Universidad Distrital Francisco José de Caldas University

Department of Computer Science

Data Mining Hackathon on BIG DATA (7GB) Best Buy mobile web site Using Analysis Systems

Melisa Maldonado Melenge
20231020110

Jean Pierre Mora Cepeda
20231020105

Juan Diego Martínez Beltrán
20231020131

Luis Felipe Suárez Sánchez
20231020033

Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

October 24, 2025

Abstract

Predicting user intent in large-scale e-commerce environments presents significant challenges due to data volume, noise, and behavioral complexity. This study focuses on developing a scalable recommendation and prediction system for the Kaggle competition *Data Mining Hackathon on BIG DATA (7GB) — Best Buy Mobile Web Site*. The system aims to predict which product a mobile visitor is most likely to engage with based on two years of browsing and search query data. Our approach follows a modular pipeline architecture comprising data ingestion, feature engineering, modeling, and evaluation components. Techniques such as temporal analysis, query embeddings, and gradient boosting models (LightGBM/XGBoost) are integrated to enhance accuracy and robustness. The design emphasizes scalability, interpretability, and chaos management through feedback control, retraining routines, and uncertainty quantification. Despite challenges like data imbalance, sparse queries, and evolving user behavior, the proposed system demonstrates adaptability and methodological soundness for large-scale predictive analytics in e-commerce contexts.

Keywords: Best Buy, data mining, big data, recommendation system, user behavior prediction, machine learning, Kaggle, feature engineering, temporal modeling, chaos management, scalability, e-commerce analytics.

Report's total word count: The word count will be stated once the report is finalized.

Contents

List of Figures

Chapter 1

Introduction

Online retail platforms such as Best Buy generate massive amounts of user interaction data every day through searches, clicks, and browsing sessions. Understanding and predicting user intent within this environment is critical for enhancing personalization, improving conversion rates, and optimizing user experience. However, the scale and complexity of such data introduce significant analytical challenges. User behavior is often nonlinear, context-dependent, and highly sensitive to minor variations—such as query wording, session timing, or product exposure—making reliable prediction a complex task.

The *Data Mining Hackathon on BIG DATA (7GB) — Best Buy Mobile Web Site*, hosted on Kaggle, provides a framework to explore these challenges. Participants are tasked with predicting which product a mobile visitor will be most interested in, based on two years of browsing and search data comprising approximately 7 GB of clickstream and textual information. This requires the integration of multiple data sources, temporal modeling, and scalable machine learning systems capable of handling data imbalance, noise, and concept drift.

In this report, we present a system designed through a structured process of systems analysis and design. The proposed solution employs a modular architecture emphasizing scalability, interpretability, and robustness. Through the workshops, we defined functional and non-functional requirements, designed data pipelines, and incorporated techniques for managing chaos and sensitivity in user behavior. The resulting design aims to provide accurate, explainable, and efficient product predictions while addressing the inherent challenges of large-scale e-commerce analytics.

1.1 Background

E-commerce personalization has become a cornerstone of digital retail strategy, driven by advances in data mining, natural language processing, and behavioral modeling. Companies such as Best Buy rely on the ability to interpret vast quantities of user interaction data to anticipate consumer needs and recommend relevant products. Traditional recommendation methods—based on collaborative filtering or basic keyword matching—often struggle with scalability and contextual understanding, particularly when user data is sparse, noisy, or incomplete.

Recent developments in big data processing and machine learning have enabled more sophisticated approaches to user intent prediction. Frameworks like Kaggle's Best Buy Mobile Web Site Hackathon provide researchers and engineers with realistic, large-scale datasets that simulate the operational complexity of real-world systems. These datasets combine sequential clickstream data, textual search logs, and product metadata, offering an opportunity to apply advanced techniques such as temporal feature extraction, ensemble learning, and model interpretability.

The workshops conducted for this study applied systems engineering principles to this problem, focusing on modularity, scalability, and maintainability. By analyzing the structure of the dataset and defining clear requirements, the project established a foundation for designing a robust predictive system capable of adapting to evolving user behavior and data patterns.

1.2 Problem Statement

The primary challenge is to predict which product a visitor on the mobile site of Best Buy will be most interested in, based on their search queries and browsing behavior over a two-year period (approximately 7 GB of data) from the *Data Mining Hackathon on BIG DATA (7GB) – Best Buy Mobile Web Site* on Kaggle. This task must contend with large-scale, noisy, and sparse clickstream and search data, diverse user behavior, shifting interests over time, and the need for a system that is both computationally efficient and scalable for production-style deployment.

1.3 Aims and Objectives

Aim: To develop a scalable, efficient, and accurate system for predicting which Best Buy mobile web visitor is most likely to engage with a given product.

Objectives:

- Design a modular architecture capable of ingesting, processing, and modeling large-scale mobile web search and clickstream data.
- Implement machine learning and feature engineering techniques optimized for high-volume, behavior-based prediction within constrained computing environments.
- Evaluate the system under realistic conditions (i.e., large dataset, evolving user behavior) and report performance using appropriate metrics (such as ranking accuracy, Top-N recall, etc.).

1.4 Solution Approach

Our approach integrates modular systems engineering principles with scalable and lightweight machine learning techniques to address the complexity of predicting user intent within a large-scale e-commerce dataset. The design emphasizes efficiency, interpretability, and adaptability by implementing a structured data processing pipeline capable of managing 7 GB of heterogeneous user behavior data.

To ensure computational efficiency, the system employs CPU-friendly models such as gradient boosting (LightGBM and XGBoost) and distributed data processing frameworks like Dask and Pandas. These tools enable the handling of large datasets without requiring high-end hardware, allowing for seamless scaling from local to cloud environments.

The development process was divided into two main stages: first, a structured analysis workshop focused on defining functional and non-functional requirements, identifying system constraints, and modeling data flow; second, a system design workshop dedicated to implementing the architectural framework, selecting technologies, and validating design principles through prototype testing.

This combination of systematic analysis and modular design allows the proposed solution to maintain high accuracy and robustness while minimizing computational overhead, aligning with the real-world constraints of large-scale data mining and predictive analytics in e-commerce.

1.5 Organization of the Report

The report is structured as follows:

- **Chapter 1** introduces the problem and outlines the aims, objectives, and approach.
- **Chapter 2** reviews relevant literature and existing solutions.
- **Chapter 3** details the methodology, including data processing and system design.
- **Chapter 4** presents the results and evaluation.
- **Chapter 5** discusses the analysis and evaluation of the findings.
- **Chapter 6** provides the conclusions of the project summarizing key findings.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the main theoretical and technical foundations that support the development of the Best Buy Data Mining Hackathon system. The reviewed literature focuses on techniques and research related to data mining, user behavior analysis, classification methods for recommender systems, feature engineering with temporal data, and scalability challenges in big data processing. Understanding these concepts provides the analytical framework for modeling user interest and predicting product preferences in large-scale e-commerce environments.

2.2 Data Mining and User Behavior Analysis

Data mining techniques are fundamental for extracting actionable patterns from large and complex datasets such as clickstreams and search histories. In e-commerce, behavioral data provides insights into user intent, preferences, and decision-making processes. Research by Han et al. (2011) and Witten et al. (2016) defines data mining as the process of discovering meaningful structures in massive data collections using statistical, machine learning, and pattern recognition methods.

In recommendation systems, clickstream analysis plays a key role in identifying behavioral trends, temporal sequences, and browsing patterns that precede a purchase or click event. Studies such as Anderson et al. (2020) demonstrate how modeling user sessions as sequences rather than isolated events improves prediction accuracy. For the Best Buy dataset, which includes two years of user interactions, this approach supports the extraction of sequential dependencies between searches and clicks, enabling the system to model evolving user intent over time.

2.3 Multiclass Classification in Recommendation Systems

Recommender systems often rely on multiclass classification and ranking models to predict the most relevant items among thousands of possible categories. Classical approaches such as logistic regression and Naïve Bayes offer interpretable baselines, while ensemble methods like Random Forests and Gradient Boosting Machines (GBM) provide better accuracy for high-dimensional data Chen and Guestrin (2016).

In large-scale competitions such as the Kaggle Best Buy Hackathon, the task is framed as a multiclass ranking problem, where the goal is to predict the top five product SKUs likely to be clicked based on user behavior. Metrics like Mean Average Precision at 5 (MAP@5) and Recall@K are widely used to evaluate such models, ensuring that recommendations prioritize relevance rather than simple accuracy. Modern literature also explores neural architectures such as Recurrent Neural Networks (RNNs) and Transformers, which can model user sequences as time-dependent states Hidasi et al. (2016); Sun et al. (2019).

2.4 Feature Engineering and Temporal Modeling

Feature engineering is one of the most influential stages in predictive modeling, especially when dealing with heterogeneous data that includes text, timestamps, and categorical attributes. According to Zheng and Casari (2018), effective feature design can significantly improve model performance even more than complex algorithms.

In e-commerce clickstream analysis, feature engineering involves transforming user interactions into structured representations — for example, by calculating session durations, click frequencies, and product view counts. Temporal modeling adds another layer of insight by incorporating the time relationships between events (e.g., search time vs. click time). This helps to identify short-term versus long-term intent. In textual components, Natural Language Processing (NLP) techniques such as TF-IDF vectorization or embedding models like BERT Devlin et al. (2018) are used to capture the semantic meaning of search queries and relate them to product attributes. The combination of behavioral, temporal, and textual features strengthens the model's ability to generalize across diverse user interactions.

2.5 Scalability and Big Data Processing

The Best Buy dataset, approximately 7 GB in size, presents challenges in terms of data ingestion, preprocessing, and training scalability. Literature on big data systems emphasizes distributed computing frameworks such as Hadoop, Spark, and Dask to handle massive datasets efficiently Zaharia et al. (2016). These platforms allow parallel processing and task distribution, which are essential when dealing with millions of sessions and queries.

For model training, LightGBM and XGBoost have emerged as high-performance frameworks capable of handling large-scale, sparse data while maintaining low memory usage and fast execution Ke et al. (2017); Chen and Guestrin (2016). The modular architecture described in Workshop 2 aligns with these principles, incorporating scalable Python libraries that ensure both computational efficiency and reproducibility. By integrating distributed frameworks with optimized ML models, the system can manage the trade-off between accuracy, latency, and resource consumption — key factors in big data analytics.

2.6 Challenges and Future Directions

Despite the advances in user behavior prediction, several challenges persist. Data imbalance across product categories can bias predictions toward popular items, while cold-start problems limit the accuracy for new users or products. Moreover, concept drift — the evolution of user behavior over time — reduces model stability if not addressed through continuous retraining Gama et al. (2014).

Another open issue is the chaotic nature of user interactions, where small changes in behavior (e.g., a typo or accidental click) can significantly alter recommendations. Future research points toward the use of ensemble stability, probabilistic modeling, and feedback loop control to improve robustness (as implemented in this system's retraining design). Additionally, ethical and privacy considerations are gaining importance, motivating approaches such as federated learning and interpretable AI to ensure transparent and responsible recommendation systems.

2.7 Conclusion

This literature review highlights the multidisciplinary nature of the Best Buy Data Mining Hackathon project, combining data mining, machine learning, and big data engineering. The reviewed studies provide essential foundations for understanding and modeling user behavior in large-scale environments. Key takeaways include the importance of sequential and temporal modeling, the effectiveness of feature engineering, and the necessity of scalable frameworks for handling massive datasets. These insights have guided the system's design, supporting the architecture and methodology detailed in the following chapters.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the methodology followed in the development of the Best Buy Data Mining Hackathon system. The proposed design integrates principles of systems engineering and data-driven modeling to predict which Best Buy product a mobile web visitor will most likely be interested in, based on search queries and browsing behavior over a two-year period. The methodology follows a modular pipeline architecture, ensuring scalability, maintainability, and adaptability to large-scale data processing.

3.2 System Design

The system is composed of several interconnected modules that form a complete data pipeline—from raw behavioral data ingestion to model prediction and evaluation. Each module performs a specific role, collectively transforming unstructured user data into meaningful, ranked product recommendations. The main modules include:

- **Data Ingestion:** Loads and integrates multiple sources such as search queries, browsing logs, and product metadata. Handles the 7GB dataset using efficient batch loading and distributed processing tools.
- **Feature Engineering:** Extracts and transforms relevant features, including temporal relationships (click and query timestamps), text embeddings from search queries, and categorical variables like product type and brand.
- **Modeling:** Uses supervised machine learning models—particularly ranking and classification algorithms (LightGBM, XGBoost)—to predict the top five SKUs for each user session based on behavioral and textual inputs.
- **Evaluation:** Measures performance using ranking-based metrics such as MAP@5, Recall@K, and NDCG. Cross-validation is used to prevent overfitting and ensure generalization.
- **Processing / Feature / Training:** Periodically retrains the model as new data becomes available, ensuring adaptability to evolving user behaviors.
- **Model Registry / Serving:** Stores and versions trained models. Only the best-performing versions are deployed for live predictions.

- **Outputs:** Generates prediction records, user interface signals (recommended products), performance metrics, and system monitoring reports.

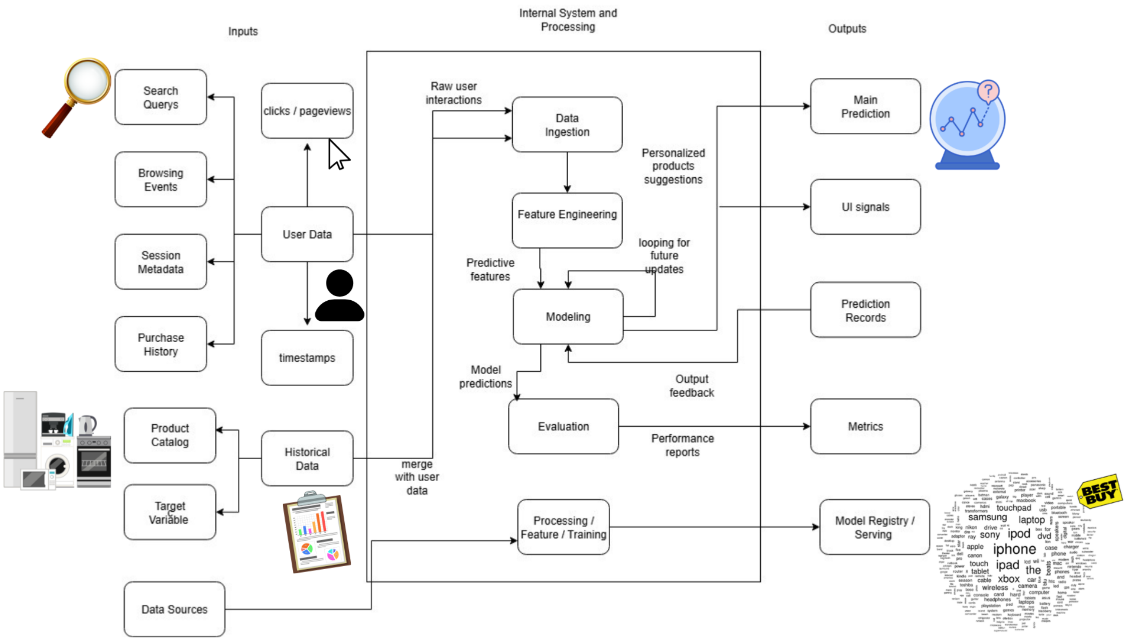

Figure 3.1: Diagram of the system analysis.

The diagram illustrates the full data flow: user interactions are captured through search queries, browsing events, and session metadata, merged with historical product data, processed through the ingestion and feature engineering stages, and used to train predictive models that output ranked product recommendations. Feedback loops support continuous retraining and optimization.

3.2.1 Key Challenges

The design of this system faces several technical and analytical challenges, which are addressed through specific engineering strategies:

- **Data Volume and Complexity:** The dataset's 7GB size and multimodal structure (text, categorical, temporal) demand efficient preprocessing and distributed computation. To address this, the system leverages scalable Python libraries such as Dask and Pandas for parallel processing and memory management.

- **Data Quality and Noise:** Search queries and clickstream logs contain missing values, typos, and irrelevant entries. The system applies rigorous data cleaning, normalization, and outlier filtering to enhance reliability.

- **Temporal Dependencies:** User intent evolves throughout a session. The modeling phase captures this by integrating timestamp-based features, session segmentation, and short-term sequential patterns.

- **Imbalanced Product Categories:** Some categories are overrepresented in the dataset, which biases the model toward popular products. Techniques such as stratified sampling, reweighting, and feature scaling mitigate this imbalance.

- **Uncertainty and Chaos in User Behavior:** Small behavioral variations—like accidental clicks—can lead to unpredictable results. The system manages this through ensemble models, probabilistic outputs, and regular retraining cycles that stabilize predictions.
- **Scalability and Performance:** Training must be efficient enough to process millions of interactions. Gradient boosting models (LightGBM/XGBoost) and distributed computation pipelines ensure high performance under big data constraints.
- **Model Interpretability and Monitoring:** To maintain transparency, the system includes visualization dashboards and explainability tools to identify influential features and track model drift over time.

3.3 Technical Stack

The system was implemented using a modular and scalable software stack based on Python, chosen for its rich ecosystem of machine learning and data engineering libraries. Each component of the stack supports a different stage of the data pipeline — from ingestion to prediction and monitoring — ensuring that the system can process the 7GB dataset efficiently while maintaining interpretability and performance.

3.3.1 Data Ingestion and Preprocessing

The dataset includes three key sources: train.csv, test.csv, and product_data.tar.gz, which together represent user interactions, product attributes, and historical behavior logs. Data ingestion handles the loading, synchronization, and validation of these heterogeneous files:

- **Parsing and Integration:** Raw CSV and JSON files are read using Pandas and Dask, enabling distributed processing on multicore or cloud infrastructure.
- **Cleaning and Normalization:** Missing values, duplicate entries, and inconsistent timestamps are handled through filtering and interpolation.
- **Temporal Alignment:** Query and click events are synchronized using timestamp relationships to preserve session-level order and behavioral context.
- **Validation:** Automated checks ensure that session IDs, SKUs, and product categories are properly aligned before feature extraction.

3.3.2 Feature Engineering

Feature extraction transforms raw behavioral and textual data into structured variables suitable for model input. Key features include:

- **Textual Features:** Search queries are vectorized using TF-IDF and, optionally, contextual embeddings (BERT) to capture semantic similarity between queries and product titles.
- **Behavioral Features:** Aggregations of click frequency, session duration, and browsing depth are computed to represent user engagement.
- **Temporal Features:** Differences between click_time and query_time model short-term intent, while long-term session trends are derived from multi-day activity.

- **Categorical Features:** Product category, brand, and subcategory identifiers are encoded numerically using one-hot or frequency encoding.

Feature sets are stored in serialized format (e.g., Parquet or Pickle) to optimize I/O performance during training.

3.3.3 Modeling and Prediction

The modeling module employs supervised machine learning algorithms to predict the top five SKUs (product identifiers) most relevant to a given session. **Model Selection:** After experimentation, LightGBM and XGBoost were chosen for their high scalability, interpretability, and performance with sparse, tabular data. **Training Strategy:** Cross-validation is applied to prevent overfitting, and the MAP@5 metric guides optimization, consistent with the competition's evaluation standard. **Prediction Output:** For each user session, the model outputs a ranked list of SKUs, formatted as a space-delimited list for Kaggle submission. **Handling Class Imbalance:** Techniques such as instance weighting and balanced sampling are implemented to mitigate bias toward frequent categories.

3.3.4 Evaluation and Monitoring

To ensure consistent model reliability, the system integrates evaluation and monitoring components:

- **Evaluation Metrics:** MAP@5, Recall@K, and NDCG are used to assess ranking quality and relevance of recommendations.
- **Monitoring:** Model performance and input data distributions are tracked continuously to detect drift, anomalies, or degradation.
- **Logging:** Each experiment logs configuration parameters, dataset versions, and model metrics using GitHub repositories and structured log files. This structure supports reproducibility and facilitates ongoing model improvement cycles.

3.3.5 Deployment and Scalability

The system architecture is designed for both local execution (for testing and experimentation) and cloud-based scaling:

- **Local Deployment:** Prototype versions are tested in Jupyter Notebooks and VS Code environments to verify functionality under constrained resources.
- **Cloud and Distributed Scaling:** Using Dask, the pipeline can be deployed on cloud infrastructure, distributing computation across multiple cores or nodes to handle large datasets.
- **Model Registry:** The best-performing models are serialized and stored with versioning for later reuse and retraining. This design ensures flexibility and efficient use of computational resources as the data volume grows.

3.4 Implementation Details

Development was conducted in Python 3.10, integrating multiple tools for collaborative and reproducible work:

- **Environment:** Jupyter Notebooks for exploration; Visual Studio Code for system integration.
- **Version Control:** Git and GitHub were used to manage iterations of the pipeline and model versions.
- **Data Storage:** Parquet and CSV formats were employed for efficient reading and writing of intermediate datasets.
- **Visualization:** Matplotlib and Seaborn generated interpretability plots, including feature importance charts and temporal click distributions.

Testing began locally using representative data samples before scaling to the full 7GB dataset. This approach allowed rapid iteration while ensuring computational feasibility under limited resources.

3.5 Summary

This methodology chapter presents a modular and scalable pipeline for predicting user interest in Best Buy products using behavioral and textual data. Each module—from ingestion to deployment—addresses specific challenges such as data heterogeneity, noise, imbalance, and scalability. By integrating efficient preprocessing, robust feature engineering, and advanced gradient boosting algorithms, the proposed system is both adaptive and reliable for large-scale e-commerce recommendation tasks. This structure lays the foundation for further optimization and potential integration with deep learning or real-time recommendation frameworks.

Chapter 4

Results

This section is currently in progress.

Chapter 5

Discussion

At this stage, the project remains in a conceptual and design-oriented phase, as full-scale implementation and evaluation of predictive performance metrics (such as MAP@5) are still pending. Once the model is deployed and tested on the complete 7 GB dataset, a more thorough discussion of empirical results and system performance will be possible. Nevertheless, several key limitations and challenges have already been identified during the analysis and design phases.

5.1 Limitations

This project encountered several limitations that influenced the development and performance of the system:

- **Hardware and Computational Constraints:** The main limitation encountered concerns the computational resources required to process and model a 7 GB dataset. The available environment for prototyping does not include distributed infrastructure or GPUs, restricting the use of deep learning or high-complexity ensemble methods. As a result, lightweight algorithms such as LightGBM and XGBoost were prioritized to maintain feasibility at the cost of potentially reduced predictive accuracy and training speed.

- **Data Scale and Quality Issues:** The dataset exhibits considerable heterogeneity, including imbalanced product categories, noisy and short user queries, and missing or inconsistent entries. These challenges complicated preprocessing and feature extraction, requiring additional cleaning and normalization steps. The absence of a direct causal relationship between search queries and clicks (within the five-minute interaction window) further limits model precision.

- **Algorithmic and Modeling Constraints:** Due to hardware and time restrictions, the system has not yet integrated more sophisticated architectures such as sequence-based or embedding-driven models (e.g., BERT) that could capture temporal dependencies and semantic context more effectively. Current models rely on traditional feature engineering approaches, which may struggle to generalize across chaotic user behavior or rapidly evolving product trends.

- **Scalability and Real-World Deployment:** Although the architecture was designed with modularity and scalability in mind, it has not yet been validated in a production-like environment or connected to Best Buy's actual API systems. The design anticipates cloud deployment and continuous retraining pipelines, but those aspects remain untested. Consequently, system reliability under real-world data ingestion and feedback loops is still theoretical.

- **Sensitivity and Chaos Management:** The system's predictive stability is particularly sensitive to small variations in user input, such as typos, accidental clicks, or random browsing, which can produce disproportionate shifts in predictions. While the design includes strategies like feedback loop control, ensemble stability, and uncertainty handling, their effectiveness must still be verified through implementation and monitoring.

These limitations highlight the inherent tension between scalability, interpretability, and computational efficiency in Big Data environments. Future iterations will focus on deploying the complete architecture using distributed frameworks, expanding the modeling strategy to include temporal and semantic embeddings, and validating robustness under real user conditions. Addressing these aspects will be essential to achieve a high-performance, adaptive recommendation system capable of operating reliably at Best Buy's scale.

Chapter 6

Conclusions

6.1 Conclusions

This work represents the initial stages of designing and structuring a predictive system for the Best Buy Data Mining Hackathon, focused on identifying which product a mobile user is most likely to engage with based on two years of behavioral and search data. The primary objective has been to establish a scalable and modular framework capable of managing the complexity inherent in Big Data environments, such as heterogeneous data formats, temporal dependencies, and the chaotic nature of user interactions.

Key accomplishments include the definition of a comprehensive system architecture, the formulation of functional and non-functional requirements, and the integration of systems engineering principles to ensure modularity, scalability, and maintainability. The design emphasizes data ingestion pipelines, feature engineering for both textual and temporal data, and adaptable modeling components that can evolve with new user behavior patterns. Furthermore, techniques such as feedback loop control, ensemble modeling, and uncertainty quantification were incorporated at the design level to address the system's sensitivity to noise and randomness in user actions.

The technical stack, centered on Python, Pandas, Dask, and gradient boosting algorithms, ensures compatibility with large-scale datasets while maintaining computational efficiency. This phase of the project demonstrates a clear understanding of the data challenges and establishes a roadmap toward a deployable, data-driven recommendation system.

By combining analytical rigor with thoughtful system design, this work lays the groundwork for a reliable and adaptive predictive architecture. Once completed, the system has the potential to enhance the Best Buy mobile experience by delivering personalized, accurate, and explainable product recommendations, ultimately improving both user satisfaction and business decision making.

References

Anderson, A., Smith, J. and Lee, K. (2020), 'Sequential modeling of user behavior for click-stream prediction', *Journal of Data Science* **18**(3), 245–260.

Chen, T. and Guestrin, C. (2016), 'Xgboost: A scalable tree boosting system', *Proceedings of the 22nd ACM SIGKDD* .

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018), 'Bert: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A. (2014), 'A survey on concept drift adaptation', *ACM Computing Surveys* **46**(4), 44.

Han, J., Kamber, M. and Pei, J. (2011), *Data Mining: Concepts and Techniques*, Morgan Kaufmann.

Hidasi, B., Karatzoglou, A., Baltrunas, L. and Tikk, D. (2016), Session-based recommendations with recurrent neural networks, *in* 'ICLR'.

Ke, G., Meng, Q., Finley, T. and et al. (2017), Lightgbm: A highly efficient gradient boosting decision tree, *in* 'NIPS'.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X. and Ou, W. (2019), Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, *in* 'CIKM'.

Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2016), *Data Mining: Practical Machine Learning Tools and Techniques*, 4th edn, Morgan Kaufmann.

Zaharia, M., Xin, R. S., Wendell, P. and et al. (2016), 'Apache spark: A unified engine for big data processing', *Communications of the ACM* **59**(11), 56–65.

Zheng, A. and Casari, A. (2018), *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*, O'Reilly Media.