INFS 3200 : Practice Three

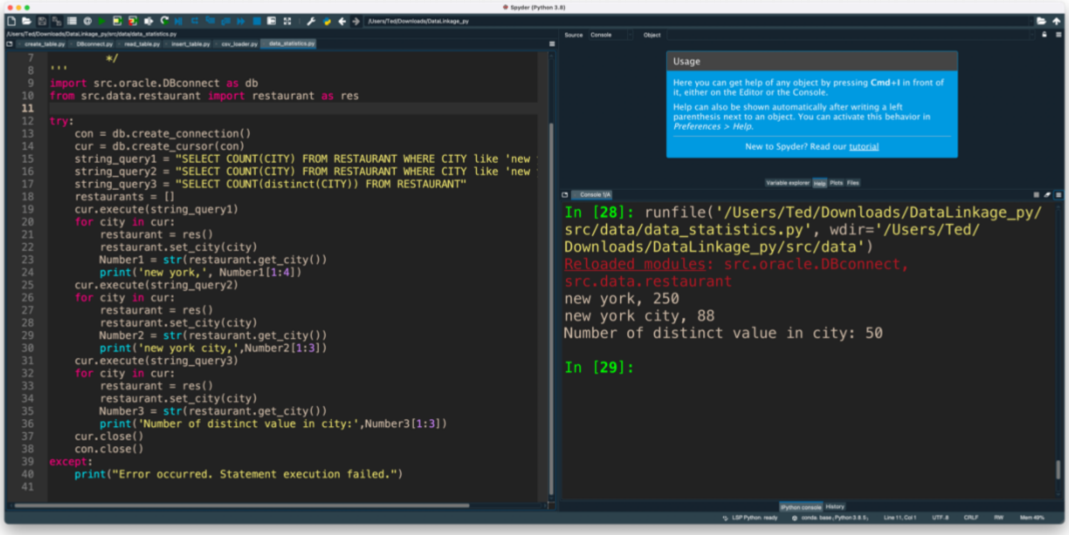
Name : Peng Yu

Student ID : 46635884

Task 1

Answer:

City	Count
“new york”	250
“new york city”	88
Number of distinct value in city	50



```
        */
'''
import src.oracle.DBconnect as db
from src.data.restaurant import restaurant as res

try:
    con = db.create_connection()
    cur = db.create_cursor(con)
    string_query1 = "SELECT COUNT(CITY) FROM RESTAURANT WHERE CITY like 'new
    string_query2 = "SELECT COUNT(CITY) FROM RESTAURANT WHERE CITY like 'new
    string_query3 = "SELECT COUNT(distinct(CITY)) FROM RESTAURANT"
    restaurants = []
    cur.execute(string_query1)
    for city in cur:
        restaurant = res()
        restaurant.set_city(city)
        Number1 = str(restaurant.get_city())
        print('new york,', Number1[1:4])
    cur.execute(string_query2)
    for city in cur:
        restaurant = res()
        restaurant.set_city(city)
        Number2 = str(restaurant.get_city())
        print('new york city,',Number2[1:3])
    cur.execute(string_query3)
    for city in cur:
        restaurant = res()
        restaurant.set_city(city)
        Number3 = str(restaurant.get_city())
        print('Number of distinct value in city:',Number3[1:3])
    cur.close()
    con.close()
except:
    print("Error occurred. Statement execution failed.")
```

Usage

Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
In [28]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/data_statistics.py', wdir='/Users/Ted/
Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.restaurant
new york, 250
new york city, 88
Number of distinct value in city: 50

In [29]:
```

Task 2

Answer (1):

(1) If an instance is a positive class and the connection result is a positive class, it is a true class (True Positive TP)

(2) If an instance is a positive class, but the connection result is a negative class, it is a false negative type (False Negative FN)

(3) If an instance is a negative class, but the connection result is a positive class, it is a false positive class (False Positive FP)

(4) If an instance is a negative class, but the connection result is a negative class, it is a true negative class (True Negative TN)

TP: Number of correct connection results
FN: false negatives, the link result did not find the number of correct matches
FP: False positive, the result of the link is incorrect/mismatch
TN: Number of non--links correctly rejected

Answer (2):

Precision: The formula of precision is P = TP / (TP + FP), which calculates that all "correctly retrieved items (TP)" account for all "actually retrieved items (TP + FP)" The ratio. In this case, it represents the ratio of the correct and misjudged restaurants that are correctly retrieved. Simply put, precision means how correct the link is matched. which can be known from the code, and the calculation method is precision = count / len(results)-- > Total length of matching results.

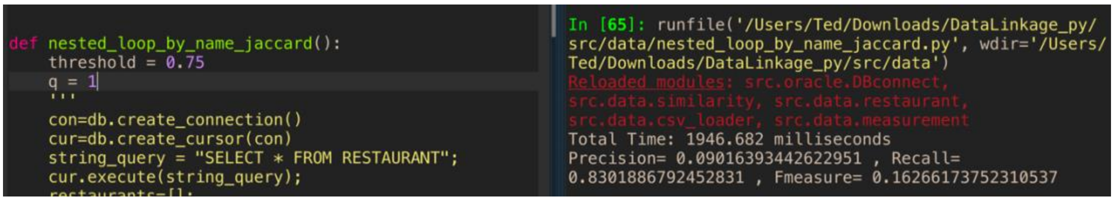
Recall: The formula for the recall is R = TP / (TP + FN), which calculates that all "correctly retrieved items (TP)" account for all "items that should be retrieved (TP+FN)" proportion. Simply put, Recall means how comprehensive the matching link is. In this case, it represents the proportion of restaurants that want to be retrieved correctly in all restaurants that are retrieved correctly.

Task 3

The default value of q and threshold is 3 and 0.75.
First I choice 5 values of q which is 0 1 2 3 4, keep threshold as 0.75 and the result show below:

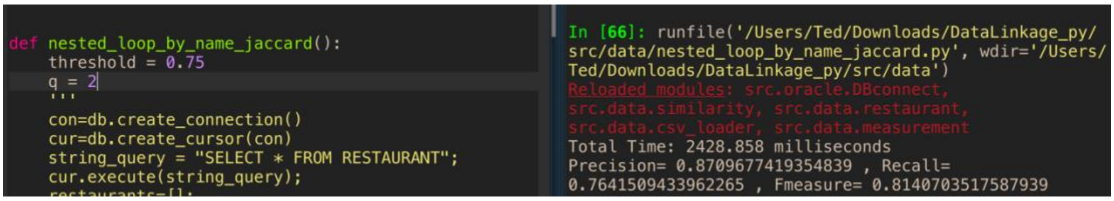
q	Precision	Recall	Fmeasure
1	0.0901639344262295	0.8301886792452831	0.1622661737523105
2	0.8709677419354839	0.7641509433962265	0.8140703517587939
3	0.9069767441860465	0.7358490566037735	0.8124999999999999
4	0.9058823529411765	0.7264150943396226	0.8062827225130891
5	0.9058823529411765	0.7264150943396226	0.8062827225130891

As q increases, the Precision gradually increases, reaches a maximum when q=3, and then gradually decreases as q increases again. When q increases, the Recall decreases. When q = 2, Fmeasure has the best result.



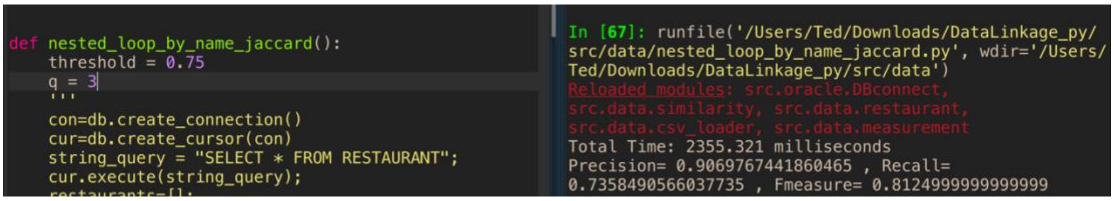
```
def nested_loop_by_name_jaccard():
    threshold = 0.75
    q = 1
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
```

```
In [65]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 1946.682 milliseconds
Precision= 0.09016393442622951 , Recall=
0.8301886792452831 , Fmeasure= 0.16266173752310537
```



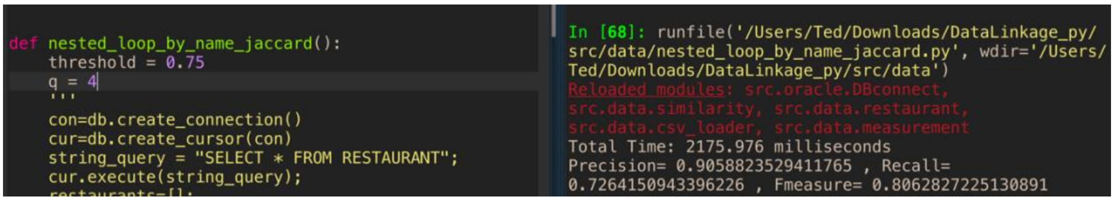
```
def nested_loop_by_name_jaccard():
    threshold = 0.75
    q = 2
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
```

```
In [66]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2428.858 milliseconds
Precision= 0.8709677419354839 , Recall=
0.7641509433962265 , Fmeasure= 0.8140703517587939
```



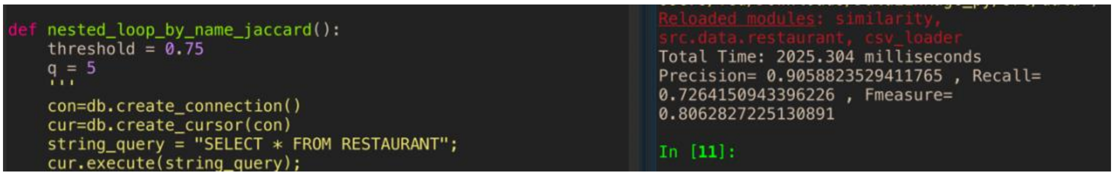
```
def nested_loop_by_name_jaccard():
    threshold = 0.75
    q = 3
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
```

```
In [67]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2355.321 milliseconds
Precision= 0.9069767441860465 , Recall=
0.7358490566037735 , Fmeasure= 0.8124999999999999
```



```
def nested_loop_by_name_jaccard():
    threshold = 0.75
    q = 4
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
```

```
In [68]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2175.976 milliseconds
Precision= 0.9058823529411765 , Recall=
0.7264150943396226 , Fmeasure= 0.8062827225130891
```



```
def nested_loop_by_name_jaccard():
    threshold = 0.75
    q = 5
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
```

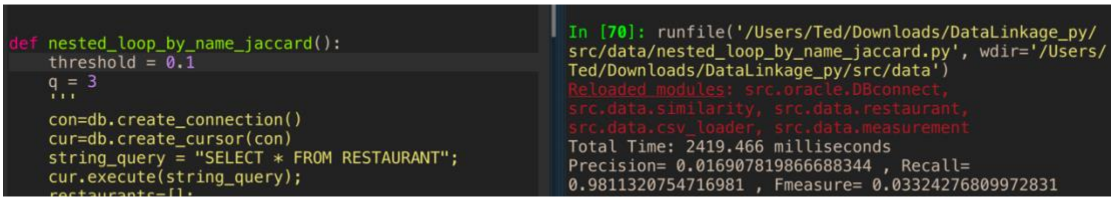
```
Reloaded modules: similarity,
src.data.restaurant, csv_loader
Total Time: 2025.304 milliseconds
Precision= 0.9058823529411765 , Recall=
0.7264150943396226 , Fmeasure=
0.8062827225130891

In [11]:
```

Second I keep q value as 3, and change five threshold value which is : 0.10, 0.25, 0.50, 0.70, 0.90 and the result show below:

threshold	Precision	Recall	Fmeasure
0.10	0.01690781986668834	0.981132075471698	0.0332427680997283
0.25	0.11314285714285714	0.933962264150943	0.2018348623853211
0.50	0.6541353383458647	0.820754716981132	0.7280334728033473
0.70	0.8764044943820225	0.735849056603773	0.8
0.90	0.9156626506024096	0.716981132075471	0.8042328042328042

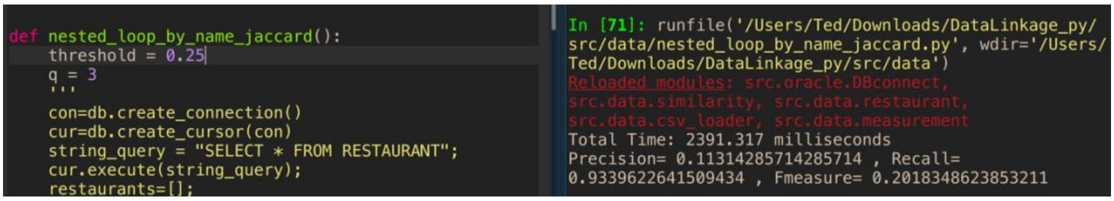
As the threshold increases, Precision increases, Recall decreases. When threshold = 0.90, Fmeasure has the best result.

In summary, when threshold = 0.90 and q = 2, the predicted result is the best.



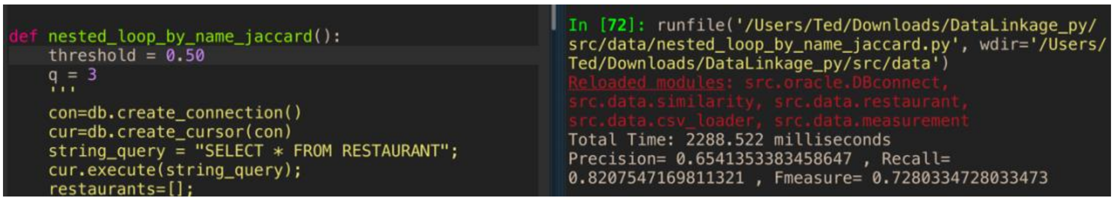
```
def nested_loop_by_name_jaccard():
    threshold = 0.1
    q = 3
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
    restaurants=[];
```

```
In [70]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2419.466 milliseconds
Precision= 0.016907819866688344 , Recall=
0.9811320754716981 , Fmeasure= 0.03324276809972831
```



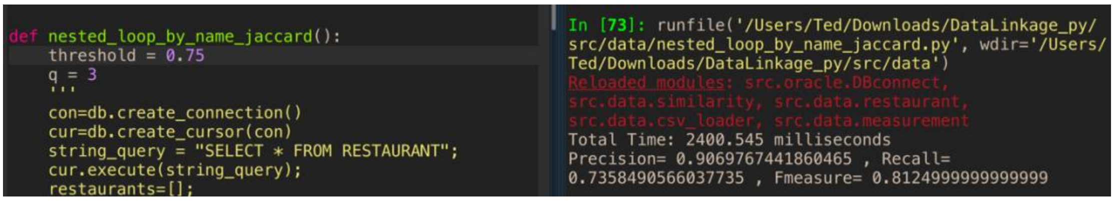
```
def nested_loop_by_name_jaccard():
    threshold = 0.25
    q = 3
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
    restaurants=[];
```

```
In [71]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2391.317 milliseconds
Precision= 0.11314285714285714 , Recall=
0.9339622641509434 , Fmeasure= 0.2018348623853211
```



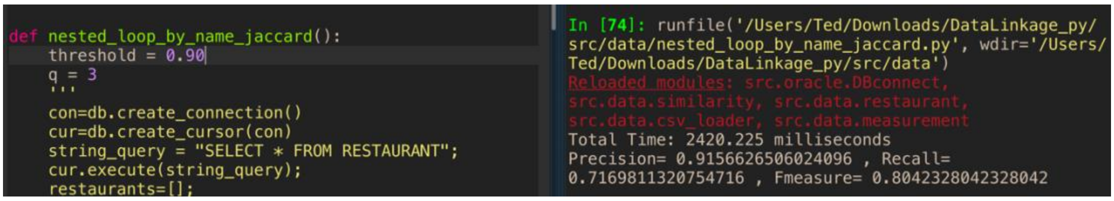
```
def nested_loop_by_name_jaccard():
    threshold = 0.50
    q = 3
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
    restaurants=[];
```

```
In [72]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2288.522 milliseconds
Precision= 0.6541353383458647 , Recall=
0.8207547169811321 , Fmeasure= 0.7280334728033473
```



```
def nested_loop_by_name_jaccard():
    threshold = 0.75
    q = 3
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
    restaurants=[];
```

```
In [73]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2400.545 milliseconds
Precision= 0.9069767441860465 , Recall=
0.7358490566037735 , Fmeasure= 0.8124999999999999
```

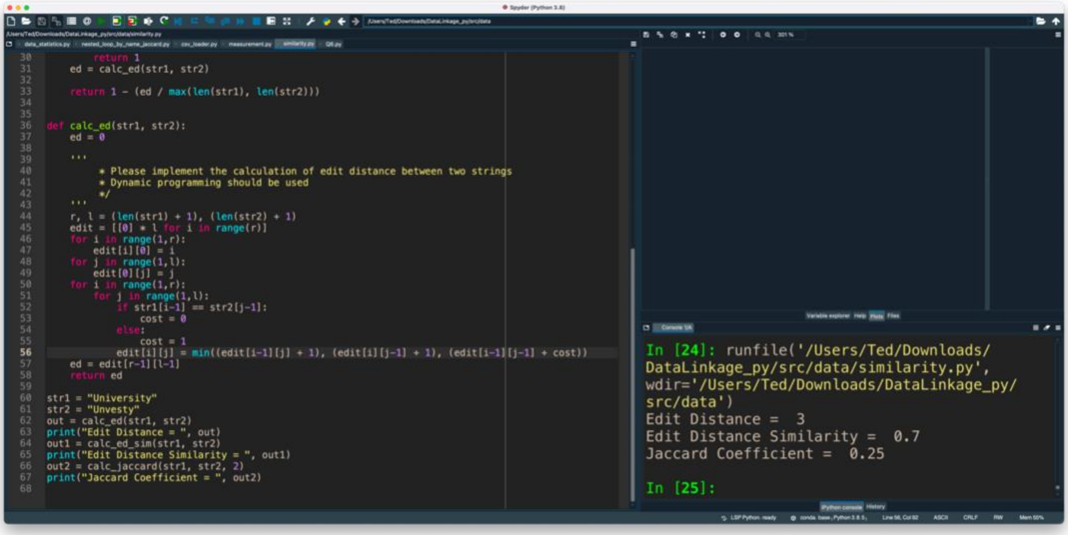


```
def nested_loop_by_name_jaccard():
    threshold = 0.90
    q = 3
    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
    string_query = "SELECT * FROM RESTAURANT";
    cur.execute(string_query);
    restaurants=[];
```

```
In [74]: runfile('/Users/Ted/Downloads/DataLinkage_py/
src/data/nested_loop_by_name_jaccard.py', wdir='/Users/
Ted/Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 2420.225 milliseconds
Precision= 0.9156626506024096 , Recall=
0.7169811320754716 , Fmeasure= 0.8042328042328042
```

Task 4

Edit Distance	3
Edit Distance Similarity	0.7
Jaccard Coefficient	0.25



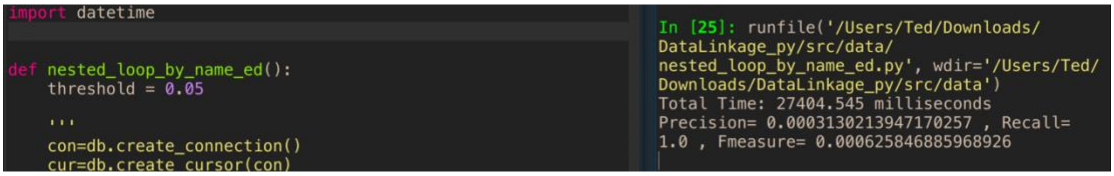
```
            return 1
    ed = calc_ed(str1, str2)

    return 1 - (ed / max(len(str1), len(str2)))


def calc_ed(str1, str2):
    ed = 0

    '''
        * Please implement the calculation of edit distance between two strings
        * Dynamic programming should be used
        */
    '''
    r, l = (len(str1) + 1), (len(str2) + 1)
    edit = [[0] * l for i in range(r)]
    for i in range(1,r):
        edit[i][0] = i
    for j in range(1,l):
        edit[0][j] = j
    for i in range(1,r):
        for j in range(1,l):
            if str1[i-1] == str2[j-1]:
                cost = 0
            else:
                cost = 1
            edit[i][j] = min((edit[i-1][j] + 1), (edit[i][j-1] + 1), (edit[i-1][j-1] + cost))
    ed = edit[r-1][l-1]
    return ed

str1 = "University"
str2 = "Unvesty"
out = calc_ed(str1, str2)
print("Edit Distance = ", out)
out1 = calc_ed_sim(str1, str2)
print("Edit Distance Similarity = ", out1)
out2 = calc_jaccard(str1, str2, 2)
print("Jaccard Coefficient = ", out2)
```

```
In [24]: runfile('/Users/Ted/Downloads/
DataLinkage_py/src/data/similarity.py',
wdir='/Users/Ted/Downloads/DataLinkage_py/
src/data')
Edit Distance =  3
Edit Distance Similarity =  0.7
Jaccard Coefficient =  0.25

In [25]:
```

I change five threshold value which is : 0.05, 0.20, 0.45, 0.75, 0.90 and the result show below:

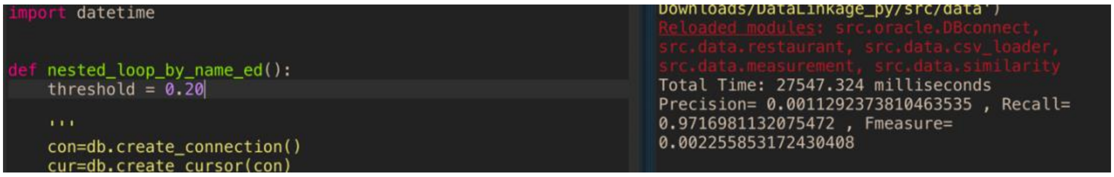
threshold	Precision	Recall	Fmeasure
0.05	0.00031302139471	1.0	0.00062584688596
0.20	0.00112923738104	0.9716981132075	0.00225585317243
0.45	0.04759441282979	0.86792452830188	0.09024031387935
0.75	0.72477064220183	0.74528301886792	0.73488372093023
0.90	0.89534883720930	0.72641509433962	0.80208333333333

As the threshold increases, the Precision increases, Recall decreases. When threshold = 0.90, Fmeasure has the best result.



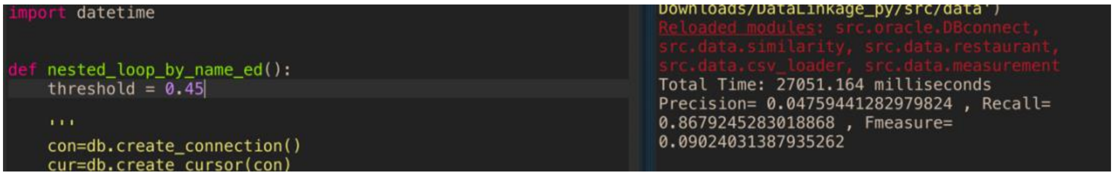
```
import datetime


def nested_loop_by_name_ed():
    threshold = 0.05

    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
```

```
In [25]: runfile('/Users/Ted/Downloads/
DataLinkage_py/src/data/
nested_loop_by_name_ed.py', wdir='/Users/Ted/
Downloads/DataLinkage_py/src/data')
Total Time: 27404.545 milliseconds
Precision= 0.0003130213947170257 , Recall=
1.0 , Fmeasure= 0.0006258468859 68926
```



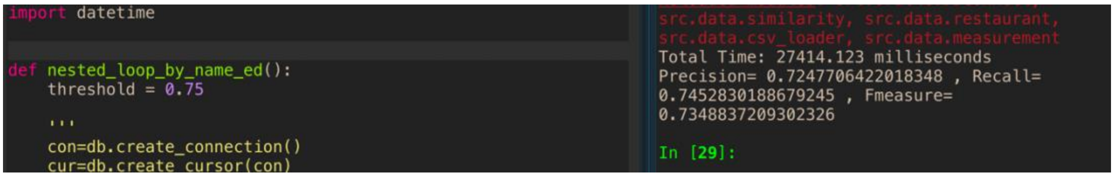
```
import datetime


def nested_loop_by_name_ed():
    threshold = 0.20

    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
```

```
Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.restaurant, src.data.csv_loader,
src.data.measurement, src.data.similarity
Total Time: 27547.324 milliseconds
Precision= 0.0011292373810463535 , Recall=
0.9716981132075472 , Fmeasure=
0.002255853172430408
```



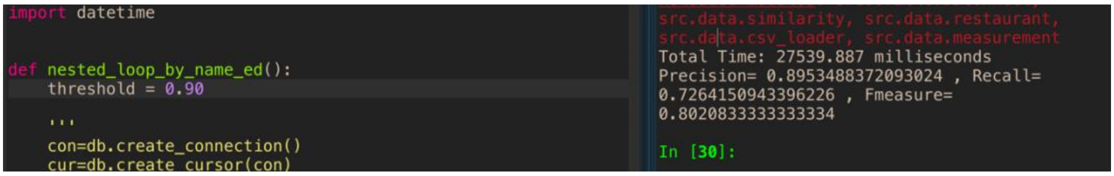
```
import datetime


def nested_loop_by_name_ed():
    threshold = 0.45

    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
```

```
Downloads/DataLinkage_py/src/data')
Reloaded modules: src.oracle.DBconnect,
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 27051.164 milliseconds
Precision= 0.047594412829798 24 , Recall=
0.8679245283018868 , Fmeasure=
0.09024031387935262
```



```
import datetime


def nested_loop_by_name_ed():
    threshold = 0.75

    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
```

```
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 27414.123 milliseconds
Precision= 0.7247706422018348 , Recall=
0.7452830188679245 , Fmeasure=
0.7348837209302326

In [29]:
```



```
import datetime


def nested_loop_by_name_ed():
    threshold = 0.90

    '''
    con=db.create_connection()
    cur=db.create_cursor(con)
```

```
src.data.similarity, src.data.restaurant,
src.data.csv_loader, src.data.measurement
Total Time: 27539.887 milliseconds
Precision= 0.8953488372093024 , Recall=
0.7264150943396226 , Fmeasure=
0.8020833333333334

In [30]:
```