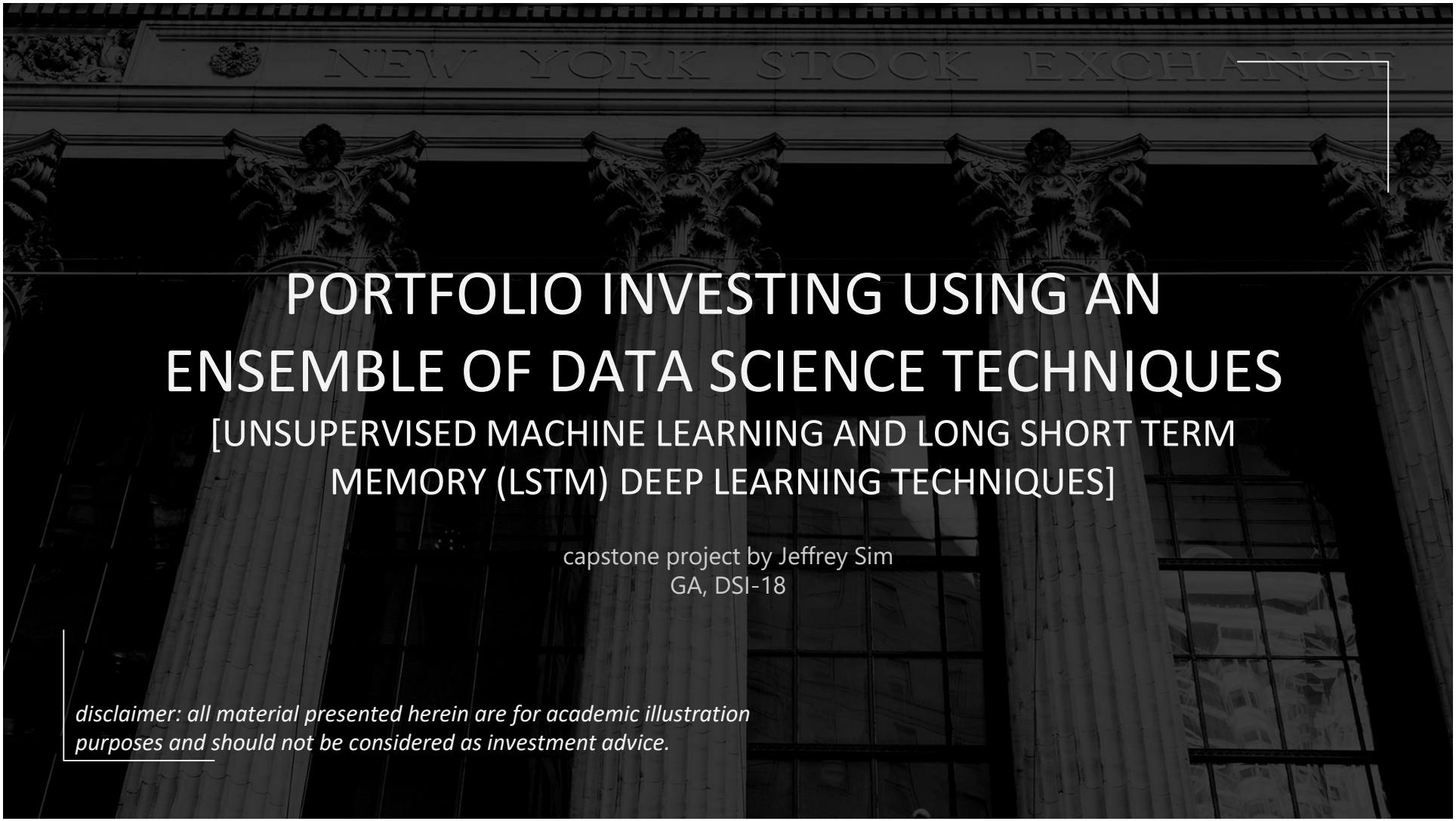


PORTFOLIO INVESTING USING AN ENSEMBLE OF DATA SCIENCE TECHNIQUES

[UNSUPERVISED MACHINE LEARNING AND LONG SHORT TERM MEMORY (LSTM) DEEP LEARNING TECHNIQUES]

capstone project by Jeffrey Sim
GA, DSI-18

disclaimer: all material presented herein are for academic illustration purposes and should not be considered as investment advice.

Agenda

00 Introduction

11 Problem Statement

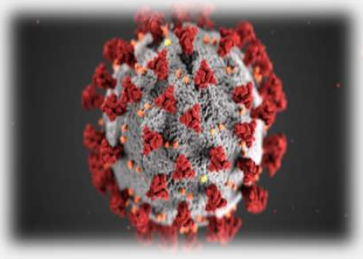
22 Data Science Methdology Overview

33 Data Collection EDA

44 Portfolio Building

55 Repeatable Prediction Model

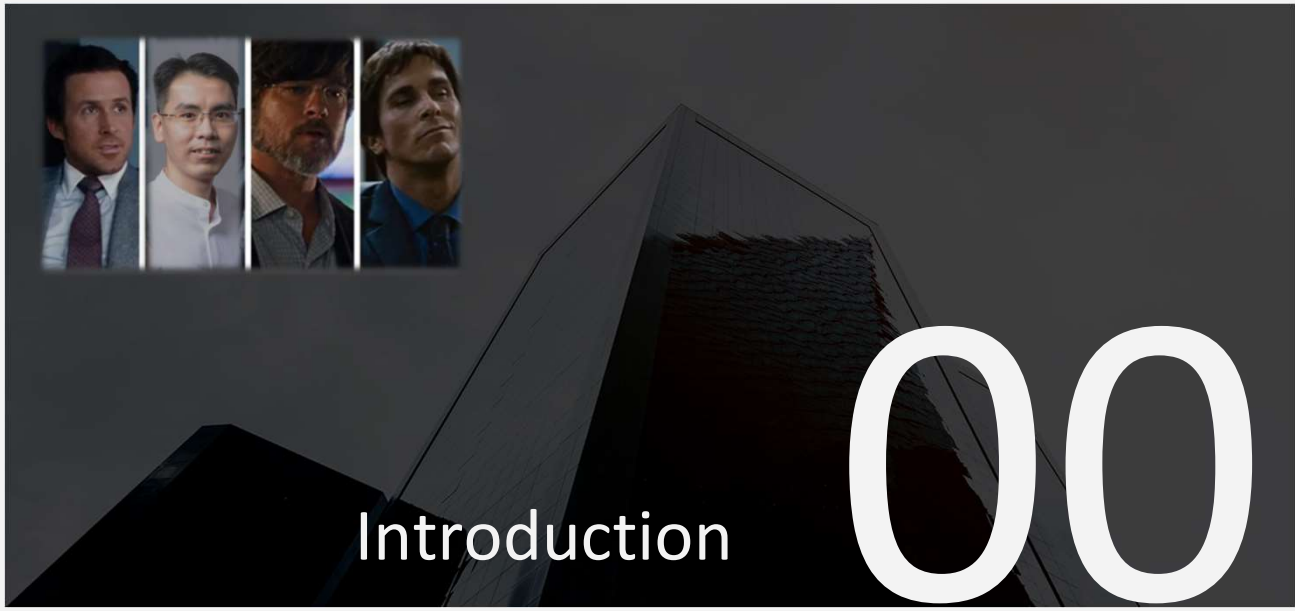
66 Performance Conclusion Recommendation



random 42

“every crisis brings opportunities, and this is no lesser in the world of investing”



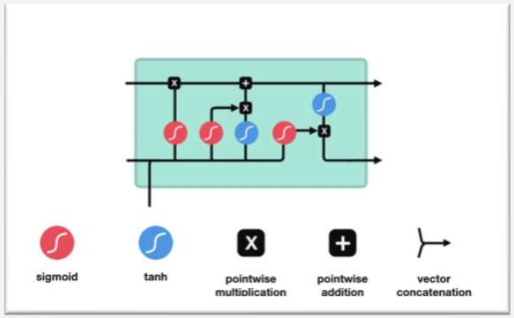
Introduction 00

Traditional Way

"by the time you hear it it is too late"

Keras pandas scikit learn TensorFlow

Beautiful Soup NumPy pandas Requests repl.it



Data Science Techniques

11 Problem Statements

1. **Too time consuming to collect and analyze information to build a viable portfolio**
2. **Not having a reliable and repeatable way to predict stock price movement**
3. **Not having an automated trading strategy that wins consistently**

Metrics of success

Our modeled portfolios should achieve:

1. **Above 50% of right calls**
2. **Positive returns higher than bank interest (@ 1.5% per yr)**

Solution Methodology Overview

22

1

Data Pipeline & Collection

- **Web scrapping** with BeautifulSoup, Regex, JSON, yfinance API, etc.
- Fundamentals indicators, Analyst's Recommendations, News articles, Historical prices
- Valence Aware Dictionary and Sentiment Reasoner (**VADER**)

2

Unsupervised Machine Learning

EDA, Domain Knowledge & Kmeans Clustering

- **EDA** to create **"Human portfolio"**
- **Standard Scaler pre-processing**
- **Kmeans** clustering to create **"AI portfolio"**

Ensemble of techniques

3

LSTM Modeling & Tuning

- **MinMax Scaler** pre-processing
- **Generic LSTM model** with fixed layer and nodes for all (i.e. un-tune baseline)
- Keras RandomSearch **Tuned LSTM models** for each company (i.e. tuned portfolios)
- Trained with 20yrs data

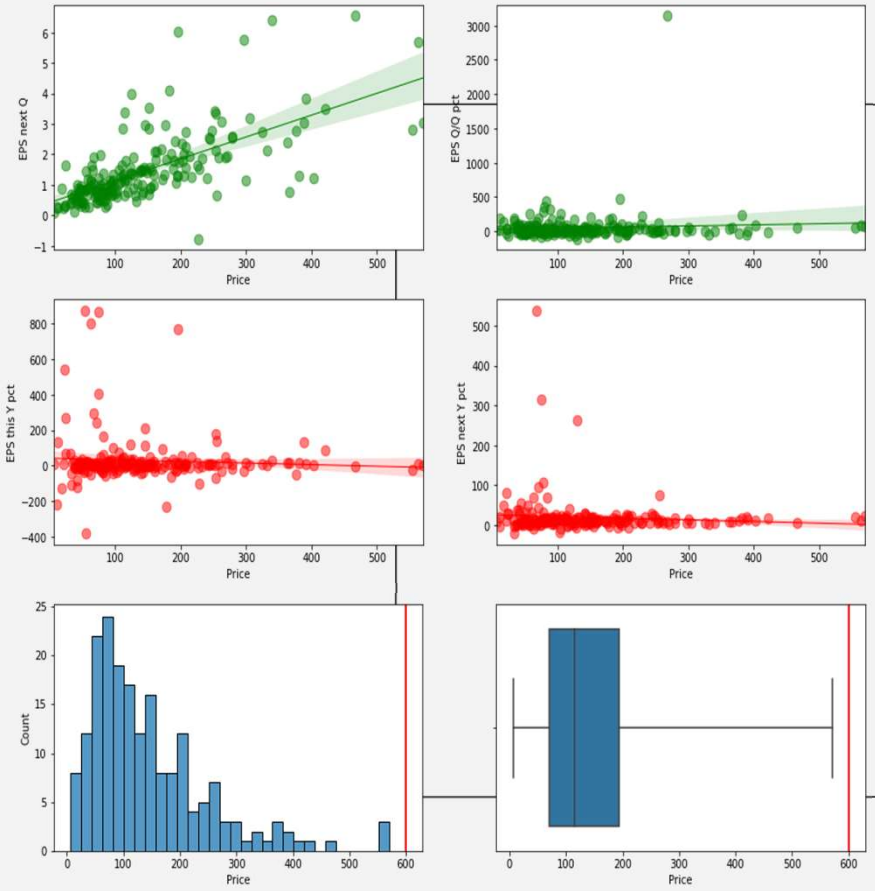
Deep Machine Learning Neural Networks

4

Trading Strategy Back Testing

- **Simple Buy low, Sell High, No Short**
- **Trading Account Class & Psuedo Trading Bot**
- Long-Term & Short-Term Performance

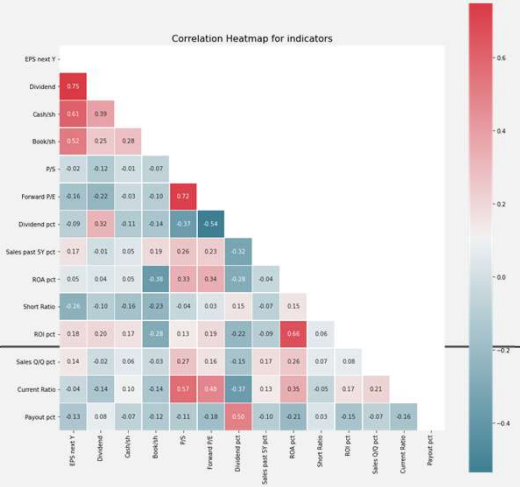

3
Data
Collection
& EDA
yfinance
finviz
yahoo! finance
MORNINGSTAR
MarketWatch
ALPHA VANTAGE
Google Finance
WALL ST bets



"snapshots of company's financial performance and how the market views its performance"

fundamental indicators

Market Capitalization, Volume and Price
Price-to-something (P/E, P/B, P/FCF)
Earning per Share
Insider, Institutional and shares float/outstd
Technical Indicators (SMA, RSI)
Dividends
Profits, Earnings, Debts
Financial ratios (ROE, ROI, ROA)

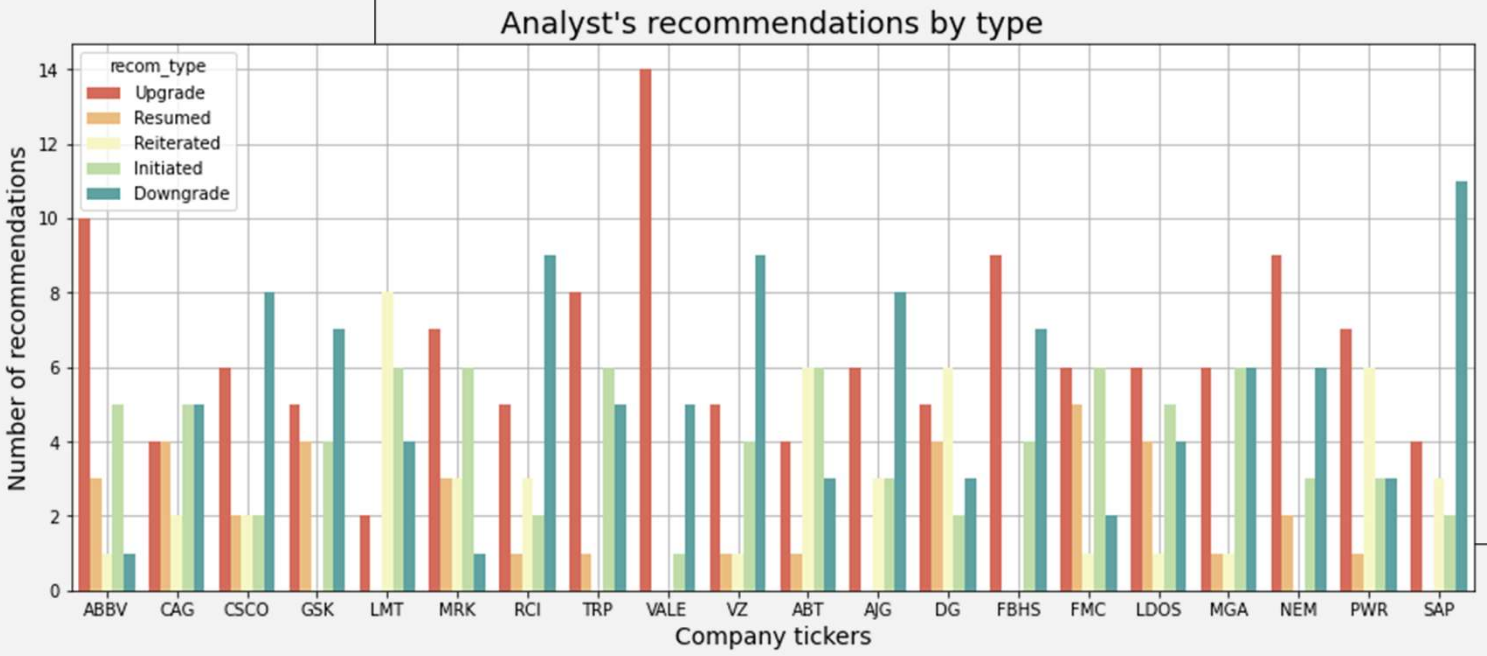


DATASET ONE
for Unsupervised Machine Learning Automated Portfolio Building

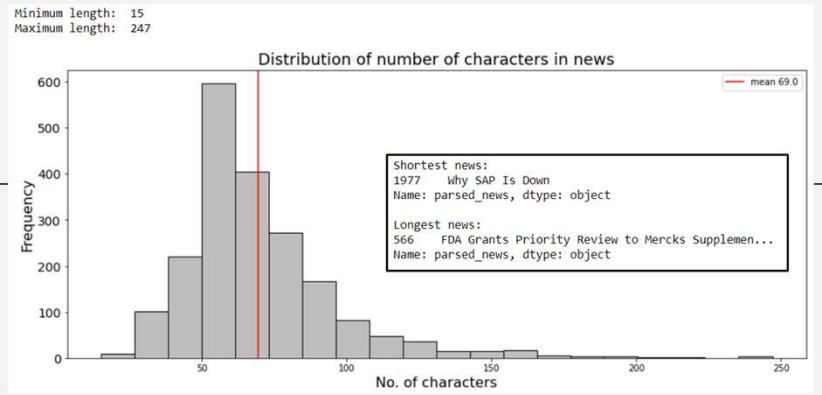
analyst's recommendations

"opinions from financial professionals who had spend a lot of time studying the financial reports a company"

*400 recommendations for 20 companies spanning from **2015-10-23 to 2021-01-21***



5 - Upgrade
4 - Resumed
3 - Re-interated
2 - Initiated
1 - Downgrade



VADER (Valence Aware Dictionary and Sentiment Reasoner)

2000 articles for 20 companies spanning from 2020-01-03 to 2021-01-22

positive sentiment: compound score >= 0.05
neutral sentiment: (compound score > -0.05 and < 0.05)
negative sentiment: compound score <= -0.05

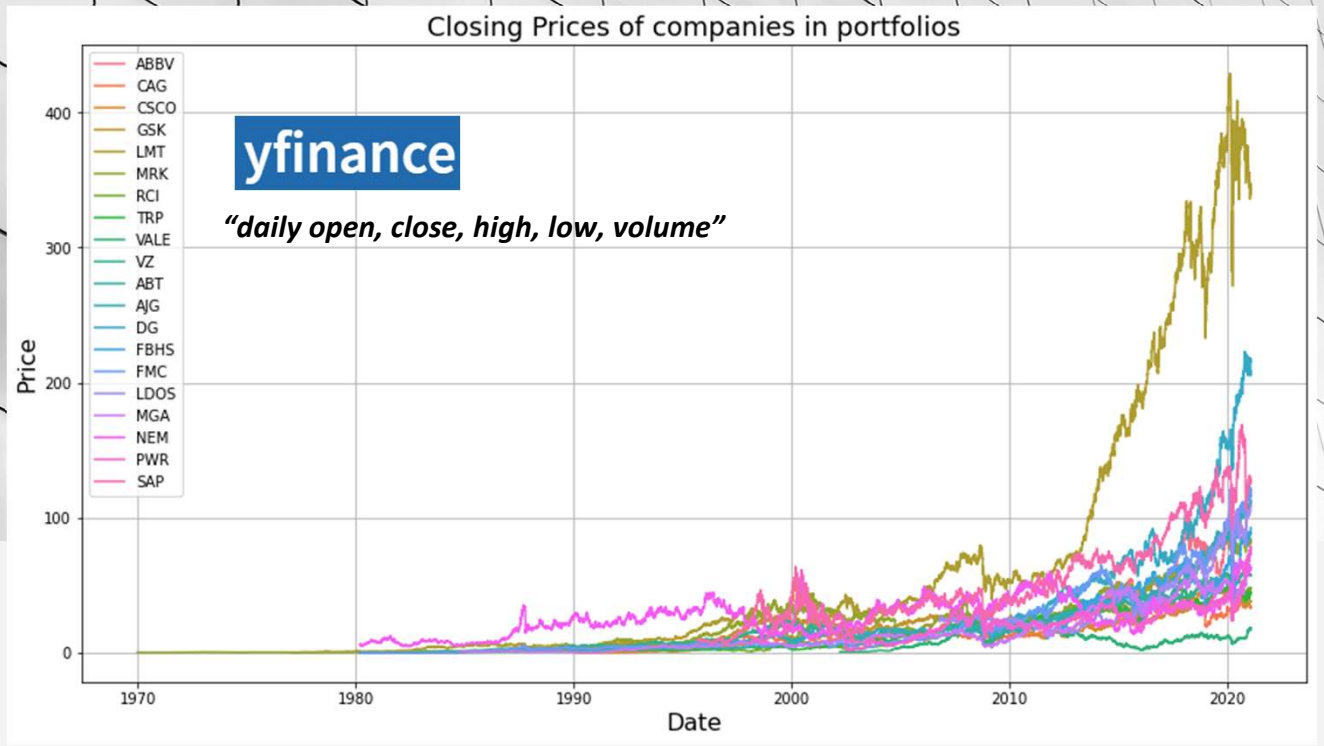
	ticker	date	time	parsed_news	neg	neu	pos	compound
494	LMT	2020-12-08	04:03PM	Boeing, Kratos To Build Skyborg Drones For Air...	0.00	1.000	0.000	0.0000
1707	NEM	2021-01-05	08:55AM	Newmont to Provide Reserves and Exploration Up...	0.00	0.759	0.241	0.2263
1122	AJG	2020-12-09	09:24AM	Arthur J. Gallagher (AJG) Announces Cool Insur...	0.00	0.753	0.247	0.3182
996	VZ	2020-12-18	05:58PM	5G Auction Soars to $34 Billion With Verizon L...	0.00	1.000	0.000	0.0000
415	LMT	2021-01-12	03:03PM	More companies pause political donations follo...	0.32	0.680	0.000	-0.5106

	ticker	date	compound	Interpretation
49	NEM	2021-01-05	0.226300	Positive
17	ABBV	2020-12-15	0.051033	Positive
17	VZ	2021-01-09	0.000000	Neutral
29	SAP	2020-12-31	0.102700	Positive
26	ABT	2020-12-05	-0.381800	Negative

"noise from the market that can swing prices"

News articles & sentimental analysis

historical prices & volumes



yfinance

"daily open, close, high, low, volume"

TA-Lib

Momentum indicators
- SMA20 –Simple Moving Average
- RSI14 Relative Strength Index
- MOM5 – Momentum

Volume indicators
- OBV – On-balance Volume

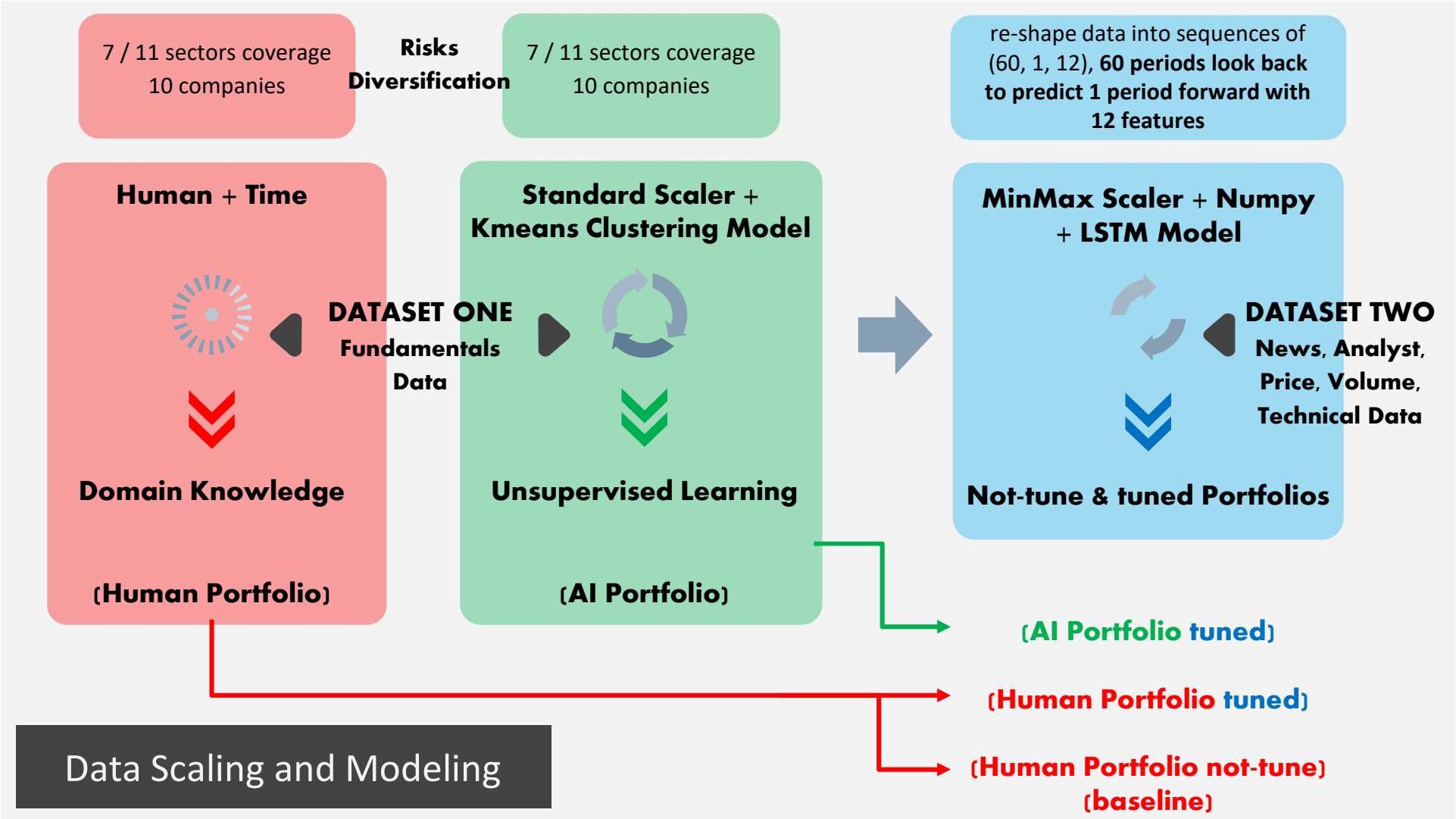
Volatility indicators
- ATR14 – Average True Range

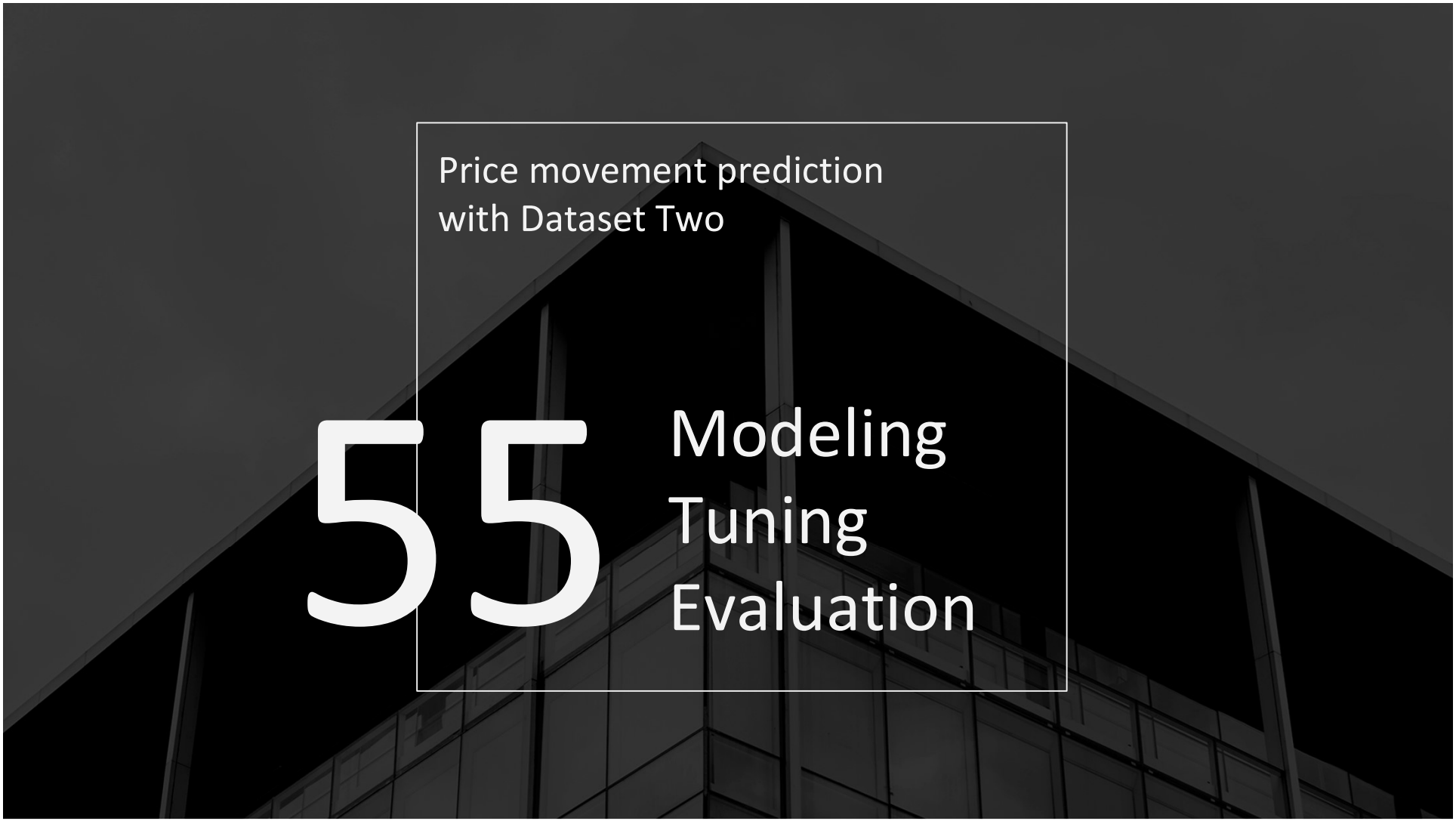
DATASET TWO

for Deep Machine Learning (LSTM) Modeling for a repeatable price movement prediction model

Final merged data:
- *News, Analysts, Historical Prices, Volumes, Technical Indicators*
- *20 companies*
- *more than **150K+ rows***
- ***13 features***


Portfolio
Building
Clustering model
with Dataset One and
Data Pre-processing
44


7 / 11 sectors coverage
10 companies
Risks
Diversification
7 / 11 sectors coverage
10 companies
re-shape data into sequences of (60, 1, 12), 60 periods look back to predict 1 period forward with 12 features
Human + Time
DATASET ONE
Fundamentals Data
Standard Scaler + Kmeans Clustering Model
MinMax Scaler + Numpy + LSTM Model
DATASET TWO
News, Analyst, Price, Volume, Technical Data
Domain Knowledge
Unsupervised Learning
Not-tune & tuned Portfolios
(Human Portfolio)
(AI Portfolio)
(AI Portfolio tuned)
(Human Portfolio tuned)
(Human Portfolio not-tune)
(baseline)
Data Scaling and Modeling


Price movement prediction
with Dataset Two
55
Modeling
Tuning
Evaluation

Modeling

```python
# instantiation
lstm_model = Sequential()

# set activation
activ = 'tanh'

# adding the input and first LSTM layer
lstm_model.add(LSTM(100, activation=activ, return_sequences=True,
                    input_shape=(n_per_in, n_features)))

# # final hidden layer w/o return_sequences
lstm_model.add(LSTM(100, activation=activ))

# output layer
lstm_model.add(Dense(n_per_out))
```

executed in 492ms, finished 11:39:34 2021-01-24

```python
lstm_model.summary()
```

executed in 15ms, finished 11:39:34 2021-01-24

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 60, 100)           45200
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               80400
_________________________________________________________________
dense (Dense)                (None, 1)                 101
=================================================================
Total params: 125,701
Trainable params: 125,701
Non-trainable params: 0
```

Generic LSTM (not-tune)

trials results

```python
def build_model(hp):

    # instantiation
    lstm_model = Sequential()

    # set activation
    activ = 'tanh'

    # adding the input and first LSTM layer
    lstm_model.add(LSTM(units=hp.Int("input_units", min_value=64, max_value=256, step=32),
                        activation=activ, return_sequences=True,
                        input_shape=(n_per_in, n_features)))

    # hidden layers with Dropout regularisation
    for i in range(hp.Int("n_layers", 1, 2)):
        lstm_model.add(LSTM(units=hp.Int(f"hidden_{i}_units", min_value=64, max_value=256, step=32),
                            activation=activ, return_sequences=True))
        lstm_model.add(Dropout(0.2))

    # final hidden layer w/o return_sequences
    lstm_model.add(LSTM(units=hp.Int("final_units", min_value=64, max_value=256, step=32),
                        activation=activ))

    # output layer
    lstm_model.add(Dense(n_per_out))

    # Compiling the RNN
    lstm_model.compile(optimizer=Adam(lr=0.01), loss='mse')

    return lstm_model
```

```python
#initializing randomsearch
tuner = RandomSearch(build_model,
                     objective = 'val_loss',
                     max_trials = 3,   # how many random permutations do you want
                     executions_per_trial = 2, # how many time to train for each permutations
                     directory = LOG_DIR,
                     project_name = 'SEEDED(42) RandomSearch')
```

```python
# display the search results and params
tuner.results_summary()
```

executed in 15ms, finished 00:59:18 2021-01-25

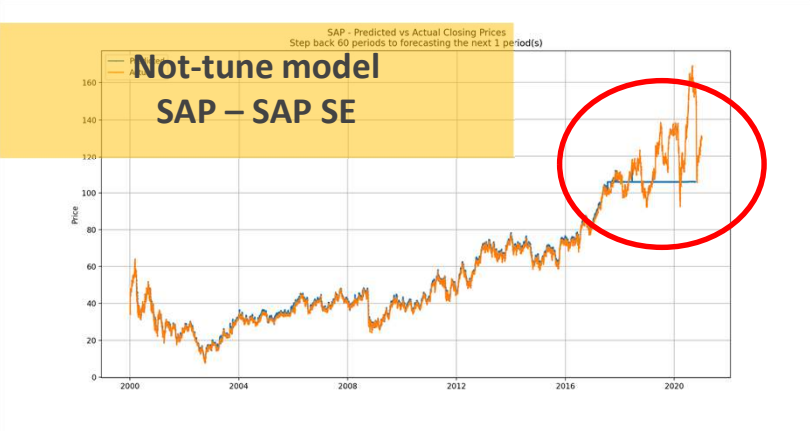
```
Results summary
Results in keras_tuner_logs/RCI_1611507447\s
Showing 10 best trials
Objective(name='val_loss', direction='min')
Trial summary
Hyperparameters:
input_units: 160
n_layers: 1
hidden_0_units: 160
final_units: 64
hidden_1_units: 160
Score: 0.0003206969955172837
Trial summary
Hyperparameters:
input_units: 192
n_layers: 2
hidden_0_units: 64
final_units: 224
hidden_1_units: 64
Score: 0.0032147193560376763
Trial summary
Hyperparameters:
input_units: 128
n_layers: 1
hidden_0_units: 160
final_units: 128
hidden_1_units: 192
Score: 0.17871791124343872
```

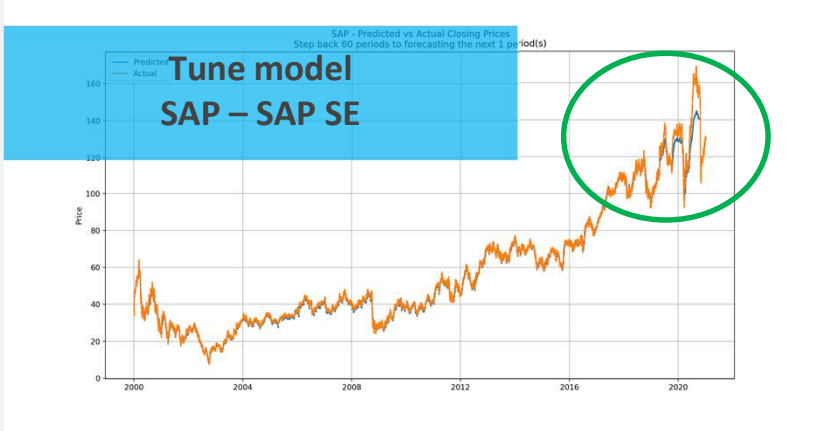
Dynamic LSTM (tuned)

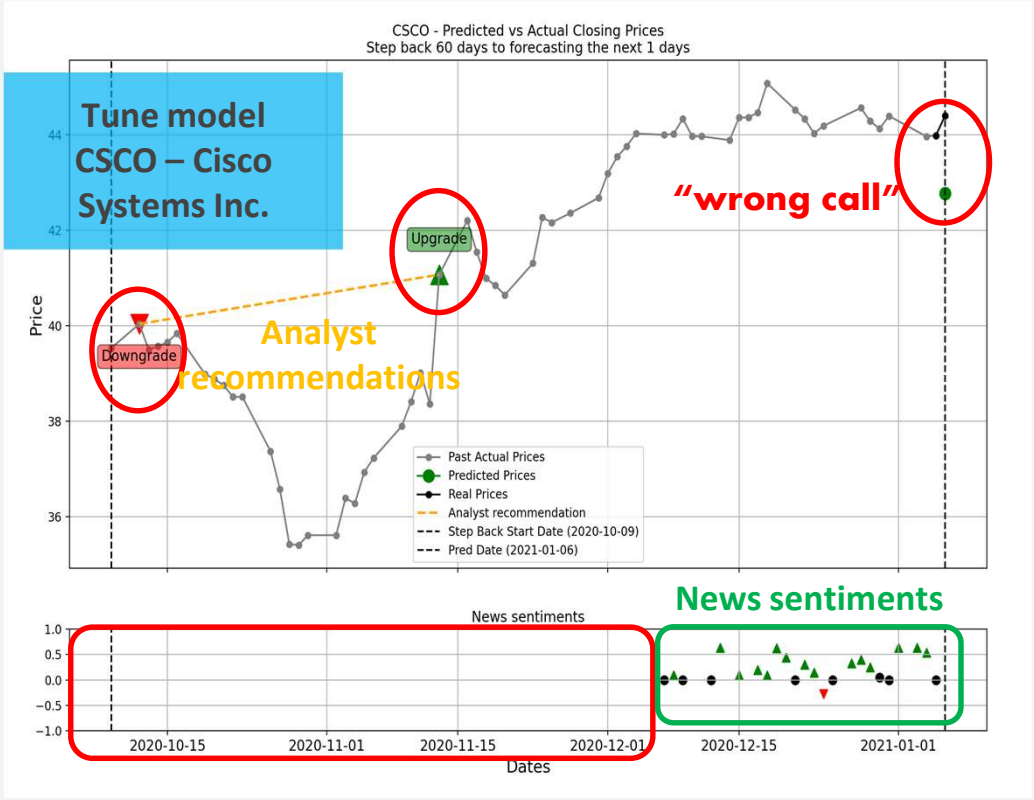
TensorFlow Keras

Evaluation

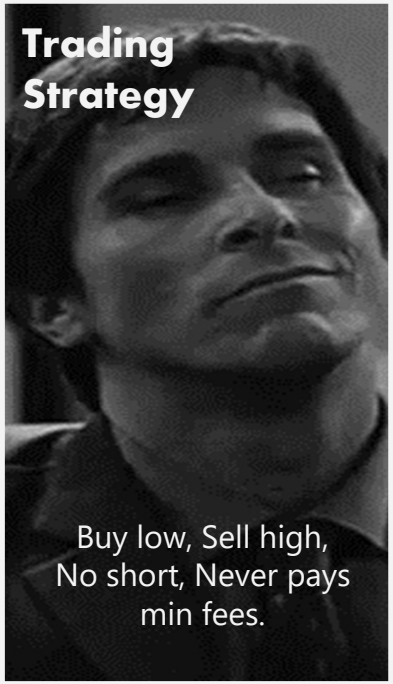
Examples of prediction results

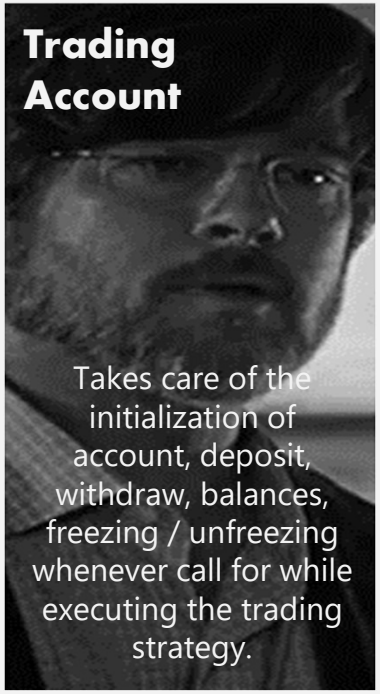






Based on DBS OET US fees
Commission = 0.15% on transaction amount
Min. Fees = US 18 per transaction

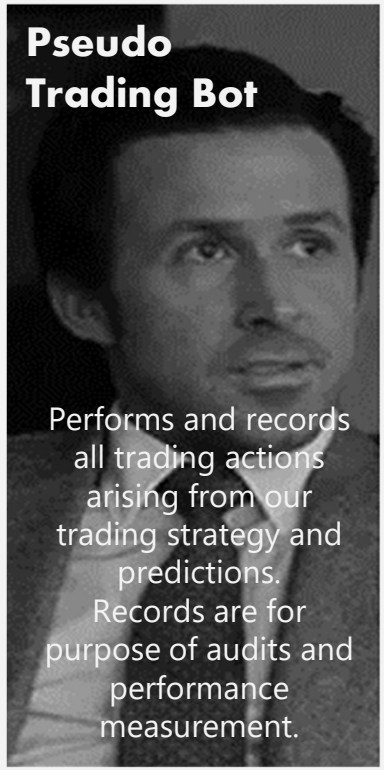




Account Starting Balance = 50K each company (i.e 500K each portfolio)

Min. predicted percentage gain = 5 %

Meet Our Team



Short Selling – selling what you do not have at a high price, hoping to buy the require quantity at a lower price (very high risk)

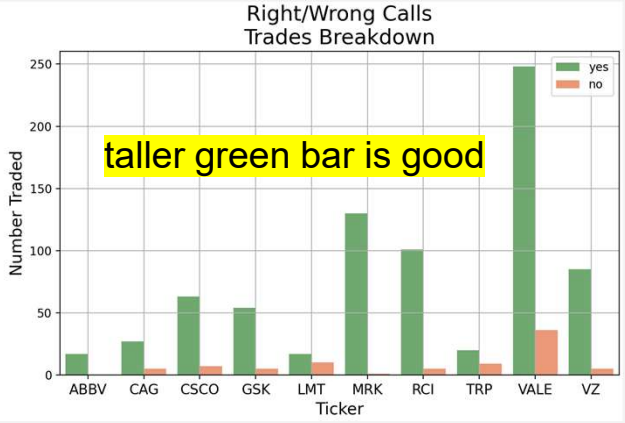
Conclusion Recommendation

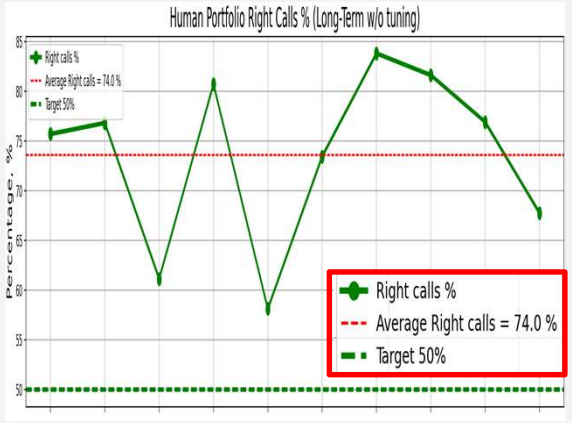
66

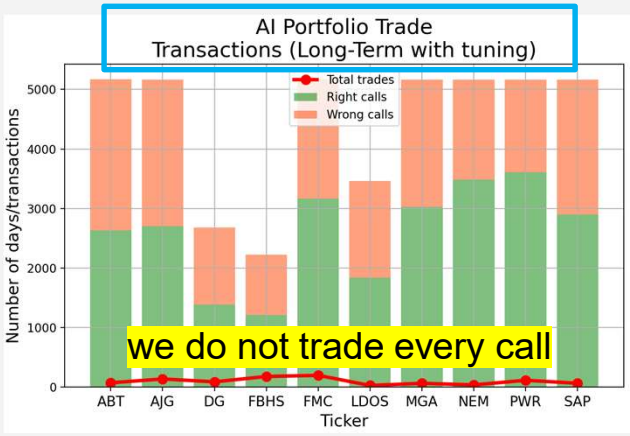
Conclusion

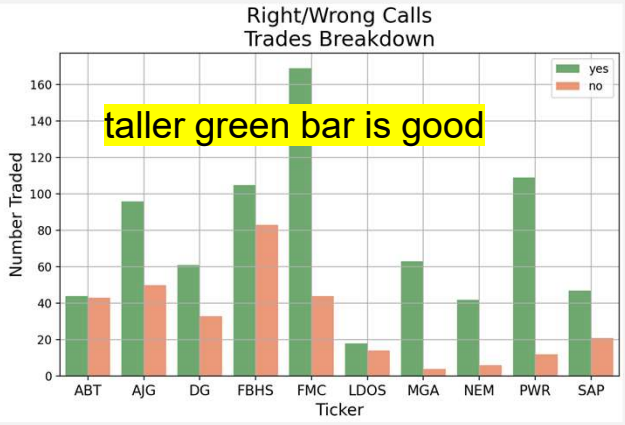
How often did we get it right?

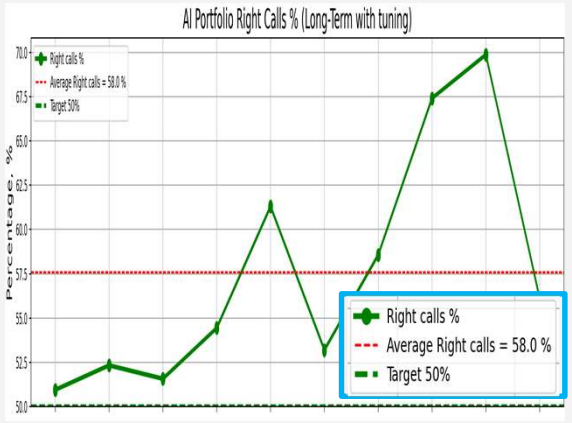
@ success metric - above 50% of right calls







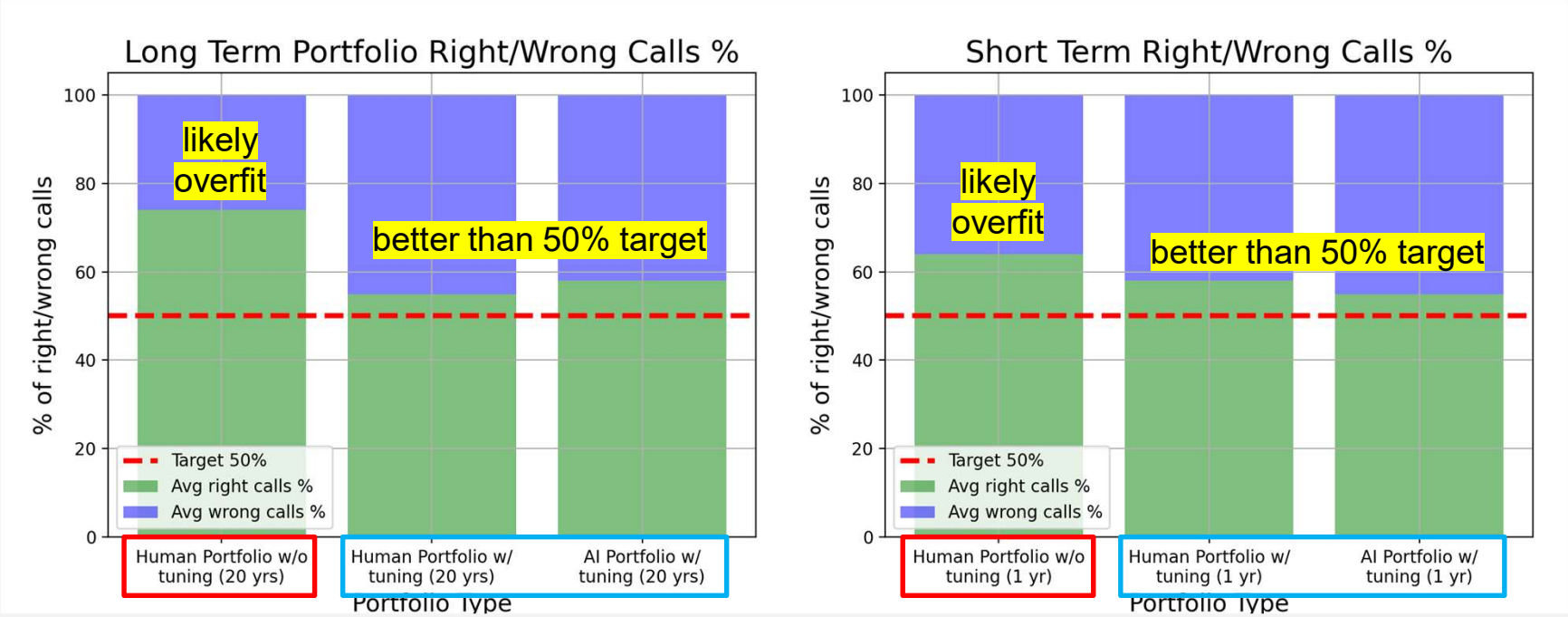




Conclusion

How often did we get it right?

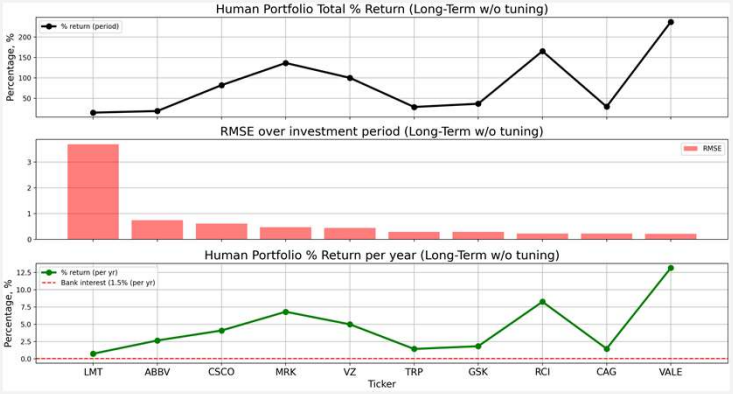
@ success metric - above 50% of right calls

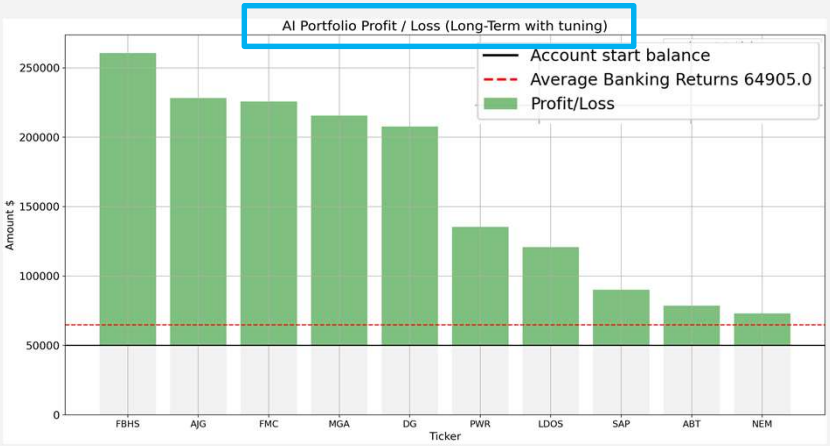


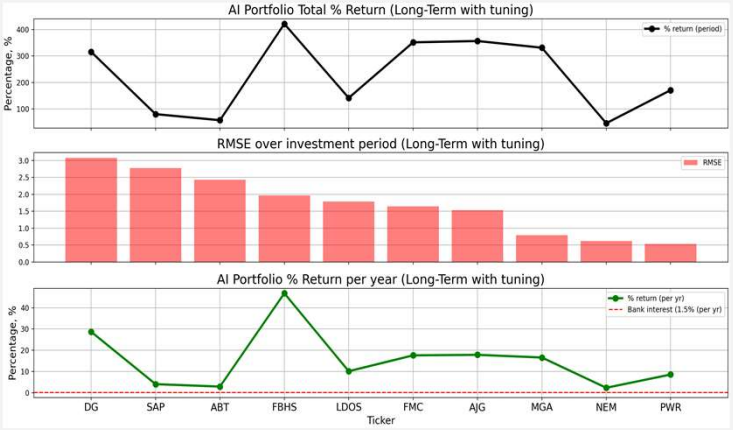
Conclusion

How much did we make or lose?

@ success metric - positive returns higher than bank interest (1.5% per yr)



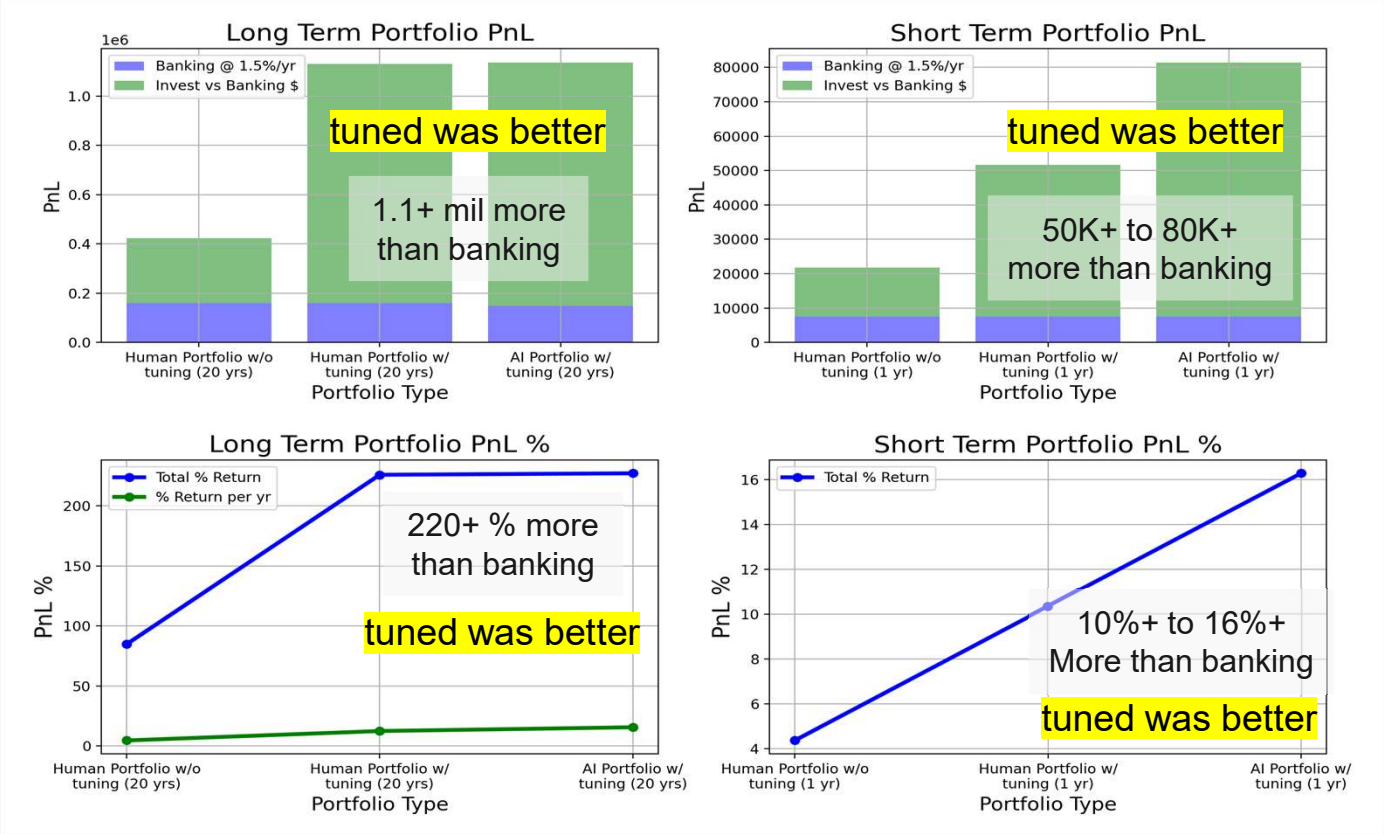




Conclusion

How much did we make or lose?

@ success metric - positive returns higher than bank interest (1.5% per yr)



11 Problem Statements

1. **Too time consuming to collect and analyze information to build a viable portfolio.**
2. **Not having a reliable and repeatable way to predict stock price movement.**
3. **Not having an automated trading strategy that wins consistently.**

Metrics of success

Our modeled portfolios should achieve:

1. **Above 50% of right calls**
2. **Positive returns higher than bank interest (@ 1.5% per yr)**

Recommendations

- Getting more diversified data sources into the data pipe
- Making a few AI portfolios with more companies in each to further diversified the risk
- Building an image classification model to identify technical trends on candlestick charts
- Re-tune the dynamic model according to your investment horizons and time-frame
- Expanding on the trading strategy to include short selling and integrating it with trading brokerage APIs
- Deploying the solution on a platform with more computational resources
- Designing a dashboard to easily track performance regularly

Credits

Our sincere and heartfelt thanks to Divya, Ryan, Alexis and the rest of GA Team for their un-reserving dedication to help us and for sharing their knowledge and experience.

Kudos to all classmates. We did it !!
See you all in the data science world.

random 42

The End

disclaimer: all material presented herein are for academic illustration purposes and should not be considered as investment advice.