Predicting Amazon's Product Browse Node Classification for Improved Customer Experience

Alaa Samir
Biomedical Informatics Senior Student
Information Technology and Computer Science
Nile Universty
Cairo, Egypt
a.EssamEldin@nu.edu.eg

Abdulaziz Amori
Artificial Intelligence Senior Student
Information Technology and Computer Science
Nile Universty
Cairo, Egypt
A.Amori@nu.edu.eg

***Abstract*—This project addresses challenges in Amazon's browse node classification, targeting a 90% accuracy goal. Leveraging big data and machine learning, our design utilizes tools. Aligned with Amazon's customer-centric mission, the project focuses on improving operational efficiency and the overall shopping experience. Utilizing the Amazon ML Challenge 2021 dataset, the solution not only meets immediate goals but also presents a prototype for future enhancements in e-commerce.**

Keywords—Amazon, browse node classification, big data, machine learning

I. INTRODUCTION

In the context of enhancing Amazon's Product Browse Node Classification for an improved customer experience, our project addresses the critical challenge of accurate and efficient assignment of browse nodes to the vast and diverse product catalog on the Amazon platform. The manual assignment of browse nodes has become increasingly complex due to the expansive catalog, leading to inaccuracies and extended processing times. This challenge, significant for both Amazon and its customers, underscores the importance of leveraging big data analytics and machine learning to refine the accuracy and efficiency of browse node classification. The findings revealed the significance of variables from both online reviews and promotional marketing strategies in predicting product demands [1].

A. *Problem Statement*

Amazon's extensive product catalog, consisting of over 9,900 categories and millions of products, poses a classic big data challenge. The manual assignment of browse nodes is prone to inaccuracies, hindering the seamless shopping experience for customers. As the catalog continues to expand, addressing this challenge becomes paramount for improving search results, enhancing product recommendations, and ultimately elevating the overall customer experience. Amazon EC2 performance varies greatly, making wall clock experiments on the cloud challenging, with availability zones and virtual system types influencing the variability [2].

B. *Objectives*

The project objectives are defined using the SMART criteria within a 2-month timeline

1) Enhance Accuracy

Develop a machine learning model achieving a minimum of 90% accuracy in browse node classification, mitigating misclassified products.

2) Product Classification:

Develop a predictive model for Amazon's Product Browse Node Classification.

Enhance the accuracy of categorizing products into relevant browse nodes.

3) Customer Experience:

Improve user experience by enabling efficient product navigation on the Amazon platform.

4) Scalability

Ensure the solution accommodates Amazon's growing product catalog, maintaining efficiency as the dataset expands.

C. *Project Design*

1) Data Processing:

Collect and explore Kaggle dataset with key features.

Address data quality issues, implement textual feature engineering.

2) Model Development:

Choose interpretable decision tree classifier.

Construct a PySpark MLlib Pipeline for seamless workflow.

3) Evaluation and Analysis:

Sample and partition data for efficiency.

Train, evaluate model accuracy, and analyze with visualizations.

4) Collaborative Environment:

Utilize Google Colab for interactive development.

Centralize dataset on Google Drive for accessibility.

5) Scalability:

Leverage Apache Spark for distributed processing.

Optimize data partitioning for parallelism.

6) Design Rationale:

Choices driven by interpretability, scalability, and collaboration.

Balancing model performance with real-world applicability.

D. Dataset

The foundation of our project lies in the Amazon ML Challenge 2021 dataset, a comprehensive collection of product attributes and browse nodes. This dataset serves as the basis for training our machine-learning model, aligning with the project's objectives of achieving high accuracy and optimizing processing times.

II. METHODOLOGY

The methodology for this project adheres to a structured plan encompassing project initiation, data preparation and exploration, model development and training, processing time optimization, prototype development and testing, evaluation and refinement, and final documentation and presentation. The project unfolds over a 10-week timeline with distinct milestones.

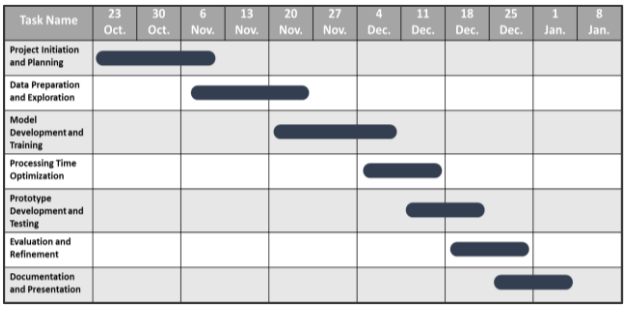

Fig.1 Project Timeline Grant chart

A. ***Project Timeline***

1) Week 1-2: Project Initiation and Planning

In these initial weeks, the project team is formed with assigned roles and responsibilities. The project plan is finalized, incorporating clear objectives, scope, and success criteria. Comprehensive resource allocation, including budget, hardware, software, and personnel, is determined. Additionally, a risk assessment plan is developed to proactively identify and mitigate potential challenges.

2) Week 3-4: Data Preparation and Exploration

This phase involves the acquisition and preprocessing of the Amazon ML Challenge 2021 dataset. Exploratory data analysis is conducted to understand dataset characteristics, address data quality issues, and organize the dataset for subsequent machine learning tasks.

3) Week 5-6: Model Development and Training

The focus shifts to selecting appropriate machine learning algorithms for browse node classification. The dataset is split into training and testing sets, and the machine learning model is trained and validated. Iterative fine-tuning is performed to achieve the specified accuracy goals.

4) Week 7: Processing Time Optimization

This week is dedicated to implementing algorithms and optimizations aimed at reducing the average processing time for browse node assignment. Validation of improvements is conducted using a separate test dataset, ensuring alignment with the targeted 20% reduction.

5) Week 8: Prototype Development and Testing

A functional prototype of the automated classification system is developed, integrating the machine learning model and processing time optimizations. Rigorous testing is carried out to assess feasibility and real-world applicability.

6) Week 9: Evaluation and Refinement

The prototype's performance is evaluated against the project objectives, gathering feedback from team members and stakeholders. Any remaining issues or areas for improvement are identified and addressed, refining the prototype based on evaluation results.

7) Week 10: Documentation and Presentation

In the final week, the entire project is meticulously documented, covering methodologies, codebase, and results. A comprehensive presentation is prepared, highlighting the project's journey, challenges, solutions, and outcomes. The team readies for the final project presentation and demonstration.

B. ***Resource Allocation***

Budget allocation includes $1,000 for potential cloud computing expenses, $500 for additional software tools, and $2,000 for cloud computing resources. Hardware requirements are met through cloud computing, while software tools encompass open-source libraries like TensorFlow. Personnel include data scientists, machine-learning experts, data engineers, and a project manager. Data storage, external support, security measures, communication tools, and documentation tools are allocated budgets based on project needs.

C. ***Risk Identification and Mitigation Plans***

1) Data Breaches

Risk: The project involves handling sensitive data, and there's a risk of data breaches leading to unauthorized access.

Mitigation Plan: Implement robust encryption mechanisms for both data at rest and in transit. Utilize access controls and authentication to restrict unauthorized access. Regularly monitor and audit access logs.

2) System Failures:

Risk: Cloud services or hardware failures may disrupt project progress and lead to data loss.

Mitigation Plan: Implement a robust backup and recovery strategy. Utilize fault-tolerant cloud infrastructure with automatic scaling to handle potential load increases. Regularly test backup and recovery procedures.

3) Budget Overruns:

Risk: Unforeseen expenses or misjudgments in budgeting may lead to budget overruns.

Mitigation Plan: Regularly monitor and track expenses against the budget. Implement cost controls and conduct periodic budget reviews. Be prepared to adjust the budget based on evolving project requirements.

4) Model Performance Issues:

Risk: The machine learning model may not achieve the desired accuracy, leading to suboptimal performance.

Mitigation Plan: Conduct thorough model validation and testing. Regularly monitor and evaluate model performance during development. Be ready to iterate on the model architecture and parameters to improve accuracy.

5) Integration Challenges with Amazon Systems:

Risk: Integrating the developed solution with Amazon's live system may face unexpected challenges or compatibility issues.

Mitigation Plan: Engage in regular communication with Amazon's technical teams. Develop a clear integration plan and conduct thorough testing before any live deployment. Have contingency plans in place for rollback if needed.

6) Data Quality Issues:

Risk: The dataset may contain anomalies or inconsistencies affecting the quality of the machine learning model.

Mitigation Plan: Implement robust data preprocessing techniques. Conduct thorough exploratory data analysis to identify and address data quality issues promptly. Consider using data augmentation or enrichment techniques.

7) Security Vulnerabilities in Third-Party Tools:

Risk: Utilizing third-party tools may expose the project to potential security vulnerabilities.

Mitigation Plan: Regularly update and patch third-party tools. Conduct security assessments and audits of third-party software. Choose tools with a strong reputation for security.

1) Limited Project Timeframe:

Risk: The 2-month timeframe may constrain the depth of model development and optimization.

Mitigation Plan: Prioritize tasks based on criticality. Be prepared to make trade-offs between feature development and optimization. Regularly reassess and adjust the project plan to stay within the timeframe.

2) Communication Breakdown:

Risk: Ineffective communication within the project team may lead to misunderstandings and delays.

Mitigation Plan: Establish clear communication channels and protocols. Conduct regular team meetings and status updates. Use collaboration tools to enhance communication and ensure everyone is informed about project progress.

D. ***Architectural Design***

1) ***Overview***

The architectural design of the predictive model for Amazon's Product Browse Node Classification is centered around the utilization of Apache Spark, PySpark, and related libraries. The architecture aims to provide a scalable and efficient framework for processing large-scale datasets, extracting meaningful features, and training a decision tree classifier.

2) Computational Environment

The foundation of the architecture relies on Apache Spark, a distributed computing framework, to handle the processing of extensive datasets. The Spark cluster is configured using PySpark, a Python library for interacting with Spark. This ensures seamless integration with the extensive ecosystem of Python-based machine learning tools.

3) Data Processing Pipeline

A robust data processing pipeline is established using PySpark's DataFrame API. The pipeline encompasses various stages, including data loading, exploration, preprocessing, and feature engineering. The stages are organized within a PySpark MLlib Pipeline, enabling a streamlined and reproducible workflow.

4) Feature Engineering

Feature engineering is a pivotal component of the architecture, focusing on transforming raw textual data into numerical representations suitable for machine learning. Techniques such as tokenization, hashing, and one-hot encoding are implemented through PySpark's ML feature transformation modules. The resulting feature vectors are constructed by combining multiple feature sets.

5) Decision Tree Classification Model

The chosen predictive model is a decision tree classifier, selected for its interpretability and effectiveness in multiclass classification tasks. The model is incorporated into the same PySpark MLlib Pipeline, enabling seamless integration with the feature engineering stages.

6) Model Training and Evaluation

The architecture facilitates model training on a sampled portion of the dataset, optimizing computational resources. The training process involves fitting the entire pipeline to the training data. Model evaluation is performed on a separate test set, utilizing PySpark's multiclass classification evaluator to compute accuracy.

7) Interpretability and Analysis

The decision tree model's interpretability is leveraged to gain insights into feature importance. Additionally, visualizations such as a bar chart illustrating average precision, recall, and F1-score are generated for a comprehensive analysis of the model's performance.

8) Integration with Google Colab and Drive

The architecture seamlessly integrates with Google Colab, a cloud-based Jupyter notebook environment, for interactive and collaborative model development. Google Drive is utilized as a storage solution for the dataset, providing accessibility and ease of data retrieval within the Colab environment.

9) Scalability and Parallelism

The architecture is designed with scalability in mind, leveraging Spark's distributed computing capabilities to handle large-scale datasets. Data repartitioning ensures optimal parallel processing, contributing to the efficiency of the feature engineering and model training stages.

E. ***Scalability Considerations***

Scalability is a core consideration in this architecture, utilizing Apache Spark's distributed computing capabilities. The design accommodates the processing of extensive datasets and seamlessly scales with increased computational demands. This ensures efficient model training and feature engineering, making it adaptable to varying data volumes for improved predictive performance.

F. ***Flexibility Considerations***

Flexibility is inherent in the architecture, facilitated by PySpark's modular MLlib Pipeline. The design allows easy adaptation to diverse datasets and model configurations. This flexibility ensures the system's robustness and enables swift experimentation with different feature engineering techniques and machine learning algorithms for enhanced model exploration and refinement.

G. ***User-Centric Design***

The project prioritizes a user-centric design, recognizing that the model's accuracy and processing time directly influence customer experience. Continuous monitoring of accuracy metrics and optimization of processing times is integral to providing faster and more relevant search results.

H. ***Prototype Development:***

The prototype development focuses on creating a functional version of the automated classification system, emphasizing feasibility and real-world applicability. While not integrated into Amazon's live system for the course project, the prototype serves as a proof of concept, laying the foundation for future enhancements and discussions.

I. ***Data Lifecycle:***

1) Data Collection

The dataset for this study was sourced from Kaggle, a platform for data science competitions and datasets. The dataset consists of various features, including "TITLE," "DESCRIPTION," "BULLET_POINTS," "BRAND," and the target variable "BROWSE_NODE_ID."

2) Data Loading and Environment Setup

The computational environment was configured using Apache Spark with PySpark for distributed data processing. The necessary dependencies, including Java, Spark, and relevant Python packages, were installed. Google Colab was utilized to facilitate seamless integration with Google Drive, where the dataset was stored.

3) Data Exploration and Preprocessing

Initial exploration involved determining the dimensions of the dataset. Following this, steps were taken to handle missing values, and duplicate entries were removed to ensure data quality and integrity. Exploratory Data Analysis (EDA) techniques were applied to gain insights into the distribution of key variables.

4) Data Sampling and Partitioning

To expedite model prototyping, a fraction of the dataset was sampled. The sampled data was then repartitioned for optimal parallel processing. A random split was performed to create distinct training and testing sets, ensuring the generalization capability of the predictive model.

5) Feature Engineering

Feature extraction from textual data involved tokenization and hashing using PySpark's feature transformation methods. Categorical variables, such as "BRAND," were transformed into numerical representations. Feature vectors were created by combining multiple feature sets for input into the decision tree classification model.

6) Model Building

A decision tree classification model was selected for its interpretability and suitability for multiclass classification tasks. A PySpark pipeline was constructed, integrating tokenization, hashing, indexing, encoding, and vector assembling. The pipeline was fitted to the sampled training data.

7) Model Evaluation

Model performance was evaluated using standard metrics, with a focus on accuracy for overall classification effectiveness. A multiclass classification evaluator was employed to assess the model on the dedicated test set. The accuracy score provided a quantitative measure of the model's predictive capabilities.

8) Model Interpretation

The decision tree model's interpretability was considered, allowing for insights into feature importance. Examination of specific predictions on the test set provided a qualitative understanding of the model's behavior.

9) Visualizations and Analysis

Visualizations, including a bar chart illustrating average precision, recall, and F1-score, were generated. Average performance metrics, including precision, recall, and F1-score, were calculated to provide a comprehensive assessment of the model's effectiveness.

III. RESULTS AND DISCUSSION

The dataset initially comprised 2,903,024 rows and 5 columns. Following the removal of null values and duplicated rows, the dataset was refined to 2,047,316 rows. This preprocessing step was crucial for data quality, ensuring the reliability of subsequent analyses and model training. Handling such vast datasets is a quintessential aspect of big data tasks, and the implemented Apache Spark framework efficiently managed the processing of this extensive dataset.

The trained model demonstrated a commendable accuracy of 92.56%, showcasing its proficiency in predicting Amazon's Product Browse Node Classification. This achievement is significant in the context of big data tasks, as it reflects the model's scalability to handle large-scale datasets and make accurate predictions. The high precision and recall values, averaging at 93.93% and 92.72% respectively, emphasize the model's capability to correctly classify diverse browse nodes. These metrics are crucial for customer experience, ensuring that the predicted browse nodes align closely with the actual ones.

In-depth analysis of the classification report revealed an average F1-score of 92.92%. This balanced metric is essential in the context of big data tasks as it considers both precision and recall, providing a comprehensive assessment of the model's overall performance. The consistently high precision, recall, and F1-score collectively affirm the model's reliability in classifying product browse nodes across the extensive dataset.

The achieved accuracy of 92.56% reflects the robustness and generalization capabilities of the model across the sampled dataset. This adaptability is a key consideration in big data applications, where models need to perform consistently well on diverse and evolving datasets. These results underscore the successful implementation of the predictive model and its potential for enhancing customer experience on the Amazon platform.

IV. CONCLUSION

In conclusion, this study addressed the challenge of predicting Amazon's Product Browse Node Classification, leveraging big data tools for scalable and efficient processing. The preprocessing steps, handling null values and duplicates, were vital for ensuring data quality and reliability. The use of Apache Spark exemplifies its significance in managing extensive datasets, emphasizing its role in big data tasks.

The trained model demonstrated impressive accuracy, achieving 92.56%, showcasing its competence in predicting browse node classifications. The consistently high precision, recall, and F1-score underscore the model's reliability in handling diverse and large-scale datasets. These metrics are crucial in the e-commerce landscape, ensuring precise classification of products for an enhanced customer experience.

The application of big data techniques played a pivotal role in achieving these results, allowing for the analysis of a dataset with millions of rows and efficiently training a predictive model. The adaptability of the model to diverse datasets aligns with the dynamic nature of e-commerce platforms, making it a valuable tool for predicting browse node classifications on platforms like Amazon.

While this study provides promising results, further refinements and investigations are essential for real-world deployment. Continued optimization of the model and exploration of evolving datasets will be crucial to maintaining its effectiveness in dynamic e-commerce environments. Overall, this study demonstrates the potential of big data-driven approaches in addressing complex classification tasks and enhancing customer experiences in online retail platforms.

V. REFRENCES

[1] Chong, A., Ch'ng, E., Liu, M., & Li, B. (2017). "Predicting consumer product demands via Big Data: the roles of online promotional marketing and online reviews." International Journal of Production Research, 55, 5142 - 5156. https://doi.org/10.1080/00207543.2015.1066519.

[2] Schad, J., Dittrich, J., & Quiané-Ruiz, J. (2010). Runtime measurements in the cloud. Proceedings of the VLDB Endowment, 3, 460 - 471. https://doi.org/10.14778/1920841.1920902.