Parichaymandal

April 17, 2020

1 Package loading and basic configurations

[1]:
```
%load_ext autoreload
%autoreload 2

# load dependencies'
import pandas as pd
import geopandas as gpd

from envirocar import TrackAPI, DownloadClient, BboxSelector, ECConfig

# create an initial but optional config and an api client
config = ECConfig()
track_api = TrackAPI(api_client=DownloadClient(config=config))
```

2 Querying enviroCar Tracks

The following cell queries tracks from the enviroCar API. It defines a bbox for the area of Münster (Germany) and requests 50 tracks. The result is a GeoDataFrame, which is a geo-extended Pandas dataframe from the GeoPandas library. It contains all information of the track in a flat dataframe format including a specific geometry column.

[2]:
```
bbox = BboxSelector([
    7.601165771484375, # min_x
    51.94807412325402, # min_y
    7.648200988769531, # max_x
    51.97261482608728  # max_y
])

# issue a query
track_df = track_api.get_tracks(bbox=bbox, num_results=50) # requesting 50␣
↪tracks inside the bbox
track_df
```

[2]:

	id	time	geometry	\
0	5e8b930965b80c5d6b4d7cd1	2020-03-07T12:33:15	POINT (7.64069 51.95733)	
1	5e8b930965b80c5d6b4d7cd3	2020-03-07T12:33:20	POINT (7.64118 51.95712)	

```
2    5e8b930965b80c5d6b4d7cd4  2020-03-07T12:33:26  POINT (7.64162 51.95690)
3    5e8b930965b80c5d6b4d7cd5  2020-03-07T12:33:31  POINT (7.64210 51.95672)
4    5e8b930965b80c5d6b4d7cd6  2020-03-07T12:33:36  POINT (7.64264 51.95650)
..                        …                    …                        …
283  5dc986e844ea856b702e3e0b  2019-10-28T16:34:55  POINT (7.59523 51.96505)
284  5dc986e844ea856b702e3e0c  2019-10-28T16:35:00  POINT (7.59425 51.96512)
285  5dc986e844ea856b702e3e0d  2019-10-28T16:35:05  POINT (7.59327 51.96518)
286  5dc986e844ea856b702e3e0e  2019-10-28T16:35:10  POINT (7.59225 51.96525)
287  5dc986e844ea856b702e3e0f  2019-10-28T16:35:15  POINT (7.59123 51.96531)

     GPS PDOP.value GPS PDOP.unit  Speed.value Speed.unit  GPS Altitude.value  \
0          1.090631     precision    28.999999       km/h          110.381939
1          1.000000     precision    28.000000       km/h          108.260375
2          1.257198     precision    28.000001       km/h          105.826028
3          1.000000     precision    30.000000       km/h          104.395998
4          1.026727     precision    31.409419       km/h          101.516865
..              …             …            …         …                   …
283        1.700000     precision    47.999999       km/h          109.652212
284        1.497088     precision    48.297297       km/h          110.122771
285        1.688911     precision    49.000001       km/h          110.573987
286        1.300000     precision    51.000000       km/h          111.140661
287        1.423253     precision    50.000001       km/h          111.891658

    GPS Altitude.unit  GPS Bearing.value  … Consumption.value  \
0                   m         124.858622  …               NaN
1                   m         125.020801  …               NaN
2                   m         121.203960  …               NaN
3                   m         123.412759  …               NaN
4                   m         122.170479  …               NaN
..                …                   …  …               …
283                 m         276.419653  …          3.122268
284                 m         276.271049  …          2.853618
285                 m         275.808021  …          4.657916
286                 m         275.411387  …          3.445271
287                 m         276.124438  …          3.248333

     Consumption.unit track.appVersion  track.touVersion  \
0                 NaN              NaN               NaN
1                 NaN              NaN               NaN
2                 NaN              NaN               NaN
3                 NaN              NaN               NaN
4                 NaN              NaN               NaN
..                …                …                 …
283               l/h              NaN               NaN
284               l/h              NaN               NaN
285               l/h              NaN               NaN
286               l/h              NaN               NaN
```

```
287               l/h              NaN               NaN

    O2 Lambda Voltage ER.value  O2 Lambda Voltage ER.unit MAF.value  MAF.unit  \
0                          NaN                        NaN       NaN       NaN
1                          NaN                        NaN       NaN       NaN
2                          NaN                        NaN       NaN       NaN
3                          NaN                        NaN       NaN       NaN
4                          NaN                        NaN       NaN       NaN
..                         …                          …         …         …
283                        NaN                        NaN       NaN       NaN
284                        NaN                        NaN       NaN       NaN
285                        NaN                        NaN       NaN       NaN
286                        NaN                        NaN       NaN       NaN
287                        NaN                        NaN       NaN       NaN

    O2 Lambda Voltage.value  O2 Lambda Voltage.unit
0                       NaN                     NaN
1                       NaN                     NaN
2                       NaN                     NaN
3                       NaN                     NaN
4                       NaN                     NaN
..                      …                       …
283                     NaN                     NaN
284                     NaN                     NaN
285                     NaN                     NaN
286                     NaN                     NaN
287                     NaN                     NaN

[9944 rows x 54 columns]
```

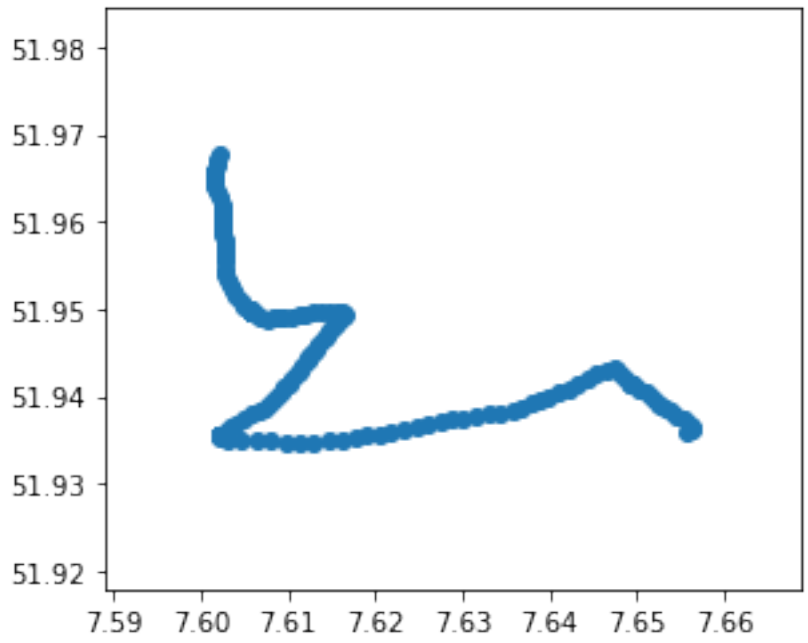
[3]:
```
track_df.plot(figsize=(8, 10))
```

[3]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x12038d090>
```

3 Inspecting a single Track

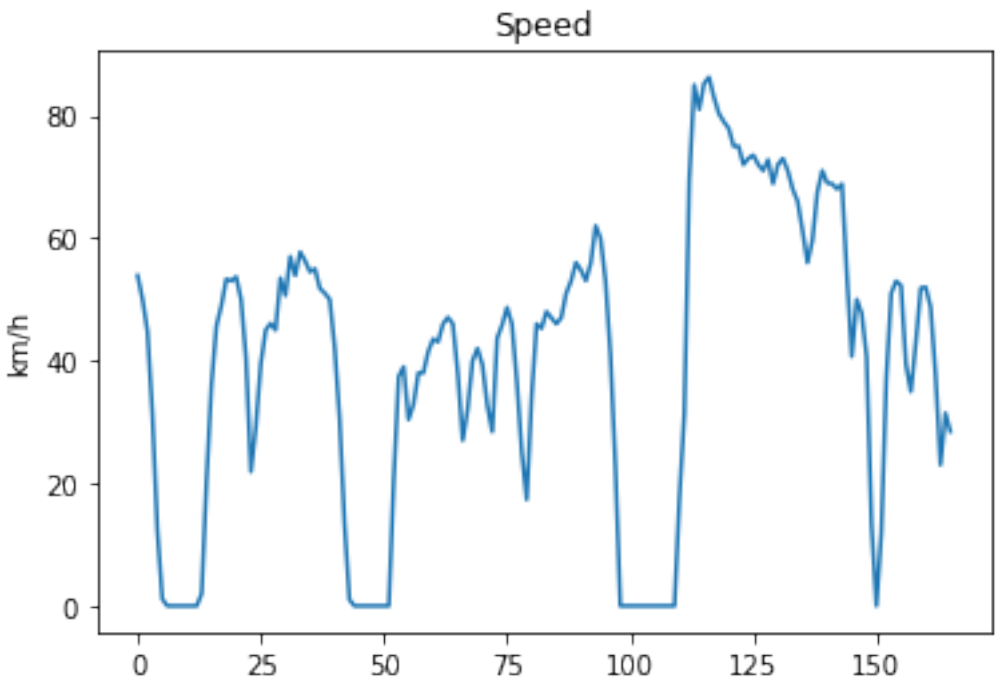
[4]:
```
some_track_id = track_df['track.id'].unique()[5]
some_track = track_df[track_df['track.id'] == some_track_id]
some_track.plot()
```

[4]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x120cd9650>
```



[5]:
```
ax = some_track['Speed.value'].plot()
ax.set_title("Speed")
ax.set_ylabel(some_track['Speed.unit'][0])
ax
```

[5]: <matplotlib.axes._subplots.AxesSubplot at 0x120d36a50>



3.1 Interactive Map

The following map-based visualization makes use of folium. It allows to visualizate geospatial data based on an interactive leaflet map. Since the data in the GeoDataframe is modelled as a set of Point instead of a LineString, we have to manually create a polyline

[6]:
```
import folium

lats = list(some_track['geometry'].apply(lambda coord: coord.y))
lngs = list(some_track['geometry'].apply(lambda coord: coord.x))

avg_lat = sum(lats) / len(lats)
avg_lngs = sum(lngs) / len(lngs)

m = folium.Map(location=[avg_lat, avg_lngs], zoom_start=13)
folium.PolyLine([coords for coords in zip(lats, lngs)], color='blue').add_to(m)
m
```

[6]:
```
<folium.folium.Map at 0x121ec5a50>
```

4 Example: Visualization with pydeck (deck.gl)

The pydeck library makes use of the basemap tiles from Mapbox. In case you want to visualize the map with basemap tiles, you need to register with MapBox, and configure a specific access token.

The service is free until a certain level of traffic is esceeded.

You can either configure it via your terminal (i.e. `export MAPBOX_API_KEY=<mapbox-key-here>`), which pydeck will automatically read, or you can pass it as a variable to the generation of pydeck (i.e. `pdk.Deck(mapbox_key=<mapbox-key-here>, ...)`.

[11]:
```
import pydeck as pdk

# for pydeck the attributes have to be flat
track_df['lat'] = track_df['geometry'].apply(lambda coord: coord.y)
track_df['lng'] = track_df['geometry'].apply(lambda coord: coord.x)
vis_df = pd.DataFrame(track_df)
vis_df['speed'] = vis_df['Speed.value']

# omit unit columns
vis_df_cols = [col for col in vis_df.columns if col.lower()[len(col)-4:
 ↪len(col)] != 'unit']
vis_df = vis_df[vis_df_cols]

layer = pdk.Layer(
    'ScatterplotLayer',
    data=vis_df,
    get_position='[lng, lat]',
    auto_highlight=True,
    get_radius=10,          # Radius is given in meters
    get_fill_color='[speed < 20 ? 0 : (speed - 20)*8.5, speed < 50 ? 255 : 255␣
 ↪- (speed-50)*8.5, 0, 140]',  # Set an RGBA value for fill
    pickable=True
)



# Set the viewport location
view_state = pdk.ViewState(
    longitude=7.5963592529296875,
    latitude=51.96246168188569,
    zoom=12,
    min_zoom=10,
    max_zoom=25,
    pitch=40.5,
    bearing=-27.36)

r = pdk.Deck(
    width=200,
    layers=[layer],
    initial_view_state=view_state,
```

```
    mapbox_key="pk.
 ↪eyJ1IjoicGFyaWNoYXkiLCJhIjoiY2s1bDAzdnFwMDd4YTNnbno3aTcwaDd5aiJ9.
 ↪NeEhX7ksiQ8Q1hkgtsIZXw"
)
r.to_html('tracks_muenster.html', iframe_width=900)
```

```
<IPython.lib.display.IFrame at 0x123e13090>
```

[11]:
```
'/Users/parichay/Desktop/Desktop/Academic/Semester 2/FCDA/envirocar-
py/examples/tracks_muenster.html'
```

[13]:
```
vis_df['consumption'] = vis_df['Consumption.value']

view = pdk.ViewState(
    longitude=7.5963592529296875,
    latitude=51.96246168188569,
    zoom=12,
    min_zoom=10,
    max_zoom=25,
    pitch=40.5,
    bearing=-27.36)

consumption = pdk.Layer(
    "HeatmapLayer",
    data=vis_df,
    opacity=0.9,
    get_position=["lng", "lat"],
    aggregation='"MEAN"',
    threshold=.5,
    get_weight="consumption",
    pickable=True,
)



r = pdk.Deck(
    width=200,
    layers=[consumption],
    initial_view_state=view,
    mapbox_key="pk.
 ↪eyJ1IjoicGFyaWNoYXkiLCJhIjoiY2s1bDAzdnFwMDd4YTNnbno3aTcwaDd5aiJ9.
 ↪NeEhX7ksiQ8Q1hkgtsIZXw"

)

r.to_html("consumption_muenster.html", iframe_width=900)
```

```
<IPython.lib.display.IFrame at 0x121d06a10>
```

[13]:
```
'/Users/parichay/Desktop/Desktop/Academic/Semester 2/FCDA/envirocar-
py/examples/consumption_muenster.html'
```

4.1 Problem I faced

When I was trying run the notebook it was showing an "pandas" attribute error for "json_normalize". I did everything currect but still I was unable to solve this problem. It took my valuable hours to find out the actual problem.

4.2 How I overcame

Later on I found that problem actually relies on the version of "pandas". Upgrading "pandas" solved my problem.

4.3 Result

Modified Result