**Calculating distance between two geo-locations in Python**

Map

Description automatically generated

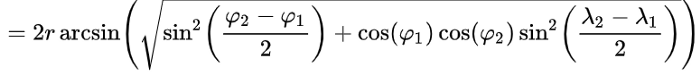
Photo by [Capturing the human heart.](https://unsplash.com/@dead____artist?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com/?utm_source=medium&utm_medium=referral" \t "_blank)

A few months back I was working on a freelance project of visualizing geo-location data(i.e. latitude and longitude) in which I have to visualize central facilities and customers location on a map. As per one of the client’s requirements, I have to find all the customers locations that are within the range of 3 km from different facilities. To do this I have to calculate the distance between all the locations. It was the first time I was working with raw coordinates, so I tried a naive attempt to calculate distance using Euclidean distance, but sooner realized that this approach was wrong.

Euclidean Distance works for the flat surface like a Cartesian plain however, Earth is not flat. So we have to use a special type of formula known as **Haversine Distance.**

*Haversine Distance**can be defined as**the angular distance between two locations on the Earth’s surface.*

Haversine distance can be calculated as:



Source: <https://en.wikipedia.org/wiki/Haversine_formula>

Looks Daunting, yes it would be daunting if you have to apply it using raw python code, but thanks to the python’s vibrant developers community that we have a dedicated library to calculate Haversine distance called haversine(one of the perks of using python).

That’s it with the introduction lets get started with its implementation:

**Step 1: Installing “haversine”**

To install haversine type following command in jupyter notebook.

!pip install haversine

If you are installing through anaconda prompt remove the “!” mark from the above command.

**Step 2: Importing library**

After installing the library import it

import haversine as hs

**Step 3: Calculating distance between two locations**

loc1=(28.426846,77.088834)  
loc2=(28.394231,77.050308)  
hs.haversine(loc1,loc2)

Output: 5.229712941541709

By default the haversine function returns distance in km. If you want to change the unit of distance to miles or meters you can use unit parameter of haversine function as shown below:

from haversine import Unit  
#To calculate distance in meters   
hs.haversine(loc1,loc2,unit=Unit.METERS)

Output: 5229.7129415417085

#To calculate distance in miles   
hs.haversine(loc1,loc2,unit=Unit.MILES)

Output: 3.2495929643035977

Similarly you can calculate distance in inches also.

Calculating distance between two locations is a basic requirement if you are working with raw location data. It not only helps you to visualize better but it also provides an edge to your Machine learning algorithm. It might give an edge to your model and improves its overall efficiency by adding a new dimension “distance”.

This is the output of the project that i mentioned earlier in this article:

Diagram

Description automatically generated

Image by author

If you want to see the whole code you can visit my github page: <https://github.com/ashutoshb418/Foodies-Visualization>