Stage IV Report

Chang Guo (cguo42@wisc.edu)
Yuncong Hao(hyuncong@wisc.edu)
Qun Zou(qzou5@wisc.edu)

- **How did you combine the two tables A and B to obtain E? Did you add any other table? When you did the combination, did you run into any issues? Discuss the combination process in detail, e.g., when you merge tuples, what are the merging functions (such as to merge two age values, always select the age value from the tuple from Table A, unless this value is missing in which case we select the value from the tuple in Table B).**

  We ran the classifier got from last stage on total candidate set to get matching pairs. Then we extracted tuples from original data set according to the pairing ID to get table A and B. Then we combined the two tables (no extra tables) using Python script. We merged the common attributes from two tables together, including shop name, shop address, postal code and phone number and kept the prices, ratings from two tables as different attributes. So, for merged attributes, we just picked the values from relatively more complete table, named table A. And also we've found in actual case, when value in A is missing, it is also missing in B, so we left the missing value as blank. For separate attributes (e.g. price, ratings), we just save them as A_price and B_price separately.

- **Statistics on Table E: specifically, what is the schema of Table E, how many tuples are in Table E? Give at least four sample tuples from Table E.**

  There are 1360 tuples in Table E.

ID	Shop Name	Postal Code	Phone Number	Street Address	Yelp Price	Yelp Number of Reviews	Trip-Advisor Star	Trip-Advisor Price	Trip-Advisor Number Of Reviews
1	sugarfish by sushi nozawa	90017	2136273000	600 w 7th st	$$$	1776	4.5	$$-$$$	280
2	rock sugar pan asian kitchen	90067	3105529988	10250 santa monica blvd,	$$	1725,	4.5,	$$ - $$$,	336
3	katsuya hollywood	90028	3233912757	6300 hollywood blvd	$$$	1535	4.5,	$$$$	400
4	pagoda bar at yama-	90068	3234665125	1999 n sycamore ave	$$$	14	4	$$$$	618

	shiro hollywood								

- **append the code of the Python script to the end of this pdf file.**

```
import csv
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
from sklearn.naive_bayes import GaussianNB
from sklearn import linear_model
from sklearn.cross_validation import KFold
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
import numpy as np

with open("data_features.csv", 'r') as csvfile:
    reader = csv.DictReader(csvfile)
    s_features = []
    for row in reader:
        feature = []
        feature.append(float(row['name']))
        feature.append(float(row['addr']))
        feature.append(float(row['post']))
        feature.append(float(row['phone']))
        s_features.append(feature)

yelp = []
advisor = []
cols = ["shop name", "postal code", "phone number", "street address", "star", "price", "number of
reviews"]
with open("advisor_clean.csv", 'r') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        advisor.append(row)
with open("yelp_clean.csv", 'rb') as csvfile:
    reader = csv.DictReader(csvfile)
```

```
    for row in reader:
        yelp.append(row)

data = []
Matches = []
i = 0
for l_id, r_id in id_pairs:
    pair  = []
    pair.append(i, yelp[r_id][cols[0]], yelp[r_id][cols[1]], yelp[r_id][cols[2]], yelp[r_id][cols[3]], \
    yelp[r_id][cols[4]], yelp[r_id][cols[5]], yelp[r_id][cols[6]], advisor[l_id][cols[0]], \
    advisor[l_id][cols[1]], advisor[l_id][cols[2]], advisor[l_id][cols[3]], advisor[l_id][cols[4]],\
    advisor[l_id][cols[5]], advisor[l_id][cols[6]]);
    Matches.append(pair)
    data.append((i, yelp[r_id][cols[0]], yelp[r_id][cols[1]], \
                yelp[r_id][cols[2]], yelp[r_id][cols[3]], \
                yelp[r_id][cols[5]], yelp[r_id][cols[6]], advisor[l_id][cols[4]], \
                advisor[l_id][cols[5]], advisor[l_id][cols[6]]))
    i += 1

csvfile = file("E.csv", 'wb')
writer = csv.writer(csvfile)
writer.writerow(["ID", "shop name", "postal code", "phone number", "street address",  \
        "yelp price", "yelp number of reviews", \
        "advisor star", "advisor price", "advisor number of reviews"])
writer.writerows(data)
csvfile.close()

csvfile = file("matches.csv", 'wb')
writer = csv.writer(csvfile)
writer.writerows(Matches)
csvfile.close()
```