

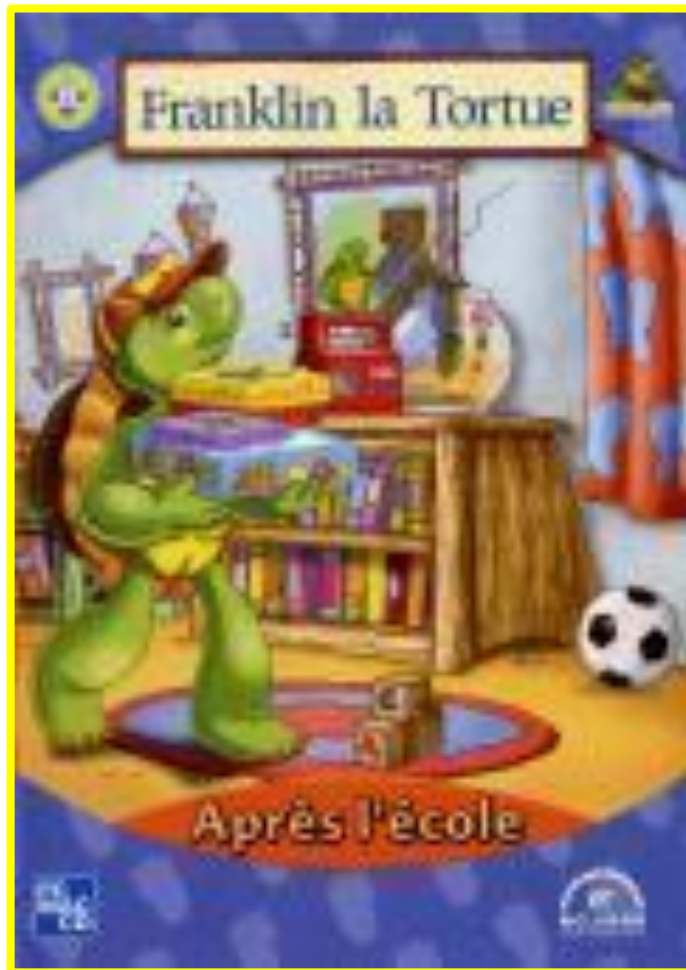


ALGORITHMIC TRADING SYSTEM FOR STOCKS

Group Two

Lidiya Maltseva, King Leung, Teddy McMillan, George Cruz, Franklin Bueno



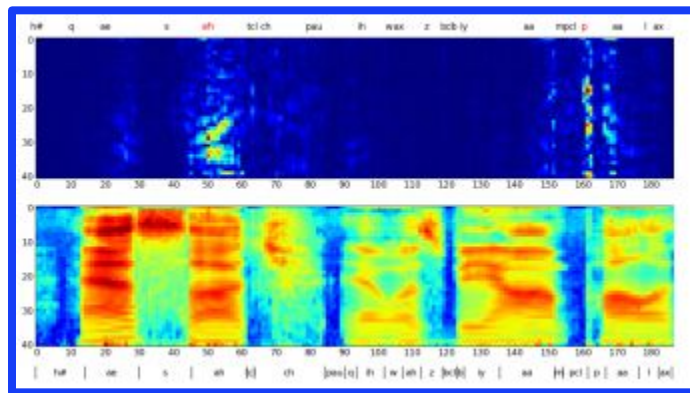


We've Almost Finished Our Boot Camp Together...



What Have We Learned?
What Can We Build?

We've Learned a Lot Together.



- Recurrent Neural Network (RNN) Modeling (Graphic-Input Sensitivity of RNN from University of Toronto Department of Computer Science)
- Long Short Term Memory (LSTM) Modeling
- Autoregressive Moving Average (ARMA) Modeling
- Autoregressive Integrated Moving Average (ARIMA) Modeling
- Generalized Autoregressive Conditional Heteroskedasticity (GARCH)
- Natural Language Processing (NLP), Valence Aware Dictionary and sEntiment Reasoner (VADER)
- Technical Indicators, Including Exponentially Weighted Moving Average (EWM)

What Can We Build from What We've Learned?



- Let's be ambitious and include as many lessons as possible...
- We could put everything we've learned together into a cohesive, unified, and original system.
- Let's build an automated system that
 - executes stock trades on a real account,
 - works off of a learning model from multiple signals that we code ourselves, and
 - leverages signals formed from NLP analysis and technical indicators.

Let's Start with One Specific Company



- Tesla (NASDAQ: TSLA) stock exhibits a great deal of volatility.
- Trading this stock offers the potential of great rewards, but also demands risk management and technical analysis.
- Plus, with NLP, we can analyze possible effects on Tesla stock price from the tweets of the Chief Executive Officer (CEO)...

ELON REEVE MUSK



- Mr. Elon Reeve Musk serves as the CEO and product architect of Tesla.
- Mr. Musk is (in)famous for his tweets.
- We could analyze and score his tweets with NLP and then form a signal for TSLA.
- Greatest Hits Tweets
 - “TESLA SHARE PRICE IS TOO HIGH IMO”
 - “I SELL ALMOST ALL PHYSICAL GOODS. WILL NOT OWN A HOUSE.”
 - “CORONAVIRUS PANIC IS STUPID”

How Do We Build This System Together?

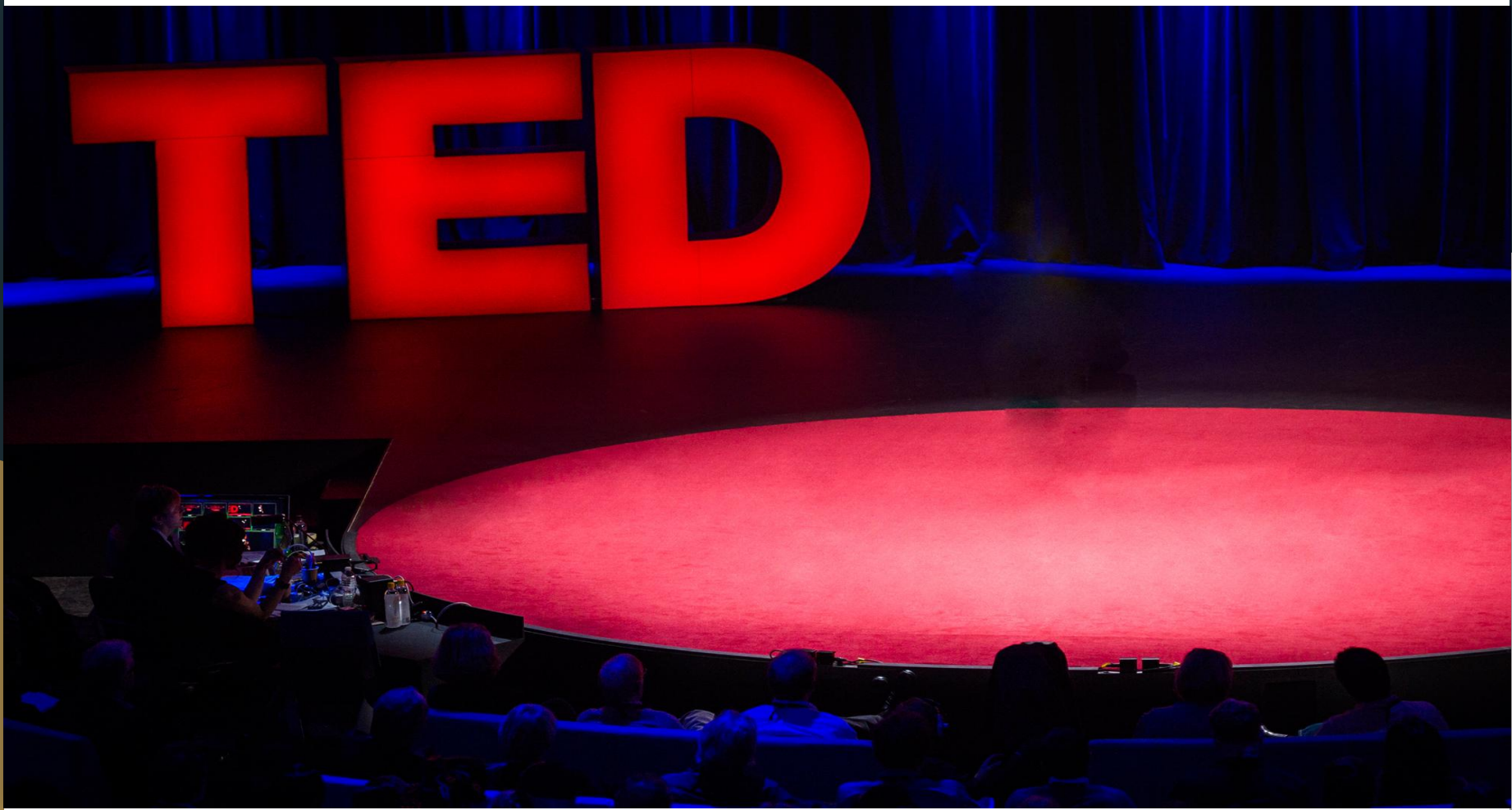


- We started by collecting data from Application Programming Interfaces (APIs), kaggle.com, and other sources, and then we cleaned and sorted the data.
- In addition, we used the alpaca.com API for the live trading component.
- We cooperated on Google Colabs, Jupyter Notebooks, and, of course, GitHub.

We Formed Our Plan.
Then, We Executed.



Teddy Continues.



MODEL and SIGNALS

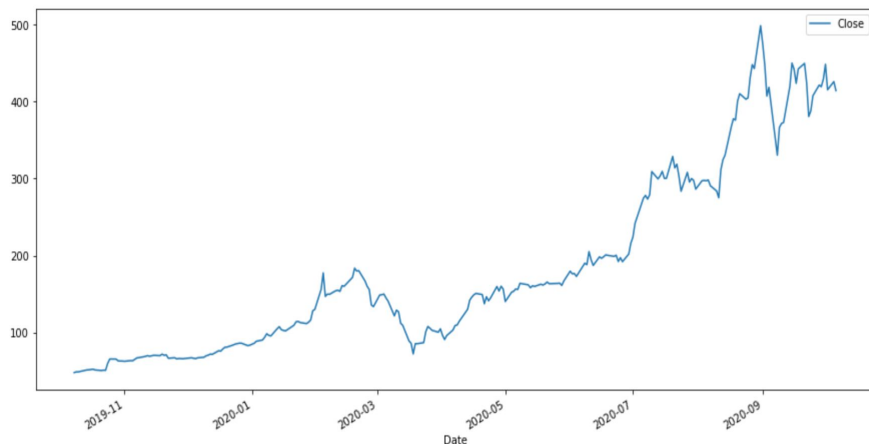
Models	Daily Closing Prices	EWM	Bollinger Band	Volatility	VADER Score "Twitter"	Google Trends
RNN	Yes	Yes	Yes	Yes	Yes	NaN
LSTM	Yes	NaN	NaN	NaN	Yes	Yes
ARMA	Yes	NaN	NaN	NaN	NaN	NaN
ARIMA	Yes	NaN	NaN	NaN	NaN	NaN
GARCH	Yes	NaN	NaN	NaN	NaN	NaN

Building a Trading Model

- Data Sources (ALPACA, Yahoo Finance, Google Trends, Kaggle)
- Frequency
 - Daily Closing Prices (Model Prediction and Trading Frequency)
 - Multiple Twitter per Day versus Zero Tweets per day/Weekend Tweets
 - Monthly Google Trends
- DateTime Formatting
- Data Concatenation
- Process of Constructing the Model

Performance of RNN Model vs TSLA Stock

TSLA Stock Movement over the Same Period

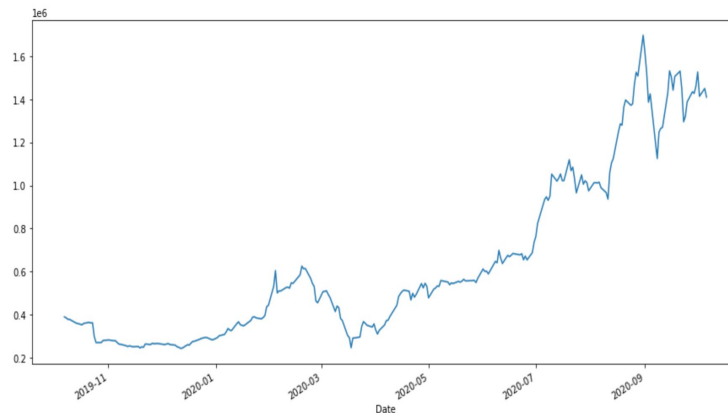


RNN Model Predictions

```
# Set initial capital allocation. (1 Signal EWM Windows = 50, 200)
initial_capital = 400000

# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))

<AxesSubplot: xlabel='Date'>
```



https://github.com/klleung72781/fintech_project_2/blob/main/TRADING_MODEL.ipynb



WHEN
GENIUS
FAILED

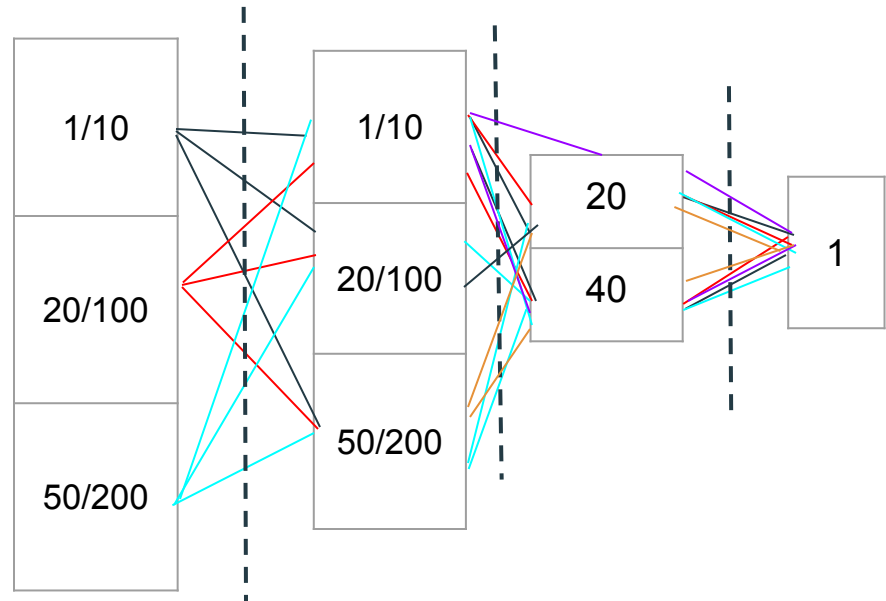
The image depicts a theater stage. Red curtains are pulled back on either side of a central spotlight. The spotlight illuminates a white rectangular sign on the wooden stage floor. The sign contains the text 'WHEN GENIUS FAILED' in a bold, sans-serif font. The word 'GENIUS' is on the second line, and the letter 'N' is highlighted in red. A yellow horizontal bar is positioned at the bottom of the sign, and a red arrow points downwards from the bottom of the red 'N' towards the floor.

Losing Money with Bad Models

We 27 Re-Ran a RNN Model Twenty Seven Times with Different Combination Signals to Find a Golden Combination

RNN Signal Combination Matrix

- Short/Long Term EWM Windows
 - Days: 1/10, 20/100, 50/200
- Short/Long Term Volatility
 - Days: 1/10, 20/100, 50/200
- Bollinger Band
 - Days: 20, 40
- Twitter Sentiment

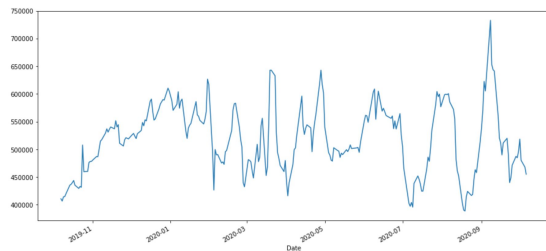



```
# Set initial capital allocation. (3 Signals EWM Windows = 1, 10, Bol-10, VOL-1, 10)
```

```
initial_capital = 400000
```

```
# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))
```

```
<AxesSubplot: xlabel='Date'>
```

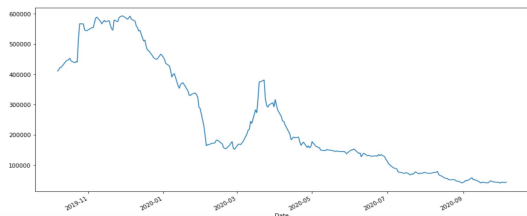


```
# Set initial capital allocation. (3 Signals EWM Windows = 20, 100, Bol-20, VOL-20, 100)
```

```
initial_capital = 400000
```

```
# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))
```

```
<AxesSubplot: xlabel='Date'>
```



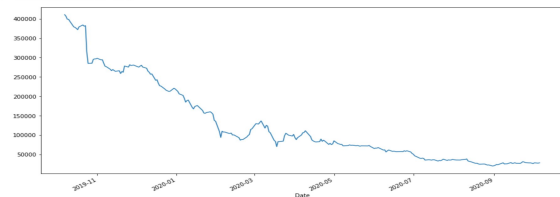
```
# Set initial capital allocation. (3 Signals EWM Windows = 1, 10, Bol-20)
```

```
## 'vol_trend_signal' and 'SEN' and 'bollinger_signal',
```

```
initial_capital = 400000
```

```
# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))
```

```
<AxesSubplot: xlabel='Date'>
```

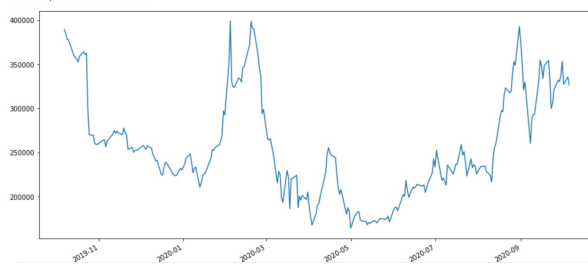


```
# Set initial capital allocation. (3 Signals EWM Windows = 50, 200, Bol-40, VOL-50, 200)
```

```
initial_capital = 400000
```

```
# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))
```

```
<AxesSubplot: xlabel='Date'>
```



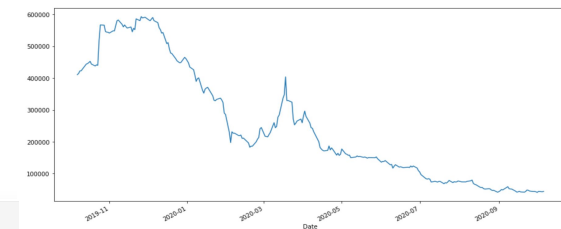
```
# Set initial capital allocation. (3 Signals EWM Windows = 10, 100)
```

```
## 'vol_trend_signal' and 'SEN' and 'bollinger_signal',
```

```
initial_capital = 400000
```

```
# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))
```

```
<AxesSubplot: xlabel='Date'>
```



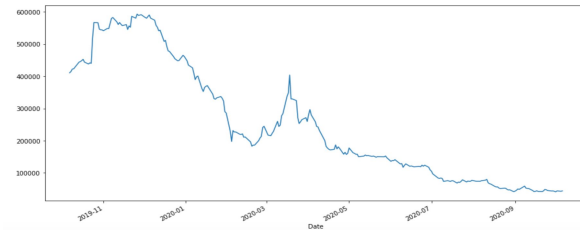
```
# Set initial capital allocation. (3 Signals EWM Windows = 20, 100, Bol-20)
```

```
## 'vol_trend_signal' and 'SEN'
```

```
initial_capital = 400000
```

```
# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))
```

```
<AxesSubplot: xlabel='Date'>
```

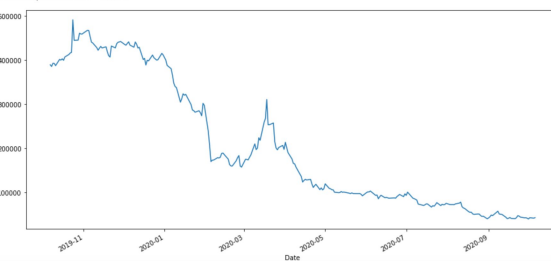


```
# Set initial capital allocation. "SEN"
```

```
initial_capital = 400000
```

```
# Plot cumulative return of model in terms of capital
cumulative_return_capital = initial_capital * (1 + (daily_df['Daily_return'] * daily_df['Predicted_Value'])).cumprod()
cumulative_return_capital.plot(figsize=(15,7))
```

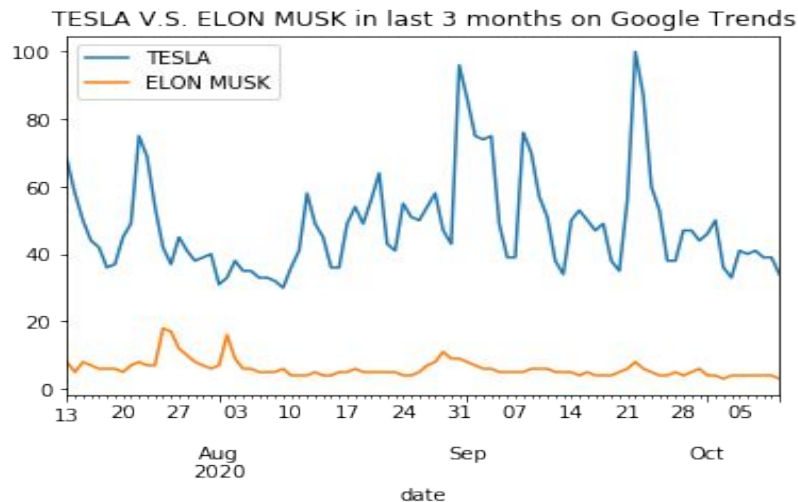
```
<AxesSubplot: xlabel='Date'>
```



→ 92% (25 out of 27) of RNN model combinations lost money on a \$400,000 investment.

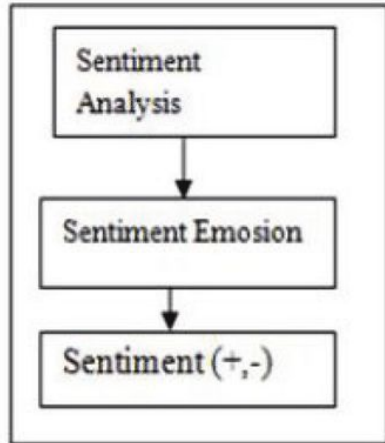
Sentiment Analysis

- Applied Twitter, and Google Trends as signals of the models.
- Google Trends inside of the LSTM model with no significance for the specific model on price fluctuations.
- We tested tweets on the RNN model.

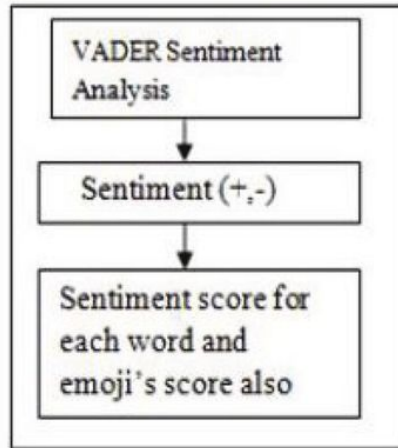


Sentiment Analysis

Sentiment Analysis



VADER Sentiment Analysis



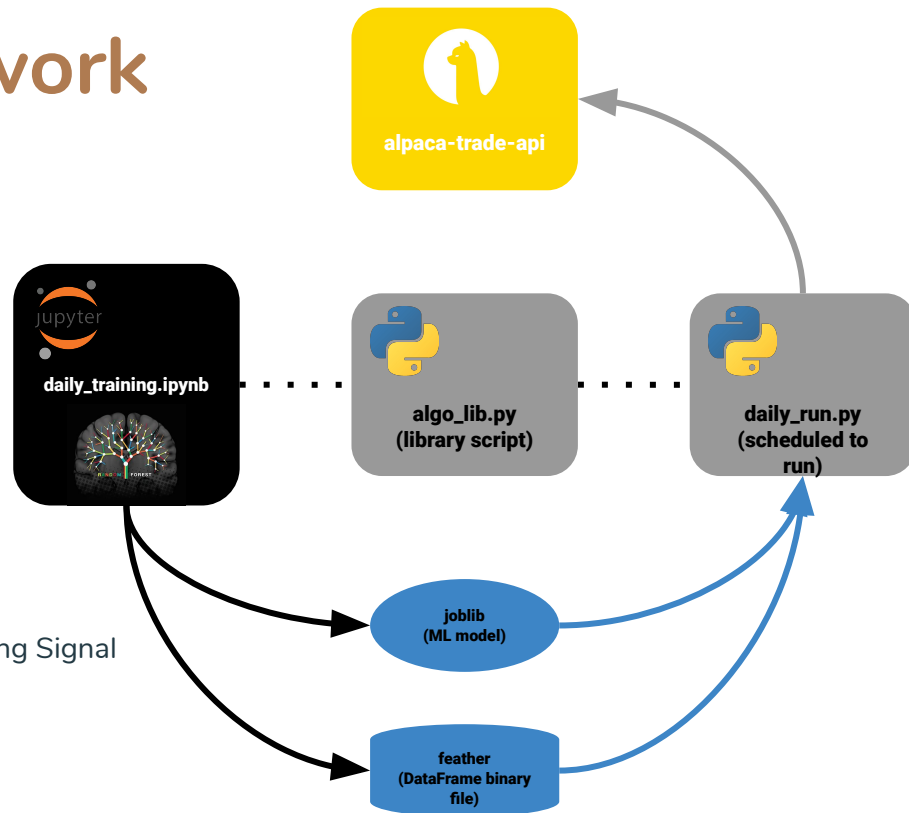
KING

Technical Stuff



King - Basic Framework

- **Jupyter Notebook (daily_training.ipynb)**
 - Trains/Retrains Model
 - Exports Price DataFrame
 - Accepts Other Tickers
- **Library script (algo_lib.py)**
 - Gathers Historical data via yfinance
 - Generates Crossover Signal
- **Trading script (daily_run.py)**
 - Gathers Latest Price
 - Append Data to DataFrame While Generating Signal
 - Import model to Make Prediction
 - API Call Buy, Sell, or Hold



King

DEMONSTRATION

What We Learned



- King-I learned that basic web-scraping and data collection is still the most important portion of an ML project.
- Frank-I learned that leveraging `df.droplevel()` can help isolate columns in an alpaca.com dataframe.
- Ted-I learned ... Models are only as good and the people constructing them.
- Lidiya-I learned not to get frustrated from data cleaning when dealing with Time Series Analysis!
- George-I learned that the goal of NLP is really to allow non programmers to obtain information about computer systems.

