7506-2020-2 Parcialito de Pandas

Juan Bautista Xifro

TOTAL POINTS

85 / 100

QUESTION 1

1 Pandas 85 / 100

✓ **- 15 pts Calcula mal la cantidad de sesiones por usuario / cuenta con count los session id / no dropea duplicados**

1. Deberías dropear los duplicados por session id ya que como menciona el enunciado:
"Una sesión está compuesta por al menos un evento."
Para contar las sesiones del usuario deberias contar la cantidad de valores diferentes que hay para "session_id"

2. checkout_mean

ExamenPandas

October 15, 2020

[167]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import random
```

[168]:
```
#genero dataframe
df_events = pd.DataFrame(columns = ['uid', 'sessionid','event', 'value',␣
 ↪'datetime'], dtype = int)
uid = [2, 9, 28, 6, 1, 25]
datetime = ['2020-12-06 22:00:00','2019-09-06 22:00:00', '2020-04-20 21:00:00',␣
 ↪'2019-12-06 23:00:00']
sessionid = [1011, 1012, 1013, 1014, 1015, 1016, 1017]
event = ['ecommerce.view-product', 'ecommerce.checkout', 'ecommerce.
 ↪conversion', 'evento4', 'evento5', 'evento6', 'evento7']
value = [200, 2000, 2800, 100, 100, 2500, 25]
for i in range(500):
    df_events = df_events.append({
        'uid': random.choice(uid),
        'sessionid': random.choice(sessionid),
        'event': random.choice(event),
        'value': random.choice(value),
        'datetime': random.choice(datetime)
    }, ignore_index = True)
```

[169]:
```
#casteo la columna 'datetime' a un formato datetime64
df_events['datetime'] = pd.to_datetime( df_events['datetime'], errors='coerce')
df_events.head()
```

[169]:

	uid	sessionid	event	value	datetime
0	6	1017	ecommerce.checkout	100	2020-04-20 21:00:00
1	1	1012	ecommerce.view-product	2800	2019-12-06 23:00:00
2	6	1013	evento4	2800	2019-12-06 23:00:00
3	6	1011	evento5	2800	2020-12-06 22:00:00
4	9	1016	evento7	100	2019-12-06 23:00:00

[170]:
```
usuarios = df_events.groupby('uid').agg({'sessionid':'count'})
usuarios = usuarios.reset_index()
usuarios = usuarios.rename(columns = {'sessionid' : 'cant_sessions'})
usuarios.head()
```

[170]:

	uid	cant_sessions
0	1	77
1	2	84
2	6	80
3	9	102
4	25	88

[171]:
```
promedio_sessions = usuarios['cant_sessions'].mean()
promedio_sessions
```

[171]: 83.33333333333333

[172]:
```
usuarios_filtrados = usuarios[usuarios['cant_sessions'] > promedio_sessions]
usuarios_filtrados = usuarios_filtrados.reset_index()
usuarios_filtrados['uid']
```

[172]:
```
0     2
1     9
2    25
Name: uid, dtype: int64
```

b) Calcular el valor promedio de los eventos 'ecommerce.view-product','ecommerce.checkout', 'ecommerce.conversion' para aquellos usuarios que hayan realizadomás de 25 sesiones. Los resultados deben ser presentados en un dataframe con las siguientescolumnas ('uid','ecommerce_view_product_mean', 'ecommerce_checkout_mean','ecommerce_conversion_mean')v

[173]:
```
events_filtrado = df_events[(df_events['event'] == 'ecommerce.view-product') |
                            (df_events['event'] == 'ecommerce.checkout')|
                            (df_events['event'] == 'ecommerce.conversion')]
usuarios = usuarios[usuarios['cant_sessions'] > 25]
events_merge = events_filtrado.merge(usuarios, on = 'uid')
events_merge
```

[173]:

	uid	sessionid	event	value	datetime	\
0	6	1017	ecommerce.checkout	100	2020-04-20 21:00:00	
1	6	1011	ecommerce.conversion	2500	2019-12-06 23:00:00	
2	6	1017	ecommerce.view-product	100	2020-12-06 22:00:00	
3	6	1011	ecommerce.conversion	25	2019-12-06 23:00:00	
4	6	1014	ecommerce.checkout	2500	2019-12-06 23:00:00	
..	…	…	…	…	…	
199	2	1017	ecommerce.view-product	2500	2020-04-20 21:00:00	

```
200    2       1013    ecommerce.conversion   2000 2019-12-06 23:00:00
201    2       1011      ecommerce.checkout     25 2020-12-06 22:00:00
202    2       1012      ecommerce.checkout   2500 2020-04-20 21:00:00
203    2       1011      ecommerce.checkout    100 2019-12-06 23:00:00

     cant_sessions
0               80
1               80
2               80
3               80
4               80
..             …
199             84
200             84
201             84
202             84
203             84

[204 rows x 6 columns]
```

[174]:
```
pivoted = events_merge.pivot_table(index = 'uid', columns = 'event', values =␣
↪'value', aggfunc = 'mean')
pivoted = pivoted.reset_index()
pivoted
```

[174]:
```
event  uid  ecommerce.checkout  ecommerce.conversion  ecommerce.view-product
0        1            1287.500            770.312500             1527.941176
1        2             987.500           1493.181818             2740.000000
2        6             731.250            750.000000             1075.000000
3        9            1675.000           1439.583333              565.000000
4       25            1143.750           2255.555556             1171.428571
5       28            1140.625           1225.000000             1829.545455
```

[175]:
```
pivoted = pivoted.rename(columns = {'ecommerce.checkout':
↪'ecommerce_view_product_mean',
                                    'ecommerce.conversion' : 'ecommerce.
↪conversion_mean',
                                    'ecommerce.view-product': 'ecommerce.
↪view-product_mean'})
pivoted
```

[175]:
```
event  uid  ecommerce_view_product_mean  ecommerce.conversion_mean  \
0        1                    1287.500                 770.312500
1        2                     987.500                1493.181818
2        6                     731.250                 750.000000
3        9                    1675.000                1439.583333
4       25                    1143.750                2255.555556
```

```
5       28                     1140.625                1225.000000
```

```
event  ecommerce.view-product_mean
0                      1527.941176
1                      2740.000000
2                      1075.000000
3                       565.000000
4                      1171.428571
5                      1829.545455
```

[ ]:

1 Pandas 85 / 100

✓ **- 15 pts Calcula mal la cantidad de sesiones por usuario / cuenta con count los session id / no dropea duplicados**

1. Deberías dropear los duplicados por session id ya que como menciona el enunciado:
"Una sesión está compuesta por al menos un evento."
Para contar las sesiones del usuario deberias contar la cantidad de valores diferentes que hay para "session_id"

2. checkout_mean