APAN PS5400: Managing Data

Week 11: Case Studies

Lecturer: François Scharffe

Columbia University
School of Professional Studies

Last Week

- Architectural look at Cassandra and MongoDB
- Interacting with Cassandra and MongoDB using Python

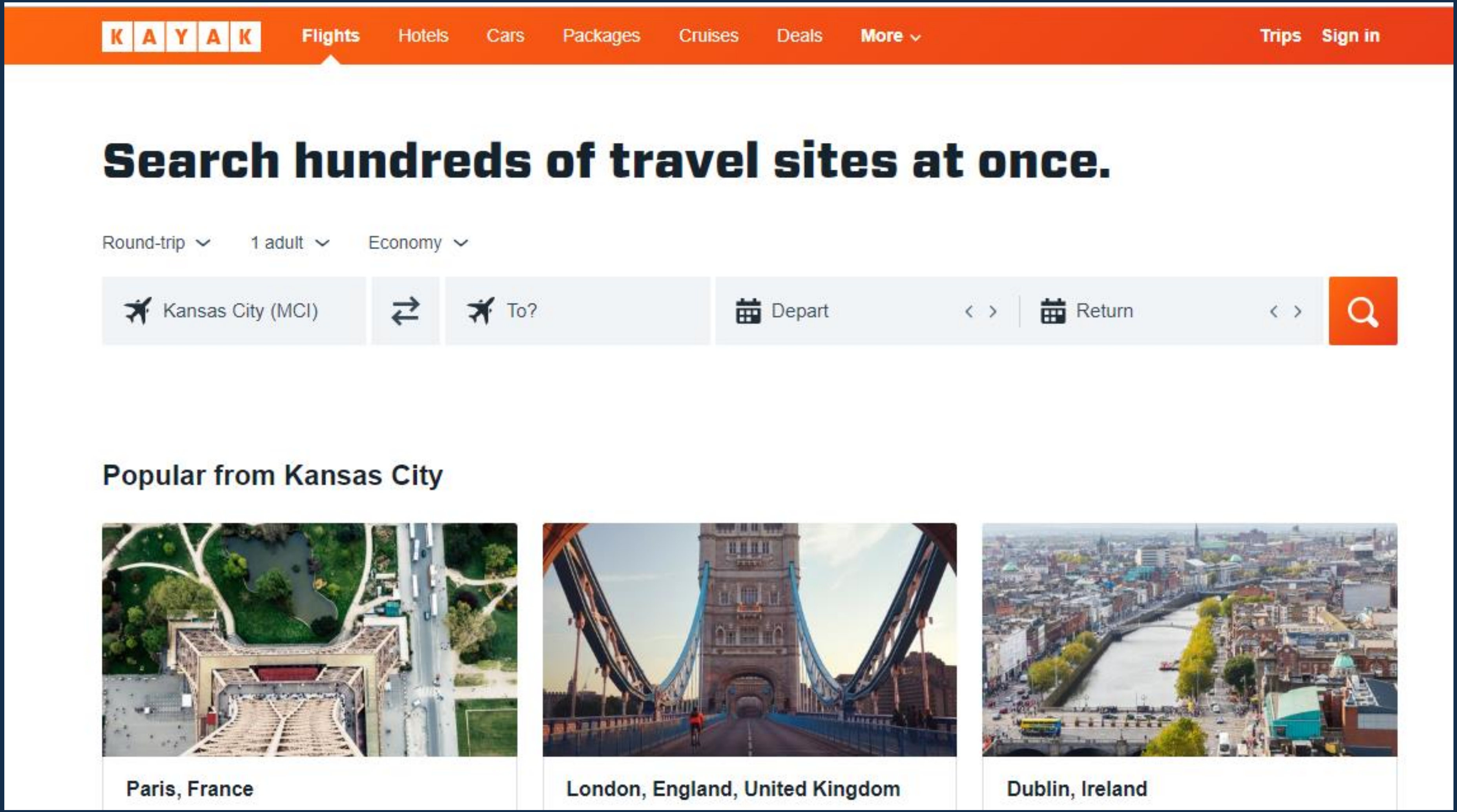
This week

- Analyzing data management needs of a very large consumer facing application—Kayak
- Imagine YOU are tasked with creating the data management needed to support the features of this application
  - Don’t expect your lecturer to spoon feed the answers
  - Not that there any one set of correct answers

KAYAK

What is Kayak?

- Kayak is a search engine for
  - Flights
  - Hotels
  - Cars
  - Packages
  - Cruises
  - Deals
  - More (let us ignore the ‘More’)
- It also seems to be a recommendation engine
  - Recommends popular destinations from your current location



Search hundreds of travel sites at once.

Round-trip 1 adult Economy

Kansas City (MCI) ⇄ To? Depart Return

Popular from Kansas City

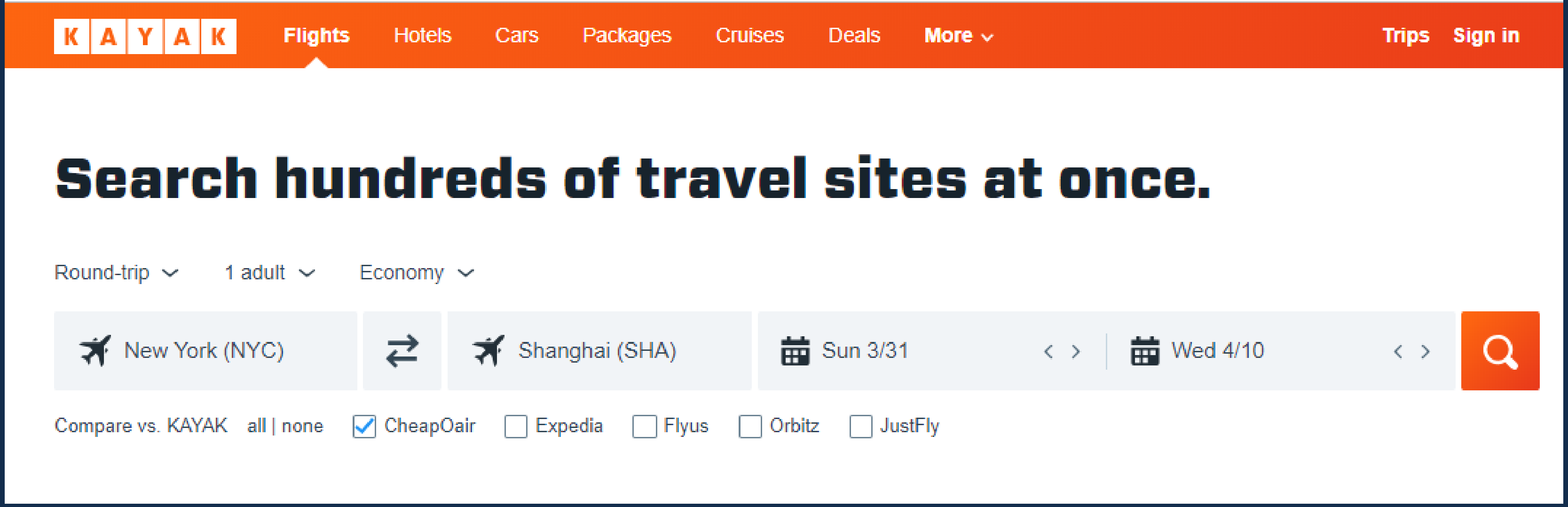
Paris, France

London, England, United Kingdom

Dublin, Ireland

Note that it shows most popular destinations from the city you are located in.

Destination Recommendations: What kind of data is needed to support this feature and how does it need to be stored and managed?



Search hundreds of travel sites at once.

Round-trip 1 adult Economy

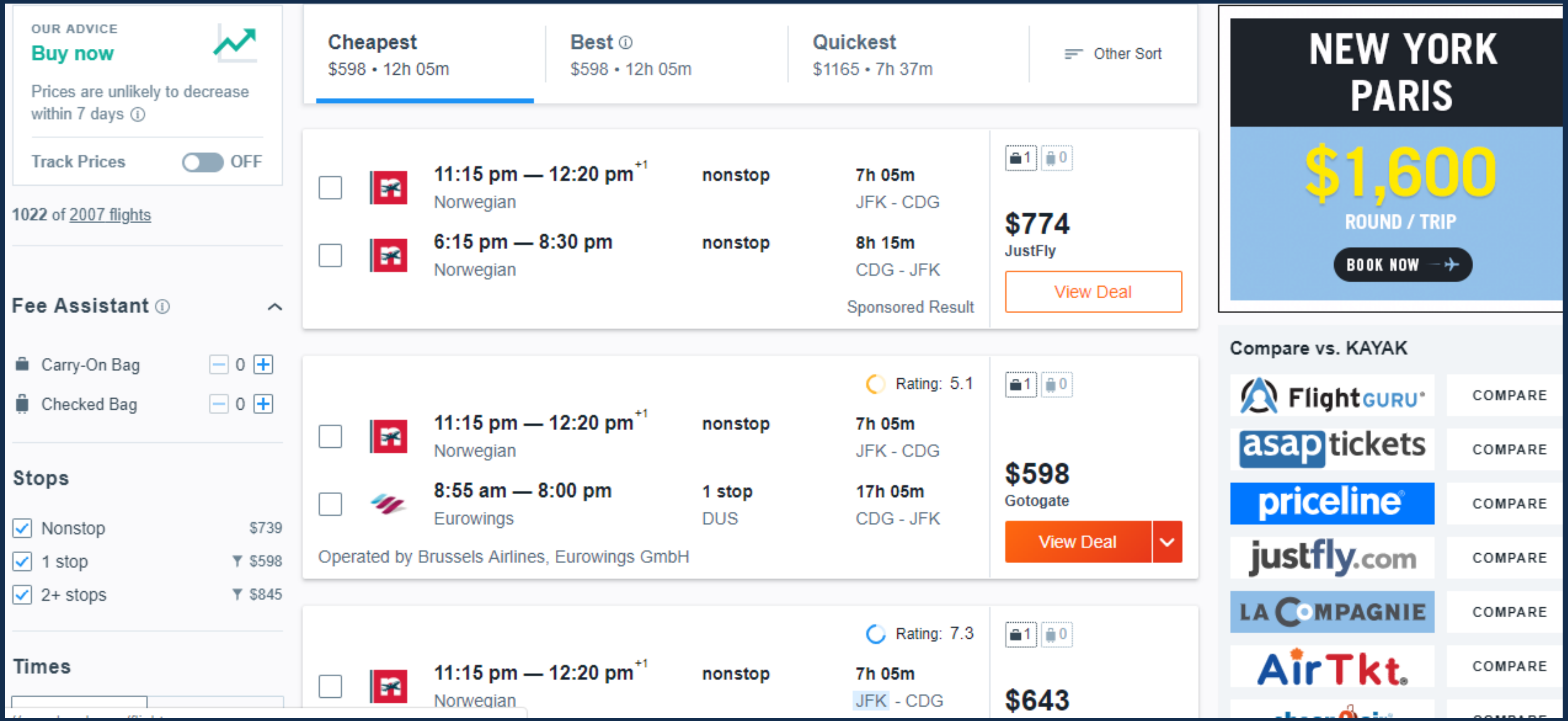
New York (NYC) ⇄ Shanghai (SHA) Sun 3/31 Wed 4/10

Compare vs. KAYAK all | none ☑ CheapOair □ Expedia □ Flyus □ Orbitz □ JustFly

- For flights, it returns not only results from its data but allows you to compare the results of doing the search on other sites, like Expedia.

- As a user types the name of a city, it provides the names of all the airports in the city and their three letter codes.
  - Airports and Codes: What kind of data is needed to support this feature and how does it need to be stored and managed?

Buy Now or Later Recommendation



OUR ADVICE
Buy now
Prices are unlikely to decrease within 7 days ⓘ

Track Prices OFF

1022 of 2007 flights

Fee Assistant ⓘ

Carry-On Bag − 0 +
Checked Bag − 0 +

Stops

☑ Nonstop $739
☑ 1 stop $598
☑ 2+ stops $845

Times

Cheapest	Best ⓘ	Quickest	Other Sort
$598 • 12h 05m	$598 • 12h 05m	$1165 • 7h 37m	

□ 11:15 pm — 12:20 pm +1 · Norwegian · nonstop · 7h 05m · JFK - CDG
□ 6:15 pm — 8:30 pm · Norwegian · nonstop · 8h 15m · CDG - JFK
Sponsored Result
1 · 0
$774
JustFly
View Deal

Rating: 5.1
□ 11:15 pm — 12:20 pm +1 · Norwegian · nonstop · 7h 05m · JFK - CDG
□ 8:55 am — 8:00 pm · Eurowings · 1 stop DUS · 17h 05m · CDG - JFK
Operated by Brussels Airlines, Eurowings GmbH
1 · 0
$598
Gotogate
View Deal

Rating: 7.3
□ 11:15 pm — 12:20 pm +1 · Norwegian · nonstop · 7h 05m · JFK - CDG
1 · 0
$643

NEW YORK
PARIS
$1,600
ROUND / TRIP
BOOK NOW

Compare vs. KAYAK

FlightGURU	COMPARE
asaptickets	COMPARE
priceline	COMPARE
justfly.com	COMPARE
LA COMPAGNIE	COMPARE
AirTkt	COMPARE

Price Comparisons

- Buy Now or Later Recommendation: What kind of data is required for this feature?
- Price Comparisons: How is this data acquired? Managed?
- Search results: Where does this data come from?

Columbia University
School of Professional Studies

Flights: Features and Data

We have identified the following features requiring data management. If you were designing this system, what data management services would you build to support these features???

- Destination recommendations
- Airports and Codes
- Buy Now or Later recommendations
- Search results
  - Flight info (take-off, landing, stop-overs, etc.)
  - Price
  - Sometimes, number of seats left
- Price comparisons

Flights: Destination Recommendations

- Recommendations are relative to the location of the user
  - Location must be inferred from the IP address of the user
- For each location, data about trips originating from these locations
  - From Kayak’s own bookings but also from third-party data
    - Perhaps from social media
  - This data is constantly being updated
  - Would you use a relational DB or a NoSQL DB to support this feature?
    - Keep in mind that destination recommendations don’t have to be based on the most up-to-date data

Airports and Codes

- This is static data
- There is no reason why the data can’t be stored in a relational DB
- But keep in mind when a part of an airport matches a number of airports, they may be presented in a certain order
    - E.g., ‘Lo’ may match ‘Heathrow, London’ , ‘Stansted, London’, ‘London, Ontario’, ‘Londonderry, Ireland’, ‘Los Angeles’, and many more.
- In what order should they be presented?
    - What data is needed to make this determination?
    - If ‘Lo’ is in the destination field, then should it depend on what origin city is chosen?
    - How dynamic is this data? How dynamic is the order?
        - E.g., In winter, matching Florida airports may be ranked higher; in summer European airports may be ranked higher

Take 5 minutes to think about this and be prepared to discuss your solution.

Flights: Buy Now or Later Recommendations

- Presumably, these recommendations are based on a predictive model about when prices will go up, stay the same, or go down. What data do we need for that?
  - Need to distinguish between data needed to build the model (historical data) and data needed to make a prediction (input features data)
- Historical data needed depends on the features selected for the ML model
  - Holiday season? Tourist season? Any major events in the region? How much percentage of seats are already filled up? Price of jet fuel? Etc.
- Input features data—Percentage of seats already filled up
  - Combination of data about total seat capacity on a given route during a time period (relatively static), and
  - Real-time data (streaming data?) about how many seats are filled up on that route
  - Where does this data come from? In what form? Stored as raw data? Processed on demand?
- Input features data—Price of jet fuel
  - Combination of spot prices and future options on jet fuel

Take 5 minutes to think about how to handle ‘Input Features Data—Percentage of Seats Already Filled Up’ and be prepared to discuss your solution.

Flights: Search Results

- Customers search based on origin, destination, departure date, and return date (if return flight).
- The search can be done on data stored in a relational database.
  - This is very straight-forward
- What about how search results are ordered?

"There are several factors that go into how we list results after you start a search. It can be a combination of information like price and traveler ratings. For flights, we even pick up on elements that can affect your comfort, like duration and stops." from https://www.kayak.com/company

Can the factors listed above be stored and retrieved from a relational database?

How Kayak gets data

“We search hundreds of travel, airline, hotel and rental car sites at once to bring you options from across the web…. At the very least, provider sites should give us an updated inventory of prices every 24 hours. ” from https://www.kayak.com/company

- Presumably, this data comes through APIs or some other types of feeds.
  - The data may be in the form of JSON objects
  - It may come in unpredictable intervals
  - It may come very frequently if number of seats info is to be displayed in search result
  - How should the data be ingested and used?

Flights: Price comparisons

- Price comparisons from selected alternate sites like Priceline, Expedia, etc.
- Does data from these sites needs to be stored or just displayed?
    - If stored does it need to be stored before results are displayed?
    - Would storing this data be useful? If so, how exactly might it be used?

Hotel Recommendations

“With hotels, the “Recommended” algorithm is based on a few key factors. We mainly rely on the price, the hotel’s guest rating and its popularity (measured in clicks).” from https://www.kayak.com/company

- Do the clicks on each hotel need to be stored as data (with a timestamp) or do we need to keep a running count of clicks for each hotel (within a time frame)? What do you recommend?
- If we wanted to personalize recommendations—either as per that individual or as per a demographic group that person is a member of--then would we need to also store who clicked on which hotel.
  - In that case would we need to store data about each click as opposed to just keeping a running count?

Hotel static data requirements

"Before we can query for hotel availability we need to load static information into our databases. Ideally, your hotel catalog should be updated daily to ensure that we always have the most up-to-date catalog…We accept multiple data exchange formats. The preferred option is to use an existing provider static data API or scheduled dump of static data. In both cases, we will extract all useful data " from https://www.kayak.com/static-data-requirements

- Does it make sense to store this "raw data" (e.g., JSON objects) or should we just "extract all useful data" and store it in our database?
- What might be some reason to store this "raw data?"
  - Should we store it in a data lake? (note the data could come in "multiple data exchange formats")

Recap

- Examine the features of the application
- What are sort of data is needed for this feature?
- Where would this data come from? In what form?
- What is the volume and velocity of this data?
- How should it be stored?
- How is it processed and served?

Next week

- Introduction to the Hadoop ecosystem
- Inverted Index data model
- Map-Reduce
- HDFS
- Installing Cloudera Hadoop on Google Cloud