UNIVERSIDAD EAFIT
MAESTRÍA EN CIENCIA DE DATOS Y ANALÍTICA
ST1800 ALMACENAMIENTO Y RECUPERACIÓN DE INFORMACIÓN
TRABAJO FINAL – DOCUMENTO TECNICO

Integrantes:

- Andrés Franco Zapata
- Juliana Ochoa Ramírez
- Alejandro Palacio Vásquez
- Juan Esteban Torres Marulanda

Todos los detalles del trabajo se encuentran en el repositorio Github: https://github.com/franco18/st1800eafit-final

1. Entendimiento de los datos

Se tiene una base de datos con información no estructurada en formato .csv, la cual almacenamos en un Bucket (S3) en la suite de AWS. Posteriormente, una vez tenemos el cluster en AWS corriendo realizamos el procesamiento de los datos a través de pandas. A continuación resumimos la información de la base de datos:

	id	id_news	title	publication	author	date	year	month	url	content
0	0	17283	House Republicans Fret About Winning Their Hea...	New York Times	Carl Hulse	2016-12-31	2016.0	12.0	NaN	WASHINGTON — Congressional Republicans have...
1	1	17284	Rift Between Officers and Residents as Killing...	New York Times	Benjamin Mueller and Al Baker	2017-06-19	2017.0	6.0	NaN	After the bullet shells get counted, the blood...
2	2	17285	Tyrus Wong, 'Bambi' Artist Thwarted by Racial ...	New York Times	Margalit Fox	2017-01-06	2017.0	1.0	NaN	When Walt Disney's "Bambi" opened in 1942, cri...
3	3	17286	Among Deaths in 2016, a Heavy Toll in Pop Musi...	New York Times	William McDonald	2017-04-10	2017.0	4.0	NaN	Death may be the great equalizer, but it isn't...
4	4	17287	Kim Jong-un Says North Korea Is Preparing to T...	New York Times	Choe Sang-Hun	2017-01-02	2017.0	1.0	NaN	SEOUL, South Korea — North Korea's leader, ...

```
df.info()
```

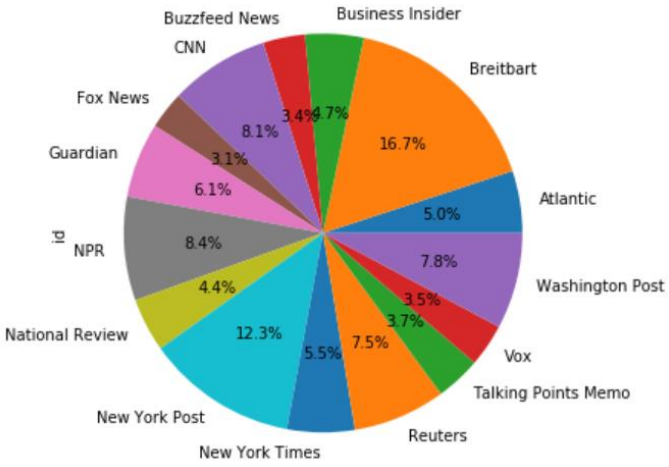
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142570 entries, 0 to 142569
Data columns (total 10 columns):
id              142570 non-null int64
id_news         142570 non-null int64
title           142568 non-null object
publication     142570 non-null object
author          126694 non-null object
date            139929 non-null object
year            139929 non-null float64
month           139929 non-null float64
url             85559 non-null object
content         142570 non-null object
dtypes: float64(2), int64(2), object(6)
memory usage: 10.9+ MB
```

Se tiene una base de datos que contiene una muestra de 142.570 noticias en ingles con la siguiente información por cada noticia:

Id: consecutivo de cada notica que hace las veces de índice.
Id_news: indicativo de la noticia en el respectivo diario.
Title: titular de cada noticia en el respectivo diario.
Publication: Diario que publico la noticia.
Author: Autor que redacto la noticia.
Date: fecha de publicación.
Year: año de publicación
Month: mes de publicación.
url: dirección web donde se publicó la noticia.
Content: texto completo de la noticia publicada.

Son 15 los diarios que aportaron las noticias publicadas:

publication	
Atlantic	7179
Breitbart	23781
Business Insider	6757
Buzzfeed News	4854
CNN	11488
Fox News	4354
Guardian	8681
NPR	11992
National Review	6203
New York Post	17493
New York Times	7803
Reuters	10710
Talking Points Memo	5214
Vox	4947
Washington Post	11114



Se identifican noticias de 15.647 autores.

```
array(['Carl Hulse', 'Benjamin Mueller and Al Baker', 'Margalit Fox',
...,
       'Robert Greenstein', 'Judy Mollen Walters', 'John Yearwood'],
      dtype=object)
```

Luego de realizar la tokenización de los datos y limpieza, observamos que las noticias están relacionadas con el entorno político de las elecciones presidenciales de los Estados Unidos de América, donde los nombres de Donald Trump, Michelle y Barack Obama tienen una alta frecuencia en las noticias e igualmente términos económicos.

El objetivo es enfrentar diferentes retos de negocio con el fin de realizar procesamiento de textos para altos volúmenes de datos.

Para nuestro ejercicio vamos a trabajar para cada noticia con las columnas “title”, “author”, “content”.

2. Especificar los retos o casos de negocio sobre los datos

Realizar indexación (solo en META): Teniendo en cuenta que tenemos un alto volumen de datos utilizamos un framework que nos permita procesar la información de forma eficiente. Para nuestro ejercicio, META proporciona un framework de trabajo desarrollado para el procesamiento de texto de forma eficiente. La primera actividad consiste en la limpieza de los datos o tokenización que nos permita llegar a la indexación de las noticas a través del Bag of Words, inverted index y forward index. A través de META vamos a efectuar el desarrollo de estas actividades permitiendo generar información mas detallada del corpus, es decir de las diferentes noticias, vocabulario, numero de documentos, longitud de los documentos, entre otros.

Realizar búsqueda y recuperación (Solo en META): A través de META pretendemos realizar una herramienta que nos permita efectuar consultas (Query) relacionados con temas políticos con el fin de recuperar los documentos más relevantes o asociados con la consulta a través de un ranking que utiliza como criterio la función BM25. Se construye el inverted index que genera el score para el ranking de noticias. En META vamos a utilizar BM25 con k=1.2 y b=0.75 que son los parámetros más utilizados en el desarrollo de procesamiento de texto y se utilizan en META para el BM25 castigar los documentos y las palabras más frecuentes..

Modelado de tópicos (en SparkML o AWS comprehend o META): A través de META y SparkML buscamos aplicar la modelación de tópicos con Latent Dirichlet Allocation (LDA). Este modelo de aprendizaje automático permite que conjuntos de observaciones puedan ser explicados por grupos no observados, donde la clave del modelo es que no parte del orden de las palabras sino de su probabilidad de ocurrencia, donde al mirar las frecuencias de las palabras en todo el corpus y no individualmente por documentos se encuentra posible tópicos o temas en común para todos los documentos. Con la aplicación de este modelo pretendemos encontrar la probabilidad que un documento sea consistente con el contenido del tópico.

Análisis de sentimientos (en SparkML): Hicimos un análisis de sentimientos de las noticias. A través de Textblob, el cual nos entrega un puntaje de polarización entre [-1 1] se determina si una noticia es positiva o negativa. y posteriormente vamos a implementar el modelo nltk.NaiveBayesClassifier para identificar su comportamiento.

3. Ingeniería de Datos:

Almacenamiento y Clúster de procesamiento en SparkML/Meta/NLTK en Amazon AWS

Se realizaron las siguientes actividades para el proceso de inicio, configuración y uso de AWS:

Creación y activación de cuentas para los integrantes del grupo en AWS Educate.

A. Adecuación de ambiente en nube AWS
   A. Creación y configuración de llaves: Key Pair.
   - EMR-AWS Console: Configuración del Cluster "Francocluster" (ejecución video tutorial para configuración jupyterhub 100%) se efectuaron los siguientes pasos:
   - AWS management console: EC2 dashboard para la creación del cluster desde el cual vamos a trabajar. Launch mode Cluster. S3 folder para base de datos.
   - Configuracion de software: EMR-5.24.0 con aplicaciones por defecto Spark 2.4.0 on Hadoop 2.8.5. YARN with Ganglia 3.7.2 and Zeppelin 0.8.1.
   - Configuracion Hardware: Instance type m3.xlarge con 1 master y 2 core nodes.
   - Acceso y seguridad: Se asignan las llaves y roles por defecto. EMR_defaultRole y EMR_EC2_DefaultRole.

B. Base de datos en S3

   - Se crea el bucket finaltext donde se carga el csv con las noticias y cargamos la información que necesita META para ejecutar como .dat.

C. Instalación jupyterhub en cluster para acceder remoto a los servicios de notebooks (100%).
   - Nos conectamos al servidor remoto - master del EMR-AWS, que corresponde a una instancia EC2.
   - Nos conectamos como administrador y cambiamos la clave al usuario.
   - Instalamos node con referencia dada en github e instalamos las respectivas dependencias necesarias para la ejecución del proyecto.
   - Luego creamos un ambiente virtual (virtualenv) para trabajar en Python y no tener problemas de permisos, ya que desde los notebooks nativos de EMR

tuvimos problemas para acceder al file sytem del cluster. Nuevamente con el usuario hadoop, ingresamos al directorio env, lo que nos permite tener un ambiente propio y no global de la máquina, esto es de gran ayuda para ambientes pre productivos y asi no alterar los componentes del ambiente en producción.

- Por último instalamos el JupyterHub y todas sus dependencias en el ambiente virtual. Todo en el cluster master EMR.
- Iniciamos el JupyterHub, con el fin de poder utilizar la herramienta y poder mostrar los resultados de manera más interactiva. Para esto se ejecuta desde el puerto 8080 para conectarnos todos los integrantes del equipo con el usuario hadoop y password asignado.
- Ingresamos al servidor de jupyter y comenzamos con el desarrollo del proyecto.
- Procedemos con la creación del notebook donde vamos a trabajar los puntos del trabajo final.

Indexación, búsqueda y recuperación con META

- Instalamos y utilizamos en el notebook la librería boto3 para conectarnos a la base de datos S3 (Bucket='finaltext') y descargar el news.csv
- Realizamos análisis descriptivo de las noticas para responder las preguntas 1 y 2 del trabajo relacionadas con entendimiento de los datos, donde se incluyó la construcción del Bag of Words con su term frecuency (ver github notebook AnalisisDescriptivo.ipynb).
- Se crea y parametriza el archivo .toml para META. ('news.toml'). Instalamos Metapy y demas librerias requeridas en jupyter (ver github notebook final.ipynb).
- Con META generamos el inverted_index y generamos la información necesaria para realizar consultas y ranking con BM25 (ver github notebook final.ipynb con información generada).

Modelado de tópicos

- Para el modelo de tópicos en el notebook se utilizó pyspark y las demás librerías de apoyo (ver github notebook Spark-Noteboook.ipynb).
- Para este cargamos la información desde los dataframes de pyspark y cargamos el archivo .csv.
- Realizamos la limpieza de la información utilizando lematización, stemming y removiendo los stop-words.
- Se construye el Bag of Words para generar el diccionario de palabras y la frecuencia de cada una de estas.
- Se ejecuta el modelo de aprendizaje automatico LDA para identificar los tópicos de cada una de las noticias.

4. Desarrollo de los retos o casos de negocio.

Para cada uno de los siguientes puntos referirse a los siguientes links:

1. Análisis descriptivo y análisis de sentimiento: https://github.com/franco18/st1800eafit-final/blob/master/AnalisisDescriptivo.ipynb
2. Búsqueda e indexación: https://github.com/franco18/st1800eafit-final/blob/master/final.ipynb
3. Análisis de Tópicos: https://github.com/franco18/st1800eafit-final/blob/master/Spark-Noteboook.ipynb