algolia

What it takes to build marketplace search

TABLE OF CONTENTS

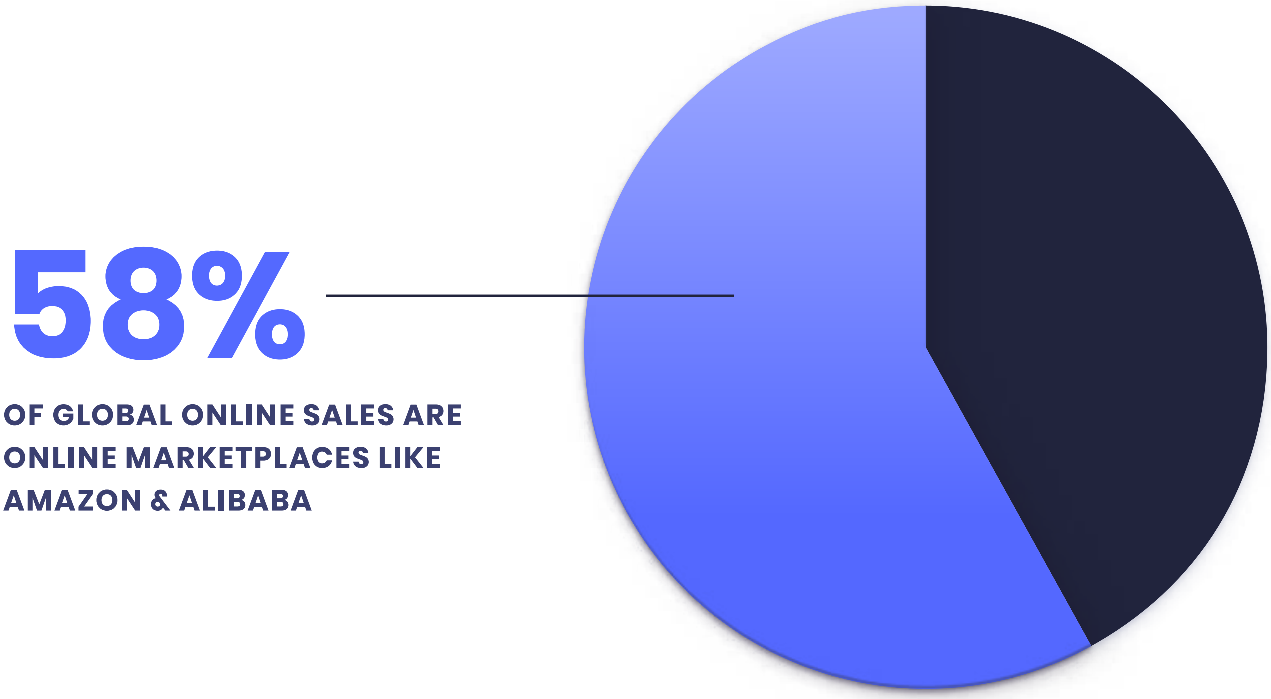
In 2019, online marketplaces accounted for 58% of global online sales.

This has led to the creation of a very competitive market that is already dominated by gigantic players such as Amazon and Alibaba. To remain viable despite low margins, marketplaces must quickly develop economies of scale via organic growth, as well as rapid internationalization and catalog expansion. A loyal customer base is also critical to keeping a competitive edge.

What creates that loyal customer base is a state-of-the-art experience on the digital properties.

Offering shoppers the best Search and Discovery experience across a marketplace's large and often diverse catalog is a key lever in differentiating from the competition.

In this guide, we cover the requirements for a great marketplace experience and what it takes to build one.



Requirements for a great marketplace Search and Discovery experience

INDEXING AT VERY LARGE SCALE

Players such as Newegg have more than 20 million active products in their catalogs; Etsy has more than 40m. Marketplaces' underlying systems must be robust enough to support such massive catalogs.

The first of these systems is the search engine. The challenges here are numerous:

- The total number of items to index
- The total size of the index
- The heterogeneity of catalogs for those selling a wide variety of items
- The large volume of transactions: items going out of stock, popularity changing rapidly, sellers modifying pricing

All that information needs to be promptly updated in the marketplace search experience, and therefore in the index, requiring above-average indexing speed.

SEARCH ENGINE CONSIDERATIONS

✓ NUMBER OF ITEMS TO INDEX

✓ CATALOG VARIETY

✓ NUMBER OF TRANSACTIONS

✓ TOTAL SIZE OF THE INDEX

ANSWERING QUERIES AT VERY LARGE SCALE

Once a marketplace's catalogs are indexed, the next challenge is to return relevant results to the shoppers.

This is very demanding for the search engine in at least two ways:

1. The search engine must return relevant results from a very large catalog.
2. Marketplaces usually have higher traffic than typical e-commerce websites, leading to very large volumes of search queries that the engine must manage.

In addition, satisfying Search and Discovery experiences require fast response time (down to a few milliseconds for search-as-you-type experiences), as well as performant typo tolerance, management of synonyms, and more.

SELLER MANAGEMENT

Managing sellers' requests and expectations makes marketplace search a unique challenge. To attract more sellers, especially at the early stages, marketplaces often allow sellers to list products more easily, which can unfortunately result in poor-quality product data.

To overcome potentially low data quality, a marketplace must prioritize good seller behavior in the relevance logic, such as:

— Fully completing product listings

— Having good shopper feedback on delivery and payment

— Quickly responding to user questions

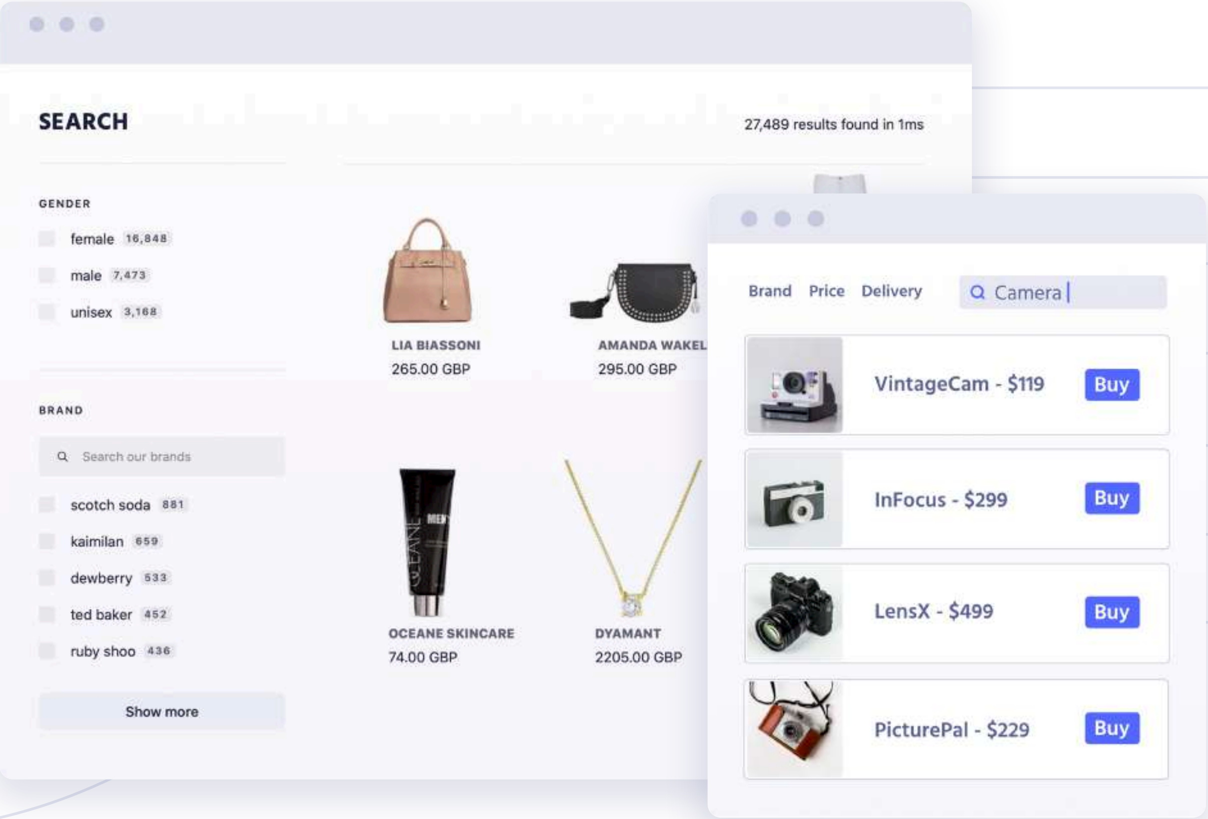
Along with the data quality and consistency that are key to driving relevance across the marketplace, various other features can be added to the seller back office, such as the ability to label/tag each product leveraging a global (cross-seller) tag index, leading to easier input for the seller (and even more consistency of the data).

EASE OF FRONT-END IMPLEMENTATION

Like any other system, a marketplace search engine is useless if it powers a simple search bar. It's meant to power the product Search and Discovery experiences on the marketplace's various front ends.

Marketplace search engines need to either power or easily integrate with advanced Search and Discovery scenarios — full search-result pages including dynamic faceting, content carousels, autocomplete dropdowns, and federated search.

They also need to provide a unified omnichannel experience across web, mobile web, and mobile applications.



ADVANCED RELEVANCE AND PERSONALIZATION

Each marketplace has its own unique catalog, audience, and business strategy:

- The ways marketplaces return products to their shoppers are unique to each of them, and can also be competitive advantages. Marketplace search engines must support customization of their relevance logic, using various signals such as product popularity, margins, or conversion rate.
- Unlike traditional retailers, marketplaces must manage several vendors for the same item, and determine how to expose each vendor based on their ratings, volumes, shipping terms, or even negotiated terms.

Today's consumers expect personalized experiences. In addition to supporting advanced relevance, a marketplace search engine must be able to:

- Personalize search results to each individual shopper
- Personalize a very large catalog, for a very large volume of queries, in a few milliseconds

CONTROL AND VISIBILITY OF THE SEARCH EXPERIENCE

Marketplaces have merchandising needs, such as executing on promotional campaigns, managing and benefiting from seasonality, and showcasing best-selling items. This implies providing nontechnical teams with the right tools to assess the performance of the Search and Discovery experience and merchandise it as necessary.

The marketplace search engine must support any merchandising logic, which means it must come with an interface that's easy for nontechnical teams to use, so they can make changes on the fly without requiring developer/IT time.

> “ Any marketplace search engine downtime translates into a loss of revenue

RELIABILITY, RESILIENCE, MAINTENANCE, AND SECURITY

Marketplace search experiences are often shoppers’ primary paths to products. Some marketplaces run sophisticated promotional campaigns using search, for example, treasure hunts that hide heavily discounted items in random categories or include them in search results for random queries. This drives huge spikes in search traffic.

As such, any marketplace search engine downtime translates into a loss of revenue. Search downtime is the immediate critical failure, of course, but a downtime in indexing that is not solved rapidly can also become critical.

Performance drops, while less consequential than downtime, can also affect the marketplace bottom line. Marketplace search engines must seamlessly manage challenges such as traffic surges from various geographic locations.

Finally, marketplace search engines must be protected from attacks attempting to take them down, alter their operation for malicious purposes, access sensitive data, or simply scrape the marketplace catalog

Building marketplace search with open source

PROVISION THESE RESOURCES ACCORDING TO YOUR NEEDS

✓ CPU

✓ NETWORK

✓ RAM

✓ DISK

In this section, we look at what it takes to build marketplace search functionality that fulfills the requirements listed above, using open-source solutions such as Elasticsearch or Apache Solr.

DIMENSION, PROVISION, DEPLOY, AND SCALE INFRASTRUCTURE

Whether you choose to deploy your marketplace search engine on bare metal servers or on your cloud provider’s virtual machines, the first thing you need to do is understand the architecture you’ll put in place, and accordingly dimension your infrastructure for the initial and future needs of your marketplace, while accounting for traffic and indexing spikes, backups, redundancy, and more.

Marketplace search engines require various resources for various scenarios. You must define how much your indexing and query strategies require CPU, RAM, network, and disk throughout the activity of your marketplace, and provision accordingly. All that while being mindful of optimizing operating costs so as to stay competitive with the fees you apply within your marketplace.

As your marketplace grows, so will your product catalog and the size of your index. You'll need to think about scaling. There are many options available with open-source search engines such as Elasticsearch. Do you want to scale horizontally or vertically, or use a mix? Keep in mind that every architecture decision has an impact on the performance of the different aspects of your search engine (query processing speed, indexing speed, relevance).

You must carefully define how to logically and physically split and allocate your index across your infrastructure. Solutions such as Elasticsearch offer the concepts of clusters, nodes, and shards, physical and/or logical subdivisions of an Elasticsearch instance indices that provide scaling flexibility. However, configuration and balance of those elements is very complex and impacts both performance and scalability. Elastic users are urged to closely monitor the data set being searched to ensure that the shard configurations are set up correctly.

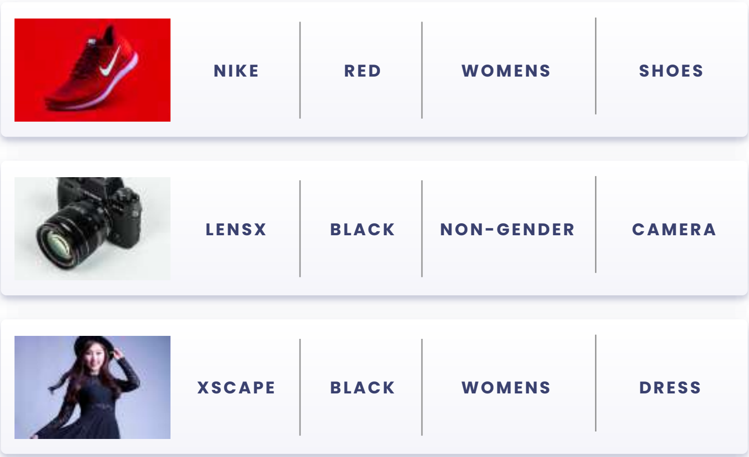
Additionally, if you have shoppers in multiple geographic locations, you need to geo-replicate your search instances to make sure that regardless of where they are, they can enjoy a fast Search and Discovery experience.

Finally, when you make your search infrastructure decisions, you must still deploy the search engine over the infrastructure, then configure both the engine and your deployment process to support geo-replication and scaling.

▶ **SKILLS REQUIRED: INFRASTRUCTURE, NETWORK, ELASTICSEARCH/SOLR**

PREPARE DATA AND CONFIGURE INDEXING

You now have a search engine running on your infrastructure, and you know how it will scale. The next step is to upload your marketplace catalog into it.



The first step is to collect and aggregate the data, which may be scattered across different systems (such as product information management, PIM; marketplace commerce platform, Digital Asset Management).

Then, it is critical to correctly structure the data. Marketplaces tend to have very diverse catalogs, including items that can have very different attributes. For instance, if you sell both laptops and apparel, you need to select the best data scheme to accommodate laptop screen sizes, CPU, and RAM, as well as clothing brands, colors, and sizes. This information is key for letting your shoppers search and filter, as well as critical for best performance and relevance.

Open-source engines use analyzers, which process the uploaded data before indexing it. Text analysis enables Elasticsearch to perform full-text search, which returns all relevant results rather than just exact matches. You can choose from multiple analyzers depending on parameters such as your data set, the language of your experience, and your desired end-user experience.

Next, you must define the best indexing strategy (e.g., regular batches, real-time updates) to keep the indexed data fresh while preserving your infrastructure and the performance of your search (indexing can be very CPU-intensive).

When you have a clear strategy and everything is ready, you'll send the data to the search engine via an API client or the REST API of your search engine, depending on the languages available.

▶ **SKILLS REQUIRED: ELASTICSEARCH/SOLR, BACK-END ENGINEERING**

CONFIGURE SEARCH

Frictionless search user experiences hinge on several factors:

— How relevant the search results are, regardless of which words the user enters and which typos they include

— How fast the search engine returns those results

— How easily the user can browse the results and refine their searches

Matching those requirements is what makes building search engines so complex. Let's break down the steps to creating great search for your marketplace with Elasticsearch.

— Enabling textual relevance. Elasticsearch provides features to handle typo tolerance (fuzzy search), synonyms, and other expected behaviors, offering varying support in different languages. Those features do not come out of the box; they require knowledge of the impact of activating them. Each feature has a performance cost, either at indexing time or at query time, which impacts relevance and scaling. It can become a never-

OPTIMUM SEARCH USER EXPERIENCE

✓ RELEVANCE

✓ SPEED

✓ USABILITY

ending job to balance the flexibility you give users when they enter queries with search performance, relevance, and scalability.

- Balancing flexibility with performance. The challenge is to balance the flexibility you give users when they enter queries with search performance, relevance, and scalability. User- or vendor-generated content such as product reviews and product descriptions can lead to errors and inconsistencies in the data users search through. This requires broadening typo tolerance and adding synonyms, but not too much, as you don't want to return a too-broad (irrelevant) result set, nor affect the performance too much.
- Fine-tuning textual relevance. It is likely that the products in your marketplace will have more than one attribute you want to make searchable. You will likely want shopper queries to match not only product names but brand names, product types, categories, descriptions, colors, or several of these at the same time. The complexity of marketplace catalogs combined with user-generated product listings will require you to constantly fine-tune how your textual relevance works. A

> **Configure your marketplace search engine so that a match at the beginning of a product description is more important than one at the end**

match on a product name will likely be more important than a match on a brand name, which is more important than a match on a description.

- In addition, it isn't unusual for vendors to use clever ways to make their products appear higher in marketplace search results, including adding popular keywords at the end of product descriptions. Therefore, you may want to configure your marketplace search engine so that a match at the beginning of a product description is more important than one at the end. Solutions like Elasticsearch offer various options and parameters to meet those requirements. The complexity comes in choosing the best options, understanding how they will interact with your other search engine configurations (such as typo tolerance), and knowing how they will impact search performance, as some of them can be very CPU intensive.

There are other important elements, such as providing search-as-you-type experiences and configuring highlighting and snippeting logic to help shoppers understand and navigate their results.

The bottom line: there are many considerations for building and configuring search that fits marketplace needs using open-source solutions such as Elasticsearch. The complexity lies in choosing the right features and solutions, and figuring out how each element will impact the others, as well as how each decision will impact search performance and relevance.

▶ **SKILLS REQUIRED: ELASTICSEARCH/SOLR**

IMPLEMENT ON YOUR FRONT ENDS

You now have a search engine up and running, capable of answering search queries. Next, you must connect your search API to your front end. And this can be a challenging task.

Most search engines, including open-source solutions such as Elasticsearch, return JSON answers to search queries. Very basic, static implementations can be as easy as sending user search queries to the engine once they're fully typed and formatting the JSON response. However, things can quickly get more complex when you want to power the types of experiences that help shoppers navigate large and diverse marketplace catalogs.

For example, you should be able to power search-as-you-type experiences (in which technically, each keystroke is a search query), add filters and facets to the experience, and power rich autocompletes with federated search (that could, for instance, include relevant product results, as well as brand names, categories, and query suggestions). The open-source community has built front-end libraries to ease the implementation of these types of Search and Discovery experiences, but none of them are officially supported by open-source search solutions.

Furthermore, once you've created your desktop marketplace search, you'll need to adapt the search experience for mobile web, and rebuild everything for your potential mobile applications, which have a special set of complex UX considerations.

▶ **SKILLS REQUIRED: ELASTICSEARCH/SOLR, FRONT-END ENGINEERING**

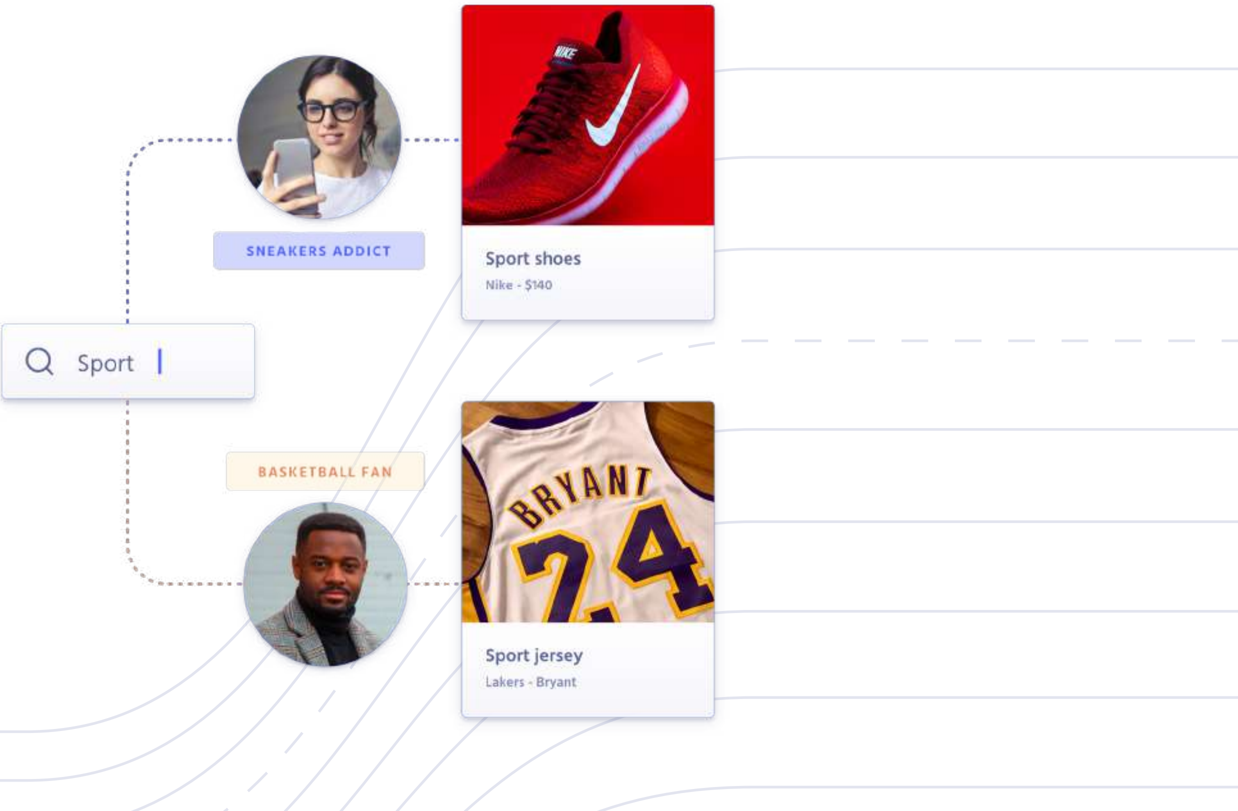
CONFIGURE RELEVANCE AND PERSONALIZATION

A textually relevant, functioning marketplace search engine is a good starting point. It returns iPhone cases when shoppers look for iPhone cases. That's your basic textual relevance.

Now, how do you define which iPhone cases to return first? The best rated, the ones with the shortest shipping time, those that most benefit your business, or the ones for which vendors pay higher fees? Or, most likely, a mix of business criteria relevant to your specific marketplace.

Introducing business logic to the search engine and mixing it with textual relevance is extremely complex to do in a Lucene-based search engine, as it combines all relevance criteria in a single score computed using a mathematical formula. This makes it hard to understand when one criterion overtakes another.

This single formula is supposed to work across all searches. As such, adding business criteria can lead to search results containing high-profit products that are completely meaningless in terms of textual relevance. It's a never-



ending job to fine-tune the relevance formula to work consistently across the marketplace.

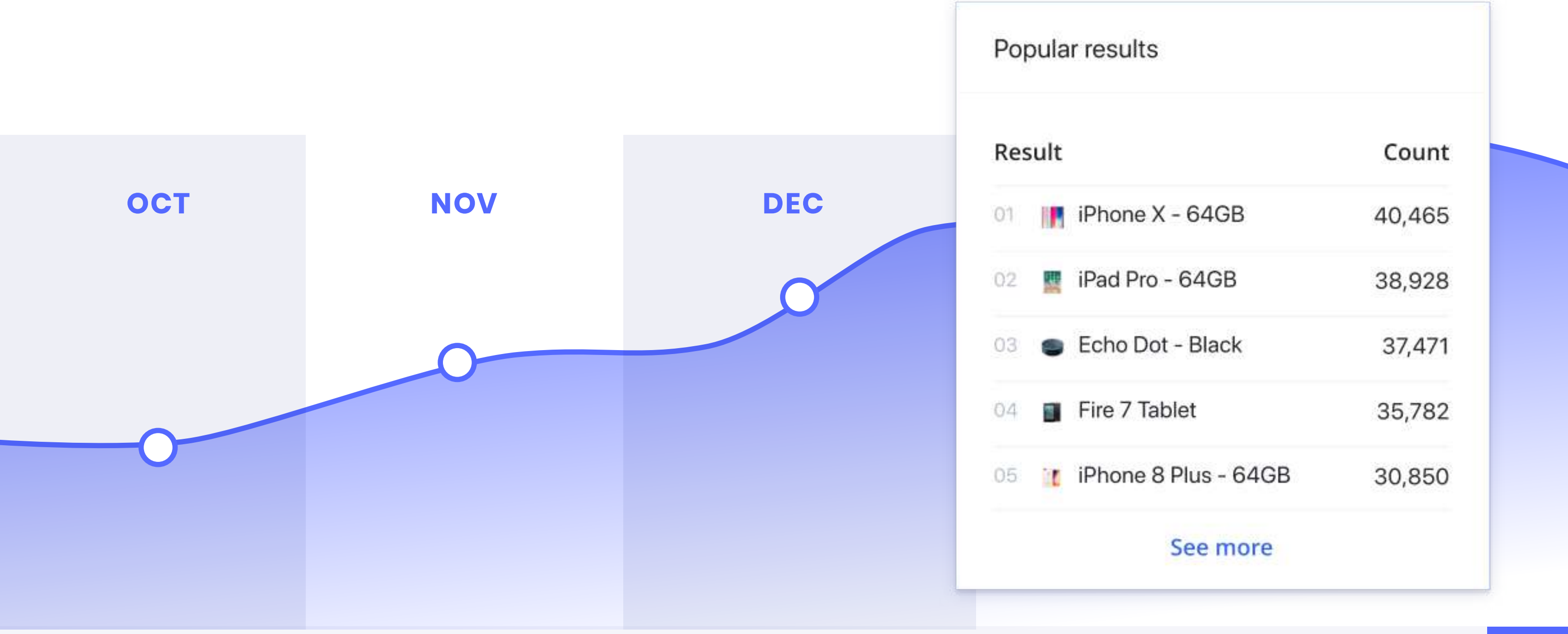
An additional way to refine your marketplace search engine is to serve users personalized search results based on their previous searches, actions, and purchases. Make sure to build all the required functionality to track and collect user signals, build user profiles from them, store the profiles, and modify the search results based on the profiles, while not discarding the textual and business relevance you worked hard on.

▶ **SKILLS REQUIRED: DATA SCIENCE, DATA ENGINEERING, ELASTICSEARCH/SOLR, BACK-END ENGINEERING**

ADD TOOLS FOR YOUR BUSINESS TEAMS

Search and Discovery is a great opportunity for merchandisers, category managers, and other business teams to get insights from shopper searches and actions, as well as to merchandise effectively.

The first step in giving business users what they need is to gain a comprehensive understanding of how the current experience is performing.



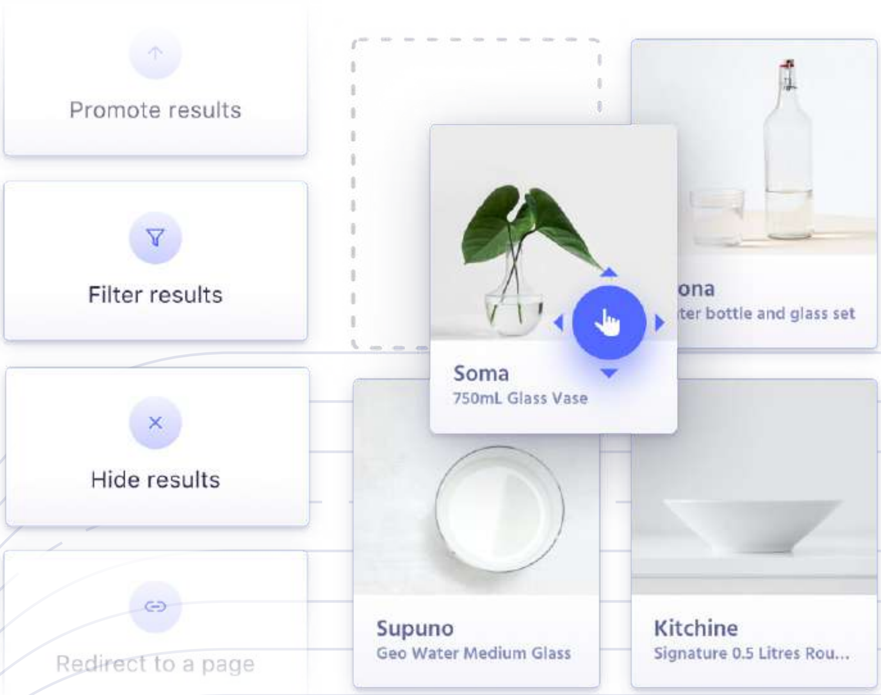
Popular results

Result		Count
01	iPhone X - 64GB	40,465
02	iPad Pro - 64GB	38,928
03	Echo Dot - Black	37,471
04	Fire 7 Tablet	35,782
05	iPhone 8 Plus - 64GB	30,850

See more

This requires search analytics tracking various elements such as the queries that users enter, the most popular results, most-bought items after a search, and most-clicked filters. Building search analytics is an ambitious engineering project that consists of building the data-tracking pipeline, normalizing queries (think about synonyms, typo tolerance, search as you type), storing the data in a secure, privacy-compliant, and efficient manner, running the right analysis and displaying the insights to users, and building search analytics.

When you have search analytics, you can go further and build an A/B testing tool to compare the performance of different types of relevance and merchandising strategies or merchandising strategies. Here, the challenge is to build a mechanism that continually routes a defined share of search traffic to a given variant while collecting and correctly routing all necessary data to compute the final results and statistical confidence score of the A/B tests, as well as removing outliers such as bots from the results.



As for managing the experience, business teams may be interested in impacting the relevance logic itself. Be careful to expose a complex relevance formula in a relatively easy way, and make sure the entire experience is not broken by someone inadvertently changing a parameter or two.

Another key capability that business users would expect from a modern marketplace search engine is the ability to merchandise the experience, for instance to promote a given product, product category, or vendor for specific queries or in specific categories. Implementing such rules in a Lucene-based engine without risking alteration of the global relevance can require years of custom development. Not only must you build your own technology to deeply customize search, including a model to generate rules, you need to develop the ability to apply the rules in real time. When you achieve this, you can think about creating a UI to allow your teams to manage the rules.

▶ **SKILLS REQUIRED: DATA SCIENCE, DATA ENGINEERING, ELASTICSEARCH/SOLR, BACK-END ENGINEERING, FRONT-END ENGINEERING**

ENSURE HIGH AVAILABILITY, MAINTENANCE, AND SECURITY

Ensuring high availability of a marketplace search engine isn’t a small project. While solutions such as Elasticsearch have built-in snapshot and restore features to revert to a previous state if you have a critical failure, you’ll need to decide where and how to store those backups.

Of course, restoring a backup is a last-resort option and introduces discontinuity to your search. Ensuring high availability often means redundancy, replicating clusters across data centers or your cloud provider availability regions, even over several cloud providers for maximum availability. This requires you to create multiple deployments of your search engine instance and synchronize them yourself.

Maintenance tasks such as updating your open-source search engine to a new version can lead to downtime and needing to redevelop part of the custom tools you built on top of your engine.

On the security side, solutions such as Elasticsearch were built to be deployed behind the firewall with your own layers of security. Users of Elasticsearch are urged to purchase Elastic's Enterprise Security for Elasticsearch and possess strong security expertise to implement and configure the solution.

▶ **SKILLS REQUIRED: NETWORK, INFRASTRUCTURE, ELASTICSEARCH/SOLR, SECURITY**

ITERATE

A marketplace search engine isn't a ship-and-forget project. As the catalog evolves, as shoppers look for new items, as the seasons change, the requirements of your search experience will change. Iterating on your search engine is mandatory, whether you're focused on relevance, the UX and other features you built on top of it, or the infrastructure behind it. Every change will likely mean in-depth work fine-tuning the relevance, making sure the infrastructure can support the new load or that the change doesn't bring security weaknesses.

▶ **SKILLS REQUIRED: DEPENDS ON ITERATION**

The alternatives

“Build a production-ready, fully featured search engine in a fraction of the typical time and with a fraction of the resources

The great open-source tools today allow for immense flexibility, letting marketplaces leverage them to build search engines that meet their specific needs and power the specific experiences that shoppers expect. Virtually anything is possible, but you’ll need extensive resources and expertise to achieve the level of relevance, performance, reliability, security, and manageability that a marketplace search engine requires in production.

At Algolia, we provide an alternative that allows you to accomplish the same results: build a production-ready, fully featured search engine in a fraction of the typical time and with a fraction of the resources.

Here’s a summary comparison of engineering expertise required to expertly build great marketplace search by open-source solutions vs. Algolia:

Step	Elasticsearch / Solr Expertise	Algolia Expertise
Provision and deploy infrastructure	— Infrastructure — Network — Advanced Elasticsearch/Solr	— None Algolia manages its own search infrastructure and abstracts this layer from customers. The customer must consider only the expected number of products to index and the search traffic.
Prepare data and configure indexing	— Elasticsearch/Solr	— Algolia Working on the structure of data to index will be required with every solution on the market.
Configure search	— Advanced Elasticsearch/Solr	— Algolia Out of the box, Algolia comes with a series of features that all consumer-facing search experiences need, including search as you type, typo tolerance, synonyms, highlighting, and snippeting. Algolia provides the option to customize configuration of those features without risking an impact on performance or breaking the relevance.
Implement search on your front ends	— Elasticsearch/Solr — Front-end engineering	— Front-end engineering Algolia comes with extensive front-end libraries for the most common front-end frameworks, as well as API clients for iOS and Android. All the logic complexity is managed by the libraries.
Configure advanced relevance and personalization	— Data science — Data engineering — Advanced Elasticsearch/Solr — Back-end engineering	— None Algolia relevance is based on a tie-breaking algorithm, which makes it very easy to introduce new ranking criteria and understand how they will impact the experience. This algorithm can be managed via API or a user-friendly dashboard. Algolia comes with a built-in personalization feature and allows users to customize the personalization algorithm.

Add tools for your business users	— Data science — Data engineering — Elasticsearch/Solr — Back-end engineering — Front-end engineering	— None Algolia comes with built in search analytics, merchandising tools, a user- friendly dashboard, A/B testing, and more.
High availability, aintenance, and security	— Network — Infrastructure — Elasticsearch/Solr — Security	— None Algolia manages its own search infrastructure. Every search application is hosted on a cluster composed of 3 identical servers distributed across 2 data centers in 2 availability regions. Being a SaaS product, each upgrade and new feature is transparently brought to each customer without service discontinuity. Algolia is SOC 2 & SOC 3 compliant and has a full-fledged security team.
Summary	— Advanced Elasticsearch/Solr — Infrastructure — Network — Back-end engineering — Front-end engineering — Data science — Data engineering — Security	— Algolia — Back-end engineering — Front-end engineering

algolia

Thank you for reading!

ABOUT ALGOLIA

Algolia enables e-commerce businesses to create fast, relevant and intuitive search and discovery experiences that increase conversions, engagement and customer loyalty. We deliver personalized shopper experiences and fuel your digital strategy with sophisticated search technologies like conversational search. More than 9,000 companies like Under Armour, Lacoste, Birchbox, Dior, L'Occitane En Provence and Bringmeister rely on Algolia.

To learn more, visit **resources.algolia.com**