Data Processing at Scale

PROJECT 1: NoSQL Report

MARCO ANTONIO SANCHEZ ORTEGA

Lessons Learned:

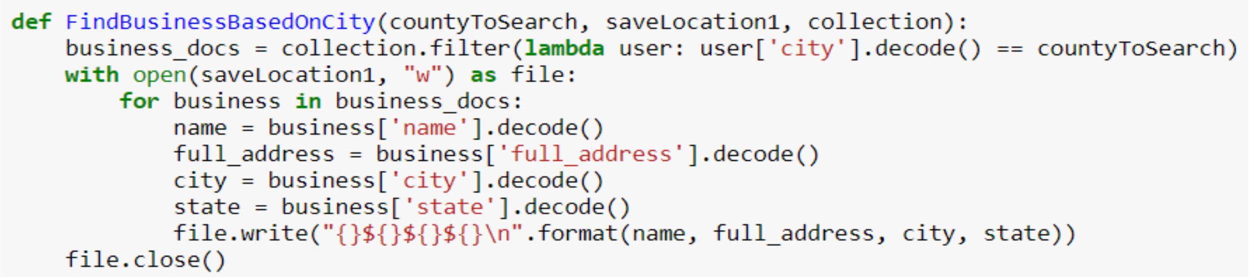
Throughout this project I understood the importance of NoSQL in real problems, specifically I have the opportunity to improve my programming skills in Python.

Reflection:

Python Functions

FindBusinessBasedOnCity Function

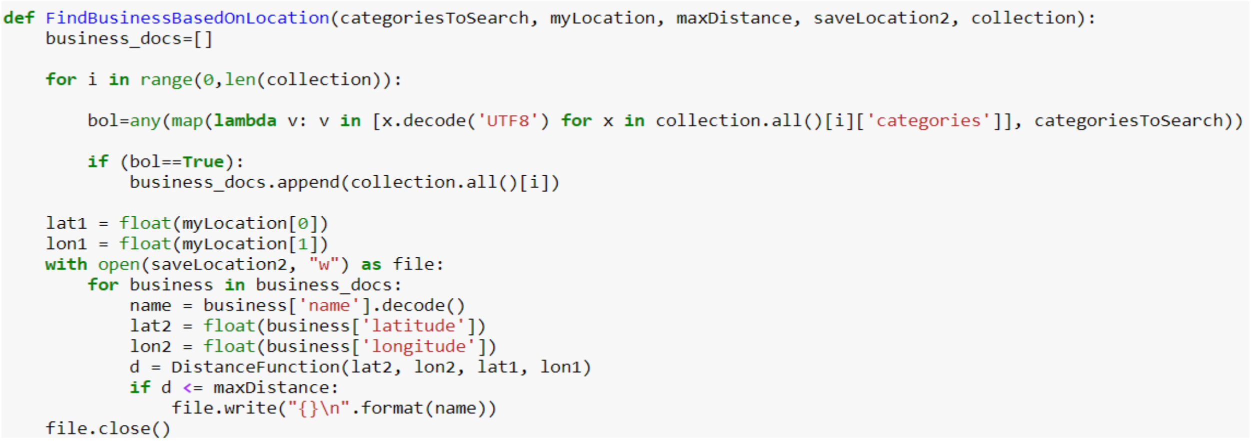
This function searches the 'collection' given to find all the businesses present in the city provided in 'cityToSearch' and saves it to 'saveLocation1'. For each business you found, you should store the name, full address, city, and state of the business in the following format. Each line of the saved file will contain: Name$FullAddress$City$State. ($ is the separator and must be present.)



```python
def FindBusinessBasedOnCity(countyToSearch, saveLocation1, collection):
    business_docs = collection.filter(lambda user: user['city'].decode() == countyToSearch)
    with open(saveLocation1, "w") as file:
        for business in business_docs:
            name = business['name'].decode()
            full_address = business['full_address'].decode()
            city = business['city'].decode()
            state = business['state'].decode()
            file.write("{}${}${}${}\n".format(name, full_address, city, state))
    file.close()
```

FindBusinessBasedOnLocation Function

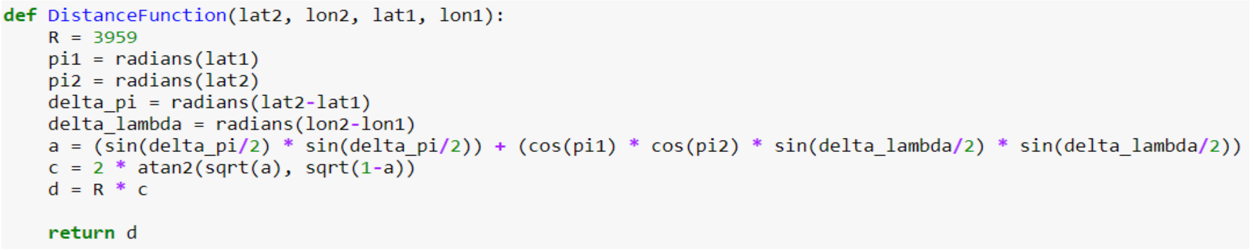
This function searches the 'collection' given to find the name of all the businesses present in the 'maxDistance' from the given 'myLocation' (please use the distance algorithm given below) and saves them to 'saveLocation2'. Each line of the output file will contain the name of the business only.



```
def FindBusinessBasedOnLocation(categoriesToSearch, myLocation, maxDistance, saveLocation2, collection):
    business_docs=[]

    for i in range(0,len(collection)):

        bol=any(map(lambda v: v in [x.decode('UTF8') for x in collection.all()[i]['categories']], categoriesToSearch))

        if (bol==True):
            business_docs.append(collection.all()[i])

    lat1 = float(myLocation[0])
    lon1 = float(myLocation[1])
    with open(saveLocation2, "w") as file:
        for business in business_docs:
            name = business['name'].decode()
            lat2 = float(business['latitude'])
            lon2 = float(business['longitude'])
            d = DistanceFunction(lat2, lon2, lat1, lon1)
            if d <= maxDistance:
                file.write("{}\n".format(name))
    file.close()
```

Distance Function

Given two pairs of latitude and longitude as [lat2, lon2] and [lat1, lon1], you can calculate the distance between them using the formula given below:

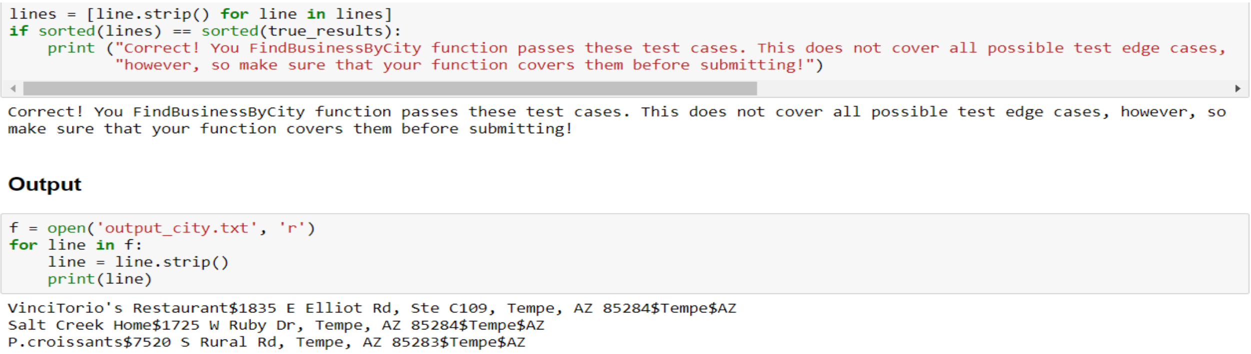


```
def DistanceFunction(lat2, lon2, lat1, lon1):
    R = 3959
    pi1 = radians(lat1)
    pi2 = radians(lat2)
    delta_pi = radians(lat2-lat1)
    delta_lambda = radians(lon2-lon1)
    a = (sin(delta_pi/2) * sin(delta_pi/2)) + (cos(pi1) * cos(pi2) * sin(delta_lambda/2) * sin(delta_lambda/2))
    c = 2 * atan2(sqrt(a), sqrt(1-a))
    d = R * c

    return d
```

Output & Results

Business on Cities Tests

Test 0:



```
lines = [line.strip() for line in lines]
if sorted(lines) == sorted(true_results):
    print ("Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases,
           "however, so make sure that your function covers them before submitting!")
```

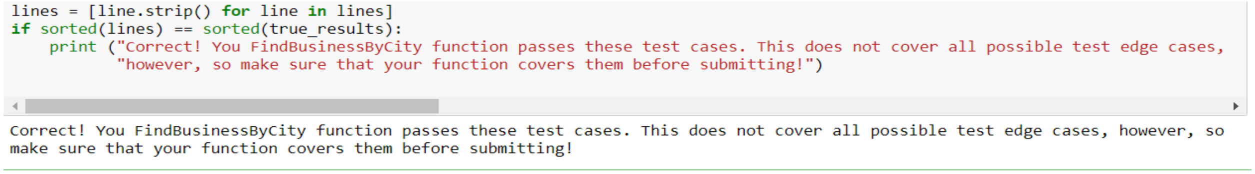
```
Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases, however, so
make sure that your function covers them before submitting!
```

Output

```
f = open('output_city.txt', 'r')
for line in f:
    line = line.strip()
    print(line)
```

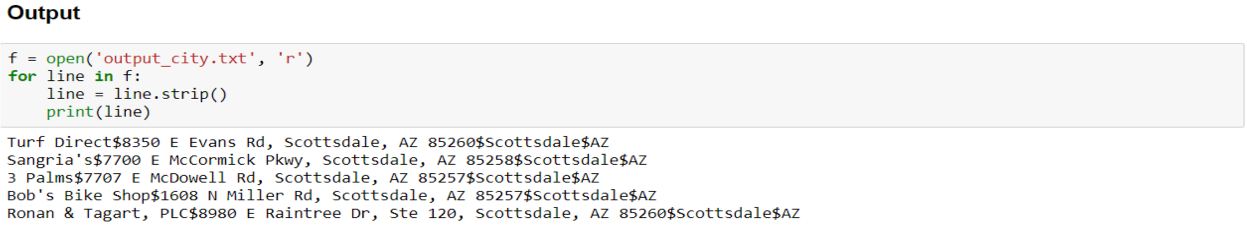
```
VinciTorio's Restaurant$1835 E Elliot Rd, Ste C109, Tempe, AZ 85284$Tempe$AZ
Salt Creek Home$1725 W Ruby Dr, Tempe, AZ 85284$Tempe$AZ
P.croissants$7520 S Rural Rd, Tempe, AZ 85283$Tempe$AZ
```

Test 1:



```
lines = [line.strip() for line in lines]
if sorted(lines) == sorted(true_results):
    print ("Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases,
          "however, so make sure that your function covers them before submitting!")
```

```
Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases, however, so
make sure that your function covers them before submitting!
```

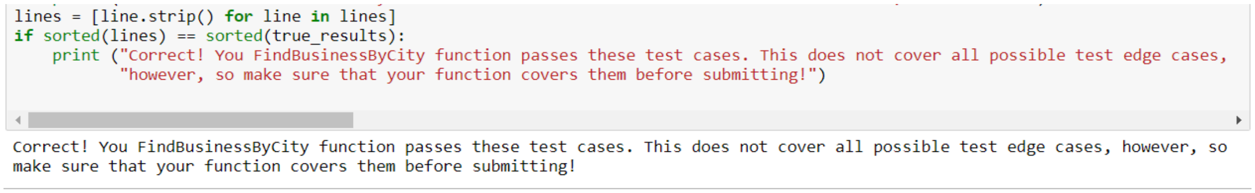


Output

```
f = open('output_city.txt', 'r')
for line in f:
    line = line.strip()
    print(line)
```

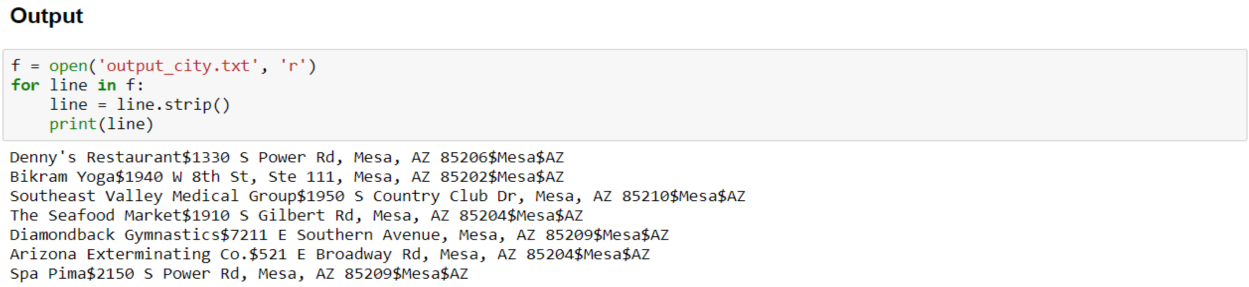
```
Turf Direct$8350 E Evans Rd, Scottsdale, AZ 85260$Scottsdale$AZ
Sangria's$7700 E McCormick Pkwy, Scottsdale, AZ 85258$Scottsdale$AZ
3 Palms$7707 E McDowell Rd, Scottsdale, AZ 85257$Scottsdale$AZ
Bob's Bike Shop$1608 N Miller Rd, Scottsdale, AZ 85257$Scottsdale$AZ
Ronan & Tagart, PLC$8980 E Raintree Dr, Ste 120, Scottsdale, AZ 85260$Scottsdale$AZ
```

Test 2:



```
lines = [line.strip() for line in lines]
if sorted(lines) == sorted(true_results):
    print ("Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases,
          "however, so make sure that your function covers them before submitting!")
```

```
Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases, however, so
make sure that your function covers them before submitting!
```



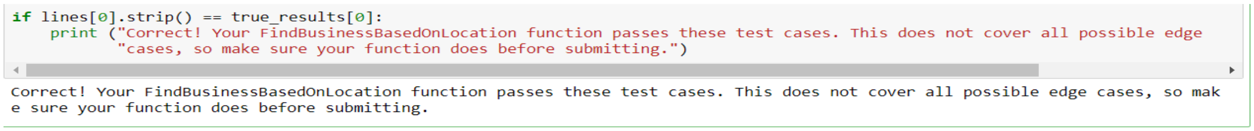
Output

```
f = open('output_city.txt', 'r')
for line in f:
    line = line.strip()
    print(line)
```

```
Denny's Restaurant$1330 S Power Rd, Mesa, AZ 85206$Mesa$AZ
Bikram Yoga$1940 W 8th St, Ste 111, Mesa, AZ 85202$Mesa$AZ
Southeast Valley Medical Group$1950 S Country Club Dr, Mesa, AZ 85210$Mesa$AZ
The Seafood Market$1910 S Gilbert Rd, Mesa, AZ 85204$Mesa$AZ
Diamondback Gymnastics$7211 E Southern Avenue, Mesa, AZ 85209$Mesa$AZ
Arizona Exterminating Co.$521 E Broadway Rd, Mesa, AZ 85204$Mesa$AZ
Spa Pima$2150 S Power Rd, Mesa, AZ 85209$Mesa$AZ
```

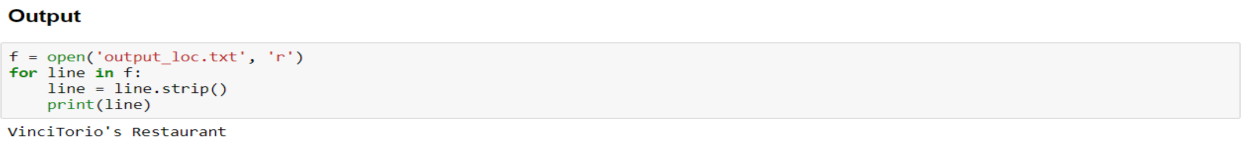
Business on Location Tests

Test 0:



```
if lines[0].strip() == true_results[0]:
    print ("Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge
          "cases, so make sure your function does before submitting.")
```

```
Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge cases, so mak
e sure your function does before submitting.
```

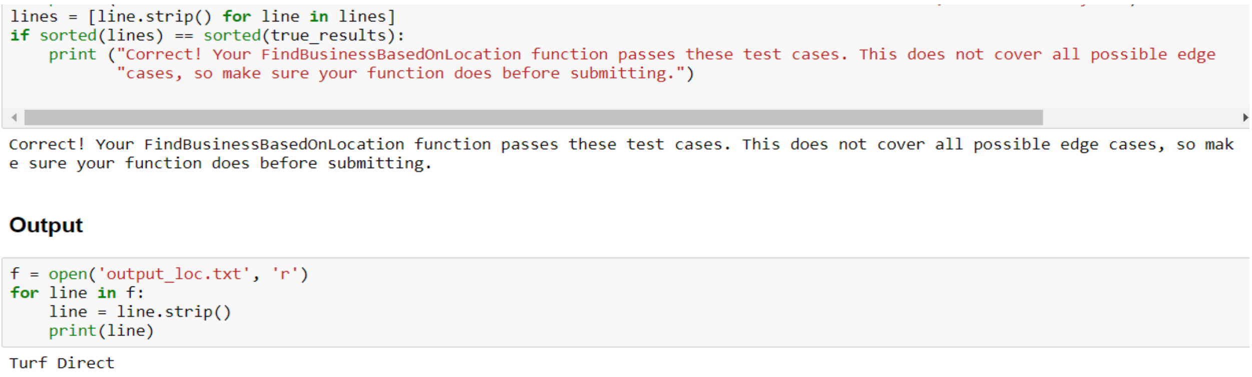


Output

```
f = open('output_loc.txt', 'r')
for line in f:
    line = line.strip()
    print(line)
```

```
VinciTorio's Restaurant
```

Test 1:



```
lines = [line.strip() for line in lines]
if sorted(lines) == sorted(true_results):
    print ("Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge
           "cases, so make sure your function does before submitting.")
```

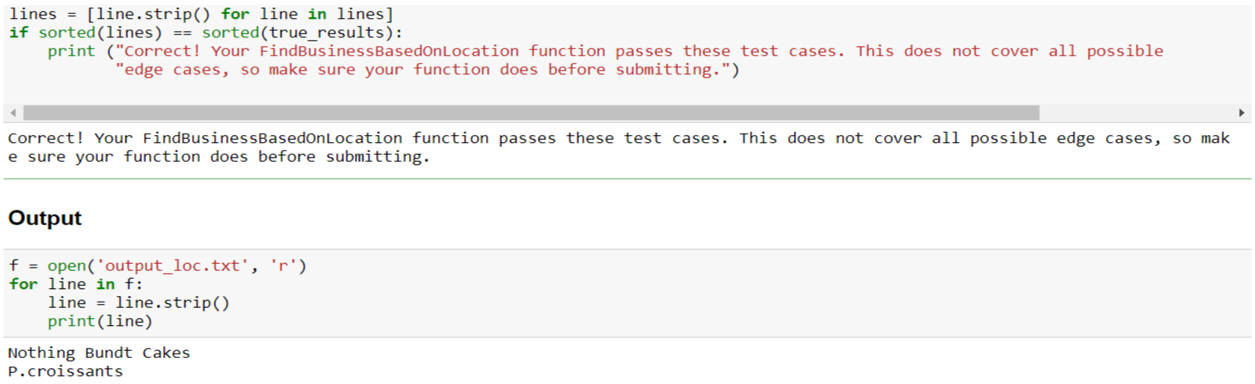
```
Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge cases, so mak
e sure your function does before submitting.
```

Output

```
f = open('output_loc.txt', 'r')
for line in f:
    line = line.strip()
    print(line)
```

```
Turf Direct
```

Test 2:



```
lines = [line.strip() for line in lines]
if sorted(lines) == sorted(true_results):
    print ("Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible
           "edge cases, so make sure your function does before submitting.")
```

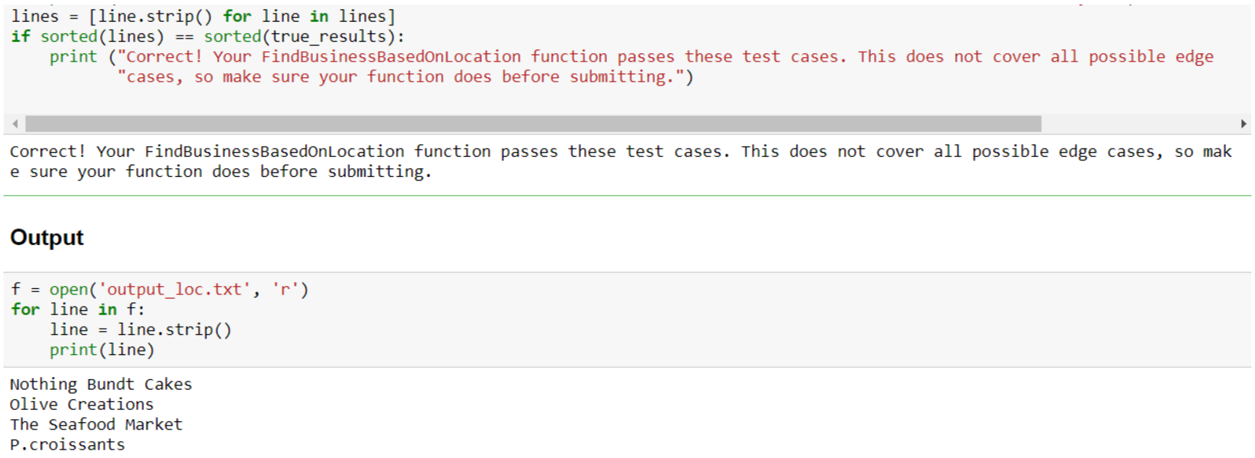
```
Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge cases, so mak
e sure your function does before submitting.
```

Output

```
f = open('output_loc.txt', 'r')
for line in f:
    line = line.strip()
    print(line)
```

```
Nothing Bundt Cakes
P.croissants
```

Test 3:



```
lines = [line.strip() for line in lines]
if sorted(lines) == sorted(true_results):
    print ("Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge
           "cases, so make sure your function does before submitting.")
```

```
Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge cases, so mak
e sure your function does before submitting.
```

Output

```
f = open('output_loc.txt', 'r')
for line in f:
    line = line.strip()
    print(line)
```

```
Nothing Bundt Cakes
Olive Creations
The Seafood Market
P.croissants
```