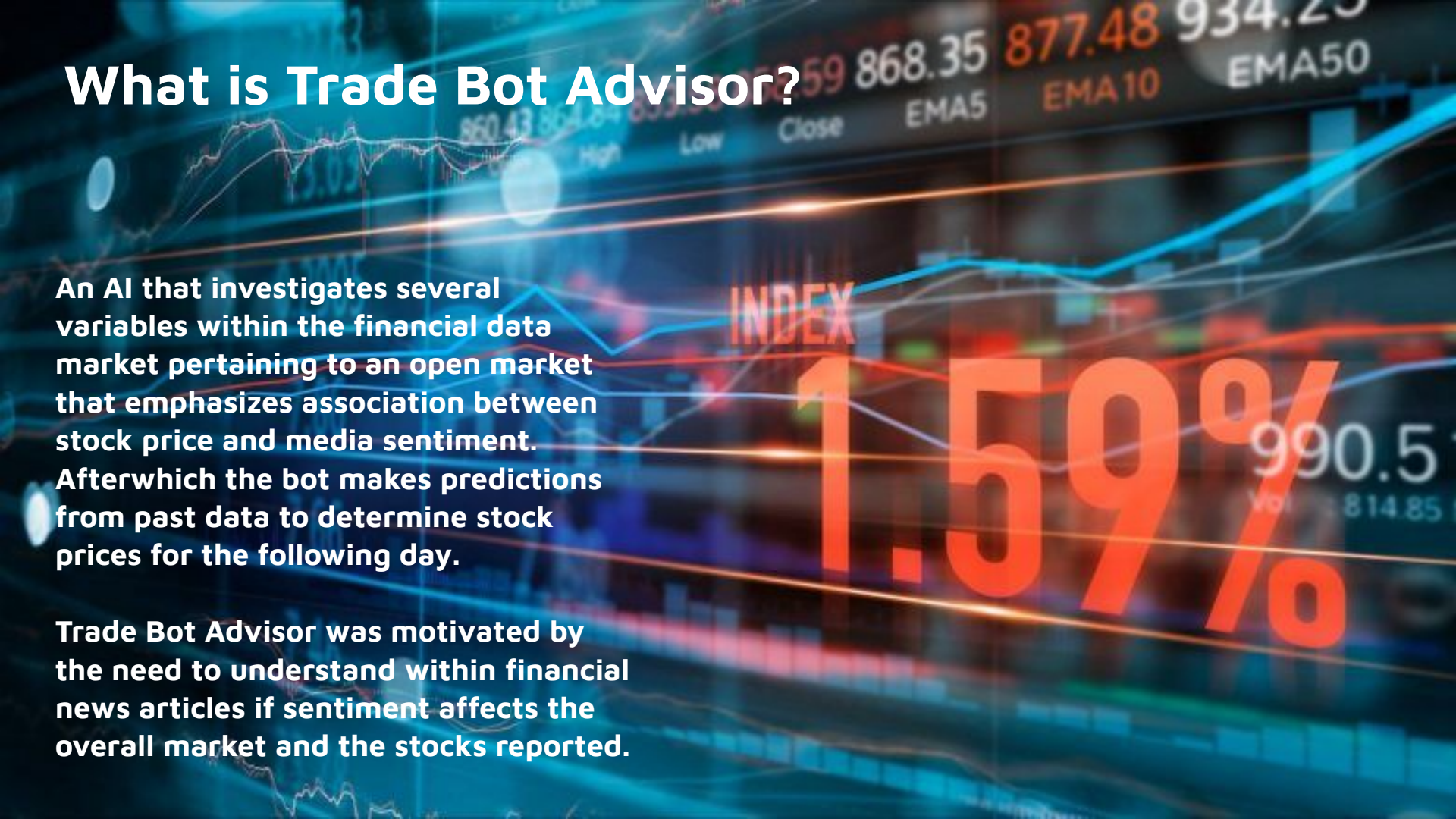
Trade Bot Advisor

Contributors:
Alpa Sheladia, Brian Withrow, Mischelle Massey, and Nathan Kleiner, Lee Redfearn

What is Trade Bot Advisor?



An AI that investigates several variables within the financial data market pertaining to an open market that emphasizes association between stock price and media sentiment. Afterwhich the bot makes predictions from past data to determine stock prices for the following day.

Trade Bot Advisor was motivated by the need to understand within financial news articles if sentiment affects the overall market and the stocks reported.

Data Source

Trade Bot Advisor utilizes customized class(Stockscrub) to gather data from the following features and their sources to make its predictions:

- **Sentiment Scores (GoogleNews API, Newspaper3k API, LM dictionary)**
- **VIX indicator (Quandl API)**
- **Stock price (AlphaVantage API)**
- **Volume (AlphaVantage API)**
- **S&P500 Beta Indicator (Yahoo finance API)**


News events and social media comments
Positive reactions
Negative reactions
Client Sentiment increases
Client Sentiment decreases
Shares are overbought
Shares are oversold
Price of shares fall
Price of shares rise

Cleaning Data

- We used multiple APIs such as Yahoo FinanceAPI, Google News API , Quandl API, and Alphadvantage API to pull necessary data. Thereafter, duplicate were removed using links (as links are unique identifiers).

- In order to achieve the desired historical data from the S&P 500, we used a Yahoo Finance API to extract specific relevant data.

- A unique class(StockScrub) was created to import, organize, and clean the data to organize into a dataframe using a customized function get_compile() to combine all data retrieved from the APIs.

- Finally, in order to compute sentiment scores we used Newspaper3k API to extract required content from the article links provided by Google API which was rather difficult using get.sentiment() function of the Stockscrub class. . Manual tokenization was required to produce articles without NLTK using regular expressions. LM Dictionary was used to clean various sentiment using specific financial stopwords.

Code to Clean Data

```
####################################     GET VIX FUNCTION       ########################################
def VIX_update():
    """Function pulls in VIX data from Quandl API"""                                    # Description
    result=quandl.get("CHRIS/CBOE_VX1", authtoken=quandl_key, collapse="daily")         # request pulling VIX data
    result.to_csv('Quandl_VIX_Data/VIX_DAILY.csv')                                      # saves VIX data to csv file
    print("Data successfully stored.")


####################################     GET SP500 FUNCTION       ########################################
def sp500_update():
    """Function pulls in S&P500 data from Quandl API"""                                 # Description
    spx = yf.Ticker("^GSPC")                                                            # ticker used to identify S&P500
    df = spx.history(period="max")                                                      # gets all available data
    df.to_csv("Yahoo_Finance_Data/SP500_DAILY.csv")                                     # saves S&P500 data to csv file



####################################       PROCESS TEXT         ########################################
def process_text(text):
    """Function recieves text, cleans it, and returns tokens"""
    regex = re.compile("[^a-zA-Z ]")                                                    # Making our regular expression
    clean_text = regex.sub(' ', text)  #consider hyphenated words                       # erasing nonalphabetic characters
    clean_text = re.findall('[a-zA-Z]+',clean_text)                                     # performing tokenization on clean text
    clean_text= [token.upper() for token in clean_text]                                 # makes tokens uppercase for dictionary
    return clean_text                                                                   # returning cleaned tokens
```

Apple's dataframe

```
[291]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8755915250>
```

Saving the Model for Future Use

Sentiment Analysis

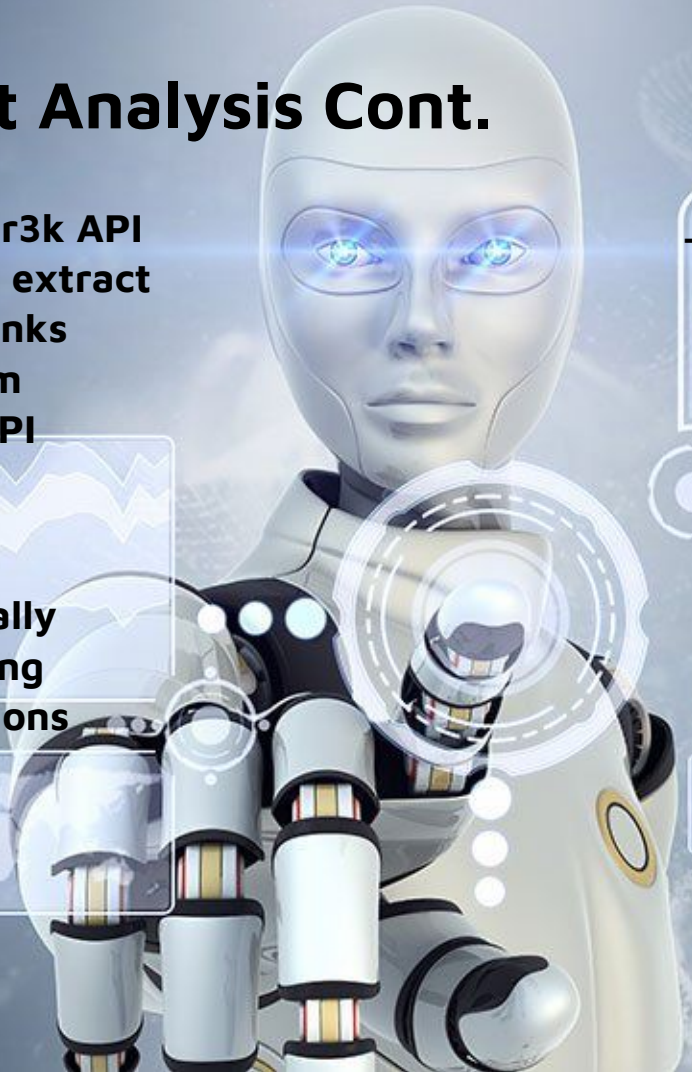
Terms in finance can vary in meaning depending on the context:

- **We used the LM Dictionary to provide a refined sentiment score since standard methods prove to be inefficient.**

  - **Ex: "decrease" in dividends or "decrease" in overall expenses**

Sentiment Analysis Cont.

- The Newspaper3k API was utilized to extract articles from links generated from GoogleNews API
- Tokenization of articles was executed manually without nltk using regular expressions



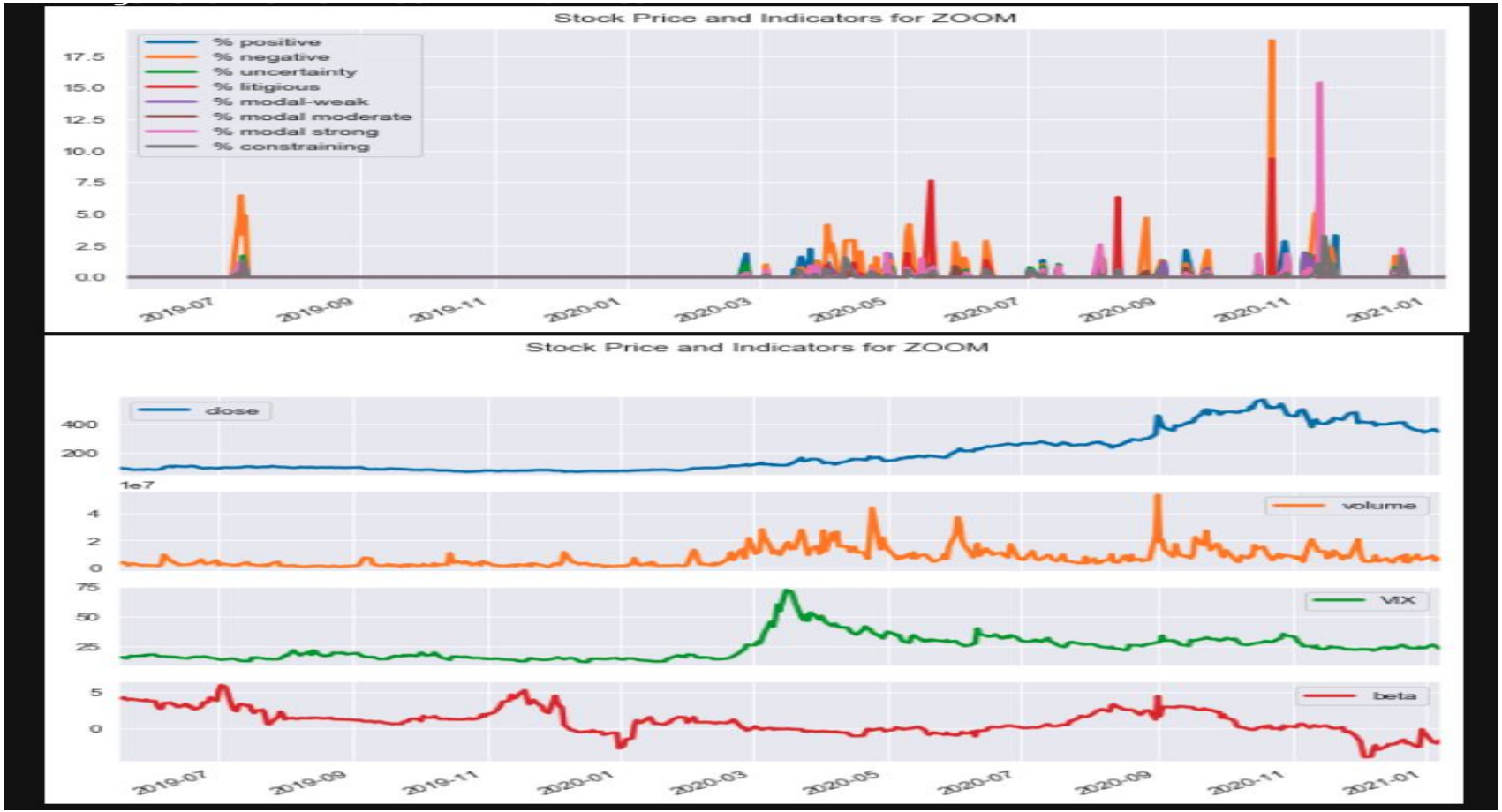
- LM used to compute various sentiment proportions (positive, negative, uncertainty,etc.)
- Results were compiled into a DataFrame object and stored for further use

[22]:
```
zoom=StockScrub(['ZM'])
zoom_df=zoom.data_compile(start='2016-01-01')
zoom_df.head()
```

[22]:

	close	volume	VIX	beta	number of words	% positive	% negative	% uncertainty	% litigious	% modal-weak	% modal moderate	% modal strong	% constraining	average word length	vocabulary
2019-05-17	89.98	3442549.0	16.02	4.050317	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-05-20	84.67	3666775.0	16.02	4.220943	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-05-21	85.44	2576024.0	15.05	3.945936	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-05-22	82.43	1596424.0	16.22	3.927525	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2019-05-23	78.76	2855970.0	17.35	4.032602	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Actual Sentiment scores for ZOOM

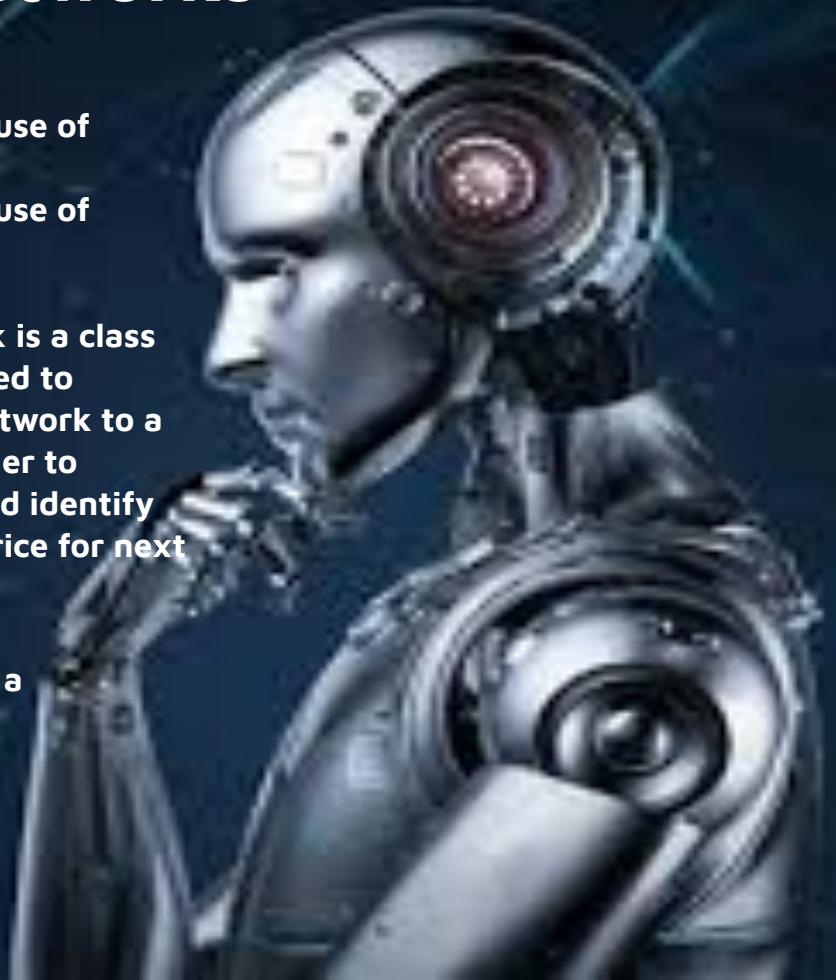


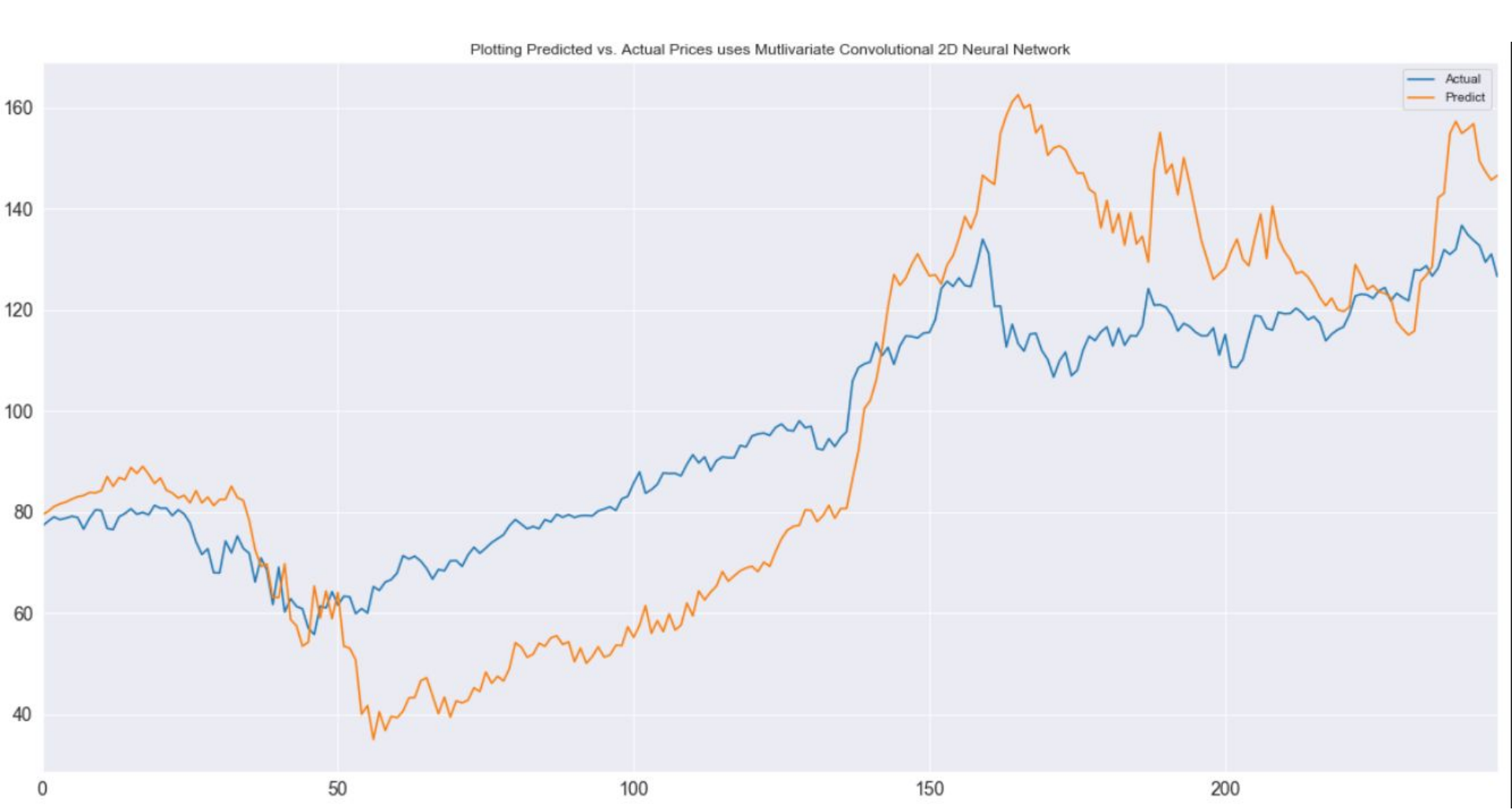
Convolutional Neural Networks

- All techniques all models from class only made use of univariate data (previous prices of asset)
- All techniques all models from class only made use of univariate data (previous prices of asset)

- In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. We repurposed the network to a multivariate signal to get readable results in order to incorporate the information from all features and identify the feature that was significant to predict the price for next day.

- Since our data is multivariate, it is analogous to a multivariate signal.

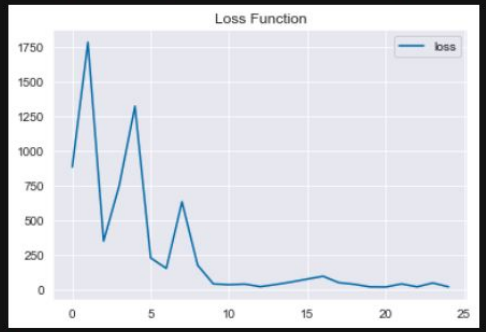



Plotting Predicted vs. Actual Prices uses Mutlivariate Convolutional 2D Neural Network
Actual
Predict
160
140
120
100
80
60
40
0
50
100
150
200

Machine Learning

- We trained CNN 2D model to fit all the assets

```
#predicted = model.predict(X_test)                                    # making predictions using test data
#predicted_prices = scaler.inverse_transform(predicted)               # Recover the original prices instead of the scaled versio
```



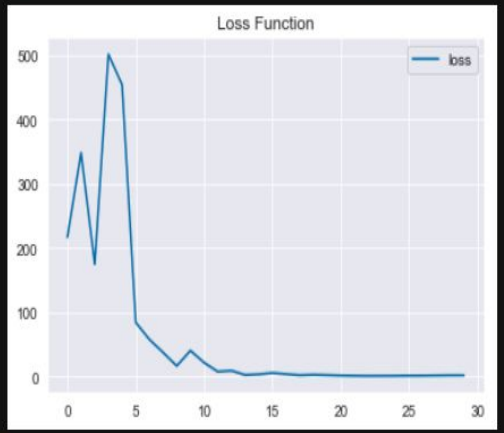
[290]:
```
# Create a DataFrame of Real and Predicted values
predictions = model.predict(data_test_wide_partial)
conv_acc_df = pd.DataFrame()
conv_acc_df['Actual'] = y_test[:,0]
conv_acc_df['Predict'] = predictions[:,0]
conv_acc_df.head(10)

model.evaluate(data_test_wide_partial, y_test)                        # evaluating performance of the model
```

```
7/7 [==============================] - 0s 8ms/step - loss: 54.2262
```

[290]:
```
54.22616195678711
```

[12]:
```
plt.plot(hist.history["loss"])
plt.title("Loss Function")
plt.legend(["loss"])
plt.show()
#predicted = model.predict(X_test)                                    # making predictions using test data
#predicted_prices = scaler.inverse_transform(predicted)               # Recover the original prices instead of the scaled version
```



[13]:
```
# Create a DataFrame of Real and Predicted values
predictions = model.predict(data_test_wide_partial)
conv_acc_df = pd.DataFrame()
conv_acc_df['Actual'] = y_test[:,0]
conv_acc_df['Predict'] = predictions[:,0]
conv_acc_df.head(10)

model.evaluate(data_test_wide_partial, y_test)                       # evaluating performance of the model
```

```
8/8 [==============================] - 0s 5ms/step - loss: 11.1579
```

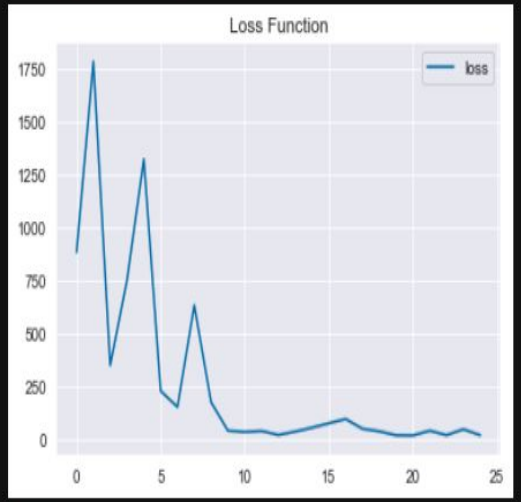
[13]:
```
11.157930374145508
```


Plotting Predicted vs. Actual Prices uses Mutlivariate Convolutional 2D Neural Network: PFE
Actual
Predict
42.5
40.0
37.5
35.0
32.5
30.0
27.5
25.0
0
50
100
150
200

[291]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8755915250>

Saving the Model for Future Use

```
#predicted = model.predict(X_test)                                    # making predictions using test data
#predicted_prices = scaler.inverse_transform(predicted)               # Recover the original prices instead of the scaled versio
```



[290]:
```
# Create a DataFrame of Real and Predicted values
predictions = model.predict(data_test_wide_partial)
conv_acc_df = pd.DataFrame()
conv_acc_df['Actual'] = y_test[:,0]
conv_acc_df['Predict'] = predictions[:,0]
conv_acc_df.head(10)

model.evaluate(data_test_wide_partial, y_test)                        # evaluating performance of the model
```

```
7/7 [==============================] - 0s 8ms/step - loss: 54.2262
```

[290]:
```
54.22616195678711
```

Final Results

What information or problem was resolved for the User?

- User will see reliable accurate information regarding the acquisition of stock and use the data to determine whether a User should buy, sell, or hold.

How much does the media sentiment influence the price of stocks?

- Extreme events led to biased and unbiased media coverage which affected stock prices based on the financial sentiment analysis.

How accurate were the predictions?

- With sentiment feature, mean test error rate of Apple's was 74.92, Pfizer;s was 20.43, American Airline's was 530.42, Exxon's was 565.90 and Zoom's was 44518.51

What event occurred in the media caused these signals.

- During the time period that was discussed the main event that occurred causing shifts in the data was COVID-19 pandemic

Our Discovery

- What are your findings of the news media's contribution in the stock market volatility over past three-four years?

We used several features and academic research to figure out that media plays crucial role in market volatility. We redefine sentiment and multivariate to extract data from news articles to see the affect on future prices of stocks.

- Is this a supervised or unsupervised learning task? what are data features and what is the target variable?

This is supervised learning task since the data is provided with target variable is next day's price . Data features include parameters, prices,VIX, Sentiment, Closing Price and Volume.

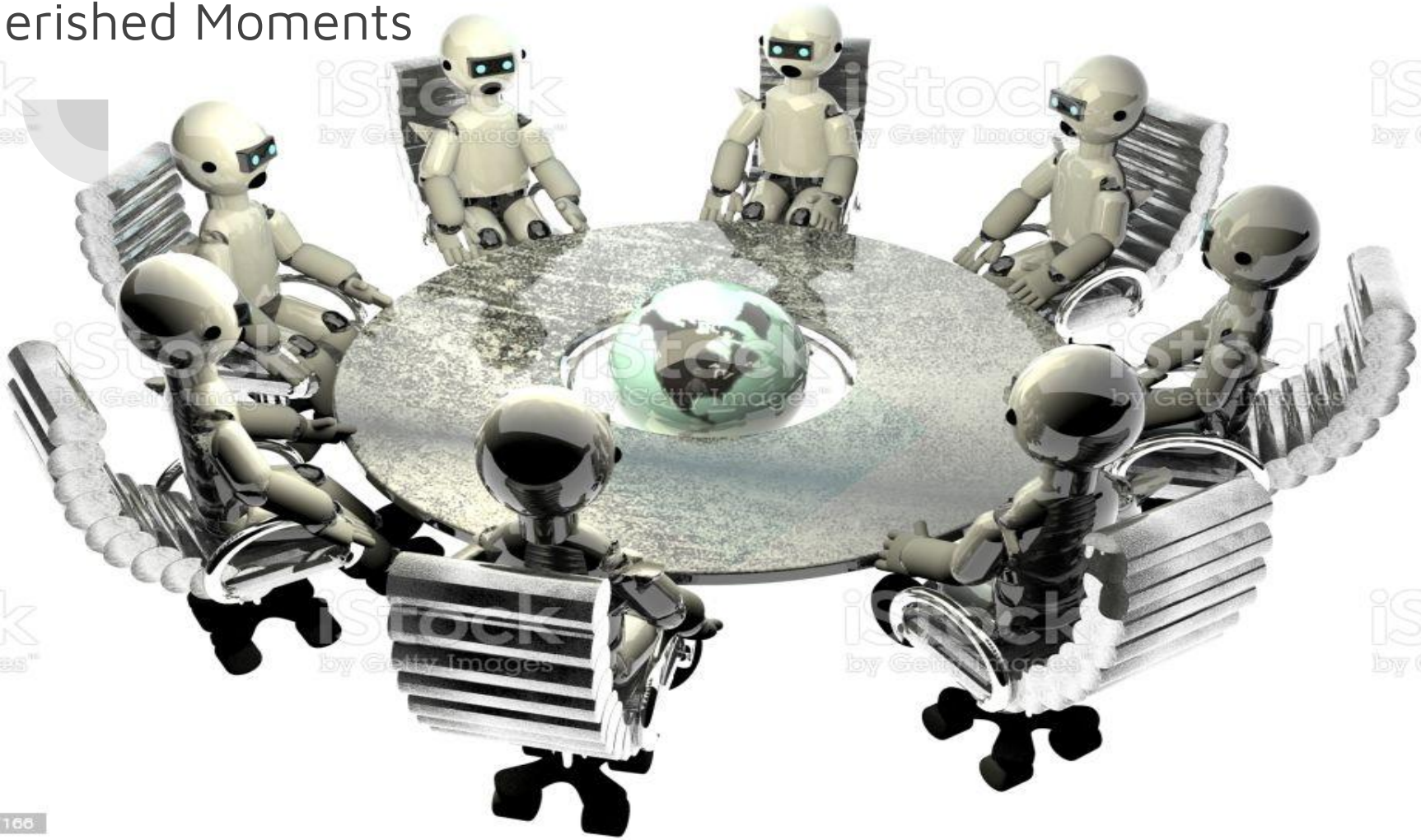
- How did the Machine learning play a role in it?

In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics. We repurposed it to multivarious signal to get readable result in order to incorporate the information from all features and identifies the features significant to predict the price for the next day.

- Which specific stock most affected by news media?

Zoom was doing well during Covid-19 events till On Aug. 20th 2020. Zoom crashed tremendously .

Cherished Moments



Questions and Discussion