EXECUTIVE SUMMARY

Objective

This project aims to enhance the customer experience on AB InBev's B2B e-commerce app, BEES, by implementing a robust product ranking framework. Our primary goal was to provide users with personalized and dynamic product recommendations, thus improving user experience and driving sales.

Methods

We developed a model using a comprehensive dataset, including customer order history, customer and product metadata, and user-app interaction data for customers in Paraguay. The initial models focused on two types of product recommendations: 'suggested next order' and 'upsell'. The former simplifies reordering by predicting customer needs based on past purchases, while the latter encourages customers to explore and purchase a broader range of products.

Various classification models were tested, including interaction data, and were surprisingly unable to surpass the performance of models without interaction data. Moreover, the team tested AB InBev's ranking algorithm using LightFM, logistic regression, and LightGBM, with LightGBM emerging as the most effective. This model effectively ranked products for each customer using their preferences, historical interactions, and contextual information.

Results

The framework's performance was measured using precision metrics. The model showed improved precision in the top three and ten ranked items in the suggested order category. The upsell model displayed even more significant improvements, with substantial increases in precision for the first-ranked item and the top three and ten items.

By offering more relevant and user-specific product recommendations, we anticipate heightened user engagement, customer loyalty, and order frequency. Future efforts should focus on refining existing models and exploring precise yet computationally heavy classification and ranking models on larger kernels.

AB InBev: Optimal Ranking Framework for Personalized Product Recommendations on B2B E-Commerce Platform

Team Supermodels - Iris Brook, Pranav Girish, Emily Hahn, Mackenzie Lees

15.572- Analytics Lab

November 30, 2023

ABSTRACT

This paper presents a comprehensive study on developing and implementing a product ranking framework within AB InBev's B2B e-commerce app, BEES. The primary objective of this project was to enhance the user experience by delivering personalized and dynamic product recommendations.

Our models use an extensive dataset encompassing customer order history, customer metadata, product metadata, and user-app interaction data to improve product recommendations in two categories: suggested next order and upsell. The suggested next order recommendation aims to facilitate the reordering process for users by recommending products based on past orders, making it convenient for them to reorder routinely purchased items or items typically bought at specific dates or times. The upsell recommendation seeks to diversify users' purchases by suggesting products they have not yet bought or have not bought in a while. This strategy aims to introduce users to a broader range of products and boost overall sales and profit.

To accomplish this, we experimented with various classification models, including Random Forest and XGBoost. We ultimately focused on improving AB InBev's ranking algorithm by exploring LightFM, logistic regression, and LightGBM. Our most successful model was the LightGBM ranking algorithm, leveraging user preferences, historical interactions, and contextual information to rank each customer's products.

We assessed our framework's performance using the precision metrics at various levels for the suggested order and upsell categories. Specifically, we evaluated the precision for the first-ranked item, the combined precision for the top three items, and the combined precision for the top ten items.

Our model demonstrated a solid performance in the suggested order category, showcasing improvements in the top three and top ten cumulative precisions by approximately two percentage points. The results were even more promising for the upsell model, with significant enhancements. The first-ranked item precision improved by eight percentage points, while the combined precisions for the top three and top ten items improved by five and three percentage points, respectively.

INTRODUCTION

In the fast-evolving e-commerce landscape, personalized recommendations are imperative to improve customer engagement and satisfaction. Both consumers and vendors benefit greatly from tailored experiences that cater to users' unique preferences and needs. Recognizing this potential, AB InBev, the largest multinational brewing company, has embarked on a project to enhance the user experience within its B2B e-commerce app, BEES.

The BEES app is an essential platform connecting AB InBev with its valued customers, with a diverse portfolio of over 500+ brands and 2,000+ products in any given country. However, until now, the product recommendations have yet to account for each customer's behaviors within the app and are primarily based on a user's purchase history.

Recognizing the potential of a more sophisticated and personalized approach to product recommendations, AB InBev sought to explore its untapped customer-specific app behavior data, including instances of product views, additions and removals to the cart, and other insights. By doing so, AB InBev aims to refine its product recommendation within BEES, ensuring each customer has a curated selection of products that align with their unique preferences and requirements. Furthermore, this new approach may add more variety to future purchases.

The primary objective of this project is to construct a robust product ranking framework that creates more relevant and personalized recommendations within the BEES app. This framework seeks to leverage diverse modeling approaches that account for user preferences, historical interactions, and contextual information to rank each customer's products. The term "ranking" here pertains to the order in which recommendations appear on the page, with the most relevant and engaging products positioned at the top. This strategic placement is pivotal in influencing whether customers accept the recommended products, making it a critical aspect of the e-commerce experience.

This paper explores our team's journey in elevating the BEES app by implementing our new product ranking framework. Through advanced data analytics, machine learning, and a commitment to enhancing the customer experience, this project strives to produce a personalized recommendation engine that redefines the user experience within BEES.

METHODS

Data Preprocessing and Exploration

Our initial dataset encompassed various aspects of customer app interactions, product details, and transactional records. This dataset combined information from several structured tables, each offering unique insights into different facets of the business operation.

The invoice table includes detailed information about direct sales transactions, such as gross and net revenues, quantities sold, invoice dates, and customer and product identifiers. By analyzing this data, we could grasp the financial aspects of transactions and identify frequently ordered products.

The customer details table provides comprehensive customer information, including geographic location, customer segments, and credit limits. Understanding customer segments, such as distinguishing between wholesalers and bars, was crucial in tailoring recommendations based on customer types.

The product details table provides critical insights into the range of products. This table sheds light on product categories, brands, packaging types (e.g., bottles vs cans), and volumes. The diversity in product specifications allowed us to explore preferences across different customer segments.

We also used tables detailing customer interactions on the BEES app. These tables include data on completed BEES orders, events of product details viewed, events where customers added and removed products from the cart, and instances where customers viewed product lists. These tables were pivotal in understanding customer engagement, providing a quantifiable measure of interest and intent through counts of various interactions. During our exploratory data analysis, we realized that the counts of interactions were key predictors for our model as they offered a direct measure of customer engagement and interest in specific products. Additionally, the nuances of product information, such as packaging type and product category, emerged as significant factors. These insights were instrumental in constructing a more nuanced recommendation system that goes beyond mere transactional data, incorporating behavioral and preferential aspects.

Random Forest Models

Our first initiative was to develop a model that could accurately predict customer purchasing behavior based on previous interaction data. We opted for a Random Forest model because of its effectiveness in handling complex, real-world data, its ability to process both categorical and continuous inputs, and its robustness in preventing overfitting.

We combined multiple data sources, aligning them based on shared keys, including customer, product, week number (week 0 representing the earliest week in our dataset), and day of the week. This alignment was critical for aggregating order counts, product interactions, and customer demographics into a unified dataset, forming the foundation for subsequent analyses.

To manage the scale of our data, we applied a 1% sampling strategy for initial testing. We introduced a binary 'label' column to indicate the occurrence of a purchase on the BEES app. Subsequently, we constructed a pipeline that integrated string indexing, feature assembly, and the Random Forest algorithm. This pipeline facilitated the transformation of raw data into a suitable format for our model. Following this, we divided our dataset randomly into training and test sets, with an 80-20 split.

Despite the robust pipeline and the theoretical strengths of the RandomForestClassifier, our team discovered several areas for improvement. The primary issue lay in our predictions' inadequate precision and recall, indicating a need to generalize learned patterns to new data more accurately. This problem persisted despite various feature combinations and hyperparameter tuning efforts. The problem was likely attributed to data leakage within the model, specifically related to collinearity among the features.

After our initial setback, we pivoted our model with a strategic reconfiguration of the dataset. We revisited our data aggregation process, focusing on a more dynamic representation of customer-product interactions. Each product-customer combination and stock-keeping unit (SKU) was analyzed through various interactions, including orders, product details views, and additions and removals from the cart. A key innovation within this pivot was implementing a 30-day rolling sum for these metrics within a moving window, capturing evolving trends and patterns in customer behavior. This approach enriched our dataset, offering a multifaceted view of customer interactions with products over time.

We once again divided our dataset through an 80-20 split for training and testing, but this time, we did so with a time split, ensuring a balanced representation for model assessment. Model optimization began with a meticulous grid search, exploring various tree numbers and depth configurations, complemented by a five-fold cross-validation. This detailed approach enabled us to fine-tune the model, optimizing its predictive accuracy.

Post-training, we focused on refining the model's classification efficacy. We recalculated probability thresholds, considering the predicted probabilities' average and standard deviation. This recalibration allowed us to set a more informed threshold, sharpening our classification results. The final predictions, categorized using this adjusted threshold, demonstrated improved class differentiation.

Leveraging the model's predictions, we developed a ranking system for the SKUs for each customer based on the likelihood of purchase. This ranking, informed by the refined Random Forest model, could enable AB InBev to make more tailored marketing efforts and announced inventory management, pivoting on the predicted customer engagement with specific products. However, this model failed to accurately generalize the learned patterns to the new data, indicating a deficient number of recommended products per customer. This shortcoming was due to a relatively large class imbalance (96% negative to 4% positive) within the model's setup, as most customers did not place frequent orders for their items.

LightFM Ranking Algorithm

Following our difficulties in creating a classification model, we decided to pivot towards creating a ranking algorithm that would bypass making predictions to rank and instead would output a ranking directly. Building upon our enhanced data preparation efforts, we focused on developing sophisticated ranking algorithms. This transition marked a significant step in our project, aiming to leverage predictive modeling for generating actionable insights in ranking customer-product interactions.

We began with a LightFM model, an advanced hybrid recommendation algorithm combining user-item interaction data and content features. Our dataset was segmented into training and testing sets using a split date set one month before the current date, thus allowing the model to be trained on historical data and validated against recent trends.

Our implementation of LightFM utilized a learning rate of 0.05 and the adoption of the WARP loss function. The WARP (weighted approximate rank pairwise) loss function aimed to optimize the ranking of items in recommendation systems, emphasizing the accuracy of the top recommendations. It dynamically samples negative examples and calculates loss based on the number of tries it takes to find a mis-ranked negative example, prioritizing corrections in the higher ranks of the recommendation list. The training phase, encompassing ten epochs, was followed by a rigorous evaluation process. We employed precision and AUC metrics to assess the model's performance, scrutinizing its effectiveness across the training and testing datasets.

The problems faced while implementing this model using our dataset are twofold. Firstly, the size of our dataset made it very computationally intensive to generate the sparse matrix of customer product combinations required as an input for the LightFM model. Secondly, the LightFM architecture is such that it intrinsically considers similarities between users (in our case, customers), and due to this, the ranks produced include implicit feedback. This structure could be favorable, but the company directed us to focus solely on prediction without considering similarities between customers to make the model more generalizable to different regions. For these reasons, we shifted our focus to a more lightweight but accurate framework for ranking that can rank products using interaction data for a relevance score: LightGBM.

LightGBM Ranking Algorithm

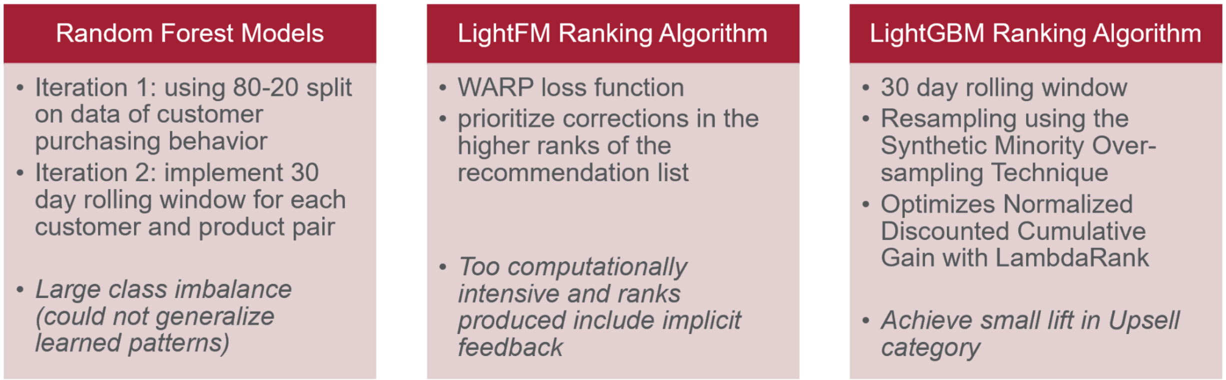
Upon further research, we found that LightGBM, a gradient-boosting framework, significantly outperformed LightFM in the context of our model. Furthermore, LightGBM tended to run faster than LightFM, could handle deep learning architectures such as PyTorch and TensorFlow, and utilized a wider variety of hyperparameters. A critical aspect of our methodology was the meticulous feature engineering and transformation of the dataset. This process involved creating a comprehensive framework that captured various dimensions of customer interactions. These dimensions included product addition and removal from the cart, product views, and completed orders. These interactions were quantified over a specified historical period, offering a granular view of customer behavior and preferences. We then aggregated and transformed the data via filtering and joining. These actions resulted in a multidimensional dataset that formed the backbone of our predictive modeling.

For the development of the predictive model, the dataset was divided into training and testing sets based on specific time frames, as we did in the LightFM model. The training data underwent a crucial resampling step using the Synthetic Minority Over-sampling Technique (SMOTE), addressing the challenge of class imbalance—a common issue in recommendation systems where the number of non-purchased products vastly exceeded the purchased ones. The features were then standardized using the StandardScaler, ensuring that all variables contributed equally to the model's prediction.

Two distinct modeling approaches were employed to achieve our final ranking algorithm: Logistic Regression and LightGBM. Through Logistic Regression, a foundational statistical analysis, and a machine learning model, we predicted an individual score for each POC ID and SKU pairing's likelihood of purchase. Using LightGBM, a gradient-boosting framework, we collected the scores from the Logistic Regression and output rankings for each POC ID and SKU pairing. The LightGBM model was trained using the *'lambdarank'* objective and the *'ndcg'* metric, optimizing for the quality of the ranked order of products. LambdaRank is a ranking algorithm optimizing the Normalized Discounted Cumulative Gain (NDCG), a performance measure for ranking models that evaluates the gain of item placements discounted at lower ranks, ensuring the relevance of top-ranked items.

After training the model, the predictions were used to rank the products for each customer. The ranking was determined based on the model's confidence in the product's relevance to the customer, ensuring that the most pertinent products were positioned at the top of the recommendation list. We further refined this ranked list by applying a scoring system that scaled the model's output to a more interpretable range. The scores were normalized and mapped to a predefined range, facilitating easier comprehension and application in a business context.

Our ranking framework extended its application to suggested order and upsell product recommendations, allowing us to derive AB InBev's current product recommendations for each customer as its initial input. This comprehensive approach enhanced the quality of product recommendations, contributing significantly to our project's success.



Random Forest Models

- Iteration 1: using 80-20 split on data of customer purchasing behavior
- Iteration 2: implement 30 day rolling window for each customer and product pair
- *Large class imbalance (could not generalize learned patterns)*

LightFM Ranking Algorithm

- WARP loss function
- prioritize corrections in the higher ranks of the recommendation list
- *Too computationally intensive and ranks produced include implicit feedback*

LightGBM Ranking Algorithm

- 30 day rolling window
- Resampling using the Synthetic Minority Over-sampling Technique
- Optimizes Normalized Discounted Cumulative Gain with LambdaRank
- *Achieve small lift in Upsell category*

Chart 1: Summary of Methods

RESULTS

In our comprehensive exploration to refine predictive modeling techniques, we developed two distinct Random Forest models and experimented with two separate ranking algorithms. This iterative process of trial and adaptation proved to be a cornerstone in our steps towards enhancing prediction accuracy. Our final model uses LightGBM architecture for learning-to-rank from a generated set of recommendations obtained from a random forest classifier. We achieved results that showed improved model performance, effectively utilizing customer app behaviour data in conjunction with historical purchase data. Our algorithm led to a 1 percentage point increase in overall precision and a 3 percentage point increase in recall for the suggested order category. For the upsell category, we observed Mean Average Precision (MAP) values of approximately 20% at rank 1 and 12% at rank 3 for the LightGBM, a substantial enhancement over the current implementation, which scored around 14% at rank 1 and 7% at rank 3.

Date	**Current Overall Precision**	***Our Overall Precision***	**Current Rank 3 Precision**	***Our Rank 3 Precision***
10/15	0.405	*0.418*	0.446	*0.467*
10/16	0.3998	*0.412*	0.435	*0.448*
10/18	0.3967	*0.411*	0.437	*0.461*
10/19	0.371	*0.383*	0.414	*0.431*

Table 1: Suggested Order Product Ranking Metrics

Date	**Current MAP at Rank 1**	***Our MAP at Rank 1***	**Current MAP at Rank 3**	***Our MAP at Rank 3***
10/15	0.1297	*0.203*	0.065	*0.114*
10/16	0.133	*0.219*	0.069	*0.123*
10/18	0.14	*0.213*	0.071	*0.121*
10/19	0.144	*0.211*	0.068	*0.115*

Table 2: Upsell Product Ranking Metrics

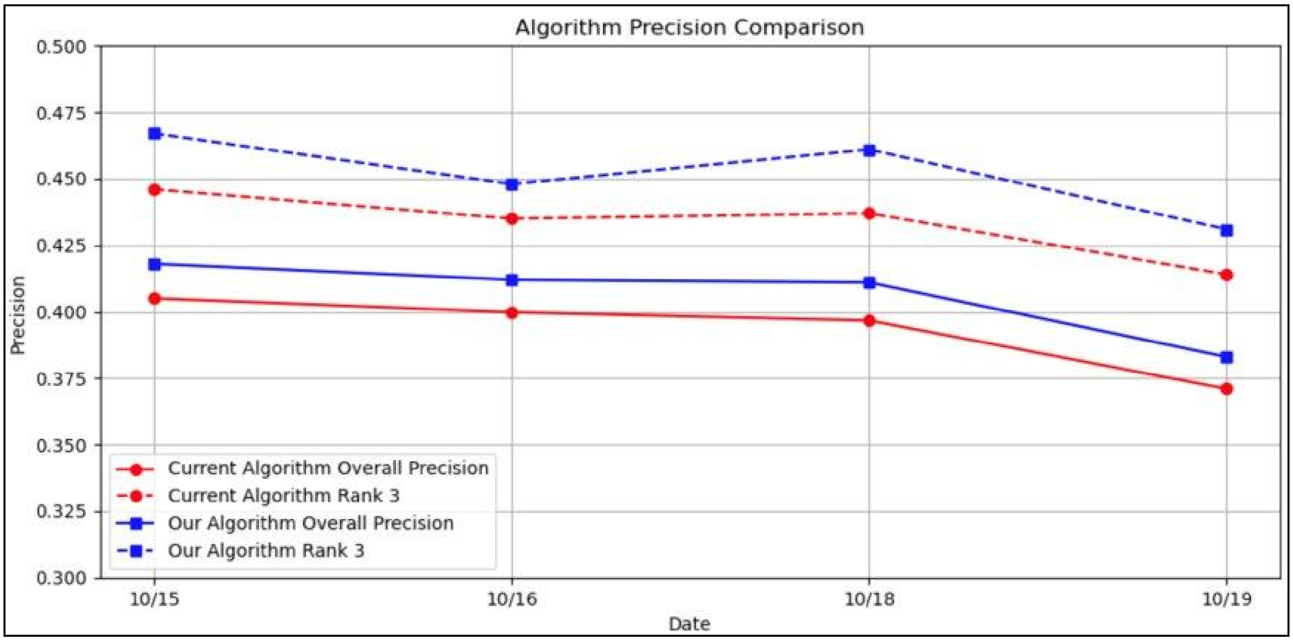


Figure 1: Plot of Suggested Order Product Ranking Metrics

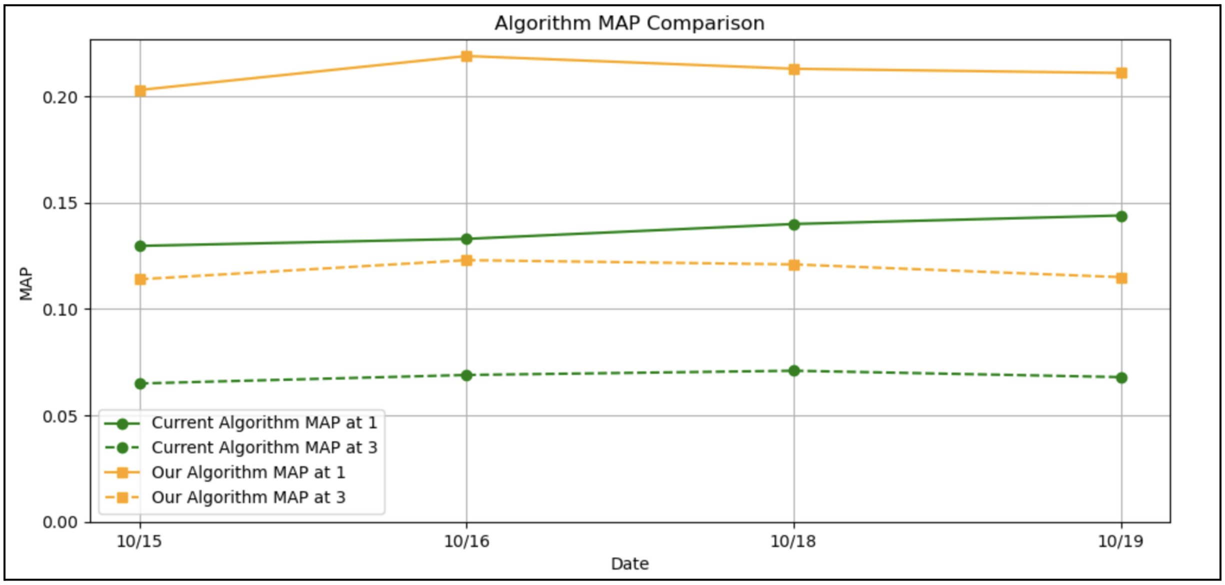


Figure 2: Plot of Upsell Product Ranking Metrics

DISCUSSION

In reflecting upon our results, it is evident that AB InBev's existing model is already operating at a high level of optimization. Our efforts, while comprehensive, resulted in marginal improvements, underscoring the current model's advanced design and the challenges inherent in significantly enhancing such a sophisticated system.

In the future, AB InBev should launch an A/B testing experiment with an enhanced model that utilizes customer behavior data. This experiment may provide valuable insights into the impacts of these improved recommendations on customer engagement and sales diversity.

Additionally, we recommend a strategy that balances model enhancement with operational efficiency. We suggest optimizing the existing model's efficiency and exploring innovative feature engineering techniques. Future endeavors should also include developing a more streamlined data pipeline and incorporating real-time data processing, enhancing the model's responsiveness and precision.

Our project represents a significant step in advancing AB InBev's product recommendation capabilities within the BEES app. Our journey underscores the importance of continuous innovation and optimization in the dynamic world of e-commerce, aiming to deliver a more personalized and effective user experience.