OTA INSIGHT

2021

VTK Hackathon

Intro to OTA Insight

OTA Insight - Who are we?

OTA INSIGHT

Founded 2012 in London

200+ stellar employees

London, Ghent, Dallas, Denver, Singapore

Scale up in the hospitality industry

Help the hospitality industry to visualise their data

Our company history

Where it all started

OTA INSIGHT

SEPT 2012

A billion-dollar idea
(So they thought)

The London Olympics

MARCH 2013

Pivot

First customer visits
First of many pivots
Pooling money

SEPT 2013

Incubator

Smart money
Learning the industry
16/24 – 7/7

JAN 2014

MVP

First employees
MVP
Early adopter

MARCH 2014

Product - Market fit

Strategic ground work

Our company history

Where it all started

OTA INSIGHT

2016

Expansion

Product team
Sales team grows
OTA Insight goes to the US

2018

100 employees!

OTA Insight keeps growing
Launch RI
30,000 partners

2019

Continuous growth

Scaling up
#200 employees

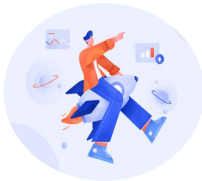
2020

Getting ready for the upturn

Market insight launch
Tripadvisor Partnership
Getting ready for upturn

2021

Road to recovery



Getting back to rocket mode
250 employees!

Where will you find our teams?

With more than 30 nationalities on the team, we truly are a global company

OTA INSIGHT

LONDON

OTA Insight's HQ
All teams

GHENT

OTA Insight's tech hub
Finance
HR

DENVER & DALLAS

Commercial
Marketing
CS - OPS - SD
HR
Finance

SINGAPORE

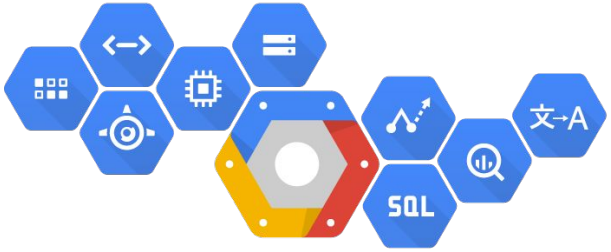
Commercial
Marketing
CS - OPS - SD

Facts & Figures

Signed hotels	55,000
Signed locations	185
Employees	200+
Data collection	1.5B
Products	4

Facts & Figures

- We crunch (actual) big data
  - 100 TB of data processed per day (= binge-watching Netflix for 4 years)
  - 3 x 10^{12} of rates data points in database (= amount of fish in the sea)
  - 2 million reservations processed per day
- We are passionate about technology



elastic

Google Cloud Platform

Stackdriver

Flask

scikit learn

RUNDECK

jupyter

GitLab

dbt

Technology company at heart

python

Grafana

docker

Go

MySQL

ember

kubernetes

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

FoundationDB

dj django

we're building the #bestteamever

OTA INSIGHT

Frontend engineers

You'll be creating complex **single-page MVC applications**, visualizing large amounts of data in an actionable way.

Data engineers

You'll build **high-performance data processing pipelines** which can guarantee industry's best data integrity and quality.

Backend engineers

You'll architect, code and maintain our **API backend and microservices infrastructure**.

More info @ www.careers.otainsight.com

Workshop content

Content of tonight:

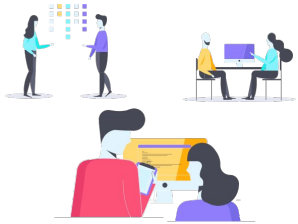
OTA INSIGHT

We in the company do a lot of data gathering and analysis of that gathered data, and tonight you'll do the same (on a very limited level)

- First part: web scraping of a website to gather data
- Second part: data science on existing dataset

Both parts can/should be done in parallel

There are questions for you to answer to gain points



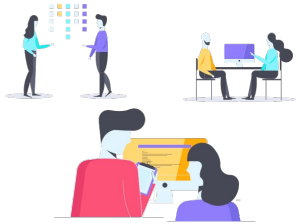
Content of tonight:

OTA INSIGHT

https://github.com/OTA-Insight/vtk-hackathon-2021

For the website crawling part: read through to see the questions you need to answer! You can answer questions partially to us!

First to answer a particular question gets more points than the next one.

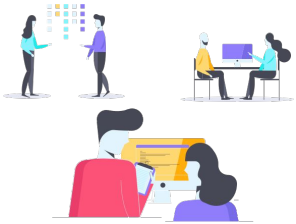


Content of tonight:

OTA INSIGHT

https://github.com/OTA-Insight/vtk-hackathon-2021

- Contains a readme with the workshop instructions
- Contains skeleton code to start with



Content of tonight:

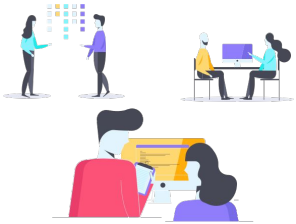
OTA INSIGHT

https://github.com/OTA-Insight/vtk-hackathon-2021

We are available for questions and help

At the end of the night (let's see how everybody progresses for an end time) we'll shortly go over the solutions

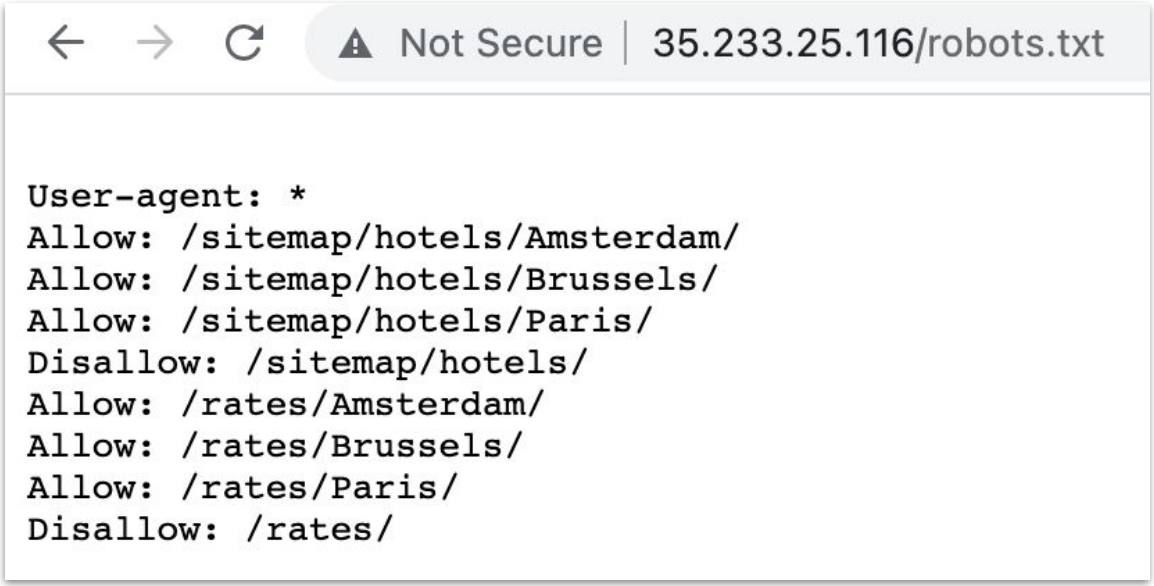
Feedback welcome! mathieu@otainsight.com / lana@otainsight.com



Solutions

Web crawling: countermeasures

- /robots.txt



- https://moz.com/learn/seo/robotstxt

Web crawling: countermeasures

- User Agent

```
19
20  class UserAgentDownloaderMiddleware:
21      user_agent_choices = [
22          "Edg/93", "Edg/94", "Edg/95", "Edg/96",
23          "Chrome/93", "Chrome/94", "Chrome/95", "Chrome/96",
24          "Firefox/93", "Firefox/94", "Firefox/95", "Firefox/96",
25          "Safari/603", "Safari/604", "Safari/605", "Safari/606",
26          "Trident/7",
27      ]
28
29      def process_request(self, request, spider):
30          # user_agent = request.headers.get('User-Agent')
31          # Look at what user_agent is by default ...
32
33          # Replace by a random choice from a good list
34          request.headers['User-Agent'] = random.choice(self.user_agent_choices)
35
36          return None
37
```

- Default = ‘Scrapy/****’

Web crawling: countermeasures

- Rate limiting

```
23
24  // rateLimit will apply a ratelimit for certain hotels + rates
25  // rate limit consits of the destination - ip address - user agent
26  func rateLimit(DestinationID string) adapter {
27      return func(h http.Handler) http.Handler {
28          return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
29
30              // Call the getVisitor function to retrieve the rate limiter for the current key.
31              limiter := getVisitor(DestinationID + r.RemoteAddr + toUserAgentType(r.UserAgent()))
32              if !limiter.Allow() {
33                  WriteJsonResponse(w, http.StatusTooManyRequests, nil)
34                  return
35              }
36
37              h.ServeHTTP(w, r)
38          })
39      }
40  }
41
```

- https://newbedev.com/how-to-handle-a-429-too-many-requests-response-in-scrapy

Web crawling: countermeasures

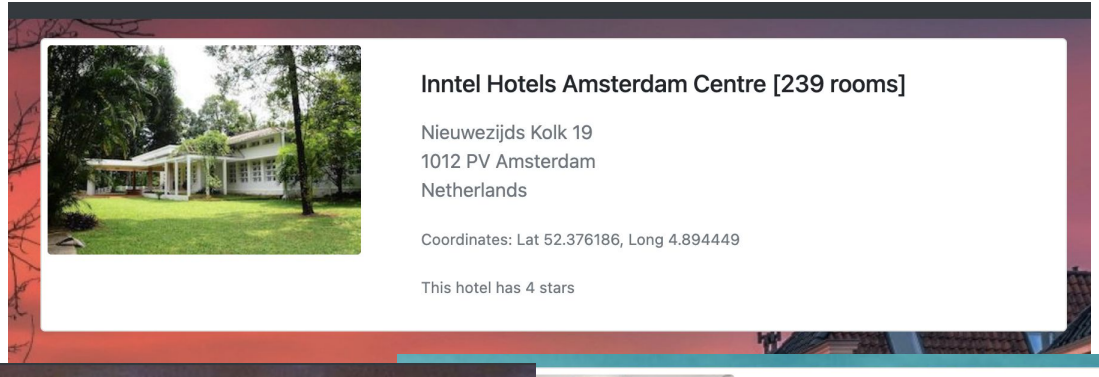
- Rate limiting

```
38
39  class RateLimitRetryMiddleware(RetryMiddleware):
40      def process_response(self, request, response, spider):
41          if request.meta.get('dont_retry', False):
42              return response
43          elif response.status == 429:
44              self.crawler.engine.pause()
45              time.sleep(2)
46              self.crawler.engine.unpause()
47              reason = response_status_message(response.status)
48              return self._retry(request, reason, spider) or response
49          elif response.status in self.retry_http_codes:
50              reason = response_status_message(response.status)
51              return self._retry(request, reason, spider) or response
52          return response
53
54
```

- https://newbedev.com/how-to-handle-a-429-too-many-requests-response-in-scrapy

Web crawling: countermeasures

- Content differences: room count & star rating

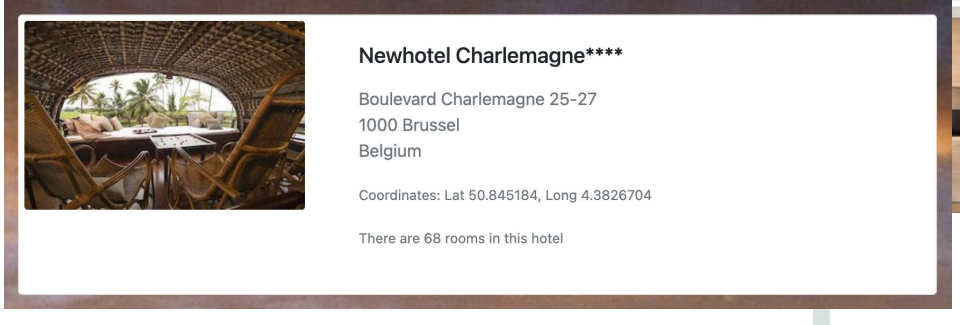


Inntel Hotels Amsterdam Centre [239 rooms]

Nieuwezijds Kolk 19
1012 PV Amsterdam
Netherlands

Coordinates: Lat 52.376186, Long 4.894449

This hotel has 4 stars

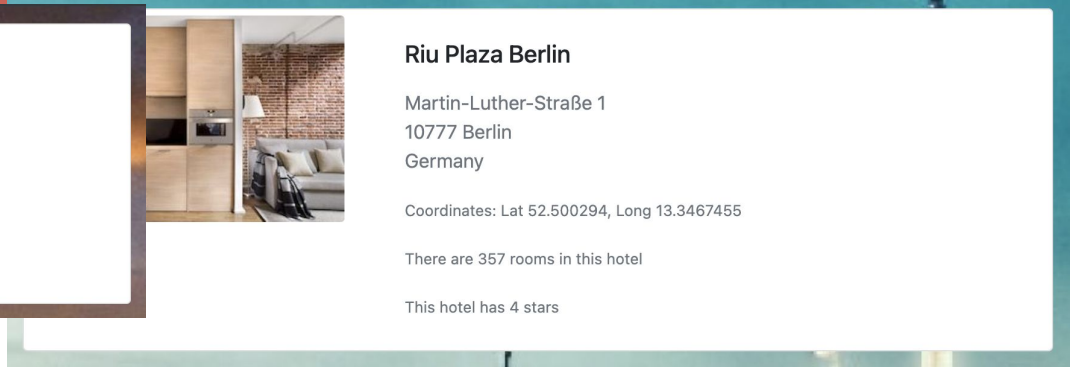


Newhotel Charlemagne****

Boulevard Charlemagne 25-27
1000 Brussel
Belgium

Coordinates: Lat 50.845184, Long 4.3826704

There are 68 rooms in this hotel



Riu Plaza Berlin

Martin-Luther-Straße 1
10777 Berlin
Germany

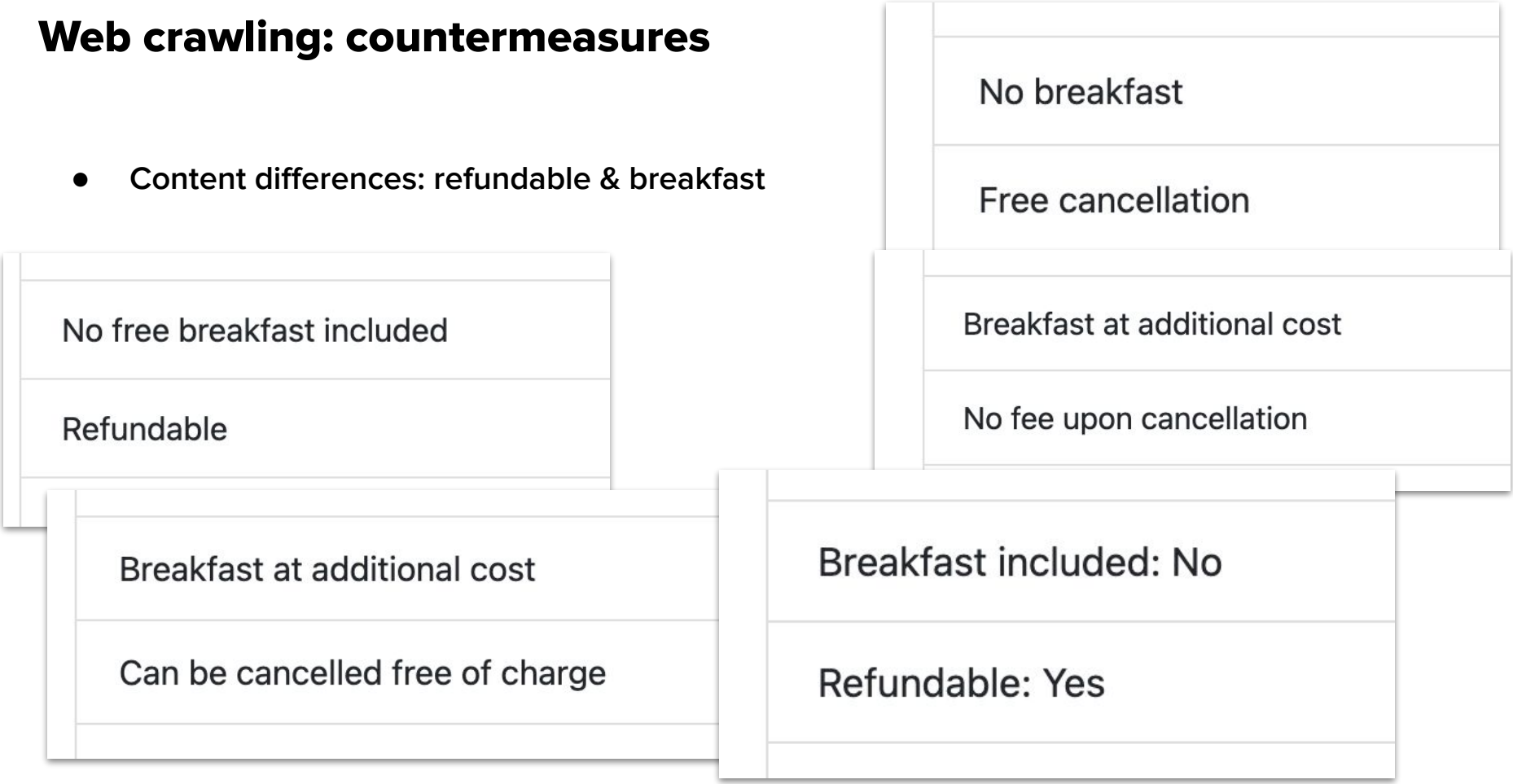
Coordinates: Lat 52.500294, Long 13.3467455

There are 357 rooms in this hotel

This hotel has 4 stars

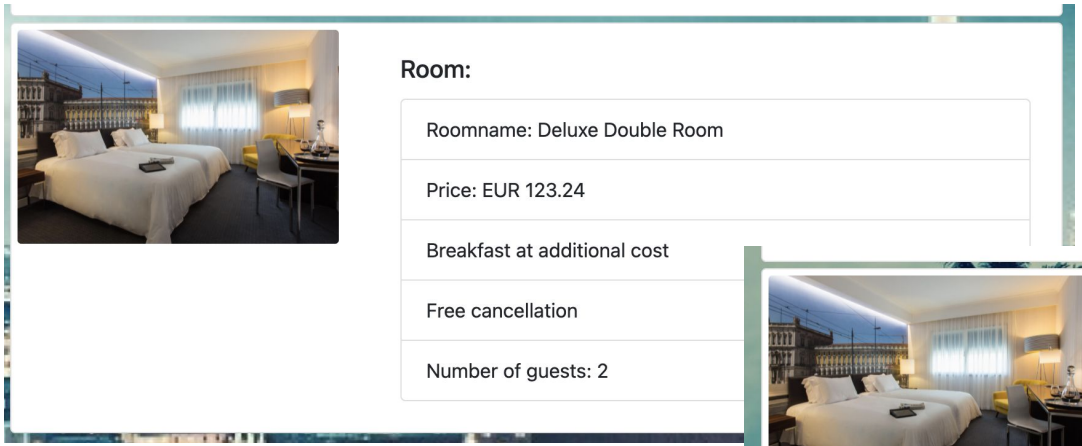
Web crawling: countermeasures

- Content differences: refundable & breakfast



Web crawling: countermeasures

- Broken html with similar look but different



Room:

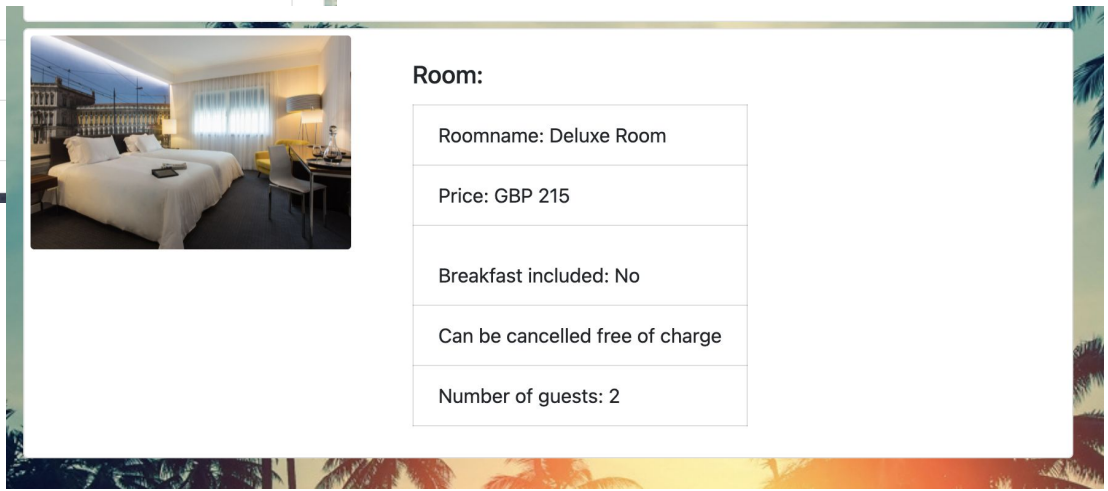
Roomname: Deluxe Double Room

Price: EUR 123.24

Breakfast at additional cost

Free cancellation

Number of guests: 2



Room:

Roomname: Deluxe Room

Price: GBP 215

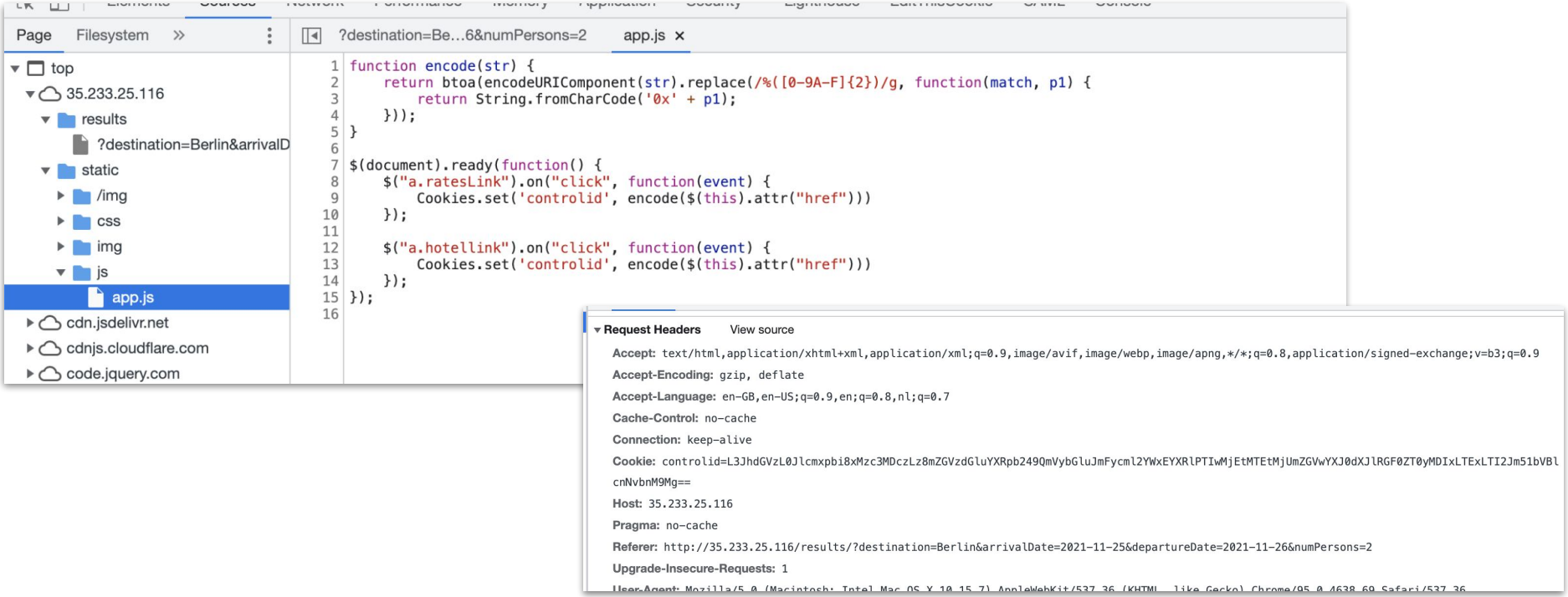
Breakfast included: No

Can be cancelled free of charge

Number of guests: 2

Web crawling: countermeasures

- Header required: base64 encoded path



Web crawling: countermeasures

- Header required: base64 encoded path

```
54
55  class ControlIDCookiesMiddleware:
56      def process_request(self, request, spider):
57          if '/Berlin' in request.url:
58              path = request.url.removeprefix(server_location)
59              encoded_path = base64.b64encode(path.encode()).decode('utf-8')
60              request.cookies['controlid'] = encoded_path
61          return None
62
```

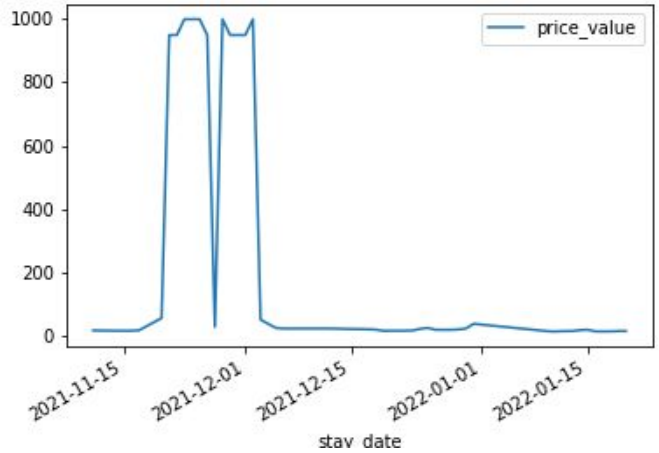
Data science: exceptional deals

- Normalize prices to price index
- Sort by price index to find most exceptional prices
- Normalizer should be robust

Data science: exceptional deals

Normalizing by mean price value

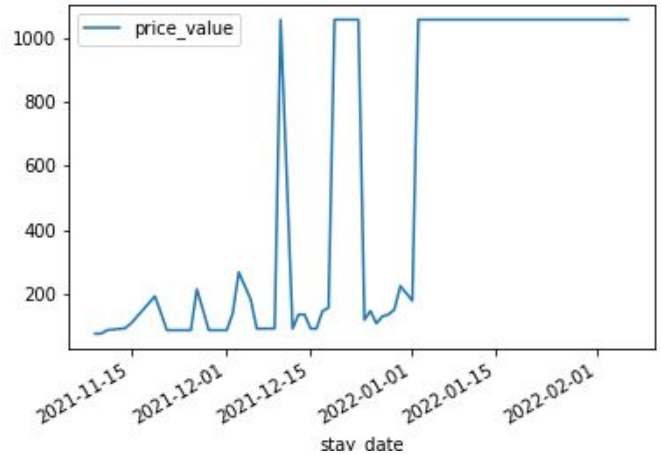
- Mean is sensitive to outliers
- Lowest price index rate:
  - price value: 14 euro
  - normalizer: 222 euro
  - index: 0.06



Data science: exceptional deals

Normalizing by median price value

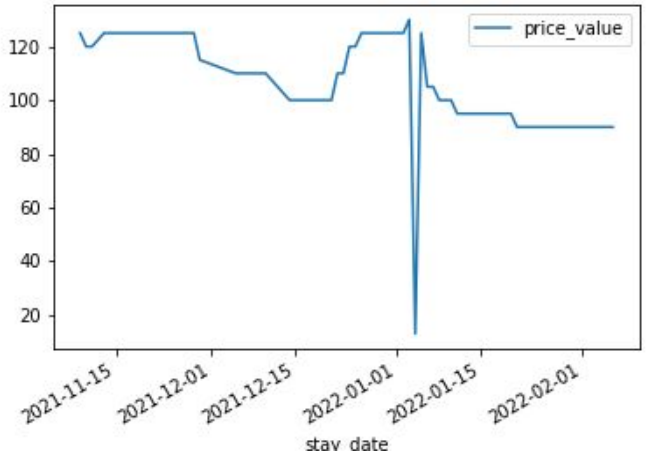
- Issue when more than 50% is unreal price
- Lowest price index rate:
  - price value: 76 euro
  - normalizer: 1056 euro
  - index: 0.07



Data science: exceptional deals

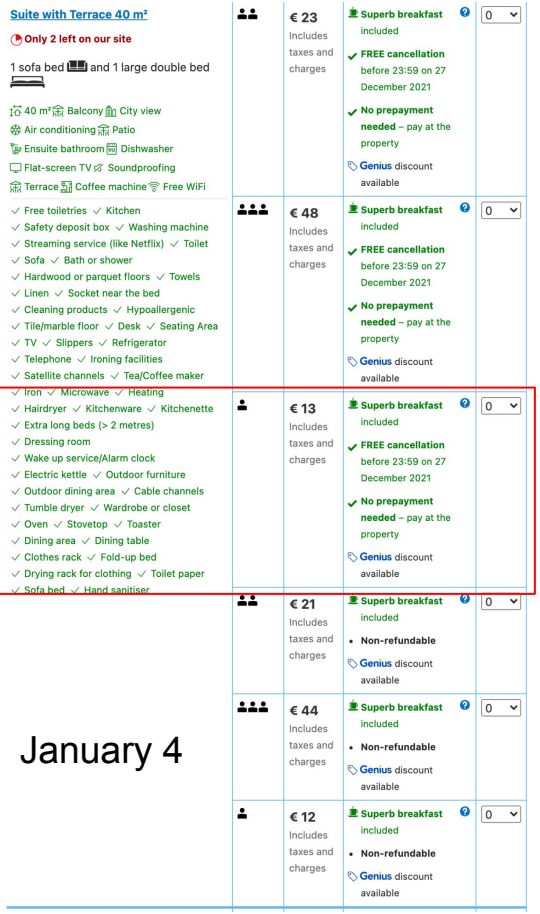
Normalizing by 10th percentile price value

- Robust method of describing usual price value
- Lowest price index rate:
  - price value: 13 euro
  - normalizer: 90 euro
  - index: 0.14

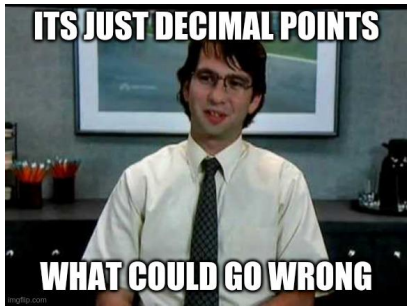


Data science: exceptional deals

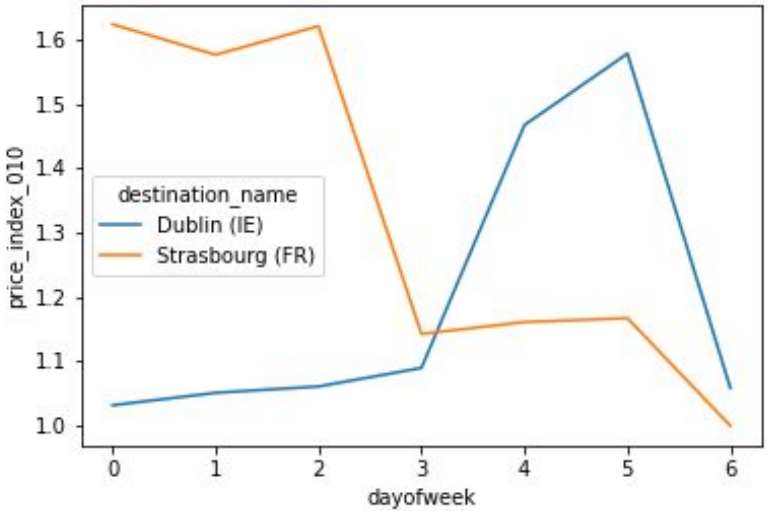
January 5



January 4



Data science: destination patterns



Winners!

And the winners are...

OTA INSIGHT

1st place

2nd place

3rd place

Croqueaton

And the winners are...

OTA INSIGHT

1st place

2nd place

The Brogrammers

3rd place

Croqueaton

And the winners are...

OTA INSIGHT

1st place

The Dream Team

2nd place

The Brogrammers

3rd place

Croqueaton

OTAINSIGHT

Q&A

ota-insight @otainsight / otainsight www.otainsight.com