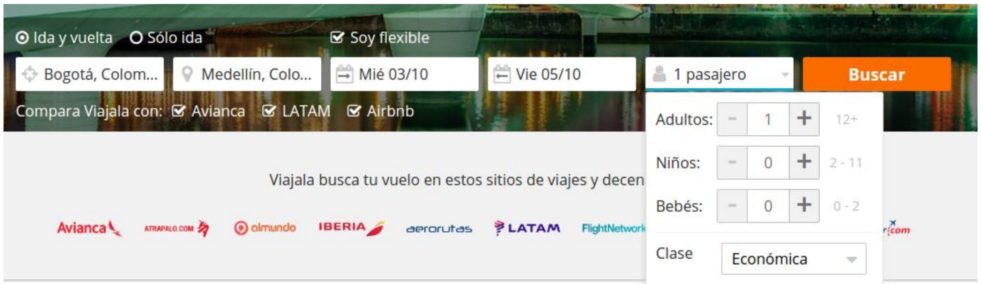
Conversion prediction

Technical test for data scientist

Contact : If you have any question regarding this test, please write to victor@viajala.com.

Context

To search for a flight on Viajala (https://viajala.com.co/), the user enters some characteristics of the travel : origin, destination, travel type, dates of travel, number of passengers and travel class.



● Ida y vuelta ○ Sólo ida ☑ Soy flexible

Bogotá, Colom... | Medellín, Colo... | Mié 03/10 | Vie 05/10 | 1 pasajero

Compara Viajala con: ☑ Avianca ☑ LATAM ☑ Airbnb

Adultos: 1 (12+)

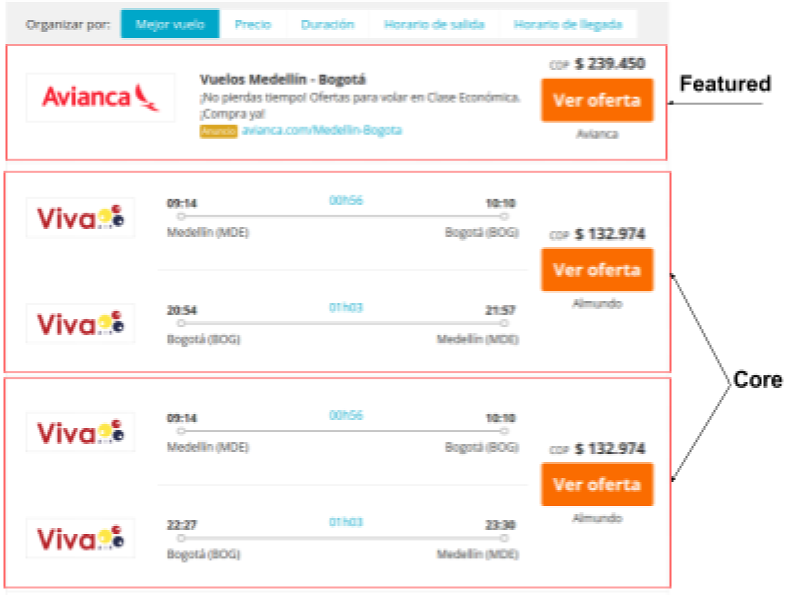
Niños: 0 (2 - 11)

Bebés: 0 (0 - 2)

Clase: Económica

Viajala busca tu vuelo en estos sitios de viajes y decen

This step leads him to the results page which displays a set of available offers provided by our partners (airlines, travel agencies, etc…). There are two categories of results : **core** and **featured**. A core result is an offer associated to a specific flight. A featured result is an advertising position paid by a partner.



Organizar por: Mejor vuelo | Precio | Duración | Horario de salida | Horario de llegada

Avianca — Vuelos Medellin - Bogotá
¡No pierdas tiempo! Ofertas para volar en Clase Económica. ¡Compra ya!
Anuncio avianca.com/Medellin-Bogota
COP $ 239.450 — Ver oferta — Avianca

Viva — 09:14 Medellin (MDE) — 00h56 — 10:10 Bogotá (BOG)
Viva — 20:54 Bogotá (BOG) — 01h03 — 21:57 Medellin (MDE)
COP $ 132.974 — Ver oferta — Almundo

Viva — 09:14 Medellin (MDE) — 00h56 — 10:10 Bogotá (BOG)
Viva — 22:27 Bogotá (BOG) — 01h03 — 23:30 Medellin (MDE)
COP $ 132.974 — Ver oferta — Almundo

To book or get more information about a flight (= a core result), the user will click on it and be redirected to the partner page. In this test, this action is called a **conversion**.

Problem

In order to improve user experience, we would like to hide featured results when a conversion is likely to happen during a search. In this context, the test aims at estimating the likelihood of a search to convert.

To make this estimation, two sets of features are available :

- The characteristics of the analyzed search : origin, destination, dates, number of passengers, etc…
- The historic of searches done by the current user.

Data

You will find attached to this document two datasets : *train.csv* and *test.csv.*
Both represent a sample of searches of an unique day (28/03/2018).
Only **train.csv** provides the binary category *conversion*, which defines if there is a conversion or not.

Data fields

- **search_id** : id of the search
- **conversion** : This is the target variable you are going to predict. **(Absent from *test.csv*)**
- **search_country** : country code of the search (Viajala markets are Colombia, Brasil, Argentina, Chile, Peru, Mexico and Ecuador)
- **search_date** : date of the search
- **origin** : iata of the origin
- **destination** : iata of the destination
- **destination_country** : country code of the destination
- **travel_type** : *RT* for Round-Trip, *OW* for One-Way
- **device** : device from which the search has been done
- **source** : name of the source of the user
- **medium** : general category of the source
- **departure_date :** date of flight departure
- **return_date :** date of flight return (only for travel_type = ‘RT’)
- **nb_passengers :** sum of the number of adults, children, and babies entered in the search.
- **previous_searches :** dictionary representing the search historic of the current user on the last 7 days. Values are lists of same length, each position defining a past search. Empty lists means no search

Objective

The objective of this exercise is to build a classifier **maximizing the F1 score**. As results, you will **provide two files**.

1) In a first document, you will explain each step of your work : exploration, feature engineering, model selection, hyperparameter optimization and performance measurement.
   As a conclusion, you will discuss ideas of improvement (data to collect, model to investigate, metrics to optimize, etc...).
   For the presentation, we recommend you to use a Python or R notebook. You can find great examples on Kaggle (https://www.kaggle.com/kernels).

2) You will also provide a function in a R or Python script . It will takes as inputs *train.csv* and *test.csv*. Once the model trained on *train.csv,* the function will estimate the conversion probabilities of *test.csv.* Its outputs will be an array of **probabilities** associated to the searches of *test.csv*.