수집된 모든 텍스트데이터와 메타데이터 전체 결합 및 재분리

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import time
import random
from datetime import datetime, timedelta
np.random.seed(1)
```

In [3]:
```
parse_dates=['date_created','date_deadline','date_launched']
ks_dataset = pd.read_csv('./ks_overall_dataset.csv', parse_dates=parse_dates)
```

In [4]:
```
ks_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200232 entries, 0 to 200231
Data columns (total 27 columns):
blurb                       200230 non-null object
category_name               200232 non-null object
category_specific           200232 non-null object
country_displayable_name    200232 non-null object
created_at                  200232 non-null int64
date_created                200232 non-null datetime64[ns]
deadline                    200232 non-null int64
date_deadline               200232 non-null datetime64[ns]
disable_communication       200232 non-null bool
id                          200232 non-null int64
is_starrable                200232 non-null bool
launched_at                 200232 non-null int64
date_launched               200232 non-null datetime64[ns]
date_launched_year          200232 non-null int64
date_launched_month         200232 non-null int64
location_name               200025 non-null object
name                        200232 non-null object
photo_link                  200232 non-null object
state                       200232 non-null object
state_changed_at            200232 non-null int64
date_state_changed          200232 non-null object
url_project                 200232 non-null object
usd_goal                    200232 non-null float64
usd_pledged                 200232 non-null float64
set_fundraising_period      200232 non-null int64
launching_delay_time        200232 non-null int64
target_goal_period_rate     200232 non-null float64
dtypes: bool(2), datetime64[ns](3), float64(3), int64(9), object(10)
memory usage: 38.6+ MB
```

In [10]:
```
ks_dataset[['name','blurb','photo_link']]
# 추가) photo 2015~2019년 다시 크롤링 하기위해 'photo_link' 컬럼 가져오기
```

Out[10]:

	name	blurb	photo_link
0	Louli à l'école	Un livre enfant pour l'apprentissage des émoti...	https://ksr-ugc.imgix.net/assets/025/618/188/1...
1	Strange Wit, an original graphic novel about J...	The true biography of the historical figure, w...	https://ksr-ugc.imgix.net/assets/012/216/438/f...
2	FAM - FIND A MOTIVE MOBILE APP	FAM is the new mobile app which combines event...	https://ksr-ugc.imgix.net/assets/021/506/749/1...
3	Destiny, NY - FINAL HOURS!	A graphic novel about two magical ladies in love.	https://ksr-ugc.imgix.net/assets/013/616/177/6...
4	Publishing Magus Magazine	We are publishing a magazine that focuses on t...	https://ksr-ugc.imgix.net/assets/011/292/182/9...
...	...	...	...
200227	Making Jenna Bialostosky's awesome debut album.	I'm raising money to make my debut, self-relea...	https://ksr-ugc.imgix.net/assets/011/264/421/7...
200228	Hoerner Bros Hot Sauce Company startup with a ...	We are starting a hot sauce company from the g...	https://ksr-ugc.imgix.net/assets/021/869/677/f...
200229	Unpacked Urban Tote	Artist Designed Urban Tote\nPunKin Apparel by ...	https://ksr-ugc.imgix.net/assets/025/451/684/5...
200230	The Pumpkin Roll Express/Mean Bean Cafe Express	Looking to expand my pumpkin roll sales to a m...	https://ksr-ugc.imgix.net/assets/011/756/585/c...
200231	ECO-FRIENDLY MOBILE FARM STAND FOR URBAN FOOD ...	All-electric mobile farm stand for transportin...	https://ksr-ugc.imgix.net/assets/011/534/972/8...

200232 rows × 3 columns

In [18]:
```
# meta dataset 읽어오기

ks_meta = pd.read_csv('./ks_meta_dataset.csv')
```

In [19]:
```
# meta dataset에 name, blurb 컬럼 추가

ks_meta[['name','blurb','photo_url']] = ks_dataset[['name','blurb','photo_link']]
```

In [20]:
```
ks_meta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200232 entries, 0 to 200231
Data columns (total 15 columns):
category_name               200232 non-null object
location_name               200025 non-null object
country_displayable_name    200232 non-null object
usd_goal                    200232 non-null float64
set_fundraising_period      200232 non-null int64
target_goal_period_rate     200232 non-null float64
date_launched_year          200232 non-null int64
date_launched_month         200232 non-null int64
launching_delay_time        200232 non-null int64
disable_communication       200232 non-null bool
is_starrable                200232 non-null bool
state                       200232 non-null object
name                        200232 non-null object
blurb                       200230 non-null object
photo_url                   200232 non-null object
dtypes: bool(2), float64(2), int64(4), object(7)
memory usage: 20.2+ MB
```

In [21]:
```
ks_meta
```

Out[21]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_de
0	Children's Books	Ste.-Maxime	France	684.396336	5184000	0.000132	2019	6	
1	Graphic Novels	Minneapolis	the United States	12000.000000	2592000	0.004630	2015	8	
2	Apps	London	the United Kingdom	132713.121000	5184000	0.025601	2018	6	
3	Graphic Novels	New York	the United States	20000.000000	2565470	0.007796	2016	10	
4	Periodicals	Logan	the United States	5000.000000	2592000	0.001929	2011	9	
...	...	...	...	...	...	...	...	...	...
200227	Jazz	Glendale	the United States	3000.000000	4664051	0.000643	2010	7	
200228	Small Batch	Tampa	the United States	5000.000000	3888000	0.001286	2018	7	
200229	Textiles	Seattle	the United States	700.000000	2657079	0.000263	2019	6	
200230	Food Trucks	Locust Grove	the United States	20000.000000	3551605	0.005631	2014	7	
200231	Farmer's Markets	Chicago	the United States	19000.000000	2592000	0.007330	2015	3	5

200232 rows × 15 columns

2015~2019년 meta data만 가져오기

In [22]:
```
ks_meta_2015_2019 = ks_meta[ks_meta['date_launched_year'].isin([2015,2016,2017,2018,2019])]
ks_meta_2015_2019
```

Out[22]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_de
0	Children's Books	Ste.-Maxime	France	684.396336	5184000	0.000132	2019	6	
1	Graphic Novels	Minneapolis	the United States	12000.000000	2592000	0.004630	2015	8	
2	Apps	London	the United Kingdom	132713.121000	5184000	0.025601	2018	6	
3	Graphic Novels	New York	the United States	20000.000000	2565470	0.007796	2016	10	
5	Drama	Sydney	Australia	72438.903150	2592000	0.027947	2016	8	
...	...	...	...	...	...	...	...	...	
200225	Metal	Victoria	Canada	2993.848400	5184000	0.000578	2015	12	
200226	Faith	Washington	the United States	20000.000000	5011887	0.003991	2019	4	
200228	Small Batch	Tampa	the United States	5000.000000	3888000	0.001286	2018	7	
200229	Textiles	Seattle	the United States	700.000000	2657079	0.000263	2019	6	
200231	Farmer's Markets	Chicago	the United States	19000.000000	2592000	0.007330	2015	3	5

153576 rows × 15 columns

In [23]:
```
ks_meta_2015_2019.to_csv('./data_2015_2019/ks_meta_2015_2019_with_name_blurb_photo_url.csv', index=False, encoding='utf-8')
```

In [42]:
```
# ks_meta_2015_2019_indexed.to_csv('ks_meta_2015_2019_with_index.csv', index=False, encoding='utf-8')
```

In [24]:
```
ks_meta_2015_2019['state'].value_counts()
```

Out[24]:
```
successful    93112
failed        60464
Name: state, dtype: int64
```

In [ ]:
```
# 일단 각 년도별 메타데이터셋에 content crawl한거 붙이고 (index에 맞춰서)
# 그리고 그것들을 concat해버리기
# 굳이 원래 큰 데이터셋에 맞출필요 없잖아?
```

2015~2019 연도별 meta dataset 생성

In [25]:
```
ks_meta_2015 = ks_meta_2015_2019[ks_meta_2015_2019['date_launched_year'] == 2015]
ks_meta_2015.shape
```

Out[25]: (34696, 15)

In [26]:
```
ks_meta_2015.reset_index(drop=True, inplace=True) # index 재정렬
```

In [27]:
```
ks_meta_2016 = ks_meta_2015_2019[ks_meta_2015_2019['date_launched_year'] == 2016]
ks_meta_2016.shape
```

Out[27]: (27413, 15)

In [28]:
```
ks_meta_2016.reset_index(drop=True, inplace=True)
```

In [29]:
```
ks_meta_2017 = ks_meta_2015_2019[ks_meta_2015_2019['date_launched_year'] == 2017]
ks_meta_2017.shape
```

Out[29]: (26866, 15)

In [30]:
```
ks_meta_2017.reset_index(drop=True, inplace=True)
```

In [31]:
```
ks_meta_2018 = ks_meta_2015_2019[ks_meta_2015_2019['date_launched_year'] == 2018]
ks_meta_2018.shape
```

Out[31]: (27398, 15)

In [32]:
```
ks_meta_2018.reset_index(drop=True, inplace=True)
```

In [33]:
```
ks_meta_2019 = ks_meta_2015_2019[ks_meta_2015_2019['date_launched_year'] == 2019]
ks_meta_2019.shape
```

Out[33]: (37203, 15)

In [34]:
```
ks_meta_2019.reset_index(drop=True, inplace=True)
```

연도별 meta dataset 각각 저장

In [70]:
```
# ks_meta_2015.to_csv('./data_2015/ks_meta_2015.csv', index=False, encoding='utf-8')
```

In [71]:
```
# ks_meta_2016.to_csv('./data_2016/ks_meta_2016.csv', index=False, encoding='utf-8')
```

In [72]:
```
# ks_meta_2017.to_csv('./data_2017/ks_meta_2017.csv', index=False, encoding='utf-8')
```

In [73]:
```
# ks_meta_2018.to_csv('./data_2018/ks_meta_2018.csv', index=False, encoding='utf-8')
```

In [74]:
```
# ks_meta_2019.to_csv('./data_2019/ks_meta_2019.csv', index=False, encoding='utf-8')
```

In [ ]:
```

```

crawled된 text파일들 불러오기

In [35]:
```
ks_text_2015 = pd.read_csv('./data_2015/ks_content_2015_recrawled_0_34695.csv')
```

In [36]:
```
ks_text_2015.shape
```

Out[36]: (34696, 3)

In [37]:
```
ks_text_2016 = pd.read_csv('./data_2016/ks_content_2016_recrawled_0_27412.csv')
```

In [38]:
```
ks_text_2016.shape
```

Out[38]: (27413, 3)

In [39]:
```
ks_text_2017 = pd.read_csv('./data_2017/ks_content_2017_recrawled_0_26865.csv')
```

In [40]:
```
ks_text_2017.shape
```

Out[40]: (26866, 3)

In [41]:
```
ks_text_2018 = pd.read_csv('./data_2018/ks_content_2018_recrawled_0_27397.csv')
```

In [42]:
```
ks_text_2018.shape
```

Out[42]: (27398, 3)

In [43]:
```
ks_text_2019 = pd.read_csv('./data_2019/ks_content_2019_recrawled_0_37202.csv')
```

In [44]:
```
ks_text_2019.shape
```

Out[44]: (37203, 3)

In [ ]:
```

```

각 연도별로 meta data와 text data 병합하기

- ks_meta_2015 + ks_text_2015

In [45]:
```
pd.concat([ks_meta_2015, ks_text_2015], axis=1).head(2)
```

Out[45]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_delay_time	d
0	Graphic Novels	Minneapolis	the United States	12000.0	2592000	0.004630	2015	8	871654	
1	Architecture	New York	the United States	500.0	2588400	0.000193	2015	2	1407876	

In [46]:
```
# 2015년 병합
meta_text_merged_2015 = pd.concat([ks_meta_2015, ks_text_2015.set_index('index')], axis=1)
meta_text_merged_2015.head(2)
```

Out[46]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_delay_time	d
0	Graphic Novels	Minneapolis	the United States	12000.0	2592000	0.004630	2015	8	871654	
1	Architecture	New York	the United States	500.0	2588400	0.000193	2015	2	1407876	

In [116]:
```
meta_text_merged_2015.to_csv('./data_2015/meta_text_merged_2015.csv', index=False, encoding='utf-8')
```

In [47]:
```
# 2016년 병합
meta_text_merged_2016 = pd.concat([ks_meta_2016, ks_text_2016.set_index('index')], axis=1)
meta_text_merged_2016.head(2)
```

Out[47]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_delay_time
0	Graphic Novels	New York	the United States	20000.00000	2565470	0.007796	2016	10	2414894
1	Drama	Sydney	Australia	72438.90315	2592000	0.027947	2016	8	1698096

In [117]:
```
meta_text_merged_2016.to_csv('./data_2016/meta_text_merged_2016.csv', index=False, encoding='utf-8')
```

In [48]:
```
# 2017년 병합
meta_text_merged_2017 = pd.concat([ks_meta_2017, ks_text_2017.set_index('index')], axis=1)
meta_text_merged_2017.head(2)
```

Out[48]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_delay_time	d
0	Drinks	Raleigh	the United States	25000.0	2592000	0.009645	2017	11	4170124	
1	Drama	Orlando	the United States	10000.0	2592000	0.003858	2017	7	14353763	

In [118]:
```
meta_text_merged_2017.to_csv('./data_2017/meta_text_merged_2017.csv', index=False, encoding='utf-8')
```

In [49]:
```
# 2018년 병합
meta_text_merged_2018 = pd.concat([ks_meta_2018, ks_text_2018.set_index('index')], axis=1)
meta_text_merged_2018.head(2)
```

Out[49]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_delay_time
0	Apps	London	the United Kingdom	132713.121	5184000	0.025601	2018	6	1021832
1	Art Books	Denver	the United States	5000.000	2595600	0.001926	2018	10	145454

In [119]:
```
meta_text_merged_2018.to_csv('./data_2018/meta_text_merged_2018.csv', index=False, encoding='utf-8')
```

In [50]:
```
# 2019년 병합
meta_text_merged_2019 = pd.concat([ks_meta_2019, ks_text_2019.set_index('index')], axis=1)
meta_text_merged_2019.head(2)
```

Out[50]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_delay_tim
0	Children's Books	Ste.-Maxime	France	684.396336	5184000	0.000132	2019	6	554
1	Apps	Pittsburgh	the United States	30000.000000	2592000	0.011574	2019	8	1238439

In [120]:
```
meta_text_merged_2019.to_csv('./data_2019/meta_text_merged_2019.csv', index=False, encoding='utf-8')
```

In [51]:
```
meta_text_merged_2019.shape
```

Out[51]:
```
(37203, 17)
```

meta data 그리고 text data 병합한 연도별 파일들 세로로 병합(concat)

In [52]:
```
meta_text_dataset = pd.concat([meta_text_merged_2015, meta_text_merged_2016, meta_text_merged_2017, meta_text_merged_2018, meta_text_merged_2019], ignore_index=True)
meta_text_dataset
```

Out[52]:

	category_name	location_name	country_displayable_name	usd_goal	set_fundraising_period	target_goal_period_rate	date_launched_year	date_launched_month	launching_dela
0	Graphic Novels	Minneapolis	the United States	12000.000000	2592000	0.004630	2015	8	
1	Architecture	New York	the United States	500.000000	2588400	0.000193	2015	2	1
2	Gaming Hardware	Oshkosh	the United States	10000.000000	3884400	0.002574	2015	2	
3	Drama	Manchester	the United Kingdom	998.226229	2306670	0.000433	2015	3	
4	Flight	South Florida	the United States	17500.000000	3020400	0.005794	2015	3	1
...	...	...	...	...	...	...	...	...	
153571	Accessories	Sydney	Australia	393.166059	2592000	0.000152	2019	3	
153572	Accessories	London	the United Kingdom	1305.500070	2592000	0.000504	2019	4	
153573	Web	Prague	Germany	7893.879210	1728000	0.004568	2019	4	
153574	Faith	Washington	the United States	20000.000000	5011887	0.003991	2019	4	2
153575	Textiles	Seattle	the United States	700.000000	2657079	0.000263	2019	6	2

153576 rows × 17 columns

In [53]:
```
len(ks_meta_2015_2019) # 데이터 수 일치
```

Out[53]:
```
153576
```

In [54]:
```
# 파일 저장
# meta_text_dataset.to_csv('./data_2015_2019/meta_text_dataset_2015_2019.csv', index=False, encoding='utf-8')
```

In [ ]:
```
meta_text_dataset.to_csv('./data_2015_2019/meta_text_dataset_ordered_2015_2019_with_photo_url.csv', index=False, encoding='utf-8')
```

meta와 text(name, blurb 포함) 분리

In [128]:
```
text_dataset = meta_text_dataset[['name','blurb','content','risk_challenge']]
```

In [133]:
```
# text_dataset.to_csv('./data_2015_2019/text_dataset.csv', index=False, encoding='utf-8')
```

In [132]:
```
meta_dataset = meta_text_dataset.drop(columns=['name','blurb','content','risk_challenge'])
```

In [134]:
```
# meta_dataset.to_csv('./data_2015_2019/meta_dataset.csv', index=False, encoding='utf-8')
```

In [ ]:
```

```