Candidate Assignment

Introduction

Welcome at the Zeelo's **Python Engineer** candidate assignment! With this exercise we would like to assess your ability to write clean/maintainable code as well as your ability to interact with remote APIs to retrieve and crunch data.

The exercise consists in **a Python class (or classes) that provide functionalities to retrieve data from a remote service, process them an visualise the results**. In particular, your program will calculate and present the *quality* of the public transport options between major cities in the United Kingdom and Victoria Station in London, defined as the ratio between the travel time by car and other transport means.

Assignment description

The data sources are:

- an open dataset called **`worldcitypop`** (containing a list of world major cities and their population), available at OpenDataSoft[1], which exposes a free RESTFul API to query the datasets in its catalogue;
- Google Maps Directions API[2] to calculate travel times (you can create a free API key to solve this exercise; let us know if you encounter any problems).

Your program has to:

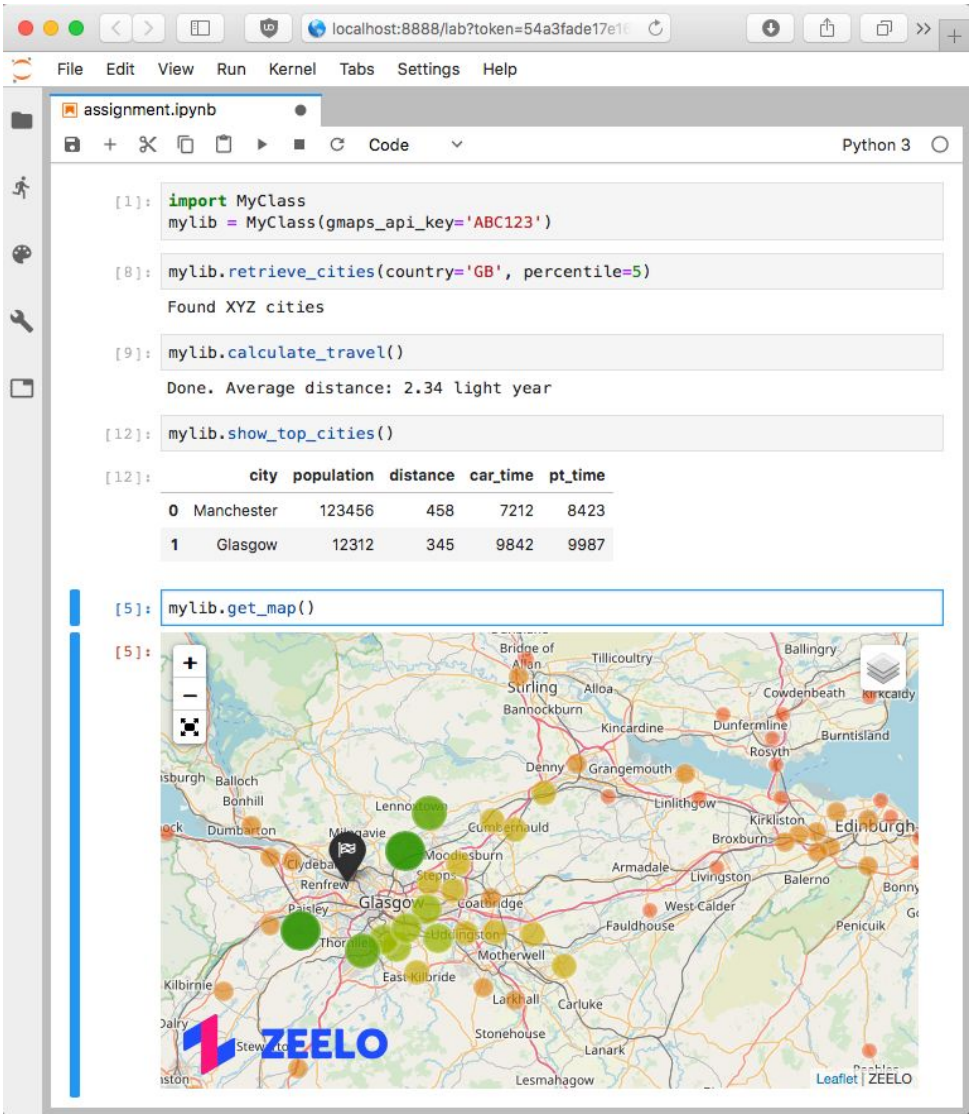
- retrieve a list of UK cities from OpenDataSoft;
- preferably (but not mandatory), load the results in a Pandas DataFrame;
- select the cities in the top 5th percentile, sorted by population;
- calculate the travel times between the resulting cities and "Victoria Station, London", using two different transport modes, namely "driving" (private car) and "transit" (public transportation);
- [BONUS POINT] generate a map (using the tool you prefer, such as Google Maps APIs or Folium/LeafletJS) showing the ratio[3] between the two travel times (collected at the previous point) in a meaningful/informative way. Alternatively, export a CSV.

[1] https://public.opendatasoft.com/explore/dataset/worldcitiespop/
[2] https://developers.google.com/maps/documentation/directions/start
[3] For example: from Glasgow to London (Victoria Station), you need 4:57h by train ("transit") and 6:57 by car ("driving"), with a ratio of 297 minutes / 417 minutes = 0.7. In this case the ratio is <1.0, which means that there are good public transport options (or at least more convenient than a car) connecting these two points.

Suggested setup & results

Your class (or classes) should be provided by a single Python module. We would highly appreciate if your code is imported and executed within a Jupyter notebook, which will show the steps listed above and the resulting visualisation.

The optimal result would look like this (where **`MyClass`** is provided by **`mymodule.py`**):



In case you do not feel comfortable using Jupyter, feel free to adopt your own strategy to achieve the same functionalities.

Requirements / Suggestions

- We would like you to send back your **results in 7 days** after you receive the assignment. During these 7 days, there is no limit on the time you wish to dedicate (although we expect the test to take between 3 and 6 hours) and number of sessions.
- The exercise has to be solved with **Python 3 using the object-oriented paradigm**. You are free to choose your preferred libraries and packages.
- Feel free to make use **third party tools** and try to **not over engineer** your solution, but, in this case, please explain your reasons.
- You should **use GIT** as your VCS. The result has to be handed over in the form of a GIT repository, using a hosting service such as GitHub/Bitbucket/GitLab with a private repo or, alternatively, a local GIT bundle[4], at your choice. Please be aware that we will want to review your commit history.
- The main objectives are: code **maintainability**, **scalability**, **efficiency**, **readability**. We highly favour simpler solutions over complex ones.
- Please clearly **comment your code and make sure that your decisions and assumptions are clear** for other developers.
- Add any comment you find meaningful as Markdown cells in your Jupyter notebook or in an external file.

Contacts and questions

If you have any doubts or questions, please feel free to contact us:

- Pierdomenico Fiadino (Data Science Lead) <piero@zeelo.co>
- Daniel Ruiz (CTO) <daniel@zeelo.co>

Good luck!

[4] https://git-scm.com/docs/git-bundle