assignment03_SafsafiAchraf

April 12, 2021

Achraf Safsafi

DSC650

3.1 Assignment

[1]:
```
import os
import sys
import gzip
import json
from pathlib import Path
import csv

import pandas as pd
import s3fs
import pyarrow as pa
from pyarrow.json import read_json
import pyarrow.parquet as pq
import fastavro
import pygeohash
import snappy
import jsonschema
from jsonschema.exceptions import ValidationError


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

current_dir = Path(os.getcwd()).absolute()
schema_dir = current_dir.joinpath('schemas')
schema_dir.mkdir(parents=True, exist_ok=True)
results_dir = current_dir.joinpath('results')
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 = 'data/processed/openflights/routes.jsonl.gz'
    with s3.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

records = read_jsonl_data()
```

[2]:
```
records[0]
```

[2]:
```
{'airline': {'airline_id': 410,
  'name': 'Aerocondor',
  'alias': 'ANA All Nippon Airways',
  'iata': '2B',
  'icao': 'ARD',
  'callsign': 'AEROCONDOR',
  'country': 'Portugal',
  'active': True},
 'src_airport': {'airport_id': 2965,
  'name': 'Sochi International Airport',
  'city': 'Sochi',
  'country': 'Russia',
  'iata': 'AER',
  'icao': 'URSS',
  'latitude': 43.449902,
  'longitude': 39.9566,
  'altitude': 89,
  'timezone': 3.0,
  'dst': 'N',
  'tz_id': 'Europe/Moscow',
  'type': 'airport',
  'source': 'OurAirports'},
 'dst_airport': {'airport_id': 2990,
  'name': 'Kazan International Airport',
  'city': 'Kazan',
  'country': 'Russia',
  'iata': 'KZN',
  'icao': 'UWKD',
  'latitude': 55.606201171875,
  'longitude': 49.278701782227,
  'altitude': 411,
  'timezone': 3.0,
```

```
  'dst': 'N',
  'tz_id': 'Europe/Moscow',
  'type': 'airport',
  'source': 'OurAirports'},
 'codeshare': False,
 'equipment': ['CR2']}
```

3.1

3.1.a JSON Schema

[3]:
```
import requests

def validate_jsonl_data(records):


    schema_path = schema_dir.joinpath("routes-schema.json")
    with open(schema_path) as f:
        validation_csv_path = results_dir.joinpath("validation-results.csv")
        schema = json.load(f)

    with open(validation_csv_path, 'w') as f:
        for i, record in enumerate(records):
            try:
                jsonschema.validate(instance=record,schema= schema)
            except ValidationError as e:
                print('e')
                msg='json routes is invalid'
                return False,msg
            msg='json routes is valid'
            return True,msg



validate_jsonl_data(records)
```

[3]:
```
(True, 'json routes is valid')
```

3.1.b Avro

[4]:
```
import fastavro
from fastavro.schema import load_schema
from fastavro import writer

def create_avro_dataset(records):
    schema_path = schema_dir.joinpath('routes.avsc')
```

```
    data_path = results_dir.joinpath('routes.avro')
    parsed_schema = load_schema(schema_path)
    with open(data_path,'wb') as out:
        writer(out,parsed_schema,records)


create_avro_dataset(records)
```

[5]:
```
from fastavro import reader
with open('routes.avro', 'rb') as fo:
    avro_reader = reader(fo)
    for record in avro_reader:
        pass
print(record)
```

```
{'airline': {'airline_id': 19016, 'name': 'Apache Air', 'alias': 'Apache',
'iata': 'ZM', 'icao': 'IWA', 'callsign': 'APACHE', 'country': 'United States',
'active': True}, 'src_airport': {'airport_id': 2913, 'name': 'Osh Airport',
'city': 'Osh', 'iata': 'OSS', 'icao': 'UAFO', 'latitude': 40.6090011597,
'longitude': 72.793296814, 'timezone': 6.0, 'dst': 'U', 'tz_id': 'Asia/Bishkek',
'type': 'airport', 'source': 'OurAirports'}, 'dst_airport': {'airport_id': 2912,
'name': 'Manas International Airport', 'city': 'Bishkek', 'iata': 'FRU', 'icao':
'UAFM', 'latitude': 43.0612983704, 'longitude': 74.4776000977, 'timezone': 6.0,
'dst': 'U', 'tz_id': 'Asia/Bishkek', 'type': 'airport', 'source':
'OurAirports'}, 'codeshare': False, 'stops': 0, 'equipment': ['734']}
```

3.1.c Parquet

[6]:
```
def create_parquet_dataset():
    src_data_path = 'data/processed/openflights/routes.jsonl.gz'
    parquet_output_path = results_dir.joinpath('routes.parquet')
    s3 = s3fs.S3FileSystem(
        anon=True,
        client_kwargs={
            'endpoint_url': endpoint_url
        }
    )

    with s3.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()]
            df = pd.DataFrame(records)
            table = pa.Table.from_pandas(df)
            pq.write_table(table,parquet_output_path)
            return parquet_output_path

create_parquet_dataset()
```

[6]: PosixPath('/home/jovyan/dsc650/schemas/results/routes.parquet')

3.1.d Protocol Buffers

[7]:
```
sys.path.insert(0, os.path.abspath('routes_pb2'))

import routes_pb2

def _airport_to_proto_obj(airport):
    obj = routes_pb2.Airport()
    if airport is None:
        return None
    if airport.get('airport_id') is None:
        return None

    obj.airport_id = airport.get('airport_id')
    if airport.get('name'):
        obj.name = airport.get('name')
    if airport.get('city'):
        obj.city = airport.get('city')
    if airport.get('iata'):
        obj.iata = airport.get('iata')
    if airport.get('icao'):
        obj.icao = airport.get('icao')
    if airport.get('altitude'):
        obj.altitude = airport.get('altitude')
    if airport.get('timezone'):
        obj.timezone = airport.get('timezone')
    if airport.get('dst'):
        obj.dst = airport.get('dst')
    if airport.get('tz_id'):
        obj.tz_id = airport.get('tz_id')
    if airport.get('type'):
        obj.type = airport.get('type')
    if airport.get('source'):
        obj.source = airport.get('source')
    obj.latitude = airport.get('latitude')
    obj.longitude = airport.get('longitude')

    return obj


def _airline_to_proto_obj(airline):
    obj = routes_pb2.Airline()
    if not airline.get('name'):
        return None
```

```
    if not airline.get('airline_id'):
        return None

    obj.airline_id = airline.get('airline_id')
    obj.name = airline.get('name')

    return obj


def create_protobuf_dataset(records):
    routes = routes_pb2.Routes()
    for record in records:
        route = routes_pb2.Route()
        airline = _airline_to_proto_obj(record.get('airline',{}))
        if airline:
             route.airline.CopyForm(airline)
        src_airport = _airport_to_proto_obj(record.get('src_airport',{}))
        dst_airport = _airport_to_proto_obj(record.get('dst_airport',{}))



        if src_airport:
            src_airport = _airport_to_proto_obj(record.get('src_airport',{}))
            route.src_airport.CopyForm(src_airport)
            routes.route.append(route)

        if dst_airport:
            dst_airport = _airport_to_proto_obj(record.get('dst_airport',{}))
            route.dst_airport.CopyForm(dst_airport)
            routes.route.append(route)



    data_path = results_dir.joinpath('routes.pb')

    with open(data_path, 'wb') as f:
        f.write(routes.SerializeToString())

    compressed_path = results_dir.joinpath('routes.pb.snappy')

    with open(compressed_path, 'wb') as f:
        f.write(snappy.compress(routes.SerializeToString()))

create_protobuf_dataset(records)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
```

```
<ipython-input-7-fdca173c4c46> in <module>
     84         f.write(snappy.compress(routes.SerializeToString()))
     85
---> 86 create_protobuf_dataset(records)

<ipython-input-7-fdca173c4c46> in create_protobuf_dataset(records)
     56         airline = _airline_to_proto_obj(record.get('airline',{}))
     57         if airline:
---> 58             route.airline.CopyForm(airline)
     59         src_airport = _airport_to_proto_obj(record.get('src_airport',{} )
     60         dst_airport = _airport_to_proto_obj(record.get('dst_airport',{} )

AttributeError: CopyForm
```

3.1.e Output Sizes

Format	Uncompressed	gzip	snappy
jsonl			
Avro	18.7 MB		
Parquet	1.88 MB		
Protocol Buffers			

3.2

3.2.a Simple Geohash Index

[8]:
```
df = pd.json_normalize(records)

df = df.rename({'dst_airport.latitude': 'dst_airport_latitude', 'dst_airport.
↪longitude': 'dst_airport_longitude'}, axis=1)
df.head()
```

[8]:
```
   codeshare equipment  airline.airline_id airline.name  \
0      False     [CR2]                 410   Aerocondor
1      False     [CR2]                 410   Aerocondor
2      False     [CR2]                 410   Aerocondor
3      False     [CR2]                 410   Aerocondor
4      False     [CR2]                 410   Aerocondor

            airline.alias airline.iata airline.icao airline.callsign  \
0  ANA All Nippon Airways           2B          ARD       AEROCONDOR
1  ANA All Nippon Airways           2B          ARD       AEROCONDOR
2  ANA All Nippon Airways           2B          ARD       AEROCONDOR
3  ANA All Nippon Airways           2B          ARD       AEROCONDOR
4  ANA All Nippon Airways           2B          ARD       AEROCONDOR
```

```
  airline.country  airline.active  …  dst_airport_latitude  \
0        Portugal            True  …             55.606201
1        Portugal            True  …             55.606201
2        Portugal            True  …             44.225101
3        Portugal            True  …             55.606201
4        Portugal            True  …             55.012600

  dst_airport_longitude dst_airport.altitude dst_airport.timezone  \
0             49.278702                411.0                  3.0
1             49.278702                411.0                  3.0
2             43.081902               1054.0                  3.0
3             49.278702                411.0                  3.0
4             82.650703                365.0                  7.0

  dst_airport.dst dst_airport.tz_id  dst_airport.type  dst_airport.source  \
0               N     Europe/Moscow           airport         OurAirports
1               N     Europe/Moscow           airport         OurAirports
2               N     Europe/Moscow           airport         OurAirports
3               N     Europe/Moscow           airport         OurAirports
4               N  Asia/Krasnoyarsk           airport         OurAirports

   dst_airport  src_airport
0          NaN          NaN
1          NaN          NaN
2          NaN          NaN
3          NaN          NaN
4          NaN          NaN

[5 rows x 40 columns]
```

[9]:
```
dst_airport_latitude = df['dst_airport_latitude']
dst_airport_longitude = df['dst_airport_longitude']

df['geohash'] = df.apply(lambda x: pygeohash.encode(x.dst_airport_latitude,x.
 ↪dst_airport_longitude,precision=5), axis=1)
df.head(5)
```

[9]:
```
   codeshare equipment  airline.airline_id airline.name  \
0      False     [CR2]                 410   Aerocondor
1      False     [CR2]                 410   Aerocondor
2      False     [CR2]                 410   Aerocondor
3      False     [CR2]                 410   Aerocondor
4      False     [CR2]                 410   Aerocondor

            airline.alias airline.iata airline.icao airline.callsign  \
0  ANA All Nippon Airways           2B          ARD       AEROCONDOR
```

```
1  ANA All Nippon Airways           2B          ARD       AEROCONDOR
2  ANA All Nippon Airways           2B          ARD       AEROCONDOR
3  ANA All Nippon Airways           2B          ARD       AEROCONDOR
4  ANA All Nippon Airways           2B          ARD       AEROCONDOR

  airline.country  airline.active  …  dst_airport_longitude  \
0        Portugal            True  …              49.278702
1        Portugal            True  …              49.278702
2        Portugal            True  …              43.081902
3        Portugal            True  …              49.278702
4        Portugal            True  …              82.650703

  dst_airport.altitude dst_airport.timezone dst_airport.dst dst_airport.tz_id  \
0                411.0                 3.0               N     Europe/Moscow
1                411.0                 3.0               N     Europe/Moscow
2               1054.0                 3.0               N     Europe/Moscow
3                411.0                 3.0               N     Europe/Moscow
4                365.0                 7.0               N  Asia/Krasnoyarsk

  dst_airport.type  dst_airport.source  dst_airport  src_airport  geohash
0          airport         OurAirports          NaN          NaN    v1gh3
1          airport         OurAirports          NaN          NaN    v1gh3
2          airport         OurAirports          NaN          NaN    szyes
3          airport         OurAirports          NaN          NaN    v1gh3
4          airport         OurAirports          NaN          NaN    vcfbb

[5 rows x 41 columns]
```

[10]:
```
df['geohash']
```

[10]:
```
0        v1gh3
1        v1gh3
2        szyes
3        v1gh3
4        vcfbb
         …
67658    r1f90
67659    txsuy
67660    ucfgn
67661    tx5z0
67662    txsuy
Name: geohash, Length: 67663, dtype: object
```

[11]:
```
df.to_json(r'/home/jovyan/dsc650/schemas/results/geoindex\geoindex.json')
```

3.2.b Simple Search Feature

[12]:
```
df = pd.json_normalize(records)

df = df.rename({'dst_airport.latitude': 'dst_airport_latitude', 'dst_airport.
 ↪longitude': 'dst_airport_longitude'}, axis=1)  # new method
dst_airport_latitude = df['dst_airport_latitude']
dst_airport_longitude = df['dst_airport_longitude']

df['geohash'] = df.apply(lambda x: pygeohash.encode(x.dst_airport_latitude,x.
 ↪dst_airport_longitude,precision=5), axis=1)
df.head(5)
```

[12]:

	codeshare	equipment	airline.airline_id	airline.name	\
0	False	[CR2]	410	Aerocondor	
1	False	[CR2]	410	Aerocondor	
2	False	[CR2]	410	Aerocondor	
3	False	[CR2]	410	Aerocondor	
4	False	[CR2]	410	Aerocondor	

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

	airline.country	airline.active	…	dst_airport_longitude	\
0	Portugal	True	…	49.278702	
1	Portugal	True	…	49.278702	
2	Portugal	True	…	43.081902	
3	Portugal	True	…	49.278702	
4	Portugal	True	…	82.650703	

	dst_airport.altitude	dst_airport.timezone	dst_airport.dst	dst_airport.tz_id	\
0	411.0	3.0	N	Europe/Moscow	
1	411.0	3.0	N	Europe/Moscow	
2	1054.0	3.0	N	Europe/Moscow	
3	411.0	3.0	N	Europe/Moscow	
4	365.0	7.0	N	Asia/Krasnoyarsk	

	dst_airport.type	dst_airport.source	dst_airport	src_airport	geohash
0	airport	OurAirports	NaN	NaN	v1gh3
1	airport	OurAirports	NaN	NaN	v1gh3
2	airport	OurAirports	NaN	NaN	szyes
3	airport	OurAirports	NaN	NaN	v1gh3
4	airport	OurAirports	NaN	NaN	vcfbb

[5 rows x 41 columns]

the airports which is in a Radius of 600 Km from Bellevue University (Using The Haversine Distance)

[13]:
```
from math import radians, cos, sin, asin, sqrt
def haversine(lon1, lat1, lon2, lat2):
  # convert decimal degrees to radians
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])

  # haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    r = 6371 # Radius of earth in kilometers. Use 3956 for miles
    return c * r
```

[14]:
```
Bellevue_University_long_lat=(-95.91779,41.1499988)
def row_hsign(row):
    return␣
 ↪haversine(*Bellevue_University_long_lat,row['dst_airport_longitude'],row['dst_airport.
 ↪altitude'])

df['distance']=df.apply(row_hsign,axis=1)
df[df['distance']<=600]
```

[14]:

	codeshare	equipment	airline.airline_id	\
1132	False	[SU9]	17885	
1148	False	[320]	17885	
2035	True	[ATR]	1623	
2046	True	[ATR, AT4]	1623	
2135	False	[320]	2850	
…	…	…	…	
64398	False	[73W]	4547	
64474	False	[73W]	4547	
64494	False	[73W, 733, 73C]	4547	
64530	False	[738, 73W, 73H, 73C]	4547	
64552	False	[73W]	4547	

	airline.name	airline.alias	airline.iata	\
1132	Interjet (ABC Aerolineas)	nan	4O	
1148	Interjet (ABC Aerolineas)	nan	4O	
2035	Canadian North	SN Brussels Airlines	5T	
2046	Canadian North	SN Brussels Airlines	5T	

```
2135             IndiGo Airlines      Horizon Airlines           6E
…                           …                     …          …
64398         Southwest Airlines              SkyWork           WN
64474         Southwest Airlines              SkyWork           WN
64494         Southwest Airlines              SkyWork           WN
64530         Southwest Airlines              SkyWork           WN
64552         Southwest Airlines              SkyWork           WN

      airline.icao airline.callsign airline.country  airline.active  …  \
1132           IBS         INTERJET          Mexico            True  …
1148           IBS         INTERJET          Mexico            True  …
2035           MPE          EMPRESS          Canada            True  …
2046           MPE          EMPRESS          Canada            True  …
2135           IGO             IFLY           India            True  …
…            …               …              …                …  …
64398          SWA        SOUTHWEST   United States            True  …
64474          SWA        SOUTHWEST   United States            True  …
64494          SWA        SOUTHWEST   United States            True  …
64530          SWA        SOUTHWEST   United States            True  …
64552          SWA        SOUTHWEST   United States            True  …

       dst_airport.altitude dst_airport.timezone dst_airport.dst  \
1132                   36.0                 -6.0               S
1148                   46.0                 -6.0               S
2035                   40.0                 -6.0               A
2046                   40.0                 -6.0               A
2135                  138.0                  5.5               N
…                     …                    …                 …
64398                  36.0                 -6.0               A
64474                  46.0                 -6.0               A
64494                  46.0                 -6.0               A
64530                  46.0                 -6.0               A
64552                  46.0                 -6.0               A

         dst_airport.tz_id dst_airport.type dst_airport.source  dst_airport  \
1132   America/Mexico_City          airport        OurAirports          NaN
1148   America/Mexico_City          airport        OurAirports          NaN
2035      America/Winnipeg          airport        OurAirports          NaN
2046      America/Winnipeg          airport        OurAirports          NaN
2135         Asia/Calcutta          airport        OurAirports          NaN
…                     …                …                  …          …
64398      America/Chicago          airport        OurAirports          NaN
64474      America/Chicago          airport        OurAirports          NaN
64494      America/Chicago          airport        OurAirports          NaN
64530      America/Chicago          airport        OurAirports          NaN
64552      America/Chicago          airport        OurAirports          NaN
```

```
      src_airport  geohash    distance
1132          NaN    9ghwz  584.308235
1148          NaN    9gjyr  594.200493
2035          NaN    cgn4h  308.131696
2046          NaN    cgn4h  308.131696
2135          NaN    tgt50  172.553604
…             …      …          …
64398         NaN    9udt1  592.180525
64474         NaN    9vk0x  541.740413
64494         NaN    9vk0x  541.740413
64530         NaN    9vk0x  541.740413
64552         NaN    9vk0x  541.740413

[138 rows x 42 columns]
```