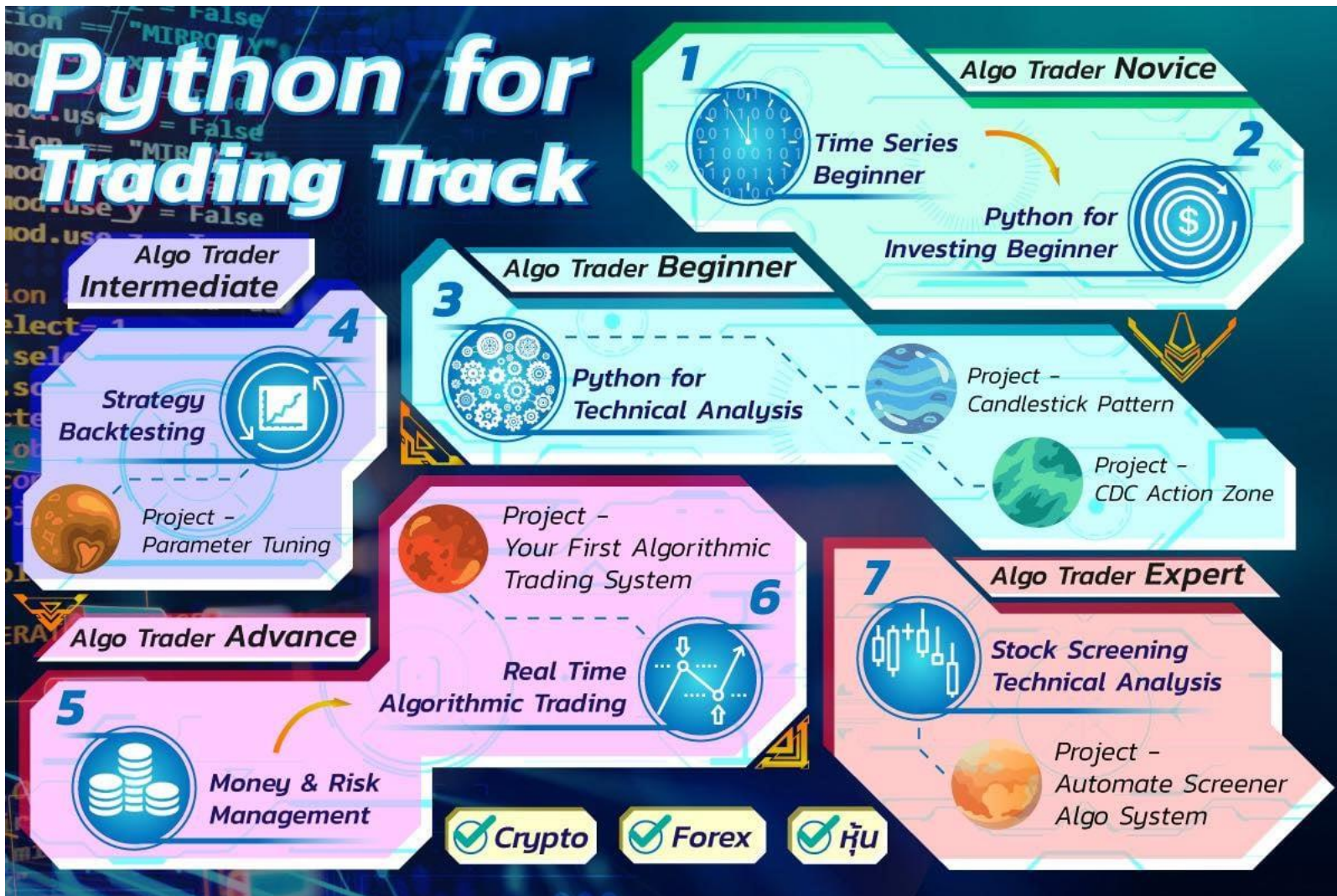




# PYTHON FOR TRADING TRACK COURSE



กำหนดการ Live และเริ่มคอร์สเรียน

1. Python for Time Series [23 พ.ค.]
2. Python for Investing Beginner [30 พ.ค.]
3. Technical Analysis [13 มิ.ย.]
4. Strategy Backtesting [27 มิ.ย.]
5. Money & Risk Management [18 ก.ค.]
6. Realtime Algorithmic Trading [1 ส.ค.]
7. Stock screening Technical Analysis [15 ส.ค.]

เริ่ม 1 ทุ่มตรง



# Algo Trader Beginner

```
df
```

	Population	Capital	Country
0	11190846	Brussels	Belgium
1	1303171035	New Delhi	India
2	207847520	Brasilia	Brazil

```
df[df['Population']>1200000000]
```

	Population	Capital	Country
1	1303171035	New Delhi	India

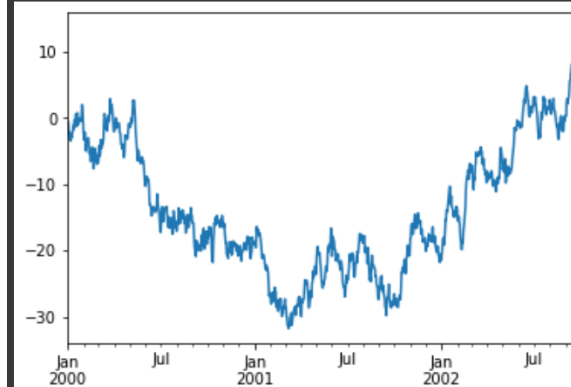
```
b = df.apply(lambda x: 'pass' if x.Math >= 80  
              and x.Science >70 else 'fail', axis=1)
```

```
print(b)
```

```
0    fail  
1    fail  
2    fail  
3    pass  
dtype: object
```

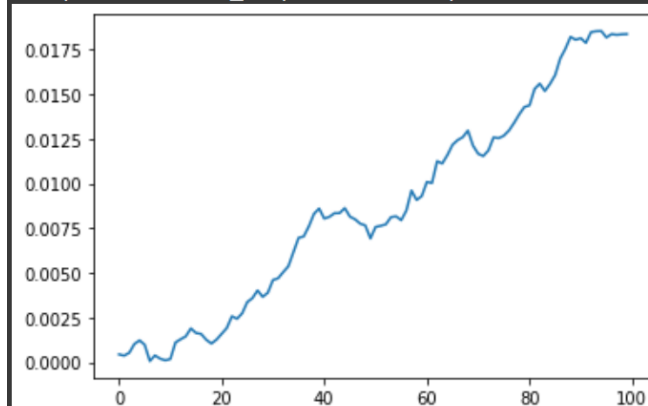
```
ts = pd.Series(np.random.randn(1000),  
              index=pd.date_range('1/1/2000', periods=1000))  
ts = ts.cumsum()  
ts.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f85432c3310>
```

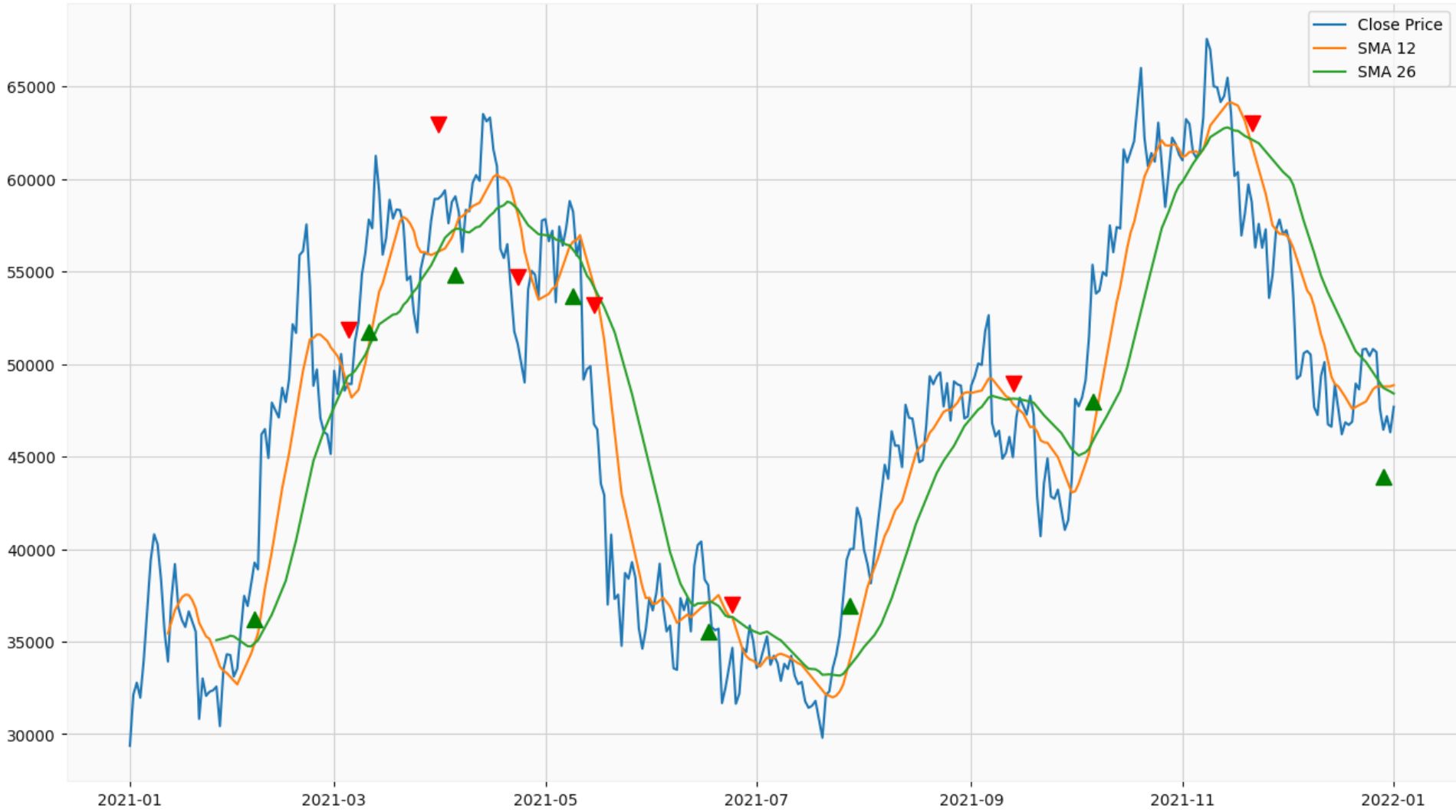


```
b = (a+1).cumprod()-1  
b.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f853843cfd0>
```



# Algo Trader Beginner





# Algo Trader Intermediate

**Algo Trader**  
**Intermediate**

- ✓ Backtest Overview
- ✓ การทำ Backtesting ผ่าน vectorbt
- ✓ ตั้ง Stop Loss และ Take Profit
- ✓ สร้าง Metric วัดผลตัวกลยุทธ์

**Strategy Backtesting**

Project - Parameter Tuning

4

1. Time Series Beginner

2. Python for Investing Beginner

3. Python for Technical Analysis

4. Strategy Backtesting

5. Money & Risk Management

6. Real Time Algorithmic Trading

7. Stock Screening Technical Analysis

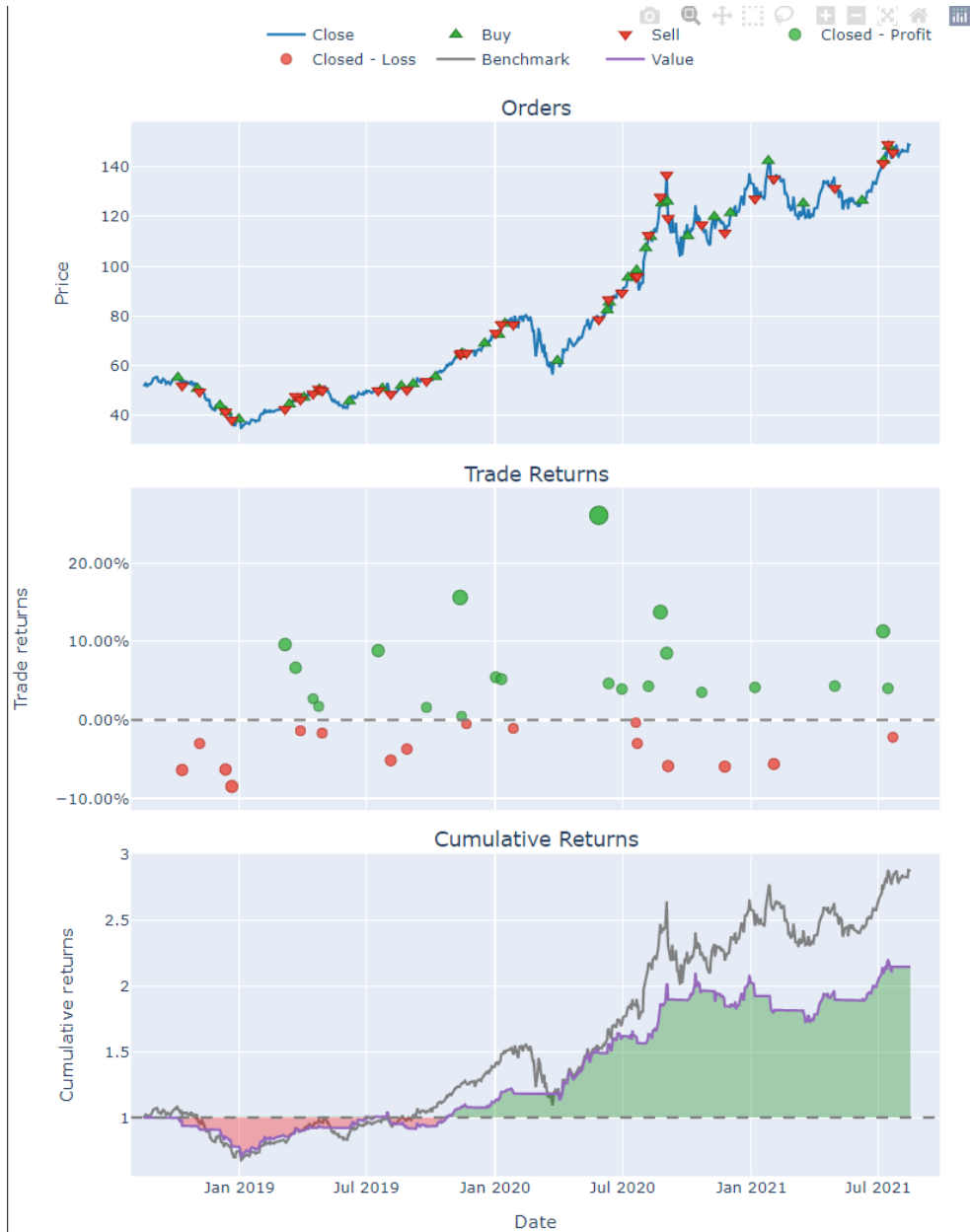
NODE 06

BLOCK 01





# Algo Trader Intermediate



**Algo Trader Advance**

1. Time Series Beginner  
2. Python for Investing Beginner  
3. Python for Technical Analysis  
4. Strategy Backtesting  
5. Money & Risk Management  
6. Real Time Algorithmic Trading  
7. Stock Screening Technical Analysis

**5**

**Money & Risk Management**

- ✓ Money Management ด้วยหลักการ Kelly Criterion
- ✓ การคำนวณ Metric ความเสี่ยงต่างๆ
- ✓ คำนวณ Value at Risk

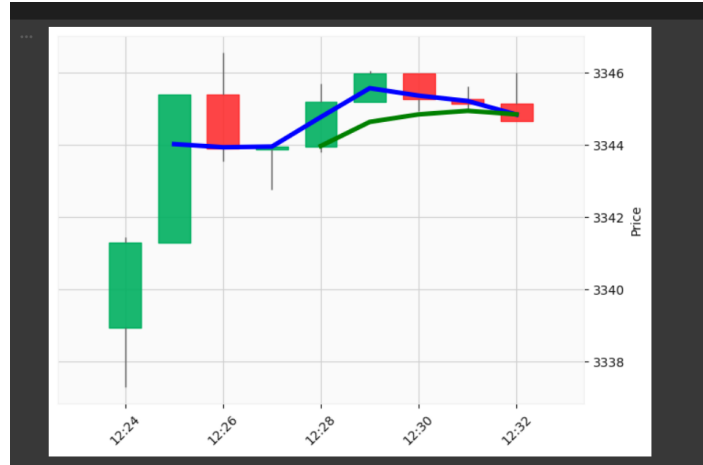
Project - Your First Algorithmic Trading System

**6**

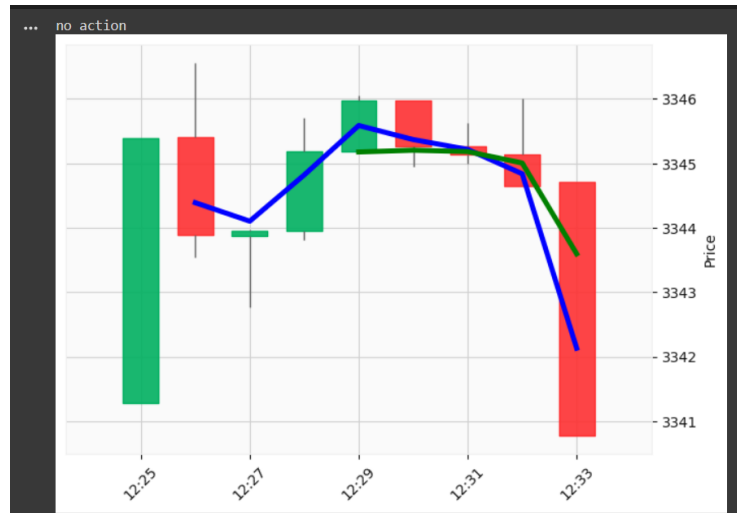
**Real Time Algorithmic Trading**

- ✓ เชื่อมต่อและสั่งคำสั่งซื้อขายผ่าน Bitkub, Binance
- ✓ สั่งคำสั่งซื้อขายหุ้นไทย, หุ้นต่างประเทศ
- ✓ สร้างบอทส่งคำสั่งซื้อขายจริงแบบ Real Time บนสัญญาณ Technical Indicator

# Algo Trader Advance



Date	Pair	Type	Side	Average	Price	Executed	Amount	Total
+ 03-28 19:33:00	ETH/BUSD	Market	Sell	3,344.71	Market	0.0060	0.0060	20.07



**Algo Trader Expert**

Project -  
Automate Screener Algo System

**Stock Screening  
Technical Analysis**

- ✓ การดึงข้อมูลหลายๆ Ticker พร้อมๆ กัน
- ✓ Price Gap Detection
- ✓ สแกนด้วย Volume
- ✓ สแกนย่อจาก 52 week high
- ✓ สแกนด้วย Technical Analysis และเทคนิคอื่นๆ

1. Time Series Beginner
2. Python for Investing Beginner
3. Python for Technical Analysis
4. Strategy Backtesting
5. Money & Risk Management
6. Real Time Algorithmic Trading
7. Stock Screening Technical Analysis

กำหนดการ Live และเริ่มคอร์สเรียน

1. Python for Time Series [23 พ.ค.]
2. Python for Investing Beginner [30 พ.ค.]
3. Technical Analysis [13 มิ.ย.]
4. Strategy Backtesting [27 มิ.ย.]
5. Money & Risk Management [18 ก.ค.]
6. Realtime Algorithmic Trading [1 ส.ค.]
7. Stock screening Technical Analysis [15 ส.ค.]

เริ่ม 1 ทุ่มตรง

## – Python Basic

0/8

### 2.2 Basic Python

[Preview](#)

### 2.3 Number

### 2.4 String

### 2.5 List

### 2.6 List Method

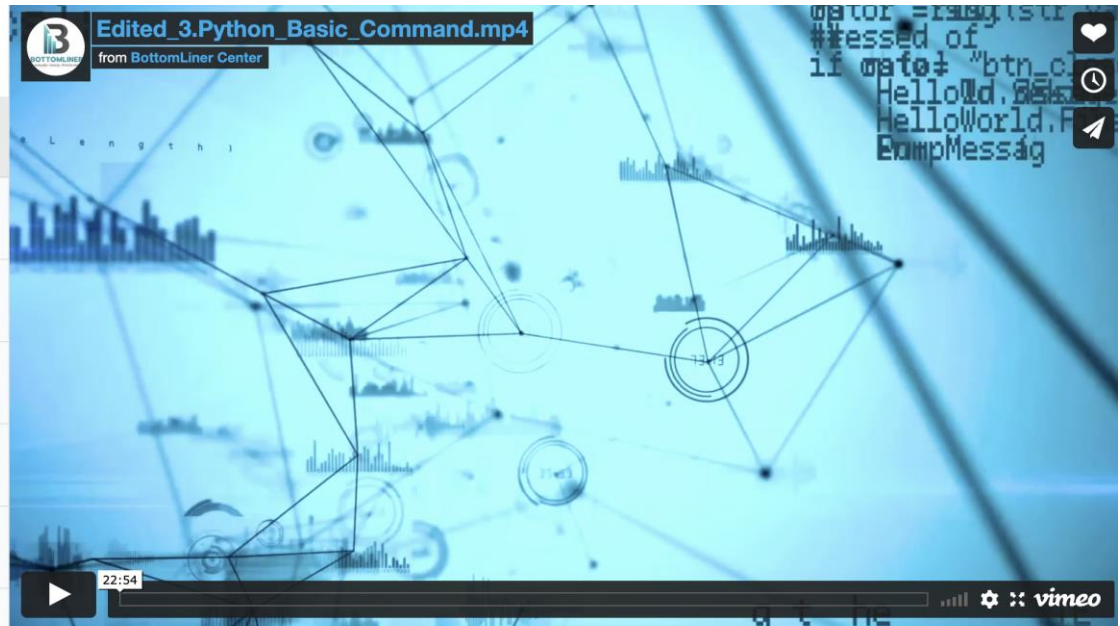
### 2.7 Dictionary

### 2.8 Basic Operation

## – Python Numpy and Pandas

0/8

### 3.1 Timeseries



## Basic Python

Number

◀ Prev

Next ▶



## Workshop – Create RSI strategy

1. งบหาสัญญาณซื้อขาย RSI 14 วัน ของหุ้น NVDA โดยดึงข้อมูล Nvidia ในระดับ Daily

โดยคำนวณจากสูตร

$$RSI = 100 - \frac{100}{1 + RS}$$

Number

```

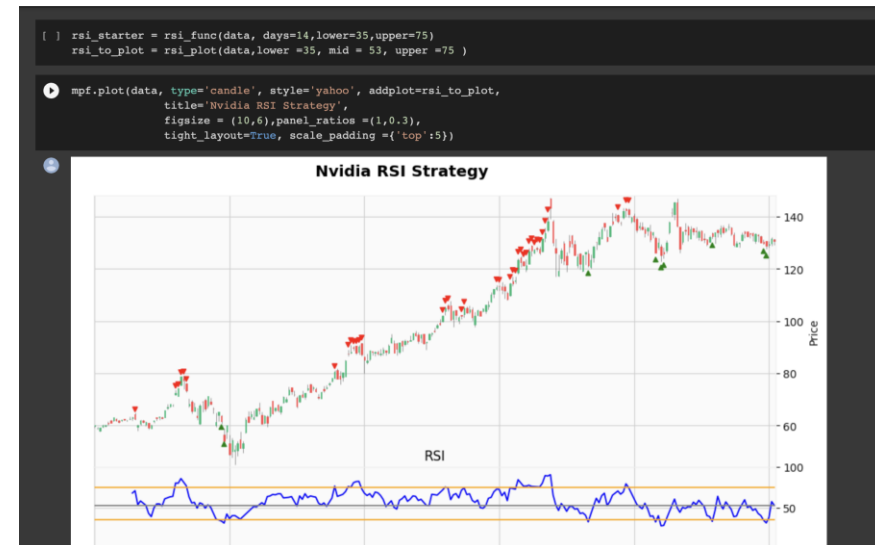
main.py
1 #ปรี้นเลข 7 ผ่านตัวแปร myint
2 myint =
3 print(myint)
4
5 #ปรี้นเลข 7 แบบทศนิยม
6 myfloat1 =
7 print(myfloat1)
8
9 #ปรี้นเลข 7 แบบทศนิยมผ่านการให้ float()
10 myfloat2 =
11 print(myfloat2)

```

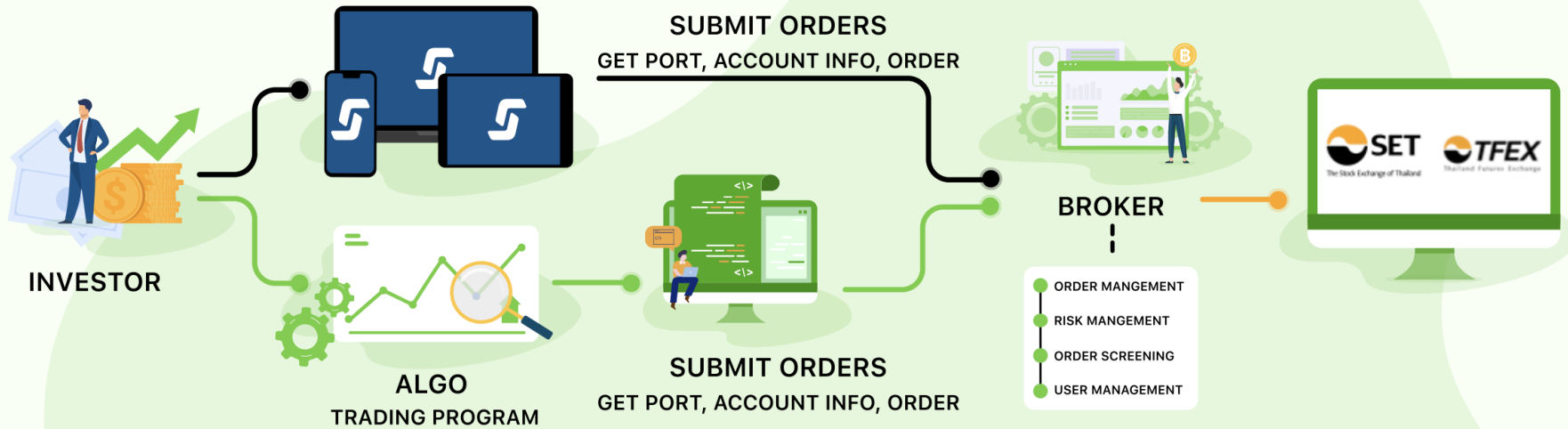
Relative Strength Index (RSI)							
	DATE	Close	Gain	Loss	Avg. Gain	Avg. Loss	RS RSI
1	2/11/2021	\$ 811.66					
2	2/12/2021	\$ 816.12	4.46	0.00			
3	2/16/2021	\$ 796.22	0.00	19.90			
4	2/17/2021	\$ 798.15	1.93	0.00			
5	2/18/2021	\$ 787.38	0.00	10.77			
6	2/19/2021	\$ 781.30	0.00	6.08			
7	2/22/2021	\$ 714.50	0.00	66.80			
8	2/23/2021	\$ 698.84	0.00	15.66			
9	2/24/2021	\$ 742.02	43.18	0.00			
10	2/25/2021	\$ 682.22	0.00	59.80			
11	2/26/2021	\$ 675.50	0.00	6.72			
12	3/1/2021	\$ 718.43	42.93	0.00			
13	3/2/2021	\$ 686.44	0.00	31.99			
14	3/3/2021	\$ 653.20	0.00	33.24			
14	3/4/2021	\$ 621.44	0.00	31.76	6.61	20.19	25

Annotations:

- Avg. Gain on Day 14 = A Simple Average Gain of 14 periods
- Avg. Loss on Day 14 = A Simple Average Loss of 14 periods
- Avg. Loss from Day 15 = (Prev Day Avg Loss \* 13) + Current Day loss



# Robot Trading



## Create API

Creating an API private key provides access to markets and real-time trading services on Binance via a third-party site or application.

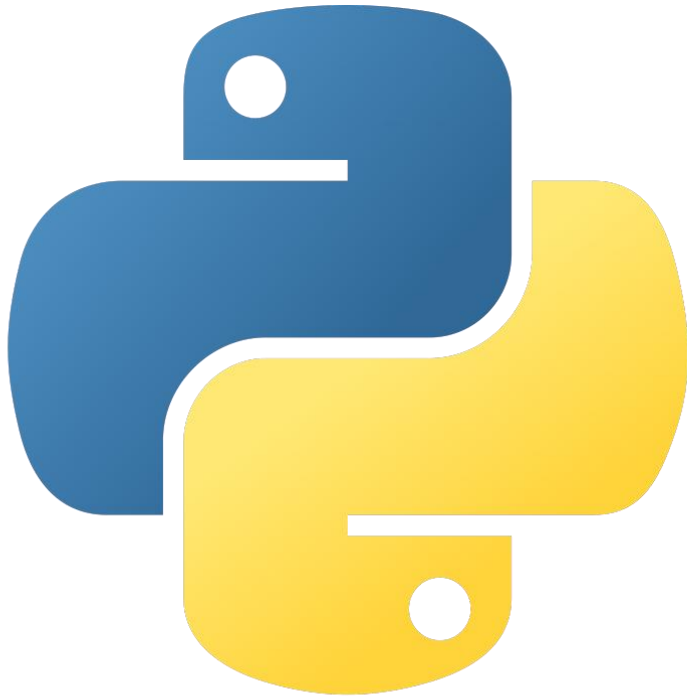
Label API key to proceed

Create API

[Create Tax Report API](#)

[View API documentation](#)



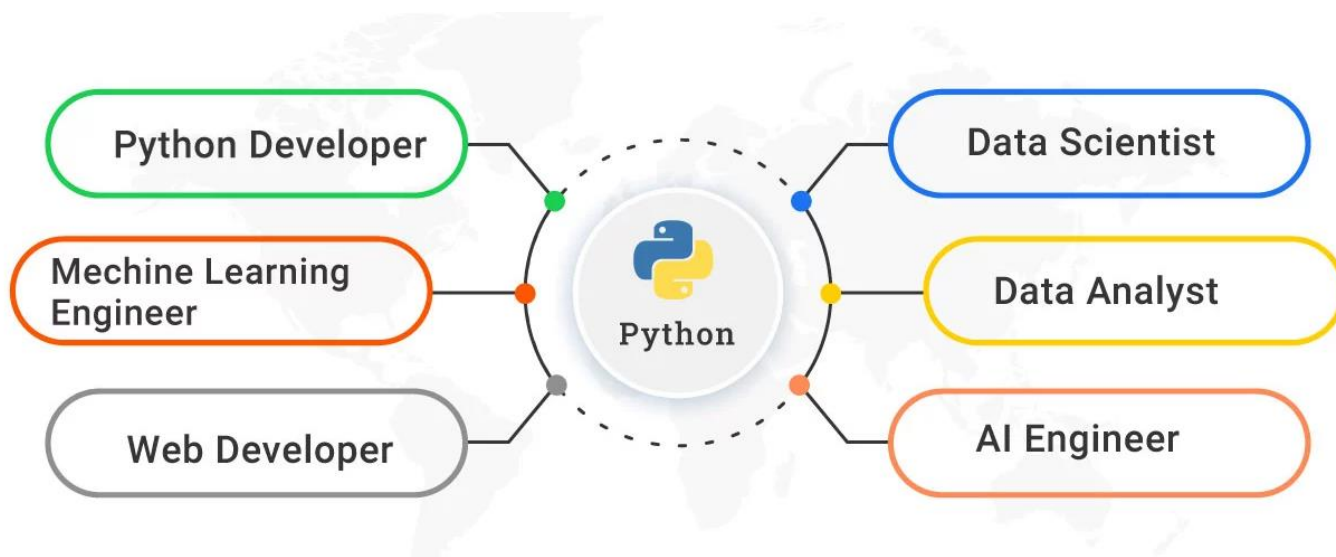


WHAT IS

PYTHON

# Advantages

- FREE, Easy to learn and use
- More productive as codes can be written in fewer lines (**Copy paste from Internet**)
- Provides excellent **library** support and has a large developer community



# Syntax is too easy

**C**

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

**JAVA**

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Prints the string to the
        console.
    }
}
```

**C++**

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
    return 0;
}
```

**Python**

Print("Hello World!")

Print(1+1)

>> Hello World!

>> 2

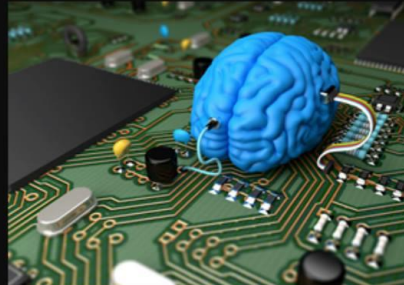


# Deep Learning



Set Stroke

What society thinks I do



What my friends think I do



What other computer scientists think I do



What mathematicians think I do

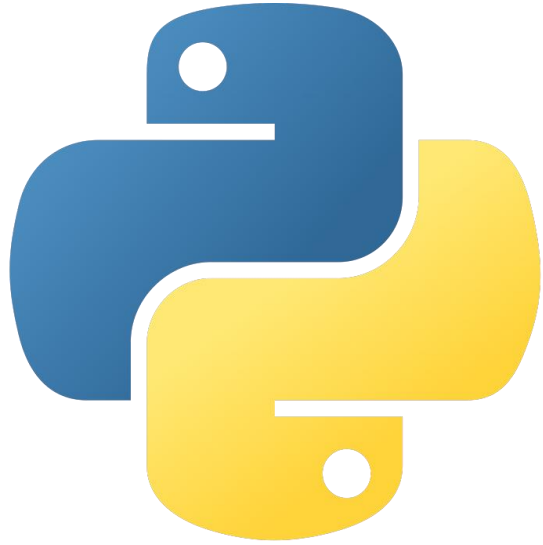


What I think I do

```
import keras
```

What I actually do





# WHAT PYTHON

Can

# Python Web Development

## Pros

Use in Prototyping

01

02

Easy to Use & Read

Enterprise Application Integration

03

04

Less-Limited Programming Approach

Server-Side Scripting

05

06

Asynchronous Coding

Easily accessible

07

08

Open-Source Perks

App Scripting & Software Testing

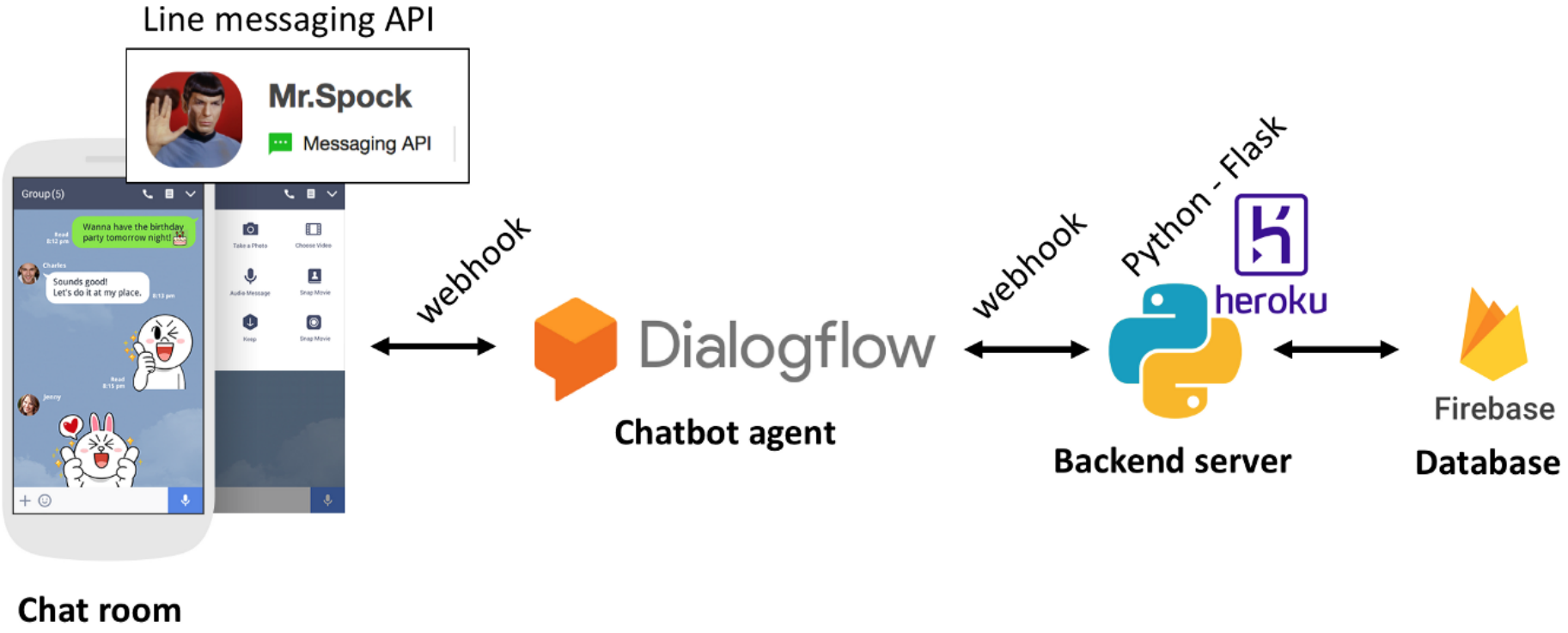
09

10

Use In ML & AI



# Chat Bot





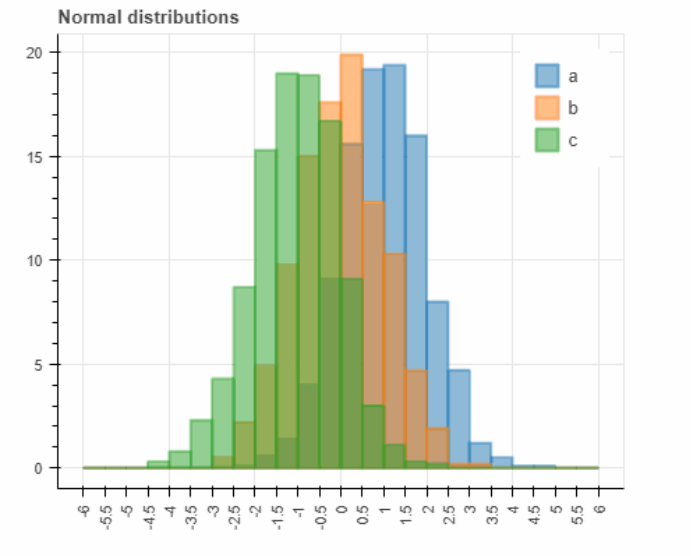
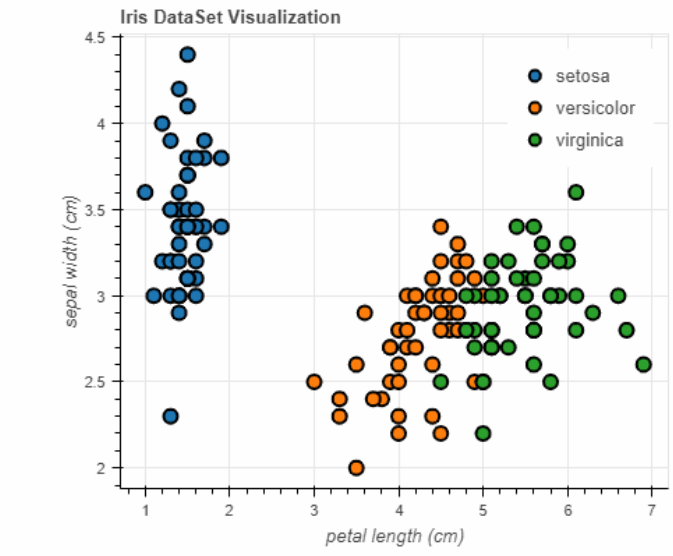
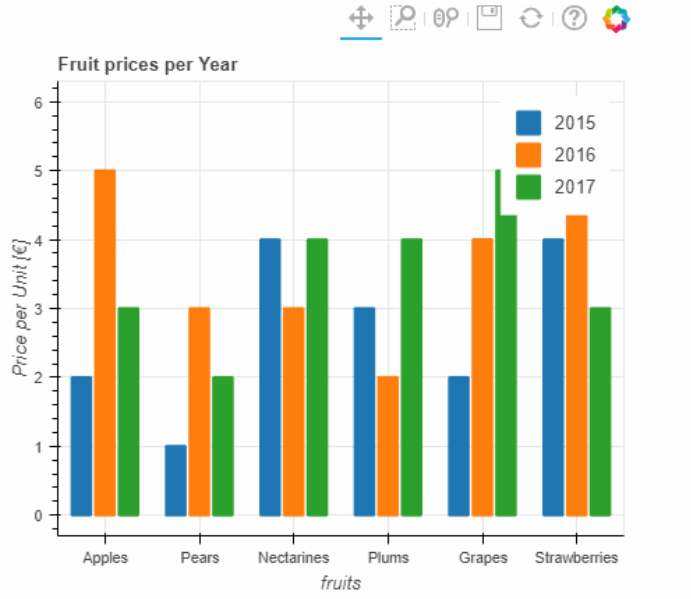
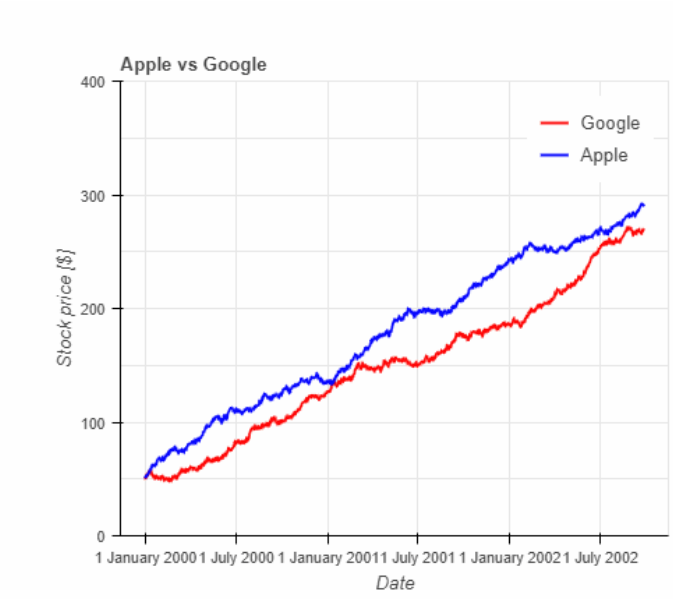
The screenshot shows an IDE window titled "untitled1" with a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help) and a toolbar. The left sidebar shows a project structure with "untitled1" selected. The main editor area displays a JSON file named "deployment.template.json" with the following content:

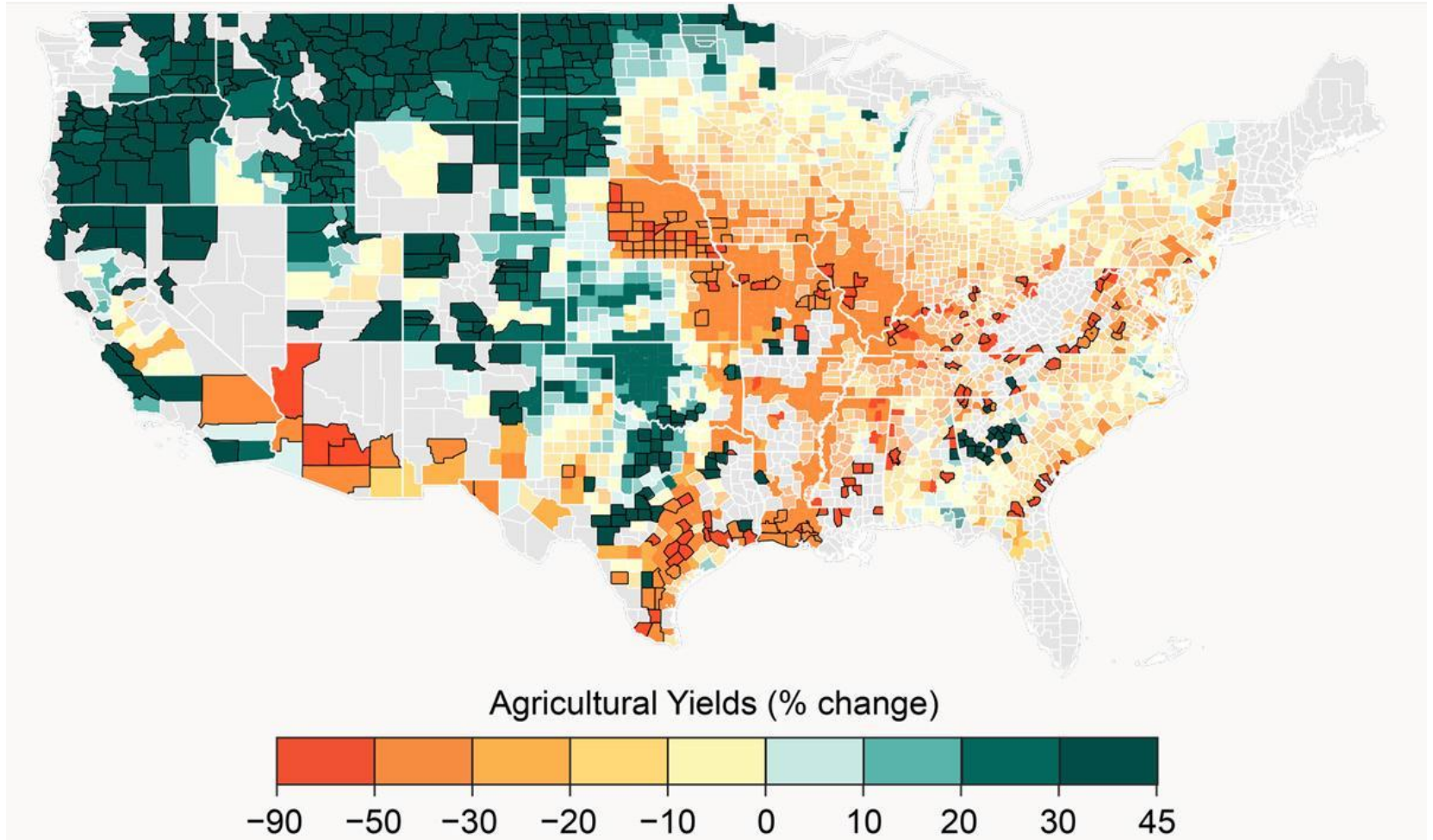
```
55 },
56 "$edgeHub": {
57   "properties.desired": {
58     "schemaVersion": "1.0",
59     "routes": {
60       "PythonModuleToIoTHub": "FROM /messages/modules/Py
61       "sensorToPythonModule": "FROM /messages/modules/te
62     },
63     "storeAndForwardConfiguration": {
64       "timeToLiveSecs": 7200
65     }
66   }
67 },
68 "PythonModule": {
```

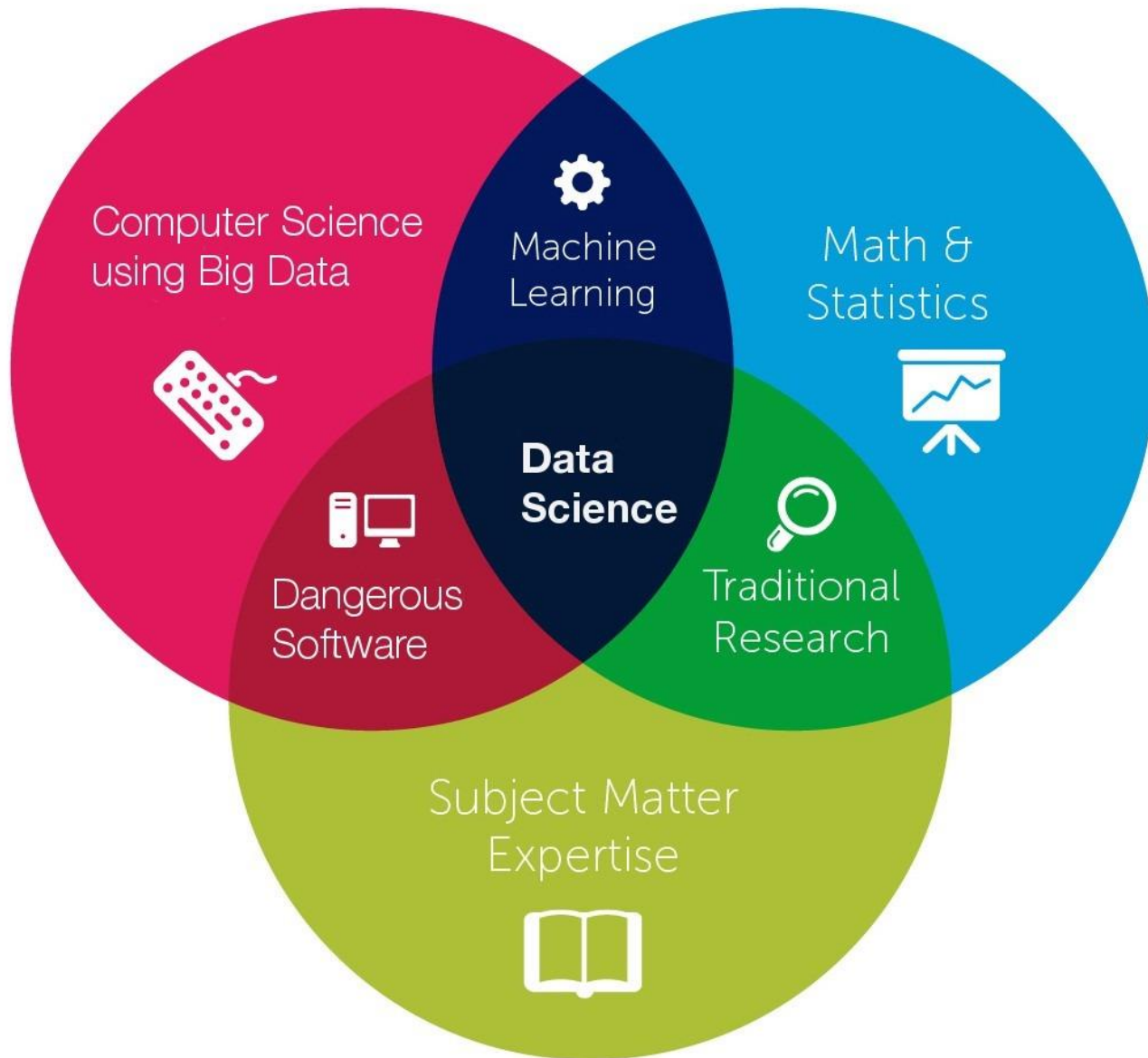


# AI & Machine Learning









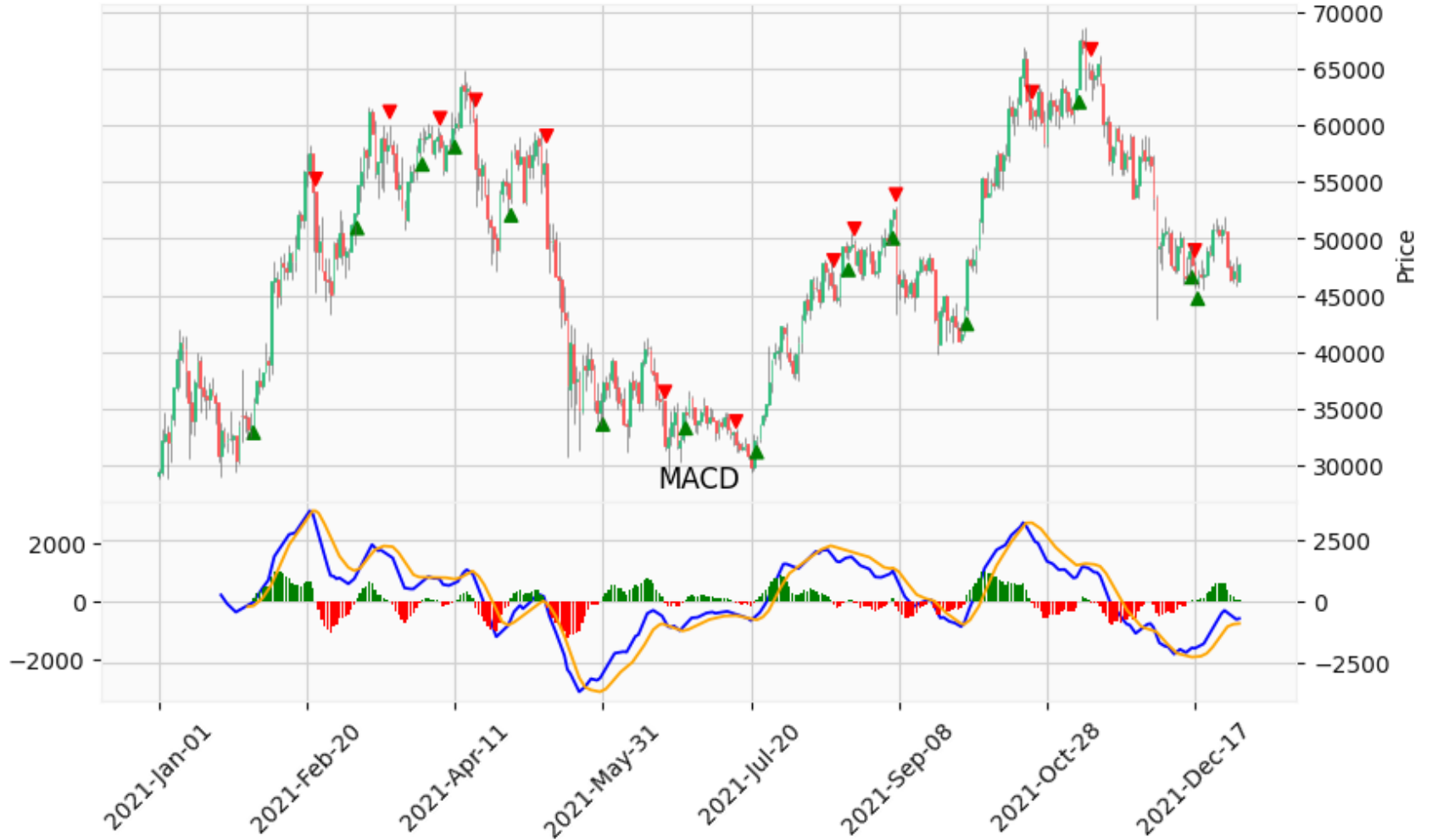




PYTHON IN THE

Investing World

### Bitcoin MACD



## เช็คใน binance

Date	Pair	Type	Side	Average	Price	Executed	Amount	Total
+ 03-28 19:33:00	ETH/BUSD	Market	Sell	3,344.71	Market	0.0060	0.0060	20.07

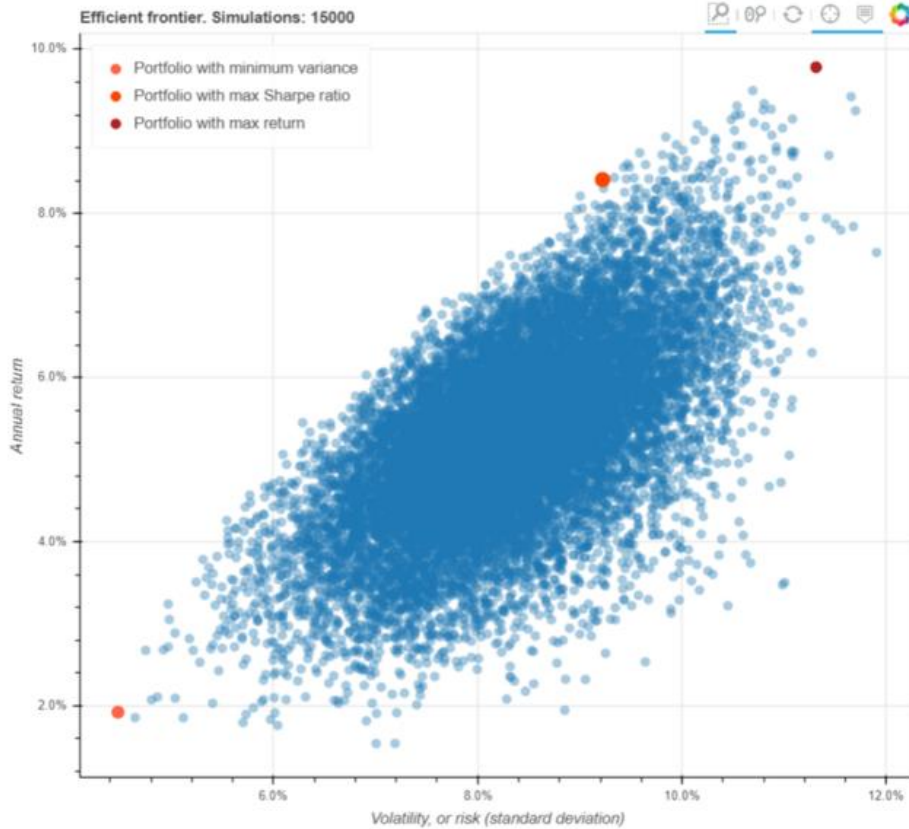
## แท่งถัดไป

... no action





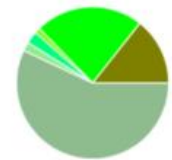
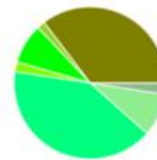
# Portfolio Management



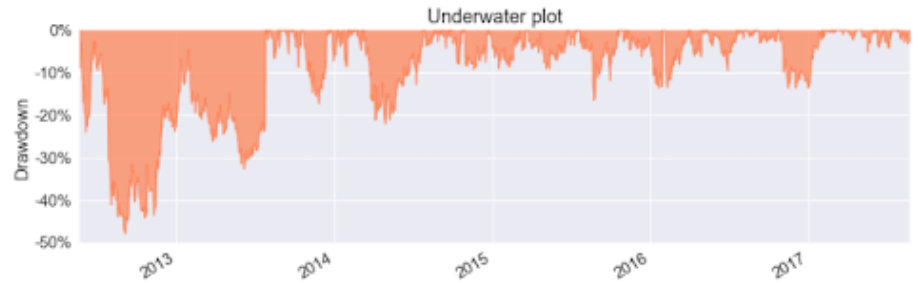
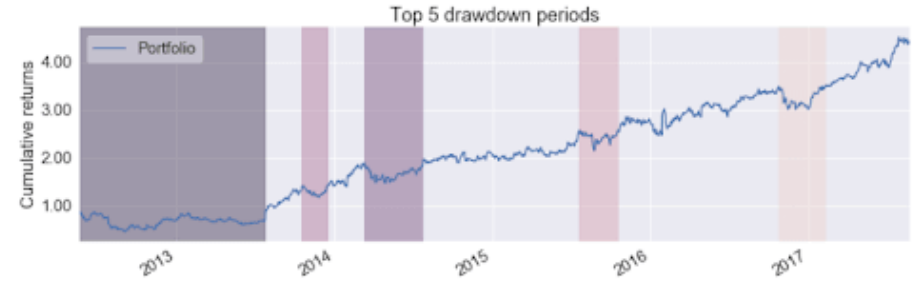
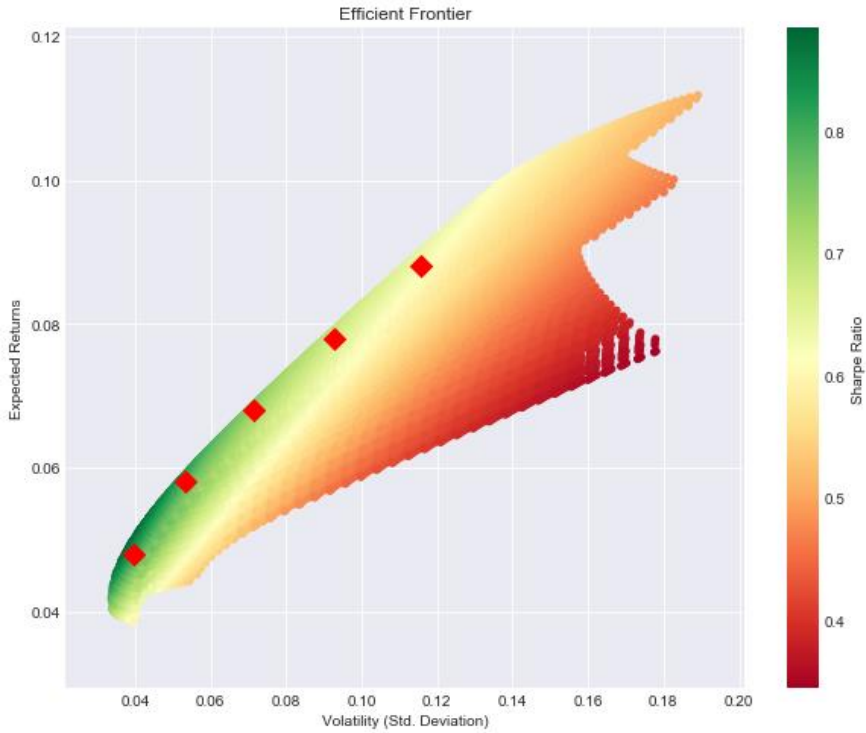
Portfolio with minimum variance

Portfolio with max Sharpe ratio

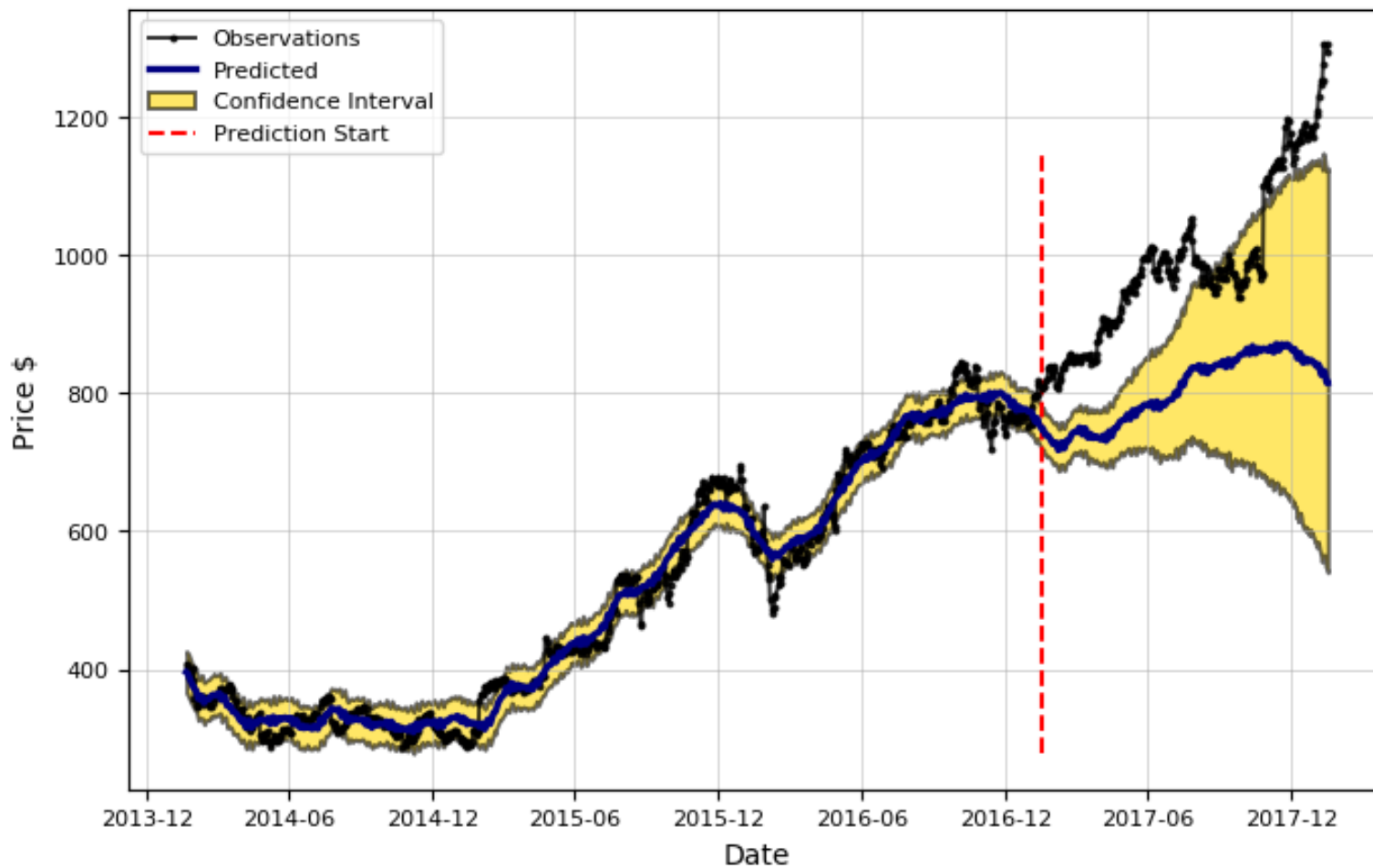
Portfolio with max return



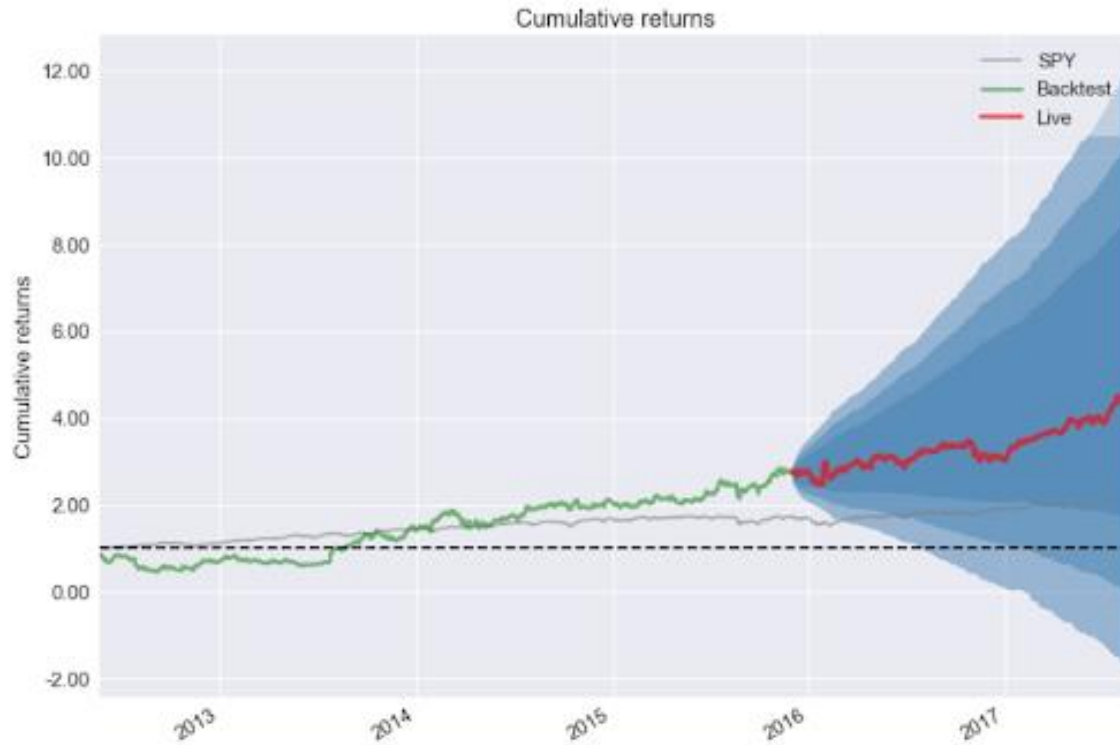
# Portfolio Management



AMZN Model Evaluation from 2017-01-18 to 2018-01-18.



# Price Prediction



## How Twitter Feels About the 2016 Election Candidates

