In [1]:

```
import pandas as pd
import numpy as np
import os
import sys
import logging as log
import google.cloud.logging
from google.cloud import bigquery
from argparse import ArgumentParser
from time import sleep
from datetime import datetime
from pytz import timezone
from random import randint
import subprocess
```

In [11]:

```
%%bigquery data

SELECT  query_v2.query_key as CustomerID,   query_v2.raw,  query_v2.ordered,  query_v2.ordered_stem_
wo_stop_words as stem_stop,
        item.id as StockCode, count(*) as freq , item.title

FROM relatedsearch.dim_cat_trimmed
inner join  relatedsearch.item on dim_cat_trimmed.category_code = item.cat
inner join  relatedsearch.query_source_mapping_v2 on query_source_mapping_v2.source_key = item.key
inner join  relatedsearch.query_v2 on query_v2.query_key = query_source_mapping_v2.query_key

where fourth_sub_category  = "Tablet" and
    DATE(dim_cat_trimmed.source_time) between "2018-01-20" and "2020-07-28" and
    DATE(item.source_time) between "2018-01-20" and "2020-07-28" and
    DATE(query_source_mapping_v2.source_time) between "2018-01-20" and "2020-07-28" and
    DATE(query_v2.source_time) between "2018-01-20" and "2020-07-28"

group by  query_v2.query_key,  item.id, query_v2.raw,  query_v2.ordered,  query_v2.ordered_stem_wo_s
top_words, item.title

order by item.id
```

In [12]:

```
data
```

Out[12]:

	CustomerID	raw	ordered	stem_sto
0	2UEnXzvG18mt6DUwaa7ecmlw1AqZ7L0ZkXr2mk/eBxI=	ucuz tablet	ucuz tablet	ucuz table
1	k8NdVRWT3FgGtLj4ErfH9mcocsWzmQHU6kb1fkMuPog=	en ucuz tabletler	en ucuz tabletler	en ucu table
2	W+2FvWmlUCqPZFpYoF6Xn5S963e5Vxmc6EKeD/vAElc=	tablet	tablet	table
3	W+2FvWmlUCqPZFpYoF6Xn5S963e5Vxmc6EKeD/vAElc=	tablet	tablet	table
4	niW5+dzcTvXhZF49NpFskturkVlwHfM55nuLYCMK/G8=	Çaycı	çaycı	çayc
...	...	...	...	.
118578	zb41GIT07VU1pi4VgsDtIXV610ZJfY7D8rPBT/iKdLc=	general+mobile+e+tab+5+ekran	general mobile e tab 5 ekran	genera mobile tab 5 ekra
118579	Ds/nxyYoSfyIxCR2C9QXOG0KT/OzvJP/H6r5sxrtvig=	general mobile e tab 5	general mobile e tab 5	genera mobile tab
118580	K48NghUz/kLMyAJKYKORCJ+8TC8ruPD9bgltytF/bas=	ipad 2 el	ipad 2 el	ipad 2 e
118581	7Mv5YZqXkfU7nvOBU8qIZK5anixNrdu3ffQ2RoQwevs=	ipat+air+2	ipad air 2	ipad air
118582	Zs/VCxs5AVvReb+74HzoIZcXACR7k10EHzl7VzUBTlo=	2+el+tablet	2 el tablet	2 el table

118583 rows × 7 columns

In [ ]:

```
```

In [ ]:

```
```

In [4]:

```
import pandas as pd
import numpy as np
import random
from tqdm import tqdm
from gensim.models import Word2Vec
import matplotlib.pyplot as plt
%matplotlib inline

import warnings;
warnings.filterwarnings('ignore')
```

In [14]:

```
df=data
```

In [15]:

```
#
df['StockCode']= df['StockCode'].astype(str)
```

In [16]:

```
# customer ID's
customers = df["CustomerID"].unique().tolist()
len(customers)
```

Out[16]:

```
18274
```

In [17]:

```
# extract 90% of customer ID's
customers_train = [customers[i] for i in range(round(0.9*len(customers)))]

# split data into train and validation set
train_df = df[df['CustomerID'].isin(customers_train)]
validation_df = df[~df['CustomerID'].isin(customers_train)]
```

In [18]:

```
# list to capture purchase history of the customers
purchases_train = []

# populate the list with the product codes
for i in tqdm(customers_train):
    temp = train_df[train_df["CustomerID"] == i]["StockCode"].tolist()
    purchases_train.append(temp)
```

```
100%|██████████| 16447/16447 [02:15<00:00, 121.70it/s]
```

In [ ]:

```

```

In [159]:

```
customers_train[0]
```

Out[159]:

```
'2UEnXzvG18mt6DUwaa7ecmlw1AqZ7L0ZkXr2mk/eBxI='
```

In [151]:

```
purchases_train[0][0]
```

Out[151]:

```
'165786711'
```

In [19]:

```
# list to capture purchase history of the customers
purchases_val = []

# populate the list with the product codes
for i in tqdm(validation_df['CustomerID'].unique()):
    temp = validation_df[validation_df["CustomerID"] == i]["StockCode"].tolist()
    purchases_val.append(temp)
```

```
100%|██████████| 1827/1827 [00:01<00:00, 1302.15it/s]
```

In [ ]:

```

```

In [20]:

```
# train word2vec model
model = Word2Vec(window = 10, sg = 1, hs = 0,
                 negative = 10, # for negative sampling
                 alpha=0.03, min_alpha=0.0007,
                 seed = 14)

model.build_vocab(purchases_train, progress_per=200)

model.train(purchases_train, total_examples = model.corpus_count,
            epochs=10, report_delay=1)
```

Out[20]:

```
(1033111, 1161980)
```

In [21]:

```
# save word2vec model
model.save("word2vec_2.model")
```

In [22]:

```
model.init_sims(replace=True)
```

In [23]:

```
print(model)
```

```
Word2Vec(vocab=6960, size=100, alpha=0.03)
```

In [ ]:

```

```

In [24]:

```
# extract all vectors
X = model[model.wv.vocab]

X.shape
```

Out[24]:

```
(6960, 100)
```

In [42]:

```
X
```

Out[42]:

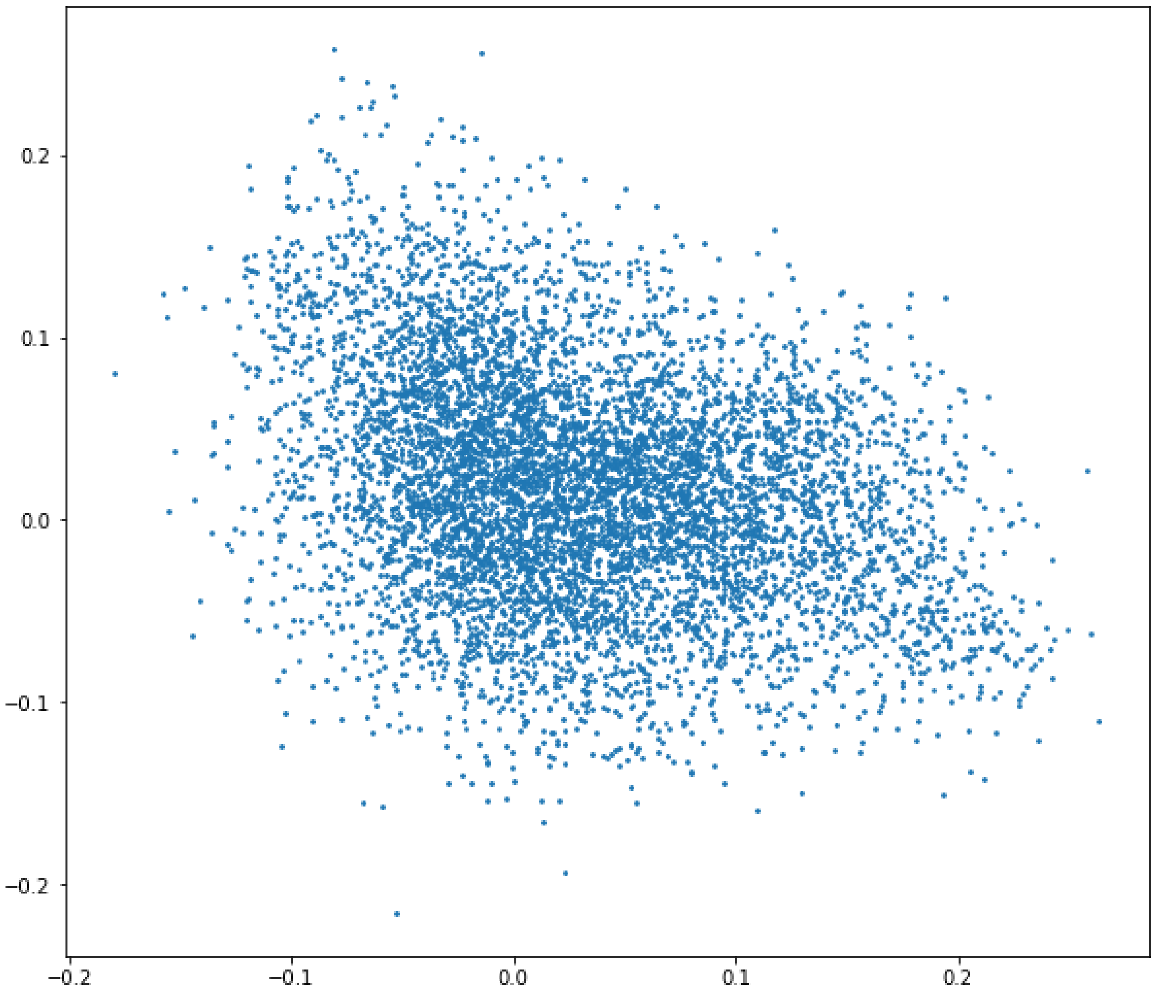
```
array([[-0.05284742,  0.00712433,  0.20459254, ..., -0.01402055,
        -0.11476439,  0.09722929],
       [-0.03493589,  0.0191889 ,  0.1708321 , ...,  0.02028203,
        -0.15433466,  0.00557681],
       [-0.04775743, -0.0607722 ,  0.16536613, ..., -0.02587532,
         0.04191248,  0.09388705],
       ...,
       [ 0.09830297, -0.03798198, -0.03748675, ...,  0.01540025,
         0.01946338,  0.08791307],
       [ 0.0818606 ,  0.00990511, -0.03087101, ..., -0.06615732,
         0.03966628,  0.11811795],
       [-0.1558237 ,  0.11166029,  0.01835854, ...,  0.00354234,
        -0.15070991,  0.09588706]], dtype=float32)
```

In [74]:

```
plt.figure(figsize=(10,9))
plt.scatter(X[:, 0], X[:, 1], s=3, cmap='Spectral')
```

Out[74]:

```
<matplotlib.collections.PathCollection at 0x7f42bacf1e50>
```



In [169]:

```
products = train_df[["StockCode", "ordered"]]

# remove duplicates
#products.drop_duplicates(inplace=True, subset='StockCode', keep="last")

# create product-ID and product-description dictionary
products_dict = products.groupby('StockCode')['ordered'].apply(list).to_dict()
```

In [170]:

```
products
```

Out[170]:

	StockCode	ordered
0	165786711	ucuz tablet
1	165786711	en ucuz tabletler
2	165786711	tablet
3	166429944	tablet
4	181456675	çaycı
...	...	...
118578	580673686	general mobile e tab 5 ekran
118579	580673686	general mobile e tab 5
118580	580705948	ipad 2 el
118581	580705948	ipad air 2
118582	580714545	2 el tablet

116198 rows × 2 columns

In [ ]:

```

```

In [187]:

```
# test the dictionary
products_dict['165786711']
```

Out[187]:

```
['ucuz tablet', 'en ucuz tabletler', 'tablet']
```

In [173]:

```
def similar_products(v, n = 9):

    # extract most similar products for the input vector
    ms = model.similar_by_vector(v, topn= n+1)[0:]

    # extract name and similarity score of the similar products
    new_ms = []
    # new_ms.append(ms)
    for j in ms:
        pair = (j[0], j[1],  products_dict[j[0]][0] )
        new_ms.append(pair)

    return new_ms
```

In [174]:

```
similar_products(model['556534244'])
```

Out[174]:

```
[('556534244', 1.0, 'samsung tablet'),
 ('554937714', 0.954814076423645, '4 gb ram tablet'),
 ('555864992', 0.936670184135437, 'galaxy tab'),
 ('556782532', 0.9223512411117554, 'samsung t510'),
 ('555048663', 0.9208989143371582, 'samsung tablet'),
 ('554374345', 0.9140273928642273, 'samsung galaxy tab'),
 ('556313357', 0.9132769703865051, 'iphone tablet'),
 ('557320325', 0.9101274013519287, 'samsung tablet'),
 ('554374351', 0.9078882932662964, 'samsung galaxy tab a 10 1 2019'),
 ('557894635', 0.9039982557296753, 'samsung galaxy tab a sm t510 10 1')]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [125]:

```
def aggregate_vectors(products):
    product_vec = []
    for i in products:
        try:
            product_vec.append(model[i])
        except KeyError:
            continue

    return np.mean(product_vec, axis=0)
```

In [126]:

```
aggregate_vectors(purchases_val[0]).shape
```

Out[126]:

```
(100,)
```

In [188]:

```
similar_products(aggregate_vectors(purchases_val[20]))
```

Out[188]:

```
[('555501237', 0.965438961982727, 'windows tablet'),
 ('555109726', 0.965438961982727, 'alcatel tablet'),
 ('555683587', 0.9452964067459106, 'hometech tablet'),
 ('554804715', 0.9421983957290649, 'sim kartlı tablet'),
 ('556560707', 0.9396722316741943, 'teşhir'),
 ('555158381', 0.9385358095169067, 'doğubank tablet'),
 ('554640560', 0.9372175931930542, 'alcatel 3 16 gb t8'),
 ('555604088', 0.9371612071990967, '2 gb ram tabletler'),
 ('554462533', 0.9360650181770325, 'tablet'),
 ('554942967', 0.9339286088943481, 'hometech tablet')]
```

In [193]:

```
products_dict['555501237']
```

Out[193]:

```
['windows tablet',
 'xiaomi tablet',
 '2 el tablet',
 'ikisi bir arada bilgisayar',
 'hdmi tablet',
 'outlet tablet',
 'tablet',
 'tablet',
 'outlet']
```

In [192]:

```
similar_products(aggregate_vectors(purchases_val[1]))
```

Out[192]:

```
[('555001334', 1.0, 'apple ipad'),
 ('554100949', 0.9686295986175537, 'ipad'),
 ('554262381', 0.9668571352958679, 'ipad 2'),
 ('553809807', 0.9642125368118286, 'ipad'),
 ('554374340', 0.9616502523422241, 'apple tablet'),
 ('555039506', 0.9609588384628296, 'teşhir tablet'),
 ('555481997', 0.9580819606781006, 'ipad'),
 ('554098463', 0.9576786756515503, 'ipad 4 nesil'),
 ('554336537', 0.9565437436103821, 'grafik tablet'),
 ('554349803', 0.9557768106460571, 'apple ipad pro')]
```

In [ ]:

```
products_dict['165786711']
```

In [190]:

```
purchases_val[20][-5:]
```

Out[190]:

```
['555109726', '555501237', '555832120']
```

In [204]:

```
purchases_val[40]
```

Out[204]:

```
['555465752', '556563987']
```

In [205]:

```
purchases_val[1]
```

Out[205]:

```
['554937714']
```

In [208]:

```
similar_products(aggregate_vectors(['555001334']))
```

Out[208]:

```
[('555001334', 1.0, 'apple ipad'),
 ('554100949', 0.9686295986175537, 'ipad'),
 ('554262381', 0.9668571352958679, 'ipad 2'),
 ('553809807', 0.9642125368118286, 'ipad'),
 ('554374340', 0.9616502523422241, 'apple tablet'),
 ('555039506', 0.9609588384628296, 'teşhir tablet'),
 ('555481997', 0.9580819606781006, 'ipad'),
 ('554098463', 0.9576786756515503, 'ipad 4 nesil'),
 ('554336537', 0.9565437436103821, 'grafik tablet'),
 ('554349803', 0.9557768106460571, 'apple ipad pro')]
```

In [ ]:

```

```