Tesla Stock Price Forecasting with LSTM and Tweet Sentiment Analysis

Prepared for

IS6423 AI for Business Applications

Prepared by

Ace Team

WEI He 56580119

CHEN Yanru 58767895

DENG Yao 58555026

LI Jiting 59226369

LU Wanshan 58876682

YE Zhencheng 59083618

Submission Date:

April 20, 2025

Table of Contents

1. Executive Summary

This report explores the relationship between social media sentiment and stock price trends, focusing on Tesla (TSLA). We aim to assess whether public sentiment can predict market trends by leveraging Twitter data and historical stock prices. The analysis employs sentiment analysis to quantify tweet emotions and LSTM-based time series modeling to forecast stock prices, incorporating sentiment scores and broader market indices (S&P 500 and Nasdaq) as key features.

The main steps and methods of our experiment are as follows:

- Data Processing: Extracting Tesla-related tweets and aligning them with daily stock data, addressing inconsistencies such as nontrading days.
- Sentiment Analysis: Generating sentiment scores using NLP techniques to capture market attitudes.
- Time Series Forecasting: Developing LSTM models to predict Tesla's closing prices, comparing performance against baseline methods (e.g., moving averages), and evaluating the incremental value of sentiment features.

There are several key challenges addressed in our project:

- Sentiment-Impact Quantification: While Tesla's stock is known to react sharply to public narratives (e.g., Elon Musk's tweets or media coverage), the extent to which sentiment systematically affects its price remains unclear. Isolating sentiment's predictive power from broader market trends (e.g., S&P 500) is critical for actionable insights.
- Data Integration: Social media data (e.g., tweets) and stock data operate on different timelines—tweets flow continuously, while markets are active only on trading days. Aligning these datasets while preserving sentiment's temporal relevance (e.g., lag effects) requires careful methodological design.

- Model Practicality: Many advanced forecasting models, such as LSTMs, are theoretically robust but face skepticism in real-world trading due to "black-box" opacity. Demonstrating that sentiment-augmented models outperform traditional baselines (e.g., moving averages) could enhance their adoption.

Our study seeks to provide insights into the interplay between social media discourse and financial markets, offering a framework for integrating unstructured data into quantitative trading strategies. Findings may inform investors and analysts of the potential of sentiment-driven indicators in high-volatility stocks like Tesla.

2. Business Problems

A complex interplay of quantitative data and qualitative factors, including public sentiment, inherently influences the stock market. For companies like Tesla—a high-profile, volatility-prone stock—social media discourse often amplifies market reactions, creating risks and opportunities for investors. However, traditional financial models primarily rely on structured data (e.g., historical prices, volume, and macroeconomic indicators), overlooking unstructured data like social media sentiment, which may contain early signals of price movements. This gap limits the ability to anticipate sudden market shifts driven by news, rumors, or collective investor emotions.

Therefore, we have identified the following specific business problems:

a) Can social media sentiment improve Tesla stock price predictions beyond traditional market data? Our project tests whether Twitter sentiment provides unique predictive signals not captured by stock indices (e.g., Nasdaq and S&P 500).

b) How can noisy, real-time tweet data be aligned effectively with discrete trading-day stock prices? Our project addresses challenges like nontrading days and sentiment lag effects (e.g., weekend tweets impacting Monday prices).

c) Does sentiment analysis offer practical value for high-volatility stocks like Tesla? Our project evaluates if sentiment-based models outperform simple baselines (e.g., moving averages) in real-world trading scenarios.

d) Can LSTM models reliably integrate sentiment and market data for actionable insights? Our project assesses whether complex ML models justify their "black-box" nature with measurable accuracy gains.

3. Data Processing

3.1 Dataset Description

Our project utilizes four datasets for research purposes, which are described as follows.

3.1.1 stock_tweets.csv

Source: kaggle.com

Description: 80793 tweets for the stock of 25 companies from 2021.9.30 to 2022.9.29, including date, tweet content, stock name, and company name.

	Date	Tweet	Stock Name	Company Name
0	2022-09-29 23:41:16+00:00	Mainstream media has done an amazing job at br...	TSLA	Tesla, Inc.
1	2022-09-29 23:24:43+00:00	Tesla delivery estimates are at around 364k fr...	TSLA	Tesla, Inc.
2	2022-09-29 23:18:08+00:00	3/ Even if I include 63.0M unvested RSUs as of...	TSLA	Tesla, Inc.
3	2022-09-29 22:40:07+00:00	@RealDanODowd @WholeMarsBlog @Tesla Hahaha why...	TSLA	Tesla, Inc.
4	2022-09-29 22:27:05+00:00	@RealDanODowd @Tesla Stop trying to kill kids,...	TSLA	Tesla, Inc.

Figure 1: head(5) of stock_tweets.csv

3.1.2 stock_yfinance_data.csv

Source: kaggle.com

Description: Together with stock_tweets.csv, the dataset includes stock data for 25 companies from 2021.9.30 to 2022.9.29, covering information such as date, open price, high price, low price, close price, adjusted close price, volume, and stock name.

	Date	Open	High	Low	Close	Adj Close	Volume	Stock Name
0	2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000	TSLA
1	2021-10-01	259.466675	260.260010	254.529999	258.406677	258.406677	51094200	TSLA
2	2021-10-04	265.500000	268.989990	258.706665	260.510010	260.510010	91449900	TSLA
3	2021-10-05	261.600006	265.769989	258.066681	260.196655	260.196655	55297800	TSLA
4	2021-10-06	258.733337	262.220001	257.739990	260.916656	260.916656	43898400	TSLA

Figure 2: head(5) of stock_yfinance_data.csv

3.1.3 Nasdaq_Index.csv

Source: nasdaq.com

Description: Nasdaq Index data from the official Nasdaq website has been filtered to include the index values for each day from 2021.9.30 to 2022.9.29.

	Date	Nasdaq_Index
0	2021-09-30	14689.62
1	2021-10-01	14791.87
2	2021-10-04	14472.12
3	2021-10-05	14674.15
4	2021-10-06	14766.75

Figure 3: head(5) ofNasdaq_Index.csv

3.1.4 sp500_index.csv

Source: kaggle.com

Description: S&P500 Index data obtained from the Kaggle, including Index data from 2021.9.30 to

2022.9.29.

	Date	S&P500
0	2021-09-30	4307.54
1	2021-10-01	4357.04
2	2021-10-04	4300.46
3	2021-10-05	4345.72
4	2021-10-06	4363.55

Figure 4: head(5) of sp500_index.csv

3.2 Data Processing and Visualization

In this part, the original datasets are processed with data cleaning to check for missing values or duplicated data and ensure that the time alignment of all data is synchronized from 2021.9.30 to 2022.9.29.

We choose data from Tesla, Inc. to be our main analysis target, as the tweet count of TSLA accounts for the most significant part of tweets data (46.2%), as shown in **Figure 5**. The dataset contains 37422 tweet comments on TSLA stock, providing sufficient data for sentiment analysis.

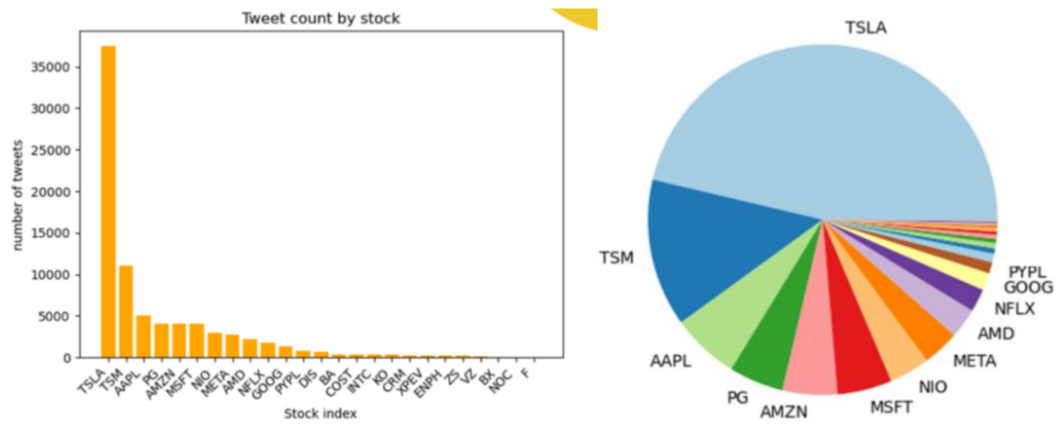


Figure 5: Tweet count by stock name

After filtering out Tesla's tweets and stock datasets, the tweets and stock datasets are checked for duplicated and missing values. Figure 6 shows the daily tweet counts of Tesla for one year, with an average of 102 tweets.

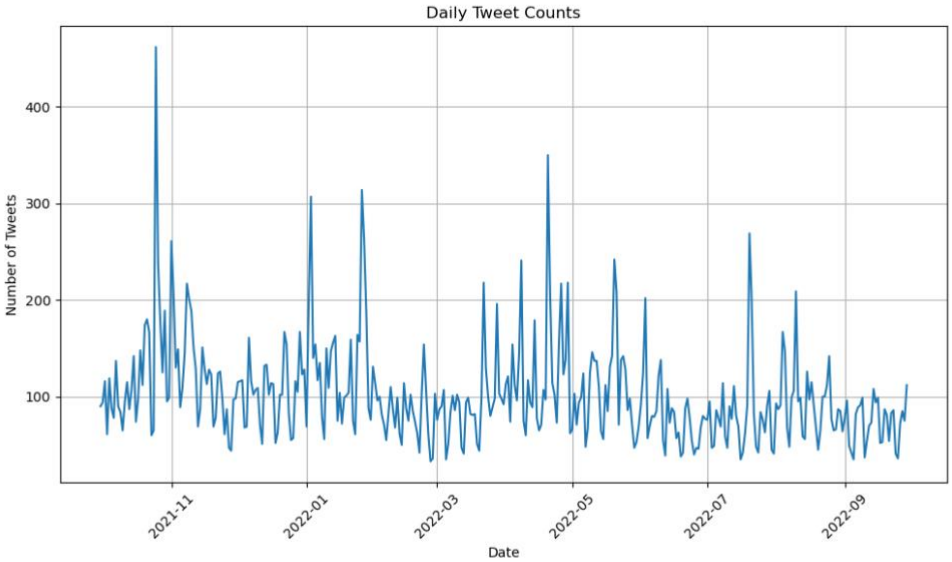


Figure 6: Daily tweet counts of Tesla

Figure 7 visualizes the four types of Tesla stock prices from 2021.9 to 2022.9. In this project, we use close price as one feature for LSTM model training and our prediction target.

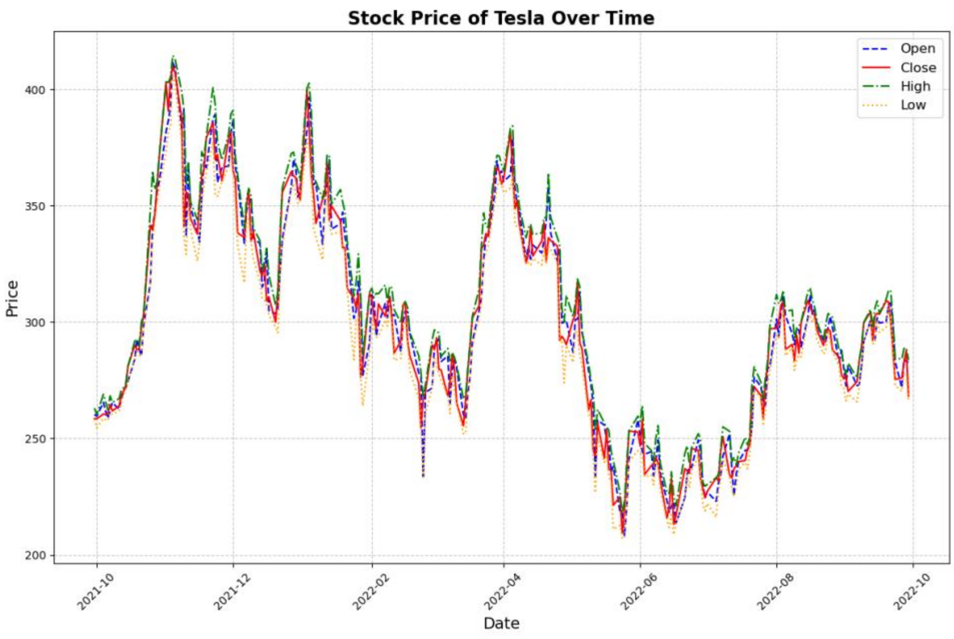


Figure 7: Stock price of Tesla (Open, Close, High and Low)

Figure 8 and **Figure 8** show the historical trend of two stock indexes, the Nasdaq Index and the S&P500 Index.

The Nasdaq Index is a stock market index that includes all companies listed on the Nasdaq stock exchange. At the same time, the S&P 500 Index is a market-capitalization-weighted index of the 500 leading publicly traded companies in the United States.

We apply these two indexes as our LSTM time series model feature to compare the model performance between sentiment score and index.

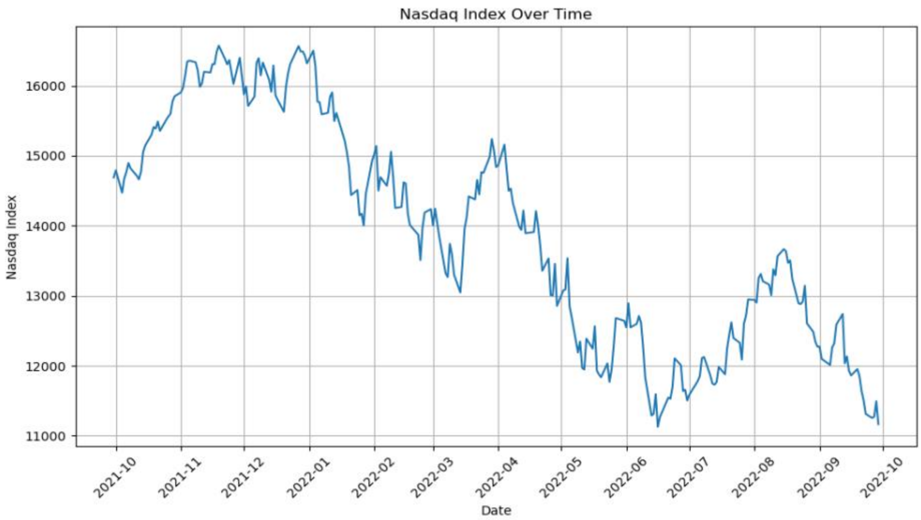


Figure 8: Nasdaq index 2021.9-2022.9

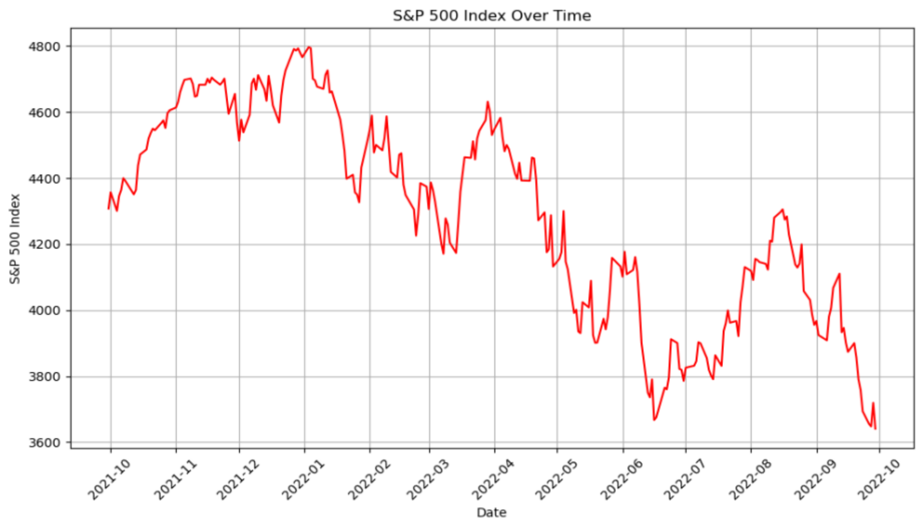


Figure 9: S&P500 index 2021.9-2022.9

These two indexes are merged into one dataset, as shown in **Figure 10**, and saved as index.csv for further experimentation.

	Date	S&P500	Nasdaq_Index
0	2021-09-30	4307.54	14689.62
1	2021-10-01	4357.04	14791.87
2	2021-10-04	4300.46	14472.12
3	2021-10-05	4345.72	14674.15
4	2021-10-06	4363.55	14766.75
...	...	...	...
247	2022-09-23	3693.23	11311.24
248	2022-09-26	3655.04	11254.11
249	2022-09-27	3647.29	11271.75
250	2022-09-28	3719.04	11493.83
251	2022-09-29	3640.47	11164.78

252 rows × 3 columns

Figure 10: Merged index dataset: Index.csv

4. Sentiment Analysis

4.1 Label the Tweets

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a tool in NLTK for sentiment analysis based on the VADER dictionary. Since our dataset has no sentiment classification labels, we use SentimentIntensityAnalyzer to label the tweets. VADER combines lexical sentiment polarity and rules, and is specifically optimized for social media texts. It considers the word's sentiment and context, tone, and emotions.

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

4.2 Processing steps

1. Creating sentiment analysis result columns:

2. Create sent_df as a copy of tsla_data.

3. Add four columns to sent_df to store the sentiment scores calculated later:

sentiment_score (comprehensive sentiment score)

Positive (positive sentiment score)

Neutral (neutral sentiment score)

Negative (negative sentiment score)

	Date	Tweet	sentiment_score	Positive	Neutral	Negative
0	2021-09-30	In other words, AMD has been giving Tesla pref...				
1	2021-09-30	Get ready for a $TSLA _ _ _ _ _ _ Q3 delivery...				
2	2021-09-30	Hold. On. Tight. $TSLA				
3	2021-09-30	I agree with @freshjiva that $TSLA 's EV busin...				
4	2021-09-30	Playing in the dirt and #chasingsunsets\n@tesl...				

4. Use SentimentIntensityAnalyzer to calculate the sentiment score of text data.

Each row contains the original tweet, the corresponding sentiment score, and positive, neutral, and negative values.

	Date	Tweet	sentiment_score	Positive	Neutral	Negative
0	2021-09-30	In other words, AMD has been giving Tesla pref...	0.659	0.166	0.834	0.0
1	2021-09-30	Get ready for a $TSLA _ _ _ _ _ _ Q3 delivery...	0.4215	0.257	0.743	0.0
2	2021-09-30	Hold. On. Tight. $TSLA	0.0	0.0	1.0	0.0
3	2021-09-30	I agree with @freshjiva that $TSLA 's EV busin...	0.5719	0.175	0.747	0.078
4	2021-09-30	Playing in the dirt and #chasingsunsets\n@tesl...	-0.1531	0.148	0.656	0.197

4.3 Aggregation by time

4.3.1 Daily average

Use the groupby() function to group by 'Date', calculate the daily average sentiment score (sentiment_score), and store the result in twitter_df. The daily average sentiment score changes are as follows:

```
:  Date
   2021-09-30    0.231552
   2021-10-01    0.233704
   2021-10-02     0.27194
   2021-10-03     0.27157
   2021-10-04    0.135388
   2021-10-05    0.069445
   2021-10-06     0.19994
   2021-10-07    0.192548
   2021-10-08    0.220011
   2021-10-09    0.294931
   2021-10-10    0.244551
```

4.3.2 Weekly Average

Use the groupby() function and pd.Grouper to group by week ('W', which defaults to ending on Sunday) and calculate the average sentiment score for each week. The average sentiment score for each week is as follows:

```
Date
2021-10-03    0.251852
2021-10-10    0.188605
2021-10-17    0.166151
2021-10-24    0.184919
2021-10-31    0.196194
Freq: W-SUN, Name: sentiment_score, dtype: object
```

4.3.3 Monthly average

Use the groupby() function and pd.Grouper to group by the last day of each month ('ME') and calculate the average sentiment score for each month. The monthly average sentiment score is as follows:

```
Date
2021-09-30    0.231552
2021-10-31    0.191146
2021-11-30    0.191373
2021-12-31    0.188472
2022-01-31     0.14224
Freq: ME, Name: sentiment_score, dtype: object
```

4.4 Sentiment classification

Text classification:

Define a function to classify a sentiment_score as "positive", "negative", or "neutral" based on its value:

Greater than or equal to 0.5: positive

Less than or equal to -0.5: negative

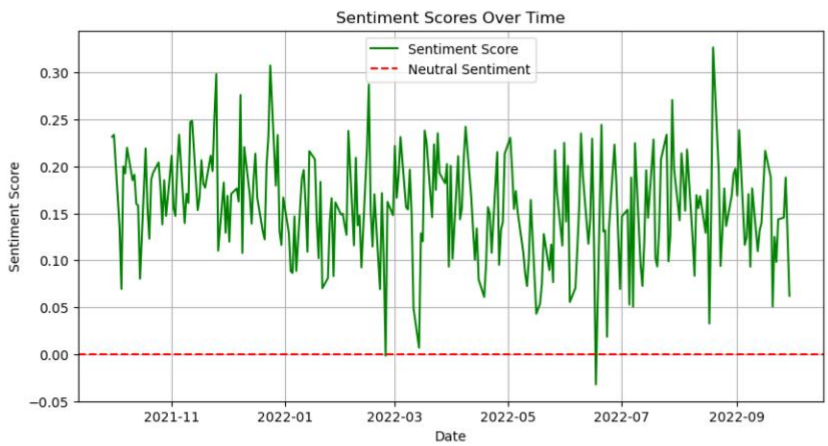
In between: neutral

	Date	Tweet	sentiment_score	sentiment
0	2021-09-30	In other words, AMD has been giving Tesla pref...	0.659	positive
1	2021-09-30	Get ready for a $TSLA ______ Q3 delivery...	0.4215	neutral
2	2021-09-30	Hold. On. Tight. $TSLA	0.0	neutral
3	2021-09-30	I agree with @freshjiva that $TSLA 's EV busin...	0.5719	positive
4	2021-09-30	Playing in the dirt and #chasingsunsets\n@tesl...	-0.1531	neutral

4.5 Visualization

In order to understand the temporal dynamics and overall distribution characteristics of emotional information extracted from tweets, we conducted a visual analysis of the calculated emotional scores.

4.5.1 Sentiment Scores Over Time



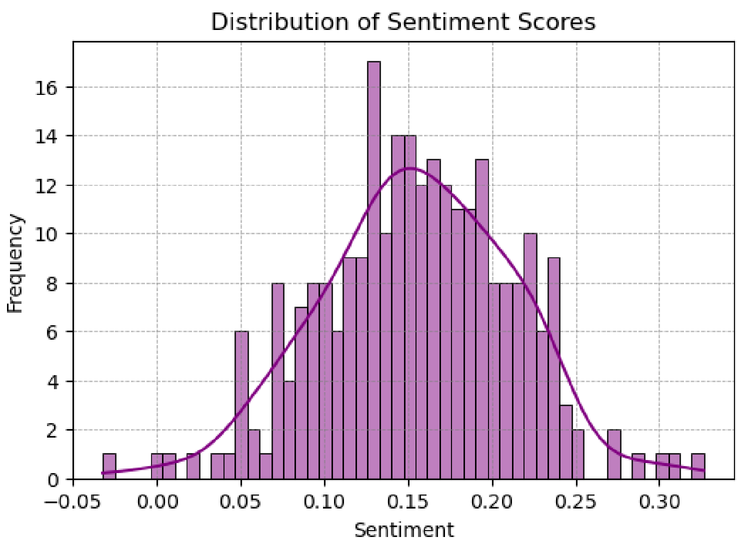
Overall trend: During the observation period, the sentiment score was mainly above the neutral baseline (0), indicating that the overall sentiment of tweets about Tesla during this period tended to be weakly positive.

Fluctuation range: The score mainly fluctuates within the range of 0.05 to 0.25, with occasional peaks approaching 0.30 and troughs approaching or briefly falling below 0 (such as around June and July 2022).

Volatility: Emotional scores exhibit significant high-frequency fluctuations, with no apparent long-term upward or downward trend observed. This reflects the rapid changes in public sentiment or discussion hotspots over time.

Key points: Several significant peaks (such as late 2021 and late February 2022) and valleys (such as early 2022 and June 2022) can be seen in the graph, which may be associated with specific market events or news.

4.5.2 Distribution of Sentiment Scores



Distribution shape: The distribution of emotional scores shows an unimodal shape, roughly approaching a normal (bell-shaped) distribution, with the peak in the positive region. The distribution may be slightly skewed to the right (positive bias).

Central trend: The peak of the distribution (i.e., the most common daily sentiment score) is between 0.13 and 0.15.

Data concentration: Most sentiment scores are concentrated in the range of 0.05 to 0.25, with a low frequency of extreme values (very close to 0 or significantly higher than 0.25). This is consistent with the observation results of the time series chart.

4.6 Data Processing and Feature Engineering

After generating baseline classifications for our dataset, we began our sentiment analysis based on the methods in the tutorial. In order to extract adequate sentiment information from tweets, we first systematically preprocessed the original tweet texts. The process mainly includes the following steps:

4.6.1 Text cleaning

Tokenization: Use the nltk.word_tokenize library to split each tweet into tokens.

Lowercase conversion: Convert all words to lowercase to eliminate uppercase and lowercase differences.

Punctuation removal: Remove all punctuation.

Non-letter word filtering: Remove all tokens not composed of pure letters (for example, numbers, mixed symbols, etc.).

Stopword removal: Use the standard English stopword list provided by nltk to remove common words that do not contribute much to sentiment analysis (such as "the", "a", "is", etc.).

Result storage: Store the processed word list in review_lines for subsequent word embedding training. This processing involves a total of 37,422 tweets.

4.6.2 Word Embedding - Word2Vec

In order to convert text data into numerical representations that can be understood by machine learning models and capture the semantic relationship between words, we use the Word2Vec model for word embedding training:

1. Model selection: Use the Word2Vec model in the gensim library.

2. Training data: Use the tweet word list cleaned in the previous step as the training corpus.

3. Parameter settings:

- vector_size (embedding dimension EMBEDDING_DIM): Set to 100, that is, each word is represented as a 100-dimensional vector.
- Window (context window size): Set to 5, which means that the five words before and after the current word are considered when predicting the current word.
- min_count (minimum word frequency): Set to 1, which means that all words that have appeared in the corpus will be included in the vocabulary and generate word vectors.

4. Vocabulary construction: The trained vocabulary size is 30,515 independent words.

5. Model saving and loading: The trained Word2Vec model (word vector) is saved as text format and then loaded into a dictionary named embeddings_index for subsequent use in the LSTM model.

4.7 LSTM Model Data Preparation

Before training the LSTM sentiment classification model, it is necessary to convert the text data and sentiment labels into a format suitable for the model input:

a. Text serialization:

- Build a vocabulary index (vocab/word_index) that maps each word to a unique integer index (the index starts from 1, and 0 is reserved for padding).

• Convert each processed tweet (word list) into a corresponding integer index sequence.

b. Padding: Due to the varying lengths of tweets and the fact that LSTM models typically require fixed length inputs, integer sequences are filled in.

c. Tag preparation: Convert the sentiment tags corresponding to the tweet (in numerical form, such as 0 representing negative/neutral and 1 representing positive) into Tensor.

d. Dataset partitioning: Randomly divide the dataset into an 80% training set and 20% validation/test set to obtain *X_train_mad, y_train, X_test_pad, and y_test*. (The partitioned training set contains 29938 samples, and the validation set contains 7484 samples.)

4.8 LSTM Model Construction

We built an LSTM model for sentiment binary classification of tweets using PyTorch.

- Embedding Layer:
  - The dimension is (numw_words, EMBEDDIND_SIM), which is (30515+1, 100).
  - Key: Use nn. Embedding. from _pretained loads the previously trained Word2Vec word vector (embedding_matrix) and sets freeze=True not to train the embedding layer, utilizing pre-trained semantic information. Padding_idx=0 ensures that the padding position does not affect the calculation.
- LSTM layer:
  - input_size: 100 (consistent with word embedding dimension).
  - hidden_size: 32, which is the hidden state dimension of the LSTM unit.
  - num_layers: 1. Indicates the use of a single-layer LSTM.
  - batch_first=True， The dimension order of the input data is (batch, seq_1en, feature).

- Output Layer:
    - Obtain the output of the last time step (lstm_out [:, -1,:]) from the LSTM layer, with dimensions of (batch_2, hidden_2)=(batch_2, 32).
    - Connect a fully connected layer (nn. Linear) to map the 32 dimensional hidden state to the 1-dimensional output (logit).
    - Use nn. Sigmoid activation function converts the output logit into a probability value between 0 and 1, representing the probability of predicting a positive emotion.
- Model training settings:
    - Loss Function: nn.BCELoss (Binary Cross Entropy Loss) is suitable for binary classification problems.
    - Optimizer: optimm.Adam, with a learning rate set to 0.001.
    - Batch size: 128.
    - Training epochs: 15.

4.9 Model Training Process

The model was trained on the training set for 15 epochs and evaluated on the validation set after each epoch. The key indicators during the training process change as follows:

- Loss:

  The Train Loss steadily decreased from an initial value of approximately 0.6354 to a final value of approximately 0.4776.

  The Validation Loss decreased from approximately 0.6046 and reached its lowest point around the 10th epoch (approximately 0.4995), fluctuated slightly or stabilized, and finally reached

0.4998. This indicates that the model's performance tends to saturate the validation set, and continuing training may result in slight overfitting.

- Accuracy:

The training accuracy improved from approximately 64.37% to a final level of approximately 77.73%.

The Validation Accuracy improved from approximately 67.04%, reaching its peak at around 76.60% in the 12th and 14th epochs and ultimately reaching 76.31% in the 15th epoch.

4.10 Final Performance Evaluation of the Model

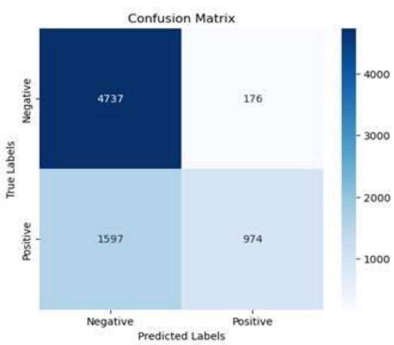
After training, the final model was evaluated on an independent test set (i.e., the previous validation set X_test_cd, y_test):

- Test Loss: 0.4996
- Test Accuracy: 76.31%

This indicates that our trained sentiment analysis model based on LSTM and Word2Vec can distinguish the sentiment tendency of tweets with an accuracy of approximately 76.31% (based on the binary classification setting in the project).

4.11 Detailed performance evaluation of the model

1. Confusion Matrix:



The specific values of the confusion matrix are as follows:

- True Negative (TN): 4737 (correctly predicted as negative sample)
- False Negative (FN): 1597 (actually positive but incorrectly predicted as negative sample)
- False Positive (FP): 176 (actually negative but incorrectly predicted as positive sample)
- True Positive - TP: 974 (correctly predicted as positive sample)

2. Classification Report:

Classification Report:

	precision	recall	f1-score	support
Negative	0.75	0.96	0.84	4913
Positive	0.85	0.38	0.52	2571
accuracy			0.76	7484
macro avg	0.80	0.67	0.68	7484
weighted avg	0.78	0.76	0.73	7484

This report provides the precision, recall, F1 score, macro avg, and weighted avg for each category ("Negative", "Positive"). The overall accuracy is 0.76, consistent with the validation accuracy observed during training.

4.12 Analysis and Discussion

Model effectiveness: 76.31% accuracy indicates that the model can distinguish emotions and learn patterns related to emotions from tweet texts. This accuracy level is above average in social media sentiment analysis tasks, and the specific effectiveness needs to be compared with baseline models (such as simple TF-IDF+logistic regression) or relevant research.

Word2Vec embedding: Using pre-trained Word2Vec embeddings as features and freezing their weights can help leverage the general semantic information learned from large-scale corpora and potentially accelerate model convergence. Exploratory checks on most_similar and analogies in the code revealed some reasonable results (such as synonyms for 'car') and some results that may be specific to the dataset or slightly noisy (such as synonyms and analogies for 'tesla'), suggesting that there may be room for improvement in word vector quality (for example, adjusting Word2Vec parameters or training with larger, more relevant corpora).

Project contribution: This sentiment analysis model is an important component of our Tesla stock price prediction project. Its output (such as sentiment scores or categories predicted for future tweets) can be used as a new feature, along with historical stock prices, trading volumes, and other data, to be input into the final stock price prediction model to test whether social media sentiment can improve the accuracy of stock price predictions.

Overall evaluation: Although accuracy reaches 76%, model performance is seriously imbalanced across different categories. It tends to predict tweets as negative and has a weaker ability to recognize positive emotional signals. This bias may be caused by the imbalance of training data or the characteristics of the model itself (such as single-layer LSTM may not be sufficient to capture complex positive expressions).

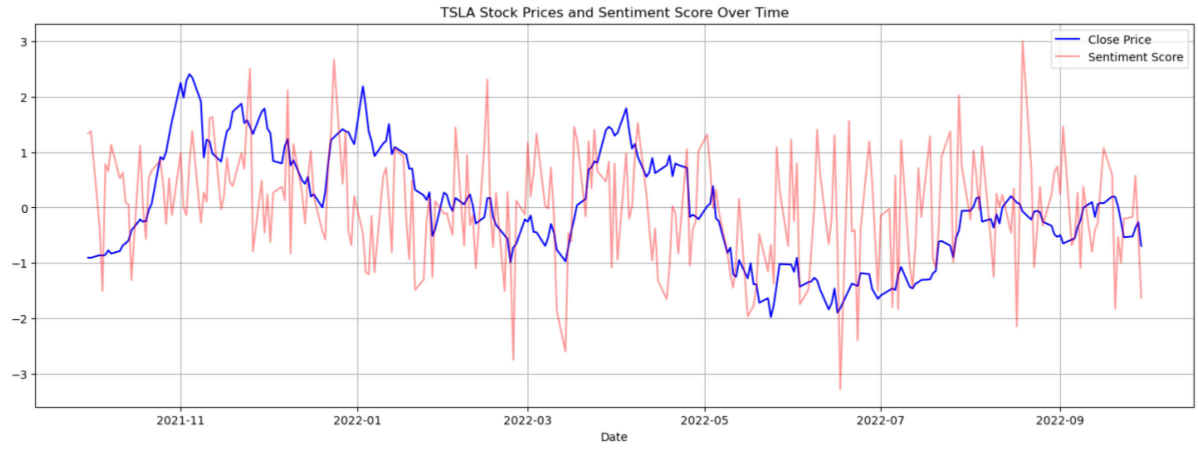
This accuracy level is above average in social media sentiment analysis tasks. However, the labels generated by the pretrained model, Vader, are still more accurate. Therefore, we choose to use the labels generated by the Vader model for the following time series analysis.

5. Time Series

5.1 Exploratory Analysis

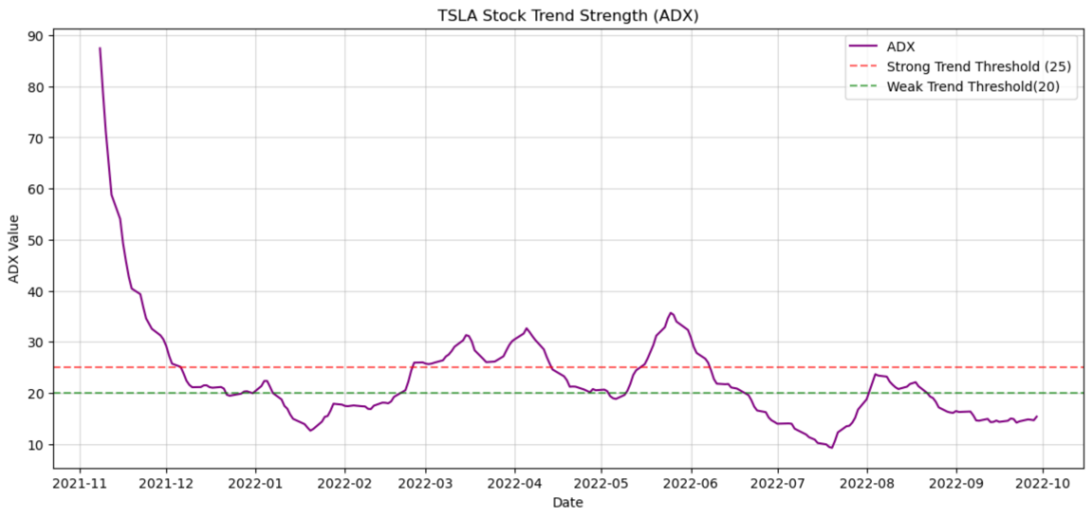
5.1.1 Sentiment_score vs. Close_price

From the previous section, we obtained sentiment score values. Observing the two trends in the overlay chart below shows that the sentiment score and closing price follow roughly the same pattern.



5.1.2 Trend Strength

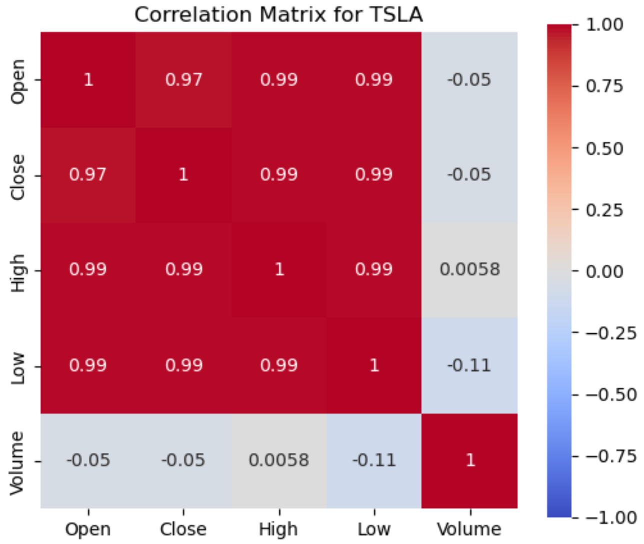
The following chart shows the trend intensity trend of Tesla stock within one year of trading. Approximately 40% of the time is in a strong trend state, approximately 30% is in a weak trend state, and the rest is in a trend transition zone. In the figure below, it can be seen that there is a strong initial trend, with an ADX close to 90, indicating the existence of a strong single trend, which may be in a period of intense market volatility. At the same time, multiple moderate-intensity trends formed, but the duration was not long and had a certain periodicity.



5.2 Feature Engineering

5.2.1 Stock Prices Features

The original data includes closing, opening, highest, and lowest prices. From the figure below, we can see the price trends of four features. Among these features, the closing price incorporates all information from the entire trading day, including market sentiment and trading activities. Unlike other intraday prices, closing prices typically have lower volatility, less noise, and more substantial settlement value. Therefore, the closing price is selected as the label in the stock price prediction model.



From the Correlation Matrix, we can see that these for features are highly correlated.

5.2.2 Sentiment Features

To improve the model's processing capability, the following features are introduced:

Feature	What it is	How it's calculated	How it helps
Sentiment_score	A number that tells us how people feel about Tesla based on Tweets.	Generated in the previous sentiment analysis section.	• It gives the model an idea of the overall mood, which can influence stock prices. • If a lot of people are feeling very positive or very negative, it might cause prices to go up or down.
Lagged_sentiment	The sentiment score from previous days.	Generated in the previous sentiment analysis section.	• It adds historical mood data so the model can recognize delayed effects of investor sentiment. • The market might react to sentiment with a delay.
Sentiment_volatility	Measures how much the sentiment score is jumping around.	The standard deviation of the sentiment scores over the past 3 days.	• It tells the model if the market mood is calm or chaotic, which can affect price movements. • If sentiment is very unstable (going up and down a lot), it might mean uncertainty or potential big.
Future_returns_lag	Measures how much the stock price will changes in the future.	Price(t) is the closing price today. n is days into the future.	• It shows the trend of the stock price in the future. • It helps train the model by showing what actually happened after each point in time.

$$Future\ Return(t, t+n) = \frac{Price(t+n) - Price(t-n)}{Price(t)} \times 100\%$$

The stock market only opens on weekdays, while sentiment data is continuously produced daily, and inconsistencies must be addressed. According to the 'Monday Effect', emotions and information accumulated during nontrading hours will be re-released when the market reopens. The impact of emotional changes on weekends is usually reflected on the next trading day (usually the following Monday). Therefore, emotional data is processed as follows:

a. Weekend (Saturday and Sunday) data is integrated and averaged to create a "weekend sentiment indicator".

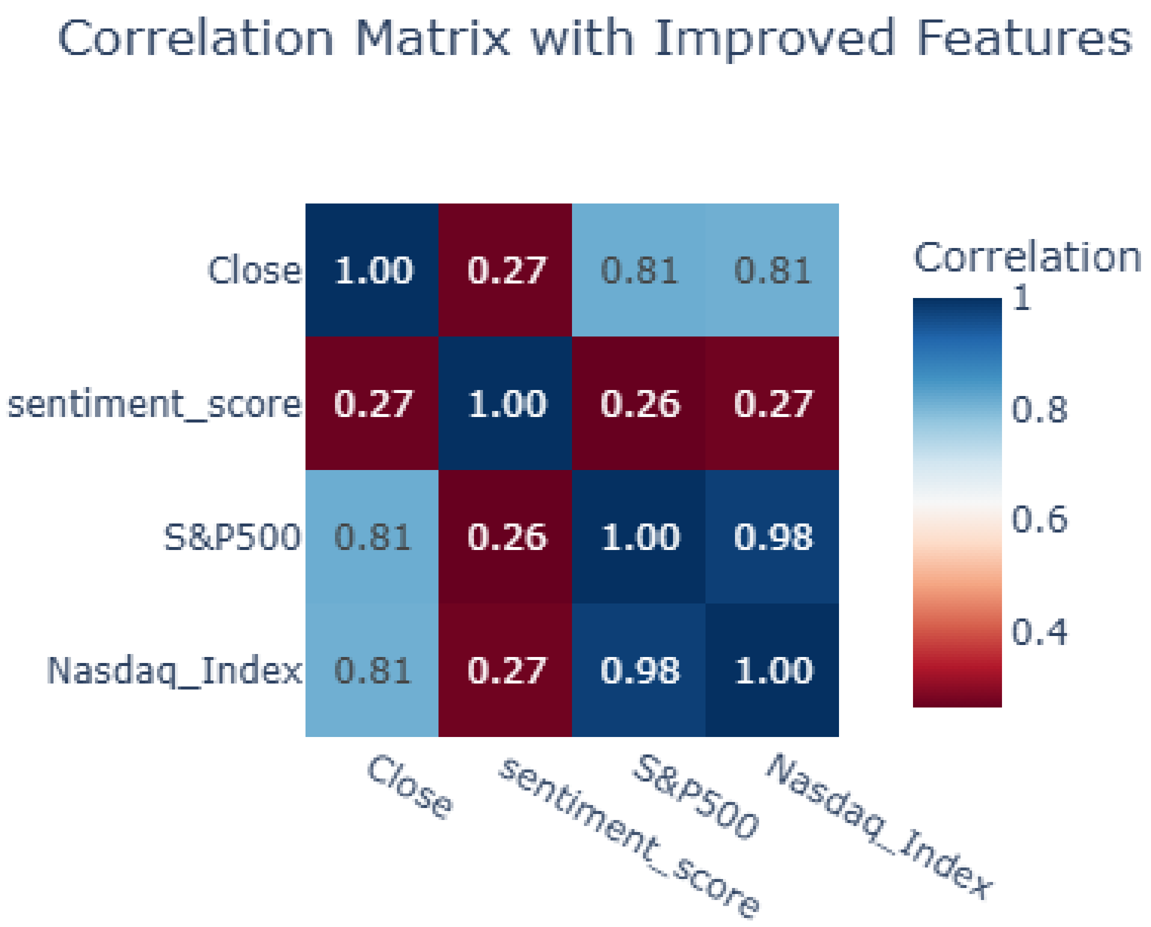
b. The calculated weekend sentiment impact is mapped to the next trading day.

5.2.3 Index Features

Feature	What it is	Characteristics	How it helps
S&P 500 Index	A stock index made up of the 500 largest publicly traded companies in the U.S.	• Represents about 80% of the total U.S. stock market value.	• It reflects the overall health and direction of the U.S. economy. • TSLA, being a major U.S. company, often moves in relation to this index.
Nasdaq Index	A stock index focusing on technology and innovation-driven companies.	• Contains many tech-heavy and high-growth companies. • Known for higher volatility and stronger growth potential.	• It is sensitive to changes in the tech industry and innovation trends. • TSLA is often seen as a tech-centric company, so its performance is more closely tied to the Nasdaq.

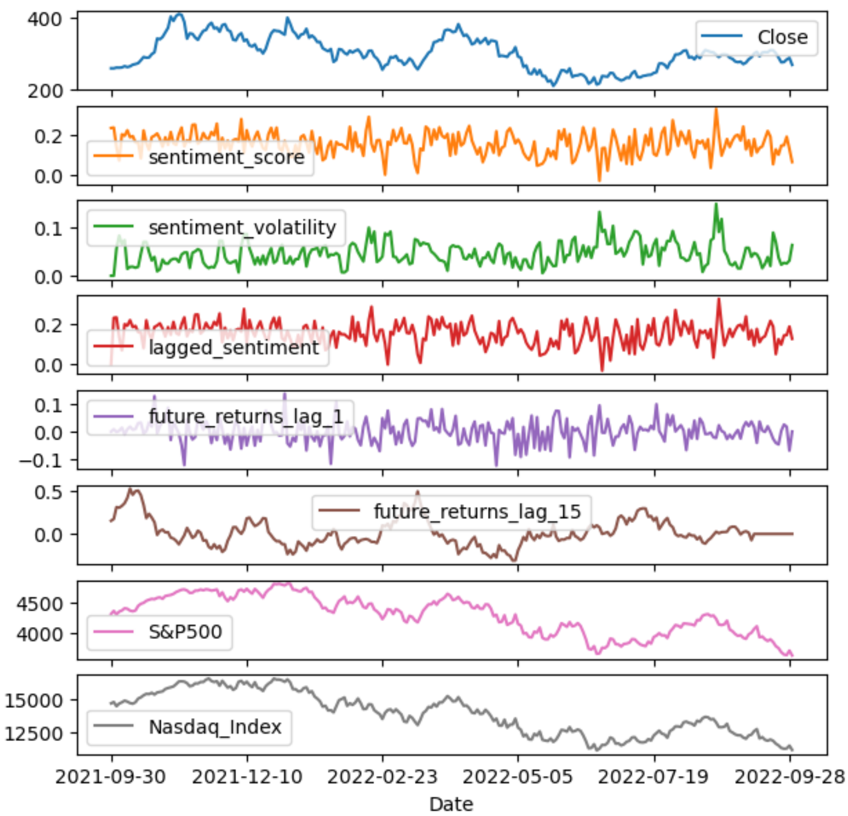
5.2.4 Correlation Matrix

As shown in the feature correlation coefficient chart below, the correlation coefficient between Tesla's closing price and the S&P500 and Nasdaq Index is 0.81, indicating that a wide range of market factors strongly influences Tesla's stock price, and about 81% of its changes can be explained by the overall market. The correlation coefficient between the closing price and sentiment score is 0.27, indicating that social media sentiment, one of the factors affecting Tesla's stock price, has a weak positive correlation. At the same time, the correlation coefficient between the sentiment index and the other two indices is around 0.26, indicating that sentiment data may simultaneously capture information independent of market trends.



5.2.5 Feature Visualization

Visualize all features to check their elemental distributions, as shown in the figure below.



5.3 Experiment Design

Model	Input Features	Purpose
Baseline: Moving Average	• TSLA close prices	Establish a **benchmark** to evaluate the effectiveness of LSTM-based models.
LSTM_1	• TSLA close prices	Capture sequential dependencies in TSLA price trends using deep learning.
LSTM_2	• TSLA close price • **Future return lag**	Capture sequential dependencies and future return patterns in TSLA price trends
LSTM_3	• TSLA close prices • **Sentiment scores**	Examine the impact of public sentiment on TSLA stock price movements.
LSTM_4	• TSLA close prices • **Sentiment scores** • **Sentiment volatility**	Examine the impact of public sentiment and its volatility on TSLA stock price movements.
LSTM_5	• TSLA close prices • **S&P 500 Index** • **Nasdaq Composite Index**	Understand how broader market trends influence TSLA performance.
LSTM_6	• TSLA close prices • **Sentiment scores** • **Sentiment volatility** • **S&P 500 Index** • **Nasdaq Composite Index**	Build a comprehensive model that leverages both market data and sentiment.

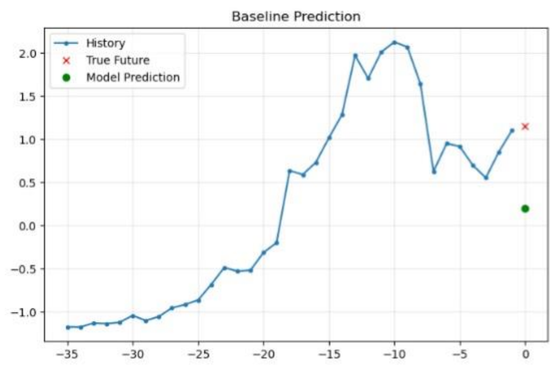
Using the Moving Average Baseline Model for simple predictions can evaluate the predictability of the trial market and provide performance limits, ensuring that complex models can bring substantial improvements. If LSTM cannot significantly outperform the moving average limit model, it may support weak market efficiency.

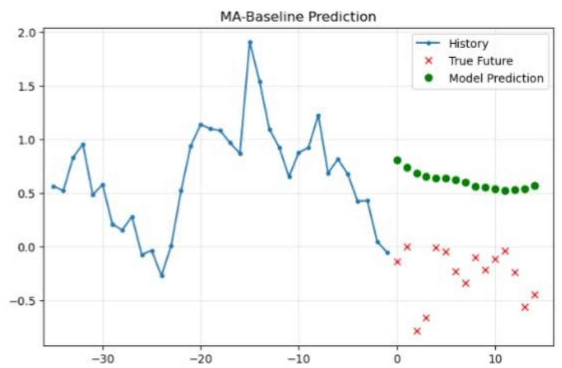
Unlike statistical models such as ARIMA, LSTM can automatically capture nonlinear relationships. Compared to simple RNN, LSTM can better capture long-term effects and handle the temporal relationships between multiple variables. It also has strong resistance to noise and outliers.

5.4 Model Configuration

5.4.1 Moving Average (Baseline)

We use the average of the last 35 observations of Tesla close prices to predict the next day close price(left) and the next 15 days' close prices(right).





5.4.2 LSTM Configurations

Build a multi-layer unidirectional LSTM neural network (two LSTM layers + fully connected layer) with parameters configured as shown in the table below.

	Input_size	Hidden_size	Batch_first	Num_layers	Dropout	Output
LSTM1	x_train.shape	64	True	1	0	-
LSTM2	64	32	True	2	0.4	-

Dense	32	-	-	-	-	Future step

We implement an early stopping strategy to stop training before the model overfits the training data, preserving the model with the best loss on the validation set.

5.5 Model Evaluations

5.5.1 Model Evaluation

For stock price prediction, since stock prices are easily affected by market conditions and can show outliers (market crashes, sudden stock surges, etc., which produce extreme data), using only MSE or RMSE makes it challenging to capture overall trends and can lead to extreme errors.

This paper uses a weighted sum of MAE and RMSE as the model's loss function, which is important in financial forecasting. This combination helps the model maintain robustness while remaining sensitive to price movements, making it suitable for handling the complex nature of stock price fluctuations.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum(\mathrm{true\_value} - \mathrm{predict\_value})^2}$$

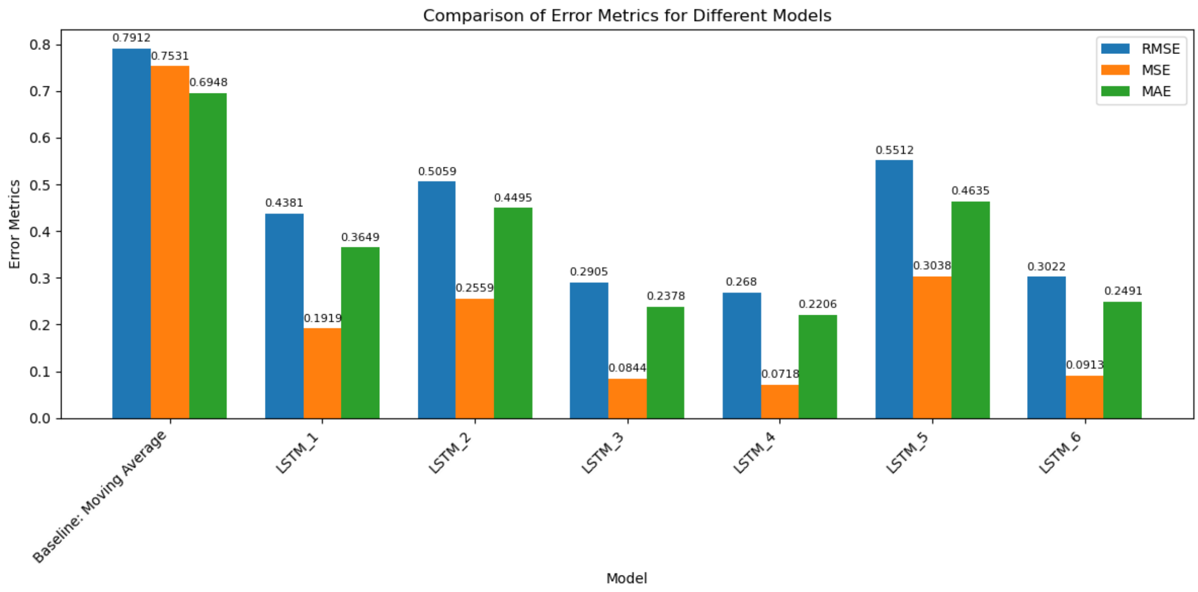
$$\mathrm{MAE} = \frac{1}{n}\sum|\mathrm{true\_value} - \mathrm{predict\_value}|$$

$$\mathrm{Loss} = \alpha \cdot \mathrm{MAE} + \beta \cdot \mathrm{RMSE}$$

The equations above show the calculation method of the loss function used in this paper, where α and β are weighting parameters that adapt to different prediction emphases.

Model	Features considered	RMSE	MSE	MAE
Baseline: Moving Average	• TSLA close prices	0.7912	0.7531	0.6948
LSTM_1	• TSLA close prices	0.4381	0.1919	0.3649
LSTM_2	• TSLA close price • **Future return lag**	0.5059	0.2559	0.4495
LSTM_3	• TSLA close prices • **Sentiment scores**	0.2905	0.0844	0.2378
LSTM_4	• TSLA close prices • **Sentiment scores** • **Sentiment volatility**	0.2680	0.0718	0.2206
LSTM_5	• TSLA close prices • **S&P 500 Index** • **Nasdaq Composite Index**	0.5512	0.3038	0.4635
LSTM_6	• TSLA close prices • **Sentiment scores** • **Sentiment volatility** • **S&P 500 Index** • **Nasdaq Composite Index**	0.3022	0.0913	0.2491

5.5.2 Model Comparison



LSTM_1 only considers close price. Compared with baseline, LSTM_1 significantly reduces all error metrics. This indicates that LSTM can effectively capture the nonlinear patterns and dependencies in time series data, and can improve the prediction accuracy even when using only a single feature.

LSTM_2 combines close price and the lagged values of future returns. However, the error metrics slightly increase. This may be because the introduction of the lagged values may introduce noise.

LSTM_3 combines close price and sentiment score. The error metrics are significantly reduced, indicating that public sentiment has a significant impact on Tesla's stock price, and the addition of the sentiment score helps the model better capture the reasons for the stock price fluctuations.

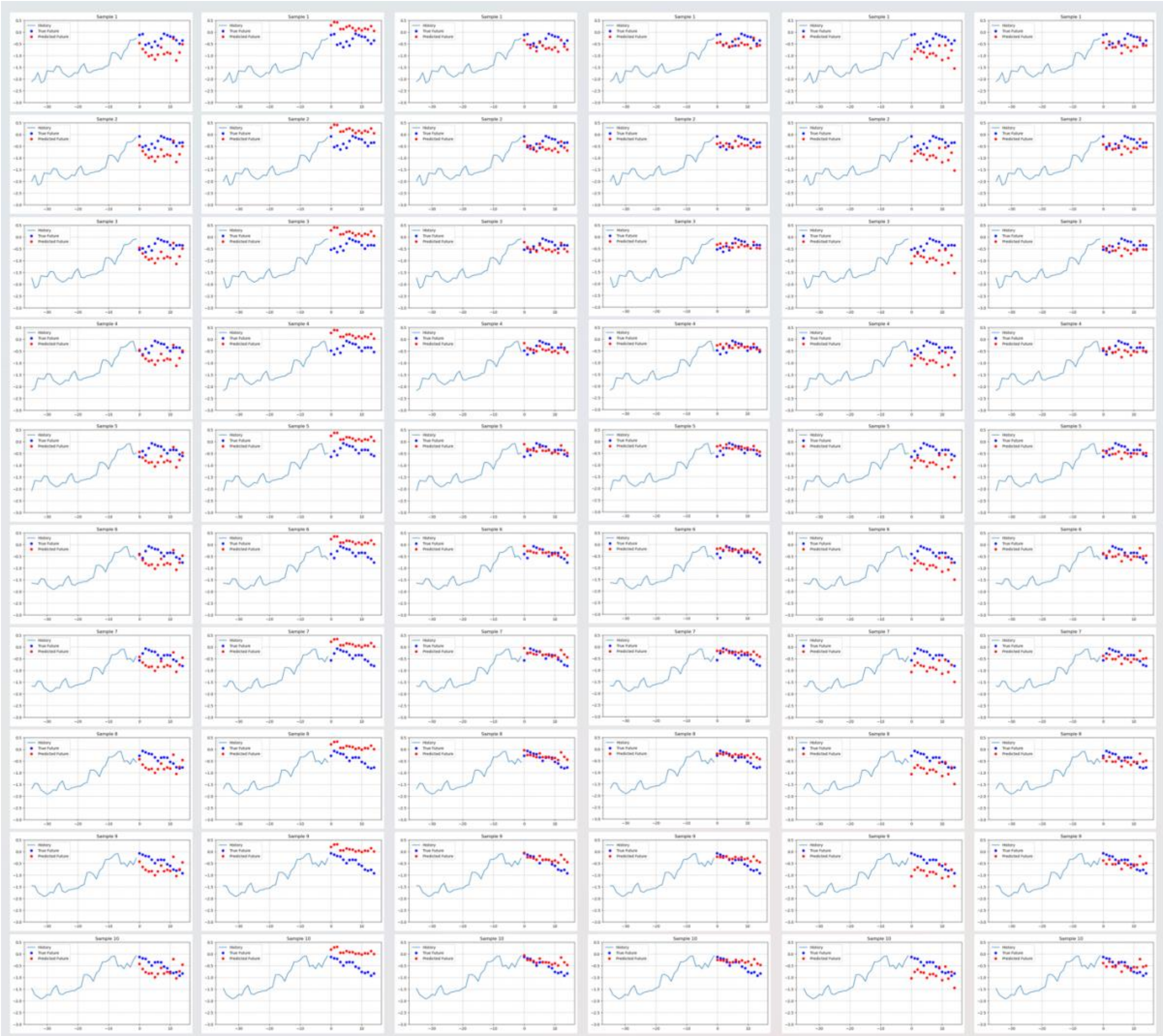
LSTM_4 further adds the sentiment volatility on the basis of LSTM_3. The error metrics of LSTM_4 are further optimized and reach the lowest level, indicating that not only is the sentiment score itself important, but the sentiment volatility can also reflect the changing speed of market sentiment, thus providing a more comprehensive explanation for the stock price fluctuations.

LSTM_5 uses close price and attempts to capture the impact of the overall market trend on Tesla by adding the S&P 500 index and the NASDAQ Composite index. However, the error metrics are significantly higher than those of LSTM_3 and LSTM_4, indicating that these market indices fail to significantly improve the model performance.

LSTM_6 combines the market indices (S&P 500 and NASDAQ Composite index) and sentiment-related data (sentiment score and sentiment volatility), and the error metrics are slightly higher than those of LSTM_4. Nevertheless, it is still better than most other models, indicating that the importance of sentiment-related data is much greater than that of market indices.

5.5.3 Sample Predictions

From left to right columns are the sample predictions of LSTM_1 to LSTM_6 models. Each column contains samples with indices ranged 1 to 10 from validation dataset.



LSTM_4 performs the best, indicating that sentiment scores and sentiment volatility are key factors in predicting Tesla's stock price. LSTM_3 and LSTM_6 follow closely. Both utilize sentiment-related data, but LSTM_6 experiences a slight drop in performance due to the introduction of market indices. However, baseline and LSTM_5 perform poorly, with the former being overly simplistic and the latter failing to fully leverage the advantages of sentiment data.

6. Implications

6.1 Theoretical Implications

Our results reinforced that LSTM can capture complex temporal dependencies in stock price data. It is better than baseline models like moving averages.

Feature engineering matters, but not all features are useful. In our case, sentimental factors are more impactful than Market indices like the S&P 500 and Nasdaq.

6.2 Practical Implications

The results suggest that using different data types (like stock prices, indices, and sentiment) could help prediction, but real-world use needs careful testing. For highly volatile stocks like Tesla, sentimental factors are crucial for predicting stock prices.

We should choose the right features. In our case, the role of market indices is relatively small, and more refined feature engineering may be needed to reflect their value. Irrelevant data may hurt model performance. Also, we should consider the explainability of the model; although LSTMs work well, they are hard to interpret.

6.3 Limitations

Many factors would impact the stock price; other than features considered in this report, external factors like Elon Musk's tweets, regulatory changes, and international conflicts also heavily influence.

LSTM models perform well in dealing with time series data, but once trained, they may not be able to adapt to future situations, especially when a sudden event happens (like when Donald Trump won). Thus, retraining is required.

There may be sentiment data bias. Our data is from Yahoo Finance, where specific age groups or regions (maybe old American people) comment and the sentiment analysis may reflect their views more.

7. Conclusion

This report explores the potential of integrating social media sentiment with traditional market data to predict Tesla stock prices. After data selection, EDA, processing, and calculation of sentiment scores, we performed LSTM models with close price, sentimental factors, S&P500, and Nasdaq index. Based on the particular dataset we used, we found out the LSTM models outperform the baseline model significantly, as RMSE, MSE, and MAE values showed. The model performed worse when adding sentimental factors or market indices, indicating that adding more features does not always lead to better performance. However, it shows the best performance when all available features are included. This suggests that combining strategically a comprehensive set of features can enhance forecasting accuracy.

Our report highlights some theoretical and practical implications. LSTM can effectively capture complex temporal dependencies, but not all features contribute equally. Considering various types of data can enhance the model's accuracy. Also, adding more features does not guarantee better results, and preprocessing may help. Additionally, the "black box" nature of the model makes it difficult to interpret, so explainability is needed.

Notable Limitations are pointed out in this report. External events like Elon Musk's tweets, regulatory changes, and international conflicts will significantly impact stock trends, and LSTM models cannot adapt once trained. Furthermore, sentiment data can be biased because specific social media sites, such as Yahoo Finance, only reflect the views of specific demographics.

Here are some future directions we may improve. Firstly, alternative architectures like transformer-based or hybrid models that combine LSTM with attention mechanisms should be explored. Feature refinement, like giving weight to sentiment scores according to different sources (some are reputable and reliable, some are not), is also one way to improve. Lastly, we need to consider model explainability. Using techniques like SHAP can help interpret.

8. References

[1] https://www.kaggle.com/datasets/equinxx/stock-tweets-for-sentiment-analysis-and-prediction

[2] https://www.kaggle.com/code/divyeshjain/stock-prediction-gan-twitter-sentiment-analysis