SKYSCANNER

IN COLLABORATION WITH

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC)

FINAL DEGREE PROJECT

Skyscanner Heatmap

Skyscanner's data domain representation

Author:
Fèlix Arribas

Director:
Francisco López
University supervisor:
Maria José Casañ

A Project for the Computer Engineering Degree *in the*
Software Engineering and Information Systems department
Facultat d'Informàtica de Barcelona (FIB)
working with
DeLorean squad *from* Marketplace Engine tribe

Wednesday 10th January, 2018

Universitat Politècnica de Catalunya (UPC)

Abstract

Facultat d'Informàtica de Barcelona (FIB)
Software Engineering and Information Systems department

Computer Engineering Degree

Skyscanner Heatmap

by Fèlix Arribas

In the last century, the world has became smaller. Communications are easier and faster than fifty years ago. Back then, you could talk through a fix phone, but you were not able to send any kind of media, like photos, videos, etc. Only the latest technology of that moment was able to do that. Since the smart phone revolution in 2007 almost everyone can text messages, sending images, share live videos or almost whatever you can imagine in less than a second.

But the internet, phones and communications are not the only thing that made the world smaller. Ways of traveling helped to this earth flattering too. In 1918 visiting another place was very difficult. If you wanted to go through the sea, you had to do it by boat. The fastest way to travel very far in a continent was by train, but not all places were connected with rails. Nowadays, all along with the internet revolution, anyone can travel to the other side of the world in less than a day by plane. Even for traveling inside the same country people use planes.

But, is the air industry as efficient enough? Are all airlines users satisfied with their purchases and possibilities? Skyscanner provides an easy to use tool to search cheap flights from any airport to another. Sadly, sometimes is difficult for users to find what they really want.

This project wants to help solving this problem, providing a HeatMap to explore differences and similarities between what users search and what airlines provides. Being able to compare between specific dates to guess user behavior.

Contents

Chapter 1

Context

This is a project developed in *Skyscanner* and evaluated by the *Universitat Politècnica de Catalunya (UPC)* as a Final Degree Project.

The main goal of this project is creating a tool for *Skyscanner* to ease the routes comparison by different parameters, taking into account values like **user demand** and **flights provided** by airlines.

Using this comparison, flights advertisement could be improved according to user demand. The company could also develop complex software using the huge amount of data it will compare through an Application Programming Interface.

Skyscanner have more than 75 million flights information and all its users queries. In order compare all the data available and get significant results, the software should take into account all possible risks working with Big Data frameworks.

1.1 Skyscanner

Skyscanner[1] was formed in 2004 when a group of people was frustrated by the difficulties of finding cheap flights.

In 12 years has evolved from a little office in the suburbs of Edinburgh to a world wide company with ten offices in seven different countries. Having more than 4 million visitors every day and more or less half a million pounds of revenue per day.

Now, is one of the top travel fare aggregate website. In the next 5 years, Skyscanner wants to become the travel experience that people prefer to the myriad confusing and unconnected travel apps.

This growth is possible thanks to the revenue Skyscanner gets from the App and Website, but how does this company make money? Does it get money from its adds as Google and other top tech companies do? Or it sells valuable information to its stakeholders such as user trends like Facebook or Twitter?

Since Skyscanner does not actually sell the flights (or hotel rooms or car hires) it cannot take a percent of the purchase. Skyscanner serves to the user a lot of data from different providers and once the user has selected what he wants to buy, it is redirected to the provider website to finish the acquisition.

The provider knows where the user comes from and they give a percent of the profit to Skyscanner.

1.2 Marketplace Engine tribe

This tribe[2][1] is one of the most important tribes in Skyscanner, its mission is to provide the most comprehensive and accurate flight inventory for Skyscanner and her partners with minimum latency.

Its main goal is to evolve the search, pricing, routes and browse services to be horizontally scalable and set us up to build a lightning fast, super accurate and fully comprehensive flight search engine, enabling the traveler to instantly find the best flight at the best price with minimum effort.

1.3 DeLorean squad

DeLorean[3] is a squad of Marketplace Engine tribe, its mission is to provide the best data and services around the routes, timetables and modes of transportation to go from one point on Earth to another.

The squad now provides a very fast service that serves flights logistic information between a given origin and destination. Some information you can find in a route is the fight number, carriers, stops, date ranges, etc.

[1] Learn more about *Skyscanner structure*

Chapter 2

State-of-the-art

2.1 Kayak

2.2 Google Flights

Chapter 3

Scope of the project

3.1 Mission

In any moment, Skyscanner uses user sessions or user queries to guess trends.

Found that gap in Skyscanner services a bunch of new ideas appeared. After some talks with product owners of different projects and some senior engineers a promising idea showed up.

Comparison of **user demand** and **flights offered** by airlines to find *over-requested*[1] routes or airports.

The team where I currently work, DeLorean squad, manages a huge amount of data: All flights planned for the next years, this are more than 75 million records. And the database with all user queries in the website is even bigger. I there were only one query per visitor the database would have 4 million new records per day. Not much more information needed to say that this is **Big Data** problem.

Finally, with DeLorean squadproduct owner's help, we found some use cases for the professing of those 75 million routes and all user session's queries to get some significant results.
Provide a visual tool to find routes with much more demand than offer and be able to obseve the evolution of it.

A route with a lot of demand but not enough offer to cover it will be named **over requested route**. A route with much more offer but not that amount of demand will be named **non profitable route**.

3.2 Scope

Skyscanner already provides services to get routes information, user requests and quotes of the routes.

[1]Being over-requested routes or airports those which the amount of user queries are higher than flights offered by airlines.

3.2.1 Pipeline

3.2.2 Service

3.2.3 Visual representation

3.2.4 Other applications

3.3 Risks

3.4 Methodology and rigor

3.4.1 Skyscanner structure

3.4.2 Extreme Programming

3.4.3 GitLab

3.4.4 Jira

3.4.5 Other tools

Chapter 4

Requirements analysis

4.1 Actors

Initially it seemed difficult to find stakeholders and actors in these project apart from the providers. It is not a tool for the user of Skyscanner so, as explained before, one risk of these project was not finding enough support.

After walking with the Squad Lead and then the Product Own of DeLorean squad a lot of stakeholders appeared: DeLorean Squad, Marketing Automation Squad, Fuel RaTS Squad, etc. Each of these stakeholders has different use cases and the project became very interesting for a considerable part of Skyscanner .

4.1.1 DeLorean squad

DeLorean's Single Flight Number service, also known as *Timetable SFN Service*, provides all the **current** flights. This is a little bit of a problem when trying to get past routes: Timetable SFN Service does not provide past flights information, it is always **up-to-date**. In order to get this data it is needed to go one step back in the whole DeLorean data processing: *Timetable Pipeline*.

The heatmap must reference old versions of the file created by the Timetable Pipeline to get older routes.

Timetable SFN Service

The *timetable SFN* endpoint returns details for time tabled Single Flight Number itineraries series. Note that SFNs are not ticket-able, so they do not include itineraries which cannot be bought on their own, neither the price nor restrictions.

Timetable Pipeline

This phase, basically collects all the OAG[1] from a provider and maps it into routes in JSON[5] format. For each different version of the OAG file, the pipeline creates a new file with all the routes.

Product Owner

Jen Agerton is the Product Owner of DeLorean Squad. She found that the Heatmap is very useful for other squads like Marketing Automation squad and providers (air carrier companies).

[1]OAG file (also know as WTF file), is a CSV[4] file which each row represents a timetable for a Single Flight Number.

DeLorean's Squad Lead

Francisco López is also the supervisor of this project. Me and had the initial idea for this project. He oriented it for a Machine Learning purpose: The information that the heatmap stores is very useful for constructed routes.

4.1.2 Fuel RaTS squad

Routes and Timetable Servies Squad provides the best data and services around the routes, timetables and modes of transportation to go from one point on Earth to another. Fuel RaTS has the same mission as DeLorean Squad, but develop different services. Since Fuel RaTS provides basic routes data, pricing, live update information and multi-destination combinationcs, DeLorean squad provides a very fast service for only routes.

4.1.3 Marketing Automation squad

Marketing Automation squad enables scalable growth by automating workflows, and the collection of insightful data. They have three main goals:

- Provide data to support decision making
- Automated, data driven campaign management
- Budget process automation

4.1.4 User

The user of this project can change a lot in the future. Now it will be only Marketing Automation Squad developers and Skyscanner employeers. But it is oriented for

4.2 Functional requirements

4.3 Non functional requirements

4.4 Use cases

Name	Routes offer and demand comparison **heatmap**
ID	UC0
Description	Heatmap of the comparison between providers offer and user demand. The heat is represented by the *over requests* of a route.
Actors	User
Triggers	Loading home page
Precondition	
Postcondition	Wolrd heatmap with most relevant routes and their heat.
Basic Flow	
Alternate Flow	
Exceptions	

TABLE 4.1: *Routes offer and demand comparison **heatmap*** use case

Name	Offer and demand plot of route
ID	UC1
Description	Compare the user demand and the providers offer of a specific route from city A to city B in a given date in a plot with two data sets, offer and demand.
Actors	User
Triggers	Request to get comparison of route from city A to city B in a specific date.
Precondition	City A and city B exists and there is some connection (SFN or Constructed) in the date.
Postcondition	Plot with the evolution through time of the user demand and air carrier offer. Time limit goes from fist offer apperance to arrival date or current date, depending which comes first.
Basic Flow	1. System provides a list of cities under *origin* tag. 2. User selects an origin city. 3. System provides another list of cities. Now with *destination* tag. 4. User selects destination (See exception 1). 5. System provides an interactive calendar. 6. User selects a date of the calendar (See exception 2). 7. System provides the plot of the demand and offer evolution of the route.
Alternate Flow	Alternate course 1 1. User **changes** destination city (See exception 1). 2. Return to basic flow step 6. Alternate course 2 1. User **changes** date (See exception 2). 2. Return to basic flow step 7.
Exceptions	1. There are no connections between to given cities. 2. There are connections between to given cities, but not in the given date.

TABLE 4.2: *Offer and demand plot of route* use case

Name	Offer and demand data set of route
ID	UC2
Description	Data set of the evolution of the user demand and providers offer in order to create metrics, alerts, etc.
Actors	Marketing Automation squad, DeLorean squad
Triggers	Request to get data set of route from city A to city B in a specific date.
Precondition	City A and city B exists and there is some connection (SFN or Constructed) in the date
Postcondition	Plot with the evolution through time of the user demand and air carrier offer. Time limit goes from fist offer apperance to arrival date or current date, depending which comes first.
Basic Flow	1. System provides an HTTP endpoint to request data. 2. The developer does a GET request to the endpoint with an origin, destination and a date (See exception 1). 3. System provides a data set in JSON format with all the demand and offers of the entity.
Alternate Flow	
Exceptions	1. There no connections between city A and city B in the given date.

TABLE 4.3: *Offer and demand data set of route* use case

Name	name
ID	id
Description	description
Actors	actors
Organzational Benefits	benefits
Frequency of Use	frequency
Triggers	trigger
Precondition	pre
Postcondition	post
Basic Flow	main
Alternate Flow	alt
Exceptions	exc

TABLE 4.4: *title* use case

Bibliography

[1] Mark Logan. *Skyscanner's Strategy*. 2015. URL: https://skyspace.sharepoint.com/sites/CxOGMblogs/Marksblog/Lists/Posts/Post.aspx?ID=2 (visited on 02/27/2018).

[2] Francisco Lopez. *Marketplace Engine Tribe Home*. 2017. URL: https://confluence.skyscannertools.net/display/MET/Marketplace+Engine+Tribe+Home (visited on 02/27/2018).

[3] Francisco Lopez. *DeLorean Home*. 2017. URL: https://confluence.skyscannertools.net/display/DEL (visited on 01/28/2018).

[4] Y. Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. 2005. URL: https://tools.ietf.org/html/rfc4180 (visited on 01/30/2018).

[5] JSON.org. *The JavaScript Object Notation (JSON) Data Interchange Format*. 2017. URL: https://tools.ietf.org/html/rfc8259 (visited on 01/30/2018).