r/WSB vs r/stocks Subreddit Classification

Group 3

Cathy, Gilbert, Jun Pin

Outline

- **Intro**
  - Project **background**
  - **Subreddit** info
  - Project **objectives**
- **EDA**
- **Modeling**
- **Sentiment analysis**

Project Background

- Client: **Robinhood Markets Inc.** Robinhood
    - Zero-commision **online trading platform** for stocks, ETFs, options, and crypto
    - No account mins, no maintenance fees, **gamified** trading experience
    - User base: **inexperienced** new investors who trades frequently

- Problem: **Accused** of encouraging active trading behavior and **fined** by financial regulatory institute for not equipping its customers with sufficient knowledge

Project Background

- Response from management:
    - Provide **more educational resources** on the platform
    - Work on the **user base**: attract more experienced, **long-term investors**

- Approach: **Targeted web advertising** to users more inclined to passive investing.

- Our Role: **use NLP** to **identify posts** from two investment-related **subreddits**:
    - **r/WallStreetBets** & **r/Stocks**

Subreddits

- r/WallStreetBets
  - Stocks and option trading
  - **Aggressive** trading strategies
  - **Memes**
  - Ideas for **extremely risky stock / option plays**

- r/Stocks
  - More **serious** discussion on stocks, option trading / investing
  - **Analysis and discussions** on various stocks and companies
  - **Stringent content policy** over discussions on ‘Penny Stocks’ (i.e.: stocks with low market capitalization and volume)
  - More geared towards **serious long-term investment**

Project Objectives

- **Primary: Targeting advertisement**
  - Use **NLP** to **classify** an unseen post, for ads targeting posts on r/stocks

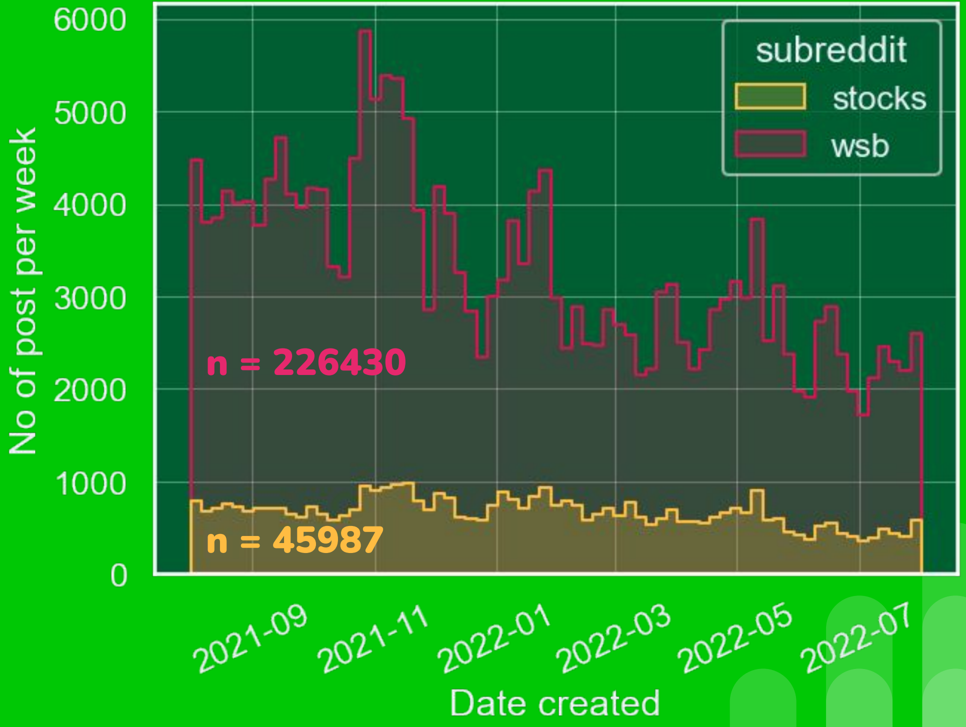
- **Secondary: Inform investment decision *(exploratory)***
  - Analyze **correlation** between the **sentiment** of a particular stock to the **future performance** of that stock.

EDA

Distribution of Post

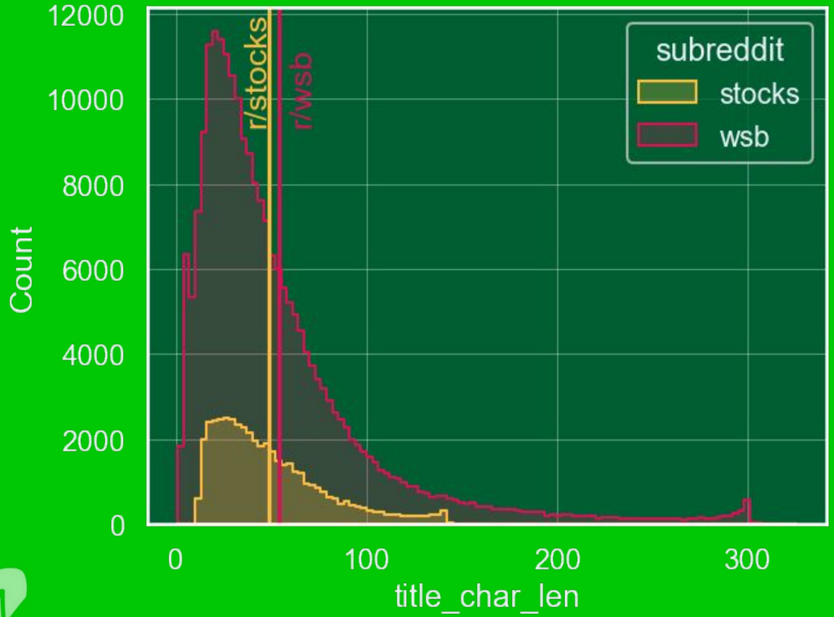
r/wsb has **5x more posts** than r/stocks

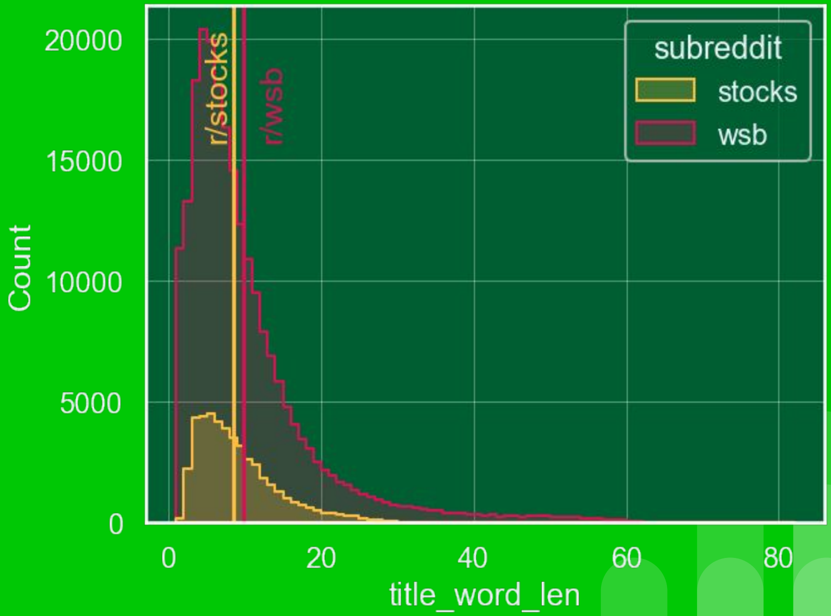
(highly imbalanced dataset)



Length of Post

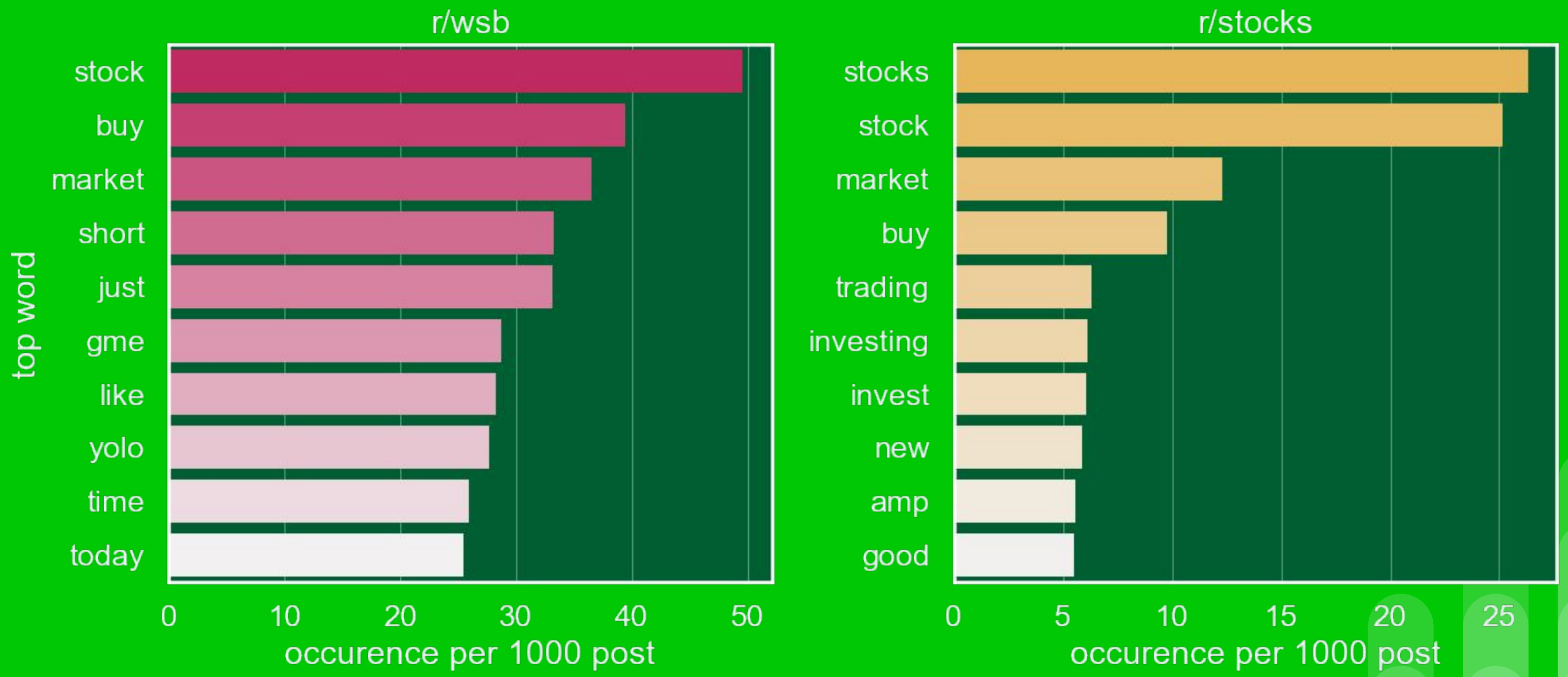
Wsb is very slightly longer





Text-Based Eda

Top 10 words in each subreddit

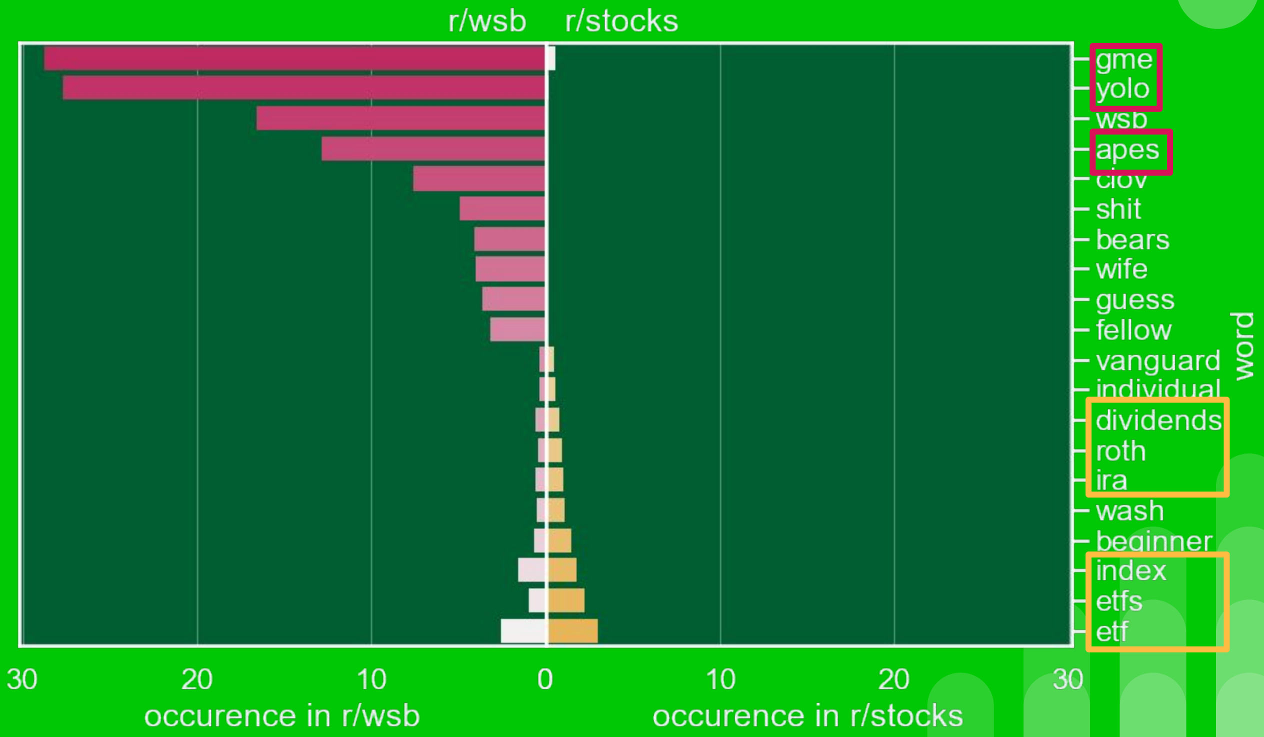


Text-Based Eda

While **r/wsb** is **YOLO**-ing their money in **$GME** with their fellow **apes**...

r/stocks is getting **dividends** from their **ETFs**, **index** funds, and **Roth IRA**

(individual retirement account)



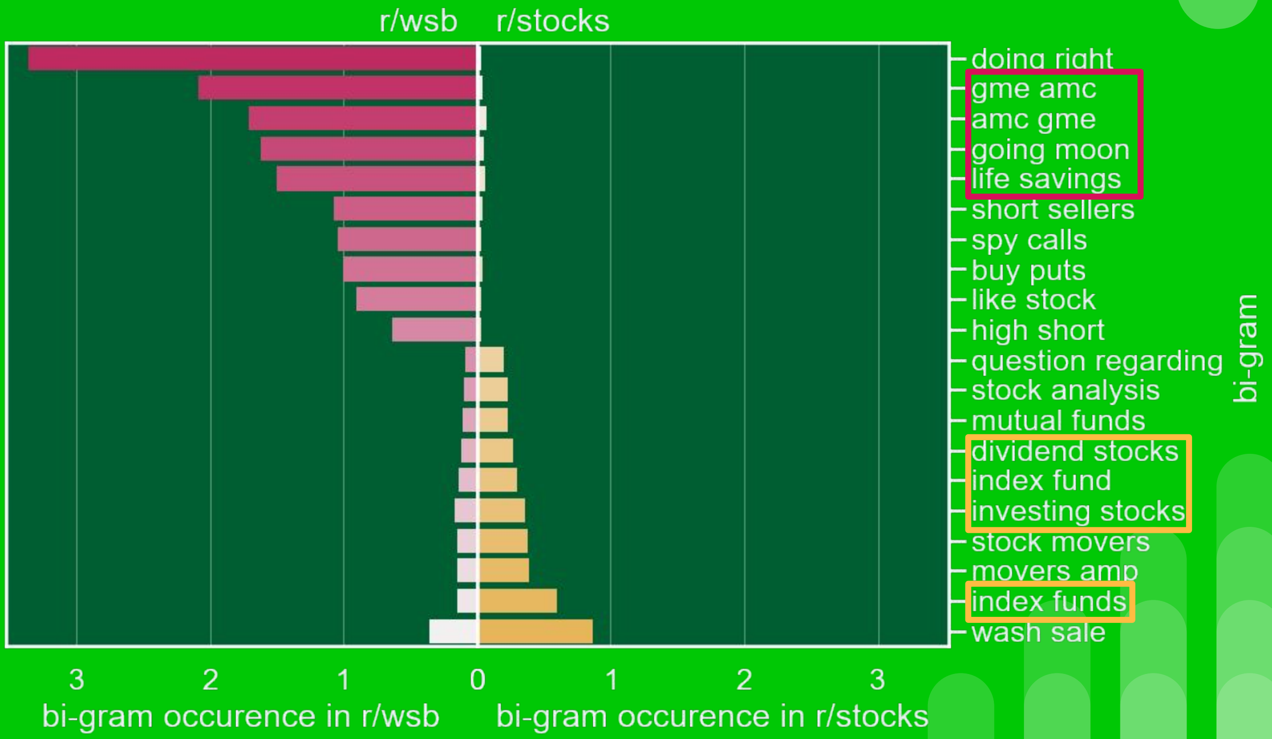
Text-Based Eda

While **r/wsb** is putting their **life savings** into **$GME** and **$AMC** and hoping that it is **going** (to the) **moon**

r/stocks is doing **stock analysis** on the **index funds** and **dividend stocks**

There is a difference!

We will train a model to classify between the two subreddits



Modelling

Success Evaluation

True Positive
correctly classifying and serving the advertisement to the target class (r/stocks)

False Positive
incorrectly classifying the target class (r/stocks), and instead serving the advertisement to the wrong subreddit (r/wsb)

False Negative
incorrectly classifying the other class (r/wsb) which resulted in not serving the advertisement to the target class (r/stocks)

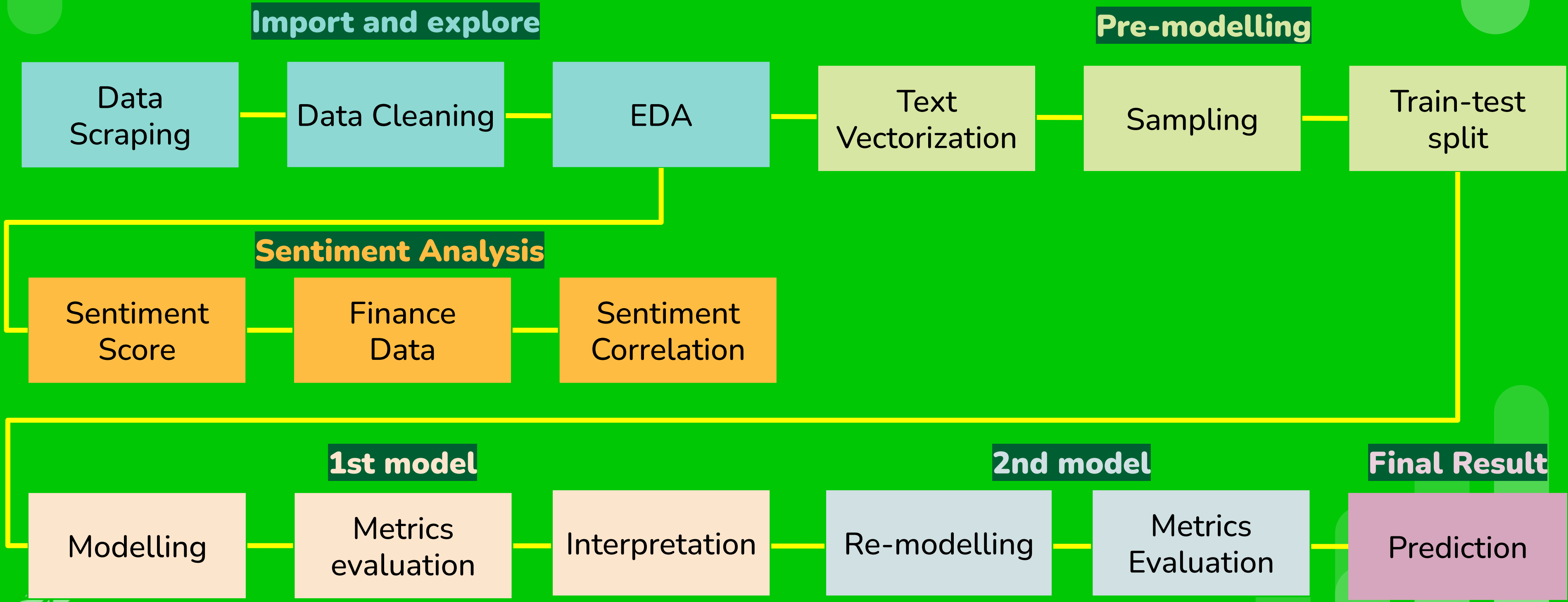
True Negative
correctly clasifying the other class (r/wsb) and not serving the advertisement

Precision
ratio of advertisement served to the correct class

Recall
ratio of posts in the correct class that is correctly served the advertisement

F1-Score
Taking both Precisions and Recall into consideration

Process Flowchart



Import and explore

Data Scraping

Data Cleaning

EDA

Pre-modelling

Text Vectorization

Sampling

Train-test split

Sentiment Analysis

Sentiment Score

Finance Data

Sentiment Correlation

1st model

Modelling

Metrics evaluation

Interpretation

2nd model

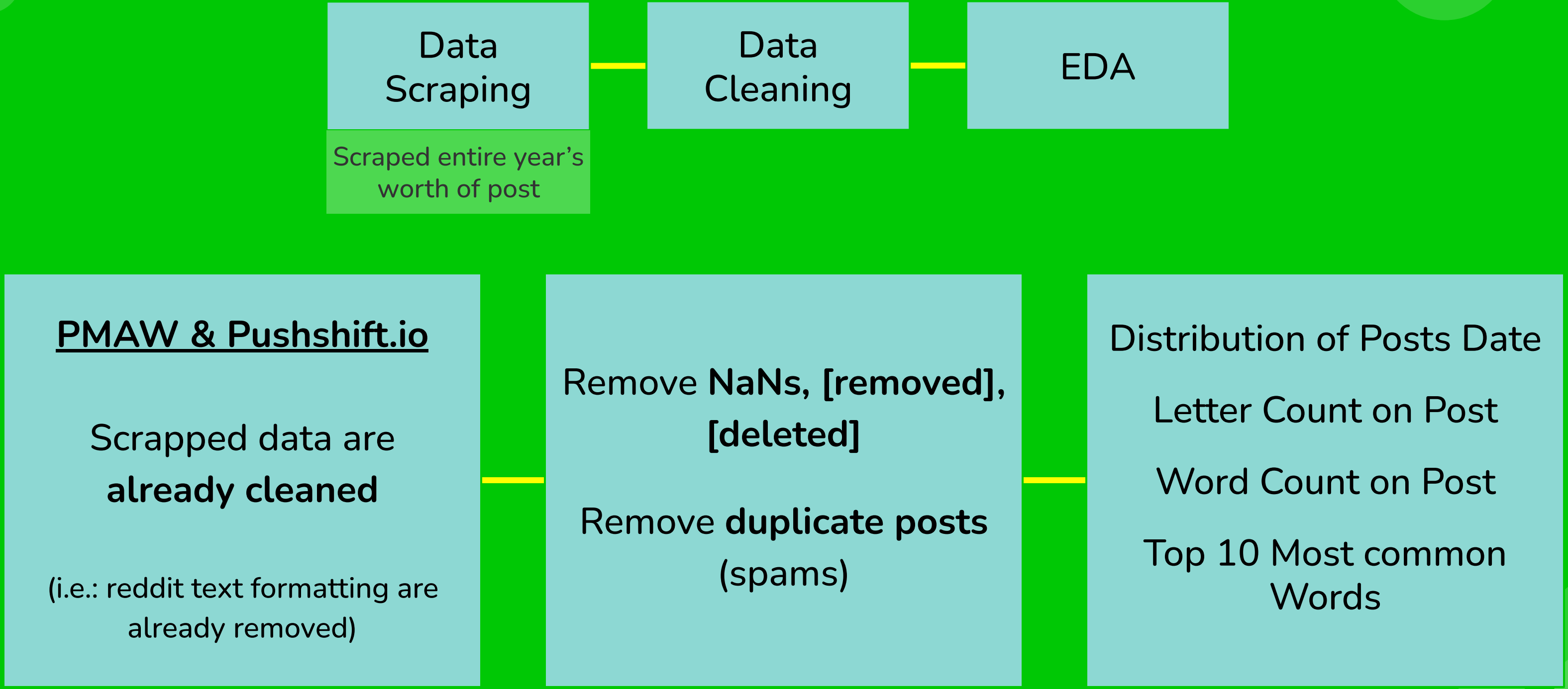
Re-modelling

Metrics Evaluation

Final Result

Prediction

Import & Explore



Data Scraping

Scraped entire year's worth of post

Data Cleaning

EDA

PMAW & Pushshift.io

Scrapped data are **already cleaned**

(i.e.: reddit text formatting are already removed)

Remove **NaNs, [removed], [deleted]**

Remove **duplicate posts** (spams)

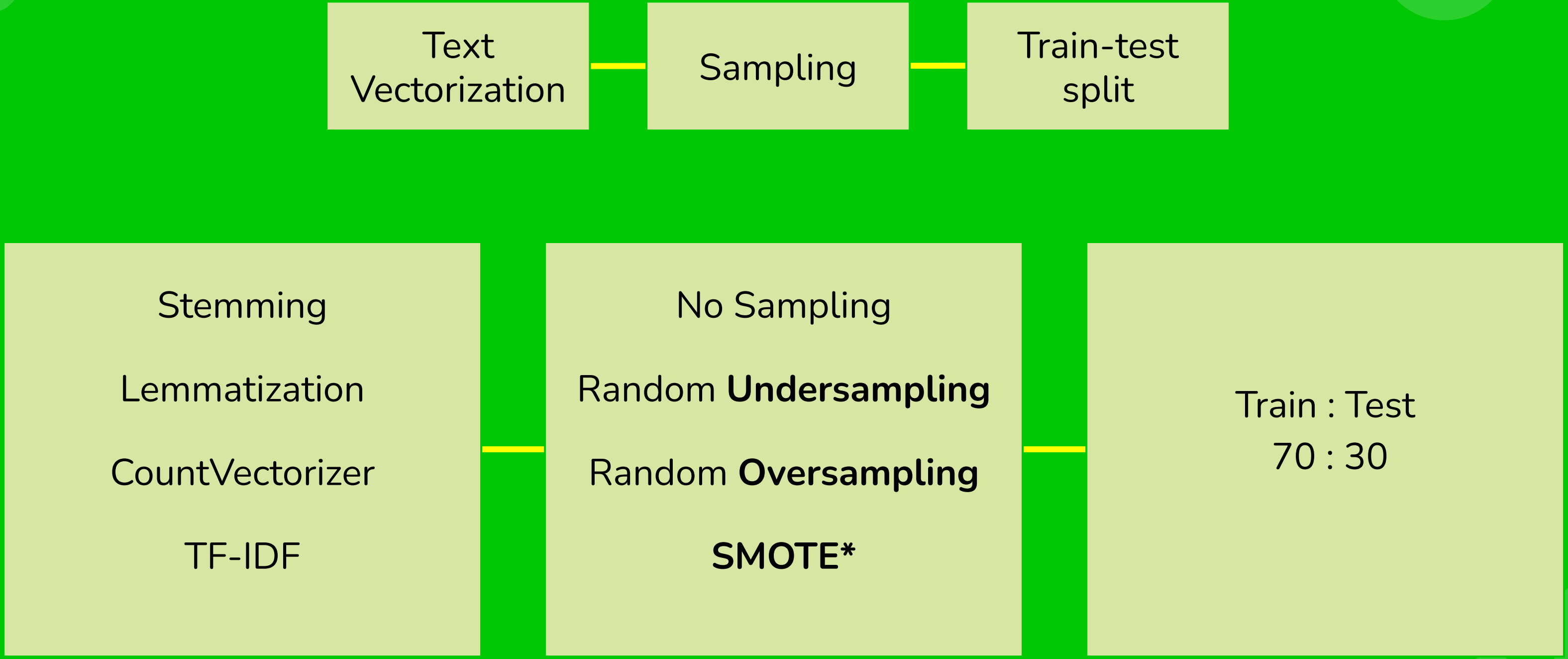
Distribution of Posts Date

Letter Count on Post

Word Count on Post

Top 10 Most common Words

Pre-Modeling



Text Vectorization — Sampling — Train-test split

Stemming

Lemmatization

CountVectorizer

TF-IDF

No Sampling

Random **Undersampling**

Random **Oversampling**

SMOTE*

Train : Test

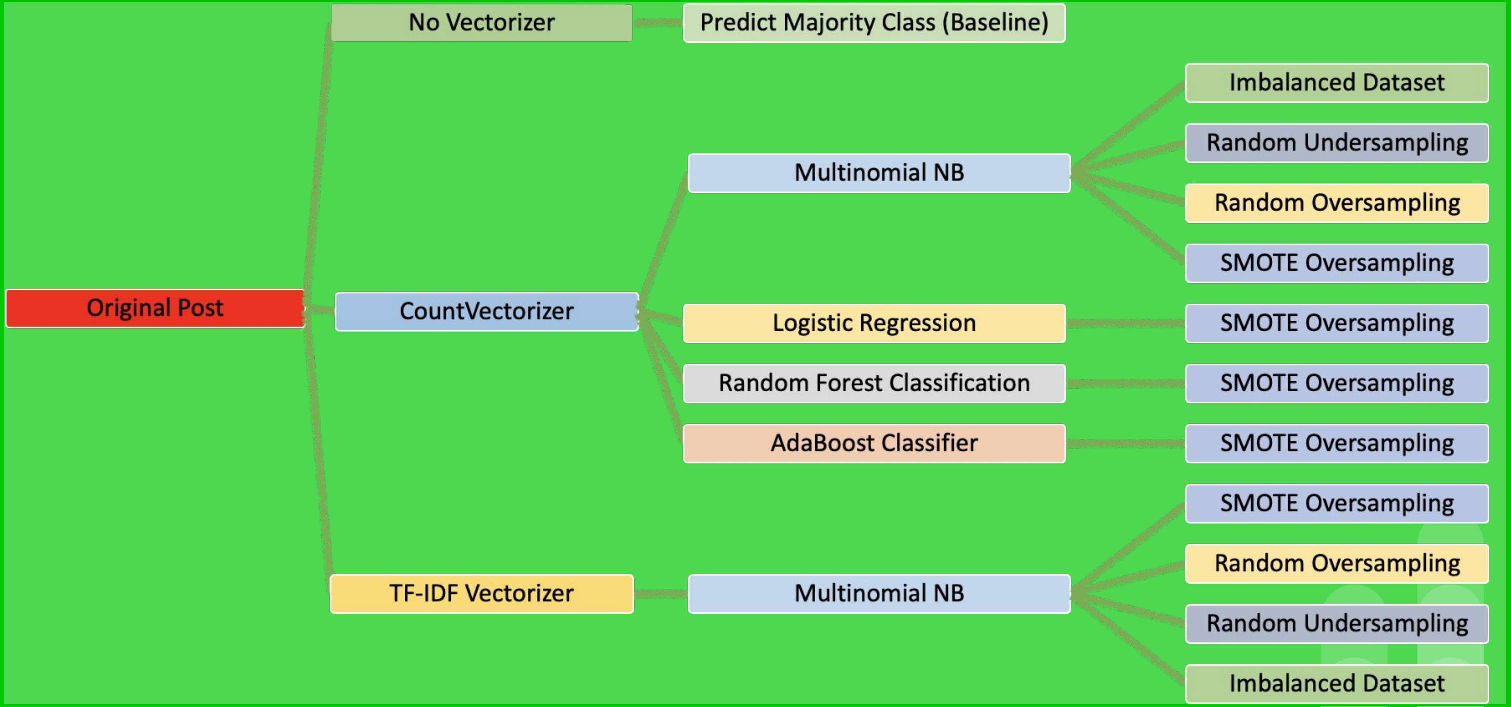
70 : 30

*Synthetic Minority Over-sampling Technique
synthetic samples are generated for the minority class.

1st Model

Modelling

Metrics evaluation



Original Post

No Vectorizer

Predict Majority Class (Baseline)

CountVectorizer

Multinomial NB

Imbalanced Dataset

Random Undersampling

Random Oversampling

SMOTE Oversampling

Logistic Regression

SMOTE Oversampling

Random Forest Classification

SMOTE Oversampling

AdaBoost Classifier

SMOTE Oversampling

TF-IDF Vectorizer

Multinomial NB

SMOTE Oversampling

Random Oversampling

Random Undersampling

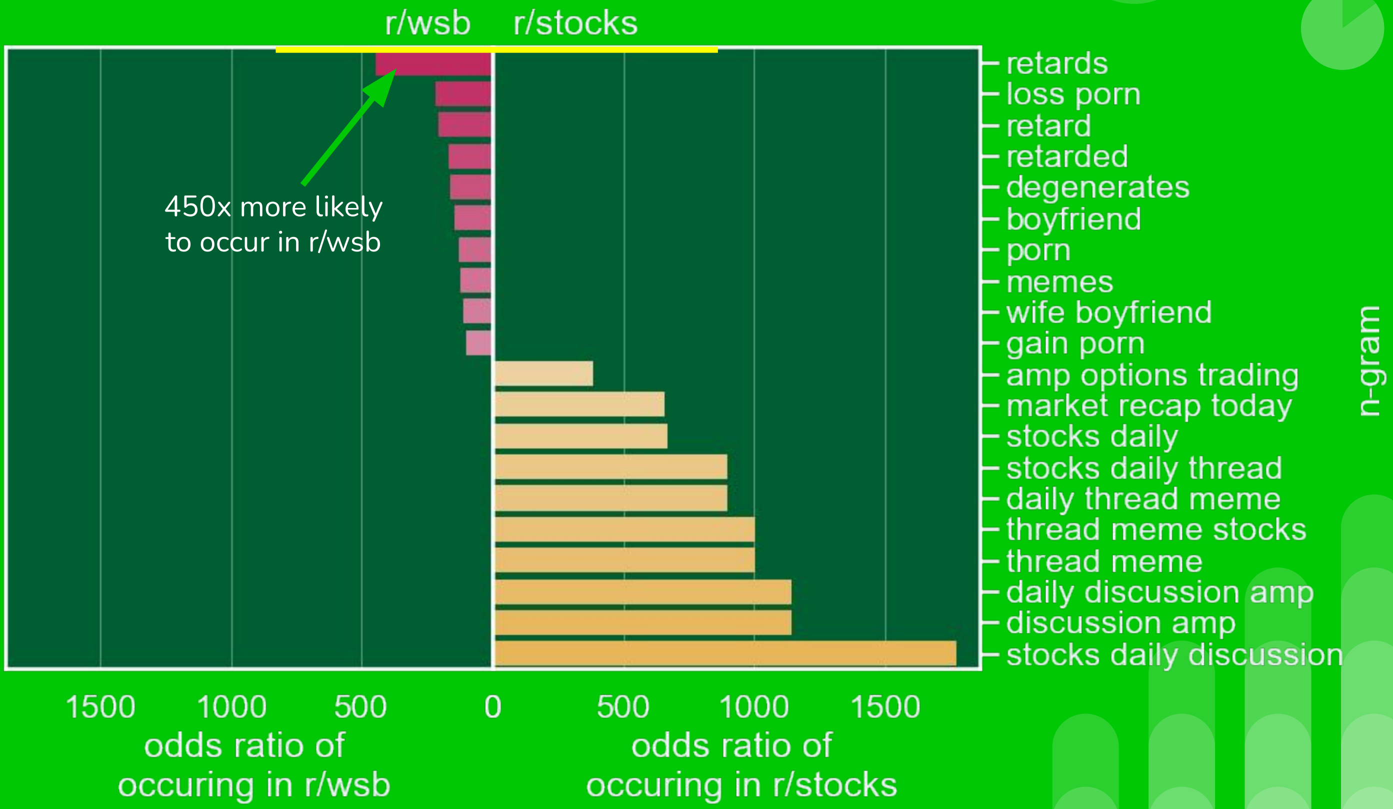
Imbalanced Dataset

Model Performance (Test Scores)

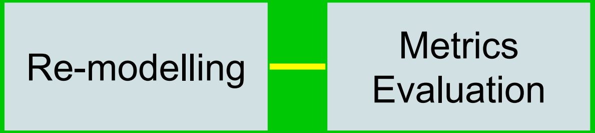
id	vectorizer	model	comments	accuracy	precision	recall	c1_f1	c0_f1	avg_f1	wt_avg_f1
0a	N/A	Predict Majority	Baseline Model	0.831	0.000	0.000	0.000	0.908	0.454	0.755
0b	N/A	Predict Target	Baseline Model	0.169	0.169	1.000	0.289	0.000	0.144	0.049
1a	CVEC	Multinomial NB	Imbalanced dataset	0.853	0.747	0.196	0.311	0.918	0.614	0.815
1b	CVEC	Multinomial NB	Random Undersampling	0.755	0.386	0.773	0.515	0.836	0.675	0.782
1c	CVEC	Multinomial NB	Random Oversampling	0.793	0.432	0.725	0.542	0.866	0.704	0.811
1d	CVEC	Multinomial NB	SMOTE Oversampling	0.844	0.541	0.506	0.523	0.907	0.715	0.842
2a	TF-IDF	Multinomial NB	Imbalanced dataset	0.84	0.857	0.061	0.114	0.912	0.513	0.777
2b	TF-IDF	Multinomial NB	Random Undersampling	0.735	0.369	0.796	0.504	0.820	0.662	0.766
2c	TF-IDF	Multinomial NB	Random Oversampling	0.795	0.434	0.698	0.535	0.869	0.702	0.813
2d	TF-IDF	Multinomial NB	SMOTE Oversampling	0.829	0.496	0.622	0.552	0.895	0.723	0.837
3	CVEC	Log-Reg	SMOTE Oversampling	0.788	0.396	0.481	0.434	0.870	0.652	0.796
4	CVEC	RFC	SMOTE Oversampling	0.68	0.291	0.624	0.397	0.782	0.589	0.717
5	CVEC	AdaBoost	SMOTE Oversampling	0.716	0.326	0.64	0.432	0.81	0.621	0.747

Interpretation

Comparing the **odds ratio** of each **feature** being present in r/wsb or r/stocks



2nd Model (Test Scores)


Re-modelling

Metrics Evaluation

comments	accuracy	c1_precision	c1_recall	c1_f1	c0_f1	avg_f1	wt_avg_f1
Final model (Multi-NB, CVEC, SMOTE)	0.844	0.540	0.504	0.522	0.907	0.714	0.842
only consider posts with **3 words or more**	0.846	0.577	0.489	0.529	0.908	0.718	0.840
only consider posts with **10 words or more**	0.865	0.681	0.291	0.408	0.924	0.666	0.841
converting **emoji** into text	0.845	0.542	0.514	0.528	0.907	0.718	0.843
combining **title and selftext**	0.832	0.502	0.494	0.498	0.899	0.699	0.831
change **n-gram** range to (1,1)	0.797	0.432	0.640	0.516	0.872	0.694	0.812
change **n-gram** range to (1,2)	0.831	0.500	0.565	0.531	0.897	0.714	0.835
change **max-features** to 118167 (10% of default)	0.816	0.465	0.623	0.533	0.885	0.709	0.826
change **max-features** to 59083 (5% of default)	0.814	0.462	0.622	0.530	0.884	0.707	0.824
change **max-features** to 11817 (1% of default)	0.811	0.455	0.601	0.518	0.882	0.700	0.821

Data Drift

Train-test split on **one year**’s worth of data

Train on **one month** of data

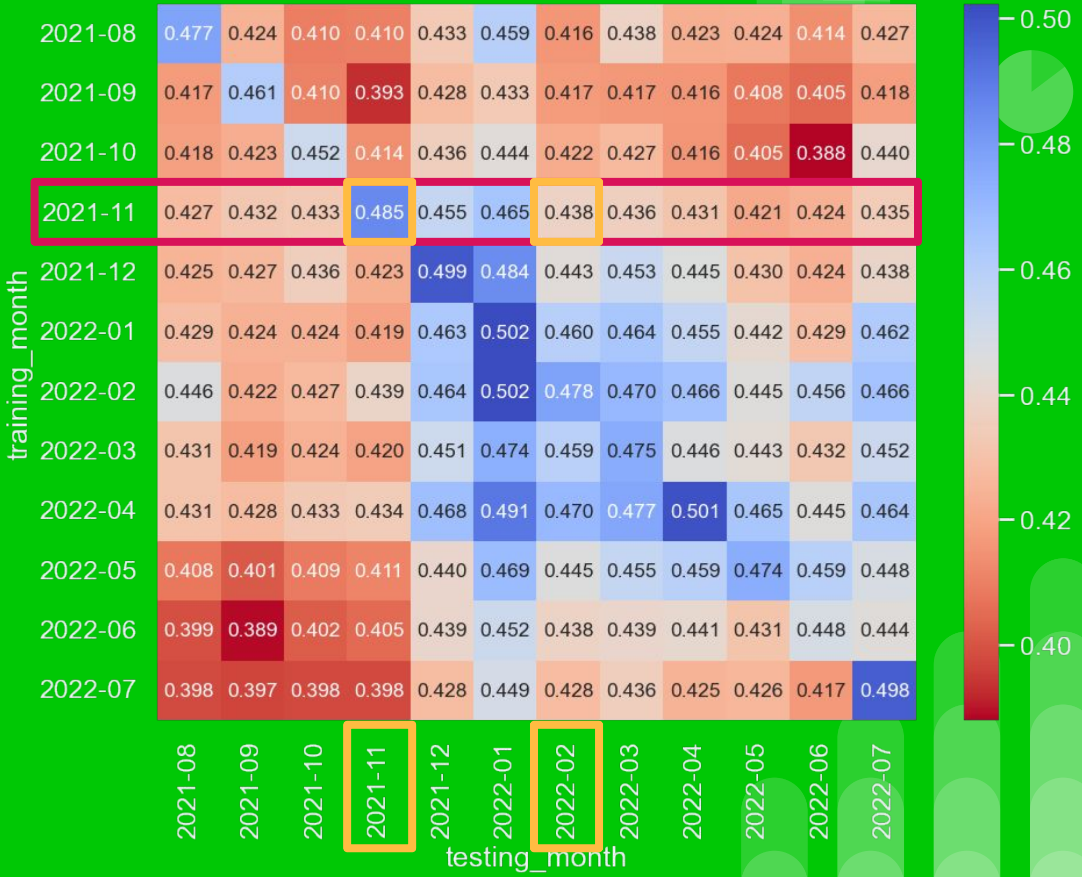
Predict other (**next**) month’s data

Data Drift

Each row is a **model**

Each cell shows the **performance** of each model *(row)* based on the testing month *(column)*

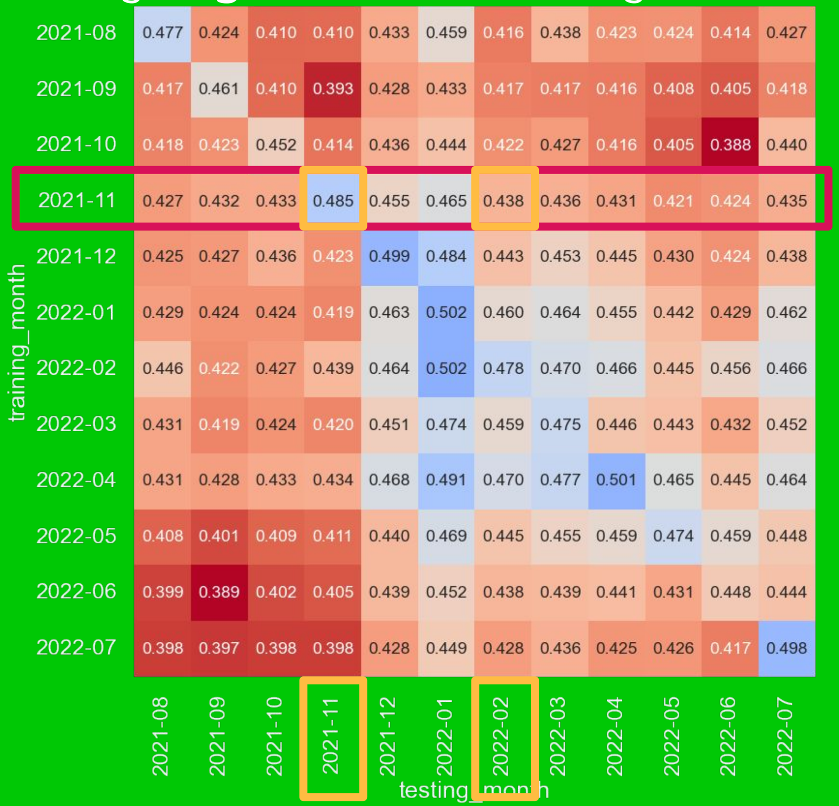
The model performs **worse** when predicting **other months**



	2021-08	2021-09	2021-10	2021-11	2021-12	2022-01	2022-02	2022-03	2022-04	2022-05	2022-06	2022-07
2021-08	0.477	0.424	0.410	0.410	0.433	0.459	0.416	0.438	0.423	0.424	0.414	0.427
2021-09	0.417	0.461	0.410	0.393	0.428	0.433	0.417	0.417	0.416	0.408	0.405	0.418
2021-10	0.418	0.423	0.452	0.414	0.436	0.444	0.422	0.427	0.416	0.405	0.388	0.440
2021-11	0.427	0.432	0.433	0.485	0.455	0.465	0.438	0.436	0.431	0.421	0.424	0.435
2021-12	0.425	0.427	0.436	0.423	0.499	0.484	0.443	0.453	0.445	0.430	0.424	0.438
2022-01	0.429	0.424	0.424	0.419	0.463	0.502	0.460	0.464	0.455	0.442	0.429	0.462
2022-02	0.446	0.422	0.427	0.439	0.464	0.502	0.478	0.470	0.466	0.445	0.456	0.466
2022-03	0.431	0.419	0.424	0.420	0.451	0.474	0.459	0.475	0.446	0.443	0.432	0.452
2022-04	0.431	0.428	0.433	0.434	0.468	0.491	0.470	0.477	0.501	0.465	0.445	0.464
2022-05	0.408	0.401	0.409	0.411	0.440	0.469	0.445	0.455	0.459	0.474	0.459	0.448
2022-06	0.399	0.389	0.402	0.405	0.439	0.452	0.438	0.439	0.441	0.431	0.448	0.444
2022-07	0.398	0.397	0.398	0.398	0.428	0.449	0.428	0.436	0.425	0.426	0.417	0.498

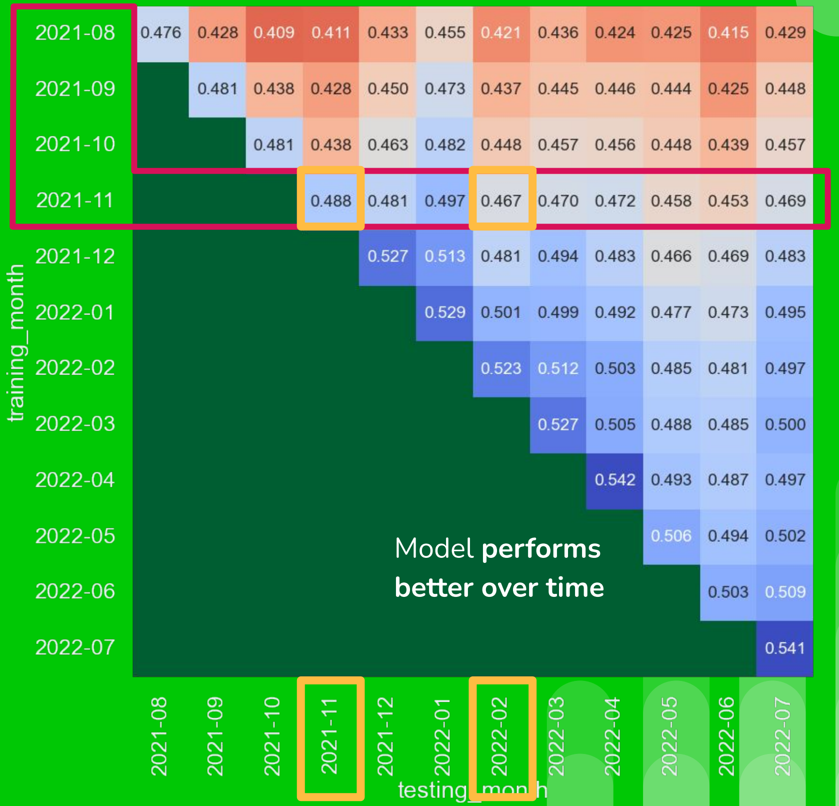
Data Drift

Using **single month** for training



	2021-08	2021-09	2021-10	2021-11	2021-12	2022-01	2022-02	2022-03	2022-04	2022-05	2022-06	2022-07
2021-08	0.477	0.424	0.410	0.410	0.433	0.459	0.416	0.438	0.423	0.424	0.414	0.427
2021-09	0.417	0.461	0.410	0.393	0.428	0.433	0.417	0.417	0.416	0.408	0.405	0.418
2021-10	0.418	0.423	0.452	0.414	0.436	0.444	0.422	0.427	0.416	0.405	0.388	0.440
2021-11	0.427	0.432	0.433	0.485	0.455	0.465	0.438	0.436	0.431	0.421	0.424	0.435
2021-12	0.425	0.427	0.436	0.423	0.499	0.484	0.443	0.453	0.445	0.430	0.424	0.438
2022-01	0.429	0.424	0.424	0.419	0.463	0.502	0.460	0.464	0.455	0.442	0.429	0.462
2022-02	0.446	0.422	0.427	0.439	0.464	0.502	0.478	0.470	0.466	0.445	0.456	0.466
2022-03	0.431	0.419	0.424	0.420	0.451	0.474	0.459	0.475	0.446	0.443	0.432	0.452
2022-04	0.431	0.428	0.433	0.434	0.468	0.491	0.470	0.477	0.501	0.465	0.445	0.464
2022-05	0.408	0.401	0.409	0.411	0.440	0.469	0.445	0.455	0.459	0.474	0.459	0.448
2022-06	0.399	0.389	0.402	0.405	0.439	0.452	0.438	0.439	0.441	0.431	0.448	0.444
2022-07	0.398	0.397	0.398	0.398	0.428	0.449	0.428	0.436	0.425	0.426	0.417	0.498

Using **cumulative** data

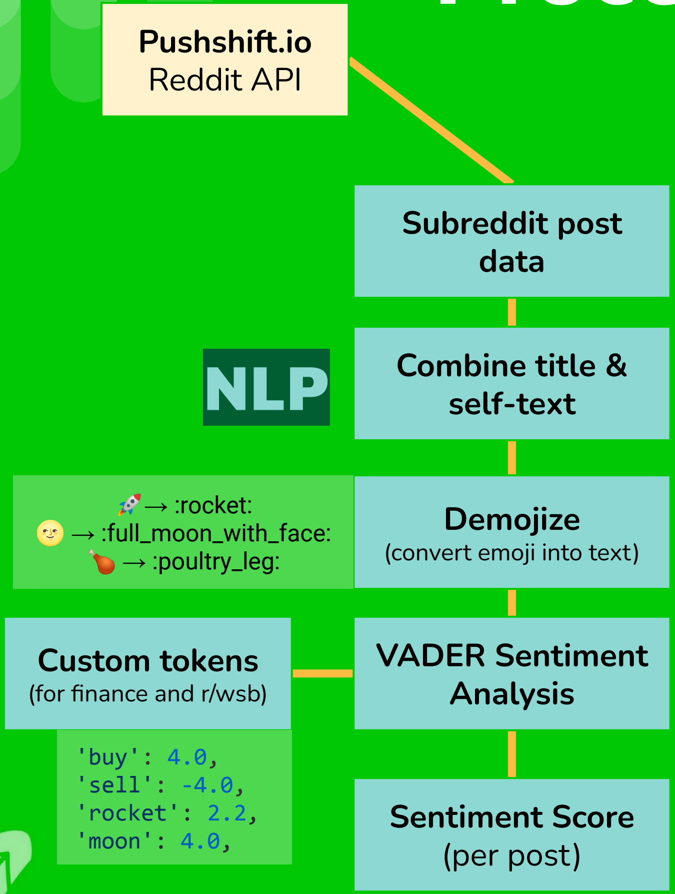


	2021-08	2021-09	2021-10	2021-11	2021-12	2022-01	2022-02	2022-03	2022-04	2022-05	2022-06	2022-07
2021-08	0.476	0.428	0.409	0.411	0.433	0.455	0.421	0.436	0.424	0.425	0.415	0.429
2021-09		0.481	0.438	0.428	0.450	0.473	0.437	0.445	0.446	0.444	0.425	0.448
2021-10			0.481	0.438	0.463	0.482	0.448	0.457	0.456	0.448	0.439	0.457
2021-11				0.488	0.481	0.497	0.467	0.470	0.472	0.458	0.453	0.469
2021-12					0.527	0.513	0.481	0.494	0.483	0.466	0.469	0.483
2022-01						0.529	0.501	0.499	0.492	0.477	0.473	0.495
2022-02							0.523	0.512	0.503	0.485	0.481	0.497
2022-03								0.527	0.505	0.488	0.485	0.500
2022-04									0.542	0.493	0.487	0.497
2022-05										0.506	0.494	0.502
2022-06											0.503	0.509
2022-07												0.541

Model **performs better over time**

Sentiment & Stocks Analysis

Process Flowchart



Pushshift.io
Reddit API

Subreddit post data

NLP

Combine title & self-text

🚀 → :rocket:
🌝 → :full_moon_with_face:
🍗 → :poultry_leg:

Demojize
(convert emoji into text)

Custom tokens
(for finance and r/wsb)

```
'buy': 4.0,
'sell': -4.0,
'rocket': 2.2,
'moon': 4.0,
```

VADER Sentiment Analysis

Sentiment Score
(per post)

Sentiment Histogram

Right skew

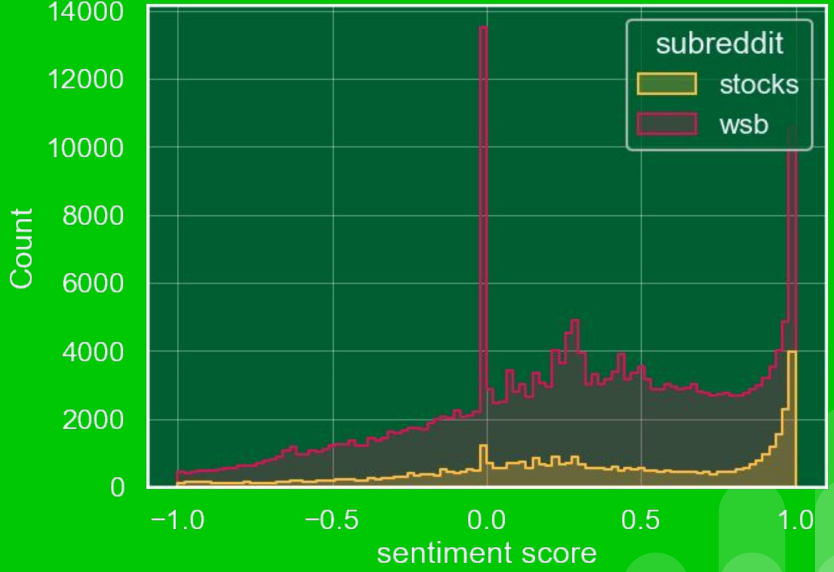
(more posts w/ positive sentiment)

Peaks at 0.0 and +1.0

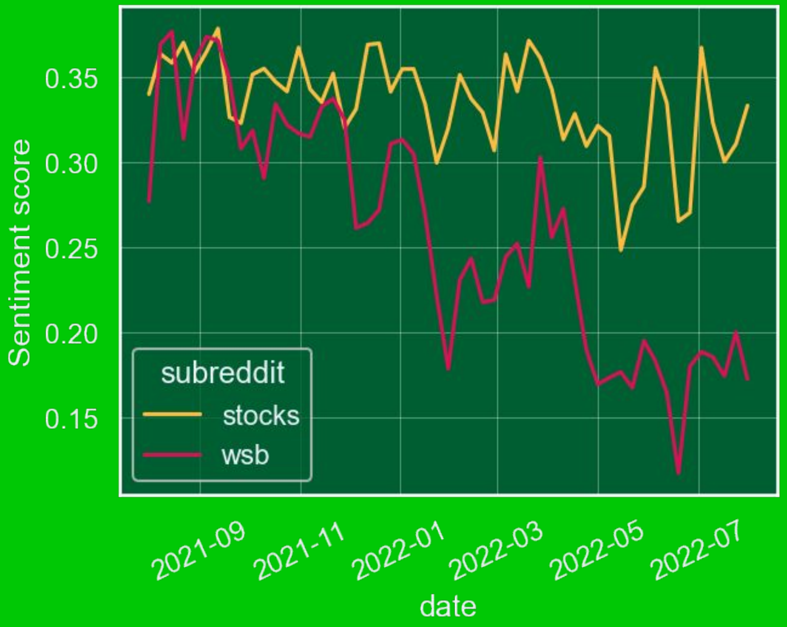
(high number of neutral and +ve posts)

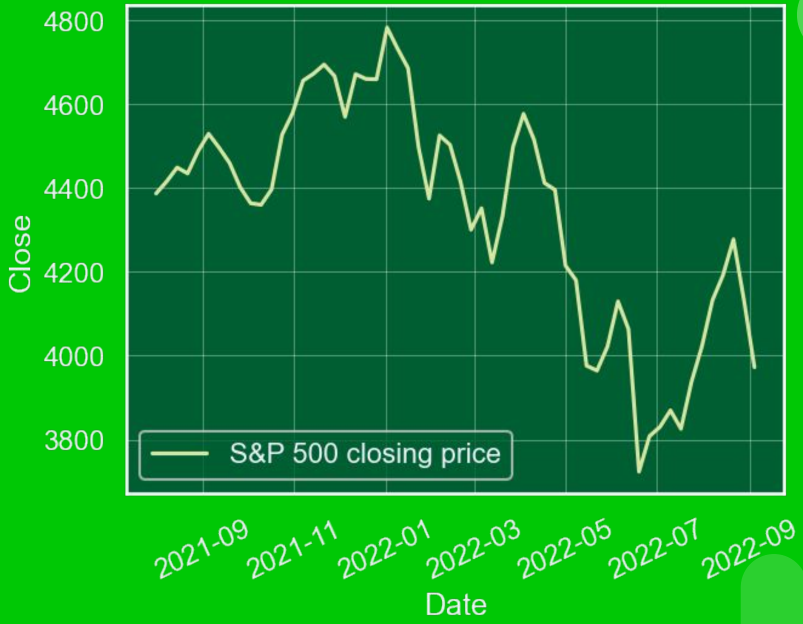
Similar distribution

between r/stocks and r/wsb



Sentiment Trend





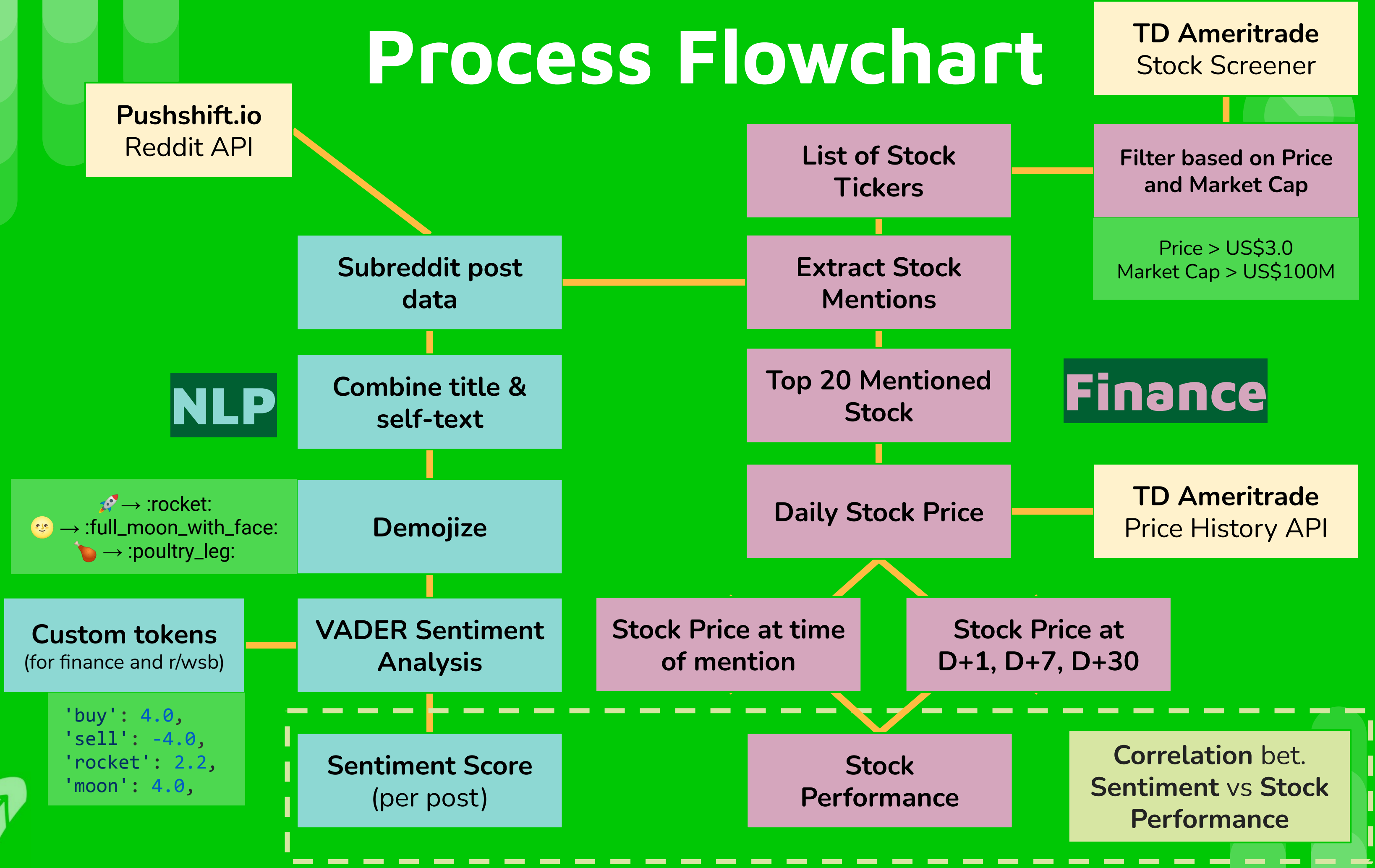
r/wsb = downward trend

r/stocks = slightly downwards trend

Similar to S&P500 price trend

(market sentiment is reflected in the subreddits)

Process Flowchart



Pushshift.io
Reddit API

Subreddit post data

Combine title & self-text

NLP

Demojize

🚀 → :rocket:
🌝 → :full_moon_with_face:
🍗 → :poultry_leg:

VADER Sentiment Analysis

Custom tokens
(for finance and r/wsb)

```
'buy': 4.0,
'sell': -4.0,
'rocket': 2.2,
'moon': 4.0,
```

Sentiment Score
(per post)

TD Ameritrade
Stock Screener

Filter based on Price and Market Cap

Price > US$3.0
Market Cap > US$100M

List of Stock Tickers

Extract Stock Mentions

Top 20 Mentioned Stock

Finance

Daily Stock Price

TD Ameritrade
Price History API

Stock Price at time of mention

Stock Price at D+1, D+7, D+30

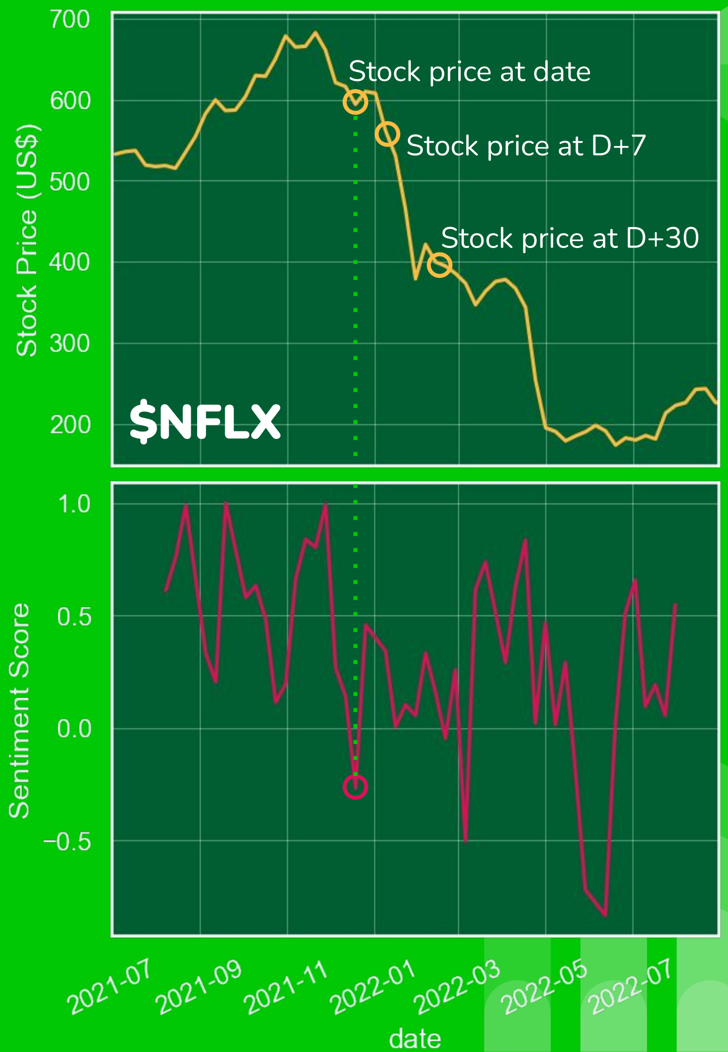
Stock Performance

Correlation bet. Sentiment vs Stock Performance

Stock price vs Sentiment Score

Process:

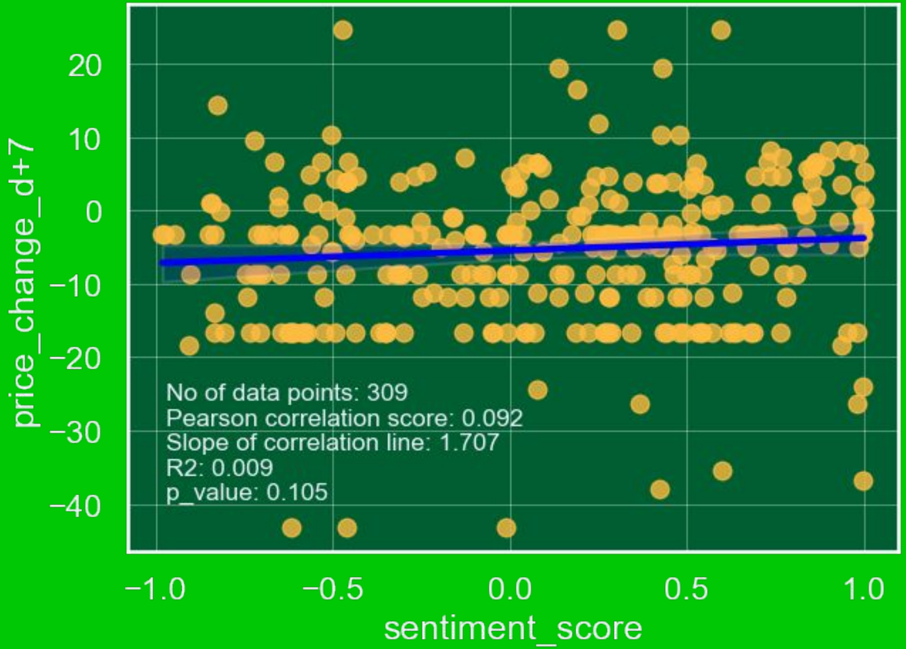
- Analyze **sentiment scores**
- Filter based on **ticker mention**
- Get **stock price** at each **post date**
- Get **stock price** at **D+7** and **D+30**
- Calculate **price change** in price
- **Compare** against sentiment



Results (???)

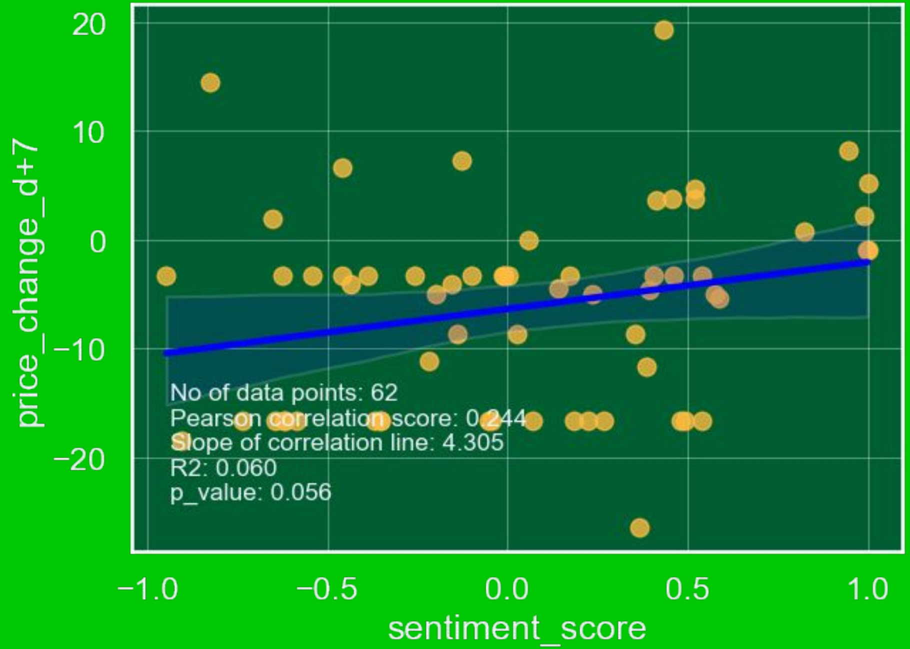
Looking at **NFLX D+7 price** vs **sentiments**

Very noisy! Possible to filter based on **post score**



Results (??)

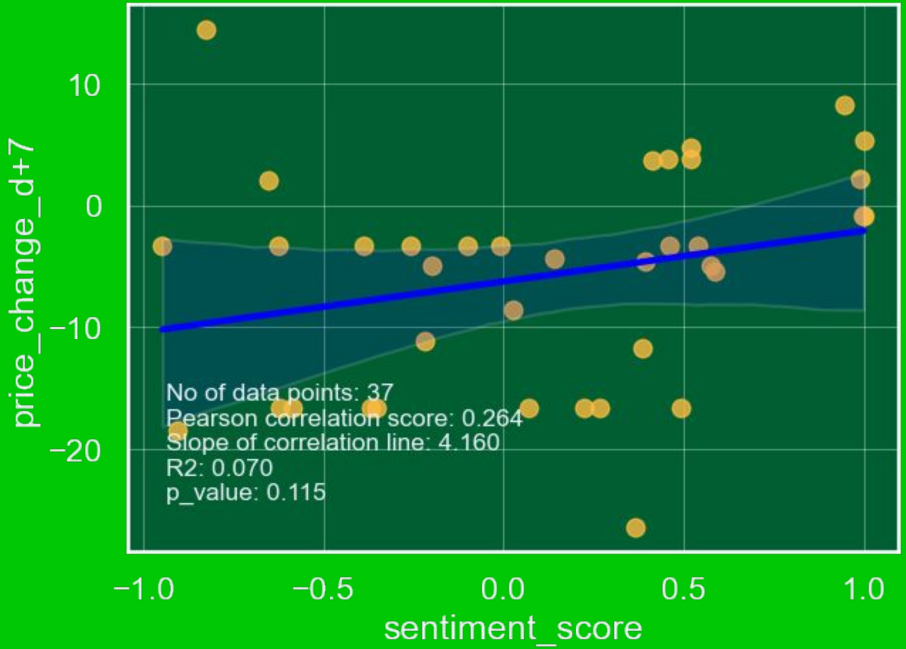
Looking at **NFLX D+7 price** vs **sentiments**
[Filtering based on **post score > 20**]



No of data points: 62
Pearson correlation score: 0.244
Slope of correlation line: 4.305
R2: 0.060
p_value: 0.056

Results (?!)

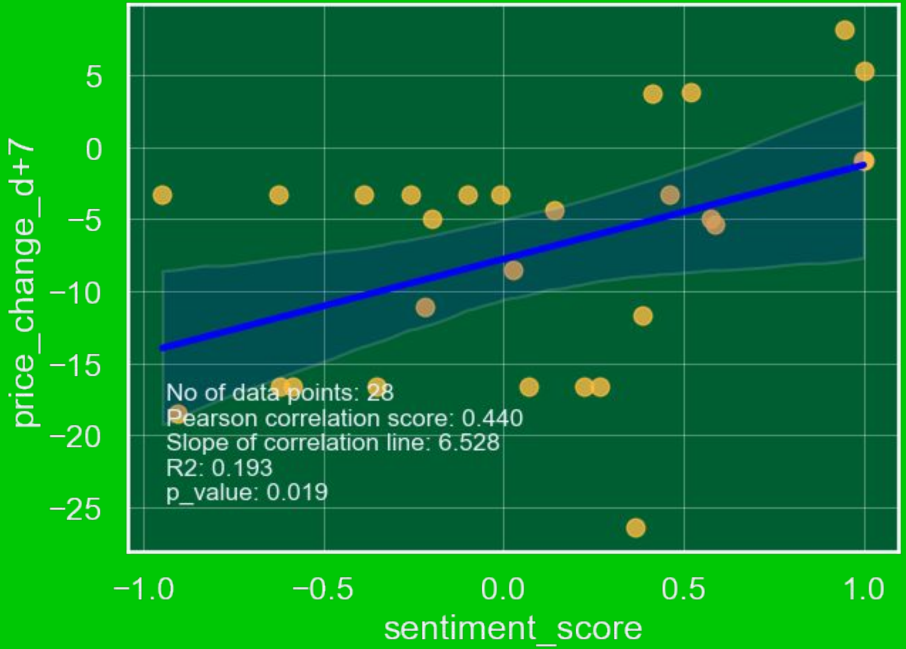
Looking at **NFLX D+7 price** vs **sentiments**
[Filtering based on **post score > 50**]



No of data points: 37
Pearson correlation score: 0.264
Slope of correlation line: 4.160
R2: 0.070
p_value: 0.115

Results (!)

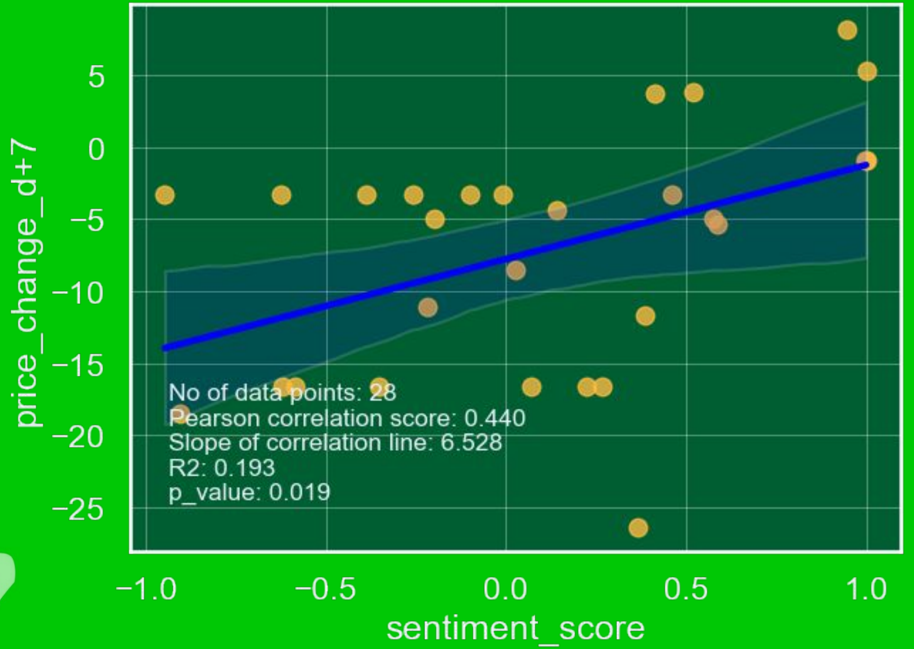
Looking at **NFLX D+7 price** vs **sentiments**
[Filtering based on **post score > 100**]



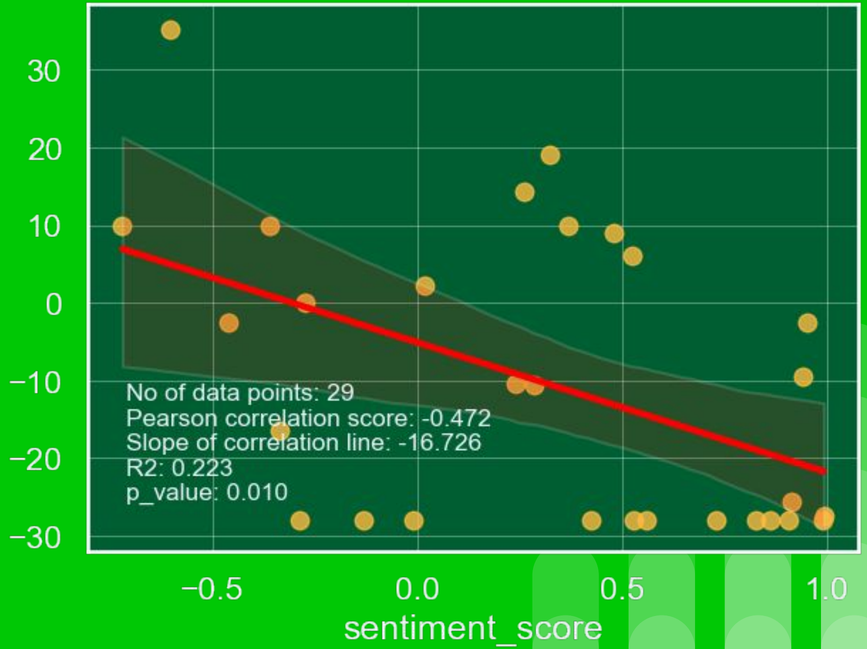
No of data points: 28
Pearson correlation score: 0.440
Slope of correlation line: 6.528
R2: 0.193
p_value: 0.019

In reality...

Looking at **$DWAC** (Digital World Aqcuisition Corp), with the **same settings and filters**:



No of data points: 28
Pearson correlation score: 0.440
Slope of correlation line: 6.528
R2: 0.193
p_value: 0.019



No of data points: 29
Pearson correlation score: -0.472
Slope of correlation line: -16.726
R2: 0.223
p_value: 0.010

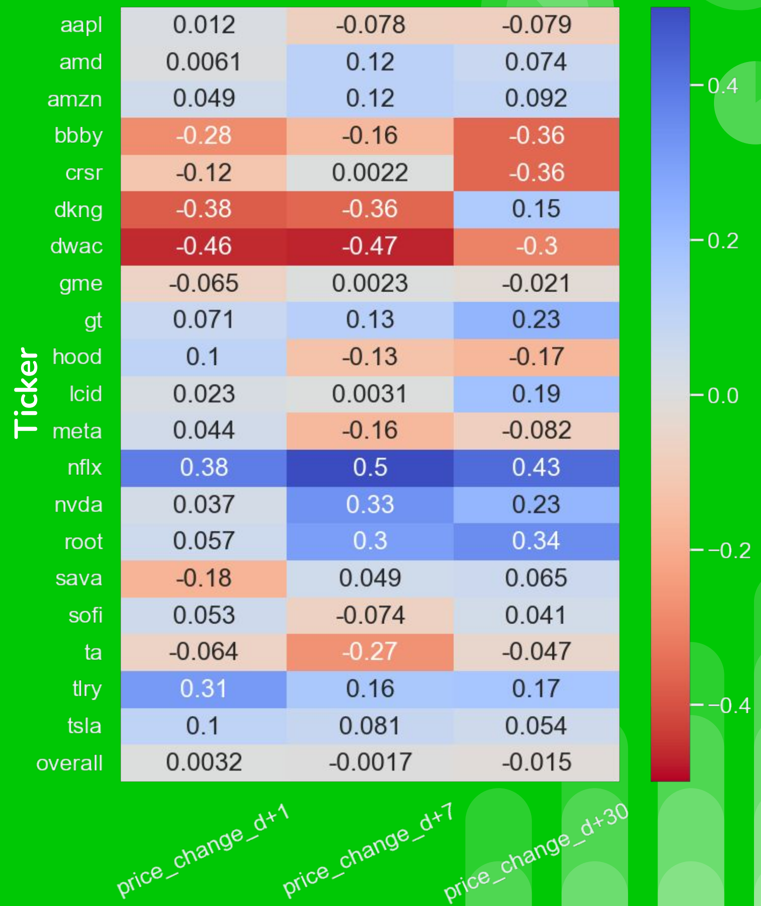
In reality…

The correlations between sentiment score and stock performance is **entirely random**

Looking at the **overall trend**, there is **no meaningful correlation**

[YouTube] I Gave My Goldfish $50,000 to Trade Stocks

Pearson correlation coef.
(Sentiment vs stock performance)



Ticker	price_change_d+1	price_change_d+7	price_change_d+30
aapl	0.012	-0.078	-0.079
amd	0.0061	0.12	0.074
amzn	0.049	0.12	0.092
bbby	-0.28	-0.16	-0.36
crsr	-0.12	0.0022	-0.36
dkng	-0.38	-0.36	0.15
dwac	-0.46	-0.47	-0.3
gme	-0.065	0.0023	-0.021
gt	0.071	0.13	0.23
hood	0.1	-0.13	-0.17
lcid	0.023	0.0031	0.19
meta	0.044	-0.16	-0.082
nflx	0.38	0.5	0.43
nvda	0.037	0.33	0.23
root	0.057	0.3	0.34
sava	-0.18	0.049	0.065
sofi	0.053	-0.074	0.041
ta	-0.064	-0.27	-0.047
tlry	0.31	0.16	0.17
tsla	0.1	0.081	0.054
overall	0.0032	-0.0017	-0.015

Summary

- Tasked with classifying **r/wsb** vs **r/stocks** for **targeted advertising**
    - **Highly imbalanced** dataset

- Used various vectorizer, sampling method, and classifier model
    - Best performance: **Multinomial NB** w/ **CVEC** + **SMOTE**

- Presence of **Data Drift**
    - Models trained on one month performs worse on other months
    - Using **cumulative data** results in better prediction

- Posts in both subreddits tend to have **positive sentiment**
- Sentiment of the subreddits **NOT able to predict** stock performance

Future Works

Part 1

- Obtain more posts data from **previous months**
- Trying **other models** (e.g.: XGBoost, kNN, etc)
- Observe **misclassified** posts
- **Productionize** model

Part 2

- **Consider all stock tickers** in analysis (incl. penny stocks), as wsb is known for analysis on those types of stocks
- Manual modification for **emoji-to-text mapping**

THANK YOU
FOR YOUR ATTENTION
makeameme.org

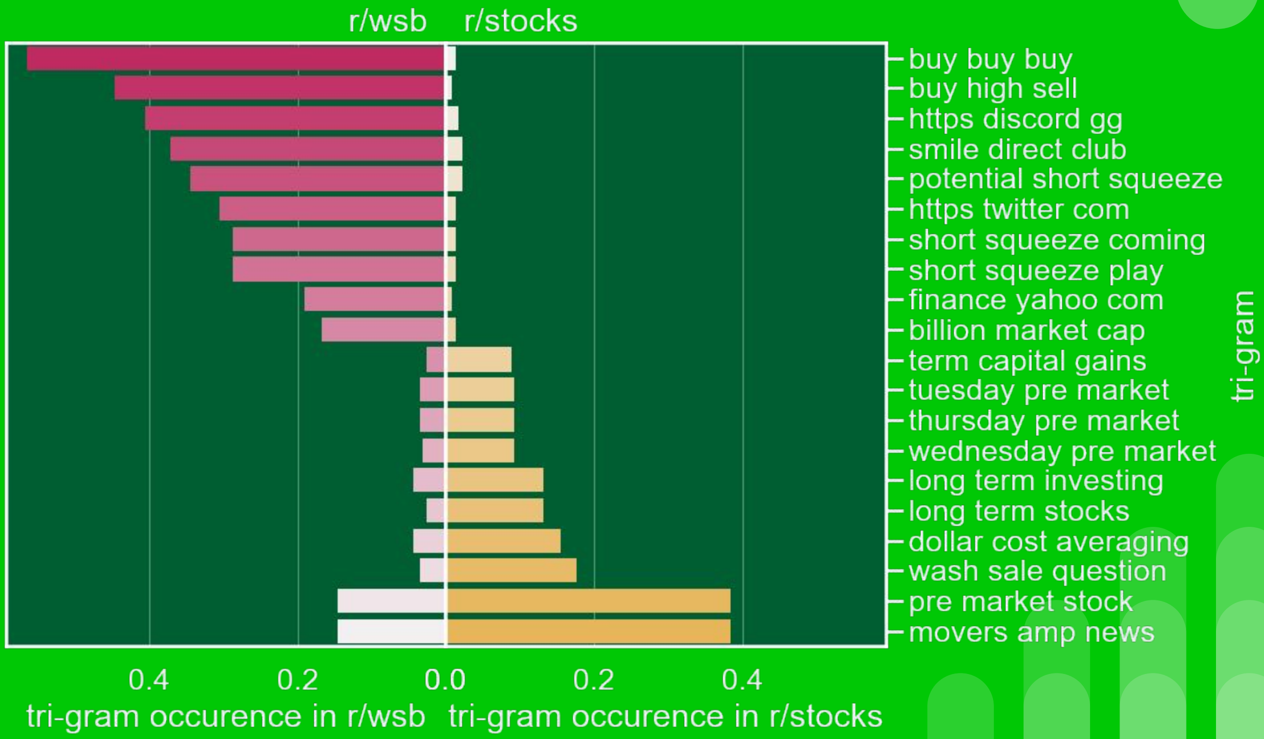
Project Objectives

- **Primary: Targeting advertisement**
  - Use **NLP** to **classify** an unseen post, for ads targeting posts on r/stocks

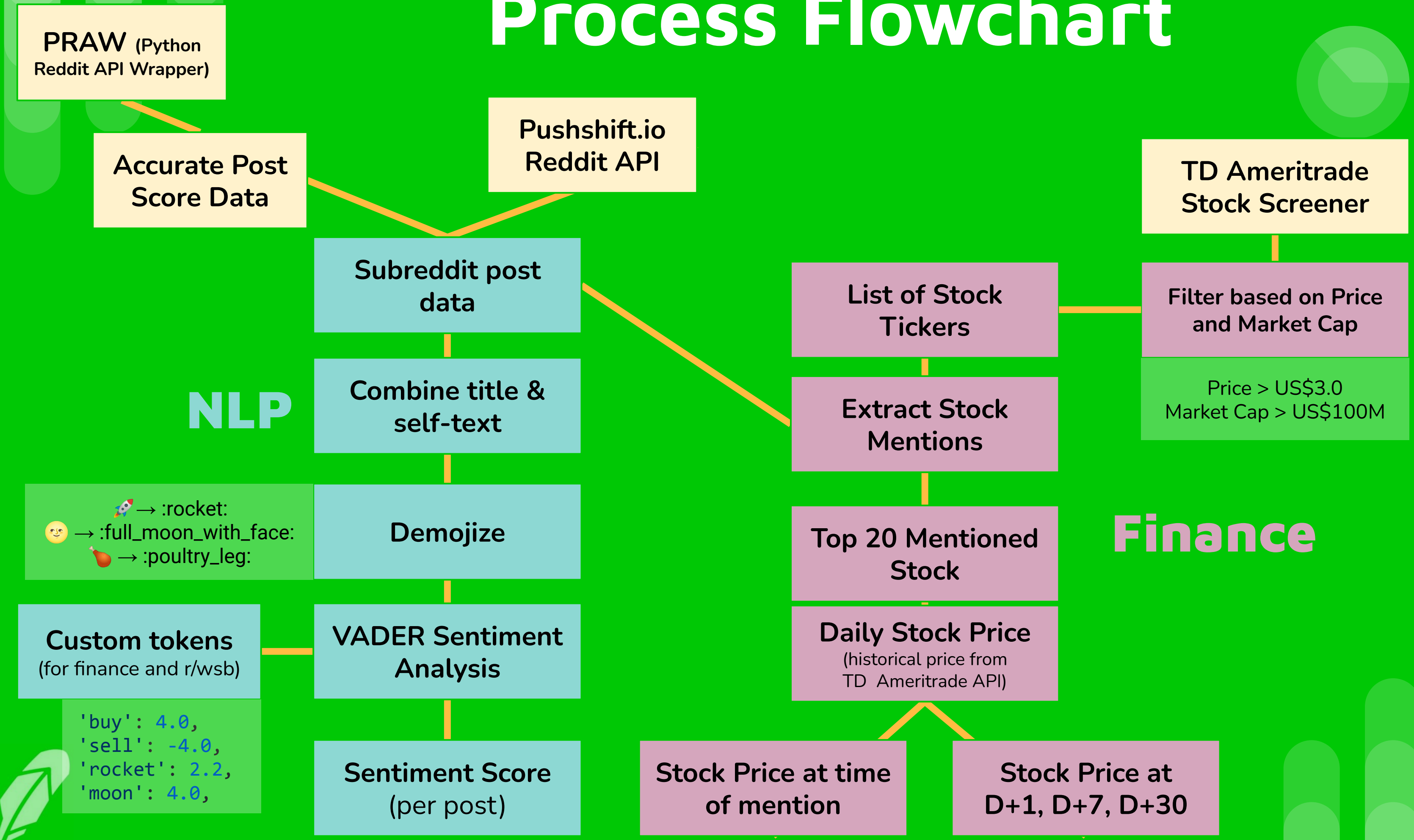
- **Secondary: Inform investment decision *(exploratory)***
  - Analyze **correlation** between the **sentiment** of a particular stock to the **future performance** of that stock.

- **Data scope: Aug 2021 - Aug 2022**
  - Discussions revolving **GameStop** lasted until around Jun/Jul 2021
  - Contextual data **deviated** much from the norm
  - **Excluded** this abnormality from this classification project

Text-Based Eda



Process Flowchart



PRAW (Python Reddit API Wrapper)

Accurate Post Score Data

Pushshift.io Reddit API

Subreddit post data

NLP

Combine title & self-text

Demojize

🚀 → :rocket:
🌝 → :full_moon_with_face:
🍗 → :poultry_leg:

VADER Sentiment Analysis

Custom tokens
(for finance and r/wsb)

```
'buy': 4.0,
'sell': -4.0,
'rocket': 2.2,
'moon': 4.0,
```

Sentiment Score
(per post)

Finance

TD Ameritrade Stock Screener

Filter based on Price and Market Cap

Price > US$3.0
Market Cap > US$100M

List of Stock Tickers

Extract Stock Mentions

Top 20 Mentioned Stock

Daily Stock Price
(historical price from TD Ameritrade API)

Stock Price at time of mention

Stock Price at D+1, D+7, D+30