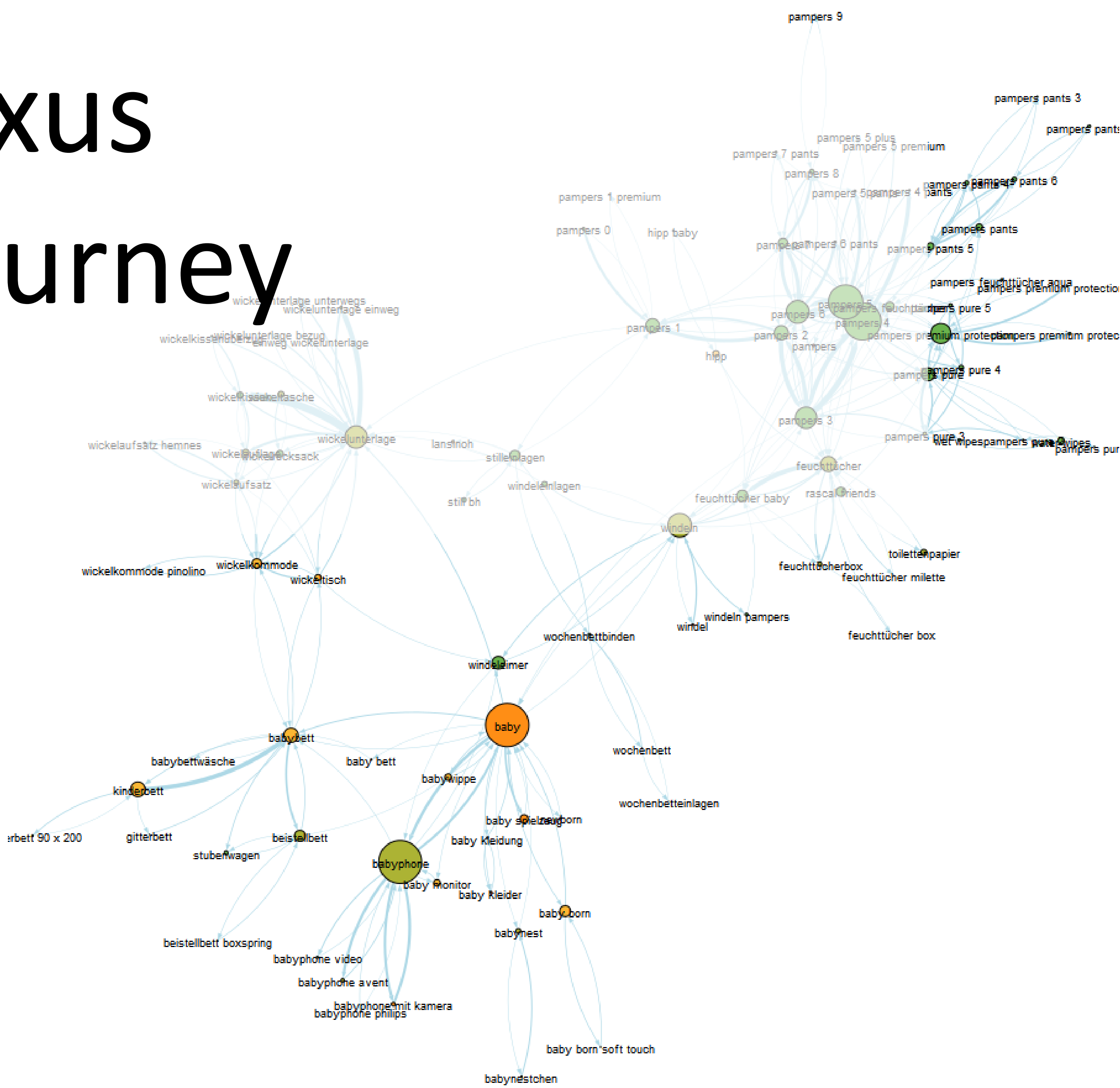
The Digitec Galaxus Vector Search Journey



Abel Camacho Guardian, Joel Widmer

GALAXUS

Digitec Galaxus

7 Countries
Austria, Belgium, Italy, France, Germany, Netherland, Switzerland

5 Languages
Dutch, English, Italian, French, German

350 Million Searches
in 2024

Over 2 Million Searches
on Black Friday (2024)

100k
Daily Vector Searches

Search at Digitec Galaxus

Two teams, 13 people

- **One team focuses on frontend and filtering**
- **Second team focuses on search relevance**
- **Shared platform for infrastructure**

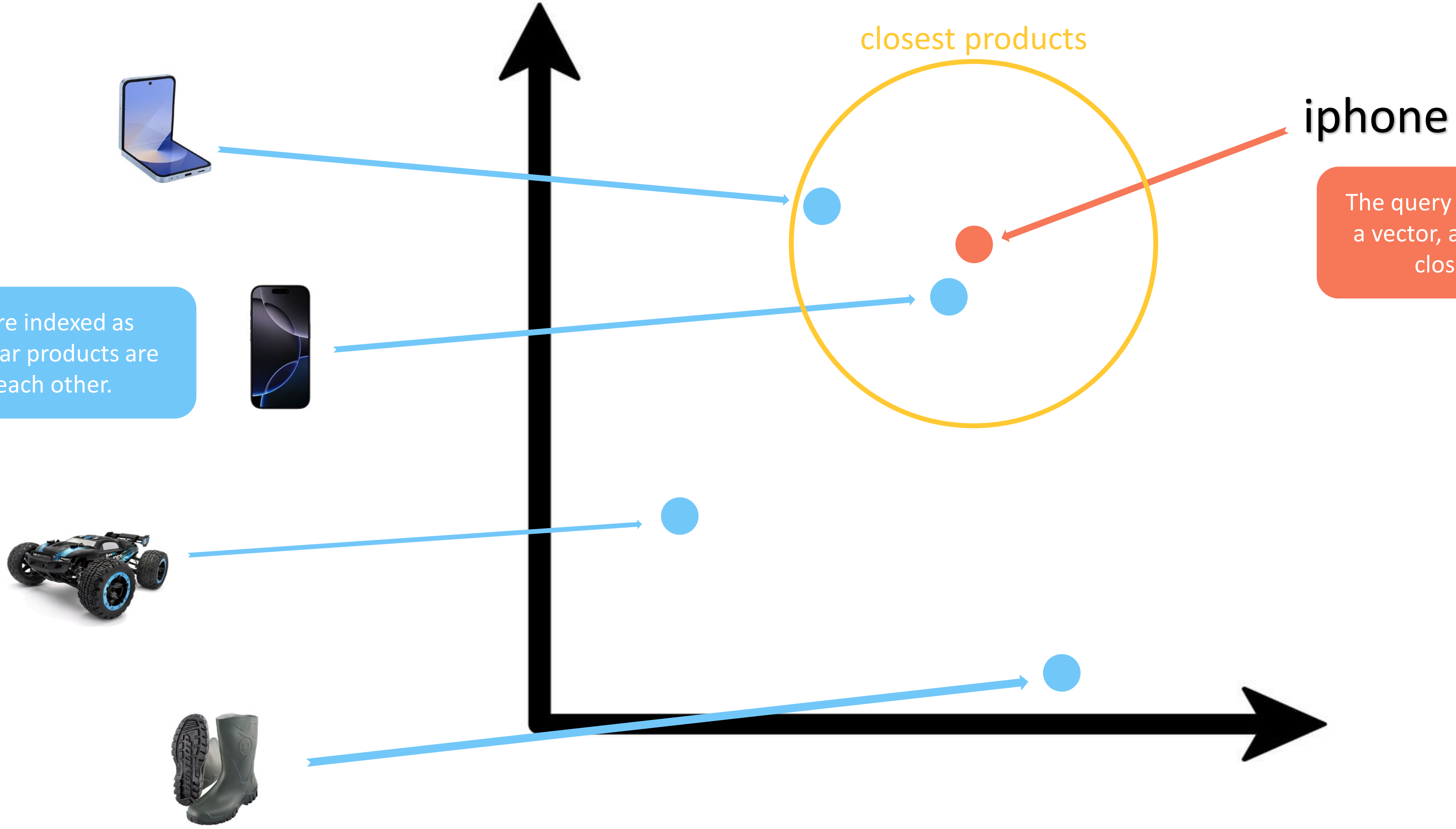
Abel Camacho Guardian

Senior Analytics Engineer

Joel Widmer

Search Engineer

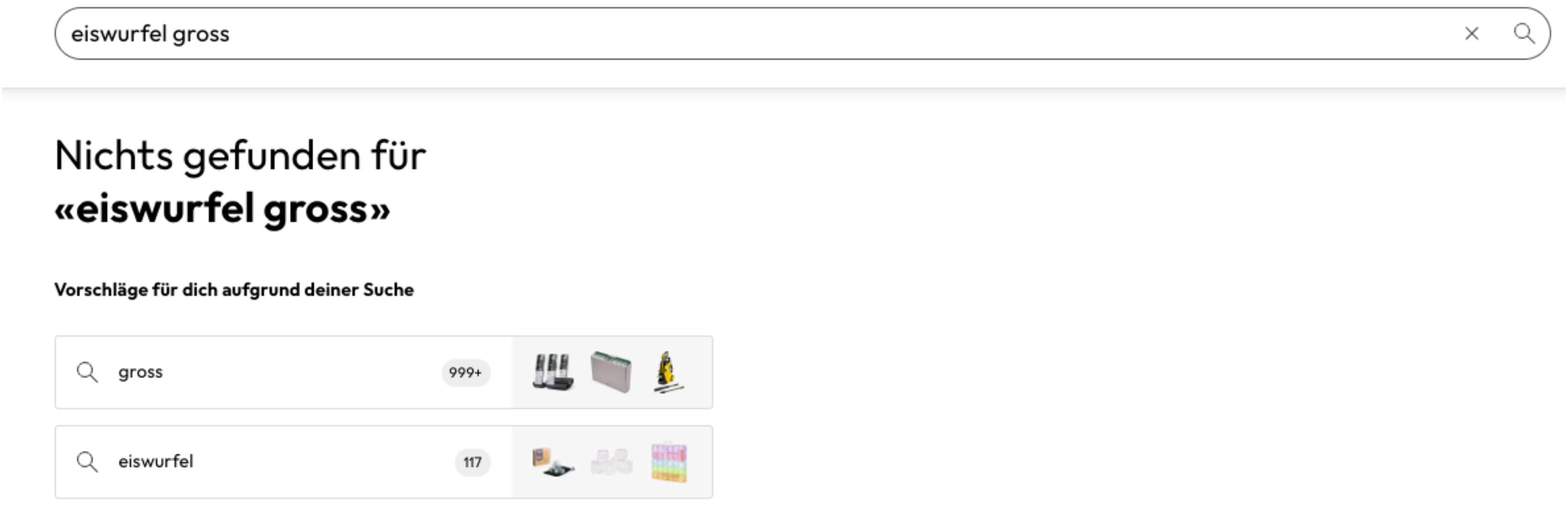
What is vector search?



Products are indexed as vectors. Similar products are close to each other.

The query is also translated to a vector, and we retrieve the closest products.

Why do we need vector search?



- Roughly 10% of all searches ended up on a zero results page
- For many of these searches we do have relevant products which are not retrieved with keyword search

The journey starts at MICES

Vectorizing consumer electronic goods - Ruchi Juneja, Johannes Peter - MICES 2024

How semantic search projects fail - Roman Grebennikov - MICES 2024

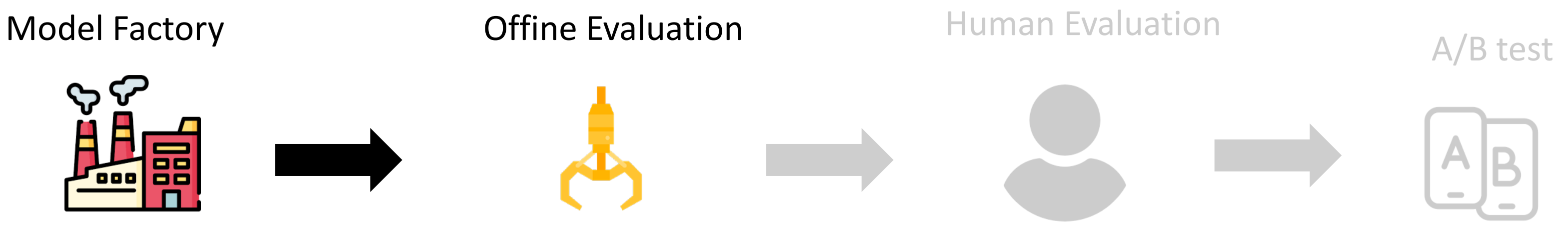
A bouquet of insights from vector search AB-Tests

Our process to bring a vector search model into an AB-Test

Model Factory: Create many model candidates

- Create a fine-tuning pipeline from raw data to model
- Many models never see the light of an AB-Test

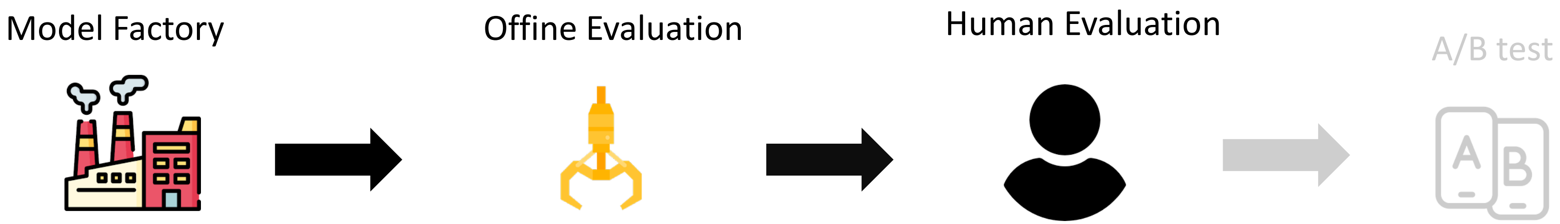
Our process to bring a vector search model into an AB-Test



Offline Evaluation: First step of model selection

- Give each simple tasks to the models and filter out the bad ones
  - Out of 100 products, which one fits best for “iphone”?
  - How many of the top 10 products for “iphone” are from the category “smartphone”?
- Hypothesis: If a model is bad at those simple tasks, it is also bad at vector search
- The top models according to the offline evaluation are considered for the next step

Our process to bring a vector search model into an AB-Test

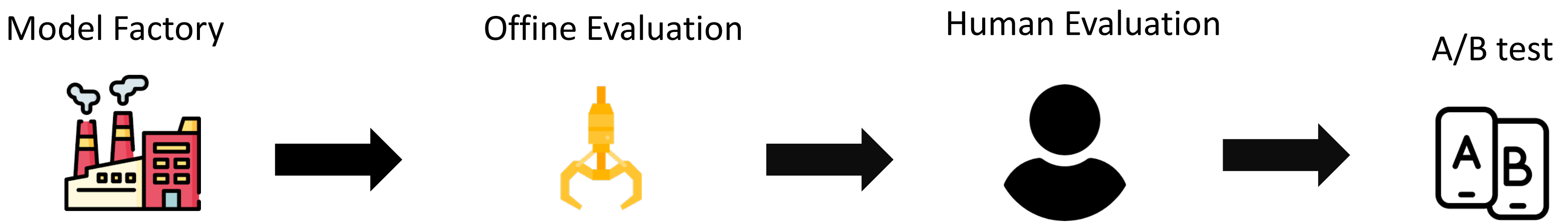


Human Evaluation: Manual grading by an expert

- Actual zero result queries have very low volume, so implicit labels are unreliable
- We can get a feeling of the strengths and weaknesses of the models
- The two best models go into an AB-Test

We are in the process of enhancing human evaluation with LLM-as-a-judge

Our process to bring a vector search model into an AB-Test



AB-Test: The ONLY true signal of model quality

- Only after the AB-Test we can truly say which model is better
- Custom metric to compare performance between vector search and zero results
- Ran AB-Tests only for one week, since signals were so strong

Three challenging areas of vector search

- Indexing
- Query Time Latency
- **Results Quality**

Measuring result quality

- How can we measure and compare result quality if our control group does not even show products?
    - CTR does not work for 0-result pages (control group).

We define a new metric “Search Success Rate”

💡 Our data shows that a direct refinement happens within the first 30 seconds

💡 Search Success = A click on any product within the first 30 seconds

Pre-trained models are not good enough

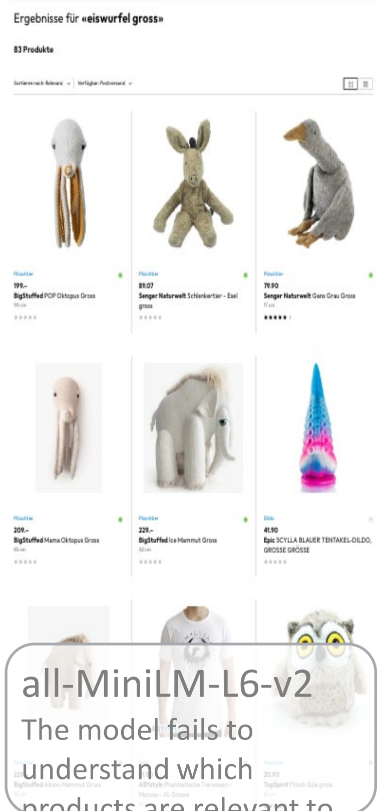
- Pre-trained embedding models struggle to capture the nuance required for e-commerce search
- Fine-tuning only with product data is not enough.

💡 Showing bad vector search results was overall worse than showing zero results pages

💡 We saw a significantly lower search success rate and a significantly higher exit rate

Pre-trained models are not good enough

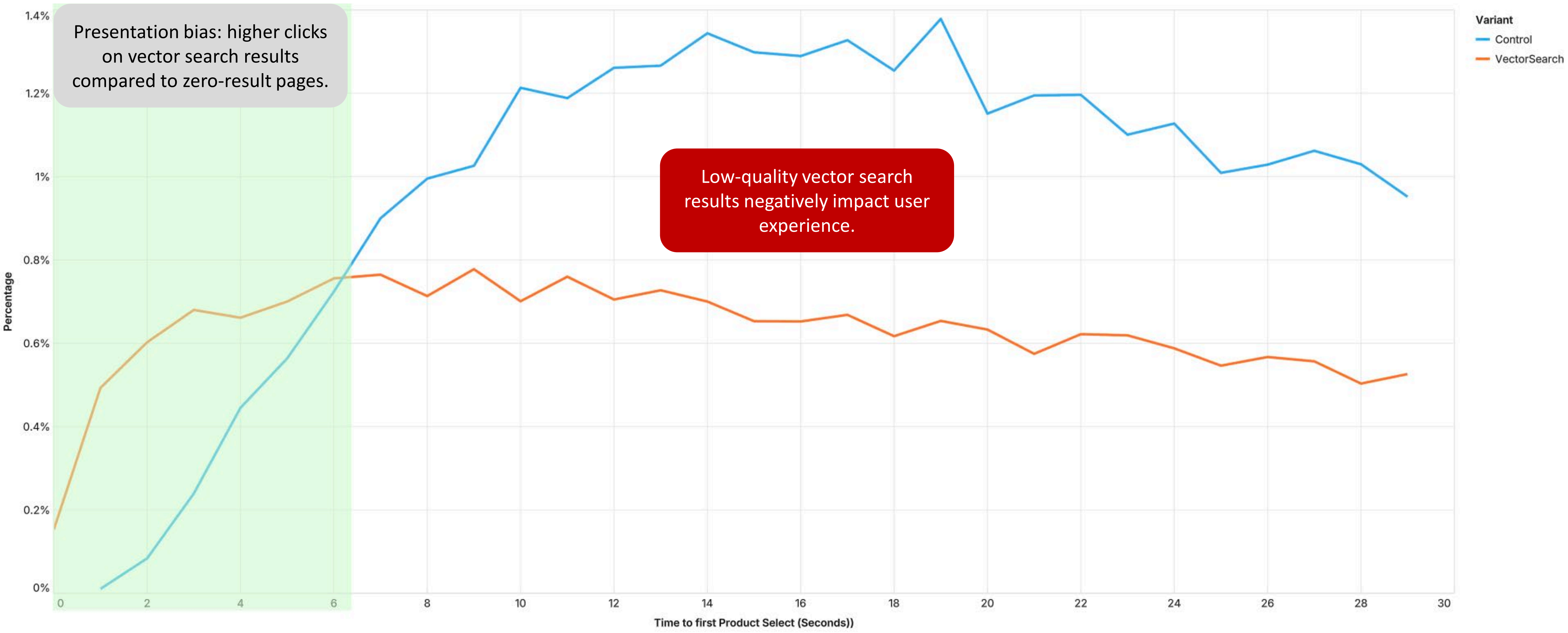
eiswurfel gross



all-MiniLM-L6-v2

The model fails to understand which products are relevant to the ‘eiswurfel gross’ query.

Showing bad vector search results was overall worse than showing zero results pages



Fine-tuning a pre-trained model

- We use a combination of product data and user behavioral data to fine-tune a pre-trained model

💡 Product Data: Put our products into the context of natural language

💡 The model learns about the products in different languages.

💡 Behavioral Data: Link queries to products - postive examples (query, product, 1)

💡 The model learns which products are relevant for a query.

💡 The model learns which queries are similar.

Hard negative examples are the key for good fine-tuning

💡 We use our existing product taxonomy to create hard negative examples

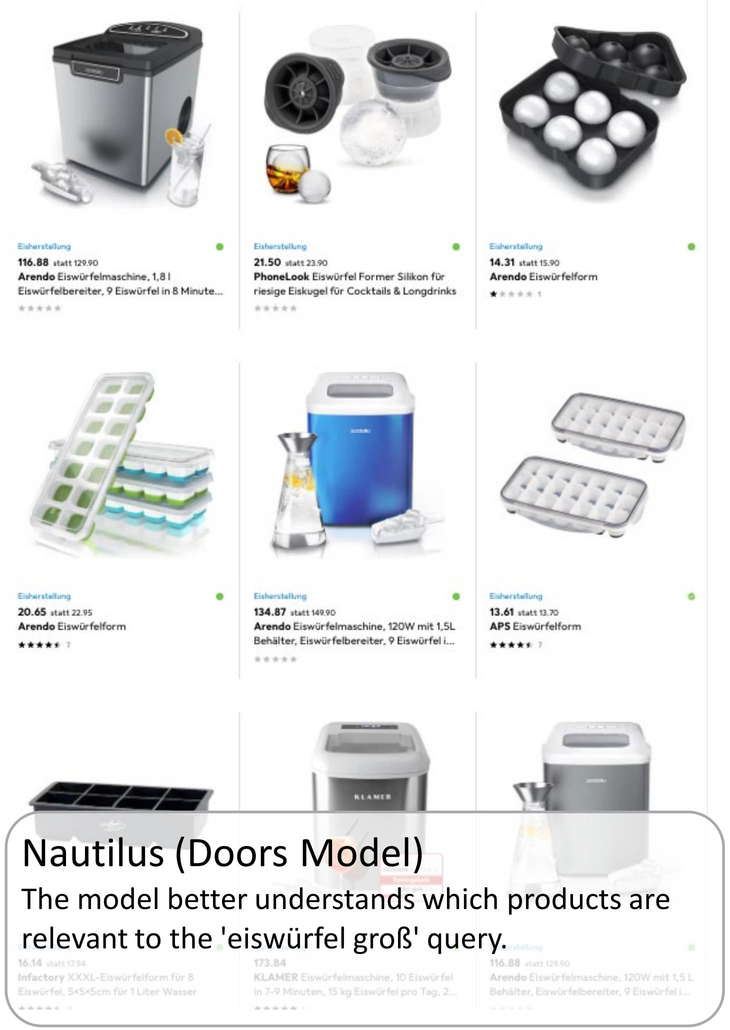
💡 We need hard negative examples to teach the model nuance

A hard example is a query-product pair where the product is very close to the intent but still wrong.

(iphone, smartphone case iphone, 0) - Although smartphone cases are taxonomically close to smartphones, they tend to attract less user engagement when user search for iPhone.

Behavioral data provides signals that help determine relevant products for queries.

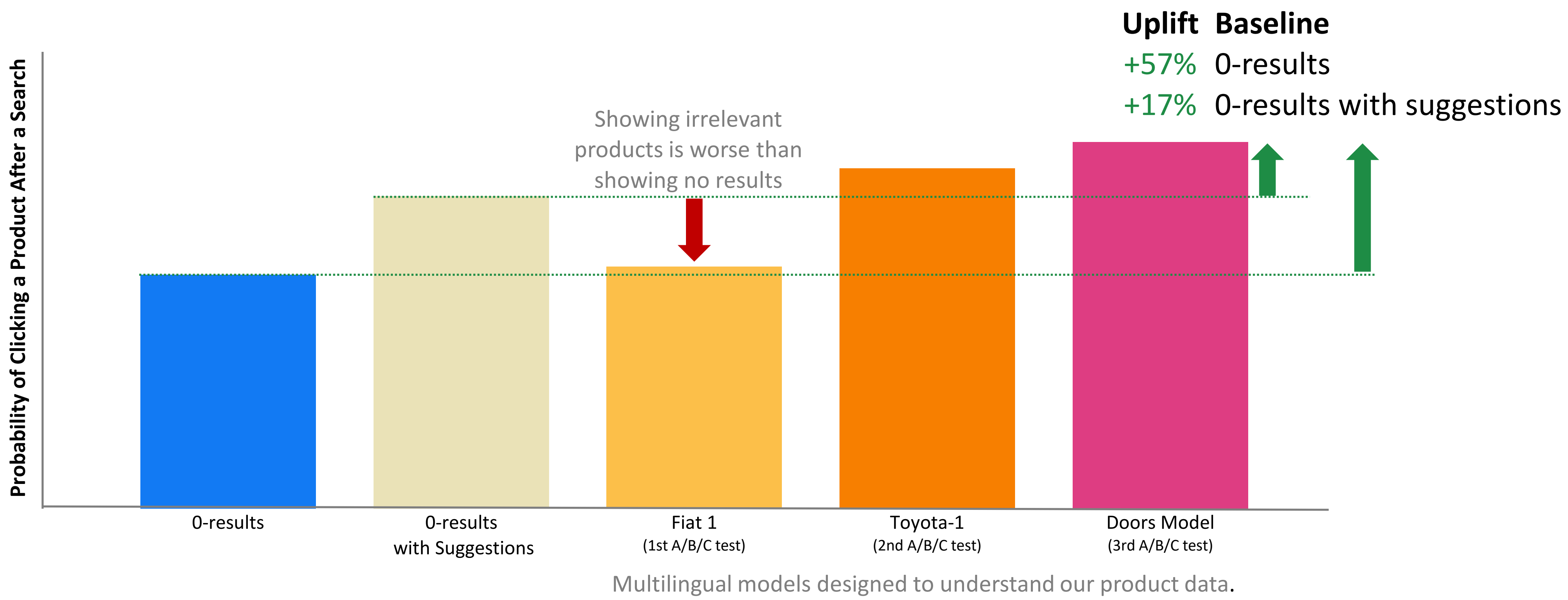
eiswurfel gross



Nautilus (Doors Model)

The model better understands which products are relevant to the 'eiswürfel groß' query.

After fine-tuning our model with product and behavioral data, along with several attempts, we achieved a significant uplift in multiple business metrics



Conclusion

💡 A custom metric helps us comparing zero result and vector search pages

💡 Hard negative examples are crucial for fine-tuning. Use your product taxonomy!!

💡 A helpful zero results page can be better than poor vector search results

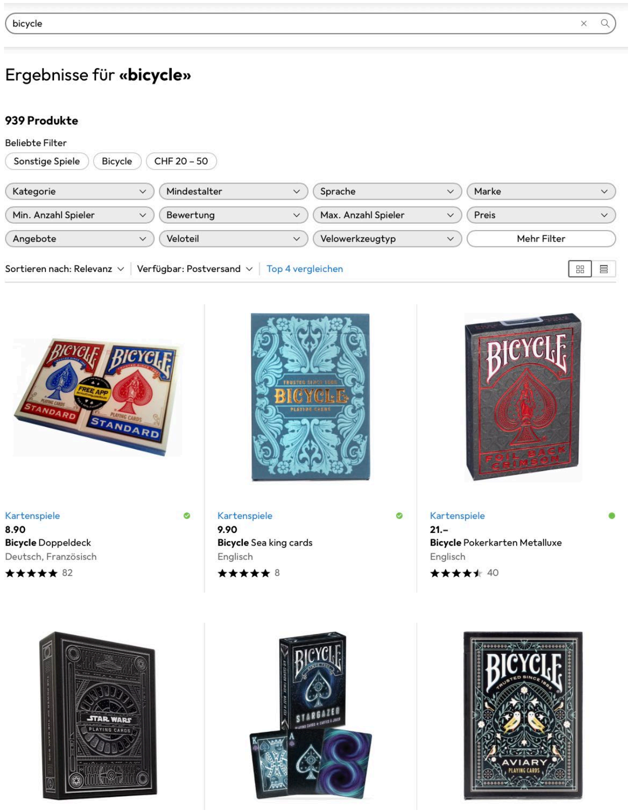
💡 If you want to build semantic search, start with the technology you know. In our case, we use Elasticsearch.

💡 Implementing semantic search is not enough, you must prove its value to both users and the business

Outlook

💡 Introduce Hybrid Search for Low-Performance Keyword Queries

💡 Improve the embedding model by better handling presentation bias in the training data used for fine-tuning.



Thank you

Join Us