Ranking: Why you never go past the first page on a search engine

Overview of the presentation

1. Brief introduction to Information Retrieval (IR)
    a. Stating the problem
2. Search related concepts
3. Tools
    a. Learning to Rank
    b. Recommender systems
4. Stages of a ranking project
5. Advices

Based on The NIST Text Retrieval Conference (TREC) Logo

Introduction to IR

- IR is part of computer science which studies the retrieval of information from a collection of written documents. The retrieved documents aim at satisfying a user information need usually expressed in natural language.[1]

1 -Modern Information Retrieval by Baeza-Yates and Ribeiro-Neto

Introduction to IR - Stating the problem

Tobias

Introduction to IR - Stating the problem

Tobias

Introduction to IR - Stating the problem

Tobias

Introduction to IR - Stating the problem

Tobias

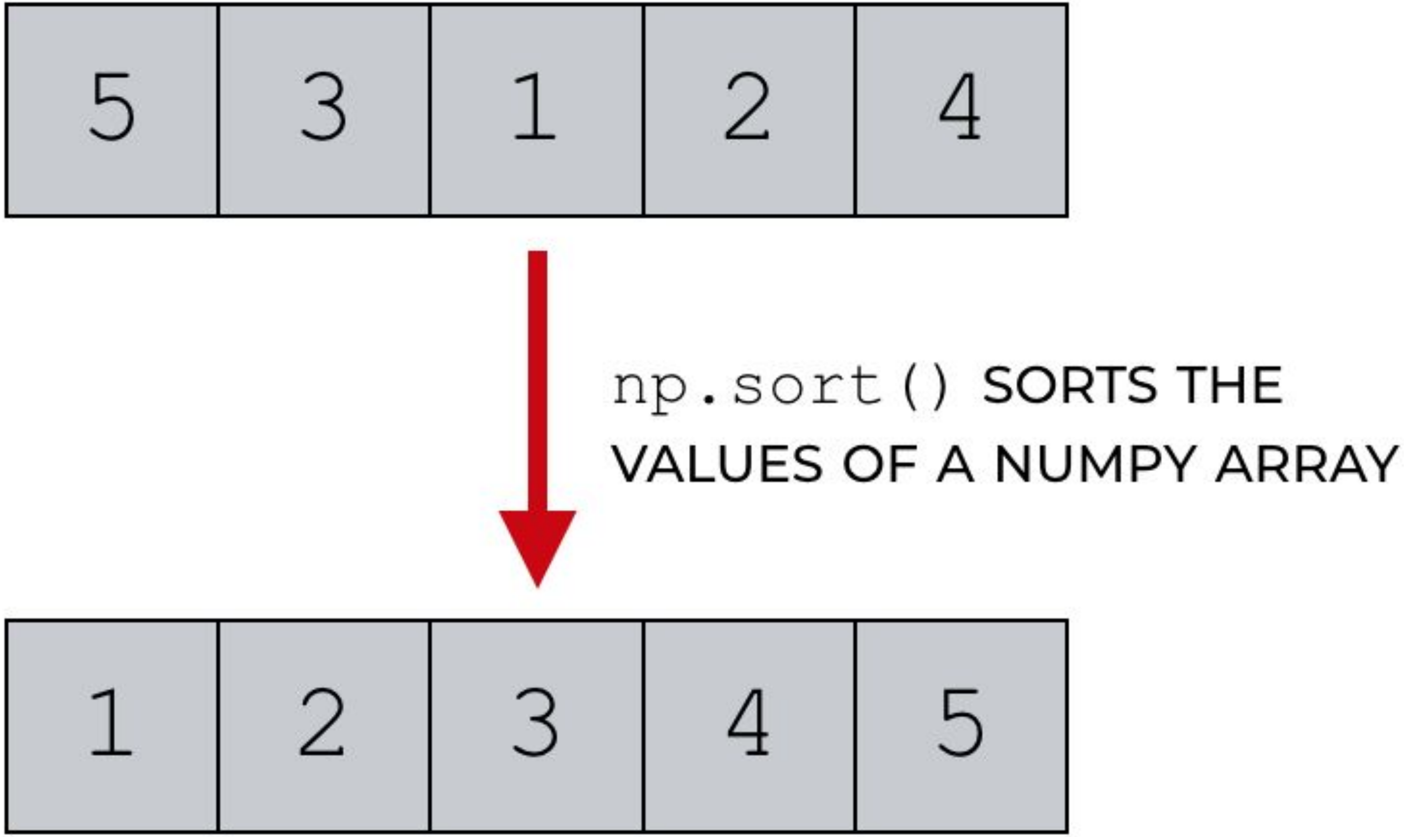
1-48 of over 40,000 results for **Clothing, Shoes & Jewelry** : **Men** : **Clothing** : **Shirts** : **T-Shirts**

Introduction to IR - Stating the problem

Tobias

1-48 of over 40,000 results for **Clothing, Shoes & Jewelry** : **Men** : **Clothing** : **Shirts** : T-Shirts

40k T-shirts


5 3 1 2 4
np.sort() SORTS THE VALUES OF A NUMPY ARRAY
1 2 3 4 5

Introduction to IR - Stating the problem

1. User - Tobias
2. Documents - 10 million products
3. Query - Filter -> Clothing Men -> T-shirts

Introduction to IR - Stating the problem

1. User - Tobias
2. Documents - 10 million products
3. Query - Filter -> Clothing Men -> T-shirts

A. Tom - Business owner
B. Ron - Engineer
C. Harry - Architect

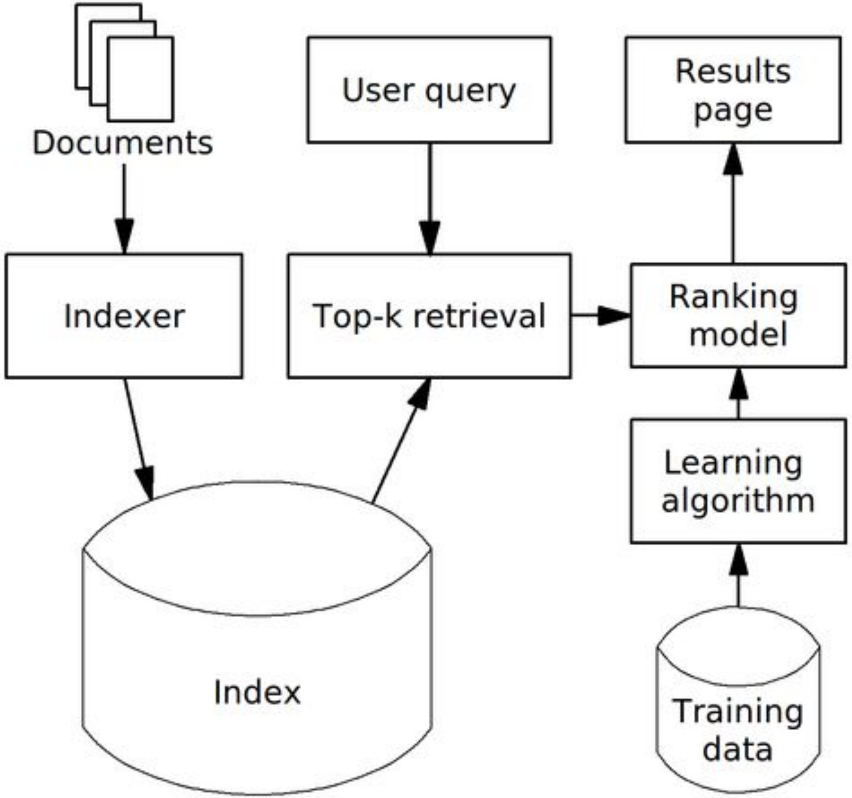
Introduction to IR - Stating the problem

1. User - Tobias
2. Documents - 10 million products
3. Query - Filter -> Clothing Men -> T-shirts

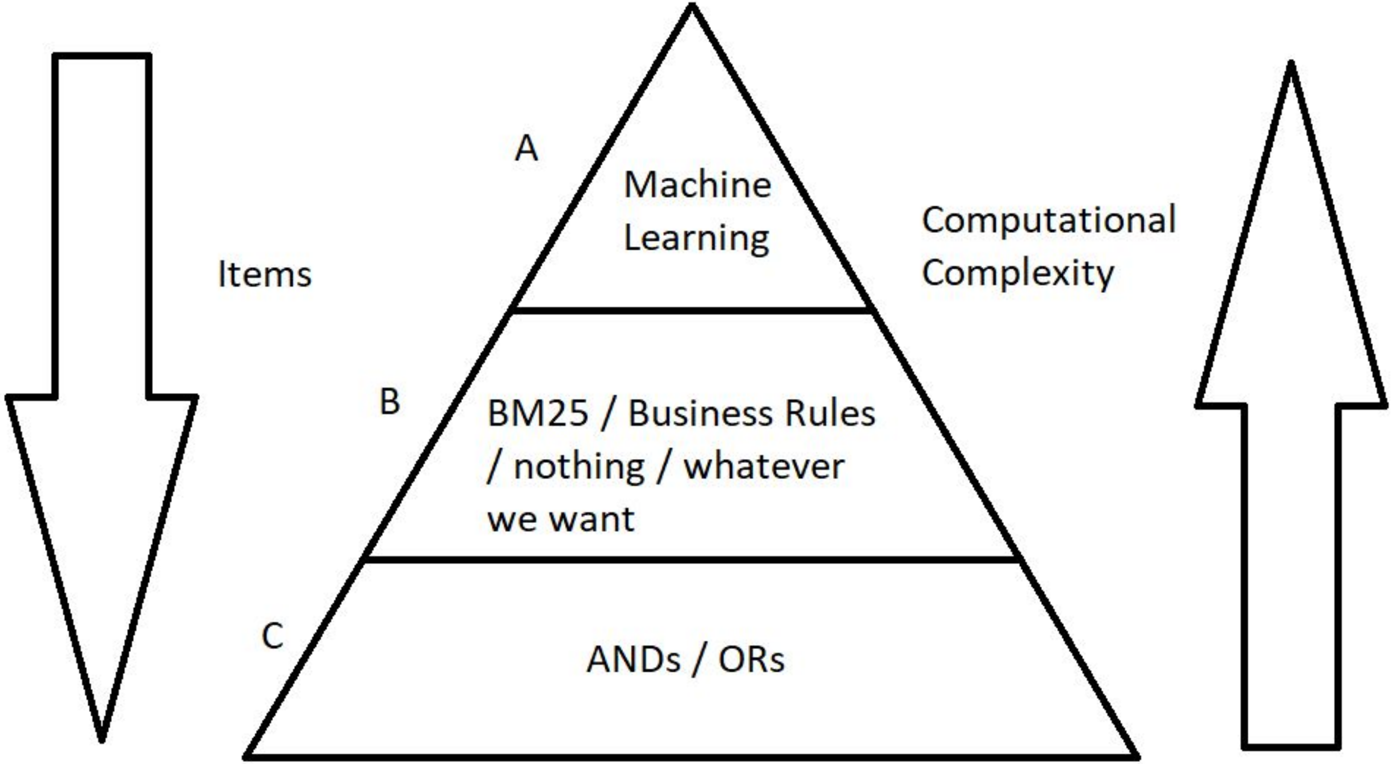
A. Tom - Business owner
B. Ron - Engineer
C. Harry - Architect

<- Data scientist

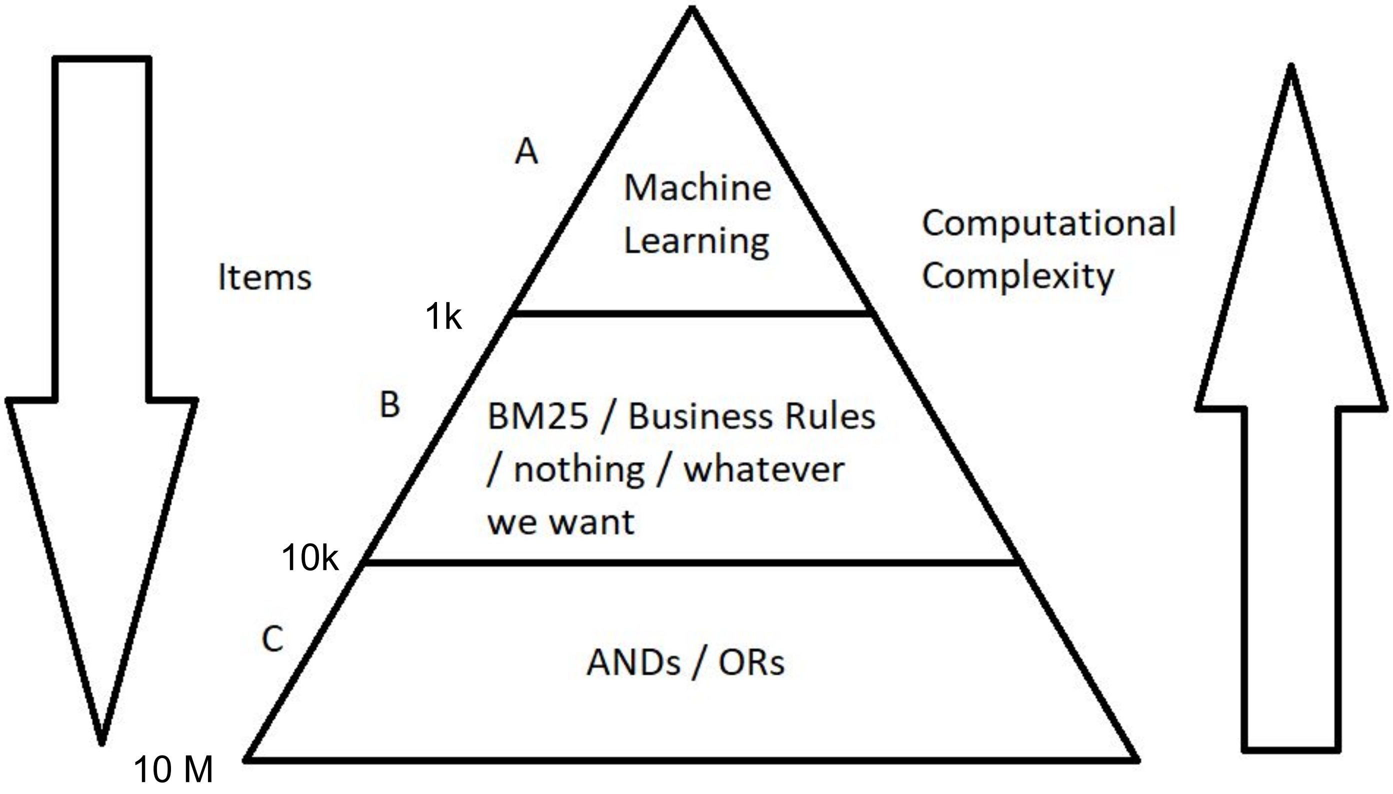
Search related concepts - Example of a Search Engine



Search related concepts - Example of a Search Engine

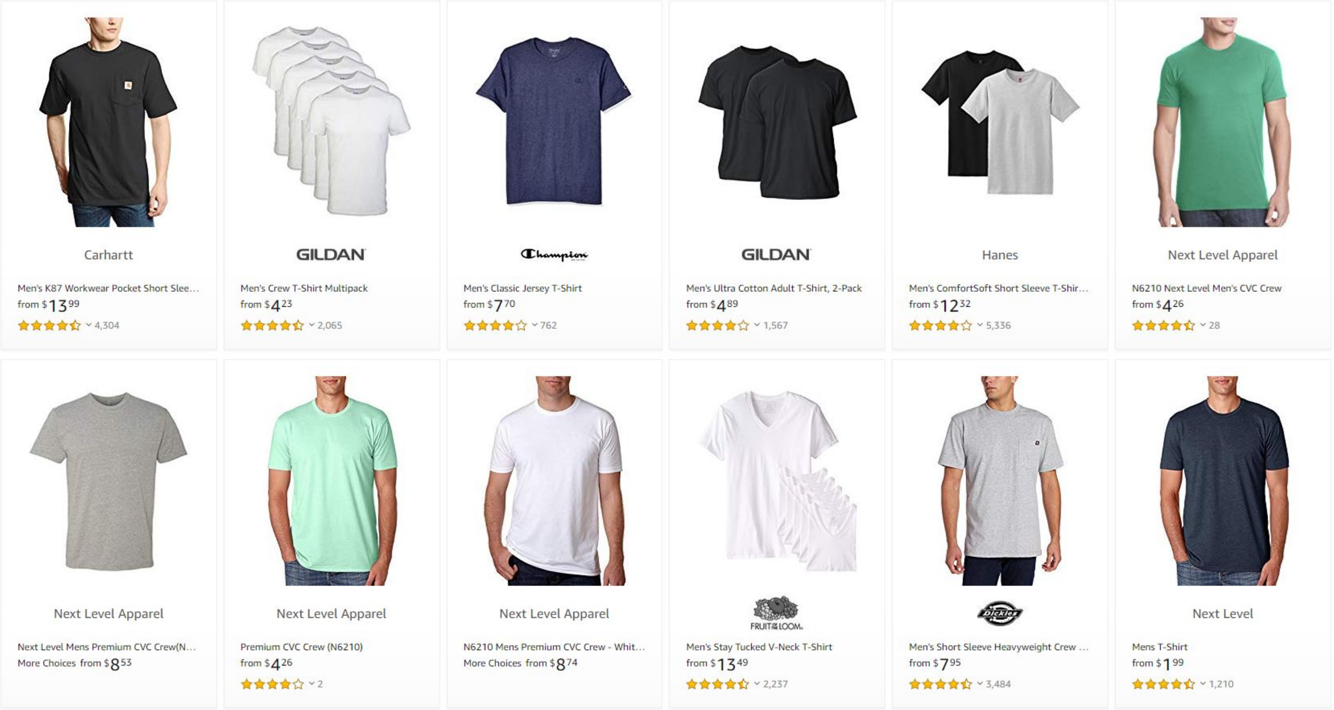


Search related concepts - Example of a Search Engine



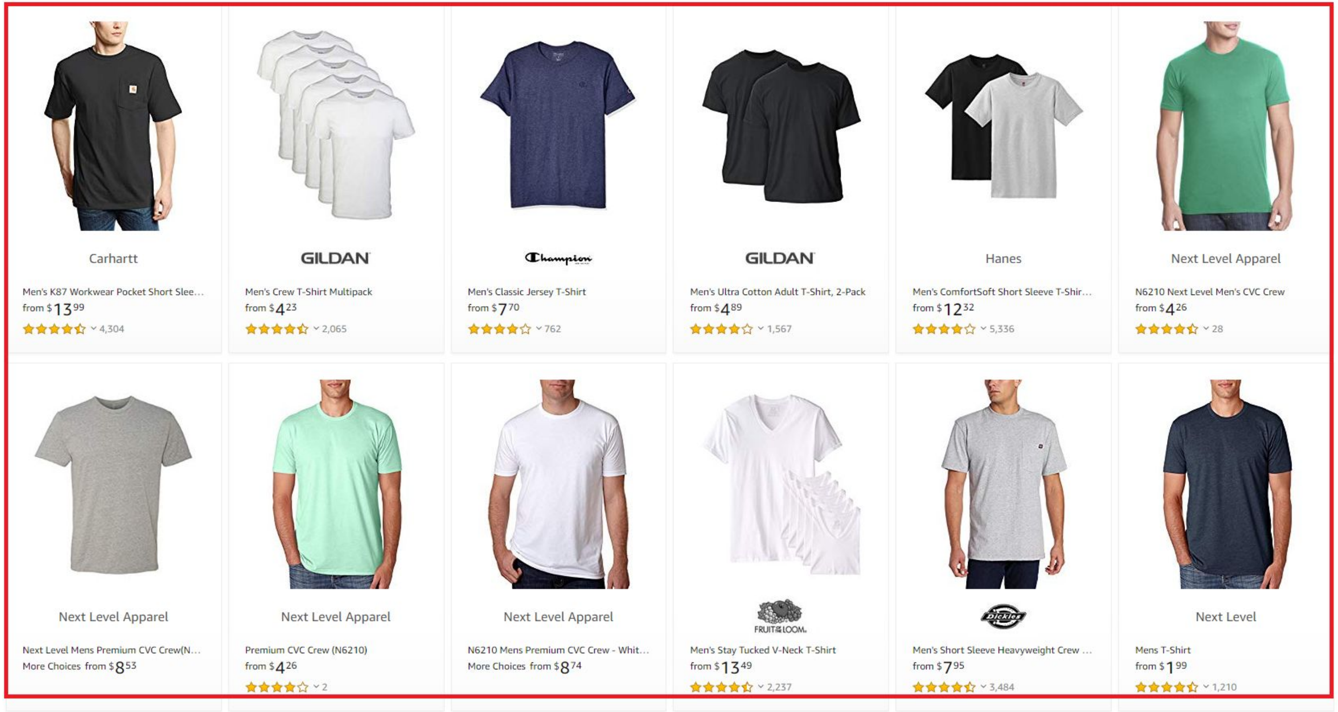
Tools - Learning to Rank

1. Query
2. Document
3. Relevance



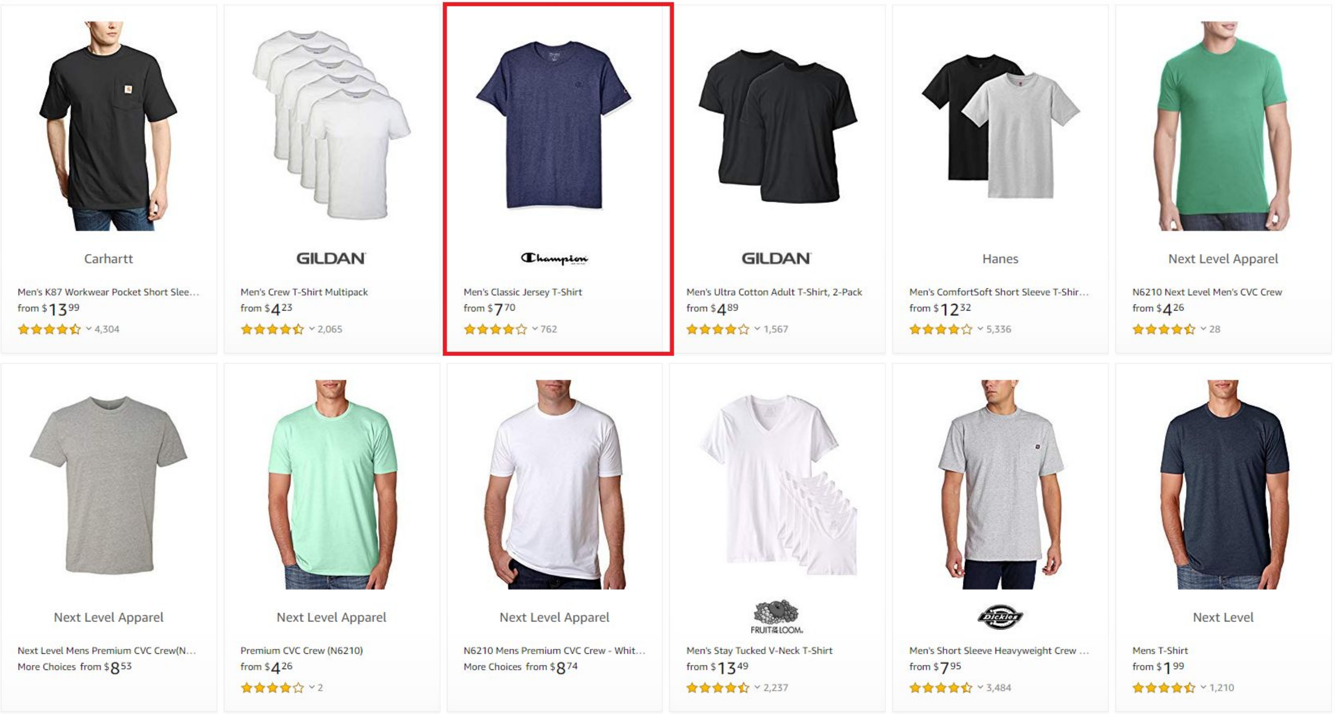
Tools - Learning to Rank

1. Query
2. Document
3. Relevance



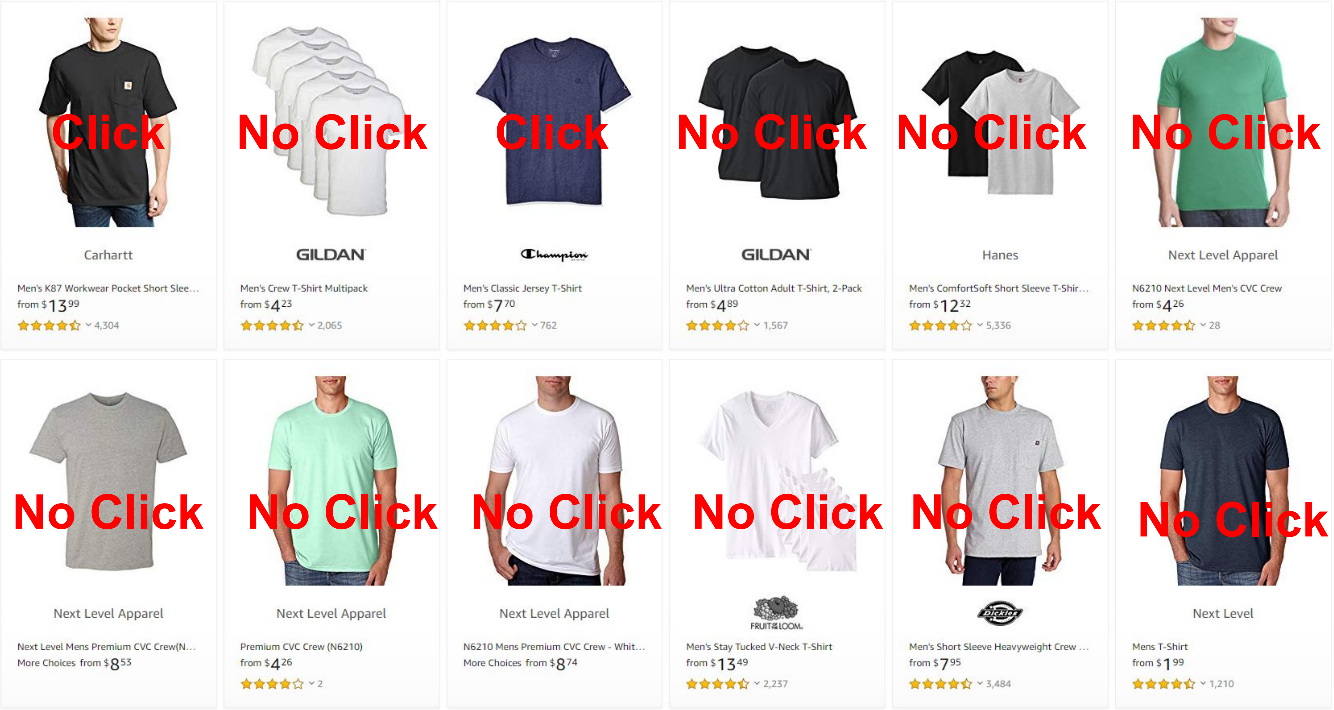
Tools - Learning to Rank

1. Query
2. Document
3. Relevance



Tools - Learning to Rank

1. Query
2. Document
3. Relevance

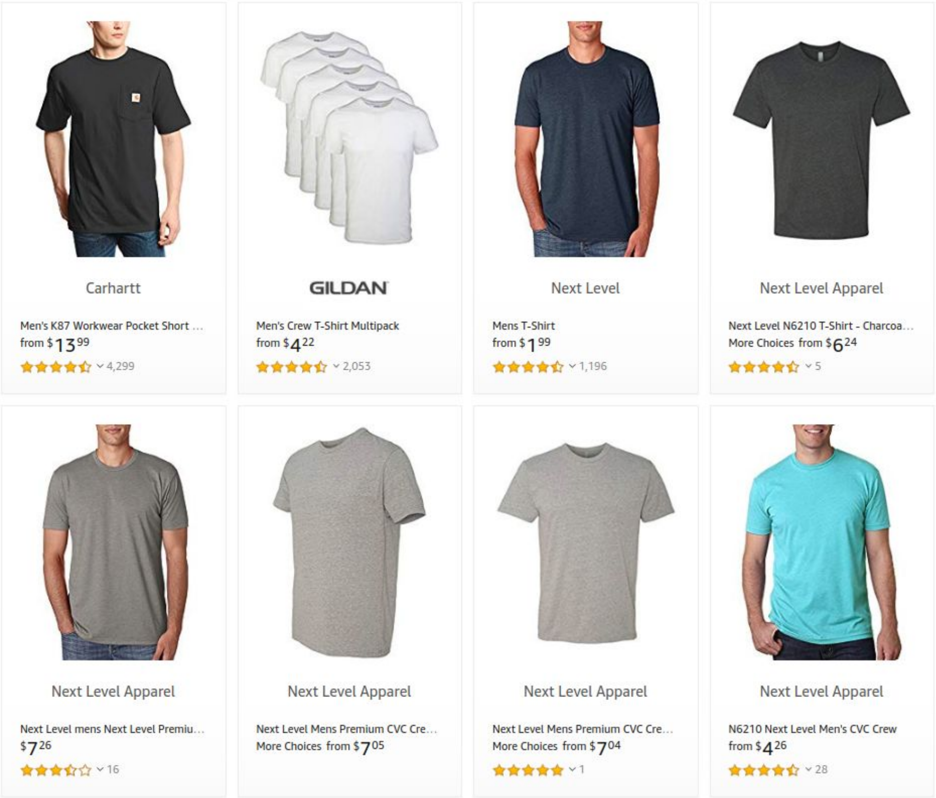


Tools - Learning to Rank

	A	B	C	D
1	query_id	product_id	relevance	features ...
2	1	1	1	
3	1	2	0	
4	1	3	1	
5	1	4	0	
6	1	5	0	
7	1	6	0	
8	1	7	5	
9	2	6	0	
10	2	7	0	
11	2	8	0	
12	2	1	0	
13	2	2	0	
14	2	3	1	
15	2	4	0	

Tools - Learning to Rank - How to measure relevance

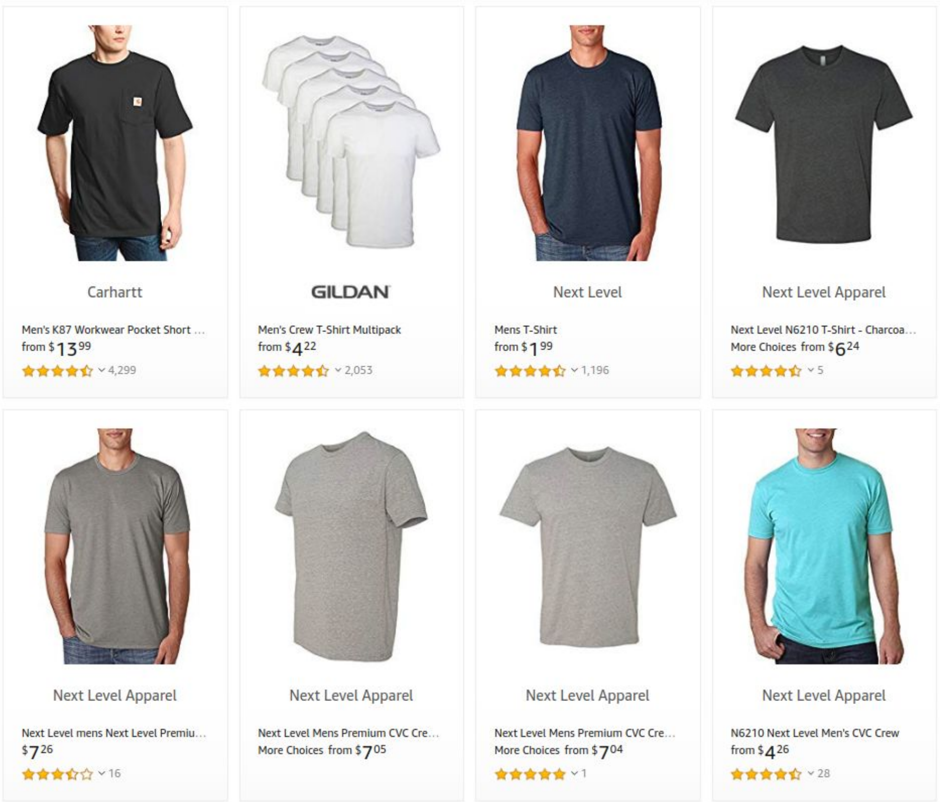
How should we determine what is relevant to a user?



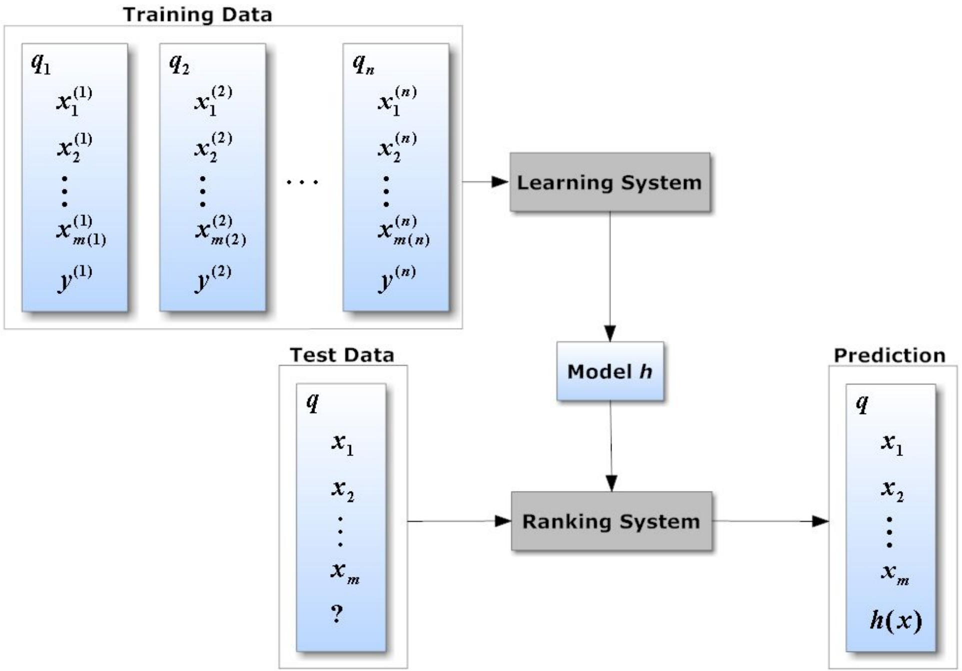
Tools - Learning to Rank - How to measure relevance

How should we determine what is relevant to a user?

- Experts manually label observations
- Using business related metrics to infer relevancy:
  - 5 for transactions
  - 1 for click
  - 0 for seen and did nothing



Tools - Learning to Rank



Tools - Learning to Rank

Pointwise	Pairwise	Listwise
Take **one candidate**	Take **a pair of candidates**	Take **the entire list**
⇩	⇩	⇩
Compute **score** between candidate and query	Given a **pair of candidates** decide which one **rank higher**	optimise its **order**
	hypothesis: more important the relative position	

algorithms:
anything that deals with regression problems

algorithms:
RankNet
LambdaRank*
...

LambdaMART

algorithms:
SoftRank
ListNet
AdaRank
LambdaRank*

Tools - Learning to Rank - RankNet

```
import numpy as np
from keras import backend
from keras.layers import Activation, Dense, Input, Subtract
from keras.models import Model


INPUT_DIM = 50

# Model.
h_1 = Dense(128, activation="relu")
h_2 = Dense(64, activation="relu")
h_3 = Dense(32, activation="relu")
s = Dense(1)

# Relevant document score.
rel_doc = Input(shape=(INPUT_DIM,), dtype="float32")
h_1_rel = h_1(rel_doc)
h_2_rel = h_2(h_1_rel)
h_3_rel = h_3(h_2_rel)
rel_score = s(h_3_rel)
```

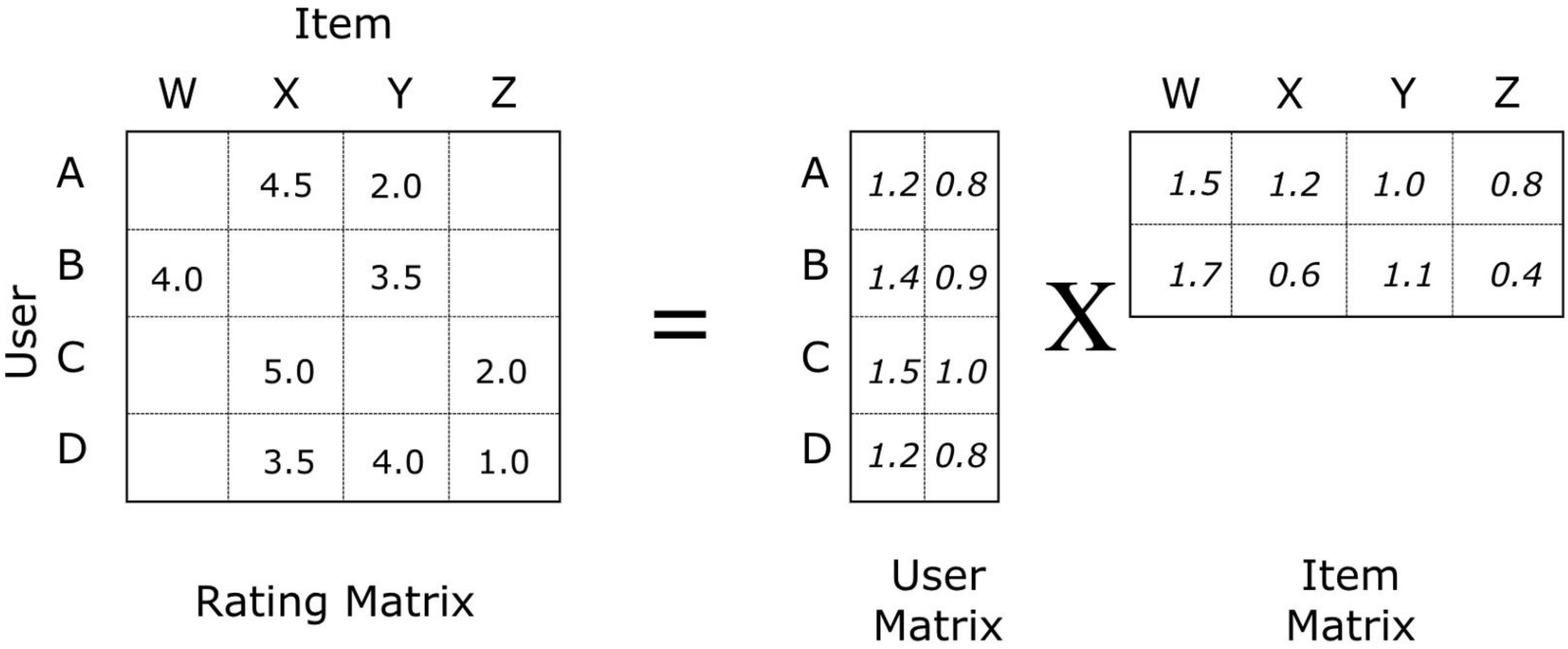
```
# Irrelevant document score.
irr_doc = Input(shape=(INPUT_DIM,), dtype="float32")
h_1_irr = h_1(irr_doc)
h_2_irr = h_2(h_1_irr)
h_3_irr = h_3(h_2_irr)
irr_score = s(h_3_irr)

# Subtract scores.
diff = Subtract()([rel_score, irr_score])

# Pass difference through sigmoid function.
prob = Activation("sigmoid")(diff)

# Build model.
model = Model(inputs=[rel_doc, irr_doc], outputs=prob)
model.compile(optimizer="sgd", loss="binary_crossentropy")
```

Credits: https://github.com/airalcorn2/RankNet/blob/master/ranknet.py

Tools - Learning to Rank - Evaluation

- **Mean Average Precision**
- **NDCG@k**
- **Recall**
- **AB test with relevant metrics**

Tools -   Recommender systems - Matrix factorization



Stages of a ranking project - Airbnb

Credit: Machine Learning-Powered Search Ranking of Airbnb Experiences blogpost by Mihajlo Grbovic

Stages of a ranking project

Stage 1
Offline ML model

- **Data size:** Small
- **Signals:**
  - Experience Features
- **Scoring:** Offline

Example:

Predict the best position for the products by country and update them on a weekly basis.

Credit: Machine Learning-Powered Search Ranking of Airbnb Experiences blogpost by Mihajlo Grbovic

Stages of a ranking project

Stage 2 Personalized Offline ML model

- **Data size**: Medium
- **Signals:**
  - Experience Features
  - User Features
- **Scoring**: Offline

Example:

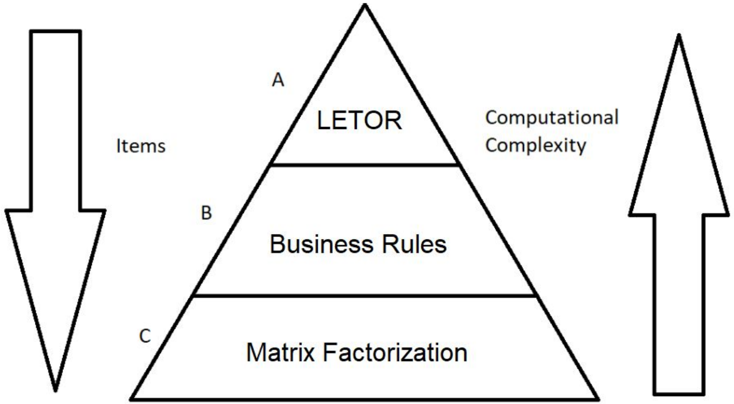
Predict the best position for the products by country **and several user segments** and update them on a **daily** basis.

Credit: Machine Learning-Powered Search Ranking of Airbnb Experiences blogpost by Mihajlo Grbovic

Stages of a ranking project

Stage 3 Personalized Online ML model

- **Data size**: Large
- **Signals:**
  - Experience Features
  - User Features
  - Query Features
- **Scoring**: Online



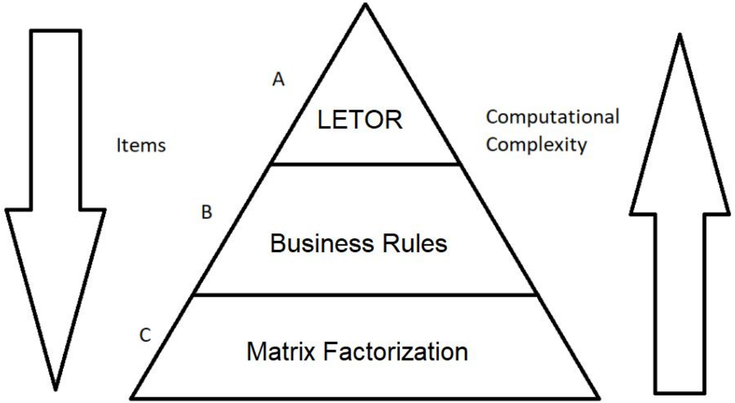
Credit:
- Machine Learning-Powered Search Ranking of Airbnb Experiences blogpost by Mihajlo Grbovic
- https://github.com/layer6ai-labs/vl6_recsys2018

You haven’t answered the question in the title for the talk yet!

You haven’t answered the question in the title for the talk yet!

- Relevant items are finite. It’s possible that all of them are on the first page

- It’s also possible that the second page is loaded using a cheaper method



Advices

- There are a lot of interesting ways of building a ranking engine, just remember the core concepts
- Take it one step at a time
- Building a ranking engine in a notebook is very different from building one in a production environment
- Have fun

Questions?

Articles:

- Two-stage Model for Automatic Playlist Continuation at Scale by Maksims Volkovs
- On Application of Learning to Rank for E-Commerce Search by Shubhra Santu et al.
- Turning Clicks into Purchases: Revenue Optimization for Product Search in E-Commerce by Liang Wu et al.
- From RankNet to LambdaRank to LambdaMART: An Overview by Christopher J.C. Burges
- Cascade Ranking for Operational E-commerce Search by Shichen Liu, Fei Xiao et al.
- Amazon Search: The Joy of Ranking Products by Daria Sorokina et al.

Books:

- Learning to Rank for Information Retrieval by By Tie-Yan Liu
- Learning to Rank for Information Retrieval and Natural Language Processing by Hang Li

Blog posts:

- Machine Learning-Powered Search Ranking of Airbnb Experiences blogpost by Mihajlo Grbovic