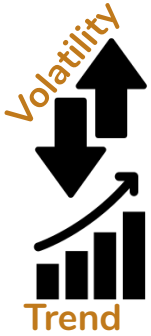
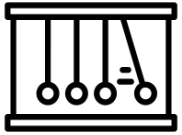
Team 5:
Signal Model Max

Atef Ajmal, Joaquim Curvo Neto, Philip Bonner, Kellye Rogers

SIGNALS

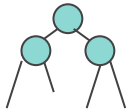
FEATURES



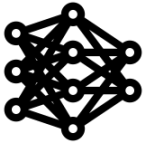


MODELS

Random Forest

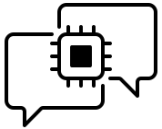


Neural Network

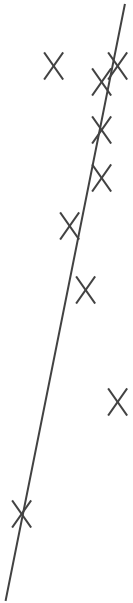


Clustering

MLP



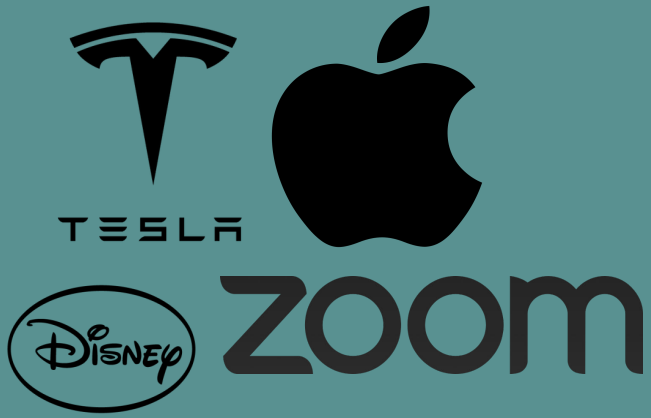
TRAIN



EVALUATION

PORTFOLIO BUILDER TO MAXIMIZE RETURN

The Data



Algo Proof of Concept

Build a Portfolio

We used Twitter, but the idea is that once the signals and models are all in place, you could run any stock through our algorithm to match it with the right signals, features, and model that should govern the buy/sell/hold strategy for that stock, to create the greatest chance to maximize return.

Signals used for the trading decision:

Stock: Twitter (TWTR)
Library: ta (Technical Analysis)

Volatility:

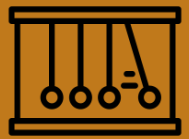
- *Crossover*
- *Vol_trend*
- *Bollinger band*

Trend:

- *Vortex indicator*
- *KST oscillator*

Volume:

- *Negative Volume Index*
- *Money Flow index*



Momentum:

- *MLP - Twitter*
- *Relative Strength Index (RSI)*
- *Stochastic Oscillator (SR)*

What the data looks like:

Features (X):

```
X.head()
```

Date	bollinger_signal	mfi_signal	nvi_signal	rsi_signal	vi_signal	crossover_signal	vol_trend_signal	kst_signal	nlp_signal
2015-10-05	0.0	0.0	0.0	1	0.0	0.0	0.0	0.0	1
2015-10-06	0.0	1.0	0.0	1	1.0	-1.0	0.0	0.0	1
2015-10-07	0.0	1.0	0.0	-1	1.0	1.0	1.0	1.0	-1
2015-10-08	0.0	1.0	0.0	-1	1.0	1.0	1.0	1.0	0
2015-10-09	0.0	1.0	1.0	-1	1.0	1.0	1.0	1.0	0

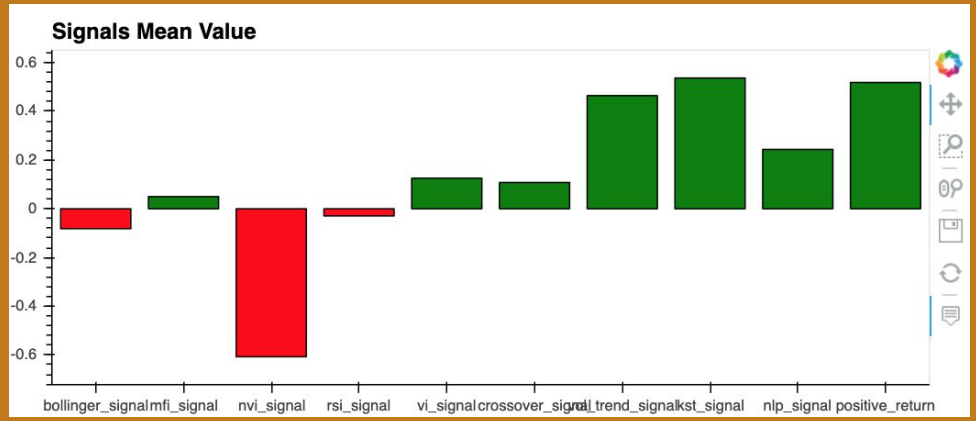
Target (Y):

```
Y_returns.head()
```

Date	daily_returns	positive_return
2015-10-05	0.000000	0.0
2015-10-06	-0.018828	0.0
2015-10-07	0.080014	1.0
2015-10-08	0.016426	1.0
2015-10-09	0.017480	1.0

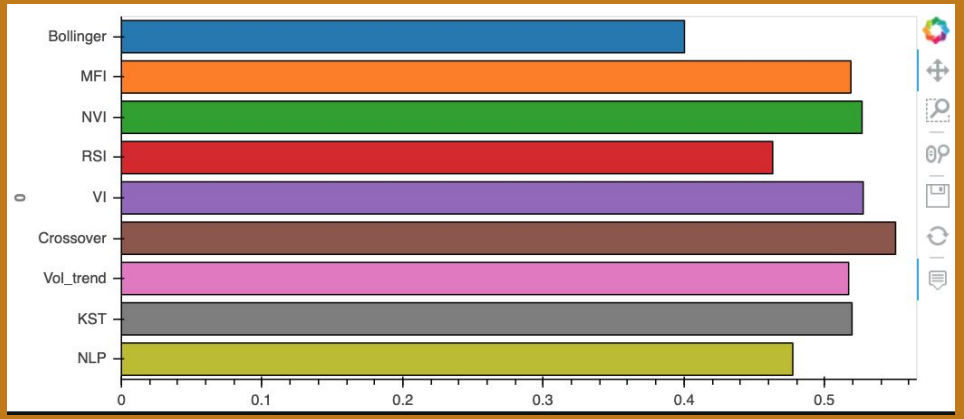
Time window: 5 years

Details of the signals dataset

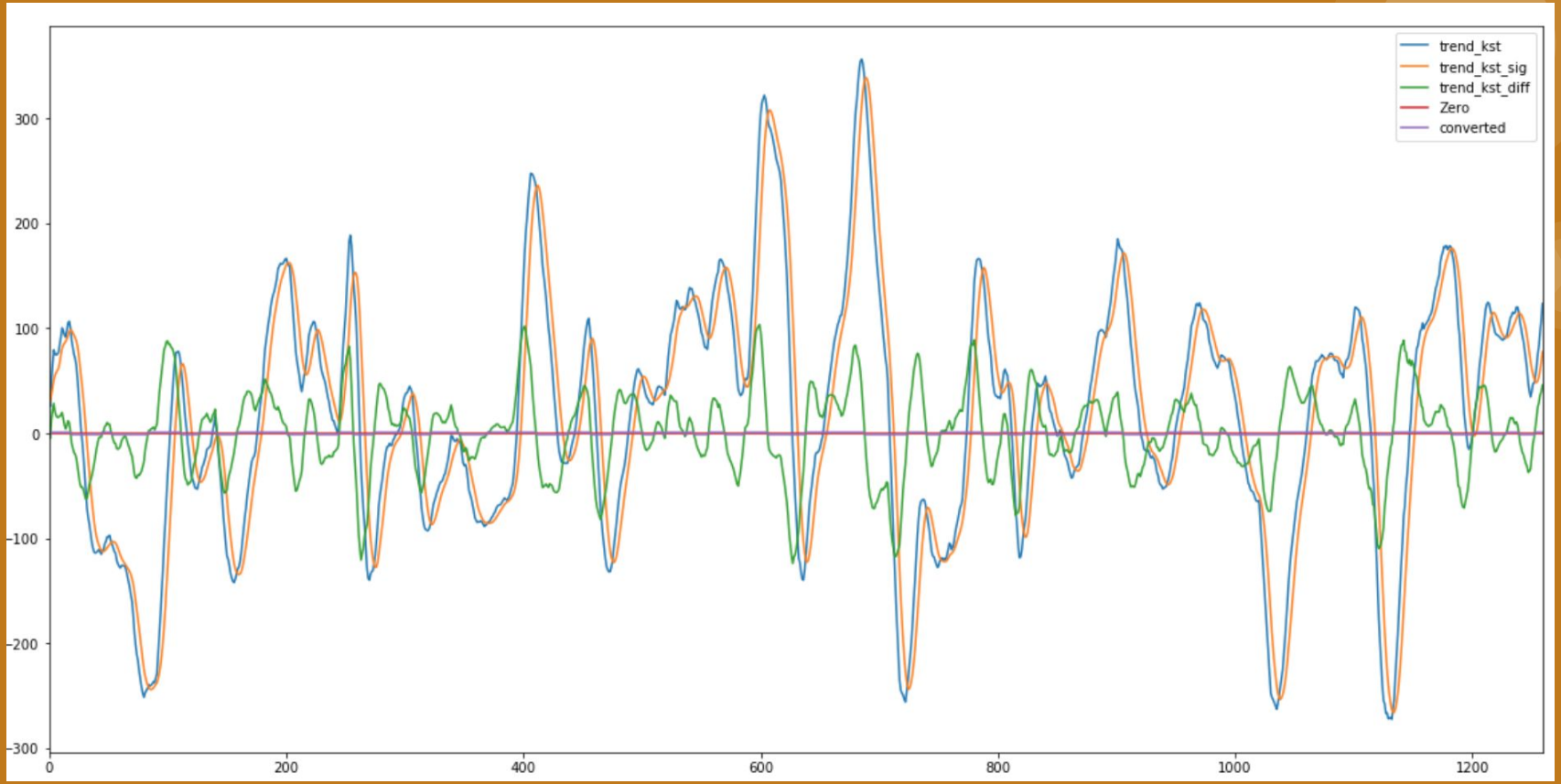


Features in red indicate a bearish approach to the stock in question.
Features in green have a majority bullish position.

Crossover is the feature that most matched the Actual Results of the stock.



Details of the signals dataset



Natural Language Processing

Tweepy

An easy-to-use Python library for accessing the Twitter API.

```
# Open/Create a file to append data
csvFile = open(sys.argv[1]+'tweets.csv', 'a')
#Use csv Writer
fields = ('date','text', 'followers')
csvWriter = csv.writer(csvFile, lineterminator= '\n')

for tweet in tweepy.Cursor(api.search,q='#Twitter', count=20000, lang="en", since_id="2015-10-1
    print(tweet.created_at, tweet.text)
    follower_count = tweet.user.followers_count
    #if tweet.created_at >= datetime.datetime:
    csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8'),follower_count])
```

```
api = tweepy.API(auth)
colnames=['date', 'text', 'followers']

df = pandas.read_csv(sys.argv[1]+'tweets.csv',encoding='latin-1', names=colnames, header=None)
df['polarity'] = 0.0000
df['sentiment_confidence'] = 0.0000

for index,row in df.iterrows():
    analysis = TextBlob(df['text'][index])
    sentiment, confidence = analysis.sentiment
    df.at[index,'polarity'] = sentiment
    df.at[index,'sentiment_confidence'] = confidence

df.to_csv(sys.argv[1]+'sentiment.csv')
```

Twitter Data

1.

2020-10-10 14:46:53	b'Dear @verified: please stop recommending me people I went to school with/that live in town with me to follow. I hav\xe2\x80\xa6 https://t.co/SnTOsw51tW'	232
2020-10-10 14:46:50	b"Sadly you can walk into several 'foreign countries' in any city in the UK. That is if you have a death wish and no\xe2\x80\xa6 https://t.co/Y7c4YSeyYu"	51
2020-10-10 14:46:50	37\xf0\x9f\x91\x87\nVankaPro\xe2\x98\x80\xef\xb8\x8f\xf0\x9f\x8c\x8a\xf0\x9f\x8c\x8a\xf0\x9f\x8c\x8a \n\nTwitter Will Turn\xe2\x80\xa6 https://t.co/zlpliMMrNv'	20970

2.

date	text	followers	polarity	sentiment_confidence
2020-10-10 14:47:04	l\xd8\xae\xd9\x8a\xd8\xb1'	922	-0.3	0.4
2020-10-10 14:46:53	s https://t.co/SnTOsw51tW'	232	0.13636363636363635	0.5

3.

followers	polarity	sentiment_confidence	clusters	Date	High	Low
922	-0.3	0.4	4	10/5/15	28.25	26.32999992
232	0.136363636	0.5	3	10/6/15	28.39999962	26.75

4.

```
[204]: trigger = 100
for index, row in twitter_sentiment_df.iterrows():
    if trigger == 100:
        twitter_sentiment_df.loc[index, "nlp_signal"] = "1"
    elif row["polarity"] > 0 :
        twitter_sentiment_df.loc[index, "nlp_signal"] = "1"
    elif row["polarity"] < 0:
        twitter_sentiment_df.loc[index, "nlp_signal"] = "-1"
    else:
        twitter_sentiment_df.loc[index, "nlp_signal"] = "0"
    trigger = row["polarity"]

twitter_sentiment_df
```

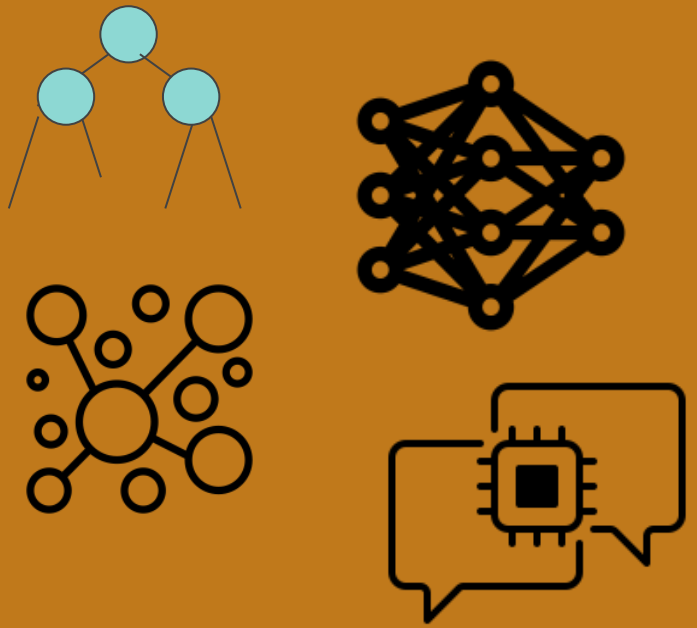
[204]:

	polarity	nlp_signal
0	-0.300000	1
1	0.136364	1
2	-0.208333	-1
3	0.000000	0
4	0.000000	0

Models

Models

- Random forest
- Neural networks
- Decision Tree
- MLP Model



For each model we:

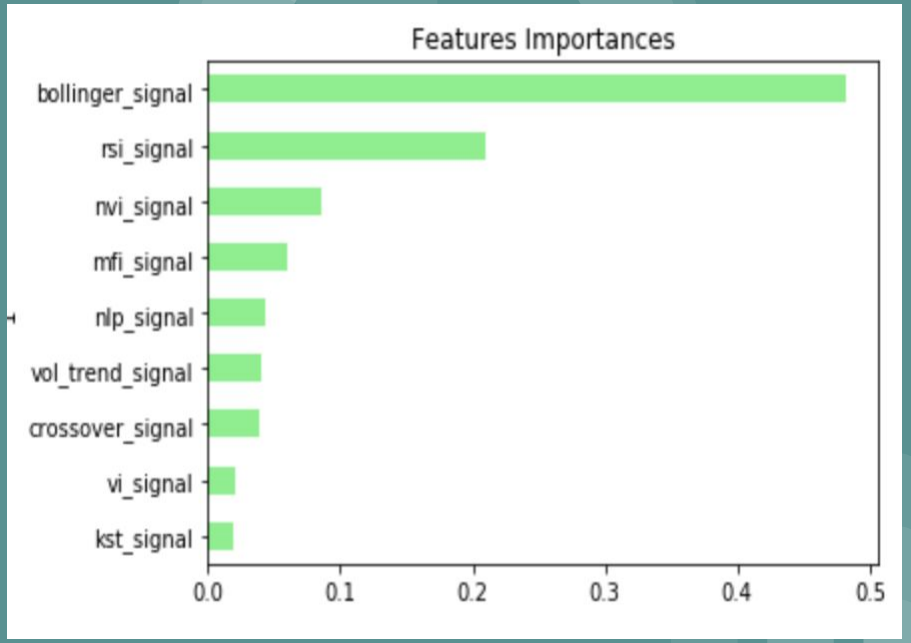
1. Studied the signals in different strategies
2. Built the features, or found examples online
3. Trained the model
4. Tested the model
5. And then we did the prediction and evaluation
6. This will tell you if you had a positive return or a negative one

Random Forest Results

	Predicted 0	Predicted 1
Actual 0	84	68
Actual 1	60	103

Accuracy Score : 0.5936507936507937
Classification Report

	precision	recall	f1-score	support
0.0	0.58	0.55	0.57	152
1.0	0.60	0.63	0.62	163
accuracy			0.59	315
macro avg	0.59	0.59	0.59	315
weighted avg	0.59	0.59	0.59	315



All 9 Features

	Predicted 0	Predicted 1
Actual 0	84	68
Actual 1	60	103

Accuracy Score : 0.5936507936507937

Classification Report

	precision	recall	f1-score	support
0.0	0.58	0.55	0.57	152
1.0	0.60	0.63	0.62	163
accuracy			0.59	315
macro avg	0.59	0.59	0.59	315
weighted avg	0.59	0.59	0.59	315

Top 5 Features

	Predicted 0	Predicted 1
Actual 0	94	58
Actual 1	74	89

Accuracy Score : 0.580952380952381

Classification Report

	precision	recall	f1-score	support
0.0	0.56	0.62	0.59	152
1.0	0.61	0.55	0.57	163
accuracy			0.58	315
macro avg	0.58	0.58	0.58	315
weighted avg	0.58	0.58	0.58	315

Top 3 Features

	Predicted 0	Predicted 1
Actual 0	97	55
Actual 1	82	81

Accuracy Score : 0.5650793650793651

Classification Report

	precision	recall	f1-score	support
0.0	0.54	0.64	0.59	152
1.0	0.60	0.50	0.54	163
accuracy			0.57	315
macro avg	0.57	0.57	0.56	315
weighted avg	0.57	0.57	0.56	315

The Top Feature

	Predicted 0	Predicted 1
Actual 0	114	38
Actual 1	106	57

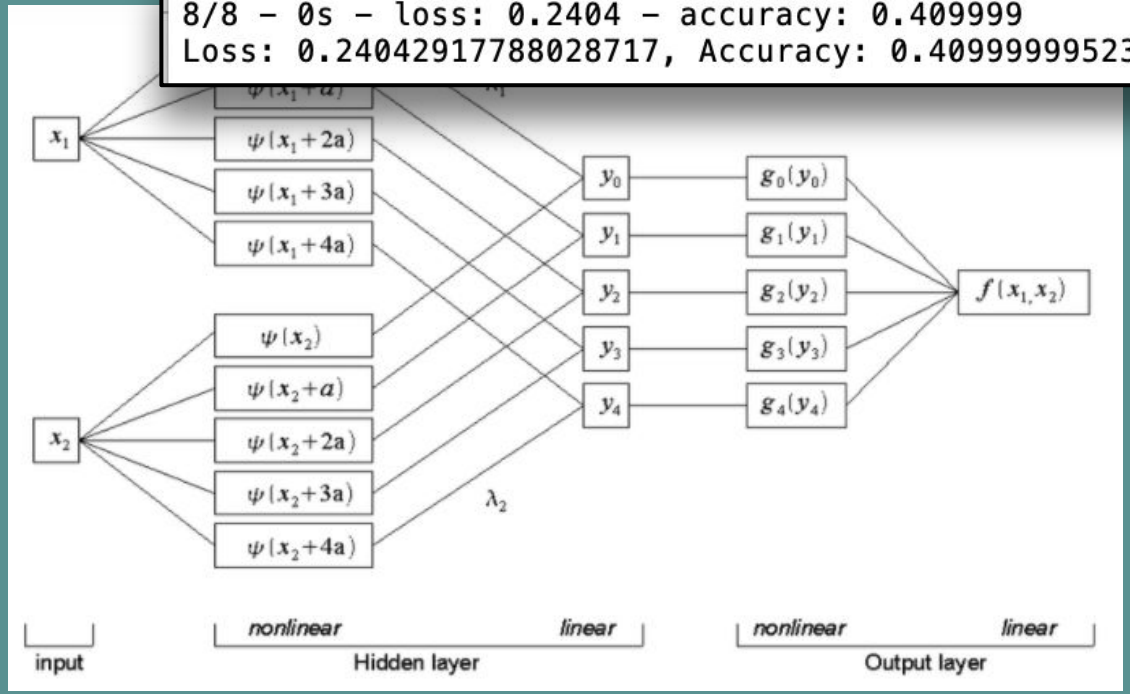
Accuracy Score : 0.5428571428571428

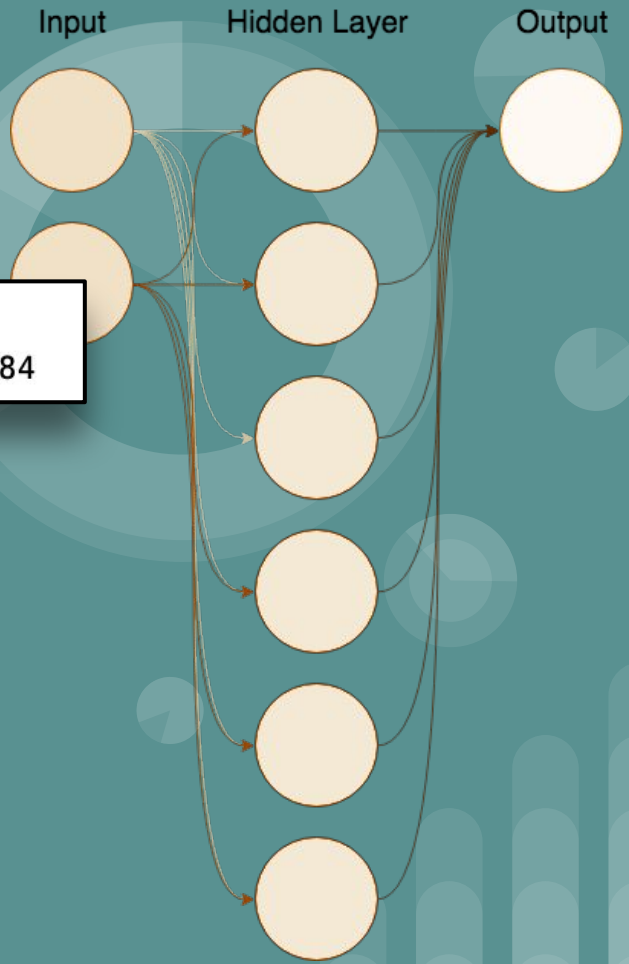
Classification Report

	precision	recall	f1-score	support
0.0	0.52	0.75	0.61	152
1.0	0.60	0.35	0.44	163
accuracy			0.54	315
macro avg	0.56	0.55	0.53	315
weighted avg	0.56	0.54	0.52	315

Neural Networks Results

```
8/8 - 0s - loss: 0.2404 - accuracy: 0.409999
Loss: 0.24042917788028717, Accuracy: 0.4099999952316284
```





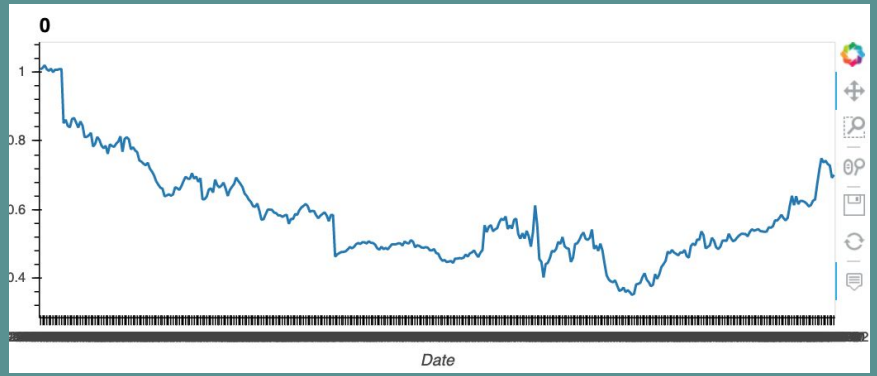
Multi Layer Perceptron Results

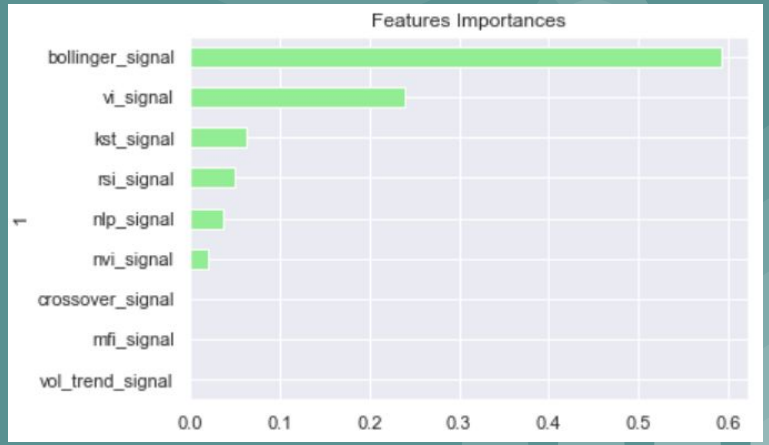
```
[[92 42]
 [32 86]]
              precision    recall  f1-score   support

           0       0.74      0.69      0.71       134
           1       0.67      0.73      0.70       118

    accuracy                           0.71       252
   macro avg       0.71      0.71      0.71       252
weighted avg       0.71      0.71      0.71       252
```

Decision Tree

Best results were achieved with:
criterion='gini', max_depth=3

	precision	recall	f1-score	support
0.0	0.62	0.66	0.64	182
1.0	0.66	0.62	0.64	196
accuracy			0.64	378
macro avg	0.64	0.64	0.64	378
weighted avg	0.64	0.64	0.64	378



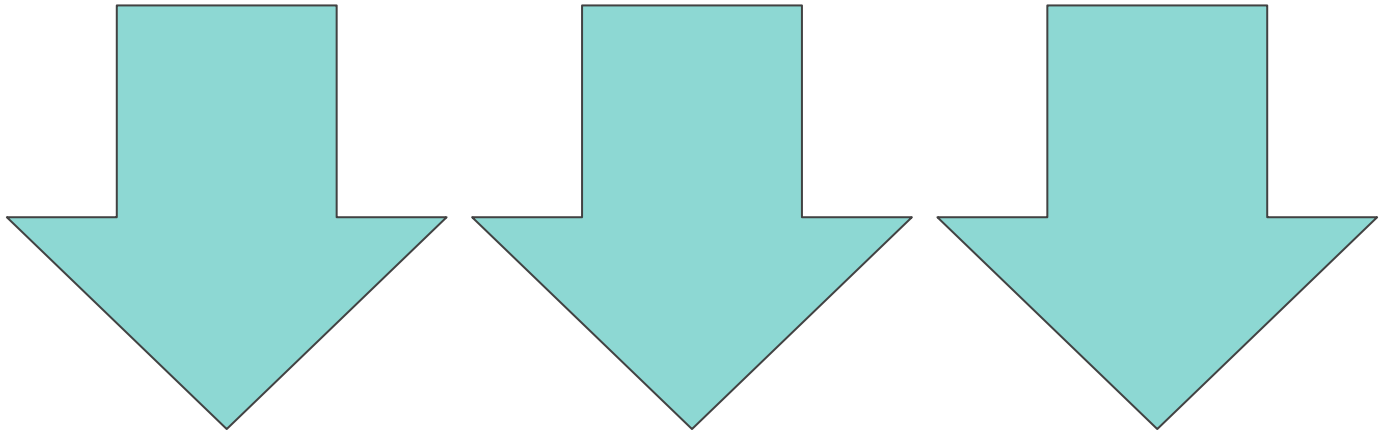


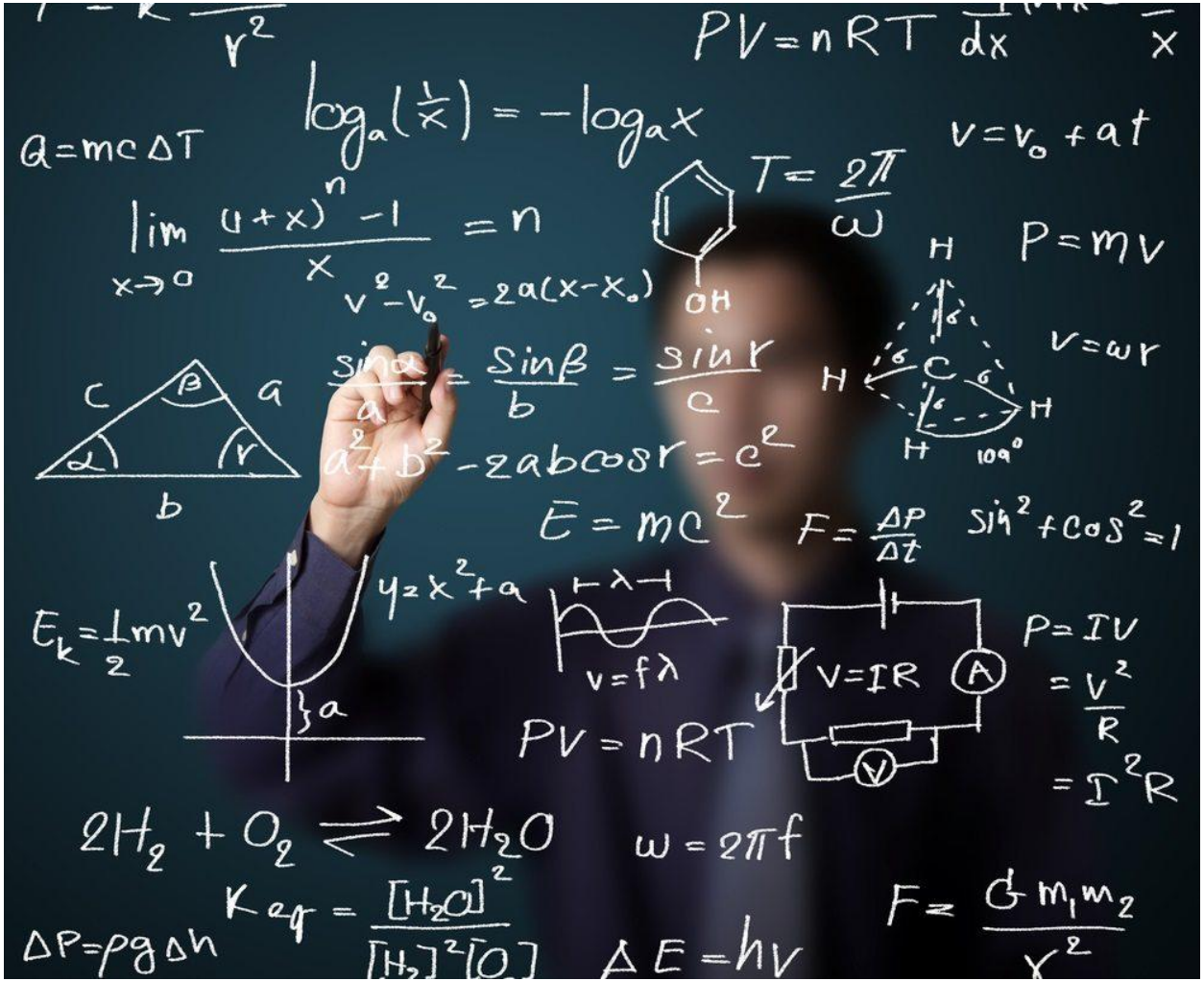
Questions?

The End.

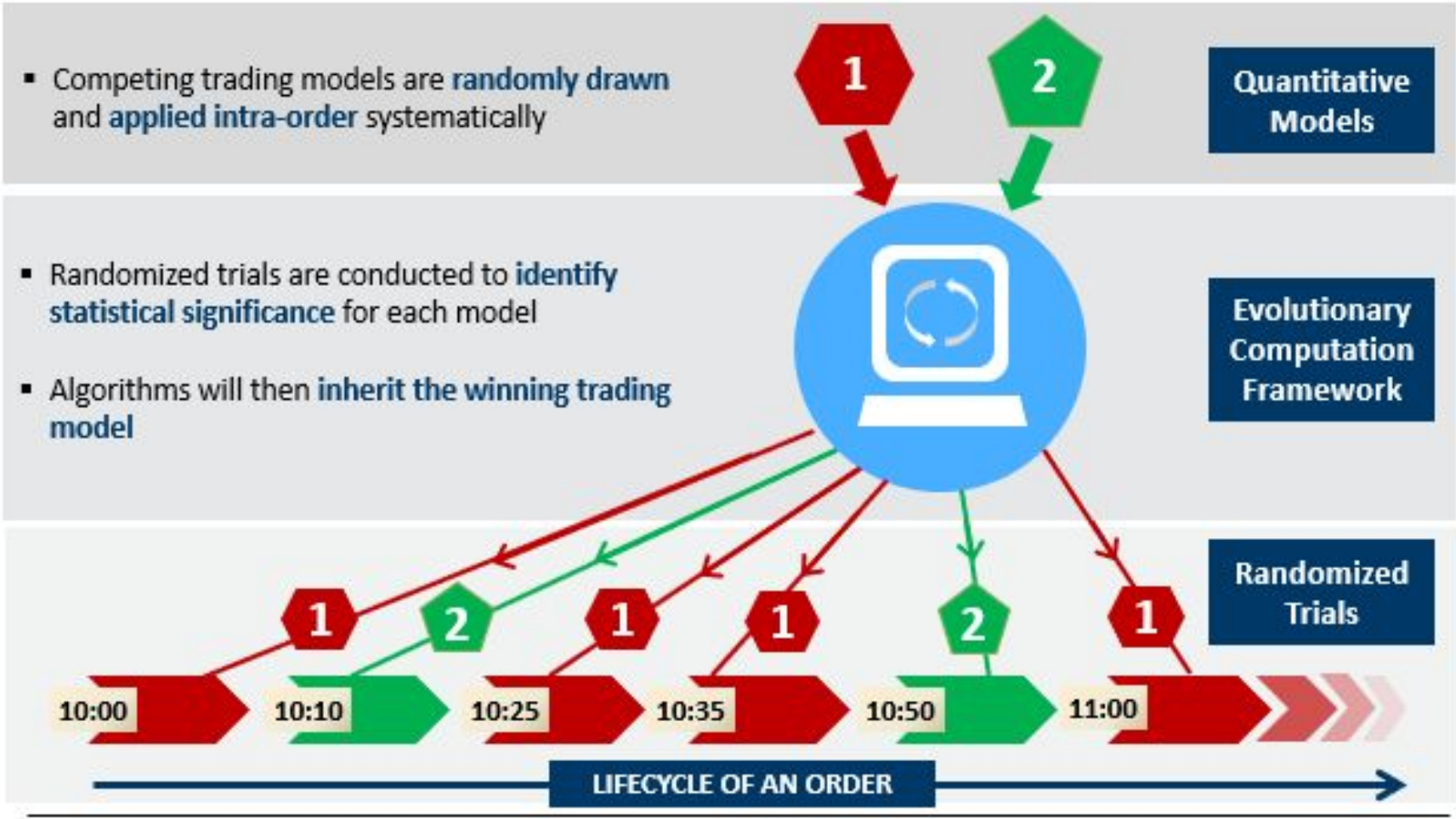
Thank you!

Slides below here are stock slides to pull from.




$PV=nRT$
$\log_a\left(\frac{1}{x}\right)=-\log_a x$
$Q=mc\Delta T$
$v=v_0+at$
$T=\frac{2\pi}{\omega}$
$\lim_{x\to 0}\frac{(1+x)^n-1}{x}=n$
$P=mv$
$v^2-v_0^2=2a(x-x_0)$
OH
$v=\omega r$
$\frac{\sin\alpha}{a}=\frac{\sin\beta}{b}=\frac{\sin\gamma}{c}$
$a^2+b^2-2ab\cos\gamma=c^2$
$E=mc^2$
$F=\frac{\Delta P}{\Delta t}$
$\sin^2+\cos^2=1$
$E_k=\frac{1}{2}mv^2$
$y=x^2+a$
$v=f\lambda$
$V=IR$
$P=IV$
$=\frac{V^2}{R}$
$=I^2R$
$PV=nRT$
$2H_2+O_2\rightleftharpoons 2H_2O$
$\omega=2\pi f$
$K_{eq}=\frac{[H_2O]^2}{[H_2]^2[O_2]}$
$F=\frac{Gm_1m_2}{r^2}$
$\Delta P=\rho g\Delta h$
$\Delta E=h\nu$
109°

ALGORITHMIC TRADING


Competing trading models are randomly drawn and applied intra-order systematically
Quantitative Models
Randomized trials are conducted to identify statistical significance for each model
Algorithms will then inherit the winning trading model
Evolutionary Computation Framework
Randomized Trials
1
2
1
1
2
1
10:00
10:10
10:25
10:35
10:50
11:00
LIFECYCLE OF AN ORDER