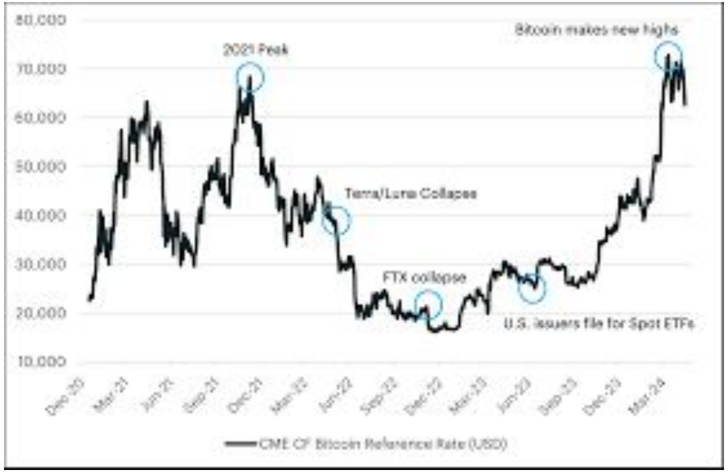
Cryptocurrency Market Sentiment Analysis: A Transformers, Lexicon and classical Models based approach

Teddy Thomas

Motivation

Crypto market is highly sentiment-driven
Retail investors rely on platforms like **StockTwits**
Predicting sentiment helps with:

- Automated trading systems
- Market trend analysis
- Retail investor sentiment tracking



Problem Statement

? How can we accurately classify the sentiment (**Bullish, Bearish, or Neutral**) of user-generated posts on **StockTwits** related to **cryptocurrencies** such as **Bitcoin, Ethereum, and Shiba Inu** using a combination of **traditional machine learning**, **lexicon-based,** and **transformer-based NLP models?**

$BTC to the moon 🚀!

Objective

Classify crypto-related StockTwits posts into:

- **Bullish (2)**
- **Bearish (1)**
- **Neutral (0)**

Compare performance of:

- Lexicon-based models
- Traditional ML classifiers
- Word embeddings
- Transformer-based models

Dataset Description: ElKulako/stocktwits-crypto

Platform: StockTwits

Collection Period:

Training: 1 Nov 2021 – 15 Jun 2022
Testing: 16 Jun 2022 – 30 Jun 2022

Volume:

Total StockTwits Posts: **1.875 million**
Training Set: **1.332 million posts**
Test Set: **83,257 posts**

Sentiment Labels:

Bullish / Bearish (self-labeled by users)

Neutral inferred when no label is given

Cryptocurrencies Covered:

Bitcoin (BTC.X)

Ethereum (ETH.X)

Shiba Inu (SHIB.X)
(Only these are used for supervised training)

Preprocessing Steps:

Removed: URLs, usernames (**@**), cashtags (**$**), hashtags (**#**), wallet addresses, emojis (except for LUKE use), non-English scripts

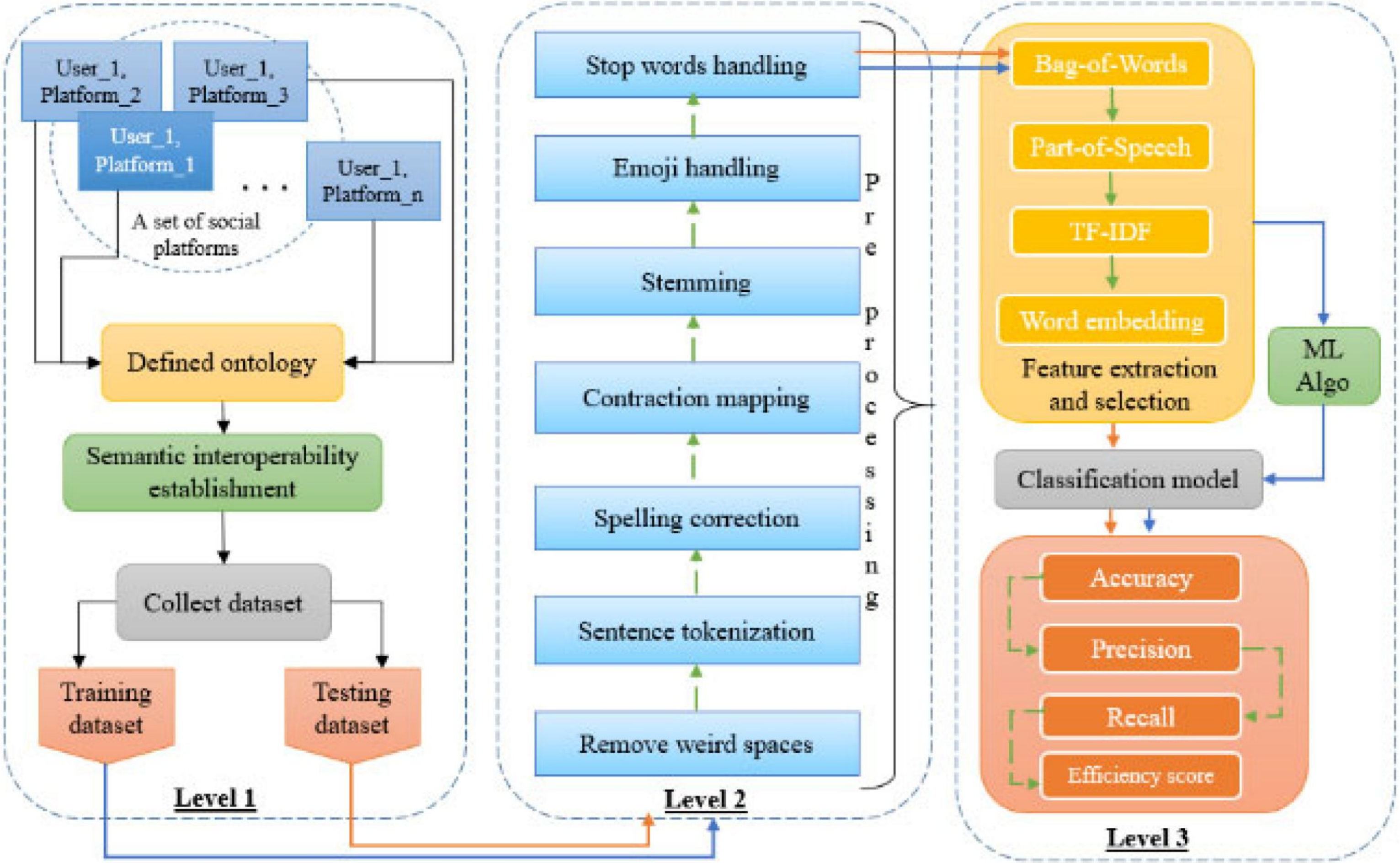
Fixed: encoding issues, multiple dots/spaces

Lowercased text, dropped duplicates and short posts (<4 words)

@elonmusk $BTC 😎😎💥💥
->just BTC

Citation :: Dataset: M. Kulakowski and F. Frasincar, "Sentiment Classification of Cryptocurrency-Related Social Media Posts," in IEEE Intelligent Systems, vol. 38, no. 4, pp. 5-9, July-Aug. 2023, doi: 10.1109/MIS.2023.3283170.

Sentiment Analysis: Methodology



Methodology Overview

Lexicon-Based: VADER SentimentIntensityAnalyzer

Classical ML:

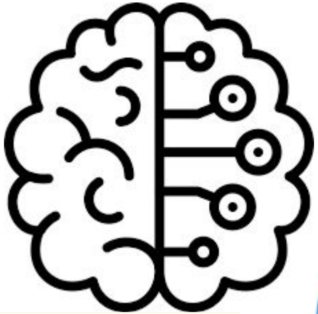
- CountVectorizer + Naive Bayes
- CountVectorizer + Logistic Regression
- TF-IDF + Linear SVM

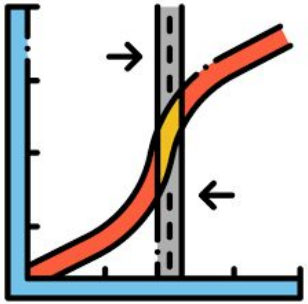
Embeddings:

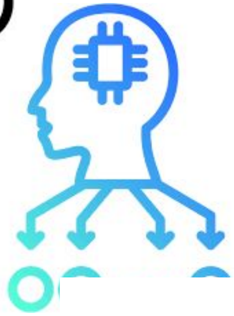
- Word2Vec + Logistic Regression

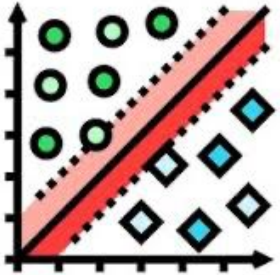
Transformer-Based:

- BERT Sentence Embeddings + ClassifierDL (Spark NLP)



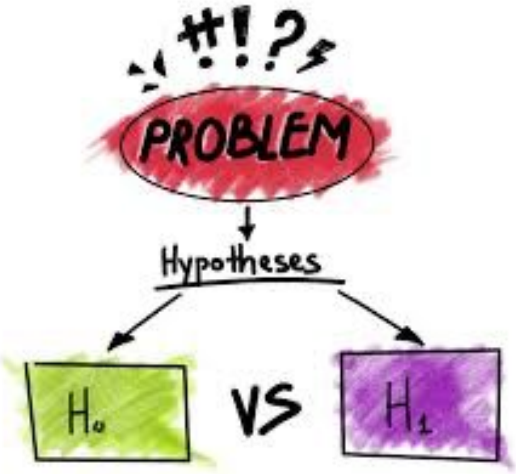






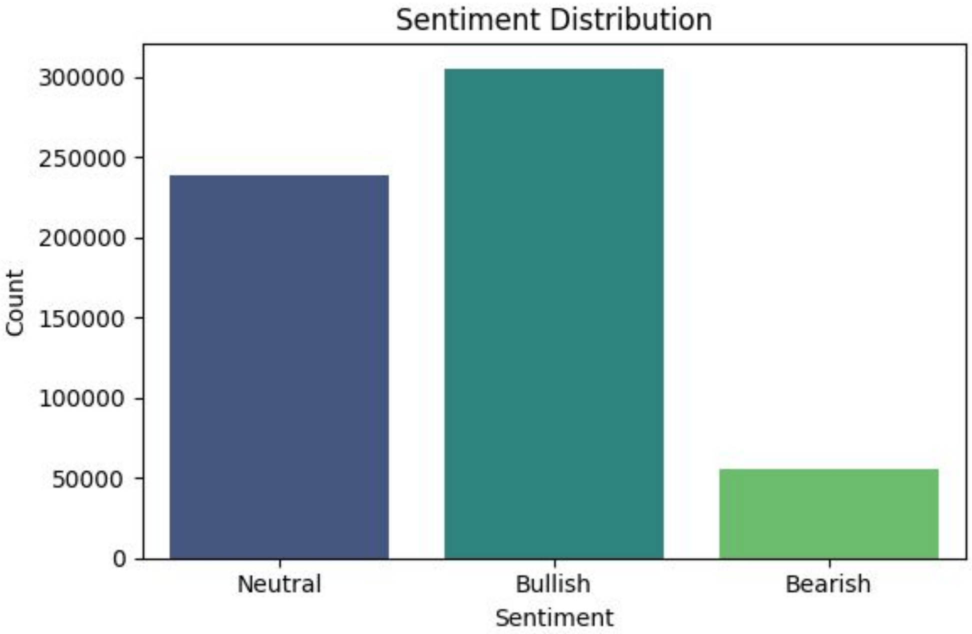
🧠 Choosing What to Use

Hypothesis

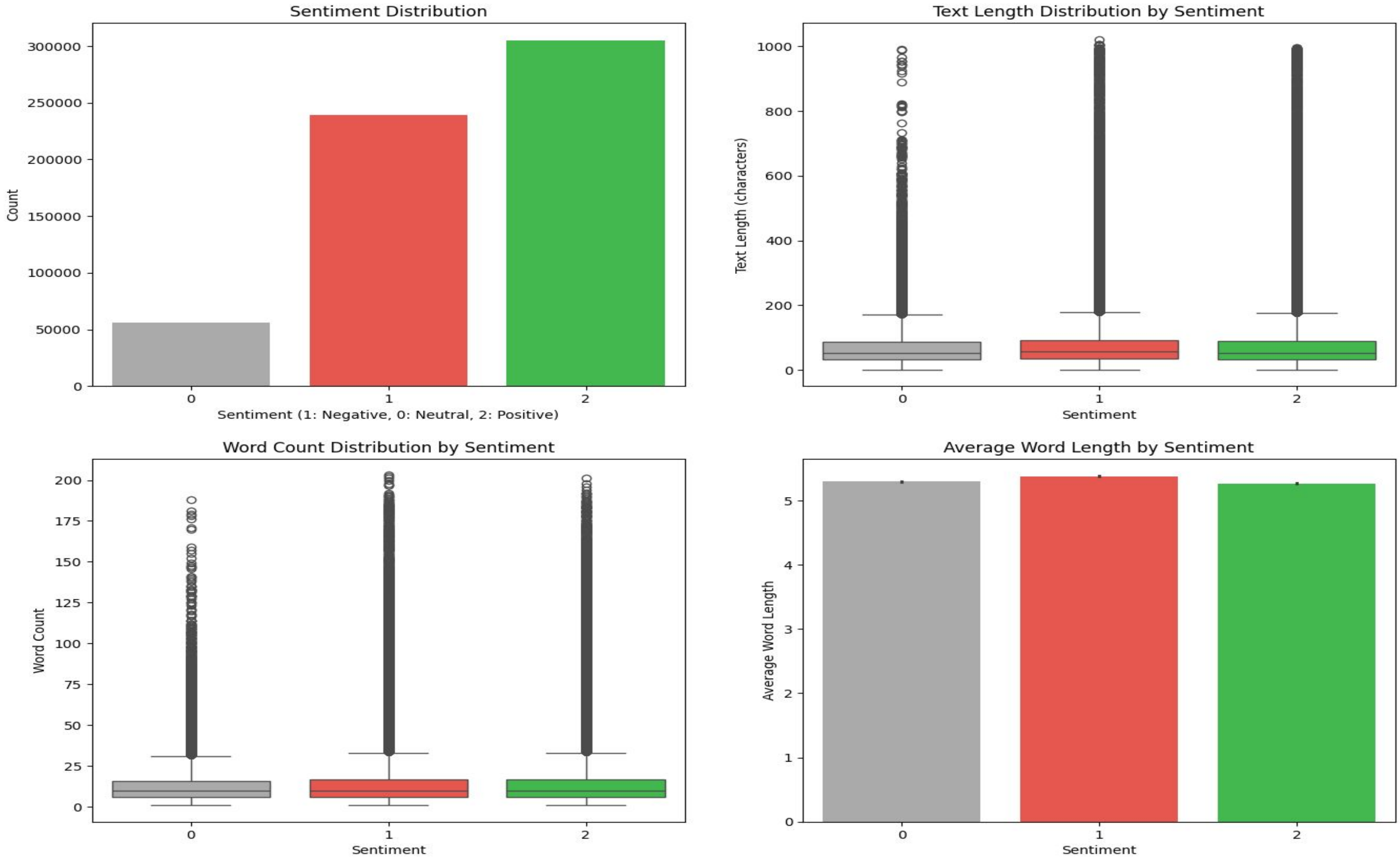


Goal	Best Choice
Simple model + small data	CountVectorizer or TF-IDF
Classical ML + interpretability	TF-IDF
Deep learning	Word2Vec / GloVe / FastText
Best accuracy	BERT embeddings / fine-tuning
Sentence-level meaning	Sentence Transformers / USE
Real-time / speed-sensitive	Hashing Vectorizer or distilled models

Exploratory Data Analysis



Label distribution:
sentiment
Bullish 305092
Neutral 239054
Bearish 55854

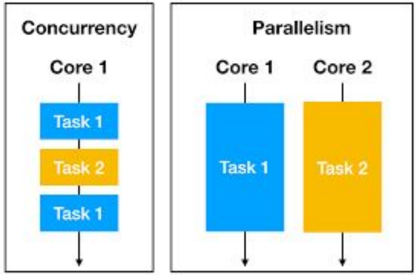


Sentiment Distribution
Count
300000
250000
200000
150000
100000
50000
0
0
1
2
Sentiment (1: Negative, 0: Neutral, 2: Positive)
Text Length Distribution by Sentiment
Text Length (characters)
1000
800
600
400
200
0
Sentiment
Word Count Distribution by Sentiment
Word Count
200
175
150
125
100
75
50
25
0
Sentiment
Average Word Length by Sentiment
Average Word Length
5
4
3
2
1
0
Sentiment


Correlation Between Text Features and Sentiment
label
text_length
word_count
avg_word_length
1
0.004271067
0.01085273
−0.03119113
0.989487
0.1722451
0.09710967
1
0.8
0.6
0.4
0.2
0

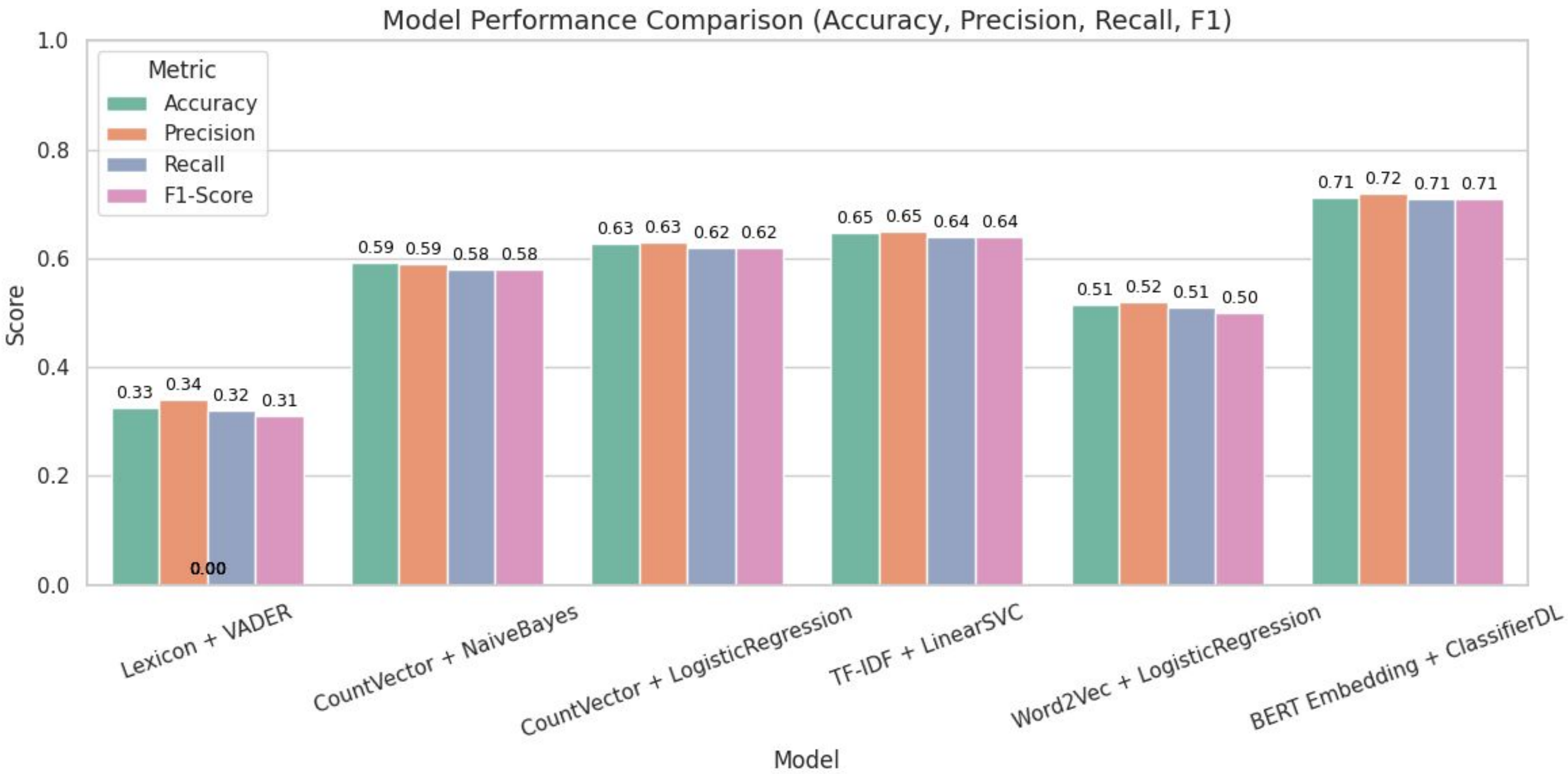
Big Data & Scalability

Apache Spark on Databricks: Used for distributed processing of 1.8M+ posts using Spark DataFrames, MLlib pipelines, and UDFs.
Parallelism: Tokenization, vectorization (TF-IDF, Word2Vec), and model training were executed in parallel across Spark workers for scalability.
Batch Processing: Data was processed in partitions to handle memory efficiently, especially during transformation and model inference stages.
CBCB HPC Cluster: Employed for heavy transformer-based models like BERT; used SLURM for job scheduling and high-performance GPU computation.
Hybrid Architecture: Combined cloud (Databricks) and HPC (CBCB) for optimal resource usage and faster execution of large-scale NLP tasks.



Model Evaluation

Model	Accuracy	Precision	Recall	F1-Score
VADER	0.3253	0.34	0.32	0.31
Naive Bayes	0.5911	0.59	0.58	0.58
Logistic Regression	0.6267	0.63	0.62	0.62
Linear SVC	0.6472	0.65	0.64	0.64
Word2Vec + LR	0.5145	0.52	0.51	0.50
BERT (Spark NLP)	0.7118	0.72	0.71	0.71


Model Performance Comparison (Accuracy, Precision, Recall, F1)
Metric
Accuracy
Precision
Recall
F1-Score
Score
1.0
0.8
0.6
0.4
0.2
0.0
0.33 0.34 0.32 0.31
0.00
0.59 0.59 0.58 0.58
0.63 0.63 0.62 0.62
0.65 0.65 0.64 0.64
0.51 0.52 0.51 0.50
0.71 0.72 0.71 0.71
Lexicon + VADER
CountVector + NaiveBayes
CountVector + LogisticRegression
TF-IDF + LinearSVC
Word2Vec + LogisticRegression
BERT Embedding + ClassifierDL
Model

Analysis and Insights

Lexicon-based approach underperforms on short, informal text

Classical models improve with better vectorization

Transformer-based model performed best

Word2Vec was sensitive to training data quality

Trade-off: Simpler models are faster, but less accurate

Challenges

Handling imbalanced and noisy sentiment labels

Short length of social media posts

Sparse text with emojis, tickers, slang

Limited labeled neutral examples (inferred)

References & Citations

[1] Dataset: M. Kulakowski and F. Frasincar, "Sentiment Classification of Cryptocurrency-Related Social Media Posts," in IEEE Intelligent Systems, vol. 38, no. 4, pp. 5-9, July-Aug. 2023, doi: 10.1109/MIS.2023.3283170.

[2] J. Devlin et al.,“BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT), 2019, pp. 4171–4186.

[3] © Databricks 2025. All rights reserved. Apache, Apache Spark, Spark and the Spark logo are trademarks of the Apache Software Foundation.

Thank YOU!

What is the Ground Truth Here?

Your dataset provides sentiment labels based on **user annotations**:

- **Bullish** → Label: 2(Positive sentiment)
- **Bearish** → Label: 1X(Negative sentiment)
- **Neutral** → Label: 0 (Inferred when no explicit sentiment label is provided)

These sentiment labels are either:

- **Explicitly provided by users** (Bullish/Bearish tags on StockTwits), or
- **Inferred** (Neutral, when no tag exists).