

Protocol Buffers

April 10, 2021

1 3.1.d Protocol Buffers

```
[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')
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]: 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.CopyFrom(airline)
        src_airport = _airport_to_proto_obj(record.get('src_airport', {}))

        dst_airport = _airport_to_proto_obj(record.get('dst_airport', {}))
        route.codeshare = record.get('codeshare')

        if src_airport:
            route.src_airport.CopyFrom(src_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)

```

ModuleNotFoundError

Traceback (most recent call last)

<ipython-input-2-5de818a796a3> in <module>

```
1 sys.path.insert(0, os.path.abspath('routes_pb2'))
2
----> 3 import routes_pb2
4
5 def _airport_to_proto_obj(airport):
```

```
ModuleNotFoundError: No module named 'routes_pb2'
```

```
[ ]:
```

```
[ ]:
```