Data Product Design

A real world example: Choice analytics

TOC

Overview

A data project is like any other project, with risks that must be mitigated in project planning.

There are some specificities to data products that we need to take into account.

A good project will consider all the risks and plan accordingly.

The importance of the TFM

Developing a project from idea to completion is an integral part of being a Data Scientist. The TFM is your best training for this.

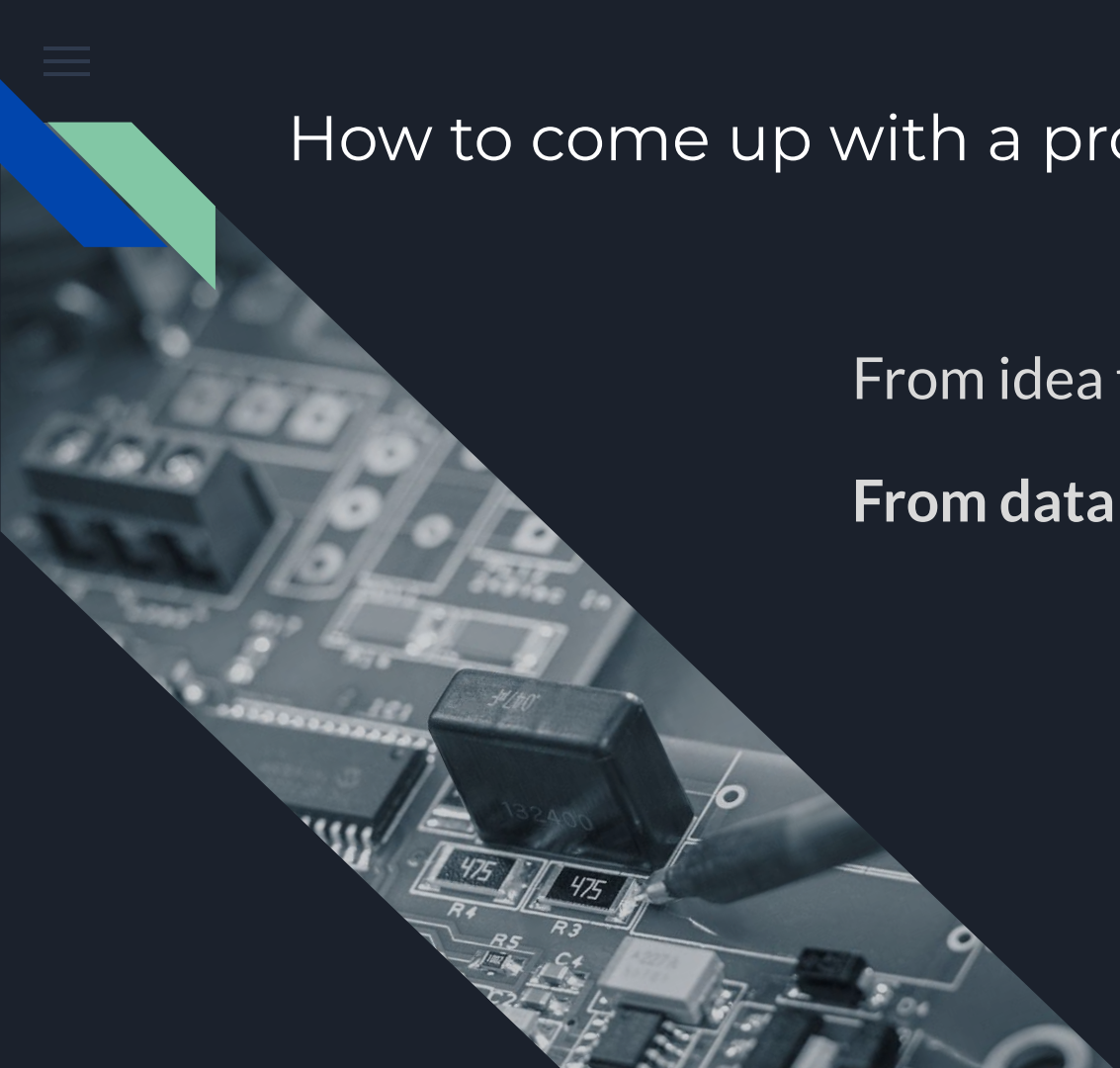
A portfolio is your best calling card.

Thinking is done best with your hands.

Pain is weakness leaving your body

We are your safety net.

How to come up with a project idea



From idea to data

From data to idea

Finding data sources

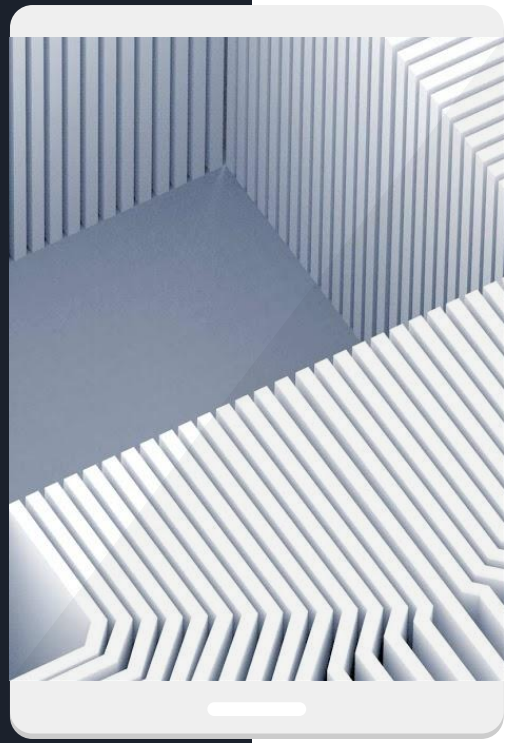
Public datasets

Statistical organizations

APIs

Scraping

Kaggle



Private datasets

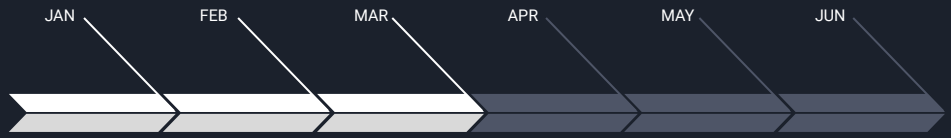
Contacting a company

Generating data

Handling private data

https://www.transtats.bts.gov/Fields.asp
https://github.com/awesomedata/awesome-public-datasets

Planning a project



Build an MVP quick (Minimum Viable Product).

Have a tight iteration loop

Establish checkpoints and intermediate goals. Fail fast.

Set actual dates for checkpoint completion.

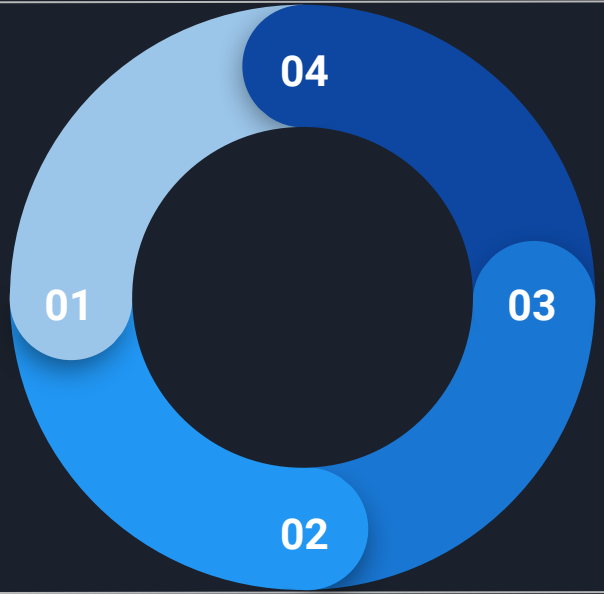
Be ready to pivot if your original idea turns out to not be attainable.

Budget a lot of time to obtain/generate/clean the data. It will still not be enough.

Project iteration loop

Data cleaning

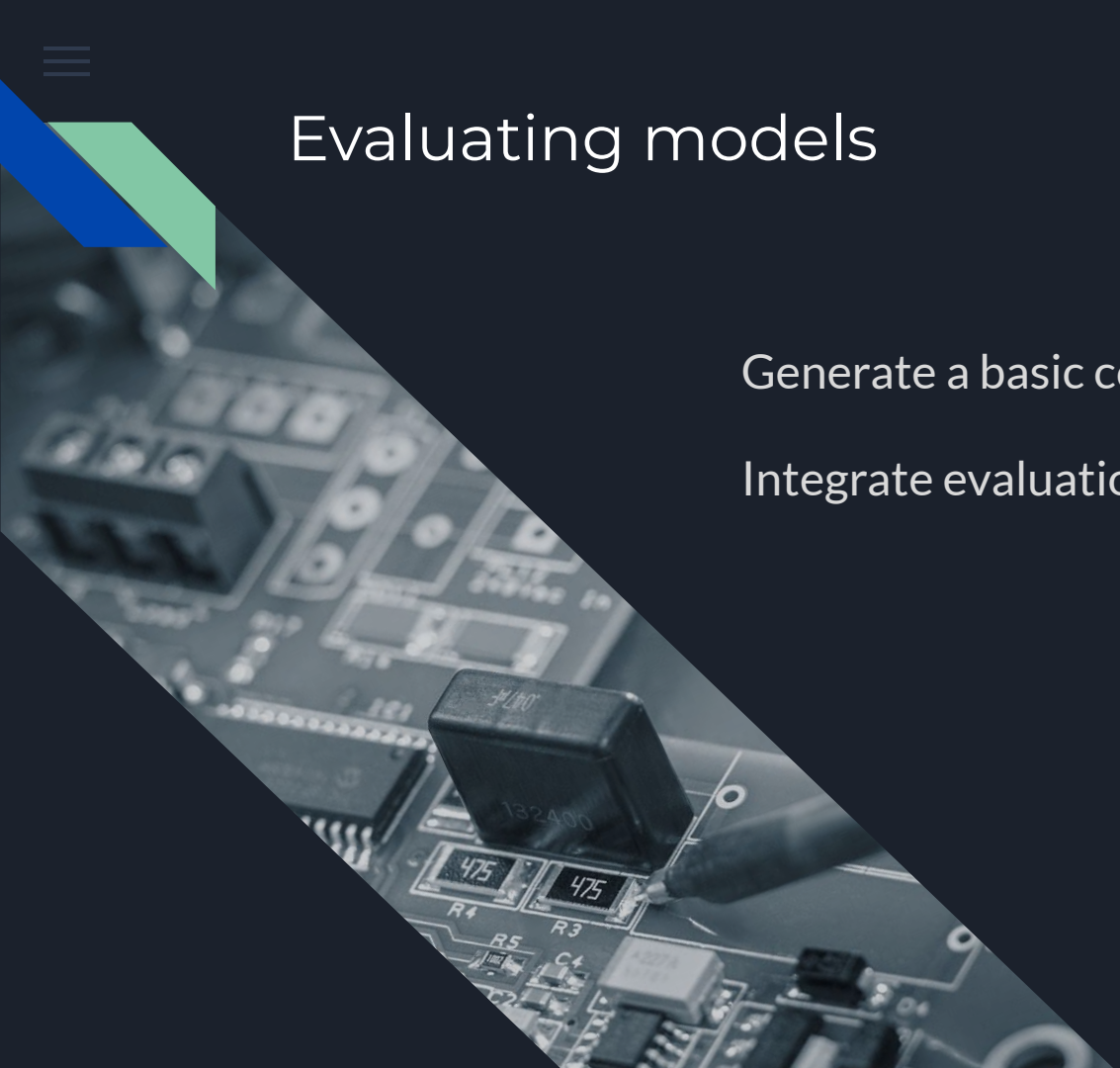
Visualization



Evaluation

Modelling

Evaluating models



Generate a basic common sense baseline

Integrate evaluation with deployment

Kinds of projects

Exploratory, white paper style.

Modelling/predictive

Clustering/market segmentation

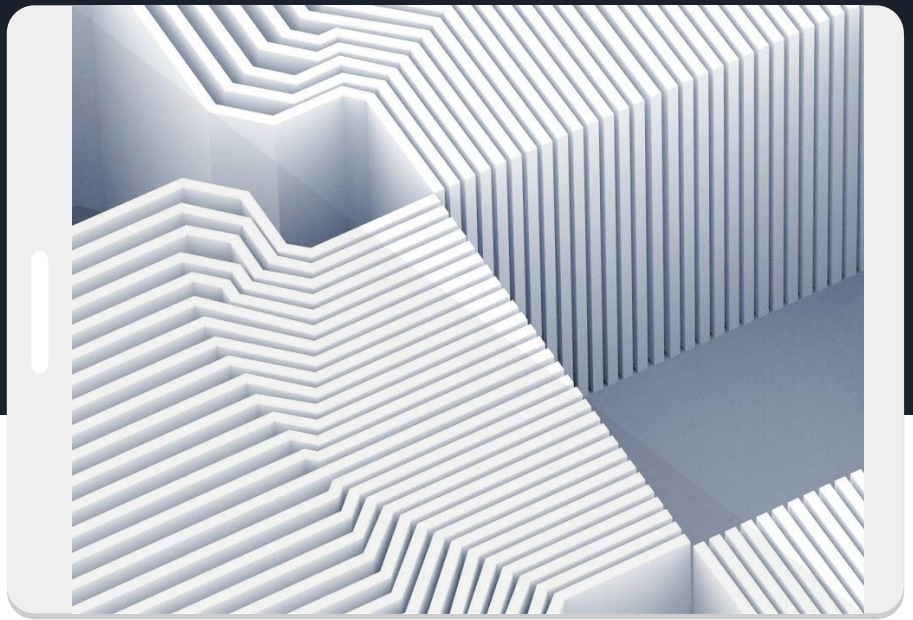
All or nothing. Deep Learning.

Recommenders

The importance of a frontend

A sexy visualization will be a great presentation

Underline the strong points of your project



Workshop

Selling a data product

Background on Choice Analytics example

Company X is an important company within the IT travel market. In particular they provide IT products, consulting services and support to airlines.

Company X has access (or create!) many interesting data feeds, but very often, selling data is not enough...

A promising data feed: **matched shopping**

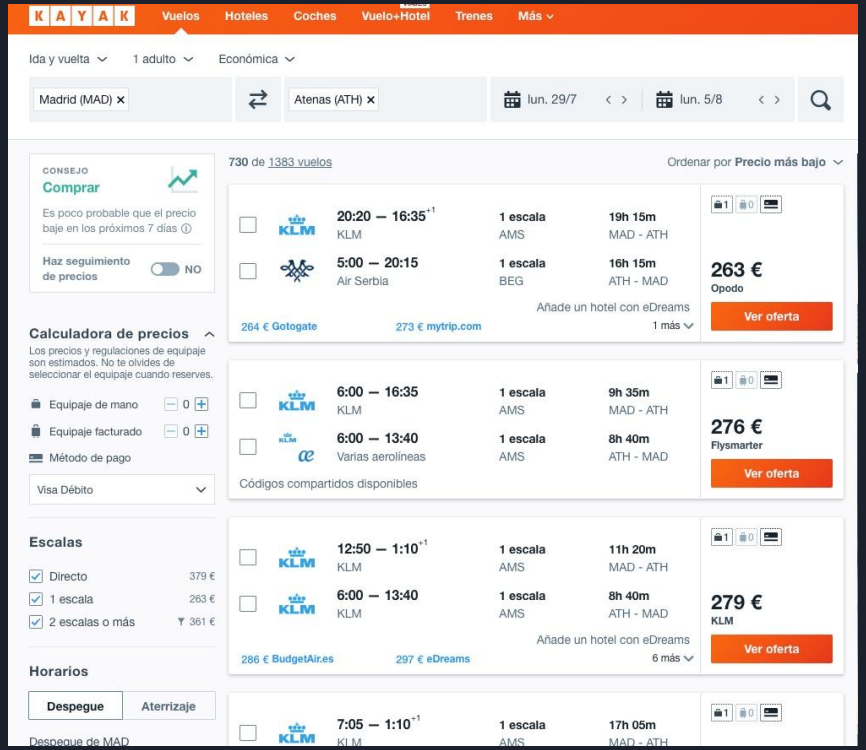
Product Manager S is very enthusiastic about the new data feed that she has to sell.

It is a data feed providing for millions of searches (flying from point A to point B) done online, the recommendations shown to the customer and the bookings done (it comes from matching Searches and bookings, similar to Challenge done in this masters with Panda).

This way, airlines could see the customer preferences when selecting the recommendation...

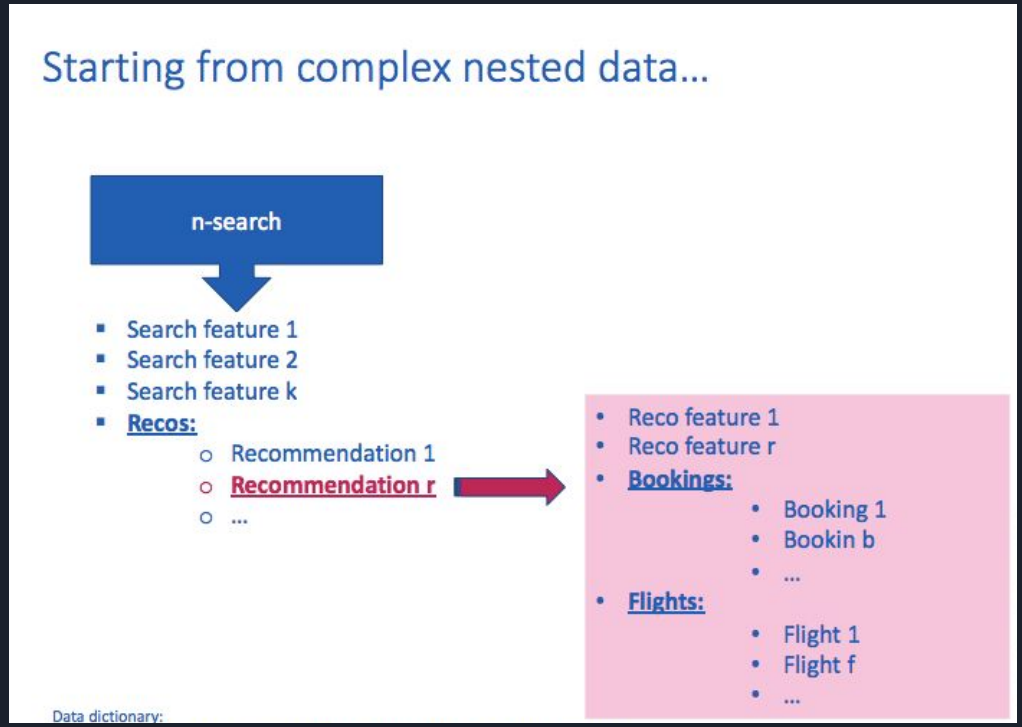
This data feed will be a BEST-SELLER!!!

Visual example



Check data dictionary provided.

Data scheme


Starting from complex nested data…
n-search
Search feature 1
Search feature 2
Search feature k
Recos:
Recommendation 1
Recommendation r
…
Reco feature 1
Reco feature r
Bookings:
Booking 1
Bookin b
…
Flights:
Flight 1
Flight f
…
Data dictionary:

Brainstorming… How could we bring value from this data?

1. Make sure you have a clear (high-level) understanding of the proposed data feed.
2. Find / understand limitations of the data feed.
3. Propose products that could bring VALUE to the airline.
4. Give reasonable arguments to support your product.
5. Select the best of them and prepare an Elevator Pitch for it.

Choice Modeling Project

A Real World Example

Real world scenario

Challenges found:

- Parser was given in python without data dictionary provided.
- Data feed was complex - nested values.
    - Search - recos - bookings/flights
- Data feed was dirty - duplicates, duplicated ids… and not suited for the data product designed.
- Matching process unclear with lack of documentation/info provided.
- New project without clear path - flexibility needed!

Feature Extraction

Feature extraction

Present status and WiP

search_id	Needed to group by search
search_date	Search date
advance_purchase	Days between search and booking
origin	Origin of search, mixing between city and airports
destination	Destination of search, mixing between city and airports
departure_date	Date of first flight departure
return_date	Date of return (empty for One Ways)
persona	Classification of pax Business/Holidays/Weekend/Oneway/U
geography	Domestic, Continental or Intercontinental
distance	Distance of trip
Rank (ID instead)	Rank given by search, ordered by price, different numbers despite same prices.
price	Price in different currencies
Currency - USD	Currency of the search/booking
eft	Elapsed flight time
CNX	Number of connections
aircraft_type	Type of Aircraft
marketing_carrier	Marketing carrier
itinerary_type	One way or Round trip Flag

Crossing with Scheduling (WiP)

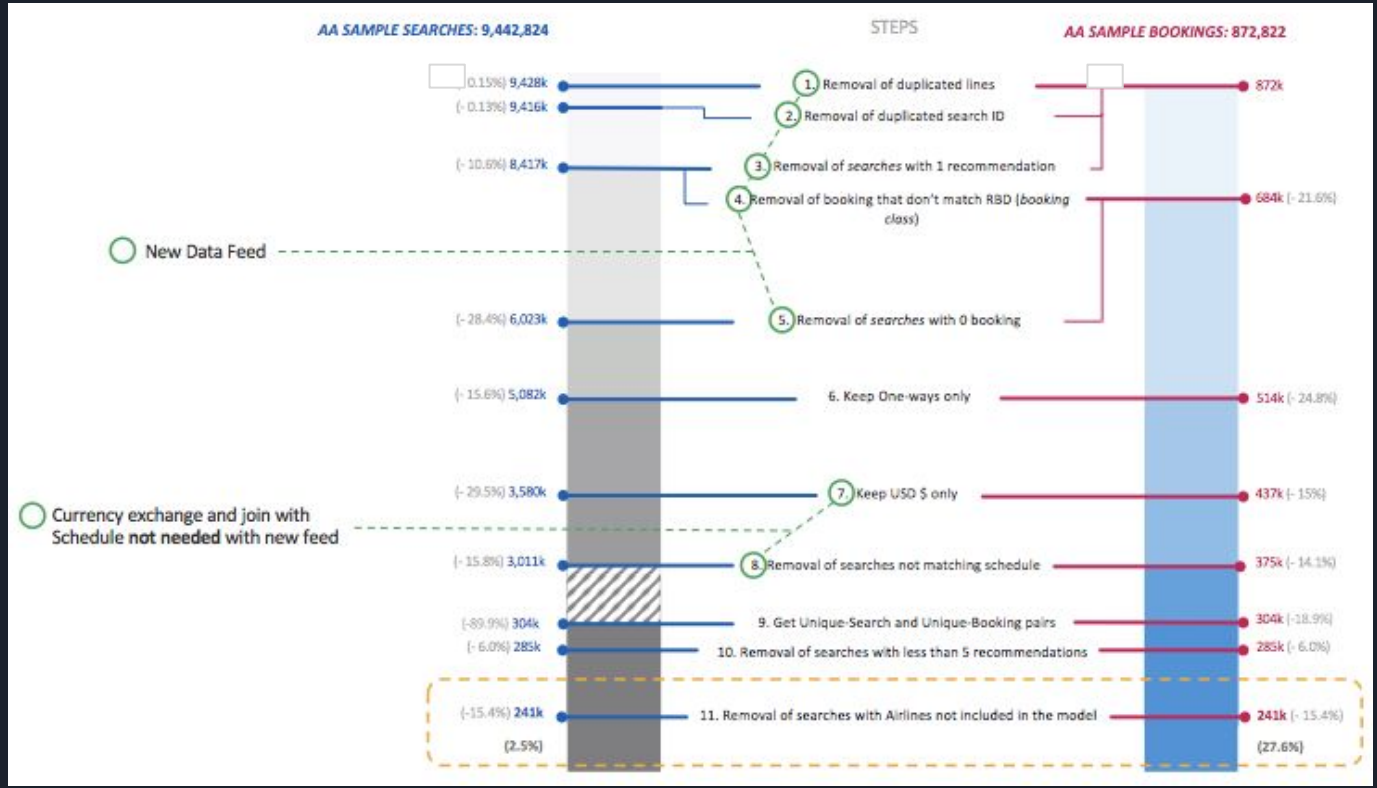
Operating_carrier	At figh level (fixing Nones)
STX	number of the stops for **each flight**
type_of_service	Type of service at **recommendation** level
Interline (flag)	Operating carrier = marketing carrier AND marketing carrier 1 ≠ marketing carrier 2. **Recommnedation level.**
Codeshare(flag)	Operating carrier ≠ marketing carrier. **Flight and recommendation level.**
eft_penalty	Penalty for each flight depending on eft compared with the NS case.

The coefficient related to CNX would be a connect penalty (this would be 0 or 1), STX would be a penalty applied per stop, Interline flag would be an interline penalty, and Codeshare would be a codeshare penalty. => **Penalties calculated at search level.**

Data cleaning & preparation process

1. Parsing data (from python to Spark-Scala).
2. Remove duplicates: this data feed comes from a complex joining/matching job.
3. Remove duplicate keys (search ids).
4. Remove searches with only one recommendation (it is about choice…).
5. Remove searches w/ bookings not matching booking class (after many iterations).
6. Filter one ways searches only (~70% of data), complexity increased for round trips!
7. Filter currency for proper price comparison.
8. Join with Schedule to add #stops.
9. Get unique searches - unique bookings! (non trivial for Big Data).
10. Extra data cleaning before modeling.

Data preparation process



Modeling

Model overview

Multinomial logit model

https://www.youtube.com/watch?v=8Hm2GCKdd5g

- We trained a multinomial logit model in order to obtain a utility function.
- In economics, utility function is a concept that measures preferences over a set of goods and services. Utility is measured in units called utils, which represent the welfare or satisfaction of a consumer from consuming a certain number of goods.
- In this case, utility is the score assigned to every recommendation.

Modeling

Utility function

- Considered variables: Departure time (hour), Connecting time (hour) and Number of Connections

$$UTILITY = \beta_{N\_Connections} \cdot N\_Connections + \beta_{connectingTime} \cdot connectingTime + \beta_{dpt\_hour}$$

- Reference value for Connections is 0
- Reference value for Connecting_time is 0
- Reference departure hour is 12:00 PM

Results with interpretation

	VALUE	STD ERR
B_CONNECTING_TIME	-0.21	0.007
B_CONNECTIONS	-2.64	0.068
B_TIME_00	0.37	0.111
B_TIME_01	0.51	0.110
B_TIME_02	0.96	0.123
B_TIME_03	0.51	0.224
B_TIME_04	-4.46	24.249
B_TIME_05	2.30	0.322
B_TIME_06	-0.10	0.209
B_TIME_07	0.21	0.163
B_TIME_08	1.26	0.130
B_TIME_09	0.69	0.153
B_TIME_10	-0.04	0.154
B_TIME_11	1.01	0.103
B_TIME_13	-0.70	0.177
B_TIME_14	0.10	0.220
B_TIME_15	-0.37	0.235
B_TIME_16	0.08	0.197
B_TIME_17	-0.30	0.240
B_TIME_18	-0.16	0.180
B_TIME_19	0.39	0.138
B_TIME_20	0.54	0.181
B_TIME_21	0.40	0.236
B_TIME_22	0.21	0.343
B_TIME_23	0.67	0.118

Recommendation 1
Recommendation with no connections (0 connecting time), departing at 12 pm.

Utility_1 = 0

Recommendation 2
Recommendation with 1 connection and 1 hour of connecting time, departing at 8 AM.

Utility_2 = -2.64*1 + (-0.21*1) + 1.26 < 0

Conclusion
Recommendation 1 has a higher score than Recommendation 2 and is thus preferred by travelers.

Results with interpretation

	VALUE	STD ERR
B_CONNECTING_TIME	-0.21	0.007
B_CONNECTIONS	-2.64	0.068
B_TIME_00	0.37	0.111
B_TIME_01	0.51	0.110
B_TIME_02	0.96	0.123
B_TIME_03	0.51	0.224
B_TIME_04	-4.46	24.249
B_TIME_05	2.30	0.322
B_TIME_06	-0.10	0.209
B_TIME_07	0.21	0.163
B_TIME_08	1.26	0.130
B_TIME_09	0.69	0.153
B_TIME_10	-0.04	0.154
B_TIME_11	1.01	0.103
B_TIME_13	-0.70	0.177
B_TIME_14	0.10	0.220
B_TIME_15	-0.37	0.235
B_TIME_16	0.08	0.197
B_TIME_17	-0.30	0.240
B_TIME_18	-0.16	0.180
B_TIME_19	0.39	0.138
B_TIME_20	0.54	0.181
B_TIME_21	0.40	0.236
B_TIME_22	0.21	0.343
B_TIME_23	0.67	0.118

Example 1

According to the results, direct flights are usually preferred, and any extra connection has an important penalty (**-2.64**)

Example 2

According to the results, an extra hour in connecting time (-0.21) could be balanced changing departure hour from 12 (utility zero) to 22 (utility +0.21).

Comment: Large standard error is due to the limited data sample.

Product delivery: Brainstorming

Prepare a mock-up and/or exposition on the results for the client, including further improvements planned.

Examples from past editions

https://github.com/carmenvaron/Airbnb-Project

https://github.com/PauTen/twitterindicators

https://github.com/AlejandroCantera/Urban-Data-Classification

https://github.com/sergio-valmorisco-sierra/tfm_data_science_sergio_valmorisco_sierra

https://github.com/ElsaDuran/reves

https://github.com/antonioramos1/master-data-science-final-project

https://github.com/vlavandeira/tfm-bicimad

https://github.com/sergioberdiales/TFM_KSchool_Gijon_Air_Pollution

https://github.com/msanzsanz/PFM_EconomicNewsImpact

Thank you!