Sample rows/columns of City location [data](https://simplemaps.com/data/world-cities) provided by **Company X** isas below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **City** | **Lat** | **Lng** | **Country** | **Capital** | **Population** |
| Tokyo | 35.6897 | 139.6922 | Japan | primary | 37977000 |
| Jakarta | -6.2146 | 106.8451 | Indonesia | primary | 34540000 |
| Delhi | 28.66 | 77.23 | India | admin | 29617000 |
| Mumbai | 18.9667 | 72.8333 | India | admin | 23355000 |
| Manila | 14.5958 | 120.9772 | Philippines | primary | 23088000 |
| Shanghai | 31.1667 | 121.4667 | China | admin | 22120000 |
| São Paulo | -23.5504 | -46.6339 | Brazil | admin | 22046000 |
| Seoul | 37.5833 | 127 | Korea, South | primary | 21794000 |
| Mexico City | 19.4333 | -99.1333 | Mexico | primary | 20996000 |
| Guangzhou | 23.1288 | 113.259 | China | admin | 20902000 |
| Beijing | 39.905 | 116.3914 | China | primary | 19433000 |
| Cairo | 30.0561 | 31.2394 | Egypt | primary | 19372000 |
| New York | 40.6943 | -73.9249 | United States |  | 18713220 |
| Kolkāta | 22.5411 | 88.3378 | India | admin | 17560000 |
| Moscow | 55.7558 | 37.6178 | Russia | primary | 17125000 |
| …… | …… | ……… | …….. | ……… | ……….. |

Relevant column details:

|  |  |  |
| --- | --- | --- |
| **Sl#** | **Column Title** | **Details** |
| 1 | City | City name |
| 2 | Country | Country name |
| 3 | Lat | City latitude |
| 4 | Lng | City longitude |
| 5 | Capital | Whether primary/admin capital |
| 6 | Population | Current city population |

To address the subject requirement, ***Foursquare location data service(FLDS)*** will be used. More specifically:

* **venues/categories** : Returns a hierarchical list of categories applied to venues.
* **venues/search**: Returns a list of venues near the current location, optionally matching a search term.

Above apis will be used. Using *venues/categories* api we will get all the venue category details used by FLDS. Each venue category also contains sub categories.

Using *venues/search* api along with target *city lat/lng* data and top level venue *categoryId*s we will fetch all target venues in the vicinity of any target city.

Important columns of *venues data-frame* are as below:

|  |  |  |
| --- | --- | --- |
| **Sl#** | **Column Title** | **Details** |
| 1 | Name | Venue name |
| 2 | Distance | Distance in m from city center |
| 3 | Lat | Venue latitude |
| 4 | Lng | Venue longitude |
| 5 | Postal-code | Postal code of venue |
| 6 | Address | Venue address |

After that, applying ***k-means clustering*** algorithm on venue location data and varying cluster size we can get the suitable delivery depot latitude & longitude from the kmeans cluster centers.

Finally, to get the actual physical address of the proposed delivery depots given by ***k-means clustering*** we will use reverse Geo-coding service of [*https://revgeocode.search.hereapi.com*](https://revgeocode.search.hereapi.com/) *api.*

To plot the venues and proposed depot location we will use ***folium*** library.

City data : <https://github.com/dasBikash84/Coursera_Capstone/blob/master/worldcities.csv>

Notebook: <https://github.com/dasBikash84/Coursera_Capstone/blob/master/courier_distribution_hub_calculator.ipynb>

Sample city map with all venues: https://github.com/dasBikash84/Coursera\_Capstone/blob/master/Toronto\_only\_venues.html

Sample city map with proposed depot locations: https://github.com/dasBikash84/Coursera\_Capstone/blob/master/Toronto\_venues\_with\_depots.html