Pontificia Universidad Javeriana Cali

Facultad de Ingeniería
Ingeniería de Sistemas y Computación

PROCESAMIENTO DE GRANDES VOLÚMENES DE DATOS

Integrantes:
Ana María García, Andrés Felipe Delgado, Leonardo Sáez, Katherine Camacho

Project 2 (Spark Streaming + Kafka/Flume + MLlib)

Período 2020-2

1. Correcciones Proyecto 1

En el proyecto 1 se presentó un sobre ajuste tanto en la técnica de Decision Tree como en la técnica de Random Forest. Para este proyecto se revisaron ambas técnicas y se encontró que este problema sucedió debido a que la profundidad que se les indicó a los árboles como parámetro era más grande de lo recomendado. Se cambió la profundidad de los árboles teniendo en cuenta el número de atributos del conjunto de datos, que para este caso fue una profundidad de 10.
Ya con estos cambios, los resultados de los modelos utilizados fueron los siguientes:

- Accuracy score de Regresión Logística = 0.5613997879109226
- Accuracy Score de Árbol de Decisión = 0.8598091198303287
- Accuracy Score de Bosque Aleatorio = 0.8914103923647932

Como podemos ver en los resultados, el Random Forest fue la técnica que obtuvo mejor desempeño. Por lo cual este será el modelo escogido para el segundo proyecto.

2. Conjunto de Datos

El conjunto de datos Avila está compuesto por páginas encontradas pertenecientes a la Biblia Avila, que han sido transcritas por distintos autores.

- El conjunto cuenta con 23127 registros, donde cada registro representa una página
- El conjunto cuenta con 10 atributos (F1,F2,F3,F4,F5,F6,F7,F8,F9,F10), que representan las características de la página
- El conjunto cuenta con un atributo clasificador “AuthorNum” el cual representa al autor de la página. Existen 12 posibles autores (0.0,1.0,2.0,...,11.0).

A continuación, se adjunta una tabla de cómo está distribuido el conjunto de datos.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	AuthorNum
0	0.117948	-0.220579	-3.210528	-1.623238	0.261718	-0.349509	0.257927	-0.385979	-0.247731	-0.331310	0.0
1	-0.005490	-0.322644	0.100483	-0.519439	0.261718	-0.307984	0.069175	-0.574114	0.282354	-0.182852	0.0
2	0.105604	-0.087108	0.367214	1.522618	0.261718	1.186911	0.635431	0.574967	0.656532	0.817580	0.0
3	0.241386	-0.126364	0.417003	0.096091	0.172340	-0.515608	-2.724356	1.342012	0.812439	-2.088249	0.0
4	-3.498799	0.203385	0.954020	-2.222565	-4.922215	-1.802878	-3.290612	-1.652310	-2.492797	-2.821900	0.0
...	...	...	...	...	...	...	...	...	...	...	...
23122	-0.005490	0.054213	0.320980	0.804804	0.172340	2.141982	0.559930	0.843424	1.685520	1.758302	4.0
23123	-0.326430	-0.157769	0.424116	1.924954	0.172340	1.602159	-0.346080	0.605799	1.623157	1.853621	4.0
23124	-0.326430	-0.157769	0.424116	1.924954	0.172340	1.602159	-0.346080	0.605799	1.623157	1.853621	4.0
23125	-0.128929	-0.040001	0.057807	0.557894	0.261718	-0.930856	-0.044076	1.158458	2.277968	-0.699884	4.0
23126	-0.128929	-0.040001	0.057807	0.557894	0.261718	-0.930856	-0.044076	1.158458	2.277968	-0.699884	4.0

23127 rows × 11 columns

Figura 1: Conjunto de Datos Avila

CONTEXTUALIZACIÓN DEL PROBLEMA:
Teniendo en cuenta que este conjunto de datos no maneja atributos de fechas o tiempo, se realizan las siguientes suposiciones:

- Inicialmente hay archivadas 16.142 páginas (Lo que equivale al 70 % del conjunto de datos total), donde se conoce con exactitud el autor de cada una de ellas. Por este motivo, se usan para realizar el entrenamiento del modelo Random Forest.
- Posterior a esto, se encontraron en diferentes expediciones varios grupos de páginas nuevas. Conforme se iban encontrando los grupos de páginas, se iban entregando al modelo para su clasificación.

3. Descripción de la Arquitectura de Transmisión

La arquitectura que se propuso para el desarrollo de este proyecto(Figura 2) esta compuesta primeramente por un productor el cual esta encargado de dividir el conjunto de datos entre "Trainz "Test", luego de esto se crea un pipeline con dos

etapas:(1.VectorAssembler,2.RandomForest) y este se entrena con el conjunto de "Train". Despues se procede a guardar el pipeline ya entrenado y el conjunto de datos de test se divide en N particiones, siendo N un numero natural mayor que cero. El componente encargado del proceso de streaming esta conformado por una carpeta(TEST) donde se encuentran los archivos de test previamente creados por el productor, también se hace uso del modulo de spark streaming que nos ayudara a manejar el flujo de baches de información a través del tiempo y por ultimo se usara el pipeline previamente entrenado para la obtención de las predicciones.

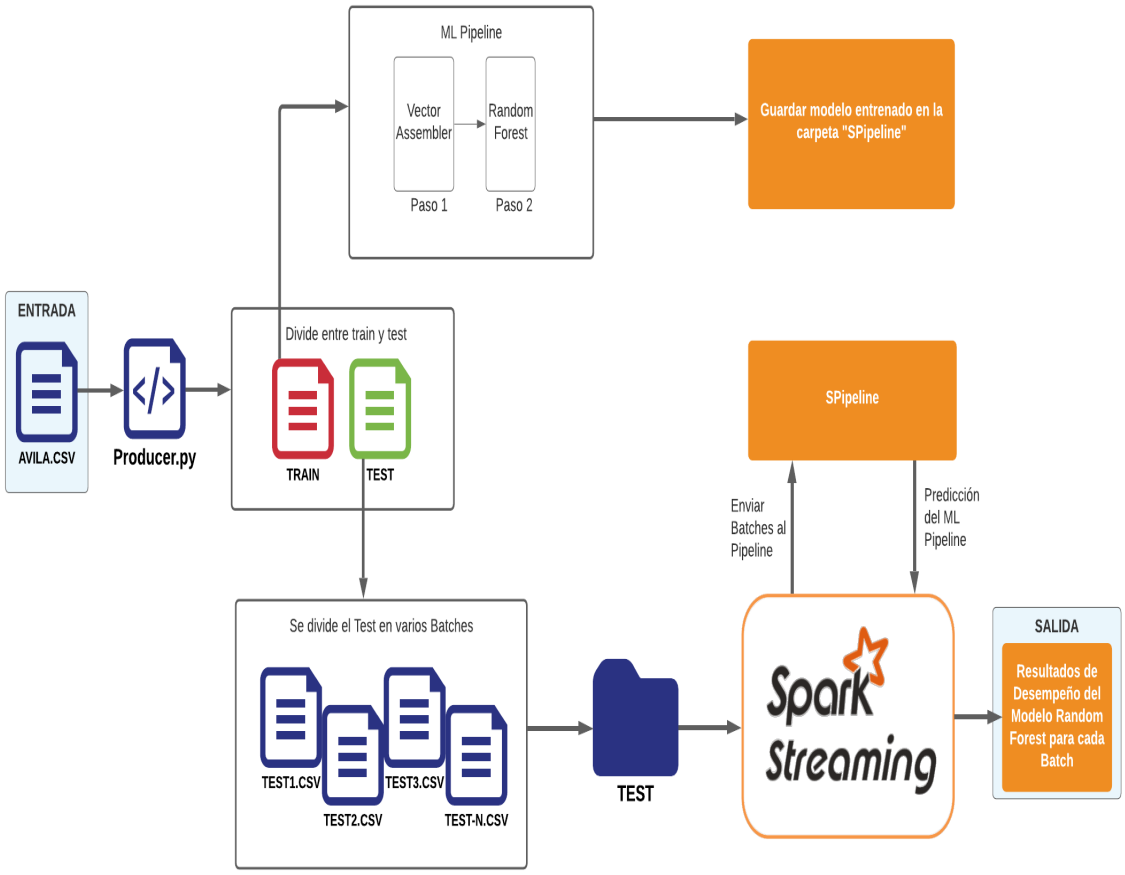

Figura 2: Arquitectura de Transmisión

3.1. Tecnologías del Ecosistema

Las tecnologías que se utilizaron para llevar a cabo este proyecto fueron las siguientes:

- **Spark Streaming + PySpark:** Spark Streaming es una extensión de la API principal de Spark que permite el procesamiento de flujos escalable, de alto rendimiento y tolerante a fallas de flujos de datos en vivo. Los datos se pueden ingerir de muchas fuentes como Kafka, Kinesis o sockets TCP, y se pueden procesar utilizando algoritmos complejos expresados con funciones de alto nivel como mapear, reducir, unir y ventana. Finalmente, los datos procesados se pueden enviar a sistemas de archivos, bases de datos y paneles en vivo. De hecho, puede aplicar los algoritmos de procesamiento de grafos y aprendizaje automático de Spark en flujos de datos.
- **ML Pipeline:** ML Pipelines es una API de alto nivel para MLlib que se encuentra en el paquete "spark.ml". Una tubería consta de una secuencia de etapas. Hay dos tipos básicos de etapas de canalización: transformador y estimador. Un transformador toma un conjunto de datos como entrada y produce un conjunto de datos aumentado como salida. Por ejemplo, un tokenizador es un transformador que transforma un conjunto de datos con texto en un conjunto de datos con palabras tokenizadas. Primero se debe ajustar un estimador en el conjunto de datos de entrada para producir un modelo, que es un transformador que transforma el conjunto de datos de entrada. Por ejemplo en nuestro caso, Random Forest es un estimador que se entrena en un conjunto de datos con etiquetas y características y produce un modelo de Random Forest.

4. Análisis del impacto de la información

El algoritmo se probó con diferentes cantidades de microbatches con el fin de analizar su comportamiento, tal como se muestra a continuación:

```
----WORKING ON BATCH----
.................
# ROWS: 3524 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8867763904653803
----WORKING ON BATCH----
.................
# ROWS: 3508 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.886259977194983
```

Figura 3: Análisis del impacto de la información con 2 microbatches

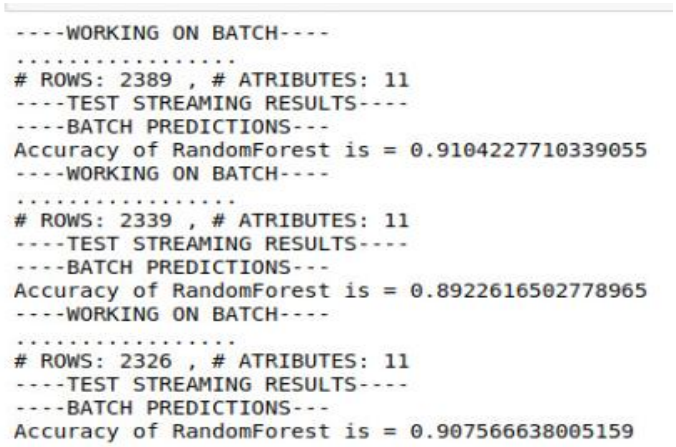

Figura 4: Análisis del impacto de la información con 3 microbatches

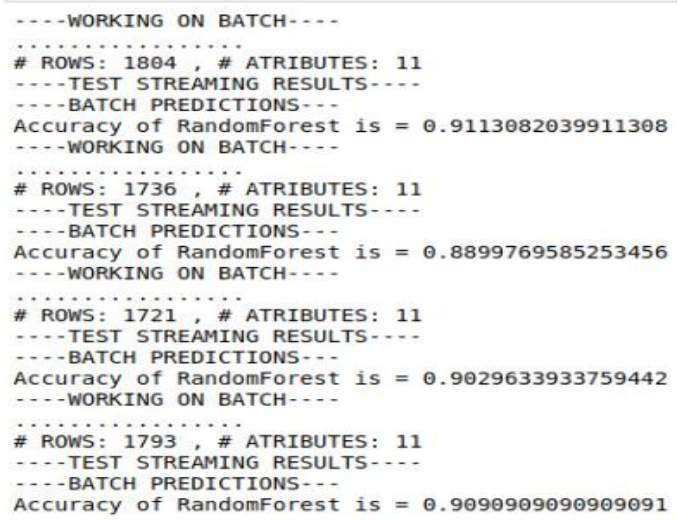

Figura 5: Análisis del impacto de la información con cuatro microbatches

```
----WORKING ON BATCH----
.................
# ROWS: 1428 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.873249299719888
----WORKING ON BATCH----
.................
# ROWS: 1417 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8637967537050106
----WORKING ON BATCH----
.................
# ROWS: 1405 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8597864768683274
----WORKING ON BATCH----
.................
# ROWS: 1374 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.868995633187773
----WORKING ON BATCH----
.................
# ROWS: 1431 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8735150244584207
```

Figura 6: Análisis del impacto de la información con cinco microbatches

```
----WORKING ON BATCH----
.................
# ROWS: 1196 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8795986622073578
----WORKING ON BATCH----
.................
# ROWS: 1190 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8554621848739495
----WORKING ON BATCH----
.................
# ROWS: 1152 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8680555555555556
----WORKING ON BATCH----
.................
# ROWS: 1190 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8579831932773109
----WORKING ON BATCH----
.................
# ROWS: 1157 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.874675885911841
----WORKING ON BATCH----
.................
# ROWS: 1170 , # ATRIBUTES: 11
----TEST STREAMING RESULTS----
----BATCH PREDICTIONS---
Accuracy of RandomForest is = 0.8717948717948718
```

Figura 7: Análisis del impacto de la información con seis microbatches

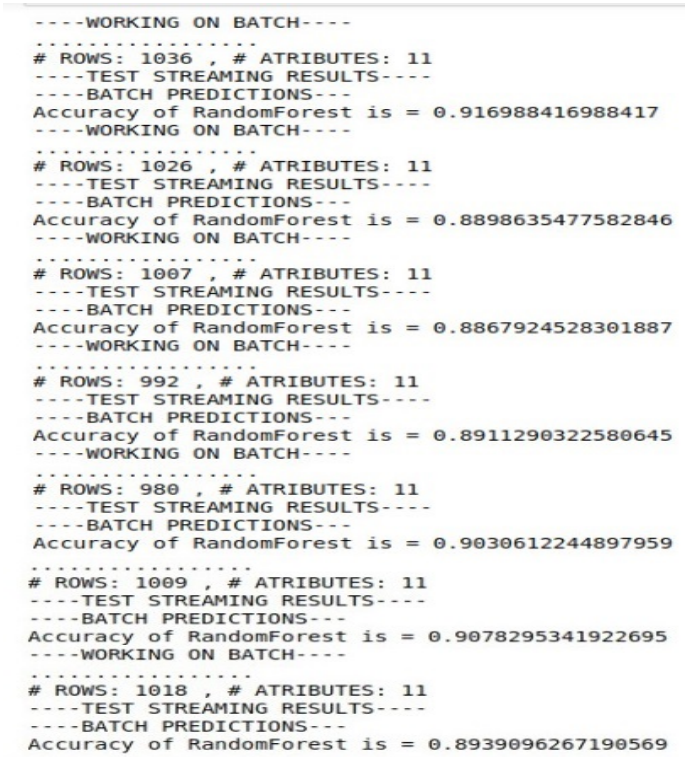

Figura 8: Análisis del impacto de la información con siete microbatches

4.1. ¿Cuánta información se necesita para hacer predicciones con la misma precisión que para el proyecto 1?

Con dos microbatches se obtiene en promedio un accuracy de 0.88645, el cual representa el valor más aproximado al accuracy presentando en el primer proyecto que era de 0.8914.

4.2. ¿Qué tan grande es el tamaño del lote (en el tiempo)?

Para el análisis del impacto de la información con dos microbatches, cada uno de estos, presentó en promedio un total de 3.528 registros. Cada lote se recibió en intévalos de 1 segundo.