Project Report

Kickstarter Project Analytics

A NLP Approach to Increase the Chance of Building a Successful Crowdfunding Project

Date: 12th April 2021

1. Introduction

1.1 Description of the Project

Kickstarter is a global crowdfunding platform for projects. Anyone can submit any creative projects varying from music, art, design to technology. Once the project initiator sets the project's funding goal and deadline, interested parties can then pledge money to make the project happen. At the end of the crowdfunding period, projects can either be successfully funded by reaching/ exceeding their goal or fail by not meeting their goal. For successful projects, 5% of the goal will be collected by Kickstarter as part of the platform fee. No charge will be made for projects that fail, and all funds will be returned to the backers.

1.2 The Business Problem

According to Kickstarter, **38.61% of projects fail** (Kickstarter, 2021)**.** That amounts to an estimate of USD **229,775,700 in opportunity cost** for the Kickstarter management team as of today.

1.3 Our Solution

Our goal for this project is to utilize text mining techniques to help Kickstarter fundraisers to increase their chances of success for their projects, and to help reduce the opportunity cost for the Kickstarter management team. The business use cases are as follows,

1) Fundraisers will be provided with the sentiments of backers towards project categories of their choice.
2) Fundraisers will be recommended with related and similar projects.

1.4 Our Datasets

Two main datasets were used for this project - one collected from WebRobots and another collected from Kickstarter by the team. For WebRobots Kickstarter, the data is crawled and published once a month, and includes short project descriptions, location, the amount raised, goal, and project status. Our team has also crawled data from the Kickstarter website itself using Octoparse for project comments from the Kickstarter projects.

The final datasets were combined through project names and there are **605 projects and 25158 comments.** Please refer to Appendix II for the data columns. The team used the Langid library to identify the language of the comments.

1.5 The Stakeholders

One of the potential users would be aspiring project starters who aim to raise funds successfully. Using the various analytics conducted, fundraisers could use these to investigate which topics the backers are interested in or attracted to, evaluate their current projects or even initiate projects that are often preferred by the backers. They can further use these analytics to facilitate their research and increase their chance of building successful projects, instead of having to conduct manual research on their own.

The Kickstarter management team would also be able to benefit by using our tool to fine-tune their algorithm to help fundraisers, which will directly help to improve their revenue at the same time.

2. Solution Overview

Techniques	Goal
Topic Modelling	To identify trending topics and keywords.
Project Topic Classification	To help recommend the optimal tags for a project to increase viewership.
Sentiment Analysis	To find out the general attitude and keywords towards different types of projects.
Content-based Recommender	To recommend similar or related projects to fundraisers.

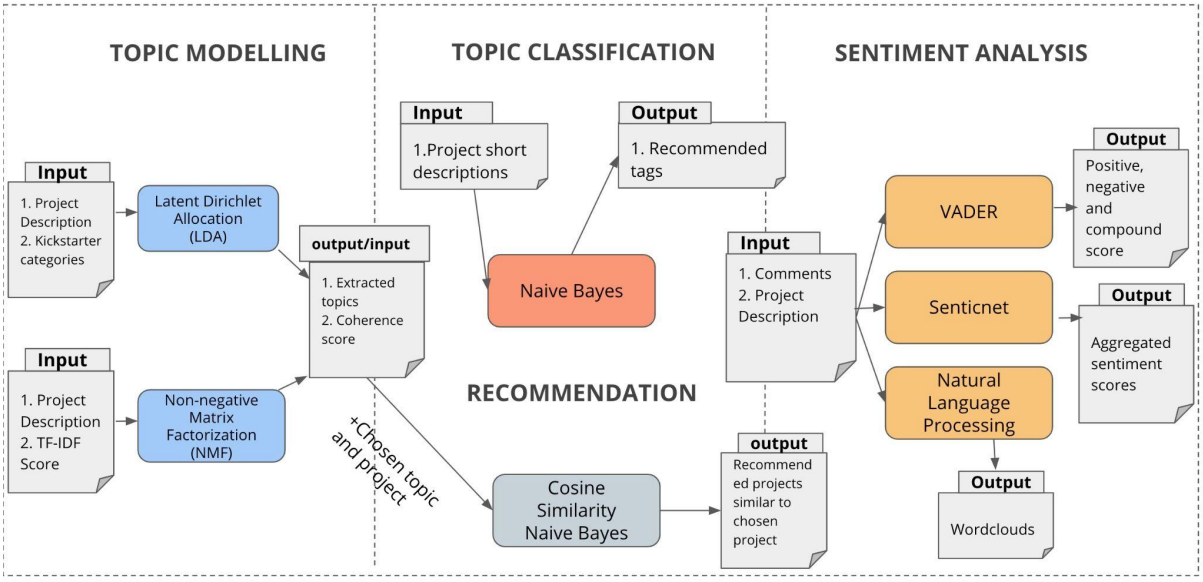


Figure 1: Solution Overview Diagram

2.1 Topic Modelling

In this analysis, our team used each of the project's description and title to identify new key trending topics and keywords for prospective fundraisers who want to start a new project. There were two approaches used - Latent Dirichlet Allocation (LDA) approach and Non-negative Matrix Factorization (NMF). LDA is an unsupervised topic modelling approach on documents using project description and Kickstarter categories from the dataset. On the other hand, NMF is the application of TF-IDF vector to documents and model topics using project description and TF-IDF scores

2.2 Project Topic Classification

This analysis makes use of project short descriptions to optimally recommend categories/ tags for prospective fundraisers on their new projects. For this section, the Naive Bayes Classification Technique was used. 100 documents for each category were used to train the model. Then, 20 unique documents per category (20% volume) were used to validate the model. The model was validated with 99.28% accuracy. Lastly, the model was tested by using 10 unseen unique documents per category (10% volume).

2.3 Sentiment Analysis

Valence Aware Dictionary for Sentiment Reasoning (VADER) and SenticNet models are used and compared in sentiment analysis on project comments and short descriptions. Word clouds are used to visualise the most frequent words for different groups of projects. We use sentiment analysis to draw business insights in a top down manner.

<table>
<tr><th>Step</th><th>Approach</th><th>Project Comments</th><th>Short Project Description</th><th>Goal</th></tr>
<tr><td>1</td><td>Sentiment Analysis</td><td colspan="2">To:
1. Find out the sentiment scores on project level;
2. Classify the projects based on their sentiment score and pass them to word cloud.</td><td rowspan="2">Derive business insights</td></tr>
<tr><td>2</td><td>Word Cloud</td><td>Visualise the most frequent words used by commenters based on different groups of projects such as categories.</td><td>Visualise the most frequent words used by projects with negative sentiment scores in project description</td></tr>
</table>

2.4 Content-based Recommender

This analysis is to recommend similar and related projects to fundraisers to assist their research on their new projects. This analysis was based on topic modeling, grouping projects into different groups and then applying TF-IDF cosine similarity to find related projects. After applying the cosine similarity, a comparison was done between normal cosine similarity and topic based cosine similarity

3. Solution Details

3.0 Exploratory Data Analysis

By grouping the comments by project id, it is noticed that projects receive varying amounts of comments from 1 to 1700. Most projects in our datasets are successful and the majority are from the US, the UK, Hong Kong, Germany and Canada. Category ‘technology, 3D printing’ receives the most comments, and ‘art, illustration’ has the highest number of projects in our dataset. We also noticed that for those projects that raised more than their goals, the top five countries are the US, the UK, Canada, Australia and Hong Kong, which is different from that for all the projects. Please refer to Appendix I for the charts.

3.1 Project Topic Modelling using LDA

	name	category	blurb
0	Albert's Cookie- A Children's Book	{"id":46,"name":"Children's Books","slug":"pub...	find happens albert decides play food takes hi...
1	SNEEKY PEEK presents a Picturesque Guide Book ...	{"id":349,"name":"Letterpress","slug":"publish...	following success hastings leonard guide book ...
2	REACHING the STARS with SpaceTripreneurs	{"id":49,"name":"Periodicals","slug":"publishi...	join amazing journey impacting million children
3	Unfettered Hexes: Queer Tales of Insatiable Da...	{"id":324,"name":"Anthologies","slug":"publish...	anthology queer witchery
4	Walking Into Winter: Writing to Warm the Frost...	{"id":50,"name":"Poetry","slug":"publishing/po...	written chronically ill girl commenting ups do...
...	...	...	...
14492	Bug Free Monkey Hut	{"id":258,"name":"Architecture","slug":"design...	inspired burning man building waterproof pvc p...
14493	Love Story, Palestine	{"id":6,"name":"Dance","slug":"dance","positio...	international collaboration yoshiko chuma scho...
14494	UPside Dance Presents 'Strum' with Composer Ma...	{"id":254,"name":"Performances","slug":"dance/...	strum vibrant dance performance celebrating co...
14495	AGNI: luxury soy wax candles that give back	{"id":343,"name":"Candles","slug":"crafts/cand...	want support people experiencing stress anxiet...
14496	Spreading hope through dance to Rwandan street...	{"id":254,"name":"Performances","slug":"dance/...	bringing contemporary dance street children rw...

Figure 2: Data before preprocessing

3.1.1 Pre-processing

The dataset from the web robots kickstart dataset was used for the LDA model. The name, category and short description (blurb) were extracted from the dataset. In this model, non-English rows were removed and three preprocessing procedures were conducted. Firstly, the words were tokenized and converted to lowercase. A stopword list was then used to remove the stopwords. Lastly stemming and lemmatization were used to generate the root form of the words.

3.1.2 Creating Bigram Models

We built a Bigram model in order to allow our model to perform better and for it to identify phrases. For the bigram model, we only passed nouns, adjectives, verbs and adjectives so that the LDA model better clusters the topics.

3.1.3 Creating Dictionary and Corpus

```
[[(id2word[id], freq) for id, freq in cp] for cp in corpus[:1]]
```

```
[[('adventure', 1),
  ('albert', 1),
  ('decide', 1),
  ('find', 1),
  ('flying', 1),
  ('food', 1),
  ('happen', 1),
  ('high', 1),
  ('outer', 1),
  ('play', 1),
  ('space', 1),
  ('take', 1)]]
```

Figure 3: A dictionary with word id and its frequency

A unique id for each word in the document. The unique words were added to the dictionary. The corpus was also populated with mapping of each word (word_id, word_frequency). These inputs will be used later in the LDA model section.

3.1.4 Choosing LDA Model

There were many implementations of the LDA models that were represented to us during our research. The algorithms vary in these models and it can be noted Mallet is better as it uses Gibbs Sampling which is more precise than other models like Gensim's LDA models which uses Variational Bayes.

For our LDA models, we decided to do a comparison between Gensim's LDA Mallet and LDA models to evaluate which one is more suitable to use. Mallet is a Java-based console application for language processing while Gensim is a python package based on numpy and scipy packages. After tweaking the hyperparameters and topics, the LDA mallet still produced better results. For instance, at 50 topics, Gensim LDA had a coherence score of 0.37742211926646135 while Mallet LDA had a coherence score of 0.39238381934372023, reflecting a slightly better result than Gensim LDA.

```
[(32,
  '0.000*"kristina" + 0.000*"apollonia" + 0.000*"dobrowo" + 0.000*"holzer" + '
  '0.000*"phe" + 0.000*"olga" + 0.000*"mensah" + 0.000*"clement" + '
  '0.000*"harris" + 0.000*"wilson"'),
 (30,
  '0.016*"way" + 0.014*"come" + 0.003*"idea" + 0.001*"bloodbath" + '
  '0.000*"mayhem" + 0.000*"ornament" + 0.000*"birthstone" + 0.000*"lingerie" + '
  '0.000*"birthday" + 0.000*"lego"'),
 (8,
  '0.000*"succulent" + 0.000*"orchard" + 0.000*"geometric" + 0.000*"planter" + '
  '0.000*"flesh" + 0.000*"orange" + 0.000*"tangerine" + 0.000*"blood" + '
  '0.000*"unveil" + 0.000*"ass"'),
 (27,
  '0.000*"æra" + 0.000*"organic_cotton" + 0.000*"basket" + 0.000*"catscratch" '
  '+ 0.000*"narcissus" + 0.000*"envelop" + 0.000*"grapple" + 0.000*"haunted" + '
  '0.000*"olga" + 0.000*"mensah"'),
 (39,
  '0.000*"æra" + 0.000*"organic_cotton" + 0.000*"basket" + 0.000*"catscratch" '
  '+ 0.000*"narcissus" + 0.000*"envelop" + 0.000*"grapple" + 0.000*"haunted" + '
  '0.000*"olga" + 0.000*"mensah"'),
```

Figure 4: Gensim Generated Topics

```
[(48,
  [('bring', 0.19208573784006594),
   ('life', 0.106760098928277),
   ('back', 0.07625721352019786),
   ('order', 0.03297609233305853),
   ('plan', 0.02967848309975268),
   ('baby', 0.025556471558120363),
   ('big', 0.015663643858202802),
   ('vision', 0.015251442704039572),
   ('dead', 0.014014839241549877),
   ('dedicated', 0.013602638087386645)]),
 (31,
  [('film', 0.18560338743824983),
   ('short', 0.07374735356386733),
   ('feature', 0.07374735356386733),
   ('horror', 0.057515878616796605),
   ('movie', 0.0497529992942837),
   ('comedy', 0.0356386732533521 54),
   ('documentary', 0.03211009174311927),
   ('animate', 0.030345800988002825),
   ('female', 0.0222300635144671 85),
   ('short_film', 0.02117148906139732)]),
 (22,
  [('pin', 0.07665260196905767),
   ('set', 0.07419127988748242),
   ('collection', 0.06188466947960619),
   ('inspire', 0.05379746835443038),
   ('feature', 0.05063291139240506),
   ('enamel_pin', 0.04008438818565401),
   ('theme', 0.03691983122362869),
   ('cat', 0.03410689170182841),
   ('animal', 0.03270042194092827),
   ('cute', 0.02180028129395218)]),
 (16,
  [('day', 0.0947455523376086),
   ('modern', 0.05585436491518411),
   ('create', 0.04757964418700869),
   ('social', 0.0289615225486139 84),
   ('tea', 0.02813405047579644),
   ('blend', 0.026479106330161355),
   ('traditional', 0.0244104261481175),
   ('night', 0.022341745966073644),
   ('classic', 0.022341745966073644),
   ('twist', 0.022341745966073644)]),
```

Figure 5: Mallet Generated Topics

Upon physical examination of the generated topics, Mallet also produced better results and the words were more interrelated to one another. The LDA mallet model was thus selected for further tuning and was used for the final comparison against the NMF model. However, it can be noted that Mallet does not offer as much tweaking and flexibility in the steps like Gensim. Given more computation resources and further tweaking of hyperparameters, Gensim may produce better results.

3.1.5 Selecting optimal number of topics for further tuning

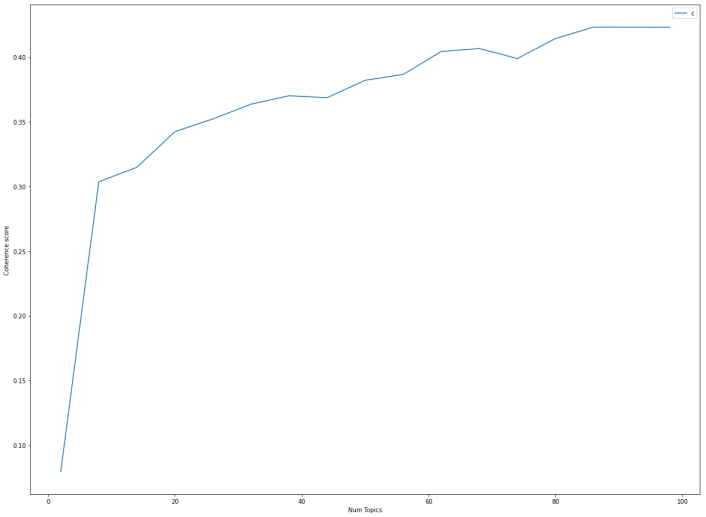


Figure 6: Coherence vs Number of Topics

The optimal number of topics were first chosen through the coherence measure. By testing the number of topics from 0 to 100, we found that topics 40-50 produced the best coherence score. Even though a higher number of topics like 60 topics can produce a higher coherence score and more subtopics, the same words started to appear across multiple topics and some of the topic words were not related to each other.

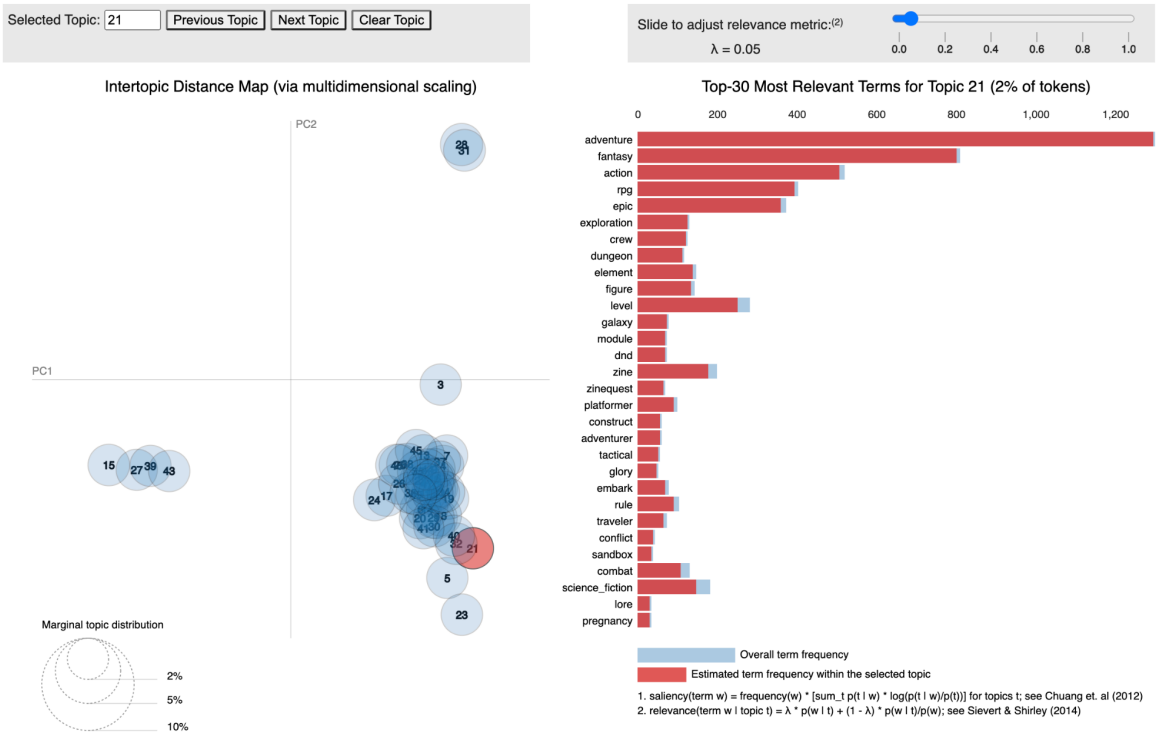


Figure 7: Intertopic Distance, pyLDAvis (50 Topics)

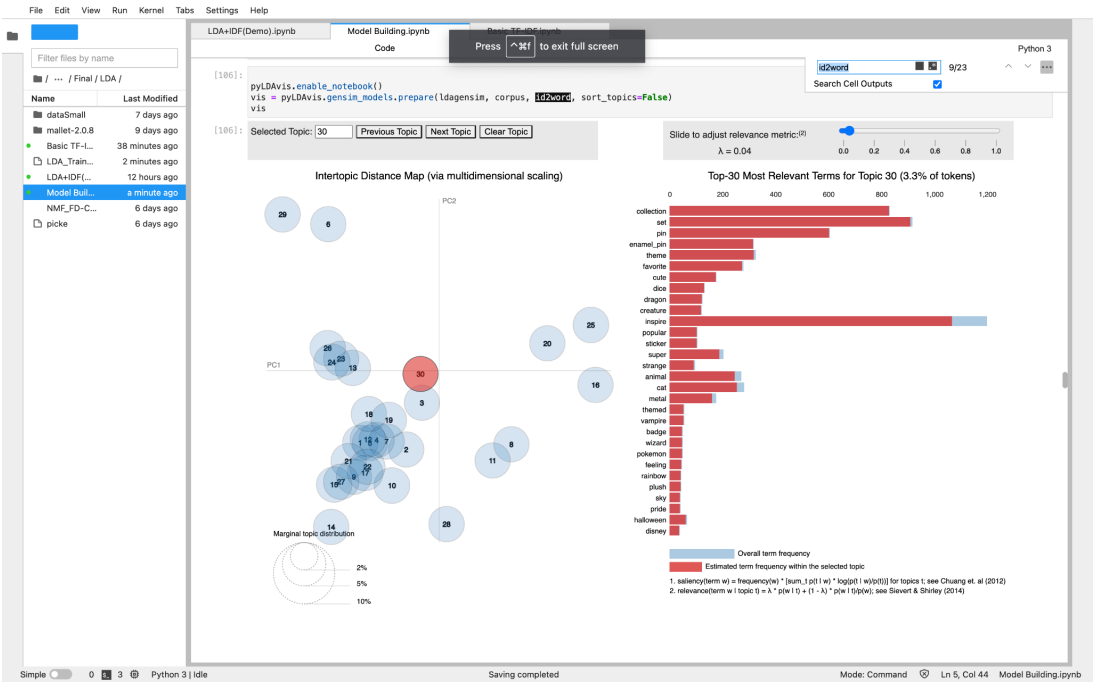


Figure 8: Intertopic Distance, pyLDAvis (40 Topics)

Furthermore, pyLDAvis was used to visualize topic relationships. The circles each represent a specific topic number and the distance between them is the topic relatedness. These are mapped through dimensionality reduction (PCA/t-sne) on distances between each topic's probability distributions into 2D space. From this, we found that 40 topics is the better number of topics even though it has a lower coherence score as the circle overlap is minimized.

3.2 Project Topic Modelling using NMF

	backers_count	blurb	category	converted_pledged_amount	country	country_displayable_name	created_at	
0	13	'Sects' is a comedy webseries - about two girl...	{"id":33,"name":"Webseries","slug":"film & vid...	556	US	the United States	1426711317	{"id":116751006
1	207	Terrence McNally's pioneering 5-decades in the...	{"id":30,"name":"Documentary","slug":"film & v...	42766	US	the United States	1489540002	{"id":1365
2	0	A romantic Comedy of falling for your true lov...	{"id":300,"name":"Romance","slug":"film & vide...	0	US	the United States	1464218765	{"id":72893975
3	0	I	{"id":301,"name":"Science Fiction","slug":"fil...	0	US	the United States	1466370982	{"id":57567281
4	70	It's hard being Gavin. Especially if you're Ga...	{"id":292,"name":"Comedy","slug":"film & video...	23404	AU	Australia	1475582845	{"id":17880562

Figure 9: Data before processing

3.2.1 Pre-processing

Similar to LDA, WebRobots kickstarter dataset was used for the NMF model. To compare with the LDA model, we first filtered out non-English corpus and then proceeded with removing the stop words using NLTK stop words library. We then tokenized the remaining words and with the remaining words we stemmed and lemmentized them to prepare for the NMF model. Below shows the code as well as the processed text after running through all the functions (shown in the second picture with header processed text).

```
def preprocess(sentence):
    sentence=str(sentence)
    sentence = sentence.lower()
    sentence=sentence.replace('{html}',"")
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, '', sentence)
    rem_url=re.sub(r'http\S+', '',cleantext)
    rem_num = re.sub('[0-9]+', '', rem_url)
    tokenizer = RegexpTokenizer(r'\w+')
    tokens = tokenizer.tokenize(rem_num)
    filtered_words = [w for w in tokens if len(w) > 2 if not w in stopwords.words('english')]
    stem_words=[stemmer.stem(w) for w in filtered_words]
    lemma_words=[lemmatizer.lemmatize(w) for w in stem_words]
    return " ".join(filtered_words)
df['blurb']=df['blurb'].map(lambda s:preprocess(s))



# use langid lib to do language detection
df['language'] = df['blurb'].apply(lambda x: langid.classify(x)[0])
df = df[df['language'] == 'en']
```

...	spotlight	staff_pick	state	state_changed_at	static_usd_rate	urls	usd_pledged	usd_type	language	processed_text
...	True	False	successful	1428510545	1.000000	{"web": {"project":"https://www.kickstarter.com...	556.000000	international	en	[sect, comedi, webseri, girl, accident, join, ...

Figure 10: Pre-Processing Codes

3.2.2 Coherence score

Before moving on to the models, we used coherence scores to identify the optimal number of topics that could be used to run the model to get the optimal result when executing topic extraction. To run the coherence score, we tapped on Gensim library for the coherence model and passed in the processed text to create a dictionary which maps the word to their integer ID. We set a range from 5 to 75 for the number of topics. The NMF model was then run and placed into the coherence model to generate the best amount of topic which would be used in later sections.

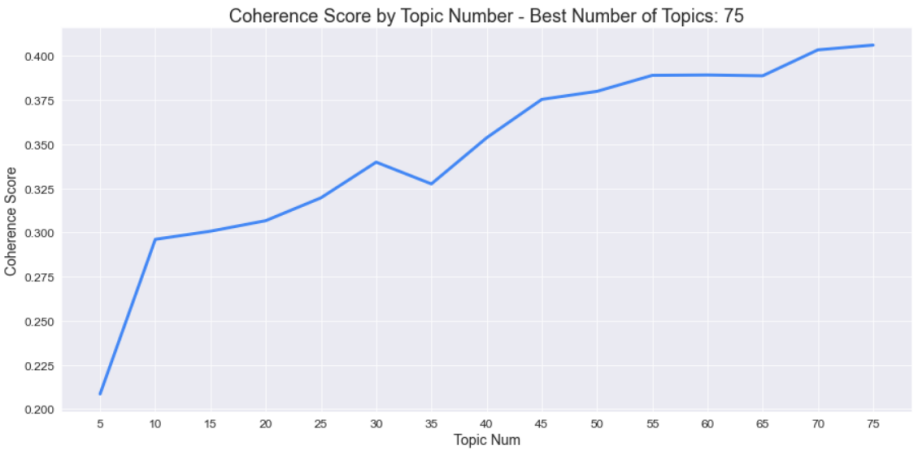


Figure 11: Coherence Score by Topic Number

Based on the graph generated above, we then observed that the best topic to use is 75 with a coherence score of 0.42 based on the results. We would apply this to the NMF model later for the best results. Results have a steady increase after 40 topics, this gives us more possibility to look at more topics beyond maximum of 75 which might future improve the model.

3.2.3 TF-IDF vector

Once we had prepared the corpus, we proceeded to vectorize the TF-IDF based on the word that was tokenized. This TF-IDF weight that was generated will be placed through a TF-IDF vectorizer using SK-learn library. Based on the coherence score generated and the processed text, we then created a TF-IDF to be used in the NMF model. Once we have the TF-IDF, we then put it through the transformer to be placed in a vector.

```
  :     :
(0, 726)     0.39078071357201066   (5897, 685)   0.19738389083681937
(0, 922)     0.5343612679700439    (5897, 687)   0.3250248879721085
(0, 1671)    0.3709616369757349    (5897, 910)   0.29786976955995964
(0, 2114)    0.3868348445846099    (5897, 1585)  0.1795399288874588
(0, 4303)    0.5239224523735175    (5897, 2190)  0.2260755416619342
(1, 60)      0.2776201319885188    (5897, 2283)  0.2284996067982979
(1, 104)     0.2750951463787931    (5897, 2284)  0.29786976955995964
(1, 984)     0.380854896097702     (5897, 2830)  0.25021559163871043
(1, 1284)    0.2585318203985653    (5897, 3091)  0.303101069463003
(1, 2023)    0.20087280696225268   (5897, 3119)  0.254829925174061
(1, 2226)    0.36848358474984383   (5897, 3324)  0.316282892435244466
(1, 2923)    0.38844318327911054   (5897, 3730)  0.18834169034692513
(1, 3314)    0.3045245258306988    (5898, 28)    0.2701862417506428
(1, 3814)    0.31261257798445535   (5898, 769)   0.2317449320879573
(1, 3973)    0.34333746449203983   (5898, 802)   0.2929636247535959
(2, 409)     0.6249202276512127    (5898, 1205)  0.30946679400081367
(2, 726)     0.18360063135321433   (5898, 1230)  0.2965051464583692
(2, 1362)    0.20440601582448192   (5898, 1312)  0.3285967987413937
(2, 1671)    0.5228670842069295    (5898, 1609)  0.15075230365621065
(2, 2303)    0.22900589623273296   (5898, 1623)  0.3376791481720166
(2, 2309)    0.13965364228638275   (5898, 2225)  0.2356996738237046
(2, 2439)    0.18235278154583856   (5898, 2424)  0.3211759789247988
(2, 3342)    0.2264592501031253    (5898, 2948)  0.234071383757015572
(2, 4002)    0.26395944558400103   (5898, 3126)  0.24008043811596333
(2, 4096)    0.18556926487370023   (5898, 3127)  0.2965051464583692
```

Figure 12: TF-IDF for NMF model

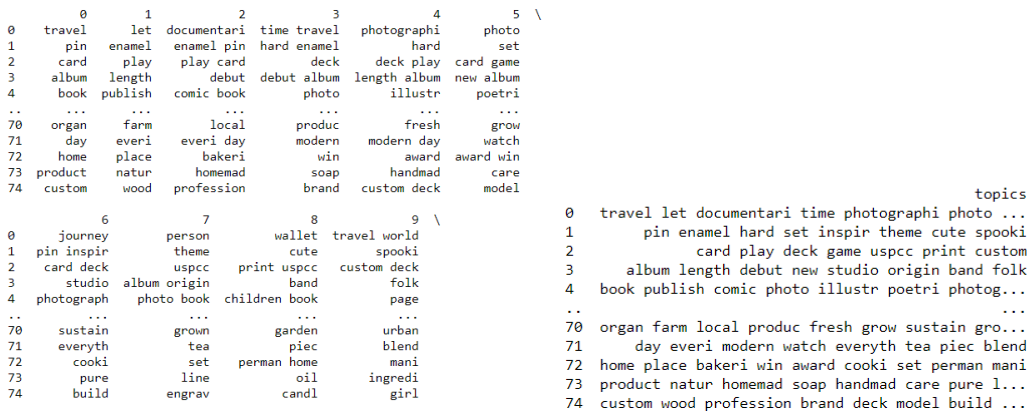
3.2.4 NMF model

Having created the TF-IDF vector, we proceed to combine the TF-IDF vector with the best number of topics. This generates the score of the different clusters which is also a vector that will then be combined with the topics to generate the list of top words in the topics. This would then allow us to extract a list of words in the topics and the number of topics that is generated from the coherence score.

```
# Use the top words for each cluster by tfidf weight
# to create 'topics'

# Getting a df with each topic by document
docweights = nmf.transform(tfidf_vectorizer.transform(texts))

n_top_words = 10

topic_df = topic_table(
    nmf,
    tfidf_fn,
    n_top_words
).T

# Cleaning up the top words to create topic summaries
topic_df['topics'] = topic_df.apply(lambda x: [' '.join(x)], axis=1) # Joining each word into a list
topic_df['topics'] = topic_df['topics'].str[0]  # Removing the list brackets
topic_df['topics'] = topic_df['topics'].apply(lambda x: whitespace_tokenizer(x)) # tokenize
topic_df['topics'] = topic_df['topics'].apply(lambda x: unique_words(x))  # Removing duplicate words
topic_df['topics'] = topic_df['topics'].apply(lambda x: [' '.join(x)])  # Joining each word into a list
topic_df['topics'] = topic_df['topics'].str[0]  # Removing the list brackets


topic_df.head()
print(topic_df)
```

Figure 13: Codes Combining TF-IDF with best number of Topics

We first extracted the topics based on the weight from our TF-IDF transform vector which would be used later. We then created a table based on the NMF model, the TF-IDF with its feature name, and the number of words we wanted in each topic. In this case we chose 10 to match the LDA model.



	0	1	2	3	4	5
0	travel	let	documentari	time travel	photographi	photo
1	pin	enamel	enamel pin	hard enamel	hard	set
2	card	play	play card	deck	deck play	card game
3	album	length	debut	debut album	length album	new album
4	book	publish	comic book	photo	illustr	poetri
..	...	...	...	...	...	...
70	organ	farm	local	produc	fresh	grow
71	day	everi	everi day	modern	modern day	watch
72	home	place	bakeri	win	award	award win
73	product	natur	homemad	soap	handmad	care
74	custom	wood	profession	brand	custom deck	model

	6	7	8	9
0	journey	person	wallet	travel world
1	pin inspir	theme	cute	spooki
2	card deck	uspcc	print uspcc	custom deck
3	studio	album origin	band	folk
4	photograph	photo book	children book	page
..	...	...	...	...
70	sustain	grown	garden	urban
71	everyth	tea	piec	blend
72	cooki	set	perman home	mani
73	pure	line	oil	ingredi
74	build	engrav	candl	girl

	topics
0	travel let documentari time photographi photo ...
1	pin enamel hard set inspir theme cute spooki
2	card play deck game uspcc print custom
3	album length debut new studio origin band folk
4	book publish comic photo illustr poetri photog...
..	...
70	organ farm local produc fresh grow sustain gro...
71	day everi modern watch everyth tea piec blend
72	home place bakeri win award cooki set perman mani
73	product natur homemad soap handmad care pure l...
74	custom wood profession brand deck model build ...

Figure 14: Table based on NMF model

Based on the screenshot above, you can see some of the topics that the NMF model produced, with topic 2 on card games and topic 3 on music album, most of the topics easily tell us what is about by looking at the 10 words.

After creating a list of words for the topics, we proceeded to combine it with our corpus to help us determine which topic the project belongs to based on the short description that was provided. We joined the topic based on the url of the project as the urls are unique and act as an identifier during the joining process.

We can produce the matrices of the 3 components generated by the NMF to see the vector size we are working with.

3.2.5 Residuals

Residuals was run to help us compare the quality of each topic. This shows the topic that most closely matches the project description. We can scale our analysis based on these stronger topics as well.

We first got the sum of residual squares to get a rough idea of the average strength of all the topics. This was done by taking the Frobenius norm of the TF-IDF weight and subtracting it from the dot product of coefficients of the topics and the topics. Once we obtained that, we moved on to plot the graph to show the topics which are the strongest.

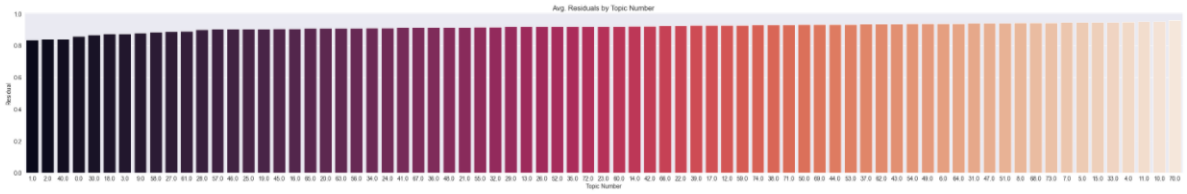


Figure 15: A graph showing the strongest topics

Based on the graph generated from the code above, we can see that the top 10 strongest topics are 1, 2, 40, 0, 30, 16, 3, 9, 58, 27. The lower the residual score, the stronger the topics which depict the description. From this we can choose these 10 topics generated for future projects.

3.2.6 Challenges

The challenge posed using LDA is choosing the number of topics and finding the right hyperparameters to be applied in the model. For the NMF, the challenge was choosing the right number of topics and setting TF-IDF to work with NMF.

3.3 Project Topic Classification

3.3.1 Details:

We started by preprocessing the data to remove all symbols and lower the cases of all the data.

The initial category data consists of:

{"id":25,"name":"Sculpture","slug":"art/sculpture","position":10,"parent_id":1,"parent_name":"Art","color":16760235,"urls":{"web":{"discover":"http://www.kickstarter.com/discover/categories/art/sculpture"}}}

After cleaning:

```
art sculpture
```

We removed stop words, stemmed, and lemmatized the words for each column as well.

For testing purposes, we utilized the general categories found in our current dataset and chose them as our labels. There were 7 in total.

```
['art', 'comics', 'film', 'music', 'photography', 'publishing', 'technology']
```

We then loaded the processed documents and created a dictionary:

```
Dictionary(11367 unique tokens: ['ancient', 'announc', 'architectur', 'art', 'artist']...)
```

After which, we prepared the labeled training dictionary and performed Naïve Bayes Classification, and the trained classifier was able to correctly predict the training document accurately.

We then validated the model with 20 unique documents per category (20% volume):

The validation returned an accuracy of **99.29%**.

Lastly, we tested the model by utilizing 10 unique documents per category (10% volume):

```
...
art_2_000008.txt ⟶ art and art (Single tag prediction)
art_2_000009.txt ⟶ art and art
art_2_000010.txt ⟶ art and comics  (Multi-tag prediction)
comics_10_000001.txt ⟶ comics and publishing
comics_10_000002.txt ⟶ publishing and comics
...
```

3.3.2 Issues encountered and resolution

There were non-English short descriptions found in the CSV file itself and we removed them from the CSV. There was overfitting present during the initial run for the validation. We discovered this issue when we fitted more datasets into the validation model as the validation accuracy fell from 100% to 1.36%. The issue was primarily due to insufficient documents loaded for training and validation, despite having thousands of records within that document. That was easily rectified by splitting the single large document into multiple smaller documents to properly train the model, and also providing sufficient document split to validate and test.

3.3.3 How does the solution address the business problem?

In essence, this solution helps fundraisers accurately recommend tags based on successful projects' short descriptions in specific categories. By using the appropriate tags, the solution could help them gain more viewership, which in turn, increases their chances of success as people are more likely to view and back their projects. This also lowers the opportunity costs associated with failed projects for the Kickstarter management team.

3.4 Sentiment Analysis

Through sentiment analysis, fundraisers are able to find out the general sentiments towards all projects and different categories of projects.

3.4.1 Detailed Approach - Model Selection

Preliminary research is done to find out the most suitable models for sentiment analysis for our datasets. Given the nature of the datasets, being both short online reviews and one that contains many Internet languages such as emoticons and slangs, we chose VADER and SenticNet due to their better performance and easy access (Ribeiro et al., 2016). VADER uses a combination of sentiment lexicon (phrases, emoticons, acronyms) and grammatical rules (Ma, 2020). It returns sentiment scores that include positive, negative, neutral and compound scores[1] on the sentence level. SenticNet adopts a concept-level sentiment analysis approach and it focuses on semantic analysis through the use of semantic networks (Lowry-Duda, 2020). It returns polarity scores on the word level. Hence, we created functions to calculate the average scores to find out the sentence sentiment score for SenticNet, and for both models on the project level. For SenticNet, functions are created to lemmatise and stem the input words using NLTK to return a bag of words (for example, ('amazing', 'amaz')) so that they can be matched with the SenticNet database. If the word is not found in the SenticNet library, it is ignored.

```
from nltk.stem import WordNetLemmatizer
from nltk.stem.snowball import EnglishStemmer
import re
from senticnet.senticnet import SenticNet

def get_words_bag(message):
    wnl = WordNetLemmatizer()
    es = EnglishStemmer()
    for word in re.finditer("\w+", message):
        yield (wnl.lemmatize(word.group()), es.stem(word.group()))
```

```
def get_avg_polarity(message):
#     threshold = 0.3
    sn = SenticNet()
    count = 0
    summ = 0
    for word_options in get_words_bag(message):
        polarity = 0
        for word in word_options:
            try:
#                 print(sn.concept(word))
                concept = sn.concept(word)
                polarity = float(concept['polarity_value'])
                break
            except:
                pass #Do next
#         if abs(polarity) > threshold:
        summ += polarity
        count += 1

    if count == 0:
        return 0
    return summ / count
```

Figure 16: Functions used to calculate the sentiment score from SenticNet

Manual inspection is then used to decide on the performance of the two sentiment analysis models given the nature of our unlabelled dataset. Our results show that VADER performed better than SenticNet.

[1] Positive, neutral and negative scores from VADER refer to the percentage of the respective sentiments in a sentence. Compound score is a normalised score of all three and hence it is used as a standard to compare different sentences.

Model	Most Positive Comments	Most Negative Comments
VADER	Congrats Sam, it is a great idea and I look forward to watching and if given the chance to participate I'll be there. I'm a truckin biker of the old school I make more time to ride these days as I have been a successful owner-operator in the trucking business. As a cold war vet from '69 to '71 I relish experiencing our great countries history and meeting the folks who hold secrets and stories not readily available to most. Best wishes and best of luck.	F*ck it, why not. But you better hit someone with that stick.
SenticNet	Nicely presented Kickstarter - Good Luck! @ jerice50: Thank you very much.	Congratulations on funding!

SenticNet did not perform well because it gives negative scores to words such as ‘funding’, and many sentiments are ignored when they cannot be matched with the SenticNet library.

3.4.2 Detailed Approach - Sentiment Analysis on Comments and Project Description

Based on the sentiment scores on project comments of VADER, we are able to find out the most favourable project categories. As emotional appeal plays an important part in project description, short project descriptions are also analysed. Additional attention is given to those with negative compound scores to find out why.

3.4.3 Detailed Approach - Word Cloud Visualization

Natural language processing techniques such as tokenization and removing stop words are used to preprocess the texts. We used unigram to find out the key words and bigram and trigram for key phrases on word clouds. Both project descriptions and comments were used as inputs. We visualise the most frequently used words for all projects, different project categories, successful projects, exceptional projects (those raised two times of their goal), and projects with negative sentiment scores. Difficulties were encountered when creating word cloud visualisation due to the lack of documentation on the Wordcloud library. It is solved by trial and errors.

3.4.5 Challenges

Sentiments for non-English comments are removed and misspellings may cause inaccuracies. There is also a limitation to the VADER model such as detecting sarcasm or irony and grammatical mistakes.

3.5 Content Based Recommender

3.5.1 Creating Corpus

	index	name	category	blurb	Dominant_Topic
0	10	Cook-a-long Children's Book: Growing Dill	children's books publishing children's books	children cook long book helping parents get children involved kitchen exploring new foods fun story format	27.0
1	45	Read-a-Feeling: the books about emotions	children's books publishing children's books	hello project children book series focuses emotional education help teach kids emotions	27.0
2	73	Reno Vegas Photography story book	photobooks photography photobooks	hypnotic new look behind scenes las vegas eyes one las vegas natives love town	27.0
3	178	2016 Fantasy Releases: Two New Worlds!	fiction publishing fiction	thrilled present two new breathtaking fantasy worlds love middle grade books love young adult covered	27.0
4	191	Fractured Fairy Tales	fiction publishing fiction	told retold fairy tales form backbone storytelling heritage treasure trove old new retold	27.0

Figure 17: Topics assigned based on LDA/NMF

Based on the topics generated in the LDA and NMF models, each project in the corpus is assigned a topic. In total 3 corpus were created, one without any topics and the other two with LDA topics and NMF topics respectively.

3.5.2 Selecting Projects for Analysis

Based on a selected project (i.e. Albert's Cookie- A Children's Book - Topic 1) the corresponding projects with the same topic (Topic 1) was extracted into a new corpus. This step was repeated for both LDA topics and NMF topics. TF-IDF and cosine similarity was then applied on this respective corpus.

3.5.3 Generating Cosine Similarity

```
# instantiating and generating the count matrix using CountVectorizer
count = CountVectorizer()
count_matrix = count.fit_transform(dfTopic['bag_of_words'])
```

```
# generating the cosine similarity matrix
cosine_sim = cosine_similarity(count_matrix, count_matrix)
print(cosine_sim)
```

```
[[1.         0.3796283  0.05976143 ... 0.         0.         0.06454972]
 [0.3796283  1.         0.         ... 0.         0.         0.        ]
 [0.05976143 0.         1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.13245324]
 [0.06454972 0.         0.         ... 0.         0.13245324 1.        ]]
```

Figure 18: TF-IDF Score

The category and blurb were tokenized and a bag of words was created. A count matrix was then instantiated and generated using the CountVectorizer based on the bag of words. Subsequently, a Cosine similarity matrix was generated based on the count matrix.

3.5.4 Returning Similar Projects

```
[52]: def recommendations(name, cosine_sim = cosine_sim):

          recommended = []

          # getting the index with the given title
          idx = indices[indices == name].index[0]

          # creating a Series with the similarity scores in descending order
          score_series = pd.Series(cosine_sim[idx]).sort_values(ascending = False)

          # getting the indexes of the 10 most similar
          top_10_indexes = list(score_series.iloc[1:11].index)

          # populating the list with the titles of the best 10 matching
          for i in top_10_indexes:
              recommended.append(list(dfTopic['name'])[i])

          return recommended
```

```
[56]: recommendations('Reno Vegas Photography story book')
```

```
[56]: ['Iran - A Mystery Narrated Through Photos',
       'Forsaken Official Release',
       'Eyes on Main Street 5th Edition',
       "The Knife's Edge - Revised (New Edition)",
       'MODEL WAREHOUSE magazine',
       'Observations in 6x6',
       'Unearth Women: The 1st Travel Magazine For Women, By Women',
       '2016 Fantasy Releases: Two New Worlds!',
       'Wild Darlings Photography & Art Studio Project',
       'Skatercross, pro skateboard race']
```

Figure 19: List of Recommended Projects

Based on the project title, the top ten other projects with the highest cosine similarity will be returned to the users.

3.5.5 Challenges

One of the challenges posed in this analysis is using short descriptions or categories may not be enough for topic modelling and finding similar projects. Also, cosine similarity only matches the exact words as well as the challenge posed by Lexical Ambiguity.

3.7 Demo - Webapp

The final system would be a web application for our fundraisers. They would be able to input the description of the project they have in mind to see what are the potential or latent labels that it belongs to.

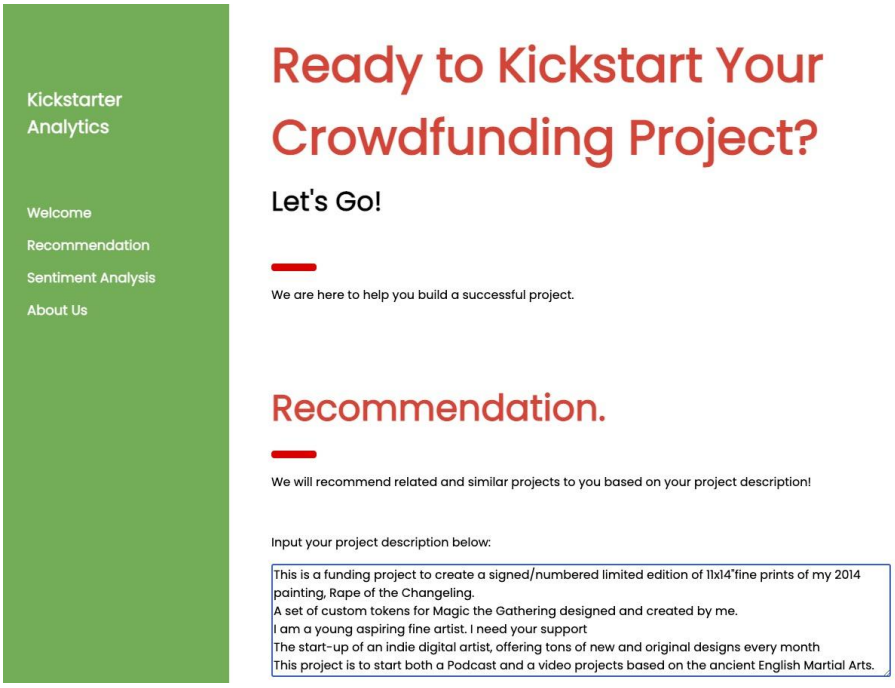


Figure 20: Webapp Demo

Our system would then show them the top three related projects and the users are able to access them via hyperlinks.

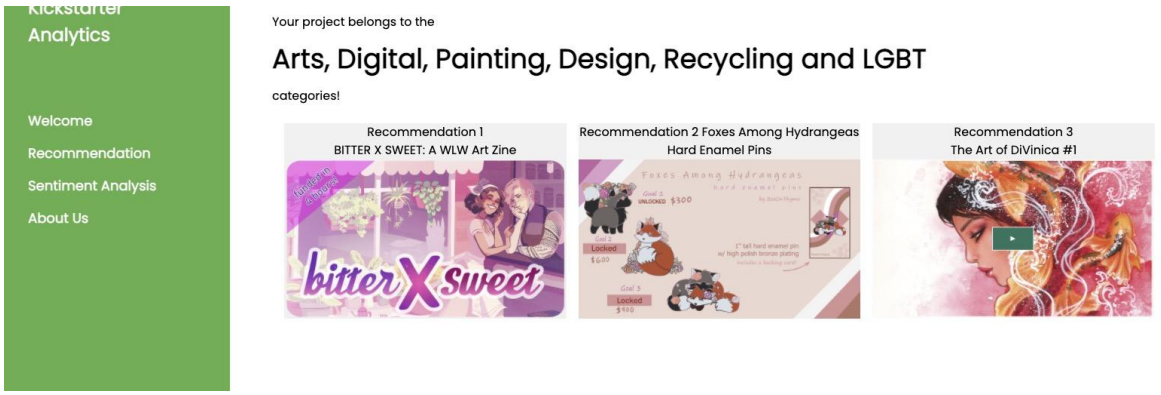


Figure 21: Webapp Demo showing results on related projects

Users are also able to view the general sentiments across all projects when they enter the page. They can also choose a specific category such as 'Art, illustration' to view the most commonly used words by commenters and other project starters in the project descriptions. They can also view the projects with the most positive sentiments under the chosen category.

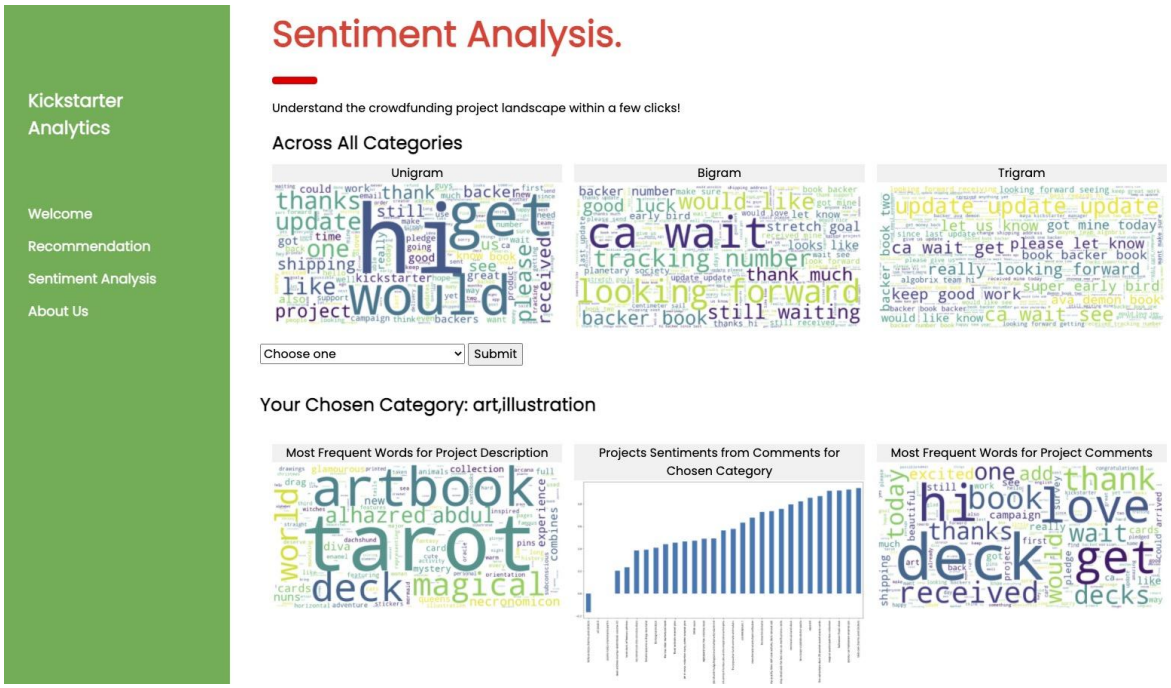


Figure 22: Webapp Demo showing results on sentiment analysis

4. Results and Analyses

4.1 Project Topic Modelling using LDA

4.1.1 Evaluation

```
]: pprint(ldamallet.show_topics(formatted=True)[0:4])
   print('\nCoherence Score: ', coherence_ldamallet)

   [(35,
     '0.244*"game" + 0.078*"play" + 0.043*"mobile" + 0.040*"player" + '
     '0.032*"card" + 0.028*"puzzle" + 0.025*"battle" + 0.025*"board" + '
     '0.022*"building" + 0.016*"win"'),
    (22,
     '0.077*"pin" + 0.074*"set" + 0.062*"collection" + 0.054*"inspire" + '
     '0.051*"feature" + 0.040*"enamel_pin" + 0.037*"theme" + 0.034*"cat" + '
     '0.033*"animal" + 0.022*"cute"'),
    (18,
     '0.078*"give" + 0.074*"young" + 0.073*"girl" + 0.051*"short_film" + '
     '0.048*"follow" + 0.034*"man" + 0.029*"boy" + 0.019*"opportunity" + '
     '0.019*"chance" + 0.016*"freedom"'),
    (37,
     '0.080*"support" + 0.078*"coffee" + 0.046*"company" + 0.039*"launch" + '
     '0.031*"shop" + 0.031*"move" + 0.030*"expand" + 0.027*"bar" + 0.026*"summer" '
     '+ 0.020*"cafe"')]

   Coherence Score:  0.38113971067952596
```

Figure 23: Coherence Source

The model had a coherence score of 0.38. Human evaluation was also done for this model which will be outlined in the comparison results of the LDA vs NMF models in the later part of the report.

4.1.2 Topics Generated

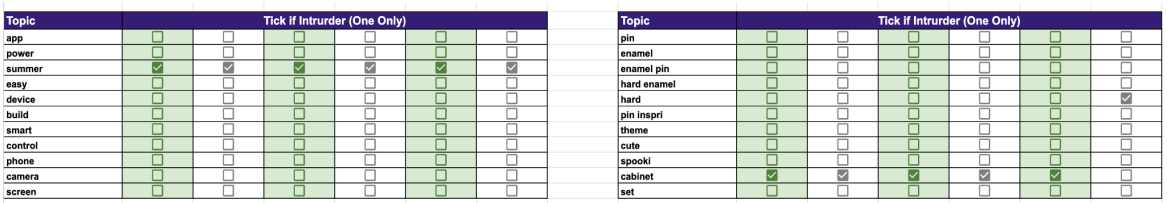
Figure 24: 10 Random Generated Topic

These word clouds show 10 random topics from the LDA topic modelling. The larger the word, the higher the term frequency within the selected topic. We can see that most of the words in each topic are interrelated.

4.2 Project Topic Modelling using NMF

For NMF, the coherence score was 0.42 and topics of 75 were chosen. As mentioned above, we went through human evaluation as well to compare the two models created.

4.2.1 Evaluation Metrics



Topic	Tick if Intrurder (One Only)					
app	☐	☐	☐	☐	☐	☐
power	☐	☐	☐	☐	☐	☐
summer	☑	☑	☑	☑	☑	☑
easy	☐	☐	☐	☐	☐	☐
device	☐	☐	☐	☐	☐	☐
build	☐	☐	☐	☐	☐	☐
smart	☐	☐	☐	☐	☐	☐
control	☐	☐	☐	☐	☐	☐
phone	☐	☐	☐	☐	☐	☐
camera	☐	☐	☐	☐	☐	☐
screen	☐	☐	☐	☐	☐	☐

Topic	Tick if Intrurder (One Only)					
pin	☐	☐	☐	☐	☐	☐
enamel	☐	☐	☐	☐	☐	☐
enamel pin	☐	☐	☐	☐	☐	☐
hard enamel	☐	☐	☐	☐	☐	☐
hard	☐	☐	☐	☐	☐	☑
pin inspri	☐	☐	☐	☐	☐	☐
theme	☐	☐	☐	☐	☐	☐
cute	☐	☐	☐	☐	☐	☐
spooki	☐	☐	☐	☐	☐	☐
cabinet	☑	☑	☑	☑	☑	☐
set	☐	☐	☐	☐	☐	☐

Figure 25: Intruder Test

To determine which topic model is the best, we asked users to do a word intrusion task based on 10 random topics generated from each model. Each task considered 10 words from the topics and an intruder word. The goal of this human evaluation was to evaluate whether human subjects can identify this intruder word. This process was repeated for each of the 10 topics, and the resulting precision is computed. The coherence and the precision of each model as follows.

Model	LDA	NMF
Coherence Score	0.39	0.42
Word intrusion	0.27	0.40

We can infer that the NMF provides better results for topic modeling.

Using NMF, we found out that **Film and Horror** was the topic that appeared the most. This means that most of the people that ask for funds have projects under the film and horror category. From this, we could also infer that the common trends among people asking for funds. Hence, this could help prospective kickstarters to gauge fundings based on specific topics.

4.3 Project Topic Classification

4.3.1 Performance analysis:

The validation returned an accuracy of 99.29%.

4.3.2 Sample Output:

```
...
art_2_000008.txt ⟶ art and art (Single tag prediction)
art_2_000009.txt ⟶ art and art
art_2_000010.txt ⟶ art and comics  (Multi-tag prediction)
comics_10_000001.txt ⟶ comics and publishing
comics_10_000002.txt ⟶ publishing and comics (Wrong main tag)
...
```

4.3.3 Summary of output

The model was able to accurately suggest tags that are relevant to the short descriptions provided with sample labels provided. With actual working labels, more unique labels such as sculpture, digital art and etc should be able to generate similar results to the solution provided above. With this, the model would be able to generate 2 additional tags for the Kickstarter project. More tags can be recommended if more labels are provided.

4.3.4 Error Analysis

It seems like the comics_10_000002.txt document's main tag/ category has been labeled incorrectly. Despite it being a comics document, it was labeled mainly a publishing document rather than a comics document. Upon comparing comics_10_000001.txt with comics_10_000002.txt, the former document contained more “comic” than the latter. The former had 8 counts of “comic” whereas the latter only had 2. This seems to alter the accuracy of the predicted main tag labeling. However, both predicted labels are correct. The error could be part of the Naive Bayes classifier limitation as it is sensitive to skewed data.

4.4 Sentiment Analysis

4.4.1 VADER Sentiment Score

For project comments, most of the projects score positive. For project description, many projects are neutral or positive, but a significant portion score negative which is interesting. We will dive in details in the later section.

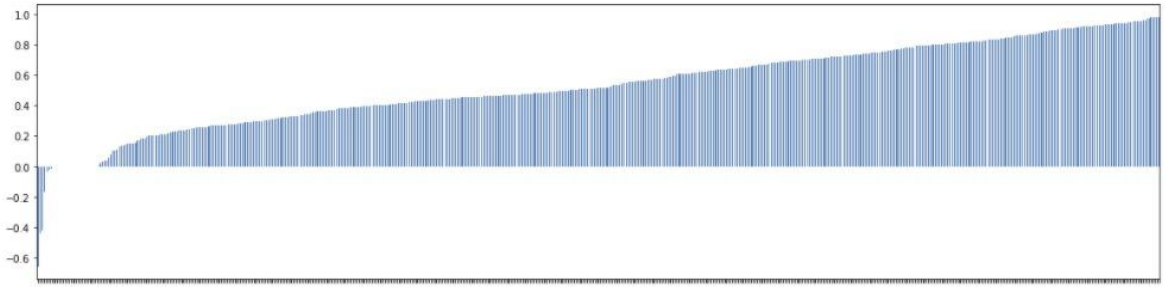


Figure 26: Compound scores for comments on project level

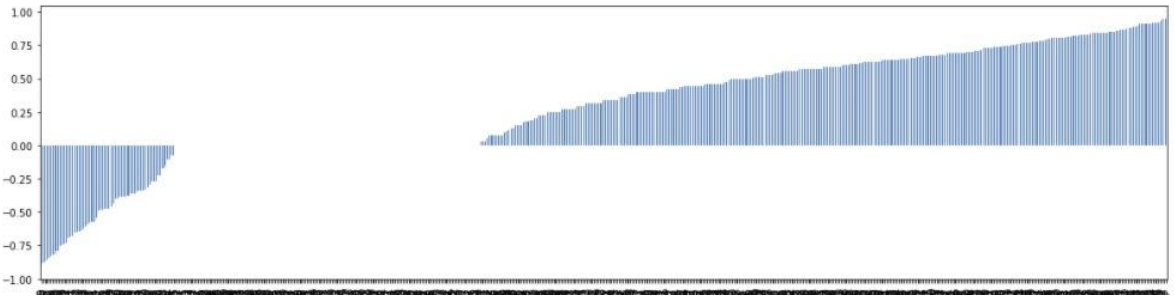


Figure 27: Compound scores for short project description on project level

For comments, projects under ‘Film & video, television’ scored the highest, reflecting that Kickstarter viewers like this category the most. ‘Comics’ category scores the lowest, but on the category level all the sentiments are all positive overall.

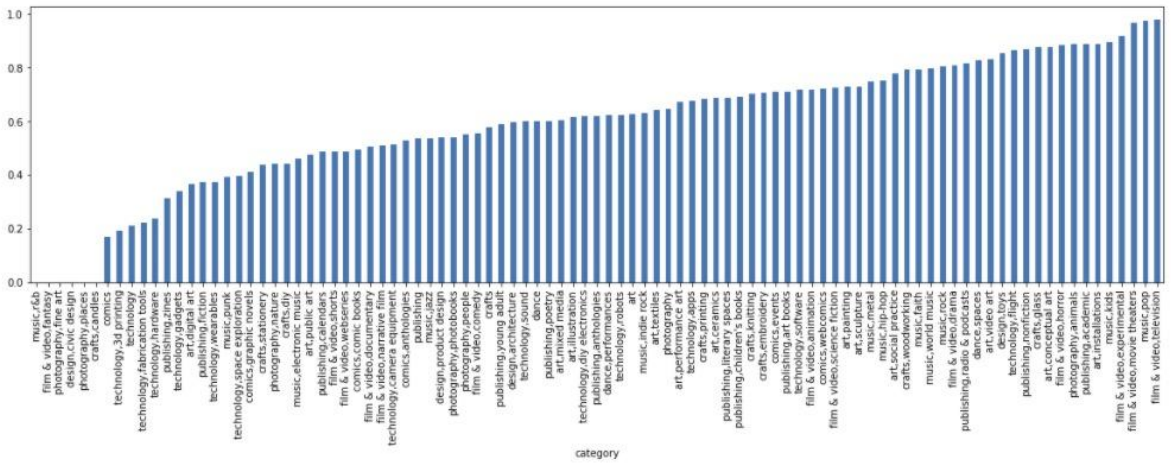


Figure 28: Compound scores for comments for all categories on project level

Exceptional projects are defined as those that raised more than 200% of their goals. It is noticed that the proportion and degree of the projects scoring negative is less than that of all projects, which makes sense.

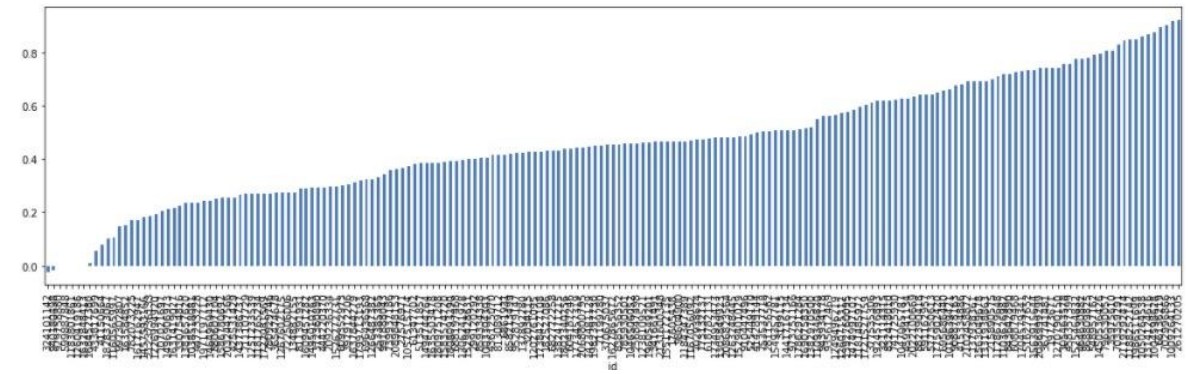


Figure 29: Compound scores for comments for exceptional projects on project level

4.4.2 Word Cloud

Across all projects, we noticed that the general attitude from commenters are anticipating, encouraging and polite. Shipment and updates on projects concern the investors the most.

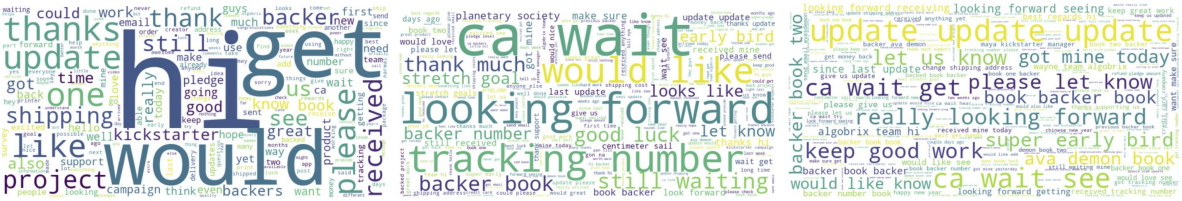


Figure 30: Unigram, bigram and trigram for frequent words in all project comments

Across all projects, ‘help’ and its associations are used the most frequently. Arts related products such as books, comic books and enamel pins are popular among fundraisers.

Figure 31: Unigram, bigram and trigram for frequent words in all project description

No observable differences between comments for exceptional projects and across all projects. This may lead to the fact that there is no observable links between sentiments of the commenters and the possibility of building a successful project. Those who invest may

not leave positive comments. It is noticed that for exceptional project description, words related to technology and more specific words such as ‘propulsion’ and ‘space citizen’. This shows by using specific words in project description helps with gaining viewership, instead of simply asking for help.

Figure 32: Unigram for exceptional project comments and description

When zoomed in on the project category level, there is a difference on the comments across categories. This indicates that commenters discuss category specific topics in the comment section.

Figure 33: Bigram for art,illustration and film & video,television project comments

The most popular products for the most popular topics are printers with different functionalities and original tarot deck. Card games such as Tarot and Fictional characters such as *Alhazred Abdul* are popular. This means that the audience and backers tend to look for products that are not easily found on mainstream markets on Kickstarter. Therefore, prospective project starters or fundraisers should pay more attention to projects that are of original content and in the field of horror/weird fiction and fortune telling.

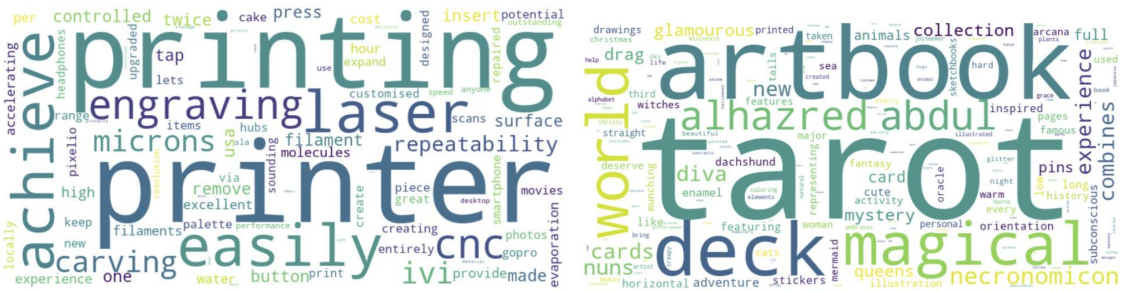


Figure 34: Unigram for technology,3d printing and art,illustration project description

We noticed that some projects do include negative words in the project description, but it is likely that those words are used for horror fiction and films or to alleviate social issues.

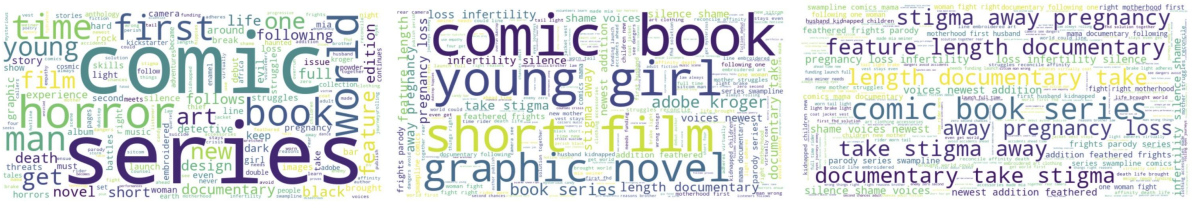


Figure 35: Unigram, bigram and trigram from project description for projects with negative compound score for vader

4.4.3 Other Insights

We found that **Technology-related** projects received the most comments while **art or illustration related** projects raised the highest amount of funds. This means that there is an imbalance between the ones who get public attention to those who get the fundings. Therefore, this could mean that although many people are looking forward to advanced technology or technology related projects, most of them are also taken aback by the chances of failure.

We also noticed that successful projects in Hong Kong and Germany did not receive as much comments as the successful projects in other countries. This could be attributed to the fact that the people in Hong Kong and Germany are non-native English speakers and hence, their project descriptions are not translated to English. Hence, an **auto-translation system** can be implemented to translate project descriptions of those projects to expose them to non-English speaking countries and vice versa.

4.5 Content Based Recommender

4.5.1 Evaluation Metrics

Project	Bicycle Gnomes Playing Cards	Tick if related					
Recommender One	Evolve Bicycle® Playing Cards Deck	☑	☑	☑	☑	☑	☑
	Royal House Limited Edition by Edgy Brothers	☑	☑	☑	☑	☑	☑
	Kingdoms of Erden Limited Printing Plastic Playing Cards	☑	☑	☑	☑	☑	☑
	CATastrophe Tarot Deck	☐	☐	☑	☑	☑	☐
	HEXANNE PLAYING CARDS	☑	☑	☑	☑	☑	☑
	BICYCLE CARDISTRY B/W - THE NEXT LEVEL IN DESIGN	☑	☑	☑	☑	☑	☑
	Parasol Tree - Dark Phoenix Playing Cards	☑	☑	☑	☑	☑	☑
	Fiveaside®: retro-inspired football card game.	☐	☐	☐	☑	☑	☐
	Beamz Interactive Laser Music & Gaming Controller	☐	☐	☐	☐	☐	☐
	Bios:Mesofauna & Galenus	☑	☐	☐	☑	☑	☐
Recommender Two	⚜ Florentia Playing Cards ⚜',	☑	☑	☑	☑	☑	☑
	"EH B'Y Newfoundland Playing Cards",	☑	☑	☑	☑	☑	☑
	Floppy Disk Playing Cards',	☑	☑	☑	☑	☑	☑
	"MUERTOS" Playing Cards',	☑	☑	☑	☑	☑	☑
	BOWL-A-RAMA Playing Cards',	☑	☑	☑	☑	☑	☑
	The Music Box Playing Cards Relaunch',	☑	☑	☑	☑	☑	☑
	Knights Bicycle Playing Cards Poker Size Deck Custom Limited',	☑	☑	☑	☑	☑	☑
	Royal House Limited Edition by Edgy Brothers',	☑	☑	☑	☑	☑	☑
	ZDV2: retro',	☑	☑	☑	☑	☑	☑
	Myndset: Get inspired to create ideas in 30 minutes or less',	☑	☐	☑	☑	☑	☐
Recommender Three	Æsir Gold Playing Cards and Prints	☑	☑	☑	☑	☑	☑
	⚜ Florentia Playing Cards ⚜	☑	☑	☑	☑	☑	☐
	Bicycle Koi Playing Cards	☑	☑	☑	☑	☑	☑
	🌈 Vaivorykštė - Historic, Playing Cards from 1933 🌈	☑	☑	☑	☑	☑	☑
	Drink Kings	☑	☐	☑	☑	☑	☑
	White Rabbit: Back Down the Rabbit Hole Relaunch	☑	☐	☑	☑	☑	☑
	Betrayers Playing Cards	☑	☑	☐	☑	☑	☑
	Bicycle Four Seasons Limited Edition (Summer) Playing Cards	☑	☑	☑	☑	☑	☑
	LINIA: Custom Playing Cards	☑	☑	☑	☑	☑	☑
	Treble Clef Deck of Playing Cards V1 Red. Limitied Edition	☑	☑	☑	☑	☑	☑

Figure 36: Evaluation Metrics

To calculate how accurate and relevant our recommendations are to the users, we used the precision @ K. 10 recommendations were given for each recommender system. Recommender one was the combination of LDA and TF-IDF while recommender two was the combination of NMF and TF-IDF and recommender three was TF-IDF only.

The formula for Precision @ K is as follows:

```
Precision@k = (# of recommended items @k that are relevant) / (# of
   recommended items @k)
```

Precision @ K	LDA + TF-IDF	0.636
	NMF + TF-IDF	0.472
	TF-IDF	0.916

Based on the results, the TF-IDF still produces better results than the other models. This could mean that the topic models may still need further tuning or a different or larger corpus for generating the topics. However, in some cases, the LDA + TF-IDF still outperformed the TF-IDF model which showed that this approach was still useful in recommending similar projects to our users.

5. Discussions and Gap Analysis

5.1 What went well

There were several things that went well for the project, we found out that the best model for Topic Modelling is NMF as there was better coherence results and users were better able to detect word intrusion, while for Cosine Similarity works better for the Recommender as it returned better results and achieve a higher precision@k score. For sentiment analysis, VADER worked better as it is able to find the most positive and negative comments in comparison to Senticnet.

5.2 What did not go well

Our team has also tried Content Based Recommender with topic modelling however it did not produce optimal results in comparison with Cosine Similarity method. Senticnet did not perform well for our sentiment analysis as some of the sentiments were categorised wrongly.

5.3 Gap Analysis

For all the analysis conducted in the project, one of the common challenges or problems faced was non-English comments or projects that had to be filtered out because they could not be analysed with the tools that we used. Although this had to be done because of limited time given, non-English comments and projects in fact, play a huge part in Kickstarter. According to the Statista, the top 10 countries with the most Kickstarter pledge amounts are as follows: USA ($663.32m), UK ($54.43m), Canada ($44.91m), Australia ($31.78m), Germany ($21.61m), France ($10.13), Sweden ($7.15m), Japan ($7.14m), Netherlands ($7.03m), and Singapore ($6.71m) (Petronzio, 2014). From these statistics, it is evident that 50% of the top 10 countries use non-English language. This means that our results could have been heavily affected and new insights could have been drawn if we were to include any non-English analysis in our project.

Although there are methods suggested to conduct topic modelling, sentiment analysis and topic classification for non-English language, many of them are at infancy stage and still possess other challenges. For instance, for sentiment analysis of non-English texts, there is a huge number of unlabeled data available on the internet and labeled non-English dataset

is scarcely available, obstructing building effective supervised machine learning system in non-English data (Djatmiko et al., 2019).

Our dataset may be biased in representing the entire Kickstarter project repository. Some projects receive more comments than the rest. This can be solved by adopting a stratified sampling method for the projects from different categories, sizes, states etc. Our analytics approach can be scaled up and reapplied to the improved dataset.

5.4 Improvements Methods

Results for the LDA model **further tuning of hyperparameters** could help in improving the results further. Automated Hyperparameters tuning techniques could also be applied to find the best hyperparameter combination for the best results. Applications such as Amazon Sage (AWS, n.d.) which has an in-built model tuning could also be used to find the right hyperparameter combination for producing the best results.

One of the possible improvements for sentiment analysis method is **use sentiment lexicon available for non-English sentiments**. Additional research could be done in order to look out for non-English sentiment lexicon available. One example, for German language there is an available sentiment lexicon SentiWS (Universitat Leipzig, n.d.) from there the sentence polarity could be computed from the values of individual words.

For the content based recommender, one of the improvements could be **tackling lexical ambiguity** to produce better results. Using word sense disambiguation (WSD) would help to assign the meaning of the word in the context. Unknown words could also be put in a dictionary to refer to it for better recommendation results.

6. Future Work and Conclusion

6.1 Extension of the Project

As the Kickstarter website is used worldwide it is not surprising to see projects in different languages. Hence, the sentiments of other languages also plays a part in accurately recommending the right words to the prospective project starters. This method requires time to be completed as the words from the native language have to be collected and should be huge enough to be trained for the best classification accuracy results. Training a huge

dataset would take time but it will produce the best results for non-English sentiments (Koning, 2020).

Other techniques could also be applied such as transfer learning for smaller datasets. Transfer learning involves a neural network model being first trained on a problem similar to the problem that is being solved (Brownlee, 2020).

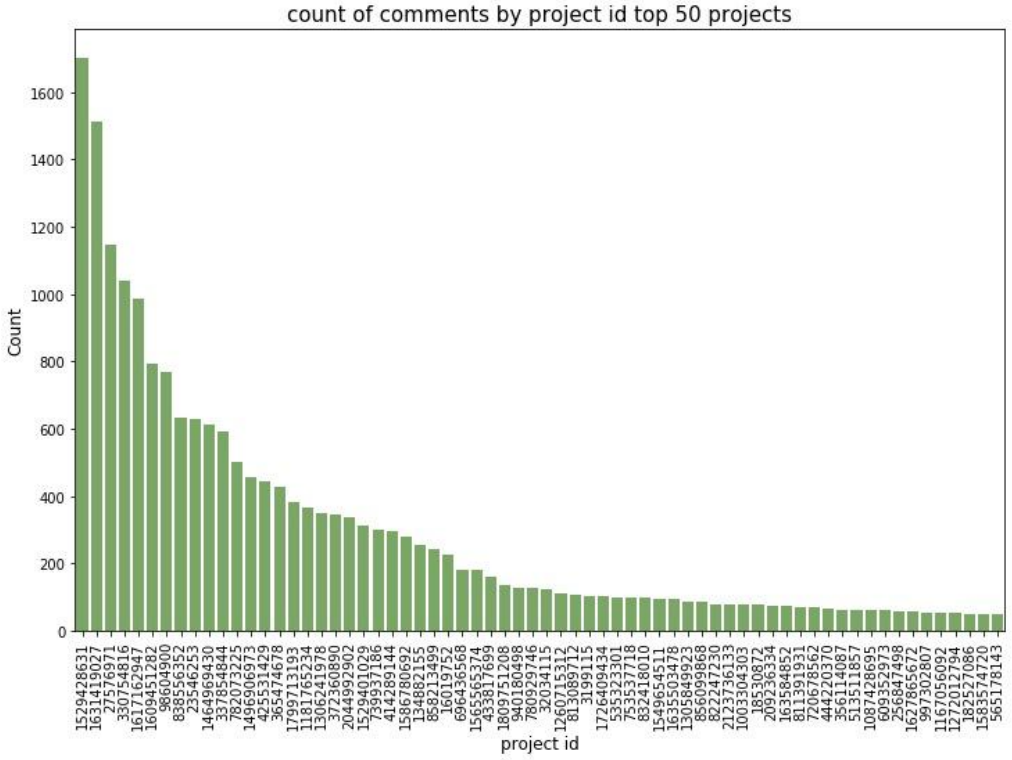
To refine the project topic classification with producing more accurate results, smoothing techniques such as laplace smoothing could be applied together with Naive Bayes to tackle the problem of zero probability. There are also other various smoothing methods such as Jelinek-Mercer, Dirichlet, Absolute Discounting and Two-stage smoothing that could be used with Naive Bayes to improve the model and accuracy of its results (Yuan et al., 2012). Therefore, our system helps the project starters tag and categorise projects accurately to increase their chances of getting fundings.

6.2 Summary of the Project

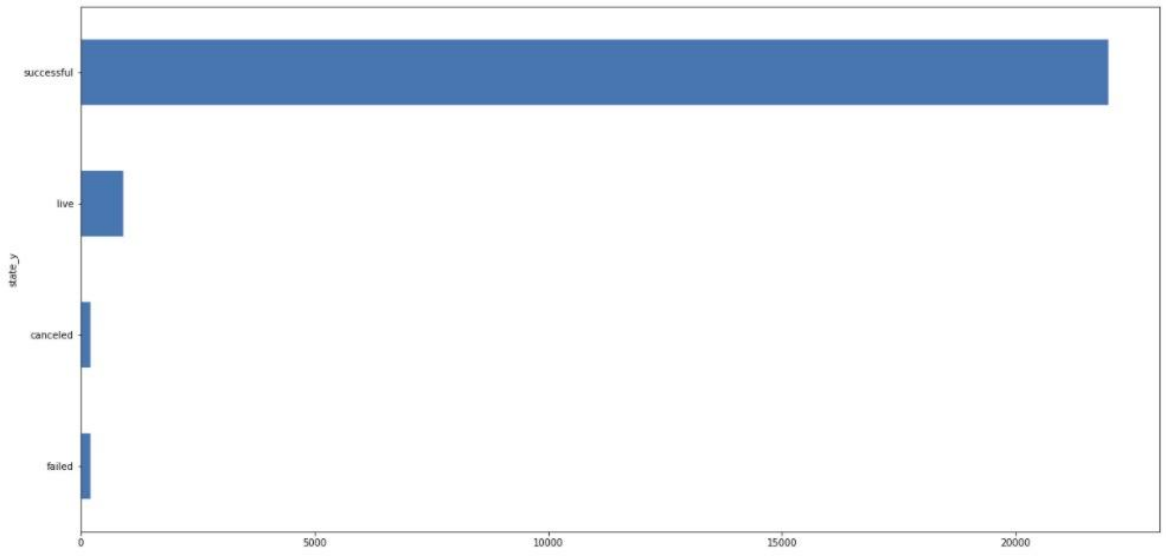
Overall, combining the things we learned in class and additional resources found in the internet beyond the class helped us to achieve our goals for the project. Our team managed to produce required results through experimenting different methods and comparing them on what works best to solve the business problem. Our team met challenges along the way but managed to tackle them through further research and consulting one another. This project taught us to apply concepts as well as work with one another to complement each other's strengths and weaknesses.

7. Appendix

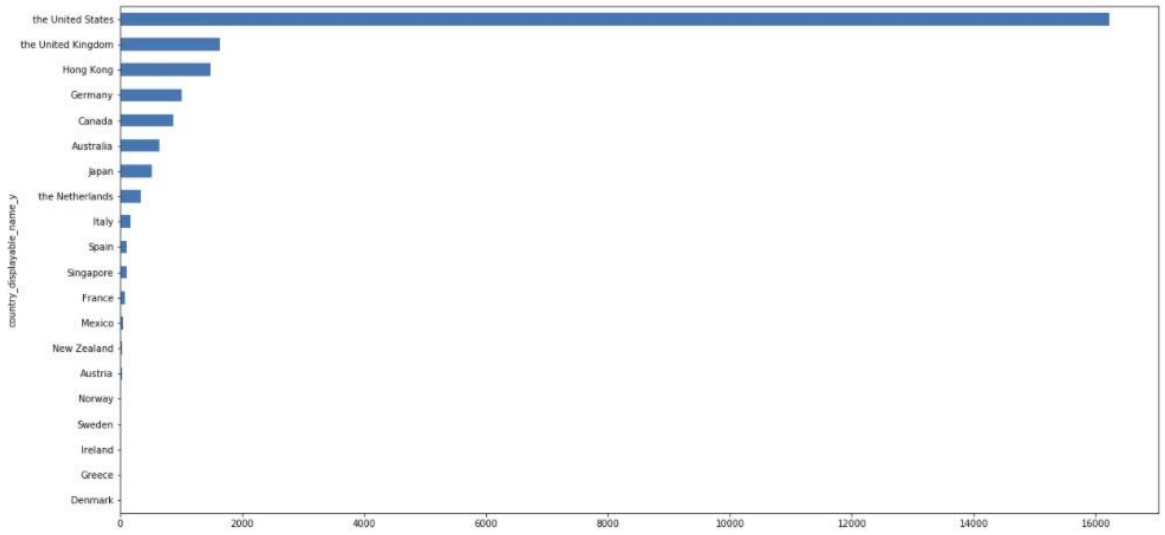
Appendix I: EDA Results



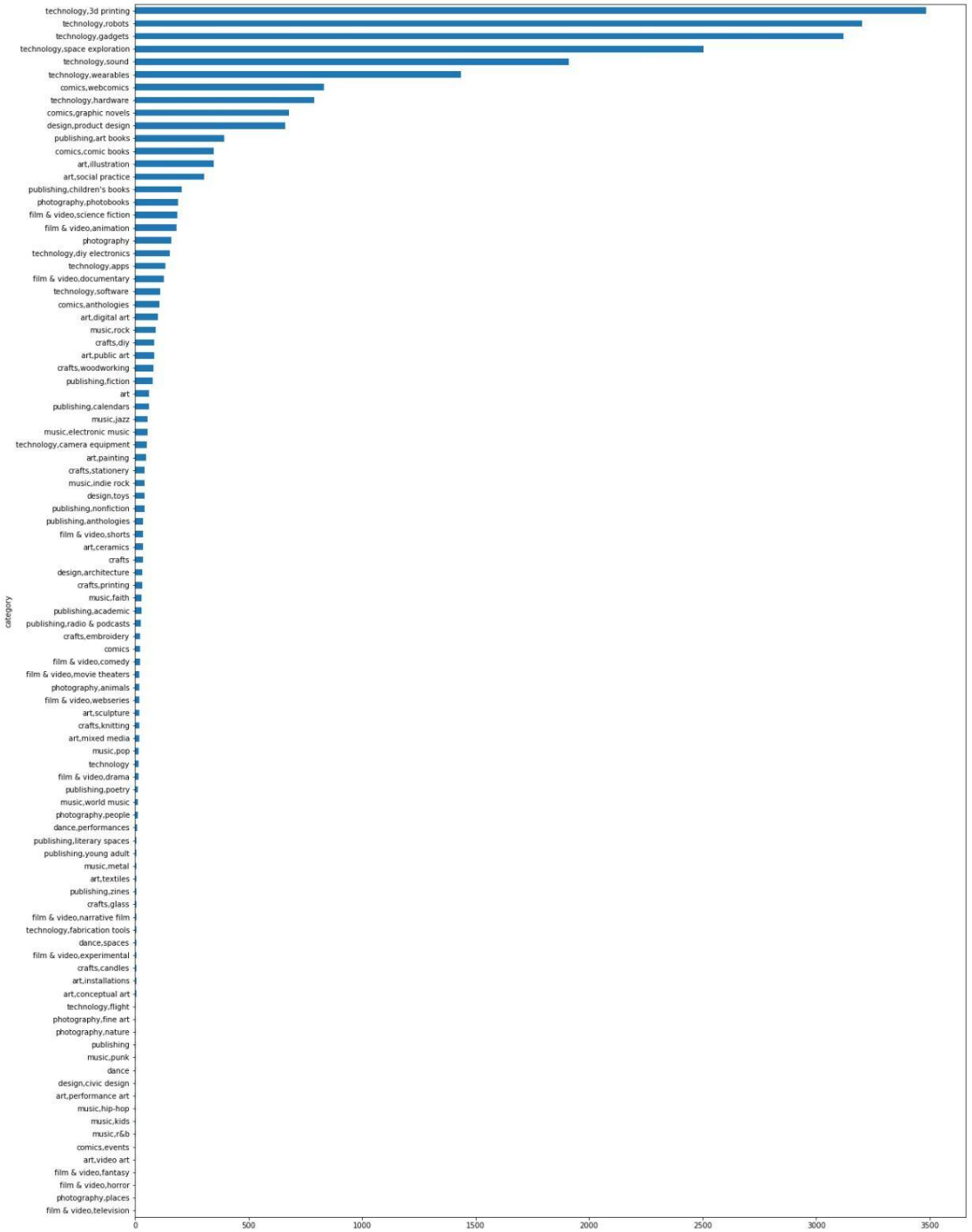
Count of comments by projects



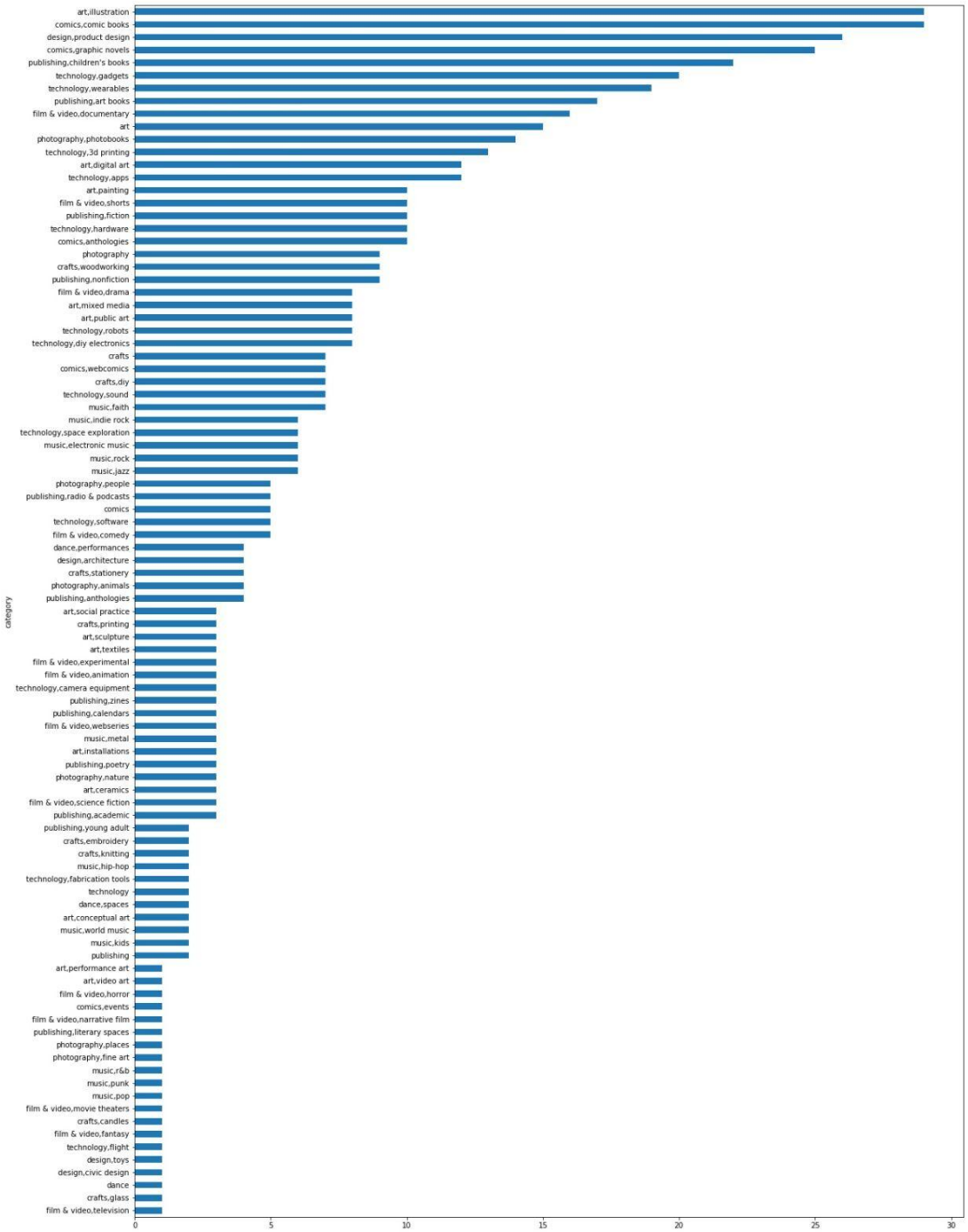
Count of projects by states



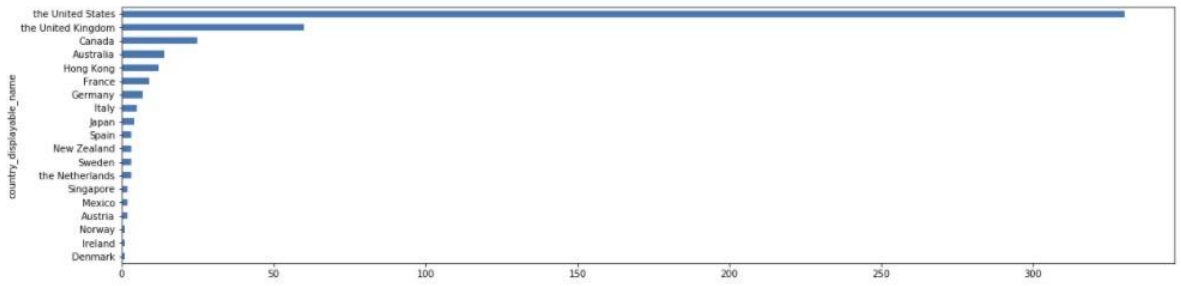
Count of projects by countries



Count of comments by project categories



Count of project by categories



Count of projects that raised more than 100% of their goals by countries

Appendix II: Data Columns

WebRobots	Crawled data by team	Final_data
backers_count	Title	Title
blurb	Image	Name
category	Name	Type
converted_pledged_amount	Type	Comment_URL
country	Comment_URL	Period
country_displayable_name	Period	project
created_at		project_name
creator		backers_count
currency		blurb
currency_symbol		category
currency_trailing_code		converted_pledged_amount
current_currency		country
deadline		country_displayable_name
disable_communication		currency
friends		current_currency
fx_rate		fx_rate
goal		goal
id		id
is_backing		location
is_starrable		name
is_starred		pledged
launched_at		profile
location		slug
name		state
permissions		static_usd_rate
photo		urls
pledged		usd_pledged
profile		has
slug		
source_url		
spotlight		
staff_pick		
state		
state_changed_at		
static_usd_rate		
urls		
usd_pledged		
usd_type		

 # Title here refers to the project comments.

8. References

AWS. (n.d.). *Tune an LDA Model*. https://docs.aws.amazon.com/sagemaker/latest/dg/lda-tuning.html

Brownlee, J. (2020, August 18). *Transfer Learning in Keras with Computer Vision Models*. Machine Learning Mastery. https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

Djatmiko, F., Ferdiana, R., & Faris, M. (2019). *A Review of Sentiment Analysis for Non-English Language*. https://doi.org/10.1109/ICAIIT.2019.8834552

Kickstarter and Taxes — Kickstarter. (n.d.). KickStarter. https://www.kickstarter.com/help/taxes

Kickstarter Stats — Kickstarter. (2021, April 12). Kickstarter. https://www.kickstarter.com/help/stats#:%7E:text=Funding%20on%20Kickstarter%20is%20all,their%20goal%20were%20successfully%20funded

Koning, M. (2020, May 5). *To translate or not to translate, best practices in non-English sentiment analysis*. Towards Data Science. https://towardsdatascience.com/to-translate-or-not-to-translate-best-practices-in-non-english-sentiment-analysis-144a53613913

Lowry-Duda, J. (2020, December 31). *Concept-level sentiment analysis: The next level of understanding emotion in text feedback*. Luminoso. https://www.luminoso.com/post/concept-level-sentiment

Ma, Y. M. (2020, February 5). *NLP: How does NLTK.Vader Calculate Sentiment?* Medium. https://medium.com/ro-data-team-blog/nlp-how-does-nltk-vader-calculate-sentiment-6c32d0f5046b

Petronzio, M. (2014, March 5). *The Top 10 Countries by Money Pledged on Kickstarter*. Mashable. https://mashable.com/2014/03/04/kickstarter-countries/

Ribeiro, F. N., Araújo, M., Gonçalves, P., André Gonçalves, M., & Benevenuto, F. (2016). SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, *5*(1), 1. https://doi.org/10.1140/epjds/s13688-016-0085-1

Universitat Leipzig. (n.d.). *Download page of the project Deutscher Wortschatz / Leipzig Corpora Collection*. https://wortschatz.uni-leipzig.de/de/download#sentiWSDownload

Yuan, Q., Cong, G., & Thalmann, N. M. (2012). *Enhancing Naive Bayes with Various Smoothing Methods for Short Text Classification*. https://personal.ntu.edu.sg/gaocong/papers/wpp095-yuan.pdf