Project 3

Web APIs & Classification

Linus, Clarence, Ganesh, Willy

Content

1) Overview
2) Data Gathering / Exploration
3) Cleaning
4) Baseline modelling (Naive Bayes)
5) Alternative Models
6) Key findings & problems
7) Conclusions

Business Objective

- Helping the company to better classify posts into their individual categories

Problem Statement

- Scrape posts from 2 Sub-Reddit forums and develop a Natural Language Processing model that can accurately identify which Sub-Reddit forum a post belongs to.

Data Collection

1. Identified the URL (WorldNews + Today I Learned)

url = 'https://www.reddit.com/r/worldnews/.json'

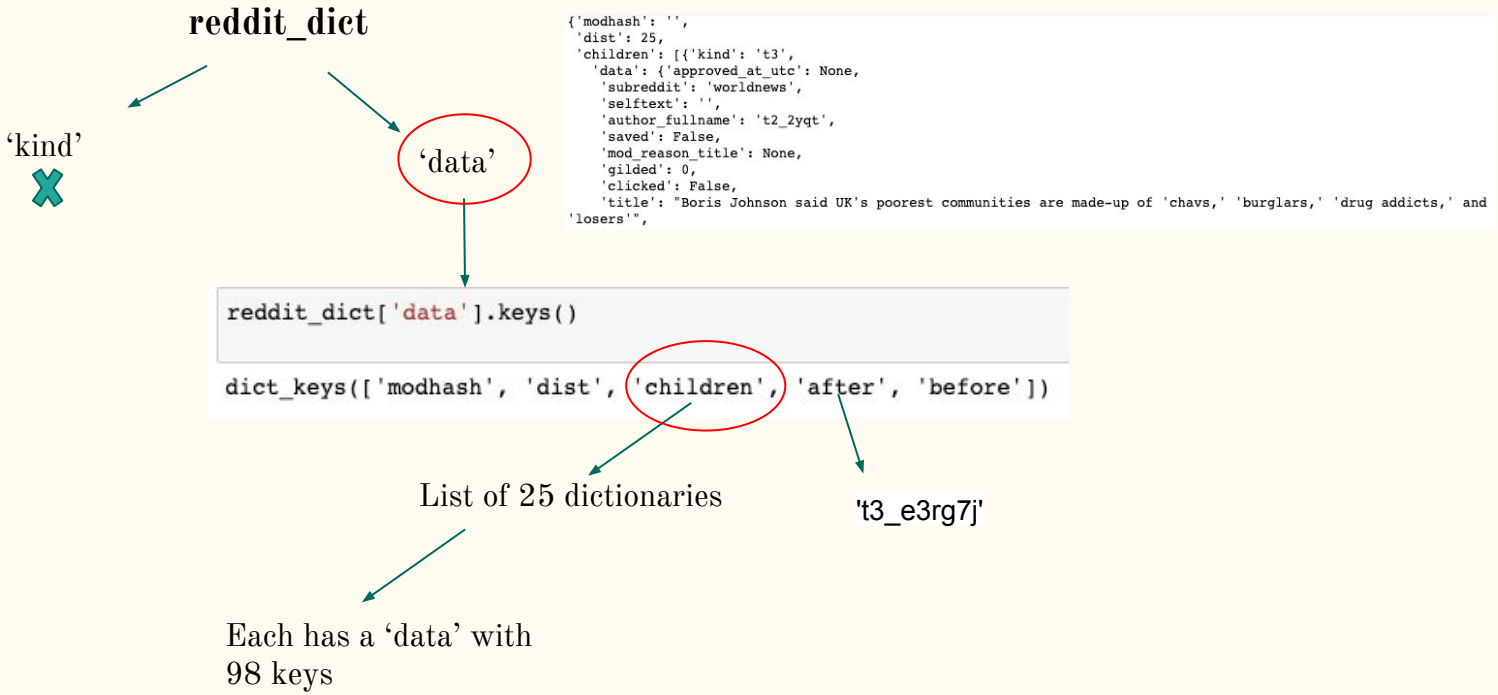
2. Called the API

res = requests.get(url)

3. Converted the API output to .json

reddit_dict = res.json()

Data Exploration



Data Exploration

List of 25 dictionaries

	approved_at_utc	subreddit	selftext	author_fullname	saved	mod_reason_title	gilded	clicked	title
0	None	worldnews		t2_2yqt	False	None	0	False	Boris Johnson said UK's poorest communities ar...
1	None	worldnews		t2_612zd	False	None	0	False	One of Malta's wealthiest men, Yorgen Fenech, ...
2	None	worldnews		t2_174cr0	False	None	1	False	Over 1,000 climate protesters storm German coa...

	subreddit	title
0	worldnews	Boris Johnson said UK's poorest communities ar...
1	worldnews	One of Malta's wealthiest men, Yorgen Fenech, ...
2	worldnews	Over 1,000 climate protesters storm German coa...
3	worldnews	Thousands demand Netanyahu's resignation at Te...
4	worldnews	Chinese diplomat clashes with BBC over definit...

Data Exploration

1. Accessed a sub-reddit url

2. Path of interest:

   reddit_dict['data']['children'] > 25 dictionaries > ['data']['title']

3. 1 x API call = 25 subreddit posts

   40 calls <= 1000 subreddit posts

Data Collection

```
posts = pd.DataFrame()
after = None

for i in range(4):
    if after == None:
        params = {}
    else:
        params ={'after' : after}

    url = 'https://www.reddit.com/r/worldnews/.json'

    res = requests.get(url, params = params, headers={'User-agent': 'Pony Inc 1.0'})

    if res.status_code == 200:
        the_json = res.json()

        wn_df = pd.DataFrame(the_json['data']['children'])
        wn_df = wn_df.data.apply(pd.Series)
        wn_df = wn_df[['subreddit','title']]

        posts = pd.concat([posts, wn_df])

        after = the_json['data']['after']
        print(i)

    else:
        print(res.status_code)
        break

    time.sleep(1)
```

Pseudo Code:

-If there is an after code, take note of it, otherwise, ignore

-Use the following URL to make an API call

-Ensure that the call is successful

-Access the relevant values in from the API output

-Append that output to a table of contents

-Store the after code to access the next 25 posts

-Sleep before repeating

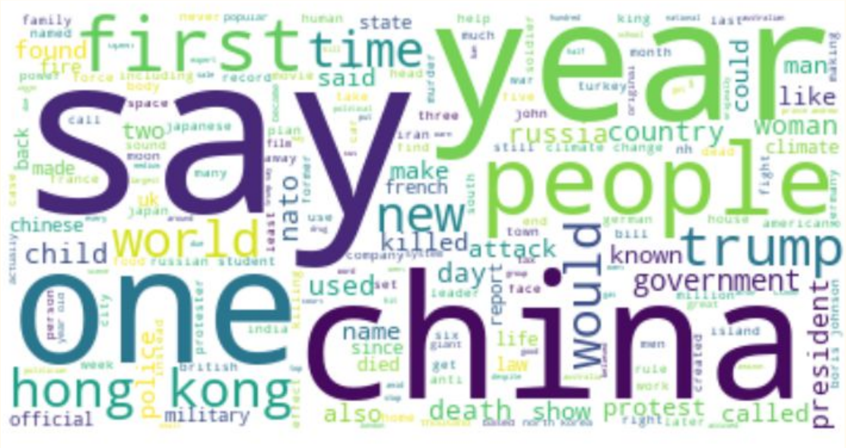
Raw Scrapes


year
China
people
one
TIL
London Bridge
Japan
died
originally
President
world
instead
called
known
use
city
fire
French
still
used
American
named
found
day
death
man
say
first
time
new
Hong Kong
protest
government
state
killed
attack
two
US
UK
police
children
home
name
made
make
said
three
month
million
show
protesters
many
will
car
end
group
Trump
terrorist
military
Germany
take
part
despite
Great
actually
movie
right
thousand
British
including
former
official
human
body
island
person
country
back
leaderelection
Boris Johnson

Minus stop words

til	year	people	london	death	china	say	world	used	time
743	97	66	60	58	56	55	51	49	48

Minus ‘Til’


say
year
first
time
people
one
china
new
world
trump
would
hong kong
government
death
show
protest
called
also
president
nato
child
chinese
official
military
police
used
day
attack
killed
known
country
russia
woman
climate
like
could
man
said
state
family
found
fire
make
french
two
made
back
uk
name
since
died
life
law
boris johnson


people
one
first
year
man
time
new
city
killed
protest
police
american
death
called
hong kong
china
used
say
london bridge
attack
japan
woman
world
day
million
group
two
trump
made
died
back
known
official
president
name
found
child
island
government
country
election
russia
british
terrorist
named
japanese
state
show
make
french
uk
take
body
never
movie
former
murder
boris johnson
germany
said
home

Data Cleaning

The following steps were performed to clean our data to get it ready for modeling:

1. Strip HTML - Remove any remnants of HTML after the scraping
2. Remove Accented Characters - e.g é, ó
3. Expand contractions - You'll to you will, y'all to you all
4. Lowercases the text - Convert everything to lowercase for uniformity
5. Remove extra newlines - Remove any lingering "/"
6. Lemmatize the text - Break each word down to its root word - "Playing to play"
7. Removes special characters & digits - Any punctuation marks or digits
8. Removes stopwords - Remove the most common words to get rid of the 'noise'

Data Cleaning

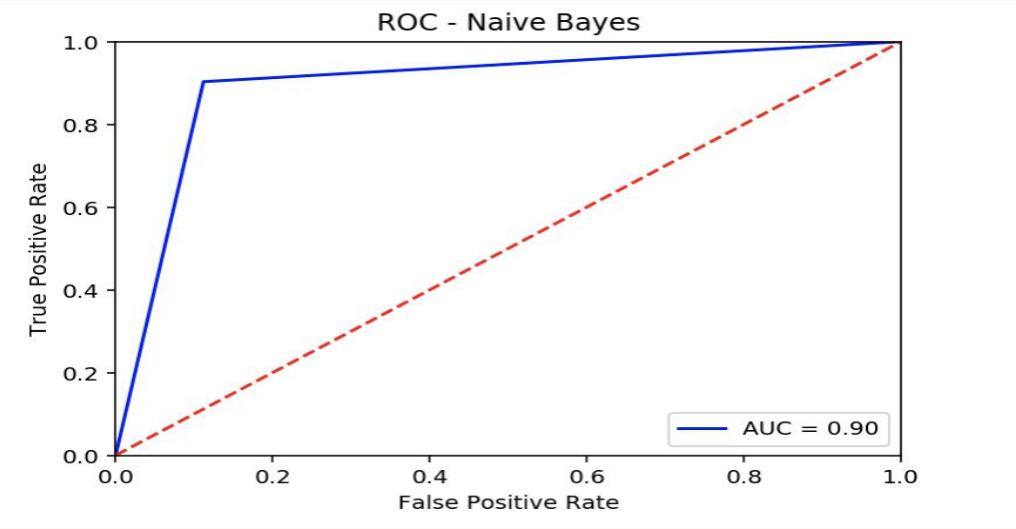
The effects of the transformation:

Original post : '\'We Don\'t Know a Planet Like This\': CO2 Levels Hit 415 PPM for 1st Time in 3 Million+ Yrs - "How is this not breaking news on all channels all over the world?"'

After Cleaning: not know planet like co level hit ppm st time million yrs not break news channel world

Baseline Modelling - MultinomialNB

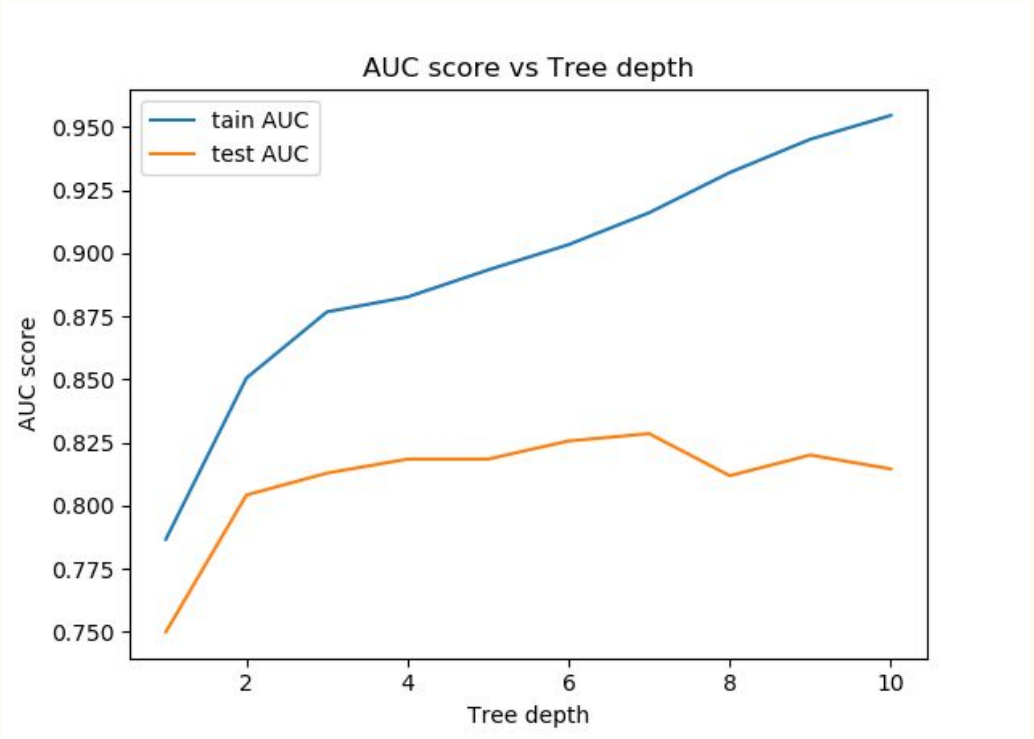
- Relatively simple model compared to some of the other classifiers but faired better than some of the other models we fitted
- Model fitted using a Countvectorizer with a max frequency of 0.3 to deal with words with high frequency in a subreddit. E.g “TIL, today, learned”



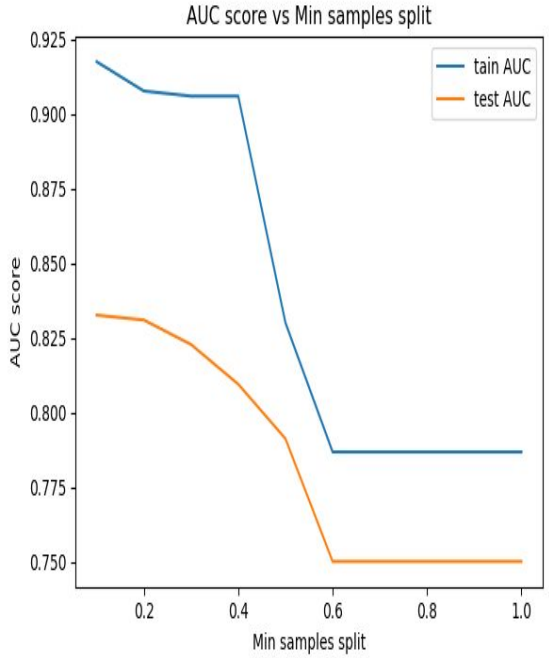
Alternative Models - Decision Tree Classifier 1

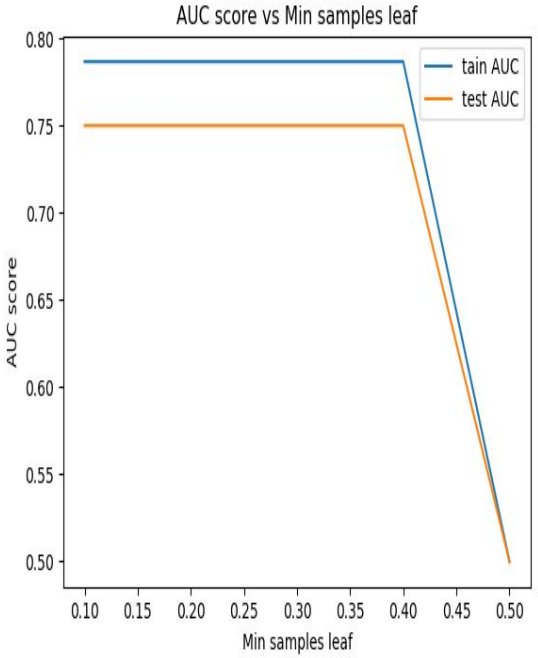
Hyperparameter tuning using AUC scores for:

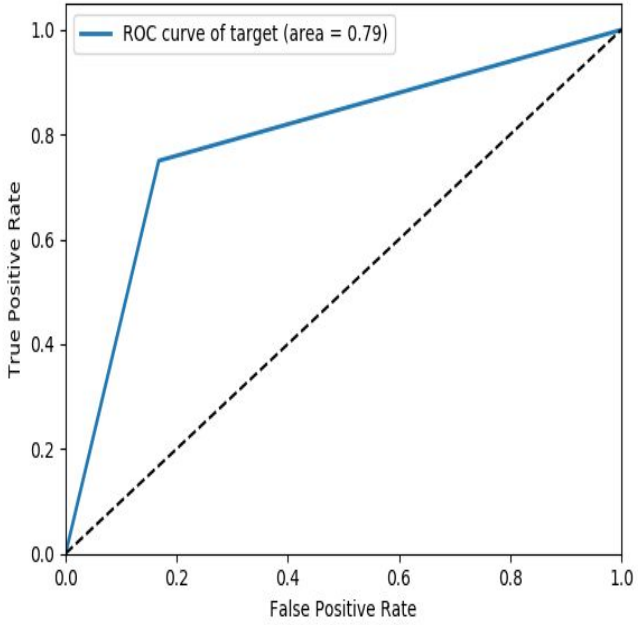
- Tree depth
- Minimum samples for split
- Minimum samples for leaf



Alternative Models - Decision Tree Classifier 2







Alternative Models - Others

VotingClassifier comprises:

- Multinomial Naive Bayes
- ExtraTreesClassifier
- RandomForestClassifier
- AdaBoostClassifier
- Logistic Regression

model_name	split_test_score	ext_test_score
VotingClassifier	0.854484	0.903585
LogisticRegression	0.832487	0.873651
ExtraTreesClassifier	0.840948	0.854159
RandomForestClassifier	0.813875	0.855204
AdaBoostClassifier	0.825719	0.863209
MultinomialNB	0.861252	0.895580

Key Findings

- Removal/Non-Removal of stopwords
  - Lower accuracy (Below 0.9)
  - TIL

- Correlation of stopwords
  - Length of titles

Key Problems

- Lower accuracy for future classification
  - Number of new posts increasing
  - New words/acronyms
  - Content of posts (Diverse topic/Similarity)

Conclusions

- Web Scraping subreddits
  - (WorldNews, Today I Learned)
- Data Exploration / Cleaning
  - (Subreddit, Title)
- Natural Language Processing
  - (Stopwords, Lemmatize, HTML, Punctuation, Lowercase, TIL)
- Modelling
  - (VotingClassifier, Naive Bayes)
- Evaluation
  - (VotingClassifier)