Informe Proyecto 2: Grandes Volúmenes de Datos

Juan Manuel Cuellar Borrero
Nicolás Ibagón Rivera
Santiago Uribe Pastás
Pontificia Universidad Javeriana Cali

Octubre 2020

1. Abstract

En el siguiente documento se presenta un informe sobre el streaming de datos para posteriormente realizar predicciones sobre estos batches. Este proyecto se toma como continuación del proyecto 1. Se trabajará con el mismo data set del proyecto pasado el cual es Adult. Este conjunto cuenta con 48842 instancias y 14 atributos, entre estos hay atributos categóricos y numéricos. Su creador es Barry Becker, quien extrajo datos de un censo en EE.UU. en el año 1994. La idea del proyecto es hacer uso de Apache Kakfa para realizar streaming y enviar batches de datos por medio de un **Productor** al clúster de Kafka para que posteriormente un **Consumidor** los reciba y realice predicciones de estos haciendo uso de la tecnica de machine learning Gradient Boost Tree.

2. Arquitectura del Streaming

A continuación se presenta la arquitectura realizada para el streaming de datos y se hablará acerca de sus componentes y el trabajo que realizan en dicha arquitectura.

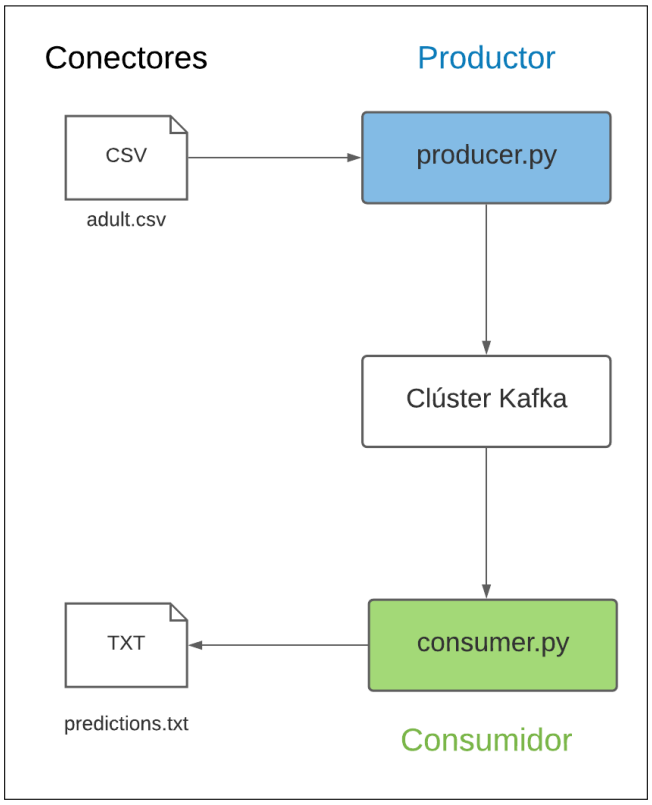

Figura 1: Arquitectura del Streaming

2.1. Clúster de Kafka

El clúster de Kafka generalmente consta de varios brokers para mantener el equilibrio de carga. Los brokers de Kafka no tienen estado, por lo que utilizan ZooKeeper para mantener su estado del clúster. Las aplicaciones se conectan a este sistema y transfieren un registro (mensaje) al topic. Los datos enviados se almacenan en este clúster hasta que transcurre un período de retención especificado. En la figura 2 se puede observar la estructura interna del clúster.

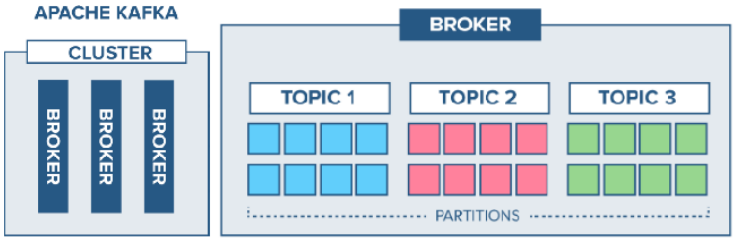

Figura 2: Arquitectura interna del clúster de Kafka

2.2. Productor

Un productor se define como una aplicación la cual es la fuente del flujo de datos. Genera mensajes y los publica en uno o más topics en el cluster de Kafka. Este fue implementado con la libreria `confluent kafka` de Python con el modulo `Producer`. En este caso el productor se encargará de entregar los mensajes al cluster de kafka de manera asíncrona con la instrucción `produce` pasando el valor del mensaje y el topic respectivo, el cual sera **`test`**. Antes de cerrar el productor se hace uso de la instrucción `flush` para garantizar que se entreguen todos los mensajes pendientes.

2.3. Consumidor

Un consumidor se define como una aplicacion la cual se suscribe al clúster de Kafka para leer y procesar estos mensajes. Los consumidores pueden unirse al clúster en cualquier momento, por lo que se maneja una comunicación asíncrona. El consumidor emite una solicitud de extracción al broker para tener un búfer de bytes listo para consumir. Este al igual que el productor fue implementado con la libreria `confluent kafka` con el modulo `Consumer`. El consumidor esta suscrito al topic **`test`**, posteriormente se centra en un bucle de consumo, que llama repetidamente al método `poll` para recuperar mensajes uno por uno. Ya recibido el mensaje este se parsea y se ingresa en el batch, el cual tiene una cantidad mínima de datos para ser enviado al procesamiento. Una vez esta condición se cumple el batch es enviado para realizar las respectivas predicciones sobre este usando la técnica Gradient Boost Tree.

2.4. Conectores

Los conectores son los componentes de Kafka los cuales se pueden configurar para escuchar los cambios que se producen en una fuente de datos como un archivo o una base de datos, y extraer esos cambios automáticamente. En este caso utilizaremos dos conectores, el primero tendrá la funcionalidad de enviar los datos respectivos del archivo CSV al productor, y el segundo tendrá la funcionalidad de enviar el resultado de las predicciones a un archivo de texto.

3. Análisis del impacto de la información

Para hacer un análisis completo respecto a la información obtenida es importante recordar los resultados obtenidos en el primer proyecto, estos se encuentran plasmados en la Tabla 1.

Accuracy	Precision	Recall	F1
0,8558	0,9396	0,8773	0,9074

Tabla 1: Métricas obtenidas en el primer proyecto con GBT

Debido a que el data set utilizado no tiene atributos de fechas para tomar batches de tiempo, el acercamiento que se uso en este proyecto fue que se tomaron 7 batches con un tamaño lo mas parecido posible (teniendo 6 batches de tamaño 7.000 y 1 batch de tamaño 6.842) y se simulo que este ultimo batch es el batch del "futuro" al cual se le quieren hacer las predicciones pertinentes. Por ende cada batch del 1 al 6 se usa como entrenamiento del modelo y con el batch 7 se hace el test. En la Tabla 2 se ven los promedios de 3 simulaciones distintas.

Batch	Accuracy	Precision	Recall	F1
Batch 1	0,8128	0,9242	0,8427	0,8815
Batch 2	0,8302	0,9254	0,8599	0,8915
Batch 3	0,8223	0,9359	0,845	0,8881
Batch 4	0,8353	0,9387	0,8565	0,8957
Batch 5	0,838	0,9177	0,8737	0,8951
Batch 6	0,8244	0,9072	0,8661	0,8862

Tabla 2: Promedio de métricas para cada batch después de 3 simulaciones distintas

En la tabla 3 se muestra las medidas estadísticas como el promedio y la desviación estándar de los datos obtenidos en la tabla 2.

	Accuracy	Precision	Recall	F1
Promedio	0,8271	0,9248	0,8573	0,8896
Desviación Estándar	0,0092	0,0116	0,0119	0,0054

Tabla 3: Promedio de las medidas estadísticas

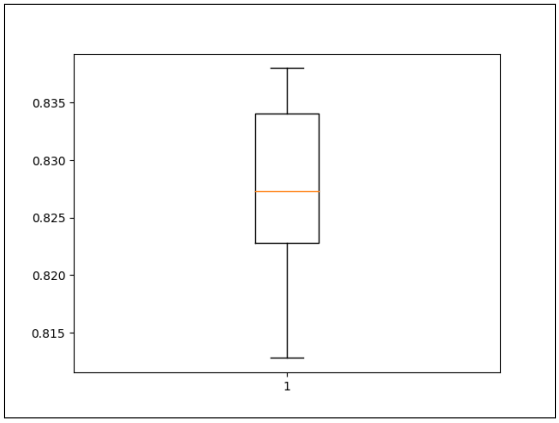

Figura 3: Boxplot del Accuracy de los datos

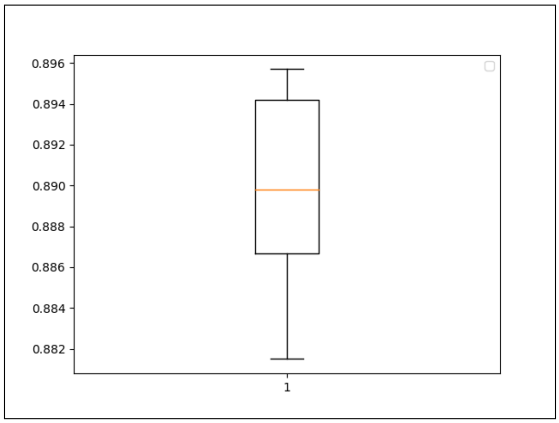

Figura 4: Boxplot del F1 score de los datos

En primer lugar se puede apreciar que el F1 Score en general tiene una media más alta que que el accuracy, además de esto la dispersión de los valores en ambas medidas es bastante similar, siendo la del F1 Score ligeramente mayor, por ultimo no se encontraron valores atípicos en las simulaciones, lo cual indica que ningún batch tuvo un comportamiento por fuera de los esperado en cuanto a su entrenamiento.

	Accuracy	F1
Error Medio	0.028633	0.017716
Desviación Estándar	0.008470	0.005007

Tabla 4: Error medio y análisis

En la tabla anterior se puede ver que el error medio que se tuvo con respecto a los resultados obtenidos en el primer proyecto es mínimo, lo que indica que el realizar el machine learning sin tener que esperar por todos los datos es viable, esto hace que en una situación real sea sumamente provechoso el uso de streaming pues realizar análisis o predicciones con una cantidad de datos menor abre la oportunidad de adaptarse a cambios de forma mas rápida y eficiente.

Referencias

[1] "Confluent-Kafka 1.4.0 Documentation", docs.confluent.io. [Online]. Available: `https://docs.confluent.io/current/clients/confluent-kafka-python/`.

[2] L. Johansson, "Apache Kafka for beginners - What is Apache Kafka?", cloudkarafka.com, 2019. [Online]. Available: `https://www.cloudkarafka.com/blog/2016-11-30-part1-kafka-for-beginners-what-is-apache-kafka.html`.