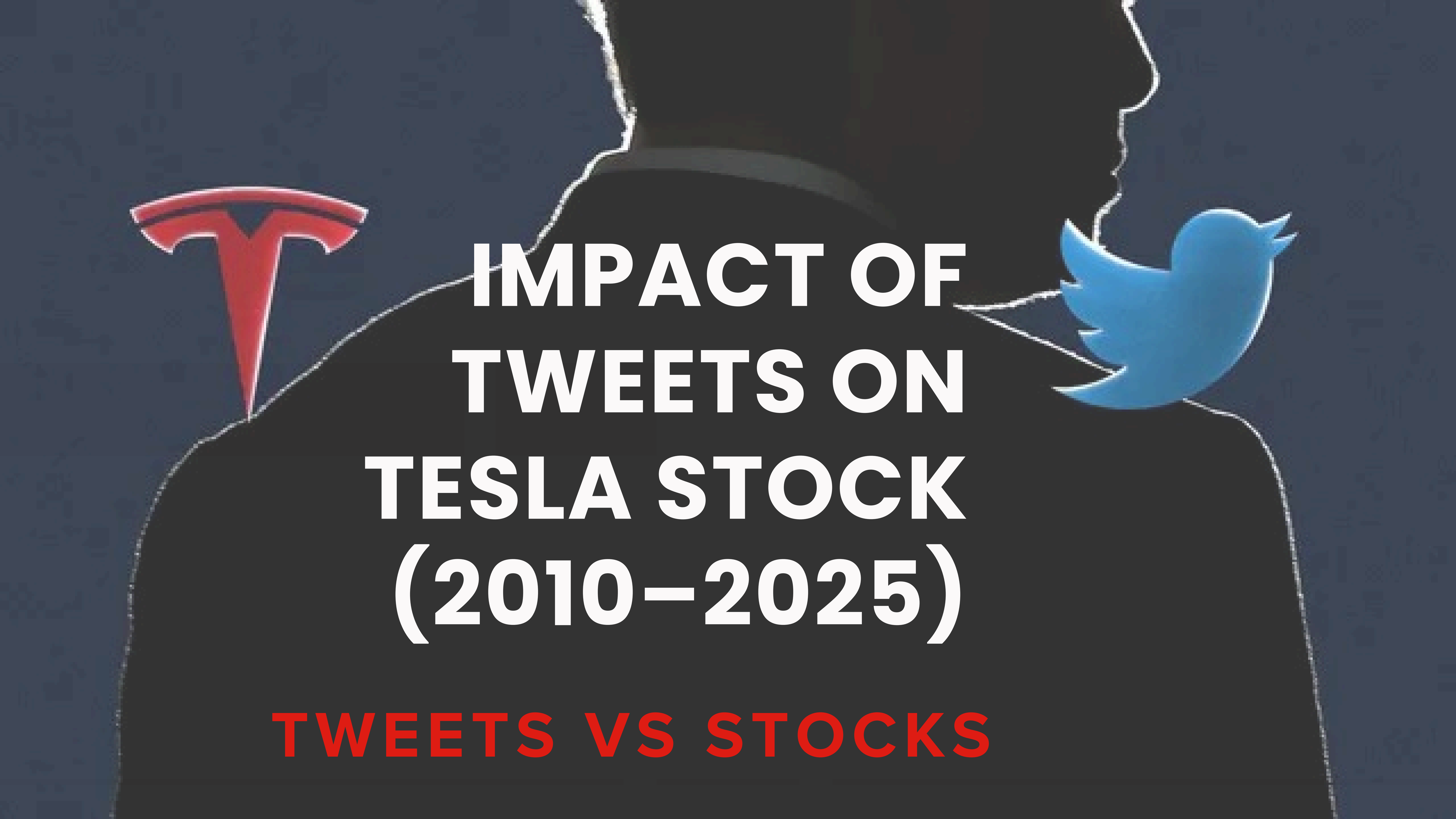

IMPACT OF TWEETS ON TESLA STOCK (2010–2025)
TWEETS VS STOCKS

# PURPOSE

- Explore and quantify the relationship between Elon Musk's tweets and Tesla’s stock price movements from 2010 to 2025.

# RESEARCH QUESTIONS

01 Do Elon Musk's tweets impact short-term Tesla stock price movement?

02 Can sentiment and content of tweets be used to predict stock price direction or volatility?

03 How accurately can a model forecast price movements using tweet metadata and sentiment scores?

| tweet_body | roberta_sentiment | roberta_pos_score | r |
|---|---|---|---|
| Yup | neutral | 0.290412 | |
| Massive public manipulation | negative | 0.009549 | |
| 🤣 🤣 | neutral | 0.310404 | |
| Prescient | neutral | 0.204723 | |
| Congratulations Tesla team on a great year!! | positive | 0.991541 | |

# DATA & WORKFLOW

- 2 Kaggle Data sets
  - 1 data set of all Elon Musk tweets from 2015 to 2025
  - 1 data set of Tesla stock price info from 2000 to 2025

## Data Cleaning:

- Filter the stock data set to match dates of tweets.
- Drop null and noisy columns (tweet url, original username if not Elon).
- Concat tweets data frame and stocks data frame together.

## Machine Learning Logic:

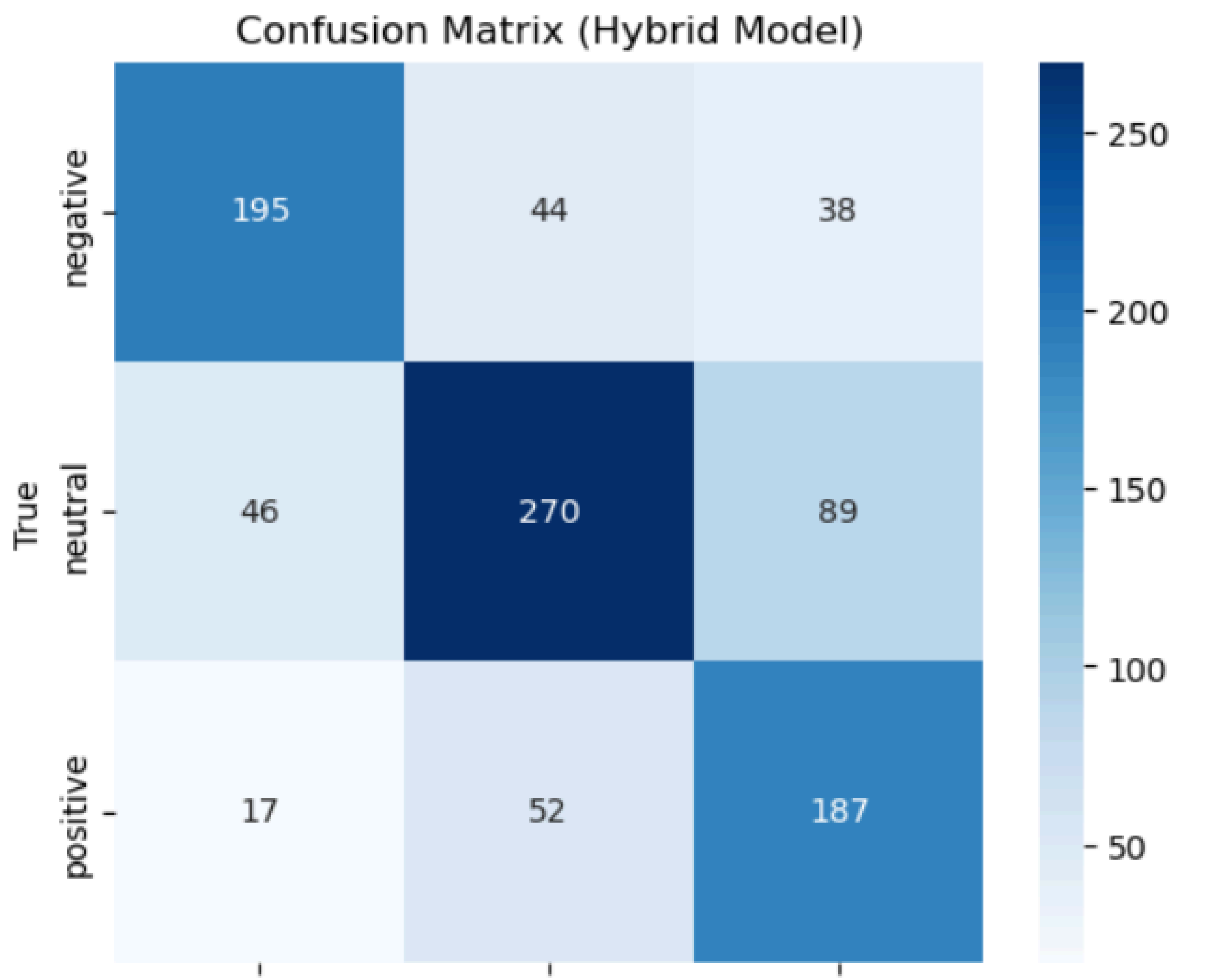
- Leveraging natural language processing (NLP), we applied sentiment analysis to classify each tweet as positive, negative, or neutral. We also quanitifed an engagement score for each tweet (likes, replies, retweets, etc). These features were used as input variables in our prediction model.

# MACHINE LEARNING TESTS

## LSTM Model

```python
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Embedding, LSTM, Dense, Dropout, Concatenate

# Text input branch
text_input = Input(shape=(X.shape[1],), name='text_input')
x = Embedding(input_dim=10000, output_dim=128)(text_input)
x = LSTM(128)(x)
x = Dropout(0.5)(x)

# Numeric input branch
num_input = Input(shape=(X_num_train.shape[1],), name='num_input')
n = Dense(64, activation='relu')(num_input)
n = Dropout(0.3)(n)

# Combine branches
combined = Concatenate()([x, n])
z = Dense(64, activation='relu')(combined)
z = Dropout(0.5)(z)
output = Dense(3, activation='softmax')(z)

# Create model
model = Model(inputs=[text_input, num_input], outputs=output)

# Compile
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negative | 0.76 | 0.70 | 0.73 | 277 |
| neutral | 0.74 | 0.67 | 0.70 | 405 |
| positive | 0.60 | 0.73 | 0.66 | 256 |
| accuracy | | | 0.70 | 938 |
| macro avg | 0.70 | 0.70 | 0.70 | 938 |
| weighted avg | 0.70 | 0.70 | 0.70 | 938 |



# MACHINE LEARNING TESTS

## Random Forest & XGBoost

```python
features = [
    "roberta_pos_score",
    "roberta_neg_score",
    "roberta_neu_score",
    "sentiment_polarity",
    "engagement_score"
]

target = "pct_change"

df_model = df[features + [target]].dropna()
X = df_model[features]
y = df_model[target]
```

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```python
##RandomTreeForest and XGBoost seem to be the most promising based on our data

rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)


xgb_model = XGBRegressor(n_estimators=100, learning_rate=0.1, random_state=42)
xgb_model.fit(X_train, y_train)
xgb_preds = xgb_model.predict(X_test)
```

```
Random Forest Evaluation:
  MAE:   2.4367
  MSE:   9.5523
  RMSE:  3.0907
  R²:    -0.0858

XGBoost Evaluation:
  MAE:   2.3957
  MSE:   9.2622
  RMSE:  3.0434
  R²:    -0.0528
```

# IMPORTANT TOOLS

## RoBERTA

A pre-trained NLP model from Hugging Face for understanding the meaning and emotion behind text. Gave us a classification of positive, negative, or neutral for each tweet, based on the text.

## Tokenizer

Needed to break the tweets down into “tokens” which RoBERTA could process and understand.

## TextBlob

While RoBERTA gave each tweet a classification of sentiment, TextBlob gave us a numerical value for sentiment polarity.

# EDA

## Tweet Content Overview

- Dataset includes 2,000+ tweets from Elon Musk
- Common topics: Tesla, SpaceX, AI, Dogecoin, X (Twitter)
- Most used hashtags: #Tesla, #SpaceX, #AI, #Dogecoin
- Frequent keywords: “launch,” “update,” “working on,” “fun,” “free speech”
- Word cloud revealed emphasis on tech, innovation, and finance

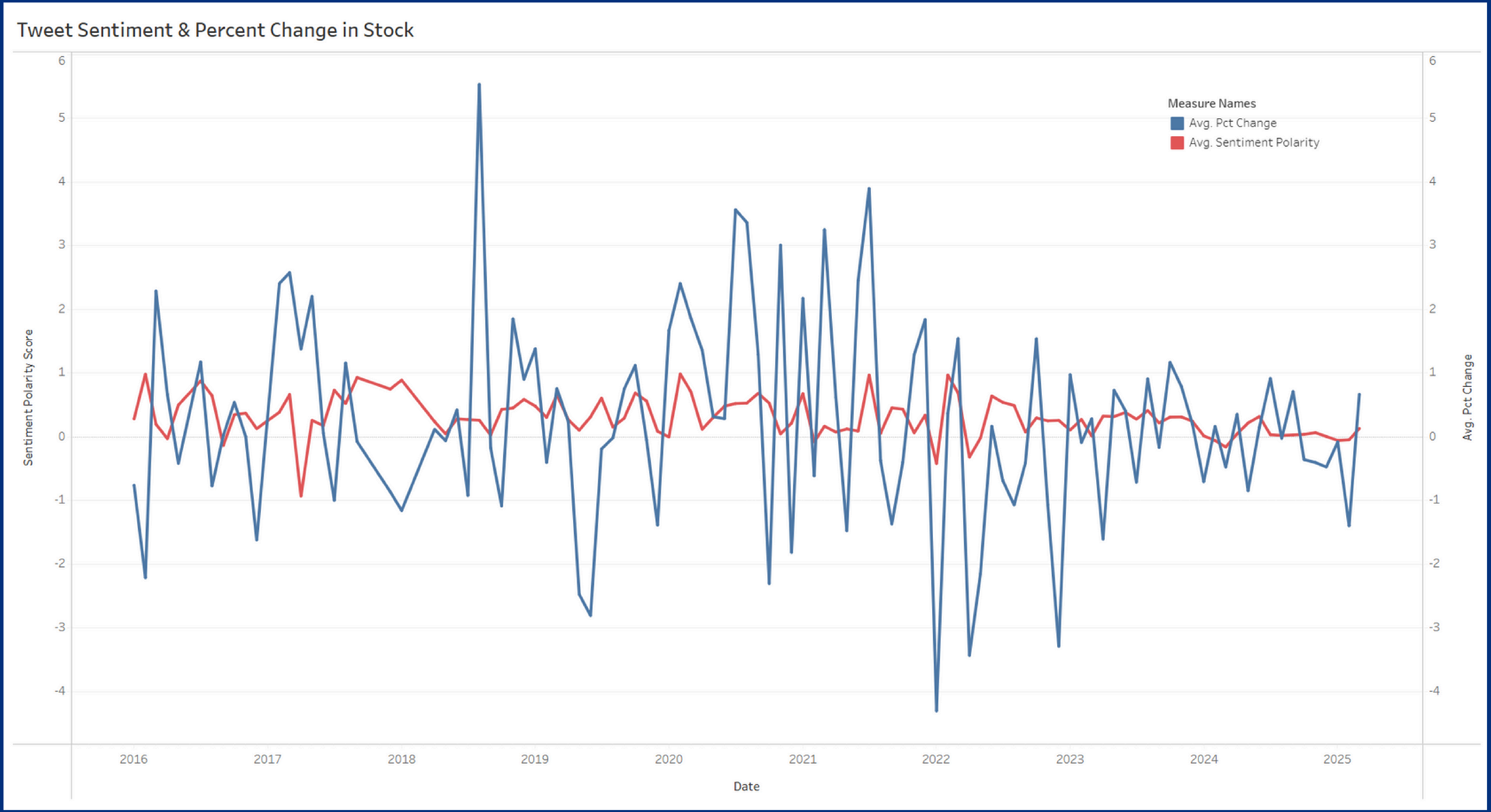
## Engagement & Sentiment Insights

- Avg. likes: 150K | Avg. retweets: 25K
- Top tweet: 500K+ likes, often tied to product updates or memes
- Sentiment (VADER):
- Positive: 42%, Neutral: 38%, Negative: 20%
- Positive sentiment linked to innovation and humor, negative tied to controversies

# LIVE DEMO

lalibedrosian.pythonanywhere.com

# CONCLUSION



Tweet Sentiment & Percent Change in Stock
Measure Names
Avg. Pct Change
Avg. Sentiment Polarity
Sentiment Polarity Score
Avg. Pct Change
6
5
4
3
2
1
0
-1
-2
-3
-4
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
Date

# ANY QUESTIONS?


THANK
YOU