CSSCOMMERCE

MARK’S PLUMBING

Enhancing Search Experience and Sales Performance for Mark's Plumbing using Solr, Perfion, and Process Mining

Mark's Plumbing is a major B2B and B2C plumbing supplier in the United States, with over 1 million products available online. They faced challenges related to slow search and query speeds, sub-optimal search results, and high rates of no-results.
To address these challenges, CSS Commerce implemented a unified search engine using Solr, integrated custom databases and Perfion (PIM) with MSFT GP (GABI), and employed process mining concepts for continuous improvement.
This case study provides a detailed account of the solutions implemented, their impacts, and the overall success of the project.

Mark's

CSSCOMMERCE

TECHNICAL DETAILS

1. Unified Search Engine with Solr:

Solr was used as a powerful, open-source search engine to handle the large scale of Mark's Plumbing's product catalog. Solr helped optimize search speed and relevance, funneling search activities from multiple entry points into a single engine. This enabled a consistent search experience across the platform, from predictive search results to layered navigation and product detail pages. With Solr, no-results cases were significantly reduced, and query response times were shortened to mere seconds or fractions of a second.

2. Integration with Perfion (PIM) and MSFT GP (GABI):

Mark's Plumbing used Perfion, a Product Information Management (PIM) system, to manage product data across multiple channels. This PIM was integrated with their existing Microsoft Dynamics GP (Great Plains) system, which managed the financial and inventory aspects of the business.
The integration allowed seamless communication between the two systems, ensuring accurate product information and pricing in real-time. Solr was then configured to index data from Perfion, allowing for comprehensive search results that included product attributes, availability, and pricing.

3. Custom Database Integration:

Mark's Plumbing had several custom databases containing valuable information, such as product specifications, and manufacturer information. These databases were integrated into the Solr search engine to enrich search results with relevant data. This was accomplished by creating custom adapters to extract data from these sources and transform it into a format compatible with Solr.

MACRO PROCESS

a) Data Extraction

Custom scripts were written to extract data from the various custom databases. These scripts were responsible for pulling out the required information and transforming it into an appropriate format. This included converting data types, handling special characters, and formatting text data.

b) Data Transformation

Once the data was extracted, it needed to be transformed into a format compatible with Solr.

This involved several steps, including:

i. Schema Definition: A Solr schema was designed to accommodate the data from the custom databases. The schema defined the fields, data types, and indexing properties for each piece of information. It also specified how the data should be tokenized, filtered, and analyzed during the indexing process.

ii. Field Mapping: The extracted data from the custom databases was mapped to the appropriate fields in the Solr schema. This involved converting field names, matching data types, and ensuring that the data was stored in the correct format.

iii. Data Normalization: The extracted data was normalized to ensure consistency across records. This involved standardizing date formats, resolving inconsistencies in product names and descriptions, and reconciling duplicates.

iv. Data Enrichment: Additional metadata and attributes were added to the records in order to improve search relevancy and provide users with more context. For example, product records were enriched with manufacturer information.

c) Data Ingestion

Once the data was transformed into the appropriate format, it was ingested into the Solr search engine.

This process involved several steps:

i. Batch Processing: Data was ingested into Solr in batches, to ensure efficient indexing and to minimize the impact on the search engine's performance. Batch sizes were determined based on factors such as data volume, indexing speed, and server resources. erformance and reliability.

ii. Incremental Updates: To keep the search engine up-to-date with the latest information, incremental updates were performed on a regular basis. This involved identifying new or modified records in the custom databases, transforming them according to the Solr schema, and ingesting them into the search engine.

iii. Error Handling: During the ingestion process, any errors or issues that arose were logged and handled appropriately. This included identifying and resolving issues with data quality, indexing errors, or server-related issues. Robust error handling ensured that the search engine remained stable and reliable.

iv. Monitoring and Optimization: After data ingestion, the performance of the Solr search engine was closely monitored to identify any bottlenecks or areas for improvement. This involved tracking key performance metrics, such as query response times, indexing speeds, and resource utilization. Based on this monitoring, optimizations were made to the Solr configuration, indexing strategies, and server resources to ensure optimal performance and reliability.

d) Search Interface

With the Solr search engine in place and data ingested, the next step was to develop a search interface for users to access the knowledge repository.

This involved several components:

i. User Interface Design: The search interface was designed with a focus on usability, accessibility, and responsiveness. Key design elements included a clean layout, intuitive navigation, and helpful search hints to guide users in finding relevant information. The interface was also designed to be compatible with different devices and screen sizes.

ii. Query Construction: To enable users to search the knowledge repository effectively, a query construction component was developed. This component allowed users to create complex search queries using a combination of keywords, filters, and sorting options.

iii. Search Results Presentation: Once a user submitted a search query, the Solr search engine returned a list of relevant records. The search interface was responsible for presenting these results to the user in a clear and organized manner. This involved displaying key metadata for each record, such as title, manufacturer, SKU #, providing options for users to refine or sort the search results.

iv. Pagination and Navigation: To improve the user experience and manage large result sets, the search interface implemented pagination and navigation features. This allowed users to easily browse through search results and navigate to specific pages or sections.

v. Analytics and Feedback: To continually improve the search interface and better understand user needs, analytics and feedback mechanisms were implemented. This included tracking user interactions with the interface, collecting feedback on the search experience, and analyzing search trends to identify areas for improvement or enhancement.

vi. Integration with External Tools and APIs: To expand the functionality of the search interface and encourage collaboration, it was designed to integrate with external tools and APIs such as process mining platforms.

vii. Ongoing Development and Support: As the knowledge repository continued to grow and evolve, so too did the search interface. Feature enhancements were identified to ensure that the interface remained robust, reliable, and up-to-date with the latest technologies and user expectations.

e. Process Mining for Search-to-Purchase Pathway Optimization

To further enhance the search interface's effectiveness and optimize customer experience, process mining techniques were employed to validate per customer cluster common search-to-purchase pathways. This allowed the identification of the most optimal path for each customer or customer group, based on their historical interactions with the platform.

Data extraction from the Solr history involved gathering information related to search queries, user interactions, and purchase transactions. This data was then analyzed using process mining algorithms, which mapped the user journey from the initial search query to the final purchase. By organizing the data by customer and group, the analysis could identify the most common and efficient pathways taken by users to reach a purchase decision.

Delta analytics were then employed to examine divergences fromthe most optimal path per customer or customer group. These analyses aimed to identify areas of improvement in the search interface and customer experience, as well as to provide personalized recommendations based on the user's unique search and purchase behavior.

By leveraging process mining and delta analytics, the search interface was further optimized to reduce the time it took for users to find and purchase their desired products or services. This not only led to increased customer satisfaction but also improved the overall efficiency and effectiveness of the platform.

f. Leveraging Google Analytics Data for Dynamic Search Result Weighting

In addition to the techniques discussed earlier, the search interface also utilized Google Analytics (GA) data to dynamically adjust search result weights for both low and high performing items in an ecommerce context.
This was achieved by incorporating GA ecommerce data into the search ranking process, providing a more tailored and optimized search experience for users.

By analyzing the GA ecommerce data, the system could identify high and low performing items based on key metrics such as conversion rate, revenue, and user engagement. This information was then used to adjust the weights assigned to search results, effectively boosting or lowering the search ranking of these items depending on their performance.

To ensure a seamless integration with the search term, the system matched high and low performing items with relevant search queries. This allowed for a dynamic and adaptive approach to search result ranking, ensuring that users were presented with the most relevant and high-performing items first, ultimately leading to increased conversion rates and overall platform performance.

By leveraging Google Analytics data, the search interface was able to deliver a more personalized and effective search experience for users.

IMPACT

The implementation of the advanced search interface, incorporating a range of innovative techniques and technologies, has resulted in significant improvements in user experience, search relevance, and overall platform performance. A brief summary of the key components and their impact is provided below:

Advanced Query Processing:

By employing sophisticated query processing techniques, including query expansion, synonym mapping, and NLP-based methods, the search interface has significantly enhanced the relevance and accuracy of search results, leading to greater user satisfaction and higher engagement.

Machine Learning for Ranking:

The integration of machine learning algorithms to train search result ranking models has enabled the search interface to deliver more personalized and contextually relevant results, ultimately driving higher conversion rates and revenue.

Search Interface Enhancements:

a. Faceted Navigation: The introduction of faceted navigation has enabled users to quickly and easily refine search results based on specific attributes, improving the overall search experience, and increasing the likelihood of successful product discovery.

b. Autocomplete and Query Suggestions: By providing users with real-time query suggestions and autocomplete functionality, the search interface has streamlined the search process, reducing user effort and promoting more efficient navigation.

c. Fuzzy Matching: The incorporation of fuzzy matching techniques has ensured that users are still presented with relevant search results, even when queries contain typos or misspellings, reducing search friction and enhancing user satisfaction.

d. Rich Search Results: The presentation of rich search results, including images and product details, has increased user engagement and trust, further driving platform success.

e. Process Mining for Customer Cluster Pathways: Utilizing process mining to validate per customer cluster common search-to-purchase pathways has allowed for the identification of optimal paths and the detection of divergences. By offering delta analytics, the platform can now address inefficiencies and guide users more effectively towards the most successful search-to-purchase journeys, enhancing user satisfaction and increasing conversion rates.

f. Dynamic Search Result Weighting: By leveraging Google Analytics data, the search interface has been able to adjust search result weights for low and high-performing items based on ecommerce performance data. This dynamic boosting or lowering of search rankings allows for greater visibility of high-performing items and a more tailored search experience for users.

Overall, the advanced search interface has significantly improved user experience, search relevance, and the overall effectiveness of the platform. By incorporating advanced query processing, search interface enhancements, process mining, and integration of Google Analytics data, the platform has achieved higher engagement, conversion rates, and revenue while providing users with a powerful, intuitive, and flexible tool to access the wealth of information stored within.