assignment07_SafsafiAchraf

May 7, 2021

Achraf Safsafi

DSC650

Assignment 7

Assignment 7.1 :

A :

[1]:
```
import os
import hashlib
import pygeohash as pgh
import numpy as np
import pandas as pd
from pathlib import Path
```

[2]:
```
df = pd.read_parquet('routes.parquet', engine='fastparquet')
df.head()
```

[2]:
```
   codeshare equipment  airline.active  airline.airline_id  \
0      False     [CR2]            True                 410
1      False     [CR2]            True                 410
2      False     [CR2]            True                 410
3      False     [CR2]            True                 410
4      False     [CR2]            True                 410

             airline.alias airline.callsign airline.country airline.iata  \
0  ANA All Nippon Airways       AEROCONDOR        Portugal           2B
1  ANA All Nippon Airways       AEROCONDOR        Portugal           2B
2  ANA All Nippon Airways       AEROCONDOR        Portugal           2B
3  ANA All Nippon Airways       AEROCONDOR        Portugal           2B
4  ANA All Nippon Airways       AEROCONDOR        Portugal           2B

  airline.icao airline.name  …  dst_airport.dst  dst_airport.iata  \
```

```
0          ARD   Aerocondor  …                N               KZN
1          ARD   Aerocondor  …                N               KZN
2          ARD   Aerocondor  …                N               MRV
3          ARD   Aerocondor  …                N               KZN
4          ARD   Aerocondor  …                N               OVB

  dst_airport.icao dst_airport.latitude dst_airport.longitude  \
0             UWKD            55.606201             49.278702
1             UWKD            55.606201             49.278702
2             URMM            44.225101             43.081902
3             UWKD            55.606201             49.278702
4             UNNT            55.012600             82.650703

              dst_airport.name dst_airport.source  dst_airport.timezone  \
0  Kazan International Airport        OurAirports                   3.0
1  Kazan International Airport        OurAirports                   3.0
2     Mineralnyye Vody Airport        OurAirports                   3.0
3  Kazan International Airport        OurAirports                   3.0
4           Tolmachevo Airport        OurAirports                   7.0

   dst_airport.type dst_airport.tz_id
0           airport     Europe/Moscow
1           airport     Europe/Moscow
2           airport     Europe/Moscow
3           airport     Europe/Moscow
4           airport  Asia/Krasnoyarsk

[5 rows x 38 columns]
```

[3]:
```
df['key'] = df['src_airport.iata'].astype(str) + df['dst_airport.iata'].
 ↪astype(str) + df['airline.iata'].astype(str)
df['key']
```

[3]:
```
0        AERKZN2B
1        ASFKZN2B
2        ASFMRV2B
3        CEKKZN2B
4        CEKOVB2B
           …
67658    WYAADLZL
67659    DMEFRUZM
67660    FRUDMEZM
67661    FRUOSSZM
67662    OSSFRUZM
Name: key, Length: 67663, dtype: object
```

[4]:
```
df['kv_key'] = df['key'].astype(str).str[0]
df['kv_key']
```

[4]:
```
0        A
1        A
2        A
3        C
4        C
        ..
67658    W
67659    D
67660    F
67661    F
67662    O
Name: kv_key, Length: 67663, dtype: object
```

[5]:
```
df.to_parquet('/home/jovyan/dsc650/results/kv/',partition_cols=['kv_key'],␣
↪engine='pyarrow')
```

results/kv Directory

[6]:
```
def scan_dir(path):
    print([os.path.abspath(f) for f in os.listdir(path)])

scan_dir('/home/jovyan/dsc650/results/kv/')
```

```
['/home/jovyan/kv_key=A', '/home/jovyan/kv_key=N', '/home/jovyan/kv_key=P',
'/home/jovyan/kv_key=Y', '/home/jovyan/kv_key=X', '/home/jovyan/kv_key=C',
'/home/jovyan/kv_key=H', '/home/jovyan/kv_key=E', '/home/jovyan/kv_key=G',
'/home/jovyan/kv_key=I', '/home/jovyan/kv_key=Q', '/home/jovyan/kv_key=R',
'/home/jovyan/kv_key=W', '/home/jovyan/kv_key=V', '/home/jovyan/kv_key=F',
'/home/jovyan/kv_key=K', '/home/jovyan/kv_key=D', '/home/jovyan/kv_key=J',
'/home/jovyan/kv_key=L', '/home/jovyan/kv_key=S', '/home/jovyan/kv_key=Z',
'/home/jovyan/kv_key=T', '/home/jovyan/kv_key=M', '/home/jovyan/kv_key=U',
'/home/jovyan/kv_key=O', '/home/jovyan/kv_key=B']
```

B :

[7]:
```
def hash_key(key):
    m = hashlib.sha256()
    m.update(str(key).encode('utf-8'))
    return m.hexdigest()
```

[8]:
```
df['hashed'] = df['key'].apply(lambda x: hash_key(x))
df['hashed']
```

[8]:
```
0        652cdec02010381f175efe499e070c8cbaac1522bac59a…
1        9eea5dd88177f8d835b2bb9cb27fb01268122b635b241a…
2        161143856af25bd4475f62c80c19f68936a139f653c1d3…
3        39aa99e6ae2757341bede9584473906ef1089e30820c90…
4        143b3389bce68eea3a13ac26a9c76c1fa583ec2bd26ea8…
                                  …
67658    f31527be84c36208c05cac57dfac8a46b48a87dda151f8…
67659    880fc35ca283ad034c90becc4e331b72ee894b9eb69f76…
67660    e976939986fbf947bb9318018cef717c0b34dff91e5e67…
67661    8b0c0b835a58a4250e020d51ec2a896e4ef3f5c3543b8e…
67662    629f14f3fb6f94ebd1522d33a3c50675942e3148d028b4…
Name: hashed, Length: 67663, dtype: object
```

[9]:
```
df['hash_key'] = df['hashed'].astype(str).str[0]
df['hash_key']
```

[9]:
```
0        6
1        9
2        1
3        3
4        1
        ..
67658    f
67659    8
67660    e
67661    8
67662    6
Name: hash_key, Length: 67663, dtype: object
```

[10]:
```
df.to_parquet('/home/jovyan/dsc650/results/hash/',partition_cols=['hash_key'],␣
↪engine='pyarrow')
```

results/hash Directory

[11]:
```
scan_dir('/home/jovyan/dsc650/results/hash/')
```

```
['/home/jovyan/hash_key=1', '/home/jovyan/hash_key=4',
'/home/jovyan/hash_key=e', '/home/jovyan/hash_key=f', '/home/jovyan/hash_key=7',
'/home/jovyan/hash_key=8', '/home/jovyan/hash_key=c', '/home/jovyan/hash_key=5',
'/home/jovyan/hash_key=2', '/home/jovyan/hash_key=d', '/home/jovyan/hash_key=9',
'/home/jovyan/hash_key=0', '/home/jovyan/hash_key=b', '/home/jovyan/hash_key=a',
'/home/jovyan/hash_key=3', '/home/jovyan/hash_key=6']
```

C :

[12]:
```
df = df.rename({'src_airport.latitude':'src_airport_latitude'
, 'src_airport.longitude': 'src_airport_longitude'}, axis=1)

df['src_airport_geodash'] = df.apply(lambda x: pgh.encode(x.
 ↪src_airport_latitude,x.src_airport_longitude,precision=5), axis=1)
df['src_airport_geodash']
```

[12]:
```
0        szsrj
1        v04pk
2        v04pk
3        v3gdx
4        v3gdx
         …
67658    r41gc
67659    ucfgn
67660    txsuy
67661    txsuy
67662    tx5z0
Name: src_airport_geodash, Length: 67663, dtype: object
```

[13]:
```
def det_loc(src_airport_geodash):
    locations = dict(west = pgh.encode(45.5945645,-121.1786823),
    central = pgh.encode(41.1544433,-96.0422378),
    east = pgh.encode(39.08344,-77.6497145))

    dists = []
    for x, y in locations.items():
        hav = pgh.geohash_approximate_distance(src_airport_geodash,y)
        dists.append(tuple((hav,x)))

    dists.sort()
    return dists[0][1]

df['location'] = df['src_airport_geodash'].apply(det_loc)
df['location']
```

[13]:
```
0        central
1        central
2        central
3        central
4        central
          …
67658    central
67659    central
67660    central
```

```
67661    central
67662    central
Name: location, Length: 67663, dtype: object
```

[14]:
```
df['location'][400:600]
```

[14]:
```
400    central
401    central
402    central
403    central
404    central
        …
595       east
596       east
597       east
598       east
599       east
Name: location, Length: 200, dtype: object
```

[15]:
```
df.to_parquet('/home/jovyan/dsc650/results/geo/',partition_cols=['location'],␣
↪engine='pyarrow')
```

results/geo Directory

[16]:
```
scan_dir('/home/jovyan/dsc650/results/geo/')
```

```
['/home/jovyan/location=west', '/home/jovyan/location=central',
'/home/jovyan/location=east']
```

D :

[17]:
```
def balance_partitions (keys, num_partitions):
    partition_counts = (len(sorted(set(keys))) / num_partitions)+1
    partitions  = []
    curRow = 1
    partNum = 1
    for i in range(len(sorted(set(keys)))):
        curKeyVal ={}
        if curRow <= partition_counts:
            curKeyVal[sorted(set(keys))[i]] = partNum
            curRow = curRow + 1
        else:
            curRow = 1
            partNum = partNum + 1
            curKeyVal[sorted(set(keys))[i]] = partNum
            curRow = curRow + 1
```

```
        partitions .append(curKeyVal)
    return partitions
```

Do an example

[18]:
```
example_list = df['src_airport_geodash'].head(100).tolist()
```

[19]:
```
balance_partitions (example_list,8)
```

[19]:
```
[{'00000': 1},
 {'6mc5t': 1},
 {'6mejp': 1},
 {'6msff': 1},
 {'6myb0': 1},
 {'6q1zc': 1},
 {'6qcdy': 2},
 {'6qdbr': 2},
 {'6r7f7': 2},
 {'ebvjy': 2},
 {'ebzzu': 2},
 {'ecuzm': 2},
 {'edeef': 3},
 {'ef4r7': 3},
 {'efnym': 3},
 {'s10gh': 3},
 {'s11sn': 3},
 {'s43s9': 3},
 {'szsrj': 4},
 {'szyes': 4},
 {'tp5w4': 4},
 {'u3y8k': 4},
 {'uc400': 4},
 {'ucfgn': 4},
 {'udts0': 5},
 {'v04pk': 5},
 {'v1gh3': 5},
 {'v1twc': 5},
 {'v1vh1': 5},
 {'v3gdx': 5},
 {'v654z': 6},
 {'vcfbb': 6},
 {'vdy6s': 6},
 {'vewrv': 6},
 {'y361r': 6},
 {'y602d': 6},
 {'y655m': 7},
 {'y90xf': 7},
```

```
{'yd31p': 7},
{'ydc9k': 7},
{'ye15g': 7},
{'ygh31': 7}]
```