

Q Stock Analyzer



Abdullah Al Marzouq
Akash Beh
Rayane Tarazi
Stanley Gunawan

Q is...

A machine learning algorithm that uses a semantic analyzer along with historical data, and any public insider trading data, to predict the movement of a stock price, and give the user a recommendation on whether to buy, sell, or hold.



Project Goals



Goal 1

Make it easy for people to accurately invest in stocks

Goal 2

Add to the knowledge base of financial stock prediction algorithms

Goal 3

Increase all team members' capital

Justifying Q

Our Experience

- AI
- ML
- Cognitive Systems
- Linear Algebra

Our Interest

We are really interested to see if there is an accurate way to predict markets using machine learning.

Our Abilities

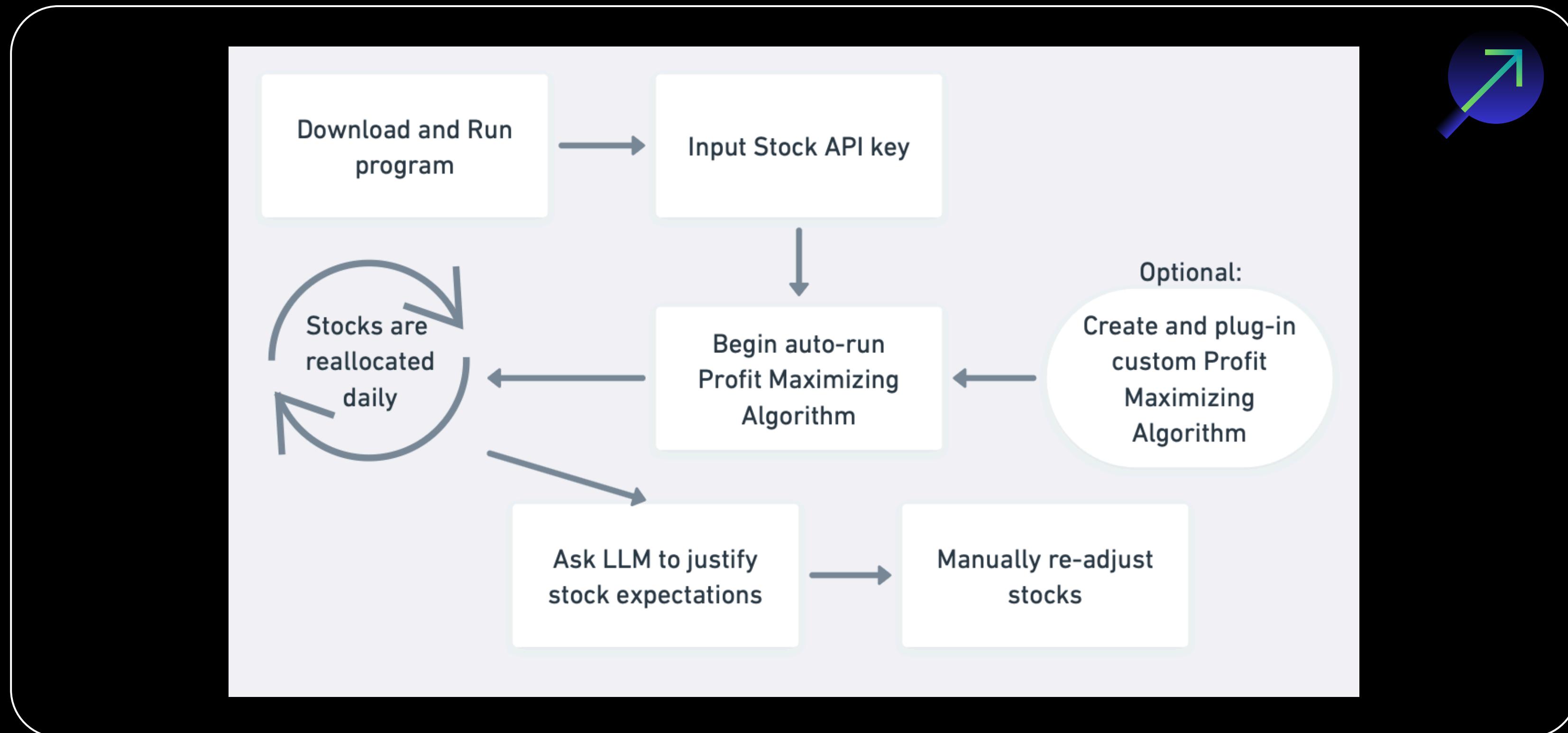
This is a challenging project that requires a lot of different moving parts to keep track of, being able to complete this will give our team immense confidence in our abilities as well as practical skills to use in the future.



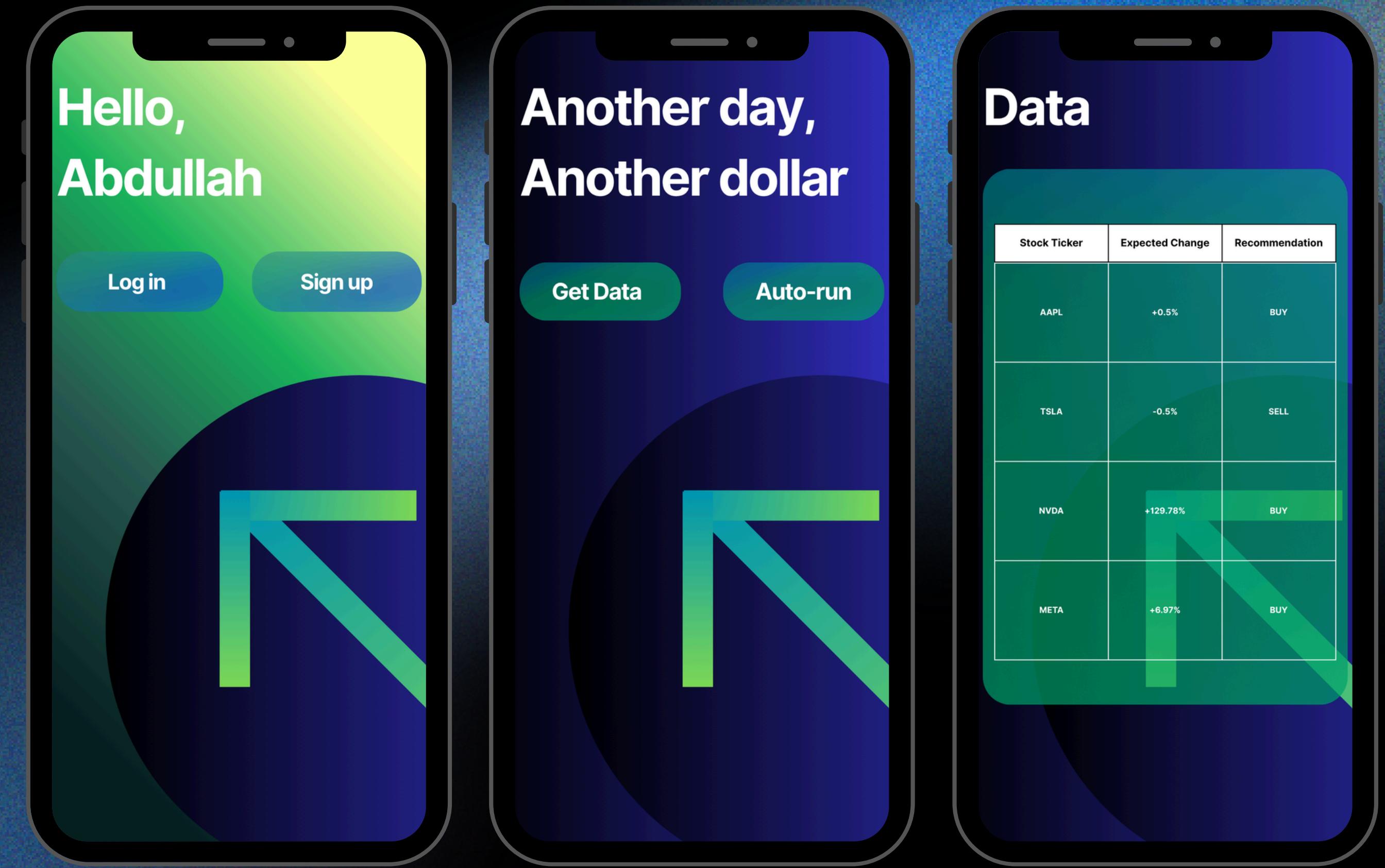
User Stories

- **Story/Scenario 1**
Broke → Finds this algorithm → Rich
- **Story/Scenario 2**
Works a 9-5 → Finds this algorithm → Retired
- **Story/Scenario 3**
Expert quant → Asks LLM about stock
→ makes better predictions
- **Story/Scenario 4**
Venture capitalist → Invests in us → Algorithm improves

Use Case Diagram



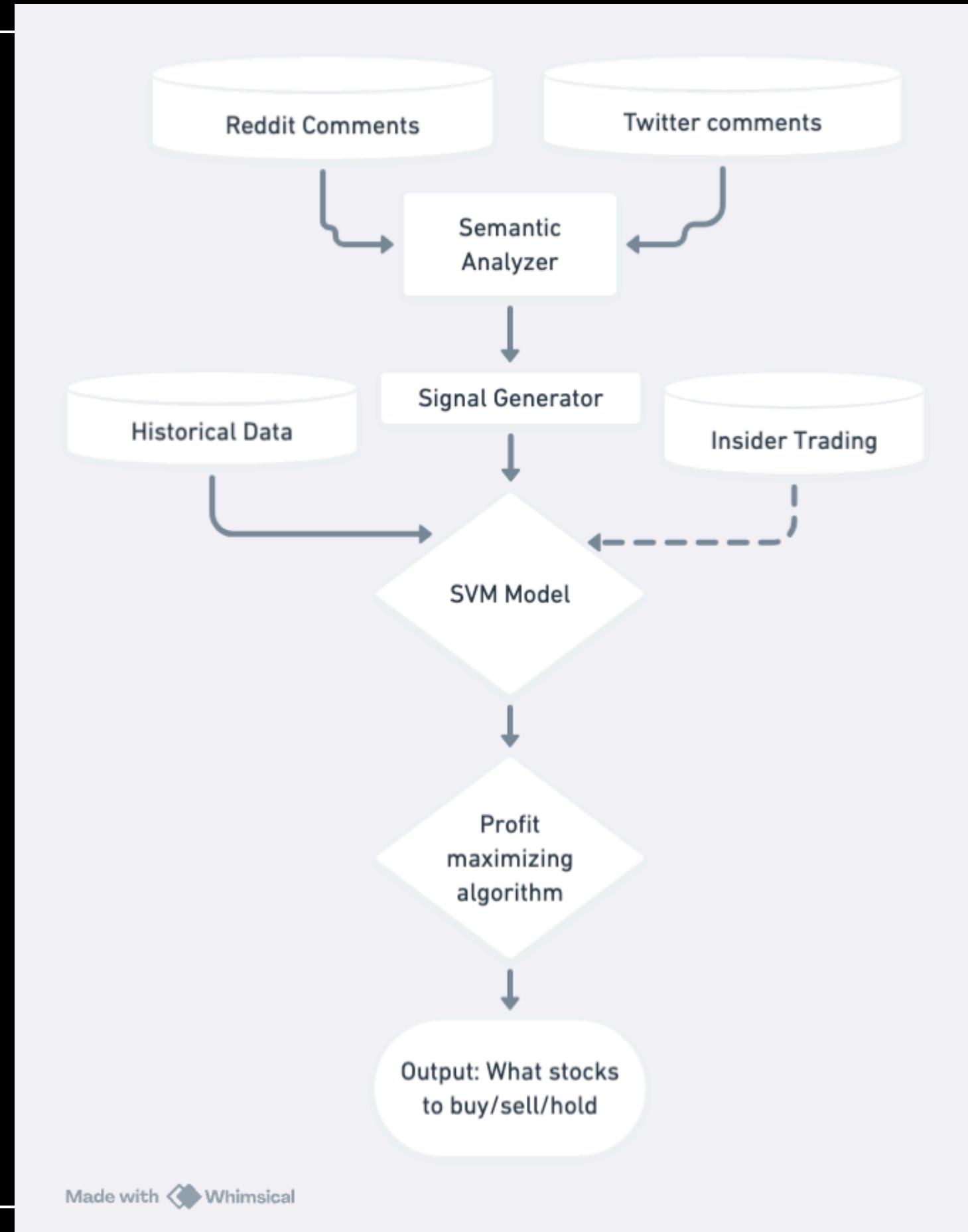
Mock UI



Disclaimer: This is just a tentative version of a UI in the case that we might expand into it. It is not a part of our current plan or project and is simply presented for clarity.

Architecture /Data Flow

We start by scraping and analyzing sentiment across social media platforms, this data is then combined with historical data collected from yfinance from 2008-01-01 to the present day, and any insider trading data that may be present. This is then used as data for training and testing our svm model. The classifications done by the svm model are then used by our profit maximizing algorithm to output the stocks that are predicted to go up, down, or stay within a hold range(eg. change is within [-3%,3%])



```
Class Distribution in dataset:  
target  
0 66.564080  
1 18.355683  
-1 15.080237  
Name: proportion, dtype: float64  
  
Training on 3639 samples, testing on 910 samples.  
Training SVM model with RBF kernel...  
Model training complete.  
  
=====  
Model Evaluation Results  
=====  
Accuracy: 94.29%  
  
Classification Report:  
precision recall f1-score support  
  
Sell (-1) 0.98 0.79 0.87 137  
Hold (0) 0.93 1.00 0.96 602  
Buy (1) 0.99 0.88 0.93 171  
  
accuracy 0.94 910  
macro avg 0.96 0.89 0.92 910  
weighted avg 0.95 0.94 0.94 910  
  
=====  
=====  
===== BACKTESTING WITH 100 AGAINST B&H =====  
=====  
  
--- Running SVM Strategy Simulation ---  
--- Running Buy and Hold Simulation ---  
--- Running Raw SVC Signal Simulation ---  
  
--- Detailed Results per Ticker (Top 10 by SVM Profit) ---  
strategy SVM Model Raw SVC Signal Buy and Hold  
ticker  
SOFI 4.24 0.00 14.04  
SNAP 3.04 0.25 13.62  
SOUN 2.17 1.99 32.51  
TV 2.15 0.00 13.90  
RKT 2.02 0.00 6.69  
TSLA 1.09 0.50 -4.32  
UAMY 1.00 0.00 -25.85  
YOU 0.51 0.00 -2.01  
WEN 0.36 0.00 2.08  
TSM 0.33 -0.28 0.94  
  
--- Average (Portfolio) Results Across All Tickers ---  
final_value profit total_return_pct risk_pct sharpe_ratio  
strategy  
SVM Model $99.23 $-0.77 -0.77% 0.25% -0.02  
Raw SVC Signal $99.98 $-0.02 -0.02% 0.03% -0.07  
Buy and Hold $98.49 $-1.51 -1.51% 3.03% 0.45  
  
=====  
ticker strategy final_value profit total_return_pct risk_pct sharpe_ratio  
0 RKLB SVM Model 94.731138 -5.268862 -0.052689 0.01037 -2.774068  
1 RKT SVM Model 102.022879 2.022879 0.020229 0.00552 2.922754  
2 RMD SVM Model 100.000000 0.000000 0.000000 0.00000 0.000000  
3 RR SVM Model 77.725871 -22.274129 -0.222741 0.02296 -5.605911  
4 RS SVM Model 100.000000 0.000000 0.000000 0.00000 0.000000  
.. ... ... ... ... ... ...  
104 XPEV Raw SVC Signal 100.000000 0.000000 0.000000 0.00000 0.000000  
105 XPO Raw SVC Signal 100.000000 0.000000 0.000000 0.00000 0.000000  
106 XYZ Raw SVC Signal 100.000000 0.000000 0.000000 0.00000 0.000000  
107 YOU Raw SVC Signal 100.000000 0.000000 0.000000 0.00000 0.000000  
108 ZIM Raw SVC Signal 100.000000 0.000000 0.000000 0.00000 0.000000
```



Demo

Challenges

Description

- Historical Data Acquisition: Our biggest challenge is that the live APIs for Reddit and Twitter cannot be used to get the deep historical data needed for backtesting. We must acquire and process a large data archive(e.g. Academic Torrent).
- Profit Maximizing Algorithm: We tried several simple formulas, but we are still working on the ideal algorithm to maximize profits, given the prediction data.
- Hyperparameter setting: We have to experiment with different hold thresholds as well as others for the profit-maximizing algorithm.
- Data Cleaning: Our scrapers and data loaders will constantly find bad data (e.g., junk tickers, non-standard timestamps like in the Twitter files, API errors) that will break our scripts if not handled.
- Backtesting Bias: We must avoid lookahead bias (e.g., scaling our entire dataset before the train/test split). Our model must beat the "Buy and Hold" benchmark on a risk-adjusted basis (Sharpe Ratio), not just on total profit



Status Update (Week 12)

Accomplishments

- Refined Twitter/X Scraper
- Finished backlog documentation
- Obtain and filter insider transactions

Plans

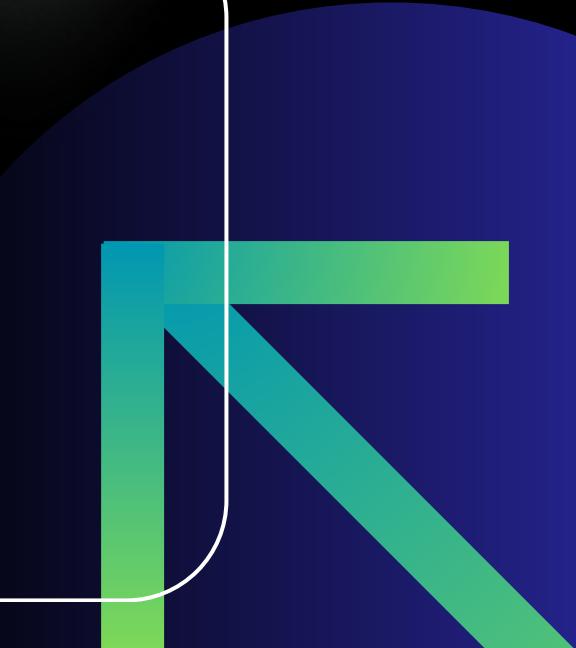
- Reach profitable status
- Find Historical Data
- Find Politician trades

Turmoil

- Reaching desired level of accuracy for sentiment analyzer

Assistance

- None :)



Thank You.
Questions?

