Overview

This code will generate two new attributes to our review data:

- `sentiment_score` : sentiment score of each review (using textblob)
- `most_rev` : max number of reviews made in a day by a profile id

Sentiment Analysis

Using textblob module to calculate polarity from -1.0 to 1.0 (whether the comment goes toward negative or positive sentiment) and the subjectivity of the review.

For more info check here

```
from google.colab import drive
drive.mount("/content/gdrive/")
```

```
Mounted at /content/gdrive/
```

```
from textblob import TextBlob
import csv
import pandas as pd

profile_data = pd.read_csv("/content/gdrive/MyDrive/MSING055_Amazon_Review-master/final_datas
reviews = profile_data['review'].tolist()
sentiment_score = []
sentiment_subjectivity=[]
review_head_sentiment=[]
for rev in reviews:
    testimonial = TextBlob(str(rev))
    sentiment_score.append(testimonial.sentiment.polarity)
    sentiment_subjectivity.append(testimonial.sentiment.subjectivity)
sentiment_score
```

```
0.5,
0.1865234375,
0.30000000000000004,
-0.08333333333333333,
-0.4035714285714286,
-0.16666666666666666,
0.056249999999999994,
0.09374999999999999,
0.3833333333333333,
0.20000000000000004,
0.175,
0.23888888888888885,
```

```
        0.16041666666666668,
        0.14545454545454545,
        -0.6953125,
        0.2583333333333333,
        0.3,
        0.05000000000000001,
        -0.18055555555555558,
        0.32,
        0.2,
        0.38,
        0.3333333333333333,
        0.5,
        -0.15555555555555553,
        0.175,
        0.45,
        -0.65,
        1.0,
        0.07999999999999999,
        0.0,
        0.4,
        0.5,
        0.0,
        0.07999999999999999,
        0.3,
        0.7,
        -0.4,
        0.21542929292929294,
        0.024537037037037048,
        -0.06785714285714287,
        0.0,
        0.13,
        0.7,
        1.0,
        0.33928571428571436,

        -0.010000000000000009,
        0.44,
        0.13375,
        0.12042857142857143,
        0.08148148148148147,
        -0.225,
        0.04866666666666667,
        0.525,
        0.1811011904761905,
        0.5,
        0.40166666666666667,
        0.437037037037037,
        0.39999999999999997,
```

```
    sentiment_subjectivity
```

```
        0.4561224489795918,
        0.625,
        0.58125,
        0.65,
        0.4666666666666666,
        0.6642857142857144,
        0.16666666666666666,
```

```
        0.35,
        0.6062500000000001,
        0.4833333333333334,
        0.5,
        0.5416666666666667,
        0.5155555555555555,
        0.5555555555555556,
        0.3666666666666667,
        0.5,
        0.675,
        0.625,
        0.3333333333333333,
        0.3138888888888889,
        0.5900000000000001,
        0.4,
        0.72,
        0.5,
        0.6,
        0.49999999999999994,
        0.7,
        0.2,
        1.0,
        0.3,
        0.68,
        0.1,
        0.7,
        0.5333333333333333,
        0.0,
        0.45999999999999996,
        0.43333333333333335,
        0.8,
        0.6,
        0.4297979797979798,
        0.5555555555555557,

        0.5857142857142857,
        0.0,
        0.44000000000000006,
        0.8,
        1.0,
        0.5238916256157636,
        0.8,
        0.6166666666666667,
        0.43375,
        0.5204285714285716,
        0.512962962962963,
        0.725,
        0.613,
        0.575,
        0.472842261904762,
        0.4375,
        0.6683333333333333,
        0.41851851851851857,
```

```
#sanity check
testimonial = TextBlob("I'm really sad with the product")
testimonial.sentiment.polarity
```

```
-0.5
```

```
#sanity check
testimonial = TextBlob("It was awesome! I've never had something like this before. Definitely
testimonial.sentiment.polarity
```

```
0.5
```

```
profile_data['review_head_sentiment'] = sentiment_score
profile_data['review_sentiment'] = sentiment_score
profile_data['sentiment_subjectivity'] = sentiment_subjectivity
profile_data
```

	Unnamed: 0	url	review_bold	ratings	
0	0.0	https://www.amazon.co.uk/gp/review/R18JE8EYCYX...	I rarely use them but..	5.0	h
1	1.0	https://www.amazon.co.uk/gp/review/R3M2146VNTD...	Best Price for Great Workout / Running Headphones	5.0	a T
2	2.0	https://www.amazon.co.uk/gp/review/R3U00QINFHP...	Amazed at how good these sound for the price.	5.0	T
3	3.0	https://www.amazon.co.uk/gp/review/R5BZDAV3JC42E/	Earbuds that give you great sound	5.0	e

```
profile_data.to_csv('/content/gdrive/MyDrive/MSING055_Amazon_Review-master/final_dataset/fina
```

Most reviews made in a day

In this section we'll calculate the highest number of reviews that a profile made within a day. The logic is pretty straightforward since the date is already sorted from most recent:

- for every profile_id, check the date
- if the current date row has the same value as the previous one, we'll add 1 to the counter `seq`
- we compare `seq` for each day, and put the highest value to the dictionary `pid_seq`,with the `profile_id` as the key

```
import datetime
import pandas as pd
profile_data = pd.read_csv("/content/gdrive/MyDrive/MSING055_Amazon_Review-master/final_datas
profile_id = profile_data['profile_id']
date = (profile_data['date'])
```

```
i = 1
pid_seq={}

while i < (len(profile_data)):
    print(i)
    if (profile_data['profile_id'][i] == profile_data['profile_id'][i-1]):
```

```
        if (profile_data['date'][i] == profile_data['date'][i-1]):
            seq += 1
        else:
            if (profile_data['profile_id'][i] in pid_seq.keys()):
                if (seq>pid_seq[profile_data['profile_id'][i]]):
                    pid_seq[profile_data['profile_id'][i]] = seq
            else:
                pid_seq[profile_data['profile_id'][i]] = seq
            seq=1
        i+=1
    else:
        seq=1
        i+=1

for pid in profile_id:
    if len(str(pid))> 30:
        profile_id.replace(pid,'')
    if pid not in pid_seq.keys():
        pid_seq[pid] = 1
```

```
32375
32376
32377
32378
32379
32380
32381
32382
32383
32384
32385
32386
32387
32388
32389
32390
32391
32392
32393
32394
32395
32396
32397
32398
32399
32400
32401
32402
32403
32404
32405
32406
32407
32408
32409
```

```
32410
32411
32412
32413
32414
32415
32416
32417
32418
32419
32420
32421
32422
32423
32424
32425
32426
32427
32428
32429
32430

32431
32432
32433
```

```
pid_seq
```

```
'AFAF75AQSFWKBVA5JFKYLC5RDNSQ': 35,
'AFANY7IRVGTM5USYBWAEMO2RQBEA': 1,
'AFAZLF4B2D2FY4WY4VOXBKBXBR6A': 6,
'AFB3UVQ2LLGBIYRXVZ47PM5RJZYA': 1,
'AFB6P6QJZ6XNQ5PRC4ZGOC6N5QPA': 1,
'AFBFWZG6VDK7IZ4GIEL4NOSOPKOQ': 1,
'AFBKPIEM6R26HRFTSM4QHDU274YA': 7,
'AFC6S7AGO5LVWFDP5EVVNGDFOWWQ': 1,
'AFCFXAAGESHAWKXZ5G3NN5IY35JQ': 1,

'AFCNZESS7HSFPC743JKQF6DONNWA': 1,
'AFCR7R53YMSUHSJWUCPV64BY3E4Q': 1,
'AFCRVPWYLMWTK43SA3OHS3SZKVCQ': 1,
'AFCWXK5CUQ7BBR6DYA7K5JZZ3RGA': 1,
'AFD2USHPSYVYLZQJIC44Y73BRURQ': 1,
'AFD64LT4SBCT6SVE7MCIFZCR5DQQ': 1,
'AFDA3T3DLM3ZIEOAVE34BROWPZVA': 4,
'AFDAMDCMVTC7W5QLZCH5L43IU3XA': 1,
'AFDC5N5OQLEOOR3OE2QVYRU2HP2Q': 1,
'AFDGWOW7VSIB24TIPFLCR4OAUSFQ': 1,
'AFDIAHMLUMYTE4HJH4VJ36WARFJA': 1,
'AFDO3RRRAZUNK7OYDM34LSXXAHOQ': 1,
'AFDPPIBXN2MGH7LYQ5Z34I6BLHEA': 5,
'AFEDDI6DU3OJJTAV43I4NMWOOL3Q': 1,
'AFEF5LQTDE3B6QKQ36J42CZ4GZRA': 1,
'AFEFEZD5QC24IVMWSV7B3BSTBTSA': 1,
'AFEJP75K2APT66KURONXWYIDYOTA': 1,
'AFEQAD3EYBX3YMWTNQVBOWO4JJPQ': 1,
'AFEV4A3JGXRIASZMMOQ74CHR2YPQ': 1,
'AFF5VC5TSX5YFUZNSAAYZX7TVPFA': 9,
```

```
 'AFF7WXFXHG42YCVUJZPBBVRFFQDA': 4,
 'AFFCWLGV7I47NGVUKPDJQADD5FFQ': 4,
 'AFFHO4X75KQ5JGO7VG6AFEZHWAFA': 1,
 'AFFNQLAWE6HEKG4AXXVVUGVM6EFQ': 1,
 'AFFRTAUYKRQUJBDQEZ7OOHA4REOQ': 1,
 'AFFTKR2SRKIDQ3S3DAAPVQWET3PQ': 5,
 'AFG5SDRMLEZCN4IEWI6VQMIFCHIQ': 1,
 'AFGBKKVTV37UTWF4TIVHOWXHH5SQ': 1,
 'AFGST3MYPWLQBDIEILCIC27ZEJOQ': 1,
 'AFHJCKPOPMIZ4CVLATOJOOKOTLPA': 1,
 'AFHKE4F7I3VKCG53X7LUDC7U47YA': 1,
 'AFHKPHVUKUYSJCCRFHIMZMGFELVQ': 1,
 'AFHUQ6VV4LG7YLCMCZOSWIECYKBA': 1,
 'AFHVXQU2HZHO4JMVRHLFBKADTSXA': 1,
 'AFICTKJGYCJKBS7DF5X2KYZWFB7A': 1,
 'AFIDFQMUSDNLBMPSGLMRVWT6YZ7A': 2,
 'AFIGMDIOXQQRRJ6VX32ZTH734KQQ': 1,
 'AFIRPQAUW5LBKTSZCPGTPA2QBLKA': 8,
 'AFJ3U5V6LBPM5KA3PYDAWAD5LJKA': 1,
 'AFJ47HU7Q3WV2GFH5N6Q54QS7IEA': 1,
 'AFJ666YPRH7T6X3F2QICG3Q2VSTQ': 1,
 'AFJ7M6NRYU54VQUC2522BWJE7VJA': 1,
 'AFJADYS5PNAQYM2XLDJUBL3SENQA': 48,
 'AFJGPKEQ5H7PZR44W7NDZKE5KENQ': 1,
 'AFJRQJUX74N3SMVHYCHQ2M4C2OKQ': 1,
 'AFJVIYHIPJTCDXMUYTMPOTTP6KZQ': 1,
 'AFJWYM6F2H7HR5M4OMVOJU5O75AQ': 1,
 'AFK7652QI4OHY6TKXMZCQCIQ2ASQ': 1,
 'AFKP7RJONDZMBZYUKXD4W2IEOFSA': 1,
```

```
len(pid_seq)
```

```
882
```

```
profile_id
```

```
0                             AFNMUCU4D7HN6NLUSQXPYSS7ZT6A
1                             AEMS5UDZTZN4L3BIJV57COUNV2FA
2                             AEMS5UDZTZN4L3BIJV57COUNV2FA
3                             AEMS5UDZTZN4L3BIJV57COUNV2FA
4                             AEMS5UDZTZN4L3BIJV57COUNV2FA
                               ...
33060                         AGPL5CCMK7H5Y4INIAGSNGHER36A
33061                                                  NaN
33062    14(L)Black Women's Lace Tops Long Sleeve Peplu...
33063                         AGPL5CCMK7H5Y4INIAGSNGHER36A
33064                         AGPL5CCMK7H5Y4INIAGSNGHER36A
Name: profile_id, Length: 33065, dtype: object
```

```
pid=[]
count_rev=[]
for k,v in pid_seq.items():
    pid.append(k)
    count_rev.append(v)
```

```
df_count = pd.DataFrame(
        {'profile_id': pid,
         'most_rev': count_rev,
        })
df_count.to_csv("/content/gdrive/MyDrive/MSING055_Amazon_Review-master/final_dataset/final_da
```

```
df_count
```

	profile_id	most_rev
0	AEMS5UDZTZN4L3BIJV57COUNV2FA	1
1	AG57ELELG6RNTZWK7RDHDO53RE4A	9
2	AHZI3QY65DLTDUDWWIPI4SQQBKVA	1
3	AHJ4TATYP5OGXMMFDJ2XZSTKJYDQ	1
4	AHPBJJIJAYS65CAV372NAHS6IZXA	1
...	...	...
877	MoonarÂ®Vintage Women's Leopard Printing Peter...	1
878	Womens Ladies Aztec Diamond Print Long Sleeve ...	1
879	Roman Women's Patterned Ribbed Knitted Dress C...	1
880	Rockport Women's Jalicia Buckle Tall Heels Boot	1
881	14(L)Black Women's Lace Tops Long Sleeve Peplu...	1

882 rows × 2 columns