Project 10 Solutions

Kevin Choe

Collaborators: (Collaborators listed here. Include names, which part of the project you gave or sought help with, and how you helped or were helped.)

TA help: Summeth Guda

Online resources used: (List of links/resources (if any) here. Include web addresses, which part of the project the resource helped with, and how you were helped.)

Question 1

```
import pandas as pd
```

```
import numpy as np
import pandas as pd
beers = pd.read_parquet("/class/datamine/data/beer/beers.parquet")
breweries = pd.read_parquet("/class/datamine/data/beer/breweries.parquet")
reviews = pd.read_parquet("/class/datamine/data/beer/reviews.parquet")
def prepare_data(myDF, min_num_reviews: int):
  myDF = myDF.loc[myDF.loc[:, "score"].notna(), :]
  myDF = myDF.loc[myDF.loc[:, "username"].notna(), :]
  myDF = myDF.loc[myDF.loc[:, "beer_id"].notna(), :]
  myDF.reset_index(drop=True)
  goodbeers = myDF.loc[:, "beer_id"].value_counts() >= min_num_reviews
  goodbeers = goodbeers.loc[goodbeers].index.values.tolist()
  goodusers = myDF.loc[:, "username"].value_counts() >= min_num_reviews
  goodusers = goodusers.loc[goodusers].index.values.tolist()
  myreturnDF = myDF.loc[myDF.loc[:, "username"].isin(goodusers)&myDF.loc[:, "beer_id"].isin(goodbeers),
  return myreturnDF
train = prepare_data(reviews, 1000)

def normalize(data):
    data['mean_score'] = data['score'].mean()
    data['std_score'] = data['score'].std()
    data['normalized'] = (data['score'] - data['mean_score'])/data['std_score']
    return data

train= train.groupby(["username"]).apply(normalize)
score_matrix = train.pivot(index='username',columns='beer_id',values='normalized')
myresults=score_matrix.mean(axis=0)
score_matrix=score_matrix.fillna(value=myresults)

from sklearn.metrics.pairwise import cosine_similarity
cosine_similarity_matrix=cosine_similarity(score_matrix)
np.fill_diagonal(cosine_similarity_matrix,0)
```

```
cosine_similarity_matrix= pd.DataFrame(cosine_similarity_matrix)
cosine_similarity_matrix.index = score_matrix.index
cosine_similarity_matrix.columns = score_matrix.index
cosine_similarity_matrix[0:4]
```

```
username    1971bernat  1Sundown2C    22Blue  ...   zimm421  zonker17    zotzot
username                                      ...
1971bernat    0.000000    0.675032  0.718968  ...  0.732694  0.691533  0.731644
1Sundown2C    0.675032    0.000000  0.793614  ...  0.810388  0.766973  0.806054
22Blue        0.718968    0.793614  0.000000  ...  0.849088  0.810966  0.853877
2GOOFY        0.719644    0.792560  0.853467  ...  0.873437  0.815647  0.868995

[4 rows x 1824 columns]
```

```
cosine_similarity_matrix[1820:]
```

```
username  1971bernat  1Sundown2C    22Blue  ...   zimm421  zonker17    zotzot
username                                    ...
zestar      0.715598    0.773764  0.805240  ...  0.860737  0.799530  0.847968
zimm421     0.732694    0.810388  0.849088  ...  0.000000  0.826829  0.877142
zonker17    0.691533    0.766973  0.810966  ...  0.826829  0.000000  0.824804
zotzot      0.731644    0.806054  0.853877  ...  0.877142  0.824804  0.000000

[4 rows x 1824 columns]
```

Question 2

```
def get_knn (cosine_similarity_matrix,user,k):
  return cosine_similarity_matrix[user].sort_values(ascending=False)[0:k].index.tolist()

k_similar=get_knn(cosine_similarity_matrix,"2GOOFY",4)
print(k_similar) # ['Phil-Fresh', 'mishi_d', 'SlightlyGrey', 'MI_beerdrinker']
```

```
['Phil-Fresh', 'mishi_d', 'SlightlyGrey', 'MI_beerdrinker']
```

Question 3

```
User="mishi_d"
similar=get_knn(cosine_similarity_matrix,User,1)[0]
similar
```

```
'GoHabsGo'
```

```
aux=pd.DataFrame()
for i in range(0,reviews.shape[0]):
  if(reviews['username'][i]==User or reviews['username'][i]==similar):
    aux=aux.append(reviews.iloc[i])

aux_matrix=aux.pivot(index='beer_id',columns='username',values='score')
aux_matrix=aux_matrix.dropna(axis=0)
aux_matrix.head
#I think that the users rated the beers similarly because of the positive trend in data moving upwards.
```

```
<bound method NDFrame.head of username  GoHabsGo  mishi_d
beer_id
61.0          4.00     3.99
```

```
65.0          2.25     2.00
104.0         3.75     3.68
129.0         4.25     4.15
155.0         4.25     4.00
...            ...      ...
95680.0       3.50     3.00
98866.0       3.50     3.48
111537.0      3.75     3.75
120830.0      3.50     3.52
222579.0      4.00     3.79

[77 rows x 2 columns]>
```

Question 4

```
def recommend_beers(train: pd.DataFrame, username: str, cosine_similarity_matrix: pd.DataFrame, k: int)
  k_similar=get_knn(cosine_similarity_matrix, username, k)
  aux = pd.DataFrame(data = train[train["username"].isin(k_similar) == True])
  myBeers = train[train["username"].isin([username]) == True]
  myBeers = myBeers["beer_id"].to_list()
  aux = aux[aux["beer_id"].isin(myBeers) == False]
  aux = aux.loc[:, ("beer_id", "normalized")].groupby(["beer_id"]).mean()
  aux = aux.sort_values(by = "normalized", ascending = False)
  aux = aux.iloc[0:5]
  return aux.index.tolist()
recommend_beers(train, "22Blue", cosine_similarity_matrix, 30) # [40057, 69522, 22172, 59672, 86487]
```

```
[40057, 69522, 22172, 59672, 86487]
```

Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together – We are Purdue.