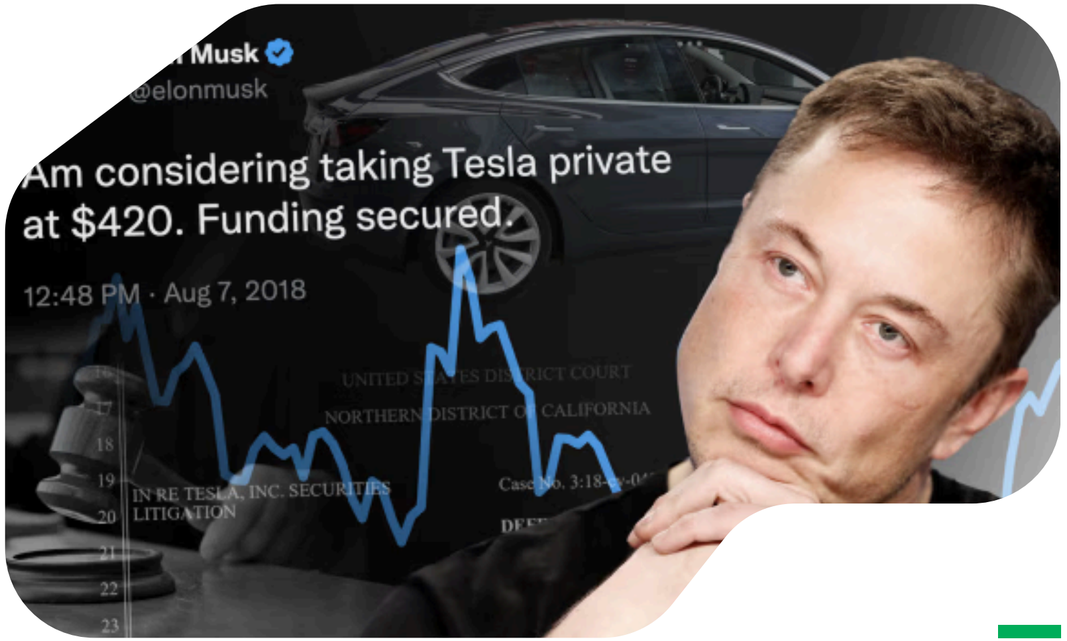
Tweet Sentiment Analysis

How do Elon Musk's tweets impact Tesla's stock price?



Presented by

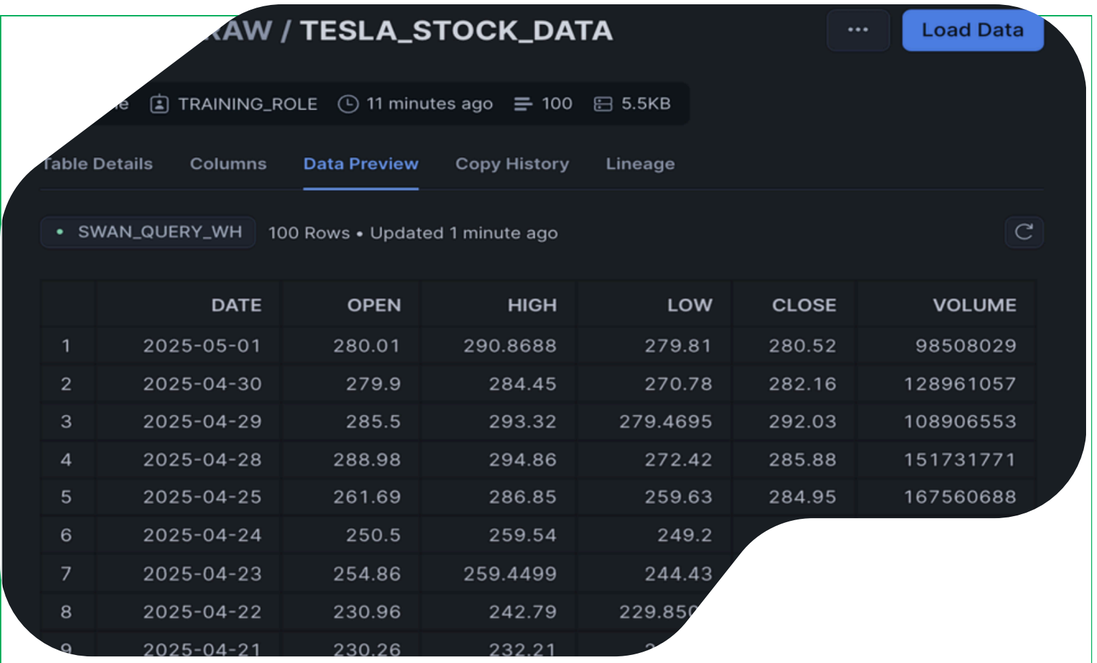
Andreah Cruz,
Daniel Kim,
Jane Heng,
Lakshmi Bharathy Kumar,
Savitha Vjayarangan

GROUP 9

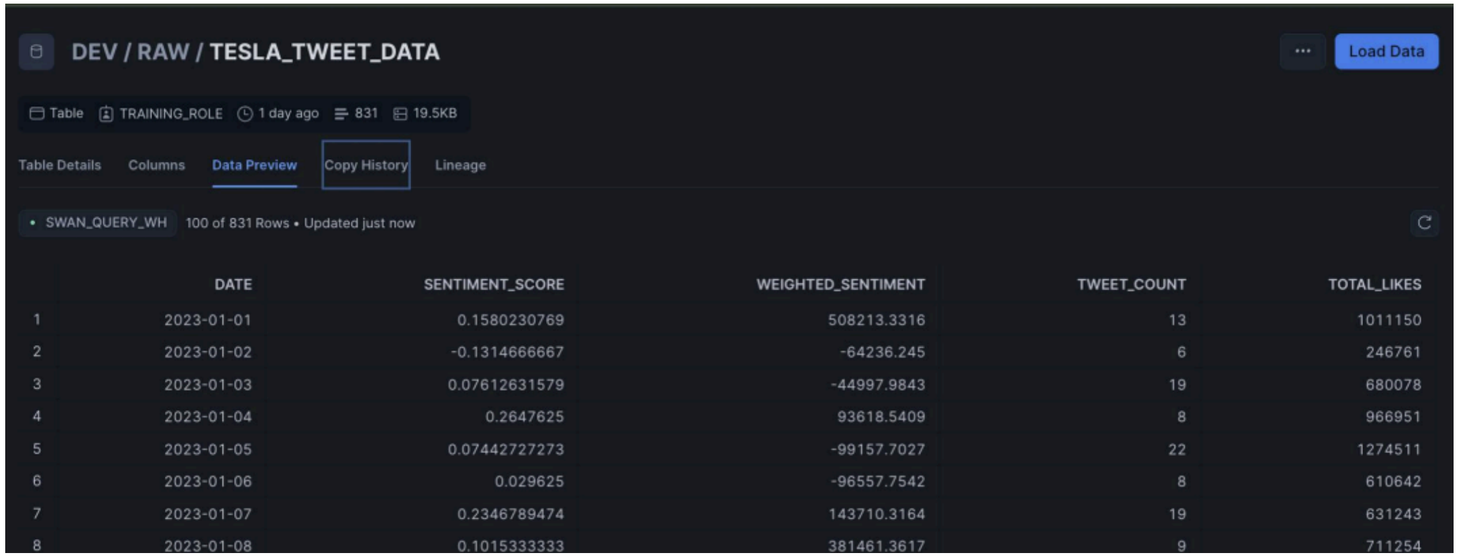
PROBLEM STATEMENT

- With social media becoming a powerful communication tool, analyzing influential figures' posts can offer **insight** into how **public sentiment** shapes **financial behavior**
- Project **predicts** Tesla's next-day closing price using **engineered sentiment features** from Musk's tweets
- Dataset combines historical tweets from Elon Musk & Tesla stock price data ==> measure sentiment trends & compare them with corresponding stock movements
- This project uses:
  - Tweet data from 2010–2025 (sourced via CSV)
  - Tesla stock data pulled through Python-based ETL
  - Tools like Airflow & dbt to automate the pipeline/transform data into analytics ready formats

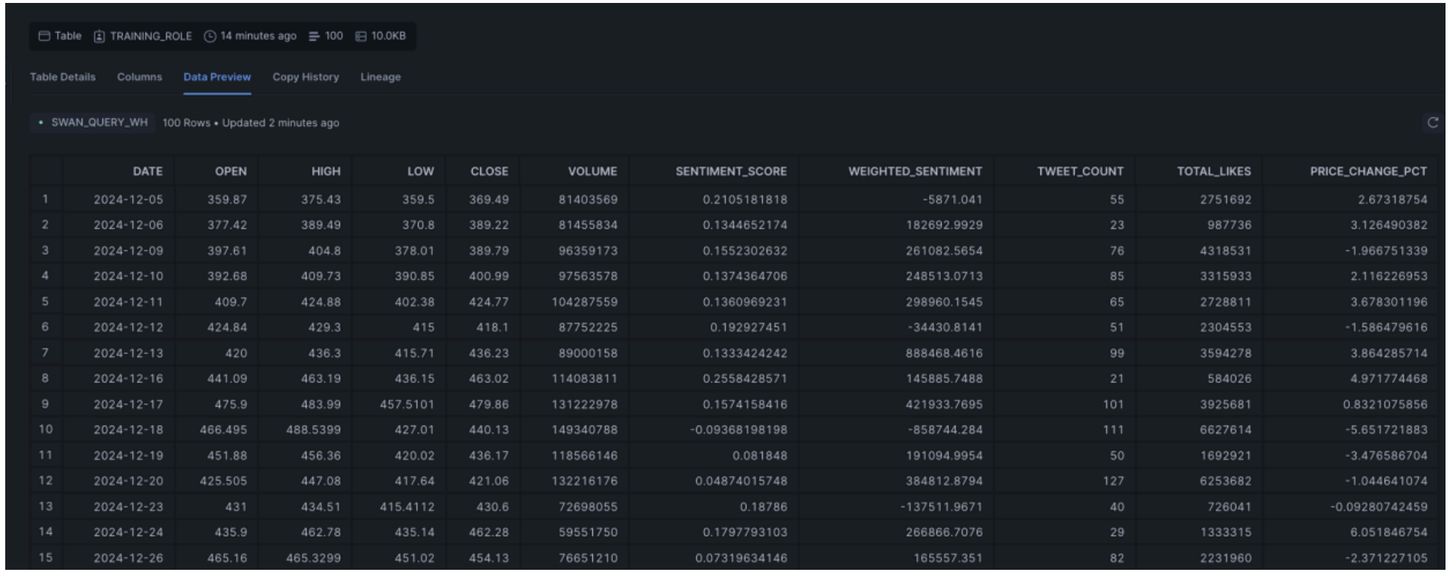
DATASET CONTEXT



	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME
1	2025-05-01	280.01	290.8688	279.81	280.52	98508029
2	2025-04-30	279.9	284.45	270.78	282.16	128961057
3	2025-04-29	285.5	293.32	279.4695	292.03	108906553
4	2025-04-28	288.98	294.86	272.42	285.88	151731771
5	2025-04-25	261.69	286.85	259.63	284.95	167560688
6	2025-04-24	250.5	259.54	249.2		
7	2025-04-23	254.86	259.4499	244.43		
8	2025-04-22	230.96	242.79	229.85		
9	2025-04-21	230.26	232.21			

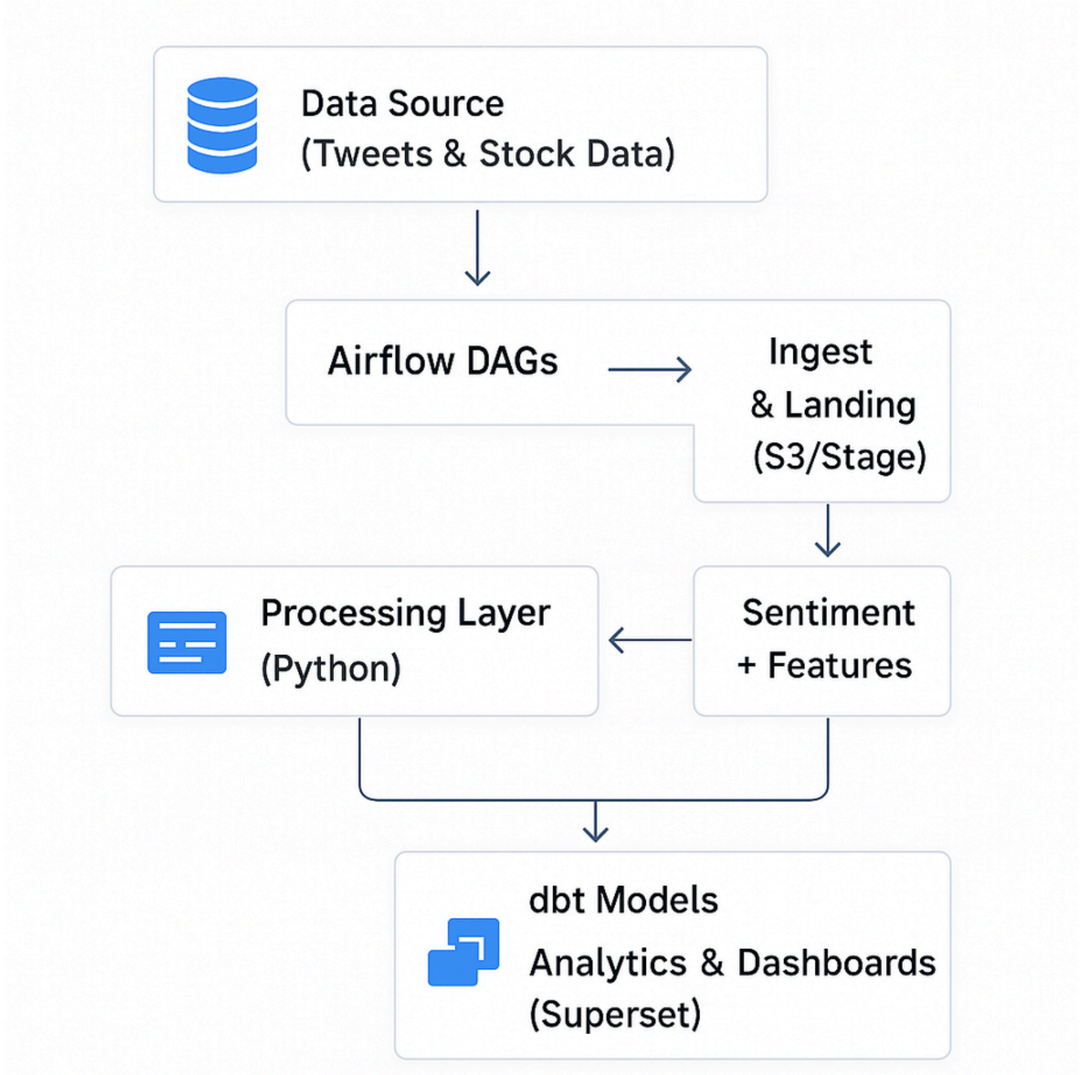


	DATE	SENTIMENT_SCORE	WEIGHTED_SENTIMENT	TWEET_COUNT	TOTAL_LIKES
1	2023-01-01	0.1580230769	508213.3316	13	1011150
2	2023-01-02	-0.1314666667	-64236.245	6	246761
3	2023-01-03	0.07612631579	-44997.9843	19	680078
4	2023-01-04	0.2647625	93618.5409	8	966951
5	2023-01-05	0.07442727273	-99157.7027	22	1274511
6	2023-01-06	0.029625	-96557.7542	8	610642
7	2023-01-07	0.2346789474	143710.3164	19	631243
8	2023-01-08	0.1015333333	381461.3617	9	711254



	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	SENTIMENT_SCORE	WEIGHTED_SENTIMENT	TWEET_COUNT	TOTAL_LIKES	PRICE_CHANGE_PCT
1	2024-12-05	359.87	375.43	359.5	369.49	81403569	0.2105181818	-5871.041	55	2751692	2.67318754
2	2024-12-06	377.42	389.49	370.8	389.22	81455834	0.1344652174	182692.9929	23	987736	3.126490382
3	2024-12-09	397.61	404.8	378.01	389.79	96359173	0.1552302632	261082.5654	76	4318531	-1.966751339
4	2024-12-10	392.68	409.73	390.85	400.99	97563578	0.1374364706	248513.0713	85	3315933	2.116226953
5	2024-12-11	409.7	424.88	402.38	424.77	104287559	0.1360969231	298960.1545	65	2728811	3.678301196
6	2024-12-12	424.84	429.3	415	418.1	87752225	0.192927451	-34430.8141	51	2304553	-1.586479616
7	2024-12-13	420	436.3	415.71	436.23	89000158	0.1333424242	88468.4616	99	3594278	3.864285714
8	2024-12-16	441.09	463.19	436.15	463.02	114083811	0.2558428571	145885.7488	21	584026	4.971774468
9	2024-12-17	475.9	483.99	457.5101	479.86	131222978	0.1574158416	421933.7695	101	3925681	0.8321075856
10	2024-12-18	466.495	488.5399	427.01	440.13	149340788	-0.09368198198	-858744.284	111	6627614	-5.651721883
11	2024-12-19	451.88	456.36	420.02	436.17	118566146	0.081848	191094.9954	50	1692921	-3.476586704
12	2024-12-20	425.505	447.08	417.64	421.06	132216176	0.04874015748	384812.8794	127	6253682	-1.044641074
13	2024-12-23	431	434.51	415.4112	430.6	72698055	0.18786	-137511.9671	40	726041	-0.09280742459
14	2024-12-24	435.9	462.78	435.14	462.28	59551750	0.1797793103	266866.7076	29	1333315	6.051846754
15	2024-12-26	465.16	465.3299	451.02	454.13	76651210	0.07319634146	165557.351	82	2231960	-2.371227105

- **Tesla Stock Prices – Yahoo Finance API**
  - Daily open, high, low, close, volume (2010 – present)
  - Collected using the **yfinance** Python package
- **Elon Musk Tweets – all_musks.csv (2023-2025)**
  - Pre-collected dataset scored using the **VADER model** in Google Colab
  - Aggregated daily sentiment (sentiment score, tweet count, likes, etc.)



System ARCHITECTURE

Data Sources

Tesla stock prices & Elon Musk tweets are extracted. Sentiment is computed in Colab using VADER & stored as daily aggregates

Airflow (Ingest Layer)

Schedules and automates the daily extraction and loading of raw stock and tweet data into Snowflake staging tables

Processing Layer

Sentiment scoring was performed externally in Colab. Airflow handles scheduling; basic pre-processing logic is built into the DAG.

Snowflake Data Warehouse

Central repository for both raw and transformed tables.

dbt Transformation Layer

SQL-based models engineer features, run the regression formula, and generate predictions

Analytics & Dashboards

Superset visualizes trends, predictions, and evaluation metrics using dbt-transformed tables

GROUP 9

View: feature_engineering_combined_features

Field Name	Data Type	Attributes	Constraints	Description
date	DATE	Not null	Primary Key with SYMBOL	Trading date
sentiment_score	FLOAT			
weighted_sentiment	FLOAT			
tweet_count	NUMBER			
total_likes	NUMBER			
lag_1_sentiment	FLOAT			
avg_3d_sentiment	FLOAT			
open_price	FLOAT			Opening price
close_price	FLOAT			Closing price
low_price	FLOAT			Lowest price
high_price	FLOAT			Highest price
volume	NUMBER			volume
Price_change_pct	FLOAT			(current close – last close) / last close
high_low_range_pct	FLOAT			High – low/low
Avg_7d_close	FLOAT			Average of the closing prices over the past 7 days
weighted_tweet_impact	FLOAT			

View: model_predict_price

Field Name	Data Type	Attributes	Constraints	Description
date	DATE	Not null	Primary Key with SYMBOL	Trading date
sentiment_score	FLOAT			
weighted_sentiment	FLOAT			
tweet_count	NUMBER			
total_likes	NUMBER			
lag_1_sentiment	FLOAT			
avg_3d_sentiment	FLOAT			
open_price	FLOAT			Opening price
close_price	FLOAT			Closing price
low_price	FLOAT			Lowest price
high_price	FLOAT			Highest price
volume	NUMBER			volume
Price_change_pct	FLOAT			(current close – last close) / last close
high_low_range_pct	FLOAT			High – low/low
Avg_7d_close	FLOAT			Average of the closing prices over the past 7 days
weighted_tweet_impact	FLOAT			
Predicted_close_price	FLOAT			Predicted price

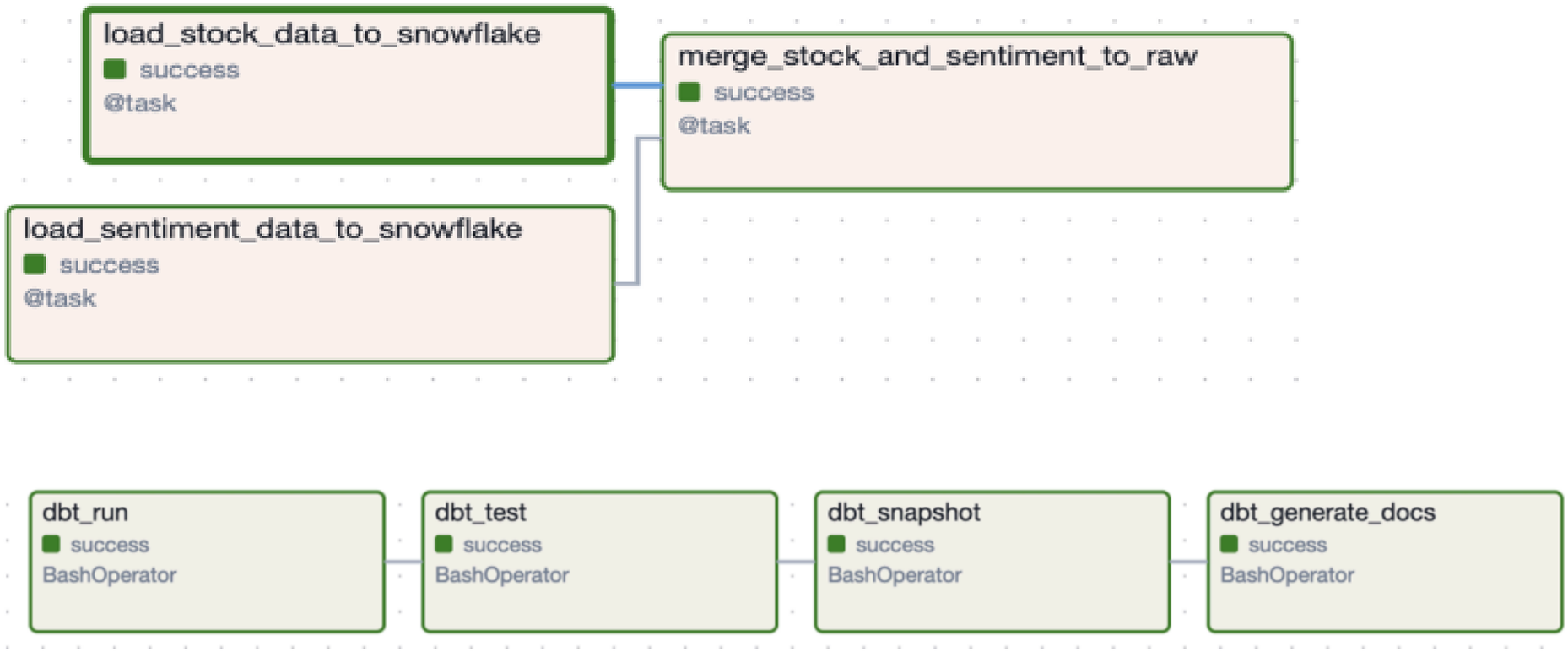
View: evaluation_prediction_output

Field Name	Data Type	Attributes	Constraints	Description
date	DATE	Not null	Primary Key with SYMBOL	Trading date
sentiment_score	FLOAT			
weighted_sentiment	FLOAT			
tweet_count	NUMBER			
total_likes	NUMBER			
open_price	FLOAT			Opening price
close_price	FLOAT			Closing price
low_price	FLOAT			Lowest price
high_price	FLOAT			Highest price
volume	NUMBER			volume
Price_change_pct	FLOAT			(current close – last close) / last close
high_low_range_pct	FLOAT			High – low/low
Avg_7d_close	FLOAT			Average of the closing prices over the past 7 days
weighted_tweet_impact	FLOAT			
Predicted_close_price	FLOAT			Predicted price
Prediction_error	FLOAT			
Relative_error	FLOAT			

DB SCHEMA

- **Raw Schema**
  - stock_data (date, symbol) PK: daily OHLCV prices
  - tweet_data (date) PK: aggregated Musk-tweet sentiment
  - final_raw (date, symbol) PK: joins stock_data + tweet_data, adds price_change_pct
- **Analytics Schema**
  - feature_engineering_combined_features (date, symbol) PK: rolling/lagged features
  - model_predict_price (date, symbol) PK: predicted next-day close
  - evaluation_prediction_output (date, symbol) PK: prediction error & relative_error
- **Key Points**
  - Composite PK (date, symbol) ensures one row per ticker per day
  - Foreign keys enforce consistency between raw tables
  - Easily extensible with dim_date or dim_symbol for richer BI filters

ETL/ELT Pipeline (Airflow + dbt)

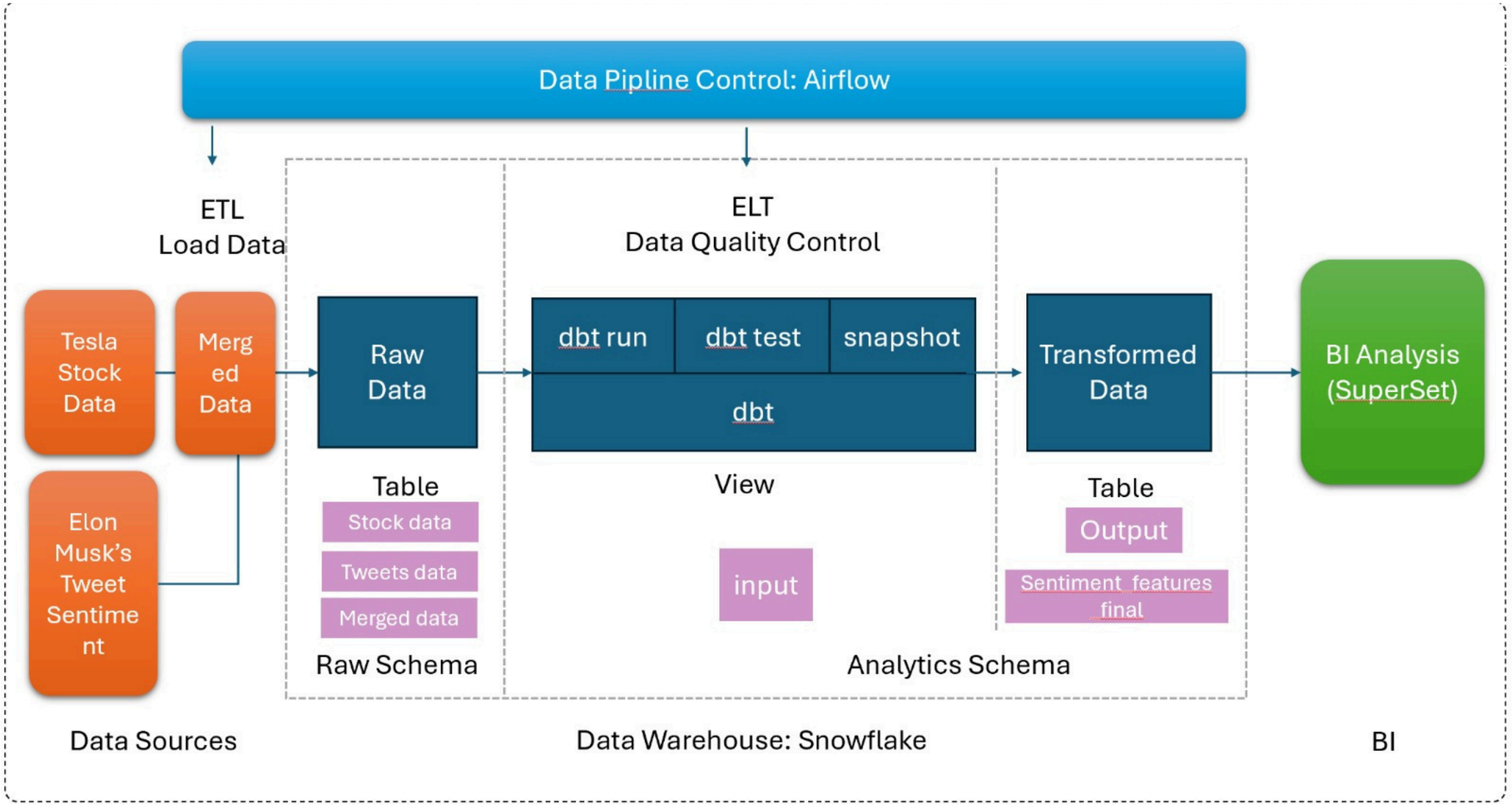


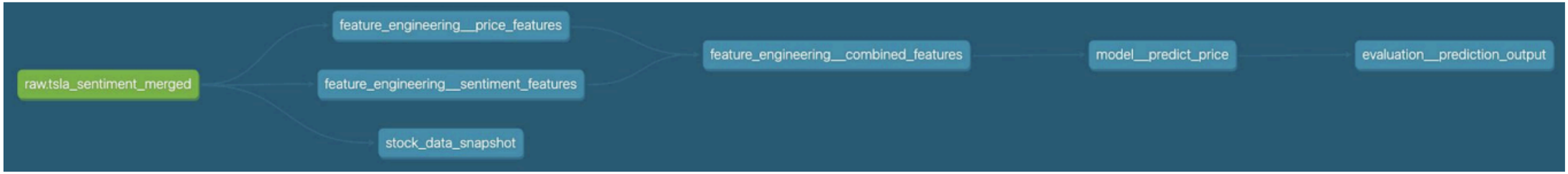
Airflow schedules:

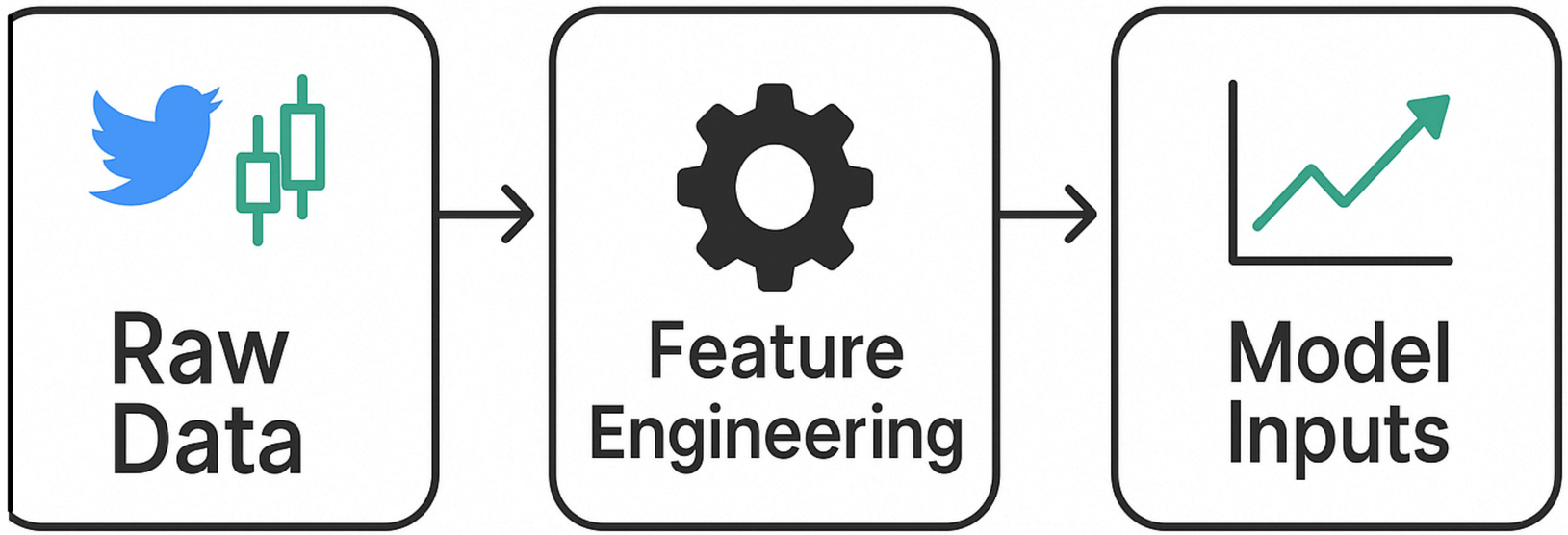
- extract_stock_data – pulls latest prices
- extract_tweet_data – fetches Elon Musk's tweets & computes daily sentiment
- load_stock_data_to_snowflake – loads raw stock_data into Snowflake
- load_sentiment_data_to_snowflake – loads rawtweet_data into Snowflake
- merge_stock_and_sentiment_to_raw – merges raw tables into final_raw_table

dbt transforms:

- dbt run – executes SQL models
- dbt test – enforces data quality checks
- dbt snapshot – captures historical versions of key tables for auditing


Data Pipline Control: Airflow
ETL
Load Data
ELT
Data Quality Control
Tesla Stock Data
Merged Data
Elon Musk's Tweet Sentiment
Raw Data
dbt run
dbt test
snapshot
dbt
Transformed Data
BI Analysis (SuperSet)
Table
Stock data
Tweets data
Merged data
Raw Schema
View
input
Table
Output
Sentiment_features_final
Analytics Schema
Data Sources
Data Warehouse: Snowflake
BI


raw.tsla_sentiment_merged
feature_engineering__price_features
feature_engineering__sentiment_features
stock_data_snapshot
feature_engineering__combined_features
model__predict_price
evaluation__prediction_output



Feature Engineering

Feature Name	Coefficient (β)	Interpretation
Intercept (β_0)	-6.9047	Base value when all features are 0
SENTIMENT_SCORE	6.2242	Higher sentiment today increases price
TWEET_COUNT	0.0030	More tweets slightly increase price
LAG_1_SENTIMENT	4.9216	Yesterday's sentiment still positively influences
AVG_3D_SENTIMENT	28.5573	Recent 3-day sentiment trend has strongest impact
AVG_7D_CLOSE	1.0163	Recent price trend also contributes positively

modeling approach

VADER Sentiment Analysis Requires Python

- VADER is a Python-based NLP tool not supported in Snowflake or SQL. Using Colab allowed fast, accurate sentiment scoring on Elon Musk's tweets.

Snowflake Doesn't Support Native ML Training (for free tiers)

- Advanced ML features like FORECAST or ML_PREDICT are part of Snowflake Cortex, which requires premium licensing not available in this setup.

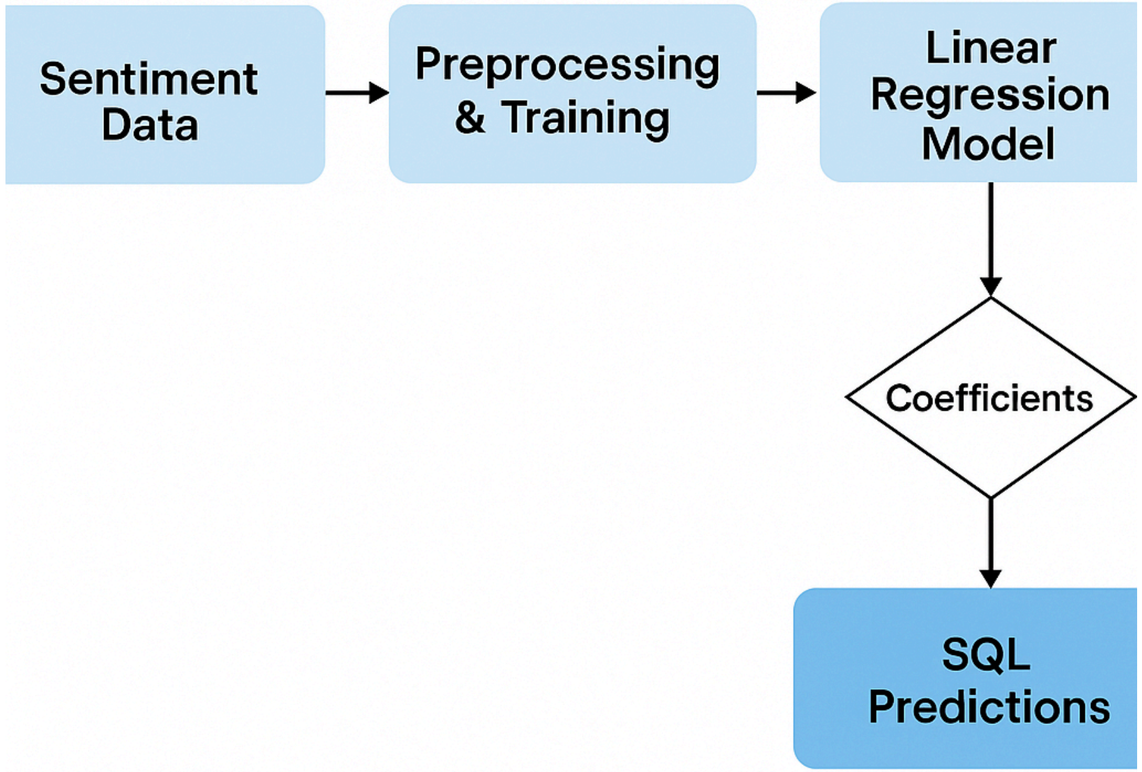
Hybrid Architecture = Best of Both Worlds

- Training and sentiment scoring were done in Python (flexible, powerful), while SQL was used for efficient, scalable in-database prediction.

In-Database Inference Enables Real-Time BI

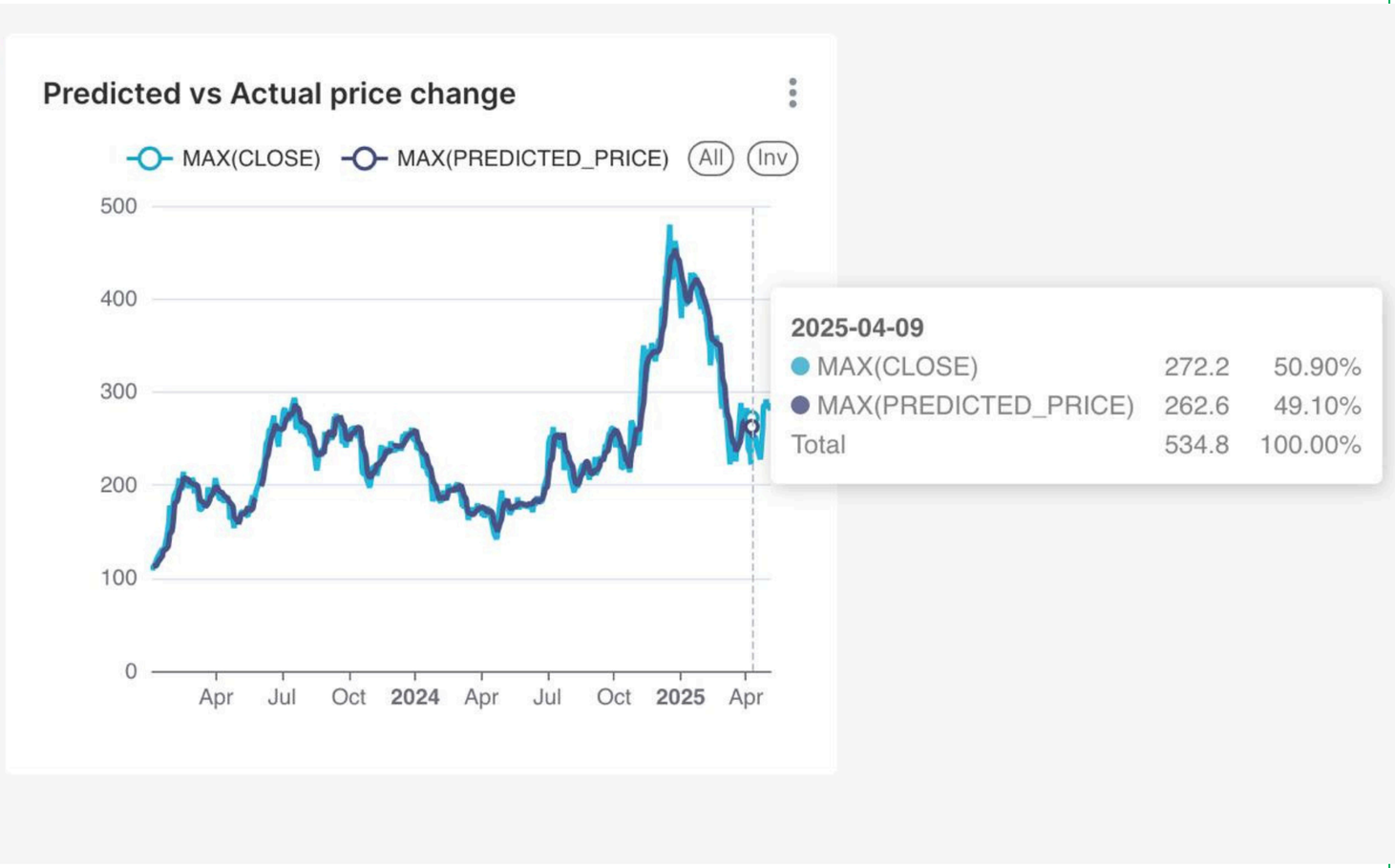
- With predictions calculated in SQL (output.sql), results can be directly consumed by dashboards in tools like Superset without data movement.

In-Database Inference Modeling



Model Evaluation

Actual vs Predicted Line Chart



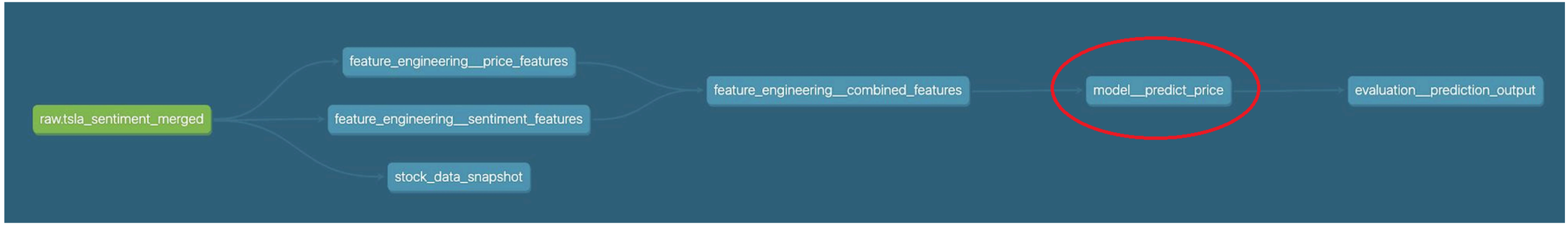
Metric	Value	Interpretation
RMSE	19.03	Avg. prediction error size in stock price units
MAE	14.58	Avg. absolute difference between predicted and actual price
R^2 Score	0.9160	Model explains ~91.6% of price variation (very strong fit)

DEMO AND VISUALS

Prediction Execution in Snowflake

Deployed regression model directly within Snowflake using SQL ==> keeps entire pipeline centralized, scalable, and auditable.

- Trained in Colab, coefficients were manually imported to output.sql
- SQL calculates predicted_close_price using feature columns like:
  - lag_1_sentiment, avg_3d_sentiment, weighted_impact, etc.
- dbt model is materialized as final table used for dashboards
- Benefits:
  - No external model serving required
  - Fast, fully auditable, and easy to update
  - Leverages Snowflake's compute engine for scalability



Limitations & Challenges

- Model constraints:
  - Linear regression assumes feature independence and linearity
- Feature limitations:
  - Some engineered features (like volume) were non-informative in this setup
- Snowflake ML not used:
  - Snowflake's in-database ML was unavailable due to licensing in environment, so prediction logic was implemented manually in SQL

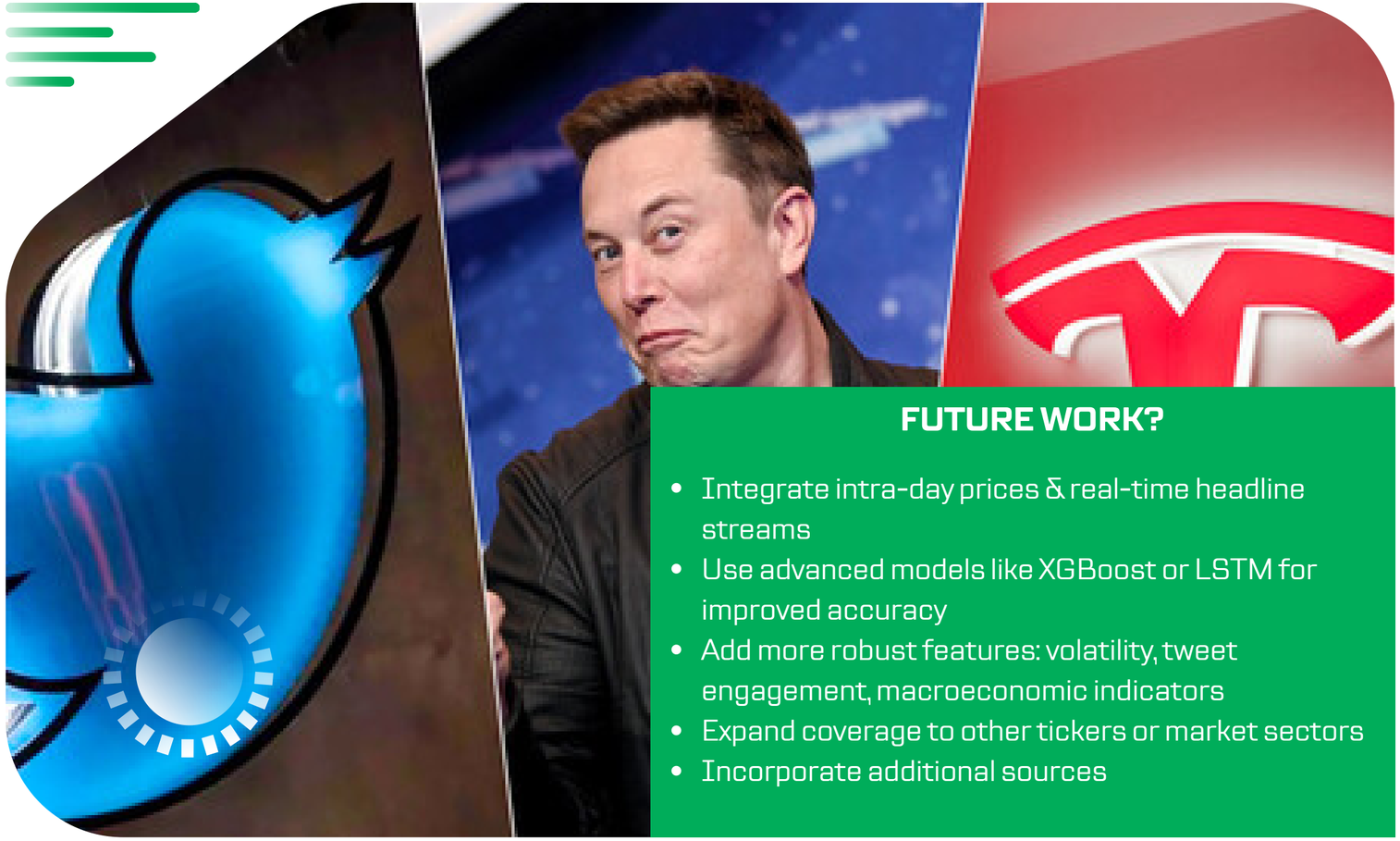
GROUP 9

KEY FINDINGS:

- News or tweets sound negative ==> Tesla's stock usually dips the next day
- Positive news pushes prices up, but not by a huge amount; bigger market forces still play a role
- Price move shows up most clearly the day after the sentiment hit, so traders aren't reacting instantly

WHAT WE LEARNED/MODEL LEARNED:

- R^2 = 0.916 shows strong explanatory power; engineered sentiment features like lag_1_sentiment & avg_3d_sentiment were highly informative.
- Built a transparent, version-controlled pipeline using dbt & SQL—easy to scale and audit..
- Demonstrated that social sentiment can predict price, but only with thoughtful feature engineering and lag-aware modeling.



FUTURE WORK?

- Integrate intra-day prices & real-time headline streams
- Use advanced models like XGBoost or LSTM for improved accuracy
- Add more robust features: volatility, tweet engagement, macroeconomic indicators
- Expand coverage to other tickers or market sectors
- Incorporate additional sources

CONCLUSION

BORCELLE

THANK YOU!

GROUP 9