Cryptocurrency Sentiment Analysis Using NLP Technology Based On Tweet

Po-Han Lu
- Cross College Elite Program, National Cheng Kung University
- F14081088

Chun-Rong Huang | Professor
- Cross College Elite Program, National Cheng Kung University
- Advisor

Abstract

The rise of web3.0 products has fueled cryptocurrency popularity. Knowing public sentiment is crucial for product managers and marketers. This project applies Natural Language Processing (NLP) [1] to analyze Bitcoin-related tweets. Through data collection, preprocessing, and sentiment analysis model development, accurate classification of tweets into positive and negative sentiments is achieved. Model performance is evaluated using the Confusion Matrix and Accuracy.

Motivation

- Transformative Potential: Analyzing Bitcoin sentiment on Twitter, which provides insights into investor sentiment, market trends, and price volatility.
- Actionable Insights: Providing information for product managers and marketing teams based on sentiment analysis of Bitcoin-related tweets.
- Personal Development: Gaining hands-on experience in NLP, sentiment analysis, and programming while building a strong portfolio.

Related Work

Natural Language Processing, NLP

NLP helps computers analyze human language. By applying it to tweet sentiment analysis, insights are extracted from short messages, aiding researchers in understanding public opinion and social media trends.

Dataset [2]

id	target	user_id	date	flag	user	text
446717	0 (Negative)	2068495354	Sun Jun 07 14:10:44 PDT 2009	NO_QUERY	AshVicious	@Kolbijean The same exact thing happened here last night to! We heard gunshots
847441	4 (Positive)	1564585293	Mon Apr 20 04:09:09 PDT 2009	NO_QUERY	trashGold	@zackalltimelow happy birthday dudelet this time last year i was watching you guys on trl i think

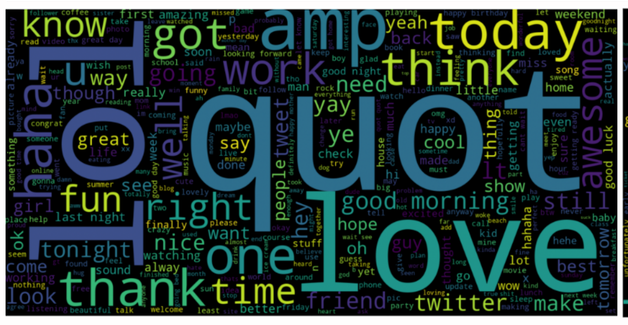


Figure Positive WordCloud

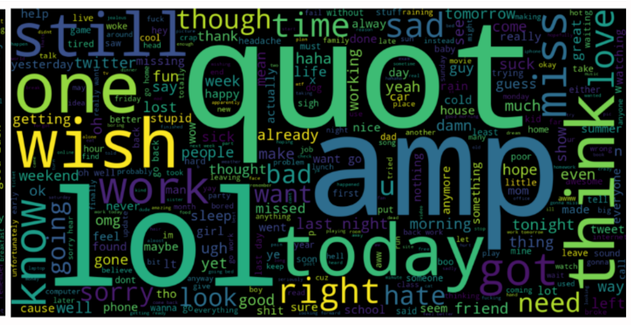


Figure Negative WordCloud

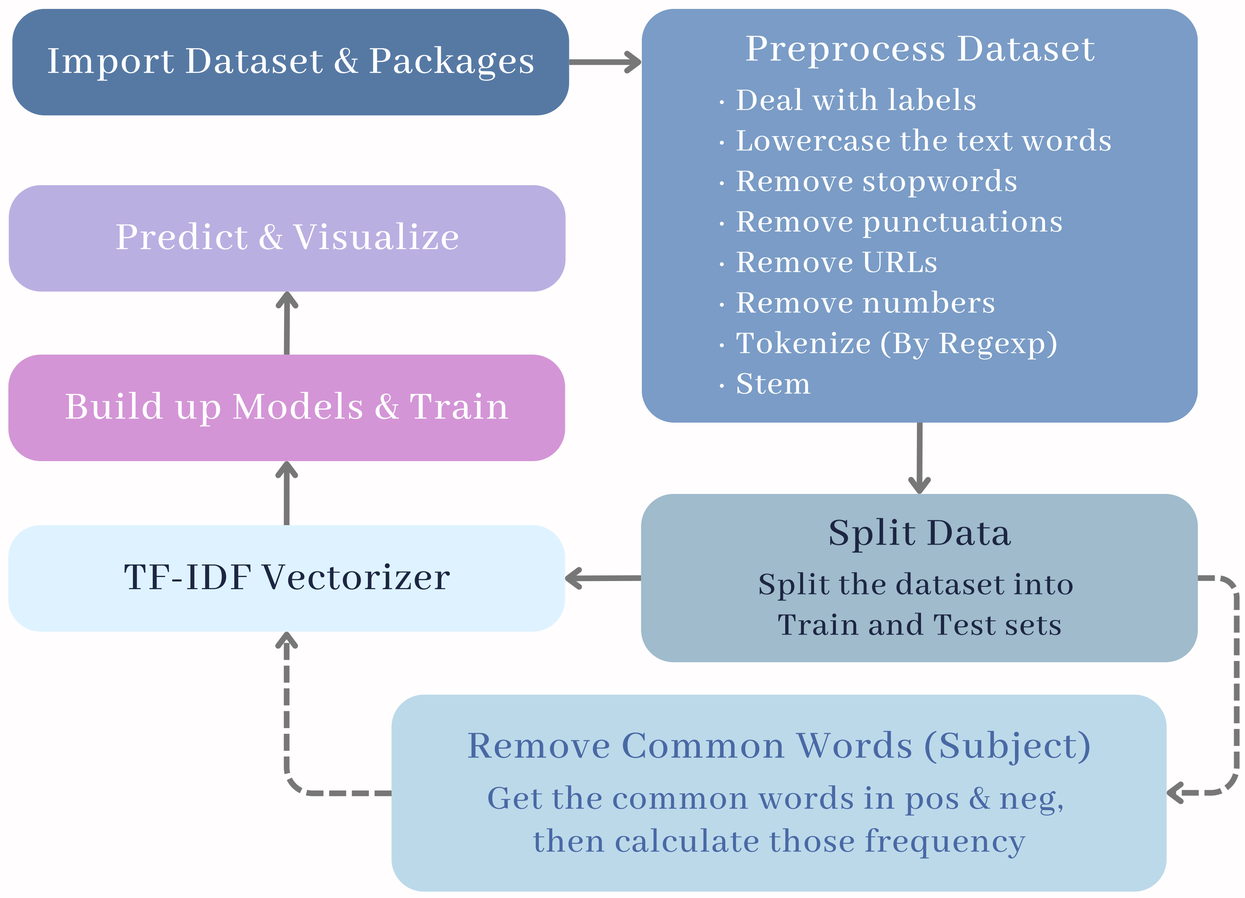
Naive Bayes Classifier [3]

Naive Bayes Classifier is a popular ML algorithm for classification tasks. It's based on Bayes' theorem, calculating event probabilities with prior knowledge. Despite simplicity and feature independence assumptions, Naive Bayes is efficient and performs well in real-world applications.

LogisticRegression [4]

Logistic Regression is a potent algorithm for binary classification. It models the relationship between features and outcomes using the logistic function, mapping inputs to probabilities from 0 to 1. By fitting the model to training data, it learns coefficients to maximize outcome likelihood. It's effective when decision boundaries are linear or approximated well by linear functions.

Flow Chart



Result: Removing common words

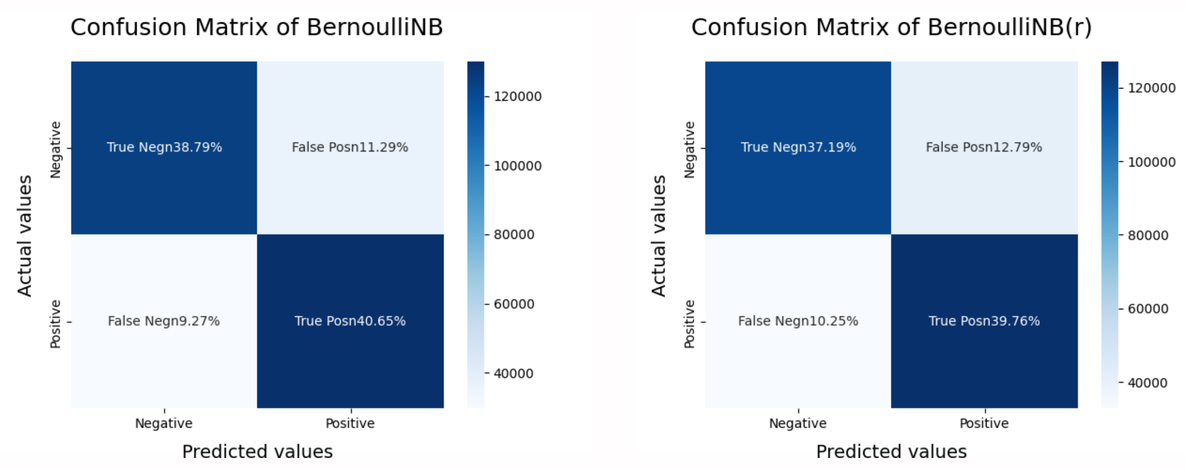
	BernoulliNB	LinearSVC	LogisticRegression	MultinomialNB	1D conv	LSTM
origin	0s	24s	130s	0s	300s	600s
remove	0s	23s	117s	1s	630s	900s

↑ Training time: Remove common words took 430s

	BernoulliNB	LinearSVC	LogisticRegression	MultinomialNB	1D conv	LSTM
origin	0.79	0.80	0.81	0.80	0.71	0.50
remove	0.77	0.77	0.78	0.77	0.53	0.49

↑ Accuracy

Result: Removing common_BernoulliNB



↑ Confusion Matrix

Reference

- [1] Clark, A. (2010). An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall.
- [2] Sentiment140 dataset with 1.6 million tweets from Kaggle
- [3] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
- [4] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. Wiley.

Conclusion: removing common words

Loss of Context

Indiscriminate removal of common words disregards contextual significance, resulting in valuable information loss. Common words carry sentiment clues within sentence usage and context.

Insufficient Data and Overfitting

Removing common words manually may introduce biases or disrupt dataset balance, which might cause overfitting, hindering the model's ability to generalize to new, unseen data.

Conclusion: 1D conv vs LSTM

Local Pattern Extraction

1D CNNs excel at capturing sequential data's local patterns. Using filters with small receptive fields, they detect specific features in the data. In sentiment analysis, where word order or phrase structure signifies sentiment, 1D CNNs effectively capture these patterns.

Robustness to Sequence Length

1D CNNs are robust to varying sequence lengths in sentiment analysis. They do not rely on the entire sequence's sequential dependencies and can handle inputs of different lengths without extra processing steps like padding or truncation required by LSTM models.