hace adecuado para este proyecto en particular.

## Arquitectura Lambda

A continuación, se visualiza la arquitectura utilizada para la ejecución y despliegue de la solución del trabajo.

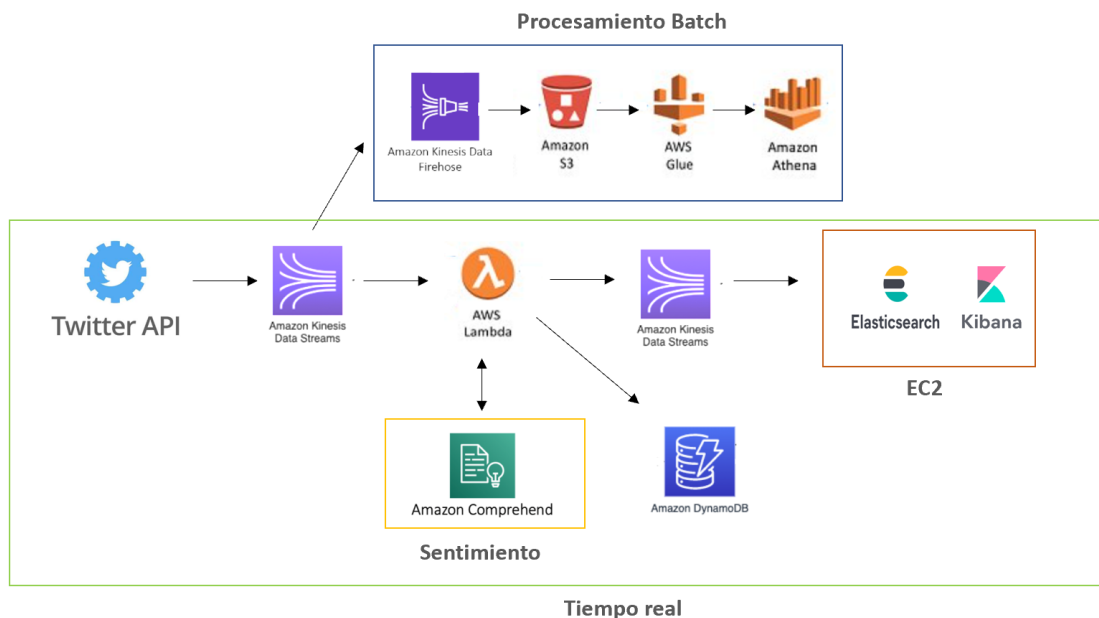


Imagen 1. Arquitectura Lambda

# Solución de la problemática

## 1. Conexión y extracción de datos de Twitter

Lo primero que realizamos, es entrar a la página para desarrolladores de Twitter y registrarnos, en el portal de desarrolladores. Creamos un nuevo proyecto y se muestra como se llevan los datos tweet hacia nuestra solución.

```
tweet_dict = {
    'id' : self.id,
    'text' : ud.unidecode(status.text),
    'created_time' : status.created_at.isoformat(),
    'source' : status.source,
    'tweet_id' : status.id_str,
    'user_name' : status.user.name,
    'user_id' : status.user.id_str,
    'run_start_time': self.start_time,
    'run_time_limit': self.limit
}
```

el envío de datos a Kinesis

```
def send_to_kinesis(self, stream_name, kinesis_client, tweet_dict):
    data = tweet_dict
    kinesis_client.put_record(
        StreamName=stream_name,
        Data=json.dumps(data),
        PartitionKey="partitionkey")
```

```
def background(stream):
    stream.filter(track = ['quinterocalle', 'medellin', 'alcalde'],
        languages = ['en', 'es'], threaded=True)

def wait(minutes):
    for i in tqdm(range(int(60*minutes - 1))): # termine antes que aparezca el mensaje de finalización de conexión
        time.sleep(1) #update each second

if __name__ == '__main__':
    MINUTES = 60
    stream_name, kinesis_client = 'new_twitter_stream', boto3.client('kinesis')

    listener = StdOutListener(app_credentials.CONSUMER_KEY, app_credentials.CONSUMER_SECRET,
        app_credentials.ACCESS_TOKEN, app_credentials.ACCESS_TOKEN_SECRET,
        60*MINUTES, stream_name, kinesis_client)

    try:
        background(listener)
        wait(MINUTES)
    except Exception as e:
        print(e)
    finally:
        listener.disconnect()

#finaliza ejecucion
print("Ejecución terminada")
```

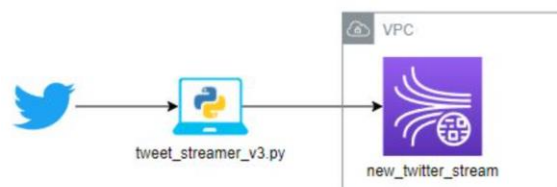
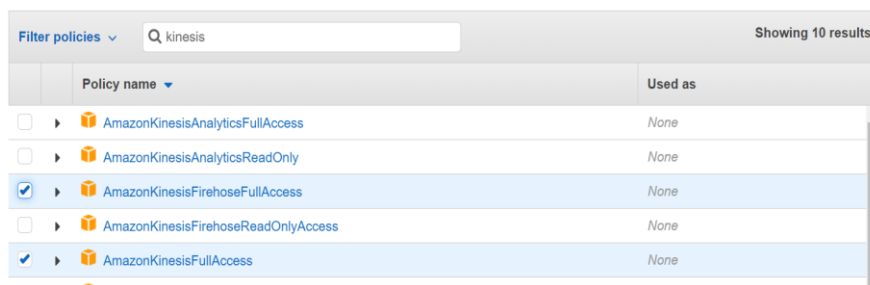


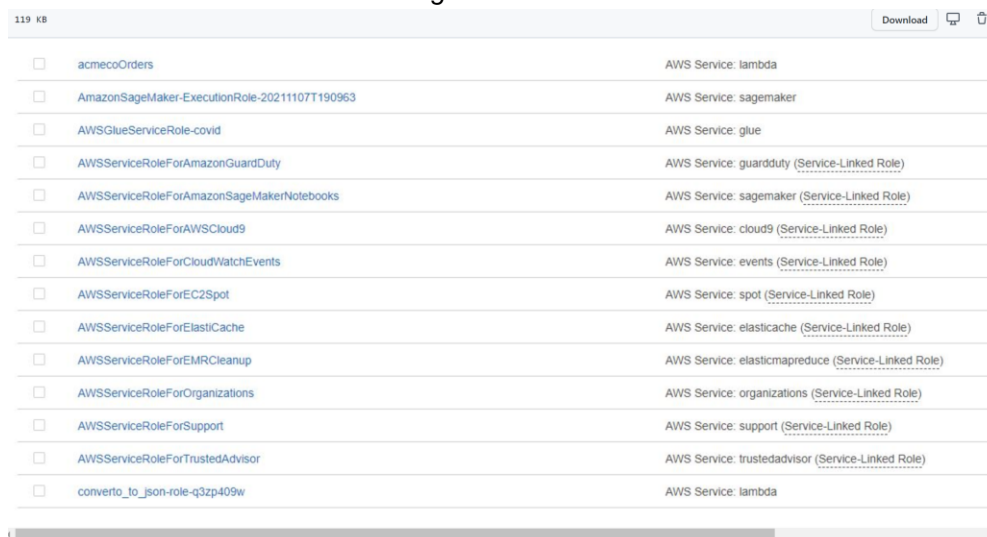
Imagen 1. Twitter

En el código anterior podemos observar los filtros utilizados para la extracción de la información a valorar; en la plataforma AWS se inicia con la configuración de los permisos en el Rol utilizado en el IAM, estos roles fueron configurados a medida que se configuraba cada servicio:



Policy name	Used as
<input type="checkbox"/> AmazonKinesisAnalyticsFullAccess	None
<input type="checkbox"/> AmazonKinesisAnalyticsReadOnly	None
<input checked="" type="checkbox"/> AmazonKinesisFirehoseFullAccess	None
<input type="checkbox"/> AmazonKinesisFirehoseReadOnlyAccess	None
<input checked="" type="checkbox"/> AmazonKinesisFullAccess	None

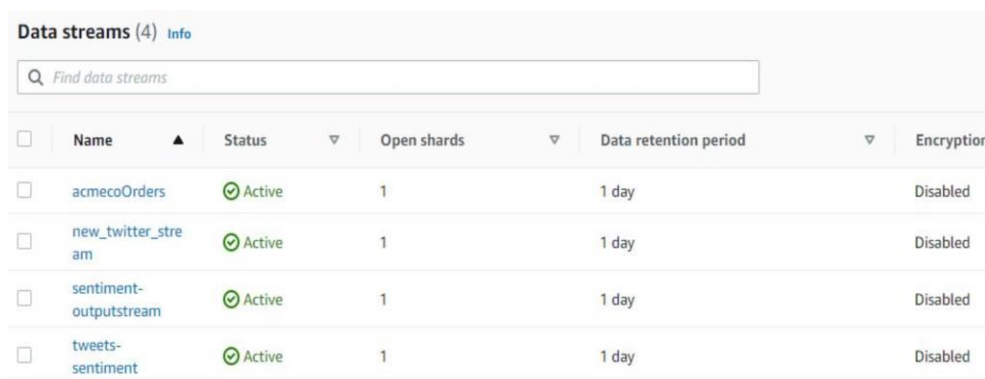
Imagen 2. Permisos



Role name	Service
acmecOrders	AWS Service: lambda
AmazonSageMaker-ExecutionRole-20211107T190963	AWS Service: sagemaker
AWSGlueServiceRole-covid	AWS Service: glue
AWSServiceRoleForAmazonGuardDuty	AWS Service: guardduty (Service-Linked Role)
AWSServiceRoleForAmazonSageMakerNotebooks	AWS Service: sagemaker (Service-Linked Role)
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked Role)
AWSServiceRoleForCloudWatchEvents	AWS Service: events (Service-Linked Role)
AWSServiceRoleForEC2Spot	AWS Service: spot (Service-Linked Role)
AWSServiceRoleForElasticCache	AWS Service: elasticache (Service-Linked Role)
AWSServiceRoleForEMRCleanup	AWS Service: elasticmapreduce (Service-Linked Role)
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked Role)
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)
convertio_to_json-role-q3zp409w	AWS Service: lambda

Imagen 3. Permisos

## 2. Entrega y configuración de datos para Kinesis Data Stream



Name	Status	Open shards	Data retention period	Encryption
acmecOrders	Active	1	1 day	Disabled
new_twitter_stream	Active	1	1 day	Disabled
sentiment-outputstream	Active	1	1 day	Disabled
tweets-sentiment	Active	1	1 day	Disabled

Imagen 4. Twitter

Con la anterior configuración, podemos pasar al **camino batch** el cual nos va a permitir tener un almacenamiento de los datos para ser utilizados en algún momento para otros estudios. El siguiente paso, es llevar la información de los datos al servicio de Kinesis Firehose:

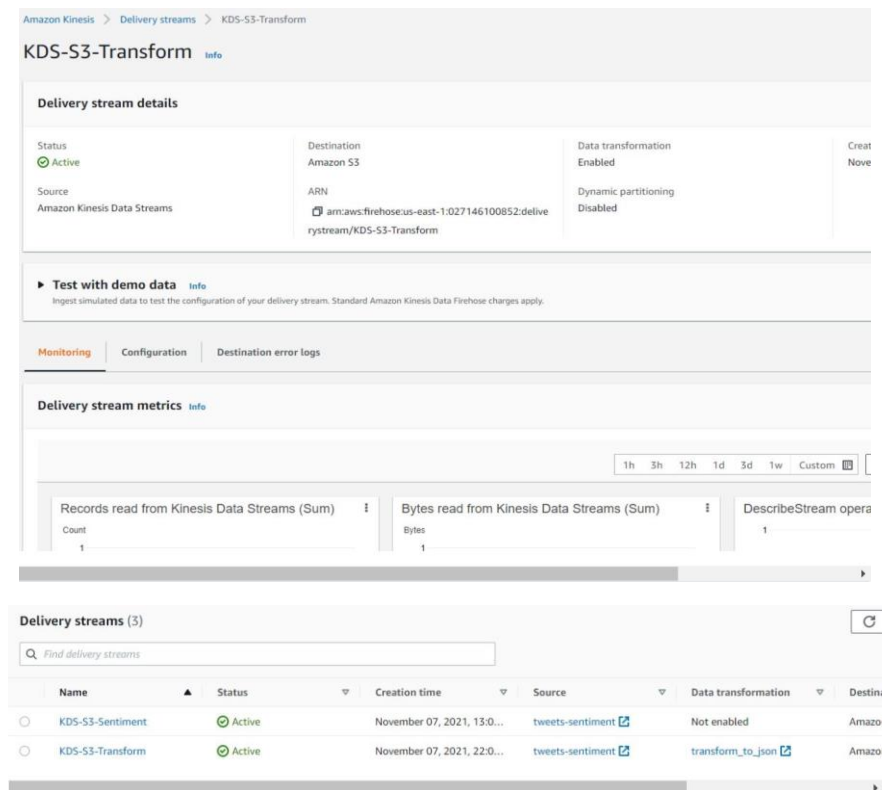


Imagen 5. Configuración Kinesis Firehose

Se muestra que el servicio queda activo y con conexión a AWS S3, e cual vamos a utilizar para el almacenamiento, a medida que se ejecuta el proceso se va realizando el proceso con un comentario demo, el cual se mantiene para verificar que el proceso este ejecutando en buen estado.

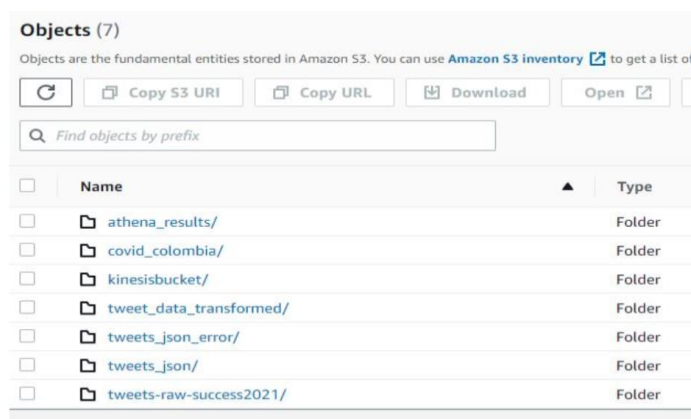


Imagen 6. S3

La anterior imagen nos muestra los buckets que fueron creados para el almacenamiento de la información, la cual fue guardada en formato Json.

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

[Add crawler](#) [Run crawler](#) [Action](#)

<input type="checkbox"/>	Name	Schedule	Status	Logs	Last runtime
<input type="checkbox"/>	covid_vaccination_crawler		Ready	<a href="#">Logs</a>	1 min
<input type="checkbox"/>	covid_vases_crawler		Ready	<a href="#">Logs</a>	52 secs
<input checked="" type="checkbox"/>	json_raw_kinesis		Ready	<a href="#">Logs</a>	1 min
<input type="checkbox"/>	logsacme		Ready	<a href="#">Logs</a>	55 secs
<input checked="" type="checkbox"/>	tweets_kinesis_transform_crawler		Ready	<a href="#">Logs</a>	1 min
<input type="checkbox"/>	vaccinated_curated_crawler		Ready	<a href="#">Logs</a>	59 secs

[Schemas](#) > [kinesis\\_tweet\\_schema](#) > 6

### Schema version details

Version ID	4c07dc2a-7c00-4a73-96cb-26aeca515880
Schema name	<a href="#">kinesis_tweet_schema</a>
Registry name	<a href="#">sentiment_test_firehose</a>
Data format	JSON
Compatibility mode	Disabled compatibility
Schema tags	-

### Schema version definition

Version 6

```
1  {
2    "$id": "http://json-schema.org/draft-07/myjsonschema",
3    "$schema": "http://json-schema.org/draft-07/schema#",
4    "title": "Tweet",
5    "properties": {
6      "id": {
7        "type": "integer",
8        "description": "Id asignado por mi programa generador"
9      },
10     "text": {
11       "type": "string",
12       "description": "texto del tweet a analizar"
13     },
14     "created_at": {
```

Imagen 7. Glue

Se crearon los crawlers para generar los catálogos de los datos, lo cuales se muestra a continuación:

Name	json_raw_kinesis
Description	Create a single schema for each S3 path
Table level	false
Security configuration	
Tags	-
State	Ready
Schedule	
Last updated	Sun Nov 07 23:01:37 GMT-500 2021
Date created	Sun Nov 07 23:01:37 GMT-500 2021
Database	kinesis
Table prefix	tweets_json_
Service role	service-role/AWSGlueServiceRole-covid
Selected classifiers	
Data store	S3
Include path	s3://rawcmurill5/tweets-raw-success2021
Connection	
Exclude patterns	
Configuration options	
Schema updates in the data store	Update the table definition in the data catalog.
Object deletion in the data store	Mark the table as deprecated in the data catalog.

id	text	created_time	source	tweet_id	user_name
181	RT @VillamilMauro: Alcalde @QuinteroCalle por que NADIE, ni la policía, ni la fiscalía, ni el ejercito (q se la pasa reduciendo muchachos...	2021-11-07T21:45:51+00:00	Twitter for iPhone	1457464340836143111	Francisco Cote
0	RT @elPeritaAFK: El mejor alcalde @JuanSGuarnizo https://t.co/SVMnv6wqrx	2021-11-08T04:28:57+00:00	Twitter for iPhone	1457565781747802114	Seiso ?? Borando
1	"RT @santiagoangel: ¡Que hacían divisiones de la "Guardia Nacional Bolivariana" este fin de semana en un evento en Facatativa señor alcalde..."	2021-11-08T04:28:58+00:00	Twitter for Android	1457565785614917635	JPA51979
2	RT @Vidales2Luis: En medellin, dirigentes de @ffecode haciendo proselitismo con el candidato comunista, Gustavo Petro https://t.co/YqwaK5rBI4	2021-11-08T04:29:03+00:00	Twitter for Android	1457565809795084289	Daniel Hernandez
3	RT @Ccarhuaz2: Como una bestia atacando a mujeres estos serenos Matones compinches de Castillo de por medio estan los billetes con su alcald...	2021-11-08T04:29:10+00:00	Twitter for Android	1457565837653528584	@Pilar Blas
4	RT @Aymaraguar: ¡Con Alegria! Estuvimos en el Mpio. #Guacara junto a nuestro Alcalde @castaneda10J entregando Morrales, Uniformes y Zapat...	2021-11-08T04:29:20+00:00	Twitter for Android	1457565878363492359	Territorio Escolar Los Guayos
5	@Proferitypatia @EpicursaB8 @juanbedoya73 Same: me ha ido super bien. ¿Sera que tienen mejor empresa de encomiendas para Medellin?	2021-11-08T04:29:35+00:00	Twitter for Android	1457565940841918464	Blankita en Problemas
6	RT @CarlentonDavila: El candidato a Alcalde del Partido Nacional pidio que trajeran a toda la estructura de su partido hacia la Capital por...	2021-11-08T04:29:35+00:00	Twitter for iPhone	1457565940846141443	Leo Rhn

Imagen 8. Catálogo de datos

Con la catalogación de los tweets almacenados al corte del proceso, podemos crear las tablas (2 tablas) que podemos consultar en Athena:

19.8 KB	Download		
<input type="checkbox"/> tweets_json_00	kinesis	s3://rawcmurill5/tweets-raw-success2021/1...	json
<input type="checkbox"/> tweets_json_07	kinesis	s3://rawcmurill5/tweets-raw-success2021/1...	json

Imagen 9. Tablas

Con las tablas, podemos hacer la consulta en SQL para visualizar los primeros 10 tweets captados:



The screenshot shows a SQL query execution interface. At the top, there's a tab for 'Query 1' with a close button. Below it, the SQL query is displayed: `SELECT * FROM "kinesis"."tweets_json_tweets_raw_success2021" limit 10;`. The interface includes buttons for 'Run again', 'Cancel', 'Save as', 'Clear', and 'Create'. A status bar indicates 'Completed' with 'Time in queue: 0.146 sec' and 'Run time: 0.928 sec'. Below this, the 'Results (10)' are shown, with a search bar and a 'Copy' button. The results are presented in a table with columns 'id', 'text', and 'creat'.

id	text	creat
306	RT @UnaBansheeMas: 303.402 votos lo eligieron alcalde 305.000 firmas para revocarlo Lidia con eso @QuinteroCalle	2021
1438	Esto le corresponde a San Miguelito. @musamicontigo @hvcarrasquilla . Esto es facil hacer lo que hace el alcalde de... <a href="https://t.co/Occ5IOwVx5">https://t.co/Occ5IOwVx5</a>	2021
228	<a href="https://t.co/IVBmLw9lf9">https://t.co/IVBmLw9lf9</a> En diciembre de 1993, fue la muerte de Pablo Emilio Escobar Gavidia, jefe del cartel de Med... <a href="https://t.co/AuUmlytflP">https://t.co/AuUmlytflP</a>	2021
0	RT @SdrodriguezT: Senor @PetroGustavo nos puede decir si los que intentaron robar la fundidora de oro en Medellin que usaban prendas de su...	2021
138	RT @CitlaCarvajal: En Mérida Yucatán somos ejemplo entre hacer bien las cosas y el panorama deprimente de una "Transformacion" La Ciudad B...	2021
1721	@luispelaej @EPMestamosahi @hidroituang @QuinteroCalle @TodosXMedellin @JDValde @AndresElGury @DimayZepol... <a href="https://t.co/xNXlBmkuUG">https://t.co/xNXlBmkuUG</a>	2021
769	RT @Joshycastillo: El Alcalde de Nagarote Juan Gabriel Hernandez en un audio que me acaban de compartir. Operacion Repela para poder cumpli...	2021

Imagen 10. Tablas

Con la anterior tabla, culminamos el proceso o ramificación que recreamos para mostrar la parte batch. A continuación, continuamos con el procesamiento en tiempo real.

### 3. Lambda

Con la conexión y configurado el servicio de Kinesis Data Streams, continuamos con la creación de una Lambda, donde inicialmente con algunas trasformaciones como cambiar el formato de los tweets a .csv, para poder utilizarlo para el análisis de sentimientos.

The screenshot shows the AWS Lambda console 'Functions' page. It displays a list of 4 functions. Each function entry includes a checkbox, the function name, a description, the package type, and the runtime.

	Function name	Description	Package type	Runtime
<input type="checkbox"/>	transform_kinesis_to_csv	-	Zip	Python 3.9
<input type="checkbox"/>	ProcessOrders	-	Zip	Python 3.9
<input type="checkbox"/>	productOrders	-	Zip	Python 3.7
<input type="checkbox"/>	transform_to_json	timeout modified	Zip	Python 3.9

```

now = datetime.now()
year, month, day, hour = now.year, now.month, now.day, now.hour

def lambda_handler(event, context):

    decoded_record_data = [base64.b64decode(record['kinesis']['data']) for record in event['Records']]
    deserialized_data = [json.loads(decoded_record) for decoded_record in decoded_record_data]

    key = f'tweets_json/{year}/{month}/{day}/{hour}/tweets.csv'
    local_file_name = '/tmp/test.csv'

    s3 = boto3.client('s3')
    s3_resource = boto3.resource('s3')
    try:
        s3.head_object(Bucket='rawcmurill5', Key=key)
        # download s3 csv file to lambda tmp folder
        s3_resource.Bucket('rawcmurill5').download_file(key, local_file_name)
        with open(local_file_name, 'a', newline='') as outfile:
            writer = csv.writer(outfile, delimiter=';')
            for data in deserialized_data:
                dict = {}
                for k, v in data.items():
                    if isinstance(v, str):
                        dict[k] = ud.unidecode(v)

```

Imagen 11. Lambda

Para el proceso de la Lambda, se hace un control o monitoreo de su funcionamiento:

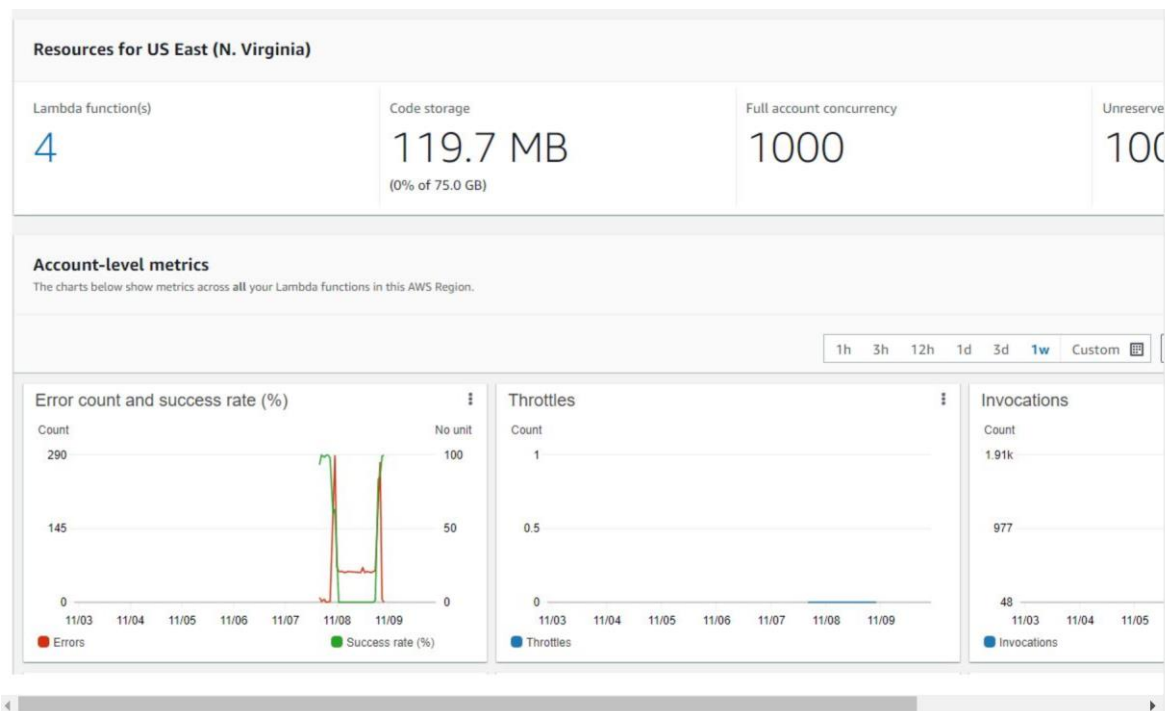


Imagen 11. Lambda

## 4. AWS Comprehend

Para realizar el análisis de sentimientos, utilizamos el servicio Comprehend de AWS, el cual ya tiene incorporado y entrenado el modelo de sentimientos que nos entregará la respuesta con varias variables, una con el scoring por cada clase (positivo, negativo, neutro, mixto) con el cual toma la decisión y el significado o categoría asignada positivo, neutro, negativo o mixto:

The screenshot displays the Amazon Comprehend console interface. On the left, a navigation pane shows the 'Real-time analysis' section expanded, with 'Analysis jobs' selected. The main area is titled 'Input text' and shows the 'Built-in' analysis type selected. The input text field contains the word 'Hola'. Below the input field, the 'Insights' section is visible, with the 'Sentiment' tab selected. The 'Analyzed text' section shows the word 'Hola'. The 'Results' section displays the sentiment scores: Neutral (0.04 confidence), Positive (0.00 confidence), Negative (0.94 confidence), and Mixed (0.00 confidence). The 'API call and API response of DetectSentiment API' section shows the JSON input and output for the API call.

**Input text**  
Supported languages [\[+\]](#)

**Analysis type**  
☒ Built-in  
View real-time insights based on AWS built-in models.  
☐ Custom  
View real-time insights based on custom models from an endpoint you've created.

**Input text**  
Hola  
4 of 5000 characters used.

**Clear text** **Analyze**

**Insights** [Info](#)

Entities | Key phrases | Language | PII | **Sentiment** | Syntax

**Analyzed text**  
Hola

**Results**  
**Sentiment**

Sentiment	Confidence
Neutral	0.04 confidence
Positive	0.00 confidence
Negative	0.94 confidence
Mixed	0.00 confidence

**Application integration**  
API call and API response of DetectSentiment API. [Info](#)

**API call**

```
1 {  
2   "Text": "Estas elecciones están malas",  
3   "LanguageCode": "es"  
4 }
```

**API response**

```
1 {  
2   "Sentiment": {  
3     "Sentiment": "NEGATIVE",  
4     "SentimentScore": {  
5       "Positive": 0.002269430732280016,  
6       "Negative": 0.948888486632996,  
7       "Neutral": 0.04428382051897084,  
8       "Mixed": 0.00534538383525276  
9     }  
10  }  
11 }
```

Imagen 12. Configuración manual Comprehend

Para que el proceso sea automático y en tiempo real, se configura en lambda que se realice el llamado automáticamente:

```
sentiment_all = comprehend.detect_sentiment(Text = text, LanguageCode = 'es')  
sentiment = sentiment_all['Sentiment']
```

Imagen 12. Configuración Lambda Comprehend

Con la información clasificada, podemos almacenar el resultado en una tabla de **DynamonDB**

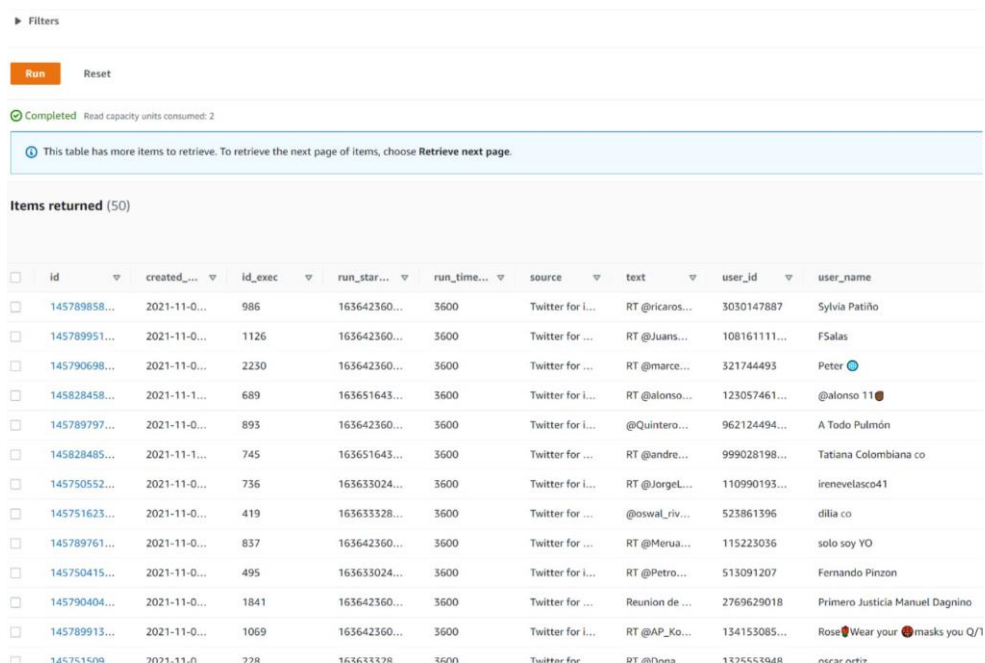
## 5. AWS DynamonDB

Para utilizar en otro momento, la información suministrada y calificada por el modelo realizado con Comprehend, se decide almacenar en una tabla en el servicio de AWS DynamonDB, a través de la Lambda:

```
with table.batch_writer() as batch_writer:
    # write to dynamo
    batch_writer.put_item(
        Item = dict_tweet
    )

with table_sentiments.batch_writer() as batch_writer:
    batch_writer.put_item(
        Item = dict_sentiment
    )
```

Imagen 13. Configuración Lambda DynamoDB



<input type="checkbox"/>	id	created_at	id_exec	run_star	run_time	source	text	user_id	user_name
<input type="checkbox"/>	145789858...	2021-11-0...	986	163642360...	3600	Twitter for i...	RT @ricaros...	3030147887	Sylvia Patiño
<input type="checkbox"/>	145789951...	2021-11-0...	1126	163642360...	3600	Twitter for ...	RT @Juans...	108161111...	FSalas
<input type="checkbox"/>	145790698...	2021-11-0...	2230	163642360...	3600	Twitter for ...	RT @marce...	321744493	Peter
<input type="checkbox"/>	145828458...	2021-11-1...	689	163651643...	3600	Twitter for i...	RT @alonso...	123057461...	@alonso 11
<input type="checkbox"/>	145789797...	2021-11-0...	893	163642360...	3600	Twitter for i...	@Quintero...	962124494...	A Todo Pulmón
<input type="checkbox"/>	145828485...	2021-11-1...	745	163651643...	3600	Twitter for ...	RT @andre...	999028198...	Tatiana Colombiana co
<input type="checkbox"/>	145750552...	2021-11-0...	736	163633024...	3600	Twitter for i...	RT @JorgeL...	110990193...	irenevelasco41
<input type="checkbox"/>	145751623...	2021-11-0...	419	163633328...	3600	Twitter for ...	@oswal_riv...	523861396	dilia co
<input type="checkbox"/>	145789761...	2021-11-0...	837	163642360...	3600	Twitter for ...	RT @Menua...	115223036	solo soy YO
<input type="checkbox"/>	145750415...	2021-11-0...	495	163633024...	3600	Twitter for i...	RT @Petro...	513091207	Fernando Pinzon
<input type="checkbox"/>	145790404...	2021-11-0...	1841	163642360...	3600	Twitter for ...	Reunion de ...	2769629018	Primerio Justicia Manuel Dagnino
<input type="checkbox"/>	145789913...	2021-11-0...	1069	163642360...	3600	Twitter for i...	RT @AP_Ko...	134153085...	Rose Wear your masks you Q/1
<input type="checkbox"/>	145751508...	2021-11-0...	728	163633328...	3600	Twitter for ...	RT @Dona...	1175555048	near net

Imagen 14. Tabla DynamoDB

Con la tabla creada, podemos ver que se almacenará de forma consecutiva a medida que ingrese nueva información, lo cual hace que la tabla esté actualizada en tiempo real en el momento que se requiera trabajar, adicional se obtiene nueva información como tiempo y sentimiento que se obtiene.

## 6. Máquina Virtual EC2

Para continuar con el proceso y llevar la información a tableros de control con Kibana, se decide montar una maquina virtual el cual se le instalan los componentes de Kibana, Logstash y elasticsearch.

- Instalar e iniciar elastic search  
bin/elasticsearch -d -p pid

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-24-85 ~]$ cd elasticsearch-7.15.1/
[ec2-user@ip-172-31-24-85 elasticsearch-7.15.1]$ bin/elasticsearch -d -p pid
```

- Instalar y ejecutar kibana  
cd kibana-7.15.1-linux-x86\_64/nohup bin/kibana &

```
[ec2-user@ip-172-31-24-85 ~]$ cd kibana-7.15.1-linux-x86_64/
[ec2-user@ip-172-31-24-85 kibana-7.15.1-linux-x86_64]$ nohup bin/kibana &
[1] 3793
[ec2-user@ip-172-31-24-85 kibana-7.15.1-linux-x86_64]$ nohup: ignoring input and appending output to 'nohup.out'
```

- Instalar plugin de logstash para trabajar con kinesis  
bin/logstash-plugin install logstash-input-kinesis
- Crear archivo de configuración en logstash para leer los tweets desde el stream de kinesis. Lo enviamos a un index llamado "twittersentiment"

```
ec2-user@ip-172-31-24-85:~/logstash-7.15.1
input {
  kinesis {
    kinesis_stream_name => "sentiment-outputstream"
    codec => json { }
  }
}

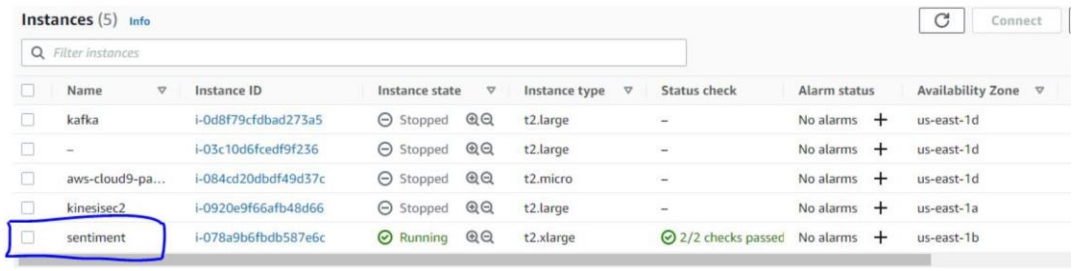
output {
  stdout { }

  elasticsearch {
    index => "twittersentiment"
  }
}
```

- Correr logstash  
bin/logstash -f etl-kinesis.conf

```
[ec2-user@ip-172-31-24-85 logstash-7.15.1]$ bin/logstash -f etl-kinesis.conf
Using bundled JDK: /home/ec2-user/logstash-7.15.1/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
```

- Se verifica la instancia creada para la máquina que este corriendo:



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	kafka	i-0d8f79cfdbad273a5	Stopped	t2.large	–	No alarms	us-east-1d
<input type="checkbox"/>	–	i-03c10d6fcedf9f236	Stopped	t2.large	–	No alarms	us-east-1d
<input type="checkbox"/>	aws-cloud9-pa...	i-084cd20dbdf49d37c	Stopped	t2.micro	–	No alarms	us-east-1d
<input type="checkbox"/>	kinesise2	i-0920e9f66afb48d66	Stopped	t2.large	–	No alarms	us-east-1a
<input type="checkbox"/>	sentiment	i-078a9b6fbdb587e6c	Running	t2.xlarge	2/2 checks passed	No alarms	us-east-1b

Imagen 15. Instancia EC2

## 7. Verificación de información

Se hace la verificación que en la máquina con las configuraciones y llamada de la instancia los datos este ingresando correctamente:

```
ec2-user@ip-172-31-24-85:~/logstash-7.15.1
{
  "@version" => "1",
  "score_positive" => 0.00658944109454751,
  "score_mixed" => 0.0021310702431946993,
  "score_negative" => 0.25456973910331726,
  "sentiment" => "NEUTRAL",
  "score_neutral" => 0.736709713935852,
  "created_time" => "2021-11-09T03:00:28+00:00",
  "text" => "RT @elcolombiano: [?]Fuentes cercanas a la administracion nos contaron que se ha barajado la id
ea de que Diana Osorio, la esposa del alcalde...",
  "id" => "1457905903232077827",
  "@timestamp" => 2021-11-10T02:34:03.874Z,
  "user_name" => "Dslibertariocolombia"
}
{
  "@version" => "1",
  "score_positive" => 0.017460636794567108,
  "score_mixed" => 0.04777759686112404,
  "score_negative" => 0.30990156531333923,
  "sentiment" => "NEUTRAL",
  "score_neutral" => 0.6248602867126465,
  "created_time" => "2021-11-09T03:00:36+00:00",
  "text" => "RT @Acorazado blind: Algunos chalecos antibalas que utilizaron en el asalto a la empresa de oro
en Medellin, fueron adquiridos por la alcal...",
  "id" => "1457905935381315584",
  "@timestamp" => 2021-11-10T02:34:03.874Z,
  "user_name" => "Fabian Lopez"
```

Imagen 16. Información en EC2

## 8. Configurar el índice en kibana

## Index patterns

Create and manage the index patterns that help you retrieve your data from Elasticsearch.

Pattern ↑

twittersentiment\*

Rows per page: 10 ↓

## Index Management

[Index Management docs](#)

[Indices](#) [Data Streams](#) [Index Templates](#) [Component Templates](#)

Update your Elasticsearch indices individually or in bulk. [Learn more](#). ☐ Include rollout indices ☐ Include hidden indices

Lifecycle status ↓ Lifecycle phase ↓ [Reload indices](#)

<input type="checkbox"/> Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
<input type="checkbox"/> twittersentiment	yellow	open	1	1	1	8.9kb	

Rows per page: 10 ↓

< 1 >

Imagen 17. Índice Kibana

Se verifica que los dos índices necesarios estén funcionando con el dato que se tiene de prueba para después verificar en tiempo real con el que ingrese.

## 9. Revisar tweets

Se verifica la información de tweets:

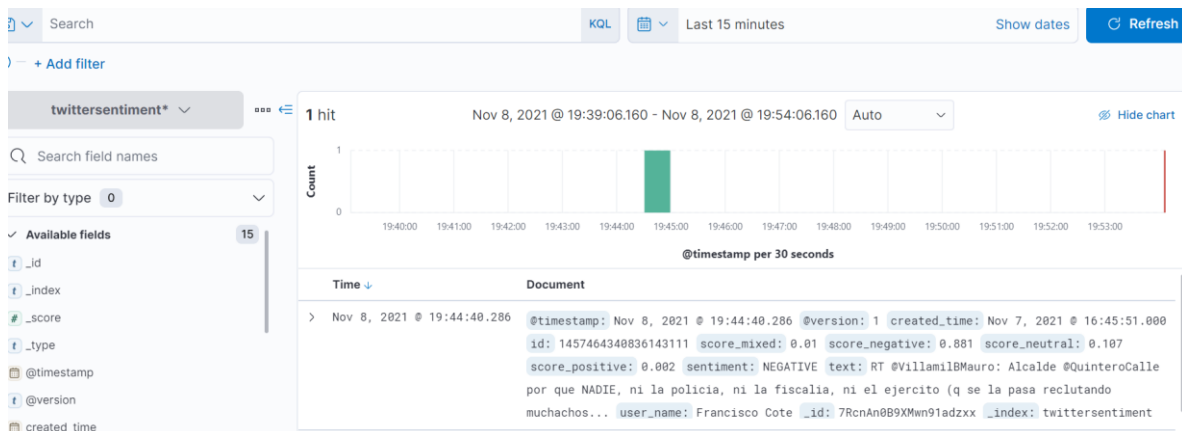


Imagen 18. Información en Kibana

## 10. Explorar los datos

Se realiza un primer análisis descriptivo de las variables de los tweets, donde se puede observar que se está obteniendo las categorías arrojadas por el modelo de sentimientos y las distribuciones que no siguen un patrón de normalidad, a corte del análisis de recibieron en 2 horas de procesamiento 1203 tweets.

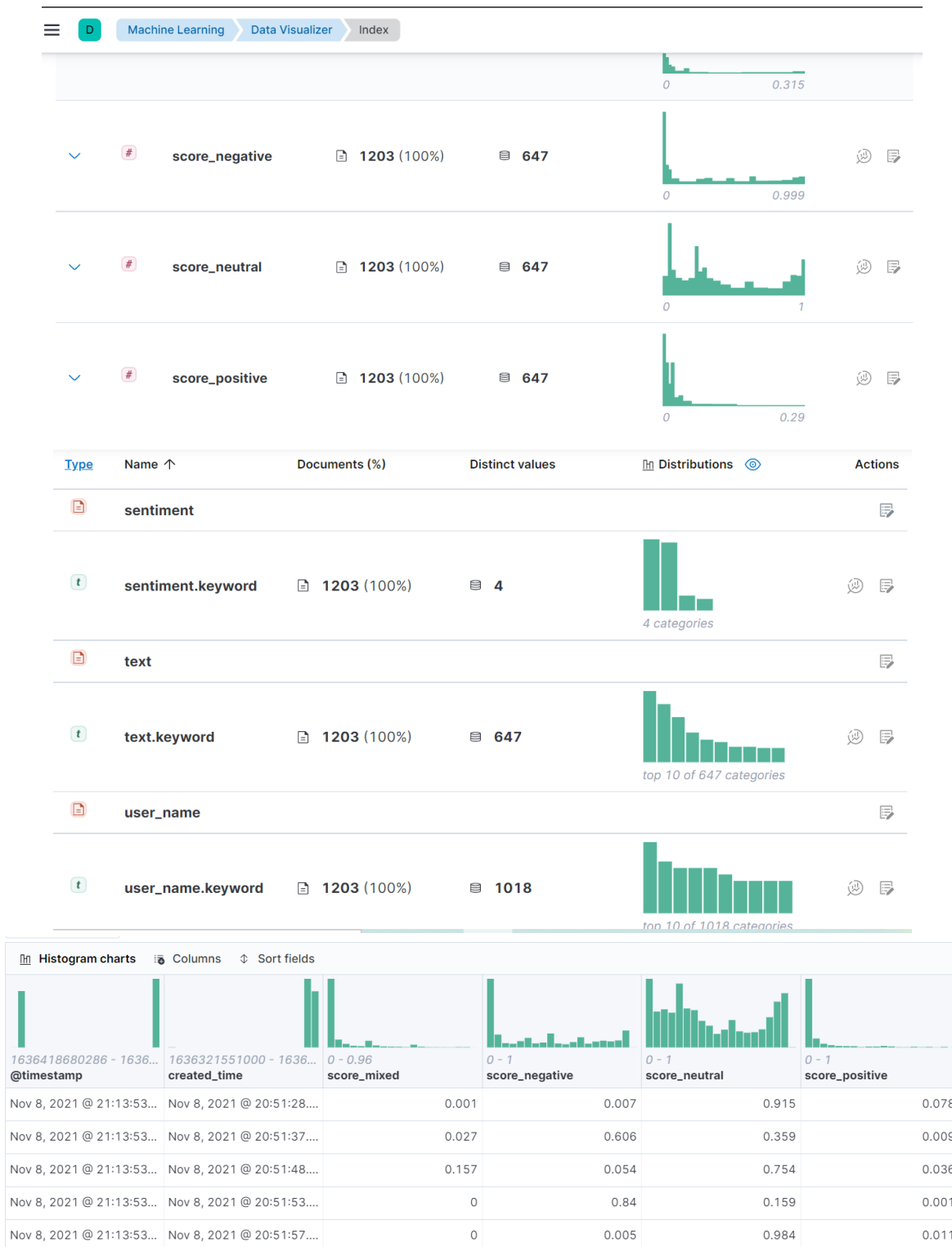




Imagen 19. descriptiva en Kibana

## 11. Visualización de datos

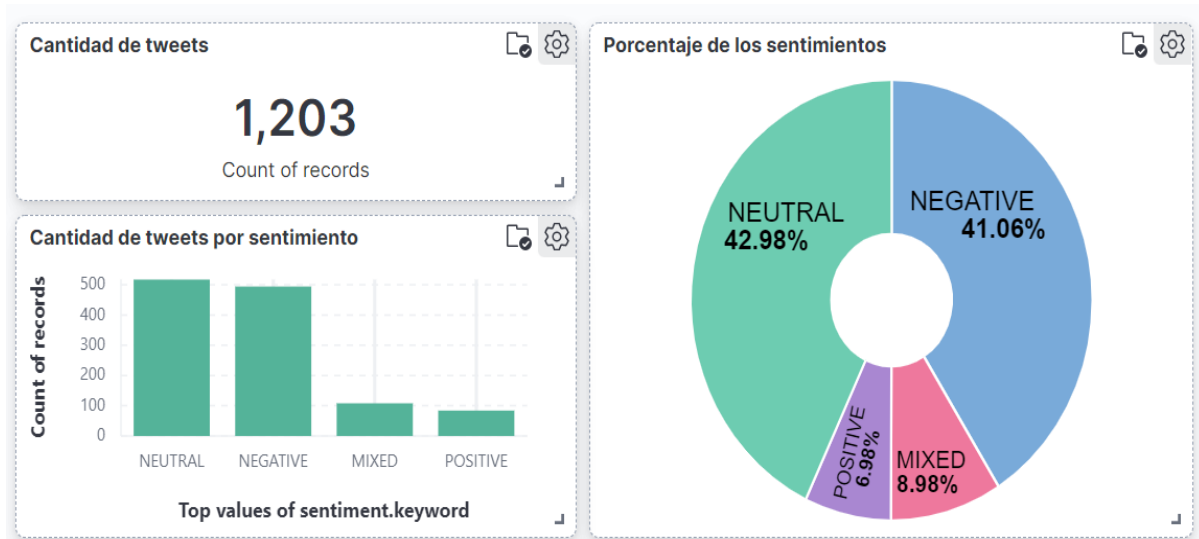


Imagen 20. Dashboard en Kibana

Para el tiempo analizado de 2 horas, se observa que los comentarios sobre Daniel Quintero marcan una posición de 43% neutral seguido por un 41% de comentarios negativos, donde resaltan temas como el mal enfoque de su administración y en un tercer lugar, el solo 7% de los tweets son positivos.

## 12. Modelo de clasificación en kibana

El equipo de trabajo decide indagar las fortalezas de Kibana con los modelos analíticos de clasificación, con la información ya recolectada, se plantea encontrar un modelo que encuentre cual es la probabilidad de que suceda alguna de las 4 categorías encontradas con el modelo de sentimientos; donde se toman los resultados numéricos o scoring obtenidos para hacer el entregamiento, para este entrenamiento se configura el 80% de la información, dejando un 30 % para validación.

## Create job

Source index pattern: twittersentiment\*

1

### Configuration

**Source index**

twittersentiment\*

**Job type**

classification

**Dependent variable**

sentiment.keyword

**Query**

--

**Training percent**

80

**Included fields**

sentiment.keyword, text.keyword, user\_name.keyword

 Edit

*Imagen 20. Configuración modelo en Kibana*

2

Additional options

Advanced configuration

Top feature importance values

0

Prediction field name

sentiment.keyword\_prediction

Model memory limit

21mb

Maximum number of threads

1

Randomized seed

--

Top classes

All classes

Hyperparameters

Lambda

--

Max trees

--

Gamma

--

Eta

--

Feature bag fraction

--

4

Validation

✓ Dependent variable

The dependent variable field contains discrete values suitable for classification.

✓ Training percent

The training percent is high enough to model patterns in the data.

✓ Top classes

Predicted probabilities will be reported for 4 categories.

✓ Analysis fields

The selected analysis fields are at least 70% populated.

✓ Kibana index pattern jobclasificacionsentimiento created.

Progress

Phase 8/8

100%

Data Frame Analytics

Return to the analytics management page.

View Results

View results for the analytics job.

Imagen 21. Configuración modelo en Kibana

- Evaluación del modelo

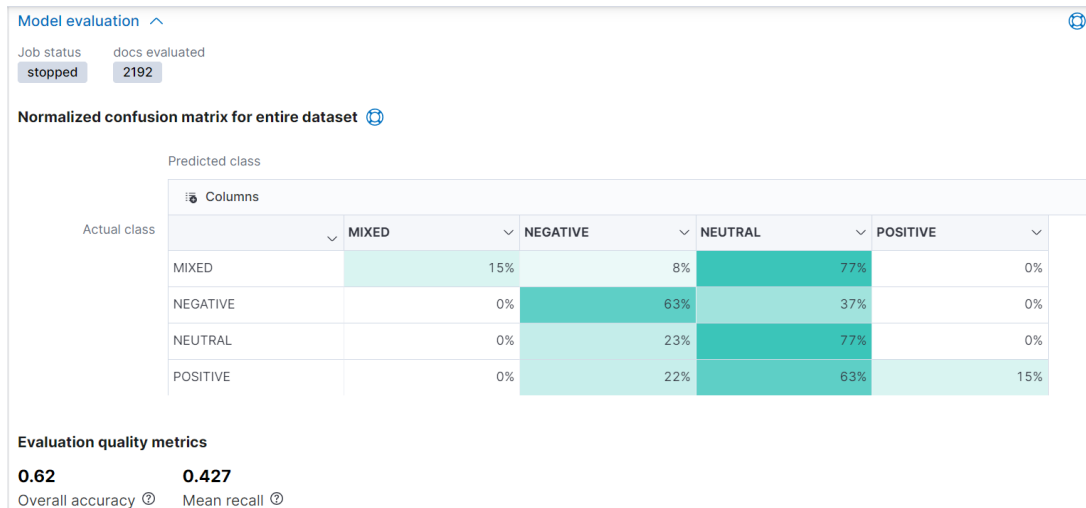


Imagen 22. Evaluación modelo en Kibana

Se verifica que el modelo en la categoría positiva presenta solamente un acierto del 15%, obteniendo unos errores en asignar una categoría de neutral del 63%, lo que significa que el modelo para predecir los tweets positivo no es el recomendable. Para la clasificación Negativa se obtiene una certeza del 63%, y para el neutral del 77% .

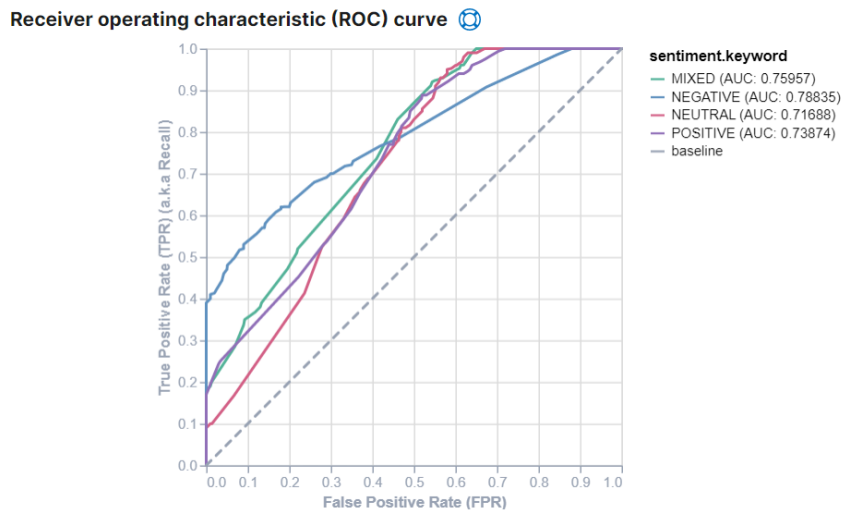


Imagen 23. evaluación modelo en Kibana

Se rectifica que el modelo tiene un buen ajuste para algunas categorías y a nivel global del modelo se tiene una eficiencia del 62%.

## Referencias

- <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=dm-crisp-help-overview>.  
[Último acceso: 6 junio 2021]. IBM. (2021). CRISP-DM Help Overview.
- Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Systematic reviews in sentiment analysis: a tertiary study. In *Artificial Intelligence Review* (Issue February). Springer Netherlands. <https://doi.org/10.1007/s10462-021-09973-3>
- Wyeld, T., Jiranantanagorn, P., Shen, H., Liao, K., & Bednarz, T. (2021). Understanding the effects of real-time sentiment analysis and morale visualisation in backchannel systems: A case study. *International Journal of Human Computer Studies*, 145(August 2020), 102524. <https://doi.org/10.1016/j.ijhcs.2020.102524>