DSI - Project 3: Web API & Sub-Reddit Classification using NLP

(Subreddits - Dogecoin and Bitcoin)

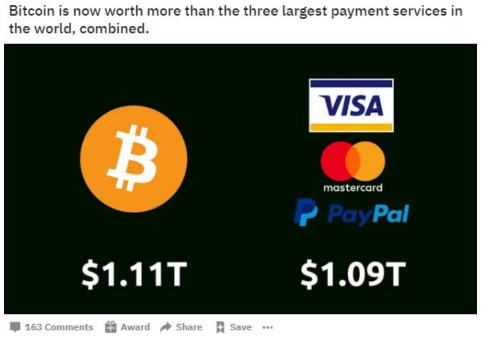
Presented by

Patchara Lertmanatchai (Bob)

Introduction

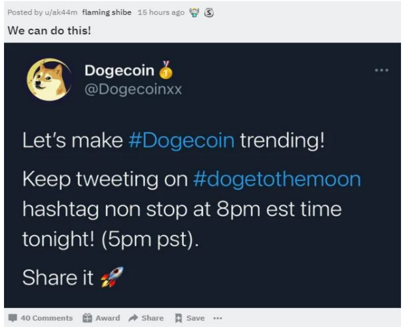
What is Cryptocurrency?

- Digital money created from code
- Blockchain Technology
- Decentralization (Not determined by a central bank)
- Ex. Bitcoin, Ethereum, Dogecoin



Bitcoin is now worth more than the three largest payment services in the world, combined.

Vs



Posted by u/ak44m flaming shibe 15 hours ago

We can do this!

Dogecoin
@Dogecoinxx

Let's make #Dogecoin trending!

Keep tweeting on #dogetothemoon hashtag non stop at 8pm est time tonight! (5pm pst).

Share it 🚀

Bitcoin - The Currency of the Internet

r/Bitcoin

Dogecoin

r/dogecoin

"Bitcoin and Dogecoin are among the most popular cryptocurrencies today"

Executive summary

Posted by u/izzak83 21 hours ago 2 2 & 4 More

Its always about the Doge. Always. Serious

Lex Fridman @lexfridman · Mar 18, 2021

This tweet is not about Bitcoin.

Not everything is about Bitcoin.

Or is it?

Elon Musk @elonmusk

Sometimes it's about Doge

4:17 PM · Mar 18, 2021

48.2K 5.2K Share this Tweet

Issues:

- Extremely large posts on reddit
- Bugs in the system or human errors of creating subreddit tags
- Misleading by contents and topic titles

Objective:

To use machine learning to classify similar content from two different subreddits, /r/Bitcoin and /r/dogecoin?

The Approach:

- Data collection : **Web Scrape sub-Reddits posts with Reddit APIs (1000 posts each)**
- Data cleaning & EDA
- Word Vectorizers : **CountVectorizer and TF-IDF Vectorizer**
- Classification Models

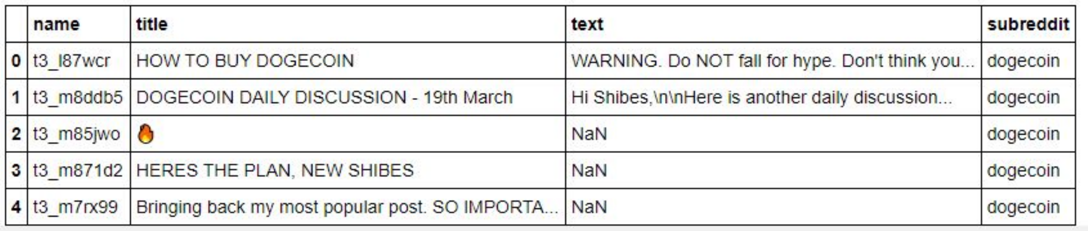
Evaluation:

- Train/Test Split Data
- Classification Matrix

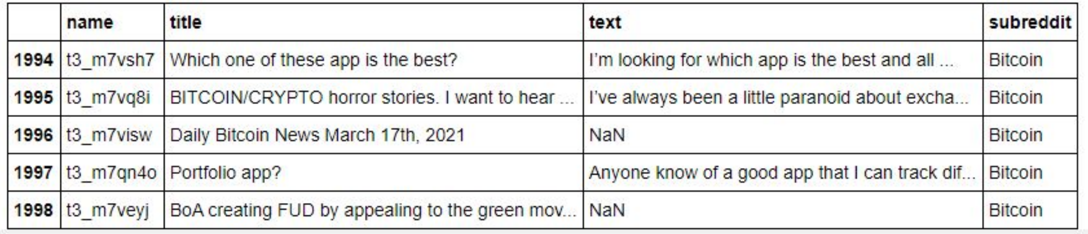
Data Collection

Reddit APIs:

- Limitation of Reddit's APIs, it can only be able to pull **25 requests per request**
- Loop these pulls **40 times** in order to get 1,000 posts each from /r/Bitcoin and /r/dogecoin
  - Name ID
  - Title
  - Selftext
  - Subreddit
- Save the results from scrape as a **JSON**. from Reddit then convert to **Pandas DataFrame** and finally save as **CSV.** file so no need to scrape posts again if something goes wrong in loop.



	name	title	text	subreddit
0	t3_l87wcr	HOW TO BUY DOGECOIN	WARNING. Do NOT fall for hype. Don't think you...	dogecoin
1	t3_m8ddb5	DOGECOIN DAILY DISCUSSION - 19th March	Hi Shibes,\n\nHere is another daily discussion...	dogecoin
2	t3_m85jwo	🔥	NaN	dogecoin
3	t3_m871d2	HERES THE PLAN, NEW SHIBES	NaN	dogecoin
4	t3_m7rx99	Bringing back my most popular post. SO IMPORTA...	NaN	dogecoin

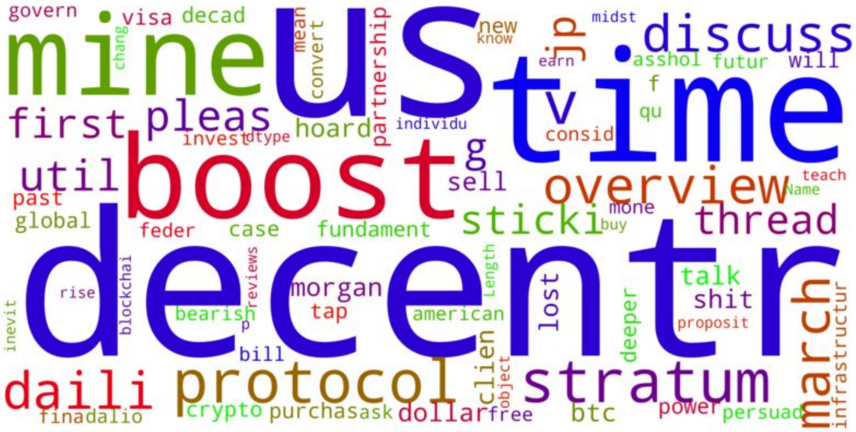


	name	title	text	subreddit
1994	t3_m7vsh7	Which one of these app is the best?	I'm looking for which app is the best and all ...	Bitcoin
1995	t3_m7vq8i	BITCOIN/CRYPTO horror stories. I want to hear ...	I've always been a little paranoid about excha...	Bitcoin
1996	t3_m7visw	Daily Bitcoin News March 17th, 2021	NaN	Bitcoin
1997	t3_m7qn4o	Portfolio app?	Anyone know of a good app that I can track dif...	Bitcoin
1998	t3_m7veyj	BoA creating FUD by appealing to the green mov...	NaN	Bitcoin

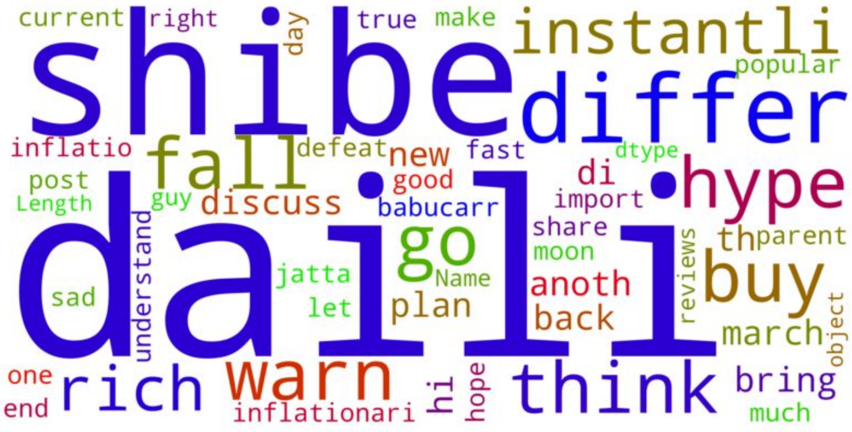
Data Cleaning and Exploratory Data Analysis (EDA)

- **Remove duplicated data (name_id)**
- **Fill NaN values with empty string for 'text', else the operation cannot be done**
- **Combined Title and Text Data into a single column**
- **Extracts an element value from an XML string (*&amp, &gt; and &lt)***
- **Create reveiew_to_words function to clean the data**
  - **Remove HTML**
  - **Remove Non-letters (numerics, new line separators, punctuations)**
  - **Convert to lower-case**
  - **Remove Stopwords**
  - **Stemming using NLTK**
- **Binarizing Target Variable subreddit**
  - **Engineer a feature to turn**
    1. **Bitcoin = 1**
    2. **dogecoin = 0**
- **Further Visualizations: Word Clouds**
- **Frequent Words (Top 20 of each subreddit)**

Word Cloud: Bitcoin and Dogecoin



Word Cloud for /r/Bitcoin



Word Cloud for /r/dogecoin

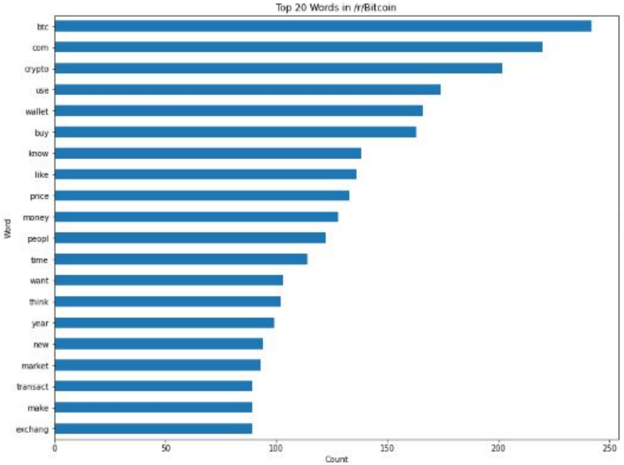
Words that are more unique to **r/Bitcoin** are:

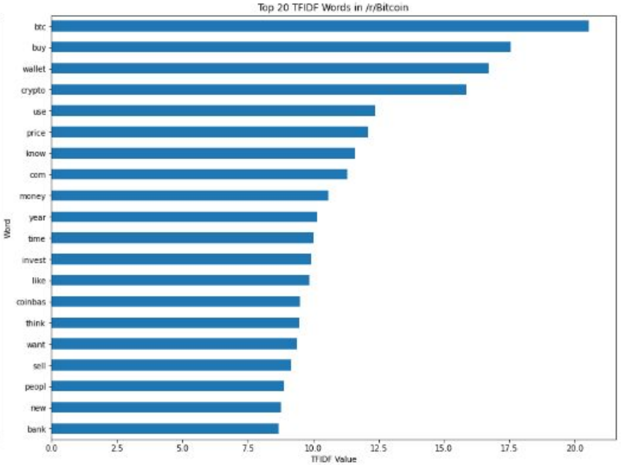
- **Decentr**
- **Time**
- **Us**
- **Mine**
- **boost**

Words that are more unique to **r/dogecoin** are:

- **Daily**
- **Shibe**
- **Differ**
- **instantli**

Top 20 words: Bitcoin and Dogecoin

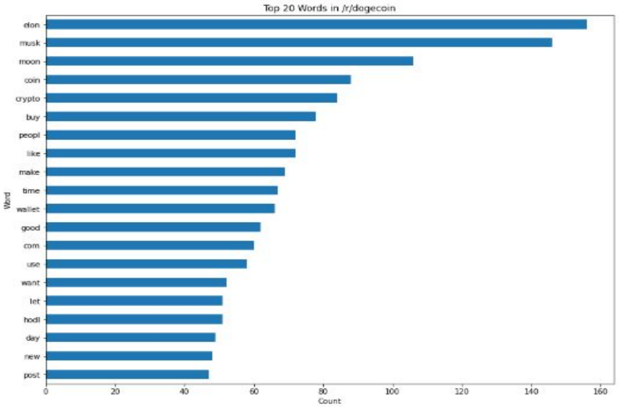


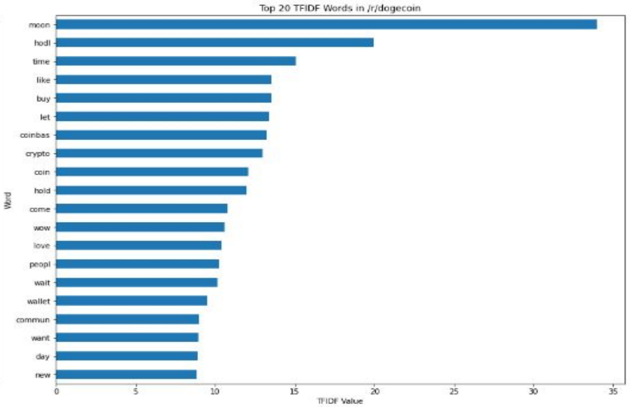


Frequent Words in /r/Bitcoin

btc crypto

buy wallet



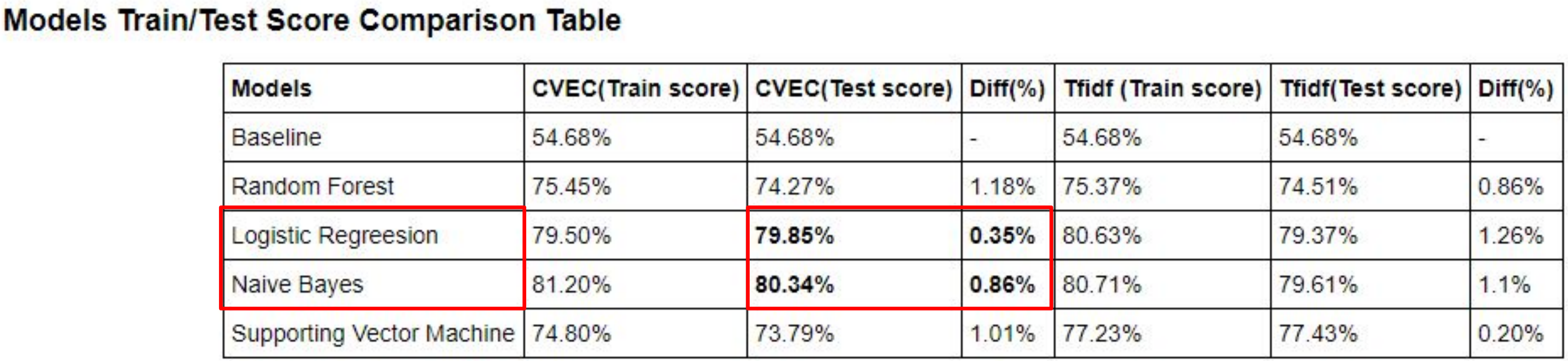


Frequent Words in /r/dogecoin

moon coin

elon musk

Preprocessing and Modeling



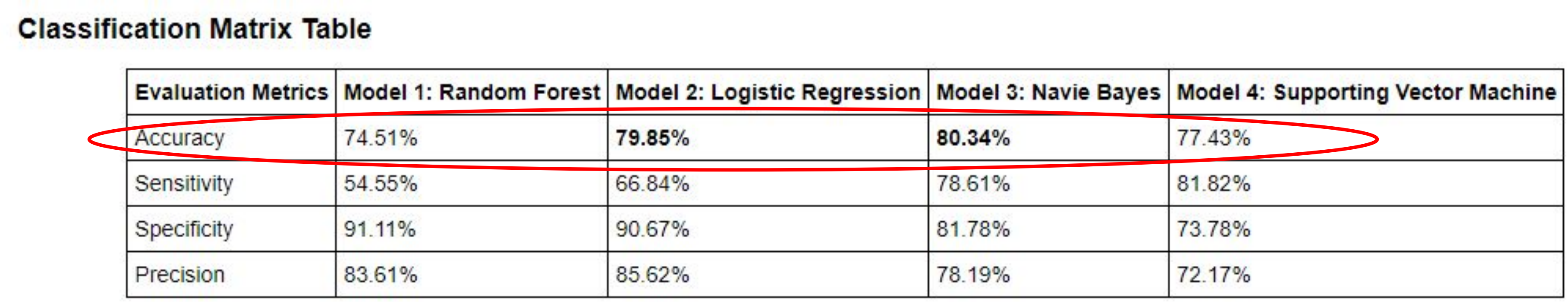
Models Train/Test Score Comparison Table

Models	CVEC(Train score)	CVEC(Test score)	Diff(%)	Tfidf (Train score)	Tfidf(Test score)	Diff(%)
Baseline	54.68%	54.68%	-	54.68%	54.68%	-
Random Forest	75.45%	74.27%	1.18%	75.37%	74.51%	0.86%
Logistic Regreesion	79.50%	**79.85%**	**0.35%**	80.63%	79.37%	1.26%
Naive Bayes	81.20%	**80.34%**	**0.86%**	80.71%	79.61%	1.1%
Supporting Vector Machine	74.80%	73.79%	1.01%	77.23%	77.43%	0.20%

Key Insights

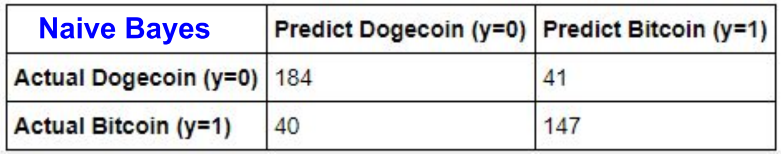
- **Baseline Accuracy** is 54.68%
  - At given any posts, we would predict that it belongs to the Dogecoin (y =0) subreddit everytime
- **Naive Bayes** does best in Testing set at 80.34%
  - A bit overfitting evident here since the train accuracy score is actually 0.86 higher than the testing score
  - Seeing how close the training and testing accuracy scores are to each other suggests that our model's performance is consistent when applied to unseen data.
- **Logis Regression** (79.85%)
  - Does better than the Random Forest and SVB since both Training and Testing accuracy scores were higher.
  - However, the testing set's accuracy score is lower than the training set's which may suggest a very small degree of overfitting.

Confusion Matrix



Classification Matrix Table

Evaluation Metrics	Model 1: Random Forest	Model 2: Logistic Regression	Model 3: Navie Bayes	Model 4: Supporting Vector Machine
Accuracy	74.51%	**79.85%**	**80.34%**	77.43%
Sensitivity	54.55%	66.84%	78.61%	81.82%
Specificity	91.11%	90.67%	81.78%	73.78%
Precision	83.61%	85.62%	78.19%	72.17%



Naive Bayes	Predict Dogecoin (y=0)	Predict Bitcoin (y=1)
Actual Dogecoin (y=0)	184	41
Actual Bitcoin (y=1)	40	147

Key Insights

- **Overall:** Almost models performed well (Accuracy > 75%)
- **Naive Bayes** outperformed all other models in terms of **Accuracy**
  - So out of the 412 testing posts, we see 184+147 = **331 posts were correctly classified**, while 40+41= **81 posts were misclassified** under the multinomial Naive Bayes

Conclusion and Recommendation

Key Takeaways

- All models exceed the baseline accuracy (54.68%)
- Almost accuracy > 75% → Good ability to accurately classify posts
- Definitely Winner: **Multinomial Naive Bayes with CountVectorizer Model (4 in 5 correct or 80%)**
  - **Implication:** Reddit Data Team need to review 1 in 5 misclassified posts
  - This accuracy is within expectation because the topics of two chosen subreddits are quite similar.

Further Improvements

- Collecting more training data not only just Text data but also Pic posted in a post as a feature to classify subreddits.
- Enhancing the model's ability to classify more than two subreddits in our classification model.
- Tuning of parameters for any models to get a better score. However, this requires a longer amount of time to tune to get the perfect parameters.