Rappi - Analytics Challenge 2023

Part 1 - Data Analysis (100 points)

Rappi's Operations team is interested in decreasing the number of orders that are not taken by any courier, due to the fact that they are not attractive enough for couriers. To solve this problem, we have provided a sample of orders created in 1 month. Your objective is to use this dataset to help Rappi **understand which factors determine whether an order will be taken by a courier, and offer recommendations to take actions based on your insights to improve Rappi's operation.** Below you can find a detailed description of the dataset. Please include any code/spreadsheet/notebook you wrote/built for the analysis and delete the dataset when you have finished the challenge. You may use any analytical tool: Python, R, etc. In addition, please detail any assumptions you may need or explain any issues you may encounter.

2.1. Perform any cleaning, exploratory data analysis (EDA), and/or visualizations of the data for this analysis (a few sentences/plots describing your approach will be enough)

Examples of questions to be solved but feel free to investigate new ones that you may come up with:

- How many orders were not taken by any courier?
- What weekday has the higher percentage of non-taken orders?

2.2. Make an analytics model to help Rappi determine whether or not an order is going to be taken by a courier. Discuss why you chose your approach, what alternatives you considered, and any concerns you may have.

2.3. Briefly discuss how Rappi can take advantage of the insights gained from your analysis to increase the number of orders taken by our couriers. (Here take into account that you may be talking to the data team or key business stakeholders)

2.4 Lastly, with your model, discuss how much can you maximize orders for Rappi. Build a rationale case and detail any assumptions you may use.

Codebook

Column name	Description
order_id	Order unique identifier (ID)
store_id	Store unique identifier (ID) of the order
to_user_distance	Distance (km) between store and user location

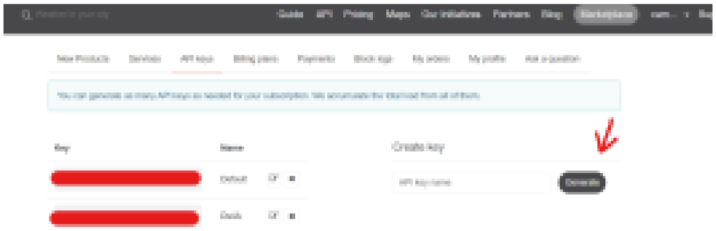
to_user_elevation	Difference in meters between the store and user altitude (m.a.s.l.)
total_earning	Courier’s earning by delivering the order
created_at	Timestamp of order creation
taken	Takes the value of 1 if the order was taken by a courier, 0 otherwise

Part 2 - Bonus - Optional

This is a bonus exercise, optional to do, but a nice challenge.

In Rappi we have all types of challenges on a daily basis, in this case, we want to know how is the current weather situation in the zones where Rappi has operations. For this, we either ask our local teams to register manually a file with the current weather conditions or we can use tech capabilities. With Open Weather tool you get real-time information about the weather in almost every location in the world. This is an open tool with free access, to limit capabilities, but the ones we need so far are free. For this, you can use their API services which the documentation can be found here.

In this challenge, we want to understand how is the weather condition plus all available information that Open Weather can give to us, in some exact locations in Bogotá, Colombia (Lat & Lng) shared in the CSV named Bogota_lat_Ing.csv.



You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name
[redacted]	Default
[redacted]	Dock

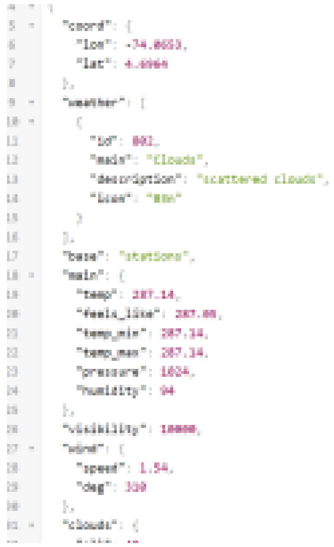
To get the available information via the API you will need to create a free account on Open Weather website and generate a personal API Key (previous image - here) to make the desired requests which are in the form:

https://api.openweathermap.org/data/2.5/weather?lat={lat_value}&lon={lon_value}&appi d={API_KEY}

for example, if you want to know how is the current weather in this location (Lat = 4.6964062, Longitude = -74.0652613) then the request will look like this:

https://api.openweathermap.org/data/2.5/weather?lat=4.6964062&lon=-74.0652613&appid=1234

where “1234” is a dummy API KEY, and you will need to replace this with the one you generate for you. The output will look like this in [JSON Format](#)



```
{
  "coord": {
    "lon": -74.0653,
    "lat": 4.6964
  },
  "weather": [
    {
      "id": 802,
      "main": "Clouds",
      "description": "scattered clouds",
      "icon": "03n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 287.14,
    "feels_like": 287.09,
    "temp_min": 287.14,
    "temp_max": 287.14,
    "pressure": 1024,
    "humidity": 94
  },
  "visibility": 10000,
  "wind": {
    "speed": 1.54,
    "deg": 330
  },
  "clouds": {
```

For all latitudes and longitudes shared in the CSV, add one column per variable you want to extract from the API (eg. weather, description, temperature, feels_like, wind, etc)

The output should look something like this (or Charts/Visualizations/Maps, what you think is the best). Please also include any code/spreadsheet/notebook you wrote/built for the analysis and delete the dataset when you have finished the challenge.

Latitude	Longitude	Current Date	Weather Temperature Feels like ... All Variables you want to add to the data extraction

For this great challenge you can use any programming language you are familiar with, though the language must be able to perform [API requests](#). Python is a good option to go because of its simplicity to perform this kind of task. Good luck!!

Tip: Be careful with any for-loop you perform using the OpenWeather endpoint because if you send more than 60 requests per minute your account can be suspended.