```
import numpy as np
import pandas as pd
from neuralforecast.models import NBEATS, NHITS, NBEATSx
from neuralforecast import NeuralForecast
import yfinance as yf
```

```
data =  yf.download("BTC-USD", start = "2014-09-17", end = "2024-04-21")
data.reset_index(inplace = True)
```

```
[*********************100%%**********************]  1 of 1 completed
```

```
import pandas as pd
data = pd.read_csv("/content/BTC-USD.csv")
data['ds'] = pd.to_datetime(data['Date'])
data['unique_id'] = 1
data['y'] = data['Adj Close']
data.drop(columns = ['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'],axis = 1, inplace = True)
data.tail(),data.columns
```

```
(             ds  unique_id             y
 3498 2024-04-15          1  63426.210938
 3499 2024-04-16          1  63811.863281
 3500 2024-04-17          1  61276.691406
 3501 2024-04-18          1  63512.753906
 3502 2024-04-19          1  62001.316406,
 Index(['ds', 'unique_id', 'y'], dtype='object'))
```

```
data
```

	ds	unique_id	y
0	2014-09-17	1	457.334015
1	2014-09-18	1	424.440002
2	2014-09-19	1	394.795990
3	2014-09-20	1	408.903992
4	2014-09-21	1	398.821014
...	...	...	...
3498	2024-04-15	1	63426.210938
3499	2024-04-16	1	63811.863281
3500	2024-04-17	1	61276.691406
3501	2024-04-18	1	63512.753906
3502	2024-04-19	1	62001.316406

3503 rows × 3 columns

Next steps: Generate code with `data` View recommended plots

```
from google.colab import output
output.enable_custom_widget_manager()
```

```
horizon =15
models =[NBEATS(h = horizon, input_size = 30),
        NHITS(h = horizon, input_size = 30),
        NBEATSx(h = horizon, input_size = 30)]

nf = NeuralForecast(models = models, freq = 'D')
nf.fit(df=data)
```

```
/usr/local/lib/python3.10/dist-packages/pytorch_lightning/utilities/parsing.py:199: UserWarning:

Attribute 'loss' is an instance of `nn.Module` and is already saved during checkpointing. It is recommended to ignore them using `self.s

INFO:lightning_fabric.utilities.seed:Seed set to 1
INFO:lightning_fabric.utilities.seed:Seed set to 1
INFO:lightning_fabric.utilities.seed:Seed set to 1
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type          | Params
-----------------------------------------------
0 | loss         | MAE           | 0
1 | padder_train | ConstantPad1d | 0
2 | scaler       | TemporalNorm  | 0
3 | blocks       | ModuleList    | 2.5 M
-----------------------------------------------
2.5 M     Trainable params
1.4 K     Non-trainable params
2.5 M     Total params
9.871     Total estimated model params size (MB)
/usr/local/lib/python3.10/dist-packages/pytorch_lightning/loops/fit_loop.py:298: PossibleUserWarning:

The number of training batches (1) is smaller than the logging interval Trainer(log_every_n_steps=50). Set a lower value for log_every_r

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_steps=1000` reached.
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type          | Params
-----------------------------------------------
0 | loss         | MAE           | 0
1 | padder_train | ConstantPad1d | 0
2 | scaler       | TemporalNorm  | 0
3 | blocks       | ModuleList    | 2.5 M
-----------------------------------------------
2.5 M     Trainable params
0         Non-trainable params
```

```
0         Non-trainable params
2.5 M     Total params
9.821     Total estimated model params size (MB)
INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_steps=1000` reached.
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type          | Params
-----------------------------------------------
0 | loss         | MAE           | 0
1 | padder_train | ConstantPad1d | 0
2 | scaler       | TemporalNorm  | 0
3 | blocks       | ModuleList    | 2.5 M
-----------------------------------------------
2.5 M     Trainable params
1.4 K     Non-trainable params
2.5 M     Total params
9.871     Total estimated model params size (MB)
INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_steps=1000` reached.
```

```
y_hat_df = nf.predict()
y_hat_df
```

```
/usr/local/lib/python3.10/dist-packages/neuralforecast/tsdataset.py:92: UserWarning:

To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(

INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
/usr/local/lib/python3.10/dist-packages/neuralforecast/core.py:184: FutureWarning:

In a future version the predictions will have the id as a column. You can set the `NIXTLA_ID_AS_COL` environment variable to adopt the r
```

	ds	NBEATS	NHITS	NBEATSx
unique_id				
1	2024-04-20	61799.042969	61866.523438	61799.042969
1	2024-04-21	63017.273438	63113.195312	63017.273438
1	2024-04-22	63995.023438	63291.687500	63995.023438
1	2024-04-23	64961.328125	64240.605469	64961.328125
1	2024-04-24	65107.343750	64323.285156	65107.343750
1	2024-04-25	65884.617188	64393.316406	65884.617188
1	2024-04-26	65866.031250	64610.238281	65866.031250
1	2024-04-27	65456.023438	64295.070312	65456.023438
1	2024-04-28	65768.835938	63961.535156	65768.835938

1	2024-04-29	65719.640625	63561.492188	65719.640625
1	2024-04-30	65359.121094	63258.097656	65359.121094
1	2024-05-01	65282.484375	62775.617188	65282.484375
1	2024-05-02	65416.191406	62559.859375	65416.191406
1	2024-05-03	66108.148438	62767.273438	66108.148438
1	2024-05-04	66542.523438	63211.625000	66542.523438

Next steps: Generate code with `y_hat_df` View recommended plots

```
import yfinance as yf
Real_Data =  yf.download("BTC-USD", start = "2024-04-20", end = "2024-05-05")
Real_Data.reset_index(inplace = True)
```

```
[*********************100%%**********************]  1 of 1 completed
```

```
from plotly.subplots import make_subplots
import plotly.graph_objs as go
fig = make_subplots(rows=1, cols=1, shared_xaxes=True, vertical_spacing=0.1)
data_df = data.iloc[-15:]
fig.add_trace(go.Scatter(x=data_df['ds'], y=data_df['y'], mode='lines', name='data'),row=1, col=1)

fig.add_trace(go.Scatter(x=Real_Data['Date'], y=Real_Data['Adj Close'], mode='lines', name="Real_Data"),row=1, col=1)
fig.add_trace(go.Scatter(x=y_hat_df['ds'], y=y_hat_df['NBEATS'], mode='lines', name='NBEATS_Prediction'),row=1, col=1)
fig.add_trace(go.Scatter(x=y_hat_df['ds'], y=y_hat_df['NHITS'], mode='lines', name='NHITS_Prediction'),row=1, col=1)
fig.add_trace(go.Scatter(x=y_hat_df['ds'], y=y_hat_df['NBEATSx'], mode='lines', name='NBEATSx_Prediction'),row=1, col=1)

fig.update_layout(title = 'Next 15 days Prediction Using NBEATS, NHITS and NBEATSx Algorithms',xaxis=dict(rangeslider=dict(visible=True)), h
fig.update_xaxes(
    rangeselector=dict(
        buttons=list([
            dict(count=1, label="1m", step="month", stepmode="backward"),
            dict(count=6, label="6m", step="month", stepmode="backward"),
            dict(count=1, label="YTD", step="year", stepmode="todate"),
            dict(count=1, label="1y", step="year", stepmode="backward"),
            dict(step="all")
        ])
    )
)
fig.show()
```


Next 15 days Prediction Using NBEATS, NHITS and NBEATSx Algorithms
1m
6m
YTD
1y
all
72k
70k
68k
66k
64k
62k
60k
58k
Apr 7
2024
Apr 10
Apr 13
Apr 16
Apr 19
Apr 22
Apr 25
Apr 28
May 1
May 4
data
Real_Data
NBEATS_Pre
NHITS_Predi
NBEATSx_Pr