#Importing Libraries

import os

import pandas as pd

import numpy as np

import turicreate

#Reading Dataset

cd\_train = pd.read\_csv('coupon\_detail\_trainN.csv')

cl\_test = pd.read\_csv('coupon\_list\_testN.csv')

cl\_train = pd.read\_csv('coupon\_list\_trainN.csv')

user\_list = pd.read\_csv('user\_listN.csv')

print(cd\_train.head(8))

|  |  |  |
| --- | --- | --- |
| **USER\_ID\_hash** | **COUPON\_ID\_hash** | **PURCHASEID\_hash** |
| 0000b53e182165208887ba65c 079fc21 ... | 38beeadfe3f97e640367eddae 4a8c1b5 ... | 1 |
| 00035b86e6884589ec8d28fbf 2fe7757 ... | 25a27d420caa1c46a8d3c0572 d27868a ... | 1 |
| 0005b1068d5f2b8f2a7c978fc fe1ca06 ... | 4a79cd05ecb2bf8672e1d955f 5faa7fa ... | 1 |
| 0005b1068d5f2b8f2a7c978fc fe1ca06 ... | f0f66195d527a5a9509e139ed 367b879 ... | 1 |
| 000cc06982785a19e2a2fdb40 b1c9d59 ... | 229ff5cc21c8d26615493be7f 3b42841 ... | 1 |
| 000cc06982785a19e2a2fdb40 b1c9d59 ... | 35ed2dd67171a5defaac71ea3 1298f07 ... | 1 |
| 000cc06982785a19e2a2fdb40 b1c9d59 ... | 4f0145ab107bb2d4202ea3fed c0e3558 ... | 1 |
| 000cc06982785a19e2a2fdb40 b1c9d59 ... | 66066a64984f6ca94538b1795 48bbe8d ... | 1 |

print(cl\_train.head())

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PRICE\_RATE** | **CATALOG\_PRICE** | **DISCOUNT\_PRICE** | **DISPFROM** | **DISPEND** | **DISPPERIOD** | **VALIDPERIOD** |
| 50 | 3000 | 1500 | 2011-07-08 12:00:00 | 2011-07-09 12:00:00 | 1 | 151.0 |
| 51 | 2080 | 1000 | 2011-07-01 12:00:00 | 2011-07-02 12:00:00 | 1 | 154.0 |
| 50 | 7000 | 3500 | 2011-07-12 12:00:00 | 2011-07-15 12:00:00 | 3 | 179.0 |
| 50 | 3000 | 1500 | 2011-07-09 12:00:00 | 2011-07-11 12:00:00 | 2 | 142.0 |
| 50 | 2000 | 1000 | 2011-07-05 12:00:00 | 2011-07-06 12:00:00 | 1 | 176.0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **USABLE\_DATE\_MON** | **USABLE\_DATE\_TUE** | **USABLE\_DATE\_WED** | **USABLE\_DATE\_THU** | **USABLE\_DATE\_FRI** | **USABLE\_DATE\_SAT** | **USABLE\_DATE\_SUN** |
| 1.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 |

|  |  |  |
| --- | --- | --- |
| **USABLE\_DATE\_HOLIDAY** | **USABLE\_DATE\_BEFORE\_HOLIDA Y ...** | **COUPON\_ID\_hash** |
| 1.0 | 0.0 | 6b263844241eea98c5a97f133 5ea82af ... |
| 1.0 | 1.0 | cc031f250e8bad1e24060263b 9fc0ddd ... |
| 1.0 | 1.0 | ba5e9b7453ca52ff711635a5d 2e8102d ... |
| 1.0 | 1.0 | 3e1ffbedca3569f9e8032d401 e8cb4e6 ... |
| 1.0 | 0.0 | 782934b6c815b4030ea204eef 7d4a734 ... |

print(user\_list.head())

|  |
| --- |
| **USER\_ID\_hash** |
| d9dca3cb44bab12ba313eaa68 1f663eb ... |
| 560574a339f1b25e57b0221e4 86907ed ... |
| e66ae91b978b3229f8fd858c8 0615b73 ... |
| 43fc18f32eafb05713ec02935 e2c2825 ... |
| dc6df8aa860f8db0d710ce9d4 839840f ... |

#Preprocessing and Convert Data to SFrame

cd\_train = cd\_train.groupby(['USER\_ID\_hash', 'COUPON\_ID\_hash'])['PURCHASEID\_hash'].count().to\_frame().reset\_index()

cl\_train.drop(['VALIDFROM','VALIDEND'],axis=1,inplace=True)

cl\_test.drop(['VALIDFROM','VALIDEND'],axis=1,inplace=True)

cl\_train.VALIDPERIOD.fillna(180, inplace=True)

cl\_test.VALIDPERIOD.fillna(180, inplace=True)

cl\_train.fillna(1, inplace=True)

cl\_test.fillna(1, inplace=True)

observation\_data = turicreate.SFrame(cd\_train)

item\_data = turicreate.SFrame(pd.concat([cl\_train,cl\_test]))

users = turicreate.SFrame(user\_list[['USER\_ID\_hash']])

items = turicreate.SFrame(cl\_test[['COUPON\_ID\_hash']])

Fitting data into recommender model using cosine similarity

cosine\_model=turicreate.recommender.item\_content\_recommender.create(

similarity\_metrics='cosine',item\_data=item\_data, item\_id='COUPON\_ID\_hash', observation\_data=observation\_data,target='PURCHASEID\_hash', user\_id='USER\_ID\_hash', item\_data\_transform='auto', verbose=True)

Applying transform:

Class : AutoVectorizer

Model Fields

------------

Features : ['CAPSULE\_TEXT', 'GENRE\_NAME', 'PRICE\_RATE', 'CATALOG\_PRICE', 'DISCOUNT\_PRICE', 'DISPFROM', 'DISPEND', 'DISPPERIOD', 'VALIDPERIOD', 'USABLE\_DATE\_MON', 'USABLE\_DATE\_TUE', 'USABLE\_DATE\_WED', 'USABLE\_DATE\_THU', 'USABLE\_DATE\_FRI', 'USABLE\_DATE\_SAT', 'USABLE\_DATE\_SUN', 'USABLE\_DATE\_HOLIDAY', 'USABLE\_DATE\_BEFORE\_HOLIDAY', 'large\_area\_name', 'ken\_name', 'small\_area\_name']

Excluded Features : ['COUPON\_ID\_hash']

Column Type Interpretation Transforms Output Type

-------------------------- ----- -------------- ---------- -----------

CAPSULE\_TEXT str categorical None str

GENRE\_NAME str categorical None str

PRICE\_RATE int numerical None int

CATALOG\_PRICE int numerical None int

DISCOUNT\_PRICE int numerical None int

DISPFROM str categorical None str

DISPEND str categorical None str

DISPPERIOD int numerical None int

VALIDPERIOD float numerical None float

USABLE\_DATE\_MON float numerical None float

USABLE\_DATE\_TUE float numerical None float

USABLE\_DATE\_WED float numerical None float

USABLE\_DATE\_THU float numerical None float

USABLE\_DATE\_FRI float numerical None float

USABLE\_DATE\_SAT float numerical None float

USABLE\_DATE\_SUN float numerical None float

USABLE\_DATE\_HOLIDAY float numerical None float

USABLE\_DATE\_BEFORE\_HOLIDAY float numerical None float

large\_area\_name str categorical None str

ken\_name str categorical None str

small\_area\_name str categorical None str

Defaulting to brute force instead of ball tree because there are multiple distance components.

Starting brute force nearest neighbors model training.

Validating distance components.

Initializing model data.

Initializing distances.

Done.

Starting pairwise querying.

+--------------+---------+-------------+--------------+

| Query points | # Pairs | % Complete. | Elapsed Time |

+--------------+---------+-------------+--------------+

| 1 | 19723 | 0.00507022 | 240.353ms |

| 242 | 4772966 | 1.22699 | 3.24s |

| 598 | 1.2e+07 | 3.03199 | 6.24s |

| 999 | 2e+07 | 5.06515 | 9.24s |

| 1417 | 2.8e+07 | 7.18451 | 12.25s |

| 1793 | 3.5e+07 | 9.09091 | 15.25s |

| 2229 | 4.4e+07 | 11.3015 | 18.24s |

| 2601 | 5.1e+07 | 13.1876 | 21.24s |

| 3051 | 6e+07 | 15.4692 | 24.25s |

| 3496 | 6.9e+07 | 17.7255 | 27.25s |

| 3921 | 7.7e+07 | 19.8803 | 30.25s |

| 4346 | 8.6e+07 | 22.0352 | 33.25s |

| 4785 | 9.4e+07 | 24.261 | 36.25s |

| 5193 | 1e+08 | 26.3297 | 39.25s |

| 5611 | 1.1e+08 | 28.449 | 42.25s |

| 6018 | 1.2e+08 | 30.5126 | 45.25s |

| 6402 | 1.3e+08 | 32.4596 | 48.29s |

| 6833 | 1.3e+08 | 34.6448 | 51.25s |

| 7245 | 1.4e+08 | 36.7338 | 54.27s |

| 7647 | 1.5e+08 | 38.772 | 57.26s |

| 8046 | 1.6e+08 | 40.795 | 1m 0s |

| 8473 | 1.7e+08 | 42.96 | 1m 3s |

| 8878 | 1.8e+08 | 45.0134 | 1m 6s |

| 9288 | 1.8e+08 | 47.0922 | 1m 9s |

| 9713 | 1.9e+08 | 49.2471 | 1m 12s |

| 10137 | 2e+08 | 51.3968 | 1m 15s |

| 10536 | 2.1e+08 | 53.4199 | 1m 18s |

Preparing data set.

Data has 158933 observations with 22782 users and 19723 items.

Data prepared in: 1.58917s

Loading user-provided nearest items.

Generating candidate set for working with new users.

Finished training in 0.156412s

#Recommend

cosine\_res = cosine\_model.recommend(users, k=2, items=items).to\_dataframe()

recommendations finished on 1000/22873 queries. users per second: 20148.7

recommendations finished on 2000/22873 queries. users per second: 37987.4

recommendations finished on 3000/22873 queries. users per second: 53857.1

recommendations finished on 4000/22873 queries. users per second: 67961.3

recommendations finished on 5000/22873 queries. users per second: 80732.4

recommendations finished on 6000/22873 queries. users per second: 92506.9

recommendations finished on 7000/22873 queries. users per second: 103316

recommendations finished on 8000/22873 queries. users per second: 113031

recommendations finished on 9000/22873 queries. users per second: 121729

recommendations finished on 10000/22873 queries. users per second: 129676

recommendations finished on 11000/22873 queries. users per second: 137255

recommendations finished on 12000/22873 queries. users per second: 143038

recommendations finished on 13000/22873 queries. users per second: 149623

recommendations finished on 14000/22873 queries. users per second: 155845

recommendations finished on 15000/22873 queries. users per second: 161737

recommendations finished on 16000/22873 queries. users per second: 162848

recommendations finished on 17000/22873 queries. users per second: 166850

recommendations finished on 18000/22873 queries. users per second: 171538

recommendations finished on 19000/22873 queries. users per second: 176122

recommendations finished on 20000/22873 queries. users per second: 180473

recommendations finished on 21000/22873 queries. users per second: 183661

#Convert to Submission Format

def clean\_prediction(row):

data = row.PURCHASED\_COUPONS

data = str("".join(str(data))[2:-2].replace("', '"," || "))

return data

cosine\_res=cosine\_res.groupby('USER\_ID\_hash')['COUPON\_ID\_hash'].apply(

list).reset\_index(name='PURCHASED\_COUPONS')

cosine\_res['PURCHASED\_COUPONS'] = cosine\_res.apply(clean\_prediction, axis=1)

cosine\_res.rename(columns = {'PURCHASED\_COUPONS':'RECOMMENDED\_COUPONS'}, inplace = True)

cosine\_res.to\_csv('sub\_cpp\_turi\_item\_content\_cosine.csv', index=False)

#Output is list of recommended coupons to each user given below

| **USER\_ID\_hash** | **RECOMMENDED\_COUPONs** |
| --- | --- |
| **0** | 0000b53e182165208887ba65c079fc21 | dd74dc95ca294afa02db40a543ae1763 || c76ea297eb... |
| **1** | 00035b86e6884589ec8d28fbf2fe7757 | f5a77f2907876411752d58e1b9030023 || d506a61810... |
| **2** | 0005b1068d5f2b8f2a7c978fcfe1ca06 | 0c015306597566b632bebfb63b7e59f3 || dd74dc95ca... |
| **3** | 000cc06982785a19e2a2fdb40b1c9d59 | c988d799bc7db9254fe865ee6cf2d4ff || 5e47b887e1... |
| **4** | 0013518e41c416cd6a181d277dd8ca0b | c988d799bc7db9254fe865ee6cf2d4ff || 86c6439131... |
| **...** | ... | ... |
| **22868** | fff1a623187cefd7a594e338709b0f40 | 6d3ea4f9c9272ee7595eaca7f96234db || c988d799bc... |
| **22869** | fff4a076cfda6ff9dbe85e1cb678791b | c988d799bc7db9254fe865ee6cf2d4ff || 6d3ea4f9c9... |
| **22870** | fff970d2014c3e10a77e38d540239017 | c988d799bc7db9254fe865ee6cf2d4ff || 23d51ce25f... |
| **22871** | fffafc024e264d5d539813444cf61199 | 0fd38be174187a3de72015ce9d5ca3a2 || c988d799bc... |
| **22872** | ffff56dbf3c782c3532f88c6c79817ba | c988d799bc7db9254fe865ee6cf2d4ff || d38cd84f81... |

22873 rows × 2 columns