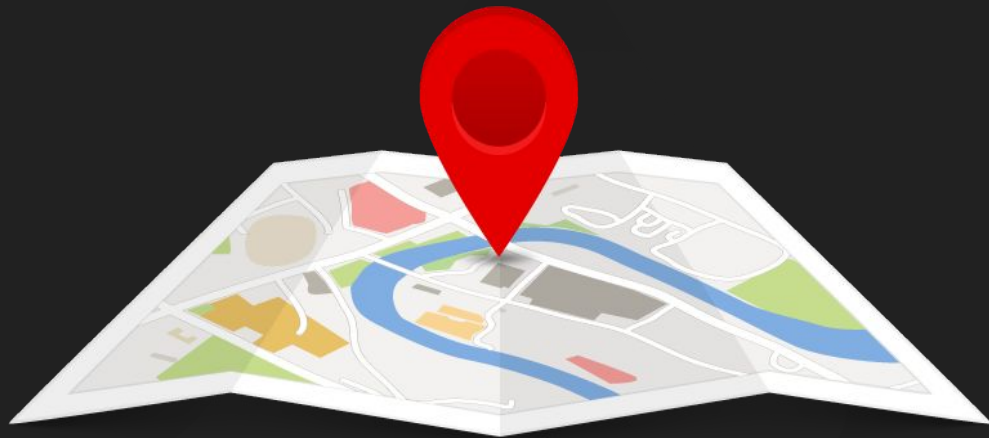






/ Geospatial Data





Coordinates: **Latitude** \updownarrow and **Longitude** \leftrightarrow

- Degrees, minutes, and seconds:
- Decimal degrees:

Latitude \updownarrow

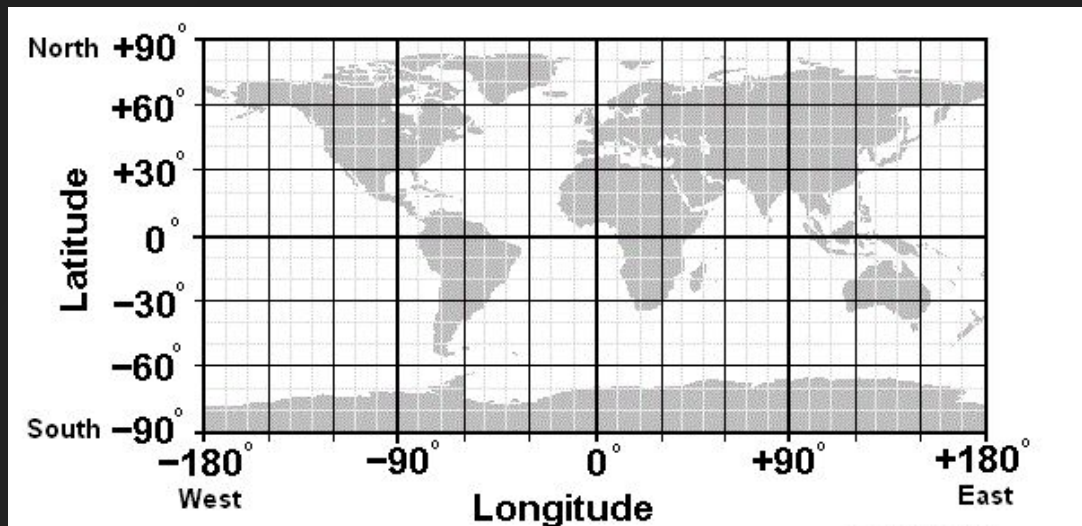
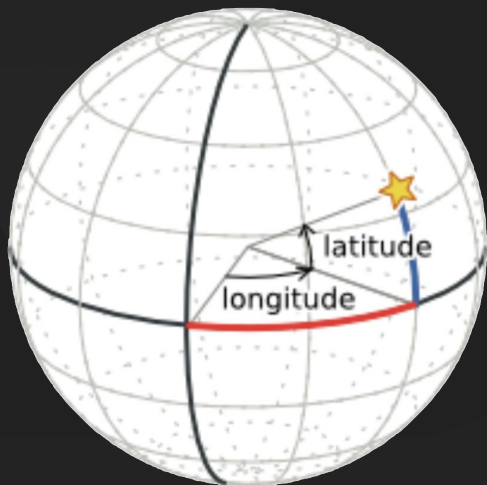
41° 24' 12.2" N

41.40338

Longitude \leftrightarrow

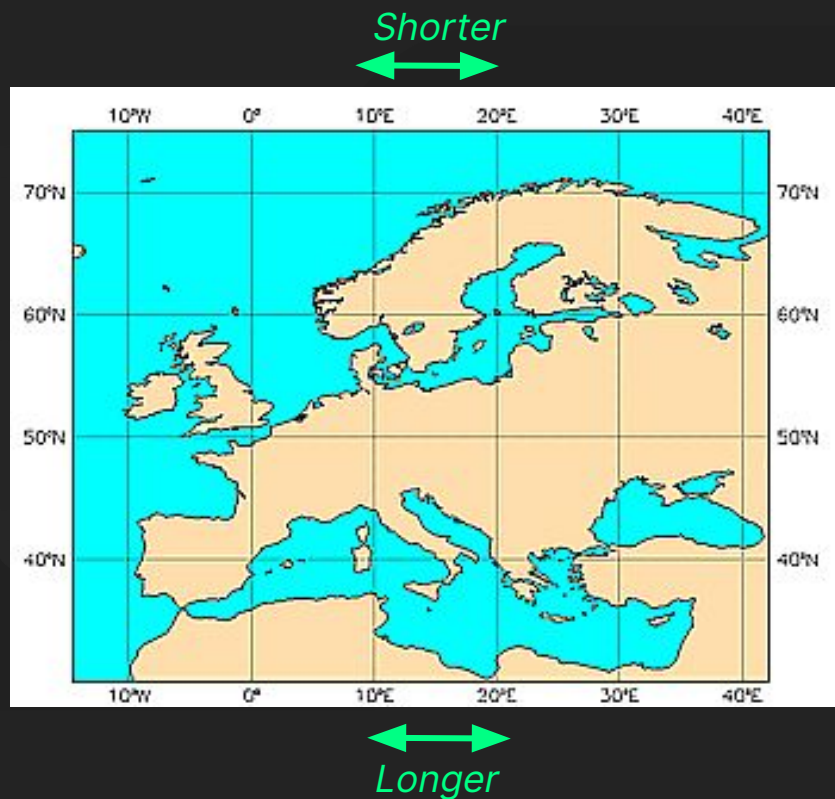
2° 10' 26.5" E

2.17403



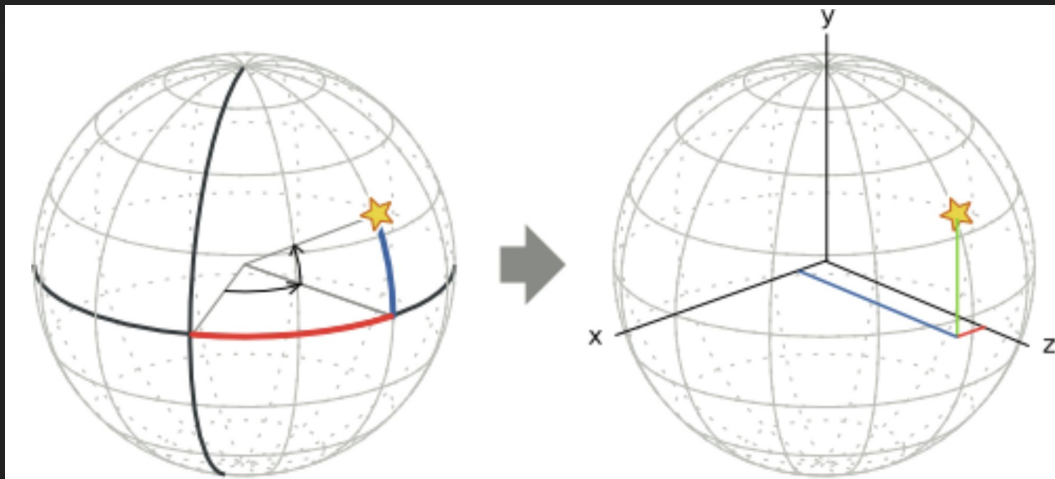


Caution! Not the same distance



 X,Y,Z coordinates

Trigonometry to the rescue!



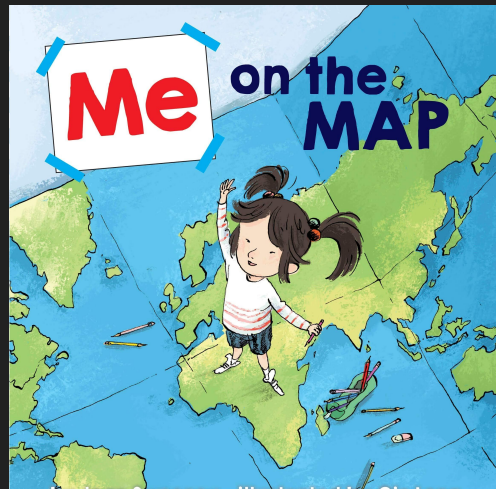
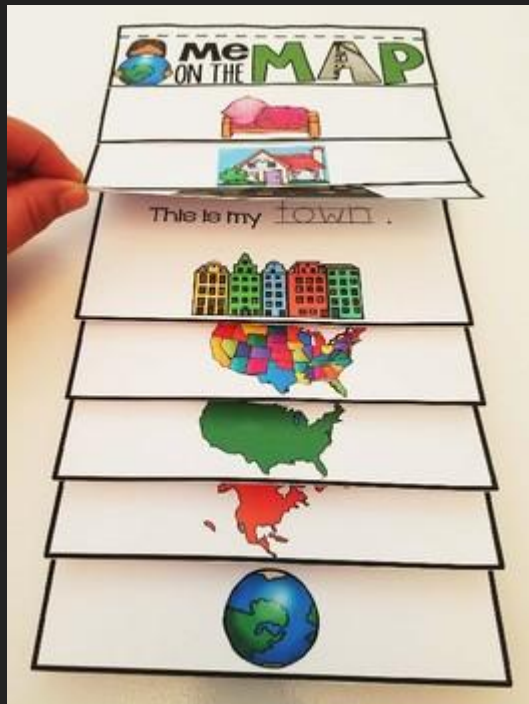
```
df["X"] = np.cos(df["Lat"]) * np.cos(df["Long"])
df["Y"] = np.cos(df["Lat"]) * np.sin(df["Long"])
df["Z"] = np.sin(df["Lat"])
```



Me on the Map!

/ Locations have lot of categorical levels. Use them to get richer features.

1. Street
2. Neighbourhood
3. City/Town
4. County
5. State/Province
6. Country
7. Continent
8. Planet
9. ...





Additional data

/ Generally you can calculate distances to important points on the map. Examples, distance to the:

- Beach
- Important monument
- Nearest food shop
- Nearest school
- Nearest hospital





Additional data

/ Generally you can calculate distances to important points on the map. Examples, distance to the:

- Beach
- Important monument
- Nearest food shop
- Nearest school
- Nearest hospital

Use the Pythagoras's Theorem:

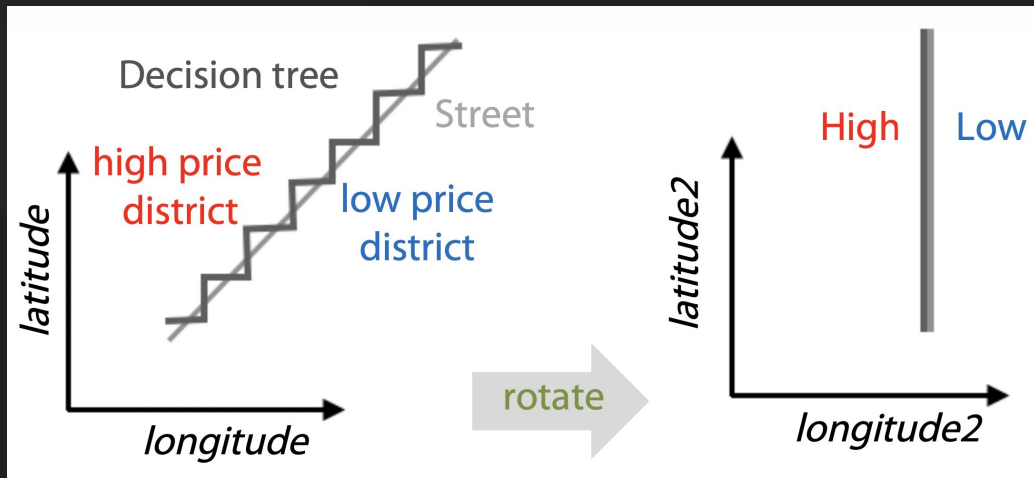
$$\text{dist} = \sqrt{\text{diff_x}^2 + \text{diff_y}^2}$$





Rotate the map!

/ Is better for Tree Model, make splits in a cartesian format.





/ Q&A

What are your doubts?

