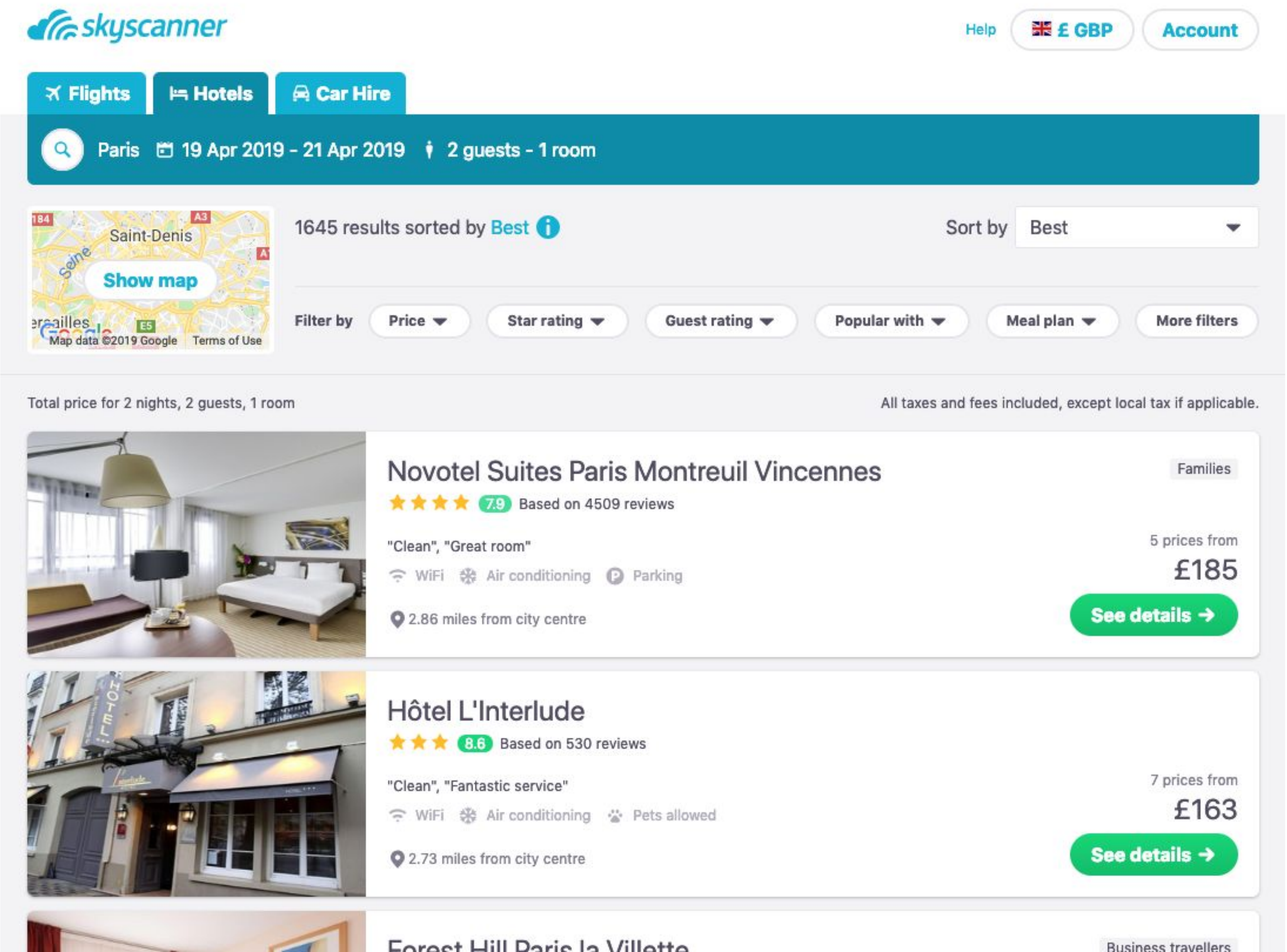
AI from zero to production: lessons from Skyscanner

Konstantin Halachev • SKYSCANNER

Using AI to rank hotels

Every business makes daily expert decisions on what to show to whom and when. Those decisions can often be improved with AI

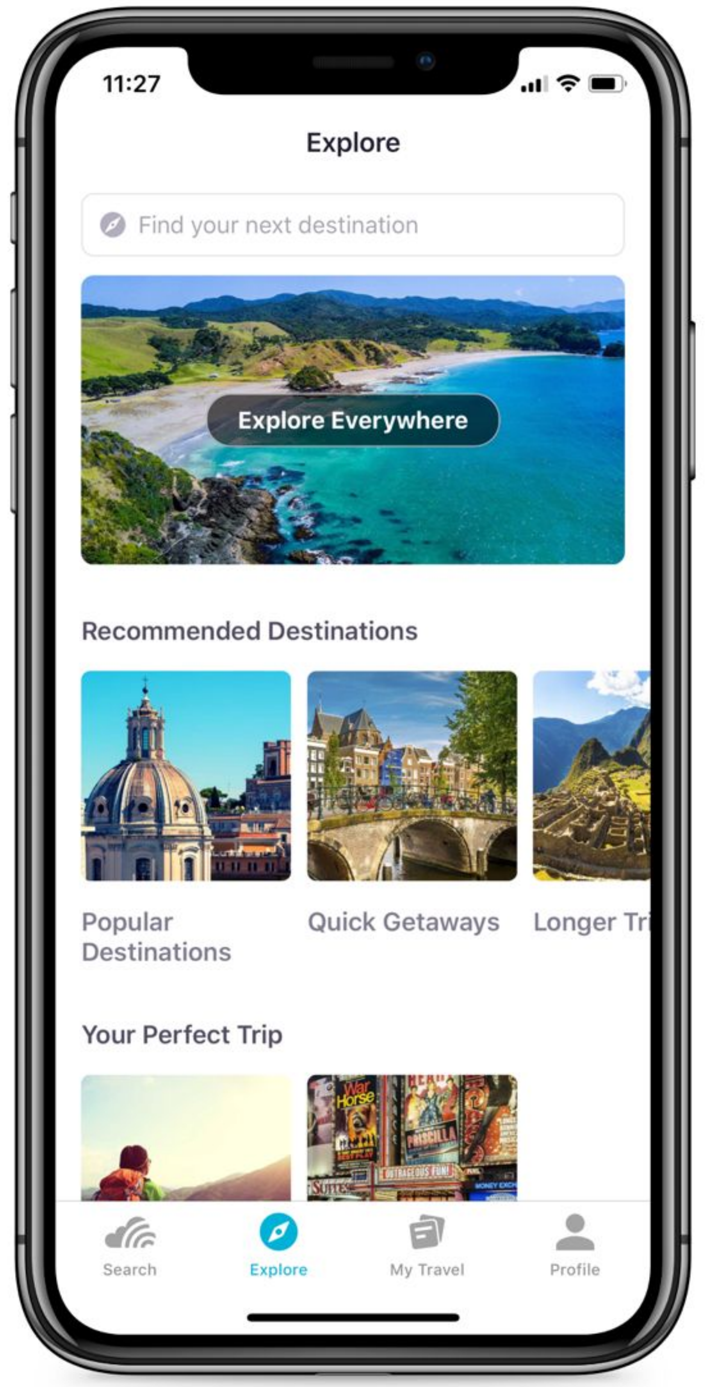


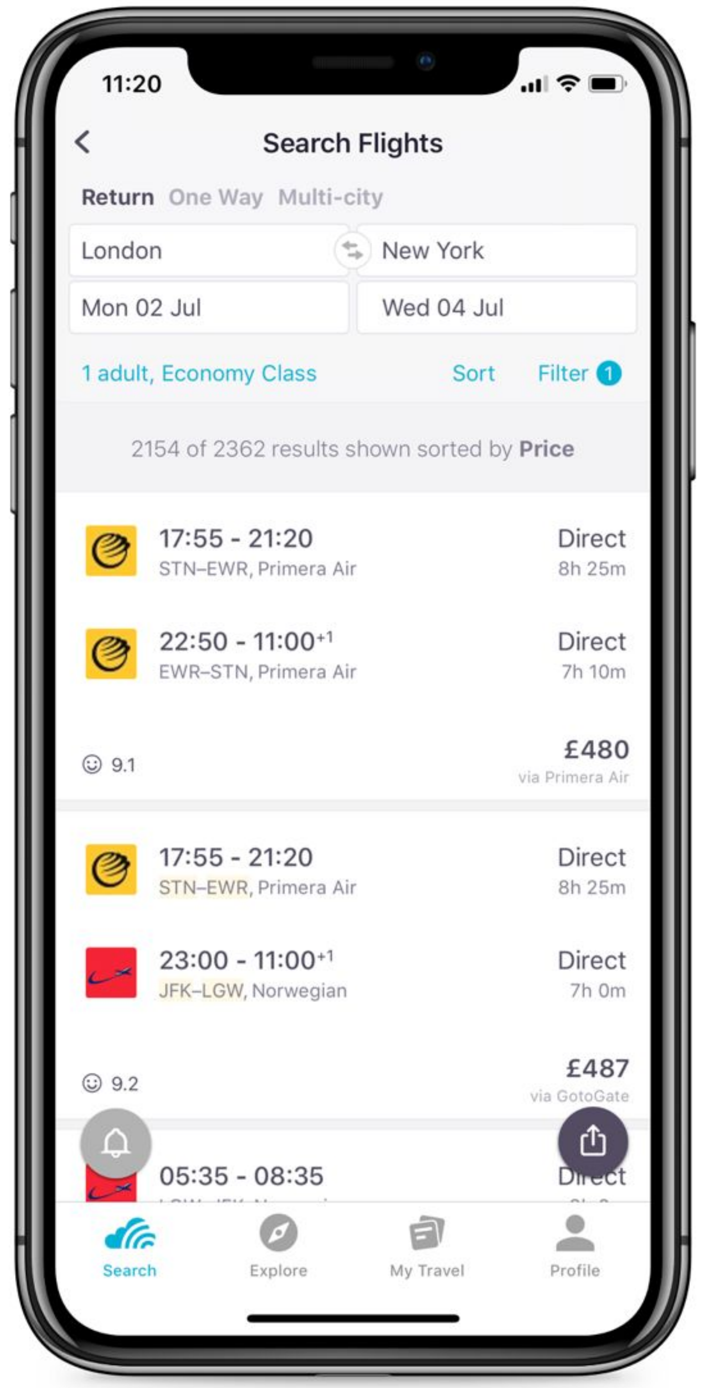
About Skyscanner

skyscanner

The world’s travel search engine

- Powering flights, hotels and car hire.
- Average 70m unique monthly visitors
- Over 70m app downloads



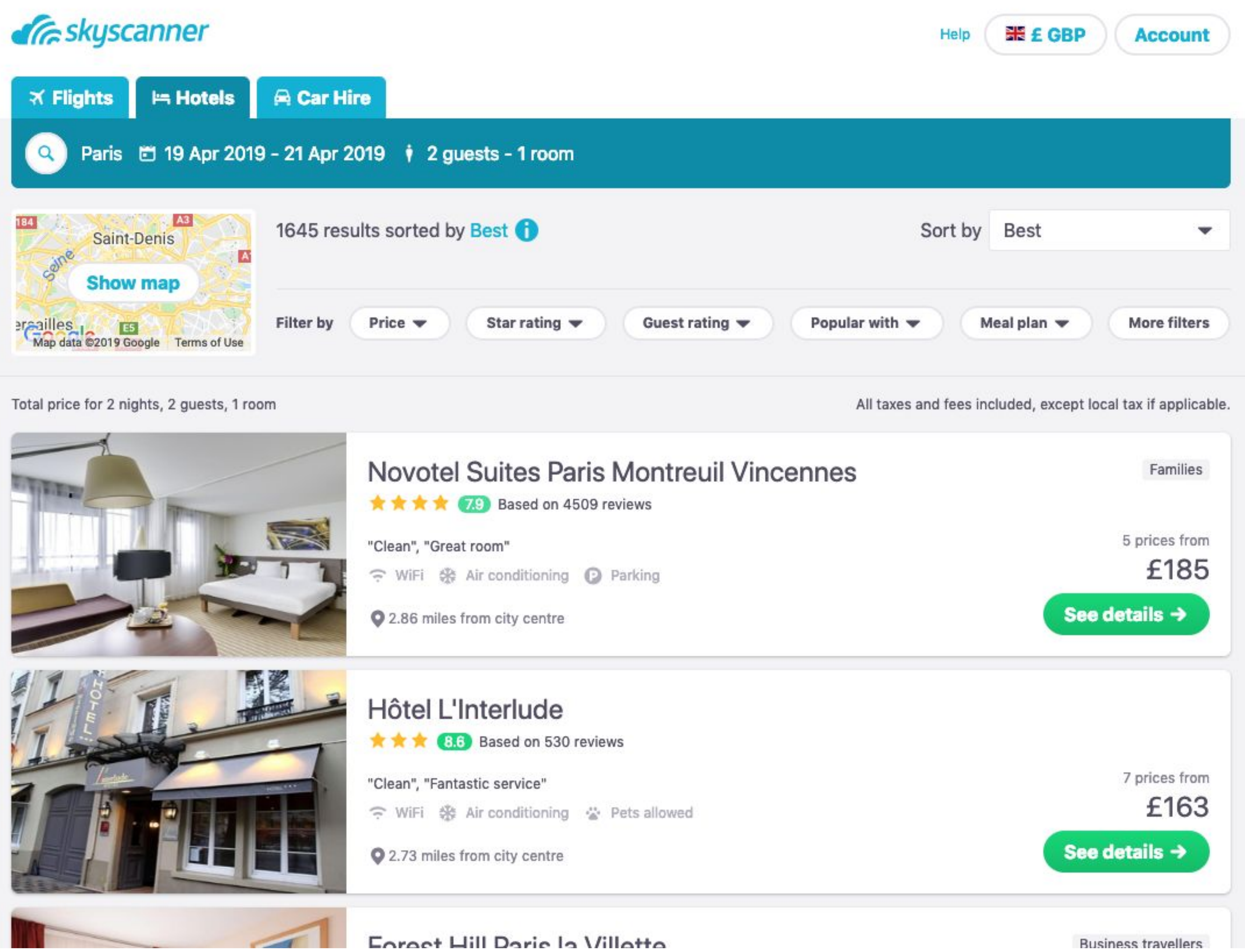


About me

- Konstantin Halachev
- Currently an Engineering manager in Skyscanner
- Previously a Data Scientist
- Previously a PhD in Bioinformatics from MPII

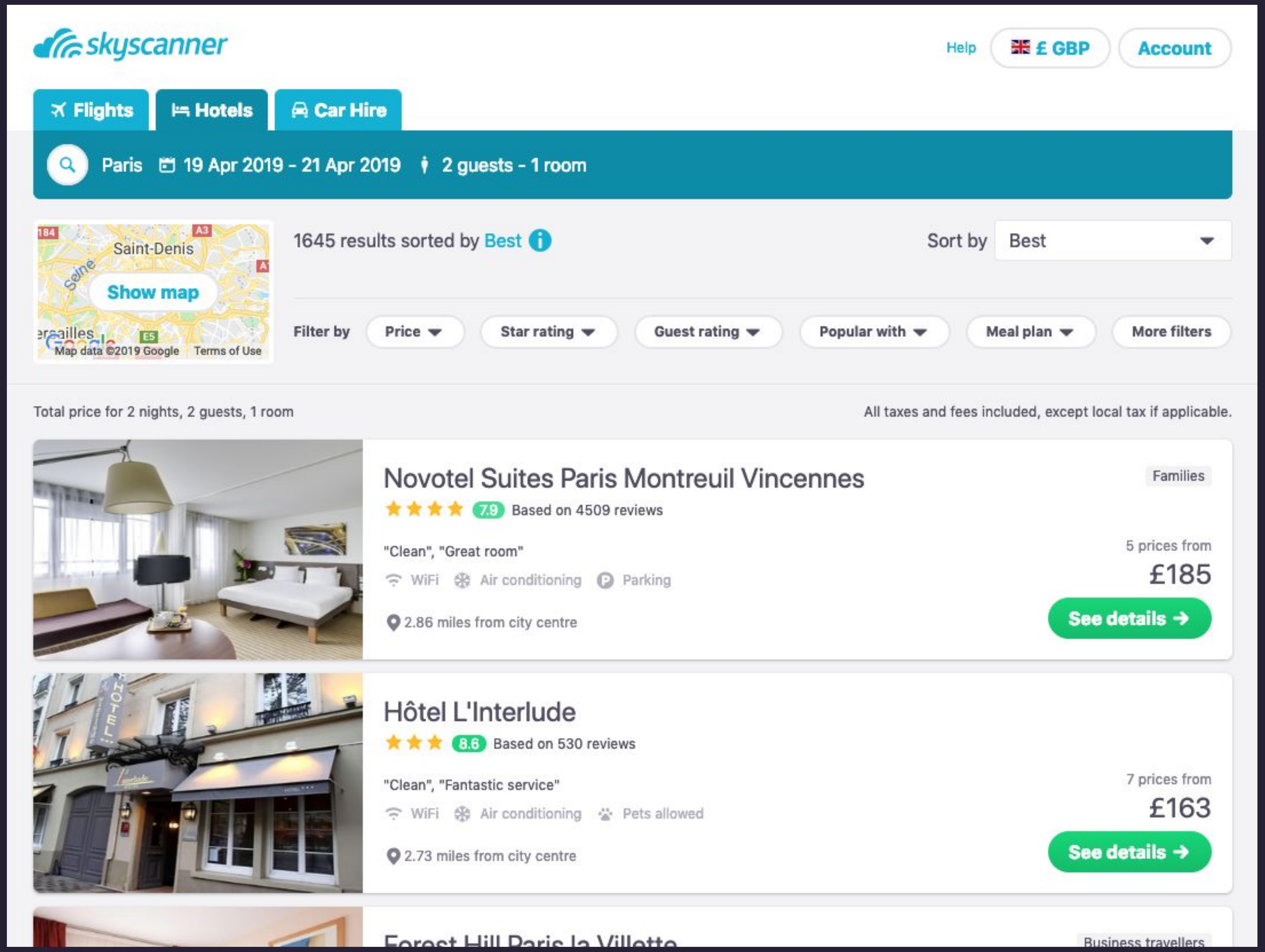
Lessons from ranking hotels

Skyscanner hotel search



Skyscanner's hotel recommendations

1. Business need



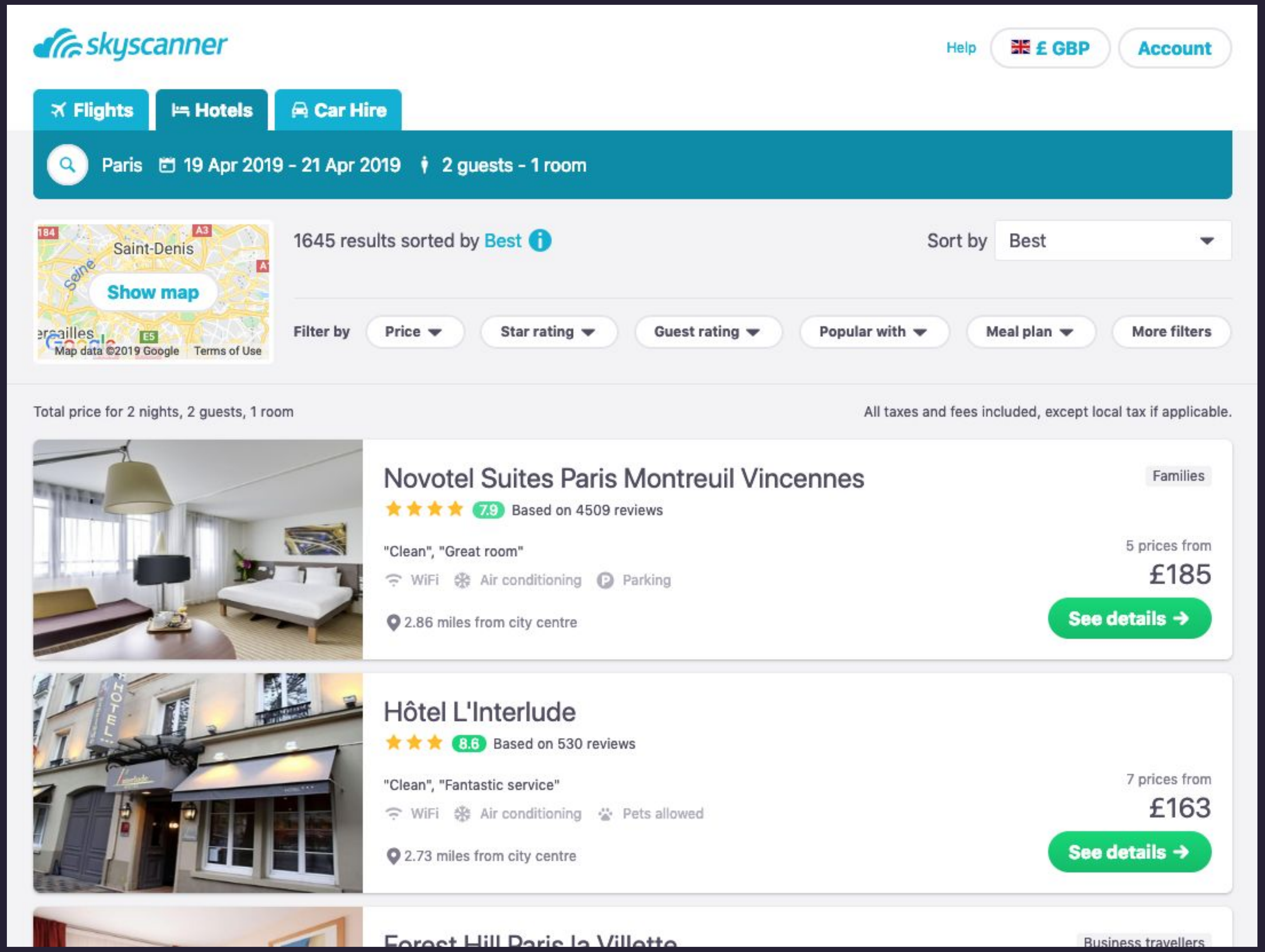
Goal of the user: *To find a good hotel*

Goal of Skyscanner: *To offer users good hotels*

Many ways to approach this …

Skyscanner's hotel recommendations

1. Business need



Goal of the user: *To find a good hotel*

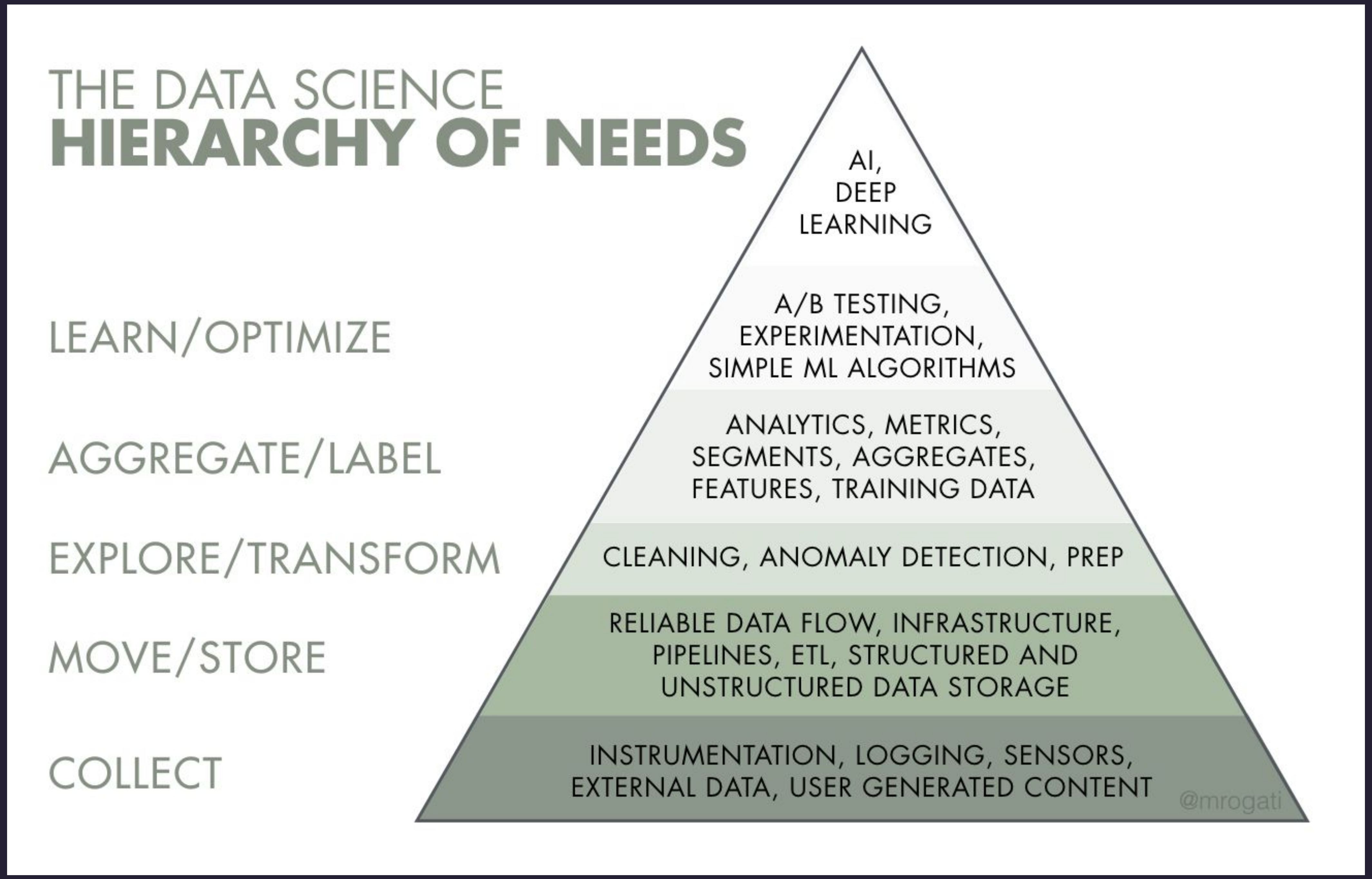
Goal of Skyscanner: *To offer users good hotels*

Many ways to approach this …

We worked on: *Which hotels are most relevant to a given user, for a given search at a specific time?*

Which hotels are most relevant to a given user, for a given search at a specific time?

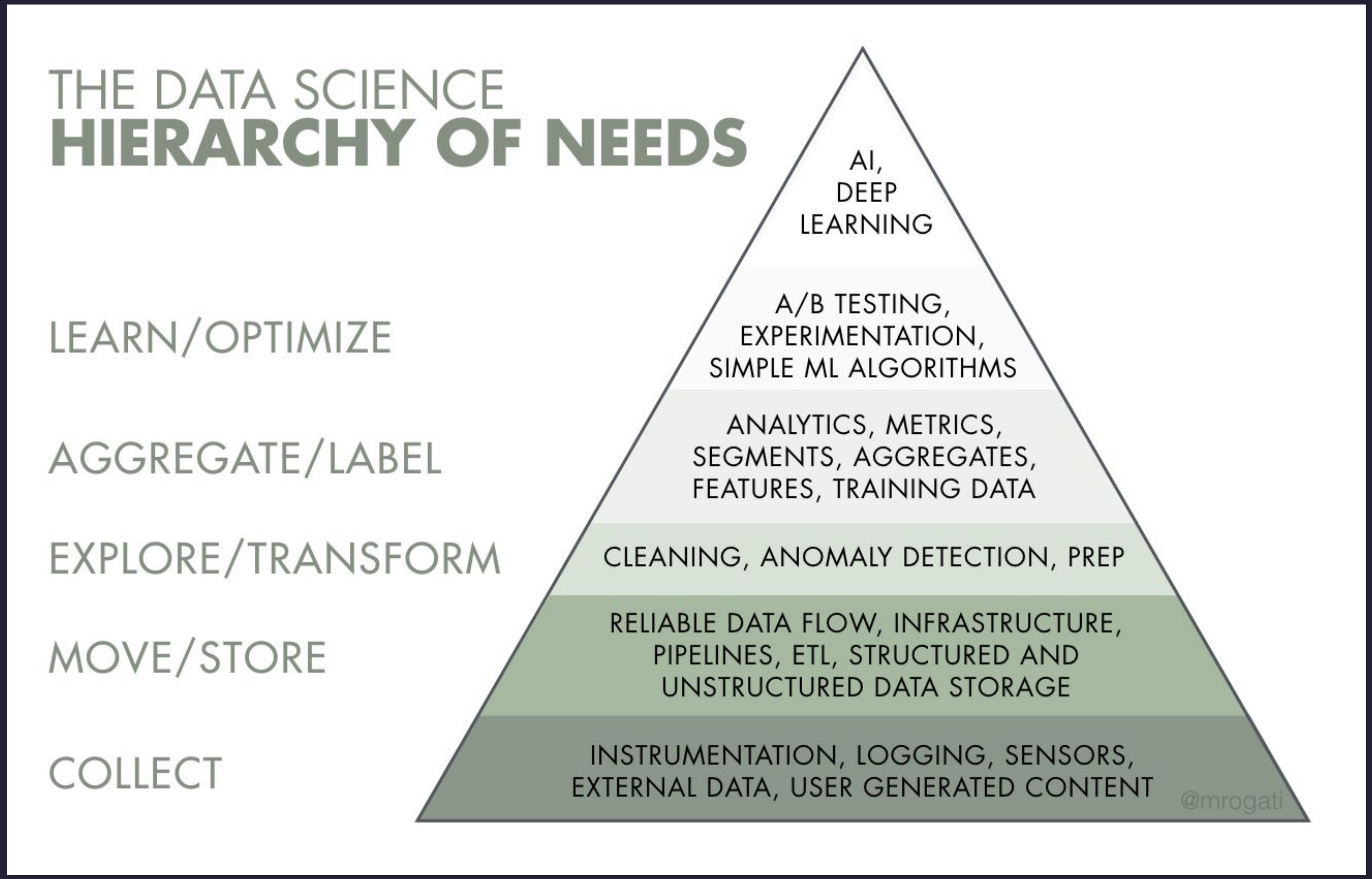
2. Data



Source: https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

Which hotels are most relevant to a given user, for a given search at a specific time?

2. Data

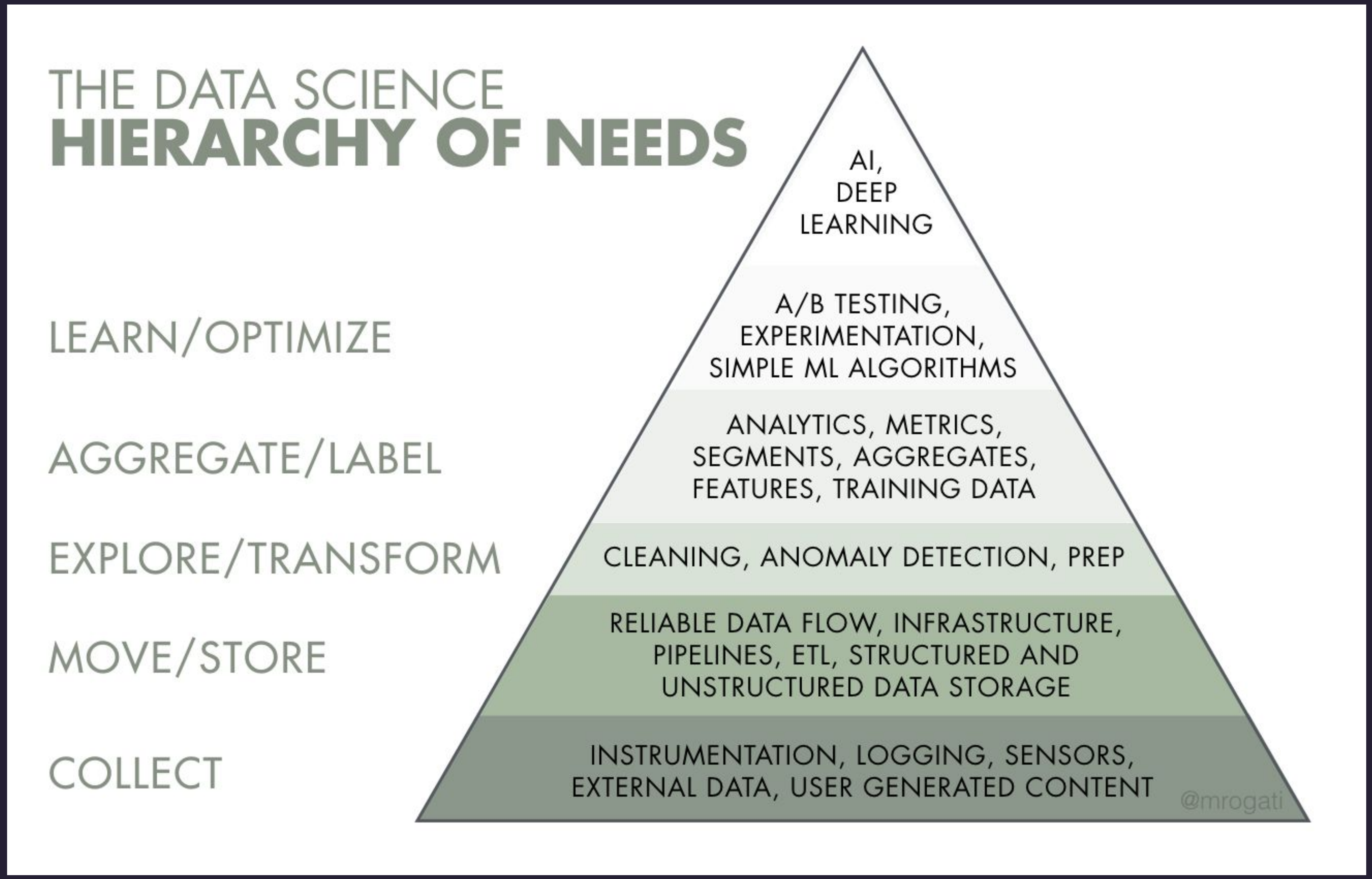


Source: https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

- Do we have data to answer the question?
  - What have users searched and seen?
  - Do we store the hotels shown?
  - Do we store their positions?
  - Do we store prices, ratings, location, etc?
- Is the data correct?
  - Do we store all searches?
  - Are the searches in the same format?
  - Are there values that do not make sense?
  - Are there outliers?
  - Has the UI or UX changed?
- Do we have enough data?
  - 100 data points are not enough
  - Data for 1 day is not enough
  - Millions of data points may be

__Which hotels are most relevant to a given user, for a given search at a specific time?__

2. Data

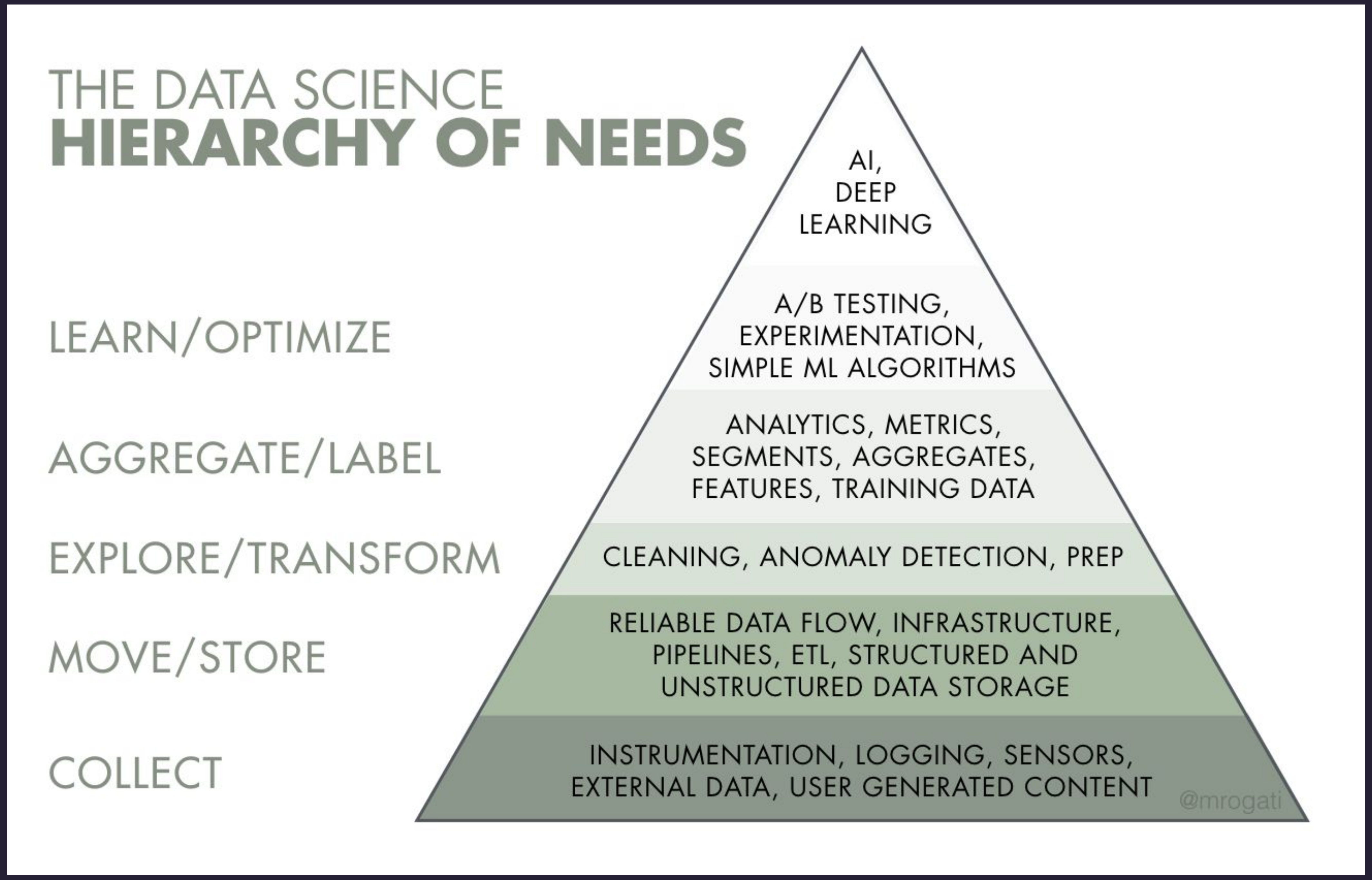


Source: https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

- Do we have data to answer the question?
  - What have users searched and seen?
  - Do w the hotels sh ?
  - eir
  - we sto ri ngs, loc tc?
- e dat c ?
  - a es?
  - the sea ch e same for a
  - e there v ue o not ma se
  - here er
  - X cha
- Do we have enough data?
  - 100 data points are not enough
  - Data for 1 day is not enough
  - Millions of data points may be

Assume we have a new ranking algorithm

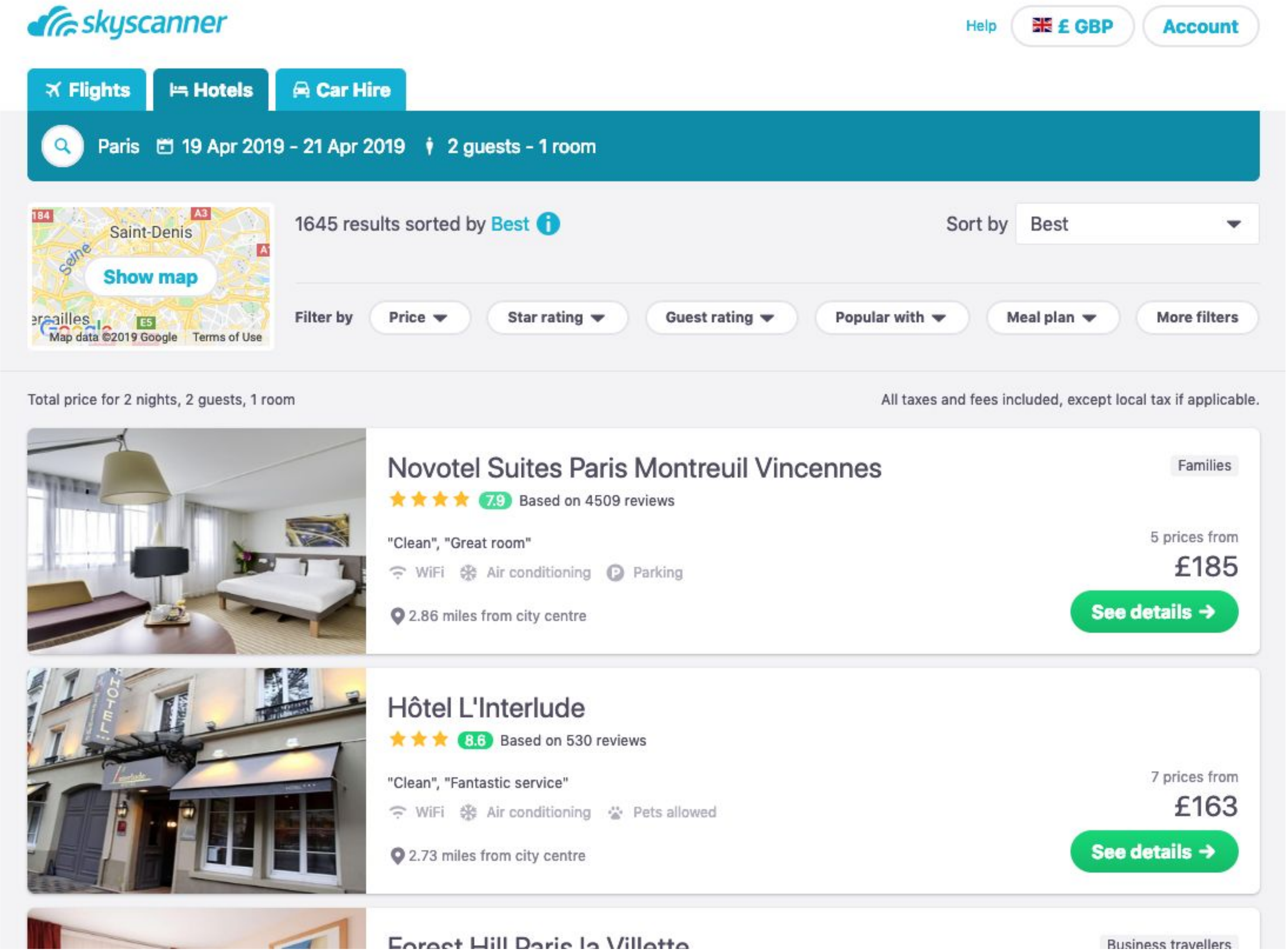
3. Evaluate



- How do we put this algorithm in front of users?
- How do we compare against the previous solution?
- What do we need to keep track of?
- Did we improve significantly?

4. Ranking algorithm

Learning to rank hotels

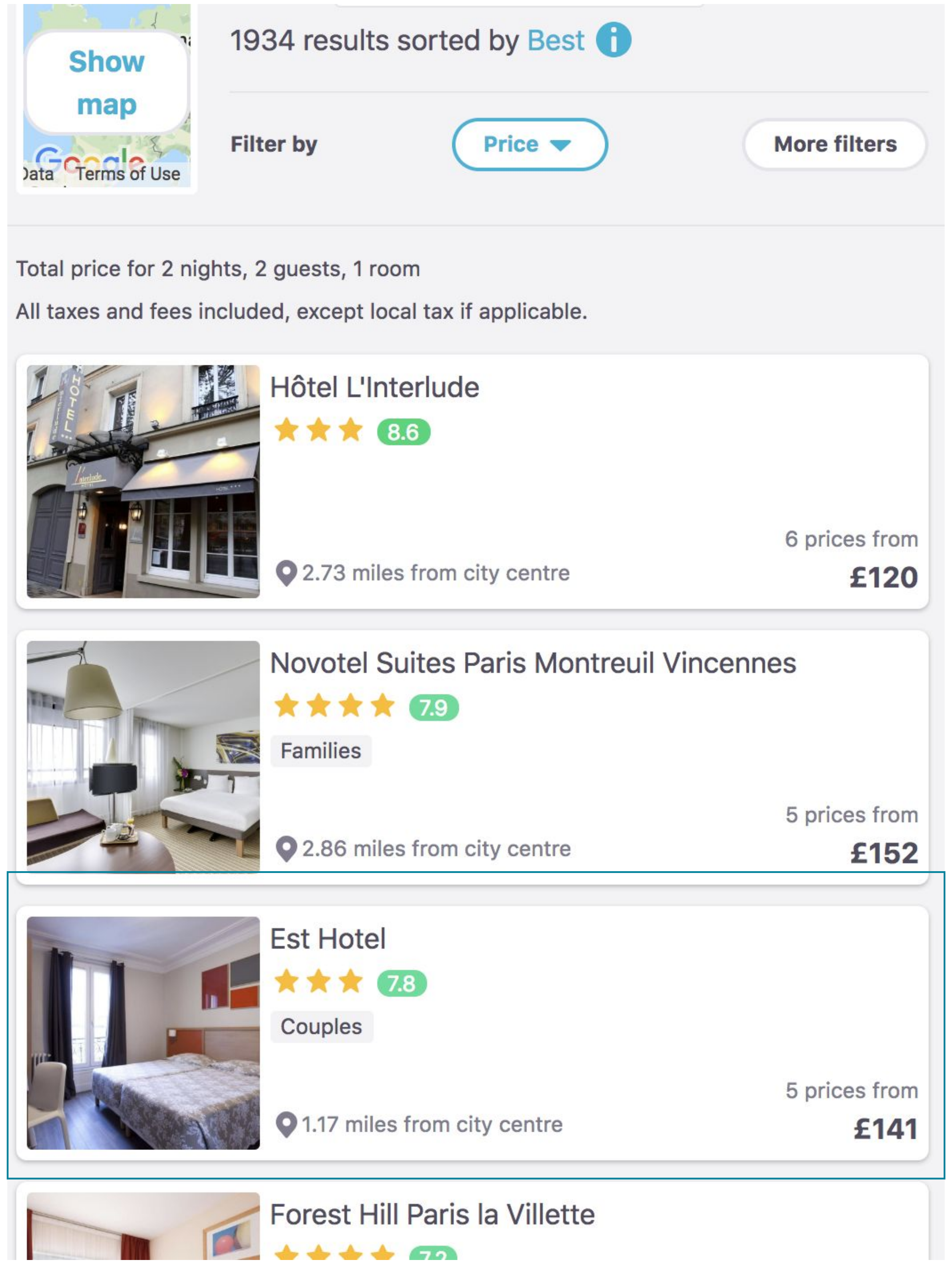


A few properties that domain experts use to order hotels

- Room price per night
- Hotel location
- User Rating
- Hotel Stars
- Hotel Amenities
- City
- Pictures

4. Ranking algorithm

4.1 Problem representation



What the user saw and did not choose

What the user saw and did not choose

What the user saw and chose

What the user did not see

4. Ranking algorithm

4.2 Exploration space

Simple vs Complex models

- Amount of data
- Need for interpretability
- Domain knowledge (features)

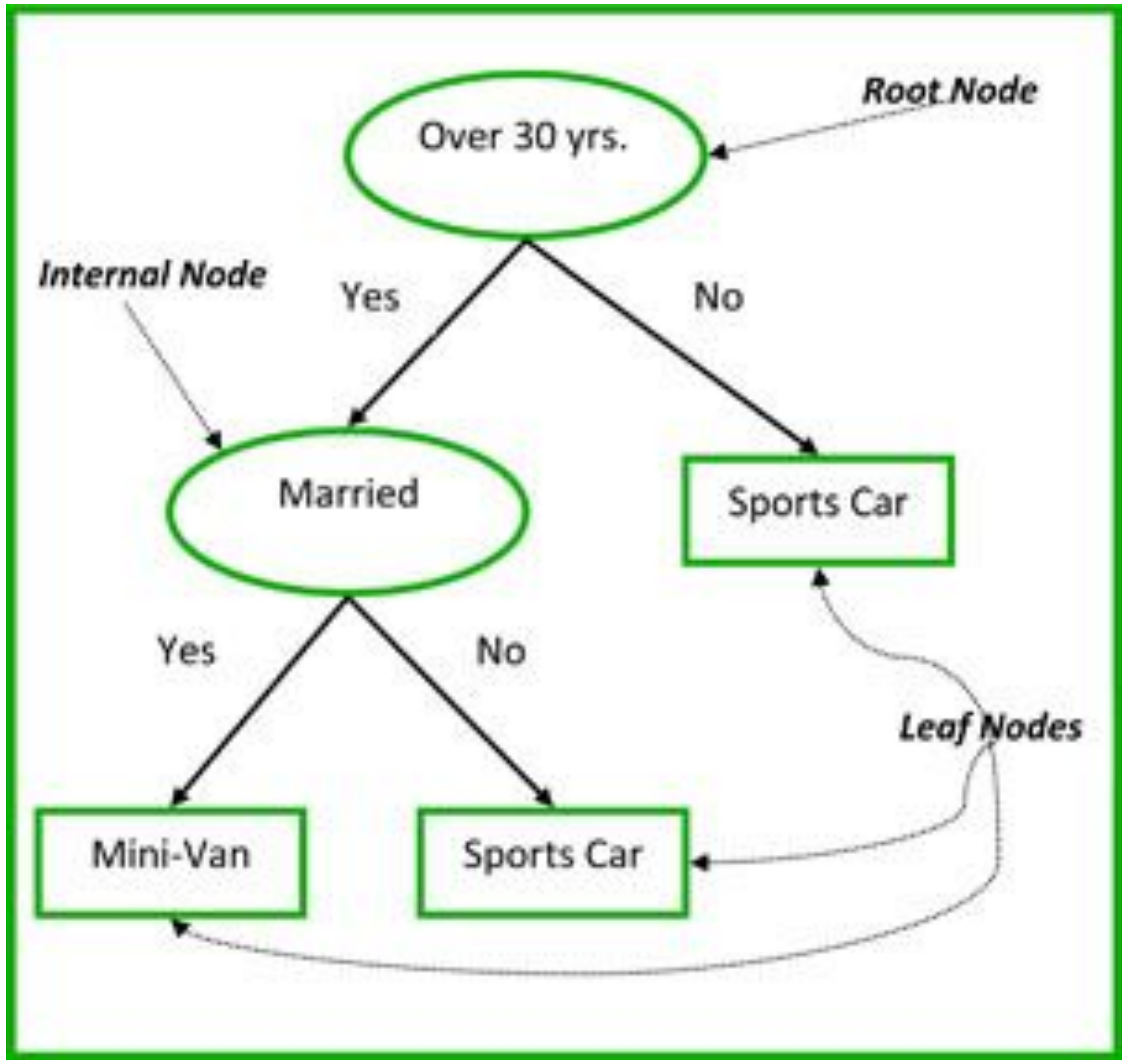
Gradient boosted trees

4. Ranking algorithm

4.2 Exploration space

DECISION TREE

Decision trees



4. Ranking algorithm

4.3 Evaluation

- ✗ Is there a ground truth?
- ✗ Can we reproduce the experiment?
- ! Can we test with real users?
- ✓ Can we test offline?

With assumption for resorting stability

Comparison of multiple models

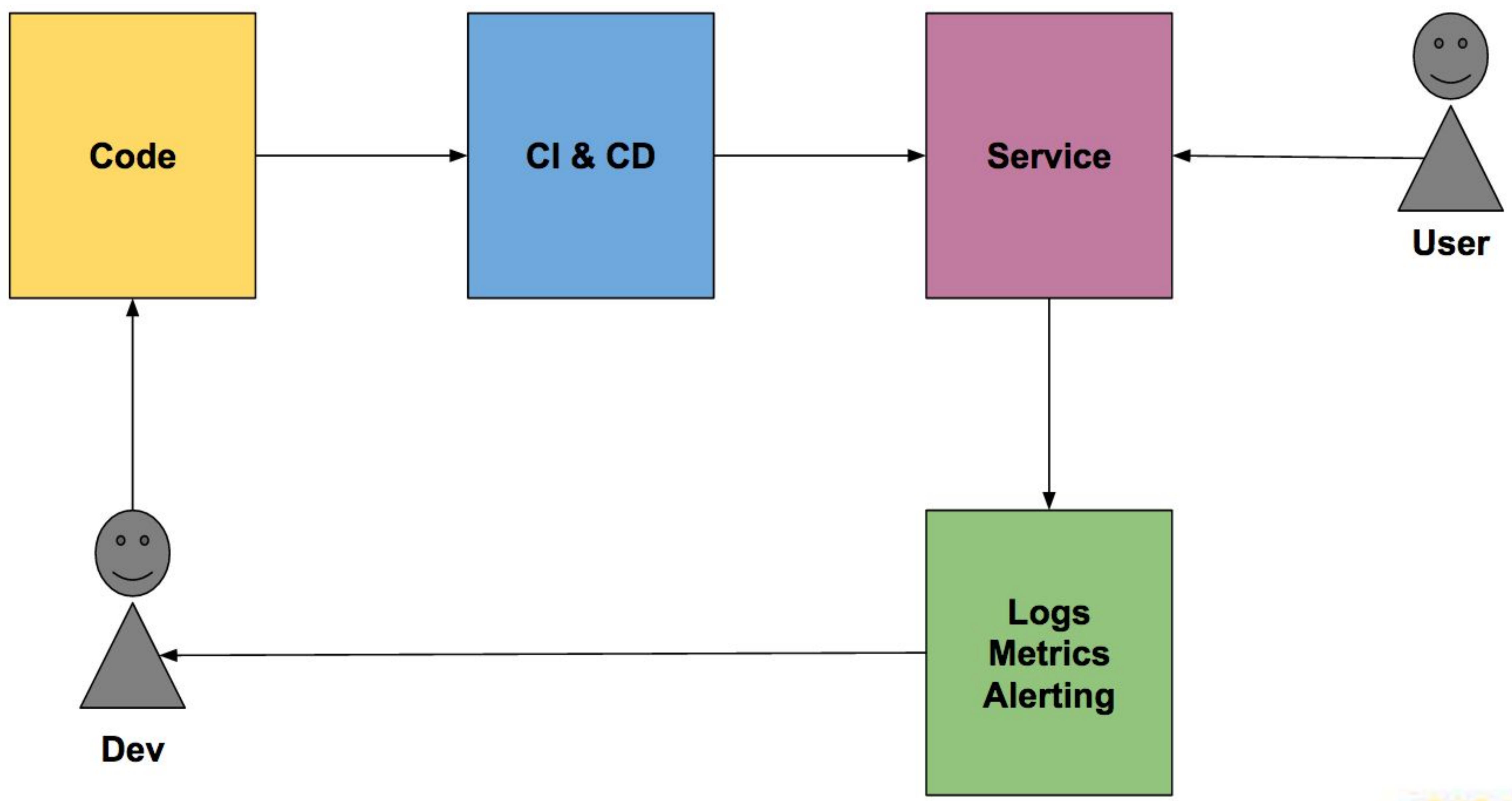
Great!

We have a model, now what?

SUBTITLE

Getting to production

- Continuous integration (CI)
- Continuous Deployment (CD)
- Testing
- Monitoring



SUBTITLE

After being in prod. It’s not over

- Effect of latency
- Bias (e.g. self-fulfilling prophesies)
- Updating models

Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.

Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison
{ebner,vchaudhary,mwyoung,jfcrespo,dennison}@google.com
Google, Inc.

Abstract

Machine learning offers a fantastically powerful toolkit for building useful complex prediction systems quickly. This paper argues it is dangerous to think of these quick wins as coming for free. Using the software engineering framework of *technical debt*, we find it is common to incur massive ongoing maintenance costs in real-world ML systems. We explore several ML-specific risk factors to account for in system design. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and a variety of system-level anti-patterns.

Source: https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

Results

Start simple and iterate

Lean 3-step approach

1. A/A`: **Ensure no harm**

Start simple and iterate

Lean 3-step approach

1. A/A`: **Ensure no harm**
2. A/B with simple model: **Success**

Start simple and iterate

Lean 3-step approach

1. A/A`: **Ensure no harm**
2. A/B with simple model: **Success**
3. A/B with more complex models: **More success**

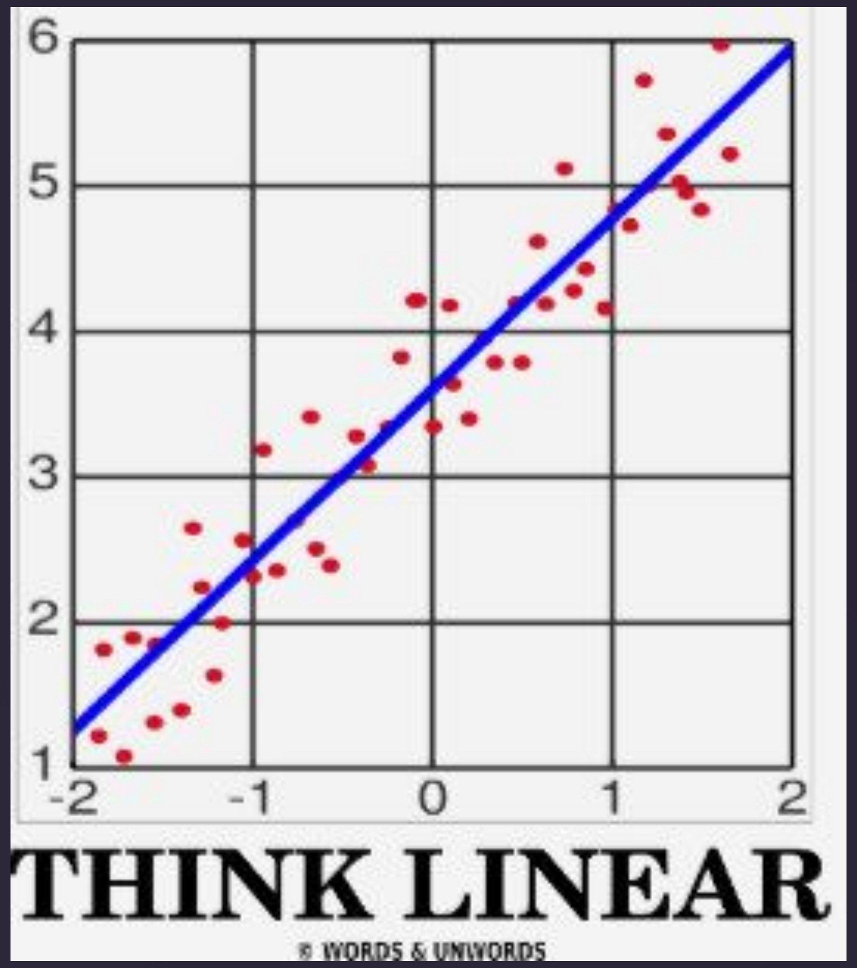
One last thing

A few properties that domain experts use to order hotels

- Room price per night
- Hotel location
- User Rating
- Hotel Stars
- Hotel Amenities
- City
- Pictures

These properties were enough to offer users better hotels than a domain expert

- Price
- ~~Location~~
- Rating
- Stars
- ~~Amenities~~
- ~~City~~
- ~~Picture~~



To summarize

- AI holds a lot of promise for businesses
- Expertise, tools and materials are available
- Even simple solutions can beat domain experts. Start simple.

Thank you.

Konstantin Halachev • SKYSCANNER