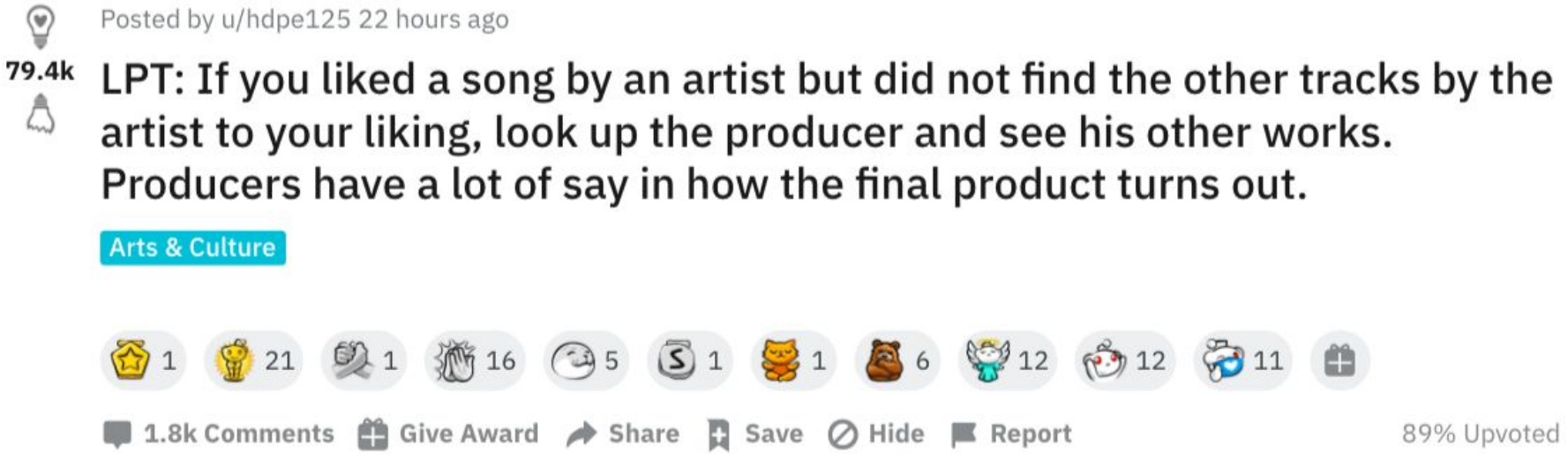
Classifying Reddit Posts

LifeProTips & UnethicalLifeProTips

Presentation by Alexis Lim

What is a Life Pro Tip?

What is a Life Pro Tip?



What is a Life Pro Tip?

r/LifeProTips

Tips that improve your life in one way or another.

18.4m
Members

18.1k
Online

Created 25 Oct 2010

LPT vs ULPT

r/LifeProTips

Tips that improve your life in one way or another.

18.4m
Members

18.1k
Online

Created 25 Oct 2010

r/UnethicalLifeProTips

An Unethical Life Pro Tip (or ULPT) is a tip that improves your life in a meaningful way, perhaps at the expense of others and/or with questionable legality. Due to their nature, do not actually follow any of these tips–they're just for fun. Share your best tips you've picked up throughout your life, and learn from others!

1.1m
Members

249
looking for unethical tips

Created 1 Mar 2016

The problem statement

How can we qualify what exactly makes a pro tip **unethical**?

Using a **classification model,** are we able to accurately predict posts for the two subreddits? We want to investigate:

- If we can **define** what is ‘unethical’ or anti-social behaviour (according to Reddit!)

- If we can **identify** significant differences between the two subreddits’ posts, given their objectives

The process

Getting the data from Reddit

Processing our data

Exploratory data analysis

Modeling & tuning

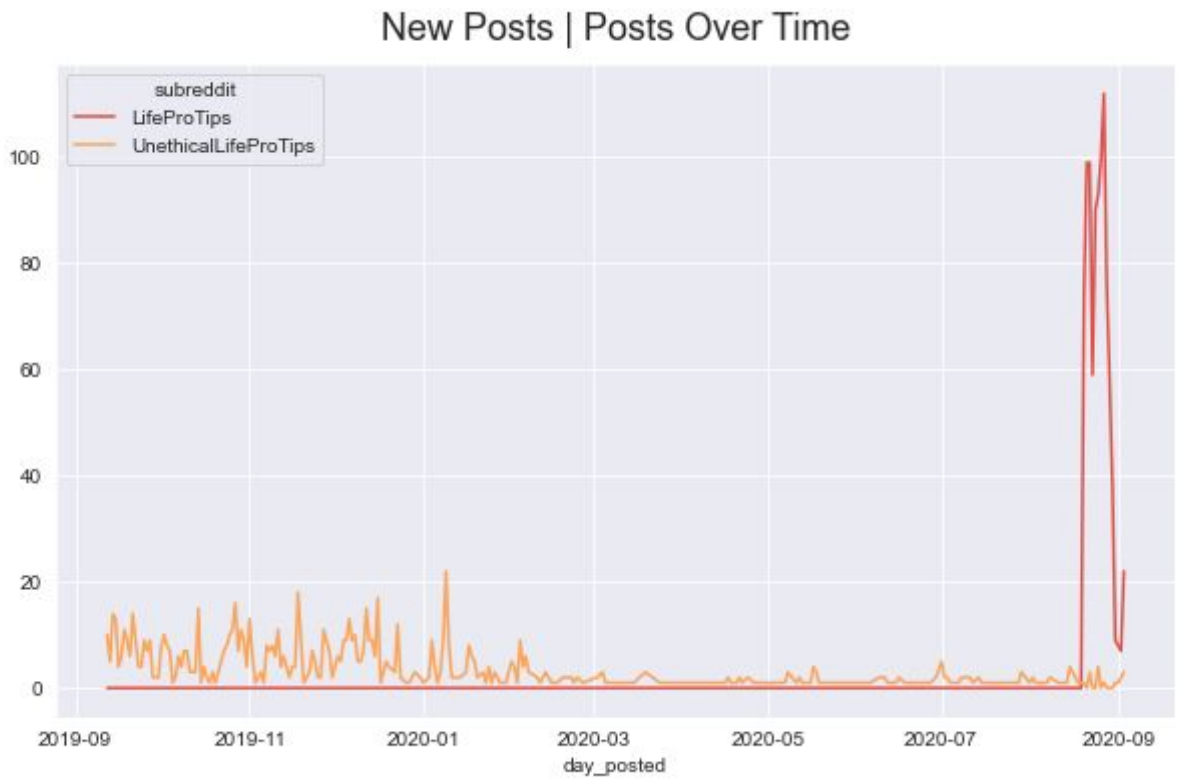
Interpretation & conclusions

Getting data from Reddit

Two ways of getting data:

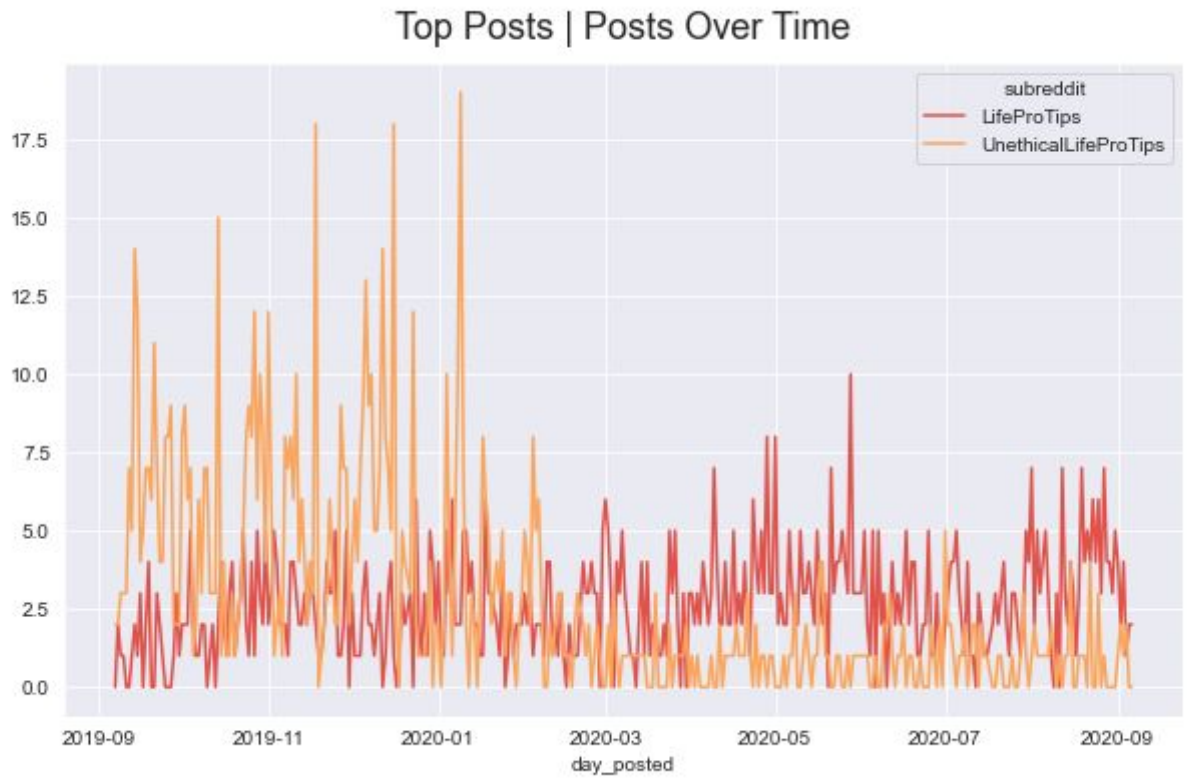
- **Webscraping,** using .json format on Reddit
  - Structured like a python dictionary and easy to parse
  - Relies on Reddit URL, depending on subreddit can be difficult to get posts
  - Can't access comments
- **PRAW**: The Python Reddit API Wrapper
  - Slightly more steps and different structure
  - Can access more things like comments, posts sorted by top etc.!

Scraping from new posts versus top posts



The LPT subreddit is much more active than the ULPT subreddit

Scraping from new posts versus top posts



Taking **the top 1000 posts for the past year** gives us a better representation

Features of interest

What features are we interested in retrieving?

- Subreddit name
- Text features (selftext, title)
- ID (name)
- Post features (number of comments, score, time created etc.)

Features of interest

	subreddit	selftext	score	title	upvote_ratio	created_utc	id	fullname	num_comments
0	LifeProTips	I had a phone interview scheduled this morning...	165153	LPT: keep your mouth shut, and don't volunteer...	0.96	1.582156e+09	f6jt5e	t3_f6jt5e	4973
1	LifeProTips	NaN	131334	LPT: If you want a smarter kid, teach your chi...	0.92	1.585245e+09	fpfwra	t3_fpfwra	3103
2	LifeProTips	I just found this quote online and wanted to s...	107877	LPT: Just because you did something wrong in t...	0.94	1.588878e+09	gfcq4g	t3_gfcq4g	2036

Removing duplicates

	subreddit	selftext	score	title	upvote_ratio	created_utc	id	fullname	num_comments	total_awards_received	author
7	LifeProTips		97685	LPT If you ever forget your WiFi password or y...	0.95	1.568801e+09	d5vknk	t3_d5vknk	2908	13	None
733	LifeProTips	Edit: Wow 2 silver! My first time, thanks kind...	3627	LPT If you ever forget your WiFi password or y...	0.96	1.575529e+09	e6dlbm	t3_e6dlbm	270	2	slackftw

Creating new features

- **Target variable:** Mapping ULPT posts to the positive class and LPT posts to the negative class

```
master_df_top['subreddit_cat'] = master_df_top['subreddit'].map({'LifeProTips': 0, 'UnethicalLifeProTips': 1})
```

- **Time:** Converting timestamp to EST and extracting day, hours, etc.

```
master_df_top['timestamp'] = master_df_top['created_utc'].apply(lambda x: datetime.fromtimestamp(x))
master_df_top['timestamp'] = master_df_top['timestamp'].dt.tz_localize('UTC')
master_df_top['timestamp'] = master_df_top['timestamp'].dt.tz_convert('US/Eastern')
```

- **Text:** Combining our title and selftext variables

```
master_df_top['selftext'] = master_df_top['selftext'].fillna("")
master_df_top['all_text'] = master_df_top['title'] + " " +  master_df_top['selftext']
```

Preparing our text for analysis

LPT: When a friend is upset, ask them one simple question before saying anything else: 'Do you want to talk about it or do you want to be distracted from it?' This is honestly one of the best thin...

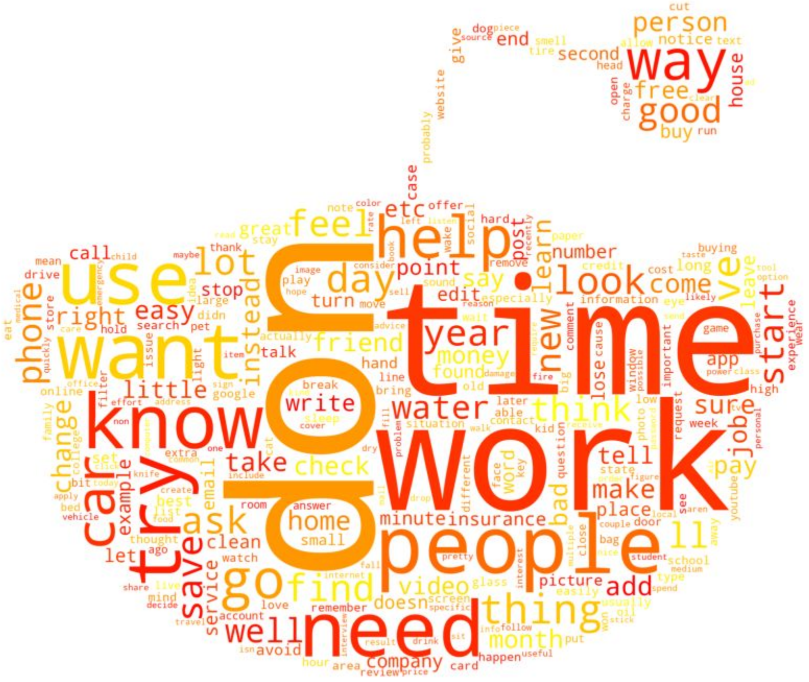
Lowercase, removal of numbers and punctuation

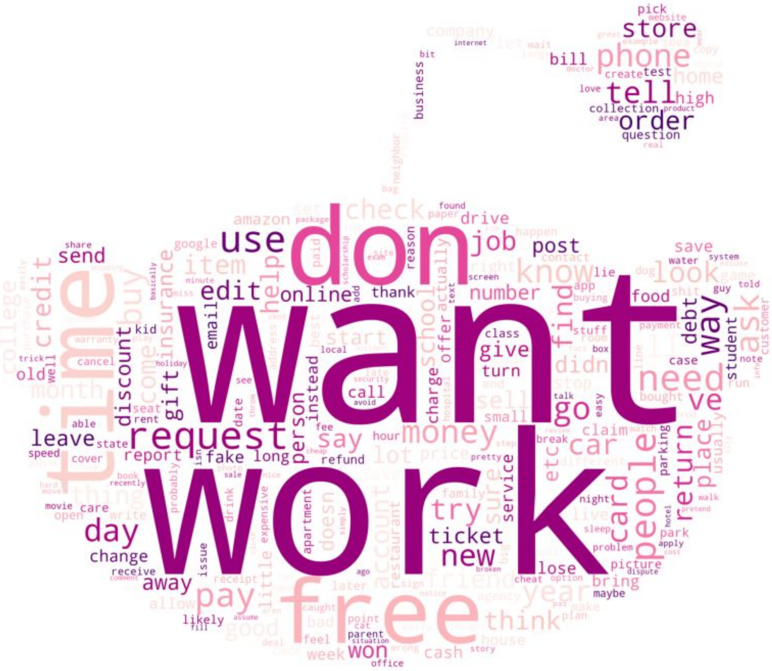
Stopwords Removal

Lemmatization

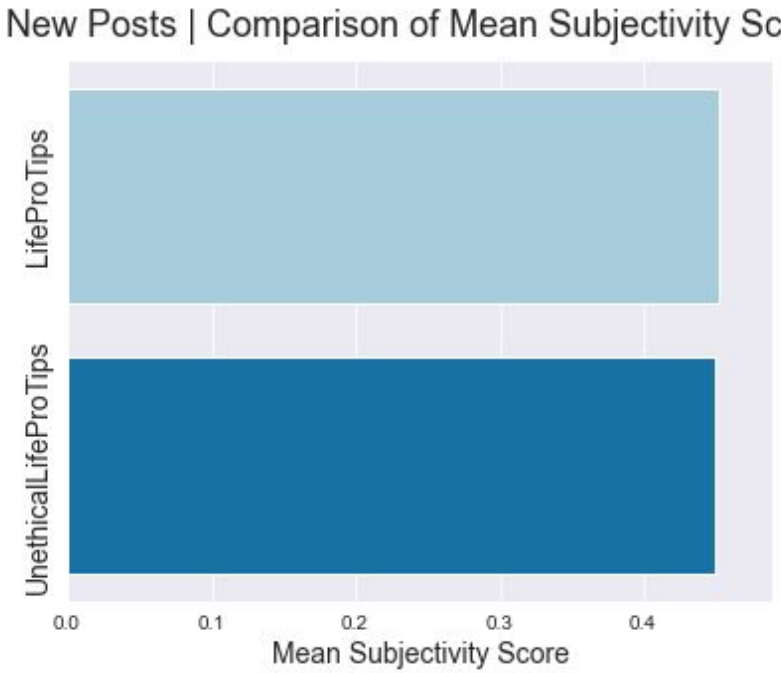
friend upset ask simple question say want talk want distract honestly best thing upset friend use time people respond people come need vent comfort follow want advice want listen time need let out...

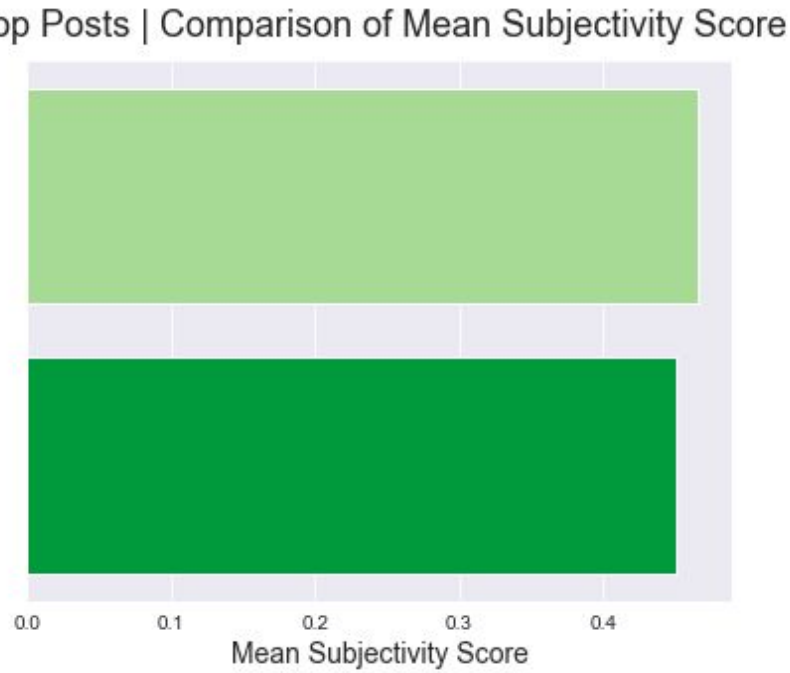
Wordcloud


don
time
work
want
use
know
people
need
try
car
help
go
find
thing
look
way
good
person

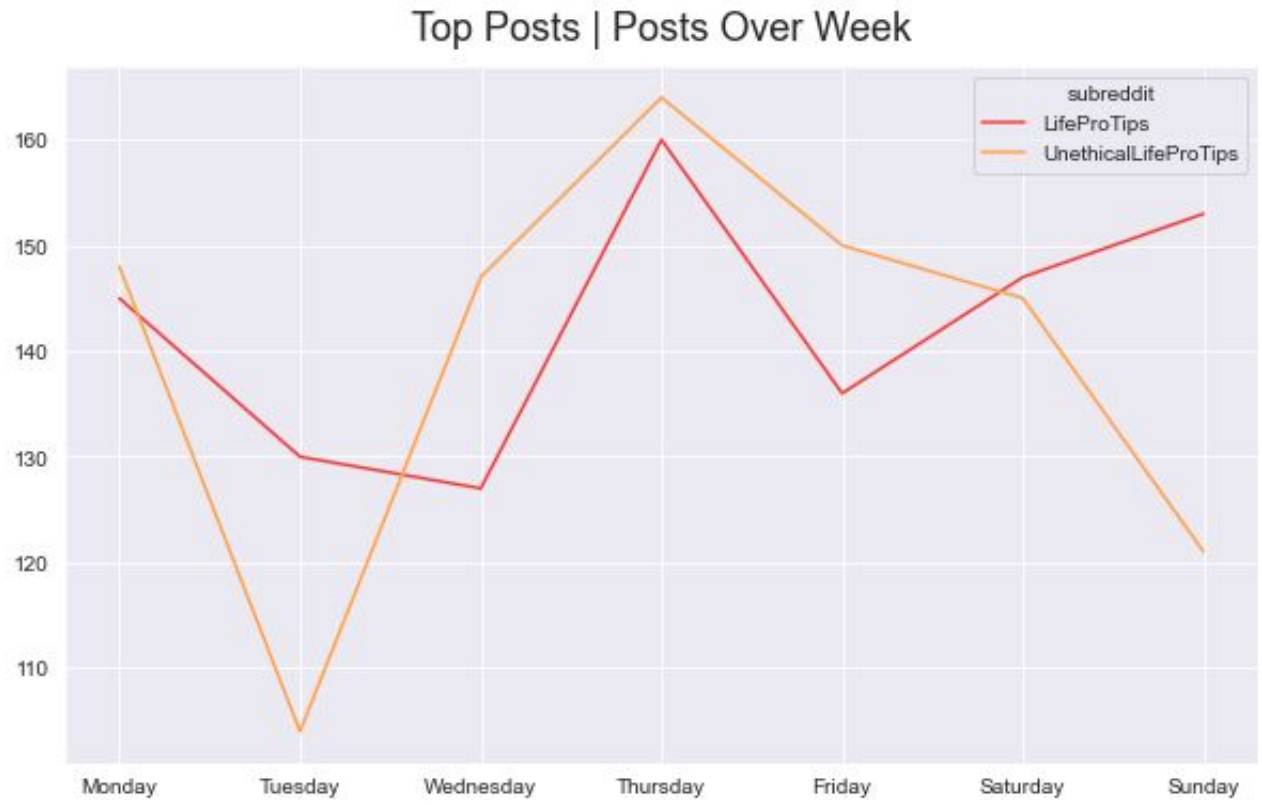

want
work
don
free
time
use
request
need
people
money
phone
tell
order
store

Polarity: positive versus negative sentiment





Posts across the week



Vectorizing and modeling

Word vectorization techniques

Count vectorization

TF-IDF vectorization

Modeling techniques

Multinomial NB

Logistic Regression

Support Vector Model

7 models were tested, including a voting classifier combining the 3 models.

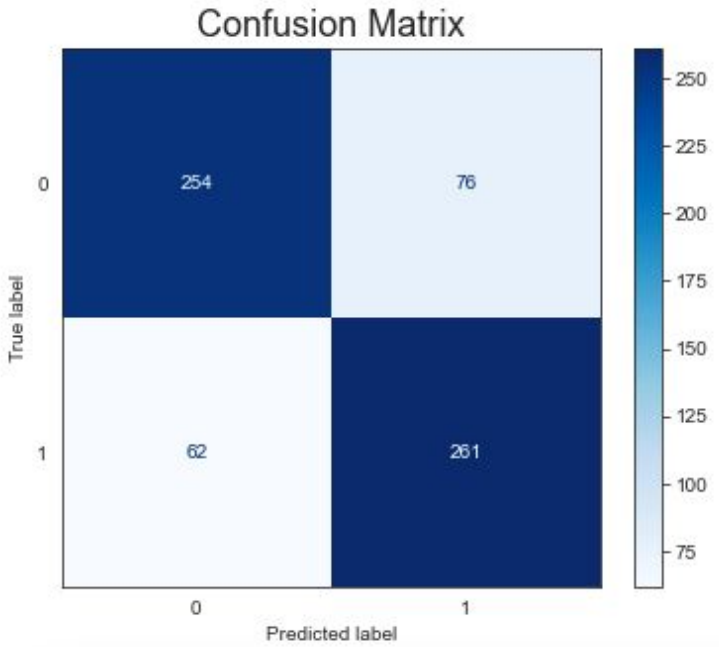
For every model - vectorizer combination:

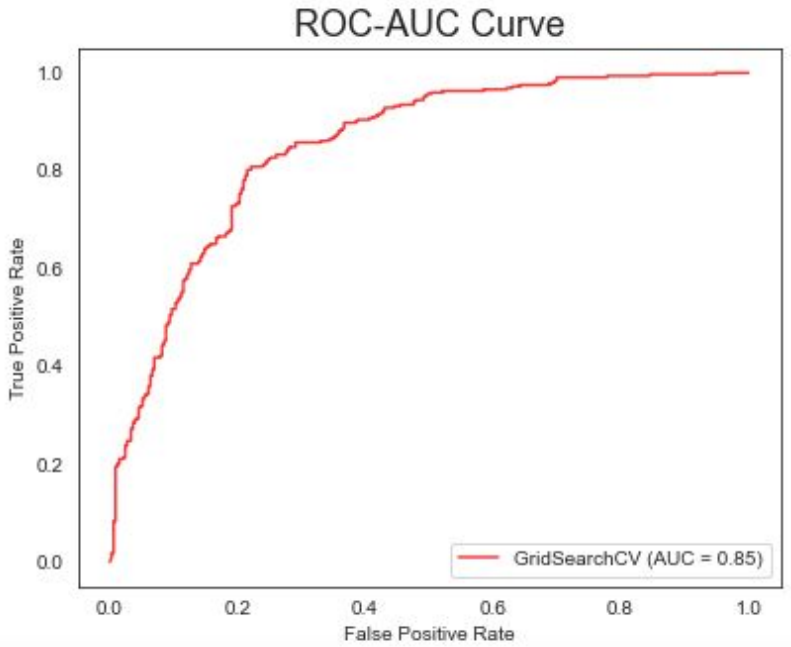
```
For cvec, nb:
Best score: 0.774
Train score: 0.9
Test score: 0.789
Best parameters: {'cvec__max_features': 3000, 'cvec__ngram_range': (1, 2), 'nb__alpha': 1.0, 'nb__fit_prior': True}

ROC-AUC Score: 0.849

True Negatives: 254
False Positives: 76
False Negatives: 62
True Positives: 261
Specificity (Positive Detection Rate): 0.808
Sensitivity (Negative Detection Rate): 0.77
```

For every model - vectorizer combination:





Initial testing results

	Model Name	GS Best Score	Train Score	Test Score	ROC-AUC Score	Specificity	Sensitivity
0	cvec_nb	0.774	0.900	0.789	0.849	0.808	0.770
1	tvec_nb	0.779	0.906	0.789	0.864	0.811	0.767
2	cvec_lr	0.758	0.987	0.784	0.867	0.799	0.770
3	tvec_lr	0.776	0.917	0.790	0.876	0.777	0.803
4	cvec_svm	0.743	0.927	0.766	0.856	0.811	0.727
5	tvec_svm	0.784	0.995	0.793	0.886	0.783	0.818
6	tvec_vote	0.777	0.954	0.804	0.878	0.786	0.788

Initial testing results

	Model Name	GS Best Score	Train Score	Test Score	ROC-AUC Score	Specificity	Sensitivity
0	cvec_nb	0.774	0.900	0.789	0.849	0.808	0.770
1	tvec_nb	0.779	0.906	0.789	0.864	0.811	0.767
2	cvec_lr	0.758	0.987	0.784	0.867	0.799	0.770
3	tvec_lr	0.776	0.917	0.790	0.876	0.777	0.803
4	cvec_svm	0.743	0.927	0.766	0.856	0.811	0.727
5	tvec_svm	0.784	0.995	0.793	0.886	0.783	0.818
6	tvec_vote	0.777	0.954	0.804	0.878	0.786	0.788

Highest accuracy score and high ROC-AUC score, but lower specificity

Initial testing results

	Model Name	GS Best Score	Train Score	Test Score	ROC-AUC Score	Specificity	Sensitivity
0	cvec_nb	0.774	0.900	0.789	0.849	0.808	0.770
1	tvec_nb	0.779	0.906	0.789	0.864	0.811	0.767
2	cvec_lr	0.758	0.987	0.784	0.867	0.799	0.770
3	tvec_lr	0.776	0.917	0.790	0.876	0.777	0.803
4	cvec_svm	0.743	0.927	0.766	0.856	0.811	0.727
5	tvec_svm	0.784	0.995	0.793	0.886	0.783	0.818
6	tvec_vote	0.777	0.954	0.804	0.878	0.786	0.788

Highest specificity but lower accuracy scores.

Selecting a model

	Model Name	GS Best Score	Train Score	Test Score	ROC-AUC Score	Specificity	Sensitivity
0	cvec_nb	0.774	0.900	0.789	0.849	0.808	0.770
1	tvec_nb	0.779	0.906	0.789	0.864	0.811	0.767
2	cvec_lr	0.758	0.987	0.784	0.867	0.799	0.770
3	tvec_lr	0.776	0.917	0.790	0.876	0.777	0.803
4	cvec_svm	0.743	0.927	0.766	0.856	0.811	0.727
5	tvec_svm	0.784	0.995	0.793	0.886	0.783	0.818
6	tvec_vote	0.777	0.954	0.804	0.878	0.786	0.788

Further tuning - adding stopwords

Predictive feature overlap

Consider words that feature heavily for both classes e.g. ‘work’, ‘people’

```
['want', 'work', 'use', 'don', 'need', 'like', 'get', 'way', 'time', 'peo
ple', 'll', 'ask', 'look', 'day', 'try', 'know']
```

Misclassified data word counts

Consider words that appear the most in our misclassified posts

Feature importance

Positive class - ULPT

-8.4200 action	-5.4161 want
-8.4200 add list	-5.4835 request
-8.4200 allergy	-5.5212 free
-8.4200 animation	-5.6137 work
-8.4200 applicant	-5.7430 use
-8.4200 apply job	-5.7728 don
-8.4200 appreciation	-5.8102 need
-8.4200 area rug	-5.8577 buy
-8.4200 art	-5.8709 like
-8.4200 assist	-5.8805 car
-8.4200 attitude	-5.9331 card
-8.4200 automatically	-5.9334 get
-8.4200 backwards	-5.9475 pay
-8.4200 bath	-5.9688 phone
-8.4200 bbt	-5.9718 way
-8.4200 behavior	-6.0008 time
-8.4200 beneficiary	-6.0052 people
-8.4200 bike	-6.0662 money
-8.4200 blade	-6.0686 return
-8.4200 braid	-6.0691 new
-8.4200 braid gray	-6.1274 ll
-8.4200 brain	-6.1342 ask
-8.4200 bread	-6.1811 look
-8.4200 breath	-6.2121 day
-8.4200 breathe	-6.2226 tell
-8.4200 breathing	-6.2239 gift
-8.4200 cam	-6.2628 check
-8.4200 capture	-6.2822 item
-8.4200 cellular	-6.2959 try
-8.4200 change mind	-6.2984 know

Negative class - LPT

-8.4995 admin	-5.4442 people
-8.4995 advance	-5.4794 don
-8.4995 airpods	-5.6798 time
-8.4995 anniversary	-5.7143 like
-8.4995 asshole	-5.7239 help
-8.4995 believable	-5.7942 work
-8.4995 borrow	-5.8272 ask
-8.4995 buy new	-5.8431 know
-8.4995 cancellation	-5.8456 thing
-8.4995 cancellation fee	-5.8459 day
-8.4995 cheat	-5.8808 edit
-8.4995 club	-5.9320 feel
-8.4995 controller	-5.9387 need
-8.4995 craigslist	-5.9894 get
-8.4995 despite	-6.0098 good
-8.4995 donor	-6.0457 way
-8.4995 felon	-6.0461 try
-8.4995 free trial	-6.0481 want
-8.4995 furniture store	-6.0681 go
-8.4995 haircut	-6.0796 person
-8.4995 hallway	-6.1079 ll
-8.4995 hassle	-6.1363 well
-8.4995 immigrant	-6.1462 ve
-8.4995 leftover	-6.1788 think
-8.4995 ll pay	-6.1867 look
-8.4995 parking	-6.2025 start
-8.4995 parking lot	-6.2256 use
-8.4995 pop	-6.2304 lot
-8.4995 radio	-6.2355 instead
-8.4995 reservation	-6.3108 say

Misclassified posts word count

Positive class - ULPT

	wordcount
want	51
don	48
people	47
ve	43
like	42
work	41
time	32
try	31
m	30
help	29
ask	26
tell	23
job	22
know	21
get	20

Negative class - LPT

	wordcount
car	99
don	59
engine	53
check	49
want	49
use	45
like	45
money	44
look	44
need	43
fuel	41
buy	40
edit	39
work	38
good	38

Final model

	Model Name	GS Best Score	Train Score	Test Score	ROC-AUC Score	Specificity	Sensitivity
0	cvec_nb	0.774	0.900000	0.789000	0.849	0.808	0.770
1	tvec_nb	0.779	0.906000	0.789000	0.864	0.811	0.767
2	cvec_lr	0.758	0.987000	0.784000	0.867	0.799	0.770
3	tvec_lr	0.776	0.917000	0.790000	0.876	0.777	0.803
4	cvec_svm	0.743	0.927000	0.766000	0.856	0.811	0.727
5	tvec_svm	0.784	0.995000	0.793000	0.886	0.783	0.818
6	tvec_vote	0.777	0.954000	0.804000	0.878	0.783	0.797
7	tvec_nb_new	NA	0.906344	0.790199	0.865	0.811	0.770

Feature importances

Positive class - ULPT

-8.4200 action	-5.4161 want
-8.4200 add list	-5.4835 request
-8.4200 allergy	-5.5212 free
-8.4200 animation	-5.6137 work
-8.4200 applicant	-5.7430 use
-8.4200 apply job	-5.7728 don
-8.4200 appreciation	-5.8102 need
-8.4200 area rug	-5.8577 buy
-8.4200 art	-5.8709 like
-8.4200 assist	-5.8805 car
-8.4200 attitude	-5.9331 card
-8.4200 automatically	-5.9334 get
-8.4200 backwards	-5.9475 pay
-8.4200 bath	-5.9688 phone
-8.4200 bbt	-5.9718 way
-8.4200 behavior	-6.0008 time
-8.4200 beneficiary	-6.0052 people
-8.4200 bike	-6.0662 money
-8.4200 blade	-6.0686 return
-8.4200 braid	-6.0691 new
-8.4200 braid gray	-6.1274 ll
-8.4200 brain	-6.1342 ask
-8.4200 bread	-6.1811 look
-8.4200 breath	-6.2121 day
-8.4200 breathe	-6.2226 tell
-8.4200 breathing	-6.2239 gift
-8.4200 cam	-6.2628 check
-8.4200 capture	-6.2822 item
-8.4200 cellular	-6.2959 try
-8.4200 change mind	-6.2984 know

Negative class - LPT

-8.4995 admin	-5.4442 people
-8.4995 advance	-5.4794 don
-8.4995 airpods	-5.6798 time
-8.4995 anniversary	-5.7143 like
-8.4995 asshole	-5.7239 help
-8.4995 believable	-5.7942 work
-8.4995 borrow	-5.8272 ask
-8.4995 buy new	-5.8431 know
-8.4995 cancellation	-5.8456 thing
-8.4995 cancellation fee	-5.8459 day
-8.4995 cheat	-5.8808 edit
-8.4995 club	-5.9320 feel
-8.4995 controller	-5.9387 need
-8.4995 craigslist	-5.9894 get
-8.4995 despite	-6.0098 good
-8.4995 donor	-6.0457 way
-8.4995 felon	-6.0461 try
-8.4995 free trial	-6.0481 want
-8.4995 furniture store	-6.0681 go
-8.4995 haircut	-6.0796 person
-8.4995 hallway	-6.1079 ll
-8.4995 hassle	-6.1363 well
-8.4995 immigrant	-6.1462 ve
-8.4995 leftover	-6.1788 think
-8.4995 ll pay	-6.1867 look
-8.4995 parking	-6.2025 start
-8.4995 parking lot	-6.2256 use
-8.4995 pop	-6.2304 lot
-8.4995 radio	-6.2355 instead
-8.4995 reservation	-6.3108 say

What features are most predictive?

Both classes

```
Overlap in Features
******************
want know
don buy
need help
go
web
tie
ask
listen
```

ULPT class

```
Features for ULPT
******************
reply
free trial
usps
busy
card
care
payment
personal information
resume
new account
moment
lock
```

LPT class

```
Features for LPT
******************
help
know
thin
dealership
edit
feel guilty
gotten
good way
perform
wife
vehicle
thing people
```

Conclusions

- **Definitive overlap** between our two subreddits: especially things to do with work or people.
- There seems to be a greater focus on **purchases and 'gaming' the system**, in our ULPT posts.
- We don’t get a strong sense of anti-social behaviour or a clear “unethical” definition - maybe a different subreddit will be better to determine this.
- We have to note that our data may be:
  - Biased to this context and group of people
  - Unreliable as it is user-generated and moderated

Potential extensions

- **Further refine NLP** by optimizing custom set of stopwords, part-of-speech analysis

- **Expand our dataset** with comments, sample posts across a period of time etc.

- **Investigate other trends** for Reddit posts: Score, upvote ratio, whether it reaches r/all

Thank you!