Assignment7

July 20, 2023

[54]:

```
#Assignment07
!pip install IPython --quiet
```

[2]:

```
import os
from pathlib import Path
import json
import gzip
import pandas as pd
import hashlib
import shutil
import pygeohash
import s3fs
import sqlite3
import uuid
import math
import itertools
import IPython
import pyarrow as pa
import pyarrow.parquet as parq
```

[75]:

```
endpoint_url='https://storage.budsc.midwest-datascience.com'

current_dir = Path(os.getcwd()).absolute()
results_dir = current_dir.joinpath('results')

if results_dir.exists():
    shutil.rmtree(results_dir)
results_dir.mkdir(parents=True, exist_ok=True)

def read_jsonl_data():
    #s3 = s3fs.S3FileSystem(anon=True,
                           #client_kwargs={'endpoint_url': endpoint_url})

    src_data_path = '/home/jovyan/DSC650/data/processed/openflights/routes.
↪jsonl.gz'
    with open(src_data_path, 'rb') as f_gz:
        with gzip.open(f_gz, 'rb') as f:
            records = [json.loads(line) for line in f.readlines()]

    return records
```

[76]:
```
def flat_tire(record):
    flat_tire=dict()
    for key, value in record.items():
        if key in ['airline','src_airport', 'dst_airport']:
            if isinstance(value,dict):
                for child_key, child_value in value.items():
                    flat_key='{}_{}'.format(key, child_key)
                    flat_tire[flat_key]=child_value
        else:
            flat_tire[key]=value
    return flat_tire
```

[77]:
```
def create_broken_van():
    passengers=read_jsonl_data()
    parquet_path=results_dir.joinpath('flat_routes.parquet')
    return pd.DataFrame.from_records([flat_tire(record) for record in␣
 ↪passengers])
```

[78]:
```
df=create_broken_van()
df['key']=df['src_airport_iata'].astype(str)+df['dst_airport_iata'].
 ↪astype(str)+df['airline_iata'].astype(str)
```

[82]:
```
df.head(1)
```

[82]:
```
   airline_airline_id airline_name           airline_alias airline_iata
0                 410   Aerocondor  ANA All Nippon Airways           2B  \

  airline_icao airline_callsign airline_country  airline_active
0          ARD       AEROCONDOR        Portugal            True  \

   src_airport_airport_id             src_airport_name  …
0                  2965.0  Sochi International Airport  …  \

  dst_airport_longitude dst_airport_altitude dst_airport_timezone
0             49.278702                411.0                  3.0  \

  dst_airport_dst  dst_airport_tz_id  dst_airport_type  dst_airport_source
0               N      Europe/Moscow           airport         OurAirports  \

   codeshare equipment       key
0      False     [CR2]  AERKZN2B

[1 rows x 39 columns]
```

[83]:

```
partitions = (
        ('A', 'A'), ('B', 'B'), ('C', 'D'), ('E', 'F'),
        ('G', 'H'), ('I', 'J'), ('K', 'L'), ('M', 'M'),
        ('N', 'N'), ('O', 'P'), ('Q', 'R'), ('S', 'T'),
        ('U', 'U'), ('V', 'V'), ('W', 'X'), ('Y', 'Z'))
```

[86]:

```
def get_kv_partitions(partitions,string):
    first_char=string[0]
    for tup in partitions:
        if first_char in tup:
            if tup.count(tup[0])==len(tup):
                return tup[0]
            else:
                return tup[0]+'-'+tup[1]
    return 'None'
```

[87]:

```
df['kv_key']=df['key'].apply(lambda x: get_kv_partitions(partitions,x))
```

[88]:

```
df.sample(1)
```

[88]:

```
       airline_airline_id    airline_name      airline_alias airline_iata
24220                2222  Etihad Airways  Emirates Airlines           EY  \

      airline_icao airline_callsign       airline_country  airline_active
24220          ETD           ETIHAD  United Arab Emirates            True  \

       src_airport_airport_id            src_airport_name  …
24220                  3156.0  Malé International Airport  …  \

      dst_airport_altitude dst_airport_timezone dst_airport_dst
24220                157.0                  5.5               N  \

      dst_airport_tz_id  dst_airport_type  dst_airport_source  codeshare
24220      Asia/Colombo           airport         OurAirports       True  \

       equipment       key kv_key
24220      [32S]  MLEHRIEY      M

[1 rows x 40 columns]
```

[100]:

```
#File Away
df.to_parquet('./results/kv',partition_cols=['kv_key'])
```

[90]:

```
#7b.
```

[91]:

```
import hashlib
```

[92]:

```
def hash_keys(key):
    ha=hashlib.sha256()
    ha.update(str(key).encode('utf-8'))
    return ha.hexdigest()
```

[95]:

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

[96]:

```
df['hash_key']=df['hashed'].apply(lambda x:x[0])
```

[111]:

```
df[['hashed','hash_key']][:7]
```

[111]:

```
                                              hashed hash_key
0  652cdec02010381f175efe499e070c8cbaac1522bac59a…        6
1  9eea5dd88177f8d835b2bb9cb27fb01268122b635b241a…        9
2  161143856af25bd4475f62c80c19f68936a139f653c1d3…        1
3  39aa99e6ae2757341bede9584473906ef1089e30820c90…        3
4  143b3389bce68eea3a13ac26a9c76c1fa583ec2bd26ea8…        1
5  e4ec7b234cd26c4afd736cd49d1d02e4ec5f294f14533a…        e
6  30114a9dc60716adbadf6c54124a899a66eea47335fdae…        3
```

[101]:

```
#File Away
df.to_parquet('.results/hash', partition_cols=['hash_key'])
```

[114]:

```
#7c.
from pandas.core.apply import frame_apply
```

[115]:

```
!pip install geolib --quiet
```

[116]:

```
import numpy as np
import sklearn.neighbors
from geolib import geohash
```

[105]:

```
df.columns
```

[105]:

```
Index(['airline_airline_id', 'airline_name', 'airline_alias', 'airline_iata',
       'airline_icao', 'airline_callsign', 'airline_country', 'airline_active',
       'src_airport_airport_id', 'src_airport_name', 'src_airport_city',
       'src_airport_country', 'src_airport_iata', 'src_airport_icao',
       'src_airport_latitude', 'src_airport_longitude', 'src_airport_altitude',
       'src_airport_timezone', 'src_airport_dst', 'src_airport_tz_id',
       'src_airport_type', 'src_airport_source', 'dst_airport_airport_id',
       'dst_airport_name', 'dst_airport_city', 'dst_airport_country',
       'dst_airport_iata', 'dst_airport_icao', 'dst_airport_latitude',
       'dst_airport_longitude', 'dst_airport_altitude', 'dst_airport_timezone',
       'dst_airport_dst', 'dst_airport_tz_id', 'dst_airport_type',
       'dst_airport_source', 'codeshare', 'equipment', 'key', 'kv_key',
       'hased', 'hashed', 'hash_key'],
      dtype='object')
```

[117]:

```
df['src_airport_geohash']=df.apply(lambda row: pygeohash.encode(row.
 ↪src_airport_latitude, row.src_airport_longitude), axis=1)
```

[118]:

```
def determine_loc(src_airport_geohash):
    loc=dict(West= pygeohash.encode(45.5945645,-121.1786823),
             Central= pygeohash.encode(41.1544433,-96.0422378),
             East= pygeohash.encode(39.08344, -77.6497145))
    dist=[]
    for location, geohash in loc.items():
        haval= pygeohash.geohash_haversine_distance(src_airport_geohash,␣
 ↪geohash)
        dist.append(tuple((haval, location)))

        dist.sort()
        return dist[0][1]
```

[119]:

```
df['location']= df['src_airport_geohash'].apply(determine_loc)
```

[120]:

```
df.to_csv('geo_test', sep=',')
```

[122]:

```
#File away
df.to_parquet('results/geo', partition_cols=['location'])
```

[123]:

```
#7d.
```

[124]:

```
def balance_partitions(keys, num_partitions):
    part_sizes= round(len(keys)/num_partitions)
    iterats= iter(keys)
    partition_iterats= iter(lambda: tuple(itertools.islice(iterats,␣
 ↪part_sizes)), ())
    partitions = [sorted(part) for part in partition_iterats]
    return partitions
```

[125]:

```
df.sample(1)
```

[125]:

```
       airline_airline_id        airline_name airline_alias airline_iata
64098                4547   Southwest Airlines       SkyWork           WN  \

      airline_icao airline_callsign airline_country  airline_active
64098          SWA        SOUTHWEST   United States            True  \

       src_airport_airport_id                      src_airport_name  …
64098                  3747.0  Chicago Midway International Airport  …  \

      dst_airport_source codeshare   equipment       key   kv_key
```

```
64098        OurAirports     False  [73W, 73H]  MDWHOUWN       M  \

                                                   hased
64098  29a9eb72bf76d11fa9439402d88639fea088ac78b813e5…  \

                                                  hashed  hash_key
64098  29a9eb72bf76d11fa9439402d88639fea088ac78b813e5…         2  \

      src_airport_geohash location
64098        dp3tenuthfcc     West

[1 rows x 45 columns]
```

[128]:
```python
airline_brand= df.airline_iata.sample(70).to_list()
partitions= balance_partitions(airline_brand, 7)
partitions
```

[128]:
```
[['5Q', '9C', 'AP', 'AY', 'CZ', 'DY', 'KN', 'SC', 'SG', 'X3'],
 ['3U', 'AA', 'AB', 'BC', 'DL', 'EK', 'KL', 'LA', 'NH', 'ZH'],
 ['9W', 'BA', 'CA', 'CZ', 'GA', 'KA', 'KM', 'S7', 'ST', 'UA'],
 ['AD', 'CX', 'DL', 'DL', 'MM', 'MU', 'OS', 'QF', 'U2', 'UA'],
 ['AA', 'CX', 'EK', 'FR', 'HY', 'HZ', 'JL', 'JT', 'KY', 'QF'],
 ['4U', '8L', 'ET', 'FR', 'G4', 'IE', 'OZ', 'SC', 'TK', 'WN'],
 ['7J', 'AA', 'AB', 'DL', 'DL', 'G3', 'TS', 'US', 'US', 'WN']]
```

[ ]: