Deep Learning
N-BEATS Model

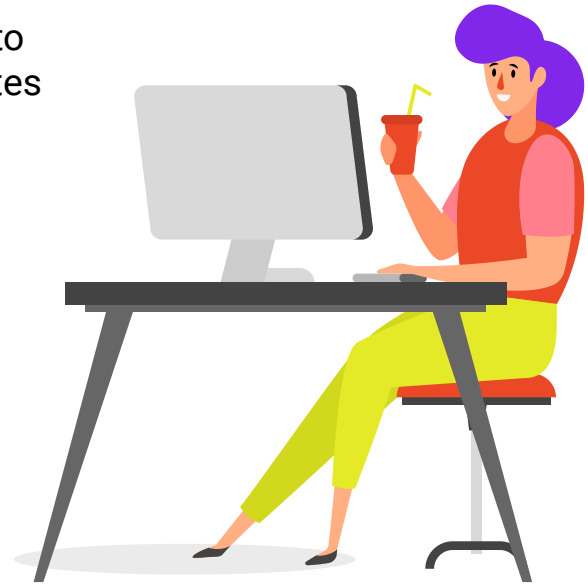
Use Case - Stock Price Prediction

Univariate Time Series Forecasting

By: Anthony, Benny, Wing
10 Aug 2022

Objective

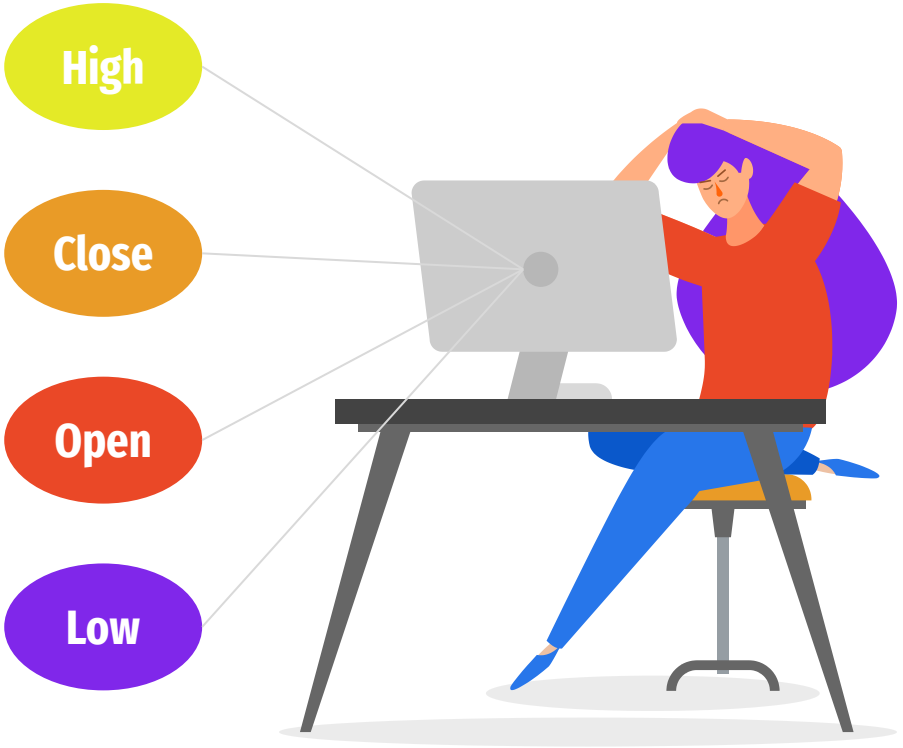
Use the past closing prices of SPY500 (in the 5-min interval) to predict **NEXT** closing price after 5, 10, 15, 20, 25, 30, 35 minutes (7 intervals).



Candlestick

Each of the candlestick represents a single 5-min interval. It tells how the stock price changes within a certain 5-min period.

Some basics about our stock dataframe



https://www.investopedia.com/terms/c/candlestick.asp

Library

01 **Tensorflow**

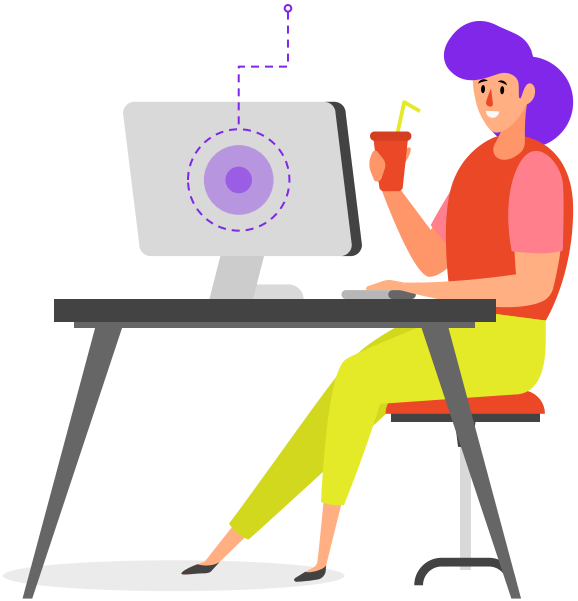
02 **train_test_split**

from sklearn.model_selection

03 **Pandas**

04 **Keras**

from tensorflow



05 **Numpy**

06 **layers**

from tensorflow.keras

07 **Matplotlib.pyplot**

08 **plot_model**

from tensorflow.keras.utils

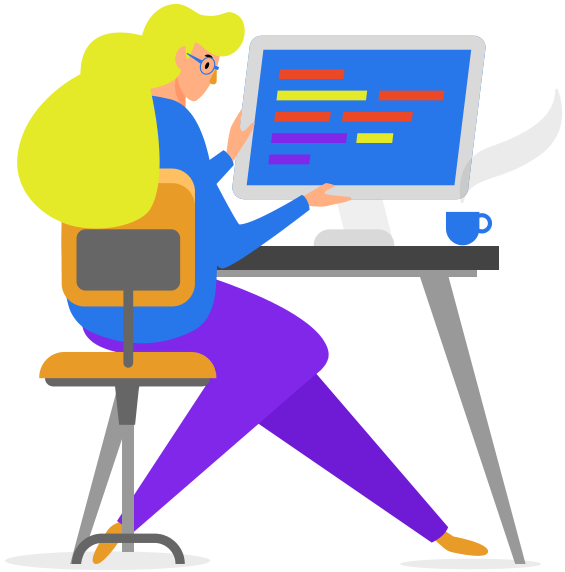
09 **Pickle**

To save the prediction output

10 **Sequential, load_model**

from tensorflow.keras.models

Steps in this end-to-end project



01 **Data Acquisition & Cleaning**

02 **Train Test Split**

80% Train & 20% Test

03 **Tensorflow**

Turn train & test arrays into tensor datasets

04 **N-BEATS**

Set parameters and fit the trained dataset

05 **N-BEATS**

Make and evaluate the predictions

06 **Streamlit**

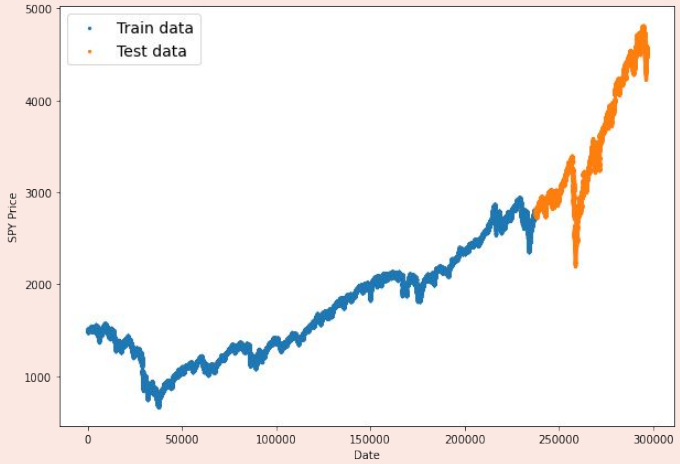
Gather the input/output function to a web-app

01 02 Data Acquisition & Cleaning & Train Test Split

Acquire SPY ETF dataset in 5-min interval (30/4/2007 - 4/2/2022)

- Dataset contains 5 columns: DateTime, Open, High, Low, Close
- Only DateTime and Close (close price) are kept
- Get timestep array and price array in the same length

80% Train & 20% Test



Data processing

Data from SPY 500 in 5-min interval (2007 Apr - 2022 Feb)					
Timestep	Datetime	Open	High	Low	Close
0	30/4/2007 9:25	*	*	*	*
1	30/4/2007 9:30	*	*	*	*
2	30/4/2007 9:35	*	*	*	*
...	...	...	...	...	...
296645	4/2/2022 16:05	*	*	*	*

296646 data points in total

	P: Price	P +i (i: one 5-min interval)						
	t0: the close price in timestep 0							
Timestep	P	P +1	P +2	P +3	P +4	P +5	P +6	P +7
0	t0	na	na	na	na	na	na	na
1	t1	t0	na	na	na	na	na	na
2	t2	t1	t0	na	na	na	na	na
3	t3	t2	t1	t0	na	na	na	na
4	t4	t3	t2	t1	t0	na	na	na
5	t5	t4	t3	t2	t1	t0	na	na
6	t6	t5	t4	t3	t2	t1	t0	na
7	t7	t6	t5	t4	t3	t2	t1	t0
8	t8	t7	t6	t5	t4	t3	t2	t1
...	...	...	...	...	...	...	...	...
Timestep	P	P +1	P +2	P +3	P +4	P +5	P +6	P +7
296646	Pred0					Known data points		(t296645 ... t296639)
296647	Pred1	Pred0						(t296645 ... t296640)
296648	Pred2	Pred1	Pred0					(t296645 ... t296641)
296649	Pred3	Pred2	Pred1	Pred0				(t296645 ... t296642)
296650	Pred4	Pred3	Pred2	Pred1	Pred0			(t296645 ... t296643)

N-BEATS Model introduction:

A deep learning method to solve the univariate times series point forecasting problem.

The architecture does not rely on time-series-specific feature engineering or input scaling but purely timesteps.

In Timestep 296646, Pred0 is the predicted close price after 5 mins.

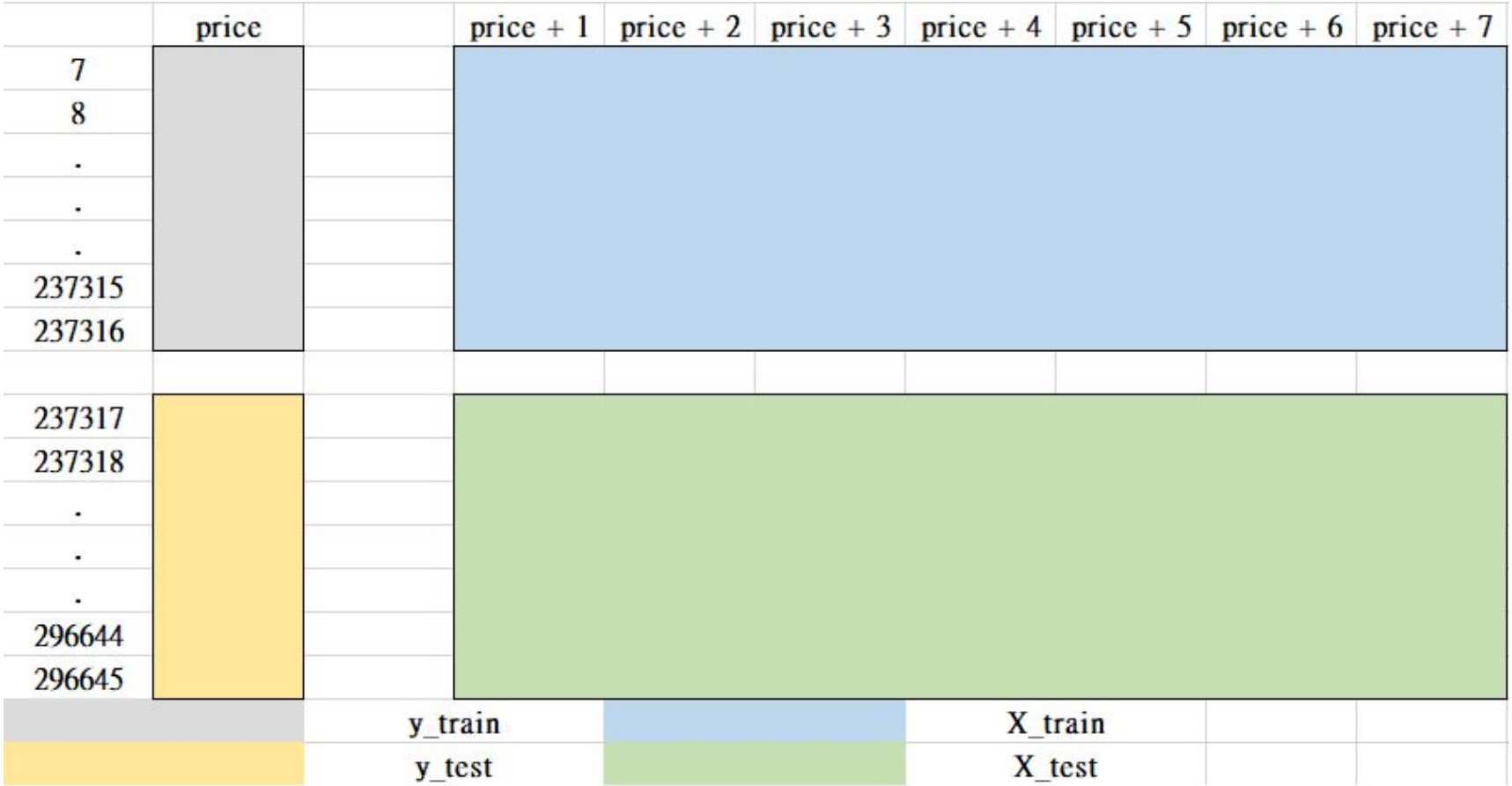
In Timestep 296647, Pred1 is the predicted close price after 10 mins.

In Timestep 296648, Pred2 is the predicted close price after 15 mins.

In Timestep 296649, Pred3 is the predicted close price after 20 mins.

In Timestep 296650, Pred4 is the predicted close price after 25 mins.

Train Test Split



In reality, however, after 5 mins, we will have the latest real close price (5-min interval) on hand and have it input to the current P +i dataframe, so the model could learn from the latest real data and predict further.

Timestep	P	P +1	P +2	P +3	P +4	P +5	P +6	P +7
296646	Pred0			Known data points		**(t296645 ... t296639)**		
296647	Pred1	Pred0				**(t296645 ... t296640)**		
296648	Pred2	Pred1	Pred0			**(t296645 ... t296641)**		

Timestep	P		P +1	P +2	P +3	P +4	P +5	P +6	P +7
296646	Real0				Known data points		**(t296645 ... t296639)**		
296647	Pred1		Real0				**(t296645 ... t296639)**		
296648	Pred2		Pred1	Real0			**(t296645 ... t296639)**		
296649	Pred3		Pred2	Pred1	Real0		**(t296645 ... t296639)**		
296650	Pred4		Pred3	Pred2	Pred1	Real0	**(t296645 ... t296639)**		

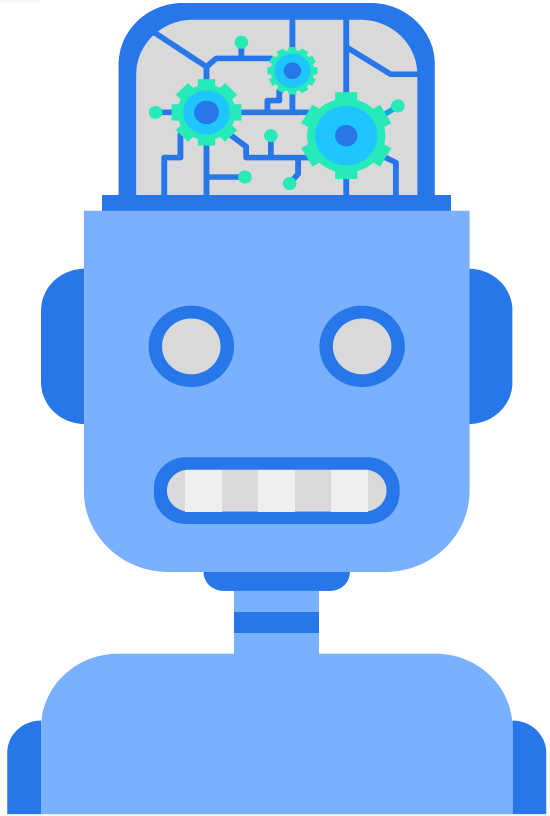
```
# Set up dummy NBeatsBlock layer to represent inputs and outputs
dummy_nbeats_block_layer = NBeatsBlock(input_size=WINDOW_SIZE,
                                       theta_size=WINDOW_SIZE+HORIZON, # backcast + forecast
                                       horizon=HORIZON,
                                       n_neurons=128,
                                       n_layers=4)
```

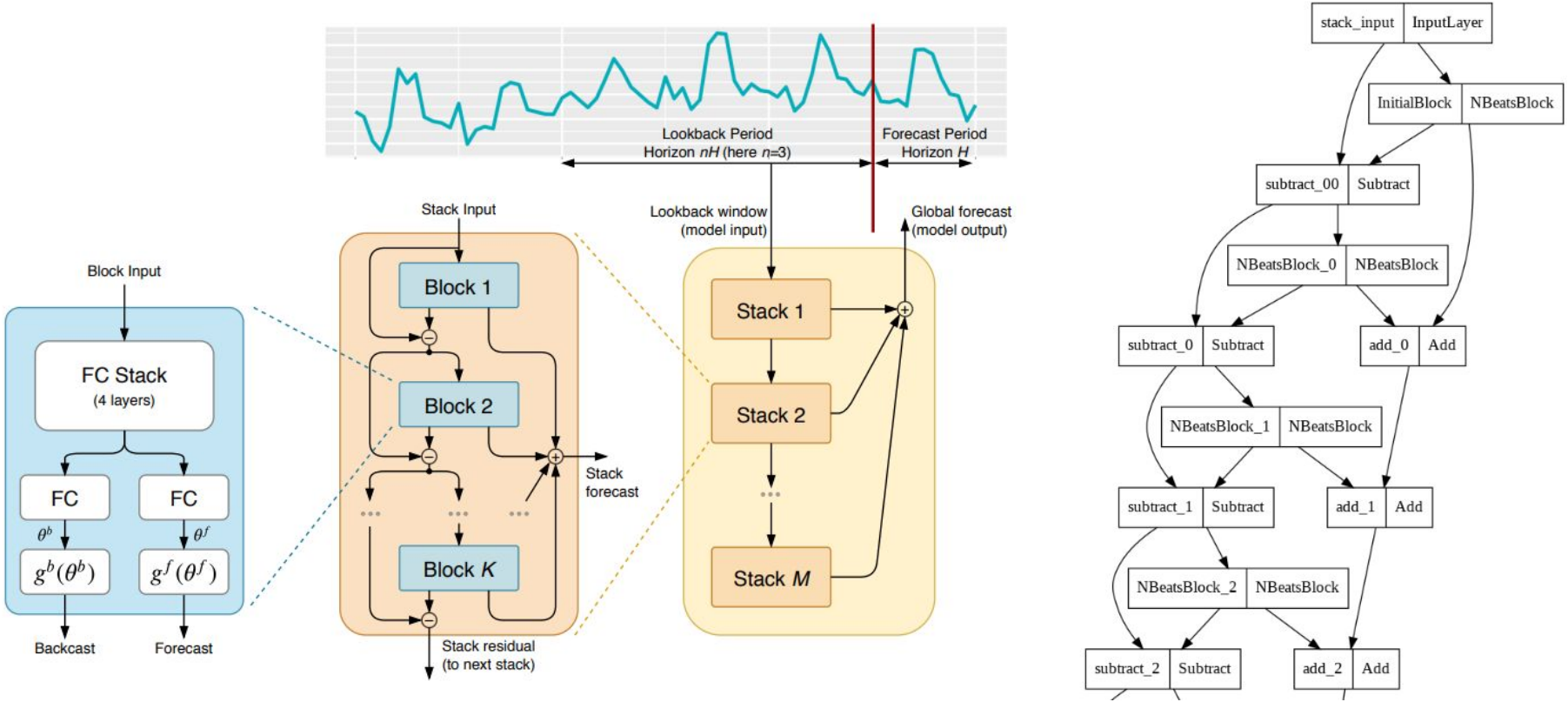
```
# Values from N-BEATS paper Figure 1 and Table 18/Appendix D

BATCH_SIZE = 1024 # taken from Appendix D in N-BEATS paper
N_EPOCHS = 5000 # called "Iterations" in Table 18
N_NEURONS = 512 # called "Width" in Table 18
N_LAYERS = 4
N_STACKS = 30

INPUT_SIZE = WINDOW_SIZE * HORIZON # called "Lookback" in Table 18
THETA_SIZE = INPUT_SIZE + HORIZON
```

Table 18: Settings of hyperparameters across subsets of M4, M3, TOURISM datasets.

	M4						M3				TOURISM		
	Yly	Qly	Mly	Wly	Dly	Hly	Yly	Qly	Mly	Other	Yly	Qly	Mly
Parameter	N-BEATS-G												
L_H	1.5	1.5	1.5	10	10	10	20	20	20	10	5	10	20
Iterations	15K	15K	15K	5K	5K	5K	20	250	10K	250	30	100	100
Losses	sMAPE/MAPE/MASE						sMAPE/MAPE/MASE				MAPE		
Width	512												
Blocks	1												
Block-layers	4												
Stacks	30												
Sharing	NO												
Lookback period	$2H, 3H, 4H, 5H, 6H, 7H$												
Batch	1024												

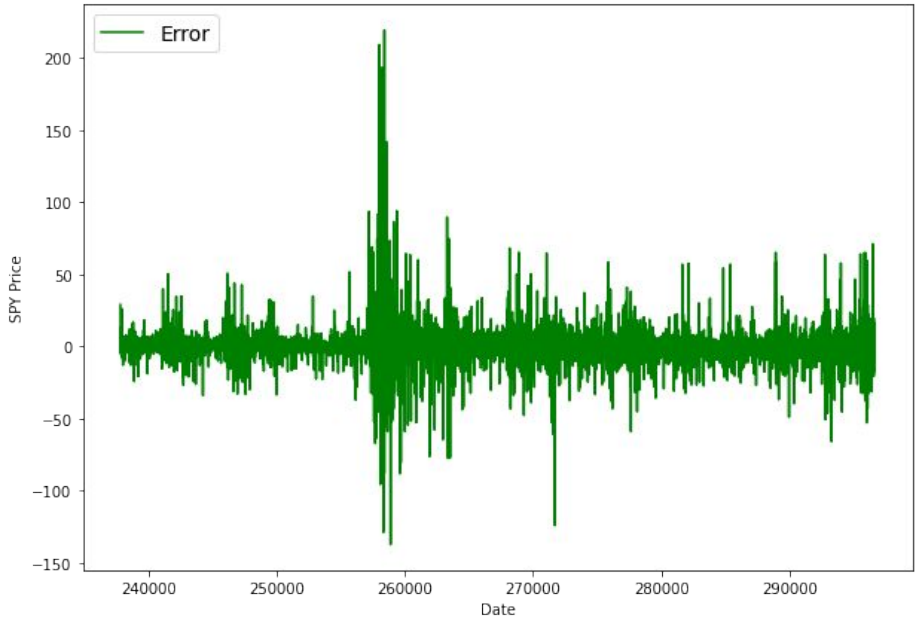
N-BEAT model



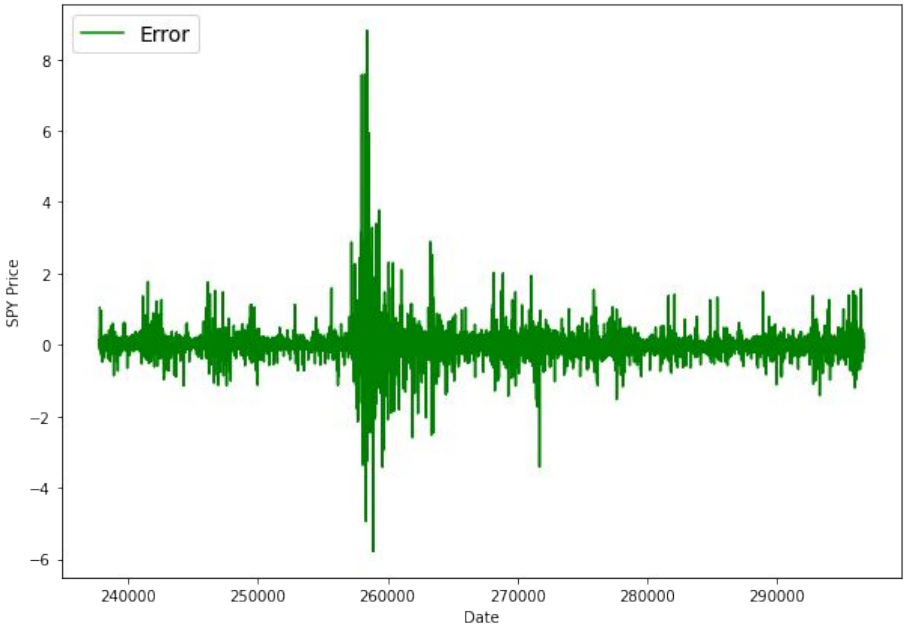

Lookback Period
Horizon nH (here n=3)
Forecast Period
Horizon H
Block Input
FC Stack
(4 layers)
FC
FC
θ^b
θ^f
$g^b(\theta^b)$
$g^f(\theta^f)$
Backcast
Forecast
Stack Input
Block 1
Block 2
Block K
Stack forecast
Stack residual
(to next stack)
Lookback window
(model input)
Global forecast
(model output)
Stack 1
Stack 2
Stack M
stack_input
InputLayer
InitialBlock
NBeatsBlock
subtract_00
Subtract
NBeatsBlock_0
NBeatsBlock
subtract_0
Subtract
add_0
Add
NBeatsBlock_1
NBeatsBlock
subtract_1
Subtract
add_1
Add
NBeatsBlock_2
NBeatsBlock
subtract_2
Subtract
add_2
Add

Model performance

Error in price



Error in percentage



Next steps

01 **Try different data granularity**

Try 1-hour, 30-min, 3-min interval, etc. to explore the doors to better accuracy.

02 **Try different stocks (current: SPY)**

The price variation of other stocks may be more predictable to the model.

03 **Adjust the Window Size (Current: up to P+7)**

Large Window Size may improve the model performance at a larger time cost.

04 **Leverage cloud computing and storing service**

E.g. IaaS VM with GPU to improve the training speed and enlarge the data size storing availability.

05 **Try different multivariate time-series model**

N-BEATS mainly solves univariate TS problem, but stock close price is probably considered as influenced by multiple variates.