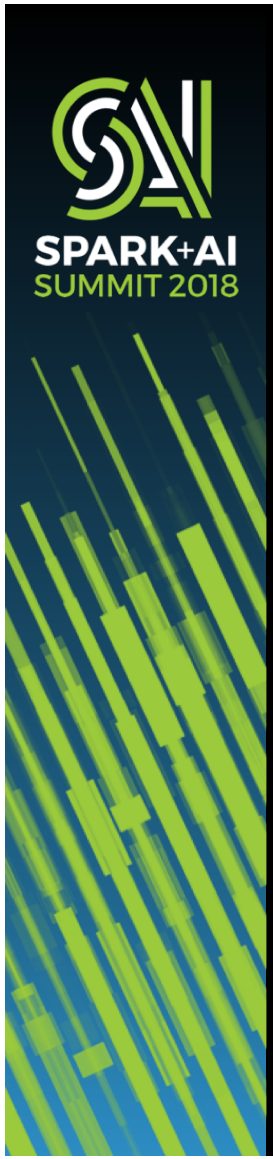


Using Spark-Solr at Scale: Productionizing Spark for Search with Apache Solr

FAIZAN AHMED

flipp

About Me

Role

Data Engineer on **Flipp's** Search Team

Education

M.Sc in Computer Science from University of Waterloo

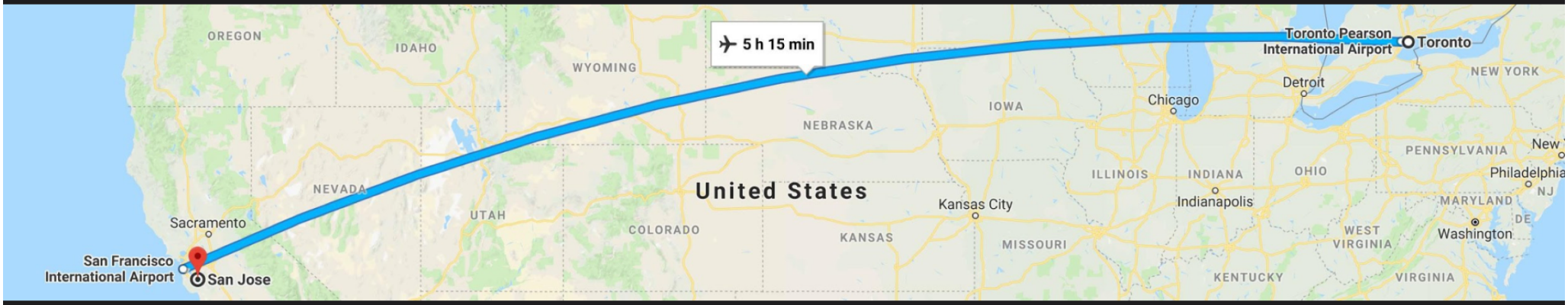
medium.com/@faizanahmed_18678

Responsibilities

Building pipelines to support machine learning models

Developing scalable systems and working on products to enhance search relevancy.

Core work includes using Spark for machine learning at scale & data warehousing for analytics.



Agenda

- About Flipp
- The Search Experience
- The Problem Domain
- The Spark-Solr Journey
  - Solution Architecture
  - Deep Dive: Content Categorization
  - Deep Dive: Understanding User Intent
- Improvements
- Lessons Learned
- Key Takeaways

About Flipp

Digital Shopping Marketplace

The Flipp App is a digital shopping marketplace with over 40M+ downloads.

Storefront Visual Merchandising

Enterprise Platform converts analog content to provide a mobile browsing experience, which showcases retailers' digital storefront.

The Flipp Platform

Flipp's platform is used by over 90% of tier one retailers in North America.

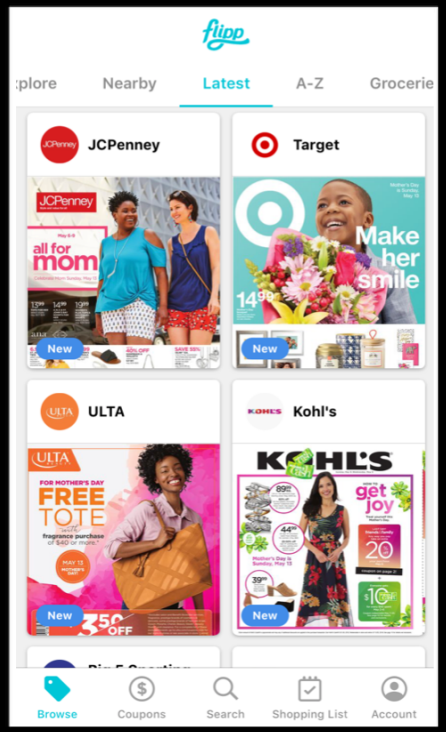
Search Platform At a Glance

- 2M Searches/day
- 30M Indexed Documents
- Team of 5 engineers

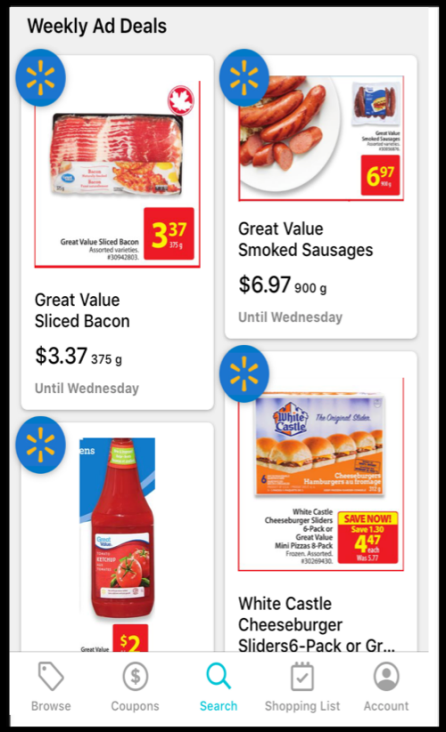

Searchable Entities

The Search Experience

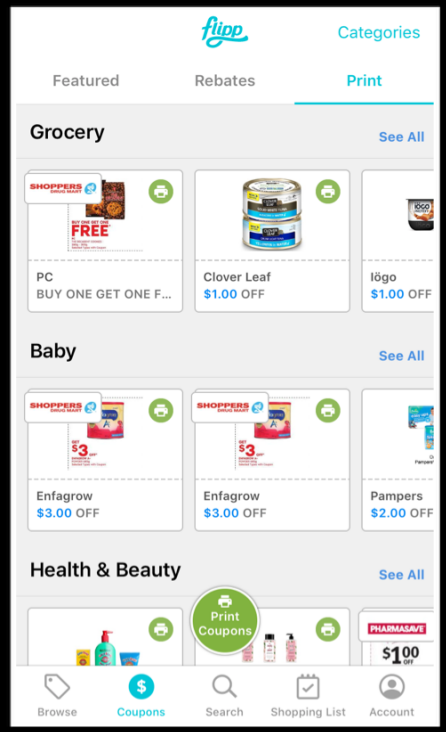
Merchant Flyers



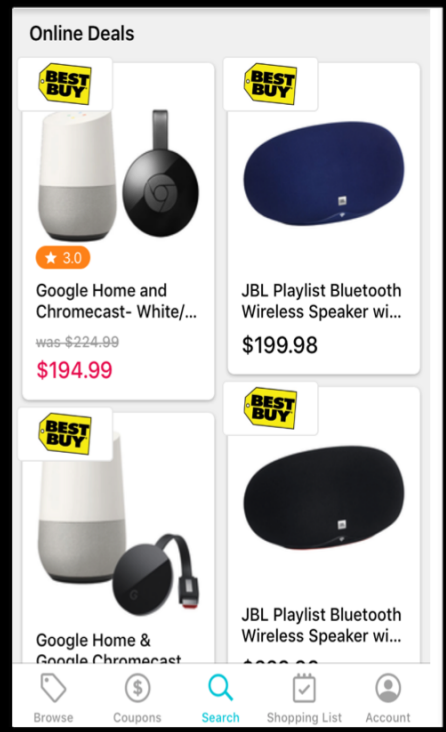
Items



Coupons

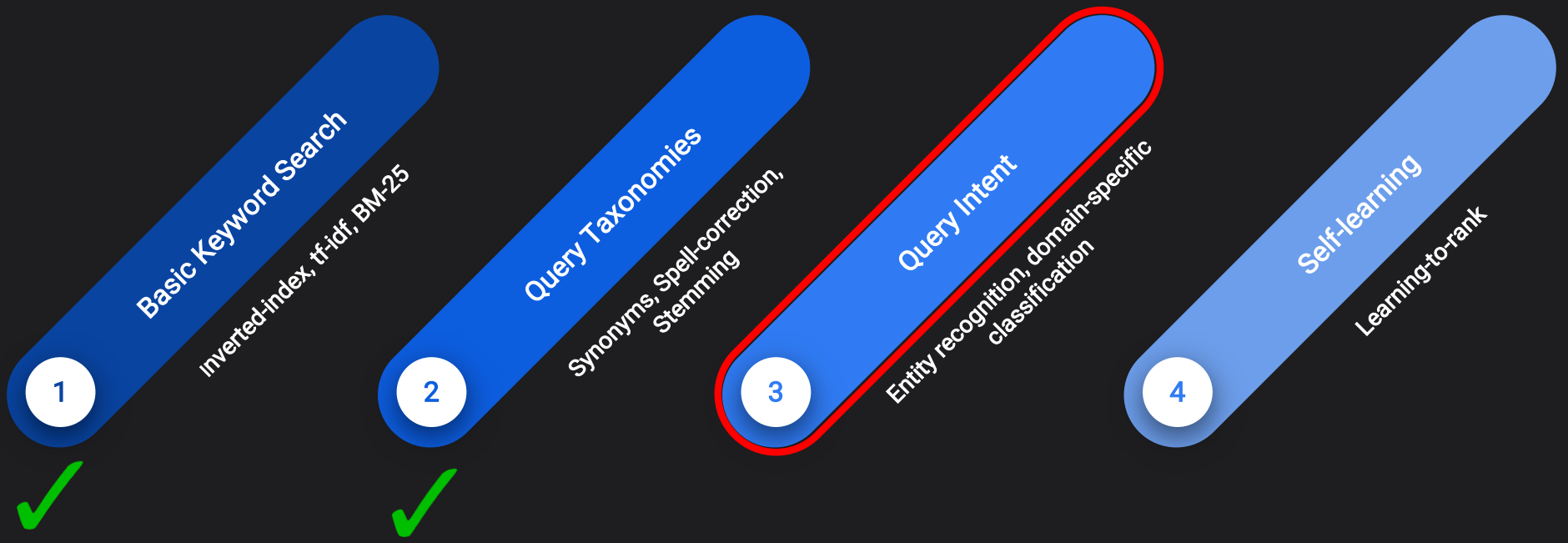


Online Deals



The Problem Domain

Search Platform Evolution

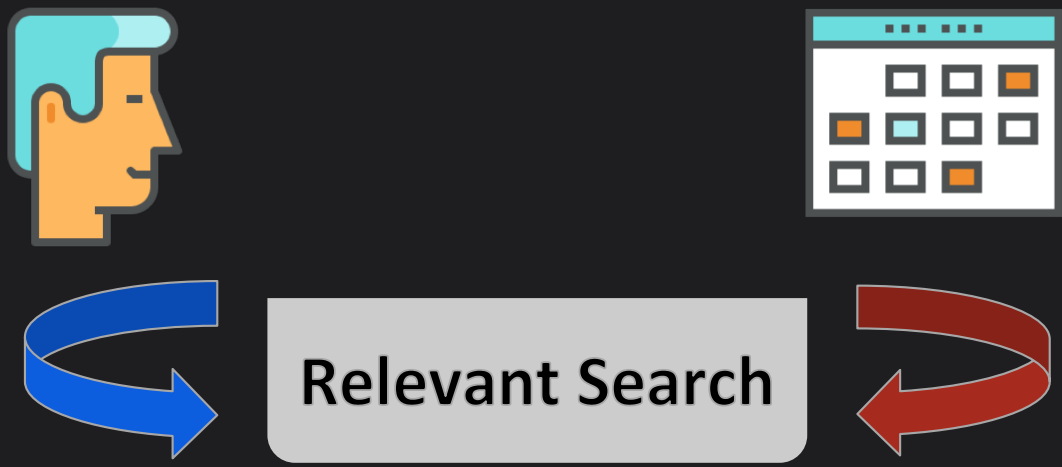


Discovering and leveraging query intent is dependent on domain specific **user & content signals**.

Signals: The Building Blocks Of Search

User Signals

Content Signals



Content Signal Challenges: Product Categorization

- Non-uniform ***category*** structure

Every item would have a *merchant-specific* category

Consumer Electronics -> Consumer Audio Headphones

Audio-> Earbuds & In-Ear Headphones-> Headphones

"***Bose SoundSport In-Ear Sport Headphones***"

Content Signal Challenges: Product Categorization

- Content tagged with correct categories lacked complete hierarchy

Tagged as "**Kale**"

Tagged as "**Greens**"

Categorization hierarchy for "*Kale*"

Food, Beverages & Tobacco > Food Items > Fruits & Vegetables > Fresh & Frozen Vegetables > Greens > Kale

User Signal Challenges

- Unable to leverage Solr features

{!boost b=log(popularity)}foo

"cheese" -> Dairy Products -> Cheese

"shredded cheese" -> ?

- Query categorization not deep enough

Food, Beverages & Tobacco > Food Items > Fruits & Vegetables

What Was Needed?

ML, ETL & Analytics

Combine Signals

Spark-Solr Integration

Reduce Productionization Time

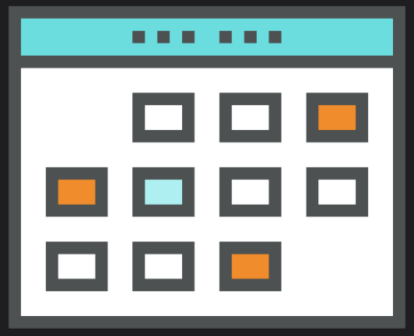
Scalable & Efficient

Cost-friendly


The Spark-Solr Journey
Implementing the solution

Action Plan

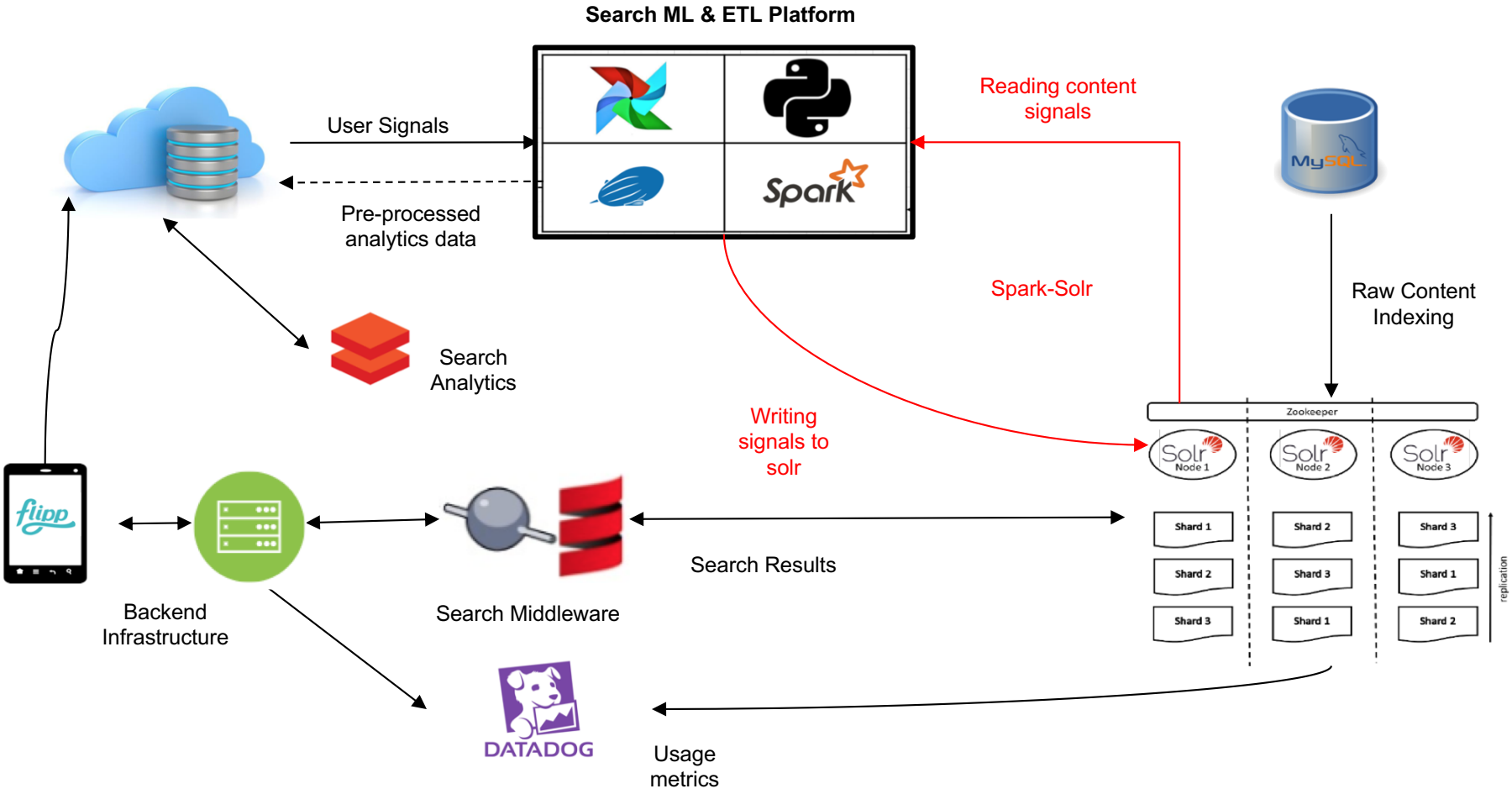
Content



Context

Collaboration

The Spark-Solr Architecture




DEEP DIVE
Content Categorization:
Separating wheat from the chaff

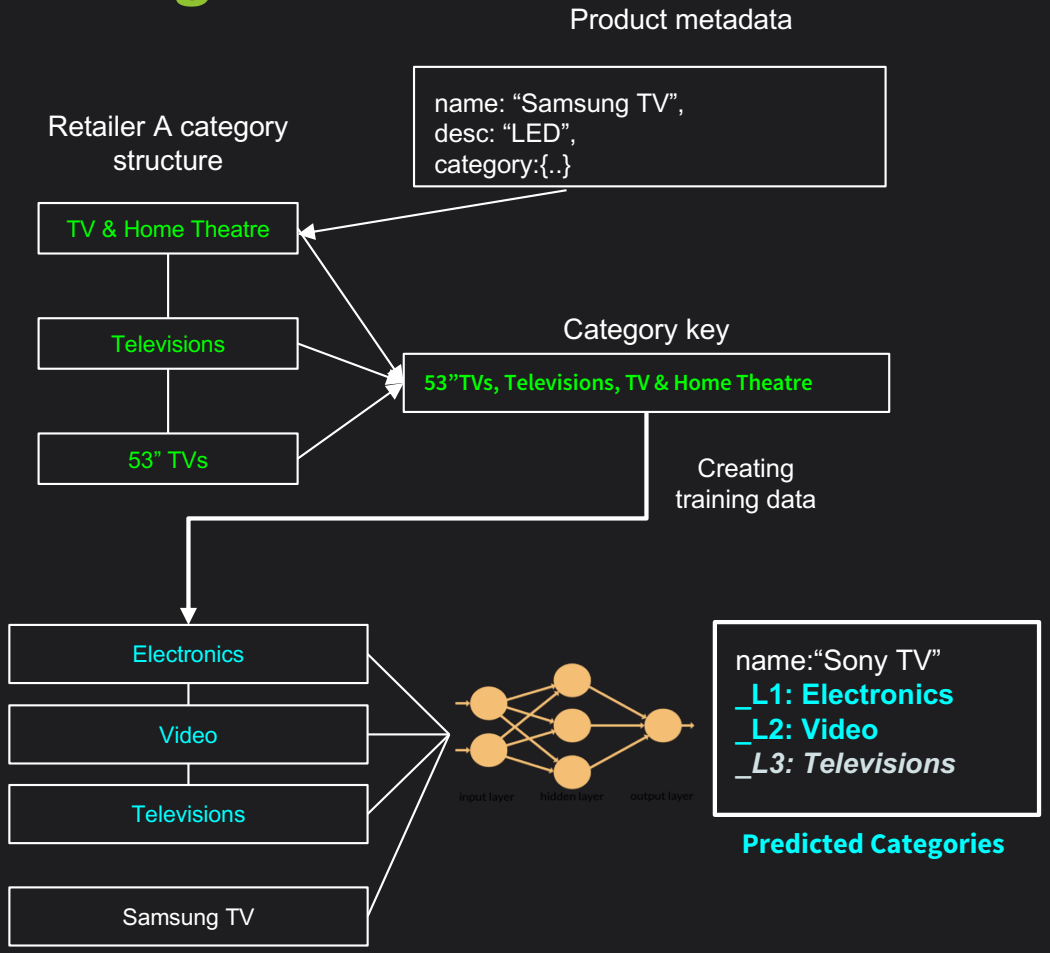
What Raw Content In Solr Looks Like


```
"docs": [
  {
    "name_v": [
      "Samsung The Frame 55\" 4K UHD HDR LED Tizen Smart TV (UN55LS003AFXZC) - Charcoal Black"
    ],
    "categories": "53 - 59 inch TVs,Televisions,TV & Home Theatre",
    "merchant_id": "210"
  },
  {
    "name_v": [
      "Samsung 55\" The Frame 4K UHD Smart LED TV with Art Mode"
    ],
    "categories": "TV's (55-64\")",
    "merchant_id": "299"
  },
  {
    "name_v": [
      "Samsung 65\" Wood Frame for The Frame TV - Beige Wood"
    ],
    "categories": "Home Theatre Accessories,TV & Home Theatre,TV Frames",
    "merchant_id": "210"
  },
  {
    "name_v": [
      "Samsung 65\" Customizable Frame for \"The Frame\" TV - Oak"
    ],
    "categories": "Other TV & Video Accessories",
    "merchant_id": "299"
  }
]
```

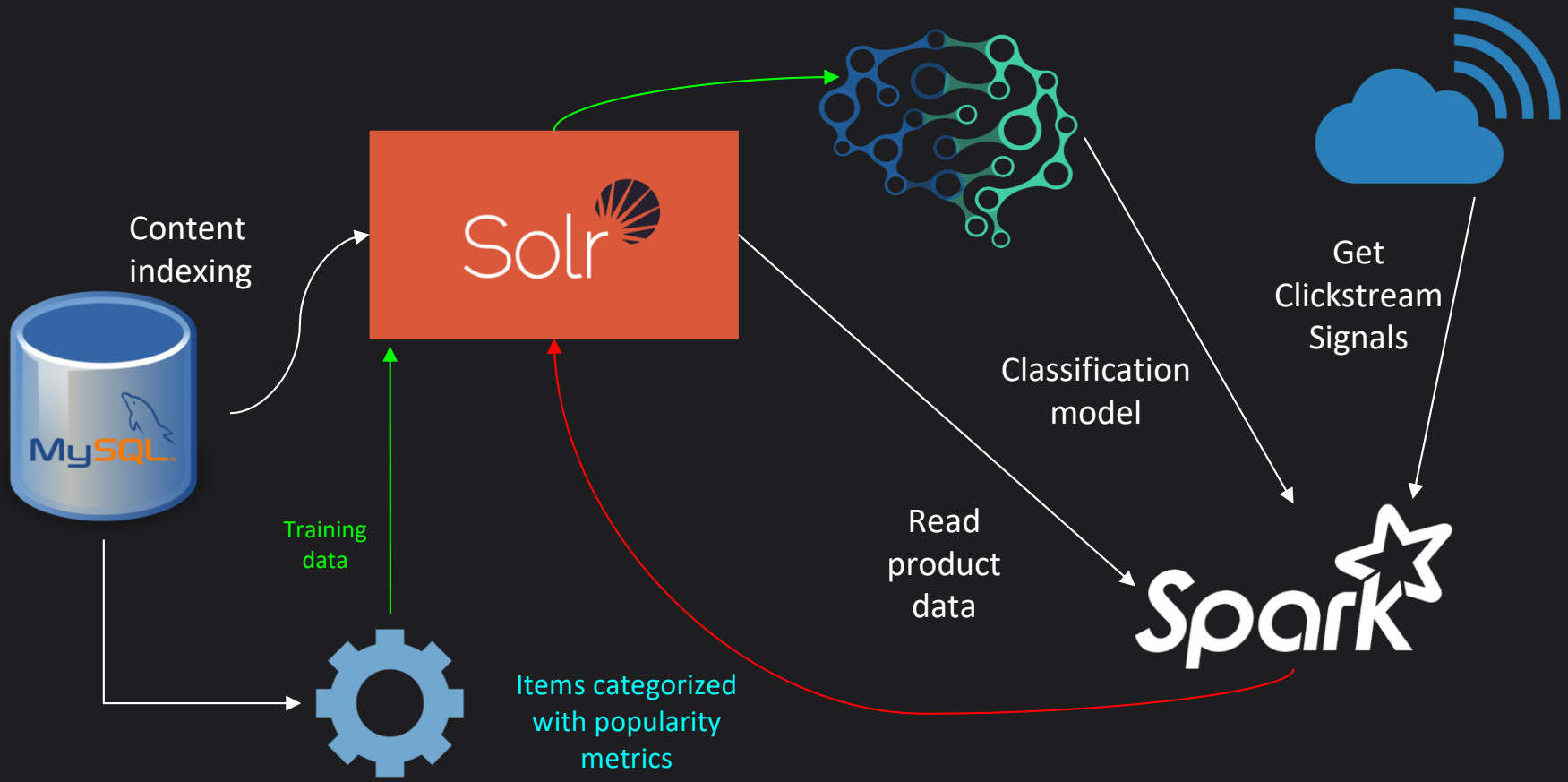

Same products but different category structure

TV accessories that would show up as noise

Categorization Algorithm



Content Categorization Workflow

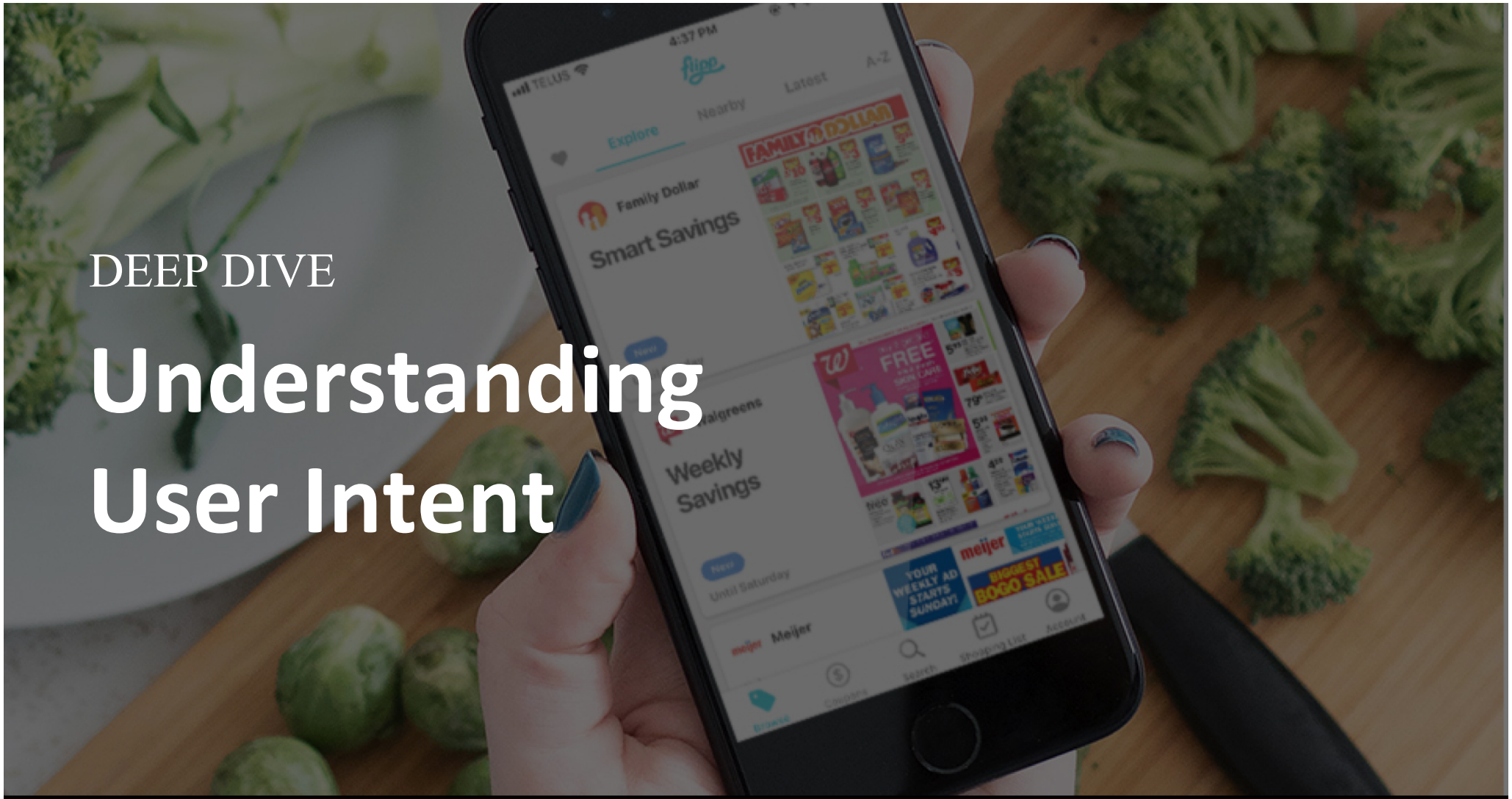


Voilà!


```
"docs": [
  {
    "name_v": [
       "Samsung The Frame 55\" 4K UHD HDR LED Tizen Smart TV (UN55LS003AFXZC) - Charcoal Black"
    ],
    "categories": "53 - 59 inch TVs,Televisions,TV & Home Theatre",
    "merchant_id": "210",
    "_L1": "Electronics",
    "_L2": "Video",
    "item_weight": 1.5928669693999906
  },
  {
    "name_v": [
       "Samsung 55\" The Frame 4K UHD Smart LED TV with Art Mode"
    ],
    "categories": "TV's (55-64\")",
    "merchant_id": "299",
    "_L1": "Electronics",
    "_L2": "Video",
    "item_weight": 0.6496946375051786
  },
  {
    "name_v": [
       "Samsung 65\" Customizable Frame for \"The Frame\" TV - Oak"
    ],
    "categories": "Other TV & Video Accessories",
    "merchant_id": "299",
    "_L1": "Furniture",
    "_L2": "Entertainment Centers & TV Stands",
    "item_weight": 4.199292645481367
  },
  {
    "name_v": [
       "Samsung 65\" Wood Frame for The Frame TV - Beige Wood"
    ],
    "categories": "Home Theatre Accessories,TV & Home Theatre,TV Frames",
    "merchant_id": "210",
    "_L1": "Furniture",
    "_L2": "Entertainment Centers & TV Stands",
    "item_weight": 1.4340761125496102
  }
```

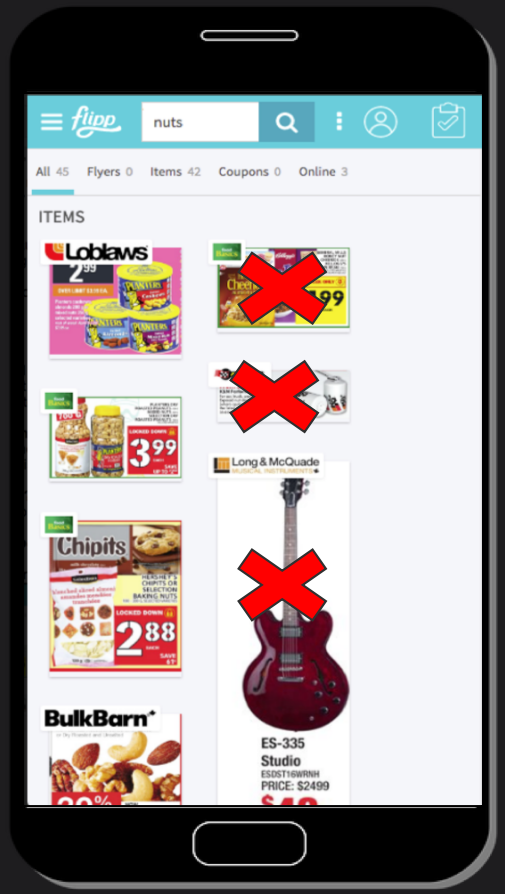

TVs are now under similar categories with popularity weight

The stands are now correctly classified as ‘Furniture’ and can be filtered


DEEP DIVE
Understanding User Intent
4:37 PM
flipp
Explore
Nearby
Latest
A-Z
Family Dollar
Smart Savings
Weekly Savings
Meijer
FREE
YOUR WEEKLY AD STARTS SUNDAY!
BIGGEST BOGO SALE

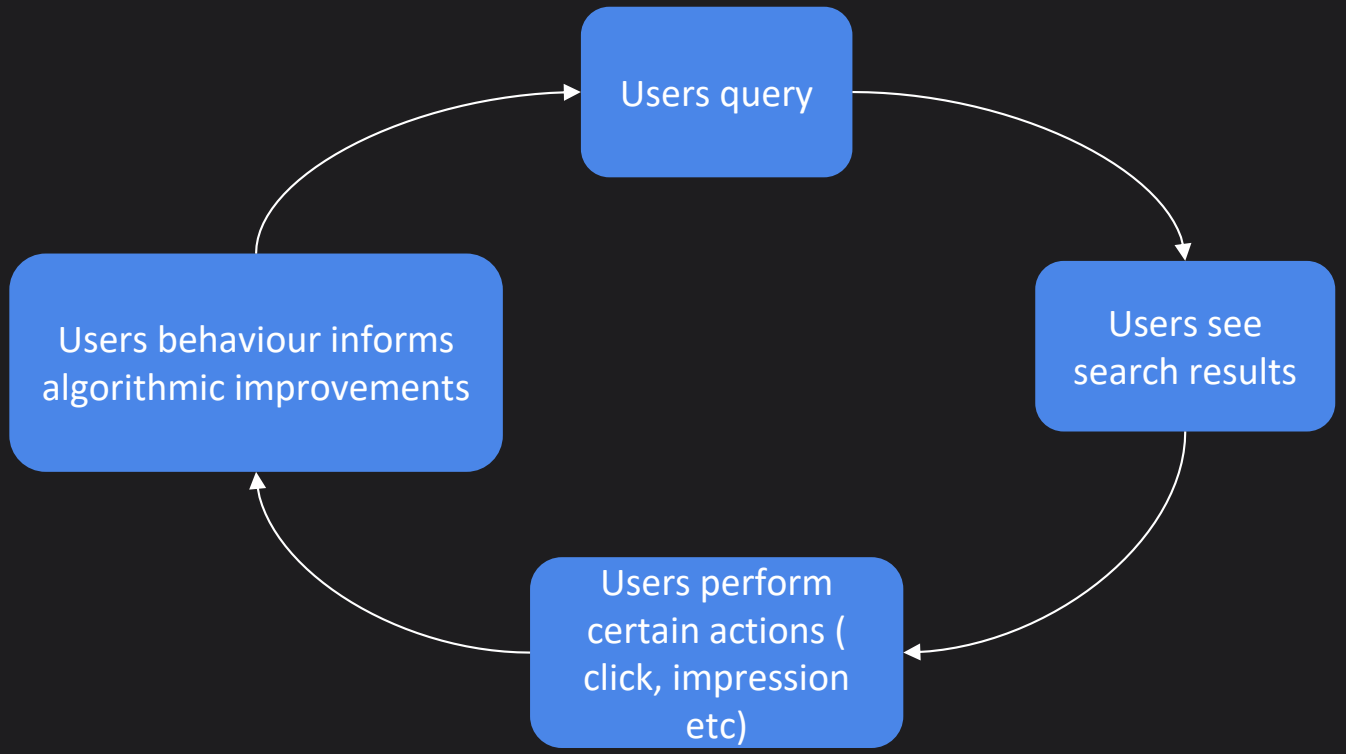
Why Decipher Query Intent?

A query for "*nuts*" ranks ***cereal***, ***guitar*** & ***car oil filter*** on top due to noisy keyword signals



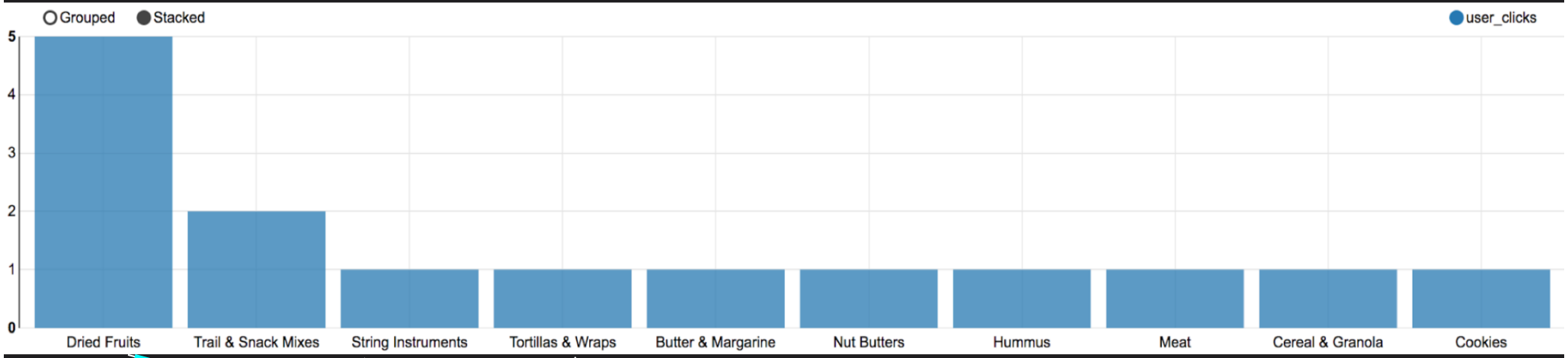
Reinforced Intelligence

- Using past impression data to improve how latest impression data is interpreted



Narrowing Down The Query Intent

*What is the probability that a user will click on a particular category class given the query "**nuts**"?*



The query "*nuts*" will be interpreted to have category "*Dried Fruits*"

One query can be associated with multiple classes

The category with the most clicks for a particular search query would be interpreted as the query category

Persisting Query Classifier Model In Solr

Model schema in solr

```
root
 |-- category_L1: string (nullable = true)
 |-- _L1_conf: double (nullable = true)
 |-- category_L2: string (nullable = true)
 |-- _L2_conf: double (nullable = true)
 |-- category_L3: string (nullable = true)
 |-- _L3_conf: double (nullable = true)
 |-- category_L4: string (nullable = true)
 |-- _L4_conf: double (nullable = true)
 |-- category_L5: string (nullable = true)
 |-- _L5_conf: double (nullable = true)
 |-- category_L6: string (nullable = true)
 |-- _L6_conf: double (nullable = true)
 |-- category_L7: string (nullable = true)
 |-- _L7_conf: double (nullable = true)
 |-- isBrand: long (nullable = true)
 |-- searchable_query: string (nullable = false)
 |-- isMerchant: long (nullable = true)
```

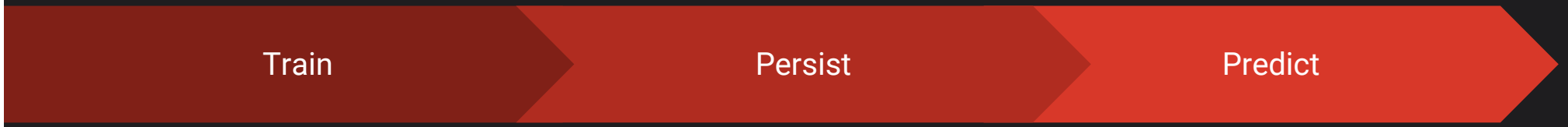
Category classifications for "*nuts*" variations

```
1  import requests
2  import pandas as pd
3  nut_queries = ['nut thins', 'nuts', 'peanut butter', 'cashews', 'mixed nuts', 'almonds']
4  cats =[]
5  for q in nut_queries:
6      cat_frame ={x:y[0]  for x,y in requests.get("{}/query?q=clean_query:{}&wt=json&\
7      fl=_L1,_L2,_L3,_L4,_L1_prob"\
8        .format(solr_url,q)).json()\
9         ['response']['docs'][0].items()}
10     cat_frame['query'] = q
11     cats.append(cat_frame)
12 display(sqlContext.createDataFrame(pd.DataFrame(cats))\
13         .selectExpr('query','_L4 as category','_L1_prob as pred_conf'))
```

▸ (3) Spark Jobs

query	category	pred_conf
nut thins	Cereal & Granola	0.93500465
nuts	Dried Fruits	0.9542686
peanut butter	Nut Butters	0.97368956
cashews	Dried Fruits	0.9877151
mixed nuts	Dried Fruits	0.9845098
almonds	Dried Fruits	0.96738935

Query Classification Workflow In Solr

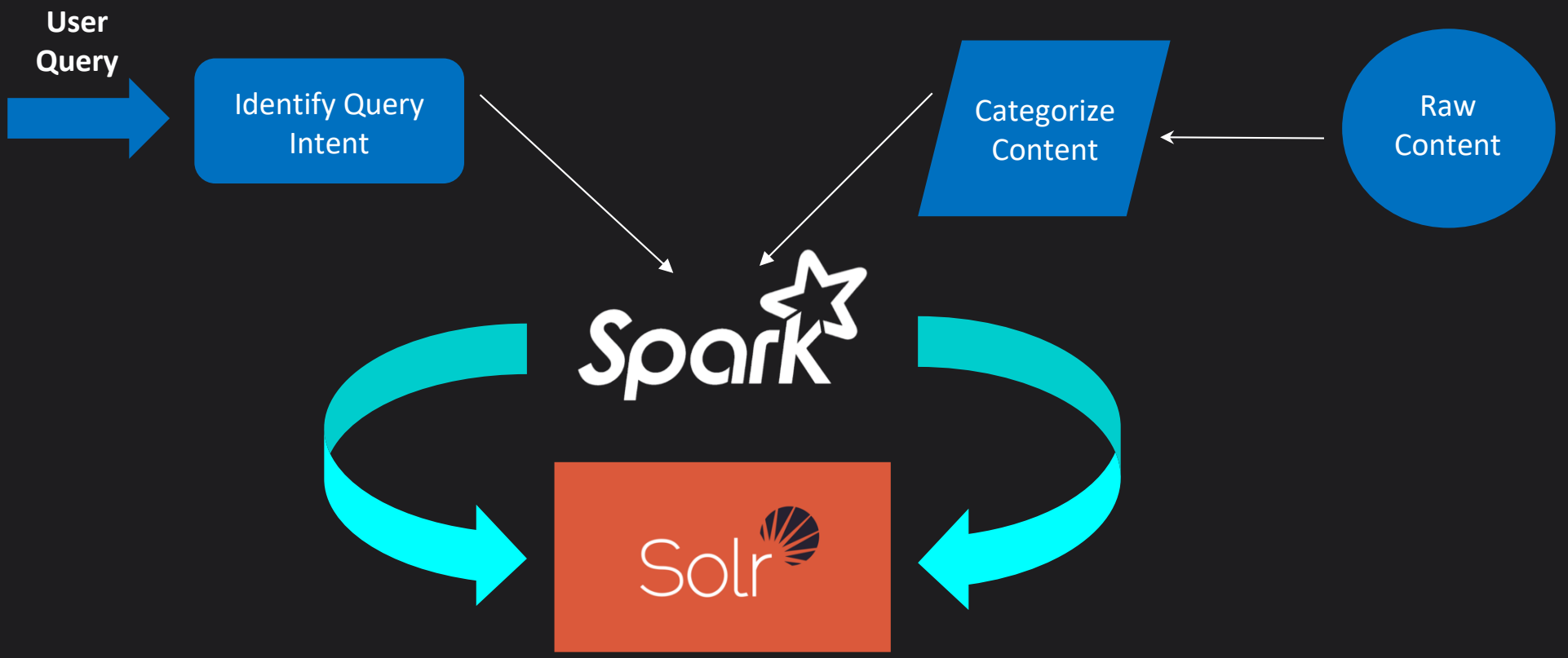


- Compute category classes
- Classify queries as domain specific entities

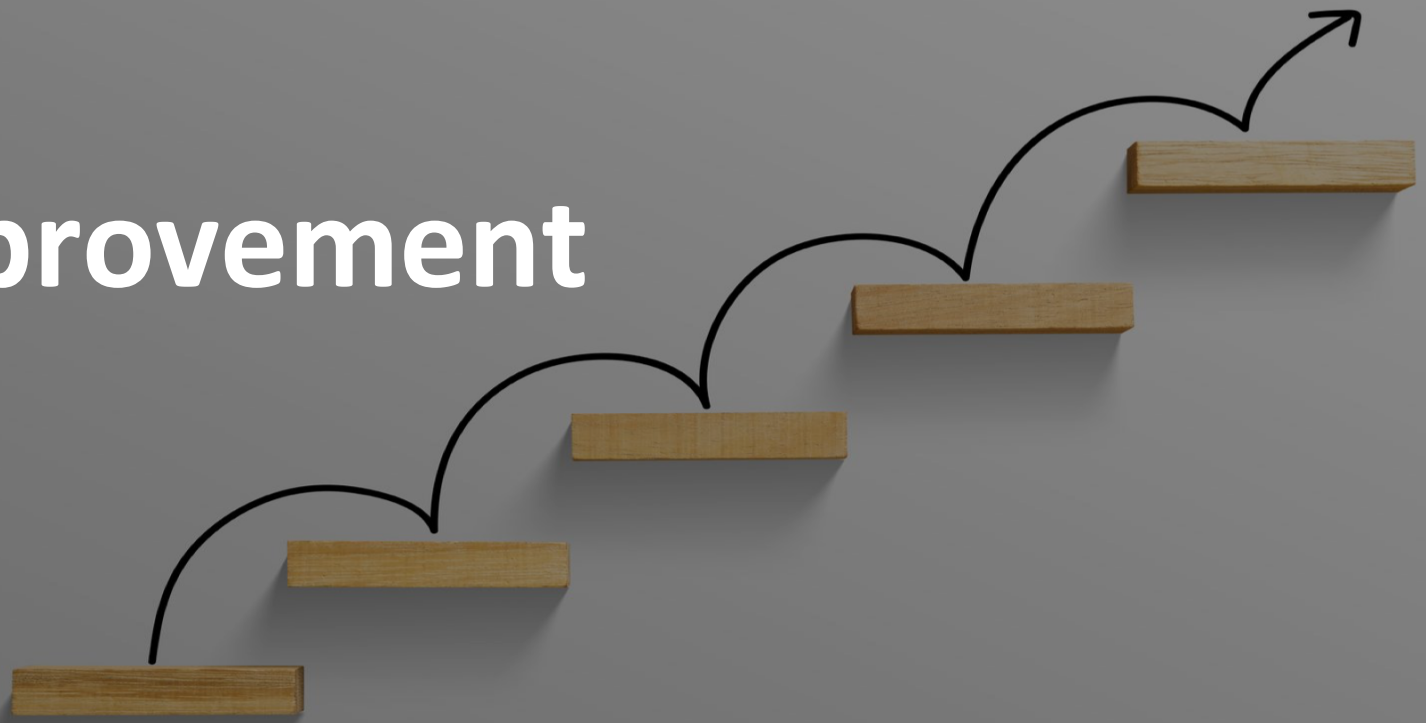
- Query as a searchable entity.
- Taxonomic features.

- Query model
- Boost content ranking

Collaboration!



Improvement



Measuring Search Relevance

Normalized Discounted Cumulative Gain (NDCG) is a popular method for measuring the quality of a set of search results. It asserts the following:

- **Cumulative Gain:** *0 => Not relevant, 1 => Near relevant, 2 => Relevant*

$$CG = \sum_{pos=1}^{n} relevance_{pos}$$

- **Discounting:** Positional bias

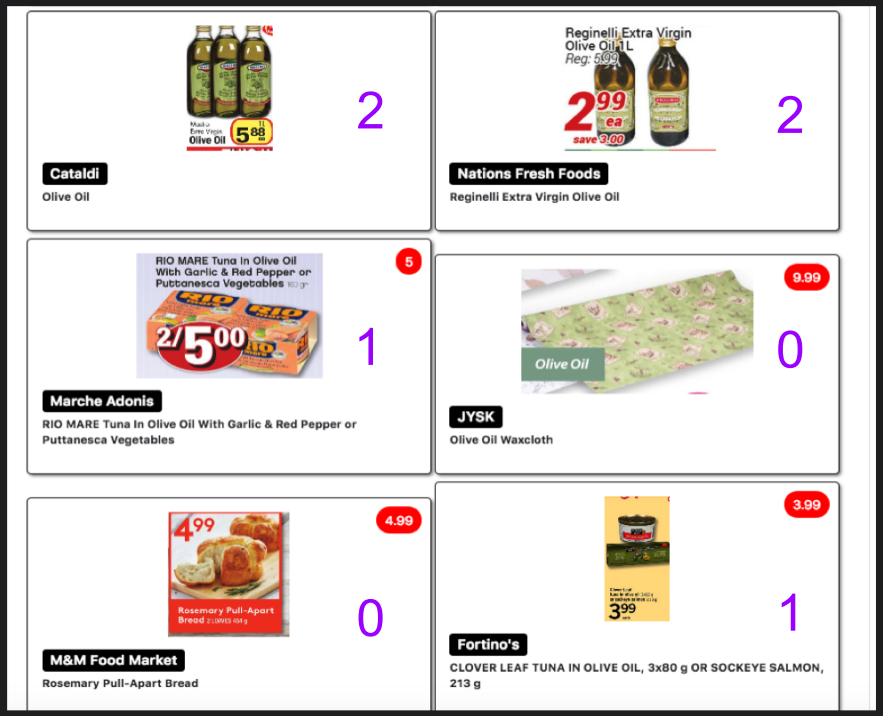
$$DCG = \sum_{pos=1}^{n} \frac{relevance_{pos}}{ln(pos+1)}$$

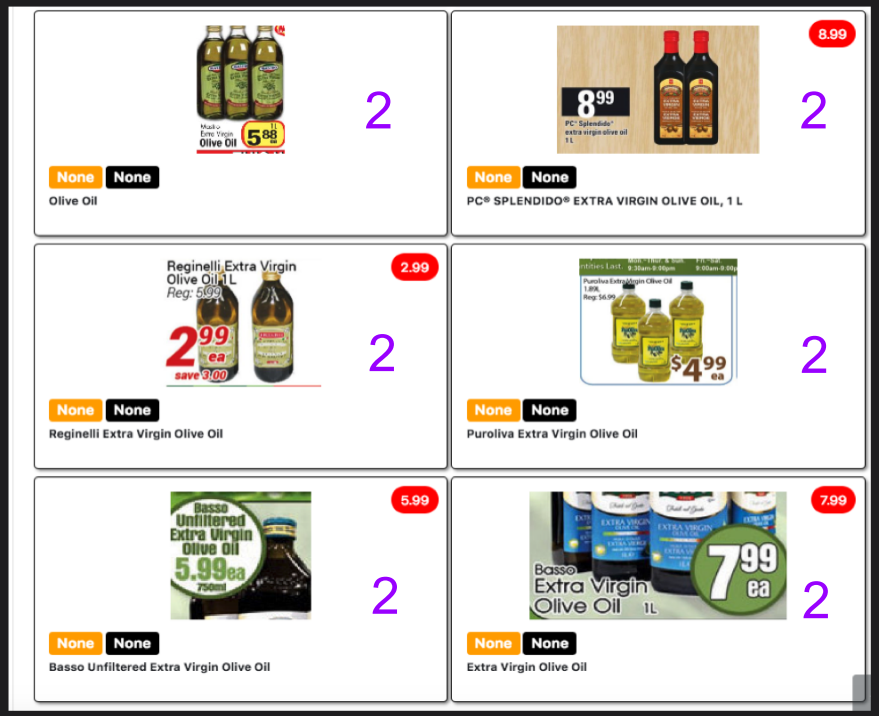
- **Normalization:** Ideal DCG (iDCG)

$$NDCG_{pos} = \frac{DCG_{pos}}{iDCG}$$

CG Improvement

Q: “Olive Oil”



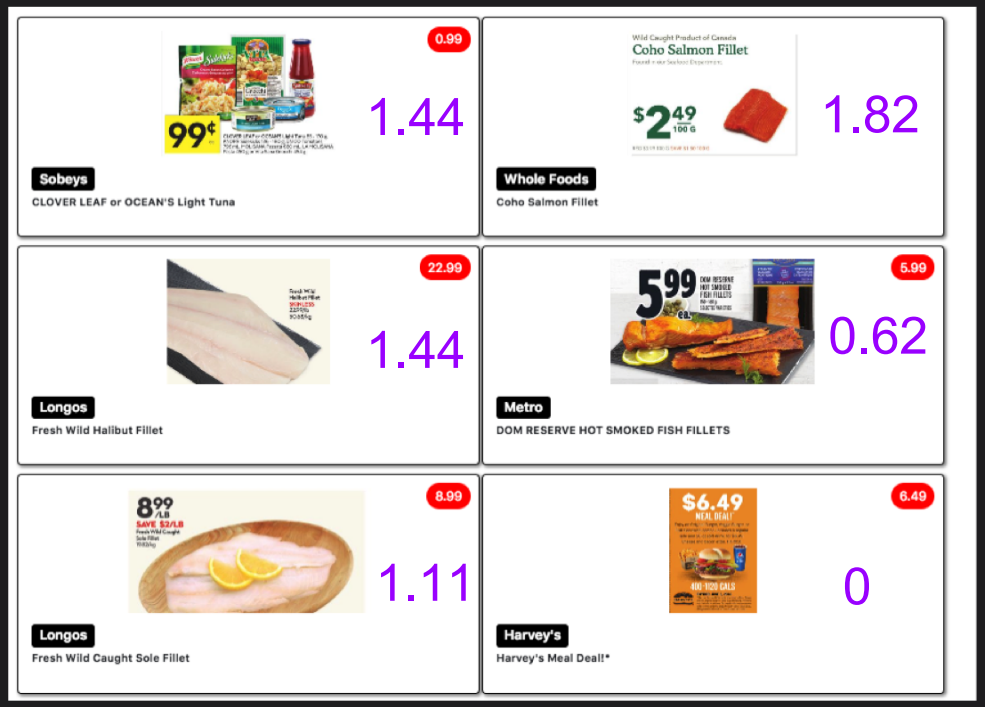


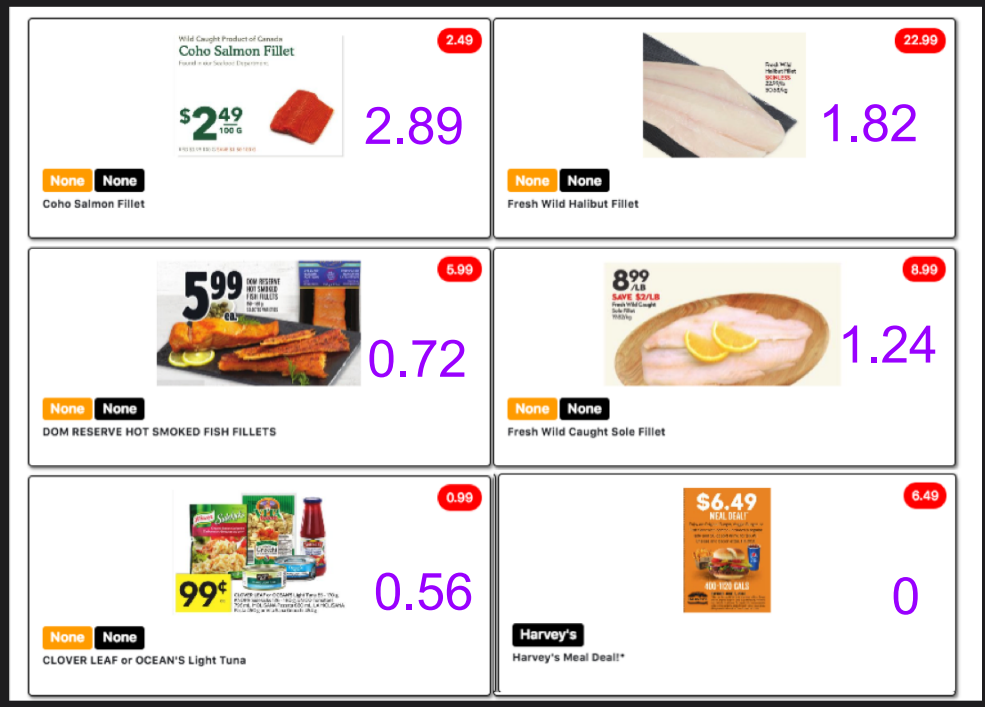
2+2+1+0+0+1= **6**

2+2+2+2+2+2= **12**

DCG Improvement

Q: “Fish”





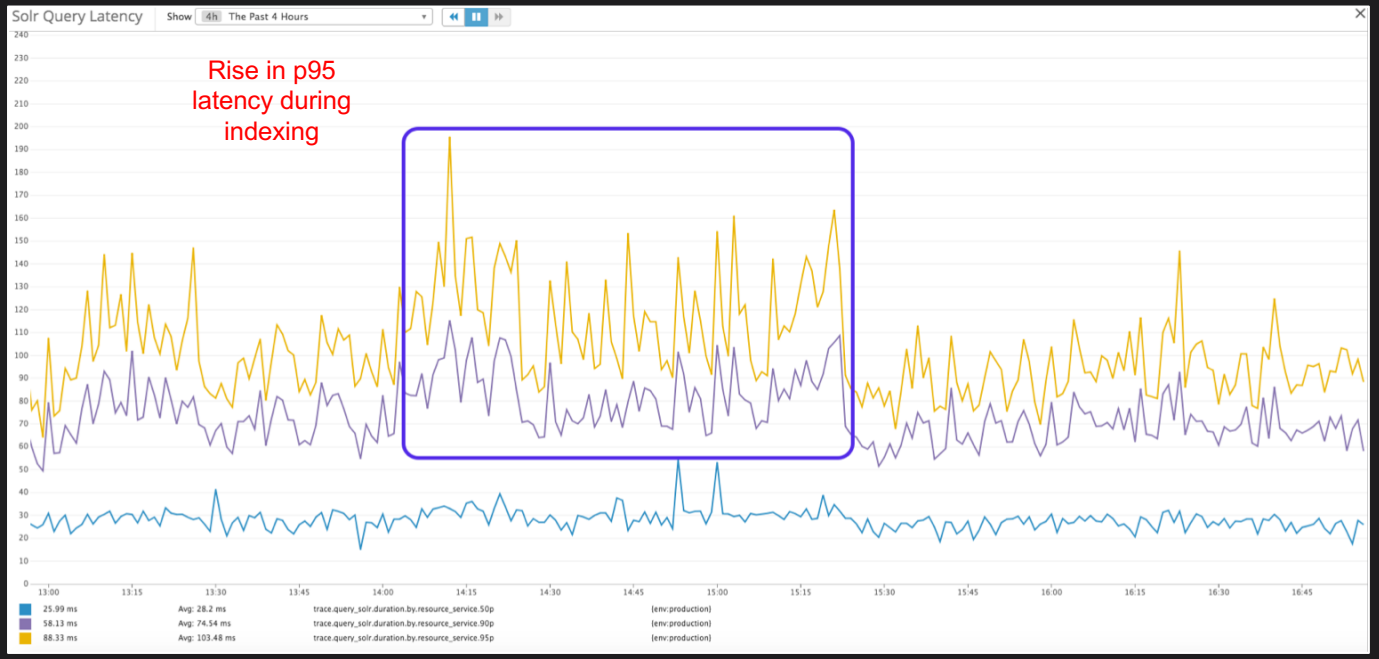
1.44+1.82+1.44+0.62+1.11+0= 6.43

2.89+1.82+0.72+1.24+0.56= **7.23**


Spark-Solr:
Lessons Learned

Lessons Learned

- Latency rise in p95 by 40% during indexing



Lessons Learned

- Read Parallelism
  - Match the number of spark executor nodes to the number of shards in your solr collection.
  - Prefer using solr filter params over `where` clause in Spark.

```
1 val zkhost = "localhost:9983"
2 val signalsOpts = Map("zkhost" -> zkhost, "collection" -> "ecommerce_signals", "query" -> "*:*", "fields" ->
  "count_i, doc_id_s, query_s", "solr.params" -> "fq=userId:[10 TO 1000]","splits" -> "true")
3 val signals = spark.read.format("solr").options(signalsOpts).load
```

- Indexing improvements
  - Multivalued fields may require extra processing.
  - Avoid using soft_commit_secs param in production.
  - Consider using smaller batch_size.

Key Takeaways

- The Spark-Solr framework unlocks a wide variety of machine learning techniques to be applied to search applications.
- Enables faster data analytics in SparkSQL than traditional SQL frameworks.
- Spark-Solr can be used to optimize solr indexing with signal data.
- Storing text classification models in solr has significant semantic benefits over other persistence layers.

Thank You!