Real-Time Streaming Feature Generation

CS4991 Capstone Report, 2024

Param Damle
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
psd9vgc@virginia.edu

ABSTRACT

As data-driven marketing provides better personalization for online ads, companies pursue transformation pipelines to efficiently convert user data into marketing predictions. A distributed streaming pipeline not only provides robust data processing, but its real-time operation allows the company to market ads to a user before they leave the website. To this end, I contributed to the development of a streaming pipeline running on Apache Flink to process user data across nodes in AWS EMR, transform raw information into feature sets, and output these features to a model (or stand-in database) for marketing prediction. This improved the system latency from 24 hours to sub-millisecond orders of magnitude. Future work can provide additional efficiency by taking advantage of pipeline-model synergy and deploying hardware accelerators for feature extraction.

1. INTRODUCTION

Skip ad. Click X. Clear your cookies. At times, it may feel impossible to browse the internet without encountering some form of digital advertising, whereby websites leverage user traffic and data to serve ads in the hopes of converting visitors into customers. As these marketers seek effective (and cost-efficient) advertisement decision making, they increasingly turn to machine learning (ML) models that use data to adjust marketing strategies and boost profits. Using ML in data-driven marketing is becoming crucial to advertising in the digital space, but advertisers often find it difficult to aggregate and feed the user data necessary for this (Abakouy, et al., 2019).

This focus on the data makes sense—the largest predictor of ML performance tends to not be the model's hyperparameters, rather the quality of the training set and the features derived from it (Wang & Shah, 2021). This has created a "data imperative" within the digital space to maximize information collection, whereby algorithms can craft the most specific profile for each user (Seaver, 2021). Specificity provides more personalized recommendations, leading to a uniquely memorable customer experience, which in turn produces more loyal sources of revenue for a company (Vas, 2021).

However, this calculation provides no utility if the user leaves the website before being served a personalized ad, which incentivizes companies to make such decisions in real-time. In a Neolane/DMA study, 69% of marketers reported real-time web marketing and dynamic content as "highly important" to creating cohesive customer experiences (Ad Tech Daily, 2013). The same study noted that major obstacles to this vision include system complexity, access to real-time data, and privacy issues; the first two are addressed by the solution in this paper, while the third is analyzed in Damle (2024).

2. RELATED WORKS

The deployment of recommendation systems requires a synergy between the algorithm and the real-time data processing platform. Prior work in feature-based recommendation showed that customer information collection, profiling, comparison, aggregation, and measurement could achieve improvements in one-to-one marketing personalization. Specifically, algorithms that factored features extracted from user data into predictions achieved a precision score of 61% in recommending films to potential viewers, compared to the 31% precision of the control group (Weng & Liu, 2004).

To process the data necessary for these systems, Xu et al. (2022) explore platforms designed to handle the volume, variety, and velocity of information, namely Apache Flink. The researchers captured data from Bilibili and fed it in real-time through a stream processing app to demonstrate that Flink could consume up to 12,000 tokens per second for sentiment analysis.

This work extends and integrates both concepts by exploring a real-time application that extracts features from user web data that will be useful in marketing prediction.

3. PROJECT DESIGN

This section reviews the developmental elements of the project.

3.1 Motivation

To generate real-time marketing revenue from users visiting its website, Capital One Financial Corporation sought to integrate feature extraction into its data pipeline. The development of the entire pipeline was distributed across front-end, ML engineering, and data science teams. The front-end team owned a preexisting real-time data collection framework, and it was assumed that the data science team would create a marketing prediction model with a cross-team accessible endpoint. Thus, the ML engineering team was tasked with preparing the data streamed from the website to be inputted into a future model. As this project was sufficiently short in nature (scoped for two months) and not intended for immediate deployment, the managers on the Card Tech Machine Learning team selected it as the summer assignment to the Firebeetle intern pod. After joining Capital One's Technology Internship Program in June 2022, I was placed on the Firebeetle pod (and by extension, this project) due to my affinity for machine learning systems.

3.2 Objectives

The resultant data pipeline needed to satisfy multiple design parameters to address the underlying problem:

- The pipeline must integrate with and channel information between the front-end data streaming source and the back-end prediction model. While the model is not deployed, the output of the pipeline can be pushed to a database as a stand-in.
- The pipeline should extract informative features from the streamed data in real-time, allowing a marketing prediction to be generated and realized on the front end before the user leaves the website.
- The system should exist on a cloud computing cluster with guarantees for uptime and safety mechanisms in the event of node failure.

In the following section, I describe our solution to the task presented above and how the Firebeetle intern pod satisfied these completion criteria.

3.3 System Design

3.3.1 Prior Setup

The Capital One website already included a user data collection component that captured the clickstream (cursor movement), web cookies indicating user preferences across all websites, and information about the user's device and network. Every night, a model transformed this data into features to input into a model, which predicted the optimal credit cards to advertise to the user the next time they visited the website. This presented an issue of staleness, measured by how much time passes between the collection of data and the input of relevant features into the algorithm (Talati, et al., 2023).

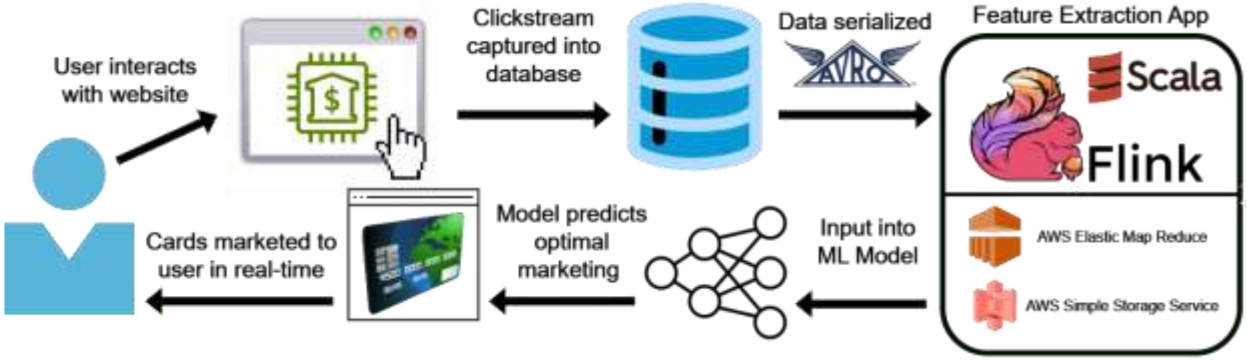
3.3.2 New Setup

The Firebeetle team replaced the previous "batch-based" processing with streaming feature generation, which Iguazio (2023) regarded as the contemporary "Holy Grail" of data science because of its ability to adapt to changes in user behavior in real-time. With this setup, the backend system could calculate the best credit card offers to market to a user as they browse the website and display the optimal ads relating to these offers before the user even leaves the website.

Figure 1 depicts the full data pipeline, with our deliverable highlighted in the black box to the right. This deliverable was an application running on the Apache Flink framework with constant uptime, processing and outputting data 24/7. This application connected (via internal company credentials) to the front-end team's streaming endpoint, which served live clickstream data serialized in Avro format. As opposed to sending the data as string characters, which are vastly inefficient in the bits required to encode information, Avro optimizes for compression by transmitting the raw bytes associated with the data in a standardized format. This platform-agnosticism makes Avro the optimal choice of vessel in which data is transmitted between applications like those in our setup.

On the other side of the pipeline, the app outputted to a DynamoDB endpoint, which recorded receipt of the features and represented the rate at which data would be inputted into a future prediction model. This entire Flink app ran on an AWS EMR cluster. Amazon Web Services (AWS) leverages the vast technological resources of Amazon to provide businesses like Capital One reliable and scalable access to a suite of cloud computing and database storage solutions. Of these, the Elastic Map Reduce (EMR) cluster consists of multiple *nodes*, each of which is an instance of a self-contained virtual computer running on a server rack in AWS's us-east-1 data center in northern Virginia. The main node in the EMR cluster receives a job and distributes its workload across several subsidiary nodes to optimize for throughput and latency. This process is optimized for highly *parallel* processes, like our use case where each incoming data point in our high-throughput channel underwent the same procedure for feature extraction.

Figure 1: **Streaming feature generation pipeline diagram**. Ideal solution cyclically presents a marketing prediction to the user whose data was profiled. Implemented solution outputted to a DynamoDB endpoint after the feature extraction step as a proxy for future model deployment.


User interacts with website
Clickstream captured into database
Data serialized
AVRO
Feature Extraction App
Scala
Flink
AWS Elastic Map Reduce
AWS Simple Storage Service
Input into ML Model
Model predicts optimal marketing
Cards marketed to user in real-time

These distributed sources were further used to store reboot credentials for nodes that crashed and rejoined the cluster. However, this only applied to the peripheral nodes in the EMR cluster as we had insufficient time in the internship to implement such contingency procedures for the main node.

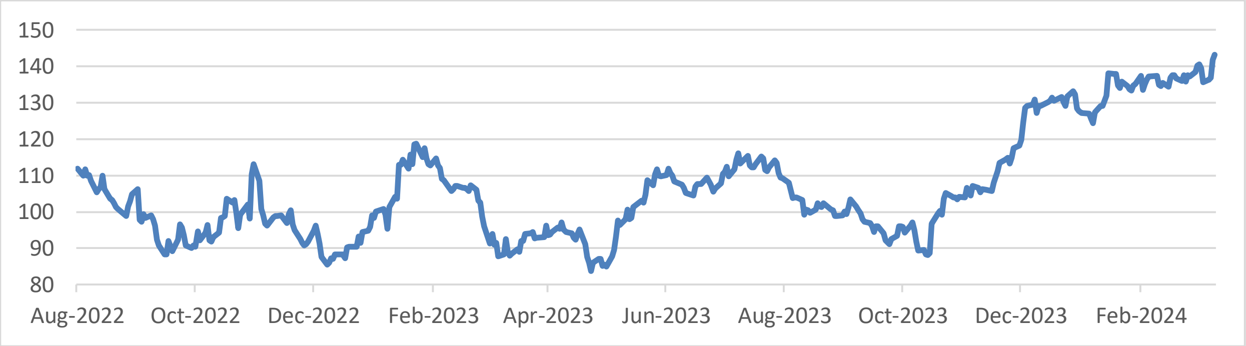
While each of these nodes had local memory to store the partial intermediate products of the data point it was currently processing, information crucial to the entire cluster (e.g. authentication credentials to connect to endpoints on either side) needed to be shared between nodes to minimize redundancy. This was achieved through credentials stored on long-term Simple Storage Services (S3, an AWS cloud storage container that integrates seamlessly with EMR) that copied over to Hadoop Distributed Filesystem (HDFS, the directory shared by all nodes in a cluster) once the cluster launched.

4. RESULTS

The success of projects in the financial technology (FinTech) sector is usually evaluated both directly on technical metrics (e.g. latency, throughput) and indirectly using changes in sales, a standard measure for the business value of a recommender system (Jannach & Zanker, 2022). As the data science team had not completed the prediction model by the end of my internship, I have no guarantee that our feature extraction pipeline was ever deployed in production. Regardless, as Figure 2 shows, public-facing information about business performance demonstrates no drastic changes in the 2 years following the end of the internship, as the stock price hovered between $90 and $120 per share until 2024 (Yahoo Finance, 2024). Thus, effectively measuring the business impacts of this pipeline would require attribution of internal marketing numbers to its deployment, perhaps by leveraging A/B testing (Nicholson, 2024). Otherwise, the technology was assessed by a panel of upper-level managers in the Card Tech Machine Learning division as a proof-of-concept for how the company could transition towards real-time stream processing across its many information systems.

After integration with either endpoint, the overall latency of the application from stream ingestion to deposition in DynamoDB averaged 100ms, a major improvement from the nightly 24-hour latency of the previous batch-based setup. Should the pipeline be fully actualized, the logic of this feature extraction app alone would be insufficient to meet the latency requirements of the full system. Careful construction of the remainder of the data pipeline is needed to ensure the turnaround time of the *entire* marketing process (as in Figure 1) is under a second.

Figure 2: **Capital One Stock Adjusted Closing Price, August 2022 – March 2024**. Historical daily stock price data was downloaded from Yahoo Finance. Price at period start (08/12/2022) was $111.95 per share and price at period end (03/21/2024) was $143.18 per share.



5. CONCLUSION

The feature extraction pipeline built in this work allowed information collected from the Capital One website to be processed in real-time and pre-processed as an input to a marketing prediction model. Such a model would leverage this information to market a credit card or other relevant financial products to the user before they even leave the website. Key results include successful deployment of the application on AWS cloud compute, integration with other software components, and reduction of data processing latency from 24 hours to 100ms.

My primary takeaways from this project included hands-on experience with the agile software development lifecycle, navigating a large repository of components, and building an industry-scale system with real-world uptime and failure-resistant design. Although it is unlikely the application was directly deployed into production, its successful connection of the necessary endpoints demonstrated a proof-of-concept for real-time feature extraction. As a result of this technology, customers visiting the website will benefit from the live feedback in the form of relevant financial offers.

6. FUTURE WORK

Since the data collection component is sufficiently robust and operational, future deployment of a feature extraction pipeline depends on a fully-furnished machine learning endpoint that can receive feature input and produce a marketing output that is fed into the website's front end. Iterations on this design could experiment with alternative data serialization schemes, such as Google's Protocol Buffers (Currier, 2022), as well as different recommendation model setups, which would affect the feature set extracted in the pipeline. Overall, real-time data processing and inference present ample opportunities to leverage machine learning to solve a problem—regardless of the problem domain—within milliseconds of the relevant problem information being collected.

7. ACKNOWLEDGMENTS

This work would not be possible without the team of software engineers and interns in the Firebeetle pod at Capital One. My co-interns David Wolfers (Cornell University, 2023), Siddhant Pandit (University of Texas Austin, 2024), and Frederick Sion (Pennsylvania State University, 2023) were instrumental in the development, deployment, and testing of the real-time feature extraction pipeline. Last, the unending support and leadership of John Thompson, Marco Yuen and Shrividhya Baikadi enabled us to work efficiently and reach all our milestones.

REFERENCES

Abakouy, R., En-naimi, E. M., El Haddadi, A., & Lotfi, E. (2019, October 2). Data-driven marketing: How machine learning will improve decision-making for marketers. Proceedings of the 4th International Conference on Smart City Applications (SCA '19). https://doi.org/10.1145/3368756.3369024

Capital One Financial Corporation (COF) Historical Prices & Data. (2024). [dataset]. Yahoo Finance. https://finance.yahoo.com/quote/COF/history?period1=1660262400&period2=1711073913

Currier, C. (2022). Protocol Buffers. In C. Hummert & D. Pawlaszczyk (Eds.), Mobile Forensics – The File Format Handbook: Common File Formats and File Systems Used in Mobile Devices (pp. 223–260). Springer International Publishing. https://doi.org/10.1007/978-3-030-98467-0_9

Damle, P. (2024). Responsible Research and Innovation of Recommendation Systems. Department of Computer Science, School of Engineering and Applied Science, University of Virginia.

Iguazio. (2023, February 7). How to Build Real-Time Feature Engineering with a Feature Store. AI Infrastructure Alliance. https://ai-infrastructure.org/how-to-build-real-time-feature-engineering-with-a-feature-store/

Jannach, D., & Zanker, M. (2022). Value and Impact of Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. Kantor, Recommender Systems Handbook (3rd ed., pp. 519–546). Springer. https://web-ainf.aau.at/pub/jannach/files/BookChapter_RS_Handbook_Value_Impact.pdf

Neolane and DMA Study Reveals Digital Channels are Crucial for Real-Time Marketing. (2013, August 1). Ad Tech Daily. https://adtechdaily.com/2013/08/01/neolane-and-dma-study-reveals-digital-channels-are-crucial-for-real-time-marketing/

Nicholson, R. (2024, March 19). How to Do A/B Testing: 15 Steps for the Perfect Split Test. HubSpot. https://blog.hubspot.com/marketing/how-to-do-a-b-testing

Seaver, N. (2021). Seeing like an infrastructure: Avidity and difference in algorithmic recommendation. Cultural Studies, 35(4–5), 771–791. https://doi.org/10.1080/09502386.2021.1895248

Talati, A., Parkhe, M., & Lukiyanov, M. (2023, February 16). Best Practices for Realtime Feature Computation on Databricks. Databricks. https://www.databricks.com/blog/2023/02/16/best-practices-realtime-feature-computation-databricks.html

Vas, G. (2021, September 21). How Recommendation Systems Comply with Privacy Regulations. Gravity Research & Development. https://www.yusp.com/blog-posts/recommendation-systems-comply-with-privacy-regulations/

Wang, A., & Shah, K. (2021, March 4). Building Riviera: A Declarative Real-Time Feature Engineering Framework. DoorDash Engineering. https://doordash.engineering/2021/03/04/building-a-declarative-real-time-feature-engineering-framework/

Weng, S.-S., & Liu, M.-J. (2004). Feature-based recommendations for one-to-one marketing. Expert Systems with Applications, 26(4), 493–508. https://doi.org/10.1016/j.eswa.2003.10.008

Xu, B., Jiang, J., & Ye, J. (2022). Information Intelligence System Solution Based on Big Data Flink Technology. 21–26. https://doi.org/10.1145/3538950.3538954