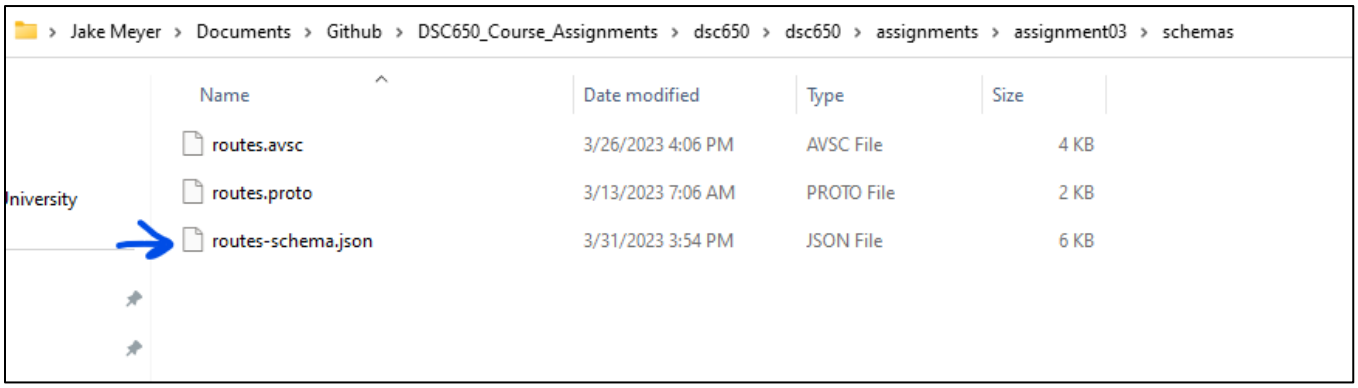
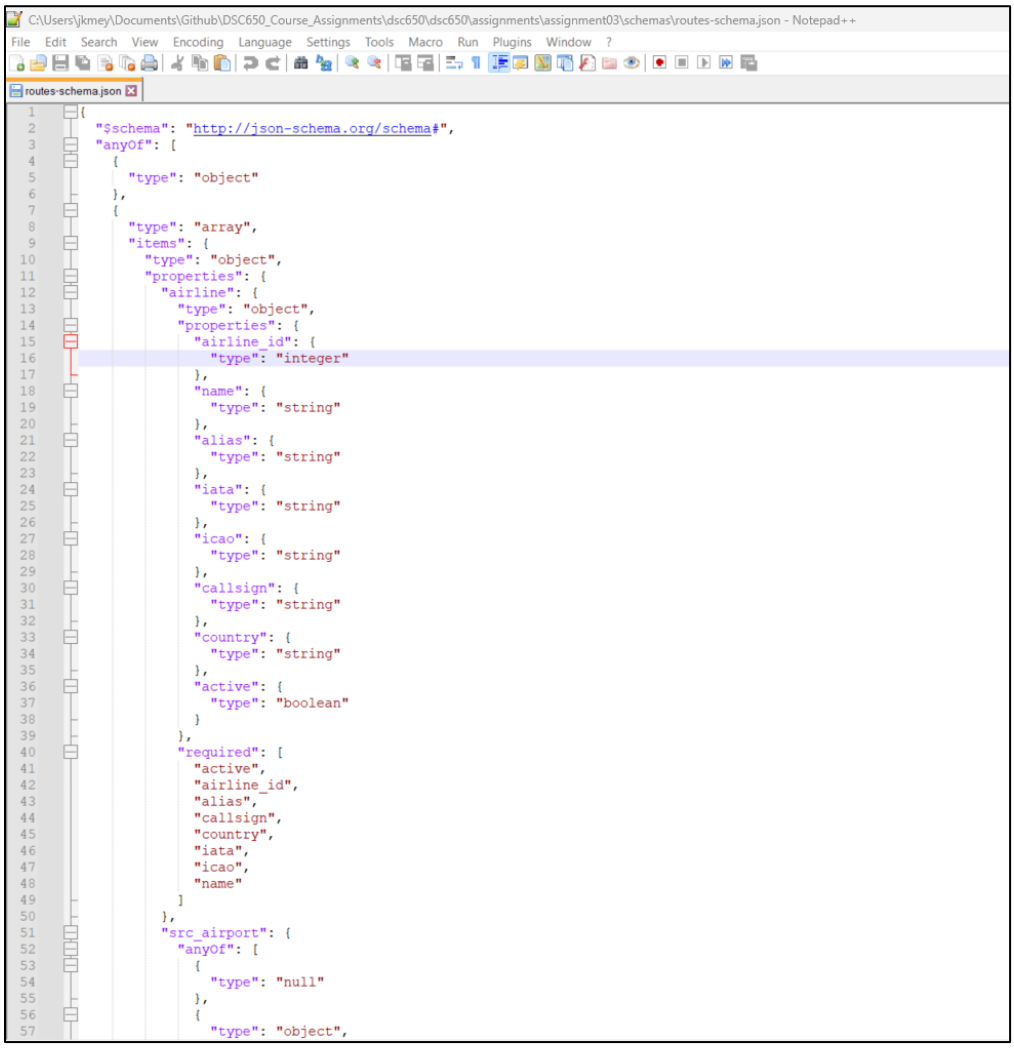
DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 03 Outputs and Code

Assignment 3.1a Code Output (JSON Schema):






```
{
  "$schema": "http://json-schema.org/schema#",
  "anyOf": [
    {
      "type": "object"
    },
    {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "airline": {
            "type": "object",
            "properties": {
              "airline_id": {
                "type": "integer"
              },
              "name": {
                "type": "string"
              },
              "alias": {
                "type": "string"
              },
              "iata": {
                "type": "string"
              },
              "icao": {
                "type": "string"
              },
              "callsign": {
                "type": "string"
              },
              "country": {
                "type": "string"
              },
              "active": {
                "type": "boolean"
              }
            },
            "required": [
              "active",
              "airline_id",
              "alias",
              "callsign",
              "country",
              "iata",
              "icao",
              "name"
            ]
          },
          "src_airport": {
            "anyOf": [
              {
                "type": "null"
              },
              {
                "type": "object",
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

jupyter Assignment 3 Last Checkpoint: 21 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
                f.write(str(e.instance))
                f.write(str(message_detail))
                return message_detail
                ## pass


validate_jsonl_data(records)
```

```
{'$schema': 'http://json-schema.org/schema#', 'anyOf': [{'type': 'object'}, {'type': 'array', 'items': {'type': 'object', 'prop
erties': {'airline': {'type': 'object', 'properties': {'airline_id': {'type': 'integer'}, 'name': {'type': 'string'}, 'alias':
{'type': 'string'}, 'iata': {'type': 'string'}, 'icao': {'type': 'string'}, 'callsign': {'type': 'string'}, 'country': {'type':
'string'}, 'active': {'type': 'boolean'}}, 'required': ['active', 'airline_id', 'alias', 'callsign', 'country', 'iata', 'icao',
'name']}, 'src_airport': {'anyOf': [{'type': 'null'}, {'type': 'object', 'properties': {'airport_id': {'type': 'integer'}, 'nam
e': {'type': 'string'}, 'city': {'type': 'string'}, 'country': {'type': 'string'}, 'iata': {'type': 'string'}, 'icao': {'type':
'string'}, 'latitude': {'type': 'number'}, 'longitude': {'type': 'number'}, 'altitude': {'type': 'integer'}, 'timezone': {'typ
e': 'number'}, 'dst': {'type': 'string'}, 'tz_id': {'type': 'string'}, 'type': {'type': 'string'}, 'source': {'type': 'strin
g'}}, 'required': ['airport_id', 'altitude', 'city', 'country', 'dst', 'iata', 'icao', 'latitude', 'longitude', 'name', 'sourc
e', 'timezone', 'type', 'tz_id']}]}, 'dst_airport': {'anyOf': [{'type': 'null'}, {'type': 'object', 'properties': {'airport_i
d': {'type': 'integer'}, 'name': {'type': 'string'}, 'city': {'type': 'string'}, 'country': {'type': 'string'}, 'iata': {'typ
e': 'string'}, 'icao': {'type': 'string'}, 'latitude': {'type': 'number'}, 'longitude': {'type': 'number'}, 'altitude': {'typ
e': 'integer'}, 'timezone': {'type': 'number'}, 'dst': {'type': 'string'}, 'tz_id': {'type': 'string'}, 'type': {'type': 'strin
g'}, 'source': {'type': 'string'}}, 'required': ['airport_id', 'altitude', 'city', 'country', 'dst', 'iata', 'icao', 'latitud
e', 'longitude', 'name', 'source', 'timezone', 'type', 'tz_id']}]}, 'codeshare': {'type': 'boolean'}, 'equipment': {'type': 'ar
ray', 'items': {'type': 'string'}}}, 'required': ['airline', 'codeshare', 'dst_airport', 'equipment', 'src_airport']}}]}
```

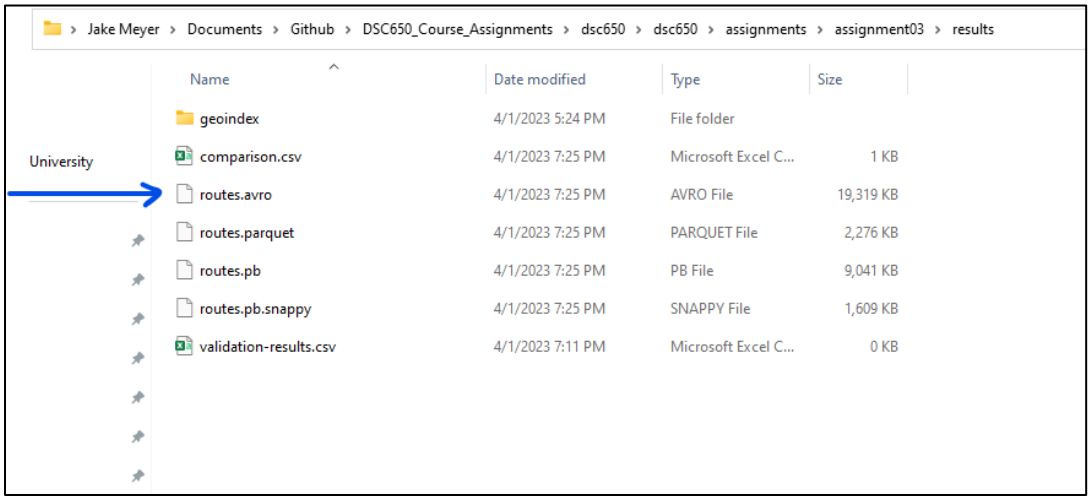
Jake Meyer › Documents › Github › DSC650_Course_Assignments › dsc650 › dsc650 › assignments › assignment03 › results

Name	Date modified	Type	Size
geoindex	4/1/2023 5:24 PM	File folder	
comparison.csv	4/1/2023 7:25 PM	Microsoft Excel C...	1 KB
routes.avro	4/1/2023 7:25 PM	AVRO File	19,319 KB
routes.parquet	4/1/2023 7:25 PM	PARQUET File	2,276 KB
routes.pb	4/1/2023 7:25 PM	PB File	9,041 KB
routes.pb.snappy	4/1/2023 7:25 PM	SNAPPY File	1,609 KB
validation-results.csv	4/1/2023 7:11 PM	Microsoft Excel C...	0 KB

If validation-results has an error, then the message will be displayed in the csv file. Otherwise, it is empty.

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

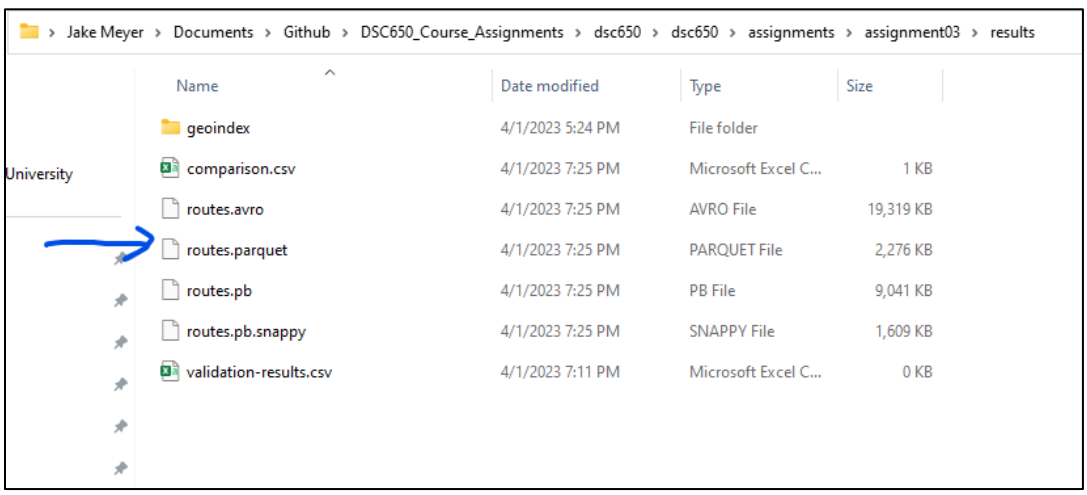
Assignment 3.1b Code Output (Avro):



Jake Meyer › Documents › Github › DSC650_Course_Assignments › dsc650 › dsc650 › assignments › assignment03 › results

Name	Date modified	Type	Size
geoindex	4/1/2023 5:24 PM	File folder	
comparison.csv	4/1/2023 7:25 PM	Microsoft Excel C...	1 KB
routes.avro	4/1/2023 7:25 PM	AVRO File	19,319 KB
routes.parquet	4/1/2023 7:25 PM	PARQUET File	2,276 KB
routes.pb	4/1/2023 7:25 PM	PB File	9,041 KB
routes.pb.snappy	4/1/2023 7:25 PM	SNAPPY File	1,609 KB
validation-results.csv	4/1/2023 7:11 PM	Microsoft Excel C...	0 KB

Assignment 3.1c Code Output (Parquet):



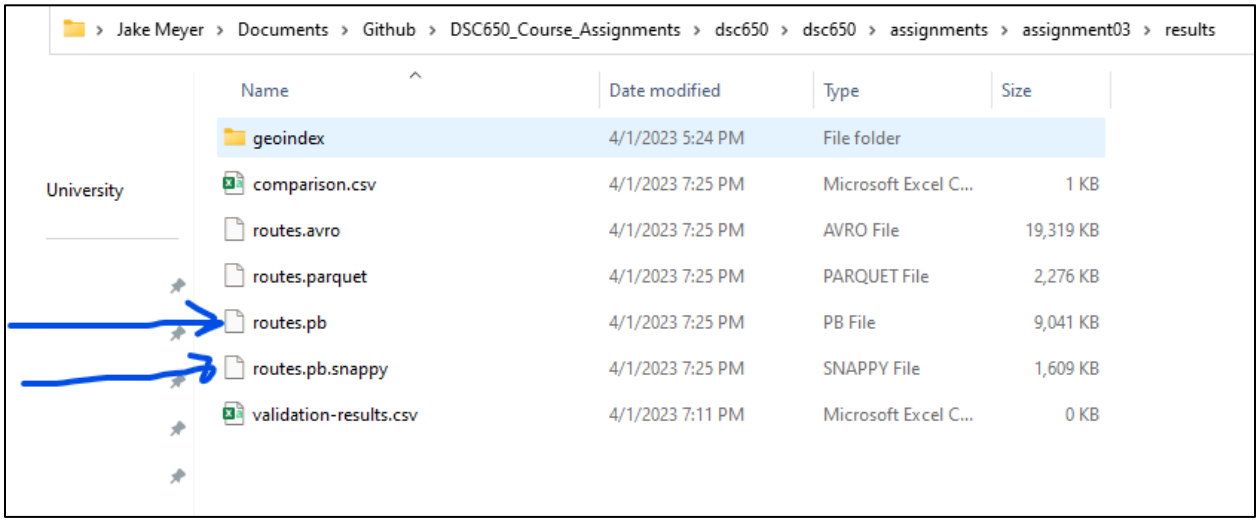
Jake Meyer › Documents › Github › DSC650_Course_Assignments › dsc650 › dsc650 › assignments › assignment03 › results

Name	Date modified	Type	Size
geoindex	4/1/2023 5:24 PM	File folder	
comparison.csv	4/1/2023 7:25 PM	Microsoft Excel C...	1 KB
routes.avro	4/1/2023 7:25 PM	AVRO File	19,319 KB
routes.parquet	4/1/2023 7:25 PM	PARQUET File	2,276 KB
routes.pb	4/1/2023 7:25 PM	PB File	9,041 KB
routes.pb.snappy	4/1/2023 7:25 PM	SNAPPY File	1,609 KB
validation-results.csv	4/1/2023 7:11 PM	Microsoft Excel C...	0 KB

```
pyarrow.Table
airline: struct<active: bool, airline_id: int64, alias: string, callsign: string, country: string, iata: string, icao: strin
g, name: string>
  child 0, active: bool
  child 1, airline_id: int64
  child 2, alias: string
  child 3, callsign: string
  child 4, country: string
  child 5, iata: string
  child 6, icao: string
  child 7, name: string
src_airport: struct<airport_id: int64, altitude: int64, city: string, country: string, dst: string, iata: string, icao: strin
g, latitude: double, longitude: double, name: string, source: string, timezone: double, type: string, tz_id: string>
  child 0, airport_id: int64
  child 1, altitude: int64
  child 2, city: string
  child 3, country: string
  child 4, dst: string
  child 5, iata: string
```

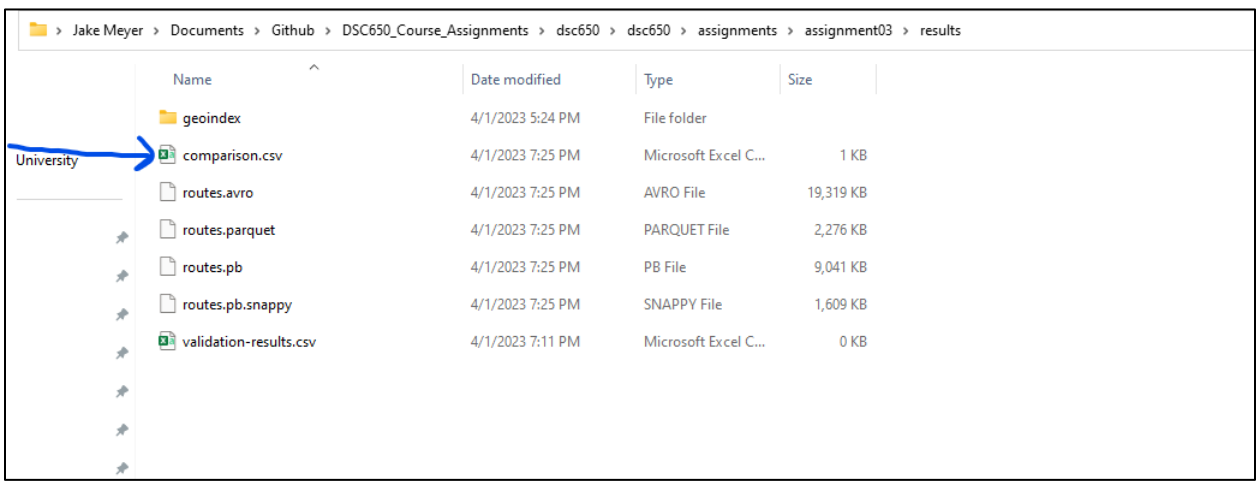
DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.1d Code Output (Protocol Buffers):

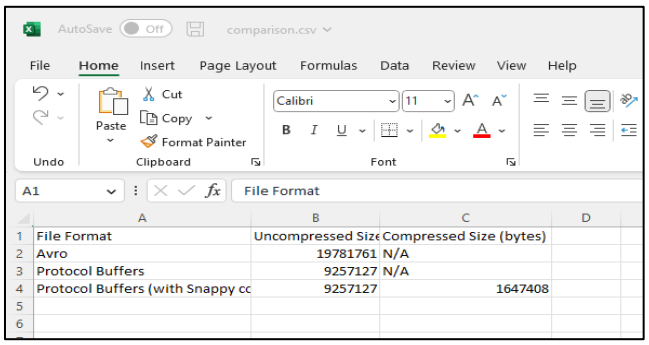


Name	Date modified	Type	Size
geoindex	4/1/2023 5:24 PM	File folder	
comparison.csv	4/1/2023 7:25 PM	Microsoft Excel C...	1 KB
routes.avro	4/1/2023 7:25 PM	AVRO File	19,319 KB
routes.parquet	4/1/2023 7:25 PM	PARQUET File	2,276 KB
routes.pb	4/1/2023 7:25 PM	PB File	9,041 KB
routes.pb.snappy	4/1/2023 7:25 PM	SNAPPY File	1,609 KB
validation-results.csv	4/1/2023 7:11 PM	Microsoft Excel C...	0 KB

Assignment 3.1e Code Output (Output Sizes):



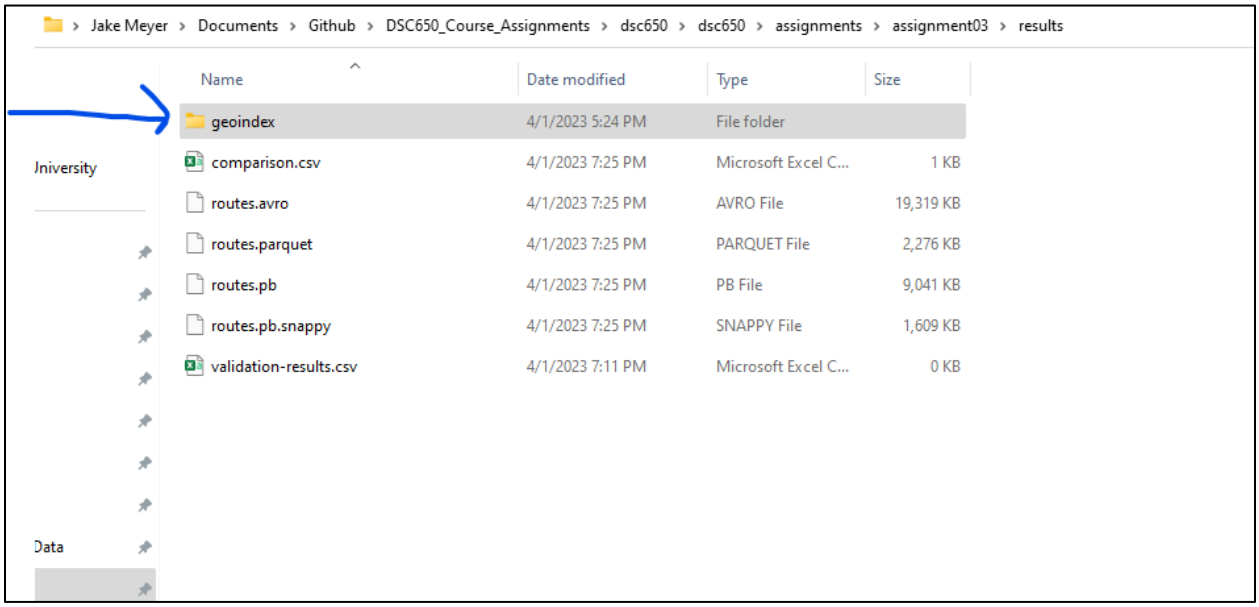
Name	Date modified	Type	Size
geoindex	4/1/2023 5:24 PM	File folder	
comparison.csv	4/1/2023 7:25 PM	Microsoft Excel C...	1 KB
routes.avro	4/1/2023 7:25 PM	AVRO File	19,319 KB
routes.parquet	4/1/2023 7:25 PM	PARQUET File	2,276 KB
routes.pb	4/1/2023 7:25 PM	PB File	9,041 KB
routes.pb.snappy	4/1/2023 7:25 PM	SNAPPY File	1,609 KB
validation-results.csv	4/1/2023 7:11 PM	Microsoft Excel C...	0 KB

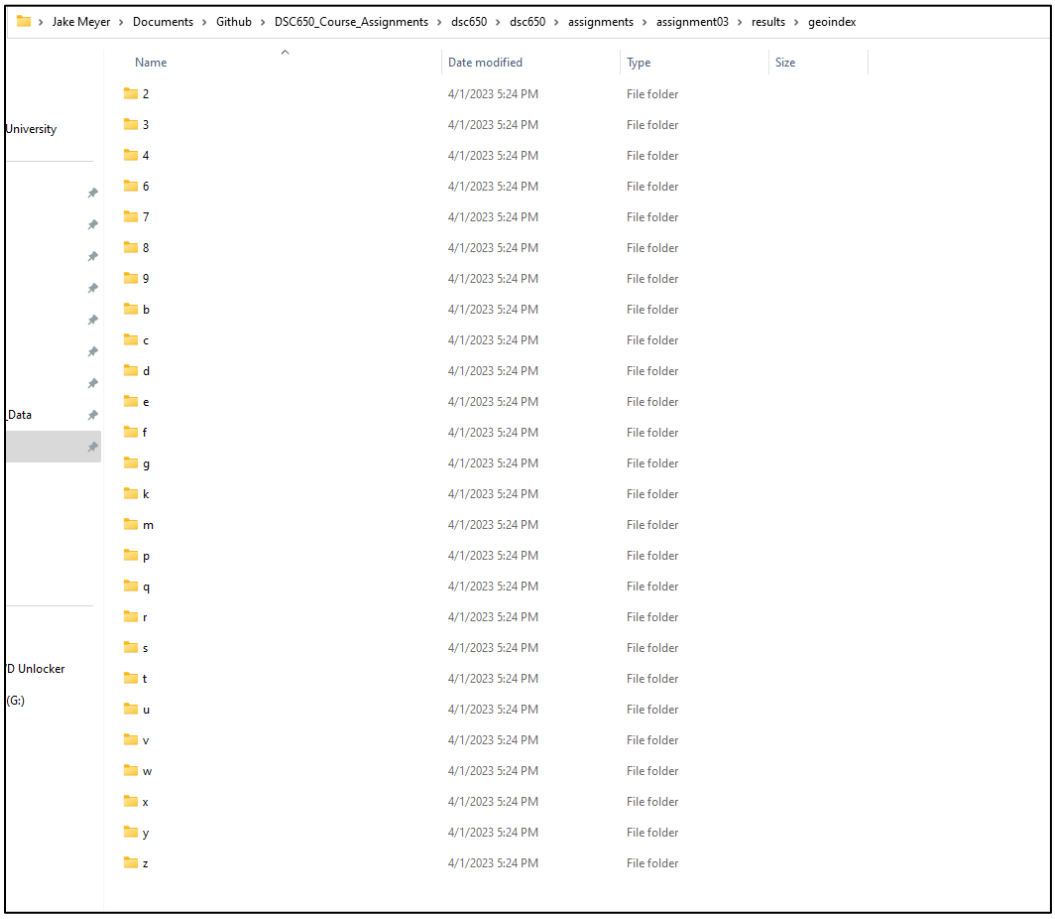


	A	B	C	D
1	File Format	Uncompressed Size	Compressed Size (bytes)	
2	Avro	19781761	N/A	
3	Protocol Buffers	9257127	N/A	
4	Protocol Buffers (with Snappy co	9257127	1647408	
5				
6				

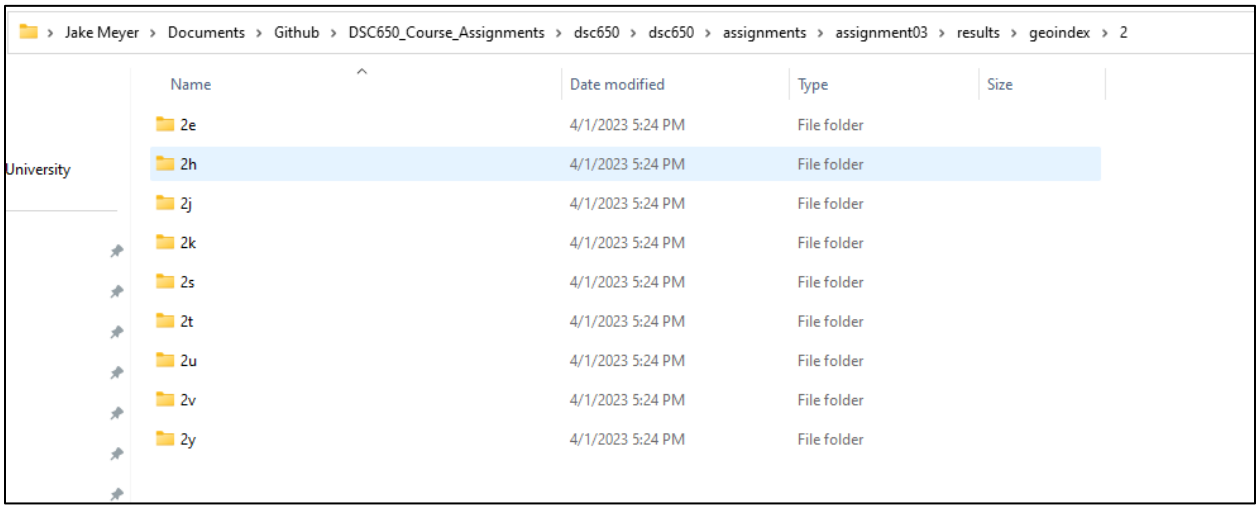
DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

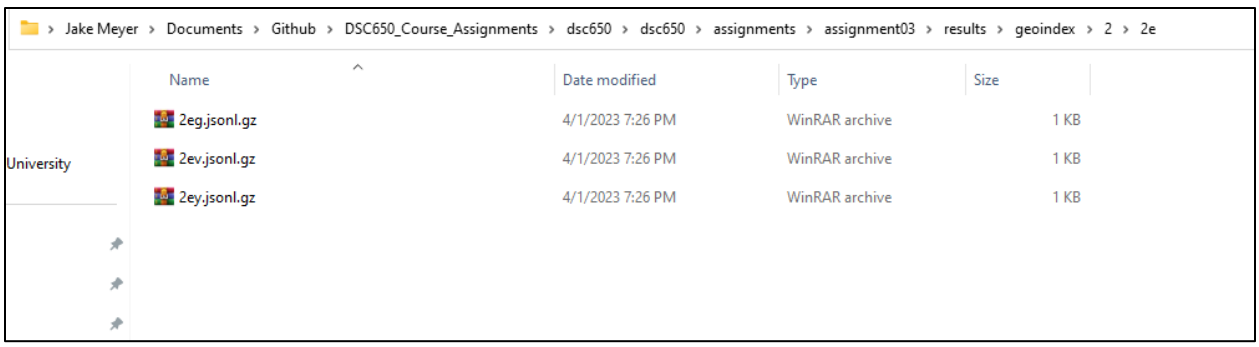
Assignment 3.2a Code Output (Create a Simple Geohash Index):



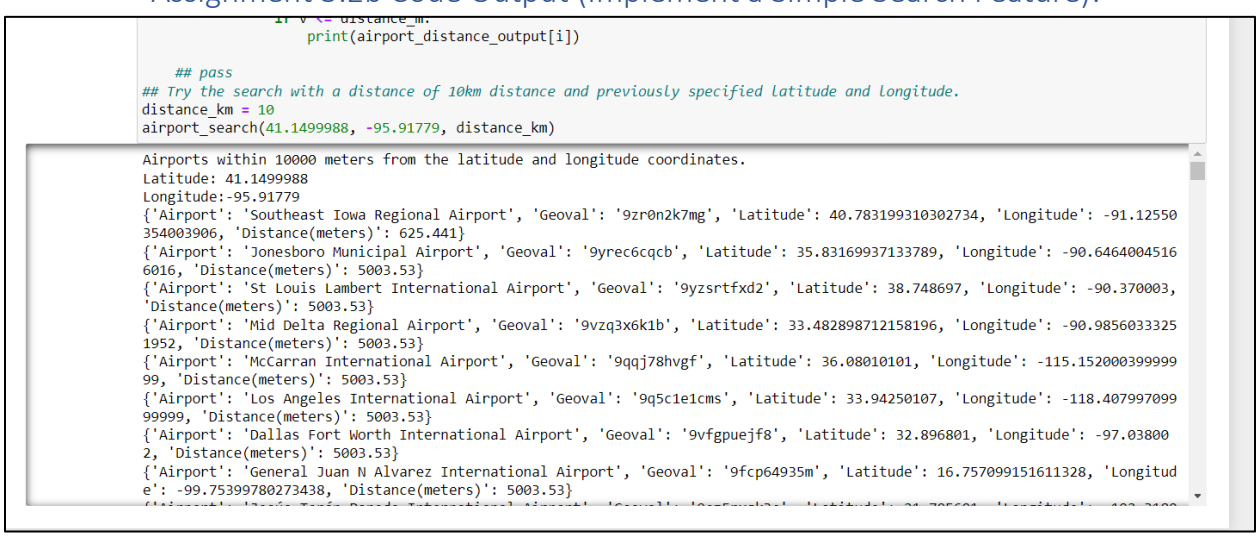


DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023





Assignment 3.2b Code Output (Implement a Simple Search Feature):



```
                print(airport_distance_output[i])

    ## pass
## Try the search with a distance of 10km distance and previously specified latitude and longitude.
distance_km = 10
airport_search(41.1499988, -95.91779, distance_km)
```

```
Airports within 10000 meters from the latitude and longitude coordinates.
Latitude: 41.1499988
Longitude:-95.91779
{'Airport': 'Southeast Iowa Regional Airport', 'Geoval': '9zr0n2k7mg', 'Latitude': 40.783199310302734, 'Longitude': -91.12550
354003906, 'Distance(meters)': 625.441}
{'Airport': 'Jonesboro Municipal Airport', 'Geoval': '9yrec6cqcb', 'Latitude': 35.83169937133789, 'Longitude': -90.6464004516
6016, 'Distance(meters)': 5003.53}
{'Airport': 'St Louis Lambert International Airport', 'Geoval': '9yzsrtfxd2', 'Latitude': 38.748697, 'Longitude': -90.370003,
'Distance(meters)': 5003.53}
{'Airport': 'Mid Delta Regional Airport', 'Geoval': '9vzq3x6k1b', 'Latitude': 33.482898712158196, 'Longitude': -90.9856033325
1952, 'Distance(meters)': 5003.53}
{'Airport': 'McCarran International Airport', 'Geoval': '9qqj78hvgf', 'Latitude': 36.08010101, 'Longitude': -115.152000399999
99, 'Distance(meters)': 5003.53}
{'Airport': 'Los Angeles International Airport', 'Geoval': '9q5c1e1cms', 'Latitude': 33.94250107, 'Longitude': -118.407997099
99999, 'Distance(meters)': 5003.53}
{'Airport': 'Dallas Fort Worth International Airport', 'Geoval': '9vfgpuejf8', 'Latitude': 32.896801, 'Longitude': -97.03800
2, 'Distance(meters)': 5003.53}
{'Airport': 'General Juan N Alvarez International Airport', 'Geoval': '9fcp64935m', 'Latitude': 16.757099151611328, 'Longitud
e': -99.75399780273438, 'Distance(meters)': 5003.53}
```

Example Code Output shown below:

```
Airports within 10000 meters from the latitude and longitude coordinates.
Latitude: 41.1499988
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

```
Longitude:-95.91779
{'Airport': 'Southeast Iowa Regional Airport', 'Geoval': '9zr0n2k7mg', 'La
titude': 40.783199310302734, 'Longitude': -91.12550354003906, 'Distance(me
ters)': 625.441}
{'Airport': 'Jonesboro Municipal Airport', 'Geoval': '9yrec6cqcb', 'Latitu
de': 35.83169937133789, 'Longitude': -90.64640045166016, 'Distance(meters)
': 5003.53}
{'Airport': 'St Louis Lambert International Airport', 'Geoval': '9yzsrtfxd
2', 'Latitude': 38.748697, 'Longitude': -90.370003, 'Distance(meters)': 50
03.53}
{'Airport': 'Mid Delta Regional Airport', 'Geoval': '9vzq3x6k1b', 'Latitud
e': 33.482898712158196, 'Longitude': -90.98560333251952, 'Distance(meters)
': 5003.53}
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.1a Code (JSON Schema):

```
def validate_jsonl_data(records):
    schema_path = schema_dir.joinpath('routes-schema.json')
    with open(schema_path) as f:
        schema = json.load(f)
    print(schema)
    ## Error surfaced stating validation_csv_path not found. Adding path.
    validation_csv_path = results_dir.joinpath('validation-results.csv')
    with open(validation_csv_path, 'w') as f:
        for i, record in enumerate(records):
            try:
                ## Validate record
                jsonschema.validate(record, schema)
                ## No message passed to validation_csv_path if validation successful.
                ## pass
            except ValidationError as e:
                ## Print message if invalid record
                message_detail = e.message
                print(message_detail)
                f.write(str(e.path))
                f.write(str(e.instance))
                f.write(str(message_detail))
                return message_detail
                ## pass
validate_jsonl_data(records)
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.1b Code (Avro):

```
## Import additional modules from fastavro
from fastavro.schema import load_schema
from fastavro import writer, reader, parse_schema

def create_avro_dataset(records):
    schema_path = schema_dir.joinpath('routes.avsc')
    data_path = results_dir.joinpath('routes.avro')

    ## Use fastavro to create Avro dataset
    ## Load the schema from the specified path. See Note at bottom of this cell for replacement of
schema file.
    avro_schema = load_schema(schema_path)
    ## Write the records to routes.avro as specified.
    with open(data_path, 'wb') as output:
        writer(output, avro_schema, records)

create_avro_dataset(records)
## Note: Default was modified from None to null in schema file to eliminate errors. Replaced schema file
with alternative file.
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.1c Code (Parquet):

```
def create_parquet_dataset():
    ## Replaced src_data_path with code Professor provided.
    src_data_path = '../../../data/processed/openflights/routes.jsonl.gz'
    parquet_output_path = results_dir.joinpath('routes.parquet')
    '''
    Commenting out S3 portion of this function and replacing with code Professor provided.
    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:
            pass
    '''
    with gzip.open(src_data_path, 'rb') as f:
        records = [json.loads(line) for line in f.readlines()]
    ## Use Apache Arrow to create Parquet table and save the dataset.
    ## Start by creating a dataframe.
    df_apache = pd.DataFrame(records)
    ## Create a table using pyarrow.
    table_apache = pa.Table.from_pandas(df_apache)
    ## Print out the results of the table.
    print(table_apache)
    ## Write to the routes.parquet file.
    pq.write_table(table_apache, parquet_output_path, compression = 'none')
create_parquet_dataset()
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.1d Code (Protocol Buffers):

```
## Needed to install python-snappy due to error with snappy.compress() showing up.
## Uninstalled snappy and python-snappy. Reinstalled python-snappy.
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
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

```
def _airline_to_proto_obj(airline):
    obj = routes_pb2.Airline()
    ## Create an Airline obj using Protocol Buffers API.
    ## Follow the similar code as specified in _airport_to_proto_obj function, except for ailrine.
    if airline is None:
        return None
    if airline.get('airline_id') is None:
        return None

    obj.airline_id = airline.get('airline_id')
    if airline.get('name'):
        obj.name = airline.get('name')
    if airline.get('alias'):
        obj.alias = airline.get('alias')
    if airline.get('iata'):
        obj.iata = airline.get('iata')
    if airline.get('icao'):
        obj.icao = airline.get('icao')
    if airline.get('callsign'):
        obj.callsign = airline.get('callsign')
    if airline.get('country'):
        obj.country = airline.get('country')
    if airline.get('active'):
        obj.active = airline.get('active')

    return obj


def create_protobuf_dataset(records):
    routes = routes_pb2.Routes()
    for record in records:
        route = routes_pb2.Route()
        ## Implement the code to create the Protocol Buffers Dataset
        ## Utilize for loop to obtain key, value pairs for airline, source airport, and distance airport.
        for key, value in record.items():
            if key== 'airline':
                airline = _airline_to_proto_obj(value)
                air_input = route.airline
                air_input.name = airline.name
                air_input.airline_id = airline.airline_id
                air_input.active = airline.active

            if key == 'src_airport' and value is not None:
                src_airport = _airport_to_proto_obj(value)
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

```
            src_airport_input = route.src_airport
            src_airport_input.name = src_airport.name
            src_airport_input.airport_id = src_airport.airport_id
            src_airport_input.latitude = src_airport.latitude
            src_airport_input.longitude = src_airport.longitude

        if key == 'dst_airport' and value is not None:
            dst_airport = _airport_to_proto_obj(value)
            dst_airport_input = route.dst_airport
            dst_airport_input.name = dst_airport.name
            dst_airport_input.airport_id = dst_airport.airport_id
            dst_airport_input.latitude = dst_airport.latitude
            dst_airport_input.longitude = dst_airport.longitude

        if key =='codeshare':
            route.codeshare = value

    routes.route.append(route)

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

  with open(data_path, 'wb') as f:
    bytes_to_string = routes.SerializeToString()
    f.write(bytes_to_string)
  '''
  Resolved an issue with snappy.compress.
  Reinstalled with python-snappy to resolve.
  '''
  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)
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.1e Code (Output Sizes):

```
'''
Working through Output Sizes of the files in compressed and uncompressed formats.
Used example code provided by classmate in Teams channel as starting point for this code.
Replaced the file paths with my local locations since I wasn't able to finish the assignment within the VM.
'''
import os
import csv

## Function get_file_size to return the file size in bytes.
def get_file_size(file_path):
    """Get the size of a file in bytes"""
    return os.stat(file_path).st_size

## Define the File paths
avro_file =
r"C:\Users\jkmey\Documents\Github\DSC650_Course_Assignments\dsc650\dsc650\assignments\assign
ment03\results\routes.avro"
pb_file =
r"C:\Users\jkmey\Documents\Github\DSC650_Course_Assignments\dsc650\dsc650\assignments\assign
ment03\results\routes.pb"
pb_snappy_file =
r"C:\Users\jkmey\Documents\Github\DSC650_Course_Assignments\dsc650\dsc650\assignments\assign
ment03\results\routes.pb.snappy"
output_file =
r"C:\Users\jkmey\Documents\Github\DSC650_Course_Assignments\dsc650\dsc650\assignments\assign
ment03\results\comparison.csv"

## Get file sizes
avro_size = get_file_size(avro_file)
pb_size = get_file_size(pb_file)
pb_snappy_size = get_file_size(pb_snappy_file)

## Write results to CSV file
with open(output_file, mode='w', newline='') as csv_file:
    fieldnames = ['File Format', 'Uncompressed Size (bytes)', 'Compressed Size (bytes)']
    writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerow({'File Format': 'Avro', 'Uncompressed Size (bytes)': avro_size, 'Compressed Size
(bytes)': 'N/A'})
    writer.writerow({'File Format': 'Protocol Buffers', 'Uncompressed Size (bytes)': pb_size, 'Compressed
Size (bytes)': 'N/A'})
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

```
    writer.writerow({'File Format': 'Protocol Buffers (with Snappy compression)', 'Uncompressed Size
(bytes)': pb_size, 'Compressed Size (bytes)': pb_snappy_size})

## Print the results for confirmation.
print("Comparison results saved to:", output_file)
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.2a Code (Create a Simple Geohash Index):

```
def create_hash_dirs(records):
    geoindex_dir = results_dir.joinpath('geoindex')
    geoindex_dir.mkdir(exist_ok=True, parents=True)
    hashes = []
    ## Create hash index
    ## Iterate through the records with a for loop.
    ## Specify the geohash values to setup the index as specified for the assignment.
    for record in records:
        for key, value in record.items():
            if key == 'src_airport' and value is not None:
                geohash_value = pygeohash.encode(value['latitude'], value['longitude'])
                geohash_value_1 = str(geohash_value)[0]
                geohash_value_2 = str(geohash_value)[0:2]
                geohash_value_3 = str(geohash_value)[0:3]+".jsonl.gz"
                geoindex_dir = results_dir.joinpath('geoindex')
                geoindex_dir_1 = geoindex_dir.joinpath(geohash_value_1)
                geoindex_dir_1.mkdir(parents=True, exist_ok = True)
                geoindex_dir_2 = geoindex_dir_1.joinpath(geohash_value_2)
                geoindex_dir_2.mkdir(parents=True, exist_ok = True)
                jsonfilename = geoindex_dir_2.joinpath(geohash_value_3)
                with gzip.GzipFile(jsonfilename, 'w') as fout:
                    fout.write(json.dumps(value).encode('utf-8'))

create_hash_dirs(records)
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

Assignment 3.2b Code (Implement a Simple Search Feature):

```
## Import unique_everseen, iteration_utilities.
from iteration_utilities import unique_everseen

## Added the distance_km as an argument in the function airport_search.
def airport_search(latitude, longitude, distance_km):
    ## Create simple search to return nearest airport
    ## Modify the input distance in Kilometers to meters (1km = 1000m)
    distance_m = distance_km * 1000
    ## Utilize pygeohash.encode().
    source_geoval = pygeohash.encode(latitude, longitude, precision = 10)
    ## Setup empty lists for the airport_distances and rec_out.
    airport_distances = []
    rec_out = []

    ## Setup for loop for unique source airports lookup
    for record in records:
        for key, value in record.items():
            if key == 'src_airport' and value is not None:
                if value not in rec_out:
                    rec_out.append(value)

    ## Iterate through elmements in rec_out to append to airport_distances list.
    ## Get the name, latitude, and longitude, geohash value, and distance (meters)
    for record in rec_out:
        distance_name = record['name']
        distance_latitude = record['latitude']
        distance_longitude = record['longitude']
        geohash_value = pygeohash.encode(distance_latitude, distance_longitude, precision = 10)
        distancem_distancegeo_1 = pygeohash.geohash_approximate_distance(source_geoval,
geohash_value) / 1000
        airport_dist = {
            "Airport" : distance_name,
            "Geoval" : geohash_value,
            "Latitude" : distance_latitude,
            "Longitude" : distance_longitude,
            "Distance(meters)" : distancem_distancegeo_1
        }
        airport_distances.append(airport_dist)

    airport_distance_output = list(unique_everseen(airport_distances))
    print("Airports within "+str(distance_m)+" meters from the latitude and longitude coordinates.")
    print("Latitude: "+str(latitude))
```

DSC650-T302 Big Data (2235-1)
Professor Iranitalab
Assignment 03 Outputs and Code
Jake Meyer
04/02/2023

```
    print("Longitude:"+str(longitude))
    for i in range(len(airport_distance_output)):
        for k, v in airport_distance_output[i].items():
            if k == 'Distance(meters)':
                if v <= distance_m:
                    print(airport_distance_output[i])

    ## pass
## Try the search with a distance of 10km distance and previously specified latitude and longitude.
distance_km = 10
airport_search(41.1499988, -95.91779, distance_km)
```