Columbia University
School of Professional Studies

NLP in Crypto Trading

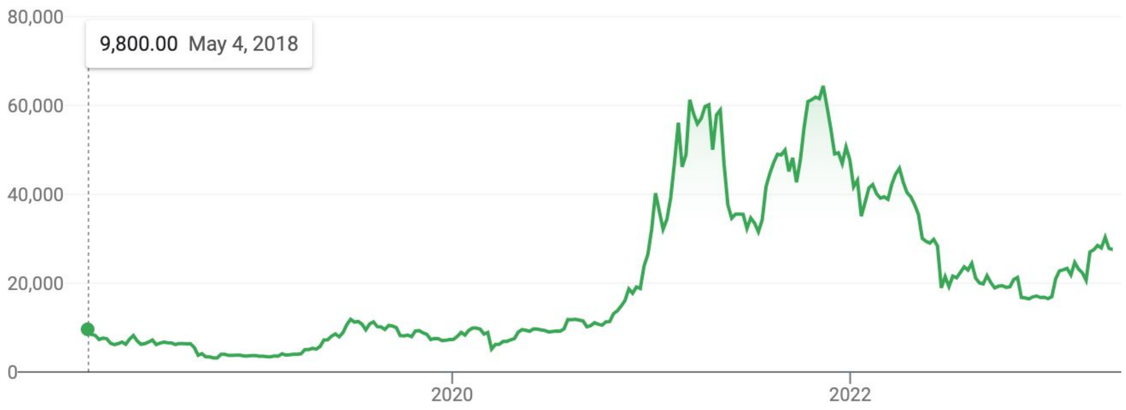
ACTU PS5842 Advanced Data Science in FI

Presented By: Emily Xiao, Yuyang Zhao

Supervised By: Yubo Wang

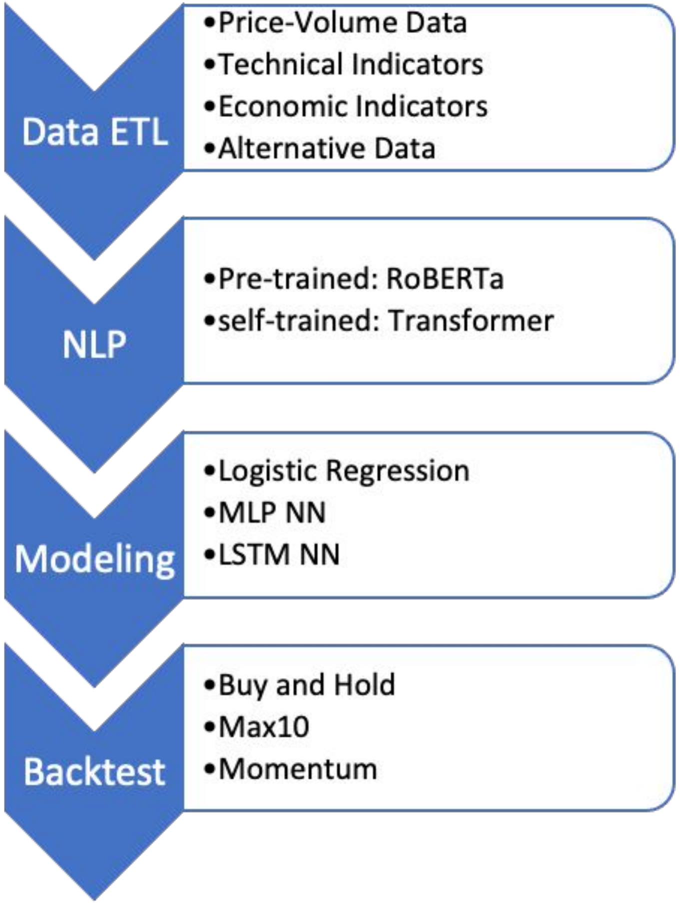
Cryptocurrencies

- 2.5% market cap equivalent to U.S. equity market cap
- Highly volatile
- Influencers
  - supply and demand
  - investor sentiments
  - government regulation
  - public attention



Goal

- To build a deep learning model to predict crypto returns
  - pricing data
  - alternative data
  - news and social feeds.
- To construct a profitable long-short equity strategy
  - based on the predicted returns
  - backtest on historical data.



Data Exploration

- Retrieve daily BTC **OHLC+volume** data from YahooFinance
- **Technical Indicators** w/ various windows
  - Moving Averages
  - Bollinger Bands
  - Relative Strength Index
  - Force Index
- **Economic indicators**: AlphaVintage API
  - US 5, 10, 30-year treasury yield
  - S&P500, Gold
  - Consumer Price Index
  - Federal Funds Rate
- Alternative Data: tweets related to BTC
  - Web-scraping **Twitter** feeds
  - Exclude trash tweets
- Transformer
  - Subjectivity
  - Polarity
  - Sentiment Label

Text Sentiment

- Use transformer to processing a given word in relation to all other words in a sentence, rather than processing them one at a time
- Simple framework of transformer
  - trained on about 1.6 million tweets
  - return three sentiment status: positive, neutral and negative with their corresponding predicted possibilities
  - Example:
    - Text data: "Covid cases are increasing fast!"

```
1) Negative 0.7236
2) Neutral 0.2287
3) Positive 0.0477
```

Transformer

- Model Training
  - Training data: Sentiment140 dataset with 1.6 million tweets
  - Variable used: “text” & “target”
  - TextVectorization
  - build_transformer_model()
- Input
  - 150 thousands bitcoin tweets
- Output
  - Subjectivity: subjective/objective
  - Polarity: positive/neutral/negative

Data source: https://www.kaggle.com/datasets/kazanova/sentiment140?resource=download

Transformer Structure


```
+--------+      +--------------+     +-----------------+     +--------------------------+     +---------+     +----------+
| Input  | --> | Embedding| --> | Transformer | --> | GlobalAvgPooling1D | --> | Dense | --> | Output |
+--------+      +--------------+     +-----------------+     +--------------------------+     +---------+     +----------+
```

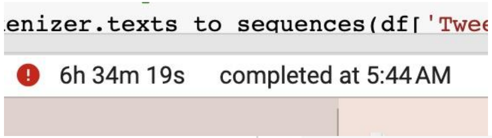
Transformer with TensorFlow

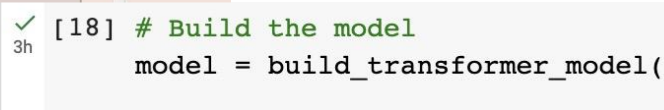
- build_transformer_model function
  - Input layer
  - Embedding layer
  - TransformerBlock layer (loop)
  - GlobalAveragePooling1D layer
  - Dense layer
  - returns a Keras model

Transformer with TensorFlow

- TransformerBlock class:
  - Inherits from tf.keras.layers.Layer
  - __init__ method
    - tf.keras.layers.MultiHeadAttention
    - tf.keras.layers.Add
    - tf.keras.layers.LayerNormalization
    - tf.keras.Sequential
  - call method
    - define how to apply these layers and operations to the input tensor

Transformer Runtime

- Training
  - LSTM version ⟶


  - Transformer simple version ⟶


  - Transformer updated version (final version) ⟶ 20 hrs

- Input to output
  - Transformer package ⟶


  - Our transformer ⟶


Logistic Regression

- Serves as a baseline model
- Classification problem: positive or negative price changes
- Use all indicators and text data
- Resulting accuracy ~50% - random guessing

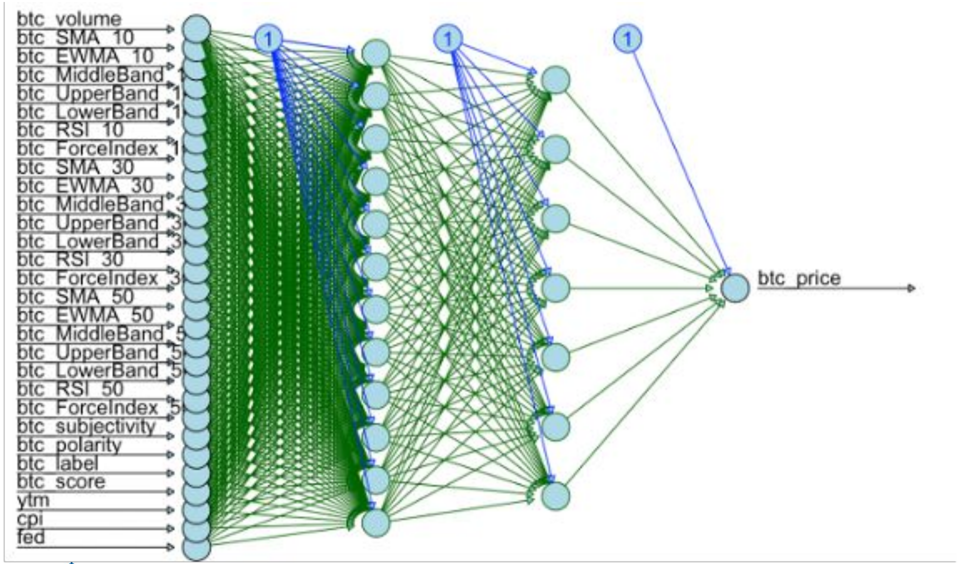
$$L(\beta|\mathbf{y}) = \prod_{i=1}^{N} \frac{n_i!}{y_i!(n_i - y_i)!} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i}$$

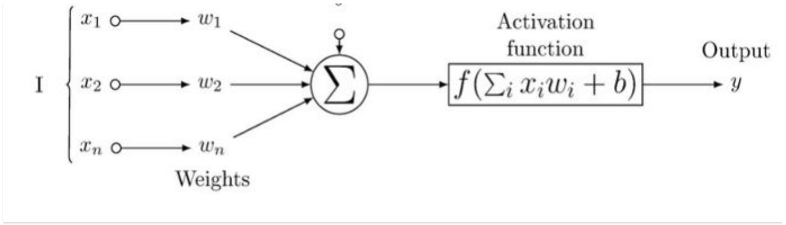
$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

	precision	recall	f1-score	support
-1	0.53	0.55	0.54	42
1	0.51	0.50	0.51	40
accuracy			0.52	82
macro avg	0.52	0.52	0.52	82
weighted avg	0.52	0.52	0.52	82

Multilayer Perceptron NN

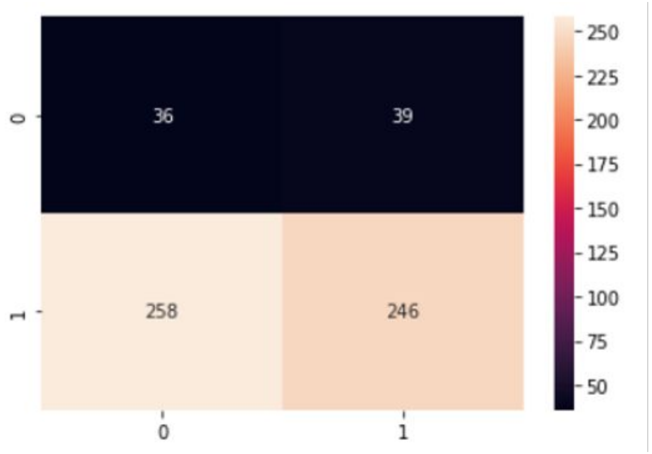
- Many layers with many neurons stacked together
- Pick out features at different scales or resolutions, combine them into higher-order features, eg. from lines to collections of lines to shapes.





Model Performance

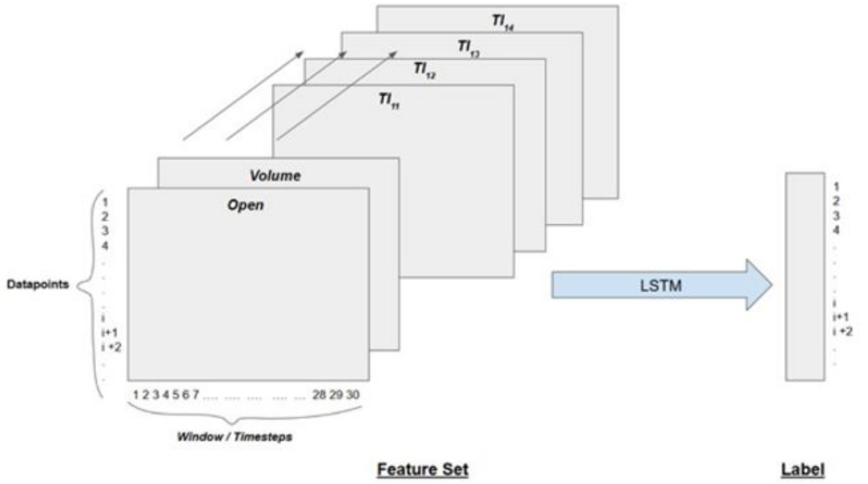
- Poor performance
- Uneven split between buy and sell predictions.
- Fails to learn in the presence of time lags between relevant input events and target signals.



	precision	recall	f1-score	support
0	0.48	0.12	0.20	294
1	0.49	0.86	0.62	285
accuracy			0.49	579
macro avg	0.48	0.49	0.41	579
weighted avg	0.48	0.49	0.41	579

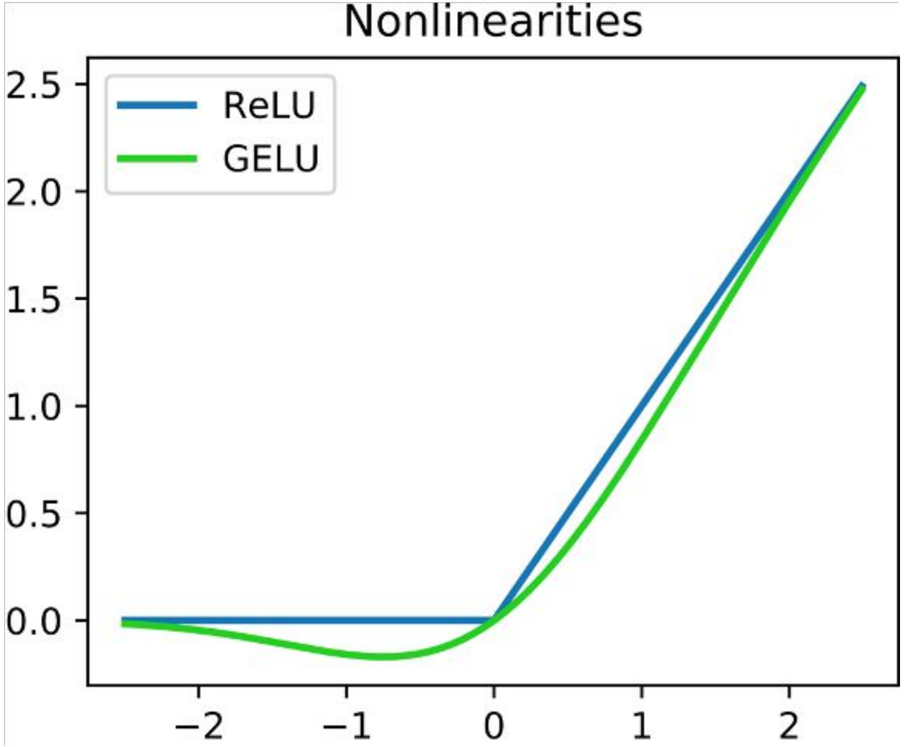
Long Short-Term Memory NN

- LSTM model learn to bridge time lags
  - More suitable to classify, process and predict time series given time lags.
- Build 3-D dataframe
  - Adding a window dimension
  - 21*30 matrix for each sample



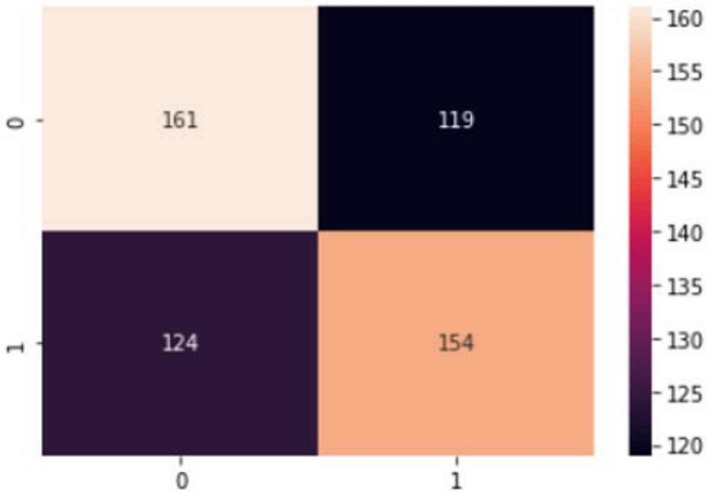
Long Short-Term Memory NN

- GeLu vs. ReLu
  - ReLu(x) = max(0, x)
  - GeLu(x) = xφ(x)
    - nonlinear weights inputs by their percentile
- Dropout layer
  - Prevent overfitting with a given rate



Model Performance

- Sensitive to seed and parameter changes
  - Seeds
  - Learning rate
  - Batches size
  - Hidden units
- Outperforms the baseline model



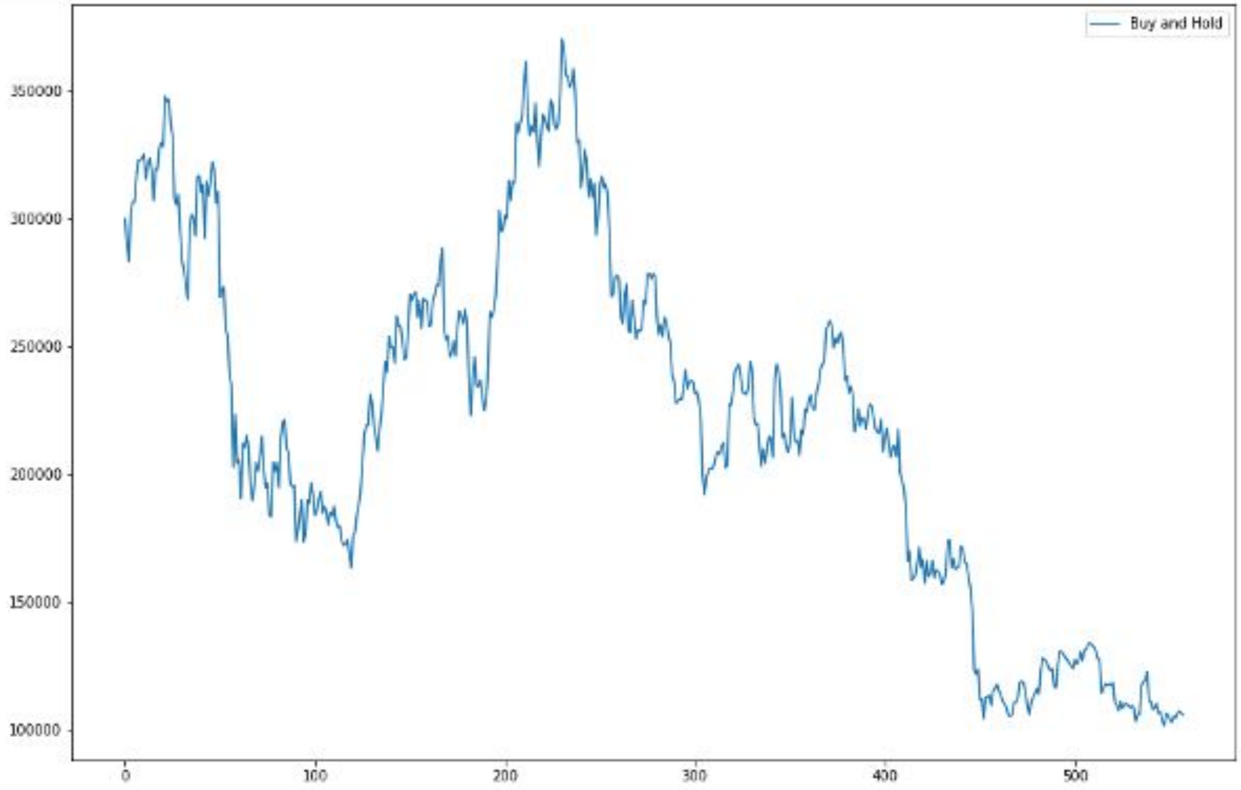
	precision	recall	f1-score	support
0	0.56	0.57	0.57	280
1	0.56	0.55	0.56	278
accuracy			0.56	558
macro avg	0.56	0.56	0.56	558
weighted avg	0.56	0.56	0.56	558

Trading Rules

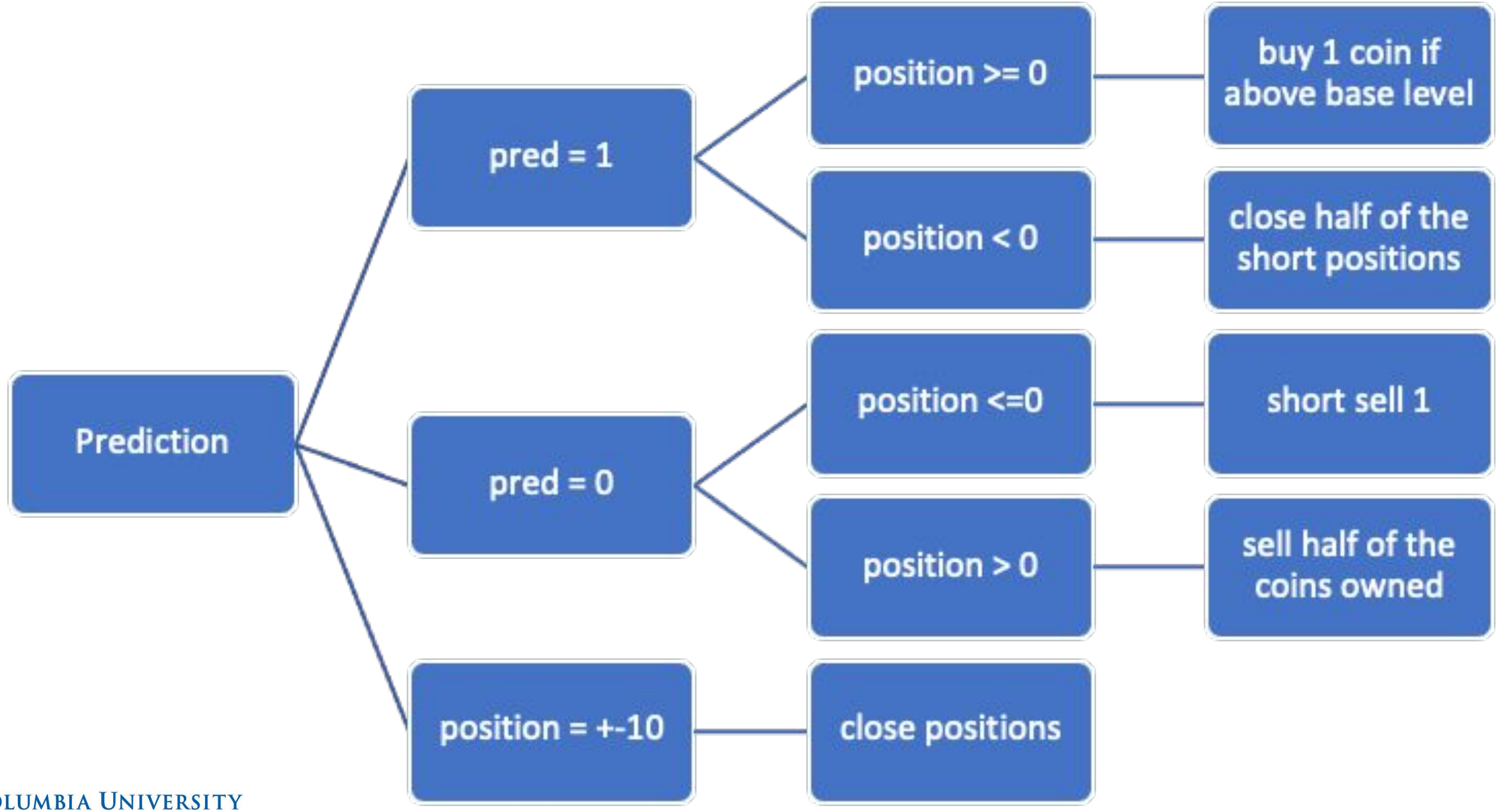
- Break down the strategy value into 3 parts: cash, coin, and margin account
- Initial cash: $300,000
- Cash base level: $5000
- Commission Fee: 0.2%
- Allow short selling
  - Initial Margin = 50%
  - Maintenance Margin = 30%
  - Interest = 0.02% per day
  - Collateral = half of the amount traded

Buy and Hold Strategy

- Benchmark

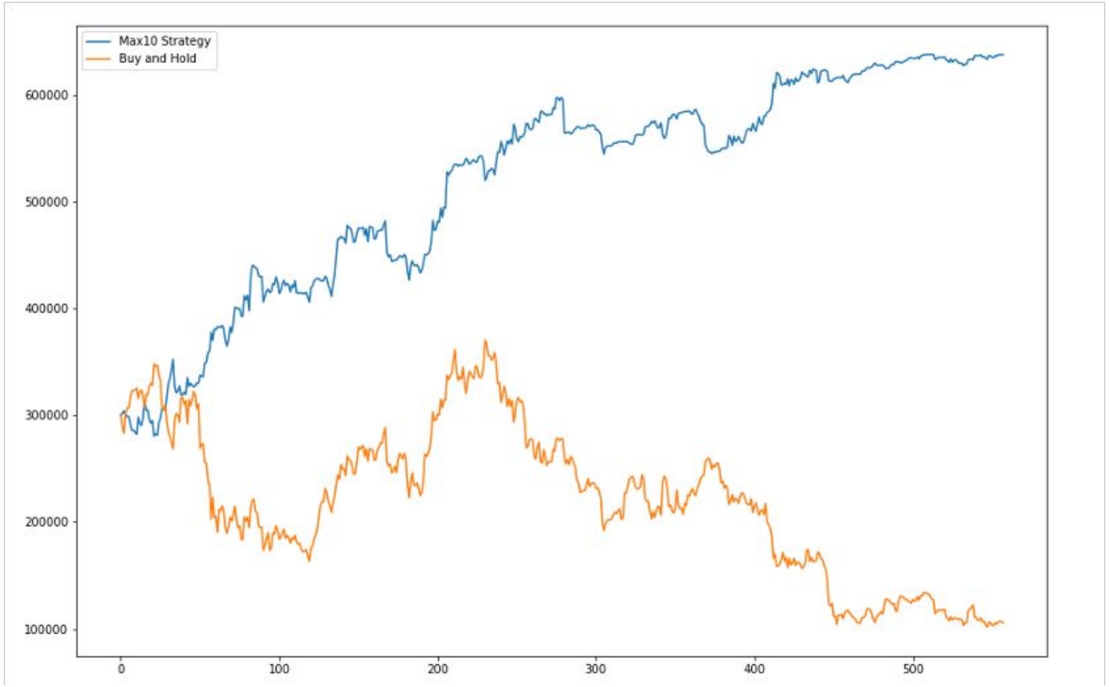


Max10 Strategy



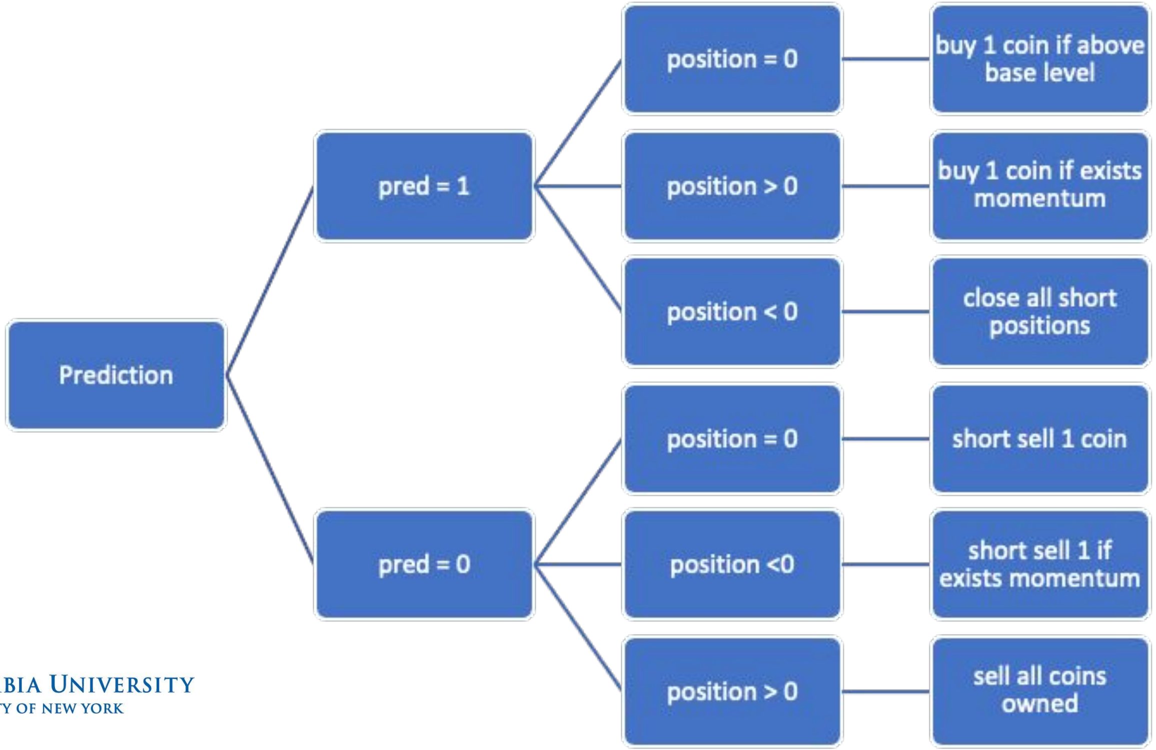
Max10 Strategy

- Strategy Performance



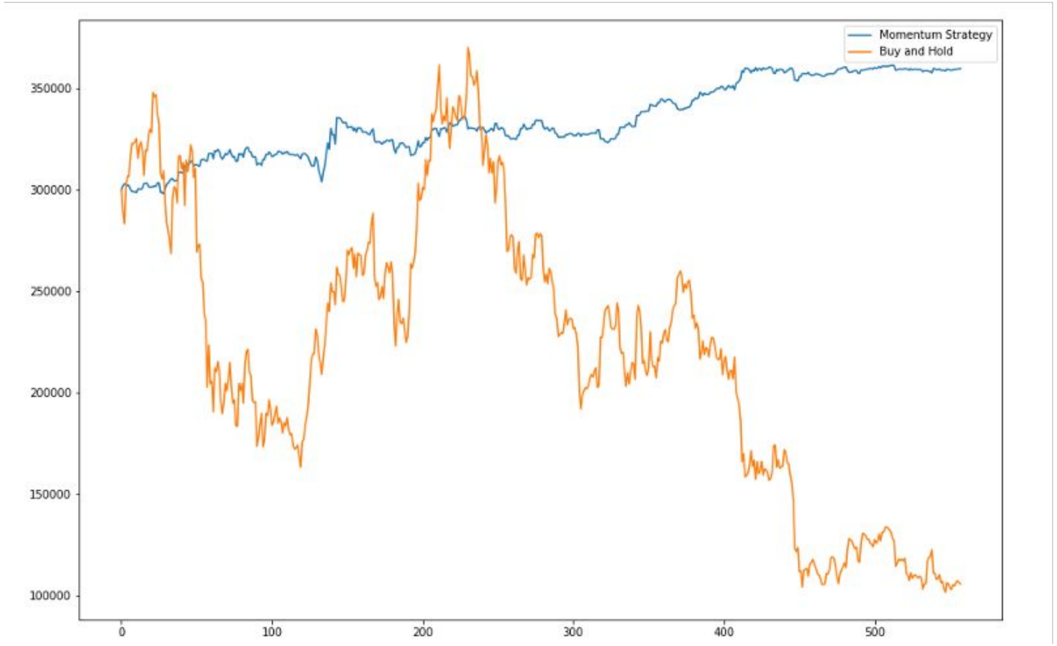
Momentum Strategy

- Same settings of cash, coin, margin account, collateral, interest, cash base level
- Exists momentum: coin value has increased/decreased 4 days in a row



Momentum Strategy

- Strategy Performance



Performance Metrics

- Max 10 Strategy vs Momentum Strategy vs Buy and Hold

Initial cash $300,000	**Max 10**	**Momentum**	**Buy and Hold**
Total Return	112.60%	19.95%	-64.72%
PnL	337795.88	59848.21	-194158.93
Sharpe Ratio	1.91	1.06	-0.63
Maximum Drawdown	-11.51%	-5.58%	-72.55%

Summary

- Increasing market capitalization, high volatile
- Data ETL
  - Include alternative data
  - NLP: Transformer -> text sentiment
- Modeling
  - MLP: fully connected, efficiency, uneven split between labels
  - LSTM: 3 dimensional dataframe, GeLu, high accuracy
- Strategy
  - Max10: possibility to exploit huge profit
  - Momentum: stable return

Futures Work

- Better quality of alternative data
  - Filter spam and advertisements
- Other cryptocurrencies, lack of alternative data
  - August 2015 - October 2022 (2884 days)
    - 952 Ethereum related Tweets
  - Major discussion in Bitcoins
  - Look into other social media platforms
- Discover more stable model
  - initial random weight matrix highly dependent on seeds

Questions?