Streamlined Website Insights: Leveraging Kafka, Redis, MongoDB, Flink, Elasticsearch, and Kibana

Vidushi Bhati
Master's in Data Analytics
San Jose State University
San Jose, USA
vidushi.bhati@sjsu.edu

Kamakshi Lahoti
Master's in Data Analytics
San Jose State University
San Jose, USA
kamakshi.lahoti@sjsu.edu

Akanksha Paspuleti
Master's in Data Analytics
San Jose State University
San Jose, USA
akanksha.paspuleti@sjsu.edu

Yesheswini Lakshmi Spandana Potti
Master's in Data Analytics
San Jose State University
San Jose, USA
yesheswinilakshmispandana.potti@sjsu.edu

Abstract*—The landscape of e-commerce analytics is rapidly evolving with the advent of real-time data processing and advanced machine learning techniques. A critical challenge in this domain is the effective integration of real-time analytics with deep offline analysis to enhance user experience and operational efficiency. Prior works have predominantly focused on either offline batch processing for user behavior analysis or real-time monitoring without deeper predictive insights. Our work bridges this gap by introducing a hybrid analytical framework that leverages both real-time stream processing and comprehensive offline machine learning models while employing cutting edge technologies. Our methodology involves generating synthetic datasets representing a hypothetical furniture website's users, sessions, and products. This data is streamed to Apache Kafka and then utilized by Redis for immediate security checks while computing unique user metrics and session counts. The data is then ingested into MongoDB for offline processing, where we implement Locality Sensitive Hashing (LSH) to identify similar users and deploy machine learning models like Random Forest, XGBoost, and Artificial Neural Networks. These models are augmented with differential privacy to safeguard user information and utilize Explainable AI technique, SHAP, for model interpretability. Concurrently, the real-time component of our pipeline leverages Apache Flink's capabilities to process the data further, which is then sent to Elasticsearch known for its powerful full-text search and analytics engine. Subsequently, Kibana is employed for its robust visualization features, enabling stakeholders to gain insights into user behavior, product performance, and session activity dynamically.This holistic approach ensures a fine balance between real-time responsiveness and the strategic depth of offline analysis. Users benefit from personalized experiences and improved service, while businesses gain a dual advantage of immediate intelligence and long-term strategic insights, paving the way for enhanced data-driven decision-making in e-commerce platforms.

I. Introduction

In the digital world, where things are always changing, a website's success depends on more than just design or content – how well it can engage users is also key. It has become important for e-commerce administrators and marketers to understand and optimize customer interactions with their websites because this is what matters most in this day and age of internet marketing. Every click, every scroll, and every purchase counts for something within this field; hence there is a need to realize that even the slightest changes can significantly affect results for better or worse. To achieve higher levels of performance enhancement in terms of functionality improvement towards securing systems designed around user friendliness; we must involve ourselves in real-time data analysis using advanced techniques alongside other applicable approaches.

We want to know more about web traffic by getting insights about it in real-time. Knowing how people behave when they visit our pages; tracking all those visits made throughout different times of day (or night), week/month/year plus identifying patterns followed by those who come to the website. In realizing these needs we have developed a new system that performs instant checks on various aspects related to traffic occurring along web pages.

Our main objective is to analyze the ten most popular products available on a hypothetical furniture website. By doing so we hope to gain a deeper understanding of what makes users tick, their preferences as well as current trends they may be following. This particularized method allows us to focus only on influential parts thus giving room for informed decision making while at the same time helping us plan strategically around them.

Apache Kafka and Apache Flink are among the leading technologies used within central areas of architecture for our system. For instance through the adoption of Apache Kafka, one can easily set up an extensible foundation that would support receipt logs from many servers without having issues related to data flow into analytical tools. In addition, Apache Flink provides the capability of real-time data stream processing thereby giving immediate feedback about how people interact with the site at any given moment. Redis cache DB is employed for initial security checks for new users in

the system. MongoDB is used for offline processing where machine learning models are trained to predict the potential customers.

This project is an example that illustrates how transformative the hybrid approach of real-time analysis with offline analysis can be in e-commerce. Administrators get live information on what visitors are doing and gain knowledge about customer behavior and analysts can gather insights from the offline techniques that will help them optimize websites and contents and make strategic decisions to acquire potential customers bringing in more profits to the company.

II. SIGNIFICANCE TO THE REAL WORLD

This project represents a significant leap in e-commerce analytics by synergizing immediate data stream processing with thorough offline analysis, catering to the emerging needs of a dynamic online retail environment. It elevates customer interaction by offering tailored content and improves operational workflows by swiftly identifying and resolving system issues. The security aspect is robustly addressed with real-time anomaly detection, providing a secure shopping experience. Strategically, the fusion of sophisticated machine learning algorithms and privacy-centric models furnishes deep insights into consumer behavior while safeguarding user anonymity, establishing a bedrock of trust. The incorporation of Explainable AI principles demystifies complex model decisions, ensuring transparency and aiding compliance with regulatory standards. The implementation of real-time visual analytics through Elasticsearch and Kibana not only empowers stakeholders with immediate clarity on actionable metrics but also equips them for rapid decision-making. Designed for scalability, the architecture promises adaptability to expanding data challenges, positioning e-commerce entities at a vantage point in a highly competitive market. As a beacon for both commercial strategy and academic inquiry, the project charts a course for future innovations in applying data science to enhance digital commerce ecosystems

III. Project Motivation

The digital world is changing quickly, and websites and other online platforms are becoming essential to social interactions, educational initiatives, and corporate operations. Real-time comprehension of user behavior is not only advantageous in this setting, but also necessary to maintain relevance and competitiveness. This project is driven by the need to leverage big data and real-time analytics to make decisions that optimize resource allocation, improve user experience, and improve content delivery. Organizations may learn about user preferences, identify new trends, and quickly address user demands by real-time analysis of clicks and user interactions. This project has multiple facets that impact different facets of marketing, strategy, and operations. Real-time user interaction data allows for the improvement of interface design, content, and navigation on a website, increasing user satisfaction and retention. Real-time click and user behavior analysis facilitates better engagement rates, higher conversion rates, and content personalization based on user preferences. Resources can be effectively allocated to website sections that elicit the greatest interest by identifying popular pages and features, hence enhancing user satisfaction and performance. Enhancing security, real-time monitoring can assist in identifying anomalous trends that might point to fraudulent behavior. Sentiments about the product can be gained in real-time. The main parties who stand to gain directly from improved user insights that promote engagement and conversion are website owners and managers. Marketing experts may use the data to improve their campaigns' targeting, hone their methods, and track the results of their work in real time. By gaining insights into the content that users are engaging with, content creators can customize their output to satisfy user preferences and needs. The project can be used by data scientists and analysts to create new models and algorithms for behavior analysis and predictive analytics. In the end, consumers gain from enhanced online experiences catered to their requirements and tastes.

IV. LITERATURE SURVEY

[1] Fang et al. (2016) conducted a fine-grained analysis of HTTP web traffic generated by mobile devices. The dataset used in this study consists of large-scale mobile network traffic traces collected from a major cellular network operator, containing HTTP traffic logs from millions of mobile subscribers. The methodology involves preprocessing the raw data, extracting relevant features, and applying clustering and classification techniques to identify patterns and gain insights into web traffic characteristics. The results demonstrate the effectiveness of the fine-grained analysis in uncovering valuable insights, such as popular web domains, content types, and user preferences. However, the paper acknowledges limitations related to user privacy concerns and the need for scalable analysis techniques to handle the ever-increasing volume of mobile web traffic.

[2] Hanamanthrao et al. (2017) addresses the problem of real-time processing and visualization of clickstream data. The dataset used in this research is clickstream data collected from web servers and client browsers. The proposed methodology involves ingesting clickstream data using Apache Kafka, processing the data in real-time with Apache Spark Streaming, and storing the processed data in Apache Cassandra.The processed data is then visualized using Apache Superset, providing insights into user behavior, traffic patterns, and website performance. The results show that the system can effectively process and analyze clickstream data in real-time, enabling website administrators and marketers to make informed decisions. However, the paper acknowledges limitations such as the need for more advanced machine learning techniques for predictive analytics and the challenge of handling diverse data formats.

[3] Ni et al. (2017) developed a method to analyze internet traffic using community detection and Apache Spark. Their work aimed to simplify the complex task of monitoring and analyzing the behavior of a large number of internet users, particularly in a large campus network. By focusing only

on IP-to-IP information and avoiding packet payloads, they created a model to find similarities in user behavior. They then built a similarity graph and applied a label propagation algorithm to identify communities of users with similar online behaviors. To handle the massive volume of data, they built a system using Apache Spark, a platform known for efficiently processing large datasets. Their experiments, conducted on a real campus network, showed that this method could not only group users with similar internet usage patterns but also identify unusual traffic behaviors, which could be vital for network management and security. The success of their approach demonstrates the potential of using big data tools like Apache Spark for large-scale internet traffic analysis.

V. PROJECT OVERVIEW AND ARCHITECTURE

A. Methodology

The methodology of this project is designed to effectively merge real-time data processing with in-depth offline analysis to enhance the operational efficiency and user experience of an e-commerce platform. Fig. 1 shows the overall workflow of our project using different real-time and offline components. The project assumes a hypothetical website of furniture products. The data is synthetically generated using Faker library. The generated data is streamed into kafka that acts as a messenger between different components. The raw data from kafka is recieved by Redis where initial checks and count metrics are calculated. The valid users are streamed to another kafka topic.

From kafka, two branches are forked, one is online and another is offline. For batch processing or offline branch, the kafka topic is consumed by MongoDB that acts as a persistent storage of the system. The data from this database is used for feature engineering, locality sensitive hashing, model training and evaluation, and explainable AI. In the real-time pipeline, kafka topic is consumed by Flink, where data is processed and send to elastic search. Elasticsearch facilitates sophisticated search facilities. Kibana uses Elasticsearch's indexed data to generate real-time dashboards and visualizations that let stakeholders track important metrics and learn about user behavior and system performance. The proposed methodology uses scalable state of the art technologies and techniques.

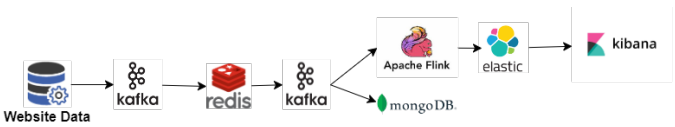

Fig. 1. Project Workflow

B. Architecture

The architecture of the e-commerce analytics project is designed to handle both real-time and batch processing needs of an e-commerce platform, ensuring scalability, flexibility, and robust data handling. The system is segmented into several key components, each serving a specific function within the larger ecosystem:

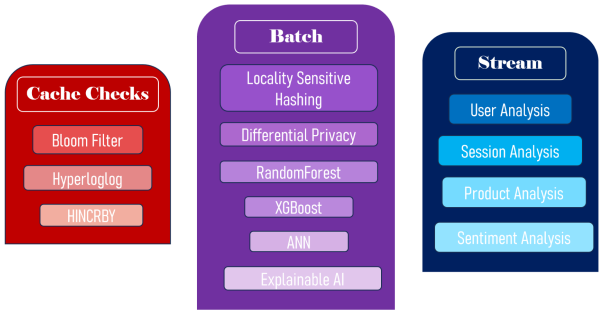

Fig. 2. Algorithms and Processing Stack

Docker: The different components of the system are containerized using docker. Docker simplifies the deployment of applications by packaging them along with their dependencies in portable containers, ensuring consistency across multiple development, testing, and production environments. This streamlined containerization approach significantly accelerates development cycles by facilitating easy scaling and management of applications.

```
(bigdata) C:\Users\bhati\BigData228\Project\BigData288-Project>docker ps
CONTAINER ID   IMAGE                                                  COMMAND                  CREATED       STATUS
     PORTS                                              NAMES
6aa2e081ed6c   docker.elastic.co/kibana/kibana:8.11.1                 "/bin/tini -- /usr/l…"   2 days ago    Up 45 secon
ds   0.0.0.0:5601->5601/tcp                             kb-container
6fd30b1a7cf7   docker.elastic.co/elasticsearch/elasticsearch:8.11.1   "/bin/tini -- /usr/l…"   2 days ago    Up 48 secon
ds   0.0.0.0:9200->9200/tcp, 9300/tcp                  es-container
f08e2b7299e1   confluentinc/cp-zookeeper:7.4.0                        "/etc/confluent/dock…"   2 days ago    Up 57 secon
ds   2888/tcp, 3888/tcp, 0.0.0.0:22181->2181/tcp      project-zookeeper-1
22a1aef1317f   redislabs/redismod:latest                              "redis-server --load…"   2 days ago    Up 51 secon
ds   0.0.0.0:6379->6379/tcp                            redis
7869d6cafcab   mongo:latest                                           "docker-entrypoint.s…"   10 days ago   Up 40 secon
ds   0.0.0.0:49153->27017/tcp                          mongodb
```

Fig. 3. Data Stored in RedisDB

Data: Synthetic data simulating user interactions, sessions, and product details is generated to mimic real-world e-commerce activity. Faker Library is used for the same. Our dataset contains all-encompassing measurements for engaging users and how they interact with the system in the digital environment. The information consists of user particulars (user, name, email, and gender); device information (device type, operating system, and browser type); session dynamics like product IDs, names, prices, categories, and user engagement actions; as well as tracking purchase transactions, review sentiments referral sources and geo-location among others. This wealth of data forms our analytics base which enlightens us on what decisions should be made towards improving sites' usability or even customer satisfaction levels through websites. Fig. 4 shows the data generated by faker and sent to Kafka.

Apache Kafka: Serves as the central messaging system that ingests and distributes the synthetic data streams to various components of the architecture, ensuring high throughput and fault tolerance. Raw data is sent to kafta topics and consumed by Redis. Also, Valid customers are sent to Kafka topic and consumed by different components. Fig. 5 shows the data that is being send by kafka producer on the topic 'user session info'.

Fig. 4. Session data for single user

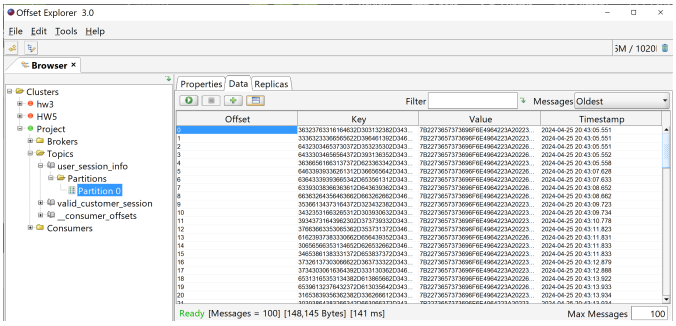

Fig. 5. Message to Kafta Topic

Data Storage - Redis: Utilized for its low-latency data processing capabilities, Redis performs immediate computations such as session counting and real-time security checks directly from the data stream provided by Kafka. This is a cache DB and in used for faster processing in the entire pipelines. For real-time analytics processing and temporary data storage for immediate computations, RedisDB comes with an in-memory data structure store that provides faster speed of processing information. Fig 2. shows the algorithm stack used in Redis DB.

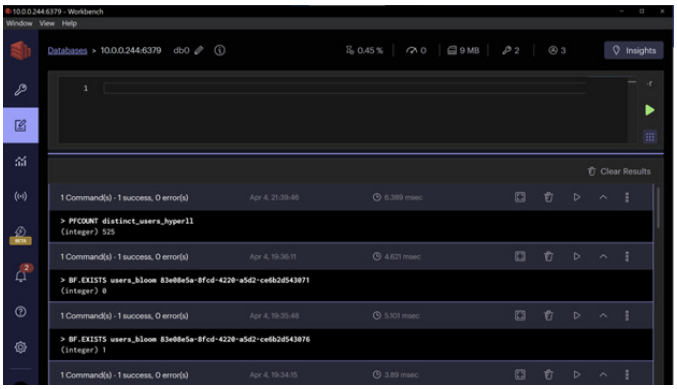

Fig. 6. Data Stored in RedisDB

Data Storage - MongoDB: Acts as the primary data store for historical data, where data is batch-loaded from Kafka. Data is fetched from the MongoDB and processed to do feature engineering, Locality Sensitive Hashing, training Random Forest, XGBoost, ANN models using differential Privacy, Explainable AI. MongoDB was a better option for the JSON format that our Data was generating with dynamic components. MongoDB being a NoSQL database is applied due to its strong storing abilities which can handle different types of datasets efficiently over longer periods. These databases work hand in hand with the system's architecture by giving fast access to data and durable storage hence supporting high throughput needs for real time website traffic analysis. This combination ensures that our system will perform well not only under various loads but also it keeps long term persistence and reliability of information necessary for reporting or analysis purposes.

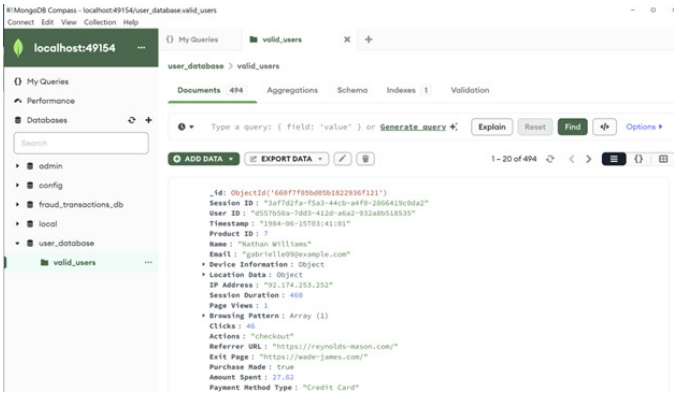

Fig. 7. Data Stored in MongoDB

Apache Flink: Apache Flink is a powerful stream processing framework that excels in handling real-time data flows. It integrates with Kafka to process streams for real-time analytics. Flink's stateful computations enable feature extraction, session analysis, and temporal data processing directly from the stream. Datastream APIs were utilized and environment is configured with necessary JAR files to ensure compatibility with Kafka for data ingestion, and Elasticsearch for indexing and visualization. The incoming data is processed to extract and modify all the information required for user, product, and session streams. Sentiment analysis of the products are also done based on the customer review in this layer. Fig. 2 shows the algorithm stack of stream processing done.

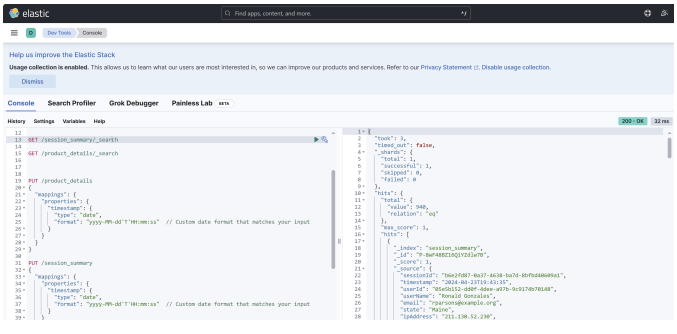

Fig. 8. Elasticsearch indexes

ElasticSearch: Configured to index large volumes of processed data for fast retrieval, supporting dynamic query requirements and acting as the analytics engine. Multiple

Streams from Flink are sinked to multiple indexes of Elasticsearch. Fig 8. shows elastic search indexes.

Kibana: Connected to Elasticsearch, it provides comprehensive visualization capabilities that allow stakeholders to monitor key performance indicators in real-time, visualize trends, and extract actionable insights. These specific visualizations within the Kibana dashboard provide an overview of our full analytics. The bar chart, which shows different percentages of neutral, positive, and negative ratings, offers a quick assessment of consumer mood for various furniture categories. In addition, a balanced breakdown of payment choices is shown by the pie chart, which shows the percentage of payments made with credit cards, PayPal, and abandoned carts. These revelations are essential to a wider range of data visualizations intended to maximize business strategy and consumer experience.

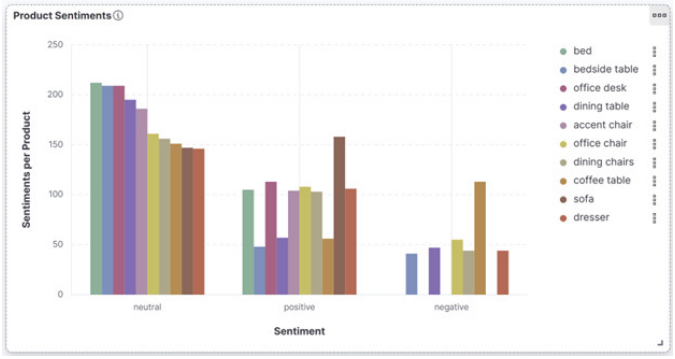

Fig. 9. Product Sentiments Visualization with Kibana

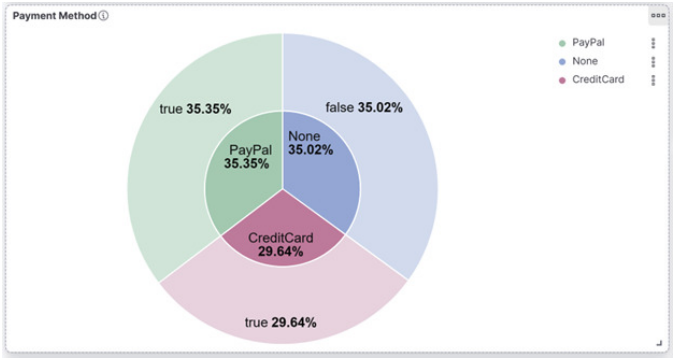

Fig. 10. Payment Method Visualization with Kibana

Machine Learning Models: Deployed to run on the data stored in MongoDB, models like Random Forest, XGBoost, and ANN utilize continuously increasing data from MongoDB to predict if user will make a purchase or not based on user behavior. Differential Privacy techniques are explored using these models. Differential privacy involves adding noise to the data or to the learning algorithms, thereby ensuring the privacy of individual entries in the dataset. We use Laplacian noise for our experiment. SHAP (SHaply Additive exPlanations) library was used for model interpretibility. Locality Sensitive Hashing Techniques are used to identify similar users. Fig 2 shows the algorithm stack of batch processing done.

C. Tools & Technologies

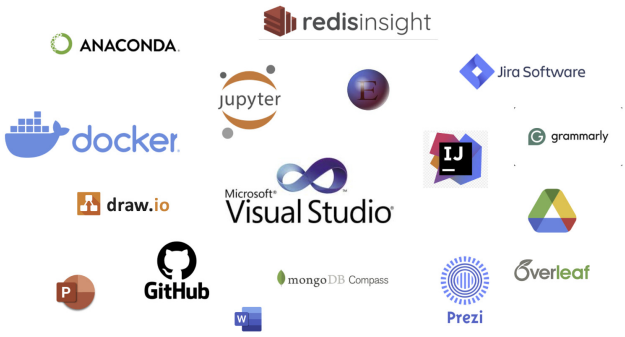

Fig. 11. Our technology stack for this project

- Anaconda - Used to simplify environment setup and dependency management as a Python distribution and package manager
- Docker - Utilized for containerization of project components, enabling consistent deployment across different environments
- IntelliJ - Utilized as an integrated development environment (IDE) for Python and other programming tasks
- Microsoft Visual Studio - Employed as an integrated development environment (IDE) for various programming tasks, including Python development
- Jupyter Notebook - Utilized as an interactive computing environment for data exploration, analysis, and visualization
- Python - Utilized for data processing tasks within the project, leveraging its powerful libraries and tools
- Apache Kafka - Employed as a distributed streaming platform for building real-time data pipelines and streaming applications
- Offset Explorer - Utilized for monitoring and managing consumer group offsets in Apache Kafka
- Apache Flink - Utilized for real-time data stream processing, offering advanced analytics capabilities and low-latency processing
- RedisDB - Employed as a high-performance, in-memory data store for caching and real-time data processing
- MongoDB - Utilized as a NoSQL database for storing and managing structured and unstructured data
- Elastic Search - Employed as a distributed, RESTful search and analytics engine for indexing and querying large volumes of data
- Kibana - Utilized as a data visualization tool for analyzing and presenting data analytics results
- GitHub and GitHub Desktop - Project version control
- Microsoft word - Utilized for general document editing and formatting tasks

- Microsoft Powerpoint - Utilized for creating and delivering presentations
- Google Drive - Utilized for cloud-based collaboration and document sharing among team members
- LaTeX (Over leaf) - Used for high-quality typesetting and formatting our documents according to IEEE standards.
- Prezi - Utilized for creating dynamic and visually engaging presentations
- Simple show video maker - Utilized for creating dynamic and visually engaging videos for project presentations and demonstrations
- draw.io - Utilized for creating diagrams and visual representations of project architectures and workflows
- JIRA Software - Utilized for managing the project lifecycle and task tracking
- Grammarly - Employed for grammar checking in project documentation to ensure clarity and professionalism

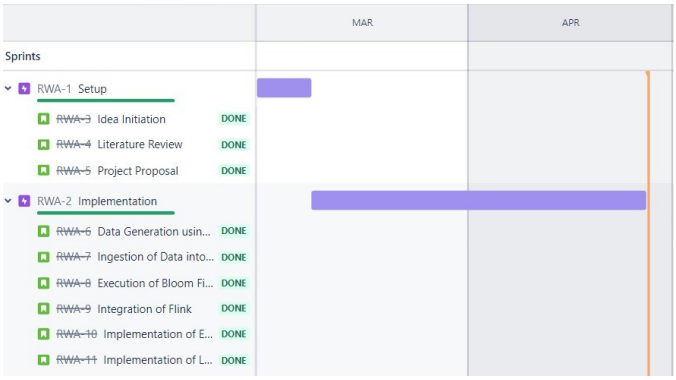

Fig. 12. JIRA for project management

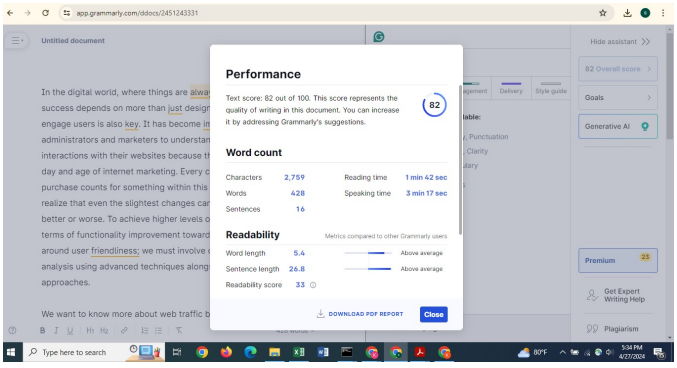

Fig. 13. Grammarly for grammar correction

D. Algorithm Implementation

Locality-Sensitive Hashing

In this project, Locality-Sensitive Hashing (LSH) plays a pivotal role in efficiently identifying and grouping users with comparable behavior patterns. Once user session data is securely stored in MongoDB, LSH is applied to process this vast dataset. The ingenuity of LSH lies in its ability to hash users who exhibit similar session characteristics—such as items browsed, time spent, and purchase history—into the same 'buckets' with a high likelihood. This clustering effect significantly speeds up the process of pinpointing users whose interactions suggest parallel preferences or activities. By utilizing this approach, the system can swiftly recommend products, tailor content, or provide personalized experiences. This, in turn, enriches user engagement and fosters a sense of connection among users with aligned interests.The application of LSH has not only optimized our system's backend operations but has also provided a direct uplift to the user experience, facilitating a more responsive and intuitive e-commerce platform.

Differential Privacy

For our e-commerce analytics project, we used a number of sophisticated analytical methods, including a deep learning model and conventional machine learning models like XGBoost and Random Forest, all of which were performance-evaluated and integrated with differential privacy mechanisms to guarantee data confidentiality.

Achieving an overall accuracy of 81%, the Random Forest model with differential privacy showed how to balance precision and recall. This strong result demonstrated the model's efficacy even with additional noise included for privacy. On the other hand, XGBoost was the best model, with the highest precision and recall values; this led to an accuracy of 88% and an F1 score of 0.88. This model was especially useful since it could manage intricate, feature-rich datasets with ease.

We also created a deep learning model with a sequential architecture consisting of layers of 128 and 64 neurons for a more in-depth investigation. This model first shown outstanding generalization abilities by achieving 100% accuracy on the training set and an astounding accuracy of 89% on the test set. But after incorporating differential privacy with DPKerasSGDOptimizer from TensorFlow Privacy, the accuracy dropped significantly to 59% with an F1 score of 0.62. This decrease brought to light the difficulties in striking a balance between data utility and privacy protection.

In order to adhere to data protection laws and uphold customer confidence, these models' implementations of differential privacy included strategies including noise addition and sensitivity adjustment. The epsilon value of the privacy budget quantified the loss of privacy and offered a clear framework for comprehending the associated trade-offs.

Explainable AI using SHAP

By integrating Explainable AI using SHAP (SHapley Additive exPlanations) into our project, we have learned how our predictive models arrive at their conclusions. which is an approach to explain the output of machine learning models. We apply SHAP to interpret the predictions of our models, helping us to understand which features are most influential in predicting whether a user will make a purchase. It shows us what each factor does to the predictions made by our machine learning algorithms about future events. High SHAP values often mean that a certain characteristic is strongly associated

with the output variable (for example, session duration or number of clicks). However, other features, such as the type of device used, may have more complex effects on the model's output—in this case, slightly lowering it when mobiles are involved. This granular understanding enables us to not only see through the eyes of website visitors but also adapt their journey accordingly. Fig 14. shows the interpretation done by SHAP for XGBoost model.

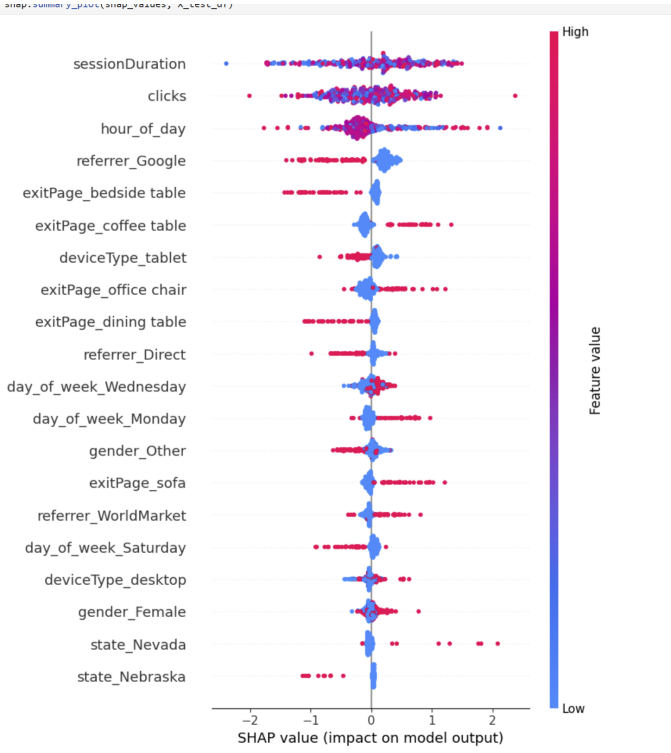

Fig. 14. SHAP (SHapely Additive exPlainations) of XGBoost

VI. Output

Fig 15. shows the user interface by Kibana. It consists of multiple visualizations. Fig 16. shows the potential customers that are spending a lot of time on viewing products but are not purchasing them. Fig 17. shows the overall estimates of number of purchases done, revenue generated, and average session duration in real-time.

VII. Lesson Learnt

This project has taught us a lot about integrating complex technologies for real-time analytics in e-commerce. We realized that having strong data pipelines is necessary in order to ensure synchronization and integrity across platforms like Kafka, Redis, MongoDB, Flink, Elasticsearch, and Kibana. Integrating Flink with other components specifically was complex due to incompatibility issues. Specific JARS has to be deployed for the integration. We also looked into scalability problems, which resulted in early design thoughts on how to handle vast amounts of data efficiently. Privacy and security were taken care of with great attention. We successfully dealt

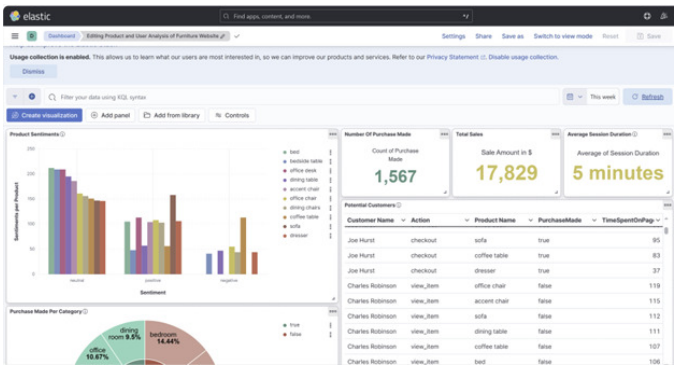

Fig. 15. User Interface

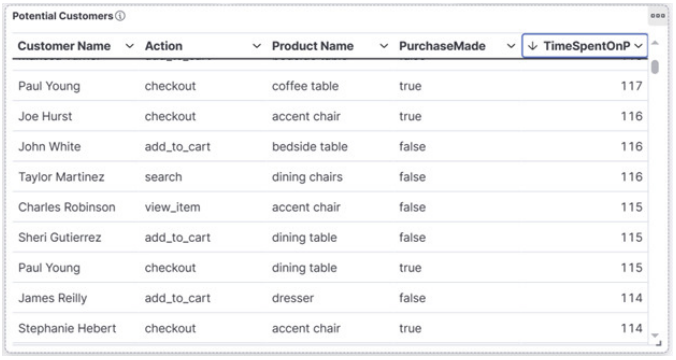

Fig. 16. Potential Customers Output

with the intricacies of combining differential privacy with real-time anomaly detection—balancing precision against ethical use of data. In the end, what speeds up decision-making is the dynamism of data visualization itself, which shows hidden user behavior patterns as demonstrated by Kibana, used for visual analytics. Moving forward, our reflections will be valuable when undertaking big-data projects in a fast-paced environment.

VIII. Innovation

Our project is unique because it merges real-time data processing with comprehensive offline analysis performed through Kafka, Redis, MongoDB, Flink, Elasticsearch, and Kibana. This is done by allowing the flow and integration of data with this method, whereby short-term operational insights as well as long-term strategic analysis can be enabled. The utilization of powerful machine learning models in the offline segment improves real-time processing capabilities for Apache Flink while ensuring quick response times and wide-reaching

Fig. 17. Output

analytic insights through Locality Sensitive Hashing (LSH). Additionally, our models incorporate explainable AI (SHAP) and differential privacy into them, which shows that we strive towards using transparently ethical information, thereby setting new standards for privacy-aware analytics centered on users. Such a framework fosters innovation by enhancing the customer experience as well as operational efficiency, and it also serves as a base for future industry developments since it can be scaled up easily.

IX. Team Work

TABLE I
TEAM MEMBER ROLES

We all worked together on each component of the pipelines discussing doubts and fixing issues together in college premise and on zoom sessions.

Team Members	Role
Vidushi Bhati	Data Generation, batch processing real-time processing
Kamakshi Lahoti	Data Generation, batch processing real-time processing
Akanksha Paspuleti	Data Generation, batch processing real-time processing
Yesheswini Lakshmi Spandana Potti	Data Generation, batch processing real-time processing

X. Pair Programming

The prosperity of our project resulted from pairing in programming, where we could make the most out of technology multiplicity. This means that when working in pairs, we were able to incorporate Apache Kafka, Apache Flink, RedisDB, and MongoDB into our system architecture. While two of us were dealing with real-time data processing through Kafka and Flink; the rest were focused on creating strong data storage systems based on RedisDB and MongoDB. Moreover, Python programming knowledge made it easy for us to implement intricate algorithms while documentation skills together with visualization tools enhanced clarity in documenting projects as well as representing them visually. In other words through this collaboration approach, we were able to utilize all our skills hence enhancing innovation which leads to the provision of high-quality solutions. We used github and gdrive for code versioning and storage.

XI. Relevance to the course

The project supports the educational goals of our course in big data systems and technologies. It applies tools like Apache Kafka and Apache Flink in practical contexts, equipping students with skills to handle massive datasets in actual work settings. Bloom filters introduce them to efficient, probabilistic data structures which help them understand complex methods for managing information. Their theoretical understanding is put into practice through the creation and installation of a real-time web traffic analysis system thus connecting what they learn inside class with hands-on problem solving ability in technology.

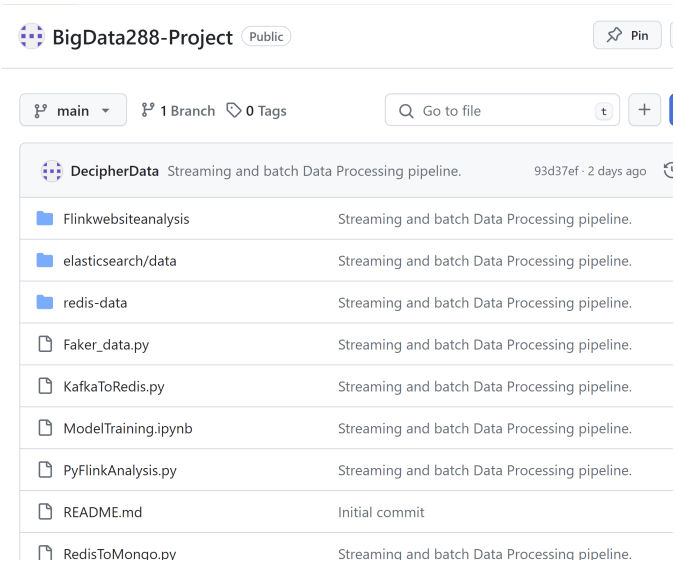

Fig. 18. Code on Github

XII. Technical Difficulties

While working on the project, we had many technical problems. This made it quite difficult for us to integrate the offline data with real-time data. We faced a big challenge in ensuring that data was synchronized smoothly across different platforms like Redis, MongoDB, Flink, Kafka, Elasticsearch, and Kibana, which called for a complete overhaul of our data pipeline architecture. Another problem came in terms of system scalability, especially when dealing with large volumes of artificially created data, which required us to optimize our data processing methods so as to be more efficient. Additionally, applying differentiated privacy and maintaining accuracy in analytics proved hard because it involved making sure users's information was secure while still getting insights out of it. In other words, these were valuable lessons that taught us how to be better technically equipped when working with complex datasets in fast-paced environments.

XIII. Novelty

The uniqueness of our project is that it integrates various technologies and methods to solve the problem of monitoring traffic at the moment of purchasing on the internet. We ensure fast data processing and detection of abnormalities by bringing together Apache Kafka, Apache Flink and Bloom filters algorithm. The model is scalable, efficient and user-centric which makes it different from other models used traditionally.

XIV. Impact

By using advanced technologies including Apache Kafka and Apache Flink, the project commits to change user involvement as well as website optimization. Administrators can boost customer experience by adapting engagements to personal tastes thus keeping people for longer and increasing income with immediate knowledge about visitor behaviours. These systems are smoothly combined so that they could be

easily expanded on any scale which contributes towards cost-effective but efficient realization of this goal while maximizing profits from the site where it was implemented as its main aim is to make sure that users have no difficulties when browsing through different pages or sections and also earn more money at the same time.

XV. Discussions and conclusion

Our project presents a unique technology that enables instantaneous analysis of web traffic, improving user experience and offering significant insights into visitor behavior. In today's digital landscape, website performance and user engagement are crucial so we developed a strong system that can process and analyze streaming data in real-time by integrating state-of-the-art technologies like Apache Kafka, Apache Flink, and Bloom filters. This innovative method exemplifies how real-time data analysis may improve security and user experience. In order to maintain our position at the forefront of web analytics and empower marketers and website administrators to make data-driven decisions for website and content optimization, we understand how critical it will be to keep up with evolving technology.

Acknowledgment

Our gratitude goes to the Department of Applied Data Science in San Jose State University. We appreciate Professor Pendyala's assistance and involvement in this project. We are also thankful for the tremendous help provided by the Instructional Student Assistants (ISAs) in addressing any questions we had.

Appendix

A. Code Walkthrough

During the presentation, we explained every segment of the code that is utilized at different levels of the architecture, and a comprehensive demonstration was given with this.

B. Discussion / Q&A

Following each presentation, we have allocated time to answer questions and discuss potential topics.

C. Demo

To display the results, a website analysis will be performed live while giving the presentation.

D. Report

We have documented each stage in a detailed manner and provided short instances for output. We maintained a formal tone throughout the report. The report was prepared with LaTeX and structured according to IEEE format.

E. Version control

To track code changes, a GitHub repository is created. We have used Git commands such as git push and git init to match our local project with the GitHub repository so that it includes project files and rubrics in the commit. https://github.com/DecipherData/BigData288-Project

F. Lessons Learned

We learned a lot from this project about incorporating real-time data analysis tools such as Kafka, Apache Flink, and Bloom filters. We had to overcome problems with accuracy of data, system integration, and optimizing performance by making strategic changes. It became clear during the event just how important it is for us to test everything properly, apply theoretical knowledge in practical situations, and stay adaptable. Real-time analytics were shown through this project to have an impact on user engagement as well as site speed improvements, which has also helped us grow technically. We have mentioned all of these learnings in the "Lessons Learnt" chapter in the report.

G. Prospects of winning competition / Publication

In this digital marketing context, the project is highly applicable because it meets crucial e-commerce needs by enhancing both user engagement and website performance; therefore, it has a high probability of publication or winning a competition. The reason behind this is that the system creatively incorporates real-time data analysis tools such as Apache Kafka and Flink, among others, like Bloom filters.

H. Innovation

This project creatively integrates Apache Kafka, Apache Flink, and Bloom filters to give real-time information about internet traffic. It does this by allowing instant data decision-making and e-commerce strategy improvement, which we have mentioned in our report.

I. Teamwork

All members of the team have taken part in all stages of the project's growth, submitting thoughts based on their respective fields. We've had many meetings where we shared ideas about the project and helped each other out when we made mistakes. Roles are described in our report under the heading "Teamwork."

J. Technical difficulty

We have included a section for the errors encountered during the process of developing the project.

K. Pair Programming

In the pair programming section of our report, we have discussed the process.

L. Practiced agile / scrum

We have been using Jira Board by Atlassian to manage the project as well as track its progress. The reason we opted for a Kanban board was because of its simplicity. We also created sprints for every stage depending on how hard the task was so that we could meet deadlines, and we followed them through with frequent Zoom meetings. ISAs are invited to join and validate the board.
https://rb.gy/c7ts9w

M. Used Grammarly / other tools

To check the report's grammatical errors, we have used Grammarly.

N. Presentation Techniques

We used Prezi to prepare our main presentation. It is not linear like Microsoft PowerPoint and has a more interactive visual approach. It is more flexible than Canva or other similar tools. We have also worked with DALL-E on generating visuals in the slides. https://prezi.com/view/PoPteC89dsubNiDpMHwi/

O. Elevator Pitch Video

We have created the Elevator Pitch Video using Simpleshow video maker to demonstrate our project. https://drive.google.com/drive/folders/1_H8Y1dinwIgy9DsYIyRCTmFUKqj9KZ_Z?usp=sharing

P. Used LaTeX

We have used IEEE LaTeX template from overleaf website. https://www.overleaf.com/read/vzqnhqzvbjvk#b6d07d

REFERENCES

[1] Fang, C., Li, J., and Lei, Z. (2016), "Fine-Grained HTTP web traffic Analysis based on Large-Scale mobile datasets," IEEE Access, 4, 4364–4373, doi: 10.1109/access.2016.2597538.

[2] Hanamanthrao, R., and Thejaswini, S. (2017), "Real-time clickstream data analytics and visualization," 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), doi: 10.1109/rteict.2017.8256978.

[3] Ni, J., Weng, W., Chen, J., and Lei, K. (2017), "Internet traffic analysis using Community Detection and Apache Spark,"2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, doi: 10.1109/cyberc.2017.85

[4] Pal, G., Li, G., and Atkinson, K. (2018), "Big Data Real Time Ingestion and Machine Learning," 2018 IEEE Second International Conference on Data Stream Mining and Processing (DSMP), doi: 10.1109/dsmp.2018.8478598.

[5] Pal, G., Li, G., and Atkinson, K. (2018), "Big Data Real-Time Clickstream Data Ingestion Paradigm for E-Commerce analytics," IEEE 4th International Conference for Convergence in Technology, doi: org/10.1109/i2ct42659.2018.9058112.

[6] Pal, G., Atkinson, K., and Li, G. (2021), "Real-time user clickstream behavior analysis based on apache storm streaming," Electronic Commerce Research, 23(3), 1829–1859, doi: 10.1007/s10660-021-09518-4.

[7] Patil, T., Anand, K., Bhateja, A., Jamal, K., Sawant-Patil, S. T., and Paygude, P. (2023), "Real-Time Clickstream Data Processing and Visualization Using Apache Tools," 2023 7th International Conference on Computing, Communication, Control and Automation (ICCUBEA), doi: 10.1109/iccubea58933.2023.10392270.

[8] RodriGuez, F., Lee, H. R., Rutherford, T., Fischer, C., Potma, E. O., and Warschauer, M. (2021), "Using Clickstream Data Mining Techniques to Understand and Support First-Generation College Students in an Online Chemistry Course," LAK21: 11th International Learning Analytics and Knowledge Conference, doi: 10.1145/3448139.3448169.

[9] Wen, X., Han, Y., Fu, J., Li, P., and Meng, F. (2020), "Design of user behavior analysis model of e-commerce website based on Spark," 2020 7th International Conference on Information Science and Control Engineering (ICISCE), doi: org/10.1109/icisce50968.2020.00112