Bachelor's thesis

Bachelor of Engineering, Information and Communications Technology

2020

Zhen Wang

SENTIMENT ANALYSIS OF CLIMATE CHANGE USING TWITTER API AND MACHINE LEARNING

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor of Engineering, Information and Communications Technology

2020 | 32 pages, 7 pages

Zhen Wang

SENTIMENT ANALYSIS OF CLIMATE CHANGE USING TWITTER API AND MACHINE LEARNING

Climate change, also called global warming, is one of the most frequently discussed topics nowadays. Also, there has been a tremendous growth in the research for sentiment analysis on social network by using natural language processing (NLP). Sentiment analysis has been widely applied in various commercial and non-commercial areas. People have different opinions on various topics with positive, negative or neutral comments on social media. This thesis work performs sentiment analysis of content from Twitter with climate change hashtags, using Twitter API for authentication and tweepy libraries. Multinomial Naive Bayes Classifier is an optimal method which is selected for training model and detecting people's opinions on climate change. The sentiment analysis has shown the result of 67% accuracy. The findings have indicated that the majority of the training group has negative sentiment on climate change, whereas the minority has an optimistic attitude towards climate change.

KEYWORDS:

sentiment analysis, machine learning, climate change, Twitter

CONTENTS

APPENDICES

FIGURES

LIST OF ABBREVIATIONS (OR) SYMBOLS

API	Application Programming Interface
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
SNA	Social Network Analysis

1 INTRODUCTION

In recent years, climate change, also called global warming, is one of the most widely discussed topics. Some people take it as a severe problem and become worried about the situation, however, some other people think of it as a hoax and political creation. The main goal of this thesis work is to analyze people's opinions on climate change over the years using the sentiment analysis method with the help of Twitter data. Social media provides the opportunities to collect the information and apply data processing.

When it comes to the Internet, people are all familiar with it. This great invention of technology has revolutionized human being everyday communications. According to the statistics from United Nations, by February of 2019, the current worldwide population is 7.7 billion. The internet has 4.2 billion users, which is more than half of the population in the world. Social media takes the greatest number of users with 3.397 billion active social media users, which is 44% of the worldwide population (Statista, 2020). Social media has had a huge effect on communication and connection of lives.

In recent years, large amount of data is generated by human on social media. Users express opinions on the social network and provide valuable and useful data. Along with the development of the social media, there has been a tremendous growth in the research for sentiment analysis on social network by using natural language processing (NLP). Sentiment analysis has been widely applied in various commercial and non-commercial areas, such as reactions and reviews to news, product experiences, movies, services etc.

In this thesis work, climate changes as a subject or study topic is analyzed by sentiment analysis on the platform, which is social network, specifically on Twitter. Datasets will be obtained from Twitter using API and split into testing and training datasets. The multinomial Naive Bayes classifier is a model that suits for classification with discrete features (Bisht, 2016), which can be used for training the model.

The idea and inspiration of this thesis work came from Nithisha Mucha's doctor's rearch, in which the author obtain the datasets about global warming from Twitter and trains classifiers to optimize for best accuracy (Mucha, 2017). The work done in this thesis is implemented using Jupyter Notebook platform and more hashtag words related to

climate changes in count. Multinomial Naive Bayes classifier model is used for training discrete features.

2 SOCIAL MEDIA AND SENTIMENT ANALYSIS

This section provides an overview of social media, sentiment analysis and relationship between them. Each field will be discussed in a separate section as below. The former briefly introduces social media, sentiment analysis and their relationship. The latter introduces sentiment analysis and the methodology in use.

2.1 Relationship of social media and sentiment analysis

The rapidly growing use of Internet and web-enabled devices has made it possible to access information from various resources, regardless of the location and time. Everyday users from all over the world create large amounts of data through web-based applications and services. Among all products and services, social media has had a huge effect on communication and connection of human lives. Social media is the collected type of product or service in the network which provides the possibilities to carry out information, ideas, pictures and videos creation, share and exchange.

Sentiment analysis for social media focuses on the study of opinions which can express positive, neutral or negative sentiments (Liu, 2015). Sentiment analysis has many benefits in application. With the help of sentiment analysis tools and methods, individuals and organizations can increasingly use the content in social media for decision making. Sentiment analysis, also called opinion mining, mainly focuses on opinions which can express or indicate positive, negative and neutral sentiments. By applying sentiment analysis from its subjects, the users are becoming more and more understood. The organizations may give extra useful and valuable advice to the users and measure the marketing campaign.

Opinions are important. They act as the key influencer of human life. Opinions from others could help one to make final decisions. This involves not only individuals, but also organizations such as the government and companies. Before the evolution of social media, traditionally, the individuals made the decisions with the help from advertisement, company or products' reputation or even from friends and family. Additionally, the government or companies collect the opinion related data by using methods such as conducting surveys, opinion polls, and focus groups. With the help of the development of social media, both individuals and companies have changed the way of accessing

opinions by browsing many user reviews and discussions in public fora on the Internet. Companies are also able to collect beneficial data from internal resources, which can be customers feedback retrieved from emails or phone calls, as well as data from surveys (Liu, 2015).

2.2 Sentiment analysis and its methodology

Sentiment analysis, which is also called opinion mining, refers to the field of study analyzing people's opinions, sentiments, attitudes, evaluations and emotions. The target includes products, services, issues, events, topics, and their attributes (Liu, 2015). The subjects of sentiment analysis (opinion mining) includes people's opinions, sentiments, attitudes, evaluations and emotions. Sentiment expresses a view or opinion that is held or expressed, whereas the evaluation is the assessment. Emotions include six basic aspects which are anger, disgust, fear, happiness, surprise and sadness. Sentiment analysis has advantages. An example of this application is the impact of tracking people's attitude on products, services and events. This allows enterprises to have the whole view and perform decision-making. Another example is city council administrators who could have the opportunity to improve the services offered to citizens and for addressing challenges of development and sustainability more efficiently based on what people feel (Liu, 2015). Although sentiment analysis brings a large number of benefits to companies and users, there are also some disadvantages in applying automatic analysis because of the ambiguity of natural language and the features of the posts. For example, Twitter uses hashtags and this may create the difficulties in this kind of situations.

When it comes to the methodology of sentiment analysis, the existing studies on sentiment analysis can be grouped by different points of views. From the technological perspective, the approaches of sentiment analysis include machine learning, lexicon-based, statistical and rule-based approaches (Collomb, 2013).

3 IMPLEMENTATION

This chapter conducts sentiment analysis implementation with Jupyter Notebook web-based application, which presents data visualization result in the development environment at the same time without leaving the environment. Dr Mucha's rearch implements with typical integrated development environment whereas the work done in this thesis uses Jupyter Notebook web-based application. Jupyter Notebook development environment provides plain text and instant data visualization features.

3.1 Implementation environment and packages

3.1.1 Jupyter Notebook and its installation on Windows 10 (64bit) environment

The Jupyter Notebook is a powerful and versatile open-source tool which provides the environment to create and share code files. Users can work with live code, equations, data visualization performance, plain text at the same time. With all these fabulous features in the application of machine learning, data mining, data visualization, Jupyter has become one of the most popular web application tools among data scientists.

Jupyter Notebook can be installed using the conda package manager miniconda:

```
conda install -c conda-forge jupyterlab
```

Figure 1. Conda package manager miniconda.

3.1.2 Packages installation

Tweepy is an open-source python library for accessing the Twitter API, which can provide access to real-time data from Twitter. Another package needed is pandas, which is an open source, BSD-licensed package can provide high-performance, easy-to-use data structures and data analysis tools for the Python programming language (Raj & Preeti, 2019).

After successful installation of Jupyter, install the tweepy and pandas packages with pip installation as below:

```
(base) C:\Users\zhenw>pip install tweepy
Collecting tweepy
  Downloading tweepy-3.8.0-py2.py3-none-any.whl (28 kB)
Collecting requests-oauthlib>=0.7.0
  Downloading requests_oauthlib-1.3.0-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: requests>=2.11.1 in
c:\users\zhenw\miniconda3\lib\site-packages (from tweepy) (2.23.0)
Requirement already satisfied: six>=1.10.0 in
c:\users\zhenw\miniconda3\lib\site-packages (from tweepy) (1.14.0)
Requirement already satisfied: PySocks>=1.5.7 in
c:\users\zhenw\miniconda3\lib\site-packages (from tweepy) (1.7.1)
Collecting oauthlib>=3.0.0
  Downloading oauthlib-3.1.0-py2.py3-none-any.whl (147 kB)
     |████████████████████████████████| 147 kB 1.3 MB/s
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in c:\users\zhenw\miniconda3\lib\site-packages (from requests>=2.11.1-
>tweepy) (1.25.8)
Requirement already satisfied: chardet<4,>=3.0.2 in
c:\users\zhenw\miniconda3\lib\site-packages (from requests>=2.11.1-
>tweepy) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in
c:\users\zhenw\miniconda3\lib\site-packages (from requests>=2.11.1-
>tweepy) (2.9)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\zhenw\miniconda3\lib\site-packages (from requests>=2.11.1-
>tweepy) (2020.4.5.1)
Installing collected packages: oauthlib, requests-oauthlib, tweepy
Successfully installed oauthlib-3.1.0 requests-oauthlib-1.3.0 tweepy-
3.8.0
```

Figure 2. Installation of tweepy package.

```
(base) C:\Users\zhenw>pip install pandas
Collecting pandas
  Downloading pandas-1.0.3-cp37-cp37m-win_amd64.whl (8.7 MB)
     |████████████████████████████████| 8.7 MB 6.8 MB/s
Collecting pytz>=2017.2
  Downloading pytz-2019.3-py2.py3-none-any.whl (509 kB)
     |████████████████████████████████| 509 kB 3.2 MB/s
Collecting numpy>=1.13.3
  Downloading numpy-1.18.2-cp37-cp37m-win_amd64.whl (12.8 MB)
     |████████████████████████████████| 12.8 MB 6.8 MB/s
Requirement already satisfied: python-dateutil>=2.6.1 in
c:\users\zhenw\miniconda3\lib\site-packages (from pandas) (2.8.1)
Requirement already satisfied: six>=1.5 in
c:\users\zhenw\miniconda3\lib\site-packages (from python-
dateutil>=2.6.1->pandas) (1.14.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.18.2 pandas-1.0.3 pytz-2019.3
```

Figure 3. Installation of pandas package.

After installation of necessary packages, testing is needed under Jupyter Notebook environment. It means that the packages have been successfully installed if no result showing in the output.

```
import tweepy as tw
import pandas as pd
import json
```

Figure 4. Import libraries.

3.2 Obtaining API keys from Twitter Developer

The next step requires authentication for the application. Authentication is the process of confirming identity. To gain access to the Twitter API, Twitter account and Twitter application is needed to get credentials and API access as the following steps:

Step 1: Go to Twitter for developers https://developer.twitter.com/

Step 2: Sign in with registered Twitter account. Register an account if not exist. Main dashboard presents after signing in. The dashboard indicates monthly, weekly requests and account activity API, API usage information.

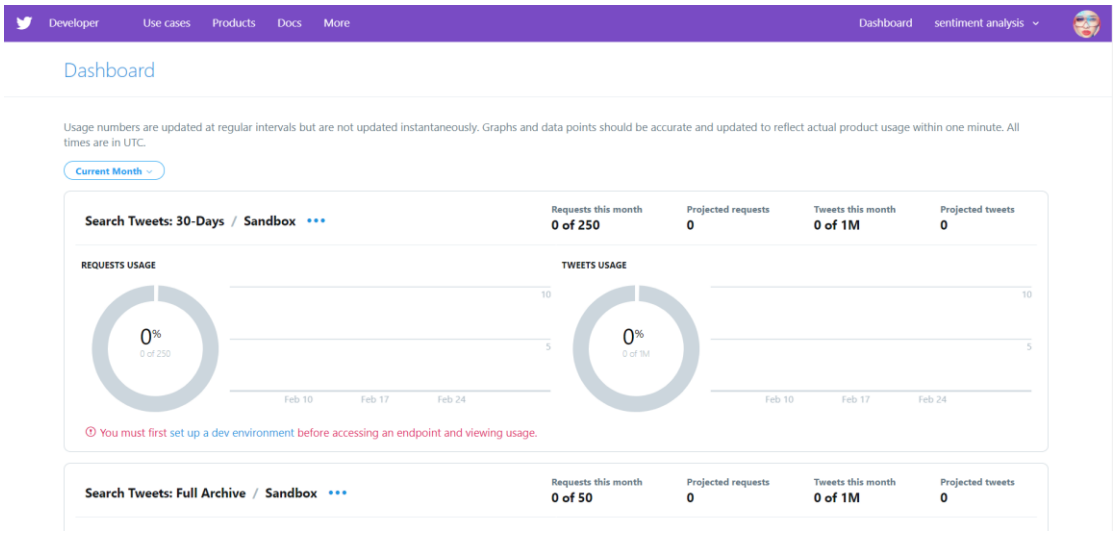

Figure 5. Sign in with registered Twitter account.

Step 3: To gain the access token, it is necessary to create an app with the app name, description, website URL, callback URLs and explanation of how this app will be used.

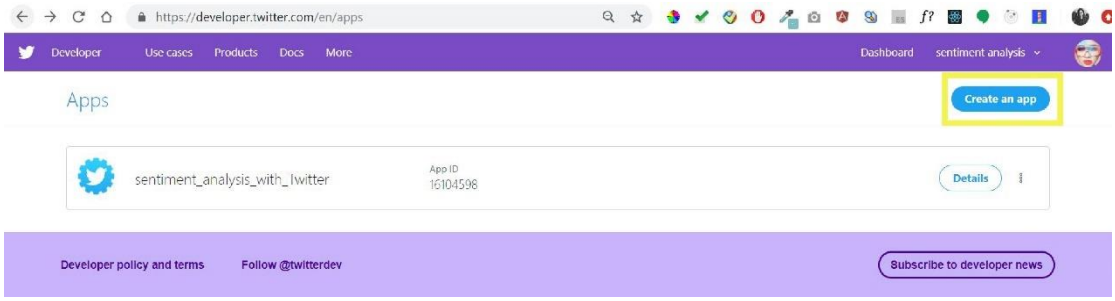

Figure 6. Create app.

Step 4: After the app has been created, under keys and tokens section, consumer API keys are needed by generating, same as access token and access token secret. With all the necessary tokens, Twitter API, credentials should be saved safely and protected from others.

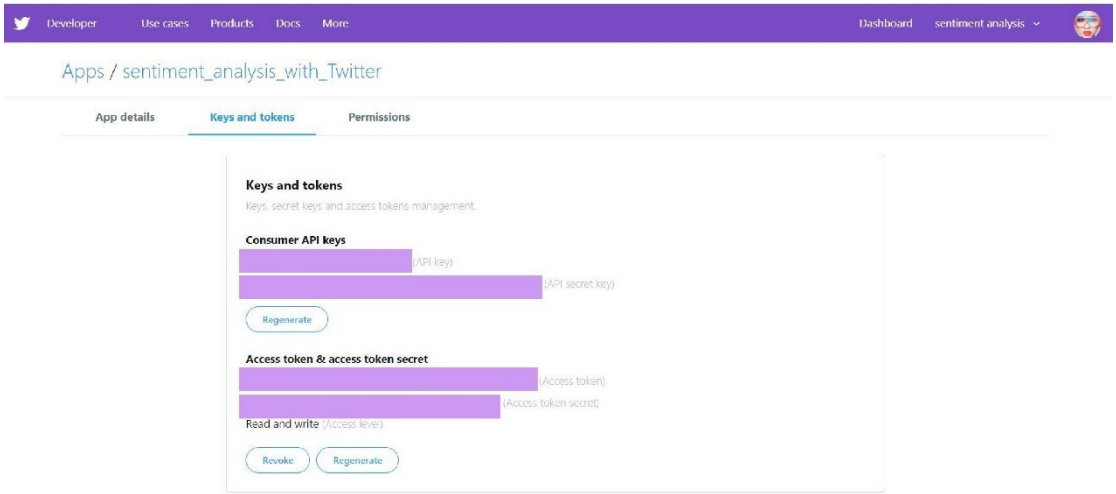

Figure 7. API keys and tokens.

3.3 Verifying authentication

3.3.1 Verifying credentials

In the previous step, credentials have been created with the Twitter developer entry and been saved in text file. This part needs to verify the credentials by taking the variables CONSUMER_KEY, CONSUMER_SECRET, ACCESS_TOKEN,

ACCESS_TOKEN_SECRET:

```
# App Auth

consumer_key = CONSUMER_KEY
consumer_secret = CONSUMER_SECRET
access_key = ACCESS_TOKEN
access_secret = ACCESS_TOKEN_SECRET
```

Figure 8. Application authentication.

3.3.2 Initialize API and test authentication

The package has been installed, tweepy, supports oauth authentication. As showing in the next part, authentication is handled by the *tweepy.AuthHandler* class.

```
# Initialize API

auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)

api = tw.API(auth)
```

Figure 9. API initialization.

After initializing API, need to test authentication using *verify credentials* method. If the authentication is successful, print "*Authentication successful*", otherwise print "*Authentication failed*".

```
# test authentication

try:
    api.verify_credentials()
    print("Authentication successful")
except:
    print("Authentication failed")
```

Figure 10. Test authentication.

As a result, the credentials are verified successfully showing ”Authentication successful”.

```
In [4]: # test authentication

        try:
            api.verify_credentials()
            print("Authentication OK")
        except:
            print("Error during authentication")

        Authentication OK
```

Figure 11. Output of test authentication.

3.4 Collecting data from Twitter

After making access to the Twitter API, next step is making request and retrieving information about any public users or tweets. Filters include also searching duration and search words.

3.4.1 Defining hashtags as search words and date as search period

In this thesis work, the subject being analyzed is tweets posted on the Twitter. Topics or content, which is related to climate changes can be retrieved and grouped by hashtags. Hashtag, which is introduced by the hash sign, #, is the way of showing specific topic or content. Here is the list of top 10 climate change hashtags, which are popular on social media of Instagram, Twitter, Facebook and Tumblr (Best-hashtags.com, ei pvm). In the work done in this thesis, first top 3 hashtags will be implemented: #climatechange, #environment, #globalwarming.

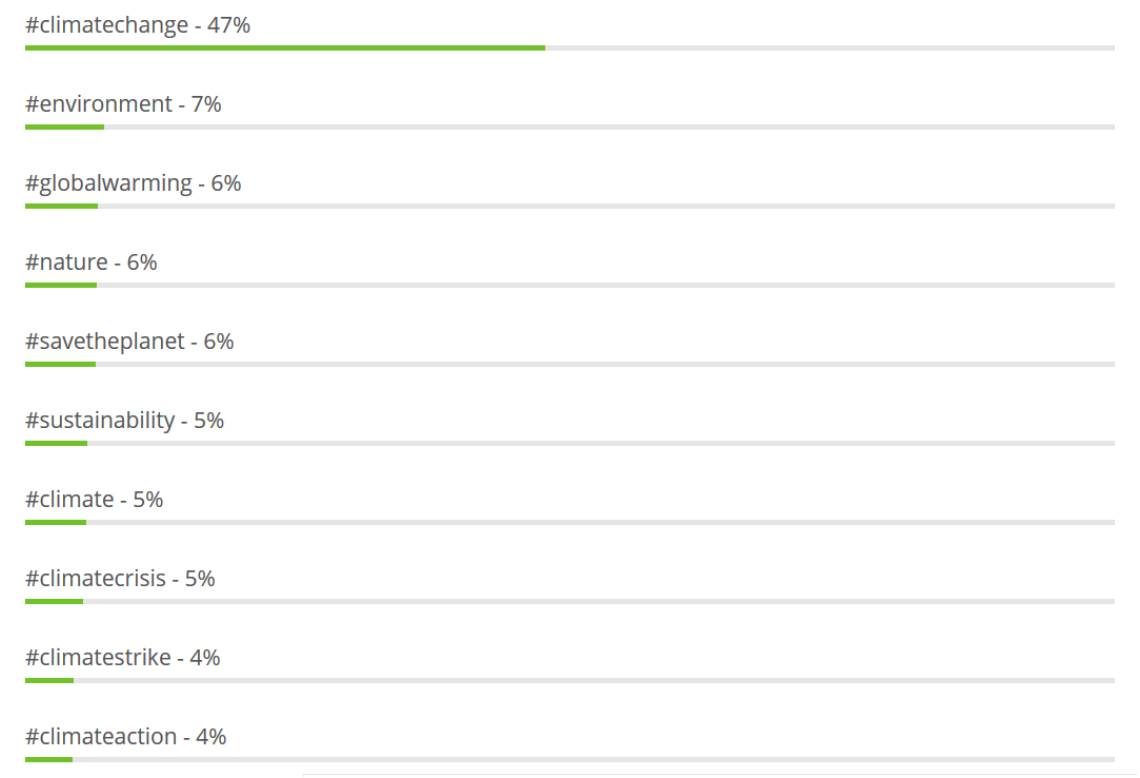

Figure 12. Top 10 climate change related hashtags and percentage of usage.

Number of tweets being analyzed: 100000 tweets

Starting date of tweets being analyzed: 1 January 2010

Assigning hashtag words to the *search_words* variable as an array.

```
# Search terms

search_words = ["#climatechange", "#environment", "#globalwarming"]

date_since = "2010-01-01"
```

Figure 13. Search terms and period definition.

3.4.2 Collecting tweets

Tweepy.Cursor method allows to get all tweets about a specific subject in a certain language in special time period. Assigning hashtag search words to query variable and define the language as English. Creating *tweets_arr* and assign as empty string.

```
# Collect tweets

tweets = tw.Cursor(api.search,
                   q=search_words,
                   lang="en",
                   since=date_since, tweet_mode='extended',
                   include_rts=True).items(100000)

tweets_arr = []
```

Figure 14. Collection of tweets using Twitter API.

```
# Iterate and print tweets

for tweet in tweets:
    tweets_arr.append(tweet._json)
print("Done")

with open("data.json", "w+") as output:
    output.write(json.dumps(tweets_arr))
```

Figure 15. Printing collected tweets.

In order to implement parsing easily, a `json` string should be saved into `data.json` file.

3.5 Preprocessing data

Preprocessing data refers to text normalization, which means clean the data to a normalized form for analyzing. This step converts human readable language to machine readable, which includes:

- Tokenization
- Stop Words Removal
- Morphological Normalization
- Collocation

By definition, tokenization is a method which splits longer strings of text into shorter pieces, or so called tokens (Mayo, 2020). This includes sentence tokenization and word tokenization. Sentence tokenization breaks longer text into sentences, whereas word tokenization breaks into smaller pieces as word. In this work done in this thesis, word tokenizations are used due to the small amount of text on tweet.

Removal of stop words is the process of removing stop words from the dataset. "Stop words" usually refers to the most common words in a language, which do not express any positive, negative or neutral opinions or emotions. Stopwords need to be excluded for better performance result of sentiment analysis. NLTK provides a rich list of stop words module called `corpus`. After word tokenization from previous step, word tokens can be checked and detected from the list of stop words provided by NLTK library.

Morphological normalization refers to normalizing the words using morphology methods. Morphology refers to the study how words are built up from smaller meaning-bearing units (morphomes) (Shahrul, et al., 2013). Two commonly used techniques are stemming and lemmatization. Stemming is the process of reducing a word to its stem, base or root format (Mayo, 2020). Similar as stemming, lemmatization aims to change the word to its base format, but it mostly applies lexical-based knowledge (Mayo, 2020).

3.5.1 Installing necessary packages

Following packages need to be installed for specific functions. This can be implemented by implementing `pip install nltk`, `pip install sklearn`, `pip install matplotlib`, `pip install seaborn`.

```
(base) C:\Users\zhenw>pip install nltk
Collecting nltk
  Downloading nltk-3.4.5.zip (1.5 MB)
     |████████████████████████████████| 1.5 MB 819 kB/s
Requirement already satisfied: six in
c:\users\zhenw\miniconda3\lib\site-packages (from nltk) (1.14.0)
Building wheels for collected packages: nltk
  Building wheel for nltk (setup.py) ... done
  Created wheel for nltk: filename=nltk-3.4.5-py3-none-any.whl
size=1449909
sha256=c4b527bb459f3fca736e7f53c52749a82e1ba23177a4b98eb2530bebd0947c9
5
  Stored in directory:
c:\users\zhenw\appdata\local\pip\cache\wheels\48\8b\7f\473521e0c731c65
66d631b281f323842bbda9bd819eb9a3ead
Successfully built nltk
Installing collected packages: nltk
Successfully installed nltk-3.4.5
```

Figure 16. Installation of nltk package.

```
(base) C:\Users\zhenw>pip install sklearn
Collecting sklearn
  Downloading sklearn-0.0.tar.gz (1.1 kB)
Collecting scikit-learn
  Downloading scikit_learn-0.22.2.post1-cp37-cp37m-win_amd64.whl (6.5
MB)
     |                                | 6.5 MB 2.2 MB/s
Requirement already satisfied: numpy>=1.11.0 in
c:\users\zhenw\miniconda3\lib\site-packages (from scikit-learn-
>sklearn) (1.18.2)
Collecting scipy>=0.17.0
  Downloading scipy-1.4.1-cp37-cp37m-win_amd64.whl (30.9 MB)
     |                                | 30.9 MB 3.2 MB/s
Collecting joblib>=0.11
  Downloading joblib-0.14.1-py2.py3-none-any.whl (294 kB)
     |                                | 294 kB 3.3 MB/s
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py) ... done
  Created wheel for sklearn: filename=sklearn-0.0-py2.py3-none-any.whl
size=1320
sha256=4889d9fba47a33365117e4d3d25234675ed51f96d19b41f566f13c1bd970dfc
c
  Stored in directory:
c:\users\zhenw\appdata\local\pip\cache\wheels\46\ef\c3\157e41f5ee1372d
1be90b09f74f82b10e391eaacca8f22d33e
Successfully built sklearn
Installing collected packages: scipy, joblib, scikit-learn, sklearn
Successfully installed joblib-0.14.1 scikit-learn-0.22.2.post1 scipy-
1.4.1 sklearn-0.0
```

Figure 17. Installation of sklearn package.

```
(base) C:\Users\zhenw>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.2.1-cp37-cp37m-win_amd64.whl (9.2 MB)
     |                                | 9.2 MB 3.3 MB/s
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Requirement already satisfied: numpy>=1.11 in
c:\users\zhenw\miniconda3\lib\site-packages (from matplotlib) (1.18.2)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.2.0-cp37-none-win_amd64.whl (57 kB)
     |                                | 57 kB 875 kB/s
Requirement already satisfied: python-dateutil>=2.1 in
c:\users\zhenw\miniconda3\lib\site-packages (from matplotlib) (2.8.1)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
     |                                | 67 kB 2.8 MB/s
Requirement already satisfied: six in
c:\users\zhenw\miniconda3\lib\site-packages (from cycler>=0.10-
>matplotlib) (1.14.0)
Installing collected packages: cycler, kiwisolver, pyparsing,
matplotlib
```

Figure 18. Installation of matplotlib package.

```
(base) C:\Users\zhenw>pip install seaborn
Collecting seaborn
  Downloading seaborn-0.10.0-py3-none-any.whl (215 kB)
     |████████████████████████████████| 215 kB 1.1 MB/s
Requirement already satisfied: matplotlib>=2.1.2 in
c:\users\zhenw\miniconda3\lib\site-packages (from seaborn) (3.2.1)
Requirement already satisfied: numpy>=1.13.3 in
c:\users\zhenw\miniconda3\lib\site-packages (from seaborn) (1.18.2)
Requirement already satisfied: scipy>=1.0.1 in
c:\users\zhenw\miniconda3\lib\site-packages (from seaborn) (1.4.1)
Requirement already satisfied: pandas>=0.22.0 in
c:\users\zhenw\miniconda3\lib\site-packages (from seaborn) (1.0.3)
Requirement already satisfied: python-dateutil>=2.1 in
c:\users\zhenw\miniconda3\lib\site-packages (from matplotlib>=2.1.2-
>seaborn) (2.8.1)
Requirement already satisfied:
pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
c:\users\zhenw\miniconda3\lib\site-packages (from matplotlib>=2.1.2-
>seaborn) (2.4.7)
Requirement already satisfied: cycler>=0.10 in
c:\users\zhenw\miniconda3\lib\site-packages (from matplotlib>=2.1.2-
>seaborn) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\zhenw\miniconda3\lib\site-packages (from matplotlib>=2.1.2-
>seaborn) (1.2.0)
Requirement already satisfied: pytz>=2017.2 in
c:\users\zhenw\miniconda3\lib\site-packages (from pandas>=0.22.0-
>seaborn) (2019.3)
Requirement already satisfied: six>=1.5 in
c:\users\zhenw\miniconda3\lib\site-packages (from python-
dateutil>=2.1->matplotlib>=2.1.2->seaborn) (1.14.0)
Installing collected packages: seaborn
Successfully installed seaborn-0.10.0
```

Figure 19. Installation of seaborn package.

3.5.2 Preprocessing data sources and instances

In order to preprocess the data, `stopwords` and `punkt` packages from NLTK need to be downloaded. A set of stop words in English is printed.

```
# NLP imports

import nltk
from nltk.tokenize import sent_tokenize
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
```

```
# sklearn imports

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
```

Figure 20. NLTK and sklearn packages import.

```
# preprocessing data sources and instances

nltk.download('stopwords')
nltk.download('punkt')
stop_words = set(stopwords.words("english"))
stemmer = PorterStemmer()
tf = TfidfVectorizer()
```

Figure 21. Preprocessing data sources and instances.

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\zhenw\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\zhenw\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Figure 22. Installation stopwords and punkt packages.

```
print(stop_words)
```

```
{"weren't", 'having', 'after', 'yourself', 'more', 'their', 'hers',
"she's", 'at', 'you', 'as', 'yours', 'these', 'can', 'other', 'do',
"you'll", 'once', 'ours', 'they', 'from', 'ourselves', 'before',
"you're", 'over', 'down', 'nor', "should've", "shan't", 'but', 'in',
'or', 'so', 'into', 'needn', 'how', 'with', 'who', 'haven', "don't",
've', 'them', 'until', 'by', 'about', 'to', 'themselves', 'ain',
'under', 'up', 'herself', 'for', 'are', 's', 'll', 'then', 'wouldn',
'had', "couldn't", 'being', 'such', 'each', 'd', 'i', 'did', 'there',
'y', "you've", 'whom', 'were', 'of', 'any', 'he', 'all', 'very',
'hasn', 'isn', 'further', 'shan', 'will', "shouldn't", 'against',
'because', 'is', 'doesn', 'and', 'his', 'aren', "you'd", 'was', 'ma',
"mustn't", "hasn't", 'an', "won't", 'your', "isn't", 'own', "it's",
'hadn', 'where', 'not', 'just', 'few', 'don', 'it', 're', 'a', 'no',
'm', 'this', 'am', 't', 'off', 'above', 'that', 'didn', 'itself',
'should', 'on', 'her', "that'll", 'yourselves', 'what', 'again',
"aren't", 'when', "mightn't", 'doing', 'some', 'have', 'wasn', 'o',
'same', "haven't", 'shouldn', 'than', 'while', 'too', 'we', 'through',
'its', 'him', 'if', 'our', 'been', 'between', "needn't", "wouldn't",
'couldn', 'here', 'me', 'below', 'won', 'during', 'himself', 'does',
'both', 'now', 'has', 'most', 'only', "didn't", 'be', "doesn't",
'mightn', 'why', 'weren', "wasn't", 'the', 'mustn', "hadn't",
'myself', 'those', 'she', 'theirs', 'my', 'which', 'out'}
```

Figure 23. Result of stop words in English from NLTK package.

3.5.3 Reading data and creating sentences from provided data source

In this thesis work, dataset for training uses *Multi-domain sentiment dataset (version 2.0).* This dataset contains product reviews taken from Amazon.com from multiple product types, which is referred to as multi-domain sentiment dataset. Domains like books and DVDs have more products than others like instruments. As a result, some domains have more reviews than others. All the reviews include star ratings which can be labeled for usage.

The dataset has the form showing as below:

`feature:<count> …. feature:<count> #label#:<label>.`

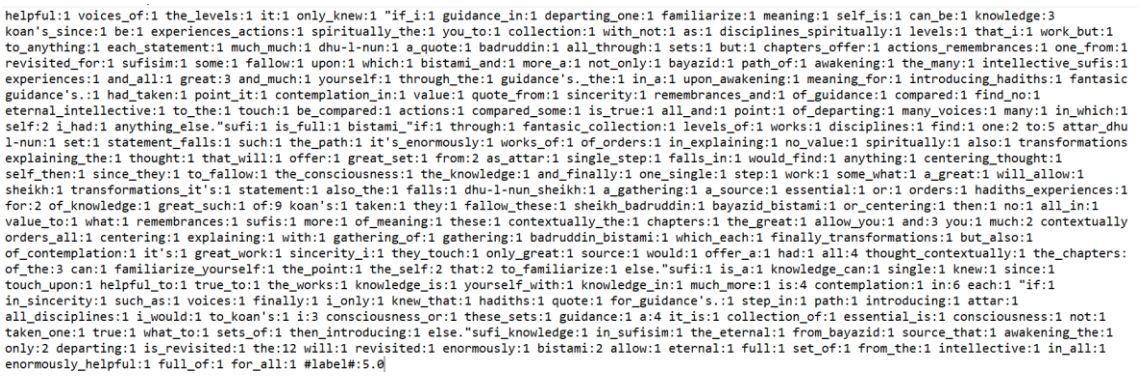

Figure 24. Snippet for dataset.

```
# Reading data and creating sentences from data source

with open("./data/processed_stars_all") as data:

    all_data = data.read().strip().split("\n")
    for n, line in enumerate(all_data):
        line_data = line.split(" ")[0:-1]
        sentence = ""
        for words_count in line_data:
            words = words_count.split(":")[0].split("_")
            count = int(words_count.split(":")[1])
            words = list(filter("<num>".__ne__, words))
            words = list(map(lambda x: re.sub(r'\`|\'|,|\.|\"', '',
x), words))

            if len(words) == 0:
                continue
            words = words * count
            sentence += " ".join(words) + " "
        sentences.append(sentence)
        Y.append(float(line.split(" ")[-1].split(":")[-1]))
```

Figure 25. Reading data and creating sentences from the data source.

```
In [20]: print(sentence)
```

```
i i i i i while living new york new york tedious boring vicariously dining found greenes a touch is not of was was wa
s reviews reviews checking page thoroughly enjoyed be speed reading there insatiable tasteless very event i reading g
ael a weekly for nothing too much i thoroughly romance about realized much information magazine her numbers dining ex
periences my id get however although hoping id experiences was just tasteless just too life savoring ability to while
experiences enjoyed restaurant reviews touch of enjoyed reading reichel insatiable to receiving very her rhapsodize i
was york magazine boring  forward to and and and page numbers and tasting get to for years york york to rhapsodize to
uch this much i looked positive midwest-quiet life numbers hoping to savoring chronicle and nothing endearing get for
ward is with a looked in in about the finesse and id speed new new with however the new the new reading reading altho
ugh gael greenes in the in the years i life however ruth reichel the chronicle york times romance dining tasting of t
oo living my to the and i was nothing found insatiable endearing end a hoping to to to to there was end restaurant ev
ent the end i realized her ability just boring very positive vicariously while to be rhapsodize finesse looked forwar
d reviews in reviews in although receiving my midwest-quiet a a a checking years this is not not weekly midwest-quiet
page reading checking positive reviews realized i thoroughly times was speed the the the the boring boring receiving
tasting vicariously information there be tedious weekly event i found endearing about living not tasteless ruth green
es restaurant a ruth gael information and romance ability magazine tedious was a finesse reichel this savoring and ti
mes i chronicle
```

Figure 26. Result of printing sentence variable.

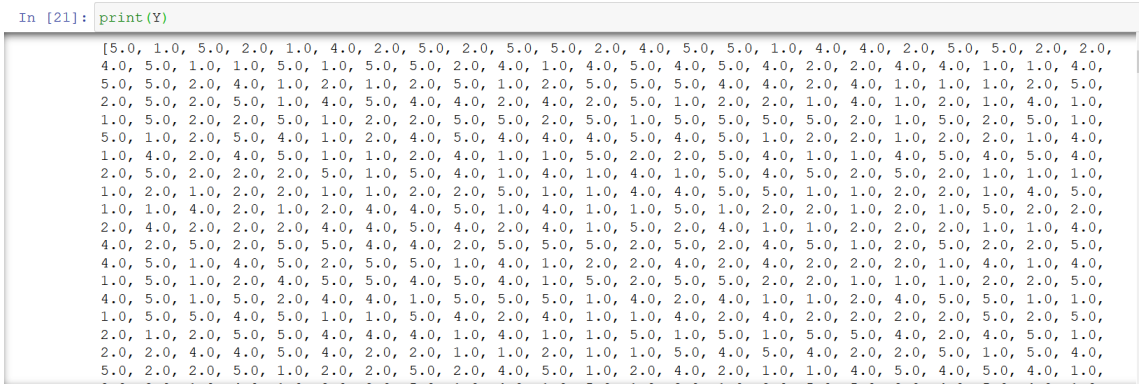

Figure 27. Result of printing Y variable.

3.5.4 Reading tweets data

This part reads tweet data using API. `full_text` object needed for collecting content, which indicates untruncated tweet message longer than 140 characters. Below is an example showing content of `full_text` in a tweet:

```
"full_text": "RT @PeterBrandt007: Save the earth. And earn valuable
rewards.\n\n(Evibacus princeps)\n\n#climatechange #environment
#globalwarming #savethepla\u2026"
```

Figure 28. full_text example from Twitter API.

Format of `full_text` is needed to get cleared by removing mentions such as RT, retweet or @ some user account, by moving hashtags and punctuations. By definition, a regular expression uses a special syntax in pattern and refers to a special sequence of characters matching strings.

```
# Reading twitter data

with open("data.json") as twd:
    data = twd.read()
    json = json.loads(data)

    for tw in json:
        text = tw['full_text']

         # remove mentions RT, retweet, @
         # remove hashtags #
         # remove punctuations ]+|,|\'|\"|\`|…

        text = re.sub(r'RT|@[a-zA-Z0-9_]+|:|#[a-zA-Z0-9_]+|[0-
9]+|,|\'|\"|\`|…', '', text)
        text = re.sub(r'  ', ' ', text.strip())
        sentences_processed_tw.append(text)
```

Figure 29. Reading Twitter data.

Printing tweet data after processing gets the following result:

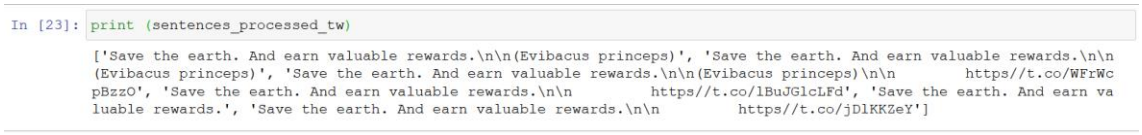

Figure 30. Sentences_processed_tw fragment.

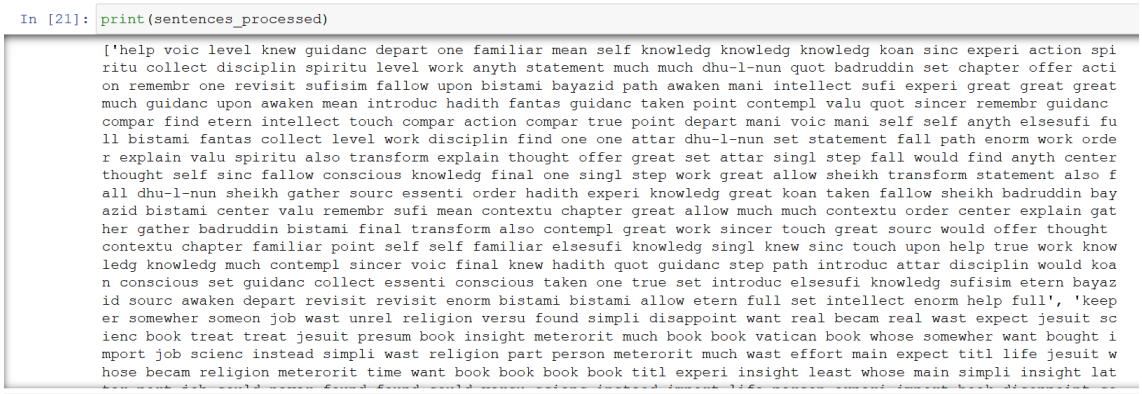

Figure 31. Sentences_processed fragment.

3.6 Training the model

This section creates a *df* (dataframe) for downstreaming analysis and applies *Multinomial Naive Bayesian Classifier*. The multinomial Naive Bayes classifier is a model that suits

for classification with discrete features (Anderson, 2019), which can be used for training the model. Since training dataset is retrieved from amazon.com comments and built up with reviewing stars from 1 to 5, *Multinomial Naive Bayesian Classifier* is the suitable model to train the data.

```
# Create a dataframe for downstream analysis

data = pd.DataFrame(data={'sentence': sentences_processed,
'sentiment': Y})

# Obtain TF-IDF transformation
text_tf = tf.fit_transform(data['sentence'])
```

Figure 32. Dataframe for downstream analysis.

As the result, printing data indicates 27570 rows and 2 columns which are sentence and sentiment.

```
In [25]: print(data)

                                                sentence  sentiment
0      help voic level knew guidanc depart one famil...        5.0
1      keeper somewher someon job wast unrel religion...        1.0
2      advic come close parent book close child reall...        5.0
3      would count system continu found simpl k-0 out...        2.0
4      yetunless sporat seem actual add amazon one bo...        1.0
...                                                  ...        ...
27565  see mean mean litter may spend empti pay autom...        2.0
27566  amount advic anyon make mess cup water counter...        2.0
27567  favorit well use favorit favorit knive knive k...        5.0
27568  pay nice featur name cuisinart effort less fea...        2.0
27569  what krup < dash-num > krup < dash-num > maker...        1.0

[27570 rows x 2 columns]
```

Figure 33. Data fragment.

`train_test_split` is a function for splitting data arrays into two subsets: dataset for testing and dataset for training.

```
train_test_split(X, y, train_size=0.*,test_size=0.*, random_state=*)
```

The model training score is 0.6666666, which indicates that Multinomial Naive Bayesian classifier model accuracy is 67%.

```
# Split data into Test-Train sets

X_train, X_test, y_train, y_test = train_test_split(
    text_tf, data['sentiment'], test_size=0.001, random_state=1)

clf = MultinomialNB().fit(X_train, y_train)
predicted = clf.predict(X_test)

print("Multinomial Naive Bayesian Classifier Model Accuracy:",
metrics.accuracy_score(y_test, predicted))
```

Figure 34. Splitting data into test-train datasets.

```
NB Model Accuracy: 0.6666666666666666
Analysing the twitter data
```

Figure 35. Model accuracy.

```
# TF-IDF transform of trained corpus

text_tf_tw = tf.transform(sentences_processed_tw)
predicted_tw = clf.predict(text_tf_tw)

print(Counter(predicted_tw))
sns.distplot(predicted_tw, kde=False, norm_hist=False)
plt.savefig("hist.png", dpi=600)
plt.figure()
sns.distplot(predicted_tw)
plt.savefig("hist_kde.png", dpi=600)
plt.show()
```

Figure 36. TF-IDF transform of trained corpus.

```
Counter({1.0: 14666, 4.0: 5348, 2.0: 4610, 5.0: 2530})
```

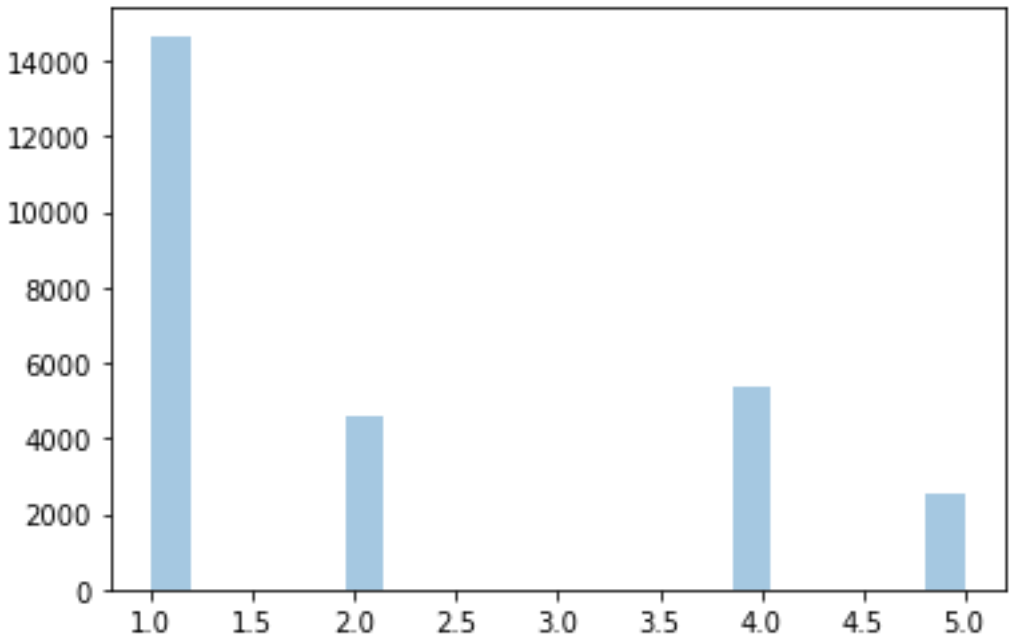

Figure 37. Counter histogram of search word #climatechange.

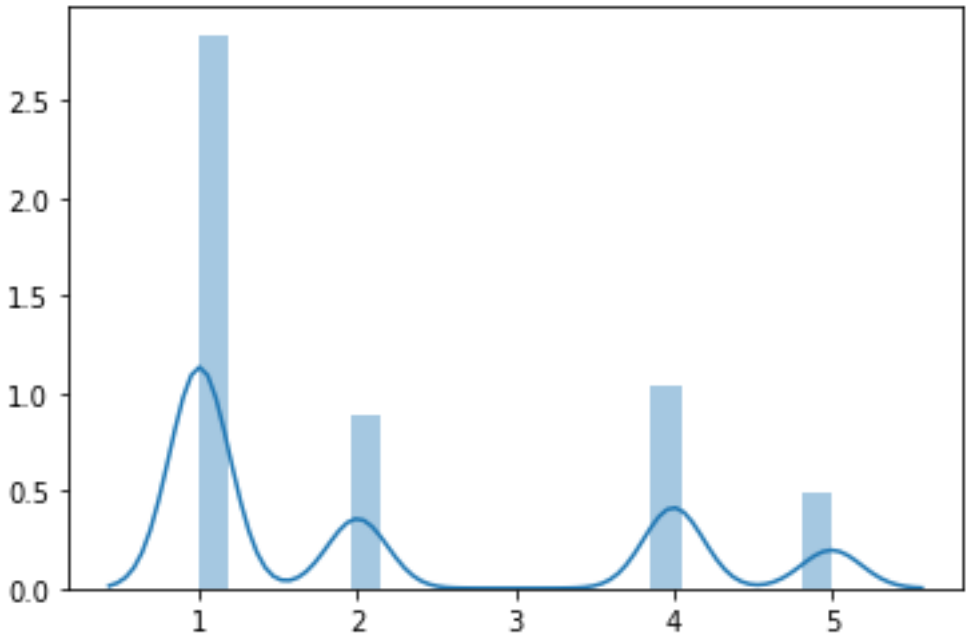

Figure 38. KDE histogram of search word #climatechange.

To include, as showing in 2 histogram figures above, 1-5 indicates negative-neutral-positive scores. 1 refers to negative sentiment / opinion whereas 5 refers to positive sentiment / opinion. 3 indicates neutral sentiment / opinion since 3 locates in the center. Detecting from the height of the bars, score 1 takes the most counter of 14666, which

means a large group of people have negative sentiment on the topics related to climate change. However, score 5 takes 2530 counters, which means even a small group of people have positive opinion on climate change. This group has an optimistic attitude towards this topic. It is clear to see that people have opinions either negative or positive. This phenomenon clearly represents the polarized nature of the society.

4 CONCLUSION

In this chapter, the conclusion of this thesis work is listed and summarized. The aim for current thesis work was to perform the sentiment analysis of global warming using Twitter data. In order to achieve the result, full processes is carried out to achieve the result. Instead of the traditional integrated development environment, the work done in this thesis uses Jupyter Notebook for superior interactive visualization features. Implementation of API initialization and functions uses Twitter API credentials and needed features. Specifically, dataset for training uses *Multi-domain sentiment dataset (version 2.0),* which contains product reviews taken from Amazon.com from multiple product types. This method provides general sentiment analysis of multi-domain data, which can be used in this thesis work. The training model uses multinomial Naive Bayes classifier, which suits for classification with discrete features (Anderson, 2019).

As a result, the training model of Naive Bayes classifier accuracy is 67%. In summary, based on the sentiment analysis, a large group of people have negative sentiment on the topics related to climate change. They are worried about the current situation and consider climate change as a severe threat to the planet. However, a small group of people has an optimistic attitude on this and hope it will become better. The findings have indicated that more attention should be taken on climate change topics.

REFERENCES

Anderson, B., 2019. *Pattern Recognition: An introduction.* s.l.:Scientific e-Resources.

Best-hashtags.com, n.d. *Hashtags For #Climatechange On Instagram, Twitter, Facebook, Tumblr.* [Online] Available at: http://best-hashtags.com/hashtag/climatechange/ [Accessed 9 April 2020].

Bisht, M., 2016. *Document Classification using Multinomial Naïve Bayes Classifier.* [Online] Available at: https://www.3pillarglobal.com/insights/document-classification-using-multinomial-naive-bayes-classifier [Accessed 22 April 2020].

Collomb, A. L. B. C. C., 2013. *A Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation.* [Online] Available at: https://www.semanticscholar.org/paper/A-Study-and-Comparison-of-Sentiment-Analysis-for-Collomb-Brunie/0e6e72c40f438c5c0e5c7ca47448d57e0a8c6e54 [Accessed 22 April 2020].

Liu, B., 2015. *A Survey of Opinion Mining and Sentiment Analysis.* s.l.:University of Illinois at Chicago.

Liu, B., 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions.* 1 ed. s.l.:Cambridge University Press.

Mayo, M., 2020. *A General Approach To Preprocessing Text Data.* [Online] Available at: https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html [Accessed 9 April 2020].

Mucha, N., 2017. *Sentiment analysis of global warming using Twitter data.* [Online] Available at: https://library.ndsu.edu/ir/bitstream/handle/10365/28166/Sentiment%20Analysis%20of%20Global%20Warming%20Using%20Twitter%20Data.pdf?sequence=1&isAllow

ed=y
[Accessed 27 April 2020].

Raj, K. & Preeti, S., 2019. *Big Data Analytics: Introduction to Hadoop, Spark, and Machine-Learning.* s.l.:McGraw-Hill Education.

Shahrul, N. A., Siti, A. N. H. S. & Haslina , A., 2013. *Soft Computing Applications and Intelligent Systems: Second International Multi-Conference on Artificial Intelligence Technology.* Springer ed. s.l.:s.n.

Statista, 2020. *Number Of Social Media Users Worldwide 2010-2021.* [Online] Available at: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/
[Accessed 7 April 2020].

Tutorialspoint.com, 2020. *Python - Regular Expressions.* [Online] Available at: https://www.tutorialspoint.com/python/python_reg_expressions.htm
[Accessed 11 April 2020].

Dataset for training model

helpful:1 voices_of:1 the_levels:1 it:1 only_knew:1 "if_i:1 guidance_in:1 departing_one:1 familiarize:1 meaning:1 self_is:1 can_be:1 knowledge:3 koan's_since:1 be:1 experiences_actions:1 spiritually_the:1 you_to:1 collection:1 with_not:1 as:1 disciplines_spiritually:1 levels:1 that_i:1 work_but:1 to_anything:1 each_statement:1 much_much:1 dhu-l-nun:1 a_quote:1 badruddin:1 all_through:1 sets:1 but:1 chapters_offer:1 actions_remembrances:1 one_from:1 revisited_for:1 sufisim:1 some:1 fallow:1 upon:1 which:1 bistami_and:1 more_a:1 not_only:1 bayazid:1 path_of:1 awakening:1 the_many:1 intellective_sufis:1 experiences:1 and_all:1 great:3 and_much:1 yourself:1 through_the:1 guidance's._the:1 in_a:1 upon_awakening:1 meaning_for:1 introducing_hadiths:1 fantasic:1 guidance's.:1 had_taken:1 point_it:1 contemplation_in:1 value:1 quote_from:1 sincerity:1 remembrances_and:1 of_guidance:1 compared:1 find_no:1 eternal_intellective:1 to_the:1 touch:1 be_compared:1 actions:1 compared_some:1 is_true:1 all_and:1 point:1 of_departing:1 many_voices:1 many:1 in_which:1 self:2 i_had:1 anything_else."sufi:1 is_full:1 bistami_"if:1 through:1 fantasic_collection:1 levels_of:1 works:1 disciplines:1 find:1 one:2 to:5 attar_dhu-l-nun:1 set:1 statement_falls:1 such:1 the_path:1 it's_enormously:1 works_of:1 of_orders:1 in_explaining:1 no_value:1 spiritually:1 also:1 transformations:1 explaining_the:1 thought:1 that_will:1 offer:1 great_set:1 from:2 as_attar:1 single_step:1 falls_in:1 would_find:1 anything:1 centering_thought:1 self_then:1 since_they:1 to_fallow:1 the_consciousness:1 the_knowledge:1 and_finally:1 one_single:1 step:1 work:1 some_what:1 a_great:1 will_allow:1 sheikh:1 transformations_it's:1 statement:1 also_the:1 falls:1 dhu-l-nun_sheikh:1 a_gathering:1 a_source:1 essential:1 or:1 orders:1 hadiths_experiences:1 for:2 of_knowledge:1 great_such:1 of:9 koan's:1 taken:1 they:1 fallow_these:1 sheikh_badruddin:1 bayazid_bistami:1 or_centering:1 then:1 no:1 all_in:1 value_to:1 what:1 remembrances:1 sufis:1 more:1 of_meaning:1 these:1 contextually_the:1 chapters:1 the_great:1 allow_you:1 and:3 you:1 much:2 contextually:1 orders_all:1 centering:1 explaining:1 with:1 gathering_of:1 gathering:1 badruddin_bistami:1 which_each:1 finally_transformations:1 but_also:1 of_contemplation:1 it's:1 great_work:1 sincerity_i:1 they_touch:1 only_great:1 source:1 would:1 offer_a:1 had:1 all:4 thought_contextually:1 the_chapters:1 of_the:3 can:1 familiarize_yourself:1 the_point:1 the_self:2 that:2 to_familiarize:1 else."sufi:1 is_a:1 knowledge_can:1 single:1 knew:1 since:1 touch_upon:1 helpful_to:1 true_to:1 the_works:1 knowledge_is:1 yourself_with:1 knowledge_in:1 much_more:1 is:4

contemplation:1 in:6 each:1 "if:1 in_sincerity:1 such_as:1 voices:1 finally:1 i_only:1 knew_that:1 hadiths:1 quote:1 for_guidance's.:1 step_in:1 path:1 introducing:1 attar:1 all_disciplines:1 i_would:1 to_koan's:1 i:3 consciousness_or:1 these_sets:1 guidance:1 a:4 it_is:1 collection_of:1 essential_is:1 consciousness:1 not:1 taken_one:1 true:1 what_to:1 sets_of:1 then_introducing:1 else."sufi_knowledge:1 in_sufisim:1 the_eternal:1 from_bayazid:1 source_that:1 awakening_the:1 only:2 departing:1 is_revisited:1 the:12 will:1 revisited:1 enormously:1 bistami:2 allow:1 eternal:1 full:1 set_of:1 from_the:1 intellective:1 in_all:1 enormously_helpful:1 full_of:1 for_all:1 #label#:5.0

i:3 keeper:1 somewhere:1 to_someone:1 job_now:1 wasted:1 unrelated:1 religion_versus:1 was_and:1 are:1 of:5 or:2 now_at:1 found:1 is_simply:1 disappointed:1 wanted_to:1 real:1 so_became:1 was:2 real_waste:1 this:1 be:2 expected:1 a_jesuit:1 science:1 book_is:1 are_'treated':1 'treated'_to:1 jesuit:1 or_presumed:1 at:2 as:1 book_with:1 insights_into:1 of_meterorites:1 an:1 much_of:1 into:1 the_book's:1 book's:1 of_the:2 of_what:1 the_vatican:1 this_book:1 whose:1 or_somewhere:1 wanted:1 and:4 bought_it:1 important:1 only_job:1 with_at:1 science_instead:1 simply:1 the_waste:1 and_the:2 to_the:1 religion:1 as_i:1 it_a:1 part_of:1 personal:1 meterorites_much:1 wasted_efforts:1 main:1 i_was:1 expected_a:1 title:1 life:1 do_so:1 jesuit_whose:1 became:1 into_religion:1 what_was:1 meterorites:1 time:1 he_wanted:1 book:3 book's_title:1 and_only:1 experience:1 some_insights:1 least_some:1 whose_main:1 we:1 simply_his:1 insights:1 the_latter:1 part:1 job:1 could:1 never_found:1 it:1 found_what:1 and_could:1 only:1 versus_science:1 in:2 what:2 instead:1 have:1 who:1 to_do:1 very:1 important_life:1 personal_experience:1 an_important:1 is:1 a_book:1 very_disappointed:1 in_this:1 could_have:1 instead_we:1 what_he:1 keeper_of:1 he:1 reason:1 vatican_seems:1 be_an:1 to:5 continued:1 a_real:1 reason_i:1 main_and:1 never:1 time_and:1 efforts:1 waste_of:2 'treated':1 with:1 the_keeper:1 waste:2 to_be:2 at_least:1 book_as:1 so:1 someone_who:1 have_continued:1 disappointed_in:1 presumed_focus:1 a:3 now:1 his:1 the_reason:1 someone:1 continued_to:1 focus_and:1 we_are:1 antartica:1 became_a:1 the:7 focus:1 experience_in:1 his_personal:1 the_book:1 antartica_or:1 presumed:1 was_very:1 much:1 in_antartica:1 some:1 do:1 vatican:1 totally:1 least:1 i_expected:1 at_the:1 versus:1 latter_part:1 be_the:1 unrelated_to:1 somewhere_totally:1 i_bought:1 seems:1 who_never:1 totally_unrelated:1 latter:1 bought:1 seems_to:1 title_or:1 of_time:1 #label#:1.0

a:3 advice:1 into_my:1 and_does:1 come_close:1 parenting_books:1 close:1 child_and:1 really_relate:1 or:1 book_has:1 and_gives:1 on:2 books_were:1 most_of:1 real:1 of:4 her:3 this:1 success:1 read:1 does:1 children:1 many_parenting:1 book_that:1 for:4 fit_the:1 that_relates:1 discipline_would:1 into:1 really:2 real_help:1 my:4 to_plan:1 to_helping:1 for_success:1 level:1 this_book:1 books_that:1 her_on:1 mold:1 didn't_fit:1 me:4 good:1 "types"_of:1 she:1 of_most:1 of_children:1 might:1 child:3 gives:1 just:1 what_might:1 her_level:1 better:1 has_given:2 child_or:1 really_good:1 has:2 insight_into:1 to_her:1 work:2 but:1 finally_real:1 and:4 advice_on:1 read_so:1 discipline:1 written_for:1 that:2 would_work:1 her_she:1 for_but:1 of_discipline:1 book:2 finally_a:1 type:1 or_what:1 might_and:1 what_type:1 on_her:1 me_understand:1 mold_of:1 me_a:2 gives_me:1 relate_to:1 to_really:1 child_this:1 did:1 some_really:1 were:1 helping_me:1 for_her:1 but_finally:1 level_and:1 have:1 insight:1 chance_to:1 come:1 finally:2 given:2 a_book:1 what:2 close_to:1 good_advice:1 fit:1 help:1 work_for:2 children_those:1 child's_personality:1 plan:1 most:1 those_books:1 a_better:1 did_not:1 parenting:1 to:5 the_mold:1 given_me:2 written:1 personality_and:1 relates_to:1 understand_my:1 on_what:1 for_my:1 type_of:1 i:1 so:1 were_written:1 books:2 the_"types":1 "types":1 my_child's:1 not:1 that_did:1 relate:1 so_many:1 my_child:3 plan_for:1 child's:1 have_read:1 understand:1 to_my:1 a_chance:1 me_some:1 the:2 didn't:1 and_to:1 and_has:1 she_just:1 i_have:1 does_work:1 not_come:1 some:1 many:1 chance:1 helping:1 better_insight:1 would:1 of_the:1 those:1 just_didn't:1 personality:1 relates:1 #label#:5.0

it_would:1 count_system:1 continually:1 found:1 simple_k-0:1 by:1 you_outdated:1 knew:1 and_even:1 book_has:1 original:1 thorpe's:1 most_of:1 contains_no:1 theory:1 of:2 isn't_a:1 this:2 by_the:1 be:1 surpassed:1 powerful_and:1 that_the:1 even:1 no:1 you:3 the_popular:1 popular:1 outdated_info:1 outdated:1 really:1 even_if:1 i_found:1 didn't_it:1 has_been:1 system_presented:1 this_book:1 terrible_one:1 you_didn't:1 powerful:1 bad:1 bland_but:1 giving_you:1 contains:1 surpassed_by:1 feeling_that:1 the_unsettling:1 terrible:1 info:1 better:1 k-0:1 elsewhere_eg:1 explains:1 simple:1 found_i:1 it_explains:1 but:2 and:3 practice_of:1 book_isn't:1 explains_the:1 has:1 been:1 k-0_system:1 book:3 feeling:1 presented_in:1 one:1 unsettling_feeling:1 more_powerful:1 get:1 you_continually:1 continually_get:1 it:2 is:1 giving:1 this_information:1 in:1 card_counting:1 if:1 i_knew:1 bad_information:1 eg_ed:1 bland:1 isn't:1 thorpe's_original:1 in_the:1 been_surpassed:1 unsettling:1 learned_elsewhere:1 the_theory:1 most:1 card:1 learned:1 personally_i:1 personally:1 practice:1 of_card:1 counting_but:1 one_it:1 and_more:1 of_this:1 that:1 ed_thorpe's:1 but_contains:1 i:2

better_learned:1 count:1 elsewhere:1 a:1 more:2 original_the:1 is_giving:1 really_bad:1 the:6 didn't:1 information_and:1 and_practice:1 presented:1 eg:1 the_book:2 ed:1 popular_more:1 book_is:1 if_you:1 be_better:1 knew_most:1 information:2 counting:1 more_simple:1 system:2 a_terrible:1 but_personally:1 would:1 system_you:1 would_be:1 no_really:1 get_the:1 the_count:1 theory_and:1 #label#:2.0

yet.unless:1 and_sporatic:1 it:5 seems_to:1 of_actual:1 them_i:1 add:1 by:3 amazon_for:1 one_at:1 just_by:1 bought_it:1 bought_and:1 harp:1 actual:1 be:2 informative_too:1 collecting:1 at:1 as:2 an:2 that_i:1 list.there:1 at_that:1 times:1 and_it:1 informative.there_are:1 seem:1 and_they:1 <num>:2 but:2 they_seem:1 only_a:1 is_only:1 writing_a:1 won't_mention:1 that_is:1 bought:2 list_or:1 reason_as:1 left:1 like:1 if_it:1 seem_to:1 it_used:1 it_really:1 collecting_cookbooks:1 reason:1 are_only:1 add_this:1 want_to:1 amazon:1 rhyme:1 rhyme_or:1 like_writing:1 outdated:2 will_not:1 weren't:1 through_them:1 is:4 better_by:1 book_even:1 around:1 the_list:1 author_and:1 not_harp:1 why_books:1 around_$3.i:1 and:4 collection_do:1 buy:1 why:1 through_with:1 i'm:1 not_buy:1 with_them:1 just:2 a_negative:1 to:5 by_merely:1 several:1 used:1 the_only:1 informative.there:1 collection:1 no_rhyme:1 by_<num>:1 $3.i_will:1 even:1 not_through:1 merely_glancing:1 buy_this:1 by_the:1 not_like:1 this:3 books_are:1 understood:1 negative_review:1 and_an:1 from:1 or_left:1 text:1 merely:1 them_since:1 for_around:1 out:1 since_i'm:1 better:1 for_collecting:1 are:3 is_understood:1 lacking:1 copyright:1 books_for:1 that_informative.there:1 this_was:1 date:1 do_not:2 with:1 want:1 plus_about:1 negative:1 or:2 actual_text:1 on:1 as_an:1 for:2 of:2 <fraction>_pages:1 they:1 book:1 an_incomplete:1 this_as:1 them:3 no:1 weren't_outdated:1 was:1 copyright_date:1 be_no:1 in_the:1 you_want:1 out_of:1 your:1 only_<num>:1 several_other:1 them_yet.unless:1 through:2 books:2 list.there_are:1 i_do:1 you:1 to_why:1 plus:1 text_from:1 harp_that:1 the_author:1 understood_by:1 won't:1 too:1 review:1 but_it:1 are_several:1 incomplete:1 writing:1 to_your:1 used_on:1 glancing:1 other:1 seems:1 of_the:1 that:5 just_bought:1 informative:1 be_better:1 glancing_through:1 really:1 author:1 since:1 one:1 are_in:1 about_this:1 review_but:1 lacking_book:1 times_just:1 in:1 <fraction>:1 sporatic_one:1 if:1 pages_of:1 as_to:1 is_not:1 yet.unless_you:1 other_books:1 a_list:1 cookbooks:1 this_to:1 your_collection:1 or_reason:1 list:2 to_be:2 left_out:1 that_it:1 but_the:1 outdated_that:1 i:4 a:3 the_list.there:1 it_is:2 is_outdated:1 on_amazon:1 not:5 only_plus:1 about:1 was_i:1 <num>_times:1 i_bought:1 sporatic:1 i'm_not:1 pages:1 the_copyright:1 an_informative:1 cookbooks_that:1 really_is:1 $3.i:1 to_add:1 not_that:1 only:3 the:5 mention_them:1 <num>_<fraction>:1 date_but:1 will:1 i_won't:1 do:2 that_seems:1

i_just:1 incomplete_and:1 it_weren't:1 from_the:1 even_if:1 a_lacking:1 list_and:1 mention:1 #label#:1.0

opportunities_exist:1 <num>_billion:1 of_article:1 marketplace_of:1 serves_poor:1 their:1 powerless:1 charged:1 trying:1 ownership:1 informal:1 global_scale:1 world's_poorest:1 by:5 how_can:1 thinking_that:1 by_engagement:1 generating:1 be:1 benefit:2 models_another:1 ignore_a:1 pyramid:1 hammondhow_can:1 profiting:1 as:3 corporations:2 authors_make:1 mainstream:1 that_runs:1 an:1 global_marketplace:1 prices_in:1 developed_markets:1 than_by:1 technologies:1 from_sweeping:1 musings_to:1 profitably:2 counter:1 as_incubators:1 collection:2 it_they:1 approach:1 reasons_why:1 efficient:1 bop:2 musings:1 tool:1 why_major:1 powerful_institutions:1 get:1 ways_than:1 might_just:1 than_simply:1 deploy:1 whole:1 year:1 for_the:2 cutting:1 tool_for:1 our_time:1 who_compete:1 poorer_customers:1 tone:1 opens:1 reaching_poorer:1 market_where:1 of_the:3 the_least:1 the:23 into_new:1 of_topics:1 another:1 more_ways:1 article_is:1 cover:1 flavor_for:1 vary:1 areas:1 from_use:1 serving_the:1 ignore:1 to_serve:1 hand_how:1 more_affluent:1 serve_a:1 distasteful:1 potential_customers:1 make:1 might:1 summarize_this:1 engaging_and:1 philosophical:1 this_collection:1 distasteful_element:1 tone_from:1 multinationals:1 to_be:1 is_a:2 how:3 well:1 serve:3 case_that:1 done_correctly:1 business:2 sales_bop:1 whole_collection:1 by_efficient:1 and_profiting:1 saturated:1 wireless_technologies:1 cover_a:1 more_from:1 lead-off:1 exist:1 goods:1 lead-off_i:1 summarize:1 <year>_or:1 much_higher:1 marketplace:1 need:1 have:1 use_by:1 as_well:1 market:2 flavor:1 allen_hammondhow:1 same_goods:1 incubators_for:1 also:1 overcome_the:1 piece_in:1 multinationals_than:1 world's_poor:1 less:1 corporate_social:1 communities_served:1 compete_for:1 authors:1 the_case:1 some_of:1 affluent_communities:1 articles:2 away_from:1 (bop)_markets:1 the_world's:2 fresh:1 served_by:1 just_get:1 the_pyramid:1 the_poor:1 broad_range:1 of_our:1 innovative:1 and_those:1 done:1 the_most:3 higher:1 prahalad:1 have_in:1 this_piece:1 when_bop:1 niche:1 why:2 for_good:1 use:3 imagining:1 who:2 edge_wireless:1 communities:3 hopes:1 pyramid_(bop):1 generating_additional:1 hopes_of:1 being_ignored:1 reasoning_and:1 makers_who:1 in_hopes:1 collection_serving:1 first:1 philosophical_musings:1 the_isolation:1 why_companies:1 developed_and:1 to_de-aggregate:1 those_who:1 adapted_into:1 front:1 ways:1 revitalize_productivity:1 pieces:1 engagement:1 is_to:2 benefit_in:1 social:1 and_leadership:1 de-aggregate:1 saturated_markets:1 are:2 on_<year>:1 in_front:1 that_can:1 giving:1 correctly:1 from_the:1 and_saturated:1 <num>:2

poor_communities:1 exploring:1 in_more:2 time_engaging:1 an_innovative:1 hand:1 make_the:1 productivity:1 reaching:1 to_overcome:1 another_strategy:1 for:6 illustrate_how:1 suffer:1 this:4 rural:1 institutions_of:1 strategy_that:1 responsibility:1 as_a:1 typical_strategies:1 market_of:1 decision:1 you_a:1 reasons:1 and:8 c.k.:1 from:5 leadership:1 they_illustrate:1 most_economically:1 article:3 where_consumers:1 eight:1 <num>_articles:1 additional:1 companies_have:1 charged_by:1 multinational_corporations:1 new_thinking:1 major:1 rigorous_academic:1 billion_potential:1 least:1 very_strong:1 engagement_with:1 profiting_from:1 de-aggregate_ownership:1 imagining_multinational:1 can:5 they:1 serves:1 new_products:1 bottom:1 thinking_is:1 ideas_in:1 (bop):1 for_their:1 most_cutting:1 or_less:1 a_broad:1 the_authors:1 that_will:1 by_exploring:1 poorer:1 serve_the:1 additional_sales:1 live_on:1 and_approaches:1 bottom_of:1 the_other:1 consumers:1 from_trying:1 of_giving:1 to_ignore:1 can_companies:2 are_done:1 correctly_corporations:1 when:1 that_serves:1 a_very:1 markets_why:1 major_growth:1 typical:1 isolation:1 this_article:1 world's:2 distribution_when:1 prahalad_:1 of_<num>:1 corporations_also:1 most_powerful:1 can_revitalize:1 global:2 illustrate:1 case:1 that:5 least_developed:1 rural_areas:1 than:3 exist_in:1 interesting:1 article_explores:1 per:2 the_global:2 the_eight:1 most:3 leadership_in:1 academic:1 runs_counter:1 first_article:1 economically_powerless:1 of_decision:1 pay_per:1 approaches:1 good:1 prices:2 and_vary:1 academic_pieces:1 eight_articles:1 strong_lead-off:1 more:3 being:1 informal_economy:1 customers:2 consumers_live:1 strategies_are:1 on_corporate:1 per_year:1 past_shied:1 to_rigorous:1 poor_rural:1 with:1 need_to:1 incubators:1 and_might:1 poorest_communities:1 strategies_need:1 companies_afford:1 innovative_approach:1 much:1 customers_is:1 technologies_in:1 explores:1 pay:1 from_being:1 less_per:1 trying_to:1 benefit_the:1 cutting_edge:1 ownership_from:1 rigorous:1 bop_markets:1 corporations_the:1 potential:1 explores_reasons:1 multinational:1 edge:1 a_distasteful:1 allen:1 to:10 exploring_pay:1 the_multinationals:1 powerless_the:1 in_tone:1 companies:3 the_bottom:1 decision_makers:1 poor:4 simply:1 past:1 can_serve:1 poorest:1 thinking:2 markets_can:1 who_can:1 for_reaching:1 vary_in:1 their_business:1 productivity_and:1 profitably_serve:1 markets_as:1 than_prices:1 products:1 very:1 powerful:1 reasoning:1 time:1 higher_for:1 the_whole:1 the_past:1 sales:1 opens_interesting:1 economically:1 lines_of:1 efficient_distribution:1 piece:1 for_new:1 that_the:1 are_typically:1 affluent:1 will_benefit:1 economy_that:1 same:1 niche_and:1 ignored_by:1 new:2 isolation_of:1 opportunities:1 or:1 some:1 on:4 profitably_:1 billion:1 developed:2 into:1 engaging:1

of:11 giving_you:1 element_to:1 strong:1 to_deploy:1 other_hand:1 approach_for:1 business_as:1 runs:1 live:1 range:1 and_how:1 topics_and:1 where:1 with_it:1 of_reasoning:1 corporate:1 article_opens:1 overcome:1 those:1 hammondhow:1 pieces_the:1 ideas_and:1 bop_strategies:1 economy:1 a_market:2 interesting_lines:1 serve_as:1 a_flavor:1 the_developed:1 in_the:3 the_informal:1 communities_and:1 on_the:2 element:1 by_the:2 afford_to:1 be_adapted:1 adapted:1 typically:1 growth:1 front_of:1 business_there:1 there:1 simply_generating:1 growth_opportunities:1 deploy_some:1 lines:1 other:1 use_models:1 articles_cover:1 counter_to:1 markets:4 poor_suffer:1 our:1 will:2 also_benefit:1 strategies:2 c.k._prahalad:1 can_use:1 suffer_more:1 models:1 ignored:1 sweeping_philosophical:1 year_on:1 <year>:1 social_responsibility:1 articles_on:1 of_poor:1 broad:1 it:1 is:4 get_fresh:1 how_typical:1 in:9 customers_this:1 products_ideas:1 serving:1 the_same:1 communities_are:1 try:1 to_mainstream:1 try_to:1 served:1 goods_than:1 collection_of:1 to_summarize:1 ideas:2 how_prices:1 distribution:1 there_is:1 scale:1 i_will:1 just:1 prices_charged:1 i:1 will_try:1 wireless:1 a:7 institutions:1 markets_to:1 well_an:1 typically_much:1 companies_profitably:1 shied_away:1 sweeping:1 revitalize:1 you:1 away:1 use_business:1 good_on:1 makers:1 this_niche:1 scale_the:1 topics:1 the_first:1 shied:1 per_use:1 in_this:1 a_tool:1 range_of:1 to_imagining:1 mainstream_thinking:1 compete:1 afford:1 poor_profitably:1 strategy:1 approaches_that:1 fresh_ideas:1 #label#:4.0

5 labeled datasets, the rest of the data are listed in:

https://drive.google.com/file/d/1KxYJSKM5kU8xPBxj3QVxCDzl57oqvRuw/view?usp=sharing