Imputations

April 21, 2021

0.1 # Imputations and Data Preparations

[95]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from openpyxl import load_workbook
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer

np.set_printoptions(suppress=True)
```

[198]:
```
xls = pd.ExcelFile('data/main dataset.xlsx')
ad_post = pd.read_excel(xls, 'Ad-Post')
ad_story = pd.read_excel(xls, 'Ad-Story')
influencer = pd.read_excel(xls, 'Influencer')
leaders_post = pd.read_excel(xls, 'Leaders-Post')
leaders_story = pd.read_excel(xls, 'Leaders-Story')
post = pd.read_excel(xls, 'Post')
story = pd.read_excel(xls, 'Story')
print('Datasets Loaded Completely.')
```

```
Datasets Loaded Completely.
```

[136]:
```
null_count_ad_story = len(ad_story[(ad_story['action'].isna()) |␣
↪(ad_story['impression'].isna())])
null_count_influencer = len(influencer[(influencer['action'].isna()) |␣
↪(influencer['impression'].isna()) | (influencer['interaction'].isna())])
null_count_leaders_story = len(leaders_story[(leaders_story['impression'].
↪isna())])
null_count_story = len(story[(story['follow'].isna()) | (story['navigation'].
↪isna()) | (story['back'].isna()) | (story['forward'].isna())
                             | (story['next'].isna()) | (story['exit'].isna())])
null_count_leaders_post = len(leaders_post[(leaders_post['view'].isna())])
print(f'Ad-Story Dataset has {null_count_ad_story} records with missing␣
↪features.')
print(f'Influencer Dataset has {null_count_influencer} records with missing␣
↪features.')
```

```
print(f'Leaders-Story Dataset has {null_count_leaders_story} records with␣
 ↪missing features.')
print(f'Story Dataset has {null_count_story} records with missing features.')
print(f'Story Dataset has {null_count_leaders_post} records with missing␣
 ↪features.')
```

```
Ad-Story Dataset has 17 records with missing features.
Influencer Dataset has 25 records with missing features.
Leaders-Story Dataset has 5 records with missing features.
Story Dataset has 6 records with missing features.
Story Dataset has 1 records with missing features.
```

0.2 Data Splitting

In this step we need to split the data for imputation. For instance if we want to impute the 'action' feature in ad_story dataset, we must put independent features (in this example, followers, view, interaction and cost) to a new matrix and train the imputer based on it and then predict the missing features with it

[141]:
```
ad_story_imputation = ad_story.drop(['ad_story_no', 'name', 'field',␣
 ↪'threshold'], 1)
influencer_imputation = influencer.drop(influencer.columns.
 ↪difference(['follower', 'view', 'action', 'cost', 'impression', 'cta',␣
 ↪'interaction']), 1)
leaders_story_imputation = leaders_story.drop(leaders_story.columns.
 ↪difference(['follower', 'view', 'action', 'interaction', 'impression']), 1)
story_imputation = story.drop(['story_no', 'type'], 1)
leaders_post_imputation = leaders_post.drop(['post_no', 'name', 'gender',␣
 ↪'l_threshold', 'h_threshold'], 1)
```

[142]:
```
ad_story_imputation_matrix = ad_story_imputation.values
influencer_imputation_matrix = influencer_imputation.values
leaders_story_imputation_matrix = leaders_story_imputation.values
story_imputation_matrix = story_imputation.values
leaders_post_imputation_matrix = leaders_post_imputation.values

imp_ad_story = IterativeImputer(max_iter = 10)
imp_influencer = IterativeImputer(max_iter = 10)
imp_story = IterativeImputer(max_iter = 10)
imp_leaders_story = IterativeImputer(max_iter = 10)
imp_leaders_post = IterativeImputer(max_iter = 10)

predicted_ad_story = np.round(imp_ad_story.
 ↪fit_transform(ad_story_imputation_matrix))
predicted_influencer = np.round(imp_influencer.
 ↪fit_transform(influencer_imputation_matrix))
predicted_story = np.round(imp_story.fit_transform(story_imputation_matrix))
```

```
predicted_leaders_story = np.round(imp_leaders_story.
 ↪fit_transform(leaders_story_imputation_matrix))
predicted_leaders_post = np.round(imp_leaders_post.
 ↪fit_transform(leaders_post_imputation_matrix))
```

```
C:\Users\Ramin\anaconda3\lib\site-packages\sklearn\impute\_iterative.py:669:
ConvergenceWarning: [IterativeImputer] Early stopping criterion not reached.
  warnings.warn("[IterativeImputer] Early stopping criterion not"
```

[143]:
```
df_final_ad_story = pd.DataFrame(predicted_ad_story, columns = ['follower',␣
 ↪'view', 'action', 'interaction', 'impression', 'cost'])
df_final_influencer = pd.DataFrame(predicted_influencer, columns = ['follower',␣
 ↪'view', 'action', 'impression', 'cta', 'interaction', 'cost'])
df_final_story = pd.DataFrame(predicted_story, columns = ['view', 'actions',␣
 ↪'reply', 'profile_visit', 'share', 'website_click', 'sticker_tap',␣
 ↪'impression',
                                                          'follow',␣
 ↪'navigation', 'back', 'forward', 'next', 'exit', 'vote'])
df_final_leaders_story = pd.DataFrame(predicted_leaders_story, columns =␣
 ↪['follower', 'view', 'action', 'interaction', 'impression'])
df_final_leaders_post = pd.DataFrame(predicted_leaders_post, columns =␣
 ↪['follower', 'view', 'like', 'comment', 'share', 'save', 'profile_visit',␣
 ↪'reach', 'impression', 'cost'])
```

[199]:
```
ad_story = ad_story.merge(df_final_ad_story, on='view', how='left')
ad_story.drop(['follower_x', 'action_x', 'interaction_x', 'impression_x',␣
 ↪'cost_x'], axis=1, inplace=True)
ad_story.rename(columns={'follower_y': 'follower',
                         'action_y': 'action',
                         'interaction_y': 'interaction',
                         'impression_y': 'impression',
                         'cost_y': 'cost'},
                inplace=True)
ad_story
```

[199]:

	ad_story_no	ad_story_no.1	name	field	view	\
0	0	1	4rahesalamat	health	6260	
1	1	2	90tv.official	news	58990	
2	2	3	ancientworld1	fact	101631	
3	3	4	ayamidooni	fact	97671	
4	4	5	banooye_khone	women	21887	
5	5	6	danestani_rooz	fact	205375	
6	6	7	diaa_graphy	art & culture	23200	
7	7	8	dialogism	art & culture	12460	
8	8	9	doctor_khabar	fact	40400	
9	9	10	filmak20	video	62300	
10	10	11	fitness_clip	health	35900	

11	11	12	fitology_group	health	18500
12	12	13	histofeed	fact	40412
13	13	14	i.wonders	fact	27544
14	14	15	inform.mag	fact	28400
15	15	16	ketab_am	art & culture	76627
16	16	17	khane.va.khanedari	women	21212
17	17	18	morphin.graphy	art & culture	64360
18	18	19	movaffagh_sho	women	2514
19	19	20	persian.dialogue	fact	21110
20	20	21	picoplay	fact	154819
21	21	22	picopry	fact	161584
22	22	23	salamatparsi	health	4365
23	23	24	shekamo_haa	women	26226
24	24	25	zheen_magazine	art & culture	62200
25	25	26	zhuaan	art & culture	78874
26	26	27	3kanstv	video	41004

	threshold	follower	action	interaction	impression	cost
0	8	686000.0	82.0	7.0	6374.0	190578.0
1	8	877000.0	234.0	90.0	58568.0	444000.0
2	8	2600000.0	273.0	218.0	94682.0	556000.0
3	8	2300000.0	365.0	488.0	92023.0	650000.0
4	8	2400000.0	239.0	38.0	74414.0	430000.0
5	8	4500000.0	850.0	523.0	206633.0	1450000.0
6	8	311000.0	33.0	116.0	1141.0	104590.0
7	8	1100000.0	42.0	30.0	12759.0	75000.0
8	8	1900000.0	135.0	202.0	49339.0	243000.0
9	8	1700000.0	188.0	311.0	53222.0	343000.0
10	8	2100000.0	114.0	179.0	50374.0	198000.0
11	8	857000.0	45.0	92.0	14124.0	111000.0
12	8	1500000.0	68.0	43.0	41053.0	222354.0
13	8	1100000.0	72.0	137.0	23240.0	151551.0
14	8	2300000.0	65.0	46.0	51587.0	157000.0
15	8	1300000.0	381.0	302.0	74765.0	800000.0
16	8	643000.0	81.0	61.0	14515.0	180000.0
17	8	852000.0	193.0	321.0	36411.0	380000.0
18	8	479000.0	38.0	14.0	2421.0	76535.0
19	8	605000.0	35.0	18.0	8764.0	116095.0
20	8	2200000.0	339.0	774.0	87158.0	585000.0
21	8	3600000.0	791.0	807.0	182013.0	1400000.0
22	8	560000.0	60.0	11.0	4640.0	132887.0
23	8	2500000.0	278.0	131.0	82249.0	500000.0
24	8	411000.0	133.0	311.0	18489.0	280410.0
25	8	2200000.0	392.0	348.0	78402.0	500000.0
26	8	1000000.0	143.0	200.0	31790.0	287000.0

[200]:
```
influencer = influencer.merge(df_final_influencer, on=df_final_influencer.index␣
↪, how='left')
influencer.drop(['key_0', 'follower_x', 'view_x', 'action_x', 'impression_x',␣
↪'cta_x', 'interaction_x', 'cost_x'], axis=1, inplace=True)
influencer.rename(columns={'follower_y': 'follower',
                           'view_y': 'view',
                           'action_y': 'action',
                           'interaction_y': 'interaction',
                           'impression_y': 'impression',
                           'cta_y': 'cta',
                           'cost_y': 'cost'},
                  inplace=True)
influencer
```

[200]:

	story_no	story_no.1	influ_name	gender	field	l_threshold	\
0	0	1	ali_bakhtiarvandi	family	lifestyle	20	
1	1	2	ali_bakhtiarvandi	family	lifestyle	20	
2	2	3	ali_bakhtiarvandi	family	lifestyle	20	
3	3	4	ali_bakhtiarvandi	family	lifestyle	20	
4	4	5	ali_bakhtiarvandi	family	lifestyle	20	
..	…	…	…	…	…	…	
97	97	98	ghazalnevis	female	health	20	
98	98	99	mahshidseydi	family	lifestyle	20	
99	99	100	mahshidseydi	family	lifestyle	20	
100	100	101	mahshidseydi	family	lifestyle	20	
101	101	102	mahshidseydi	family	lifestyle	20	

	h_threshold	follower	view	action	impression	cta	interaction	\
0	60	141000.0	3996.0	14.0	4186.0	0.0	0.0	
1	60	141000.0	3279.0	30.0	3473.0	1.0	28.0	
2	60	141000.0	3636.0	5.0	3867.0	0.0	0.0	
3	60	141000.0	3145.0	16.0	3317.0	1.0	11.0	
4	60	141000.0	3113.0	30.0	3286.0	1.0	22.0	
..	…	…	…	…	…	…	…	
97	60	45100.0	12000.0	229.0	12876.0	1.0	132.0	
98	60	89500.0	4854.0	46.0	4945.0	1.0	41.0	
99	60	89500.0	4695.0	37.0	4829.0	1.0	35.0	
100	60	89500.0	4623.0	44.0	4758.0	1.0	35.0	
101	60	89500.0	4431.0	87.0	4666.0	1.0	64.0	

	cost
0	360000.0
1	360000.0
2	360000.0
3	360000.0
4	360000.0
..	…

```
97    125000.0
98    625000.0
99    625000.0
100   625000.0
101   625000.0

[102 rows x 14 columns]
```

[201]:
```
story = story.merge(df_final_story, on=df_final_story.index , how='left')
story.drop(['key_0', 'view_x', 'actions_x', 'reply_x', 'profile_visit_x',␣
 ↪'share_x', 'website_click_x', 'sticker_tap_x', 'impression_x',
            'follow_x', 'navigation_x', 'back_x', 'forward_x', 'next_x',␣
 ↪'exit_x', 'vote_x'],
           axis=1, inplace=True)
story.rename(columns={'view_y': 'view',
                      'actions_y': 'action',
                      'reply_y': 'reply',
                      'profile_visit_y': 'profile_visit',
                      'share_y': 'share',
                      'website_click_y': 'website_click',
                      'sticker_tap_y': 'sticker_tap',
                      'impression_y': 'impression',
                      'follow_y': 'follow',
                      'navigation_y': 'navigation',
                      'back_y': 'back',
                      'forward_y': 'forward',
                      'next_y': 'next',
                      'exit_y': 'exit',
                      'vote_y': 'vote'},
             inplace=True)
story
```

[201]:

	story_no	story_no.1	type	view	action	reply	profile_visit	\
0	0	1	share	1337.0	53.0	4.0	49.0	
1	1	2	share	1164.0	114.0	2.0	110.0	
2	2	3	share	727.0	21.0	1.0	20.0	
3	3	4	share	850.0	45.0	5.0	40.0	
4	4	5	share	1294.0	69.0	8.0	58.0	
5	5	6	share	1404.0	70.0	3.0	65.0	
6	6	7	share	1277.0	118.0	6.0	54.0	
7	7	8	share	1021.0	43.0	3.0	38.0	
8	8	9	share	781.0	26.0	4.0	22.0	
9	9	10	share	668.0	21.0	3.0	17.0	
10	10	11	share	668.0	12.0	3.0	9.0	
11	11	12	share	1023.0	38.0	5.0	32.0	
12	12	13	share	825.0	36.0	5.0	30.0	
13	13	14	share	887.0	18.0	8.0	10.0	

14	14	15	share	869.0	27.0	7.0	16.0
15	15	16	share	634.0	6.0	1.0	5.0
16	16	17	share	700.0	15.0	3.0	12.0
17	17	18	share	575.0	12.0	0.0	12.0
18	18	19	share	552.0	8.0	2.0	6.0
19	19	20	share	495.0	6.0	0.0	6.0
20	20	21	share	531.0	8.0	3.0	5.0
21	21	22	share	583.0	15.0	2.0	13.0
22	22	23	share	521.0	5.0	2.0	3.0
23	23	24	share	591.0	16.0	2.0	14.0
24	24	25	share	546.0	9.0	3.0	6.0
25	25	26	share	461.0	34.0	1.0	4.0
26	26	27	share	393.0	6.0	2.0	4.0
27	27	28	share	485.0	8.0	4.0	4.0
28	28	29	share	433.0	9.0	3.0	6.0
29	29	30	poll	1434.0	27.0	3.0	24.0
30	30	31	poll	1103.0	42.0	2.0	34.0
31	31	32	poll	1267.0	16.0	3.0	11.0
32	32	33	poll	1024.0	26.0	4.0	14.0
33	33	34	poll	765.0	12.0	3.0	9.0
34	34	35	poll	578.0	14.0	3.0	10.0
35	35	36	contest	901.0	16.0	3.0	13.0
36	36	37	contest	819.0	9.0	0.0	9.0
37	37	38	contest	803.0	397.0	3.0	18.0
38	38	39	contest	776.0	193.0	2.0	22.0
39	39	40	contest	740.0	26.0	6.0	17.0

	share	website_click	sticker_tap	impression	follow	navigation	back	\
0	0.0	0.0	0.0	1380.0	0.0	1618.0	28.0	
1	1.0	1.0	0.0	1190.0	1.0	1490.0	106.0	
2	0.0	0.0	0.0	765.0	0.0	772.0	38.0	
3	0.0	0.0	0.0	930.0	1.0	1038.0	31.0	
4	0.0	3.0	0.0	1384.0	0.0	1522.0	35.0	
5	2.0	0.0	0.0	1465.0	2.0	1702.0	65.0	
6	58.0	0.0	0.0	1316.0	3.0	1649.0	224.0	
7	0.0	2.0	0.0	1097.0	0.0	1372.0	129.0	
8	0.0	0.0	0.0	806.0	0.0	944.0	14.0	
9	1.0	0.0	0.0	674.0	1.0	823.0	59.0	
10	0.0	0.0	0.0	687.0	0.0	738.0	14.0	
11	0.0	1.0	0.0	1074.0	2.0	1355.0	19.0	
12	0.0	1.0	0.0	1032.0	1.0	1032.0	78.0	
13	0.0	0.0	0.0	843.0	0.0	1121.0	63.0	
14	4.0	0.0	0.0	913.0	0.0	1147.0	73.0	
15	0.0	0.0	0.0	637.0	0.0	742.0	34.0	
16	0.0	0.0	0.0	716.0	0.0	832.0	17.0	
17	0.0	0.0	0.0	592.0	1.0	668.0	22.0	
18	0.0	0.0	0.0	571.0	0.0	652.0	6.0	

19	0.0	0.0	0.0	508.0	0.0	575.0	29.0
20	0.0	0.0	0.0	546.0	0.0	618.0	21.0
21	0.0	0.0	0.0	593.0	0.0	711.0	11.0
22	0.0	0.0	0.0	524.0	0.0	616.0	33.0
23	0.0	0.0	0.0	589.0	0.0	655.0	21.0
24	0.0	0.0	0.0	545.0	0.0	633.0	9.0
25	29.0	0.0	0.0	464.0	0.0	530.0	14.0
26	0.0	0.0	0.0	410.0	0.0	472.0	22.0
27	0.0	0.0	0.0	498.0	0.0	542.0	18.0
28	0.0	0.0	0.0	451.0	0.0	510.0	9.0
29	0.0	0.0	0.0	1457.0	1.0	1467.0	20.0
30	5.0	1.0	0.0	1107.0	1.0	1330.0	66.0
31	2.0	0.0	0.0	1346.0	0.0	1399.0	38.0
32	6.0	2.0	0.0	1072.0	0.0	1247.0	82.0
33	0.0	0.0	0.0	753.0	0.0	798.0	6.0
34	1.0	0.0	0.0	577.0	0.0	689.0	23.0
35	0.0	0.0	0.0	934.0	0.0	1061.0	13.0
36	0.0	0.0	0.0	853.0	0.0	949.0	60.0
37	80.0	0.0	296.0	836.0	3.0	1192.0	405.0
38	10.0	0.0	159.0	813.0	1.0	1058.0	181.0
39	3.0	0.0	0.0	778.0	0.0	1075.0	202.0

	forward	next	exit	vote
0	1048.0	179.0	363.0	0.0
1	919.0	119.0	350.0	0.0
2	428.0	92.0	214.0	0.0
3	531.0	125.0	351.0	0.0
4	909.0	186.0	392.0	0.0
5	1160.0	191.0	286.0	0.0
6	1106.0	102.0	217.0	0.0
7	748.0	107.0	388.0	0.0
8	589.0	136.0	205.0	0.0
9	505.0	67.0	192.0	0.0
10	443.0	110.0	171.0	0.0
11	879.0	153.0	204.0	0.0
12	772.0	25.0	157.0	0.0
13	780.0	94.0	184.0	0.0
14	876.0	43.0	155.0	0.0
15	589.0	35.0	84.0	0.0
16	596.0	79.0	140.0	0.0
17	461.0	54.0	131.0	0.0
18	464.0	69.0	113.0	0.0
19	457.0	31.0	58.0	0.0
20	431.0	49.0	117.0	0.0
21	470.0	93.0	137.0	0.0
22	458.0	53.0	72.0	0.0
23	450.0	56.0	128.0	0.0

```
24    421.0   84.0  119.0    0.0
25    409.0   36.0   71.0    0.0
26    332.0   33.0   85.0    0.0
27    396.0   36.0   92.0    0.0
28    379.0   53.0   69.0    0.0
29    968.0  264.0  215.0  109.0
30    927.0  136.0  199.0    0.0
31    900.0  255.0  206.0  165.0
32    876.0  113.0  175.0    0.0
33    551.0  131.0  110.0   74.0
34    438.0   72.0  152.0    0.0
35    742.0  153.0  153.0    0.0
36    709.0   97.0   83.0    0.0
37    539.0  -53.0  338.0    0.0
38    562.0   33.0  292.0    0.0
39    675.0   44.0  154.0    0.0
```

[202]:
```
leaders_story = leaders_story.merge(df_final_leaders_story,␣
↪on=df_final_leaders_story.index , how='left')
leaders_story.drop(['key_0', 'follower_x', 'view_x', 'action_x',␣
↪'interaction_x', 'impression_x'],
           axis=1, inplace=True)
leaders_story.rename(columns={'view_y': 'view',
                              'follower_y': 'follower',
                              'action_y': 'action',
                              'interaction_y': 'interaction',
                              'impression_y': 'impression'},
             inplace=True)
leaders_story
```

[202]:

	story_no	story_no.1	name	gender	cost	follower	\
0	0	1	aidapooryanasab	female	0	692000.0	
1	1	2	alimona.trips	family	0	73400.0	
2	2	3	amirparsaneshat	male	0	146000.0	
3	3	4	ghonche.ostovarnia	female	0	122000.0	
4	4	5	maandani	male	0	128000.0	
5	5	6	shahabjafarnejad	male	0	133000.0	
6	6	7	yaasamin_	female	0	189000.0	
7	7	8	yaasamin_	female	0	189000.0	
8	8	9	taaraa.moheb	female	0	757000.0	
9	9	10	mr.alisaa	family	0	54000.0	
10	10	11	mr.alisaa	family	0	54000.0	
11	11	12	mr.alisaa	family	0	54000.0	

	view	action	interaction	impression
0	103909.0	651.0	562.0	107902.0
1	4169.0	162.0	130.0	3548.0

```
2    26972.0   527.0        335.0     26925.0
3     8381.0   205.0        154.0      8381.0
4    10493.0   178.0        151.0     10952.0
5     7809.0   128.0         74.0      7991.0
6    12352.0   288.0        194.0     12640.0
7    15021.0    83.0          0.0     13525.0
8   148197.0  3538.0       2392.0    150001.0
9     5020.0   104.0         59.0      4229.0
10    4234.0    34.0         22.0      3483.0
11    3803.0   162.0        154.0      3002.0
```

[203]:
```
leaders_post = leaders_post.merge(df_final_leaders_post,␣
↪on=df_final_leaders_post.index , how='left')
leaders_post.drop(['key_0', 'follower_x', 'view_x', 'like_x', 'comment_x',␣
↪'share_x', 'save_x', 'profile_visit_x', 'reach_x', 'impression_x', 'cost_x'],
          axis=1, inplace=True)
leaders_post.rename(columns={'follower_y': 'follower',
                             'view_y': 'view',
                             'like_y': 'like',
                             'comment_y': 'comment',
                             'share_y': 'share',
                             'save_y': 'save',
                             'profile_visit_y': 'profile_visit',
                             'reach_y': 'reach',
                             'impression_y': 'impression',
                             'cost_y': 'cost'},
                    inplace=True)
leaders_post
```

[203]:

	post_no	post_no.1	name	gender	l_threshold	h_threshold	\
0	0	1	aidapooryanasab	female	200	400	
1	1	2	alimona.trips	family	200	400	
2	2	3	amirparsaneshat	male	200	400	
3	3	4	ghonche.ostovarnia	female	200	400	
4	4	5	maandani	male	200	400	
5	5	6	shahabjafarnejad	male	200	400	
6	6	7	yaasamin_	female	200	400	
7	7	8	taaraa.moheb	female	200	400	
8	8	9	mr.alisaa	family	200	400	

	follower	view	like	comment	share	save	profile_visit	\
0	692000.0	78137.0	17500.0	205.0	275.0	272.0	1374.0	
1	73400.0	20220.0	5099.0	140.0	238.0	138.0	463.0	
2	146000.0	128378.0	25940.0	573.0	7732.0	7207.0	2593.0	
3	122000.0	103347.0	12300.0	733.0	261.0	471.0	6611.0	
4	128000.0	15002.0	2408.0	68.0	98.0	232.0	482.0	
5	133000.0	15701.0	2766.0	35.0	46.0	73.0	125.0	

```
6  189000.0   31714.0   7890.0    211.0   499.0   272.0          427.0
7  757000.0  108552.0  12731.0    208.0   207.0   278.0         1060.0
8   54000.0    6191.0   1201.0     41.0    15.0    24.0           64.0

      reach  impression        cost
0  149048.0    162532.0  30000000.0
1   31642.0     38437.0   5000000.0
2  146276.0    180104.0  15200000.0
3  156349.0    172354.0   6000000.0
4   27562.0     30204.0   5200000.0
5   33338.0     36830.0  19000000.0
6   67071.0     74606.0  10000000.0
7  115662.0    171570.0  30000000.0
8    8311.0      9589.0   2000000.0
```

[205]:
```
book = load_workbook('data/main dataset.xlsx')
writer = pd.ExcelWriter('data/main dataset.xlsx', engine='openpyxl')
writer.book = book
writer.sheets = dict((ws.title, ws) for ws in book.worksheets)
ad_story.to_excel(writer, "Ad-Story")
influencer.to_excel(writer, "Influencer")
leaders_post.to_excel(writer, "Leaders-Post")
leaders_story.to_excel(writer, "Leaders-Story")
story.to_excel(writer, "Story")
writer.save()
```

[ ]:
```

```